

THE WAKE UP X PUBLISHING SYSTEM

BOOK 1

Break In

How to land your first role in IT and cybersecurity

From application to offer to your first ninety days

THE
WAKEUP X
SOLUTIONS

EFREN L. SALAZAR

Founder & Host, Wake Up X | IT Operations Manager & Solution Engineer
Government & Regulatory Partner, ISACA Sacramento | CompTIA Security+

wakeupxpodcast.com

Educational content — not professional, legal, financial, or career-placement advice.
Full disclaimer in DECODE, the foundation of this library.

Before You Begin

HOW TO USE THIS BOOK

This book has one job: to get you hired. Not to make you feel motivated, not to explain the industry in the abstract — to move you from applying into a role, with a system you can run again the next time you need it.

Where this book sits

Two books come first in the full Wake Up X system: the memoir I Just Have To Wake Up and the mindset book Comfort Is Expensive. This is one book in the ten-book library that follows them. Each one is short on purpose. You are not meant to read them all in order, front to back. Pick up the one that matches the problem in front of you right now.

- n **Book 0 — DECODE.** The foundation. The map of the whole path.
- n **Book 1 — Break In.** Landing your first role in IT or cybersecurity. **you are here**
- n **Book 2 — Build It.** Home labs and a portfolio that proves you can do the work.
- n **Book 3 — Think Like a Cybersecurity Professional.** Detection, triage, and incident response thinking.
- n **Book 4 — Business First.** Finance, sales, GRC, and why the business hired you.
- n **Book 5 — Lead Without a Title.** Influence, EQ, and mentoring before anyone promotes you.
- n **Book 6 — AI for IT & Cybersecurity.** Using AI as an actual working tool, not a novelty.
- n **Book 7 — Build Your Brand.** LinkedIn, content, and being findable.
- n **Book 8 — The Consultant Mindset.** Discovery, solution selling, and stakeholder management.
- n **Book 9 — Level Up.** Long-term growth, burnout, legacy, and mentoring others.

Book 0 is the foundation. *DECODE: The Wake Up X Playbook* holds the six-level path, the Curiosity Compass, and the core frameworks. Nothing in any other book contradicts it. If a term here is unfamiliar, DECODE defines it.

What is in this book

Part One — Understanding the Machine

1. Why Nobody Answers Your Applications
2. The Resume That Survives Six Seconds

Part Two — Working the System

3. T.R.A.C.K.: The Application System
4. The Approach: Talking to Strangers Who Can Hire You

Part Three — The Room

5. Telling the Story of Your Work
6. The Offer, and What's Actually Negotiable

Part Four — After the Yes

7. The First Ninety Days
8. Appendix: Templates, Scripts, and the Bullet Bank

How to actually use it

Every chapter ends with a checklist, exercises, and questions. Those are the book. The reading is the setup; the doing is the point. If you read a chapter and do nothing, you have gained nothing you could not have gotten from a podcast episode.

Work one chapter at a time. Do the exercise before moving on. Ten pages you acted on beat a hundred you skimmed.

A word on honesty. Everything in this book assumes you are telling the truth about what you can do. The tactics here make real capability visible. They do not manufacture capability you do not have, and if you try to use them that way you will be found out in the first technical interview. Build the thing, then make it visible. In that order.

I am writing this the way I would say it to you sitting across a table — not lecturing a room. Take what works, remix the rest into your own version, and leave what does not fit.

PART

ONE

Understanding the Machine

Why Nobody Answers Your Applications

MISSION

To show you exactly why your applications disappear, so you can stop repeating the thing that is costing you.

Objectives

- Understand what actually happens to an application after you click submit
- See the difference between applying and being referred
- Stop measuring effort in applications sent
- Identify which of the four gates you have never once worked

Story

I have sat on the other side of the hiring table. On one opening we received over two hundred applications in about a week. I want to be honest with you about what that felt like, because it explains everything that follows.

I did not read two hundred resumes. Nobody does. I skimmed for something that told me this person understood *our* problem. Most of them did not.

They were the same resume, sent everywhere, with a summary at the top saying the person was a "motivated IT professional seeking opportunities to grow." I believed them. It just did not help me.

The ones I stopped on had one thing in common. Something on the page matched something in the posting — a tool we actually ran, a problem we actually had, or a name I recognized. That was it. That was the entire filter.

Here is the part that should bother you: some of the people I passed over were almost certainly more capable than some of the people I interviewed. I will never know. Neither will they.

That is not a moral failure on anyone's part — it is what happens when two hundred documents arrive in a week and one tired person has to reduce them to five.

Why It Matters

Most people trying to break into IT and cybersecurity believe the problem is that they are not qualified enough. Sometimes that is true. Far more often, the problem is that nobody has

actually read the application.

You cannot fix a qualifications problem that is really a visibility problem. You will study for another certification, send another hundred applications, and get the same silence — and you will conclude, wrongly, that you are the issue. Then you will either give up or spend another year on credentials nobody asked for.

This chapter exists to stop that specific waste.

Professional Explanation

Business Perspective

A hiring manager is not shopping for the most qualified person available. They are trying to close a gap that is currently costing the business time, money, or risk.

The job posting is a description of that pain, written by a human who is tired and who probably copied half of it from the last posting.

When your application speaks to the pain, you become relevant. When it lists your attributes, you become one of two hundred.

There is also a budget dimension people miss. The role has a number attached to it, and that number was justified to someone.

Your application is competing not just against other candidates but against the option of not hiring at all and distributing the work across the existing team. Anything that makes you look like a fast return on that number helps you.

Technical Perspective

Most mid-size and large organizations run applications through an applicant tracking system. It is not magic and it is not out to get you — it is a database with search on top.

A recruiter searches it the way you would search anything: with words from the posting. If those words are not in your document, you are not in the result set.

This is not gaming the system; it is speaking the same language as the system. Writing "endpoint protection" when the posting says "EDR" is not more honest. It is just less findable.

A practical detail: keep the file readable. Clean structure, standard section headings, real text rather than text embedded in images or complicated multi-column layouts. Designer-style resumes with sidebars and icons parse badly. Save the visual flourish for your portfolio, where a human is definitely looking.

Leadership Perspective

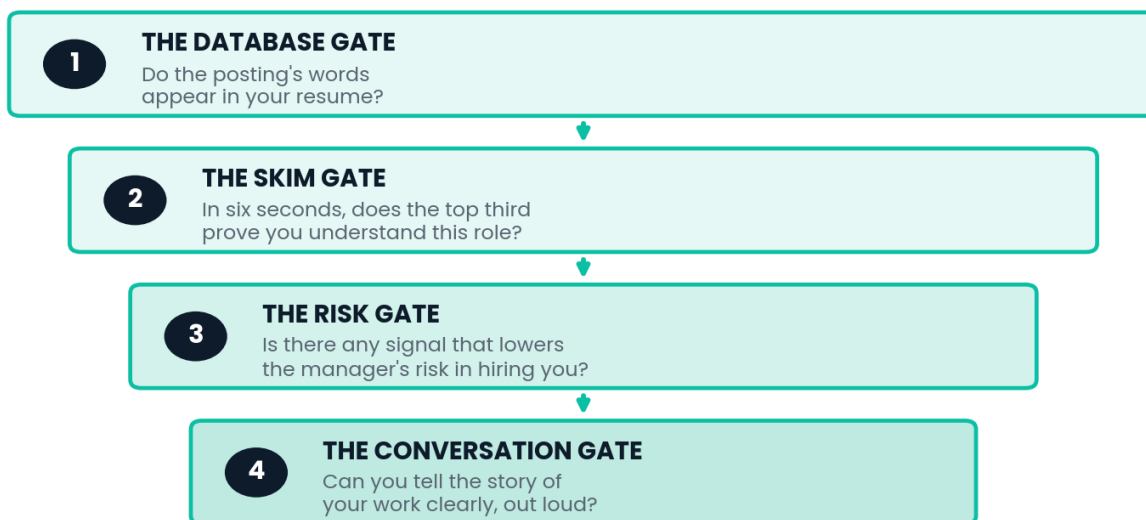
Every hire is a risk that someone puts their name on. When a manager hires you, they are spending their own credibility with their own boss. If you work out, they made a good call. If you do not, they made a bad one, and they will be reminded of it.

That is why referrals move so fast. A referral transfers risk from the unknown to someone the manager already trusts. Understanding this reframes networking entirely: you are not asking for favors, you are lowering someone's risk.

It also explains something that otherwise looks unfair — why a less-qualified referred candidate frequently beats a more-qualified stranger. It is not corruption. It is a manager choosing the option with the smaller downside.

Framework: The Four Gates

Think of every opening as four gates, not one.



Most applicants only ever work Gate 1. Gate 3 is where the leverage is.

The Four Gates. An application does not fail once — it fails at a specific gate, and each gate has a different fix.

Gate 1 — The database gate. Do the posting's words appear in your resume? Machine-level match. The fix is tailoring.

Gate 2 — The skim gate. In roughly six seconds, does the top third of your resume prove you understand this specific role? Human, fast, unforgiving. The fix is a targeted headline and evidence bullets, covered in the next chapter.

Gate 3 — The risk gate. Is there any signal that reduces the manager's risk — a referral, a portfolio, a shared connection, provable work? The fix is the Approach step in Chapter 3. It is the highest-leverage move in this entire book.

Gate 4 — The conversation gate. Can you tell the story of your work clearly, out loud, under mild pressure? The fix is Chapter 5.

Most people spend all their energy on nothing but Gate 1, then conclude the industry is broken. Gate 3 is where the real leverage is, and almost nobody works it — which is exactly why it works.

The reframe that runs through this whole book. You are not applying for jobs. You are reducing a stranger's risk in hiring you. Every tactic in these pages is a version of that one move. If you ever forget the specifics, return to that sentence and you can usually reconstruct the right action.

Real Example

Two people apply for the same help desk role. Both hold A+. Both are career changers with no formal IT experience.

Candidate A submits through the portal and waits. Over the next three weeks she applies to forty more roles the same way. She hears nothing from any of them, concludes the market is impossible, and starts studying for Network+ on the theory that one more certification will fix it.

Candidate B finds the IT manager on LinkedIn, sends a short note — no ask, just a genuine question about whether the role leans more toward end-user support or infrastructure — gets a two-line reply, then applies and mentions that exchange in the first line of his cover note. He follows up on day four and again on day ten.

Candidate B is not more qualified. He arrived at Gate 3 instead of stopping at Gate 1.

The uncomfortable part: Candidate A's instinct to study more was not irrational. It is just an answer to a different question than the one she was facing.

Case Study: The Referral Nobody Planned

A common pattern worth naming, because it looks like luck and is not.

Someone attends a local chapter meeting — ISACA, ISSA, a BSides event, a community college security club. They are not job hunting aggressively that night; they are there to learn. They talk to three people. One of them mentions, in passing, that their team is about to post a role.

Six weeks later that role posts. The candidate applies and messages the person they met. That person forwards the resume internally with a sentence: "I met this person at the chapter meeting, they seemed sharp." The candidate is interviewed within days.

Nothing about that was luck except the timing. The controllable part was being in the room and having a real conversation. If you are in a metro area, there is almost certainly a chapter meeting within driving distance this month. That is the highest-return two hours available to you.

Reflection

- Which gate have you been treating as the whole game?
- If you knew for certain that nobody was reading your applications, what would you do differently this week?
- When was the last time a human being at a target company knew your name before you applied?

Journal

Write for five minutes: *What have I been telling myself about why this is not working?* Then, underneath: *What evidence do I actually have for that?* Most people find the story is inherited rather than observed.

Exercises

1. Pull the last ten jobs you applied to. For each, write one sentence: what specific problem was that team trying to solve? If you cannot answer for more than three of them, you have found your issue.
2. Take one of those postings and highlight every noun that names a tool, a system, or a process. Now check how many of those words appear anywhere in the resume you sent. Count them. That number is your Gate 1 score.

Common Mistakes

- **Measuring effort in volume.** Two hundred generic applications is not more work than twenty tailored ones. It is less work, done more times, and it feels productive while producing nothing.
- **Waiting to be ready.** Postings are wish lists assembled by committee, not requirement lists. Almost nobody meets all of them, including the person who currently holds the job.
- **Treating silence as a verdict.** Silence usually means nobody read it. That is a process outcome, not a judgment of you.

- **Adding certifications to solve a visibility problem.** Certifications are a Gate 1 and Gate 2 asset. They do nothing for Gate 3.

AI Prompts

- "Here is a job posting. Summarize, in two sentences, what problem this team is most likely trying to solve by hiring. Do not restate the requirements."
- "Compare this posting against this resume. List only the words and skills present in the posting and absent from the resume. Do not rewrite anything."
- "Act as a hiring manager with two hundred applications and six seconds per resume. Read mine and tell me what you would conclude in those six seconds."

Books

- *So Good They Can't Ignore You* — Cal Newport. The argument that rare and valuable skills, not passion, create career leverage. Directly relevant to why evidence beats enthusiasm.
- *Never Split the Difference* — Chris Voss. More useful later in this book, at the offer stage, but the underlying idea — that you win by understanding the other side's position. Starts here.

Podcast Episodes

Look through the Wake Up X back catalogue for guest episodes where someone describes how they actually got in. Almost every one contains a Gate 3 moment that the guest describes as luck. Listen for it specifically.

Resources

- **NetworkChuck** — for building the technical foundation that makes your claims true
- **Darknet Diaries** — for understanding what this field actually is, which makes your interview answers less generic
- Your local **ISACA**, **ISSA**, or **BSides** chapter — the single most underused resource available to job seekers in this field

Checklist

- I can name the business problem behind at least one posting I care about
- I have stopped counting applications as progress
- I know which of the four gates I have never once worked
- I have identified one real human at one target organization

Key Takeaways

- Applications fail at the skim and risk gates far more than at the qualification gate
- Referrals work because they transfer risk, not because of favoritism
- Tailoring is not optional polish — it is the entire mechanism of Gate 1
- Adding credentials cannot fix a visibility problem

Next Mission

Now that you know where applications go, we build the document that survives the first two gates.

Questions for you. Which gate have you been treating as the whole game? And what would change this week if you worked Gate 3 instead of sending twenty more applications?

The Resume That Survives Six Seconds

MISSION

To build one resume you can retarget in fifteen minutes for any role, instead of one generic resume you send everywhere.

Objectives

- Structure a resume around evidence rather than duties
- Write bullets that survive a skim
- Build a reusable bullet bank so tailoring stops being painful
- Position lab work correctly when your job history does not match

Why It Matters

Your resume is not your biography. It is an argument, and the claim is: **hiring me is a low-risk decision**. Every line either supports that claim or wastes space you do not have.

Once you accept that, most resume questions answer themselves. Should you include the retail job from six years ago? Only if it supports the claim — and it might, if it involved handling escalations, training people, or working a system under pressure.

Should you list every technology you have touched? No, because a long list dilutes the claim and invites questions you cannot answer.

Professional Explanation

Duties Versus Evidence

Duties describe what you were assigned. Evidence describes what changed because you were there. Nearly every weak resume is a list of duties.

Duty: Responsible for user support tickets.

Evidence: Handled roughly 40 tickets a week; cut repeat password-reset volume by documenting a self-service guide the team still uses.

Same job. The second version answers what happened, at what scale, and what remained after you left. It also gives the interviewer something to ask about, which is the real function of a resume bullet — it is a conversation starter, not a complete account.

The Structure That Works

1. **Header.** Name, city and state, phone, email, LinkedIn, portfolio or GitHub link. No mailing address — nobody is sending you post, and it dates the document.
2. **Headline.** The role you are targeting, in the words of the posting. "SOC Analyst" if the posting says SOC Analyst. Not "Motivated IT Professional."
3. **Summary.** Three lines maximum, aimed at this specific role. This is the highest-value real estate on the page and most people waste it on adjectives.
4. **Skills.** The honest, searchable list. Split into what you have used in production versus what you have used in a lab. Do not blur that line — an interviewer will find the seam.
5. **Experience.** Evidence bullets, most relevant first.
6. **Projects.** Your home lab, treated exactly like work experience. This is where career changers win.
7. **Certifications and education.** Last, unless a certification is the specific reason you are being considered.

Home lab goes above the fold when your job history does not match. If you are coming from retail, food service, the military, teaching, or anywhere else outside technology, the lab is your evidence. It belongs high on the page — right after the summary — not buried at the bottom under "Interests." I have seen strong candidates hide the only genuinely relevant thing they had.

On Length

One page early in a career. Two pages once you have five or more years of directly relevant work. Three pages essentially never, outside academia and government.

The reason is not a rule. It is that a longer document does not get read more thoroughly — it gets skimmed at the same speed. That means your best material has more competition from your own weaker material.

Framework: The Evidence Bullet

Every bullet uses the same skeleton.

Action verb + what you did + the scale or tool + what changed.

Examples across different backgrounds:

- Built a Windows Server domain controller with Active Directory and Group Policy, managing 15 lab endpoints with enforced password and lockout policy

- Deployed Wazuh SIEM across a segmented home network, tuning rules to cut false-positive alerts by roughly half
- Segmented a home network into four VLANs on Ubiquiti equipment, isolating IoT devices from workstations and lab machines
- Documented a recurring printer-driver failure and wrote the fix into the team knowledge base, reducing repeat tickets on that issue
- Trained six new hires on the point-of-sale system and wrote the reference sheet the store still uses (*retail — and it demonstrates documentation and training, which are real IT competencies*)
- Managed inventory reconciliation for a location processing roughly 400 transactions a day, cutting month-end discrepancies to near zero (*also retail — demonstrates systems thinking and attention to data integrity*)

Notice none of those claim a title. They claim work. And notice that the last two are not technology jobs at all — the skeleton still applies, and the underlying competencies transfer.

On Numbers

Use them where you have them, and do not invent them where you do not.

"Roughly 40 tickets a week" is fine. "About half" is fine. A made-up "37% improvement" is a trap, because the first competent interviewer will ask how you measured it and the conversation will end badly.

Honest imprecision beats dishonest precision. Every time.

Step-by-Step Guide: Building the Bullet Bank

This is the single highest-return hour you will spend on your job search.

1. **Open one document.** Call it BULLET_BANK. Never write a resume from a blank page again.
2. **Create themed sections:** Support, Networking, Security, Cloud, Automation, Documentation, Leadership, Customer Service, Projects.
3. **Empty everything into it.** Every job, every lab, every volunteer thing, every side project. Do not filter yet — filtering while generating stops the generating.
4. **Rewrite each entry using the evidence skeleton.** Action, what, scale or tool, what changed.
5. **Mark each one honestly.** Tag whether it was production or lab. You will thank yourself in an interview.
6. **Now tailoring is selection, not writing.** Read the posting, pull the eight to ten bullets that match, reorder, adjust the headline and summary. Fifteen minutes.

7. **Add to it continuously.** Every time you finish a project or solve something notable, it goes in the bank that week — not eighteen months later when you are job hunting and cannot remember.

Do not inflate. The bullet bank makes it very easy to start claiming things you half-did. A bullet you cannot talk about for two minutes in an interview is a liability, not an asset. Every line must be defensible under a follow-up question, because it will get one.

Real Example

A career changer coming out of eight years in restaurant management believed she had nothing relevant. Her first resume led with her certifications and buried everything else.

What she actually had: she had run scheduling systems, trained dozens of people, handled vendor relationships, managed inventory software, and de-escalated angry customers daily. She had also built a small home lab over about six months.

The rewrite led with a targeted headline, a three-line summary aimed at help desk work, her lab projects immediately after, and then her management experience reframed as evidence. Training, documentation, systems, and pressure handling.

Nothing was invented. The same eight years produced a completely different document, because the claim being argued had changed.

Case Study: What Two Resumes Look Like Side by Side

Same person, same experience, two versions.

Version A, top third:

Motivated and detail-oriented IT professional seeking to leverage my skills in a challenging environment where I can grow and contribute to organizational success. Skills: Windows, Linux, networking, troubleshooting, communication, teamwork, Microsoft Office

Version B, top third:

Help Desk / IT Support Technician Career changer with a hands-on home lab (Active Directory, Group Policy, Windows Server) and eight years managing teams and systems under pressure. CompTIA A+. Looking for a first support role on a team that documents its work. **Production:** Windows 10/11, ticketing systems, user training **Lab:** Windows Server 2022, Active Directory, Group Policy, Hyper-V, Pi-hole

Version B tells you the target role, the evidence, the honest gap, and the person's values, in about the same number of words. Version A tells you nothing that distinguishes the writer from two hundred other applicants.

Templates: The Summary Formula

[TARGET ROLE TITLE]

[Where you're coming from] with [the most relevant concrete evidence you have]. [Certification, if it's a differentiator]. Looking for [the specific kind of role and team you actually want].

Three lines. Specific. Honest about the transition rather than hiding it, because hiding it never works and being direct about it reads as confidence.

Project

Rebuild your resume completely, from the bullet bank, targeting one specific real posting you actually want. Not a general-purpose version — a version for one job.

Then do the six-second test: hand it to someone, let them look for six seconds, take it away, and ask what role you are applying for. If they cannot say, the top third has failed.

AI Prompts

- "Here is a job posting and here is my resume. List the words and skills in the posting that do not appear anywhere in my resume. Do not rewrite anything yet."
- "Rewrite this bullet as evidence: what I did, at what scale, and what changed. Keep it to one line and do not invent numbers I did not give you."
- "Here is my non-technical work history. Identify the transferable competencies that matter for an IT support role, and phrase each as an evidence bullet using only what I have told you."

- "Act as a hiring manager for this posting. Read my resume and tell me the first three reasons you would pass on it."

Books

- *So Good They Can't Ignore You* — Cal Newport
- *The Power of Habit* — Charles Duhigg. Relevant to the bullet bank specifically: the habit of logging work as it happens is worth more than any single rewrite.

Resources

- Your target postings themselves — read twenty in your lane and the vocabulary of the field becomes obvious
- LinkedIn's skills section on people currently holding the job you want; it is a free, accurate keyword list

Checklist

- Every bullet names what changed, not just what I was assigned
- My lab projects are written like real experience, positioned high if my history does not match
- My bullet bank exists and has at least twenty entries
- My skills section separates production use from lab use
- Someone has taken the six-second test on my resume
- I have not invented a single number

Common Mistakes

- **Listing every technology you have ever touched.** It reads as noise and invites questions you cannot answer.
- **Two pages of duties.** One page of evidence beats it every time early on.
- **Hiding the lab.** If it is the strongest thing you have, treat it as the strongest thing you have.
- **Letting AI write it in its own voice.** Use AI to find gaps and pressure-test. Keep the words yours, because you will have to defend them out loud.
- **A "skills" section that is a wall of undifferentiated words.** It reads as padding and it destroys the honesty of the rest.

Key Takeaways

- Duties describe assignment; evidence describes impact
- The bullet bank turns tailoring from writing into selection
- Career changers win on the projects section, so it goes high
- Honest imprecision beats invented precision

Next Mission

The resume gets you read. The T.R.A.C.K. system gets you answered.

Questions for you. Which of your bullets could you not talk about for two full minutes? What would you have to actually go do to make that bullet defensible?

PART

TWO

Working the System

T.R.A.C.K.: The Application System

MISSION

To replace spraying applications with a repeatable sequence that gets responses.

Objectives

- Run the full T.R.A.C.K. sequence on one real opening
- Learn the follow-up cadence most candidates never use
- Build a contact ledger so no connection is ever lost

Why It Matters

Most people do exactly one step of this: they apply. T.R.A.C.K. is the full sequence, and the steps before and after the application are the ones that actually move the needle.

This is also the framework I would most encourage you to remix. Everything in the Wake Up X library is consciously borrowed from sales, marketing, and relationship-building thinking that already existed and then dragged into a field that does not naturally reach for it.

Cybersecurity and IT are unusually isolated disciplines. That isolation is exactly why importing a structured outreach sequence works so well here — you are competing against people who consider this beneath them.

Framework: T.R.A.C.K.

T.R.A.C.K. — THE APPLICATION SYSTEM



Almost everyone does only 'C'. The step that changes outcomes is 'A'.

The T.R.A.C.K. Application System. The step that changes outcomes is the one almost nobody runs.

T — Tailor

Rework the resume for this specific posting using your bullet bank. Match the headline to the job title. Adjust the summary. Reorder the bullets so the most relevant sit highest.

Fifteen minutes, not two hours. If it is taking two hours, your bullet bank is not built yet — go back to Chapter 2.

R — Research

Learn the environment before you speak. Ten focused minutes:

- What does this company actually do, and who pays them?
- What tools show up in their *other* job postings? (This is the best free intelligence available — postings leak the tech stack.)
- Who leads the team? What is their background?
- What did they announce recently — funding, an acquisition, a product, a breach, a new office?
- Are they hiring one person or eight? Eight means growth or churn. It is worth knowing which.

This ten minutes makes every later step better. It is the difference between a message that reads as thoughtful and one that reads as templated.

A — Approach

This is the step almost everyone skips. It is the whole reason the system works.

Before or right around the time you apply, reach out to a human on that team — the hiring manager, a senior engineer, someone in the department. Not a pitch. A short, genuine, specific message.

"Hi Maria — I saw the SOC analyst opening on your team. I have been running Wazuh in a home lab for the last several months and noticed the posting mentions detection tuning. Curious whether the role leans more toward triage or detection engineering. Either way I'm applying — appreciate any read you have on it."

That message works for four reasons: it is specific, it is short, it proves you did the research, and it asks a real question instead of demanding a favor. Chapter 4 covers this step in full, because it is worth its own chapter.

C — Contribute the Application

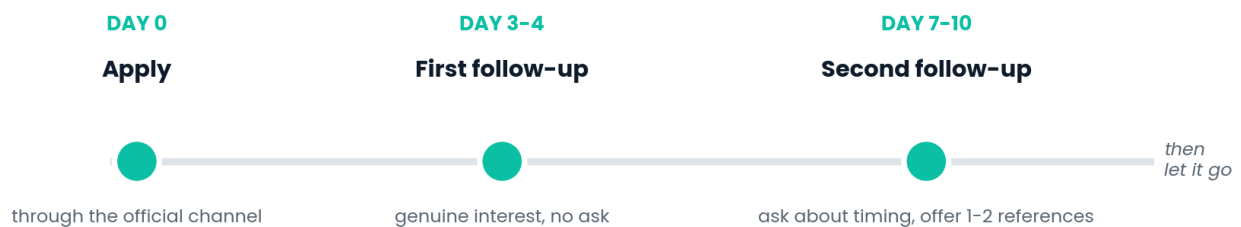
Now apply through the official channel. Always the official channel — going around it creates work for people and makes you look like you do not understand how organizations function.

In your cover note or first line, reference the conversation: *"I spoke briefly with Maria on the SOC team about the detection-tuning side of this role."* You have just arrived at Gate 3 with a name attached.

If the person never replied, you can still write: *"I reached out to Maria on your team with a question about the role."* True, honest, and it still signals initiative.

K — Keep Following Up

Here is the rhythm. It is specific on purpose.



Two follow-ups is diligence. Six is a red flag.

The follow-up cadence. Two messages, then you let it go and keep moving.

Days 3 to 4 after applying. A short email expressing genuine continued interest. No ask, no pressure. One or two sentences reaffirming why this role in particular. The purpose is presence, not information.

Days 3 to 6 after that. A direct, professional note asking when you might expect to hear back, and offering one or two references who can vouch for you. The references offer is the part people skip. It is quietly powerful — it moves you from a claim to a verifiable claim.

Two follow-ups. Then you let it go and keep moving.

Most candidates send zero. The person who sends two, professionally, stands out for reasons that have nothing to do with their skill set.

There is a line. Two thoughtful follow-ups is diligence. Six is a red flag that will follow you, because recruiters talk to each other and the field is small. Send the two, then move on. If you get a no, thank them genuinely and ask to stay in touch — that is not a formality, and Chapter 6 explains why.

Templates: The Contact Ledger

Every recruiter, every chapter contact, every person you meet at a job fair or ISACA meeting goes into one running log.

Field	Why it matters
First and last name	You will not remember the last name. Nobody does.
Phone and email	The whole point
Where they work	And update it when they move, because that move is often your opportunity
How you know them	"Met at ISACA Sac, March, talked about SIEM tuning" — this is what makes a re-contact possible without being awkward
Last contact date	Tells you when a check-in is due
Notes	Anything human. Kids' names, the conference they were heading to, the thing they were frustrated about.

This is not a CRM and it does not need to be one. A spreadsheet is fine. What matters is that it exists — because two years from now the person who could not hire you is at a different company that is hiring. You will not remember their last name.

Never burn a bridge. The industry is far smaller than it looks, and it has a long memory in both directions. The recruiter you were short with, the manager whose offer you ghosted, the colleague you criticized publicly — you will meet them again, usually at the least convenient moment.

Step-by-Step Guide: Running T.R.A.C.K. on One Opening

1. Pick one opening you actually want. Not the most likely — the one you want.
2. Tailor the resume from the bank. Fifteen minutes.
3. Research for ten minutes. Write down three things you learned.
4. Identify one human on that team. Write and send the approach message.
5. Apply through the official channel, referencing the outreach.
6. Put both follow-ups on your calendar right now, with the dates calculated. Not "remember to follow up" — actual calendar entries with the draft text in the notes field.
7. Log everyone you touched in the contact ledger.
8. Move to the next opening while the follow-ups run on their own schedule.

One job, done completely, beats thirty done partially. It is also less total work, which people find hard to believe until they try it.

Real Example

Two candidates, same posting, same certification, comparable labs.

Candidate A submits and waits, then applies to thirty more the same way.

Candidate B messages an engineer on the team with a specific question, applies referencing that exchange, follows up on day four with two sentences, and again on day nine offering references.

Candidate B has made four separate impressions on that organization. Candidate A has made one, into a database. Candidate B applied to six roles that month. Candidate A applied to thirty-one.

Candidate B got three interviews.

Case Study: What the Second Follow-Up Actually Does

A pattern worth understanding, because it explains why the rhythm is shaped the way it is.

Hiring is slow for reasons that have nothing to do with you. Someone is on leave. Budget is pending final approval. The panel cannot align calendars.

The internal candidate is still being considered. During that gap, the pile of applications sits, and candidates quietly fall out of active consideration simply by not being top of mind.

The second follow-up frequently arrives exactly when the process restarts. It is not persuasion — it is timing. You cannot know when the gap ends, so you send on a schedule and let the odds work.

The references offer does something additional: it hands the manager a way to reduce their risk without doing any work. That is a gift. It is rarely refused.

Reflection

- How many of your applications have ever had a human name attached to them?
- What is actually stopping you from sending the approach message — is it that it does not work, or that it is uncomfortable?
- If you had to get hired with only five applications, what would you do differently?

Journal

Write about the last time someone helped you professionally without being asked. What made you willing to help them, or them willing to help you? That answer is the mechanism you are trying to activate.

Exercises

1. Build your contact ledger today and backfill it with everyone you can remember from the last two years. Most people find fifteen to thirty names they had effectively lost.
2. Pick one real opening and run all five steps this week.
3. Set a recurring monthly reminder to update the ledger. It decays fast without one.

AI Prompts

- "Here is a job posting and the company name. Give me five specific things I should know before speaking to someone on this team, and tell me what you are unsure about."

- "Draft three versions of a short outreach message to an engineer on this team. Keep each under 60 words, ask a real question, and do not ask for a referral."
- "Review this follow-up email. Is there anything in it that reads as entitled, needy, or templated?"

Books

- *How to Win Friends & Influence People* — Dale Carnegie. Old, still correct, and the source most of this outreach thinking traces back to.
- *The Like Switch* — Jack Schafer. Practical mechanics of building rapport quickly, written by someone who did it professionally.

Podcast Episodes

Wake Up X guest episodes where the guest describes their entry point. Listen specifically for the human who opened the door — there almost always is one, and guests rarely frame it that way themselves.

Checklist

- I tailored using my bullet bank
- I researched the environment before reaching out
- I approached a human before or as I applied
- I referenced that conversation in the application
- Both follow-ups are on my calendar with dates and draft text
- Every contact went into the ledger

Common Mistakes

- **Approaching with an ask.** "Can you refer me?" from a stranger is a no. A real question is a yes.
- **Skipping the follow-up because it feels pushy.** Two is not pushy. Zero is invisible.
- **Never logging contacts.** You will lose them. Everyone does, without a ledger.
- **Running the system on jobs you do not want,** which wastes the effort the system needs.

Key Takeaways

- The application is step four of five, not the whole process
- Two structured follow-ups separate you from nearly everyone

- The contact ledger compounds over an entire career
- One opening done completely beats thirty done partially

Next Mission

The Approach step deserves its own chapter, because it is the one people find hardest.

The Approach: Talking to Strangers Who Can Hire You

MISSION

To make the outreach message feel possible, and to make yours good.

Objectives

- Write an approach message that gets replies
- Know who to contact and who not to
- Handle silence and rejection without it costing you anything

Why It Matters

This is the step with the highest return and the highest resistance. Everything else in this book is a document you edit alone. This one needs contacting a stranger who owes you nothing, and most people will do almost anything to avoid it — including sending two hundred more applications.

I understand the resistance. When I was young and did not speak English well, I had to fix people's computers while barely being able to explain what I had done.

I had no family network, no introductions, nobody to make a call for me. I had to social engineer and reverse engineer my own way in.

What I learned from that: people are far more willing to help than you expect, and the thing blocking you is almost never their unwillingness. It is your fear of their unwillingness.

Professional Explanation

Who to Contact

In rough order of value:

1. **The hiring manager.** Usually identifiable from the posting's language, the team structure on LinkedIn, or the department. Highest value, highest bar for the message quality.
2. **A senior engineer or analyst on the team.** Often more responsive than the manager, and their opinion carries real weight internally.
3. **Someone in the department, adjacent to the role.** Useful for context, can point you to the right person.

4. **A recruiter.** Worth contacting, but understand their incentive — they are managing a pipeline, not advocating for you personally. Still useful, and worth keeping in the ledger permanently, because recruiters change companies and take their networks with them.
5. **Anyone you already have a connection to.** Always start here if it exists. A shared alma mater, a shared chapter, a former colleague — check before you cold-contact.

Who Not to Contact

- The CEO of a large organization about a help desk role. It reads as not understanding scale.
- Five people on the same team simultaneously. They talk.
- Anyone on a personal channel they did not offer. LinkedIn or work email. Not their personal Instagram.

What Makes a Message Work

Four elements, all required:

Short. Under sixty words. Long messages from strangers do not get read.

Specific. Reference the actual role, the actual company, something actual from their posting or their work. Generic messages get generic outcomes.

A real question. Something you actually want to know and that they can answer in one line. This is what makes replying easy.

No ask. Do not request a referral, a call, or "any advice you might have" from a stranger. The ask comes later, if at all, and usually they offer before you ask.

Why "no ask" works. A request from a stranger creates obligation, and obligation creates avoidance. A genuine question creates a small, easy, satisfying interaction. People like answering questions in their area of expertise — it costs them nothing and it feels good. Start there.

Templates: Approach Messages

Three versions for different situations. Adapt them; do not send them verbatim, because templated language is detectable.

To an engineer on the team:

Hi [Name] – I saw the [role] opening on your team and I'm applying. I've been running [specific relevant thing] in a home lab and noticed the posting mentions [specific thing from posting]. Curious whether the role leans more toward [X] or [Y]? Either way, appreciate the read.

To a hiring manager:

Hi [Name] – applying for the [role] on your team this week. Before I did, I wanted to ask: what does someone doing well in this role look like six months in? Want to make sure I'm framing my application against the actual job rather than the posting.

After meeting someone at an event:

Hi [Name] – we spoke at [event] about [specific topic]. I saw [company] posted a [role] – I'm applying. If there's anything you'd flag about the team or what they're actually looking for, I'd value it. Either way, good to have met you.

To reconnect with someone in your ledger, months later:

Hi [Name] – we spoke back in [month] about [topic]. I've since [specific thing you did – built X, earned Y, started at Z]. Saw you moved to [company] – congrats. Would be glad to stay in touch.

That last one is the most valuable template in this book and the one people never use. It costs two minutes and it keeps a network alive.

Step-by-Step Guide: Sending Your First One

1. Pick the role. Pick the person.
2. Spend five minutes on their profile. Find one specific, non-creepy thing – a project, a post, a talk, their team's focus.
3. Draft under sixty words using one of the shapes above.
4. Read it aloud. If it sounds like a template, rewrite the middle sentence in your own words.

5. Check it contains a real question and no ask.
6. Send it. Do not wait for the perfect version.
7. Log it in the ledger.
8. Do not follow up on the approach message itself. If they do not reply, that is fine — you still reference it in the application.

Real Example

Compare two messages sent to the same engineer.

Message A: *"Hi, I'm very interested in opportunities at your company and would love to connect and learn more about any openings you might have. I have my Security+ and am a fast learner. Would you be able to refer me?"*

Message B: *"Hi Devin — applying for the analyst role on your team. I've been running Wazuh in a home lab for a few months and the posting mentions detection tuning. Is that mostly rule-writing on the team, or more triage with occasional tuning? Curious what the day actually looks like."*

Message A asks a stranger for a favor, contains nothing specific, and could have been sent to anyone. Message B is answerable in one line, proves the sender did real work, and gives Devin something interesting to respond to.

Devin will reply to B. He may even forward it internally, unprompted, because a specific and thoughtful message is memorable in a way that a request is not.

Case Study: The Chapter Meeting

Local professional chapters — ISACA, ISSA, BSides, OWASP, and their equivalents — are the most consistently underused resource in this field.

Here is why they work. The people who attend chapter meetings are, by definition, people invested enough in the field to spend an evening on it. They are frequently senior.

They are there to talk to people. And crucially, they are not in gatekeeping mode — nobody is screening at a chapter meeting, so the interaction starts as a peer conversation rather than a candidate one.

Partnering with a chapter, as I do with ISACA Sacramento, takes that further. But you do not need a formal role. You need to show up three times so people recognize your face, ask real questions, and follow up with the people you spoke to.

The follow-up is the part almost everyone skips. It is where the entire value is. A conversation you do not log and re-contact was entertainment.

Reflection

- What specifically are you afraid will happen if you send the message?
- Has that thing ever actually happened to you or anyone you know?
- What is the realistic worst case — and is it worse than the certain outcome of not sending it?

Journal

Write the message to the person you are most intimidated to contact. Do not send it yet. Just write it. Most people discover the writing was the hard part and the sending is trivial once the words exist.

Exercises

1. Send one approach message this week. One. To anyone, about any role.
2. Find your nearest ISACA, ISSA, BSides, or OWASP chapter and put the next meeting in your calendar.
3. Go back through your ledger and send one reconnect message to someone you have not spoken to in six months.

AI Prompts

- "Here is my draft outreach message. Tell me honestly whether it reads as templated, entitled, or needy — and identify the specific words causing it."
- "Here is a LinkedIn profile summary and a job posting. Suggest three genuine questions I could ask this person that they could answer in one sentence."
- "Rewrite this message to be under 60 words without losing the specific detail."

Books

- *How to Talk to Anyone* — Leil Lowndes
- *Influence* — Robert Cialdini. Specifically the reciprocity and liking chapters, which explain why the no-ask approach works.

Resources

- Your local chapter — genuinely, this is the highest-return item on any resource list in this book
- LinkedIn's "people also viewed" on a team member's profile, for mapping who else is on the team

Checklist

- I have sent at least one approach message
- My messages are under sixty words and contain a real question
- I never ask a stranger for a referral
- I have a chapter meeting in my calendar
- Everyone I contact goes into the ledger

Common Mistakes

- **Asking for a referral in the first message.** The single most common error.
- **Writing long.** Anything over sixty words from a stranger goes unread.
- **Being generic,** which produces a generic non-response.
- **Contacting five people on one team.** They compare notes and it looks desperate.
- **Following up on an unanswered approach message.** Let it go; the application still references it.

Key Takeaways

- The approach is the highest-return, highest-resistance step
- Short, specific, a real question, no ask
- Chapter meetings put you in rooms where nobody is screening you
- The reconnect message is the most underused tool you have

Next Mission

You have the interview. Now we make sure you can tell the story of your work.

PART

THREE

The Room

Telling the Story of Your Work

MISSION

To answer interview questions with evidence instead of adjectives.

Objectives

- Use STAR without sounding rehearsed
- Answer honestly when you do not know something
- Ask questions that make you the obvious hire
- Handle the failure question without damaging yourself

Why It Matters

The interview is where Gate 3 closes or stays open. The manager already believes you might be able to do the job — that is why you are in the room. They are now checking something different: **will you be a problem?**

Will you hide mistakes? Will you go quiet when stuck? Will you be difficult to work with at 2am during an incident? Will you still be here in eighteen months?

Almost every interview question is one of those questions in disguise.

Professional Explanation

The STAR Method, Practically

Situation — brief context. Two sentences maximum.

Task — what was actually your responsibility.

Action — what *you* did. Say "I," not "we." This is the longest part by far.

Result — what changed. Numbers if you have them, honest description if you do not.

Most people over-invest in Situation and rush Action. Flip that ratio. The interviewer does not want the backstory; they want your decisions. When you describe the setup for ninety seconds and your own actions for fifteen, you have told them nothing about yourself.

A useful discipline: after you say the situation, ask yourself internally, *have I said "I" yet?* If not, you are still in setup.

The "We" Problem

Technical people say "we" constantly, out of genuine humility and team culture. In an interview it is actively harmful, because the interviewer cannot tell what you did.

Say "we" when describing the team's context. Say "I" when describing your own actions. Both are honest. Only one is useful.

Answering What You Don't Know

This is the single highest-leverage skill in an interview. It is simple:

"I haven't worked with that directly. Here's what I do know that's adjacent, and here's exactly how I'd go about finding the answer."

That answer is strong. It demonstrates honesty, self-awareness, adjacent knowledge, and method — four things they are actively testing for.

Bluffing is fatal. Technical interviewers can tell instantly, and once they have caught one bluff they will re-examine everything else you said. You do not recover from that in the same conversation.

The Failure Question

"Tell me about a time something went wrong and it was your fault."

Rules:

- Pick a real failure. A disguised strength ("I care too much") is transparent and it insults them.
- Own it plainly. No distributed blame, no "the requirements were unclear."
- Spend most of the answer on what you did about it and what you changed afterward.
- Do not pick something catastrophic that suggests recurring judgment problems.

I have failed more times than I can count — more than a hundred, probably more than five hundred, across businesses, hires, and decisions.

The bad hire I made early on was entirely mine: I brought someone on through a friend's recommendation without a real interview, he sounded deeply technical, and what the business actually needed was someone customer-facing. He focused on his own projects over the business's goals. That was my failure of vetting and expectation-setting, not his failure of character.

That is what a real failure answer sounds like. Specific, owned, and it ends in a changed practice.

Failure is not the opposite of success. It is part of it. An interviewer who hears you say that and mean it is hearing someone who will tell them when something breaks, which is exactly what they need.

Framework: The Three Questions Behind Every Interview

Whatever they ask, they are checking three things.

1. **Can you do the work?** Answered with evidence, lab stories, and specific technical detail.
2. **Will you be a problem?** Answered by how you talk about past teams, past managers, and your own mistakes.
3. **Will you still be here in eighteen months?** Answered by genuine, specific interest in *this* role rather than any role.

Question 2 is where most candidates lose without knowing it. Never speak badly about a former employer, manager, or colleague — even when it is completely justified, and sometimes it is. The interviewer does not hear a fair complaint. They hear a preview of how you will talk about them.

If you left somewhere bad, the honest and safe version is: *"It wasn't the right fit for how I like to work, and I learned what I'm looking for from it."* True, professional, and it closes the topic.

Step-by-Step Guide: Preparing Properly

1. **Build three STAR stories.** One technical problem you solved, one time you worked with difficult constraints or people, one genuine failure.
2. **Practice them out loud.** Not in your head. The gap between thinking an answer and saying it is enormous.
3. **Record yourself once.** Painful, effective. Most people discover they never reach the Action section.
4. **Prepare your "I don't know" answer.** Pick a technology adjacent to the role that you genuinely have not used, and rehearse the honest response until it is comfortable.
5. **Research the interviewers.** Names are usually in the calendar invite. Knowing someone's background lets you pitch your answers at the right level.
6. **Prepare five real questions.** Chapter detail below.
7. **Re-read your own resume.** You will be asked about a bullet you wrote four weeks ago and forgot.

Templates: Questions Worth Asking

Ask questions a person who has genuinely thought about the job would ask.

- What does the first ninety days look like for whoever takes this role?
- What is the thing this team is most frustrated by right now?
- How does this team decide what to work on?
- What separates someone who does well here from someone who struggles?
- Is this a new role or a backfill? (*Tells you a great deal. A backfill invites the natural follow-up: what happened?*)
- How does the team handle it when something goes wrong at 2am?
- What is something you'd change about how this team works?

Avoid: questions about vacation, remote policy, or salary in a first interview — those come later and they are legitimate, just not first. Also avoid questions whose answers are on the company's homepage, which signals you did not do the ten minutes.

Real Example

An interviewer asks a career changer: "Have you worked with Active Directory in a production environment?"

Weak answer: "Yes, I have some experience with that." (*Vague, and if it is exaggerated, the next question exposes it.*)

Strong answer: "Not in production. I've built it in my lab — a domain controller with Group Policy enforcing password and lockout policy across about fifteen machines. I broke DNS on purpose once to understand what happens to authentication when it fails, which was more educational than the build. So I understand the concepts and the failure modes, but I haven't supported it at organizational scale with real users."

The second answer is more honest, more specific, and more impressive. It also invites a follow-up conversation the candidate can win.

Case Study: The Interview That Turns on One Answer

A pattern worth recognizing.

Late in an interview, a manager asks something outside the candidate's experience. The candidate has been doing well. They feel the momentum and do not want to break it, so they produce a plausible-sounding answer built on inference rather than knowledge.

The interviewer, who has done this work for a decade, recognizes the shape of a guess immediately. They do not challenge it — that would be rude. They simply make a note and

move on.

Everything after that is discounted. The candidate leaves believing it went well.

The alternative costs nothing: *"That's outside what I've done. Can I tell you how I'd approach finding out?"* Momentum is not broken by honesty. It is broken by being caught.

Project

Do a full mock interview with someone. Not a friend who will be kind — someone who will push. Give them your resume and the posting, and ask them to follow up hard on any bullet that sounds vague.

If you have nobody, record yourself answering ten questions and watch it back. It is uncomfortable and it is the fastest improvement available.

Reflection

- Which of your resume bullets would collapse under two follow-up questions?
- What is the real failure story you would rather not tell? (*That is usually the right one, told well.*)
- What do you actually want to know about this team that you have not been asking?

Journal

Write your genuine reason for wanting this specific role — not the interview version, the true one. Then look at it and find the part of it that is also true in a way you would be comfortable saying out loud. That is your answer to "why this role."

Exercises

1. Record yourself answering "tell me about a time something went wrong and it was your fault." Watch it. Count the seconds before you say "I."
2. Write your "I don't know" answer for the three technologies most likely to come up that you have not used.
3. Write five questions specific to one real company you are targeting.

AI Prompts

- "Act as a technical interviewer for this posting. Ask me one question at a time, wait for my answer, and follow up hard on anything vague."

- "Here is my STAR answer. Tell me how many seconds I spend on setup versus my own actions, and where I say 'we' when I should say 'I.'"
- "I have not used [technology]. Help me build an honest answer that shows adjacent knowledge and method, without overstating what I know."
- "Here is the company and the role. Suggest five questions I could ask that would signal I have actually thought about this job."

Books

- *Never Split the Difference* — Chris Voss. Particularly the sections on calibrated questions and mirroring, both directly applicable in an interview.
- *Extreme Ownership* — Jocko Willink and Leif Babin. The failure-question chapter above is essentially this book applied to an interview.

Podcast Episodes

Any Wake Up X episode where a guest describes a mistake. Notice how the good ones own it in one sentence and spend the rest on what changed. That is the shape you want.

Checklist

- I have three STAR stories ready, including one real failure
- I say "I" when describing my own actions
- I have a genuine, comfortable answer for a technology I do not know
- I have five real questions prepared for this specific company
- I have never rehearsed criticizing a past employer
- I have re-read my own resume this week

Common Mistakes

- Rambling because there was silence. Answer, then stop. Silence is the interviewer thinking, not an invitation.
- Saying "we" throughout, leaving your own contribution invisible.
- Having no questions, which reads as no interest.
- Memorized answers that collapse at the first follow-up.
- A disguised-strength failure answer.
- Criticizing a former employer, however fairly.

Key Takeaways

- Interviewers are assessing risk, not just skill
- "I don't know, here's how I'd find out" beats bluffing every time
- Your questions are part of your answer
- Own failures plainly and spend the answer on what changed

Next Mission

The offer conversation, and what is actually negotiable.

The Offer, and What's Actually Negotiable

MISSION

To handle the offer professionally without leaving value on the table or damaging the relationship.

Objectives

- Know what is negotiable and what is not
- Ask without jeopardizing the offer
- Get everything material in writing
- Handle a rejection so that it stays useful

Why It Matters

This conversation lasts fifteen minutes and can affect years of compensation. It is also the moment people are most afraid, because the offer feels fragile and they do not want to lose it.

Here is the reassurance: a professionally-made ask does not lose offers. Companies that rescind over a polite question were going to be difficult employers anyway, and you have just gathered useful information cheaply.

Professional Explanation

Know Which Conversation You Are In

Negotiation is real with business owners, hiring managers, and decision-makers. It is not real in a retail transaction — the person at the tone cannot change the price. Pushing makes you look like you do not understand the situation.

Some entry-level roles sit closer to the retail end: fixed bands, HR-administered, genuinely no room. Ask anyway, politely, once. The answer tells you which situation you are in, and a no costs you nothing when it is asked well.

Enthusiasm Before Ask

The sequence matters enormously.

"I'm genuinely excited about this — the team and the work are exactly what I'm looking for, and I want to accept. Before I do, is there any flexibility on the base? I'd been targeting [number] based on [reason]."

Enthusiasm first, ask second, reason third. This makes it a conversation between two people who both want the same outcome, rather than a demand from someone who might walk.

Never lead with the ask. Never imply a competing offer you do not have — it is checkable more often than people think, and it poisons the relationship if discovered.

The Whole Package

Base salary is one lever. It is frequently the least flexible one, because it is banded and because it affects internal equity across the team.

These are often much easier for a manager to approve:

Lever	Why it's often easier to say yes to
Certification and training budget	Comes from a different budget line, and it makes you better at the job
Conference attendance	Same, plus visible investment in you
Start date	Costs nothing, and two extra weeks of rest before a new role is real value
Title	Sometimes free, and it compounds for your entire career
Schedule flexibility or remote days	Costs nothing if the team already works that way
Equipment	A better laptop or a lab budget is trivially approvable
An early review	"Can we do a compensation review at six months rather than twelve?" — extremely common yes

That last one is the most underused ask in this list. It converts a no on base salary into a scheduled second conversation, at which point you will have evidence.

Get it in writing. Every time. Regardless of how much you like the person across the table, regardless of how good the conversation was. If a manager promises a review at six months, ask for it in the offer letter or in a follow-up email you send summarizing the conversation. A friendly email confirming what you agreed is not distrust; it is professionalism, and no legitimate employer objects to it.

I once walked away from a million-dollar opportunity because the other side — with a lawyer sitting right there — refused to put anything in a signed contract and wanted a verbal agreement using my own money. That took about four seconds to decide. It is also a large part of why I moved toward compliance and GRC work: I had learned firsthand what happens when nothing is written down. When someone resists documentation, the resistance itself is the information.

Before You Ask, Know Your Number

Do the work first:

- What does this role pay in this metro area, at this level? Multiple sources, not one.
- What do you actually need to live on? Different from what you want.
- What is your walk-away number, decided calmly in advance rather than emotionally in the moment?

Then, when they ask your expectations, you can give a researched range with a reason. Naming a range with a justification is far stronger than naming a number, and far stronger than deflecting entirely.

Step-by-Step Guide: The Offer Conversation

1. **Thank them genuinely, and do not accept on the call.** "This is great news, thank you. Can I take until [day] to review?" Almost always yes, and it buys you thinking time.
2. **Get the whole offer in writing** before evaluating anything — base, bonus structure, benefits, start date, title, PTO.
3. **Evaluate the whole package**, not just the number.
4. **Decide your one or two asks.** Not five. Two well-chosen asks read as considered; five reads as adversarial.
5. **Make the ask**, enthusiasm first, with a reason.
6. **Accept graciously whatever the answer is.** If it is no, "understood, I'm still in" costs you nothing and starts the relationship well.

7. **Confirm everything in writing**, including anything agreed verbally.
8. **Withdraw your other applications** and tell anyone who helped you. This matters — Chapter 4's ledger is only valuable if you maintain it, and "I got the job, thank you for the help" is the single best message you will ever send from it.

Real Example

A candidate receives an entry-level SOC offer at the bottom of the posted range. She has no competing offer and no leverage in the traditional sense.

She says: *"I'm really excited about this — the detection work is exactly what I've been building toward. I want to accept. Before I do: the range on the posting went up to [X] and I came in at the bottom.

Is there any movement, or if the base is fixed, would you be open to a certification budget and a compensation review at six months instead of twelve?"*

She got the review at six months and the certification budget. Base did not move. At six months she had evidence and got the increase.

Total cost of asking: one paragraph. Total value: several thousand dollars and a precedent that she negotiates professionally.

Case Study: The Rejection That Became an Offer

Rejection handling is part of this chapter for a reason.

A candidate reaches the final round and does not get it. The standard response is to feel bad and disengage. The better response is a short, genuine note:

"Thanks for the update and for the time your team spent with me. I enjoyed the conversation about [specific thing] and I'd be glad to be considered if something else opens up. Either way, good luck with the hire."

Three things happen. The manager remembers you as gracious. You stay in the contact ledger and they stay in yours. And with some regularity, the chosen candidate declines, or a second role opens within a few months — and the gracious near-miss is the first call.

This happens often enough that it should change how you handle every no.

Never burn a bridge. The person who could not hire you today changes companies, gets budget next quarter, or remembers you when a peer asks whether they know anyone.

Reflection

- What is your actual walk-away number, and have you decided it calmly or would you decide it in the moment?
- What matters more to you right now: the base, the learning, or the team?
- What would you regret not asking for?

Journal

Write down what you would say if they said no to everything. Practise the graceful version now, when there is no pressure, so it is available to you when there is.

Exercises

1. Research your target role's compensation in your area using at least three sources. Write down a range and a walk-away number.
2. Write your ask, out loud, in the enthusiasm-first structure. Say it three times.
3. Draft the gracious rejection response now, before you need it.

AI Prompts

- "Here is my offer and the posted range. Suggest two asks that a hiring manager is likely to be able to approve, and one that is likely fixed."
- "Review this negotiation message. Does anything read as entitled, or as a threat to walk?"
- "Draft a short email confirming what we agreed verbally in my offer call, in a way that is professional and not distrustful."

Books

- *Never Split the Difference* — Chris Voss
- *The Psychology of Money* — Morgan Housel. Useful for deciding what the number actually means to you before you negotiate for it.

Checklist

- I know what is negotiable in my situation
- I asked enthusiasm-first with a reason
- Everything material is in writing, including anything agreed verbally
- I accepted or declined graciously

- I told everyone who helped me

Common Mistakes

- Accepting on the call without reviewing
- Leading with the ask instead of the enthusiasm
- Implying a competing offer that does not exist
- Making five asks instead of two
- Accepting anything significant on a verbal agreement
- Disengaging after a rejection instead of closing it well

Key Takeaways

- Enthusiasm before ask; a professional ask does not lose offers
- The whole package is negotiable, and base is often the least flexible part
- An early compensation review converts a no into a scheduled yes
- Handle rejection well and it stays an asset

Next Mission

You are in. Now the ninety days that decide everything after it.

PART

FOUR

After the Yes

The First Ninety Days

MISSION

To convert your first three months into a reputation that carries the rest of your time there.

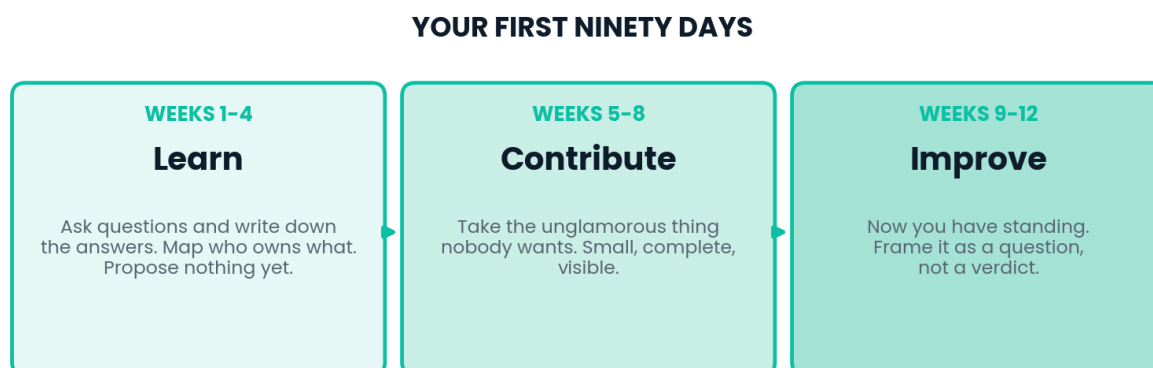
Objectives

- Enter a new role with a deliberate plan
- Build proof of value from week one
- Avoid the specific mistakes new people make

Why It Matters

People form an impression of you in the first month and then spend a year confirming it. That is not fair. It is reliably true. The good news is that the impression is largely within your control, and almost nobody deliberately manages it.

Framework: Learn, Contribute, Improve



Do these out of order and you become the new person with opinions.

The First Ninety Days. The sequence matters as much as the actions.

Weeks 1 to 4 — Learn

Ask a lot of questions and **write down the answers**. Keep a running document. New people ask the same question three times because they did not write it down, and that is what generates the "needs a lot of hand-holding" impression.

- Map who owns what. Not the org chart — who actually owns what.
- Find out how the team really communicates versus how it is supposed to.
- Learn the systems, the naming conventions, the quirks, the things everybody knows and nobody documented.
- Find out what the team is proud of, and what frustrates them.

Do not propose changes yet. Nobody wants the new person's opinion in week two, however correct it is. You have not earned the context to be right, and even when you are right you will be resented for it.

Weeks 5 to 8 — Contribute

Take the unglamorous thing nobody wants.

Documentation that is out of date. A recurring ticket type nobody has root-caused. A runbook that does not exist. The onboarding guide that is two years stale — you just went through onboarding, so you are uniquely qualified and it is genuinely useful.

Small, complete, visible. Complete matters most: a finished small thing builds far more trust than a half-finished ambitious one.

Weeks 9 to 12 — Improve

Now you have standing. Now you can suggest something.

Frame it as a question, not a verdict: *"I noticed X keeps happening. Would it help if I looked at why?"* rather than *"X is broken and here's what we should do."* The first invites collaboration. The second invites defensiveness, especially if the person who built X is in the room.

Framework: Building Provable Value

Two habits, starting week one, that compound for an entire career.

1. Ask for feedback and route it upward.

When you have genuinely helped someone, ask them how it went. When the answer is good, ask whether they would be willing to share that with your supervisor.

This is not manipulation. It is making real work visible to the people who make decisions about you. Most good work dies in the gap between doing it and anyone knowing about it — and that gap is not filled by anyone but you.

The order matters, though. Help because it is the job. Then make the help visible. Doing it in reverse — helping specifically to collect credit — is visible to everyone and it costs more than the credit is worth.

2. Document every success.

Approvals, thank-you emails, completed projects, incidents resolved, things you fixed that nobody asked you to fix.

Keep a running file. Add to it weekly, not annually. At review time you will have evidence while everyone else has recollection. At your next interview, you will have specifics while everyone else has adjectives. And in a layoff conversation, you will have a documented record of value rather than a feeling.

Most people rebuild this from memory the week reviews are due, and it always shows.

Be more valuable than your title. The people who move fastest are almost always doing something their job description does not mention.

Real Example

At a large multi-site employer, I found that our offices had effectively stopped talking to each other. Each location solving the same problems independently, none of them knowing the others had already solved them.

Nobody assigned me to fix that. I built a cross-site communication hub so the sites could actually share what they knew.

At a different site with no structure at all, I introduced regular meetings and real ticketing discipline where there had been none.

Neither of those was in a job description. Both are the reason I was trusted with more. That is the whole mechanism: the work outside your title is what tells people what your next title should be.

Case Study: Two New Hires

Two people start on the same team in the same week.

The first is technically stronger. He identifies three things wrong with the team's approach in his second week and says so, correctly, in a meeting. He is right about all three. Six months later he is regarded as difficult and is not included in planning conversations.

The second asks questions for a month and writes the answers down. In week six she rewrites the onboarding documentation, which is stale and which she just experienced.

In week ten she asks whether it would help if she looked into a recurring alert nobody had root-caused. It turns out to be a misconfiguration, and she fixes it.

She was right about fewer things. She was trusted with more, faster, because she earned the standing before she used it.

Being right is not the same as being effective. The technically stronger person often learns this late, and some never do.

Step-by-Step Guide: Your Ninety-Day Plan

Write this before you start — ideally before the final interview, because bringing it to that interview is remarkably effective.

1. **Weeks 1–4 goals.** Specific: meet every person on the team one-to-one, map the systems, document what I learn, ask about the team's biggest frustration.
2. **Weeks 5–8 goals.** Identify one unglamorous, complete, useful thing and finish it.
3. **Weeks 9–12 goals.** Identify one improvement, frame it as a question, and offer to investigate.
4. **Ongoing from week 1.** Success file started. Feedback asked for and routed. Questions written down.
5. **Day 90 review.** What did I learn, what did I contribute, what do I want the next ninety to look like?

Reflection

- What impression do you want people to have of you at day thirty?
- What is your instinct in a new environment — to prove yourself quickly, or to stay quiet too long? Both are failure modes.
- What unglamorous thing could you volunteer for that would genuinely help?

Journal

Write what you are most worried about in the new role. Then write what you would tell someone else who told you that. Most people are much kinder and more sensible toward others than toward themselves.

Exercises

1. Draft your ninety-day plan for the role you are targeting, before you have the offer.

2. Start the success file today, in your current role or your job search. Backfill the last three months.
3. Write down the three questions you most want answered in your first week.

AI Prompts

- "I am starting as a [role] on a [size] team. Give me fifteen questions I should ask in my first two weeks, organized by who to ask."
- "Here is my ninety-day plan. Tell me which items are actually measurable and which are vague intentions."
- "Help me phrase this improvement suggestion as a question rather than a verdict, without losing the substance."

Books

- *Atomic Habits* — James Clear. The success file and the weekly logging habit are exactly this book's argument applied to a career.
- *Extreme Ownership* — Jocko Willink and Leif Babin
- *The Score Takes Care of Itself* — Bill Walsh. On building a standard of performance rather than chasing outcomes — very relevant to a first ninety days.

Podcast Episodes

Wake Up X episodes on early-career transitions and first roles. Listen for what guests say they would do differently — it is almost always about the first three months.

Checklist

- I have a written ninety-day plan
- I write down the answers to questions I ask
- My success file exists and I add to it weekly
- I have asked for feedback and routed it upward at least once
- I did not propose changes in my first month
- I finished one small, complete, unglamorous thing

Common Mistakes

- Proposing changes in the first month, however correct
- Waiting passively to be given work

- Asking the same question three times because you did not write it down
- Never documenting anything, then having no evidence at review time
- Trying to prove yourself with something ambitious and unfinished

Key Takeaways

- Learn, contribute, then improve — in that order
- Being right is not the same as being effective
- Provable value beats claimed value at every stage of a career
- The work outside your title determines your next title

Next Mission

Book 2 turns the lab that got you hired into a portfolio that keeps promoting you.

Questions for you. What is the unglamorous thing on your current or future team that nobody wants to own? What would happen if you owned it?

Appendix: Templates, Scripts, and the Bullet Bank

REFERENCE

Everything from this book in one place. Copy, adapt, and make it yours.

The Bullet Bank Structure

```
BULLET_BANK.md

## Support
- [production] ...
- [lab] ...

## Networking
## Security
## Cloud
## Automation
## Documentation
## Leadership & Training
## Customer Service
## Projects

Format for every entry:
Action verb + what + scale/tool + what changed
Tag each: [production] or [lab]
```

The Resume Skeleton

NAME
City, ST | phone | email | linkedin | portfolio

[TARGET ROLE TITLE]
Three-line summary aimed at this specific role.

SKILLS
Production: ...
Lab: ...

PROJECTS (high if career-changing)
EXPERIENCE
CERTIFICATIONS & EDUCATION

The Approach Message

Hi [Name] – I saw the [role] opening on your team and I'm applying. I've been [specific relevant thing] and noticed the posting mentions [specific detail]. Curious whether the role leans more toward [X] or [Y]? Either way, appreciate the read.

Under 60 words. Real question. No ask.

Follow-Up One (Days 3–4)

Hi [Name] – following up on my application for [role], submitted [day]. Still very interested – the [specific aspect] is exactly the kind of work I'm looking to do. Happy to provide anything else that would help.

Follow-Up Two (Days 7–10)

Hi [Name] – checking in on the [role] application. Is there a timeline for when you expect to make a decision? Happy to connect you with one or two references who can speak to my work if that's useful at this stage.

The Reconnect Message

Hi [Name] – we spoke back in [month] about [topic].
I've since [specific thing you did]. Saw you moved to
[company] – congrats. Would be glad to stay in touch.

The Gracious Rejection Response

Thanks for the update and for the time your team spent
with me. I enjoyed the conversation about [specific
thing] and I'd be glad to be considered if something
else opens up. Either way, good luck with the hire.

The Negotiation Ask

I'm genuinely excited about this – [specific thing] is
exactly what I'm looking for, and I want to accept.
Before I do: is there any flexibility on [base / the
specific lever]? I'd been targeting [number] based on
[reason]. And if the base is fixed, would you be open
to [certification budget / a six-month review]?

The Contact Ledger Columns

First name | Last name | Phone | Email | Company |
How I know them | Last contact | Notes

The Ninety-Day Plan

WEEKS 1-4 LEARN

- one-to-one with every team member
- map systems and real ownership
- written answers to every question I ask
- propose nothing

WEEKS 5-8 CONTRIBUTE

- one unglamorous, complete, visible thing

WEEKS 9-12 IMPROVE

- one improvement, framed as a question

ONGOING

- success file, weekly
- feedback asked for and routed upward

The Five Interview Questions

1. What does the first ninety days look like for whoever takes this role?
 2. What is the thing this team is most frustrated by right now?
 3. How does this team decide what to work on?
 4. What separates someone who does well here from someone who struggles?
 5. Is this a new role or a backfill?
-

A Closing Note

You now have a system. Whether it works depends entirely on whether you run it, and running it means doing the uncomfortable step. The one where you contact a stranger who owes you nothing.

I want to be honest with you about why I believe you can. I was placed in special education classes as a child and told repeatedly that I could not learn. I did not speak English until around sixteen.

I earned my high school diploma as an adult, almost thirty. I started in IT at thirteen with a donated computer that had no mouse, so I learned the entire machine through keyboard shortcuts because that was the only way in.

Nobody gave me permission and the hardware was terrible. It still worked.

A sixth-grade teacher named Miss England stayed after class and taught me the fundamentals nobody thought I could learn. I have never been able to tell her what it meant.

If you are early, and this feels impossible, that is not evidence that it is. That is what early feels like.

Wake up. Start where you are. Then send the message.

Connect

If this book helped you land something, I want to hear about it — I mean that literally. Find me at [linkedin.com/in/efrenlsalazar](https://www.linkedin.com/in/efrenlsalazar) and tell me your story.

The podcast is at wakeupxpodcast.com.

Questions for the author. Three gaps where your own material would replace my framing: (1) your actual first ninety days in an IT role — what you got wrong, and the moment you realized doing more than your title was working; (2) a real negotiation you handled, and what you asked for; (3) a specific person you helped break in, and where they ended up. Chapter 7 in particular is currently carrying general framing where a story of yours would land harder.

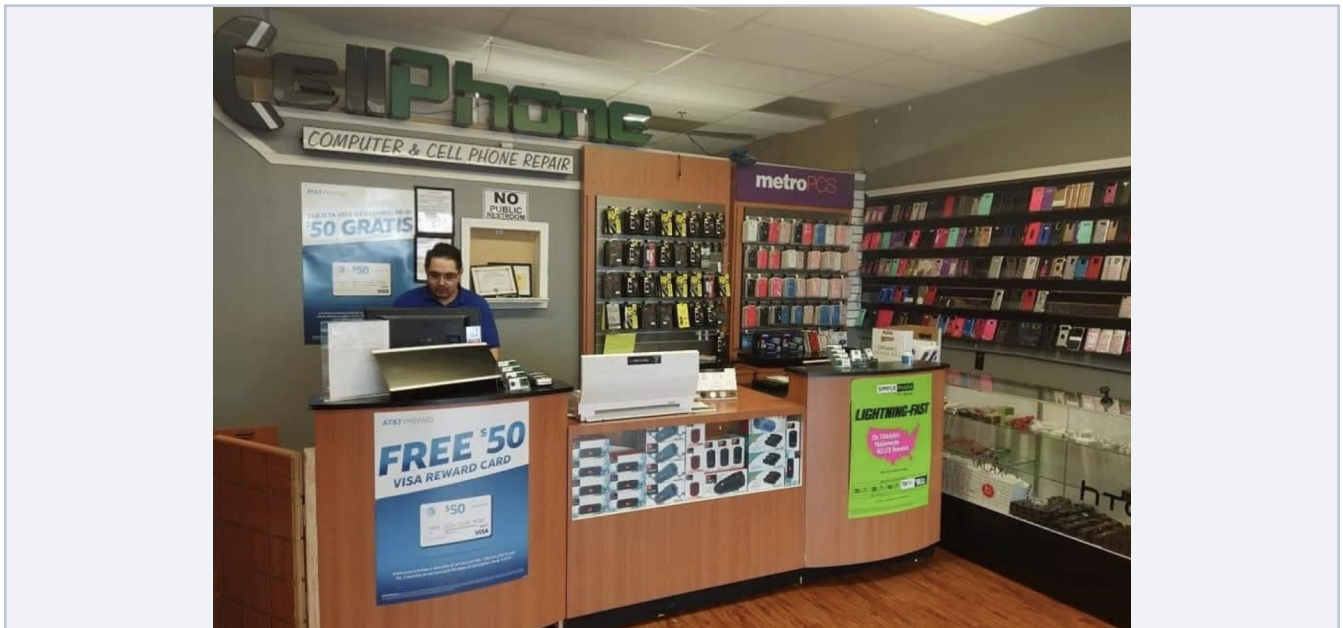
IN THE FIELD

Where It Started

Before the podcast and the frameworks, there was a repair counter. Real customers, real problems, real reps.



The Cellphone Fix, Computer and Cellphone Repair — storefront

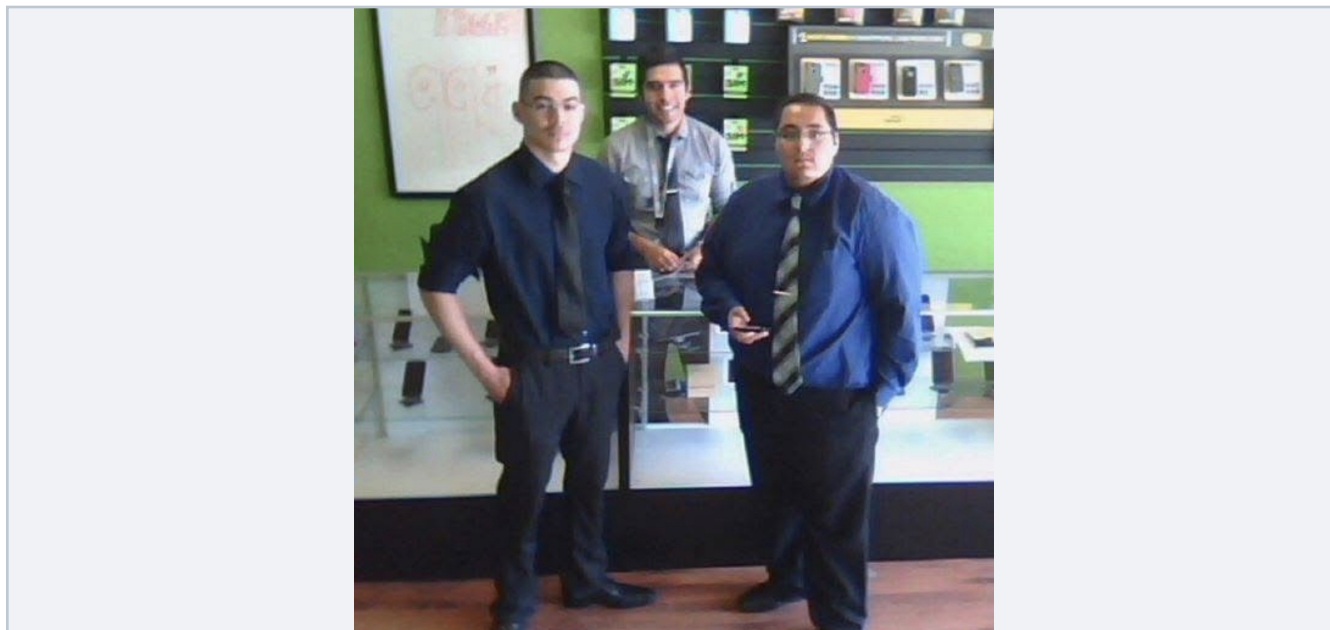


Behind the counter, helping a walk-in customer
thecellphonefix.com

IN THE FIELD

Co-Founder, Before the Title Existed

Co-founder and CTO of The Cellphone Fix, Computer and Cellphone Repair — years before "CTO" was ever on a business card.



Early days, with the store team



Running the floor at The Cellphone Fix

Before The Cellphone Fix, There Was Compu-Fix 2u

The Cellphone Fix wasn't the first business. Years earlier, a mobile setup called Compu-Fix 2u ran computer and Mac repairs, virus removal, and free diagnostics out of borrowed space — including a folding cart set up inside a hair salon in West Sacramento. No storefront, no funding, just a phone number, a laptop cart, and a habit of showing up wherever a client would let him work.

Why this matters more than the shop that came after it

It's easy to look at a real storefront and assume that's where the story started. It didn't. Compu-Fix 2u is proof the business came first, and the location came later — the same order this book asks you to work in. Build the capability and the reputation before you have the office, the title, or the budget to make it official.

KEY TAKEAWAY

The business is the skill plus the willingness to show up. Everything else — the storefront, the brand, the team — gets built on top of that, not before it.

Louis Kemp — Terminology First, Everything Else Follows

Louis Kemp is CTO of ANUL Technologies, a cybersecurity consulting firm he started himself, and a partner with Horizon3 on autonomous penetration testing. His path there started at the very bottom of a help desk, with a help desk degree — not a cybersecurity one.

Four to Five Years, Not a Weekend Bootcamp

He spent years as a level-1 analyst, then level-2, before certifying with ISC2's Certified in Cybersecurity and actually moving into the field. The full transition from help desk to cybersecurity took him roughly four to five years — and he's specific that COVID's shift to remote work is what freed up the study time that finally accelerated it.

Terminology Is the Real First Wall

His clearest advice for anyone trying to skip straight to cybersecurity: don't. "If you have a basis and a foundation, then you know this relates to this, that relates to that — you can put two and two together." Without foundational IT terms, cybersecurity language is just unfamiliar words with nothing to anchor them to.

Stability and Passion, Not Either/Or

With a young son and a retail background before IT, his stated motivation for the career change was wanting something that "moves with the economy" and could carry him to retirement — paired directly with genuine interest in the work, not framed as a tradeoff between the two.

Watch the full episode: youtu.be/0KT3Vdgdcci

Louis's Advice, In Practice

Don't skip the foundational IT terminology to rush into cybersecurity.

The words themselves are the first barrier — build the basic vocabulary before trying to layer security concepts on top of it.

Half your business can be the service, half can be giving back.

He built mentorship into ANUL Technologies' model from day one, specifically for early-career students — not as marketing, but because community mentorship is what moved his own career forward.

Annual pen testing is already becoming outdated.

His read on where the industry is headed: continuous, autonomous testing is replacing the once-a-year model — a direct, practical example of why standing still in any specialization is risky.

Career stability and genuine passion aren't in competition.

He chose cybersecurity consulting because it could support his family long-term and because he actually wanted to do the work — both reasons held equal weight in his decision.

REFLECTION

Are there foundational terms or concepts in your own field you've been skipping past, hoping the advanced material will make sense without them anyway?

BE OUR NEXT GUEST

Your Story Belongs on Wake Up X

If Break In gave you something worth using, you already have something worth sharing. The Wake Up X podcast exists to put real, unpolished career and business stories in front of people who need to hear them — and every guest so far started exactly where you are now: reading, building, and figuring it out in public.

You might be a fit if you:

Broke into tech, cybersecurity, or business without the traditional path

Built something — a lab, a business, a portfolio, a community — and can talk about how

Have a failure or turning point you're willing to tell straight, not polished

Are mentoring, leading, or building something worth other people knowing about

HOW TO APPLY

Share your story at wakeupxpodcast.com/guest, or connect directly on LinkedIn at linkedin.com/in/efrenlsalazar and mention this book.

Being a guest is also visibility — the same principle Book 7 makes the case for.

Acknowledgments

THE PEOPLE WHO MADE THIS POSSIBLE

None of Wake Up X happened alone, including Break In. Every book in this library, the podcast, and the free library exist because specific people showed up for a kid nobody expected much from. This is the record of who they are.

THE EARLY YEARS

Miss England, the 6th-grade teacher who stayed after class, unpaid, to teach the fundamentals. Sister Cora and Lewis Bair of Computers for Kids, who handed a teenager a key and a broken machine and let the work speak. Jose Luis Bravo, Angel Martinez, Isaac Murrilo, and Alex De Haro — the friends and cousins who translated the world, gave hope, drove him where he needed to go, and carried him before there was anything to carry him toward.

THE BUSINESS YEARS

Rojelio Salazar, his cousin and first mentor, RNS Computer Repairs — the reason it is called Wake Up X. Kyle Erni, his first business partner. Erick and Pancho Arauza, co-founders of The Cellphone Fix. Danny Maya, of Maya Traditional Mexican Restaurant, Taqueria Maya, and Salt and Water, who pushed him in front of a camera for the first time.

THE CAREER YEARS

Igor, who gave him the vocabulary for work he already knew how to do. Nate, who he mentored into a full-time role alongside his own. Ken Kerrick, opponent turned friend in the ISACA Sacramento election. Alvin Jue and David De los Reyes, who believed in the middle of it, not just at the finish.

THE PODCAST

Every guest who trusted an unproven show with their story, including Gustavo Mastroianni, Cathy Stephenson, Greg McCord, Aaron Portnoy, Sana Talwar, Jorge Avila, Anil Reddy, Devin Prasad, Larry Switzer, Louis Kemp, Mohana Balakrishnan, Eddie Lopez, and Michael Fierros — full episodes at wakeupxpodcast.com.

COMMUNITY PARTNERS

ISACA Sacramento and ISC2 Sacramento, The Cellphone Fix, A.N.U.L Technologies, AISOC, AspireByte, and PodGrabber — the organizations building alongside this movement.

AND

His parents, who arrived with nothing, worked since childhood, and gave away whatever was left over. Nothing about the first version of this story predicted the version you are holding now.

Continue the Journey

EVERY BOOK LEADS SOMEWHERE ELSE

Break In is one part of a 12-book system. If it helped, the next step is one page away — take whichever layer matches where you are standing.

THE LIBRARY

Layer 1 — The Story

I Just Have To Wake Up — why any of this exists.

Layer 2 — The Mindset

Comfort Is Expensive — the real cost of playing it safe.

Layer 3 — The Blueprint

DECODE: The Wake Up X Playbook — zero experience to executive thinking. Book 0.

Layer 4 — The Playbooks

1 Break In · 2 Build It · 3 Think Like a Cybersecurity Professional · 4 Business First · 5 Lead Without a Title · 6 AI for IT & Cybersecurity · 7 Build Your Brand · 8 The Consultant Mindset · 9 Level Up

FIND US EVERYWHERE

Podcast & free library: wakeupxpodcast.com · Spotify: open.spotify.com/show/28GZnRBrIV2efHOHzdiftt · Apple Podcasts: podcasts.apple.com/us/podcast/id1792821643 · YouTube: youtube.com/@Efren.wakeupx · LinkedIn: linkedin.com/in/efrenIsalazar · Substack: substack.com/@wakeupxefren

IN GOOD COMPANY

ISACA Sacramento, ISC2 Sacramento, The Cellphone Fix, A.N.U.L Technologies, AISOC, AspireByte, and PodGrabber — the partners and collaborators building alongside the movement. See the full list at wakeupxpodcast.com/community.html.

Get DECODE free, apply to be a guest, or read the story behind the movement — all of it starts at wakeupxpodcast.com.

WAKE UP X

SOLUTIONS



Efren L. Salazar

FOUNDER · HOST · CYBERSECURITY PROFESSIONAL · EDUCATOR · COMMUNITY BUILDER

STAY CONNECTED

Podcast & Blueprint Library:	wakeupxpodcast.com
LinkedIn:	linkedin.com/in/efrenlsalazar
YouTube:	youtube.com/@Efren.wakeupx
Be a Guest:	wakeupxpodcast.com/guest

THE WAKE UP X PUBLISHING SYSTEM

BOOK 2

Build It

Home labs and a portfolio that prove you can do the work

Nobody believes a resume. Everybody believes a build.

THE
WAKEUPX
SOLUTIONS

EFREN L. SALAZAR

Founder & Host, Wake Up X | IT Operations Manager & Solution Engineer
Government & Regulatory Partner, ISACA Sacramento | CompTIA Security+

wakeupxpodcast.com

Educational content — not professional, legal, financial, or career-placement advice.
Full disclaimer in DECODE, the foundation of this library.

Before You Begin

HOW TO USE THIS BOOK

This book has one job: to turn "I am studying for a certification" into "here is a thing I built, here is how it broke, and here is what it taught me." By the end you should have something a stranger can look at and understand what you can actually do.

Where this book sits

Two books come first — the memoir I Just Have To Wake Up and the mindset book Comfort Is Expensive — before this ten-book library. Each one is short on purpose. Pick up the one that matches the problem in front of you right now.

- n **Book 0 — DECODE.** The foundation. The map of the whole path.
- n **Book 1 — Break In.** Landing your first role in IT or cybersecurity.
- n **Book 2 — Build It.** Home labs and a portfolio that prove you can do the work. **you are here**
- n **Book 3 — Think Like a Cybersecurity Professional.** Detection, triage, and incident response thinking.
- n **Book 4 — Business First.** Finance, sales, GRC, and why the business hired you.
- n **Book 5 — Lead Without a Title.** Influence, EQ, and mentoring before anyone promotes you.
- n **Book 6 — AI for IT & Cybersecurity.** Using AI as an actual working tool, not a novelty.
- n **Book 7 — Build Your Brand.** LinkedIn, content, and being findable.
- n **Book 8 — The Consultant Mindset.** Discovery, solution selling, and stakeholder management.
- n **Book 9 — Level Up.** Long-term growth, burnout, legacy, and mentoring others.

Book 0 is the foundation. *DECODE: The Wake Up X Playbook* holds the six-level path, the Curiosity Compass, and the core frameworks. Nothing in any other book contradicts it. If a term here is unfamiliar, DECODE defines it.

What is in this book

Part One — Why Building Beats Studying

1. Nobody Believes a Résumé. Everybody Believes a Build.
2. What to Actually Buy, and When

Part Two — The Nine Projects

3. The Build Order
4. Breaking Things on Purpose

Part Three — Making It Count

5. The Portfolio That Does the Talking
6. From Project to Resume Bullet to Interview Answer
7. Appendix: READMEs, Logs, and the Buy Order

How to actually use it

This book is unusually action-heavy, because a book about building that you only read is a contradiction. Every chapter ends with a checklist and a project. The reading is the setup; the building is the point.

Work one project at a time and **finish it before starting the next**. Nine half-built projects is worth less than one finished and documented one. Much less, because unfinished work teaches you nothing about the hard part, which is always the last twenty percent.

A warning about scope. The most common way this book fails a reader is that they read the project list, get excited, buy hardware, start four things, and finish none. If you take one instruction from this page: pick one, finish it, document it, then pick the next.

I am writing this the way I would say it to you sitting across a table — not lecturing a room. Take what works, remix the rest into your own version, and leave what does not fit.

PART

ONE

Why Building Beats Studying

Nobody Believes a Résumé. Everybody Believes a Build.

MISSION

To convince you that the lab is your experience section, not your hobby section.

Objectives

- Understand why a lab outperforms a certificate in a hiring conversation
- Stop waiting for permission, better hardware, or more knowledge
- Adopt the loop that turns a tutorial into a competency

Story

When I was thirteen, someone gave me a donated Windows PC. It had no mouse. I did not know that was unusual, so I learned the machine entirely through keyboard shortcuts — because that was the only way in.

English was not my first language yet. I was reverse-engineering an operating system and a language at the same time, with nobody to ask, from a farm-worker family with no technology background and no one to call for help.

Nobody assigned that. There was no curriculum, no course, no certification path. I broke things constantly and fixed most of them, and the ones I could not fix taught me more than the ones I could.

Within a couple of years, people in the neighborhood were bringing me their computers and phones, and I was fixing them despite barely being able to explain in English what I had done.

That is the entire argument of this book, and I want to be clear about why I am telling you this. It is not a motivational story.

It is a structural point: **nobody gave me permission and the hardware was terrible. It still worked.** The two things you are most likely using as reasons to wait are the two things that were most absent when this started.

Why It Matters

A certification proves you passed a test on a specific day. A lab proves you can sit with something broken until it works.

Those are different claims, and hiring managers know exactly which one predicts job performance. I have interviewed people with impressive certification stacks who could not explain what DNS does when it fails.

I have interviewed people with no certifications at all who could walk me through a network they had built, what they had gotten wrong, and how they had figured it out. I know which one I wanted on my team.

More practically: if you are a career changer, the lab is the only directly relevant work experience you have. It is not a hobby. It is your experience section, and Book 1 argued it belongs high on the page.

Professional Explanation

Business Perspective

An employer is buying a reduction in their own workload. Your lab is the cheapest possible demonstration that you will reduce it rather than add to it.

It also signals something managers value more than raw skill: that you will keep learning without being managed into it. Every manager has had the direct report who stops growing the moment nobody is pushing. Someone who built a lab on their own time, unpaid, unassigned, is unlikely to be that person.

There is a cost dimension too. A new hire is unproductive for a period — typically weeks to months. Anything that shortens that ramp is worth real money. A candidate who has already stood up Active Directory in a lab is not learning what a domain controller is on company time.

Technical Perspective

Labs teach the thing courses cannot: **integration**.

Any tutorial can show you how to stand up a domain controller. Only a lab teaches you what happens when DNS is misconfigured and suddenly nothing authenticates and you have to figure out *why* at eleven at night with nobody to ask.

That debugging loop — form a hypothesis, test it, be wrong, form a better one — is the actual job. Everything else is vocabulary.

Labs also teach you the shape of failure. In a course, things work. In a lab, things fail in ways the course never mentioned, and you learn that most problems are not exotic.

They are a typo, a firewall rule, a service that did not start, a certificate that expired, or DNS. It is very often DNS.

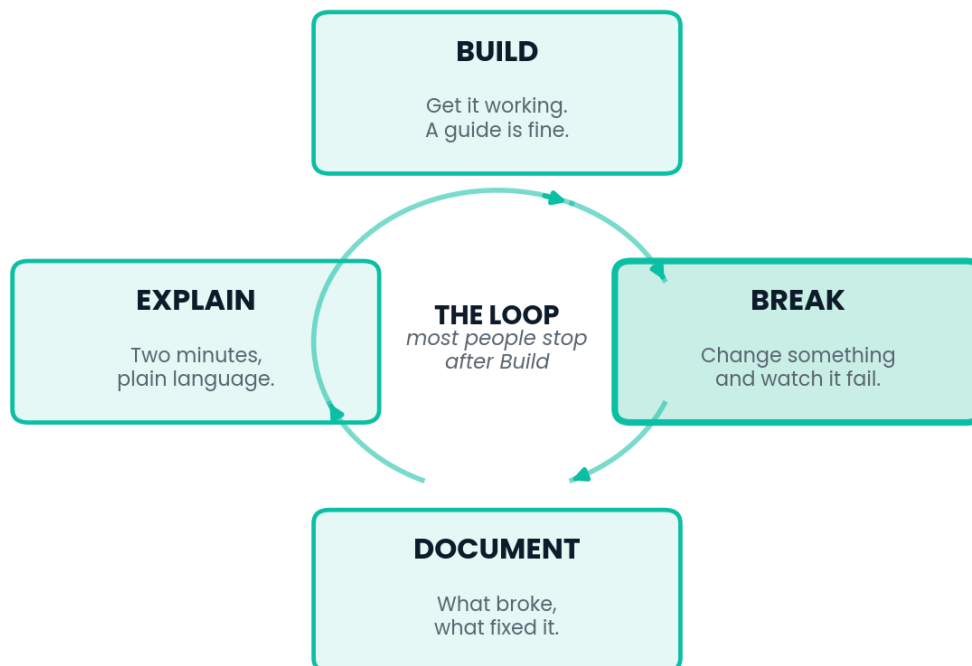
Leadership Perspective

Documenting your lab is a leadership skill wearing technical clothes.

The engineer who can explain what they built, why they chose it, and what they would do differently is the one who eventually gets handed a team. Not because documentation is glamorous, but because it is evidence of a mind that thinks in terms of other people needing to understand.

The rule. If it is not documented, it did not happen. An undocumented lab helps you learn. A documented lab helps you get hired. Same work, different outcome — and the difference is maybe forty minutes per project.

Framework: Build, Break, Document, Explain



The gap between Build and Explain is the gap between a tutorial and a job.

The loop. Most people stop after Build, then wonder why the tutorial did not make them employable.

Every project runs the same four-step loop.

1 — Build. Get it working. Follow a guide if you need to; there is no shame in that on the first pass. The goal here is a working baseline, not understanding.

2 — Break. Now change something and watch it fail. Turn off DNS. Misconfigure a firewall rule. Pull a disk. Delete a service account. This is where the learning actually is, and almost nobody does it — which is exactly why Chapter 4 is dedicated to it.

3 — Document. Write down what you built, what broke, what fixed it, and what you would do differently. Screenshots. A short README. Forty minutes.

4 — Explain. Say it out loud in under two minutes, in plain language, to someone non-technical. If you cannot, you do not understand it yet — and you will find that out in an interview instead of at home.

Most people stop after step one. The gap between step one and step four is exactly the gap between "I did a tutorial" and "I can do this job."

Real Example

Two people complete the same Active Directory tutorial in the same week.

The first follows it exactly, ends with a working domain controller, screenshots it, and moves on to the next tutorial. Three months later, asked in an interview what happens when a domain controller loses DNS, he says he is not sure.

The second follows the same tutorial, then deliberately breaks DNS to see what happens. Authentication fails in a way she did not expect. She spends two hours understanding why, discovers that domain-joined machines cache credentials and that the failure mode is stranger than she assumed, writes it down, and moves on.

Same tutorial. Same afternoon of initial work. Two hours of difference. In an interview, the second person has a story and the first has a screenshot.

Case Study: The Lab That Was Actually a Job

A pattern I have seen enough times to name it.

Someone builds a home lab for genuine personal interest, not strategically. Over six months it grows: virtualization, then a domain controller, then network segmentation because the IoT devices were bothering them, then a SIEM because they wanted to see what was happening on their own network.

They apply for a junior role. In the interview, the manager asks about the segmentation. The candidate explains their VLAN design, why they isolated the IoT devices, and the mistake they

made the first time. Putting the management interface on the wrong VLAN and locking themselves out of their own switch.

That last detail is what closed it. Nobody who read about VLANs makes that specific mistake. Only someone who actually did it does. The mistake was the credential.

This is worth internalizing: **the errors you make are as valuable as the systems you build**, provided you record them.

Common Mistakes

- **Waiting for better hardware.** An old laptop and free software is enough to start, and Chapter 2 makes the case in detail.
- **Building only what the tutorial says.** Without the break step, you learned to follow directions, which is not a scarce skill.
- **Never writing anything down.** Six months later you will not remember how you fixed it. I promise you this.
- **Building nine projects and finishing none.** One finished and documented beats five abandoned, by a wide margin.
- **Treating the lab as preparation for real work** rather than as real work. It is the work. Present it that way.

Reflection

- What have you been telling yourself you need before you can start?
- Is that thing genuinely blocking, or is it permission you are waiting for?
- When did you last deliberately break something to see what happened?

Journal

Write about the last technical thing you figured out on your own without a guide. What did the process actually feel like? That feeling — frustrated, then stubborn, then satisfied — is the job. If you like it, you are in the right field.

Exercises

1. Pick the smallest project you can genuinely finish this week. Not the impressive one — the finishable one. Momentum is the scarcest resource in this entire process.
2. Write down, right now, what you own that could host a virtual machine. Most people discover they already have enough.

Project

Take something you have already built or configured — anything, however small, including something at work — and run all four loop steps on it retroactively. Break it in a safe copy, document it, and explain it out loud to someone. You will learn more in that hour than in the original build.

AI Prompts

- "I have [specific hardware]. What is the most useful lab project I could start with today, given those constraints? Do not suggest anything requiring hardware I did not list."
- "I just built [project]. Suggest five specific ways I could deliberately break it to understand its failure modes, and what each would teach me."
- "Explain [concept] as if I have built it once but do not understand why it works."

Books

- *So Good They Can't Ignore You* — Cal Newport. The case that rare, valuable skills come from deliberate practice rather than passion.
- *The Power of Habit* — Charles Duhigg. Relevant to finishing projects rather than starting them.
- *Atomic Habits* — James Clear. Same argument, more tactical.

Podcast Episodes

Wake Up X episodes with guests who came in without a traditional path. Listen for what they actually built, and notice how often the entry point was something they made rather than something they studied.

Resources

- **NetworkChuck** — practical, project-oriented, and good at making the first step feel possible
- **John Hammond** — for when you want to understand how things break
- **David Bombal** — networking depth without assuming a formal background

Checklist

- I have accepted that the lab is experience, not a hobby
- I know the four-step loop and I can name where I usually stop

- I have picked one small, finishable first project
- I have stopped waiting for hardware I do not have

Key Takeaways

- A certification proves a test; a lab proves persistence
- Integration and failure modes are what labs teach and courses cannot
- Build, Break, Document, Explain — and most people stop at Build
- The mistakes you record are as valuable as the systems you finish

Next Mission

What to actually buy, and — more importantly — when not to.

Questions for you. What is the smallest thing you could finish this week? And what have you been calling "not ready" that is actually "not started"?

What to Actually Buy, and When

MISSION

To stop you spending money you do not need to spend, and to tell you exactly when spending it is worth it.

Objectives

- Start with zero budget and stay there as long as possible
- Know which buy unlocks which capability
- Avoid the expensive mistakes people make early

Why It Matters

Two failure modes, opposite directions, both common.

The first: never starting because you believe you need equipment. This costs you months.

The second: buying an enterprise server rack off an auction site in week one, discovering it sounds like a jet engine and triples your power bill, turning it off, and never building anything. This costs you money *and* months.

The correct approach is boring: spend nothing until the free option is genuinely the limiting factor, then spend the minimum that removes the limit.

Framework: The Five Stages

Stage 1 — Free

Your existing computer. VirtualBox or Hyper-V. Free tiers of cloud providers. Community editions.

Spend zero dollars until you have finished two projects. Not started — finished and documented. This rule exists because purchasing feels like progress and is not, and because the two-project threshold reliably separates people who will actually do this from people who enjoyed the idea of it.

Most modern laptops with 8GB of RAM will run two small virtual machines. That is enough for projects 1, 2, and 3 on the build list.

Stage 2 — The Mini-PC

A used small-form-factor desktop or refurbished mini-PC. Target 16GB of RAM and an SSD.

This is the single highest-value buy in the entire progression, and on the used market it is inexpensive — often less than a certification exam. It becomes your always-on lab host, which changes everything: services that run continuously behave differently from services you start when you feel like practising.

Look for: an Intel or AMD chip from the last several generations, 16GB minimum (32GB if the price difference is small), an SSD, and low power draw. Brands to search: Dell OptiPlex Micro, HP EliteDesk Mini, Lenovo ThinkCentre Tiny. These come off corporate leases constantly and are built to run continuously.

Stage 3 — Networking

A managed switch and a router capable of VLANs.

This unlocks project 4 and, downstream, everything security-flavored. Until you have this, network segmentation is theoretical.

You do not need enterprise gear. A small managed switch and a capable router are enough, and the concepts transfer completely. VLAN configuration on a small managed switch is the same mental model as VLAN configuration on a large one.

Stage 4 — Storage

A NAS, or simply a second drive dedicated to backups, once you have built something you would be genuinely annoyed to lose.

The value here is not storage capacity. It is that you learn backup and restore, which is one of the most consistently under-practised skills in the industry. Almost every organization has backups. A meaningful number have never verified they can restore from them.

Stage 5 — More RAM

RAM, not CPU, is what limits home labs. Almost always. When you hit a wall, check RAM first.

Do not buy a rack. Do not buy enterprise servers off auction sites to start. They are loud, they consume significant power, they generate heat, and they impress nobody who is actually hiring. A quiet mini-PC that runs twenty-four hours a day beats a rack you turn off because of the noise. I have watched people spend hundreds of dollars to end up with less capability than a refurbished office desktop.

Templates: The Purchase Decision Test

Before any buy, answer these four:

1. **What specific project is this unblocking?** If you cannot name one, stop.

2. **Have I finished the projects I can do without it?** If not, stop.
3. **Is there a free or cheaper way to learn the same concept?** Cloud free tiers cover a surprising amount.
4. **Will this still be useful in a year?** RAM and storage age well. Specialized appliances usually do not.

If you cannot pass all four, you are shopping rather than building. That is a real and common thing, and it feels productive.

Real Example

A reader with a five-year-old laptop and no budget believed he could not start.

What he actually had: enough RAM for two small virtual machines, a home router, and an internet connection. He completed projects 1, 2, and 3 — a domain controller, virtualization with snapshots, and Pi-hole — over about ten weeks, spending nothing.

He then bought a refurbished mini-PC for roughly the cost of a certification voucher, because his laptop could no longer stay on continuously without interfering with his actual work. That buy was justified by a specific limitation he had personally hit.

Three months in, he had three documented projects and one piece of hardware. The alternative version of this person buys the hardware first, feels good for a week, and has zero documented projects three months later.

Case Study: What Employers Actually Notice

Worth being blunt about something.

No hiring manager has ever been impressed by your hardware. Nobody has been hired because they owned a rack. What gets noticed is the reasoning: why you segmented the network the way you did, why you chose one tool over another, what you did when it did not work.

That reasoning is completely independent of budget. A well-reasoned three-VLAN design on inexpensive gear demonstrates exactly the same competency as one on expensive gear, and if you built it on a constraint, you also showed working within constraints — which is what every real job is.

The one exception: if the role is specifically hardware-focused, hands-on experience with the relevant class of equipment matters. Read the posting.

Step-by-Step Guide: Your First Month, Zero Budget

1. **Week 1.** Install VirtualBox or enable Hyper-V. Create one virtual machine. Take a snapshot. Break the machine. Restore the snapshot. You have now learned the most useful lab habit there is.
2. **Week 2.** Install Windows Server as a VM. Promote it to a domain controller. Join one client machine to the domain.
3. **Week 3.** Create Group Policy objects. Enforce a password policy and a lockout policy. Watch them apply.
4. **Week 4.** Break DNS deliberately. Observe what fails. Fix it. Document all of it.

At the end of that month you have spent nothing and you have a project worth talking about in an interview. Then, and only then, consider the mini-PC.

Reflection

- Have you been shopping instead of building?
- What is the actual, specific limitation you have hit — or have you not hit one yet?
- What could you finish this month with only what you already own?

Journal

Write down what you think you need to buy, and next to each item, the specific project it unblocks. Most people find at least half the list has no project attached to it.

Exercises

1. Inventory what you already own that could host a VM. Include old laptops, a spare desktop, anything.
2. Apply the four-question buy test to the thing you most want to buy right now. Be honest about the answers.

AI Prompts

- "Here is my current hardware: [list]. What lab projects can I complete without buying anything? Be specific about the limits I will hit."
- "I am considering buying [item] for a home lab. What specific capability does it unlock, and is there a free or cheaper way to learn the same concept?"
- "Compare refurbished mini-PC options in the [price] range for a home lab running 3-4 VMs. Tell me what specifications actually matter and what does not."

Books

-
- *The Psychology of Money* — Morgan Housel. Genuinely relevant: most lab overspending is emotional rather than technical.

Resources

- Refurbished corporate hardware sellers — the mini-PC market is where the value is
- Cloud provider free tiers, which cover more lab scenarios than most people realize
- Your workplace's decommissioned equipment, if the policy allows asking

Checklist

- I have finished two projects before spending anything
- Every buy I am considering unblocks a named project
- I have not bought a rack
- I know what my actual limiting factor is

Common Mistakes

- Buying before finishing anything free
- Enterprise gear for a residential space
- Optimizing CPU when RAM is the constraint
- Treating purchases as progress

Key Takeaways

- Spend nothing until two projects are finished and documented
- The refurbished mini-PC is the highest-value buy available
- RAM is nearly always the limit, not CPU
- No employer has ever been impressed by your hardware

Next Mission

The nine projects, in the order that makes sense.

PART

TWO

The Nine Projects

The Build Order

MISSION

To give you a concrete sequence instead of an overwhelming list of possibilities.

Objectives

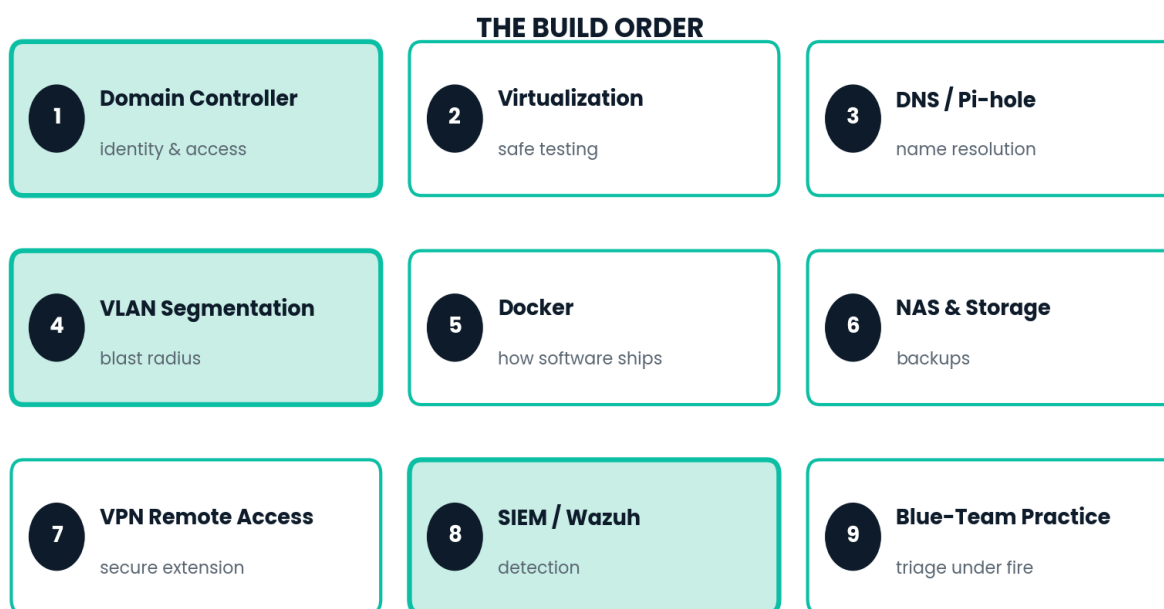
- Know which project to start with based on your lane
- Understand what each project proves to an employer
- Sequence them so each one builds on the last

Why It Matters

The most common reason people stall is not difficulty. It is choice paralysis — a list of forty possible projects, no obvious starting point, and a vague sense that they should be doing something more advanced than whatever they are considering.

This chapter removes the choice. Start at the number that matches your lane, go in order, finish each one.

Framework: The Nine Projects



Highlighted projects are the three that most change how employers see you.

The build order. Each project assumes the one before it.

1 — Home Server and Domain Controller. Windows Server, Active Directory, DNS, Group Policy.

Proves: you understand how corporate identity and access actually work. This is the single most transferable project on the list, because nearly every organization runs some version of it. If you build only one thing, build this.

2 — Virtualization and Sandbox Lab. Hyper-V, VirtualBox, or VMware. Snapshots, isolated test machines.

Proves: you can test safely and roll back — the foundation of everything else here. In practice you will build this alongside project 1 rather than after it.

3 — Ad-Blocking DNS. Pi-hole.

Proves: you understand DNS at a working level rather than a flashcard level, and can observe network-wide effects of a single service. It is also the project most likely to make your household notice, which is oddly motivating.

4 — Network Segmentation and VLAN Design. A managed switch and a VLAN-capable router. Isolate IoT from workstations from lab machines.

Proves: you think in terms of blast radius. This is the project that makes security people take you seriously. It is the one most candidates skip because networking feels less exciting than

security tools.

5 — Containerization. Docker. Images versus containers, persistent volumes, networking between containers.

Proves: you are current with how software actually ships now. Increasingly assumed rather than impressive — which is exactly why its absence is noticed.

6 — NAS and Storage. Network storage plus a real backup routine.

Proves: you understand storage, backup, and — critically — restore. Practise the restore. Everyone practises the backup.

7 — VPN and Remote Access. Tailscale or WireGuard.

Proves: you can extend a network securely. Directly relevant to every hybrid workplace, which is most of them.

8 — SIEM Monitoring. Wazuh. Collect logs, build alerts, tune out the noise.

Proves: you can do the core work of a SOC analyst. This is the project that gets blue-team interviews, and Book 3 builds directly on top of it.

9 — Blue-Team Training. LetsDefend, TryHackMe, or similar.

Proves: you have practised triage and response against real scenarios rather than only building infrastructure. Different skill from everything above.

The Tenth: Automation

Add **n8n** or equivalent on top of any of the above. Automate something small and real:

- An alert arrives, gets summarized, and posts to a channel
- A weekly report generates and sends itself
- A backup completes, gets verified, and logs the verification

Automation is the loudest signal available that you think like an engineer rather than a technician, and almost no entry-level candidate has it. If you want one differentiator from this entire book, it is this.

Framework: The Three Career Currents

Before you pick a lane, pick a current.

The Curiosity Compass in DECODE gives you six lanes: Fix, Build, Break, Defend, Develop, Translate. That is useful, and it is one level too detailed for someone just starting. Most people find it easier to answer a bigger question first.

THE THREE CAREER CURRENTS

Pick the current first. Then pick your lane inside it.



You can move between currents later. Most people do. Start with the one that sounds like you today.

The Three Career Currents. Pick the current first. Then pick your lane inside it.

1 — Engineer. You like to tinker and build. You would rather make the thing work than write about it. Your lanes are Fix, Build, and Develop.

2 — Analyst. You like to dig into things, document them, and explain them to other people. You notice patterns. Your lanes are Break, Defend, and Translate.

3 — Leader. You like growing people and running things. This one is different from the other two, because it sits on top of a lane rather than replacing it. You still need a technical home. You just get your energy from the people side.

Why this order helps. "Do I want to build things or figure things out?" is a question you can answer today. "Am I a detection engineer or a GRC analyst?" is not. Start with the current, then narrow to the lane, then pick the project.

And you are not locked in. I started as an Engineer, moved into Analyst work, and ended up in Leadership without ever planning any of it. Most people move at least once. Pick the one that sounds like you right now, and let it change later.

Step-by-Step Guide: Which One First?

Match it to your lane from the Curiosity Compass in DECODE.

Your lane	Start here	Then
Fix	1 — Domain Controller	2, then 3
Build	5 — Docker	2, then the automation project
Break	4 — VLAN Segmentation	9, then 8
Defend	8 — SIEM	9, then 4
Develop	5 — Docker	automation, then 2
Translate	1 — Domain Controller	then document it well enough for a non-technical reader — that documentation is your portfolio piece

If you do not know your lane yet, start at project 1 regardless. It is the most broadly useful and it will teach you enough to have an opinion.

Real Example

A career changer targeting help desk work started at project 1 and stopped there for two months — deliberately.

Rather than rushing to the next project, she built the domain controller, then rebuilt it from scratch without the guide, then broke it four different ways: DNS misconfigured, time skew introduced, a service account disabled, Group Policy misapplied.

Two months, one project. In her interview she was asked a routine Active Directory question and gave an answer with four failure modes in it. She was offered the role.

Depth on one project beat breadth across six. This is true more often than people expect, particularly early.

Case Study: The Project That Gets Skipped

Project 4 — network segmentation — is skipped more than any other. It is the one that most changes how you are perceived.

The reason it gets skipped is that it needs hardware, and it feels like plumbing rather than security. The reason it matters is that segmentation is how organizations limit damage, and someone who has designed a segmented network thinks about blast radius by default.

In interviews, the question "how would you limit the damage if a workstation were compromised?" separates candidates almost perfectly. People who have built VLANs answer by design. People who have not answer with tool names.

If you can only afford one hardware buy beyond the mini-PC, make it the managed switch.

Templates: The Project Log

Keep one running file with an entry per project.

```
PROJECT:      name
STARTED:      date
STATUS:       planning / building / breaking / documented
GOAL:         what I'm trying to understand
HARDWARE:     what it runs on
WHAT WORKED:
WHAT BROKE:
TIME SPENT:
RESUME BULLETS PRODUCED:
```

That last line matters more than it looks. Every project should produce at least two resume bullets, and writing them at the time is far easier than reconstructing them a year later.

Reflection

- Which project are you avoiding, and is the reason technical or emotional?
- Are you drawn to breadth because it feels like progress?
- What would two months on one project actually teach you?

Journal

Write down what you want to be able to say in an interview twelve months from now. Then work backwards: which of these nine projects produces that sentence?

Exercises

1. Pick your lane and your starting project. Write the date next to it.
2. Set up the project log file today.
3. Block two hours in your calendar this week. Not "when I have time" — a scheduled block.

Project

Complete project 1 or your lane's starting project end to end, including the break step and a written README. One project, fully finished, is the assignment for this chapter.

AI Prompts

- "I want to build [project] with [hardware]. Give me a build plan with the decisions I need to make, not a step-by-step tutorial — I want to understand the choices."
- "I have finished [project]. What is the natural next project that builds on what I now understand, and why?"
- "What does a hiring manager for a [role] want to hear about a home lab? Be specific about what impresses and what does not."

Books

- *So Good They Can't Ignore You* — Cal Newport
- *The Phoenix Project* — Gene Kim. Fiction, and the best explanation of why infrastructure work matters to a business that exists.

Podcast Episodes

Wake Up X episodes featuring practitioners describing their own labs. Note which projects come up repeatedly — the overlap with this list is not coincidental.

Resources

- **TryHackMe** and **LetsDefend** — for project 9
- **Wazuh documentation** — genuinely good, and reading real documentation is itself a skill
- **r/homelab** — for ideas, with the caveat that it will make you want to buy things

Checklist

- I know my lane and my starting project
- My project log exists
- I have finished at least one project end to end
- I have not skipped project 4 because it felt boring

Common Mistakes

- Choosing projects by what looks impressive rather than what matches your lane
- Skipping project 4

- Starting six things
- Never producing resume bullets from completed work

Key Takeaways

- There is a sensible order and it starts with identity and virtualization
- Depth on one project frequently beats breadth across six
- Segmentation is the most-skipped, highest-signal project
- Automation on top of anything is the loudest differentiator available

Next Mission

The step that separates a tutorial follower from an engineer.

Breaking Things on Purpose

MISSION

To make deliberate failure a habit, because it is where the learning actually lives.

Objectives

- Break systems safely and systematically
- Build a mental library of failure modes
- Turn each break into an interview story

Why It Matters

Here is the uncomfortable truth about tutorials: they teach you the happy path, and the happy path is maybe fifteen percent of the job.

The actual job is: something that worked yesterday does not work today, nobody knows why, and people are waiting. Nothing in a tutorial prepares you for that. Only having been there prepares you for that — and in a lab, you can choose to be there on a Saturday afternoon instead of during an outage.

The reframe. You are not testing whether it works. You already know it works. You are building a library of failure modes, so that when you see a symptom in a real environment, you recognize it instead of meeting it for the first time.

Professional Explanation

Why Snapshots Change Everything

Take a snapshot before you break anything. This one habit removes essentially all risk from experimentation. It is why project 2 sits so early in the build order.

With snapshots, the cost of breaking something is a two-minute restore. Without them, the cost is rebuilding for three hours. That means you will not do it. That means you will never learn the failure modes.

If you take one operational habit from this book: snapshot before you change anything. This transfers directly to professional work, where the equivalent is a change record and a rollback plan.

What Good Breaking Looks Like

Not random. Systematic. For each project, ask: what does this depend on, and what happens when that dependency fails?

A domain controller depends on DNS, on time synchronization, on network connectivity, on its own services running, and on healthy replication if there is more than one. That is five deliberate experiments, each teaching a distinct failure signature.

The goal is that in a future interview or incident, the symptom is *familiar*. "Authentication failing intermittently across the domain" should trigger a short list in your head, and time skew should be on it, because you did it to yourself once.

Framework: The Break Matrix

For every project you build, work through these five categories.

1 — Kill the dependency. Stop the service it relies on. DNS, DHCP, the database, the storage mount. Observe the symptom carefully — often the symptom appears nowhere near the actual cause. That is the lesson.

2 — Misconfigure it. Wrong subnet mask, wrong gateway, wrong DNS server, a firewall rule denying what should be allowed. Configuration errors are the single largest category of real-world incidents.

3 — Remove the permission. Disable the service account. Revoke the certificate. Change the password without updating where it was stored. This teaches you how identity failures present, which is rarely as an obvious "access denied."

4 — Introduce time. Skew the clock on one machine. Let a certificate expire. Kerberos and TLS both care deeply about time, and time-based failures are famously confusing until you have seen one.

5 — Add load or scale. Run it out of disk. Fill the log volume. Add enough clients that something queues. Systems fail differently under pressure, and disk-full is one of the most common real outages there is.

Step-by-Step Guide: A Break Session

Set aside ninety minutes. Do this after a project works.

1. **Snapshot everything.** Non-negotiable.
2. **Write your hypothesis first.** "If I stop the DNS service, I expect X." Writing it first is what turns this from tinkering into practice, because you find out where your mental model is wrong.

3. **Break one thing.** One. Multiple simultaneous failures teach you nothing because you cannot attribute cause.
4. **Observe before fixing.** What is the actual symptom? What do the logs say? What would a user report? That last question matters — users do not report "DNS resolution failure," they report "the internet is down."
5. **Diagnose it as though you did not cause it.** Practise the investigation, not just the fix.
6. **Fix it, then restore the snapshot anyway** if the fix was messy.
7. **Write it down.** Symptom, cause, diagnosis path, fix, and how long it took you.
8. **Repeat with the next dependency.**

Real Example

A break session on a domain controller, done properly.

Hypothesis: if I point the client at the wrong DNS server, domain login will fail immediately with a clear error.

What actually happened: the client logged in fine, using cached credentials, and the failure only appeared later when accessing a network resource — with an error message that mentioned neither DNS nor authentication.

What that taught: the symptom appeared far from the cause, delayed in time, with a misleading error. That is what most real incidents look like. It is why "when did it last work?" is often the most productive question in an investigation.

Interview value: very high. This story demonstrates understanding of caching, of failure propagation, and of investigation method, in about forty seconds.

Case Study: The Question Behind the Question

When an interviewer asks "what would you check if users could not log in?", they are not testing whether you know that DNS matters. They are testing whether you have a *method*.

A weak answer lists possible causes in no particular order. A strong answer has structure: what changed, when did it last work, is it everyone or some people, is it all resources or one, and then works from the widest signal inward.

That structure comes from having done it. Break sessions are how you get it without waiting for a production incident to hand it to you.

Templates: The Break Log

```
PROJECT:
DATE:
WHAT I BROKE:
HYPOTHESIS:      what I expected to happen
ACTUAL SYMPTOM:  what a user would have reported
LOGS SAID:
DIAGNOSIS PATH:  how I actually found it
TIME TO FIND:
FIX:
LESSON:          the one sentence worth keeping
```

The "time to find" field is worth keeping honestly. Watching it drop across sessions is the most direct evidence of improvement you will get.

Reflection

- When something breaks, is your instinct to fix it or to understand it?
- Have you ever fixed something without knowing why the fix worked?
- What would it feel like to recognize a symptom instead of meeting it new?

Journal

Write about a time something broke and you had no idea where to start. What would you do differently now? If the answer is "nothing," that is the gap this chapter is for.

Exercises

1. Run one break session this week, ninety minutes, with a written hypothesis.
2. Start the break log.
3. Break the same thing two different ways and compare the symptoms.

Project

Take your most complete project and work the entire Break Matrix against it — all five categories. Document each. This single exercise will produce more interview material than anything else in this book.

AI Prompts

- "I have built [project]. List its dependencies and, for each, what the user-visible symptom would be if it failed. Do not tell me the fixes yet."

- "Here is a symptom I created deliberately: [description]. Walk me through how an experienced engineer would diagnose this without knowing the cause, as a sequence of questions."
- "I broke [thing] and expected [X] but got [Y]. Explain why my mental model was wrong."

Books

- *The Phoenix Project* — Gene Kim
- *Site Reliability Engineering* — Google. Free online. Dense, and the sections on incident response and postmortems are directly relevant.

Resources

- **Wazuh** and **Windows Event Viewer** documentation — reading logs well is the underrated half of this
- **John Hammond's** content, for a mindset oriented around how things fail

Checklist

- I snapshot before every change
- I write my hypothesis before I break something
- I diagnose as though I did not cause it
- My break log has at least five entries
- I can describe three failure modes from memory

Common Mistakes

- Breaking several things at once, making attribution impossible
- Fixing immediately without observing the symptom
- Not writing the hypothesis, which turns practice back into tinkering
- Skipping break sessions because the project already works

Key Takeaways

- Snapshots make experimentation free; use them constantly
- Symptoms often appear far from causes, delayed and misleadingly labeled
- Break systematically using the five categories, one at a time
- The break log is your richest source of interview material

Next Mission

Turning all of this into something a stranger can see.

PART

THREE

Making It Count

The Portfolio That Does the Talking

MISSION

To make your work visible when you are not in the room.

Objectives

- Document a project so a stranger understands it in ninety seconds
- Choose the right platform and stop spreading thin
- Make the portfolio findable

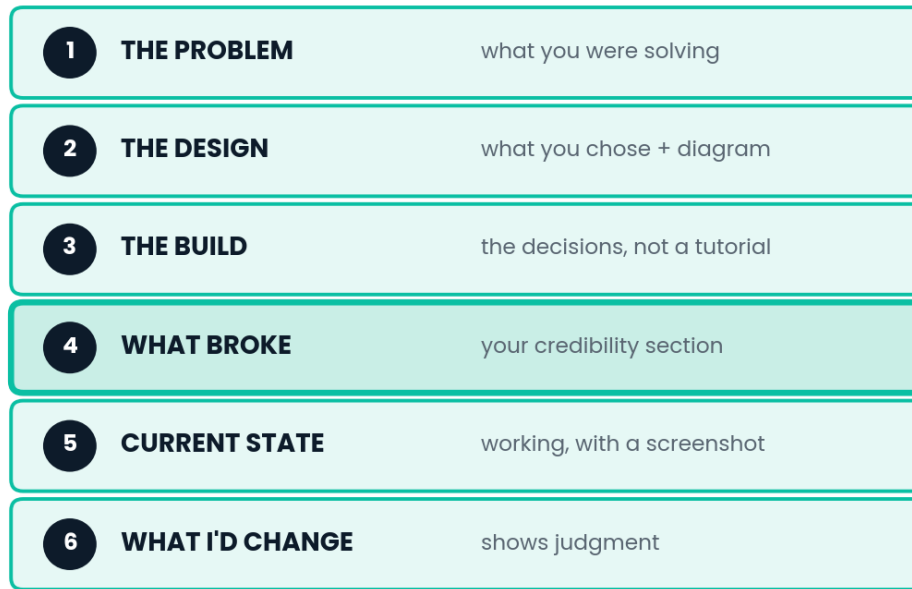
Why It Matters

Your portfolio works while you sleep. A recruiter finds it, a hiring manager checks it before the interview, someone shares it in a channel you will never see.

This is the highest-leverage documentation you will ever write, and most people never write it — which means the bar is low and the return is outsized.

Framework: The Documentation Wheel

THE DOCUMENTATION WHEEL



Anyone can screenshot a working dashboard. Only the builder knows how it failed.

The Documentation Wheel. Section four is the one that proves the work was actually yours.

1 — The problem. What were you trying to solve or learn? One or two sentences. Not "I wanted to learn Docker" — "I wanted to run several services on one machine without them interfering with each other."

2 — The design. What you chose and why. Include a diagram. The diagram matters more than the prose, and a clear hand-drawn one beats no diagram at all.

3 — The build. Key steps, key configuration, and the decisions. Not a tutorial — nobody needs another Pi-hole install guide. What did you decide, and why?

4 — What broke. Be specific and honest. This is the section that impresses experienced people, because it is the one that proves the work was real.

5 — What it does now. The working state, with a screenshot.

6 — What I would do differently. Shows judgment, and shows you are still thinking about it.

Anyone can post a screenshot of a working dashboard. Only someone who actually built it can describe the specific way it failed at eleven at night and what the fix turned out to be. That section is your credibility. It is free.

Step-by-Step Guide: Publishing It

1. **Pick one home.** GitHub for anything with configuration or code. A simple static site for anything visual. LinkedIn articles if you want reach over permanence. Pick one — spreading across three means maintaining none.
2. **Write the README first,** before polishing anything else. If the README is good, the project is presentable even if the rest is rough.
3. **Add one diagram per project.** Draw it badly if you must. Diagramming tools are free and the skill transfers directly to professional work, where the person who can draw the architecture runs the meeting.
4. **Link it from your resume header and your LinkedIn profile.** A portfolio nobody can find is a diary.
5. **Write one short post per finished project.** Not a brag — what you learned and what broke. Those posts perform far better than achievement announcements, and Book 7 covers this properly.
6. **Keep it current.** A portfolio with three finished, documented projects beats one with nine half-described ones.

Templates: The Project README

```
# Project Name
One line: what this is and why it exists.

## The problem
What I was trying to solve or understand.

## The design
What I chose and why. [diagram]

## What I built
The decisions, not the tutorial steps.

## What broke and how I fixed it
Be specific. This is the credibility section.

## Current state
What it does now. [screenshot]

## What I'd do differently
## What this taught me
```

Use the same skeleton every time. Consistency reads as professionalism, and it means you never face a blank page.

Real Example

Two write-ups of the same Pi-hole project.

Version A: a step-by-step install guide, identical to the official documentation, ending with a screenshot of the dashboard.

Version B: two sentences on why — too many devices, wanted network-wide filtering rather than per-device. A diagram showing where it sits in the network. A note on the decision to run it on dedicated hardware rather than a VM, and why. A paragraph on the outage caused when the Pi-hole went down and DNS resolution stopped for the entire household. The fix — a secondary DNS entry. Plus the observation that this is exactly why production DNS is redundant.

Version A shows that the author can follow instructions. Version B shows that the author understands single points of failure, learned it the hard way at home, and can explain it.

Same project. Forty additional minutes.

Case Study: The Portfolio That Got Read

A pattern worth naming, because it explains why the link placement matters so much.

Hiring managers frequently look candidates up before the interview — not deeply, just a quick check. What they find in that thirty seconds shapes the conversation before it starts.

If they find a GitHub with three clean READMEs, the interview begins with the manager already believing you can do the work, and the questions become "tell me about this one" rather than "have you ever actually done this?" That is a lot easier interview.

If they find nothing, the interview starts from zero.

The entire difference is a link in your resume header and forty minutes per project.

Reflection

- If someone looked you up right now, what would they find?
- Which of your projects has the best "what broke" story, and have you written it down?
- Are you spreading across platforms instead of finishing on one?

Journal

Write the "what broke" section for your most recent project, from memory, before you check any notes. What you remember without notes is what actually taught you something.

Exercises

1. Write one README using the skeleton, for a project you have already finished.
2. Hand it to someone non-technical and ask what the project does. Their confusion is your edit list.
3. Add the portfolio link to your resume header and LinkedIn today.

Project

Publish three project write-ups on one platform, each with a diagram and a real "what broke" section. Three is the number where a portfolio starts working on your behalf.

AI Prompts

- "Here are my rough notes from a lab project. Turn them into a README following this structure, keeping my voice and not inventing anything I did not describe."
- "Read this project write-up as a hiring manager with ten seconds. What do you understand, and what is unclear?"
- "My 'what broke' section is thin. Here is what happened: [description]. Ask me five questions that would make it more specific."

Books

- *Docs for Developers* — Bhatti et al. Practical technical-writing guidance that transfers directly.
- *The Pragmatic Programmer* — Hunt and Thomas. On documenting decisions rather than steps.

Resources

- **draw.io** or **Excalidraw** — free diagramming, and the skill matters professionally
- **GitHub** — free hosting and the default expectation for technical portfolios

Checklist

- Each project has a README with a diagram and a "what broke" section
- My portfolio is on one platform, not three

- The link is in my resume header and my LinkedIn profile
- Someone non-technical has read one write-up and understood it
- I have at least three published projects

Common Mistakes

- Writing tutorials instead of decision records
- Hiding the failures, which removes the only proof the work was yours
- Building a portfolio nobody has ever been given a link to
- Publishing nine incomplete write-ups instead of three complete ones

Key Takeaways

- The portfolio is the argument; the projects are the evidence
- "What broke" is the most credible section you can write
- One platform, three finished write-ups, linked everywhere
- Diagrams carry more weight than prose

Next Mission

Converting all of it into the words you will actually say.

From Project to Resume Bullet to Interview Answer

MISSION

To close the loop — turning built work into the specific sentences that get you hired.

Objectives

- Convert any project into two or three evidence bullets
- Build the two-minute spoken version
- Handle the "was this just a lab?" question

Why It Matters

Plenty of people build things and still do not get hired, because the work never becomes language. The project sits on a drive, the README sits in a repository nobody was sent, and in the interview the candidate says "I have a home lab" and stops.

This chapter is the conversion step. It is where Book 2 hands off to Book 1's interview chapter.

Framework: The Three Conversions

Every finished project converts three ways.

1 — Into resume bullets. Using the evidence skeleton from Book 1: action verb, what you did, the scale or tool, what changed. Two or three per project. Tag them honestly as lab rather than production.

2 — Into a two-minute spoken story. Problem, what you built, what broke, what you learned. Practised out loud, not in your head.

3 — Into a written artifact. The README, which is what someone reads before they meet you.

If a project has not produced all three, it is not finished — whether or not the technology is running.

Step-by-Step Guide: The Conversion

1. **Open the project log and the break log.** Everything you need is already written down, which is the entire reason those files exist.

2. **Write the bullets.** Scale matters: "15 lab endpoints" is better than "several machines." Honest numbers only.
3. **Write the two-minute version as bullet points,** not a script. Scripts sound memorized and collapse under a follow-up.
4. **Practise it out loud, twice.** Time it. Most people either run to five minutes or finish in thirty seconds.
5. **Prepare the follow-up answers.** For each project, anticipate the two hardest questions and answer them honestly.
6. **Add the bullets to your bullet bank** in the relevant themed sections.

Templates: Bullets From One Project

From a single domain controller build, honestly written:

- Built a Windows Server 2022 domain controller with Active Directory, DNS, and Group Policy, managing 15 lab endpoints with enforced password and lockout policy
- Diagnosed and resolved a domain-wide authentication failure caused by DNS misconfiguration, tracing a delayed and misleading client-side symptom back to cached credentials
- Documented the build, five deliberate failure scenarios, and their diagnosis paths in a public write-up

Three bullets, one project, all defensible, all producing natural follow-up questions you can answer well.

Handling "Was This Just a Lab?"

You will get asked. The answer is not defensive.

"Yes — it's a lab, not production. What that means is I haven't dealt with change control, or a hundred users, or the politics of a migration. What I do have is the concepts and the failure modes, because I broke it on purpose five different ways. I'd expect the first month here to be about learning how it works at real scale."

That answer is honest, specific, and it shows that you understand the difference between your experience and the job — which is exactly what the question was testing.

Never inflate a lab into production experience. Not on a resume, not in an interview. It is checkable, it is a fireable offense at some organizations, and it converts an honest strength into a permanent liability. The lab is genuinely impressive. Let it be what it is.

Real Example

Two candidates describe the same SIEM project.

Candidate A: "I set up Wazuh in my home lab and it monitors my network."

Candidate B: "I wanted to see what was actually happening on my own network, so I deployed Wazuh across four machines. The first week was almost useless because the false positive rate was so high I stopped reading the alerts — which taught me more than the build did. I spent most of my time tuning rules rather than adding detections, and I cut the noise roughly in half. If I did it again I'd start with far fewer rules enabled."

Same project. Candidate B has showed that they understand alert fatigue, tuning, and their own judgment, in about thirty seconds.

Case Study: What the Follow-Up Is Really For

When an interviewer follows up on a project claim, they are checking one thing: does the detail hold up?

A person who built something has effectively unlimited detail. They remember the annoying part, the thing that took three attempts, the setting whose name they can never recall. A person who watched a video has a summary and nothing underneath it.

You do not need to perform depth. You need to have it, which you will if you followed the break sessions. Then the follow-up question is the easiest part of the interview rather than the moment it falls apart.

Reflection

- Which of your projects could you talk about for five minutes without preparing?
- Which one would collapse at the second follow-up question?
- Have you actually said any of this out loud, to a person?

Journal

Write the two-minute version of your best project as if explaining it to a family member with no technical background. Simplifying for a non-expert is the fastest way to find the parts you do not actually understand.

Exercises

1. Convert your most complete project into three resume bullets today.
2. Record yourself doing the two-minute version. Watch it back.
3. Write down the two hardest follow-up questions for each project, and your honest answers.

Project

Take all your finished projects and complete the full conversion for each — bullets, spoken version, README. Then have someone interview you on them and follow up hard.

AI Prompts

- "Here is my project log and break log for [project]. Write three resume bullets using: action verb, what I did, scale or tool, what changed. Do not invent numbers."
- "Act as a technical interviewer. I will describe a lab project. Follow up hard on anything vague, one question at a time."
- "Here is my two-minute project description. Where would a skeptical interviewer push, and what am I glossing over?"

Books

- *Never Split the Difference* — Chris Voss
- *Made to Stick* — Chip and Dan Heath. On why concrete, specific stories are remembered and abstractions are not.

Podcast Episodes

Wake Up X episodes where a guest explains something technical to a general audience. Notice the ones who do it well — that is the skill this chapter is training.

Checklist

- Every finished project has produced at least two resume bullets
- I have practised the two-minute version out loud

- I have honest answers to the two hardest follow-ups per project
- I have never described lab work as production experience
- My bullet bank contains every project bullet

Common Mistakes

- Building without ever converting into language
- Inflating lab into production
- Scripting the spoken version, which collapses under follow-up
- Saying "I have a home lab" and stopping

Key Takeaways

- A project is not finished until it has produced bullets, a spoken version, and a README
- Honest framing of lab work is a strength, not an apology
- Depth in follow-up is what separates built from watched
- The annoying details are your credibility

Next Mission

Book 3 takes the defensive projects and turns them into the way a security professional actually thinks.

Appendix: READMEs, Logs, and the Purchase Order

REFERENCE

Everything from this book in one place.

The Project README

```
# Project Name
One line: what this is and why it exists.

## The problem
## The design          (+ diagram)
## What I built         (decisions, not steps)
## What broke and how I fixed it
## Current state       (+ screenshot)
## What I'd do differently
## What this taught me
```

The Project Log

```
PROJECT:
STARTED:
STATUS:    planning / building / breaking / documented
GOAL:
HARDWARE:
WHAT WORKED:
WHAT BROKE:
TIME SPENT:
RESUME BULLETS PRODUCED:
```

The Break Log

PROJECT:
DATE:
WHAT I BROKE:
HYPOTHESIS:
ACTUAL SYMPTOM:
LOGS SAID:
DIAGNOSIS PATH:
TIME TO FIND:
FIX:
LESSON:

The Break Matrix

1. Kill the dependency
2. Misconfigure it
3. Remove the permission
4. Introduce time (clock skew, expired certificate)
5. Add load or fill the disk

The Purchase Decision Test

1. What specific project is this unblocking?
2. Have I finished the projects I can do without it?
3. Is there a free or cheaper way to learn the same concept?
4. Will this still be useful in a year?

The Buying Order

STAGE 1	Free	existing machine, VirtualBox, free tiers
STAGE 2	Mini-PC	refurbished, 16GB RAM, SSD
STAGE 3	Networking	managed switch + VLAN-capable router
STAGE 4	Storage	NAS or dedicated backup drive
STAGE 5	More RAM	it is almost always RAM

The Nine Projects

1	Domain Controller	identity and access
2	Virtualization	safe testing, snapshots
3	Pi-hole / DNS	name resolution
4	VLAN Segmentation	blast radius
5	Docker	how software ships
6	NAS and Storage	backup and restore
7	VPN Remote Access	secure extension
8	SIEM (Wazuh)	detection
9	Blue-Team Practice	triage under pressure
+	Automation (n8n)	the loudest differentiator

The Bullet Skeleton

Action verb + what you did + scale or tool + what changed
 Tag each: [production] or [lab]

A Closing Note

The lab is not preparation for real work. It is real work, done on your own initiative, with nobody paying you and nobody checking. That is a harder thing than doing assigned work, and the people evaluating you know it.

I want to leave you with the same point I opened with, because it is the one that matters.

I started at thirteen with a donated computer that had no mouse, in a language I did not yet speak, with nobody to ask. I broke that machine constantly. Nobody gave me permission, the hardware was terrible, and I had no idea what I was doing.

It still worked. Not because I was talented — I was a kid who had been told he could not learn — but because I kept opening it up and trying things.

The hardware you have is better than mine was. The resources available to you are enormously better. The only variable left is whether you start.

Build one thing. Break it. Write it down. Then do it again.

Connect

Building something? Share it and tag me — [linkedin.com/in/efrenlsalazar](https://www.linkedin.com/in/efrenlsalazar). I mean that; I want to see it.

The podcast is at wakeupxpodcast.com.

Questions for the author. Three gaps where your material would replace my framing: (1) a specific lab project of yours that failed badly and what it taught you — Chapter 4 is currently carrying invented-but-plausible examples where a real one would land harder; (2) the actual moment you realized your documented work was generating opportunities; (3) what your own lab looks like today, which readers will ask about and which would ground the buying chapter in something real.

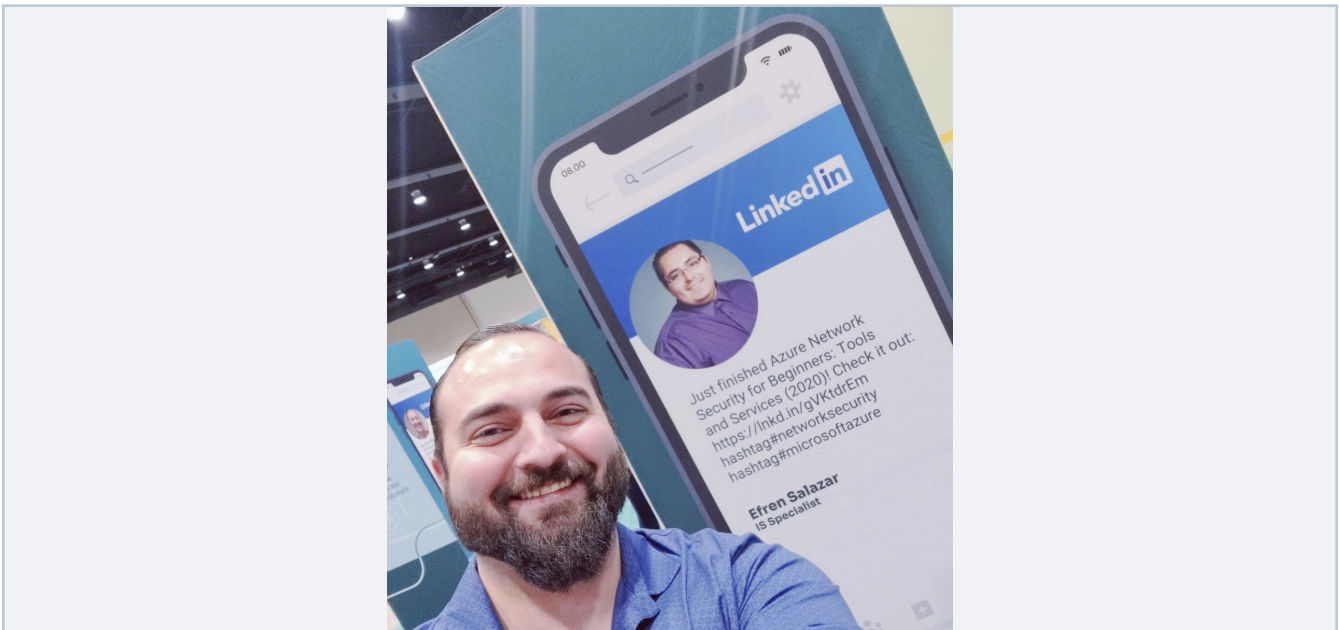
IN THE FIELD

Build It, Document It, Show It

The same habit that built the shop's second location built a public track record online — every project documented and posted.



The Cellphone Fix — second storefront era



Posting a finished cert project for the network

Devin Prasad — Everything Is a System of Components

Devin Prasad is a Network Architect whose path started in a high school electronics academy, not a computer science classroom — four years of soldering circuits before he ever touched a server.

Turned Down Money Twice — On Purpose

His first IT job offer paid \$12.50/hr. He took the \$9.50/hr offer instead, because the higher-paying agency had no path past entry-level help desk and the lower-paying one had a real tier-three ladder. He got his first promotion six months later. His rule: "you always want to work for the position you want to be in, not the one you're in."

Deliberately Went Back to Help Desk

Later in his career, with real networking experience already behind him, he chose a help-desk-first path again at a new, larger organization — on purpose. His reasoning: understanding the business, the customers, and the applications on the front end makes everything on the back end, including security, easier. "Everybody wants to skip the help desk... that's almost impossible to do."

Learn Routing Like Your Career Depends On It

His single strongest technical opinion: "if everyone knew routing in IT, you'd be the best of the best." He explains it as literally visualizing traffic like driving directions — every router is an intersection, and once you can see the whole road, security, cloud, and troubleshooting all get easier at the same time.

Watch the full episode: youtu.be/AP0U1zvp9IU

Devin's Advice, In Practice

Nothing's an all-inclusive system — it's a system of components.

The lesson from his very first electronics class stuck through his entire career: understand the pieces, and the whole becomes learnable.

If you're hungry, people will feed you.

He learned routing, F5 load balancers, and DHCP by shadowing coworkers at night during change windows — unpaid, on his own time, simply by asking.

Networking teaches you cybersecurity fundamentals whether you meant it to or not.

Understanding IPs, MACs, and traffic patterns is, in his words, already learning how attackers think.

Work for the role you want, not the one you have.

He took on senior-level projects before he had the senior title — the promotion followed the work, not the other way around.

Reverse-engineer everything, including yourself.

When his own system for executing started breaking down, he realized he'd stopped doing it for himself and was only doing it for others — that's what led him to build his personal framework from scratch.

REFLECTION

What's the one fundamental (like routing) that, if you actually mastered it, would make everything else in your field click into place?

BE OUR NEXT GUEST

Your Story Belongs on Wake Up X

If Build It gave you something worth using, you already have something worth sharing. The Wake Up X podcast exists to put real, unpolished career and business stories in front of people who need to hear them — and every guest so far started exactly where you are now: reading, building, and figuring it out in public.

You might be a fit if you:

Broke into tech, cybersecurity, or business without the traditional path

Built something — a lab, a business, a portfolio, a community — and can talk about how

Have a failure or turning point you're willing to tell straight, not polished

Are mentoring, leading, or building something worth other people knowing about

HOW TO APPLY

Share your story at wakeupxpodcast.com/guest, or connect directly on LinkedIn at linkedin.com/in/efrenlsalazar and mention this book.

Being a guest is also visibility — the same principle Book 7 makes the case for.

Acknowledgments

THE PEOPLE WHO MADE THIS POSSIBLE

None of Wake Up X happened alone, including Build It. Every book in this library, the podcast, and the free library exist because specific people showed up for a kid nobody expected much from. This is the record of who they are.

THE EARLY YEARS

Miss England, the 6th-grade teacher who stayed after class, unpaid, to teach the fundamentals. Sister Cora and Lewis Bair of Computers for Kids, who handed a teenager a key and a broken machine and let the work speak. Jose Luis Bravo, Angel Martinez, Isaac Murrilo, and Alex De Haro — the friends and cousins who translated the world, gave hope, drove him where he needed to go, and carried him before there was anything to carry him toward.

THE BUSINESS YEARS

Rojelio Salazar, his cousin and first mentor, RNS Computer Repairs — the reason it is called Wake Up X. Kyle Erni, his first business partner. Erick and Pancho Arauza, co-founders of The Cellphone Fix. Danny Maya, of Maya Traditional Mexican Restaurant, Taqueria Maya, and Salt and Water, who pushed him in front of a camera for the first time.

THE CAREER YEARS

Igor, who gave him the vocabulary for work he already knew how to do. Nate, who he mentored into a full-time role alongside his own. Ken Kerrick, opponent turned friend in the ISACA Sacramento election. Alvin Jue and David De los Reyes, who believed in the middle of it, not just at the finish.

THE PODCAST

Every guest who trusted an unproven show with their story, including Gustavo Mastroianni, Cathy Stephenson, Greg McCord, Aaron Portnoy, Sana Talwar, Jorge Avila, Anil Reddy, Devin Prasad, Larry Switzer, Louis Kemp, Mohana Balakrishnan, Eddie Lopez, and Michael Fierros — full episodes at wakeupxpodcast.com.

COMMUNITY PARTNERS

ISACA Sacramento and ISC2 Sacramento, The Cellphone Fix, A.N.U.L Technologies, AISOC, AspireByte, and PodGrabber — the organizations building alongside this movement.

AND

His parents, who arrived with nothing, worked since childhood, and gave away whatever was left over. Nothing about the first version of this story predicted the version you are holding now.

Continue the Journey

EVERY BOOK LEADS SOMEWHERE ELSE

Build It is one part of a 12-book system. If it helped, the next step is one page away — take whichever layer matches where you are standing.

THE LIBRARY

Layer 1 — The Story

I Just Have To Wake Up — why any of this exists.

Layer 2 — The Mindset

Comfort Is Expensive — the real cost of playing it safe.

Layer 3 — The Blueprint

DECODE: The Wake Up X Playbook — zero experience to executive thinking. Book 0.

Layer 4 — The Playbooks

1 Break In · 2 Build It · 3 Think Like a Cybersecurity Professional · 4 Business First · 5 Lead Without a Title · 6 AI for IT & Cybersecurity · 7 Build Your Brand · 8 The Consultant Mindset · 9 Level Up

FIND US EVERYWHERE

Podcast & free library: wakeupxpodcast.com · Spotify: open.spotify.com/show/28GZnRBrIV2efHOHzdiftt · Apple Podcasts: podcasts.apple.com/us/podcast/id1792821643 · YouTube: youtube.com/@Efren.wakeupx · LinkedIn: linkedin.com/in/efrenIsalazar · Substack: substack.com/@wakeupxefren

IN GOOD COMPANY

ISACA Sacramento, ISC2 Sacramento, The Cellphone Fix, A.N.U.L Technologies, AISOC, AspireByte, and PodGrabber — the partners and collaborators building alongside the movement. See the full list at wakeupxpodcast.com/community.html.

Get DECODE free, apply to be a guest, or read the story behind the movement — all of it starts at wakeupxpodcast.com.

WAKE UP X

SOLUTIONS



Efren L. Salazar

FOUNDER · HOST · CYBERSECURITY PROFESSIONAL · EDUCATOR · COMMUNITY BUILDER

STAY CONNECTED

Podcast & Blueprint Library:	wakeupxpodcast.com
LinkedIn:	linkedin.com/in/efrenlsalazar
YouTube:	youtube.com/@Efren.wakeupx
Be a Guest:	wakeupxpodcast.com/guest

THE WAKE UP X PUBLISHING SYSTEM

BOOK 3

Think Like a Cybersecurity Professional

Judgment, triage, and incident response

Security is a judgment problem, not a tool problem

THE
WAKEUPX
SOLUTIONS

EFREN L. SALAZAR

Founder & Host, Wake Up X | IT Operations Manager & Solution Engineer
Government & Regulatory Partner, ISACA Sacramento | CompTIA Security+

wakeupxpodcast.com

Educational content — not professional, legal, financial, or career-placement advice.
Full disclaimer in DECODE, the foundation of this library.

Before You Begin

HOW TO USE THIS BOOK

This book is about how defenders think. Not which tools exist — tools change every few years and the vendor will train you on theirs. Judgment transfers for an entire career. It is the thing nobody teaches directly.

Where this book sits

Two books come first — the memoir *I Just Have To Wake Up* and the mindset book *Comfort Is Expensive* — before this ten-book library. Each one is short on purpose. Pick up the one that matches the problem in front of you right now.

- n **Book 0 — DECODE.** The foundation. The map of the whole path.
- n **Book 1 — Break In.** Landing your first role in IT or cybersecurity.
- n **Book 2 — Build It.** Home labs and a portfolio that prove you can do the work.
- n **Book 3 — Think Like a Cybersecurity Professional.** Detection, triage, and incident response thinking. **you are here**
- n **Book 4 — Business First.** Finance, sales, GRC, and why the business hired you.
- n **Book 5 — Lead Without a Title.** Influence, EQ, and mentoring before anyone promotes you.
- n **Book 6 — AI for IT & Cybersecurity.** Using AI as an actual working tool, not a novelty.
- n **Book 7 — Build Your Brand.** LinkedIn, content, and being findable.
- n **Book 8 — The Consultant Mindset.** Discovery, solution selling, and stakeholder management.
- n **Book 9 — Level Up.** Long-term growth, burnout, legacy, and mentoring others.

Book 0 is the foundation. *DECODE: The Wake Up X Playbook* holds the six-level path, the Curiosity Compass, and the core frameworks. Nothing in any other book contradicts it. If a term here is unfamiliar, DECODE defines it.

What is in this book

Part One — The Mindset

1. Security Is a Judgment Problem, Not a Tool Problem
2. Knowing Normal Before You Can Spot Abnormal

Part Two — The Daily Work

3. Triage: The Loop You Will Actually Run
4. Tuning: How Analysts Become Engineers

Part Three — When It's Real

5. Response, and the Order That Matters
6. The Human Part: Communicating Under Pressure
7. Appendix: Runbooks, Escalations, and the Review

How to actually use it

This book pairs with Book 2's projects 8 and 9 — the SIEM build and blue-team practice. If you have not built those, the chapters here will read as abstract. If you have, you will recognize almost everything.

It also does not teach you a specific product. Deliberately. The organization that hires you will train you on their stack in a week; what they cannot train in a week is whether you think clearly under pressure.

A note on scope. This is defensive thinking — detection, triage, response. It does not cover offensive technique, and it does not teach you to attack anything. If your lane is Break rather than Defend, most of this still applies, because understanding how defenders think is exactly what makes offensive work valuable.

I am writing this the way I would say it to you sitting across a table — not lecturing a room. Take what works, remix the rest into your own version, and leave what does not fit.

PART

ONE

The Mindset

Security Is a Judgment Problem, Not a Tool Problem

MISSION

To shift you from memorizing security tools to thinking the way security professionals actually think.

Objectives

- Understand risk as the organizing idea behind all security work
- Stop treating alerts as facts
- See why "block everything" is the wrong instinct
- Write a risk sentence that a decision-maker can act on

Story

Early on, I believed the way in was to learn the tools. Which SIEM, which firewall, which framework, which acronym meant what. I studied the vocabulary hard.

Then I started doing the work, and discovered the job was almost entirely judgment. This alert or that one. This risk accepted or mitigated.

This control worth the friction it creates for two hundred people, or not. Nobody had prepared me for the fact that most security decisions have no technically correct answer — only trade-offs someone has to own.

The thing that actually changed my effectiveness was not a tool. It was learning to run a business, badly, and then better.

When you have made payroll, chosen between two things you could not afford, and watched a decision cost you real money, "we should implement this control" stops being an obvious statement and starts being a proposal with a price attached.

That is the shift this book is about.

Why It Matters

Most people entering security study tools. They learn what a SIEM is, what a firewall does, what the phases of an attack are called. Then they get into a role and discover none of that is the hard part.

The hard part is: an alert fired, you have twelve minutes of attention to give it, three other things are queued, and you have to decide whether this matters. Nothing about knowing what a SIEM is helps with that.

Tools change every few years. The judgment transfers for a career.

Professional Explanation

Business Perspective

Security exists to let the business take risks safely — not to prevent risk. That distinction is everything, and getting it wrong is the most common reason security teams become irrelevant inside their own organizations.

A security team that says no to everything gets routed around. And shadow IT — the unsanctioned tool someone adopted because you blocked the sanctioned one — is far more dangerous than whatever you blocked, because now it is invisible to you.

Every control has a cost: money, friction, someone's time, and a quantity of goodwill you are spending. Your job is not maximum security. It is *appropriate* security for what the organization is actually protecting and what it can actually afford, which needs knowing both of those things.

Technical Perspective

An alert is a hypothesis, not a fact. It says a pattern matched. It does not say something bad happened.

The core skill of a defender is turning that hypothesis into a conclusion using evidence: logs, timelines, context, and above all what normal looks like on this network. That last one is what junior analysts lack and cannot fake. It is why Chapter 2 exists.

There is a second technical reality worth internalizing early: you will be wrong regularly. Detection is a game of odds. The goal is not to never be wrong; it is to be wrong in the cheap direction. Investigating something benign costs an hour, missing something real costs much more.

Leadership Perspective

Security is a communication job wearing technical clothes.

You will spend more time convincing people to do things than doing things. The engineer who can explain risk to a finance director in language that finance director cares about is worth several who cannot. This gap widens as you become more senior.

It also determines whether you are in the room. Security decisions get made with or without you. Being technically right and absent is losing.

The reframe. You are not the department of no. You are the department of "here is what that will cost, here is how we could do it more safely, and here is what I recommend." Then the business decides — because it is their risk to accept, and Book 4's chapter on risk acceptance explains why getting that in writing protects you as much as it protects them.

Framework: The Risk Sentence

Every security concern can be written in one sentence. If it cannot, you do not understand it yet.

THE RISK SENTENCE



If you cannot write this sentence, you do not yet understand the issue.

The Risk Sentence. Four components, no tool names.

"If [threat] exploits [weakness], then [impact], and the likelihood is [assessment]."

Example: If an attacker phishes a finance user who has no MFA, then they can initiate wire transfers, and the likelihood is high because we have seen three attempts this quarter.

That sentence is what turns a technical observation into something a decision-maker can act on. Notice it contains no product names, no acronyms, and no jargon. Notice also that the last clause — likelihood — is the hardest, because it needs evidence rather than intuition. That difficulty is the job.

Practise writing these. They are the deliverable of most security work, whether or not anyone calls them that.

Real Example

A team discovers that an internal application transmits credentials without encryption on the internal network.

The weak version: "The app doesn't use TLS. We need to fix it."

The risk sentence version: "If someone with network access captures traffic on that segment, they can get application credentials for any user who logs in, and the likelihood is moderate. It needs internal access, but the segment is shared with the guest wireless, so the population of people who could do it is larger than it should be."

The second version does three things the first does not: it identifies who could exploit it, it explains why the likelihood is what it is, and it surfaces a second finding — the shared segment — that is arguably more important than the original one.

Case Study: The Control That Made Things Worse

A pattern worth recognizing, because it is common and it is embarrassing.

An organization, responding to a phishing incident, implements an aggressive email filter. Legitimate mail begins getting quarantined — client communications, invoices, calendar invitations from partners.

Within weeks, staff have developed workarounds. Some forward business email to personal accounts. Some ask clients to send documents through consumer file-sharing links. Sales starts using a personal messaging app for client communication because email "does not work."

The organization is now less secure than before the control was implemented, and the security team has no visibility into any of it.

The lesson is not that the filter was wrong. It is that the friction was not accounted for, and friction that is not accounted for gets routed around. Any control you propose should include an honest estimate of what people will do when it inconveniences them — because they will do something.

Reflection

- When you propose a control, do you estimate its friction, or only its benefit?
- Have you ever treated an alert as a fact?
- Whose risk is it, actually — and who has the authority to accept it?

Journal

Write about a security decision you disagree with, at work or somewhere you have observed. Then argue the other side as generously as you can. Most security disagreements are trade-off disagreements where each side is optimizing for something real.

Exercises

1. Write the risk sentence for three things you know are weak in a system you use — your home network, your workplace, anywhere. Notice how much harder the likelihood clause is.
2. Take a control you think should exist and write down what people will do to route around it.

Common Mistakes

- **Confusing an alert with an incident.** Most alerts are noise. Investigating is the job.
- **Recommending controls without cost.** "We should implement X" without "which costs Y and creates friction Z" gets ignored, correctly.
- **Chasing the exotic threat.** The thing that will get your organization is a phished credential or an unpatched internet-facing server. Not a nation-state zero day.
- **Security theater.** Controls that look rigorous and change nothing are worse than no control, because they buy false confidence.
- **Treating risk acceptance as failure.** It is a legitimate, and often correct, business decision.

AI Prompts

- "Here is a technical finding: [description]. Rewrite it as a risk sentence: if threat exploits weakness, then impact, likelihood assessment. Tell me what information I am missing to assess likelihood honestly."
- "I want to propose [control]. What friction will this create for ordinary users, and what workarounds should I expect?"
- "Explain [security concept] in terms of what it costs the business if it is absent, without using product names."

Books

- *Sandworm* — Andy Greenberg. For understanding what real consequence looks like.
- *The Cuckoo's Egg* — Cliff Stoll. The original investigation narrative, and still the best illustration of patient method.
- *Extreme Ownership* — Willink and Babin. Directly relevant to Chapter 6.

Podcast Episodes

Wake Up X episodes with security practitioners. Listen specifically for how they describe decisions rather than tools — the senior ones talk almost entirely in trade-offs.

Resources

- **Darknet Diaries** — narrative, accurate, and excellent for building intuition about how incidents actually unfold
- **CIS Controls** — the most practical prioritized starting point for "what should we actually do"
- **MITRE ATT&CK** — a shared vocabulary for describing adversary behavior

Checklist

- I can write a risk sentence without using product names
- I treat alerts as hypotheses
- I estimate friction when I propose controls
- I understand that risk acceptance is a business decision, not a security failure

Key Takeaways

- Security enables risk-taking; it does not prevent risk
- Alerts are hypotheses requiring evidence
- If you cannot write the risk sentence, you do not yet understand the issue
- Friction that is not accounted for gets routed around

Next Mission

The thing that has to come before any of this works: knowing what normal looks like.

Questions for you. What is the most likely way your current organization gets breached? And is that where its security spending goes?

Knowing Normal Before You Can Spot Abnormal

MISSION

To build the baseline that makes every later judgment possible.

Objectives

- Understand why baselining is the prerequisite skill
- Build a picture of normal in a real environment
- Recognize the specific mistakes that come from missing context

Why It Matters

Everything in the next four chapters depends on this one. It is the chapter most likely to be skipped because it does not feel like security work.

Here is the problem it solves. A new analyst sees a service account authenticating at 3am from an unusual host and escalates. It is a scheduled backup job that has run at 3am for four years. Nobody told them, it is not documented, and the alert did not know either.

That is not a failure of skill. It is a failure of context, and context is the thing you cannot get from a course, a certification, or a book — including this one. You can only get it from the environment.

You cannot spot abnormal until you know normal. This is the single most important sentence in this book. It is the reason your first months in a security role should be spent asking questions rather than proving yourself.

Professional Explanation

What a Baseline Actually Is

Not a document. A working mental model of:

- **Who** normally authenticates, from where, and at what hours
- **What** normally talks to what, on which ports, in which direction

- **When** the recurring things happen — backups, syncs, patch windows, batch jobs, month-end processing
- **How much** is normal — typical volumes, so that unusual volume registers
- **What changed recently** — because "it started three days ago" is frequently the whole answer

Most of this lives in people's heads rather than in documentation. Which means the way you build a baseline is largely by asking, and by writing down what you are told.

The Three-Question Habit

For anything unusual, before escalating:

1. **Has this happened before?** Search historically. If the same pattern appears every Tuesday for a year, it is a schedule, not an intrusion.
2. **Is it happening elsewhere?** One host is a problem. Every host is a change, a deployment, or a policy.
3. **Who would know?** Someone in the organization knows what that server does. Finding them takes ten minutes and saves two hours.

That third question is the one new analysts resist, because asking feels like admitting ignorance. It is the opposite — an analyst who asks the right person quickly is more valuable than one who investigates alone for an afternoon and reaches the same conclusion.

Benign True Positives

The category beginners consistently miss.

- **True positive:** it really happened and it is bad.
- **False positive:** the detection was wrong, it did not happen.
- **Benign true positive:** it really happened, the detection was correct. It is completely fine.

That third category is the largest by a wide margin. The admin really did run that command. The scanner really did generate that traffic. The user really did log in from another country, because they are on holiday.

Misclassifying benign true positives as false positives is how detections get tuned away incorrectly, which is how real events get missed later. The distinction matters operationally, not just semantically.

Framework: Building Your Baseline in Thirty Days

A structure for your first month in any new environment.

Week 1 — The map. What are the crown jewels? What does the business actually do, and which systems are load-bearing for that? Who owns what? Get the network diagram, and then find out how wrong it is.

Week 2 — The rhythms. What runs on a schedule? When are the patch windows, the backups, the batch jobs, the reporting runs? What does Monday morning look like versus Saturday night?

Week 3 — The people. Who works odd hours legitimately? Which teams travel? Which contractors have access and to what? Who has admin, and why?

Week 4 — The noise. Which alerts fire constantly? Which ones does everyone ignore, and why? What has been tuned already, and what was the reasoning? This week tells you more about the environment's real security posture than any document will.

Write all of it down. Not for anyone else — for you, in three months, when you have forgotten.

Step-by-Step Guide: Baseline Your Own Lab

You can practise this before you have a job, using the Book 2 SIEM build.

1. **Run it for two weeks without tuning anything.** Resist the urge to fix the noise immediately.
2. **Every day, note what fired.** Which alerts, how many, from where.
3. **At the end of two weeks, classify everything** into the three categories above. Be strict about the benign-true-positive distinction.
4. **Now write down what normal looks like** on your own network. You will find you cannot, entirely — and the gaps are exactly the gaps you would have in a real environment.
5. **Introduce something genuinely unusual** — a new device, a scan, traffic at an odd hour — and see whether you notice it against the baseline you built.

That final step is the exercise. Most people discover their baseline was thinner than they believed.

Real Example

An analyst sees outbound traffic from a workstation to an unfamiliar external address, at a consistent interval, small payload each time. It looks textbook — beaconing.

Before escalating, they run the three questions.

Has this happened before? Yes, for eight months.

Is it happening elsewhere? Yes, on about forty machines.

Who would know? The endpoint team, who confirm it is the telemetry function of a monitoring agent the organization pays for.

Ten minutes of checking. The alternative version escalates, wakes someone up, and burns credibility that takes months to rebuild.

Note the important part: the analyst's instinct was *correct*. The pattern genuinely does look like beaoning, and in a different environment it would be. Context is what turned a correct instinct into a correct conclusion.

Case Study: The Alert Everyone Ignored

The inverse pattern, and the more dangerous one.

An organization has an alert that fires several times daily and has been benign every time for two years. Everyone closes it without reading. It has become furniture.

Then it fires for a real reason. It is closed without reading, because that is what everyone does with that alert.

This is not carelessness. It is the predictable result of an alert with a poor signal-to-noise ratio being left in place for two years. The failure happened when someone decided to tolerate the noise instead of fixing it, which is exactly what Chapter 4 is about.

If you inherit an alert everyone ignores, you have inherited a problem, not a nuisance.

Reflection

- In your current environment — work or lab — could you describe normal?
- What is the alert you have stopped reading?
- When you do not know something, is your instinct to ask or to investigate alone?

Journal

Write down everything you actually know about the network you use most. Then mark which items you know versus assume. The assumptions are your baseline gaps.

Exercises

1. Run the two-week no-tuning observation on your lab SIEM.
2. For one environment you have access to, write down the five recurring scheduled things. If you cannot name five, you do not have a baseline.
3. Practise the three-question habit on the next unusual thing you see, before reacting.

Project

Build a written baseline document for your home network: what devices exist, what talks to what, what runs on a schedule, what normal volume looks like. Then use it. This is also an excellent portfolio piece, because almost nobody has one.

Templates: The Baseline Notes

```
ENVIRONMENT:
CROWN JEWELS:      what actually matters here
KEY SYSTEMS:       and who owns each
SCHEDULED THINGS:  what runs when
ODD-HOUR NORMAL:   who legitimately works when
KNOWN NOISY ALERTS: and why they're noisy
RECENT CHANGES:   what's different in the last 30 days
WHO TO ASK ABOUT:  [system] -> [person]
```

That last section is the most valuable and the one nobody writes.

AI Prompts

- "I am new to a security role. Give me twenty questions to ask in my first month to build an environmental baseline, organized by who to ask."
- "Here is an alert pattern: [description]. List five benign explanations I should rule out before treating this as malicious."
- "Explain the difference between a false positive and a benign true positive, with three examples of each from an enterprise environment."

Books

- *The Cuckoo's Egg* — Cliff Stoll. An entire book about building a baseline and noticing one thing that did not fit.
- *Thinking, Fast and Slow* — Daniel Kahneman. On the specific mental traps that make analysts anchor on the first explanation.

Resources

- Your own environment's documentation, however out of date — the gaps between documentation and reality are instructive
- **MITRE ATT&CK**, for a vocabulary to describe what you observe

Checklist

- I run the three questions before escalating
- I understand benign true positives as distinct from false positives
- I have written baseline notes for at least one environment
- I ask people rather than investigating alone for hours
- I know which alerts are being ignored where I work

Common Mistakes

- Escalating without checking history
- Treating "I have not seen this before" as "this is abnormal"
- Classifying benign true positives as false positives, then tuning away real detections
- Not asking, because asking feels like weakness

Key Takeaways

- You cannot spot abnormal until you know normal
- Three questions before escalating: happened before, happening elsewhere, who would know
- Benign true positives are the largest category and the most misclassified
- An ignored alert is a problem, not a nuisance

Next Mission

The loop you will run hundreds of times a week.

PART

TWO

The Daily Work

Triage: The Loop You Will Actually Run

MISSION

To give you the working method a SOC analyst uses every day.

Objectives

- Run a repeatable six-question sequence on any alert
- Know when to escalate and when to close
- Build a timeline that lets someone else pick up your work

Why It Matters

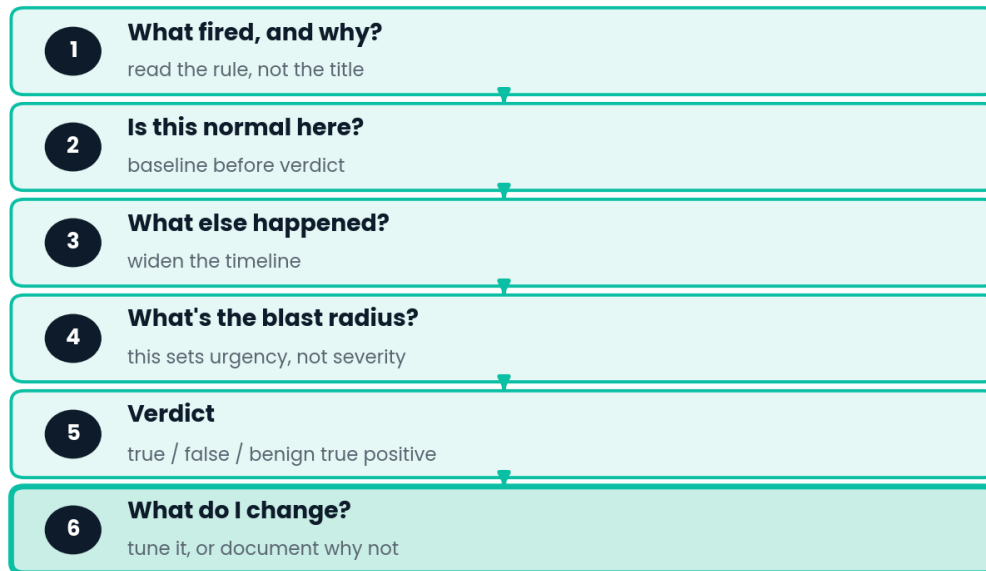
The romantic version of security is hunting an adversary. The real version is a queue.

Your value is determined almost entirely by how well you work that queue: fast on the noise, careful on the real thing, and accurate about which is which. Analysts who are slow on everything drown.

Analysts who are fast on everything miss things. The skill is triage in the medical sense — sorting by urgency, quickly, without being sloppy about the ones that matter.

Framework: The Triage Loop

THE TRIAGE LOOP



Step 6 is where analysts become detection engineers.

The Triage Loop. Same six questions, every alert, in order.

1 — What fired, and why? Read the actual detection logic. Not the alert title — the rule. Roughly half of triage ends here, because the rule was matching something benign and the title oversold it.

2 — Is this normal here? Context, from Chapter 2. Baseline before verdict.

3 — What else happened around it? Widen the timeline. Fifteen minutes before, an hour after. Single events lie; sequences tell the truth. An authentication failure is nothing. An authentication failure followed by a success followed by a privilege change is a story.

4 — What is the blast radius if this is real? What does this account or host have access to? This determines urgency — not the alert's own severity label, which was set by whoever wrote the rule and knows nothing about your environment.

5 — Verdict. True positive, false positive, or benign true positive.

6 — What do I change? Every false positive should either tune the rule or document why it cannot be tuned. Otherwise you will investigate the same nothing next week, and the week after, forever.

Step six is where careers are made. Analysts who only close tickets stay analysts. Analysts who reduce the queue's noise become detection engineers. That is the most common route out of a level-one seat.

Step-by-Step Guide: Your First Real Investigation

1. **Take the alert.** Note the exact timestamp and every entity involved — user, host, IP, process, parent process.
2. **Pull the raw log the alert came from.** Read it yourself. Do not trust the summary; summaries lose the field that mattered.
3. **Build a timeline in a document.** Literally write times down the page. This sounds primitive and it is the single most effective investigation habit there is.
4. **Answer: what did this entity do before, and after?**
5. **Check against the baseline.** Is this host supposed to talk to that destination? Is this account supposed to be active at this hour?
6. **Ask the three questions from Chapter 2** if anything is unclear.
7. **Write your conclusion as a risk sentence.**
8. **Escalate with the timeline attached, or close with the reason written down.**

Step 3 is the one people skip and the one that most improves outcomes. A timeline written down is thinking made visible; a timeline in your head is a guess you cannot check.

On Speed

You will feel pressure to be fast. Some of that pressure is real — a queue that grows faster than it drains is a genuine problem.

The resolution is not to rush every alert. It is to spend thirty seconds on step 1 and, for most alerts, correctly stop there. Speed comes from accurate early classification, not from doing every step badly.

Lab: Practise This For Real

Using the Wazuh SIEM from Book 2, project 8:

1. Generate a genuine failed-login burst against a lab machine and let it alert.
2. Run the full six-question loop against your own alert, including the written timeline.
3. Now do it again with something you did *not* create — traffic from a device you forgot about, or an alert you did not anticipate.

4. Deliberately tune one rule so a legitimate pattern stops firing, and document why.
5. Break something from the Book 2 Break Matrix and triage the resulting alerts without knowing in advance which alert corresponds to what you did.

That last exercise is the closest thing to the real job you can build at home. Documented, it is worth more in an interview than any certification you could name.

Real Example

An alert fires: "Suspicious PowerShell execution."

Step 1: the rule matches encoded PowerShell commands. Reading it, the rule matches *any* encoded command, which many legitimate tools use.

Step 2: the host is a developer workstation. Encoded commands are not unusual there.

Step 3: widening the timeline shows the command ran immediately after a software deployment tool started, and the parent process is that tool.

Step 4: the account is a standard user with no elevated access. Blast radius is small regardless.

Step 5: benign true positive. It really executed, the rule was correct. It is fine.

Step 6: the rule needs tuning — encoded PowerShell with that specific parent process on developer workstations should not alert. Document the exception and the reasoning, so that in six months nobody wonders why it was excluded.

Total time, with practice: five minutes. Without the loop: either an escalation that wastes someone's afternoon, or a close-without-reading that teaches you nothing and leaves the noise in place.

Case Study: The Escalation That Made Work

An analyst escalates an alert with a one-line summary: "Possible compromise on HOST-4472, please advise."

The senior engineer who picks it up now has to do the entire investigation from scratch, because nothing has been established — no timeline, no entities, no ruling-out. The escalation transferred the work without transferring any of the thinking.

Compare an escalation that arrives with a timeline, the entities involved, what has been ruled out and why, the blast radius, and a specific question. The senior engineer can act in five minutes.

The difference is not intelligence. It is whether the analyst treated escalation as handing off a problem or as handing off *progress*. This single habit determines how quickly people trust you.

Templates: The Escalation Note

```
WHAT:          one line, what appears to have happened
WHEN:          timestamp range
WHO/WHAT:      user, host, IP, process, parent process
TIMELINE:      the sequence, in order, with times
RULED OUT:     what I checked that came back benign
WHY I THINK IT'S REAL: the evidence, specifically
BLAST RADIUS:  what this entity can reach
WHAT I'VE DONE: containment actions already taken
WHAT I NEED:   the specific decision or action requested
```

The "ruled out" line is the one that marks an experienced analyst, and almost nobody includes it.

Reflection

- Do you read detection logic, or alert titles?
- When you close something as a false positive, do you change anything?
- Could someone else pick up your investigation from your notes alone?

Journal

Write about the last time you were wrong about a technical judgment. What did you anchor on, and what would have unstuck you earlier?

Exercises

1. Run the full six-question loop on ten alerts in your lab, with written timelines.
2. Write one escalation note using the template, even if there is nobody to send it to.
3. Time yourself. Track how long step 1 takes as you improve.

AI Prompts

- "Here is a detection rule: [rule]. Explain in plain language exactly what it matches and what legitimate activity would also match it."
- "Here is a sequence of log events. Build a timeline and tell me specifically what is unexplained."

- "What are three benign explanations for this pattern that I should rule out before escalating?"
- "Review my escalation note. What would a senior analyst still have to figure out themselves?"

Books

- *Thinking, Fast and Slow* — Daniel Kahneman
- *The Cuckoo's Egg* — Cliff Stoll. Essentially a book-length timeline investigation.

Resources

- **LetsDefend** and **TryHackMe** blue-team paths — structured triage practice with feedback
- **MITRE ATT&CK**, for naming what you find

Checklist

- I read detection logic, not just alert titles
- I build a written timeline every time
- Every false positive results in a tuning action or a documented reason
- My escalations include what I ruled out
- Someone else could continue my investigation from my notes

Common Mistakes

- Reading alert titles instead of rules
- Closing as false positive without tuning, guaranteeing a repeat
- Escalating without a timeline, which makes the senior person redo your work
- Treating alert severity as urgency — severity is set by the rule author, blast radius is set by reality
- Anchoring on the first bad explanation and gathering only confirming evidence

Key Takeaways

- Triage is a repeatable loop, not intuition
- Sequences tell the truth; single events lie
- Blast radius sets urgency, not the severity label
- Escalation should transfer progress, not just the problem

Next Mission

The step that turns an analyst into an engineer.

Tuning: How Analysts Become Engineers

MISSION

To make you the person who reduces the queue rather than the person who works it.

Objectives

- Tune detections without creating blind spots
- Document exceptions so they survive you
- Understand the trade-off you are actually making

Why It Matters

Two analysts join the same team. Both work the queue competently. Two years later one is a senior detection engineer and one is still working the queue.

The difference is almost always step 6. One of them changed the system; the other served it.

This chapter is short because the concept is simple. It is placed prominently because the concept is neglected.

Professional Explanation

What Tuning Actually Is

Every detection sits on a trade-off between missing real events and generating noise. Tuning is deliberately choosing a point on that trade-off, with your eyes open, and recording why.

The danger is obvious: tune too aggressively and you create a blind spot. The danger of *not* tuning is less obvious and more common. An alert nobody reads is a blind spot too, and a worse one, because everybody believes it is providing coverage.

An alert that is always ignored provides negative value. It consumes attention, it creates false confidence, and it trains the team to close without reading — a habit that will eventually be applied to something real.

How to Tune Without Creating Blind Spots

- **Narrow, do not disable.** Exclude the specific benign case, not the whole rule. "Exclude this parent process on this host group" rather than "turn off encoded PowerShell detection."

- **Exclude by cause, not by symptom.** If a backup service generates the noise, exclude the service account, not the whole time window — because the time window would also hide anything else that happens at 3am.
- **Document the reasoning, not just the exception.** In six months, someone — possibly you — will find the exception and need to know whether it is still valid.
- **Set a review date.** Exceptions accumulate. An exception created for a system that was decommissioned a year ago is pure blind spot.
- **Measure before and after.** How many alerts did this rule generate last week? How many now? If you cannot answer, you did not tune, you guessed.

The Conversation to Have First

Before tuning anything you did not build, find out why it exists. Detections are sometimes in place because of a compliance requirement, a past incident, or a specific concern of someone senior. Tuning it away without knowing that is a fast way to have an uncomfortable conversation.

Asking takes five minutes: "This rule generates about forty alerts a week and they have all been benign. Do you know why it was written the way it was before I narrow it?"

Framework: The Tuning Record

Every change gets an entry.

```
RULE:           name and ID
DATE:
NOISE BEFORE:   alerts per week
WHY IT'S NOISY: the actual cause
CHANGE MADE:    specific and narrow
WHAT THIS NOW MISSES: be honest about the blind spot
WHY THAT'S ACCEPTABLE:
NOISE AFTER:
REVIEW DATE:
CHANGED BY:
```

The "what this now misses" line is the one that separates engineering from convenience. Every tuning decision creates a gap. Naming it is what makes the decision defensible.

Real Example

A rule alerts on any authentication from outside the country. It generates roughly thirty alerts a week, all from employees traveling or working remotely.

The lazy fix: disable the rule. Now foreign authentication is invisible entirely.

The tuned fix: exclude authentications that come from accounts with an approved travel notice, and from the two countries where the organization has offices. Keep alerting on everything else, and add a higher-severity variant for authentication from a country where the same account also authenticated domestically within the past hour — which is physically impossible and therefore genuinely interesting.

Result: noise drops a lot, coverage improves, and a better detection now exists than before. That last part is what tuning done well looks like — it frequently produces a *better* detection, not just a quieter one.

Case Study: The Exception Nobody Reviewed

An exception is created for a legacy application that generates noisy authentication events. Sensible at the time, properly narrow, correctly documented.

Three years later the application is decommissioned. The exception remains, because nobody reviews exceptions. The service account it excluded is never disabled, because nobody reviews those either.

That account is now an excluded, unmonitored, still-valid credential. It is exactly the sort of thing that ends up in an incident report.

Exceptions need review dates for the same reason certificates have expiry dates: the world changes and the artifact does not know.

Reflection

- What is the noisiest alert you deal with, and why has nobody fixed it?
- When you tune something, do you record what it now misses?
- Are you working the system or improving it?

Journal

Write down the three things about your current queue or lab that annoy you most. Those are your engineering backlog — irritation is a reliable signal of a fixable inefficiency.

Exercises

1. In your lab, tune one rule properly using the tuning record. Measure before and after.
2. Write the "what this now misses" line honestly for that change.

3. Find one exception somewhere you have access to and check whether it is still valid.

Project

Take your lab SIEM after two weeks of running unmodified, and do a full tuning pass. Every noisy rule narrowed, every change documented with its blind spot named. Then write it up. This is one of the strongest portfolio pieces available for a blue-team role, because it demonstrates judgment rather than installation.

AI Prompts

- "Here is a noisy detection rule and examples of the benign activity triggering it. Suggest narrow exclusions that address the cause rather than the symptom, and tell me what each exclusion would now miss."
- "I want to tune this rule by [approach]. What blind spot does that create, and is there a narrower option?"
- "Help me write the reasoning for this exception so someone reading it in a year understands whether it is still valid."

Books

- *Site Reliability Engineering* — Google. Free online; the chapters on alerting philosophy apply almost directly to detection.

Resources

- **Wazuh** and **Sigma** rule documentation — reading real detection logic is the skill
- **Detection Engineering** communities and public rule repositories, for seeing how others reason

Checklist

- I narrow rather than disable
- I exclude by cause, not by symptom
- Every exception has documented reasoning and a review date
- I record what each change now misses
- I measure noise before and after

Common Mistakes

- Disabling instead of narrowing
- Excluding a time window instead of a cause
- Undocumented exceptions
- No review dates, so exceptions become permanent blind spots
- Tuning a rule without finding out why it exists

Key Takeaways

- An always-ignored alert has negative value
- Good tuning frequently produces a better detection, not just a quieter one
- Name the blind spot every change creates
- Step six is the route out of the queue

Next Mission

What happens when the answer is yes, this is real.

PART

THREE

When It's Real

Response, and the Order That Matters

MISSION

To prepare you for the moment the alert is genuine and the sequence of your actions determines the outcome.

Objectives

- Know the six phases and your role in each
- Understand why containment is not eradication
- Preserve evidence before remediating

Why It Matters

Incidents are not won by heroics. They are won by preparation and by doing things in the right order — and the wrong order is intuitive. That is why people get it wrong under pressure.

Framework: The Six Phases



Containment is not eradication. Confusing them is how incidents become disasters.

The six phases. The first and last are the ones that determine everything.

Preparation. Everything you do before. Runbooks, contact lists, backups you have actually tested, access you can actually use at 3am. Incidents are won or lost here, long before they happen.

Detection and analysis. The triage loop from Chapter 3, run under pressure with people asking for updates.

Containment. Stop the spread. Short-term containment first — isolate the host, disable the account — before you go looking for elegance.

Eradication. Remove the actual cause. Not the symptom.

Recovery. Return to normal, with monitoring turned up on the affected systems. Watch specifically for re-entry.

Lessons learned. The phase everyone skips and the only one that changes the future.

Containment Versus Eradication

Confusing these is the most consequential error in incident response.

Containment stops the bleeding. Isolate the host from the network. Disable the compromised account. Block the outbound destination. Fast, reversible, imperfect.

Eradication removes the cause. If a credential was phished, eradication is resetting it *and* revoking active sessions and tokens *and* checking for persistence the attacker established *and* finding out where else that credential was used.

Re-imaging a machine while the credential remains valid is containment mistaken for eradication. The attacker is still logged in somewhere. This specific failure appears in an enormous share of real incident reports.

The mistake that turns an incident into a disaster: wiping the machine immediately. It is an understandable instinct — make the bad thing go away — and it destroys the evidence you need to know what actually happened and whether it is still happening elsewhere. Isolate first. Preserve. Then remediate.

Step-by-Step Guide: The First Thirty Minutes

1. **Confirm it is real.** Run the triage loop. Do not escalate on a hunch, but do not sit on it either.
2. **Note the time and start a log.** Every action, timestamped. You will be asked to reconstruct this, guaranteed.
3. **Preserve before you touch.** Memory capture if you have the capability, disk image if warranted, and export the relevant logs before retention rotates them out — this last one is where people lose evidence most often.
4. **Contain.** Network isolation over shutdown, because shutdown loses volatile evidence.
5. **Identify blast radius.** What did that account or host have access to? What else used that credential?

6. **Notify per your plan.** Who needs to know at this stage is a decision you should have made in advance, not now.
7. **Then, and only then, start eradication.**

Real Example

A phished credential leads to mailbox access.

The incomplete response: reset the password, close the ticket.

The complete response: preserve the mailbox audit logs first; reset the credential *and* revoke all active sessions and tokens; check for mailbox rules the attacker created. Forwarding rules are the standard move and they survive a password reset; check whether that credential was reused on other systems; look for the same phishing message in other inboxes; check whether anything was sent from the account.

The password reset alone leaves the attacker with an active session, quietly forwarding mail to an external address. That specific gap is real, common, and it has cost organizations very large amounts of money.

Case Study: The Incident That Was Actually Three

An organization detects and remediates a compromised workstation. Machine re-imaged, credential reset, incident closed in a day.

Six weeks later, a second machine. Then a third.

The investigation that should have happened the first time — where did the initial access come from, was the credential used anywhere else, is there persistence — was skipped in favor of speed. Each incident was treated as isolated because nobody looked for the connection.

The underlying cause turned out to be a single unpatched internet-facing service that had been the entry point all three times.

This is what eradication failure looks like in practice. It rarely announces itself; it just recurs.

Templates: The Runbook Skeleton

Write one per scenario, calmly, in advance.

SCENARIO:	e.g. phished credential
TRIGGER:	how we know this is happening
FIRST ACTIONS:	in order, with owners
PRESERVE:	what evidence, and how, before touching anything
CONTAIN:	specific steps
ERADICATE:	specific steps, including sessions/tokens/persistence
VERIFY:	how we know it worked
NOTIFY:	who, when, and who decides on external notification
ESCALATE TO:	names and out-of-hours contact

A runbook written calmly is worth ten written during an incident. Writing one is also an excellent lab exercise and a strong portfolio piece.

Reflection

- If something real happened tomorrow, what would you do in the first ten minutes?
- Do you know how to preserve evidence in your environment, specifically?
- Has your organization ever tested a restore?

Journal

Write about a time you fixed something quickly and later found you had not actually fixed it. What did you skip, and why did skipping it feel reasonable at the time?

Exercises

1. Write one runbook for one scenario, before you need it.
2. In your lab, practise isolating a host and preserving logs before remediating.
3. Find out — genuinely — whether your organization has tested a restore in the last year.

AI Prompts

- "Walk me through the difference between containment and eradication for [scenario], and list the eradication steps people most commonly skip."
- "Help me write a runbook for [scenario]. Ask me questions about my environment first rather than assuming."
- "For a compromised credential, list everything that should be checked beyond resetting the password."

Books

- *Sandworm* — Andy Greenberg
- *Site Reliability Engineering* — Google, on incident command structure

Resources

- **NIST SP 800-61**, the incident handling guide — dry, and the standard reference
- **SANS** incident handling materials

Checklist

- I know the six phases and what each is for
- I preserve before I remediate
- I understand containment as distinct from eradication
- I have written at least one runbook in advance
- I log my actions with timestamps during an incident

Common Mistakes

- Wiping the machine in the first ten minutes
- Password reset without session and token revocation
- Treating repeat incidents as isolated
- No runbook, so the first ten minutes are improvised
- Skipping lessons learned because everyone is exhausted

Key Takeaways

- Preparation decides outcomes more than skill during the event
- Containment is not eradication
- Preserve first; evidence destroyed in minute five cannot be recovered in hour five
- Recurrence is the signature of incomplete eradication

Next Mission

The part that is not technical at all, and that determines how the incident is remembered.

The Human Part: Communicating Under Pressure

MISSION

To make your presence during an incident reduce chaos rather than add to it.

Objectives

- Communicate clearly while under pressure and observation
- Run a review that produces truth rather than blame
- Own your part without collapsing

Why It Matters

People remember how you behaved during the bad day far longer than they remember the incident itself. Careers are made and damaged in these hours, and almost none of it is technical.

Professional Explanation

The Three-Part Update

During an incident, every update contains three things:

1. **What we know.**
2. **What we do not know yet.**
3. **When I will next update you.**

All three, every time. The third is what stops people asking every ten minutes, and the second is what stops leadership building plans on assumptions.

Most people give only the first, which sounds more confident and creates much more chaos.

Things That Hold Up Under Pressure

- **Do not speculate on impact in front of leadership.** Guesses become quotes and quotes become commitments. "We do not know the scope yet, I will have a better answer at three" is a complete and professional answer.
- **Timestamp everything you do.** You will reconstruct it later.
- **Do not assign blame during the incident.** Ever. It stops people telling you things, and during an incident you need people telling you things more than you need to be right

about how it started.

- **Say when you are out of your depth.** Early. It is far cheaper at minute ten than at hour six.
- **Manage your own state.** Eat something. Hand off if you have been at it too long. Tired people make the expensive mistakes, and staying on out of pride has caused more damage than most technical errors.

Extreme ownership applies exactly here. When something goes wrong on your watch, own your part plainly and immediately, and go straight to what you are doing about it. No hedging, no distributed blame. I have failed more times than I can count — more than a hundred, probably more than five hundred, across businesses and hires and decisions. What I have learned is that the fastest route through a failure is straight through the middle of it, out loud.

Framework: The Blameless Review

After it is over, run the review this way or do not run it.

- 1 — Timeline first, no interpretation.** Just what happened, when. Build it collaboratively. Disagreements about the timeline are usually where the real learning is.
- 2 — Ask what made the wrong action reasonable at the time.** Not "why did you do that" — "what information did you have that made that the sensible choice?" This single reframe is the entire technique. People acted reasonably given what they knew; if the outcome was bad, the gap is usually in what they knew.
- 3 — Identify system gaps, not person gaps.** If one person's mistake could cause this, the system is the problem. A process that needs everyone to be perfect is a process that will fail.
- 4 — Assign owners and dates to fixes.** A review with no owners is a meeting.
- 5 — Publish it.** Internally, honestly. Hidden reviews teach nobody, and organizations that hide reviews repeat incidents.

Why Blameless Actually Works

It is not kindness. It is information strategy.

In a blaming culture, people withhold. They do not report the mistake, do not mention the thing they tried, do not volunteer that they clicked the link. You lose exactly the information you need, and you lose it fastest during the incidents that matter most.

In a blameless culture, someone says "I clicked it" within minutes, and that is worth more than any detection you could buy.

Real Example

Two ways of delivering the same update to leadership two hours into an incident.

Version A: "We think it's contained. It looks like it was limited to one machine. Should be back to normal soon."

Version B: "Here's where we are. We've isolated one workstation and disabled the affected account. What we don't know yet is whether that credential was used elsewhere — we're checking now. I'm not going to guess at scope until we've finished that. I'll update you at four regardless of what we find."

Version A sounds better in the moment and creates a serious problem if the scope turns out to be larger, because leadership has already told someone else it was contained.

Version B is what senior people sound like.

Case Study: The Review That Found the Real Problem

An incident is caused by an engineer applying a change without following the change process.

The blaming version identifies the engineer, reprimands, and reminds everyone to follow process. Nothing changes; the same thing happens eight months later with a different person.

The blameless version asks what made skipping the process reasonable at the time — and learns that the change process takes three days, that this change was in response to an urgent request from a senior stakeholder, and that essentially everyone on the team routinely skips it under those conditions because there is no faster path.

The finding is not "an engineer made a mistake." It is "our change process has no emergency path, so it gets bypassed under pressure by design."

One of those findings prevents recurrence. The other assigns blame and preserves the cause.

Templates: The Incident Update

STATUS: [investigating / contained / eradicating / recovering]
WHAT WE KNOW:
WHAT WE DON'T KNOW YET:
CURRENT ACTIONS:
IMPACT SO FAR: facts only, no estimates
NEXT UPDATE: specific time

Templates: The Review Agenda

1. Timeline, collaboratively built, no interpretation (30 min)
2. For each decision point: what did we know then? (20 min)
3. System gaps identified (20 min)
4. Fixes, with named owners and dates (15 min)
5. What went well – genuinely, name it (5 min)

That last item is not filler. Reviews that only find faults train people to dread them, and dreaded reviews get skipped.

Reflection

- How do you behave when things go wrong and people are watching?
- Have you ever withheld information because you were afraid of the response?
- When you run or attend reviews, do they produce truth?

Journal

Write about a mistake you have not told anyone about. You do not have to tell anyone now. But notice what kept it in — that mechanism is exactly what a blaming culture creates at scale.

Exercises

1. Write an incident update using the template for a scenario in your lab.
2. Practise the three-part update out loud. The "what we do not know" part is harder to say than it looks.
3. Run a small blameless review on something that went wrong recently, even something minor.

AI Prompts

- "Here is my incident update draft. Am I speculating anywhere? Rewrite it separating facts from estimates."
- "Help me phrase this review question so it asks what made the action reasonable rather than why the person did it."
- "I need to tell leadership that the scope is larger than we said two hours ago. Draft that message."

Books

- *Extreme Ownership* — Willink and Babin
- *The Field Guide to Understanding Human Error* — Sidney Dekker. The definitive argument for why blameless review works.
- *Crucial Conversations* — Patterson et al.

Podcast Episodes

Wake Up X episodes where guests discuss failure openly. Notice that the most senior people are the most direct about their mistakes — that correlation is not accidental.

Checklist

- My updates include what I do not know and when I will next update
- I never speculate on impact to leadership
- I timestamp my actions during an incident
- I have never assigned blame during an incident
- My reviews ask what made the action reasonable

Common Mistakes

- Updates that project false confidence
- Speculating on scope early
- Blame during the incident, which shuts down information
- Reviews that identify a person instead of a system gap
- Skipping the review because everyone is exhausted — which they will be, and it still matters

Key Takeaways

- Say what you know, what you do not, and when you will update

- Blameless review is information strategy, not kindness
- If one person's mistake could cause it, the system is the problem
- People remember how you behaved on the bad day

Next Mission

Book 4 answers the question that decides how far you go in this field: what does the business actually need from you?

Appendix: Runbooks, Escalations, and the Review

REFERENCE

Everything from this book in one place.

The Risk Sentence

```
If [threat] exploits [weakness],  
then [impact],  
and the likelihood is [assessment, with evidence].
```

The Three Questions Before Escalating

1. Has this happened before?
2. Is it happening elsewhere?
3. Who would know?

The Triage Loop

- | | |
|----------------------------------|---------------------------|
| 1. What fired, and why? | read the rule |
| 2. Is this normal here? | baseline |
| 3. What else happened around it? | widen the timeline |
| 4. What's the blast radius? | sets urgency |
| 5. Verdict | TP / FP / benign TP |
| 6. What do I change? | tune, or document why not |

The Escalation Note

WHAT: one line
WHEN: timestamp range
WHO/WHAT: user, host, IP, process, parent
TIMELINE: in order, with times
RULED OUT: what I checked that came back benign
WHY IT'S REAL: the evidence
BLAST RADIUS: what this entity can reach
WHAT I'VE DONE:
WHAT I NEED:

The Tuning Record

RULE:
DATE:
NOISE BEFORE:
WHY IT'S NOISY:
CHANGE MADE: narrow, by cause
WHAT THIS NOW MISSES: be honest
WHY THAT'S ACCEPTABLE:
NOISE AFTER:
REVIEW DATE:
CHANGED BY:

The Baseline Notes

CROWN JEWELS:
KEY SYSTEMS + OWNERS:
SCHEDULED THINGS:
ODD-HOUR NORMAL:
KNOWN NOISY ALERTS:
RECENT CHANGES:
WHO TO ASK ABOUT [system]:

The Runbook Skeleton

```
SCENARIO:
TRIGGER:
FIRST ACTIONS:    in order, with owners
PRESERVE:         before touching anything
CONTAIN:
ERADICATE:        including sessions, tokens, persistence
VERIFY:
NOTIFY:
ESCALATE TO:      names + out-of-hours
```

The Incident Update

```
STATUS:
WHAT WE KNOW:
WHAT WE DON'T KNOW YET:
CURRENT ACTIONS:
IMPACT SO FAR:    facts only
NEXT UPDATE:      specific time
```

The Six Phases

PREPARE → DETECT → CONTAIN → ERADICATE → RECOVER → LEARN

Containment is not eradication.

A Closing Note

Everything in this book comes down to one habit: slowing down enough to think, in an environment designed to make you hurry.

The queue creates urgency. Leadership creates urgency. The alert itself is written in a tone of urgency. And most of it is not urgent — most of it is noise. The discipline is to work through it steadily so that you have attention left for the small fraction that genuinely matters.

I came into this field sideways, through operations and through running a business badly enough to learn something. What that taught me is that the technical part is the part you can learn.

The judgment — knowing what matters, saying what you do not know, owning the bad day — is what takes years. It is what people are actually hiring.

You will be wrong sometimes. Everyone is. Be wrong in the cheap direction, write down what you learned, and be less wrong next time.

Read the rule. Build the timeline. Say what you don't know.

Connect

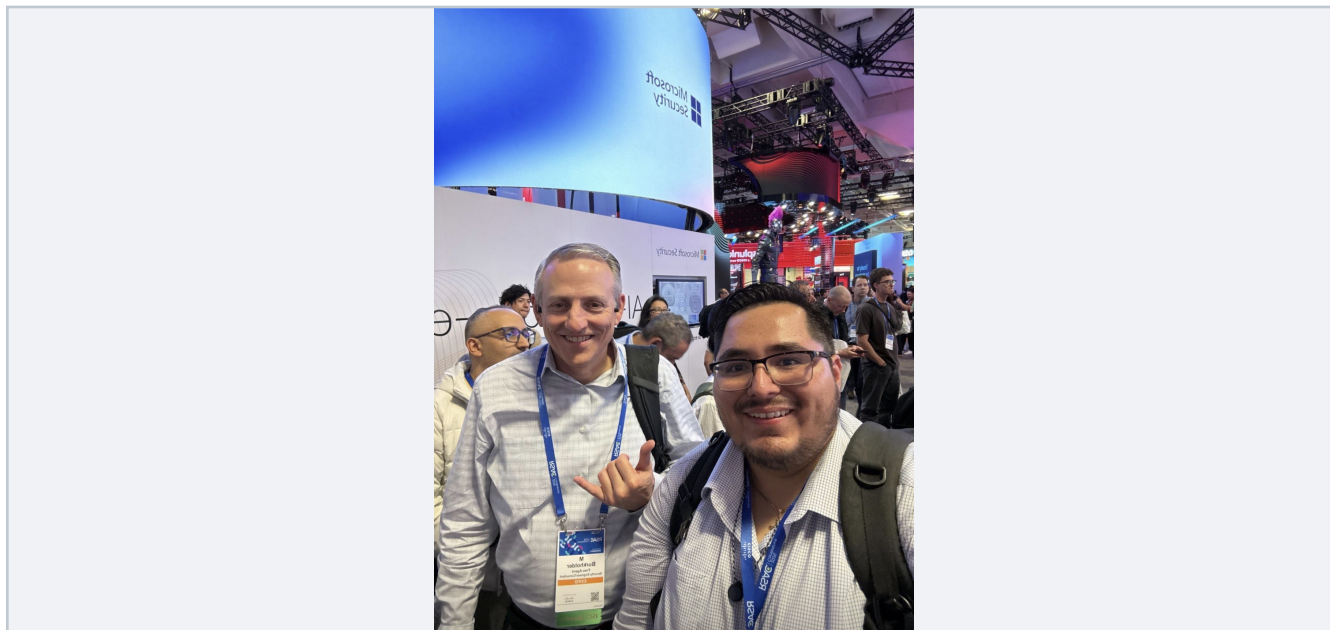
[linkedin.com/in/efrenlsalazar](https://www.linkedin.com/in/efrenlsalazar) | wakeupxpodcast.com

Questions for the author. This book has the largest gap of the three: it needs a real incident from your own career, appropriately anonymized. Even a small one. Chapters 5 and 6 are currently carrying constructed examples where a moment when you were the person everyone was looking at would land much harder. Also worth adding: your own view on whether operations background or security training served you better, since your path came in sideways and readers with similar backgrounds will want to hear it.

IN THE FIELD

In the Room With the Industry

Time on the floor at the world's largest security conferences, talking directly with the vendors building today's defenses.



RSAC 2026 — Microsoft Security expo floor



RSAC 2026 — with the Ziggiz team

Aaron Portnoy — Curiosity Is the Whole Discipline

Aaron Portnoy has been in offensive security since he was 15, learning entirely through IRC in the early 2000s, back when no cybersecurity degree existed to take. He's now Chief Product Officer at Mind Guard, an AI red-teaming company, and a fellow at Dartmouth helping shape PhD curriculum — with no certifications and no formal degree of his own.

A Vulnerability Is Just a Deviation

His clearest definition of the whole discipline: "a vulnerability is effectively just a deviation of behavior. In order to understand what a deviation is, you need to understand how it's supposed to behave." His advice for anyone starting out isn't a cert path — it's getting obsessively curious about how something actually works, down to the level where you'd notice it acting wrong.

AI Security Is Reliving the 1990s

His team has never tested an AI target without finding a vulnerability, and never found a guardrail they couldn't bypass. He predicts a major, high-profile AI breach this year — the reality-check moment that shows AI systems have the same underlying problems as any other software, just newer and less examined.

The Risk Gets Real in Healthcare

His team found real vulnerabilities in clinical AI tools — including a prescribing application that could be manipulated into recommending triple the safe dose of a controlled substance. His point: a flaw in a used-car website and a flaw in something touching a human life are not the same category of risk, even if the underlying bug looks similar.

Watch the full episode: youtu.be/OC8PqZXJHC4

Aaron's Advice, In Practice

Free resources are legitimate resources.

CTF competitions and conference talks published free on YouTube are real, hands-on ways to learn without needing a home lab or a college budget.

Don't fear AI as a full replacement — use it as a force multiplier.

"If you think of it solely as it's gonna do everything I do, then what you do is probably not that hard. If you think of it as a supplement, it's a force multiplier."

Start becoming familiar with AI before it takes you by surprise.

You don't need to be fluent — you need to understand what it can do well enough that it doesn't blindside you when it shows up in your work.

A non-traditional path is a real path, not a workaround.

He built an entire career and now teaches at the doctoral level with no certifications — curiosity and hands-on practice were the credential.

REFLECTION

What's one system or tool you use daily but don't actually understand how it works underneath? What would it take to actually find out?

BE OUR NEXT GUEST

Your Story Belongs on Wake Up X

If Think Like a Cybersecurity Professional gave you something worth using, you already have something worth sharing. The Wake Up X podcast exists to put real, unpolished career and business stories in front of people who need to hear them — and every guest so far started exactly where you are now: reading, building, and figuring it out in public.

You might be a fit if you:

Broke into tech, cybersecurity, or business without the traditional path

Built something — a lab, a business, a portfolio, a community — and can talk about how

Have a failure or turning point you're willing to tell straight, not polished

Are mentoring, leading, or building something worth other people knowing about

HOW TO APPLY

Share your story at wakeupxpodcast.com/guest, or connect directly on LinkedIn at linkedin.com/in/efrenlsalazar and mention this book.

Being a guest is also visibility — the same principle Book 7 makes the case for.

Acknowledgments

THE PEOPLE WHO MADE THIS POSSIBLE

None of Wake Up X happened alone, including Think Like a Cybersecurity Professional. Every book in this library, the podcast, and the free library exist because specific people showed up for a kid nobody expected much from. This is the record of who they are.

THE EARLY YEARS

Miss England, the 6th-grade teacher who stayed after class, unpaid, to teach the fundamentals. Sister Cora and Lewis Bair of Computers for Kids, who handed a teenager a key and a broken machine and let the work speak. Jose Luis Bravo, Angel Martinez, Isaac Murrilo, and Alex De Haro — the friends and cousins who translated the world, gave hope, drove him where he needed to go, and carried him before there was anything to carry him toward.

THE BUSINESS YEARS

Rojelio Salazar, his cousin and first mentor, RNS Computer Repairs — the reason it is called Wake Up X. Kyle Erni, his first business partner. Erick and Pancho Arauza, co-founders of The Cellphone Fix. Danny Maya, of Maya Traditional Mexican Restaurant, Taqueria Maya, and Salt and Water, who pushed him in front of a camera for the first time.

THE CAREER YEARS

Igor, who gave him the vocabulary for work he already knew how to do. Nate, who he mentored into a full-time role alongside his own. Ken Kerrick, opponent turned friend in the ISACA Sacramento election. Alvin Jue and David De los Reyes, who believed in the middle of it, not just at the finish.

THE PODCAST

Every guest who trusted an unproven show with their story, including Gustavo Mastroianni, Cathy Stephenson, Greg McCord, Aaron Portnoy, Sana Talwar, Jorge Avila, Anil Reddy, Devin Prasad, Larry Switzer, Louis Kemp, Mohana Balakrishnan, Eddie Lopez, and Michael Fierros — full episodes at wakeupxpodcast.com.

COMMUNITY PARTNERS

ISACA Sacramento and ISC2 Sacramento, The Cellphone Fix, A.N.U.L Technologies, AISOC, AspireByte, and PodGrabber — the organizations building alongside this movement.

AND

His parents, who arrived with nothing, worked since childhood, and gave away whatever was left over. Nothing about the first version of this story predicted the version you are holding now.

Continue the Journey

EVERY BOOK LEADS SOMEWHERE ELSE

Think Like a Cybersecurity Professional is one part of a 12-book system. If it helped, the next step is one page away — take whichever layer matches where you are standing.

THE LIBRARY

Layer 1 — The Story

I Just Have To Wake Up — why any of this exists.

Layer 2 — The Mindset

Comfort Is Expensive — the real cost of playing it safe.

Layer 3 — The Blueprint

DECODE: The Wake Up X Playbook — zero experience to executive thinking. Book 0.

Layer 4 — The Playbooks

1 Break In · 2 Build It · 3 Think Like a Cybersecurity Professional · 4 Business First · 5 Lead Without a Title · 6 AI for IT & Cybersecurity · 7 Build Your Brand · 8 The Consultant Mindset · 9 Level Up

FIND US EVERYWHERE

Podcast & free library: wakeupxpodcast.com · Spotify: open.spotify.com/show/28GZnRBrIV2efHOhZdiftt · Apple Podcasts: podcasts.apple.com/us/podcast/id1792821643 · YouTube: youtube.com/@Efren.wakeupx · LinkedIn: linkedin.com/in/efrenlsalazar · Substack: substack.com/@wakeupxefren

IN GOOD COMPANY

ISACA Sacramento, ISC2 Sacramento, The Cellphone Fix, A.N.U.L Technologies, AISOC, AspireByte, and PodGrabber — the partners and collaborators building alongside the movement. See the full list at wakeupxpodcast.com/community.html.

Get DECODE free, apply to be a guest, or read the story behind the movement — all of it starts at wakeupxpodcast.com.

WAKE UP X

SOLUTIONS



Efren L. Salazar

FOUNDER · HOST · CYBERSECURITY PROFESSIONAL · EDUCATOR · COMMUNITY BUILDER

STAY CONNECTED

Podcast & Blueprint Library:	wakeupxpodcast.com
LinkedIn:	linkedin.com/in/efrenlsalazar
YouTube:	youtube.com/@Efren.wakeupx
Be a Guest:	wakeupxpodcast.com/guest

THE WAKE UP X PUBLISHING SYSTEM

BOOK 4

Business First

Finance, GRC, sales, and why the business hired you

Technology is a cost center until you connect it to money and risk

THE
WAKEUPX
SOLUTIONS

EFREN L. SALAZAR

Founder & Host, Wake Up X | IT Operations Manager & Solution Engineer
Government & Regulatory Partner, ISACA Sacramento | CompTIA Security+

wakeupxpodcast.com

Educational content — not professional, legal, financial, or career-placement advice.
Full disclaimer in DECODE, the foundation of this library.

Before You Begin

HOW TO USE THIS BOOK

This book teaches the language that decides how far you go. Most technical people never learn it — which is exactly why learning it works so well.

Where this book sits

Two books come first — the memoir *I Just Have To Wake Up* and the mindset book *Comfort Is Expensive* — before this ten-book library. Each one is short on purpose. Pick up the one that matches the problem in front of you right now.

- n **Book 0 — DECODE.** The foundation. The map of the whole path.
- n **Book 1 — Break In.** Landing your first role in IT or cybersecurity.
- n **Book 2 — Build It.** Home labs and a portfolio that prove you can do the work.
- n **Book 3 — Think Like a Cybersecurity Professional.** Detection, triage, and incident response thinking.
- n **Book 4 — Business First.** Finance, sales, GRC, and why the business hired you. **you are here**
- n **Book 5 — Lead Without a Title.** Influence, EQ, and mentoring before anyone promotes you.
- n **Book 6 — AI for IT & Cybersecurity.** Using AI as an actual working tool, not a novelty.
- n **Book 7 — Build Your Brand.** LinkedIn, content, and being findable.
- n **Book 8 — The Consultant Mindset.** Discovery, solution selling, and stakeholder management.
- n **Book 9 — Level Up.** Long-term growth, burnout, legacy, and mentoring others.

Book 0 is the foundation. *DECODE: The Wake Up X Playbook* holds the six-level path, the Curiosity Compass, and the core frameworks. Nothing in any other book contradicts it. If a term here is unfamiliar, DECODE defines it.

What is in this book

Part One — How the Business Actually Works

1. The Business Did Not Hire You to Do Technology
2. Making the Case: Proposals That Get Funded

Part Two — The Lane Nobody Wants

3. GRC: Why the Boring Thing Pays
4. Risk, Written Down

Part Three — Everyone Sells

5. Everyone in Technology Sells
6. Contracts, Hiring, and the Expensive Lessons
7. Appendix: Proposals, Narratives, and Scripts

How to actually use it

This is the least technical book in the library and probably the highest-leverage one. The material here is what separates people whose careers plateau at senior engineer from people who keep going.

If you are early in your career, read Chapters 1 and 5 and come back to the rest when you have more context. If you are mid-career and feel stuck despite being technically strong, this book is likely the thing you are missing.

A warning about the wrong lesson. Learning business language is not the same as becoming less technical. The people who benefit most from this book are the ones who keep their technical depth and add this on top. Purely administrative business-adjacent roles get automated and outsourced. The combination does not.

I am writing this the way I would say it to you sitting across a table — not lecturing a room. Take what works, remix the rest into your own version, and leave what does not fit.

PART

ONE

How the Business Actually Works

The Business Did Not Hire You to Do Technology

MISSION

To make you fluent in how a company actually works and what it needs from you.

Objectives

- Understand where your role sits in how the organization makes money
- Speak about your work in business terms without becoming a suit
- Recognize why this is the largest career multiplier available to you

Story

I have run a business, not just worked in one. My first company repaired cell phones and computers. I handled the technical side; my partner handled sales and business. That partnership worked exactly because our skills did not overlap.

Here is what I learned that I did not expect: our actual competitive advantage was never repair quality. Plenty of people could fix a phone, and several of them were better at it than I was.

Our advantage was customer service — how people were treated, how clearly things were explained, whether we did what we said we would.

That is an uncomfortable thing for a technical person to hear, and it was completely true. Being the best technician in the room is not the same as being the most valuable person in the room. It took running a business to learn it, and I would rather you learn it from a chapter.

Why It Matters

Here is the structural truth most technical people avoid: **technology is a cost center in most organizations.**

You are an expense on a spreadsheet unless someone can connect your work to revenue, risk, or cost. Learning to make that connection yourself — rather than hoping your manager makes it for you — is the difference between being cut in a downturn and being protected in one.

Specialists are the ones getting laid off right now. Breadth across networking, security, IT, programming fundamentals, and hands-on

AI — plus business literacy on top — is what actually protects a career.

This is not a comfortable observation and I am not going to soften it. The narrow deep specialist is more exposed than they believe. The person who can bridge technical work and business need is more protected than their technical peers assume.

Professional Explanation

How Money Moves



You are in the third row. Your job is to talk about the fourth.

Where you sit in the money. Your work lives in the third row. Your conversations should be about the fourth.

Every organization has the same basic shape, whatever it calls things.

- **Revenue** — money in. Who pays, and for what.
- **Cost of goods sold** — what it costs to deliver the thing that was sold.
- **Operating expenses** — everything else, including almost all of IT and security.
- **Margin** — what is left. This is what leadership is actually managing.
- **Cash flow** — timing. Profitable companies die of cash flow problems with some regularity. That is why "we cannot spend it this quarter" is sometimes a real constraint rather than a brush-off.

When you propose a project, leadership is asking one of four questions:

1. Does this **make money**?
2. Does this **save money**?
3. Does this **reduce risk**?
4. Is this **required** — by law, regulation, or contract?

If your proposal answers none of those, it does not matter how technically sound it is. It will not be funded. The reason will not be explained to you.

Translating Technical Into Business

What you'd say	What they need to hear
We need to upgrade the servers	The current hardware is out of support, so a failure means a full day of downtime — roughly the cost of a day of operations
We should implement MFA	Phished credentials are the most common breach entry point; this closes it for about the cost of one laptop per year
The backups aren't tested	We have never verified we can actually restore, so our recovery plan is currently a guess
We need a SIEM	Right now we would not know about a breach until a customer told us; this gives us the ability to find it ourselves
We should segment the network	If one machine is compromised today, the attacker can reach everything. This limits it to one department.

Same facts. Second column gets funded.

Notice what the second column does not contain: product names, acronyms, and technical mechanism. That is not dumbing down. It is answering the question that was actually asked.

Learn What Your Organization Sells

An underrated move: find out how your employer actually makes money, in detail. Who the customers are. What they pay for. What the margin looks like. What the biggest cost is. What keeps leadership awake.

Most technical staff cannot answer these about their own employer. The ones who can make consistently better decisions, because they can tell which systems are genuinely load-bearing and which merely feel important.

Framework: The Value Translation

THE VALUE TRANSLATION



Notice the technical solution barely appears. Nobody buys the solution.

The Value Translation. The technical solution barely appears, deliberately.

Take any technical proposal through four steps.

1 — What breaks or is missed today? Concretely. Not "our security posture is weak" — "we cannot tell whether a compromised account was used, because we retain logs for seven days."

2 — What does that cost? Time, money, risk, or opportunity. Estimate honestly and say clearly that it is an estimate. A stated range with reasoning always beats "it depends."

3 — What does the fix cost? Money, hours, and friction. Include the friction. Leaving it out destroys your credibility when it appears anyway, and it always appears.

4 — What is the recommendation, and what happens if we do nothing? Always give the do-nothing option. It shows you are advising rather than advocating. It is the single fastest way to be treated as a peer rather than a requester.

Notice the technical solution is barely in this framework. That is deliberate. Nobody buys the solution. They buy the removal of the problem, and they are trusting you on the mechanism.

Real Example

Two versions of the same request for log retention.

Version A: "We only keep logs for seven days. Industry best practice is a year. We need to increase retention."

Version B: "Right now we keep logs for seven days. That means if we found out today that something happened three weeks ago — which is roughly the industry average time to detection — we would have no way to investigate it. We would be telling customers we do not know what was accessed. Extending to twelve months costs about [X] annually in storage. If we do nothing, we are accepting that any incident older than a week is uninvestigable."

Version A cites authority. Version B describes a specific bad afternoon that leadership can picture, prices it, and names the consequence of declining.

Case Study: The Project That Got Funded Twice

A pattern worth understanding.

A security team requests budget for an improvement. Declined — not a priority. Six months later, a minor incident occurs that the improvement would have prevented. The budget is approved immediately, at higher cost, under pressure.

The lesson is not "wait for an incident." It is that the first request failed to make the risk *feel* concrete. The incident did what the proposal should have done.

The way to make risk concrete without an incident is specificity: not "we could be breached" but "if a finance user is phished tomorrow, here is exactly what the attacker can reach, here is what we could and could not tell you afterward, and here is what that costs." Vague risk is easy to defer. Specific risk is uncomfortable to defer, and that discomfort is your leverage.

Reflection

- Do you know how your employer makes money?
- When you last proposed something, which of the four questions did it answer?
- Have you ever included friction in a proposal, honestly?

Journal

Write down what your organization's biggest business risk is — not the biggest security risk, the biggest business risk. Then ask whether your work connects to it. Most people find a gap, and that gap is often where the interesting work is.

Exercises

1. Take a technical thing you believe should happen and run it through the four-step value translation, in writing.
2. Find out three things about how your employer makes money that you did not know.

3. Rewrite one technical statement from the left column of the table above in your own context.

AI Prompts

- "Here is a technical proposal: [description]. Rewrite it answering: what breaks today, what that costs, what the fix costs including friction, and what happens if we do nothing."
- "I work at a [type] company. What are the likely top three business concerns of leadership there, and how might IT connect to each?"
- "Read this proposal as a CFO with limited budget. What would you ask, and where would you push back?"

Books

- *The Personal MBA* — Josh Kaufman. The single most efficient way to acquire business literacy from a standing start.
- *Financial Intelligence for IT Professionals* — Berman and Knight
- *The Phoenix Project* — Gene Kim. Fiction, and the best available illustration of IT connected to business outcome.

Podcast Episodes

Wake Up X episodes with founders and business leaders. Listen for how they describe technology — usually in terms of what it enables or costs, almost never in terms of what it is.

Resources

- Your own organization's annual report or investor materials, if public. Free, specific, and almost nobody in IT reads them.
- Your employer's sales team, who will happily explain what customers actually buy

Checklist

- I know how my employer makes money
- Every proposal I write answers one of the four questions
- I include friction in my cost estimates
- I always present a do-nothing option
- I can describe a technical need without product names

Common Mistakes

- Leading with the technology instead of the problem
- Refusing to estimate because you cannot be precise
- Presenting only your preferred option
- Treating budget constraints as ignorance rather than as real constraints
- Citing "best practice" as though it were an argument

Key Takeaways

- Leadership manages margin and risk; connect to those or be invisible
- Every proposal answers: makes money, saves money, reduces risk, or required
- Nobody buys the solution; they buy the removal of the problem
- Breadth plus business literacy is layoff insurance

Next Mission

The specific mechanics of a proposal that gets approved.

Making the Case: Proposals That Get Funded

MISSION

To give you a repeatable format that converts technical judgment into approved budget.

Objectives

- Write a one-page proposal that a busy executive will actually read
- Present options rather than advocating for one
- Handle the questions that follow

Why It Matters

Most technical proposals fail for reasons unrelated to their merit. They are too long, they lead with mechanism, they present one option, they omit cost, or they arrive without warning in a meeting where nobody was prepared to decide.

All of those are fixable, and fixing them is a skill you can acquire in an afternoon.

Framework: The One-Page Proposal

One page. Never more. If it needs an appendix, attach one — but the page must stand alone.

THE SITUATION	two sentences, in their words
WHAT IT'S COSTING	time, money, or risk – estimated honestly
THE OPTIONS	three, including do nothing
RECOMMENDATION	one, with why
WHAT I NEED	the specific decision or resource
TIMELINE	when, and what happens if we wait

Why Three Options

One option is advocacy. It puts you in the position of a person who wants something, and the reader's job becomes evaluating you rather than the problem.

Three options — typically a minimal version, a recommended version, and a comprehensive version, plus do-nothing — puts the reader in the position of deciding. That is the position they want to be in. It is a seriously different conversation.

Include do-nothing genuinely, with its real consequences. It shows that you are not simply trying to get your preferred thing funded, and occasionally do-nothing is the right answer. Saying so when it is buys you enormous credibility for the times it is not.

On Estimating Honestly

You will not have precise numbers. Nobody does.

Give a range with reasoning: "roughly [X] to [Y], depending on whether we need [factor]. I am confident about the lower bound and less confident about the upper." That sentence is more credible than a single confident number. It protects you when the actual figure lands somewhere in the range.

What destroys credibility is a precise number you cannot defend, or a refusal to estimate at all. The refusal reads as either evasion or incompetence, however reasonable it feels from the inside.

Step-by-Step Guide: From Idea to Approval

1. **Establish the problem exists** before you propose anything. If leadership does not agree there is a problem, no solution will be funded. Sometimes this is a separate, earlier conversation.
2. **Find out who actually decides**, and who influences them. Book 8 covers this in depth. The decider is frequently not the person you report to.
3. **Pre-socialize it**. Never let a proposal be a surprise in a meeting. Show it to one or two people beforehand and ask what is wrong with it. They will tell you, and you will fix it before it matters.
4. **Write the page**. Their words, not yours. Their metrics, not yours.
5. **Bring the appendix but do not lead with it**. Technical detail is what you use to answer questions, not what you use to make the case.
6. **Ask for a specific decision**. "I need approval for [X] by [date]" is a request. "Let me know your thoughts" is a document that gets filed.
7. **Follow up in writing**. Whatever was agreed, send a short summary. Book 1 makes this point about offers; it applies identically here.

Real Example

A proposal for MFA implementation, written badly and then well.

Badly: three pages beginning with an explanation of how TOTP works, followed by a vendor comparison matrix, ending with "we recommend proceeding."

Well:

Situation. Four employees received phishing emails last month; one entered credentials. We caught it because the login came from an unusual location, which was luck rather than process.

What it's costing. Right now a single phished password gives full access to email and file storage. We would not necessarily know for days.

Options. (1) Do nothing — accept that one phished password is a full account compromise. (2) MFA on email and remote access only — about [X], roughly two weeks, covers the highest-risk path. (3) MFA everywhere. About [Y], six to eight weeks, some friction on legacy applications.

Recommendation. Option 2 now, option 3 next year. Option 2 covers the realistic attack path at a fraction of the cost and disruption.

What I need. Approval for [X] and two weeks of my time, by the end of this month.

Timeline. If we wait until next quarter, we are accepting the current exposure through the busiest period of the year.

Half a page. Someone can decide from it.

Case Study: The Pre-Socializing Habit

The single most effective change most technical people can make to how their ideas land.

A proposal that arrives cold in a meeting has to survive first impressions, unfamiliar framing, and whatever mood the room is in. A proposal that two people have already seen arrives with those two people already understanding it, and frequently already supporting it.

More importantly, showing it early surfaces objections while there is still time to address them. The objection you hear in private on Tuesday is information. The same objection raised in the decision meeting on Friday is a rejection.

This costs two conversations. It is not politics; it is preparation.

Templates: The Follow-Up Email

Subject: Summary – [topic] discussion

[Name],

Quick summary of what we agreed:

- [decision]
- [owner and date]
- [anything still open]

Let me know if I've captured anything incorrectly.
Otherwise I'll proceed on that basis.

Two minutes. Prevents an enormous amount of later confusion, and creates a record.

Reflection

- When your ideas have not landed, was it the idea or the framing?
- Do you know who actually decides on spending in your organization?
- Have you ever pre-socialized a proposal?

Journal

Write about a good idea of yours that went nowhere. Be honest about how it was presented. Most technical people find at least one that failed on delivery rather than merit.

Exercises

1. Write a one-page proposal for something you actually want, using the template.
2. Show it to one person and ask what is wrong with it. Do not defend — just listen.
3. Rewrite it based on what they said.

AI Prompts

- "Here is my proposal. Rewrite it as a one-page document with three options including do nothing, in language a non-technical executive would follow."
- "Read this as a skeptical CFO. What are the three questions you would ask, and where is the weakest part?"
- "I need to estimate the cost of [scenario] and I do not have precise figures. Help me build a defensible range and name my assumptions."

Books

-
- *The Personal MBA* — Josh Kaufman
 - *Made to Stick* — Chip and Dan Heath. On why concrete beats abstract, which is the entire mechanic of a good proposal.
 - *Crucial Conversations* — Patterson et al.

Checklist

- My proposals fit on one page
- I always present three options including do nothing
- I estimate with ranges and state my assumptions
- I pre-socialize before any decision meeting
- I ask for a specific decision by a specific date
- I follow up in writing

Common Mistakes

- Leading with mechanism
- One option, which reads as advocacy
- No cost estimate, or a precise one you cannot defend
- Surprising the room
- Ending with "let me know your thoughts"

Key Takeaways

- One page, three options, do-nothing always included
- A defensible range beats both false precision and refusal to estimate
- Pre-socializing converts objections into information
- Ask for a specific decision by a specific date

Next Mission

The lane that pays well exactly because everyone finds it boring.

PART

TWO

The Lane Nobody Wants

GRC: Why the Boring Thing Pays

MISSION

To show you why governance, risk, and compliance is one of the most underrated career lanes in this industry.

Objectives

- Understand what the major frameworks actually need
- See GRC as translation and judgment rather than paperwork
- Know how to enter this lane from a technical background

Why It Matters

GRC is where technical knowledge, business language, and legal requirement meet. It is chronically understaffed, it pays well, and — critically — it is one of the hardest lanes for AI to automate, because it is at its core about judgment and accountability rather than output.

It is also the lane most technical people dismiss as boring. That is exactly why there is room in it.

I ended up moving toward compliance work for a specific reason, and it was not that I found it fascinating. It was that I had learned, expensively, what happens when things are not written down. That lesson is the whole discipline.

Professional Explanation: The Landscape

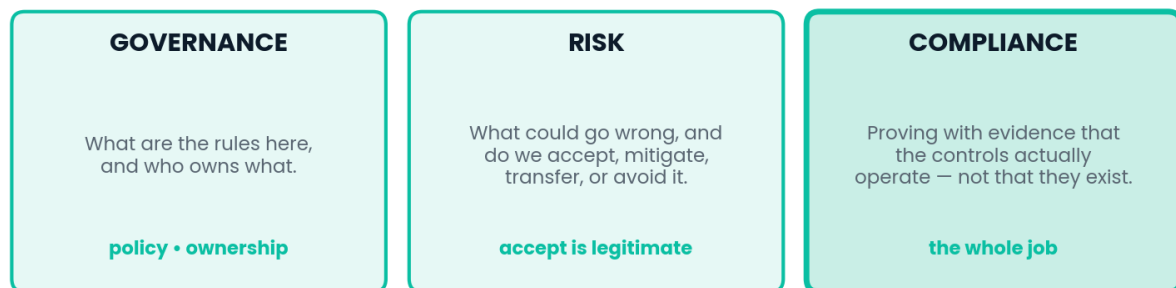
Framework	What it governs	Who it applies to
PCI DSS	Payment card data	Anyone who stores, processes, or transmits card data
HIPAA	Protected health information	Healthcare providers, insurers, and their vendors
GDPR	Personal data of EU residents	Any organization handling that data, wherever they are

Framework	What it governs	Who it applies to
SOC 2	Service organization controls — security, availability, confidentiality	Service providers, especially SaaS; usually customer-driven
ISO 27001	Information security management systems	International, certifiable, common in enterprise procurement
NIST CSF	Voluntary framework — Identify, Protect, Detect, Respond, Recover	Widely adopted in the US; often the internal organizing structure
CIS Controls	Prioritized, prescriptive technical safeguards	Anyone; the most practical starting point for a small team

If you are starting out, start with **CIS Controls**. They are ordered by impact and they tell you what to actually do, which none of the others quite manage. The others tell you what must be true; CIS tells you what to go and do on Monday.

What GRC Work Actually Is

WHAT GRC WORK ACTUALLY IS



The three loops. The third one is where the real work lives.

Governance — deciding what the rules are here, and who owns what. Policy, ownership, accountability.

Risk — identifying what could go wrong, assessing likelihood and impact, and deciding to accept, mitigate, transfer, or avoid. Note that *accept* is a legitimate outcome, and Chapter 4 is entirely about that.

Compliance — proving, with evidence, that the controls actually operate. Not that they exist. That they *operate*.

That last distinction is the entire job. A policy nobody follows is worse than no policy, because it documents that you knew what should happen and did not make it happen.

The Difference Between Existing and Operating

An access review policy exists if there is a document saying access will be reviewed quarterly.

The control *operates* if you can show, for each of the last four quarters: who performed the review, on what date, what they found, what was revoked as a result, and who approved it.

Auditors ask for the second thing. Organizations frequently have only the first. Bridging that gap is what a competent GRC practitioner does, and it needs knowing enough technically to know what evidence is even possible to produce.

Step-by-Step Guide: Entering GRC From Technical Work

1. **Start where you are.** Map the controls you already touch — access reviews, patching, logging, backups — to a framework. CIS is the easiest entry.
2. **Volunteer for the audit.** Every organization dreads evidence collection. Volunteering makes you instantly useful and teaches you the entire discipline from the inside, faster than any course.
3. **Learn to write a control narrative.** What the control is, who owns it, how often it runs, and what evidence proves it ran. This is the core artifact of the job.
4. **Get one credential once the work has started, not before.** CISA or CRISC are the recognized ones. Do them with context, or they are pure memorization.
5. **Keep your technical skills.** The valuable GRC person is the one who can read a firewall rule *and* a contract. Purely administrative compliance people get automated and outsourced; technical ones do not. I cannot stress this enough.

Templates: The Control Narrative

CONTROL:	what it is, in one sentence
FRAMEWORK REF:	which requirement this satisfies
OWNER:	a named role, not a person
FREQUENCY:	how often it runs
HOW IT WORKS:	the actual mechanism
EVIDENCE:	what artifact proves it ran
WHERE EVIDENCE LIVES:	
LAST TESTED:	
KNOWN GAPS:	be honest; auditors find them anyway

The "known gaps" line is counterintuitive and correct. Auditors trust an organization that names its own gaps much more than one that appears flawless, because the second one is either lying or not looking.

Real Example

An organization claims it performs quarterly access reviews.

What they have: a policy document stating that quarterly access reviews occur.

What an auditor asks for: evidence of the last four reviews.

What they find: two were done informally over a chat conversation with no record. One was skipped during a busy quarter. One was completed properly.

What the competent response looks like: not scrambling to fabricate the missing evidence — which is fraud, and which people genuinely attempt — but documenting the actual state, implementing a repeatable process with automatic evidence capture, and reporting the gap with a remediation date.

The organization that reports its own gap with a plan is in dramatically better shape than the one that gets caught, both practically and legally.

Case Study: Why This Lane Resists Automation

Worth understanding if you are weighing where to invest your career.

AI is good at producing documents. It can draft a policy in seconds, and it can map controls to frameworks reasonably well.

What it cannot do is *be accountable*. Compliance ultimately rests on a named human asserting that something is true, with consequences if it is not.

It also cannot navigate the genuinely political part — persuading a business unit to accept friction, negotiating with an auditor about the sufficiency of evidence, or deciding what level

of risk this specific organization should accept given its actual circumstances.

The document-production part of GRC will be automated. The judgment and accountability part will not. It is the larger part.

Reflection

- Which controls do you already perform without thinking of them as controls?
- Could you prove any of them operated last month?
- Does GRC seem boring to you, and is that a reason to avoid it or an opportunity?

Journal

Write about a time something went wrong because nothing was written down. Most people have one. That experience is the entire motivation for this discipline.

Exercises

1. Take one control you personally perform — patching, backups, access review — and write its control narrative.
2. Ask yourself what evidence would prove to a skeptical outsider that it ran last month. If you cannot answer, that is a real finding.
3. Read the CIS Controls list and identify which five your environment does worst.

Project

Produce a mini-assessment of your own home lab against the first six CIS Controls: what you have, what you do not, what evidence exists. This is an unusual and genuinely impressive portfolio piece for anyone targeting a GRC-adjacent role.

AI Prompts

- "Here is a control I perform: [description]. Write a control narrative including what evidence would prove it operated, and tell me what an auditor would likely challenge."
- "Compare what [framework A] and [framework B] need regarding [topic], and tell me where they differ in practice rather than in wording."
- "I perform these tasks: [list]. Map each to CIS Controls and identify which controls I have no coverage for."

Books

- *The Cybersecurity Manager's Guide* — Todd Barnum
- *The Personal MBA* — Josh Kaufman
- ISACA publications relevant to CISA and CRISC, once you have context

Podcast Episodes

Wake Up X episodes with guests in GRC and audit roles. Ask specifically how they came into it — almost nobody plans this lane, and the entry stories are instructive.

Resources

- **CIS Controls** — free, prioritized, practical
- **NIST CSF** — free, and a good organizing structure
- Your local **ISACA** chapter — this is the professional home of this discipline. It is where the roles get filled

Checklist

- I can name the major frameworks and who they apply to
- I understand the difference between a control existing and operating
- I have written at least one control narrative
- I am keeping technical depth alongside compliance literacy
- I have volunteered for, or asked about, the next audit

Common Mistakes

- Writing policy that describes an ideal nobody follows
- Collecting evidence that controls exist rather than that they operate
- Abandoning technical depth on entry to GRC — the career-limiting one
- Getting the certification before the context
- Presenting a flawless picture, which auditors correctly distrust

Key Takeaways

- GRC is translation and judgment, not paperwork
- Proving controls *operate* is the whole compliance discipline
- Technical depth plus compliance literacy is rare and defensible
- Naming your own gaps builds more trust than appearing perfect

Next Mission

The concept at the center of all of it, and the one most often misunderstood.

Risk, Written Down

MISSION

To make you comfortable with the idea that accepting risk is a legitimate, professional outcome.

Objectives

- Understand the four responses to risk
- Get risk acceptance owned in writing by someone with authority
- Stop treating "we cannot fix this" as failure

Why It Matters

New security and compliance people frequently believe their job is to eliminate risk. It is not, it cannot be, and behaving as though it is makes you exhausting to work with.

Your job is to make risk **visible, quantified, and owned**. What happens after that is a business decision.

Professional Explanation

The Four Responses

Every identified risk gets one of four treatments:

Mitigate. Reduce the likelihood or the impact. Implement a control. The default, and the one everyone thinks of.

Transfer. Move the financial consequence elsewhere — insurance, or contract terms with a vendor. Note that you cannot transfer reputational damage or regulatory obligation; you can only transfer money.

Avoid. Stop doing the thing that creates the risk. Decommission the system. Exit the market. Rare, and occasionally correct.

Accept. Decide the cost of addressing it exceeds the expected cost of the risk, and proceed knowingly.

Acceptance is not failure. It is frequently the right answer for a low-likelihood, low-impact risk whose fix would consume resources better spent elsewhere. An organization that mitigates everything is an organization that has stopped prioritizing.

Why Written Acceptance Protects Everyone

Risk acceptance is a business decision, not a security decision. Your job is to make the risk visible, quantified, and owned by a named person who has the authority to accept it. Once they have accepted it in writing, your job on that item is done.

That written acceptance does three things.

It protects the organization, because a documented, reasoned decision is defensible. An undocumented one looks like negligence. The difference matters enormously in a regulatory or legal context.

It protects you. If you raised it, quantified it, and someone with authority accepted it in writing, you have done your job. Without that record, the retrospective version of events is that security never flagged it — and that is the version that tends to survive.

It creates an actual decision. Verbal acceptance is often not a decision at all; it is a way of ending an uncomfortable conversation. Asking for it in writing forces the person to consider whether they genuinely mean it. A meaningful proportion of the time, the answer becomes "actually, let us fix it."

If someone refuses to accept a risk in writing but also refuses to fund the fix, that refusal is itself the information. They do not want to own it. Escalate — not aggressively, just factually: "I need this either remediated or formally accepted, and I need to know who owns that decision." Being the person who insists on this is uncomfortable for a while and then becomes the reason you are trusted.

Framework: The Risk Register Entry

RISK ID:
DESCRIPTION: as a risk sentence (see Book 3)
LIKELIHOOD: with the evidence for that assessment
IMPACT: what specifically happens, and to whom
INHERENT RATING: before any controls
CURRENT CONTROLS: what exists today
RESIDUAL RATING: after existing controls
TREATMENT: mitigate / transfer / avoid / accept
IF ACCEPTED: accepted by [name, role], date, review date
IF MITIGATED: owner, target date, cost
REVIEW DATE:

Two fields carry the weight: the evidence behind the likelihood, and the named human on acceptance. Registers without those are decoration.

Real Example

A legacy application cannot support MFA. It holds moderately sensitive data and is used by twelve people. Replacing it would cost significantly and take a year.

The unproductive response: repeatedly raise it, get nowhere, become known as the person who complains about the legacy app.

The professional response: write the risk sentence. Quantify what an attacker could access with one of those twelve credentials. Propose three options — replace, compensating controls, or accept. Recommend compensating controls: restrict the application to the internal network, enforce longer unique passwords for those accounts, monitor those twelve accounts specifically, and review quarterly.

Then get the residual risk formally accepted by the business owner, in writing, with a review date tied to the replacement roadmap.

The risk is not gone. It is now known, reduced, owned, and scheduled. That is what good looks like. It is very different from either fixing it or complaining about it.

Case Study: The Acceptance That Prevented a Blame Cycle

An organization accepts a documented risk. Eighteen months later, it materializes. There is an incident.

Because the acceptance was written — with the risk description, the reasoning, the named acceptor, and the date — the post-incident conversation is about whether the assessment was reasonable given what was known then. It is a mature conversation.

Compare the version with no record. The conversation becomes about who knew what, memories conflict, and the security team is asked why they did not raise it. They did raise it. They cannot prove it.

The written record does not prevent the incident. It prevents the incident from becoming a blame exercise that damages people who did their jobs correctly.

Reflection

- Have you ever treated risk acceptance as a failure on your part?
- Is there something you have raised repeatedly that should instead be formally accepted or escalated?

- Who in your organization actually has the authority to accept risk?

Journal

Write about a risk you know about right now that nobody owns. Then write down who *should* own it. That is usually a short conversation you have been avoiding.

Exercises

1. Write one risk register entry, complete, for something real.
2. Identify who has acceptance authority in your organization. If nobody knows, that is a governance finding.
3. Practise the sentence: "I need this either remediated or formally accepted." Say it out loud until it sounds neutral rather than confrontational.

AI Prompts

- "Here is a risk: [description]. Draft a risk register entry including likelihood with the evidence I would need, impact, and three treatment options."
- "Help me write a risk acceptance memo that is factual and non-accusatory, for a business owner to sign."
- "What compensating controls could reduce [risk] where the primary fix is not currently feasible?"

Books

- *Measuring and Managing Information Risk (FAIR)* — Freund and Jones. The serious treatment of quantifying risk.
- *How to Measure Anything in Cybersecurity Risk* — Hubbard and Seiersen

Checklist

- I understand all four treatment options
- I get acceptance in writing, from a named person with authority
- Every accepted risk has a review date
- I do not treat acceptance as a personal failure
- I escalate when someone will neither fix nor accept

Common Mistakes

- Treating acceptance as failure
- Verbal acceptance, which is not a decision
- Risk registers with no named owners
- Repeatedly raising something instead of forcing a treatment decision
- Likelihood assessments with no evidence behind them

Key Takeaways

- Four treatments: mitigate, transfer, avoid, accept
- Written acceptance protects the organization and protects you
- Refusal to accept in writing is itself information
- Your job is visibility, quantification, and ownership — not elimination

PART

THREE

Everyone Sells

Everyone in Technology Sells

MISSION

To make peace with the fact that you already sell, and to get much better at it.

Objectives

- Recognize the selling you already do daily
- Run a real discovery conversation
- Handle objections without becoming defensive

Why It Matters

Every time you propose an approach, request budget, push back on a deadline, or convince a colleague to adopt a standard, you are selling.

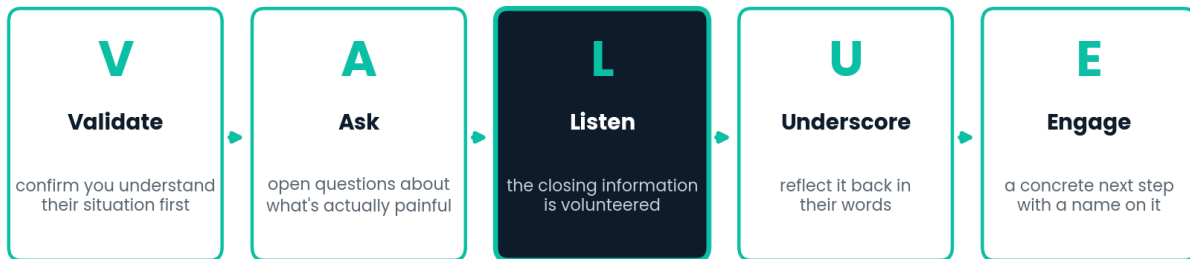
Technical people who reject this framing do it badly and without meaning to. Technical people who accept it get their ideas implemented.

Cybersecurity and IT are unusually isolated fields. They do not naturally reach for sales and communication skills, and there is a certain amount of quiet disdain for them. That is exactly why importing them creates such outsized advantage — you are competing against people who consider this beneath them.

Everything in the Wake Up X library is consciously borrowed from sales, marketing, and relationship thinking that already existed, and dragged into a field that does not reach for it. This chapter is the most direct example.

Framework: V.A.L.U.E.

V.A.L.U.E. — THE CONVERSATION SEQUENCE



The failure mode is skipping straight to Underscore because you're excited.

V.A.L.U.E. The failure mode is skipping straight to Underscore.

V — Validate. Start by confirming you understand their situation and what they care about. Not your agenda — theirs. *"Before I suggest anything, let me make sure I understand what you're dealing with."*

A — Ask. Real, open questions. What is actually painful right now? What have they already tried? What would good look like? You are not gathering ammunition; you are finding out whether your idea even applies. Sometimes it does not, and finding that out early is a win.

L — Listen. Actually listen. Most people are waiting for their turn to talk. The information that closes the conversation is almost always something the other person volunteers while you are quiet — and it usually arrives after a pause you were tempted to fill.

U — Underscore. Reflect back what you heard, in their words, and connect it explicitly to what you are proposing. *"So the real cost is the three days of rework each month. Here's how this addresses that specifically."*

E — Engage. End with a concrete next step and a name attached to it. Not "let me know what you think" — *"I'll draft the one-page version by Thursday, and if it looks right, who else needs to see it?"*

The failure mode is skipping straight to Underscore. You have a solution, you are excited about it, you lead with it. It almost never lands — because you have not established that you understand the problem, and the other person will spend the entire conversation defending their situation rather than considering your idea.

Professional Explanation

Why Validate Comes First

People do not evaluate ideas neutrally. They evaluate ideas from people, and the first thing they assess is whether you understand their situation.

An idea from someone who clearly does not understand the constraints gets dismissed regardless of quality — and often correctly, because an idea formed without the constraints usually breaks against them.

Ten minutes of genuine validation changes the reception of everything that follows. It is not a technique for manipulating people; it is how you find out whether your idea survives contact with reality.

Handling Objections

An objection is information, not rejection. Three that cover most cases:

"We don't have budget." Usually means "not a priority yet." Response: quantify the cost of doing nothing. Ask directly what would have to be true for this to become a priority. That question frequently produces the actual objection.

"We tried that before." This is gold. Ask what specifically failed. You now know the real constraint, and you can address it directly — or discover that the previous attempt failed for reasons that no longer apply.

"I need to think about it." Usually an unspoken concern. Ask gently and directly: *"Of course — is there a specific part you're unsure about? I'd rather address it now than have you working around it later."*

The pattern across all three: the stated objection is rarely the real one, and asking one more question is nearly always better than arguing with the stated version.

Real Example

An engineer wants the team to adopt infrastructure-as-code. He is right that it would help.

Skipping to Underscore: he presents at the team meeting on why the current manual process is error-prone and how Terraform would fix it. Two people become defensive because they built the current process. It goes nowhere.

Running V.A.L.U.E.: he first asks two colleagues what the most frustrating part of their week is. One says rebuilding environments after they drift. The other says onboarding new environments takes days and always misses something.

He then proposes it in their terms: *"You mentioned drift costs you most of a day every couple of weeks, and that onboarding a new environment always misses something. What if the environment definition lived in version control, so drift was visible and onboarding was a copy?"*

Same idea. Now it is their problem being solved rather than his preference being imposed. The people who built the current process are not being told their work was bad.

Case Study: Negotiation and Knowing the Room

A distinction worth being clear about.

Negotiation is real with business owners, hiring managers, and decision-makers — people with authority to change terms. It is not real in a retail transaction. Nobody at the tone can change the price, and pushing there just signals that you do not understand the situation.

Recognizing which conversation you are in is itself a skill. The signal is authority: can this person actually decide? If not, your goal is not to persuade them but to find out who can, politely.

This applies internally too. Arguing your case forcefully to someone with no budget authority wastes both of your afternoons.

Step-by-Step Guide: A Discovery Conversation

1. **Book the time properly.** "Can I get twenty minutes to understand how [thing] works for your team?" People say yes to being asked about their work.
2. **Open with validation.** State what you think their situation is and invite correction. Being corrected is the point.
3. **Ask three open questions.** What is most frustrating, what have you tried, what would good look like.
4. **Shut up.** Count to three after they finish. The second thing they say is usually the real one.
5. **Underscore in their words.** Do not translate into your vocabulary.
6. **Propose lightly,** as a question rather than a plan.
7. **End with a specific next step and a name.**
8. **Send a two-line summary afterwards.**

Reflection

- Do you lead with solutions?
- When was the last time you asked a colleague what their week is actually like?

- Which objection do you hear most, and have you ever asked what is behind it?

Journal

Write about an idea of yours that was rejected. Then write what the other person's actual constraint might have been. Most rejected technical ideas ran into a constraint that was never surfaced.

Exercises

1. Run one discovery conversation this week with someone whose work you do not fully understand.
2. Practise the three-second pause after someone finishes speaking. It is harder than it sounds and it changes conversations.
3. Take an objection you have received and write down three things it might actually mean.

Templates: The Discovery Questions

What's the most frustrating part of [process] right now?
What have you already tried?
What would "good" look like, specifically?
What's stopped this from being fixed already?
Who else does this affect?
If nothing changes, what happens?

That last one is the most useful and the least asked.

AI Prompts

- "I want to propose [change] to [role]. Give me five discovery questions to ask before I propose anything."
- "Here is an objection I received: [objection]. What might it actually mean, and what question would surface the real concern?"
- "Rewrite this proposal in the words the other person used: [their words] / [my proposal]."

Books

- *Never Split the Difference* — Chris Voss
- *SPIN Selling* — Neil Rackham. Dated in style, and still the best structured treatment of discovery questioning.

Checklist

- I open by understanding, not proposing
- I ask what they have already tried
- I pause instead of filling silence
- Every conversation ends with a named next step
- I treat objections as information

Common Mistakes

- Leading with the solution
- Treating objections as attacks
- Arguing with the stated objection rather than finding the real one
- Persuading someone who has no authority to decide
- Ending without a next step

Key Takeaways

- You already sell; the only question is whether you do it well
- V.A.L.U.E. — Validate, Ask, Listen, Underscore, Engage
- The stated objection is rarely the real one
- Ten minutes of validation changes the reception of everything after

Next Mission

The expensive lessons — hiring, contracts, and what happens when nothing is in writing.

Contracts, Hiring, and the Expensive Lessons

MISSION

To let you learn from mistakes that cost me real money.

Objectives

- Never accept anything material on a verbal agreement
- Hire against the business's actual need
- Recognize when the resistance itself is the information

Story

Two failures, both mine, both instructive.

The first employee. I hired someone on a friend's recommendation without a real interview. He sounded deeply technical, and I was impressed by that because I was technical and it was what I valued.

What the business actually needed was someone who could face customers. He focused on his own projects over the business's goals. It did not work.

That was my failure, not his. I did not vet the claims, I did not define what the role actually required, and I did not set expectations in writing. He behaved exactly as someone would when hired for skills that were not the ones the job needed and never told otherwise.

The million-dollar contract. I walked away from an opportunity worth roughly a million dollars because the other side — with a lawyer sitting right there — refused to put anything in a signed contract and wanted a verbal agreement using my own money.

That decision took about four seconds. It is also a large part of why I moved toward compliance and GRC work: I had learned firsthand what happens when nothing is written down.

Why It Matters

These are not business-owner lessons. They apply to employees constantly — every promise about a promotion, every agreed scope change, every "we'll take care of that at review time." The mechanics are identical.

Get it in writing. Every time. Regardless of how much you like the person across the table, regardless of how good the conversation was. Nothing legitimate is damaged by being written down. When someone resists documentation, the resistance itself is the information — and it is usually the most important information in the conversation.

Professional Explanation

On Hiring, and Being Hired

What the bad hire taught me, stated properly:

Hire against the actual need, not against what impresses you. I valued technical depth because I had it. The business needed customer-facing capability. Those are different, and I confused "impressive to me" with "useful to the business."

Vet the claims. A recommendation is a starting point, not a substitute for an interview. This is not distrust of the recommender; it is that they were assessing a different question.

Set expectations in writing on day one. What the role is for, what success looks like, what the priorities are when they conflict. Most performance problems are expectation problems wearing a disguise.

If you are the one being hired, invert all of this. Find out what the role is actually for. Ask what success looks like at six months. Get anything material about compensation, review timing, or scope in writing. You are protecting both of you.

On Contracts and Verbal Agreements

Things that should always be written:

- Compensation, including bonus structure and review timing
- Scope, and what happens when scope changes
- Ownership of work product
- Termination terms
- Anything promised in a conversation you would be upset to lose

The test is simple: **would I be seriously harmed if the other person remembered this differently?** If yes, write it down. Not as a legal document necessarily — an email summarizing the conversation is often enough and is far less confrontational than people fear.

Nobody reasonable objects to *"Just to confirm what we discussed: [summary]. Let me know if I've got that wrong."* If someone does object, you have learned something important very

cheaply.

Framework: The Documentation Reflex

Four situations that should automatically trigger writing something down.

1. **Any agreement about money.** Salary, budget, rates, bonuses, reimbursement.
2. **Any agreement about scope.** What is included, what is not, what happens when it changes.
3. **Any promise about the future.** A review, a promotion, a title change, a training budget.
4. **Any decision you were uncomfortable with.** Especially risk acceptance, per Chapter 4.

The reflex should be automatic and unemotional. It is not an accusation; it is professional hygiene, and the people who do it consistently are the people who do not get surprised.

Real Example

An employee is told during a hiring conversation that there will be a compensation review at six months. It is not in the offer letter.

At six months, the manager who said it has moved to another team. The new manager has no record and no obligation. The employee has nothing but a memory.

The two-minute fix that was available: an email after the conversation. *"Thanks again — just to confirm, we discussed a compensation review at six months rather than twelve. Looking forward to starting."*

That email, unanswered, is still a lot better than nothing. Answered, it is a record. It costs two minutes and it is the single most common preventable career loss I see.

Case Study: When Resistance Is the Answer

The million-dollar situation is worth examining by design rather than as a story.

A legitimate counterparty has no reason to refuse documentation. Writing things down protects both sides — it prevents honest misunderstanding as much as it prevents dishonesty. A party who wants a good-faith agreement generally *wants* it recorded.

Therefore, refusal to document is not a neutral preference. It means one of two things: either they intend to behave differently from what was said, or they want the option to. Neither is acceptable when your money or your career is the thing at stake.

The presence of a lawyer made it worse rather than better, because it removed the possibility of naivety. They knew exactly what they were asking for.

You will encounter smaller versions of this constantly. The signal is the same at every scale.

Reflection

- What have you been promised verbally that is not written anywhere?
- Have you ever avoided asking for something in writing because it felt awkward?
- If you have hired, did you hire against the need or against what impressed you?

Journal

Write down every material promise anyone has made to you about your work that exists only in memory. Then decide which of them you will confirm by email this week.

Exercises

1. Send one confirming email this week for something agreed verbally.
2. Write down what your current role is actually *for*, in one sentence, and check whether your manager would agree.
3. If you have ever hired, write down what you would do differently.

Templates: The Confirming Email

Subject: Confirming – [topic]

[Name],

Thanks for the conversation today. Just confirming what we discussed:

- [point]
- [point]

Let me know if I've misremembered anything. Otherwise I'll proceed on that basis.

Neutral, brief, and it creates a record. Send it within a day.

AI Prompts

- "Help me write a short email confirming a verbal agreement about [topic] in a way that is professional and not distrustful."

- "I am hiring for [role]. What does this role actually need to be good at, versus what would impress me technically? Ask me about the business need first."
- "Here is what I was told verbally about [promise]. What should I get in writing, and how do I ask without seeming suspicious?"

Books

- *Never Split the Difference* — Chris Voss
- *The Hard Thing About Hard Things* — Ben Horowitz. On hiring, and on the decisions that have no good option.
- *Who* — Geoff Smart and Randy Street. A structured hiring method, which is what I did not have.

Podcast Episodes

Wake Up X episodes with business owners. Listen for the expensive lessons — most founders have one story that changed how they operate permanently.

Checklist

- Everything material about my role is in writing
- I send confirming emails after significant verbal agreements
- I understand that resistance to documentation is information
- If I hire, I define the actual need before assessing candidates

Common Mistakes

- Accepting promises verbally because asking felt awkward
- Hiring against what impresses you rather than what the business needs
- Treating a recommendation as a substitute for an interview
- Not setting expectations in writing on day one
- Ignoring the signal when someone resists documentation

Key Takeaways

- Get it in writing, every time, regardless of the relationship
- Resistance to documentation is itself the information
- Hire against the actual need, not against what impresses you

- Most performance problems are expectation problems

Next Mission

Book 5: influence when you have no authority at all.

Appendix: Proposals, Narratives, and Scripts

REFERENCE

Everything from this book in one place.

The Four Questions Leadership Asks

1. Does this make money?
2. Does this save money?
3. Does this reduce risk?
4. Is this required?

Answer one, or it will not be funded.

The Value Translation

- | | |
|------------------------------------|--------------------------------|
| 1. What breaks or is missed today? | concretely |
| 2. What does that cost? | time, money, risk, opportunity |
| 3. What does the fix cost? | include the friction |
| 4. Recommendation + do nothing | always both |

The One-Page Proposal

- | | |
|-------------------|-----------------------------------|
| THE SITUATION | two sentences, in their words |
| WHAT IT'S COSTING | estimated honestly, with a range |
| THE OPTIONS | three, including do nothing |
| RECOMMENDATION | one, with why |
| WHAT I NEED | the specific decision or resource |
| TIMELINE | when, and what happens if we wait |

The Control Narrative

CONTROL:
FRAMEWORK REF:
OWNER: a named role
FREQUENCY:
HOW IT WORKS:
EVIDENCE: what proves it ran
WHERE EVIDENCE LIVES:
LAST TESTED:
KNOWN GAPS: be honest

The Risk Register Entry

RISK ID:
DESCRIPTION: as a risk sentence
LIKELIHOOD: with evidence
IMPACT:
INHERENT RATING:
CURRENT CONTROLS:
RESIDUAL RATING:
TREATMENT: mitigate / transfer / avoid / accept
IF ACCEPTED: accepted by [name, role], date, review date
IF MITIGATED: owner, target date, cost
REVIEW DATE:

V.A.L.U.E.

V Validate confirm you understand their situation
A Ask open questions about what's actually painful
L Listen the closing information is volunteered
U Underscore reflect it back in their words
E Engage a concrete next step with a name on it

The Discovery Questions

What's the most frustrating part of [process] right now?
What have you already tried?
What would "good" look like, specifically?
What's stopped this from being fixed already?
Who else does this affect?
If nothing changes, what happens?

The Confirming Email

Subject: Confirming – [topic]

Thanks for the conversation today. Just confirming what we discussed:

- [point]
- [point]

Let me know if I've misremembered anything. Otherwise I'll proceed on that basis.

The Frameworks, One Line Each

PCI DSS	payment card data
HIPAA	protected health information
GDPR	personal data of EU residents
SOC 2	service org controls, customer-driven
ISO 27001	certifiable ISMS, enterprise procurement
NIST CSF	identify, protect, detect, respond, recover
CIS Controls	prioritized and prescriptive – start here

A Closing Note

I want to be honest about why this book exists.

For a long time I thought the technical work was the work, and everything else was overhead performed by people who could not do the technical work. That belief is common in this industry and it is expensive.

It cost me a business partner's worth of skills I had to learn the hard way, one bad hire, and several good ideas that went nowhere because I presented them to people who had no

reason to care.

The people who go furthest in technology are almost never the most technical. They are the ones who stayed technical *and* learned to speak to the rest of the organization. That combination is rare because most people pick one side and dismiss the other.

You do not have to become a business person. You have to become a technical person who can be understood by one.

Learn what your company sells. Write it down. Ask before you propose.

Connect

[linkedin.com/in/efrenlsalazar](https://www.linkedin.com/in/efrenlsalazar) | wakeuypodcast.com

Questions for the author. The hiring and contract stories are yours and they carry the book. Two gaps remain: (1) a time you translated something technical to leadership and it worked, ideally with the actual numbers if they are shareable — Chapter 2 is the weakest chapter for want of one; (2) your own view on whether the GRC pivot or the business skills mattered more in your progression, since readers weighing that choice will want your actual opinion rather than a balanced summary.

IN THE FIELD

Business, On and Off Camera

From wealth-building rooms to the production side of the podcast — building the business is as much the work as running it.



Real Estate Wealth Expo, networking as a VIP guest



Behind the camera producing podcast content

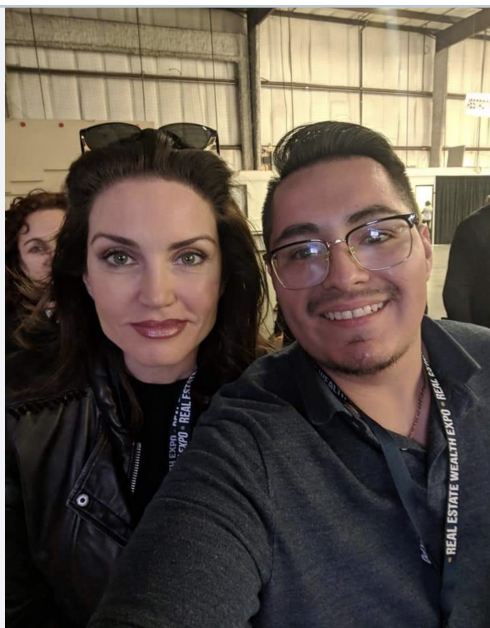
IN THE FIELD

The Team Behind the Brand

Business First is about more than one founder — it's the team and the network around the work.



"What's Your WHY" — team event



Real Estate Wealth Expo, networking as a VIP guest

GUEST SPOTLIGHT

Gustavo Mastroianni — Security-First in the Age of AI

Gustavo Mastroianni is CISO at Schools Insurance Authority, working alongside Mo Balakrishnan's CTO role. His clearest value to this library is refusing to hype or dismiss AI — he just says plainly what it's doing to the field, good and bad.

You Don't Need to Be a Data Scientist

His advice to anyone worried about AI requirements creeping into job postings: you don't need deep AI expertise, you need fluency. Start using AI tools for the searches you'd normally run through Google. "If you know how to use it... you're gonna be ahead of the game." He got his own AI certification (Certified Association AI Professional) in 2025 specifically to formalize that fluency.

Certs Prove Something Specific — Not Everything

His honest framing of what a certification actually proves: not that you know everything, but that "I was able to understand the topics... within a constrained timeframe." He holds CISM, CCSO, and is targeting ISACA's new AI-focused certification next — treating certs as one input among several, not a career in themselves.

AI Will Remove Some Jobs — Mostly Entry-Level

Asked directly whether AI will cut jobs, he doesn't soften it: yes, some jobs will disappear, mostly at the entry level. His point isn't to be alarmist — it's that new roles get created as old ones fade, the same way cybersecurity itself didn't exist as a job category not long ago.

Watch the full episode: youtu.be/RxQ1V7VWdnw

Gustavo's Advice, In Practice

Fact-check anything AI tells you before you rely on it.

Hallucinations happen, and taking one at face value can damage your reputation or your organization's. Verify what you don't already know.

Start small with automation — control the blast radius.

Before automating anything with AI, make sure the inputs are reliable and the decision logic is clearly defined. Test at low-stakes scale before scaling up.

Never rely on a single AI provider.

Building in a way that lets you switch between models or providers protects your organization from price hikes or outages — sovereignty over your own systems matters.

Be ready for your career path to change every 7–10 years.

He didn't plan to leave network engineering — his path changed anyway. Building education and certifications isn't about the short term, it's about staying adaptable long-term.

REFLECTION

Are you using AI tools regularly enough to be fluent, or are you still avoiding them?
What's one AI-heavy task you could try this week just to build that muscle?

GUEST SPOTLIGHT

Anil Reddy — Small Curiosity, Real Consultancy

Anil Reddy is founder of ARK Strategist, a consulting firm serving public-sector agencies across multiple states. After 25+ years in corporate life, he left during the pandemic to start over in a field he'd never formally worked in — and built credibility for it from scratch.

A Perfect Score, With Zero Prior Experience

His first consulting engagement was writing security policies — work he'd never done before. He self-studied and earned a perfect score on the CISM to back the work up. His own framing for how it all started isn't a business plan, it's "small curiosity": wanting to see what else was possible outside an 8-to-5 corporate job.

Chose Consulting to Avoid Getting Boxed In

Rather than specialize narrowly, his model is deliberately broad: find the problem statement, figure out how to create value, repeat across whatever the client actually needs — security policy, vulnerability assessment, penetration testing, cloud security. Four years in, ARK Strategist has expanded into Nevada, Oregon, Washington, and partnership work in New York.

Bring Solutions, Not Just Problems

His single clearest line for anyone starting a business: don't just show up with what's broken, show up with what you'd do about it. Paired with a direct point on integrity — network honestly, and the integrity itself does the selling over time.

Watch the full episode: youtu.be/IOI5IK_UezA

Anil's Advice, In Practice

Community credibility comes before business credibility.

He's explicit that ISACA Sacramento and specific state leaders willing to listen to a new idea are what let him turn "I have a value proposition" into real traction.

Grow vertically before horizontally.

Rather than chase new states or new ventures immediately, his approach was deepening the base he'd already built in public-sector work first — expansion is sequenced, not rushed.

A 2 a.m. breach is a systems problem, not a willpower problem.

His answer to handling a security incident at 2am: take a holistic approach, don't try to know everything yourself, surround yourself with people who cover what you don't, then physically step away — walk, exercise, get outside.

Study first — credibility is what lets the conversation start.

For someone new to the field: get certified first, since that's literally what gives you standing to engage with leaders in the room. Experience compounds after that credibility exists, not before.

REFLECTION

Are you showing up to conversations with problems, or with a proposed next step? What's one place you could start bringing solutions instead?

BE OUR NEXT GUEST

Your Story Belongs on Wake Up X

If Business First gave you something worth using, you already have something worth sharing. The Wake Up X podcast exists to put real, unpolished career and business stories in front of people who need to hear them — and every guest so far started exactly where you are now: reading, building, and figuring it out in public.

You might be a fit if you:

Broke into tech, cybersecurity, or business without the traditional path

Built something — a lab, a business, a portfolio, a community — and can talk about how

Have a failure or turning point you're willing to tell straight, not polished

Are mentoring, leading, or building something worth other people knowing about

HOW TO APPLY

Share your story at wakeupxpodcast.com/guest, or connect directly on LinkedIn at linkedin.com/in/efrenlsalazar and mention this book.

Being a guest is also visibility — the same principle Book 7 makes the case for.

Acknowledgments

THE PEOPLE WHO MADE THIS POSSIBLE

None of Wake Up X happened alone, including Business First. Every book in this library, the podcast, and the free library exist because specific people showed up for a kid nobody expected much from. This is the record of who they are.

THE EARLY YEARS

Miss England, the 6th-grade teacher who stayed after class, unpaid, to teach the fundamentals. Sister Cora and Lewis Bair of Computers for Kids, who handed a teenager a key and a broken machine and let the work speak. Jose Luis Bravo, Angel Martinez, Isaac Murrilo, and Alex De Haro — the friends and cousins who translated the world, gave hope, drove him where he needed to go, and carried him before there was anything to carry him toward.

THE BUSINESS YEARS

Rojelio Salazar, his cousin and first mentor, RNS Computer Repairs — the reason it is called Wake Up X. Kyle Erni, his first business partner. Erick and Pancho Arauza, co-founders of The Cellphone Fix. Danny Maya, of Maya Traditional Mexican Restaurant, Taqueria Maya, and Salt and Water, who pushed him in front of a camera for the first time.

THE CAREER YEARS

Igor, who gave him the vocabulary for work he already knew how to do. Nate, who he mentored into a full-time role alongside his own. Ken Kerrick, opponent turned friend in the ISACA Sacramento election. Alvin Jue and David De los Reyes, who believed in the middle of it, not just at the finish.

THE PODCAST

Every guest who trusted an unproven show with their story, including Gustavo Mastroianni, Cathy Stephenson, Greg McCord, Aaron Portnoy, Sana Talwar, Jorge Avila, Anil Reddy, Devin Prasad, Larry Switzer, Louis Kemp, Mohana Balakrishnan, Eddie Lopez, and Michael Fierros — full episodes at wakeupxpodcast.com.

COMMUNITY PARTNERS

ISACA Sacramento and ISC2 Sacramento, The Cellphone Fix, A.N.U.L Technologies, AISOC, AspireByte, and PodGrabber — the organizations building alongside this movement.

AND

His parents, who arrived with nothing, worked since childhood, and gave away whatever was left over. Nothing about the first version of this story predicted the version you are holding now.

Continue the Journey

EVERY BOOK LEADS SOMEWHERE ELSE

Business First is one part of a 12-book system. If it helped, the next step is one page away — take whichever layer matches where you are standing.

THE LIBRARY

Layer 1 — The Story

I Just Have To Wake Up — why any of this exists.

Layer 2 — The Mindset

Comfort Is Expensive — the real cost of playing it safe.

Layer 3 — The Blueprint

DECODE: The Wake Up X Playbook — zero experience to executive thinking. Book 0.

Layer 4 — The Playbooks

1 Break In · 2 Build It · 3 Think Like a Cybersecurity Professional · 4 Business First · 5 Lead Without a Title · 6 AI for IT & Cybersecurity · 7 Build Your Brand · 8 The Consultant Mindset · 9 Level Up

FIND US EVERYWHERE

Podcast & free library: wakeupxpodcast.com · Spotify: open.spotify.com/show/28GZnRBrIV2efHOHzdiftt · Apple Podcasts: podcasts.apple.com/us/podcast/id1792821643 · YouTube: youtube.com/@Efren.wakeupx · LinkedIn: linkedin.com/in/efrenIsalazar · Substack: substack.com/@wakeupxefren

IN GOOD COMPANY

ISACA Sacramento, ISC2 Sacramento, The Cellphone Fix, A.N.U.L Technologies, AISOC, AspireByte, and PodGrabber — the partners and collaborators building alongside the movement. See the full list at wakeupxpodcast.com/community.html.

Get DECODE free, apply to be a guest, or read the story behind the movement — all of it starts at wakeupxpodcast.com.

WAKE UP X

SOLUTIONS



Efren L. Salazar

FOUNDER · HOST · CYBERSECURITY PROFESSIONAL · EDUCATOR · COMMUNITY BUILDER

STAY CONNECTED

Podcast & Blueprint Library:	wakeupxpodcast.com
LinkedIn:	linkedin.com/in/efrenlsalazar
YouTube:	youtube.com/@Efren.wakeupx
Be a Guest:	wakeupxpodcast.com/guest

THE WAKE UP X PUBLISHING SYSTEM

BOOK 5

Lead Without a Title

Influence, feedback, and mentoring before anyone promotes you

Authority is given. Influence is earned.

THE
WAKEUPX
SOLUTIONS

EFREN L. SALAZAR

Founder & Host, Wake Up X | IT Operations Manager & Solution Engineer
Government & Regulatory Partner, ISACA Sacramento | CompTIA Security+

wakeupxpodcast.com

Educational content — not professional, legal, financial, or career-placement advice.
Full disclaimer in DECODE, the foundation of this library.

Before You Begin

HOW TO USE THIS BOOK

This book is about leading before anyone appoints you — which is the only order it has ever actually happened in.

Where this book sits

Two books come first — the memoir I Just Have To Wake Up and the mindset book Comfort Is Expensive — before this ten-book library. Each one is short on purpose. Pick up the one that matches the problem in front of you right now.

- n **Book 0 — DECODE.** The foundation. The map of the whole path.
- n **Book 1 — Break In.** Landing your first role in IT or cybersecurity.
- n **Book 2 — Build It.** Home labs and a portfolio that prove you can do the work.
- n **Book 3 — Think Like a Cybersecurity Professional.** Detection, triage, and incident response thinking.
- n **Book 4 — Business First.** Finance, sales, GRC, and why the business hired you.
- n **Book 5 — Lead Without a Title.** Influence, EQ, and mentoring before anyone promotes you.
you are here
- n **Book 6 — AI for IT & Cybersecurity.** Using AI as an actual working tool, not a novelty.
- n **Book 7 — Build Your Brand.** LinkedIn, content, and being findable.
- n **Book 8 — The Consultant Mindset.** Discovery, solution selling, and stakeholder management.
- n **Book 9 — Level Up.** Long-term growth, burnout, legacy, and mentoring others.

Book 0 is the foundation. *DECODE: The Wake Up X Playbook* holds the six-level path, the Curiosity Compass, and the core frameworks. Nothing in any other book contradicts it. If a term here is unfamiliar, DECODE defines it.

What is in this book

Part One — Influence Before Authority

1. Authority Is Given. Influence Is Earned.
2. Making Your Work Visible Without Being Insufferable

Part Two — The Human Skills

3. Disagreeing Well, and Committing After
4. Feedback, Both Directions
5. Managing Upward

Part Three — Carrying It Forward

6. Mentoring, and the Person Who Did It For You
7. Appendix: Scripts, Structures, and the Success File

How to actually use it

This is the least technical book in the library after Book 4. It is the one people most often skip. That skipping is exactly why the material works — you are competing against people who consider this soft.

Nothing here needs permission. Every practice in this book can be started tomorrow, in your current role, without telling anyone.

One caution before you start. Everything in Chapter 2 about visibility can be done cynically, and people can tell instantly when it is. Help because it is the job. Then make the help visible. That order is not a technicality — it is the difference between being trusted and being avoided.

I am writing this the way I would say it to you sitting across a table — not lecturing a room. Take what works, remix the rest into your own version, and leave what does not fit.

PART

ONE

Influence Before Authority

Authority Is Given. Influence Is Earned.

MISSION

To make you the person people listen to before anyone gives you the title.

Objectives

- Understand why influence precedes promotion rather than following it
- Build credibility deliberately rather than hoping it accumulates
- Stop waiting to be chosen

Story

At a large multi-site employer, I noticed something that nobody had put on anyone's task list: our offices had effectively stopped talking to each other.

Each location was solving the same problems independently, and none of them knew the others had already solved them. Someone in one city would spend two days on a problem that had been fixed in another city six months earlier.

Nobody assigned me to fix that. There was no project, no budget, and no request. I built a cross-site communication hub so the sites could actually share what they knew.

At a different site — one with essentially no structure at all — I introduced regular meetings and real ticketing discipline where there had been none. Again, nobody asked.

Neither of those was in a job description. Both are the reason I was trusted with more, and eventually the reason I was managing operations rather than working tickets.

I want to be precise about the mechanism, because "do more than your job" is advice people give without explaining why it works. It works because **the work you do outside your title is the only evidence anyone has about what your next title should be**. Inside your title, you are showing that you can do the job you already have.

Why It Matters

Almost everyone has the sequence backwards. They believe: get promoted, then lead.

The actual sequence is: lead, get noticed, then get promoted. Organizations promote people who are already doing the job, because promoting anyone else is a risk nobody wants to take. A manager who promotes an unproven person is spending their own credibility on a guess.

Which means the waiting is the mistake. You are waiting for permission to start something you can start today, and the starting is the qualification.

Professional Explanation

The Sources of Influence Without Authority



None of these require a title, a budget, or anyone's approval.

Five sources of influence. None require a title, a budget, or anyone's approval.

Competence. You are reliably good at something people need. Necessary but not enough — plenty of highly competent people have no influence at all, and if you have worked anywhere you have met one.

Reliability. You do what you said, when you said. This is rarer than competence and worth more. Being predictable is a genuine superpower in an environment where most people are not. It also compounds: every kept commitment makes the next one more believable.

Generosity. You help people who cannot do anything for you. This builds quietly for years and then pays all at once, usually at a moment you could not have predicted.

Clarity. You can explain complicated things simply. People gravitate toward whoever makes them feel less confused. This is the source of influence most available to technical people who have not yet developed the others.

Composure. You stay steady when things break. Everyone remembers who was calm on the bad day. Book 3's chapter on incident communication is this source applied under pressure.

Notice none of these need a title, a budget, or anyone's approval. That is the entire point of the chapter.

This is also what the Leader current from Book 2 looks like day to day. It is not a job somebody hands you. It is these five things, done for long enough that people start treating you as one.

Why Technical Excellence Alone Plateaus

The most technically capable person on a team is frequently not the most influential, and this confuses people who believe merit is straightforward.

The explanation is that technical excellence answers "can you do the work?" Influence answers a different question: "do people want to work with you, do they trust your judgment, and do they seek you out?" Those are separate assessments, and organizations weight the second more heavily than they admit.

This is not unfairness. A brilliant engineer who is difficult to work with genuinely does produce less organizational value than a good engineer who lifts the people around them. The math is real.

Framework: What Leading Without a Title Actually Looks Like

Concretely, in a normal week:

- You notice something nobody owns, and you own it
- You answer a question thoroughly instead of pointing at a link
- You write the thing down so the next person does not have to ask
- You say the uncomfortable true thing in a meeting, kindly
- You do what you said you would do, on the day you said
- You give credit specifically and publicly
- You stay calm when something breaks

None of that is heroic and none of it needs authority. Done for six months, it changes how you are regarded. Done for two years, it changes what you are offered.

Be more valuable than your title. The people who move fastest are almost always doing something their job description does not mention. That is not extra work for free — it is how the next role gets justified.

Real Example

Two people join the same team at the same level.

The first does his assigned work well and waits to be recognized. He is competent, and when he is passed over for a lead role he is genuinely confused, because he was the stronger engineer.

The second does her assigned work well, and also: writes up the deployment process nobody had documented, becomes the person people ask about the legacy system because she took the time to understand it, and consistently answers questions in a way that leaves people better off rather than merely correct.

When the lead role opens, she is already doing most of it. Promoting her is not a risk; it is a formality.

The first person's mistake was not laziness. It was believing that the assessment was about technical quality when it was about showed scope.

Case Study: The Person Everyone Asks

In most teams there is one person who is not the most senior and not the manager, but who everyone asks. Their opinion carries weight in meetings out of proportion to their title. When they say a plan will not work, the plan usually changes.

If you look closely at how they got there, it is almost never one dramatic thing. It is that they answered questions properly for two years, they were right often enough and admitted it quickly when they were not, and they never made anyone feel stupid for asking.

That position is available to anyone willing to spend the time. It is not a personality trait. It is not charisma. It is accumulated small behaviour.

Reflection

- What are you waiting for permission to start?
- Which of the five sources of influence is your strongest, and which have you never worked on?
- Is there something nobody owns that you could own?

Journal

Write about a person whose opinion you trust at work, or anywhere. Why do you trust it? Almost everyone finds the answer is reliability or clarity rather than raw capability.

Exercises

1. Identify one thing at your workplace that nobody owns and that would help people if it existed. Do not ask permission. Just start it.
2. For one week, notice every commitment you make and track whether you meet it. Most people are surprised.
3. Rate yourself honestly on the five sources. Pick the weakest and work on it for a month.

AI Prompts

- "I am an individual contributor with no direct reports. Give me ten specific ways to build influence on my team that do not need any authority or budget."
- "Here is something I want to start doing at work that nobody asked for: [description]. How do I begin it without appearing to overstep?"
- "I am strong technically but feel overlooked. Ask me diagnostic questions to help me work out whether this is a visibility problem, a relationship problem, or something else."

Books

- *Extreme Ownership* — Jocko Willink and Leif Babin. The core argument of this entire book, stated more forcefully.
- *The Score Takes Care of Itself* — Bill Walsh. On building a standard of behaviour rather than chasing outcomes.
- *How to Win Friends & Influence People* — Dale Carnegie. Old, still correct.

Podcast Episodes

Wake Up X episodes with people who moved from technical roles into leadership. Listen for when the shift actually happened — it is nearly always before the title, not after.

Resources

- Your local **ISACA** or **ISSA** chapter, where volunteering for anything is a fast route to leading something real
- Any internal committee, working group, or initiative nobody wants to run

Checklist

- I know which of the five sources of influence I am weakest on
- I have started at least one thing nobody assigned me
- I track whether I meet the commitments I make

- I have stopped waiting to be chosen

Common Mistakes

- Assuming good work speaks for itself. It does not; it whispers.
- Waiting for the title before behaving like a leader
- Taking credit for team work, which costs more than the credit is worth
- Confusing being the best engineer with being the most valuable person

Key Takeaways

- You lead first, get promoted second — organizations promote the proven
- Reliability and clarity carry more influence than raw competence
- Work outside your title is the only evidence about your next title
- The person everyone asks got there through accumulated small behaviour

Next Mission

The uncomfortable part: making sure the work is actually seen.

Questions for you. What is the thing nobody owns at your workplace? And what specifically is stopping you from owning it?

Making Your Work Visible Without Being Insufferable

MISSION

To close the gap between doing good work and anyone knowing about it.

Objectives

- Build a success file that makes reviews and interviews trivial
- Route genuine feedback upward without it feeling like self-promotion
- Give credit in a way that increases rather than spends your own standing

Why It Matters

Good work that nobody knows about does not compound. This is not politics; it is arithmetic.

Most people find this chapter uncomfortable, and I understand why. There is a real fear of becoming the person who narrates their own contributions constantly, and that person is genuinely irritating.

But the alternative most people choose — doing excellent work silently and hoping — is not humility. It is a strategy. It is a poor one.

The resolution is that visibility done well is mostly *documentation*, not announcement.

Framework: The Two Habits

1. Ask for feedback, then route it upward

When you have genuinely helped someone, ask them how it went.

If the answer is good, ask whether they would be willing to share that with your supervisor. Not "please tell my boss I did well" — more like: "I'm glad it worked."

If you get the chance, it'd help me if you mentioned it to [manager]. I'm trying to get better at showing the impact of this kind of work."

You are not fishing for praise. You are making real work visible to the people who make decisions about you, and you are being transparent about why. Most good work dies in the gap between doing it and anyone knowing, and nobody fills that gap on your behalf.

The reason this works and does not read as self-promotion: the praise comes from a third party. Third-party praise is both more credible and less awkward than your own account.

2. Document every success

Approvals, thank-you emails, completed projects, incidents resolved, things you fixed that nobody asked you to fix, problems you prevented.

Keep a running file. Add to it **weekly**, not annually.

At review time you will have evidence while everyone else has recollection. At your next interview you will have specifics while everyone else has adjectives. And in a difficult conversation about headcount, you will have a documented record of contribution rather than a feeling.

Most people rebuild this from memory the week reviews are due, and it always shows — the recent things are vivid and the first eight months are gone.

The trap. Doing any of this transactionally — helping people specifically so you can collect the credit — is visible and it poisons everything. Help because it is the job. Then make the help visible. The order matters, people can tell, and the ones who get it wrong develop a reputation that takes years to shed.

Step-by-Step Guide: Building the Success File

1. **Create one file.** Anywhere. A document, a note, a spreadsheet.
2. **Backfill three months** from your email and ticket history. Most people find a lot more than they remembered, which is itself the lesson.
3. **Add one entry per week**, minimum. Friday afternoon works well.
4. **Capture the evidence, not just the claim.** Screenshot the thank-you email. Note the ticket number. Record the actual numbers.
5. **Write it as impact, not activity.** Not "worked on the migration" but "migrated 40 users with no reported downtime; wrote the runbook the team now uses."
6. **Review it quarterly.** You will spot patterns — the kind of work you keep being pulled into, which tells you something about where you are actually valuable.

Templates: The Success File Entry

```
DATE:
WHAT I DID:      one line
WHY IT MATTERED: the impact, concretely
EVIDENCE:        email, ticket, screenshot, who saw it
NUMBERS:         if any, honestly
WHO BENEFITED:
RESUME BULLET:   the one-line version, written now
```

The last line is the highest-value field in the whole file. Writing the bullet at the moment it happened is easy. Reconstructing it eighteen months later is nearly impossible, and Book 1's bullet bank depends on it.

Professional Explanation: Giving Credit

Counterintuitively, giving credit generously increases your own standing rather than diluting it.

The mechanism: specific public credit signals that you notice good work, that you are secure enough not to need all of it, and that working with you is safe. All three of those make people want you on their projects, and being wanted on projects is how scope grows.

How to do it well:

- **Be specific.** "Priya's rewrite of the deployment script is why this went smoothly" beats "great work team."
- **Do it in front of people who matter,** not privately afterward.
- **Never take team credit as individual credit.** People find out, and the damage is permanent.
- **Credit the unglamorous work,** which nobody else credits and which is therefore worth the most to the person receiving it.

The people who hoard credit are always visible, and they are always trusted less than they think they are.

Real Example

Two engineers do comparable work over a year.

At review time, the first says he had a good year and worked hard on the migration. His manager, who has fifteen direct reports and a poor memory of March, agrees vaguely.

The second brings a one-page summary: seven projects with outcomes, three thank-you emails from other departments, the runbook she wrote that reduced a recurring ticket type,

and the specific numbers where she had them.

Same work. Completely different conversations, and completely different outcomes — not because one was better, but because one was *legible*.

Case Study: The Quiet Person Who Got Cut

An uncomfortable pattern. The reason this chapter is not optional.

During a reduction, decisions are made quickly by people who do not observe daily work. They look at what they know, what they can point to, and what other leaders say about a person.

The technically strong, quiet contributor who never made their work visible is genuinely at risk here. Not because anyone doubts their quality, but because nobody outside their immediate team has any specific evidence of it. "They seem good" loses to "she rebuilt our onboarding process and cut ticket volume."

I am not telling you this to frighten you. I am telling you because the fix costs about ten minutes a week and most people do not do it.

Reflection

- If you had to justify your value in writing tomorrow, what evidence do you have?
- Does your manager know what you actually did last quarter?
- Have you ever helped someone and then let it disappear?

Journal

List everything you accomplished in the last three months. Then check your email and see how much you forgot. That gap is the argument for the success file, made by your own memory.

Exercises

1. Create the success file today and backfill three months.
2. The next time you genuinely help someone, ask how it went — and if the answer is good, ask them to mention it.
3. Give one piece of specific public credit this week.

Project

Build a one-page "what I did this year" summary from your success file, formatted as if for a review. Even if your review is months away, having it changes how you work — you start noticing what is worth capturing.

AI Prompts

- "Here is a list of things I did this quarter. Rewrite each as impact rather than activity, without inventing numbers I did not give you."
- "Help me phrase a request asking a colleague to mention our work to my manager, in a way that is honest and not awkward."
- "Here is my success file for the year. Draft a one-page review summary organized by impact."

Books

- *Atomic Habits* — James Clear. The weekly logging habit is exactly this book's argument applied to a career.
- *The Score Takes Care of Itself* — Bill Walsh

Checklist

- My success file exists and I add to it weekly
- I write impact, not activity
- I capture evidence at the time, not from memory later
- I have asked for feedback and routed it upward at least once
- I give specific public credit regularly

Common Mistakes

- Doing this transactionally, which everyone detects
- Announcing rather than documenting
- Rebuilding the year from memory the week before reviews
- Taking team credit as individual credit
- Believing silence is humility when it is really a strategy

Key Takeaways

- Invisible work does not compound
- Third-party praise is more credible and less awkward than your own

- Write the resume bullet at the moment it happens
- Giving specific credit increases your standing rather than spending it

Next Mission

The skill that determines whether people want you in the room.

PART

TWO

The Human Skills

Disagreeing Well, and Committing After

MISSION

To let you disagree without damaging relationships, and to move on properly when the decision goes against you.

Objectives

- Frame disagreement as collaboration on a shared goal
- Say your piece once, clearly
- Commit fully to decisions you argued against

Why It Matters

Technical people disagree constantly, often correctly, and frequently in ways that cost them.

The cost is rarely the disagreement itself. It is the framing — being right in a way that makes the other person defend rather than think — and the aftermath, where people relitigate a decided matter until they are quietly excluded from the room where decisions get made.

Framework: Agree on the Goal Before You Disagree on the Method

The technique that works nearly universally.

"We both want the migration to not break anything on Monday. I'm worried the current timeline doesn't leave room to test the rollback. Can we talk about that piece?"

Three things happen in that sentence. You have stated the shared goal, which makes you a collaborator rather than an obstacle. You have named your specific concern rather than a general objection. And you have asked to discuss one piece rather than rejecting the whole plan.

Compare: *"This timeline is unrealistic and we're going to break production."* Possibly true. It leaves the other person with nothing to do except defend, and it makes the conversation about whether they planned badly rather than about the rollback window.

The content is identical. The reception is not.

The Structure

1. **State the shared goal**, in their terms

2. **Name your specific concern**, narrowly
3. **Ask a question** rather than delivering a verdict
4. **Offer what would resolve it** — "if we had one extra day for testing, I'd be comfortable"

That fourth step is the one people skip. An objection with no resolution condition is just resistance. It is exhausting to receive.

Professional Explanation: Disagree, Then Commit

When the decision goes against you: say your piece clearly, once, then support the decision fully.

This is harder than it sounds and it matters more than almost anything else in this book.

People who keep relitigating decided matters get excluded from rooms where decisions are made. Not through malice — it is simply that including them makes every meeting longer and every decision less final. The exclusion is usually never explained. That is why people who do this often have no idea why their influence is shrinking.

Committing does not mean pretending you agreed. It means:

- You do not undermine it in side conversations
- You do not say "I told you so" if it goes badly, ever
- You execute properly rather than half-heartedly
- If new information emerges, you raise it as new information rather than as a reopened argument

The "I told you so" is the single most expensive sentence available to you. It buys thirty seconds of vindication and costs you the willingness of everyone present to hear your objections next time — because now objecting has become a bid to be proven right later rather than a contribution.

When to Escalate Instead

Disagree-and-commit has a limit. If the decision is unsafe, unethical, or illegal, committing is not the right response — escalating is.

The distinction is not "I think this is a bad idea." It is "this creates a risk the decision-maker may not be aware of. I have a duty to make sure they are." State the risk factually, in writing, once, to the right person. Book 4's chapter on written risk acceptance covers what to do next.

Most disagreements are not this. Be honest with yourself about which category you are in, because treating a preference as a principle is how people burn credibility they will need later.

Real Example

An engineer believes the team should adopt a different monitoring tool. The manager decides to stay with the current one for budget reasons.

The costly version: the engineer continues raising it in meetings, mentions it whenever monitoring comes up, and when there is an outage six months later, says the other tool would have caught it. He is technically correct. He is also now the person everyone braces for.

The effective version: he makes the case once, properly, with the Book 4 one-page format. The decision goes the other way. He commits, and he makes the current tool work as well as it possibly can — which produces genuine expertise in it. A year later, when budget opens, the manager comes to him and asks what they should do. He is asked because he showed he could be overruled without becoming difficult.

The second version got what he wanted, eleven months later, with more credibility than when he started.

Case Study: The Meeting Where Nobody Objected

The inverse failure, worth naming because it is also common.

A team develops a culture where disagreement is uncomfortable. Objections stop being raised. Decisions get made in rooms where everyone nods and then complains privately afterward.

This is worse than a team that argues. Unspoken objections do not disappear; they resurface as slow execution, quiet non-compliance, and a project that stalls for reasons nobody can name.

If you are ever in a position to shape this, the fix is to ask directly and by name: *"Priya, you've been quiet. What's the part of this you're least comfortable with?"* Asking someone specifically, with a question that assumes they have a concern, gets far more than "any objections?" ever will.

Templates: The Disagreement Script

"We both want [shared goal].

My concern is specifically [narrow issue] –
[the reason, briefly].

Can we talk about that piece?

If [resolution condition], I'd be comfortable."

And after the decision:

"Understood – I'll make it work. If [specific thing]
comes up, I'll flag it early rather than relitigating."

That second script is worth memorizing. It signals commitment while leaving a legitimate door open for genuinely new information.

Reflection

- When you disagree, do you make people think or make them defend?
- Have you ever said "I told you so"? What did it cost?
- Is there a decided matter you are still quietly relitigating?

Journal

Write about a disagreement you lost. Then write what you did afterward. Most people find the aftermath cost them more than the disagreement did.

Exercises

1. Take a current disagreement and rewrite your position using the four-step structure.
2. Practise the commit script out loud. It is harder to say sincerely than it looks.
3. Identify one decided thing you have not fully committed to, and commit to it.

AI Prompts

- "Here is my objection to a plan: [objection]. Rewrite it as: shared goal, narrow specific concern, a question, and what would resolve it."
- "I disagreed with a decision and was overruled. Help me write a short message that commits properly without pretending I agreed."

- "Is this a preference or a genuine risk that should be escalated? Ask me questions to help me tell the difference: [situation]."

Books

- *Crucial Conversations* — Patterson, Grenny, McMillan, Switzler. The standard text and genuinely useful.
- *Extreme Ownership* — Willink and Babin. The chapter on disagreeing with superiors is directly applicable.
- *Thanks for the Feedback* — Stone and Heen

Checklist

- I state the shared goal before I state my concern
- I name what would resolve my objection
- I say it once, properly, and then commit
- I have never said "I told you so"
- I know the difference between a preference and a risk requiring escalation

Common Mistakes

- Rejecting the whole plan instead of naming one concern
- Objecting with no resolution condition
- Relitigating decided matters
- "I told you so"
- Treating every preference as a principle

Key Takeaways

- Agree on the goal before disagreeing on the method
- An objection without a resolution condition is just resistance
- Disagree, then commit fully — including when it goes badly
- Unspoken objections resurface as slow execution

Next Mission

The two-way skill nobody is taught properly.

Feedback, Both Directions

MISSION

To make you good at the two hardest conversations in professional life.

Objectives

- Give feedback that changes behaviour rather than creating defensiveness
- Receive feedback without explaining
- Ask for feedback in a way that gets you the truth

Professional Explanation

Giving Feedback

Four requirements, all necessary:

Specific. "The report went out with last quarter's numbers" — not "you're careless." One is fixable. The other is a judgment about who they are that produces only defensiveness.

Timely. Within days, not at the annual review. Feedback about something six months old is useless and it feels like an ambush.

Private. Criticism in private, praise in public. This is old advice because it is correct.

Behaviour, not person. Separate what was done from who they are. The behaviour can change; the person cannot, and telling someone who they are invites them to defend rather than adjust.

There is a fifth, which people miss: **make clear what you want to happen next.** Feedback that identifies a problem without a direction leaves the person anxious and no better informed.

Receiving Feedback

The only correct first response is: **"Thank you. Let me think about that."**

Not explanation. Not context. Not defense — even when you have a good one, and you often will. The explanation can come later if it is still needed, and most of the time it will not be.

The reason this matters: the person giving you feedback is taking a risk, and how you respond determines whether they ever do it again. Someone who explains immediately is remembered as someone who does not take feedback, regardless of what they do with it afterward.

If the feedback is wrong, you can address it in a day or two: *"I've been thinking about what you said. I want to add some context — not to argue, just so you have the full picture."* That works. The same content delivered in the first ten seconds does not.

The reflex is powerful and it feels like honesty. When someone criticizes your work, the urge to immediately supply the missing context is nearly overwhelming, and it feels like you are correcting the record rather than being defensive. From the outside, those look identical. Wait a day.

Asking For Feedback

Most requests for feedback fail because they are unanswerable. "Any feedback for me?" gets "you're doing great," which is worthless.

Ask narrowly:

- "What is one thing I could have done better in that meeting?"
- "If you had to name the thing that would most improve my work, what would it be?"
- "What do you think I am worse at than I realize?"

That last one is uncomfortable and it produces the most useful answers. It gives explicit permission and it signals you want the real thing.

Then — and this is the part that determines whether you ever get honest feedback again — **thank them and visibly act on it.** People give feedback to those who use it.

Framework: The Feedback Conversation

- | | |
|------------------------|--------------------------------------|
| 1. Ask permission | "Can I give you some feedback on X?" |
| 2. State the behaviour | specific, observable, recent |
| 3. State the impact | what it caused, factually |
| 4. Ask for their view | they may know something you don't |
| 5. Agree what changes | concrete and mutual |
| 6. Offer support | "what would help?" |

Step 4 is the one that separates useful feedback from a lecture. A meaningful proportion of the time, the person has information that changes your assessment entirely. The report went out with old numbers because the data source was down and they escalated and got no response.

Real Example

A colleague repeatedly joins meetings late, disrupting the start.

Weak: "You're always late, it's disrespectful." — identity claim, no specifics, invites defense.

Effective: "Can I mention something? You've come in ten minutes late to the last three standups, and it means we restart the update each time. Is something clashing on your calendar?"

That version is specific, states the actual impact, and ends with a question that assumes a reason rather than a character flaw. It frequently turns out there *is* a reason — a back-to-back meeting nobody noticed — and the fix is a calendar change rather than a behaviour change.

Case Study: The Feedback That Never Came

A pattern with real consequences.

Someone is passed over for a promotion, or let go, and is genuinely blindsided. When they ask why, the reasons turn out to have existed for two years and nobody ever told them.

Everyone involved behaved understandably. The manager avoided an uncomfortable conversation. Colleagues did not think it was their place. The person themselves never asked in a way that made honesty easy.

The single most protective habit against this: ask one person, once a quarter, "what do you think I am worse at than I realize?" It is an unpleasant question to ask and it is much less unpleasant than finding out in an exit conversation.

Reflection

- When you receive criticism, how long before you start explaining?
- Who would tell you the truth about your work? Have you asked them?
- Is there feedback you have been avoiding giving someone?

Journal

Write down the last piece of critical feedback you received and what you actually did with it. Then write what you wish you had said in the moment.

Exercises

1. Ask one person this week: "what do you think I am worse at than I realize?"
2. Practise saying "thank you, let me think about that" until it is automatic.
3. Give one piece of feedback you have been avoiding, using the six-step structure.

AI Prompts

- "Here is feedback I need to give: [situation]. Rewrite it as specific behaviour and impact rather than a judgment about the person."
- "I received this feedback: [feedback]. Help me identify what is true in it before I respond, and what I would want to add later."
- "Give me five narrow questions I could ask a colleague to get genuinely useful feedback rather than reassurance."

Books

- *Thanks for the Feedback* — Douglas Stone and Sheila Heen. The best book on the receiving side, which nobody else covers.
- *Radical Candor* — Kim Scott
- *Crucial Conversations* — Patterson et al.

Checklist

- I give feedback about behaviour, not identity
- I say "thank you, let me think about that" before anything else
- I ask narrow questions rather than "any feedback?"
- I visibly act on feedback I receive
- I have asked someone what I am worse at than I realize

Common Mistakes

- Feedback as identity claim
- Explaining immediately when receiving
- Saving it all for the annual review
- Asking unanswerably broad questions
- Skipping step 4, and missing the context that would have changed your view

Key Takeaways

- Specific, timely, private, behaviour-focused, with a clear next step
- "Thank you, let me think about that" is the entire receiving skill
- Ask what you are worse at than you realize
- People give feedback to those who use it

Next Mission

The relationship that most determines your daily experience of work.

Managing Upward

MISSION

To make your manager's job easier, because they will then make yours better.

Objectives

- Bring problems with options attached
- Never let your manager be surprised
- Ask for what you need explicitly

Why It Matters

Your manager is a person with too much to do and incomplete information. Most of what frustrates you about them is a consequence of one or both of those conditions rather than a character trait.

Managing upward is not manipulation. It is recognizing that the relationship has two participants and that only one of them — you — is thinking about it strategically.

Professional Explanation

The Four Practices

1. Bring problems with a proposed solution attached, even a bad one.

"X is broken" makes you a source of work. "X is broken, I think we should do Y, but Z is a reasonable alternative — what do you think?" makes you a source of options. That is an entirely different relationship, and it compounds quickly.

Even when your proposal is wrong, it gives the conversation somewhere to start, and it shows that you thought before escalating.

2. Never let them be surprised.

Bad news early is a manageable problem. Bad news late is a crisis with your name on it.

The instinct to wait — to see if you can fix it first, to avoid looking incompetent — is understandable and it is nearly always wrong. Managers can absorb "this might slip, here's what I'm doing about it." They cannot absorb finding out from someone else.

This applies especially to things that are your fault. Particularly those.

3. Tell them what you need.

Managers are not mind readers, and most people never actually ask for anything. They wait to be offered training, a project, a conversation about progression, and then feel overlooked when it does not arrive.

Ask directly: *"I'd like to move toward [thing]. What would I need to demonstrate for that to be realistic?"* That question is easy to answer and it puts you on the manager's radar for exactly what you want.

4. Learn how they prefer to receive information.

Some managers want detail; some want a headline and will ask for detail if they want it. Some want written; some want a two-minute conversation. Some want to hear about problems immediately; some want them batched.

Give them theirs, not yours. This is not sycophancy — it is the same courtesy you would extend to any colleague whose working style differed from your own. It dramatically improves how your input lands.

The One-to-One

Most people treat the one-to-one as a status update. That is a waste of the most valuable half hour you get.

Better structure:

- Two minutes of status, because they do need it
- The thing you actually need a decision or unblocking on
- One thing about your development or direction
- Anything they need from you

Bring an agenda, even a three-line one. Almost nobody does, and it changes the meeting entirely.

Framework: The Escalation Decision

When something goes wrong, decide quickly:

- **Can I fix it today, with no visible impact?** Fix it, then mention it in the next one-to-one. Do not hide it — mention it.
- **Might it slip or be visible to others?** Tell them now, with what you are doing about it.
- **Does it involve money, legal exposure, or another team?** Tell them now, regardless of your confidence.

■ **Is it my fault?** Tell them now, and lead with what you are doing about it rather than the explanation.

Extreme ownership: when it goes wrong on your watch, own it plainly and immediately, and go straight to what you are doing about it. No hedging, no distributed blame. I have failed more times than I can count — more than a hundred, probably more than five hundred, across businesses, hires, and decisions. What I have learned is that the fastest route through a failure is straight through the middle of it, out loud, early.

Real Example

Two people discover on Wednesday that a project will miss Friday's deadline.

The first hopes to recover it, works late, and tells the manager Friday afternoon. The manager has spent two days telling stakeholders it is on track. The problem is now the manager's credibility problem, and the engineer's reliability is what gets questioned.

The second messages Wednesday: "Heads up — Friday is at risk. The [component] is taking longer than estimated. I think we land Tuesday. If Friday is immovable, we could ship without [feature] and follow up. Want to talk through it?"

Same slip. In the second case the manager had two days to manage expectations, was given options, and now regards the engineer as reliable — which is the opposite of what most people fear will happen when they report bad news.

Case Study: The Manager Who Seemed Unsupportive

Worth considering before concluding your manager is the problem.

Someone feels overlooked: no interesting projects, no development conversations, no advocacy. They conclude their manager does not value them.

When they finally raise it directly, the manager is surprised. From the manager's side: this person never asked for anything, seemed content, handled their work independently, and never indicated any ambition. Meanwhile two other reports were asking regularly, so opportunities went to them.

Neither person behaved badly. The information simply never crossed the gap, because both assumed the other would raise it.

Some managers are genuinely bad, and no amount of managing upward fixes that — Book 9 covers when the environment is the problem. But check this pattern first, because it is more common.

Templates: The Bad News Message

Heads up — [thing] is at risk.

WHAT'S HAPPENING: one line

WHY: brief, factual, no defensiveness

WHAT I'M DOING: the action already taken

OPTIONS: two, if there are two

WHAT I NEED FROM YOU: or "nothing, just visibility"

Templates: The One-to-One Agenda

1. Status, briefly
2. [Thing I need a decision on]
3. [Thing I need unblocked]
4. Development: [one thing about direction]
5. Anything you need from me?

Reflection

- Does your manager know what you want next?
- When did you last bring a problem with options attached?
- Have you ever waited on bad news hoping to fix it first?

Journal

Write down what you want from your career in the next two years. Then ask whether your manager knows any of it. Most people find the answer is no, and that nobody but them can fix that.

Exercises

1. Bring an agenda to your next one-to-one.
2. Ask directly: "what would I need to demonstrate to move toward [thing]?"

3. The next time something goes wrong, report it the same day.

AI Prompts

- "Help me write a message telling my manager a deadline is at risk, leading with what I am doing about it rather than the explanation."
- "I want to move toward [role/area]. Draft three questions to ask my manager that make it easy for them to help."
- "Here is a problem I need to escalate. Turn it into: what's happening, why, what I've done, two options, what I need."

Books

- *Extreme Ownership* — Willink and Babin
- *The Manager's Path* — Camille Fournier. Written for managers, and extremely useful for understanding what yours is dealing with.
- *Crucial Conversations* — Patterson et al.

Checklist

- I bring options, not just problems
- My manager is never surprised
- I have explicitly said what I want next
- I know how my manager prefers to receive information
- I bring an agenda to one-to-ones

Common Mistakes

- Waiting on bad news hoping to fix it first
- Treating one-to-ones as status updates
- Never asking for anything, then feeling overlooked
- Delivering information in your preferred format rather than theirs

Key Takeaways

- Bad news early is a problem; bad news late is a crisis with your name on it
- Options make you a source of solutions rather than a source of work
- Managers are not mind readers and most people never ask

- Own failures immediately and lead with the remedy

Next Mission

The part of this book that matters most, and that has nothing to do with your own career.

PART

THREE

Carrying It Forward

Mentoring, and the Person Who Did It For You

MISSION

To make you someone else's turning point.

Story

I was placed in special education classes as a child and told, repeatedly, that I could not learn. I believed it for a long time, because when you are a child and adults tell you what you are, you tend to take their word for it.

I was a farm-worker family's kid. I did not speak English well. The system had already decided what I was, and there was no particular reason for anyone to spend extra effort on me.

In sixth grade, a teacher named Miss England stayed after class to teach me the fundamentals nobody else had bothered to teach me. She did not have to. It was not part of her job, and I was not an easy student.

She gave me hope when I had none.

I did not learn English until around sixteen. I earned my high school diploma as an adult, almost thirty. Every part of my career — the businesses, the operations work, the podcast — came after that. And none of it happens without a teacher who decided I was worth extra time.

I have never been able to tell her what it meant. So I am telling you: **somebody did that for you, or you needed somebody to and nobody came.** Either way, you now have the ability to be that person for someone else.

Why It Matters

This is not a soft closing chapter, and I want to be clear about that.

Mentoring is one of the fastest ways to develop your own capability, because teaching exposes exactly where your understanding is shallow. You cannot fake your way through explaining something to someone who is genuinely trying to learn it. They ask the question you have been avoiding, and they ask it innocently.

It is also how you build a reputation that outlives any particular job. People remember who helped them when they were nobody. Ten years from now, some of the people you help will be in positions to decide things, and they will remember.

But the honest reason is simpler than either of those. It is the one in the story above.

Framework: How to Actually Mentor

1. **Ask what they want; do not assume.** The most common mentoring failure is answering a question nobody asked. Start with "what are you trying to figure out?" rather than with your own path.
2. **Give the next step, not the whole map.** Overwhelming someone is the same as helping nobody. One thing to do this week. The full picture is discouraging when you are at the start, and you have forgotten how discouraging because you are not at the start anymore.
3. **Share the failures.** Your polished version is useless to them. The story about the bad hire, the thing you broke, the decision you got wrong for two years — that is what actually helps, because it makes the path look survivable rather than impossible.
4. **Introduce them to people.** This is the single highest-value thing you can do and it costs you almost nothing. An introduction is worth more than an hour of advice, nearly every time.
5. **Be honest, kindly.** If they are heading somewhere that will not work, say so plainly. Encouragement without honesty is just flattery with a longer timeline, and it wastes their time in a way they cannot detect until much later.
6. **Do not need gratitude.** Some will disappear. Some will take your advice and never mention it. That is fine — it was never a transaction, and treating it as one ruins it.

Professional Explanation: What Mentoring Is Not

- **It is not a formal program.** Most of the best mentoring is informal, unnamed, and consists of answering questions properly over a long period.
- **It is not one wise person and one student.** The personal board below applies in reverse: you will be one of several partial mentors to anyone you help, and that is correct.
- **It is not a time commitment you must be certain you can sustain.** Answering one question well is mentoring. You do not need to promise a relationship.
- **It is not for people exactly like you.** The most useful mentoring frequently crosses lines — different background, different lane, different starting point. Those are also the people least likely to have anyone.

Framework: The Personal Board, In Reverse

THE PERSONAL BOARD OF DIRECTORS



Most of them will never know they're on it. You just need to know who they are.

The personal board. You need one, and you are on somebody else's.

The personal board is usually framed as something you assemble for yourself:

- Someone **technically ahead** of you, who can tell you what to learn next
- Someone **in the role you want**, who can tell you what it is actually like
- Someone **outside your field entirely**, who breaks your assumptions
- Someone **behind you**, whom you mentor — because teaching exposes what you do not understand
- Someone **who will be honest with you**, even when it stings. The rarest, and the most valuable.

You do not need to formally ask anyone. Most of them will never know they are on it. You just need to know who they are and stay in genuine contact — Book 1's contact ledger is the mechanism.

The part worth sitting with: **you are the fourth seat on somebody else's board right now, or you could be.**

On failure. I have failed more than a hundred times. Probably more than five hundred. Businesses, hires, decisions, relationships with people I should have handled better. If you take one thing from this book to whoever you mentor, take this one — because the reason most people stop is that they believe failing means they were wrong to start. Failure is not the opposite of success. It is part of it.

Real Example

Someone posts a question in a community — a beginner question, phrased awkwardly, the kind that usually gets a link and a slightly impatient tone.

The typical response: a link to documentation, or "just google it."

The response that matters: a direct answer to the actual question, one sentence about the thing they will hit next, and an offer to answer follow-ups.

That takes four minutes. For the person receiving it, it is frequently the difference between continuing and quitting. Not because the information was unavailable, but because someone treated them as worth answering.

I know this because I was that person, repeatedly, and the people who answered me properly are the reason I kept going.

Case Study: What Happens Ten Years Later

A pattern worth understanding for entirely practical reasons, beyond the obvious ones.

People you help early move up. Some of them move up faster than you. In ten years, the person who was asking beginner questions is running a team, sitting on a hiring panel, or deciding which vendor to use.

They remember exactly who was generous with them and who was dismissive. Not as a grudge — just as a fact about who a person is.

This is the least important reason to mentor and it is worth knowing, because people who think of generosity as costly are usually running an incomplete math.

Reflection

- Who was your Miss England? Have you told them?
- Who is one step behind you right now?
- What would you have needed to hear when you were starting?

Journal

Write the message you wish someone had sent you when you were starting out. Then find someone to send it to.

Exercises

1. Find one person earlier on the path than you — in a community, at work, in a chapter meeting, in a comment section — and answer one question properly this week. Not a link. A real answer.
2. Make one introduction between two people who should know each other.
3. Tell someone who helped you what it meant. Do it before you cannot.

Project

Commit to one hour a month, for six months, helping people earlier on the path than you. Community, chapter, workplace, anywhere. Six hours total. Then look at what it did to your own understanding.

AI Prompts

- "Someone asked me [beginner question]. Help me answer it in a way that actually teaches rather than just linking documentation, and tell me what they will likely hit next."
- "I want to mentor someone but I am not senior. What can I genuinely offer someone two years behind me?"
- "Help me write a message to someone who helped me years ago, telling them what it meant."

Books

- *The Coaching Habit* — Michael Bungay Stanier. Short, practical, and mostly about asking rather than telling.
- *So Good They Can't Ignore You* — Cal Newport
- *Give and Take* — Adam Grant. On why generous people end up ahead, with the data.

Podcast Episodes

Nearly every Wake Up X episode contains a moment where someone credits a person who helped them. Listen for it. It is remarkably consistent, and the guests rarely frame it as the turning point it was.

Resources

- Your local chapter, where newer people show up looking for exactly this
- Community forums and Discord servers in your lane
- Anyone at your workplace who is newer than you

A note on Dr. Eric Cole. He spent a long career in cybersecurity, and he spent a lot of it teaching. If you look him up, you will find hours of him explaining hard things simply and telling people early in their careers that they belonged here.

He died in May 2026. I am including him in this chapter on purpose. He is a good example of what this whole book is arguing: the work outlasts the job title, and the people you helped remember you longer than the projects do.

Checklist

- I have at least one person I am actively helping
- I share my failures, not just my results
- I make introductions rather than only giving advice
- I ask what they want before I give it
- I have thanked someone who helped me

Common Mistakes

- Talking about yourself instead of asking about them
- Giving the whole roadmap when they needed one next step
- Only sharing wins, which makes your path look impossible
- Mentoring only people who resemble you
- Waiting until you feel senior enough

Key Takeaways

- Teaching is the fastest way to find the holes in your own knowledge
- The failures are the useful part of your story
- An introduction is worth more than an hour of advice
- Somebody did this for you, or you needed them to

Next Mission

Book 6: using AI as an actual working tool.

Appendix: Scripts, Structures, and the Success File

REFERENCE

Everything from this book in one place.

The Five Sources of Influence

COMPETENCE	reliably good at something needed
RELIABILITY	you do what you said, when you said
GENEROSITY	you help people who can't repay you
CLARITY	you make complicated things simple
COMPOSURE	you stay steady when things break

None need a title, a budget, or approval.

The Success File Entry

DATE:	
WHAT I DID:	one line
WHY IT MATTERED:	impact, concretely
EVIDENCE:	email, ticket, screenshot, who saw it
NUMBERS:	if any, honestly
WHO BENEFITED:	
RESUME BULLET:	the one-line version, written now

Add weekly. Not annually.

The Disagreement Script

"We both want [shared goal].

My concern is specifically [narrow issue] –
[reason, briefly].

Can we talk about that piece?

If [resolution condition], I'd be comfortable."

The Commit Script

"Understood – I'll make it work. If [specific thing] comes up, I'll flag it early rather than relitigating."

The Feedback Conversation

- | | |
|------------------------|-----------------------------------|
| 1. Ask permission | "Can I give you feedback on X?" |
| 2. State the behaviour | specific, observable, recent |
| 3. State the impact | factually |
| 4. Ask for their view | they may know something you don't |
| 5. Agree what changes | concrete and mutual |
| 6. Offer support | "what would help?" |

Receiving Feedback

"Thank you. Let me think about that."

Then, a day or two later if needed:

"I've been thinking about what you said. I want to add some context – not to argue, just so you have the full picture."

Questions That Get Real Feedback

What's one thing I could have done better in that meeting?
If you had to name the thing that would most improve my work, what would it be?
What do you think I'm worse at than I realize?

The Bad News Message

Heads up – [thing] is at risk.

WHAT'S HAPPENING:

WHY: factual, no defensiveness

WHAT I'M DOING: action already taken

OPTIONS: two, if there are two

WHAT I NEED FROM YOU: or "nothing, just visibility"

The One-to-One Agenda

1. Status, briefly
2. [Decision I need]
3. [Thing I need unblocked]
4. Development: [one thing about direction]
5. Anything you need from me?

The Personal Board

- | | | |
|---|----------------------|--------------------------|
| 1 | AHEAD OF YOU | what to learn next |
| 2 | IN THE ROLE YOU WANT | what it's actually like |
| 3 | OUTSIDE YOUR FIELD | breaks your assumptions |
| 4 | BEHIND YOU | teaching finds your gaps |
| 5 | HONEST WITH YOU | the rarest one |

You are seat 4 on somebody else's board.

How to Mentor

1. Ask what they want; don't assume
2. Give the next step, not the whole map
3. Share the failures
4. Introduce them to people
5. Be honest, kindly
6. Don't require gratitude

A Closing Note

Everything in this book is available to you today, without permission, without a title, and without anyone's approval.

That is either liberating or uncomfortable depending on where you are sitting when you read it, because it removes the explanation most people are using. If none of this needs authority, then not doing it is a choice.

I was told as a child that I could not learn. I did not speak the language of the country I lived in until I was sixteen. I got my diploma at nearly thirty. Nothing about my path suggested I would end up leading anything.

What changed it was not talent and it was not a plan. It was one teacher who stayed after class, and then years of doing slightly more than I was asked to do, mostly because I was curious rather than because I was strategic.

You are somebody's Miss England, or you could be. That is the whole book.

Lead first. Make it visible. Then turn around and reach back.

Connect

[linkedin.com/in/efrenlsalazar](https://www.linkedin.com/in/efrenlsalazar) | wakeupxpodcast.com

Questions for the author. The Miss England material is the strongest passage in the entire library and it is already here. What is missing closes the loop the chapter opens: **a story of someone you have mentored and where they ended up.** Right now the chapter argues that mentoring matters using only the story of being mentored. One paragraph about someone you helped — even a small thing, even someone whose name you would change — would complete it. Second gap: a moment where making your work visible directly changed an outcome for you, for Chapter 2.

Leading Inside the Community

Years of chapter volunteering, board service, and standing in front of the room — leadership built through showing up, not a title.



ISACA chapter conference, with fellow members



ISACA Sacramento Chapter, full chapter photo

IN THE FIELD

Still Showing Up

More time with the ISACA Sacramento chapter, on the floor and behind the booth.



ISACA Sacramento Chapter, working the chapter table



RE/THINK Sacramento, community advocacy

Your Network Is Your Net Worth

Every room I've stood in — a chapter meeting, a conference floor, a podcast interview — taught me something a certification never could. Your network doesn't just open doors. It carries you when a bad month, a layoff, or a bad decision would otherwise carry you alone. That's not a metaphor. It's the difference between quitting and getting a phone call at the right moment.

The Networking Engine

In Person

Chapters, conferences, meetups. Handshakes build trust faster than any message thread.

Digital

LinkedIn, comments, DMs. Show up consistently, not just when you need something.

Depth

Pick at least one community to go deep in, instead of staying surface-level in ten.



Museum of Science and Curiosity, community turnout

PROOF IN THE ROOM

Strangers Become Your Network

Most of the people in these rooms were strangers before the handshake. A conversation at a booth, a shared badge scan, a LinkedIn comment that turned into a real conversation — none of it looked like “networking” while it was happening. It just looked like showing up.



A conversation at the conference expo floor

REFLECTION

Are you spread thin across many communities, or deep in one? Which person in your network could you reach out to this week, just to check in?

You Don't Have to Take My Word For It

Miss England spent extra time after class on a kid nobody expected much from, and it changed what I thought was possible for myself. I don't know if I'll ever get to tell her that. What I can do is try to be that for someone else — and let the people I've actually helped describe it in their own words instead of mine.

In their own words

Verified LinkedIn recommendations, from wakeuypodcast.com/voices

Ghazala Ahmed

Compliance Analyst

Pointed her toward the resources she needed and stayed supportive through a job search until she landed her next role.

Dennis Maksimov

Cybersecurity Analyst, Healthcare Compliance

Ongoing troubleshooting help, learning resources, and encouragement to keep growing, described as freely given.

Jose Kyle de Luna

Computer Engineering, Device Support Technician

Called him a reliable mentor in his first months in the field — organization, technical knowledge, and teamwork included.

REFLECTION

This isn't a claim you have to take on faith — you can read the rest at wakeuypodcast.com/voices. The real question for you: who's one step behind you right now that a five-minute message could actually help?

Ryan Doctor — Sideways Into Cybersecurity

Ryan Doctor spent over 10 years in IT operations — IT service management, incident management, change management — before cybersecurity ever became the goal. It became the goal because of the work itself.

The Door Opened Sideways, Not Head-On

He didn't set out to break into cybersecurity. He was collaborating with a security team on cybersecurity-related incidents, system risk assessments, and data classification reports as part of his existing IT role — and that collaborative work was what pulled his interest toward the field, not the other way around. From there, he started attending ISC2 meetups to learn more, and a connection made there introduced him to ISACA, where he later became a chapter officer himself.

One Piece of Advice, Repeated Without Hedging

Asked what he'd tell someone trying to get into cybersecurity or just trying to learn more, his answer was immediate and singular: network. "It has opened up a lot of opportunities... that has added me opportunities for interviews." No list of certifications, no roadmap — just the one thing he's watched work, repeatedly, in his own path.

REFLECTION

Is there a project or collaboration already in front of you — not a job title — that could pull you toward the field the way this one pulled Ryan?

Watch the full episode: youtu.be/tjC2-sdFnJw

Cathy Stephenson — The Knowledge Sponge

Cathy Stephenson is on the Deputy CISO track after years in enterprise architecture, project management, and a state SOC — built entirely by walking into her manager's office and asking to learn things outside her actual job description.

She Picked the Hardest Option on the List

Choosing a degree, she scanned the options, picked cybersecurity/forensics because it was labeled the hardest, and told her counselor "yes, sign me up" when warned off it. Calls herself a "knowledge sponge" — her rule is that if she doesn't understand something, she goes and finds the person doing it, rather than settling for reading about it.

She Walked Out of a CISO Interview

Mid-interview, once she realized the advertised job wasn't the real job, she said so directly: "I don't want to waste your time... I will not benefit your agency if I come here," and ended the interview herself. Research isn't just prep for her — it's a filter she's willing to act on.

The Mentorship Goal, Stated Plainly

"My goal is when I'm mentoring someone, I want you to be my boss one day. If you could be my boss one day, then I'm successful too." Her practical version of staying in touch: check in with people she's mentoring roughly monthly — goals, progress, what she can do to help.

Watch the full episode: youtu.be/PU3tAlAKjGs

Cathy's Advice, In Practice

Once you take money off the table, the stress goes with it.

Her advice for anyone chasing "how do I get into security": figure out which specific lane (GRC, incident response, executive, technical) before anything else — not the paycheck first.

Volunteer and network while you're still in school, not after.

College is "your education career" — the connections need to already exist by the time you graduate, not start then.

Stress management is a menu, not one trick.

Weekly coffee with a friend, walking barefoot in the grass, EFT tapping — and a genuinely useful reframe: music, reading, and cooking something new all count as forms of meditation too.

Pay it forward, literally, not just as a saying.

Covered a stranger's declined purchase at a store rather than let her stay embarrassed — her rule: "you're gonna pay it forward... it'll come back."

REFLECTION

Have you ever researched an opportunity enough to realize, partway in, that it wasn't actually right — and did you act on that, or push through anyway?

BE OUR NEXT GUEST

Your Story Belongs on Wake Up X

If Lead Without a Title gave you something worth using, you already have something worth sharing. The Wake Up X podcast exists to put real, unpolished career and business stories in front of people who need to hear them — and every guest so far started exactly where you are now: reading, building, and figuring it out in public.

You might be a fit if you:

Broke into tech, cybersecurity, or business without the traditional path

Built something — a lab, a business, a portfolio, a community — and can talk about how

Have a failure or turning point you're willing to tell straight, not polished

Are mentoring, leading, or building something worth other people knowing about

HOW TO APPLY

Share your story at wakeupxpodcast.com/guest, or connect directly on LinkedIn at linkedin.com/in/efrenlsalazar and mention this book.

Being a guest is also visibility — the same principle Book 7 makes the case for.

Acknowledgments

THE PEOPLE WHO MADE THIS POSSIBLE

None of Wake Up X happened alone, including Lead Without a Title. Every book in this library, the podcast, and the free library exist because specific people showed up for a kid nobody expected much from. This is the record of who they are.

THE EARLY YEARS

Miss England, the 6th-grade teacher who stayed after class, unpaid, to teach the fundamentals. Sister Cora and Lewis Bair of Computers for Kids, who handed a teenager a key and a broken machine and let the work speak. Jose Luis Bravo, Angel Martinez, Isaac Murrilo, and Alex De Haro — the friends and cousins who translated the world, gave hope, drove him where he needed to go, and carried him before there was anything to carry him toward.

THE BUSINESS YEARS

Rojelio Salazar, his cousin and first mentor, RNS Computer Repairs — the reason it is called Wake Up X. Kyle Erni, his first business partner. Erick and Pancho Arauza, co-founders of The Cellphone Fix. Danny Maya, of Maya Traditional Mexican Restaurant, Taqueria Maya, and Salt and Water, who pushed him in front of a camera for the first time.

THE CAREER YEARS

Igor, who gave him the vocabulary for work he already knew how to do. Nate, who he mentored into a full-time role alongside his own. Ken Kerrick, opponent turned friend in the ISACA Sacramento election. Alvin Jue and David De los Reyes, who believed in the middle of it, not just at the finish.

THE PODCAST

Every guest who trusted an unproven show with their story, including Gustavo Mastroianni, Cathy Stephenson, Greg McCord, Aaron Portnoy, Sana Talwar, Jorge Avila, Anil Reddy, Devin Prasad, Larry Switzer, Louis Kemp, Mohana Balakrishnan, Eddie Lopez, and Michael Fierros — full episodes at wakeupxpodcast.com.

COMMUNITY PARTNERS

ISACA Sacramento and ISC2 Sacramento, The Cellphone Fix, A.N.U.L Technologies, AISOC, AspireByte, and PodGrabber — the organizations building alongside this movement.

AND

His parents, who arrived with nothing, worked since childhood, and gave away whatever was left over. Nothing about the first version of this story predicted the version you are holding now.

Continue the Journey

EVERY BOOK LEADS SOMEWHERE ELSE

Lead Without a Title is one part of a 12-book system. If it helped, the next step is one page away — take whichever layer matches where you are standing.

THE LIBRARY

Layer 1 — The Story

I Just Have To Wake Up — why any of this exists.

Layer 2 — The Mindset

Comfort Is Expensive — the real cost of playing it safe.

Layer 3 — The Blueprint

DECODE: The Wake Up X Playbook — zero experience to executive thinking. Book 0.

Layer 4 — The Playbooks

1 Break In · 2 Build It · 3 Think Like a Cybersecurity Professional · 4 Business First · 5 Lead Without a Title · 6 AI for IT & Cybersecurity · 7 Build Your Brand · 8 The Consultant Mindset · 9 Level Up

FIND US EVERYWHERE

Podcast & free library: wakeupxpodcast.com · Spotify: open.spotify.com/show/28GZnRBrIV2efHOhZdifft · Apple Podcasts: podcasts.apple.com/us/podcast/id1792821643 · YouTube: youtube.com/@Efren.wakeupx · LinkedIn: linkedin.com/in/efrenlsalazar · Substack: substack.com/@wakeupxefren

IN GOOD COMPANY

ISACA Sacramento, ISC2 Sacramento, The Cellphone Fix, A.N.U.L Technologies, AISOC, AspireByte, and PodGrabber — the partners and collaborators building alongside the movement. See the full list at wakeupxpodcast.com/community.html.

Get DECODE free, apply to be a guest, or read the story behind the movement — all of it starts at wakeupxpodcast.com.

WAKE UP X

SOLUTIONS



Efren L. Salazar

FOUNDER · HOST · CYBERSECURITY PROFESSIONAL · EDUCATOR · COMMUNITY BUILDER

STAY CONNECTED

Podcast & Blueprint Library:	wakeupxpodcast.com
LinkedIn:	linkedin.com/in/efrenlsalazar
YouTube:	youtube.com/@Efren.wakeupx
Be a Guest:	wakeupxpodcast.com/guest

THE WAKE UP X PUBLISHING SYSTEM

BOOK 6

AI for IT & Cybersecurity

Using AI as a working tool, not a novelty

A junior analyst who never sleeps and sometimes lies

THE
WAKEUPX
SOLUTIONS

EFREN L. SALAZAR

Founder & Host, Wake Up X | IT Operations Manager & Solution Engineer
Government & Regulatory Partner, ISACA Sacramento | CompTIA Security+

wakeupxpodcast.com

Educational content — not professional, legal, financial, or career-placement advice.
Full disclaimer in DECODE, the foundation of this library.

Before You Begin

HOW TO USE THIS BOOK

This book makes you fluent with AI as an actual instrument of work — with correct expectations about where it fails, and without letting it erode the fundamentals you need when it is wrong.

Where this book sits

Two books come first — the memoir I Just Have To Wake Up and the mindset book Comfort Is Expensive — before this ten-book library. Each one is short on purpose. Pick up the one that matches the problem in front of you right now.

- n **Book 0 — DECODE.** The foundation. The map of the whole path.
- n **Book 1 — Break In.** Landing your first role in IT or cybersecurity.
- n **Book 2 — Build It.** Home labs and a portfolio that prove you can do the work.
- n **Book 3 — Think Like a Cybersecurity Professional.** Detection, triage, and incident response thinking.
- n **Book 4 — Business First.** Finance, sales, GRC, and why the business hired you.
- n **Book 5 — Lead Without a Title.** Influence, EQ, and mentoring before anyone promotes you.
- n **Book 6 — AI for IT & Cybersecurity.** Using AI as an actual working tool, not a novelty. **you are here**
- n **Book 7 — Build Your Brand.** LinkedIn, content, and being findable.
- n **Book 8 — The Consultant Mindset.** Discovery, solution selling, and stakeholder management.
- n **Book 9 — Level Up.** Long-term growth, burnout, legacy, and mentoring others.

Book 0 is the foundation. *DECODE: The Wake Up X Playbook* holds the six-level path, the Curiosity Compass, and the core frameworks. Nothing in any other book contradicts it. If a term here is unfamiliar, DECODE defines it.

What is in this book

Part One — Correct Expectations

1. A Junior Analyst Who Never Sleeps and Sometimes Lies
2. The Rules That Keep You Employed

Part Two — Real Work

3. Where It Earns Its Keep
4. Prompting Like an Engineer

Part Three — Building With It

5. Automation: Where Fluency Becomes Capability
6. Staying Sharp While Using It
7. Appendix: Prompts, Patterns, and the Verification Gate

How to actually use it

This book deliberately avoids naming specific model versions or comparing products, because that information is stale within months and you can find it anywhere. What does not go stale is how to reason about these tools, where they fail, and how to build with them without becoming dependent.

If you use nothing else from this book, use Chapter 2. The rules there are the ones that determine whether AI helps your career or damages it.

A note on pace. This is the fastest-moving area in the library, and specifics here will age. The judgment will not. Where I describe a capability, assume it has improved; where I describe a failure mode, verify rather than assume it has been fixed.

I am writing this the way I would say it to you sitting across a table — not lecturing a room. Take what works, remix the rest into your own version, and leave what does not fit.

PART

ONE

Correct Expectations

A Junior Analyst Who Never Sleeps and Sometimes Lies

MISSION

To give you accurate expectations, so you use these tools well instead of either dismissing them or over-trusting them.

Objectives

- Understand what these tools are genuinely good and bad at
- Adopt a working mental model that holds up
- Stop worrying about the wrong risk

Why It Matters

There are two common postures and both are wrong.

The first is dismissal: it makes things up, it is a toy, real engineers do not need it. This posture is increasingly expensive, and the people holding it are being outproduced by people who are not more capable, just faster.

The second is over-trust: pasting output into production, into documents, into client communication, without reading it carefully. This is how people get embarrassed, and occasionally how they get fired.

The correct posture sits between them and needs actually understanding the failure modes.

The people losing work to AI are not being replaced by AI. They are being replaced by people who use it well. That distinction determines what you should do about it, which is: get fluent quickly, and stay technical.

Professional Explanation

What It Is Genuinely Good At

WHAT AI IS AND ISN'T FOR

GENUINELY GOOD AT

- First drafts of anything
- Explaining at your level
- Translating between formats
- Finding what's missing
- Rubber-ducking

UNRELIABLE AT

- Facts it wasn't given
- Your specific environment
- Recency
- Knowing when it's wrong

*every one of these must be
verified independently*

Treat every output as a draft from a fast, confident junior who has never seen your environment.

Correct expectations. The left column is where the value is; the right column is where careers get damaged.

First drafts of anything. Documentation, reports, policies, emails, scripts, runbooks. Editing a mediocre draft is dramatically faster than producing a blank-page one, and the blank page is where most people stall.

Explaining unfamiliar things at your level. Ask it to explain a protocol, a log format, or a regulation as if you already understand the adjacent material. This is a genuinely excellent tutor. It is patient in a way people are not — you can ask the same question four different ways without embarrassment.

Translating between formats. Log to summary. Technical to executive. Notes to structured document. Rough thinking to organized argument.

Finding what is missing. "Here is my resume and this posting. What is in the posting that is absent from my resume?" Gap-finding is much more reliable than generation, because it is a comparison task rather than a recall task.

Rubber-ducking. Explaining your problem to it frequently solves the problem before it answers. This is not a joke; articulating a problem exactly is most of debugging.

What It Is Bad At

Facts it was not given. It will produce confident, specific, false detail. Version numbers, CVE identifiers, command flags, citations, statistics, quotes. Every one of these must be verified independently, every time.

Your specific environment. It does not know your network, your constraints, your politics, or why the thing that obviously should work does not work at your organization.

Recency. Its knowledge has an edge, and it will not always tell you where that edge is.

Knowing when it is wrong. There is no reliable internal signal. Confidence in the output is not correlated with correctness — a completely fabricated CVE number arrives in the same tone as a correct one.

That last point is the one people underestimate. With a human junior analyst, uncertainty leaks through — they hedge, they pause, they say "I think." These tools do not reliably do that.

Framework: The Working Mental Model

Treat every output as a draft from a fast, confident junior analyst who has never seen your environment and will not tell you when they are guessing.

You would review that person's work carefully. You would not paste it into a client email unread. You would find their drafts genuinely useful and you would check anything factual.

That framing is correct nearly all of the time, and it generates the right behaviour without requiring you to think about it. When in doubt, ask: what would I do if a smart intern handed me this?

Why the Analogy Holds

- **Fast.** Produces in seconds what would take you an hour.
- **Confident.** Presents everything with equal certainty.
- **Junior.** Broad shallow knowledge, no judgment about what matters here.
- **Never seen your environment.** Cannot know your constraints unless you supply them.
- **Will not flag uncertainty.** The critical difference from an actual junior. The reason you cannot outsource the checking.

Real Example

Someone asks for help configuring a firewall rule and receives a confident, well-formatted answer including specific syntax and a flag that does not exist in that product.

The answer *looks* authoritative. The syntax is plausible. Someone who does not know the product would not detect it, and someone in a hurry would paste it.

This is the characteristic failure: not obviously wrong output, but plausible output with a specific fabricated detail embedded in otherwise correct material. That is harder to catch than complete nonsense. That is why the verification habit has to be structural rather than triggered by suspicion.

Case Study: The Two Colleagues

Two people on the same team adopt these tools in the same month.

The first uses them to skip understanding. He generates scripts he does not read, pastes explanations into tickets he does not follow, and produces more output than before. For about four months this works well and he looks productive.

Then something breaks in a way the tool cannot help with. A problem specific to their environment, requiring understanding of a system he has been generating around rather than learning. He cannot help. Worse, several of the scripts he deployed turn out to contain assumptions nobody checked.

The second uses them to go faster at things she already understands, and to learn things she does not. Asking for explanations, then verifying them, then building the thing herself. Her output increases too, less dramatically.

A year later she is a lot more capable than before. He is somewhat less capable than before, and dependent.

The tool did not cause either outcome. It amplified an existing disposition, which is what tools do.

Reflection

- Which posture are you closer to: dismissal or over-trust?
- Have you ever accepted an AI-generated specific without checking it?
- Are you using it to go faster, or to avoid learning?

Journal

Write about something you understand deeply, and how you came to understand it. Then ask honestly whether that process would have survived having an AI available. The answer is instructive either way.

Exercises

1. Ask a question in an area you know well and count the errors. Most people find one or two subtle ones, which calibrates trust better than any warning.
2. Take an explanation it gives you and verify every specific claim independently.
3. Notice, for one week, every time you accept output without checking.

Common Mistakes

- Trusting confident specifics
- Dismissing the tools entirely and being outproduced
- Using it to skip learning fundamentals
- Assuming a failure mode described anywhere is still current, or still fixed

Key Takeaways

- Great at drafts, explanation, translation, gap-finding, and rubber-ducking
- Unreliable on facts, your environment, recency, and its own uncertainty
- Confidence is not correlated with correctness
- The tool amplifies your existing disposition rather than changing it

Next Mission

The rules that determine whether this helps your career or damages it.

The Rules That Keep You Employed

MISSION

To keep you out of the specific trouble these tools create.

Objectives

- Know what must never be pasted into an unapproved tool
- Understand where accountability actually sits
- Handle the absence of a policy correctly

Why It Matters

This is the shortest chapter in the book and the most important one. Everything else here makes you more effective. This chapter keeps you employed.

Professional Explanation

Rule One: Data

Never paste customer data, credentials, proprietary code, personal data, or anything covered by a regulation into a tool your organization has not approved.

This is the single most common way careers get damaged with these tools right now. Not because the tools are malicious, but because:

- The data has left your control, and you cannot get it back
- You may have violated a contract commitment to a customer
- You may have violated a regulation with real penalties — GDPR, HIPAA, PCI DSS, all of which Book 4 covers
- You cannot prove what happened to it afterward

Note what is on that list: **proprietary code counts**. So do log excerpts containing internal hostnames and IP addresses, incident details, unreleased business information, and anything about an identifiable person.

If there is no policy, that is not permission — that is a gap. And you should be the person who raises it, which is also a Book 5 visibility opportunity. "We do not have guidance on this and people are using these tools daily" is a genuinely useful thing to raise. It is the kind of thing that gets noticed.

Rule Two: Accountability

The tool carries no accountability. You carry all of it.

If you send an AI-drafted email with a false claim in it, you made the false claim. If you deploy a generated script that deletes something, you deleted it. "The AI wrote it" is not a defense anywhere, and attempting it will damage you more than the original error.

This is not a legal technicality. It is the reason the verification habit has to be automatic — because the consequences flow to you regardless of where the text came from.

Rule Three: Code

Never run generated code you do not understand. Especially anything that deletes, moves, modifies at scale, or touches permissions.

The workflow that is safe:

1. Read it. All of it.
2. Ask it to explain any part you do not follow — and verify that explanation
3. Run it in a lab or against a copy first
4. Check what it would do before letting it do it, where the tool supports that
5. Then run it for real, with a rollback plan

Book 2's snapshot habit applies directly here.

Rule Four: Voice

Anything with your name on it should sound like you.

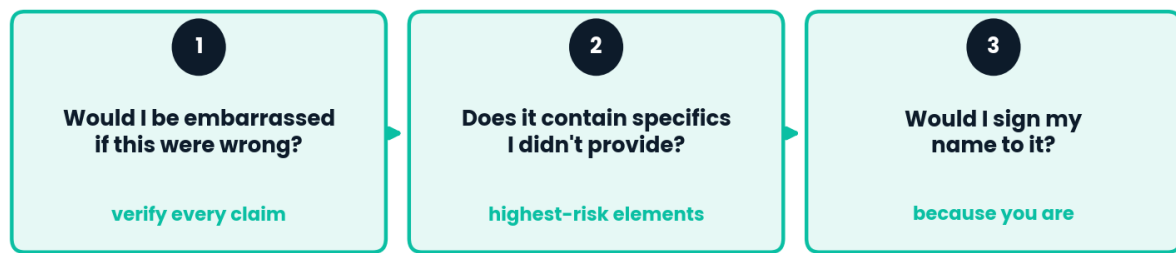
Partly this is about detection — AI-written text has a recognizable texture, and readers increasingly notice. But mostly it is that you will be asked about it.

A LinkedIn post you did not write, a document you did not think through, an explanation you cannot defend in conversation. Each of these creates a moment where the gap becomes visible.

Use it to draft, then rewrite in your own words. The rewriting is where you actually think about it.

Framework: The Verification Gate

THE VERIFICATION GATE



The tool carries no accountability. You carry all of it.

Three questions before any output leaves your hands.

1 — Would I be embarrassed if this were wrong? If yes, verify every factual claim independently, from a primary source.

2 — Does it contain specifics I did not provide? Numbers, names, versions, commands, citations, dates. Those are the highest-risk elements in any output, and they are exactly the elements that look most authoritative.

3 — Would I sign my name to it? Because you are about to.

Three questions, about fifteen seconds. Applied consistently, they prevent essentially every embarrassing outcome available with these tools.

Real Example

An analyst asks for a summary of a regulation's requirements to include in a report for leadership.

The output is well-organized and includes a specific article number and a specific penalty figure.

Without the gate: it goes into the report. The article number is wrong. Someone in legal notices. The analyst's credibility on regulatory matters is now compromised, and the report is re-examined entirely.

With the gate: question 1 fires immediately — this is going to leadership, being wrong would be embarrassing. Question 2 identifies the article number and the penalty figure as unprovided specifics. Two minutes of checking against the actual regulation text. One is right, one is not.

Total cost of the gate: two minutes. Total cost of skipping it: a year of being second-guessed.

Case Study: The Policy Nobody Wrote

An extremely common organizational situation right now.

Employees are using AI tools daily. There is no policy. Nobody has said it is allowed and nobody has said it is not. Everyone assumes someone has thought about it.

Then a customer asks, during a security questionnaire, whether their data has been processed by third-party AI services. Nobody can answer, because nobody knows what anyone pasted where.

The person who raises this before it happens is doing genuinely valuable work — and it is exactly the kind of unassigned, visible, business-relevant contribution Book 5 is about. It also happens to be a natural entry point into GRC work, per Book 4.

The framing that works: not "we should ban these tools," which is unrealistic and will be routed around, but "people are using these already and we have no guidance, so let us define what is acceptable."

Templates: A Personal Rule Set

If your organization has no policy, hold yourself to this until it does.

NEVER PASTE:

customer data	credentials or keys
personal data	proprietary code
incident details	internal hostnames / IPs
unreleased business information	
anything a contract or regulation covers	

ALWAYS:

- read generated code fully before running
- test in a lab before production
- verify every specific I did not provide
- rewrite anything with my name on it

IF UNSURE:

- assume it's sensitive and don't paste it
- ask, and raise the absence of a policy

Reflection

- Does your organization have a policy? Do you actually know, or do you assume?
- Have you ever pasted something you should not have?
- Would you be comfortable if the full contents of your prompts were reviewed?

Journal

Write down what you have pasted into these tools in the last month. Be honest with yourself. Most people find at least one item that should not have gone in. The point is not guilt — it is recognizing that the habit forms without noticing.

Exercises

1. Find out whether your organization has a policy. Actually find out.
2. If there is none, draft a one-page proposal using Book 4's format and raise it.
3. Apply the verification gate to everything for one week and count how often question 2 catches something.

AI Prompts

- "Here is a piece of text I want to work on. Before I paste it — help me identify what in it would be sensitive and should be redacted first." (*Ask this with a description, not the text itself.*)
- "Draft a one-page acceptable-use policy for AI tools at a mid-size organization, covering what may and may not be submitted."
- "Review this script and tell me every operation that is destructive, irreversible, or assumes conditions I have not verified."

Books

- *The Cybersecurity Manager's Guide* — Todd Barnum
- Your own organization's data classification policy, if one exists — the mapping to AI tools is usually direct

Resources

- **NIST AI Risk Management Framework** — the emerging structured reference
- Your legal or compliance team, who would generally much rather be asked than surprised

Checklist

- I know my organization's policy, or I know there is not one

- I have never pasted sensitive data into an unapproved tool
- I read generated code fully before running it
- I apply the three-question gate before anything leaves my hands
- Anything with my name on it sounds like me

Common Mistakes

- Treating absence of policy as permission
- Forgetting that proprietary code and log excerpts count as sensitive
- Running generated code unread
- "The AI wrote it" as an explanation
- Assuming the tool flags its own uncertainty

Key Takeaways

- Never paste sensitive data into unapproved tools — the most common career damage
- You carry all the accountability; the tool carries none
- Three-question gate, fifteen seconds, prevents nearly everything
- Raising the missing policy is valuable work and a visible contribution

Next Mission

Where these tools actually earn their keep in real work.

PART

TWO

Real Work

Where It Earns Its Keep

MISSION

To give you concrete, defensible applications rather than general enthusiasm.

Objectives

- Apply these tools to documentation, analysis, scripting, study, and career work
- Know the hard limits in each area
- Build habits rather than one-off uses

Step-by-Step Guide: The Five Areas

1. Documentation and Writing

The highest-value, lowest-risk use available to you. Feed it your rough notes and structure; get back a draft you edit.

This matters far more than in IT and security because documentation is universally under-done and universally valued. If AI makes you the person whose work is always documented, that is a visible differentiator that costs you almost nothing.

- "Here are my notes from a lab build. Turn them into a README with these sections. Do not add anything I did not describe."
- "Rewrite this incident summary for a non-technical executive. Lead with impact, keep it under 150 words."
- "Turn this policy draft into plain language at an eighth-grade reading level without changing what it needs."
- "Here is a runbook I wrote quickly. What steps have I assumed the reader knows?"

That last prompt is excellent and underused. Documentation fails mostly through assumed knowledge, and a fresh reader — even a synthetic one — catches it.

2. Log and Alert Analysis

Useful with a hard limit: it can help you *read*, it cannot be your judgment. Book 3's triage loop stays yours.

- "Explain what this log entry means, field by field."
- "Here is a sequence of events. Build a timeline and tell me what is unexplained."
- "What are three benign explanations for this pattern that I should rule out before escalating?"

That third prompt is genuinely valuable, because it counteracts the most common junior-analyst error: anchoring on the first bad explanation and then gathering only confirming evidence. Asking for benign alternatives forces the disconfirming step that Book 3 says you should do anyway.

The limit: do not paste production logs into unapproved tools. Internal hostnames, IP addresses, usernames, and file paths are all sensitive. Redact or reconstruct a sanitized version.

3. Scripting and Automation

- "Write a PowerShell script that does X. Add comments explaining each block, and tell me what could go wrong when it runs."
- "Review this script for anything destructive or anything that assumes conditions I have not verified."
- "Explain what this script does, line by line, as if I have to defend it in a change review."

The second and third prompts are more valuable than the first. Reviewing and explaining are where these tools are most reliable, and where you are most likely to catch something in your own work.

4. Study and Certification

- "Explain subnetting as if I understand binary but not networking."
- "Quiz me on Security+ domain 3. After each answer, tell me what I got wrong and why — do not just give the correct answer."
- "I got this wrong. What is the underlying concept I am actually missing?"

That third prompt changes study outcomes more than any other in this book. Wrong answers cluster around a single misunderstood concept far more often than they scatter randomly, and finding the concept fixes ten questions rather than one.

The limit: verify anything factual against the actual objectives or vendor documentation. Certification content is specific and these tools blur specifics.

5. Career Work

Book 1's chapters lean on these directly.

- "Read this posting and my resume. List the gaps. Do not rewrite."
- "Act as a skeptical hiring manager. Give me the three hardest questions you would ask about this project."
- "I need to explain to my manager why this project is worth funding. Here are the facts. Draft three options including doing nothing." (*Book 4's format.*)

- "Here is my two-minute project description. Where would a skeptical interviewer push, and what am I glossing over?"

The adversarial prompts — act as a skeptic, find the weakness, what would you push on — are the most useful category and the least used. People ask for help making things sound good. Asking for help finding what is weak is more valuable.

Framework: Draft, Check, Own

Every use follows the same three steps.

Draft. Let it produce. Do not agonize over the prompt on the first pass; iterate instead.

Check. Apply the verification gate. Verify every unprovided specific.

Own. Rewrite in your voice, take responsibility, and be able to defend every line.

If you cannot do step three, do not use the output.

Real Example

An analyst needs to write an incident summary for leadership after a long night.

Without: either a rushed summary that buries the impact, or a delay while they find the energy to write it well.

With, done properly: they dump their timeline notes in — sanitized, no internal hostnames — and ask for a 150-word executive summary leading with impact. They get a structured draft in seconds. They then rewrite it in their own voice, correct two characterizations that were slightly off, verify the timestamps against their notes, and send it.

Fifteen minutes instead of an hour, and a better document, because the structure came from something that was not exhausted.

With, done badly: they paste the raw logs including hostnames and usernames into an unapproved tool, take the output unedited, and send a summary containing a confident claim about root cause that was inferred rather than established.

Same tool. The difference is entirely process.

Case Study: The Documentation Habit

The single highest-return use of these tools for most people in this field, and worth being specific about why.

Documentation is under-done everywhere because it is tedious and unrewarded in the moment. It is also one of the most reliable ways to become the person a team depends on — Book 5's clarity source of influence, made concrete.

AI removes the tedious part almost entirely. You still have to know what to document and check that it is right, but the blank page and the formatting and the structure are gone.

Someone who adopts this habit produces, over a year: runbooks that did not exist, a documented environment, onboarding material, and post-incident write-ups. None of that is glamorous. All of it is visible, and it accumulates into a reputation.

Reflection

- Which of the five areas would help you most this week?
- Are you asking for help making things sound good, or for help finding what is weak?
- What is undocumented at your workplace that you could fix in an afternoon?

Journal

Write about something you have been putting off because it involves writing. Most people in this field have several. That list is where these tools pay off first.

Exercises

1. Take one undocumented thing at work or in your lab and document it this week using the draft-check-own loop.
2. Use an adversarial prompt on something you are proud of. Notice how it feels and whether the criticism was fair.
3. For one study session, use the "what concept am I actually missing?" prompt after every wrong answer.

Project

Document one complete system — your lab, a process at work, a runbook — using these tools for drafting and your own judgment for everything else. Time it. Compare to what it would have taken you.

Books

- *Docs for Developers* — Bhatti et al.
- *The Pragmatic Programmer* — Hunt and Thomas

Checklist

- I use it for documentation regularly
- I ask for benign explanations before escalating
- I review and explain code rather than only generating it
- I use adversarial prompts on my own work
- I never paste unsanitized production data

Common Mistakes

- Using it to produce volume rather than to save time
- Pasting raw logs with internal identifiers
- Only asking for help making things sound good
- Letting it be your judgment on an alert

Key Takeaways

- Documentation is the highest-value, lowest-risk use
- "Give me three benign explanations" counteracts anchoring
- Adversarial prompts are the most useful and least used category
- Draft, check, own — and if you cannot own it, do not use it

Next Mission

Getting a lot better output by changing how you ask.

Prompting Like an Engineer

MISSION

To get consistently better results by being specific about context and constraints.

Objectives

- Use a reusable prompt structure
- Understand why context and limits matter more than phrasing
- Iterate rather than agonizing

Why It Matters

Most poor output is a poor question. People ask vague things, receive vague answers, and conclude the tool is not useful.

The fix is not clever phrasing or magic words. It is supplying the two things the tool cannot know: your situation, and your constraints.

Framework: The Prompt Structure

```
ROLE:      act as [specific role with relevant expertise]
CONTEXT:   here is my situation and constraints: [...]
INPUT:     here is the material: [...]
TASK:      do exactly this: [...]
LIMITS:    do not invent details. If unsure, say so.
FORMAT:    respond as [...]
```

The two lines that improve output most are **CONTEXT** and **LIMITS**, and almost nobody includes either.

CONTEXT is everything the tool cannot know: the size of your organization, the constraint you are working under, the audience, what you have already tried, why the obvious answer will not work here.

LIMITS is where you tell it not to fabricate. "Do not invent numbers I did not give you." "If you are unsure, say so rather than guessing." "Only use information I have provided." These instructions genuinely change the output, and they make the failures more visible when they occur.

Why Role Helps

"Act as a hiring manager reviewing two hundred applications" produces different output from "act as a career coach." Not because the tool becomes those people, but because it selects a different register, different priorities, and different assumptions about what matters.

Be specific. "Act as a security engineer" is weaker than "act as a security engineer at a 200-person company with no dedicated SOC, reviewing this proposal for feasibility."

Step-by-Step Guide: Iterating Well

1. **Start rough.** Do not agonize over the first prompt. You will learn more from seeing an imperfect output than from a perfect prompt written blind.
2. **Diagnose the gap.** Is it too generic? Missing context. Too long? Format. Wrong angle? Role. Fabricating? Limits.
3. **Add the specific missing thing** rather than rewriting from scratch.
4. **Ask it to critique its own output** — "what is weakest about this answer?" This works surprisingly well.
5. **Ask for alternatives** — "give me a version that takes the opposite approach."
6. **Save what works.** Build a personal library of prompts you reuse.

Templates: Prompts Worth Keeping

GAP FINDING

"Compare these two things: [A] and [B]. List only what is present in A and absent from B. Do not suggest fixes yet."

ADVERSARIAL REVIEW

"Act as a skeptical [role]. Read this and give me the three hardest questions you would ask, and the weakest part of the argument."

EXPLANATION AT LEVEL

"Explain [concept] as if I already understand [adjacent thing] but have never worked with [concept]. Use one concrete example."

SAFE CODE REVIEW

"Review this script. List every operation that is destructive, irreversible, or assumes a condition I have not verified. Do not rewrite it yet."

ASSUMED KNOWLEDGE

"Read this documentation. What steps have I assumed the reader already knows?"

DISCONFIRMING

"Give me three benign explanations for this pattern that I should rule out before treating it as malicious."

SELF-CRITIQUE

"What is weakest about the answer you just gave?"

Real Example

The same request, twice.

Weak: "Help me write a security policy."

Strong: "Act as a security manager at a 150-person professional services firm with no dedicated security team and a limited budget. I need an acceptable use policy for company laptops that our staff will actually follow. Keep it under one page, plain language, no jargon. Do not include controls we could not realistically enforce with the tooling described: [tooling]. If you are unsure whether something is enforceable in that environment, flag it rather than including it."

The second version produces something usable. The difference is not phrasing skill — it is five sentences of context and constraint that only you could supply.

Case Study: The Prompt Library

A habit worth building early.

People who use these tools well accumulate a personal set of ten to twenty prompts they reuse constantly. Not because the prompts are clever, but because refining a prompt once and reusing it beats improvising every time.

Keep them in a file. Refine them when they underperform. Share them with your team, which is both useful and a Book 5 visibility opportunity. Being the person who improved how the whole team works with these tools is exactly the kind of unassigned contribution that gets noticed.

Reflection

- When output disappoints you, do you diagnose or just try again?
- Are you supplying context, or expecting it to infer your situation?
- Do you ever ask it to critique itself?

Journal

Write down the three things you ask these tools for most often. Those are your first three library prompts.

Exercises

1. Take a prompt that gave you poor output and add only CONTEXT. Compare.
2. Add LIMITS to a prompt and see whether the fabrication rate changes.
3. Start your prompt library file with three entries.

AI Prompts

- "Here is a prompt I wrote: [prompt]. What context am I failing to provide that would improve your answer?"
- "What is weakest about the answer you just gave?"
- "Give me a version of that answer that takes the opposite approach, so I can compare."

Checklist

- I include context and limits by default
- I specify a role with real specificity
- I iterate rather than rewriting from scratch

- I ask for self-critique
- My prompt library exists

Common Mistakes

- Vague questions, then blaming the tool
- Expecting it to infer your constraints
- Agonizing over the first prompt instead of iterating
- Never asking for alternatives or critique

Key Takeaways

- Context and limits improve output more than phrasing
- Specific roles produce specific registers
- Iterate by adding the missing element, not rewriting
- Build a reusable prompt library

PART

THREE

Building With It

Automation: Where Fluency Becomes Capability

MISSION

To move you from consuming these tools to building with them.

Objectives

- Build one real automation
- Understand why this is the loudest differentiator available
- Know what to automate and what not to

Why It Matters

Using AI in a chat window is now ordinary. Everyone does it.

Building something that runs without you is not. Almost no entry-level or mid-level candidate has an automation project. It signals something different from every other item on a resume: that you think in systems rather than tasks.

If you take one differentiator from this entire library, it is this one.

Framework: What to Automate

Good candidates share three properties: **repetitive, rule-based, and low-consequence if it fails.**

Start with:

- An alert arrives summarize it post to a channel with the raw link attached
- A weekly report pull the data generate the summary email it
- A new ticket classify it route it notify the owner
- A backup completes verify it log the verification alert on failure
- A form submission validate it create a record confirm to the submitter
- A scheduled check run it only notify if something is wrong

That last pattern — **only notify on exception** — is worth internalizing. Automation that reports success constantly becomes noise, and Book 3's chapter on ignored alerts applies directly.

What Not to Automate

- Anything destructive without a human confirmation step
- Anything where being wrong is expensive and detection is slow
- Anything you do not understand well enough to debug when it breaks
- A process that is broken — automating a bad process produces a bad process faster

The rule that prevents the worst outcomes: automation should fail loudly and safely. If it breaks, you should know immediately, and the failure state should be "nothing happened" rather than "something wrong happened." Design for the failure before you design for the success.

Step-by-Step Guide: Your First Automation

Using n8n or an equivalent. Book 2's project list treats this as the tenth project for good reason.

1. **Pick something genuinely annoying** that you actually do repeatedly. Not a demo — a real irritation.
2. **Map it manually first.** Write down every step you currently perform, including the decisions. You will find steps you did not know you were doing.
3. **Identify the trigger.** What starts this? A schedule, an event, a message, a file appearing.
4. **Build the simplest version.** One trigger, one action. Get that working before adding anything.
5. **Add the AI step only where judgment or language is needed** — summarizing, classifying, drafting. Not where a rule would do; rules are faster, cheaper, and deterministic.
6. **Add error handling.** What happens when the API is down? When the input is malformed? When it runs twice?
7. **Test the failure modes** using Book 2's Break Matrix.
8. **Run it for two weeks** and watch what it gets wrong.
9. **Document it** using Book 2's README skeleton.

Step 5 is where people over-reach. AI is for the fuzzy step. Everything deterministic should stay deterministic.

Real Example

A small, genuinely useful automation.

The annoyance: every Monday, someone manually checks whether the weekend backups completed, opens three different interfaces, and posts a summary to a team channel. Twenty

minutes, every week, and occasionally forgotten.

The automation: a scheduled trigger every Monday morning pulls backup status from each source, an AI step summarizes anomalies in plain language. The result posts to the channel — but only flags anything that failed or looks unusual. Success is a single line.

What it took: an afternoon to build, another week of fixing the cases it got wrong.

Why it is a strong portfolio piece: it is real, it is small enough to explain in two minutes, it demonstrates integration across systems, it uses AI for the appropriate step only, and it has a "what broke" story — because the first version posted a wall of text every week and nobody read it, which is the alert-fatigue lesson learned firsthand.

Case Study: Why This Signals Differently

Consider two candidates for the same mid-level role.

The first lists tools: SIEM, ticketing, cloud platform, scripting language. All true, all common.

The second describes an automation they built that connects three systems and removed a recurring manual task. In discussing it, they naturally cover: how the systems integrate, error handling, why they chose AI for one step and rules for the others, what broke, and how they knew it was working.

The second conversation covers systems thinking, judgment about tool selection, failure design, and operational thinking — in the course of describing one project, without being asked about any of those things directly.

That is why it signals differently. It is not that automation is impressive. It is that describing one honestly demonstrates half a dozen competencies at once.

Reflection

- What do you do repeatedly that a machine could do?
- Have you ever built something that runs without you?
- What would you automate if you had a free afternoon?

Journal

List everything you did last week that you also did the week before. That list is your automation backlog.

Exercises

1. Map one repetitive process manually, every step, including the decisions.
2. Build the simplest possible version of it. One trigger, one action.
3. Break it deliberately and see how it fails.

Project

Build, break, document, and explain one automation, following Book 2's four-step loop. Then write it up. This is the single strongest portfolio addition available from this book.

AI Prompts

- "Here is a manual process: [steps]. Which steps genuinely need judgment and which are rules? Do not suggest automating the rule-based steps with AI."
- "I built this automation: [description]. What are its failure modes, and how should it behave when each occurs?"
- "Review this workflow. Where could it run twice, produce duplicates, or fail silently?"

Books

- *The Phoenix Project* — Gene Kim
- *Site Reliability Engineering* — Google, on automation philosophy and toil reduction

Resources

- **n8n** — self-hostable, visual, and it runs well on the Book 2 mini-PC
- Your own repetitive weekly tasks, which are a better source of ideas than any tutorial

Checklist

- I have built at least one real automation
- It fails loudly and safely
- AI is used only for the judgment or language step
- I have broken it deliberately and documented what happened
- It is written up in my portfolio

Common Mistakes

- Automating a broken process
- Using AI for deterministic steps

- No error handling
- Reporting success constantly, creating noise
- Building a demo rather than solving a real annoyance

Key Takeaways

- Repetitive, rule-based, low-consequence — in that order
- Fail loudly and safely; design the failure first
- AI for the fuzzy step only
- One honest automation demonstrates six competencies at once

Next Mission

The thing that determines whether all of this makes you better or worse.

Staying Sharp While Using It

MISSION

To make sure these tools raise your ceiling rather than lower your floor.

Objectives

- Recognize the dependency pattern early
- Keep learning fundamentals while using tools that let you skip them
- Know which skills to protect deliberately

Why It Matters

This is the chapter I most want you to take seriously. It is the one that is easiest to nod at and ignore.

These tools make it possible to produce work you do not understand. That has always been somewhat true — copying from forums, following tutorials — but the scale is different now, and the output is more polished, which makes the gap harder to see from the outside and from the inside.

The failure is slow. You do not notice you are becoming dependent, because your output is fine and possibly increasing.

Professional Explanation

The Dependency Pattern

It looks like this:

1. You use it for something you could have done, to save time. Correct use.
2. You use it for something you could have figured out, to save effort. Still fine, occasionally.
3. You use it for something you could not have figured out, and you do not go back and learn it.
4. Step 3 becomes normal.
5. Something happens that the tool cannot help with, and you discover the floor has moved.

Steps 1 and 2 are healthy. Step 3 is fine *if you go back*. The failure is step 4, and it arrives without announcement.

The Test

After using it for something, ask: **could I do this again without it?**

If no, and it matters, go and learn it. Not immediately, necessarily — but put it on a list and come back.

This is not purism. There are plenty of things you should never bother learning deeply, and offloading them is exactly right. The question is whether *this specific thing* is core to what you do. Formatting a document: offload freely. Understanding why the authentication failed: do not.

Which Skills to Protect Deliberately

- **Reading and understanding code**, even if you generate more than you write
- **Debugging from symptoms**, which Book 2's break sessions build and which no tool replaces
- **The fundamentals of your lane** — networking, identity, how systems fail
- **Writing in your own voice**, because you will be asked to think out loud
- **Judgment about what matters**, which is the whole of Book 3

Notice these are the things that transfer between jobs and outlast tools. That is not a coincidence — they are worth protecting for the same reason they are valuable.

The uncomfortable version of this chapter: if the tool disappeared tomorrow, would you still be able to do your job? If the honest answer is no, you have not gained leverage — you have taken on a dependency, and dependencies are only safe while the thing you depend on stays available and stays cheap.

Framework: The Deliberate Practice Reserve

Keep some work manual on purpose.

- **Once a week, solve something without help.** Slower. It is the reps.
- **When learning something new, build it yourself first**, then ask for a review. The reverse order teaches you very little.
- **Explain what you produced to someone else.** Book 2's step four. If you cannot, you did not understand it.
- **Keep breaking things** — Book 2's break sessions are not automatable and they build the intuition nothing else does.

This is the same logic as anyone maintaining a skill they could offload. Not because offloading is wrong, but because the skill is worth having when it matters.

Real Example

Someone uses AI heavily for scripting over eighteen months. Their output is good, their scripts work, their team is happy.

Then a production incident occurs involving a script behaving unexpectedly under a condition nobody anticipated. They cannot debug it quickly, because they never developed the mental model — they had always generated, tested, and shipped without deeply reading.

They get there eventually, with help, several hours later than someone with the fundamentals would have.

Nobody blames them and nothing catastrophic happens. But they know, and the experience is genuinely unpleasant. It is entirely avoidable by reading what you deploy.

Case Study: What Actually Stays Valuable

Worth thinking about when deciding where to invest.

The tasks most exposed to automation are the ones that are well-specified, repetitive, and text-shaped: routine documentation, first-line summarization, boilerplate code, basic configuration.

The things that stay valuable are the ones requiring accountability, environmental context, judgment under uncertainty, and the ability to be responsible for an outcome. Book 4's argument about GRC is a specific instance of this general pattern.

Which means the strategic move is not to avoid these tools. It is to use them for the exposed work and spend the time they save on the lasting work. The judgment, the relationships, the business context, the accountability.

That is the whole strategy. It is available to anyone.

Reflection

- If the tool disappeared tomorrow, could you do your job?
- What have you stopped being able to do without help?
- Which skills in your lane are worth protecting deliberately?

Journal

Write about something you used to be able to do without help and now reach for a tool for. Then decide whether that matters. Sometimes it genuinely does not, and being honest about which is which is the skill.

Exercises

1. Pick one thing this week and do it entirely without help. Notice how it feels.
2. Take something you generated and read it fully until you could rewrite it.
3. List the five skills in your lane most worth protecting, and check whether you are still practising them.

AI Prompts

- "I want to understand [thing] rather than have you do it. Ask me questions that lead me to work it out myself, and do not give me the answer."
- "Here is code I generated. Quiz me on what each section does, one question at a time, and tell me where my understanding is wrong."
- "What fundamentals underpin [task] that I should understand even though you can do it for me?"

Books

- *So Good They Can't Ignore You* — Cal Newport
- *Deep Work* — Cal Newport. On protecting the capacity for the work that cannot be offloaded.
- *Range* — David Epstein

Checklist

- I ask "could I do this again without it?" for anything that matters
- I keep some work manual deliberately
- I read code before deploying it
- I still break things myself
- I know which skills in my lane I am protecting

Common Mistakes

- Confusing increased output with increased capability
- Letting step 3 become step 4
- Protecting nothing, on the theory that the tools will always be there
- Deep-learning things that genuinely should be offloaded

Key Takeaways

- The dependency pattern is slow and unannounced
- Ask: could I do this again without it?
- Protect debugging, fundamentals, judgment, and your own voice
- Use the tools for exposed work; spend the saved time on lasting work

Next Mission

Book 7: being findable by the people who would hire you.

Appendix: Prompts, Patterns, and the Gate

REFERENCE

Everything from this book in one place.

The Working Mental Model

A fast, confident junior analyst who has never seen your environment and will not tell you when they are guessing.

The Verification Gate

1. Would I be embarrassed if this were wrong?
-> verify every factual claim independently
2. Does it contain specifics I didn't provide?
-> numbers, names, versions, commands, citations
3. Would I sign my name to it?
-> because you are about to

The Personal Rule Set

NEVER PASTE:

customer data credentials or keys
personal data proprietary code
incident details internal hostnames / IPs
unreleased business information
anything a contract or regulation covers

ALWAYS:

read generated code fully before running
test in a lab before production
verify every specific I did not provide
rewrite anything with my name on it

IF UNSURE:

assume it's sensitive and don't paste it

The Prompt Structure

```
ROLE:      act as [specific role with relevant expertise]
CONTEXT:   my situation and constraints
INPUT:     the material
TASK:      do exactly this
LIMITS:    do not invent details; say so if unsure
FORMAT:    respond as [...]
```

CONTEXT and LIMITS improve output most. Almost nobody includes either.

The Prompt Library

GAP FINDING

```
"List only what is present in A and absent from B.
Do not suggest fixes yet."
```

ADVERSARIAL REVIEW

```
"Act as a skeptical [role]. Three hardest questions,
and the weakest part of the argument."
```

EXPLANATION AT LEVEL

```
"Explain [X] as if I understand [adjacent thing] but
have never worked with [X]. One concrete example."
```

SAFE CODE REVIEW

```
"List every operation that is destructive, irreversible,
or assumes a condition I have not verified."
```

ASSUMED KNOWLEDGE

```
"What steps have I assumed the reader already knows?"
```

DISCONFIRMING

```
"Three benign explanations I should rule out first."
```

SELF-CRITIQUE

```
"What is weakest about the answer you just gave?"
```

LEARNING MODE

```
"Ask me questions that lead me to work it out. Do not
give me the answer."
```

Draft, Check, Own

```
DRAFT    let it produce; iterate rather than agonize
CHECK    apply the gate; verify unprovided specifics
OWN      rewrite in your voice; be able to defend it
```

If you cannot do the third, do not use the output.

What to Automate

```
GOOD:    repetitive, rule-based, low-consequence if it fails
BAD:      destructive without confirmation
          expensive to get wrong and slow to detect
          something you can't debug
          a process that is already broken

DESIGN:   fail loudly and safely
          notify on exception, not on success
          AI for the fuzzy step only
```

The Dependency Test

```
Could I do this again without it?

If no, and it matters – go and learn it.
```

A Closing Note

I came up without any of this. A donated computer with no mouse, a language I did not speak, and nobody to ask. Every answer took days that would now take seconds.

I am not nostalgic about that. It was slow and it was frequently miserable, and if these tools had existed I would have used them constantly.

But I want to be honest about what that period gave me, because it is the thing this book is trying to protect.

It gave me the habit of sitting with something broken until it made sense. That habit is the whole job. It is the one thing these tools can quietly take from you if you let them.

So use them. Use them aggressively, for the documentation and the drafts and the explanations and the things you would otherwise put off. Build something that runs without you. Get faster.

And once a week, sit with something broken until it makes sense, without help. That is not nostalgia. That is maintenance.

Draft with it. Check everything. Own the output. Keep breaking things yourself.

Connect

[linkedin.com/in/efrenlsalazar](https://www.linkedin.com/in/efrenlsalazar) | wakeuypodcast.com

Questions for the author. Two gaps: (1) which tools you personally use and for what — readers will ask, and a specific answer from you is more useful than my deliberate vagueness; (2) **a concrete example of AI producing something confidently wrong in your own work.** Chapter 1's real example is constructed. A real one from you would do more for reader caution than any warning I can write. It is the single most valuable addition to this book.

Where the Industry Is Headed

RSAC's expo floor is where AI and security collide first — a front-row seat to where the tools in this book are actually going.



RSAC 2026 Conference, with colleagues



RSAC 2026 Conference, on the expo floor

Sana Talwar — Trial, Error, and AI Security

Sana Talwar works in product and AI security, focused on prompt-injection research and making the OWASP LLM Top 10 understandable to people newer to the field. Met at RSA Conference, her path there wasn't a straight line — it was six internships and a lot of honest "no thanks."

Six Internships, Six Honest Answers

Starting at 16, she interned every summer through high school and college: software security, network/hardware security (racking servers in cold server rooms), and embedded security on FPGAs. Every one of them ended the same way — a clear "did it, did not like it" — until product and AI security stuck. Her point: a 3-month internship tells you more about a career path than months of reading about it ever will.

On AI: Fundamentals Still Win

Her take on AI's effect on the field: it won't erase jobs, but it will change how they're done. At a 48-hour hackathon, vibe-coding with an AI tool, she got stuck on a CSS problem the AI couldn't resolve — her software-engineer husband fixed it instantly because he knew the exact property name to set. Her conclusion: AI makes execution faster, but understanding what's actually happening underneath it is what makes you good at directing it.

GUEST SPOTLIGHT

Sana's Advice, In Practice

Try it before you commit to it.

Internships exist to give you a fast, low-stakes 'no' before you spend years finding out the hard way.

Learn fundamentals even if AI writes the code.

Knowing how systems and networks actually work is what lets you direct AI well, not just prompt it and hope.

Teach what confuses beginners, even if it's obvious to you.

Her own research is paired with a deliberate goal: build the bridge that didn't exist for her when she started.

Lurking on LinkedIn still counts, but activity compounds.

She had a profile for years before actually using it — real engagement is what surfaces CFPs and community.

REFLECTION

What career path have you assumed you'd like without ever actually trying it?
What's the smallest, lowest-stakes way you could test it this month?

Mo Balakrishnan — From Physicist to CTO

Mo Balakrishnan is CTO at Schools Insurance Authority and a director at ISACA Sacramento. He didn't start in tech at all — he started as a physicist, and moved into IT almost by accident after relocating to a new city.

Get In Front Of Experiences

He started at the very bottom — desktop support, swapping RAM in modular IBM ThinkPads. A move to a London branch office forced him to learn networking, firewalls, and telephony fast, simply because the work demanded it. His rule from that stretch: "Never shy away from an experience. If you can learn to program a switch, don't let somebody else do it. Volunteer." He doesn't call his setbacks failures — he calls them experience: "Whether it's good or bad, it's an experience. You learn something from it."

What AI Actually Is (and Isn't)

His clearest explanation: an LLM has no memory of its own. Every message you send resends your entire chat history for the model to reread from scratch — which is why long chats slow down and start to lose the thread. He calls this 'tokenomics': token pricing is starting to work like an economy, and it will shape how fast AI actually gets adopted. His sharpest warning: handing out AI licenses without any control plane, logging, or policy in place is "a house of cards waiting to fall down."

Mo's Advice, In Practice

Chase the journey, not just the goal.

"If I said, well, I just wanna be a CISO, that's anticlimactic. That would be the end of the journey." Set the next goal before you hit the current one.

Certs and degrees aren't the whole path.

There's no one right route — the paths are infinite. What matters is your own clarity about where you're headed.

Think security-first, whatever your role.

Before any technology gets deployed, ask what the appropriate controls are first — the same question applies to AI as it does to any other system.

Fact-check anything an AI tells you that you don't already know.

A hallucination presented confidently can damage a reputation fast — his rule is simple: verify before you rely on it.

REFLECTION

Are you chasing a fixed goal, or are you building the habit of setting the next one before you arrive? What's one experience you've been avoiding volunteering for?

GUEST SPOTLIGHT

Jorge Avila — AI as the Great Equalizer

Jorge Avila is founder of Paleo Tech Group, a small AI startup built around one mission: teaching AI and cybersecurity skills to underserved Latino and Black communities that the digital economy keeps bypassing.

His Father's Hustle, One Generation Later

His parents were migrant farm workers from Mexico, moving between Utah, Idaho, and California. His dad worked lawn care, maintenance, and janitorial jobs while always trying to build something of his own on the side. That example is the direct reason Jorge left a stable government project-director role at the California Department of Technology to start his own company instead.

AI Has the Knowledge — You Just Have to Ask

His concrete advice for anyone starting a business: tell AI what you're good at and what you like, let it suggest a path, then ask for the specific next step. He treats it as an on-demand advisor, not a search engine. He cites a real, specific shift he's heard from employers: work that used to take 40 hours is now expected to produce roughly 140 hours of output, in the same 40-hour week.

A Knife, Not a Toy

His security framing is simple and memorable: a sharp knife cuts meat well, but it can cut your finger off too. Any small business adopting AI tools should pair that adoption with an actual security review — not skip it because there's no in-house IT team to ask.

Watch the full episode: youtu.be/ycR7h1BR5v4

Jorge's Advice, In Practice

Learn one AI tool a week for three months.

A concrete, bounded cadence beats a vague goal to "learn AI" — pick a schedule and stick to it.

Security has to scale down to small business, not just enterprise.

Paleo Tech Group's own training splits AI basics from tool-specific sessions (like QuickBooks + AI) specifically so smaller, resource-limited businesses aren't left out.

No single person or company can secure data alone.

The RSA 2026 theme that stuck with him: "your data is on my data somehow" — safeguarding it is a community effort, not a one-person job.

Being told no repeatedly doesn't have to slow you down.

He ties this directly to a shared immigrant, first-gen experience: the volume of past rejection can become the reason you stop flinching at it, not a reason to quit.

REFLECTION

What's one AI tool you could commit to actually learning this week — not just trying once, but using enough times to know if it fits your work?

BE OUR NEXT GUEST

Your Story Belongs on Wake Up X

If AI for IT & Cybersecurity gave you something worth using, you already have something worth sharing. The Wake Up X podcast exists to put real, unpolished career and business stories in front of people who need to hear them — and every guest so far started exactly where you are now: reading, building, and figuring it out in public.

You might be a fit if you:

Broke into tech, cybersecurity, or business without the traditional path

Built something — a lab, a business, a portfolio, a community — and can talk about how

Have a failure or turning point you're willing to tell straight, not polished

Are mentoring, leading, or building something worth other people knowing about

HOW TO APPLY

Share your story at wakeupxpodcast.com/guest, or connect directly on LinkedIn at linkedin.com/in/efrenlsalazar and mention this book.

Being a guest is also visibility — the same principle Book 7 makes the case for.

Acknowledgments

THE PEOPLE WHO MADE THIS POSSIBLE

None of Wake Up X happened alone, including AI for IT and Cybersecurity. Every book in this library, the podcast, and the free library exist because specific people showed up for a kid nobody expected much from. This is the record of who they are.

THE EARLY YEARS

Miss England, the 6th-grade teacher who stayed after class, unpaid, to teach the fundamentals. Sister Cora and Lewis Bair of Computers for Kids, who handed a teenager a key and a broken machine and let the work speak. Jose Luis Bravo, Angel Martinez, Isaac Murrilo, and Alex De Haro — the friends and cousins who translated the world, gave hope, drove him where he needed to go, and carried him before there was anything to carry him toward.

THE BUSINESS YEARS

Rojelio Salazar, his cousin and first mentor, RNS Computer Repairs — the reason it is called Wake Up X. Kyle Erni, his first business partner. Erick and Pancho Arauza, co-founders of The Cellphone Fix. Danny Maya, of Maya Traditional Mexican Restaurant, Taqueria Maya, and Salt and Water, who pushed him in front of a camera for the first time.

THE CAREER YEARS

Igor, who gave him the vocabulary for work he already knew how to do. Nate, who he mentored into a full-time role alongside his own. Ken Kerrick, opponent turned friend in the ISACA Sacramento election. Alvin Jue and David De los Reyes, who believed in the middle of it, not just at the finish.

THE PODCAST

Every guest who trusted an unproven show with their story, including Gustavo Mastroianni, Cathy Stephenson, Greg McCord, Aaron Portnoy, Sana Talwar, Jorge Avila, Anil Reddy, Devin Prasad, Larry Switzer, Louis Kemp, Mohana Balakrishnan, Eddie Lopez, and Michael Fierros — full episodes at wakeupxpodcast.com.

COMMUNITY PARTNERS

ISACA Sacramento and ISC2 Sacramento, The Cellphone Fix, A.N.U.L Technologies, AISOC, AspireByte, and PodGrabber — the organizations building alongside this movement.

AND

His parents, who arrived with nothing, worked since childhood, and gave away whatever was left over. Nothing about the first version of this story predicted the version you are holding now.

Continue the Journey

EVERY BOOK LEADS SOMEWHERE ELSE

AI for IT and Cybersecurity is one part of a 12-book system. If it helped, the next step is one page away — take whichever layer matches where you are standing.

THE LIBRARY

Layer 1 — The Story

I Just Have To Wake Up — why any of this exists.

Layer 2 — The Mindset

Comfort Is Expensive — the real cost of playing it safe.

Layer 3 — The Blueprint

DECODE: The Wake Up X Playbook — zero experience to executive thinking. Book 0.

Layer 4 — The Playbooks

1 Break In · 2 Build It · 3 Think Like a Cybersecurity Professional · 4 Business First · 5 Lead Without a Title · 6 AI for IT & Cybersecurity · 7 Build Your Brand · 8 The Consultant Mindset · 9 Level Up

FIND US EVERYWHERE

Podcast & free library: wakeupxpodcast.com · Spotify: open.spotify.com/show/28GZnRBrIV2efHOhZdiftt · Apple Podcasts: podcasts.apple.com/us/podcast/id1792821643 · YouTube: youtube.com/@Efren.wakeupx · LinkedIn: linkedin.com/in/efrenlsalazar · Substack: substack.com/@wakeupxefren

IN GOOD COMPANY

ISACA Sacramento, ISC2 Sacramento, The Cellphone Fix, A.N.U.L Technologies, AISOC, AspireByte, and PodGrabber — the partners and collaborators building alongside the movement. See the full list at wakeupxpodcast.com/community.html.

Get DECODE free, apply to be a guest, or read the story behind the movement — all of it starts at wakeupxpodcast.com.

WAKE UP X

SOLUTIONS



Efren L. Salazar

FOUNDER · HOST · CYBERSECURITY PROFESSIONAL · EDUCATOR · COMMUNITY BUILDER

STAY CONNECTED

Podcast & Blueprint Library:	wakeupxpodcast.com
LinkedIn:	linkedin.com/in/efrenlsalazar
YouTube:	youtube.com/@Efren.wakeupx
Be a Guest:	wakeupxpodcast.com/guest

THE WAKE UP X PUBLISHING SYSTEM

BOOK 7

Build Your Brand

LinkedIn, content, and being findable

Visibility gets you the opportunity you're not qualified for yet

THE
WAKEUPX
SOLUTIONS

EFREN L. SALAZAR

Founder & Host, Wake Up X | IT Operations Manager & Solution Engineer
Government & Regulatory Partner, ISACA Sacramento | CompTIA Security+

wakeupxpodcast.com

Educational content — not professional, legal, financial, or career-placement advice.
Full disclaimer in DECODE, the foundation of this library.

Before You Begin

HOW TO USE THIS BOOK

This book is about being the name that surfaces when someone needs what you do — and about why waiting until you feel qualified means missing the opportunities that visibility would have brought you.

Where this book sits

Two books come first — the memoir *I Just Have To Wake Up* and the mindset book *Comfort Is Expensive* — before this ten-book library. Each one is short on purpose. Pick up the one that matches the problem in front of you right now.

- n **Book 0 — DECODE.** The foundation. The map of the whole path.
- n **Book 1 — Break In.** Landing your first role in IT or cybersecurity.
- n **Book 2 — Build It.** Home labs and a portfolio that prove you can do the work.
- n **Book 3 — Think Like a Cybersecurity Professional.** Detection, triage, and incident response thinking.
- n **Book 4 — Business First.** Finance, sales, GRC, and why the business hired you.
- n **Book 5 — Lead Without a Title.** Influence, EQ, and mentoring before anyone promotes you.
- n **Book 6 — AI for IT & Cybersecurity.** Using AI as an actual working tool, not a novelty.
- n **Book 7 — Build Your Brand.** LinkedIn, content, and being findable. **you are here**
- n **Book 8 — The Consultant Mindset.** Discovery, solution selling, and stakeholder management.
- n **Book 9 — Level Up.** Long-term growth, burnout, legacy, and mentoring others.

Book 0 is the foundation. *DECODE: The Wake Up X Playbook* holds the six-level path, the Curiosity Compass, and the core frameworks. Nothing in any other book contradicts it. If a term here is unfamiliar, DECODE defines it.

What is in this book

Part One — The Case for Being Visible

1. The Opportunity You're Not Qualified For Yet
2. Findable, Credible, Known

Part Two — Actually Publishing

3. What to Publish
4. A Practice You Won't Quit
5. What Never to Post

Part Three — Going Further

6. Long-Form: Podcasts, Newsletters, and Speaking
7. Appendix: Profiles, Skeletons, and the Cadence

How to actually use it

This book depends on Book 2. Without built work, there is nothing to be visible about, and visibility without substance is the failure mode everybody can smell. If you have not built anything yet, go and build one thing first.

If you have, this book is about making sure the right person finds it.

A warning that matters more than the tactics. Chapter 5 covers what never to post. Read it before you publish anything, not after. There is exactly one way personal branding reliably damages a technology career. It is talking about your employer's environment.

I am writing this the way I would say it to you sitting across a table — not lecturing a room. Take what works, remix the rest into your own version, and leave what does not fit.

PART

ONE

The Case for Being Visible

The Opportunity You're Not Qualified For Yet

MISSION

To convince you to become visible before you feel ready, because that is the only time it matters.

Objectives

- Understand what visibility actually buys you
- Get past the belief that you have nothing to say yet
- Start before you are qualified

Why It Matters

Here is the argument. It is the whole book in one paragraph.

Visibility can get you an interview for something you are not qualified for yet. Someone sees your name, remembers a post, thinks of you when a role opens, and reaches out. That happens constantly to people who are visible and never to people who are not.

And if you pass on that opportunity — because you did not feel ready, because the timing was wrong, because you were waiting for one more certification — **it does not come back the same way.** That specific role, that specific person thinking of you at that specific moment, does not recur.

Something else might come eventually. Not that.

Which means building your name matters *before* you are ready, not after. By the time you feel qualified, you have already missed the opportunities that visibility would have brought you.

You are not building an audience. You are making sure that when someone needs what you do, your name is the one that surfaces.

Professional Explanation

What Visibility Actually Buys

Be specific about this, because "personal brand" is a vague phrase that makes technical people suspicious, and reasonably so.

Inbound opportunity. Roles that come to you rather than roles you apply for. These are by design better: less competition, more leverage, and the conversation starts from interest rather than from screening.

A shorter path through Book 1's gates. Someone who has read your work arrives at Gate 3 already. The risk gate is a lot lower for a candidate whose thinking the manager has already seen.

Credibility that precedes you. In a meeting, in a vendor conversation, in an interview. People who have encountered your work treat you differently from a stranger with the same résumé.

A network that grows without effort. People contact you. That is a completely different position from contacting people, and it builds on itself. It does not replace the T.R.A.C.K. sequence from Book 1 — it means more of your conversations start warm.

Options. Book 9's argument about what you are building depends on having options. Visibility is one of the few reliable ways to generate them.

What It Does Not Buy

- It does not substitute for capability. It amplifies whatever is actually there.
- It does not work quickly. Layer three in the next chapter takes a year or more.
- It does not need a large audience. The mechanism is one right reader, not a thousand.

That last point is the one that changes how you should approach this. You are not trying to grow a following. You are trying to be findable and credible to the twenty people who might one day need what you do.

Framework: What You Actually Have to Say

The most common objection is: *I'm not an expert, I have nothing to contribute.*

This is wrong for a specific and useful reason: **the person one step behind you needs you more than the expert does.**

An expert explaining a concept has forgotten what it was like not to understand it. They skip the part that was confusing because it stopped being confusing fifteen years ago. You have not forgotten.

You just learned it last month, you remember exactly which part was unclear, and you can explain it to the next person much better than someone who learned it long ago.

That is your material. Not thought leadership. Not predictions about the industry. **What you learned this week and what confused you about it.**

The one-step-ahead principle. You do not need to be the most qualified person writing about a topic. You need to be one step ahead of the person reading, and honest about where you are. That is a position you already occupy for somebody.

Where the Fear Comes From

People fear being wrong in public, and that fear is reasonable. But the actual risk profile is inverted from how it feels.

Being wrong in public and correcting it gracefully builds credibility. Everyone in this field has been wrong; the ones who acknowledge it are trusted more, not less. Being invisible builds nothing at all.

And the audience you are afraid of judging you is, at the start, about nobody. Your first twenty posts will be read by very few people. That is not a discouragement — it is the safest possible environment to get good in.

Real Example

Two people with comparable skills and comparable labs.

The first applies to roles through portals. She is good, and she gets interviews at roughly the rate that applications produce interviews, which is low.

The second writes one short post a week about what he is learning. The VLAN design he got wrong, the DNS failure that surprised him, the tuning pass that cut his lab's alert noise in half. Nothing polished. No large audience.

After eight months, a security manager who had been reading those posts messages him about a role that was never publicly advertised. He was not more qualified. He was known.

The important detail: he could not have predicted which post, or which reader, or when. That unpredictability is why the practice has to be consistent rather than strategic.

Case Study: The Opportunity That Did Not Return

The pattern that makes this urgent rather than merely useful.

Someone visible gets approached about a role that is a stretch — genuinely a stretch, a level above where they are. They decline: not ready, bad timing, one more certification first.

Eighteen months later they are ready. But the role was filled, the person who thought of them has moved on, and the specific circumstance that produced the opportunity does not

reassemble itself.

Something else comes eventually — visibility keeps producing — but not that. And the version of their career where they said yes to the stretch is now unavailable.

I am not telling you to accept everything. I am telling you that the calculation people run — "I'll be ready later, and the opportunity will still be there" — is wrong about the second half.

Reflection

- What are you waiting to be ready for?
- If someone searched your name right now, what would they find?
- Who is one step behind you, and what do they need to know?

Journal

Write about the last thing you learned that confused you at first. That is your first post, and you have just written most of it.

Exercises

1. Search your own name. Note what a hiring manager would conclude in ten seconds.
2. Write down three things you learned in the last month. Those are three posts.
3. Identify one person one step behind you. What would help them?

Common Mistakes

- Waiting until you are an expert, which is never
- Believing you need an audience rather than the right reader
- Posting achievements instead of learning
- Assuming being wrong in public is fatal

AI Prompts

- "I am [level] in [field]. Give me ten specific things I likely know that someone one year behind me would find useful."
- "Here is something I learned recently: [description]. Help me identify what was genuinely confusing about it, since that is the useful part."
- "Read this draft as someone brand new to the field. What did I assume you already knew?"

Books

- *So Good They Can't Ignore You* — Cal Newport
- *Show Your Work!* — Austin Kleon. Short. It is essentially this chapter as a whole book.
- *Give and Take* — Adam Grant

Podcast Episodes

Wake Up X episodes where a guest describes an opportunity that came to them. Listen for whether they were findable beforehand — they almost always were, and they rarely mention it as the cause.

Checklist

- I have searched my own name and know what surfaces
- I have stopped waiting to feel qualified
- I know who is one step behind me
- I have three post ideas written down

Key Takeaways

- Visibility gets you opportunities you are not qualified for yet, and they do not repeat
- The person one step behind you is your audience
- You need the right reader, not an audience
- Being wrong in public and correcting it builds credibility

Next Mission

The three layers, in the order they actually have to be built.

Questions for you. What would you have to believe about yourself to post something this week? And is that belief true?

Findable, Credible, Known

MISSION

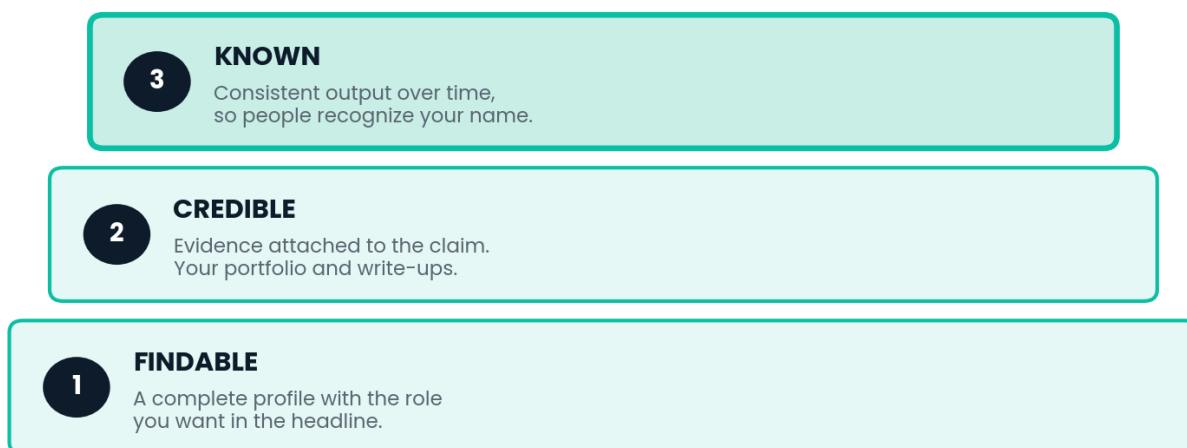
To build visibility in the order that works, instead of the order people attempt.

Objectives

- Complete layer one this week
- Understand why most content efforts fail at the foundation
- Know what each layer actually needs

Framework: The Three Layers

THE THREE LAYERS



Most people attempt layer 3 without doing 1 and 2, then conclude content doesn't work.

The three layers. Most people attempt layer three without doing one and two, then conclude content does not work.

Layer 1 — Findable. A complete profile with the role you want in the headline, your projects linked, and a real photo. This alone puts you ahead of most people, and it takes an afternoon.

Layer 2 — Credible. Evidence attached to the claim. Your portfolio, your write-ups, your documented projects. Book 2 is this layer — if you have done that work, you have already built it and may only need to link it.

Layer 3 — Known. Consistent output over time, so that people recognize your name. This is the slowest layer and the one with the largest payoff.

Most people attempt layer three without doing one and two, then conclude that content does not work. It does work — but a post that makes someone curious leads to a profile that tells them nothing, and the interest evaporates there.

Professional Explanation

Layer 1: Findable, In Detail

LinkedIn is the right answer for IT and cybersecurity careers. Not because it is enjoyable — it is frequently not — but because it is where recruiters and hiring managers already search. Being absent there is close to not existing professionally.

What actually matters, in order:

The headline. This is the single highest-value field, and most people waste it on their current job title. Put the role you want, plus what you actually do.

Weak: IT Support Specialist at [Company]

Better: IT Support | Building toward SOC analyst | Home lab: AD, Wazuh, VLAN segmentation

The second version tells a recruiter what to search for you as. It is honest about the trajectory rather than pretending to already be there.

A real photo. Not a logo, not an avatar, not a group photo cropped badly. Faces get engagement and they get trust. It does not need to be professionally shot.

The About section. Three short paragraphs: what you do, what you are building toward, and something human. Written in first person. Most About sections are written in third person and read like an obituary.

Featured links. Your portfolio, your GitHub, your best write-up. This is where layer two attaches to layer one.

Skills. The honest, searchable list — this is a search field, so the words matter. Same discipline as Book 1's resume chapter: production versus lab, clearly separated.

Experience with evidence bullets. The same bullets from your Book 1 bullet bank. Not duties.

Layer 2: Credible

This is Book 2's portfolio, linked and visible. If you have three documented projects with diagrams and "what broke" sections, layer two is done.

If you do not, stop here and go build one thing. Layer three built on nothing is the failure mode this whole book is trying to avoid. It is visible from a long way off.

Layer 3: Known

Chapters 3 and 4 cover this in full. It takes a year minimum. It is worth it. It is the layer most people quit on because the first three months produce very little visible return.

Step-by-Step Guide: Layer One in an Afternoon

1. **Rewrite your headline.** Target role, plus what you actually do. Ten minutes.
2. **Replace your photo** if it is not a clear picture of your face.
3. **Rewrite your About** in first person, three short paragraphs.
4. **Add your portfolio to Featured** and to your contact info.
5. **Fix the skills list** — honest, searchable, ordered by relevance to the role you want.
6. **Convert your experience to evidence bullets** from the bullet bank.
7. **Turn on the "open to work" signal** if appropriate — recruiter-only visibility if you are currently employed.
8. **Connect with fifty people in your target field.** Not spam — people whose work you have actually encountered.

That is a single afternoon and it puts you ahead of a large majority of people in this field, because most profiles are abandoned CVs.

Real Example

Two profiles for the same person.

Before:

Technical Support Specialist at [Company] About: "Experienced IT professional with a passion for technology and a proven track record of delivering results in fast-paced environments."

After:

IT Support SOC Analyst | Home lab: Active Directory, Wazuh SIEM, VLAN segmentation | Security+ About: "I fix things for a living and I'm moving toward security. Most of what I know about detection I learned by building a SIEM at home and spending two weeks failing to tune out the noise. I write up what I build — links below. Currently working toward a first analyst role on a team that documents its work."

The second is specific, honest about the transition, searchable, and it sounds like a person. It also gives a recruiter three search terms they could have used to find it.

Case Study: The Profile Nobody Could Find

A common and fixable failure.

Someone builds genuine skill and a real portfolio. They apply widely and hear little. They conclude the market is difficult.

Their LinkedIn headline says their current job title at their current employer. Nothing in their profile contains the words a recruiter would search for the role they want. Their portfolio is not linked anywhere. Their skills list has fourteen entries in no particular order, half of them irrelevant.

They are, functionally, invisible to search. Not because they lack anything — because the index does not contain them.

This is Book 1's Gate 1 problem, applied to your profile rather than your résumé. It is solved the same way: use the words the searcher will use.

Templates: The Headline Formula

[Current or target role] | [Trajectory, if changing] |
[Your strongest specific evidence] | [Credential if it
differentiates]

Examples:

Help Desk / Systems Admin | Home Lab: Windows Server,
AD, Group Policy | CompTIA A+

SOC Analyst | Detection tuning & incident response |
Wazuh, Splunk | Security+

IT Operations Manager | Cybersecurity & GRC |
ISACA Sacramento Partner | Security+

Templates: The About Section

Paragraph 1 —What I do now, plainly. One or two lines.

Paragraph 2 —What I'm building toward and what I've
actually done about it. The specific evidence.

Paragraph 3 —Something human. Why this field, or what
I'm curious about, or who I'm trying to help.

Links.

First person. No third-person narration. No "results-driven professional."

Reflection

- n Does your headline contain the words someone would search to find you?
- n Is your portfolio linked from your profile?
- n Would your About section distinguish you from anyone else in your field?

Journal

Write your About section as if explaining yourself to someone at a chapter meeting. That register — a real person talking — is the one that works. It is the one people abandon when they open the edit box.

Exercises

1. Do the eight-step layer-one pass this afternoon.
2. Search LinkedIn for the role you want and read ten profiles of people who hold it. Note the vocabulary.

3. Ask one person to look at your profile and tell you what they think you do.

Project

Complete all of layer one and layer two — profile fully rewritten, three documented projects linked and visible. Then, and only then, start layer three.

AI Prompts

- "Here is my current LinkedIn headline and the role I want. Give me five alternative headlines that contain the terms a recruiter would search for."
- "Here is my About section. Rewrite it in first person, three short paragraphs, without corporate language. Keep everything factual that I gave you."
- "Here are ten profiles of people in the role I want: [summaries]. What vocabulary appears repeatedly that is missing from mine?"

Books

- *Show Your Work!* — Austin Kleon
- *Building a StoryBrand* — Donald Miller. Written for businesses, and the clarity principles apply directly to a profile.

Checklist

- My headline contains the role I want
- My photo is a clear picture of my face
- My About is first person and specific
- My portfolio is linked in Featured and contact info
- My skills list is honest and ordered by relevance
- I have completed layer two before starting layer three

Common Mistakes

- Attempting layer three first
- A headline that is just your current job title
- Third-person About sections
- A portfolio that exists but is not linked anywhere
- Skills lists that are undifferentiated walls of words

Key Takeaways

- Findable, then credible, then known — in that order
- The headline is the highest-value field and most people waste it
- Layer one takes an afternoon and puts you ahead of most people
- Layer three built on nothing is visible and it fails

Next Mission

What to actually publish, once the foundation exists.

PART

TWO

Actually Publishing

What to Publish

MISSION

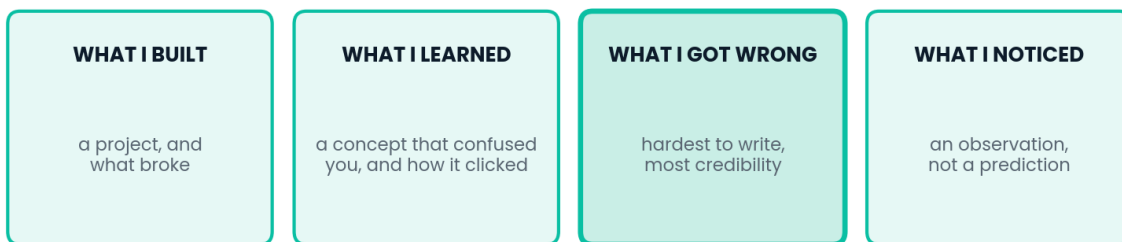
To give you a permanent supply of material, so you never face a blank page.

Objectives

- Rotate four content types so you never run out
- Write specifically rather than profoundly
- Understand why learning posts outperform achievement posts

Framework: The Four Things Worth Posting

ROTATE THESE FOUR AND YOU NEVER RUN OUT



Nobody needs your hot take on an industry you just entered.

Rotate these four and you will not run out of material.

1 — What I built. A project, what it does, what broke. Screenshot, three paragraphs, done. Book 2 generates these automatically if you are documenting.

2 — What I learned. A concept that confused you and how it finally clicked. The most useful posts on the internet are these, and they are the easiest to write because you have just lived them.

3 — What I got wrong. Highest engagement, hardest to write, most credibility-building. A mistake and what it taught you. This is the category people avoid and the one that most changes how you are perceived.

4 — What I noticed. An observation from your actual work about how the industry operates. Not a prediction — an observation. The difference matters: predictions from someone new

read as presumptuous, observations read as attentive.

Rotate those four and you will never run out, because you are generating material simply by doing the work.

Nobody needs your hot take on an industry you just entered. Everyone benefits from your specific account of learning something hard. That distinction is the entire content strategy.

Professional Explanation

Why "What I Got Wrong" Works

It feels like the riskiest category and it is the safest.

The mechanism: everyone in this field has been wrong, frequently and recently. A post admitting it does three things — it signals honesty, it gives the reader permission to have struggled too, and it shows that you actually did the work, since only someone who did it knows how it failed.

It also filters your audience usefully. People who respond badly to honest accounts of failure are not people you want in your professional network.

The format that works:

1. What I believed or attempted
2. What actually happened
3. Why I was wrong
4. What I do differently now

Four short paragraphs. No self-flagellation, no false modesty. Just the account.

Specific Beats Profound

The single most common mistake in professional content is reaching for significance.

"Security is really about people, not technology." True, generic, and it tells the reader nothing they can use.

"I spent two weeks tuning my home SIEM before I realized the reason my alert volume was unmanageable was that I'd enabled every rule by default. Starting from zero and adding detections deliberately would have taken a fraction of the time." Specific, useful, and it demonstrates the general principle without announcing it.

Write the second kind. The general truth is what the reader takes away; your job is to supply the specific instance.

On Length and Polish

Short and imperfect, published, beats long and polished, unpublished. Consistently.

Three to six short paragraphs is a good target for a post. If it needs more, it might be an article, but the bar for that is higher and the return is usually lower early on.

Do not polish for days. The post that took you twenty minutes and the post that took you four hours perform about the same, and one of them is repeatable.

Real Example

The same project, posted two ways.

Achievement post: *"Excited to share that I've completed my Wazuh SIEM deployment! Big milestone in my cybersecurity journey. Grateful for the learning opportunity. #cybersecurity #homelab #grateful"*

Learning post: *"I deployed Wazuh at home two weeks ago and immediately stopped reading my own alerts. Every rule was on by default and the volume was unusable — which turned out to be more educational than the build itself. I spent the next week tuning rather than adding detections, and cut the noise roughly in half. If I did it again I'd start with almost nothing enabled and add deliberately. Turns out alert fatigue is not a theoretical problem."*

The second is the same length. It contains a specific failure, a specific action, a specific result, and a transferable lesson. It also gives anyone reading it something to reply to.

The first tells the reader nothing except that a thing happened.

Case Study: The Post That Got the Interview

Worth understanding for what it says about the mechanism.

A hiring manager reads a post about someone's tuning pass on a home SIEM — a small post, modest engagement, nothing viral. What catches their attention is not the technology. It is that the writer described alert fatigue accurately, from experience, as a beginner.

That tells the manager something no résumé line can: this person will notice when the queue is unmanageable and will do something about it, which is the exact quality they cannot train.

They reach out. The candidate had no idea that post mattered, and could not have predicted it.

This is why consistency beats optimization. You cannot know which post reaches the right reader, so the only strategy available is to keep publishing honest, specific accounts and let the odds work.

Templates: The Post Skeleton

Line 1: the specific thing – a hook, not a headline
Lines 2-4: context – what you were doing, what happened
Lines 5-8: what you learned, concretely
Last line: a genuine question, or nothing at all

No hashtag walls. No "thoughts?" No manufactured vulnerability. Say the true thing plainly and stop.

Templates: The "What I Got Wrong" Post

I thought [belief].

What actually happened: [specific outcome].

The reason I was wrong: [the actual mechanism].

What I do now: [changed practice].

Reflection

- Which of the four categories would you find hardest to write? That is probably the one worth writing.
- Are you reaching for significance instead of specificity?
- What did you get wrong recently that you have not told anyone about?

Journal

Write a "what I got wrong" post about something from the last three months. Do not publish it yet if you are not ready. Most people find the writing is the hard part and the publishing is trivial once the words exist.

Exercises

1. Write one post from each of the four categories this week. Publish at least one.

2. Take a generic statement you believe about your field and replace it with the specific experience that taught it to you.
3. Read ten posts in your field and classify each into a category. Notice which ones you actually learned from.

AI Prompts

- "Here is a project I finished: [description]. Draft a short post about what I learned and what broke, in plain language, without hashtags or corporate enthusiasm."
- "Here is a general statement I want to make: [statement]. Ask me questions to find the specific experience behind it."
- "Read this draft post. Is it specific or is it reaching for significance? Point to the exact sentences."

Books

- *Show Your Work!* — Austin Kleon
- *Made to Stick* — Chip and Dan Heath. On why concrete beats abstract, which is this chapter's entire argument.
- *On Writing Well* — William Zinsser

Checklist

- I rotate the four categories
- My posts contain specific failures, not general truths
- I do not polish for days
- I have published at least one "what I got wrong" post
- I have never posted a hashtag wall

Common Mistakes

- Achievement posts instead of learning posts
- Reaching for profundity
- Manufactured vulnerability, which readers detect instantly
- Hot takes on an industry you just entered
- Polishing indefinitely and never publishing

Key Takeaways

-
- Built, learned, got wrong, noticed — rotate and you never run out
 - Specific beats profound, every time
 - "What I got wrong" feels riskiest and is safest
 - You cannot know which post reaches the right reader

Next Mission

The part where most people quit.

A Practice You Won't Quit

MISSION

To build a sustainable rhythm instead of a burst of enthusiasm that dies in three weeks.

Objectives

- Set a cadence you can hold on a bad week
- Engage more than you broadcast
- Stop measuring the wrong things

Why It Matters

Almost everyone who tries this quits, and they quit for the same reason: they set an unsustainable pace, saw little return in the first month, and concluded it did not work.

The return is real and it is slow. Designing for month twelve rather than week three is the entire discipline.

Step-by-Step Guide: The Practice

1. **Pick one platform.** LinkedIn for IT and cybersecurity careers. Add a second only after the first is a genuine habit — and possibly never.
2. **Set a cadence you can hold on a bad week.** Once a week beats daily-then-nothing, decisively. If once a week feels heavy, once a fortnight is a real cadence too. Consistency over volume, always.
3. **Write from your own work.** Every finished project, every problem solved, every concept understood is a post. If you are learning, you have supply — which is why Book 2 comes first.
4. **Batch when you can.** Write three posts on a Sunday when you have energy, schedule them out. This is how people with unpredictable weeks stay consistent, and it removes the "I don't feel like it" failure mode entirely.
5. **Engage more than you broadcast.** Thoughtful comments on other people's posts build more relationships than your own posts do, especially in the first six months. This is the part everyone skips. It is where the network actually forms.
6. **Ignore the metrics.** The post that matters is the one the hiring manager reads before your interview. You will never know which one that was, and engagement numbers are a poor proxy for it.
7. **Do not quit at month three.** Month three is the trough. Almost everyone who has built something visible went through it.

Professional Explanation

Why Engagement Beats Broadcasting Early

When you have no audience, posting into a void feels pointless — and mechanically, it nearly is at first.

Commenting is different. A thoughtful comment on someone else's post is seen by their audience, it starts an actual conversation with a real person, and it costs three minutes rather than thirty.

What works: adding a specific detail, asking a genuine question, or offering a different experience. What does not: "Great post!" or a link to your own thing.

Do this for two months before worrying about your own reach. You will find that people start recognizing your name, and some of them will follow you without you having posted much at all.

On Comparison

You will see people with large followings producing content you think is worse than yours. This is genuinely demoralizing and it is worth addressing directly.

Some of them started earlier. Some are optimizing for reach in ways you would not want to. Some are better at something you are undervaluing. None of it is relevant to your actual goal, which is not a following — it is being findable and credible to the right twenty people.

A person with three hundred followers, all of whom are practitioners in their field, has a more valuable network than someone with fifty thousand who are mostly bots and people in unrelated industries.

The specific trap: changing what you post to chase engagement. The posts that perform best in raw numbers are frequently not the posts that get you opportunities. Generic career advice outperforms specific technical accounts nearly every time — and specific technical accounts are what makes a hiring manager reach out. Optimize for the right reader, not the algorithm.

Framework: The Weekly Rhythm

A realistic version that takes about ninety minutes.

- **Fifteen minutes, twice a week:** read your feed and leave two thoughtful comments each time.

- **Thirty minutes, once a week:** write and publish one post from your project log or break log.
- **Fifteen minutes, once a week:** respond to anyone who replied to you. Every reply is a potential relationship and most people ignore them.
- **Ongoing:** anyone who engages meaningfully goes into Book 1's contact ledger.

Ninety minutes a week, sustained for a year, produces something that changes your career. The same ninety minutes spent applying to jobs does not.

Real Example

Two people start posting in the same month.

The first commits to daily posting. He manages eleven days, runs out of material, starts writing generic advice to fill the gap, feels increasingly uncomfortable about it, and stops entirely at week four.

The second commits to one post a week, from her project log. She misses two weeks over the year — once when she was ill and once at Christmas. She also comments on other people's work a few times a week.

At twelve months: he has eleven posts and no habit. She has fifty posts, a network of practitioners who recognize her name, three conversations that turned into opportunities, and a practice she will still have in five years.

Neither was more talented. One picked a sustainable cadence.

Case Study: The Trough at Month Three

Worth naming because it is nearly universal and it is where people quit.

Months one and two feel fine — novelty carries you. Month three is when the novelty is gone, the engagement is still low, nothing has visibly come of it, and the effort starts feeling foolish.

Everyone who has built anything visible went through this. The people who came out the other side did not have better content in month four; they simply continued.

If you know it is coming, it is much easier to sit through. Mark it in your calendar if it helps: *month three is the trough, keep going.*

Reflection

- What cadence could you genuinely hold during your worst month?
- Do you comment more than you post?
- Are you measuring engagement or opportunity?

Journal

Write down what would make you quit this. Then decide in advance what you will do when it happens. Most people quit at a predictable moment for a predictable reason.

Exercises

1. Set your cadence. Write it down. Make it slightly less than you think you can manage.
2. Leave two thoughtful comments this week on other people's posts.
3. Batch-write three posts in one sitting and schedule them.

Project

Sustain your chosen cadence for twelve weeks without missing more than one. Then look at what changed — not in metrics, but in who knows your name.

AI Prompts

- "Here is my project log for the last month. Suggest six post ideas from it, each with a one-line angle. Do not write the posts."
- "Here are three drafts. Which is most specific and which is reaching for significance?"
- "I want to comment usefully on this post: [post]. What specific detail or genuine question could I add that is not just agreement?"

Books

- *Atomic Habits* — James Clear. Directly applicable: the rhythm question is a habit-design question.
- *Show Your Work!* — Austin Kleon
- *Deep Work* — Cal Newport

Checklist

- I have a cadence I can hold on a bad week
- I comment more than I post
- I batch-write when I have energy
- I respond to everyone who replies
- I have not changed what I post to chase engagement

Common Mistakes

- Setting an unsustainable pace
- Broadcasting without engaging
- Quitting at month three
- Optimizing for the algorithm rather than the right reader
- Ignoring replies, which are the actual opportunity

Key Takeaways

- Once a week held for a year beats daily for three weeks
- Comment more than you post, especially early
- Month three is the trough; know it is coming
- The metric that matters is opportunity, not engagement

Next Mission

The one way this reliably goes wrong.

What Never to Post

MISSION

To keep you from the single mistake that turns visibility into a liability.

Objectives

- Know exactly what is off limits
- Understand why sanitizing is not enough
- Handle the grey areas

Why It Matters

Everything else in this book is upside. This chapter is the downside. It is worth reading twice.

There is essentially one way personal branding damages a technology career. It is talking about your employer's environment.

Professional Explanation

The Absolute Prohibitions

Never post about:

- **Your employer's environment.** Architecture, tooling, versions, configurations, weaknesses.
- **Incidents.** Yours or anyone's, current or historical, however long ago.
- **Vulnerabilities you found at work.** Even after they are fixed. Even without naming the employer.
- **Customer or client information.** In any form, however anonymized you believe it to be.
- **Internal frustrations.** Your manager, your team, a decision you disagree with, a vendor you find incompetent.
- **Anything under NDA,** which is more than you think.

Anonymizing is not enough. "A company I work with" is identifiable from your profile in about four seconds. "A few years ago at a previous employer" narrows it to one of two organizations. People are much better at this inference than writers expect, and the person doing the inferring is frequently the employer's security team.

Why This Is So Damaging

Three separate mechanisms, any one of which is enough:

It is a security issue. You have disclosed something about a defended environment to an audience that includes people who target such environments. That is true even if the detail seems trivial.

It is a trust issue. Every future employer now knows you will talk about their environment publicly. This is a deal-breaker for security roles specifically, where keeping things private is a core requirement rather than a nice quality.

It may be a contract or legal issue. Employment agreements, NDAs, and customer contracts frequently cover far more than people realize.

The damage is also lasting. A deleted post is not gone, and the reputation follows.

The Grey Areas

Publicly disclosed vulnerabilities. Fine to discuss — the vendor has published them. Not fine to connect to whether your employer is affected.

Your own lab. Completely fine. Break it, document it, post about it freely. This is why Book 2's material is safe content and your workplace's material is not.

Industry commentary. Fine, with care. "This class of attack is increasing" is fine. "This class of attack is increasing and we're seeing a lot of it" tells an attacker something about your employer.

Job searching while employed. Legitimate, and use recruiter-only visibility settings. Do not post about being unhappy at your current job.

Conference talks and write-ups about work. Get explicit approval in writing, from someone with authority. This happens all the time and it is fine when approved — Book 4's documentation reflex applies.

Your own certifications and achievements. Fine, though Chapter 3 explains why learning posts serve you better.

The Test

Before posting anything that touches your professional work, ask: **would I be comfortable if my employer's security team, my manager, and a future interviewer all read this?**

If the answer is anything other than yes, do not post it. There is no version of the upside that justifies the downside here.

A second, sharper test: **could an attacker use this?** If any part of the answer is "possibly, in combination with other things," that is a no.

Real Example

Someone posts: *"Spent all day dealing with a legacy system that still can't support MFA. Sometimes the hardest part of security is the stuff you inherited."*

It feels harmless. It is relatable, it is vague, it names nothing.

What it actually communicates, to anyone who looks at the profile: this specific organization has at least one system that cannot support MFA, it is in use, and someone in security is frustrated about it and has not been able to fix it.

That is a genuine intelligence contribution to anyone interested in the organization. It is the sort of thing that ends up in a security team's conversation about the person who posted it.

The equivalent safe version: *"MFA gaps in legacy systems are one of the most common findings in any environment. Compensating controls — network restriction, monitoring those accounts specifically, quarterly review — are usually more realistic than replacement."* Same knowledge, no disclosure, and it demonstrates more expertise.

Case Study: What Employers Actually Watch

Worth knowing rather than guessing.

Many organizations monitor for public disclosure about their environment, sometimes through a formal function and sometimes because a colleague noticed and mentioned it. Security teams in particular pay attention to what staff post.

This is not sinister. Managing what is publicly knowable about your environment is part of the job.

The consequence is that the post you think disappeared into a low-engagement void may well have been read by exactly the people you would least want reading it. Engagement numbers do not indicate who saw something.

Templates: The Safe Rewrite

When you want to write about something from work, convert it:

```
FROM: a specific thing at my employer
TO:   the general pattern, plus how it's typically handled

FROM: "we have [problem]"
TO:   "[problem] is common in [type of environment], and
       the usual approaches are [X] and [Y]"

FROM: an incident I worked on
TO:   the publicly documented equivalent, or my lab
       reproduction of the same class of issue
```

The general version usually demonstrates more expertise than the specific one, because it shows you understand the pattern rather than just the instance.

Reflection

- Have you posted anything that would fail the test?
- Do you know what your employment agreement says about public statements?
- Could someone infer your employer from your profile in under a minute? (*Almost certainly yes.*)

Journal

Write about the most interesting thing that happened at work recently. Then rewrite it as a general pattern with no identifying detail. Notice that the second version is usually the better piece of writing.

Exercises

1. Review everything you have posted publicly against the test. Delete anything that fails.
2. Read your employment agreement's section on confidentiality and public statements.
3. Take one work story and convert it to the general pattern.

AI Prompts

- "Here is a post I want to publish about my work. Identify anything that could be used to infer my employer's environment or that an attacker could find useful."
- "Rewrite this as a general industry pattern with no identifying detail, keeping the expertise visible."
- "What categories of information do employment agreements typically restrict from public discussion?"

Checklist

- I never post about my employer's environment
- I know that anonymizing is not enough
- I get written approval before any work-related talk or write-up
- I apply the two tests before publishing anything work-adjacent
- My lab is my content source, not my workplace

Common Mistakes

- Believing vague equals safe
- Posting frustrations about work
- Discussing incidents, however old
- Assuming low engagement means nobody saw it
- Verbal approval for a conference talk

Key Takeaways

- One reliable way to damage a career with visibility: talking about your employer's environment
- Anonymizing does not work; people infer quickly
- The general pattern demonstrates more expertise than the specific instance
- Your lab is safe content; your workplace is not

PART

THREE

Going Further

Long-Form: Podcasts, Newsletters, and Speaking

MISSION

To help you decide whether to go beyond posts, and to do it sustainably if you do.

Objectives

- Choose a format you would continue with an audience of five
- Understand what long-form actually costs
- Use speaking and chapter involvement as the fastest route

Why It Matters

Posts build layer three. Long-form builds something different: depth of association. People who listen to an hour of you thinking have a relationship with your work that a post cannot create.

It also costs much more, and most people who start a podcast or newsletter stop within six months. This chapter is about deciding honestly and then designing for continuation.

Professional Explanation

The Test Before You Start

Would you keep doing this with an audience of five?

Because for a long time, it will be five. I have run a podcast long enough to tell you that the early episodes are for you, not for anyone else. They are how you get good enough to deserve the later audience.

If the honest answer is no — if the point is the audience rather than the work — pick something else. The formats that survive are the ones where the process itself is worth your time.

What Each Format Actually Costs

A podcast. Two to four hours per episode when you include booking, preparation, recording, editing, and publishing. Interviewing is the sustainable version, because guests provide the material and the preparation is genuinely interesting. Solo is harder to sustain than people expect.

A newsletter. Two to four hours per issue if it is substantive. The advantage: it is fully yours, the audience is a list you own rather than an algorithm's, and it compounds. The risk: a weekly cadence is brutal and most people should start monthly.

Speaking. Ten to twenty hours for a first talk, much less afterward as material gets reused. Highest return per hour of anything in this chapter, and the most underused.

Long-form writing. An article is four to eight hours. One good article a quarter outperforms twelve rushed ones and is far more sustainable.

Why Speaking Is Underrated

Chapter meetings — ISACA, ISSA, BSides, OWASP, local meetups — are actively looking for speakers. They are frequently short of them.

That means the barrier is much lower than people assume. You do not need to be a recognized expert; you need a useful twenty minutes on something you actually did. A talk on "what I learned building a home SIEM" is genuinely welcome at most local chapters.

What it produces: immediate credibility in a room of practitioners, several conversations afterward that go into your contact ledger, reusable material, and — after two or three talks — a reputation as someone who contributes rather than only attends.

Partnering with a chapter, as I do with ISACA Sacramento, takes it further. But you do not need a formal role. You need to offer once.

Framework: Choosing Your Format

Match it to what you actually enjoy, because enjoyment is the only reliable predictor of continuation.

If you enjoy	Choose	Why it survives
Talking to people	Podcast, interview format	Guests supply the material and the obligation keeps you shipping
Writing and thinking alone	Newsletter, monthly	Fully owned audience, no algorithm
Explaining to a room	Speaking	Highest return per hour, and chapters need you

If you enjoy	Choose	Why it survives
Building things	Long-form write-ups	Book 2 generates the material automatically
None of the above	Stay with short posts	This is a completely legitimate answer

That last row matters. Long-form is not a requirement. A consistent short-post practice plus a strong portfolio is enough for nearly every career outcome in this field.

Step-by-Step Guide: Starting Sustainably

1. **Commit to a small number.** Ten episodes, six issues, one talk. A finite commitment you can complete beats an open-ended one you abandon.
2. **Batch the unglamorous parts.** Record three episodes in a session. Draft two newsletters at once.
3. **Lower the production bar deliberately.** Good audio and honest content beats elaborate editing. Nobody has ever stopped listening to a useful podcast because the intro music was plain.
4. **Reuse across formats.** A talk becomes an article becomes three posts. This is the single biggest efficiency available and almost nobody does it.
5. **Decide your review point in advance.** After ten episodes, ask whether you want to continue. Book 9's quarterly review logic applies directly.
6. **Do not check the numbers weekly.** They will be discouraging and they are not the point.

Real Example

Two people start podcasts in the same month.

The first plans a polished production: custom music, heavy editing, a launch strategy. Three episodes in, the editing burden is six hours per episode and he has not booked a fourth guest. He stops.

The second books three guests, records all three in one week with basic equipment, edits lightly, and publishes on a fortnightly schedule. She commits to ten episodes before deciding anything.

At episode ten she has a real skill she did not have before — asking good questions and listening properly — a network of ten guests who now know her, thirty short posts generated from episode content, and a decision to make with actual information.

The difference was not ambition. It was designing for continuation.

Case Study: Why the Early Episodes Are For You

Something worth being honest about, since I have lived it.

The first episodes of anything are not very good. The questions are too broad, the pacing is off, you talk too much or too little, you miss the interesting thread because you were thinking about your next question.

Nobody tells you this, and if you judge the early work by the standard of things you admire, you will stop.

The correct frame: the early episodes are practice, performed publicly, and the small audience is a feature rather than a failure. You are getting reps in front of very few people, which is exactly the right way to get good.

The skill that comes out of it — listening properly, asking a real follow-up, being comfortable in a conversation you do not control — transfers directly to interviews, stakeholder meetings, and everything in Book 8.

Reflection

- Would you do this with an audience of five?
- Which format matches something you actually enjoy?
- Is the point the work or the audience?

Journal

Write about why you want to do this. Be honest. If the answer is primarily about recognition, that is worth knowing before you spend two hundred hours on it.

Exercises

1. Email your local chapter and offer to give a twenty-minute talk. This week.
2. If you are considering long-form, commit to a finite number and write it down.
3. Take one thing you have already made and convert it into a different format.

Project

Give one talk at a local chapter or meetup. Twenty minutes on something you actually did. This is the single highest-return item in this book and the one people most avoid.

AI Prompts

- "I want to give a twenty-minute talk on [topic] to a local security chapter. Help me structure it: what is the one thing the audience should leave with, and what should I cut?"
- "Here is a talk outline. What questions will the audience ask that I have not prepared for?"
- "Turn this article into three short posts, each standing alone, without repeating the same angle."

Books

- *Show Your Work!* — Austin Kleon
- *Talk Like TED* — Carmine Gallo
- *On Writing Well* — William Zinsser

Podcast Episodes

The early episodes of Wake Up X, if you want a demonstration of this chapter's argument. They are not as good as the later ones. That is the point.

Resources

- Your local **ISACA**, **ISSA**, **BSides**, or **OWASP** chapter — actively looking for speakers
- Any local meetup, which is an easier first audience than a conference

Checklist

- I would do this with an audience of five
- My format matches something I genuinely enjoy
- I committed to a finite number rather than an open-ended practice
- I batch the unglamorous parts
- I reuse material across formats
- I have offered to speak at least once

Common Mistakes

- Starting because of the audience rather than the work
- Over-producing early, creating an unsustainable burden
- Open-ended commitments

- Never reusing material across formats
- Judging early work by the standard of things you admire

Key Takeaways

- Would you do it with an audience of five? If not, choose differently.
- Speaking is the highest return per hour and the most underused
- Commit to a finite number, then decide
- The early episodes are for you, and that is not a consolation

Next Mission

Book 8: the mindset that turns technical skill into a consulting career.

Appendix: Profiles, Skeletons, and the Cadence

REFERENCE

Everything from this book in one place.

The Three Layers

- | | | |
|---|----------|---|
| 1 | FINDABLE | complete profile, target role in headline, portfolio linked, real photo |
| 2 | CREDIBLE | evidence attached – Book 2's portfolio |
| 3 | KNOWN | consistent output over a year or more |

Layer 3 built on nothing is visible, and it fails.

The Headline Formula

[Current or target role] | [Trajectory if changing] |
[Strongest specific evidence] | [Credential if it
differentiates]

The About Section

- 1 – What I do now, plainly
- 2 – What I'm building toward, and the evidence
- 3 – Something human

First person. Links at the end.

Layer One Checklist

- [] Headline contains the role I want
- [] Clear photo of my face
- [] About in first person, three paragraphs
- [] Portfolio in Featured and contact info
- [] Skills honest and ordered by relevance
- [] Experience as evidence bullets
- [] Fifty connections in my target field

The Four Things Worth Posting

WHAT I BUILT	a project, and what broke
WHAT I LEARNED	a concept that confused you
WHAT I GOT WRONG	hardest to write, most credibility
WHAT I NOTICED	an observation, not a prediction

The Post Skeleton

Line 1: the specific thing – a hook, not a headline
Lines 2-4: context – what you were doing, what happened
Lines 5-8: what you learned, concretely
Last line: a genuine question, or nothing

The "What I Got Wrong" Post

I thought [belief].
What actually happened: [outcome].
The reason I was wrong: [mechanism].
What I do now: [changed practice].

The Weekly Rhythm — 90 minutes

15 min x2	read the feed, two thoughtful comments each
30 min x1	write and publish one post
15 min x1	respond to everyone who replied
ongoing	engagers go into the contact ledger

The Two Tests Before Posting

1. Would I be comfortable if my employer's security team, my manager, and a future interviewer all read this?
2. Could an attacker use this?

Anything other than yes / no – don't post it.

The Safe Rewrite

FROM: a specific thing at my employer
TO: the general pattern, plus how it's typically handled

The general version usually demonstrates more expertise.

Choosing a Long-Form Format

Enjoy talking to people	-> podcast, interview format
Enjoy writing alone	-> newsletter, monthly
Enjoy explaining to a room	-> speaking (highest return)
Enjoy building	-> long-form write-ups
None of the above	-> stay with short posts

Would you do it with an audience of five?

A Closing Note

I want to be direct about the thing this book is really arguing, because it is easy to read the tactics and miss it.

The reason to be visible is not recognition. It is that opportunity is distributed by memory – someone needs something, they think of a name, they reach out. If your name is not in anyone's head, you are competing only through portals, against two hundred other applications, at the worst possible odds.

Being findable changes which game you are playing.

And the window matters more than people accept. The opportunity that arrives when you are not quite ready is frequently the one that changes the trajectory, and declining it does not

preserve it. It ends it.

So start before you are ready. Post the thing you learned this week, badly. Offer to give the twenty-minute talk. Fix the headline this afternoon.

You are not building an audience. You are making sure that when someone needs what you do, your name is the one that surfaces.

Fix the profile today. Post what you learned this week. Offer to speak once.

Connect

[linkedin.com/in/efrenlsalazar](https://www.linkedin.com/in/efrenlsalazar) | wakeupxpodcast.com

Questions for the author. The core argument of Chapter 1 is yours and it is currently stated rather than proven. Two gaps: (1) **an opportunity that came to you because you were visible** — a specific one, with what it led to; (2) **an opportunity you passed on and regretted**, which is the sharper half of the argument and the one readers will remember. Third, smaller: what actually made you start the podcast, for Chapter 6 — the current version uses my framing where your reason would be better.

IN THE FIELD

The Brand in Practice

Wake Up X: Network With Efren Salazar — the same visibility principle in this book, applied episode after episode.



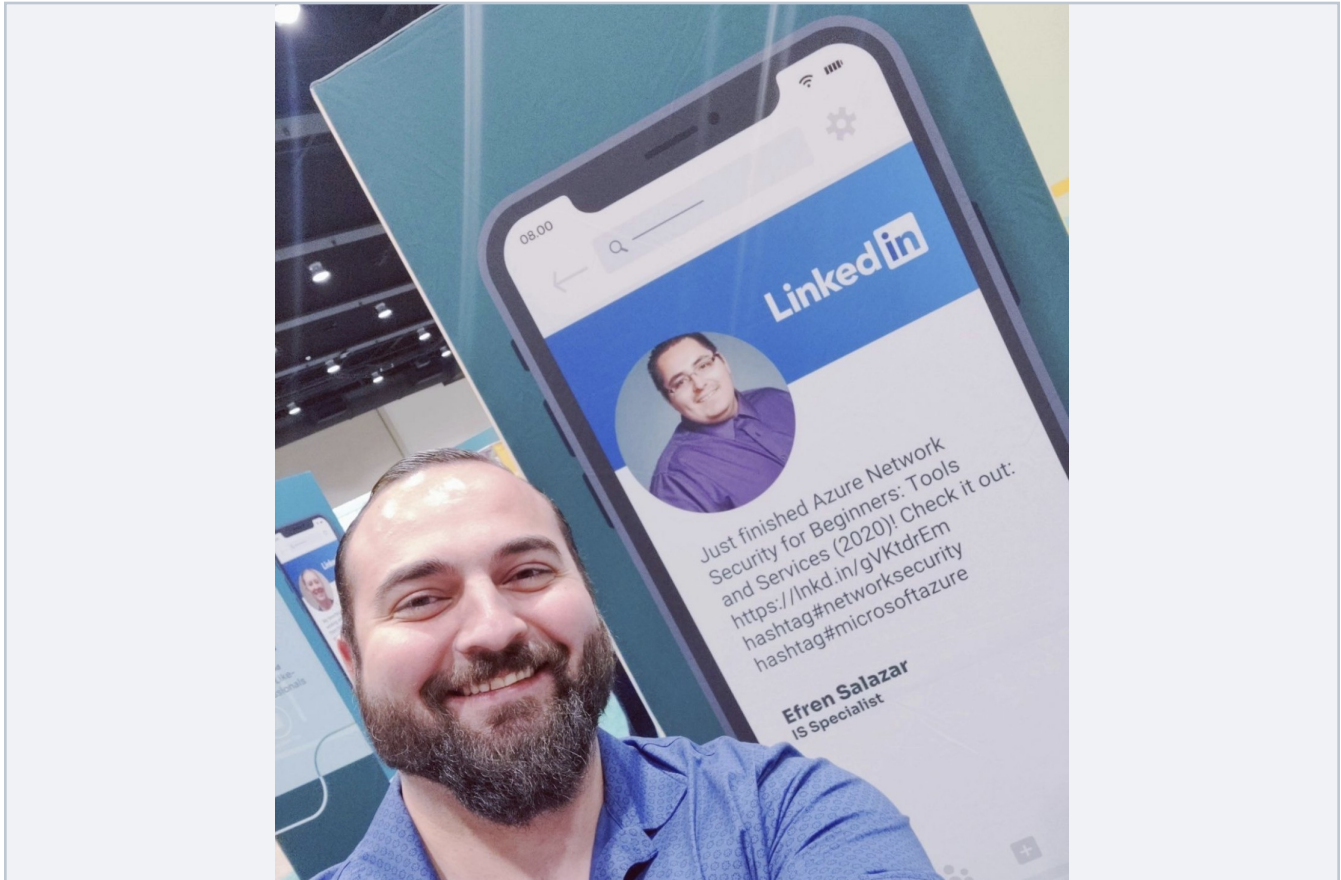
Podcast guest promo — Gustavo Mastroianni, CISO



Podcast guest promo — Devin Prasad, Security Architect

Nobody Handed Me the Opportunity

I didn't get a referral into this industry, and I didn't get handed a career. Every job, every client, every certification came from building something visible enough that people could find it — then showing up in person to back it up. Visibility didn't replace the work. It's what let the work get seen.



A project post, featured at a conference LinkedIn kiosk

IN THE FIELD

I Made My Own Path

That same post ended up back on a screen at a national conference, with a stranger pointing at it as if she already knew me. She didn't, not really — the post had done that work in advance. That's the whole argument of this book: build in public long enough, and the room starts to recognize you before you introduce yourself.



The same post, recognized on the conference floor

REFLECTION

What have you built that could be found by someone who's never met you?
If the answer is nothing yet, that's this chapter's starting assignment.

BE OUR NEXT GUEST

Your Story Belongs on Wake Up X

If Build Your Brand gave you something worth using, you already have something worth sharing. The Wake Up X podcast exists to put real, unpolished career and business stories in front of people who need to hear them — and every guest so far started exactly where you are now: reading, building, and figuring it out in public.

You might be a fit if you:

Broke into tech, cybersecurity, or business without the traditional path

Built something — a lab, a business, a portfolio, a community — and can talk about how

Have a failure or turning point you're willing to tell straight, not polished

Are mentoring, leading, or building something worth other people knowing about

HOW TO APPLY

Share your story at wakeupxpodcast.com/guest, or connect directly on LinkedIn at linkedin.com/in/efrenlsalazar and mention this book.

Being a guest is also visibility — the same principle Book 7 makes the case for.

Acknowledgments

THE PEOPLE WHO MADE THIS POSSIBLE

None of Wake Up X happened alone, including Build Your Brand. Every book in this library, the podcast, and the free library exist because specific people showed up for a kid nobody expected much from. This is the record of who they are.

THE EARLY YEARS

Miss England, the 6th-grade teacher who stayed after class, unpaid, to teach the fundamentals. Sister Cora and Lewis Bair of Computers for Kids, who handed a teenager a key and a broken machine and let the work speak. Jose Luis Bravo, Angel Martinez, Isaac Murrilo, and Alex De Haro — the friends and cousins who translated the world, gave hope, drove him where he needed to go, and carried him before there was anything to carry him toward.

THE BUSINESS YEARS

Rojelio Salazar, his cousin and first mentor, RNS Computer Repairs — the reason it is called Wake Up X. Kyle Erni, his first business partner. Erick and Pancho Arauza, co-founders of The Cellphone Fix. Danny Maya, of Maya Traditional Mexican Restaurant, Taqueria Maya, and Salt and Water, who pushed him in front of a camera for the first time.

THE CAREER YEARS

Igor, who gave him the vocabulary for work he already knew how to do. Nate, who he mentored into a full-time role alongside his own. Ken Kerrick, opponent turned friend in the ISACA Sacramento election. Alvin Jue and David De los Reyes, who believed in the middle of it, not just at the finish.

THE PODCAST

Every guest who trusted an unproven show with their story, including Gustavo Mastroianni, Cathy Stephenson, Greg McCord, Aaron Portnoy, Sana Talwar, Jorge Avila, Anil Reddy, Devin Prasad, Larry Switzer, Louis Kemp, Mohana Balakrishnan, Eddie Lopez, and Michael Fierros — full episodes at wakeupxpodcast.com.

COMMUNITY PARTNERS

ISACA Sacramento and ISC2 Sacramento, The Cellphone Fix, A.N.U.L Technologies, AISOC, AspireByte, and PodGrabber — the organizations building alongside this movement.

AND

His parents, who arrived with nothing, worked since childhood, and gave away whatever was left over. Nothing about the first version of this story predicted the version you are holding now.

Continue the Journey

EVERY BOOK LEADS SOMEWHERE ELSE

Build Your Brand is one part of a 12-book system. If it helped, the next step is one page away — take whichever layer matches where you are standing.

THE LIBRARY

Layer 1 — The Story

I Just Have To Wake Up — why any of this exists.

Layer 2 — The Mindset

Comfort Is Expensive — the real cost of playing it safe.

Layer 3 — The Blueprint

DECODE: The Wake Up X Playbook — zero experience to executive thinking. Book 0.

Layer 4 — The Playbooks

1 Break In · 2 Build It · 3 Think Like a Cybersecurity Professional · 4 Business First · 5 Lead Without a Title · 6 AI for IT & Cybersecurity · 7 Build Your Brand · 8 The Consultant Mindset · 9 Level Up

FIND US EVERYWHERE

Podcast & free library: wakeupxpodcast.com · Spotify: open.spotify.com/show/28GZnRBrIV2efHOHzdiftt · Apple Podcasts: podcasts.apple.com/us/podcast/id1792821643 · YouTube: youtube.com/@Efren.wakeupx · LinkedIn: linkedin.com/in/efrenIsalazar · Substack: substack.com/@wakeupxefren

IN GOOD COMPANY

ISACA Sacramento, ISC2 Sacramento, The Cellphone Fix, A.N.U.L Technologies, AISOC, AspireByte, and PodGrabber — the partners and collaborators building alongside the movement. See the full list at wakeupxpodcast.com/community.html.

Get DECODE free, apply to be a guest, or read the story behind the movement — all of it starts at wakeupxpodcast.com.

WAKE UP X

SOLUTIONS



Efren L. Salazar

FOUNDER · HOST · CYBERSECURITY PROFESSIONAL · EDUCATOR · COMMUNITY BUILDER

STAY CONNECTED

Podcast & Blueprint Library:	wakeupxpodcast.com
LinkedIn:	linkedin.com/in/efrenlsalazar
YouTube:	youtube.com/@Efren.wakeupx
Be a Guest:	wakeupxpodcast.com/guest

THE WAKE UP X PUBLISHING SYSTEM

BOOK 8

The Consultant Mindset

Discovery, scope, and stakeholder management

You are not being paid for the answer

THE
WAKEUPX
SOLUTIONS

EFREN L. SALAZAR

Founder & Host, Wake Up X | IT Operations Manager & Solution Engineer
Government & Regulatory Partner, ISACA Sacramento | CompTIA Security+

wakeupxpodcast.com

Educational content — not professional, legal, financial, or career-placement advice.
Full disclaimer in DECODE, the foundation of this library.

Before You Begin

HOW TO USE THIS BOOK

This book is about diagnosing instead of prescribing — the shift from being paid for hours to being paid for judgment.

Where this book sits

Two books come first — the memoir *I Just Have To Wake Up* and the mindset book *Comfort Is Expensive* — before this ten-book library. Each one is short on purpose. Pick up the one that matches the problem in front of you right now.

- n **Book 0 — DECODE.** The foundation. The map of the whole path.
- n **Book 1 — Break In.** Landing your first role in IT or cybersecurity.
- n **Book 2 — Build It.** Home labs and a portfolio that prove you can do the work.
- n **Book 3 — Think Like a Cybersecurity Professional.** Detection, triage, and incident response thinking.
- n **Book 4 — Business First.** Finance, sales, GRC, and why the business hired you.
- n **Book 5 — Lead Without a Title.** Influence, EQ, and mentoring before anyone promotes you.
- n **Book 6 — AI for IT & Cybersecurity.** Using AI as an actual working tool, not a novelty.
- n **Book 7 — Build Your Brand.** LinkedIn, content, and being findable.
- n **Book 8 — The Consultant Mindset.** Discovery, solution selling, and stakeholder management. **you are here**
- n **Book 9 — Level Up.** Long-term growth, burnout, legacy, and mentoring others.

Book 0 is the foundation. *DECODE: The Wake Up X Playbook* holds the six-level path, the Curiosity Compass, and the core frameworks. Nothing in any other book contradicts it. If a term here is unfamiliar, DECODE defines it.

What is in this book

Part One — Diagnosing Before Prescribing

1. You Are Not Being Paid for the Answer
2. The Discovery Ladder

Part Two — The People Around the Work

3. Map the Room

4. Scope, and Saying No

Part Three — Running the Work

5. The Engagement Rhythm

6. Pricing, Positioning, and Knowing Your Worth

7. Appendix: Kickoffs, Scripts, and the Ladder

How to actually use it

You do not need to be a consultant for this to apply. It applies to solution engineers, internal architects, anyone who gets asked "what should we do about X?", and anyone who has ever been handed a request that was not really the problem.

That is most people in this field within a few years.

This book builds directly on Book 4. If you have not read the chapters there on the value translation and on selling, several things here will feel like they are missing a foundation — because they are.

A note on the framework in Chapter 2. The five-step sequence there is named The Discovery Ladder. An earlier draft of the Wake Up X material labelled the same sequence V.A.L.U.E., which collides with the V.A.L.U.E. framework already defined in DECODE and used in Book 4 — same letters, different meanings, guaranteed confusion for anyone who reads both. I have kept them distinct. If you would rather they share the name, that is a decision to make once and apply everywhere.

I am writing this the way I would say it to you sitting across a table — not lecturing a room. Take what works, remix the rest into your own version, and leave what does not fit.

PART

ONE

Diagnosing Before Prescribing

You Are Not Being Paid for the Answer

MISSION

To shift you from delivering solutions to diagnosing problems.

Objectives

- Understand what clients and stakeholders actually buy
- Recognize the request that is not the real problem
- Earn the right to disagree with the request

Story

I have been on both sides of this. I have run a business where clients came to me with a stated problem. I have been the internal person handed a request from leadership with a solution already attached.

The pattern is the same in both directions, and I learned it the expensive way in my repair business.

Customers came in saying "my phone needs a new screen." Sometimes true. Frequently the actual issue was different — a failing battery causing a symptom that looked like a display problem, or water damage they had not mentioned, or a phone that was genuinely not worth repairing and what they really needed was honest advice about replacing it.

If I had simply done what was asked every time, I would have taken money for work that did not solve the problem, and the customer would have come back angry. Not because I did the job badly, but because I did the wrong job well.

Our competitive advantage was never repair quality. Plenty of people could replace a screen. It was that people trusted what we told them, and that trust came from diagnosing rather than order-taking.

Why It Matters

A technician receives a request and fulfills it. A consultant receives a request and figures out whether it is the right request.

That single difference determines whether you are paid for hours or for judgment — and judgment scales, hours do not. It also determines whether you spend your career executing other people's diagnoses, which is a ceiling most technical people hit without recognizing what it is.

Professional Explanation

The Presented Problem Is Rarely the Real One

Someone says: *"We need a new ticketing system."*

Maybe. Or:

- The current system is fine and nobody was trained on it
- The process around it is broken and no tool fixes a broken process
- Two teams disagree about ownership, and a tool buy is the politically easier move than the conversation
- One executive had a bad experience last quarter and this is the response
- The reporting is poor, which is a configuration problem rather than a product problem

If you deliver the new ticketing system without finding out which of those is true, one of two things happens. You solve the problem — possible, and you got lucky. Or you spend six months and significant money and the same complaints resume within a quarter.

The second outcome is far more common, and here is the part that matters for you: the blame lands on the implementation. Which is to say, on you. Nobody remembers that the diagnosis came pre-attached to the request.

The consultant's first job is to earn the right to disagree with the request. You do that by showing that you understand their situation better than they expected you to. Then, and only then, can you say "I think the actual issue is somewhere else" and have it land as insight rather than as resistance.

What They Are Actually Buying

Four things. It is rarely the one you think.

Certainty. They want to stop being unsure about this. A confident, reasoned recommendation has value independent of which option it recommends.

Risk transfer. If it goes wrong, someone else was accountable for the recommendation. This is a legitimate and rational thing to want, and understanding it explains a great deal about how engagements are structured.

Speed. You have seen this before; they have not. What takes them three months of learning takes you a week.

Permission. Sometimes they already know the answer and need an outside voice to say it so it can be acted on. This is far more common than most people realize — internally as much as externally.

That last one is worth recognizing when it happens, because it changes what you should do. If someone already knows the answer, your job is not to discover it independently and present it as your own.

It is to validate it, strengthen it with reasoning they can use, and give them something they can take to whoever needs convincing.

Diagnosing Without Being Obstructive

There is a failure mode on the other side of this. It is worth naming: the person who questions every request, treats every stated problem as suspect, and cannot simply do the thing when the thing is obviously correct.

That person is exhausting, and they do not get asked back.

The judgment is: **how much is riding on this being the right problem?** A small request with a low cost of being wrong — just do it. A six-month project with a significant budget — diagnose properly. Match the diagnostic effort to the stakes.

Real Example

An internal team asks a security engineer to implement a data loss prevention tool. Budget approved, vendor selected, timeline set.

Order-taking: implement it. Six months later it generates enormous false-positive volume, is largely bypassed, and is regarded as a failed project.

Diagnosing: the engineer asks what triggered the request. It emerges that an employee emailed a customer list to a personal account before resigning, and leadership wanted assurance it would not happen again.

That is a specific problem with a much narrower solution — monitoring for bulk data movement to personal accounts, plus an offboarding process change. It costs a fraction, deploys in weeks, and actually addresses the thing that happened.

The engineer did not refuse the request. She found out what the request was *for*, and offered a better route to the same destination.

Case Study: The Question That Changes the Engagement

One question does more diagnostic work than any other.

"What made this a priority now?"

Every request has a trigger. Something happened, or someone said something, or a number moved. Finding the trigger tells you what the request is actually for, which is frequently different from what it says.

Variations that work:

- "What changed recently that made this urgent?"
- "What would happen if we did nothing for six months?"
- "Who is most keen for this to happen, and why?"

The answers reframe engagements regularly. A request that looked like a technology decision turns out to be a response to an audit finding, a reaction to a competitor, or a new executive establishing their agenda. All three need different work.

Reflection

- When did you last do exactly what was asked without finding out why?
- Do you match diagnostic effort to stakes, or apply the same level to everything?
- Have you ever been blamed for an implementation that failed because the diagnosis was wrong?

Journal

Write about a request you received that turned out not to be the real problem. What was the signal you noticed, or what did you miss?

Exercises

1. Take the last request someone made of you. Write down three plausible underlying problems that would produce that same request.
2. Work out which one it actually was. Most people find they never checked.
3. Ask "what made this a priority now?" on the next request you receive.

Common Mistakes

- Prescribing before diagnosing
- Accepting the presented problem at face value
- Questioning everything, including obvious small requests
- Discovering the real problem and presenting it as though the requester was foolish

AI Prompts

- "Someone has asked for [request]. List six different underlying problems that could produce that same request, and the question I would ask to distinguish between them."
- "Here is a stated requirement. Help me identify what it assumes about the solution before we have diagnosed the problem."
- "I found out the real issue is different from the request. Help me phrase that so it lands as insight rather than as telling them they were wrong."

Books

- *The Trusted Advisor* — Maister, Green, Galford. The standard text, and genuinely good.
- *Flawless Consulting* — Peter Block
- *Never Split the Difference* — Chris Voss

Podcast Episodes

Wake Up X episodes with consultants and solution engineers. Listen for how they describe the first conversation with a client — the good ones spend it almost entirely asking.

Checklist

- I ask what made this a priority now
- I match diagnostic effort to what is riding on it
- I earn the right to disagree before disagreeing
- I recognize when someone needs permission rather than an answer

Key Takeaways

- The presented problem is rarely the real problem
- Clients buy certainty, risk transfer, speed, and permission
- Blame for a failed implementation lands on the implementer, not the diagnosis
- "What made this a priority now?" does more work than any other question

Next Mission

The five rungs, in order, none skipped.

The Discovery Ladder

MISSION

To give you a repeatable sequence for any engagement, internal or external.

Objectives

- Run all five rungs in order
- Understand what failure looks like at each rung
- Surface objections before they go quiet

Framework: The Five Rungs



Skip a rung and the engagement fails there, not where you notice it.

The Discovery Ladder. Skip a rung and the engagement fails there, not where you notice it.

1 — Verify. Confirm the facts of the situation as stated. What is actually in place today? Not what the documentation says, not what they believe, not what the org chart implies — what is true.

This rung alone frequently changes the entire engagement. The backup that everyone believes runs nightly has been failing for four months. The process everyone describes is not the process anyone follows. The system that "everyone uses" has eleven active users.

Failure at this rung looks like: building a solution for an environment that does not exist.

2 — Anchor. Establish what "good" looks like to *them*, in their terms, and what it is worth.

Without an anchor you have no way to evaluate any option and no way to know when you are done. "Better security" is not an anchor. "We can answer a customer questionnaire without three days of scrambling" is an anchor.

Failure at this rung looks like: an engagement that never ends, because nobody defined completion.

3 — Lay out. Present the options plainly, including doing nothing. Three is the right number. Give honest trade-offs for each, including the trade-offs of the one you prefer.

Book 4's one-page proposal is the artifact for this rung.

Failure at this rung looks like: being treated as an advocate for your preferred solution rather than as an advisor.

4 — Uncover. Surface objections *before* they become quiet resistance. Ask directly: "What is the part of this you are least comfortable with?"

An unspoken objection kills implementations months later, and by then nobody remembers it was never addressed. It presents as slow adoption, missed meetings, a team that "hasn't had time" — never as the original objection.

Failure at this rung looks like: a project that stalls for reasons nobody can name.

5 — End. Close with a specific next step, a named owner, and a date. Not "let me know." An engagement without a named owner is a document that gets filed.

Failure at this rung looks like: a good recommendation that nothing happened about.

Professional Explanation

Why the Order Cannot Change

Each rung depends on the one below it.

You cannot anchor on "good" until you know what is actually in place, because the gap between current and good is the whole engagement. You cannot lay out options until you know what good looks like, or you are proposing against your own standard rather than theirs.

You cannot uncover objections to options you have not presented. And you cannot close on something nobody has objected to yet, because the objection will simply arrive later, silently.

People skip rung 1 most often, because it is tedious and because asking "is that actually true?" feels like doubting the client. It is the rung that most changes outcomes.

People skip rung 4 most often when things are going well — the meeting is positive, everyone is agreeable, and asking for objections feels like inviting trouble. That is exactly when it matters.

How to Verify Without Being Insulting

The framing that works: verification is about *you*, not about them.

- "Before I recommend anything I'd like to see the current setup myself — I've been wrong often enough about what I assumed was in place."
- "Can I look at the actual configuration? I want to make sure I'm not building on an assumption."
- "Would it be possible to see the last three runs of that report?"

Each of those attributes the caution to you rather than implying they are unreliable. Nobody objects, and you find out what is actually true.

Uncovering Objections Well

The question matters. "Any concerns?" gets "no." The questions that work assume there is a concern and make it easy to say:

- "What's the part of this you're least comfortable with?"
- "If this goes wrong in six months, what will have gone wrong?"
- "Who on your team is going to hate this, and why?"
- "What haven't I asked about that I should have?"

That last one is the most productive question in this book. Ask it at the end of every discovery conversation.

Step-by-Step Guide: Running the Ladder

1. **Before the first meeting:** write down what you think you already know, so you can check it.
2. **Verify:** see the actual thing. Configurations, logs, reports, the process being performed.
3. **Anchor:** ask what good looks like, in their words, and what it is worth to them. Write down their exact phrasing.
4. **Lay out:** one page, three options, do-nothing included, honest trade-offs.
5. **Uncover:** ask the objection questions directly. Wait through the silence.
6. **End:** specific next step, named owner, date. Send it in writing the same day.
7. **After:** confirm in writing what was agreed, per Book 4's documentation reflex.

Real Example

An organization asks for help improving their vulnerability management.

Rung 1 — Verify. The scanner runs weekly and produces reports. Verification reveals the reports go to a shared mailbox nobody monitors, and that the scan covers roughly 60% of the estate because credentials expired on two subnets eight months ago.

Rung 2 — Anchor. "Good" turns out not to be "fewer vulnerabilities" but "being able to tell our largest customer, truthfully, that we patch critical issues within thirty days." That is a specific, measurable, contractually-driven anchor.

Rung 3 — Lay out. Three options: fix the coverage gap and assign ownership of the report. The same plus a ticketing integration so remediation is tracked; a full managed service. Plus do-nothing, with the honest consequence — the customer question cannot be answered truthfully.

Rung 4 — Uncover. The objection, surfaced by asking, is that the infrastructure team believes this will generate work they have no capacity for. Real, legitimate, and it would have killed the project silently.

Rung 5 — End. Option two, owner named on both sides, first review in thirty days, and an explicit agreement that only criticals are ticketed initially so the infrastructure team's concern is addressed.

Note that rung 4 changed the solution. That is what it is for.

Case Study: The Objection That Was Never Raised

A pattern that explains most stalled projects.

An engagement concludes with agreement. Everyone is positive. The recommendation is accepted.

Then implementation slows. Meetings are rescheduled. The team "hasn't had a chance." Six months later, nothing has happened, and the reasons given are all logistical.

The actual reason was never stated: someone in the room believed this would make their job harder, or expose something, or take away something they controlled. They did not say so, because the meeting was going well and objecting felt disruptive.

Rung 4 exists entirely to prevent this. And the reason it works is not that people are secretive. It is that a positive meeting creates pressure not to disagree, and you have to explicitly remove that pressure by asking.

Templates: The Discovery Questions

VERIFY

Can I see the current setup myself?
When did this last work as intended?
Who actually performs this today?

ANCHOR

What does "good" look like, specifically?
How would you know this had worked?
What's it worth to you to fix?

LAY OUT

[Book 4's one-page proposal – three options + do nothing]

UNCOVER

What's the part of this you're least comfortable with?
If this goes wrong in six months, what went wrong?
Who on your team is going to hate this?
What haven't I asked about that I should have?

END

Who owns the next step, and by when?
Who else needs to be comfortable before this moves?

Reflection

- Which rung do you skip most often?
- Have you ever built something for an environment that turned out not to exist?
- When did you last ask someone what they were least comfortable with?

Journal

Write about a project that stalled without explanation. What objection do you think was never voiced?

Exercises

1. Run rung 1 properly on something you currently assume you understand. Verify it.
2. Ask "what haven't I asked about that I should have?" at the end of your next work conversation.

3. Write down what "good" looks like for a current project, in the stakeholder's words rather than yours.

Project

Run the full ladder on one real piece of work — internal or external — and document what each rung surfaced. Most people find that rung 1 and rung 4 produce information they would otherwise never have had.

AI Prompts

- "I am about to start [engagement]. Give me verification questions that check what is actually in place, phrased so they do not imply I distrust the client."
- "Here is what the stakeholder said 'good' looks like: [quote]. Is that a usable anchor, or is it too vague to evaluate options against?"
- "Give me five ways to ask for objections that assume there is one, rather than 'any concerns?'"

Books

- *The Trusted Advisor* — Maister, Green, Galford
- *Flawless Consulting* — Peter Block
- *SPIN Selling* — Neil Rackham

Checklist

- I verify what is actually in place before proposing
- I capture the anchor in their words, not mine
- I present three options including do nothing
- I ask for objections in a way that assumes there is one
- Every engagement ends with a named owner and a date

Common Mistakes

- Skipping verification because it feels like doubting the client
- Vague anchors that make completion undefinable
- One option, which makes you an advocate
- Not asking for objections when the meeting is going well
- Ending without a named owner

Key Takeaways

- Verify, Anchor, Lay out, Uncover, End — in order
- Rung 1 changes engagements more than any other
- Unspoken objections present as logistics, not as objections
- "What haven't I asked about?" is the highest-return question available

PART

TWO

The People Around the Work

Map the Room

MISSION

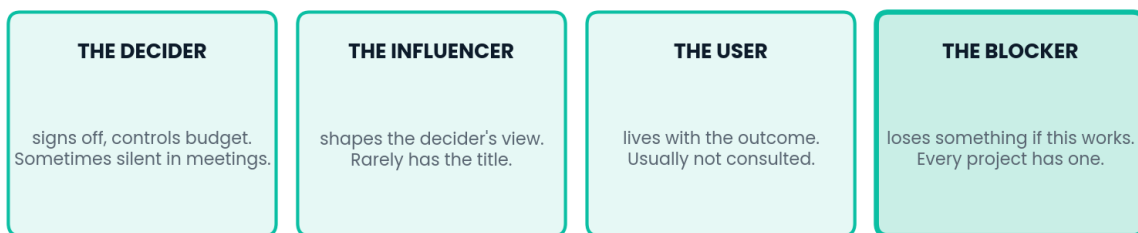
To identify who actually decides, before you spend effort persuading someone who cannot.

Objectives

- Identify the four roles in any decision
- Engage the blocker rather than routing around them
- Find the decider when the org chart is misleading

Framework: The Four Roles

MAP THE ROOM BEFORE YOU PROPOSE



"Who else needs to be comfortable with this before it moves forward?"

Four roles, and frequently not the people you would guess.

The decider. Signs off, controls budget. Sometimes silent in meetings, sometimes not in the room at all. The org chart is a poor guide here — budget authority and reporting lines diverge constantly.

The influencer. Shapes the decider's opinion. Often more important than the decider and almost never has the title. The long-tenured engineer whose judgment the executive trusts. The chief of staff. The person the decider calls afterward.

The user. Lives with the outcome. Usually not consulted, and the most common reason implementations fail after go-live. A solution the users hate does not survive, regardless of who approved it.

The blocker. Loses something if this succeeds — control, headcount, relevance, a budget line, a reason to exist. Every project has one.

On Blockers

The instinct is to route around them. Do not.

A blocker who is routed around does not disappear. They surface later, at the worst moment, with more force and more legitimacy — because now they can accurately say they were excluded.

The alternative is to engage them honestly and early. Frequently the blocker's concern is real and you had not considered it. Sometimes it is purely self-interested and cannot be resolved, but knowing that changes how you sequence the work and who you need on side.

The conversation that works: *"You know this area better than I do, and I suspect there's something about this that's going to be a problem. I'd rather hear it from you now than find out in three months."*

Most people respond to that. Being asked is different from being managed.

Professional Explanation

Finding the Decider

One question does most of the work: **"Who else needs to be comfortable with this before it moves forward?"**

It is easy to answer, it does not imply the person you are asking lacks authority, and it reliably surfaces names you did not have.

Follow-ups that help:

- "Who signed off on the last thing like this?"
- "If this needs budget, whose budget is it?"
- "Is there anyone who'd be annoyed to hear about this after the fact?"

That last one finds blockers and influencers simultaneously.

The User Problem

Users are the most consistently ignored role and the most consistently fatal.

A tool that is approved, funded, implemented, and hated will be worked around. Book 3's example about aggressive email filtering is exactly this pattern: the control existed, the users routed around it, and the organization ended up worse off.

Talk to the people who will live with the outcome. Ten minutes each, three of them. Ask what their day looks like now and what would make it worse. This is rung 1 of the ladder applied to people rather than systems.

When You Are Internal

All of this applies inside an organization, with one difference: you cannot walk away, and you have to work with these people afterward.

That makes the blocker conversation more important, not less. An external consultant who alienates a blocker leaves at the end of the engagement. You do not.

Real Example

A proposal to consolidate two ticketing systems into one.

The org chart says: the IT director decides.

Reality: the IT director will approve whatever the operations manager supports, because the operations manager has been there eleven years and the director has been there eight months. The operations manager is the influencer.

The users are two support teams, one of whom built extensive custom workflows in the system being retired. They have not been asked.

The blocker is the manager of that second team, who correctly anticipates losing workflows their team depends on, and who was not consulted.

A proposal that goes to the director alone gets approved and then fails during implementation. A proposal that engages the operations manager, talks to both support teams, and addresses the workflow question directly with the second manager gets implemented.

Same proposal. The difference is entirely who was mapped.

Case Study: The Silent Decider

Worth recognizing because it is common and it wastes enormous effort.

Someone spends weeks building a relationship and a case with the person who called the meeting, presents well, and gets enthusiastic agreement. Then nothing happens.

The person who called the meeting was not the decider. They were gathering information for someone else, who has never met you, has not seen your reasoning, and received a second-hand summary that lost most of it.

The fix is one question at the start, asked casually: "Just so I pitch this right — who ultimately makes the call on this?" Nobody is offended by it, and it saves weeks.

Templates: The Room Map

ENGAGEMENT:

DECIDER: name, and how I know
INFLUENCER: who the decider listens to
USERS: who lives with the outcome; have I spoken to them?
BLOCKER: who loses something; what specifically?
CHAMPION: who wants this to happen and why

QUESTIONS ASKED:

Who else needs to be comfortable with this?
Who signed off on the last thing like this?
Whose budget is it?
Who'd be annoyed to hear about it afterwards?

Reflection

- In your current project, who is the blocker? (*If you cannot name one, you have not found them.*)
- Have you spoken to anyone who will actually use what you are building?
- Do you know who decides, or do you know who called the meeting?

Journal

Write about a project that was approved and then failed in implementation. Which of the four roles was not mapped?

Exercises

1. Map the four roles for a current piece of work. Write down names.
2. Ask "who else needs to be comfortable with this?" this week.
3. Talk to one person who will live with something you are building.

AI Prompts

- "I am proposing [change] at a [size] organization. Who typically holds each of the four roles — decider, influencer, user, blocker — for this kind of decision?"

- "Help me phrase an approach to someone who is likely to oppose my proposal, in a way that invites their concern rather than defending against it."
- "What questions surface a decision-maker without implying the person I am talking to lacks authority?"

Books

- *The Challenger Sale* — Dixon and Adamson. On mobilizers and stakeholder mapping specifically.
- *Flawless Consulting* — Peter Block
- *Crucial Conversations* — Patterson et al.

Checklist

- I know who decides, who influences, who uses, and who blocks
- I have engaged the blocker directly rather than routing around them
- I have spoken to at least one user
- I asked who else needs to be comfortable before moving forward

Common Mistakes

- Confusing the meeting-caller with the decider
- Routing around blockers
- Never talking to users
- Assuming the org chart reflects decision authority

Key Takeaways

- Four roles: decider, influencer, user, blocker — every project has all four
- The influencer rarely has the title
- Routing around a blocker guarantees they surface later with more force
- A solution users hate does not survive approval

Next Mission

Protecting the work from the thing that most reliably destroys it.

Scope, and Saying No

MISSION

To control scope without becoming difficult, and to say no in a way that strengthens the relationship.

Objectives

- Make the cost of additions visible rather than refusing them
- Use the three forms of no
- Understand why the reluctant yes is the most expensive thing available

Why It Matters

Scope creep is not caused by bad clients. It is caused by undocumented agreements. Almost always.

The mechanism is mundane: someone asks for a small addition in a hallway or at the end of a call, you say yes because it is small and you want to be helpful, and no record exists. Repeat eleven times and the project is forty percent larger with the original timeline and budget, and everyone is unhappy. Including the client, who does not experience it as having asked for more.

Framework: The One Sentence

The fix is one sentence, used consistently:

"Yes, we can do that — here's what it does to the timeline. Want me to write it up?"

You have not said no. You have made the cost visible and offered to put it in writing.

Three things happen. Most requests evaporate right there, because the person did not know it was significant and does not want it that much. The ones that survive were genuinely worth doing. And a record exists, which is Book 4's documentation reflex applied to scope.

Note what this sentence is not: it is not "that's out of scope." That phrasing is technically correct and it makes you the obstacle. The version above makes the timeline the obstacle, which is accurate and impersonal.

Professional Explanation

The Three Forms of No

In order of preference.

1 — The trade. *"I can do that if we move X. Which matters more?"*

This puts the decision where it belongs — with the person who owns the priorities. It is not a refusal at all; it is a request for a decision. Most scope conversations should end here.

2 — The alternative. *"I wouldn't recommend that approach, and here's why. Here's what I'd do instead."*

A no with a path forward is not experienced as a no. This is also where your expertise is most visible, because declining something and offering something better demonstrates judgment in a way that agreement never does.

3 — The plain no. *"That's not something I can do well, and I'd rather tell you now than deliver something poor."*

Rare, and it earns more respect than a reluctant yes ever does. People remember who told them the truth about their own limits.

The reluctant yes is the most expensive thing in professional life. It costs you the work, the relationship, and your reputation — usually all three, and usually at once.

Why the Reluctant Yes Is So Costly

You agreed to something you knew you could not do well, or did not have capacity for, or did not believe in.

Then one of three things happens: you deliver it badly, you deliver it late, or you deliver it at the expense of something else you had committed to. All three damage you more than the original no would have.

And the person who asked would nearly always have preferred the honest answer. They asked because they wanted the problem solved, not because they wanted your agreement.

The reason people give the reluctant yes anyway is that the discomfort of saying no is immediate and the cost of saying yes is deferred. Recognizing that asymmetry is most of the fix.

On Being Internal

Everything here applies inside an organization, where scope creep is called "sure, I can look at that" and where nobody writes anything down.

The same sentence works: *"Yes — that'll push [other thing] out by about a week. Which would you rather I do first?"* Said to a manager, it is not obstruction; it is asking them to do their job, which is prioritization.

Most managers appreciate it. The ones who do not are telling you something useful about the environment, per Book 9.

Step-by-Step Guide: Handling an Addition

1. **Do not answer immediately** if it is more than trivial. "Let me think about what that involves" is a complete response.
2. **Estimate honestly**, including the parts that are not obvious.
3. **Use the sentence.** Yes, here is the cost, shall I write it up?
4. **Write it up** if they say yes — even three lines in an email.
5. **If it is genuinely small**, absorb it and say so explicitly: "That's small enough that I'll just do it — noting it so we both remember it happened." This builds enormous goodwill and it keeps the record intact.
6. **Track the cumulative total.** Eleven small absorptions is not small.

That fifth step matters. Being rigid about genuinely trivial things makes you tedious. The point is not to charge for everything; it is that nothing happens invisibly.

Real Example

Mid-project, a stakeholder asks for an additional report "while you're in there."

The costly version: "Sure, no problem." It takes two days. Nobody records it. Three more requests follow the same pattern. The project runs late and the delay is attributed to poor estimation.

The effective version: "Yes, that's doable — it's about two days, which would move the delivery to the following week. Want me to write it up as an addition, or would you rather we do it after the main delivery?"

Frequently the answer is "after." The request was real but not urgent, and the person had no idea it was two days.

Neither response refused anything. One protected the project.

Case Study: The Client Who Was Not Difficult

Worth reframing, because "difficult client" is often a misdiagnosis.

A project ends badly. The consultant's account: the client kept changing the scope and was impossible to satisfy.

The client's account: they asked for a few small things, were told yes each time, and then the project was late and the invoice was higher than expected, and nobody warned them.

Both accounts are accurate. Neither party behaved badly. The failure was that no mechanism existed to make cost visible at the moment of the request.

This is why the one sentence matters more than any amount of goodwill. Goodwill does not survive a surprise invoice; a written record prevents one.

Templates: The Scope Addition Note

Subject: Addition to [project] – [thing]

As discussed, adding [thing].

WHAT IT INVOLVES: briefly

IMPACT ON TIMELINE: specific

IMPACT ON COST: specific, or "absorbed"

REVISED DELIVERY: date

Confirming so we both have the same picture – let me know if that's not right.

Templates: The Three Nos

THE TRADE

"I can do that if we move [X]. Which matters more?"

THE ALTERNATIVE

"I wouldn't recommend that, and here's why.

Here's what I'd do instead."

THE PLAIN NO

"That's not something I can do well, and I'd rather tell you now than deliver something poor."

Reflection

- When did you last give a reluctant yes? What did it cost?

- Do you have a record of the additions on your current work?
- Is your instinct to refuse or to make the cost visible?

Journal

Write about a time you said yes when you should have said no. What made the no feel impossible in the moment?

Exercises

1. Practise the one sentence out loud until it is neutral rather than defensive.
2. Track every addition to your current work for two weeks, including absorbed ones.
3. Use the trade form once this week.

AI Prompts

- "Someone asked me to add [thing] to a project. Help me phrase a response that says yes, makes the cost visible, and offers to document it."
- "I need to decline [request] without damaging the relationship. Give me the trade version, the alternative version, and the plain version."
- "Here is a scope addition. Draft a three-line confirming note covering what it involves and the timeline impact."

Books

- *Never Split the Difference* — Chris Voss
- *The Trusted Advisor* — Maister, Green, Galford
- *Essentialism* — Greg McKeown. On the cost of the reluctant yes, at length.

Checklist

- Every addition gets written up or explicitly absorbed
- I make cost visible rather than refusing
- I use the trade before the plain no
- I track cumulative absorptions
- I have not given a reluctant yes this month

Common Mistakes

- "That's out of scope," which makes you the obstacle
- Agreeing to additions verbally
- Being rigid about genuinely trivial requests
- Absorbing eleven small things and calling each one small
- The reluctant yes

Key Takeaways

- Scope creep is an undocumented-agreement problem, not a bad-client problem
- "Yes — here's what it does to the timeline. Shall I write it up?"
- Trade, alternative, plain no — in that order
- The reluctant yes costs the work, the relationship, and the reputation

PART

THREE

Running the Work

The Engagement Rhythm

MISSION

To run work in a shape that produces repeat business and no surprises.

Objectives

- Structure any engagement, whatever its length
- Communicate on a cadence rather than on news
- Hand over properly rather than building dependency

Framework: The Five Phases



The last one costs an email and generates more repeat work than anything else.

The engagement rhythm. The last phase costs an email and generates more repeat work than anything else.

Kickoff. Agree the objective, the scope, the decision-makers, and how you will communicate. Write it down and send it. This document prevents most later disputes, and it takes twenty minutes.

Updates on a fixed cadence. Even when there is nothing to report. **Especially** then — silence is where clients invent problems. An update that says "on track, nothing to flag, next update Friday" takes ninety seconds and prevents a phone call you do not want.

A visible mid-point check. Are we still solving the right problem? Give them a formal opportunity to redirect. They will respect it, and it prevents a much worse conversation at the end.

Handover. Documentation, training, and a named owner on their side. Work that depends on you permanently is a failure, not job security.

Follow-up after the close. Thirty to sixty days later, unprompted. This costs an email and generates more repeat work than anything else you can do.

Professional Explanation

Why Cadence Beats News

Most people update when there is something to say. This feels efficient and it is the wrong model.

From the other side, silence is ambiguous. It could mean everything is fine, or that something has gone wrong and you are avoiding the conversation. People reliably assume the second, because the second is what they have experienced before.

A fixed cadence removes the ambiguity entirely. They know when they will hear from you, so silence between updates carries no information and generates no anxiety.

This applies internally with equal force. A manager who hears from you on a schedule stops checking in, which is what both of you want.

On Handover and Dependency

There is a temptation, particularly for consultants and for internal specialists, to be the only person who understands something. It feels like security.

It is not. It is a ceiling.

If you are the only person who can operate what you built, you cannot be moved to anything more interesting, you cannot take leave without anxiety, and you have made yourself a risk that someone will eventually decide to mitigate. Book 5's argument about being more valuable than your title depends on being *replaceable in your current title*.

Handover done properly: documentation someone can follow, a training session, a named owner, and a period where they do it while you watch rather than the reverse.

The Follow-Up

Thirty to sixty days after the work ends, send a short unprompted message. Not a sales message.

"Hi [name] — it's been about six weeks since we wrapped up [project]. How's it holding up? Anything behaving differently than expected?"

Three things come from this. You find out whether it actually worked, which is information you would otherwise never get. You are remembered as someone who cares about outcomes

rather than invoices. And with real frequency, they mention a new problem.

Almost nobody does this. It costs two minutes.

Step-by-Step Guide: The Kickoff Document

Twenty minutes, sent the same day.

OBJECTIVE	what success looks like, in their words
IN SCOPE	specifically
OUT OF SCOPE	specifically – prevents most disputes
DECISION MAKER	name
OTHER STAKEHOLDERS	the room map
KEY RISKS	what could derail this
CADENCE	how and when we communicate
DEFINITION OF DONE	

The out-of-scope section is the one people omit and the one that does the most work. Naming what you are *not* doing is not defensive; it is the clearest possible statement of what you are doing.

Real Example

Two engagements of similar size.

The first: kickoff by conversation, no document. Updates when there is news. That means three weeks of silence in the middle. No mid-point check.

At the end, the client says it is not quite what they expected, and there is no shared record of what was agreed. It resolves awkwardly and there is no repeat work.

The second: kickoff document sent day one. Weekly update every Friday, including two that said "on track, nothing to flag." A mid-point conversation where the client redirects one element. Small, easily accommodated at week four, expensive at week ten. Handover with documentation and a named owner. Follow-up email at week six after close.

The second engagement produced two more pieces of work over the following year. Not because the technical work was better — it was comparable. Because the experience of working with her was predictable.

Case Study: The Mid-Point Check

Why this phase earns its place, when it looks like an optional courtesy.

Requirements drift, quietly. What the client wanted at kickoff and what they want at week six are frequently different, because they have learned things during the engagement. Often from your own early work.

Without a formal opportunity to redirect, they will not raise it. Raising it feels like criticism, or like changing the deal, or like admitting they did not know what they wanted.

The mid-point check gives explicit permission: *"Now that we're halfway, I want to check we're still solving the right problem. Has anything changed in what would make this a success?"*

A redirect at the mid-point is a small adjustment. The same redirect at the end is a failed engagement. The phase costs thirty minutes and it is the highest-value meeting in the whole rhythm.

Templates: The Cadence Update

```
[Project] – update [date]

STATUS:           on track / at risk / blocked
DONE THIS WEEK:
NEXT WEEK:
ANYTHING I NEED FROM YOU:
FLAGS:            or "nothing"
NEXT UPDATE:     date
```

Ninety seconds. Send it even when it is boring.

Templates: The Follow-Up

```
Hi [name] – it's been about [six weeks] since we wrapped
up [project]. How's it holding up? Anything behaving
differently than expected?
```

```
No agenda – just curious whether it's working the way
we intended.
```

Reflection

- Do you update on a cadence or on news?
- Is there something only you can operate? Is that security or a ceiling?
- When did you last follow up on finished work, unprompted?

Journal

Write about a piece of work you finished and never checked on. What would you find if you asked?

Exercises

1. Write a kickoff document for your current work, even mid-project.
2. Set a fixed update cadence and hold it for four weeks, including boring updates.
3. Follow up on one finished piece of work, unprompted, this week.

Project

Run one complete engagement through all five phases with documentation at each. Then note which phase produced the most valuable information — for most people it is the mid-point check or the follow-up.

AI Prompts

- "Draft a kickoff document for this engagement: [description]. Be specific about the out-of-scope section."
- "Here is my weekly update draft. Is there anything ambiguous that a client could interpret as bad news being hidden?"
- "Help me phrase a mid-point check that gives the client permission to redirect without it feeling like criticism of the work so far."

Books

- *The Trusted Advisor* — Maister, Green, Galford
- *Flawless Consulting* — Peter Block
- *The Phoenix Project* — Gene Kim

Checklist

- Every engagement starts with a written kickoff
- I update on a fixed cadence, including when there is no news
- I run a formal mid-point check
- I hand over to a named owner with documentation
- I follow up thirty to sixty days after close

Common Mistakes

- Updating on news rather than cadence
- No out-of-scope section
- Skipping the mid-point check as a courtesy nobody needs
- Building dependency and calling it job security
- Never following up

Key Takeaways

- Silence is where clients invent problems
- A redirect at the mid-point is small; the same redirect at the end is a failure
- Work that depends on you permanently is a ceiling, not security
- The unprompted follow-up costs two minutes and generates more repeat work than anything else

Pricing, Positioning, and Knowing Your Worth

MISSION

To help you value your work accurately, whether you invoice for it or not.

Objectives

- Price against value rather than hours where possible
- Recognize when a client or employer is not worth having
- Get everything in writing, every time

Why It Matters

This chapter applies to independent consultants directly and to everyone else by analogy. The internal equivalent of underpricing is taking on work that does not advance you, for people who do not value it, because saying no felt impossible.

The mechanics are the same.

Professional Explanation

Hours Versus Value

Charging by the hour has one structural problem: it caps your income at your available time, and it penalizes you for getting faster. The better you get, the less you earn for the same outcome.

Value-based pricing asks a different question: what is solving this worth to the client? A week of work that prevents a recurring six-figure problem is not priced by the week.

This needs rung 2 of the Discovery Ladder. If you did not establish what good is worth to them, you have no basis for value pricing and you will default to hours.

For most people the practical path is hybrid: hourly or day-rate for open-ended and advisory work, fixed price for well-defined deliverables where you have done the discovery properly.

The Underpricing Trap

People starting out underprice for understandable reasons: uncertainty, wanting the work, imposter feelings.

The problem is that price is a signal. Very low pricing does not read as good value; it reads as inexperience or desperation. It attracts clients who are optimizing for cheapness — who are, reliably, the most difficult clients you will have.

Raising prices later with the same client is also much harder than starting correctly.

If you are uncertain, the safer error is to price at market and be excellent, rather than to price low and hope volume compensates.

When to Walk Away

Some work is not worth having at any price. The signals:

- **Resistance to documentation.** The single clearest signal there is.
- **No named decision-maker,** or a decision-maker who will not meet you.
- **Unwillingness to give you access** to verify what is actually in place — rung 1 blocked.
- **A history of disputes** with previous providers, described as entirely the providers' fault.
- **Pressure to start before terms are agreed.**
- **Scope that keeps expanding during the sales conversation,** before any agreement exists. It will not improve afterwards.

I once walked away from an opportunity worth roughly a million dollars because the other side — with a lawyer sitting right there — refused to put anything in a signed contract and wanted a verbal agreement using my own money.

That decision took about four seconds. It is also a large part of why I moved toward compliance and GRC work: I had learned firsthand what happens when nothing is written down.

The structural point is worth extracting from the story. A legitimate counterparty has no reason to refuse documentation — writing things down protects both sides equally, and prevents honest misunderstanding as much as dishonesty. Therefore refusal is not a neutral preference. It means either they intend to behave differently from what was said, or they want the option to. **When someone resists documentation, the resistance itself is the information.**

The presence of a lawyer made it worse, not better. It removed any possibility that this was naivety. Whatever scale you encounter this at — a client, an employer, a partner, a handshake agreement about equity — the signal reads the same.

Positioning

Positioning is what you are known for. It is mostly a Book 7 problem.

The useful principle: **specific beats general, and a narrow reputation travels further than a broad one.** "Security consultant" competes with everyone. "The person who helps mid-size firms get through their first SOC 2" is memorable, easy to refer, and answers a question someone is actually asking.

You can broaden later. Starting broad means starting invisible.

Framework: The Pricing Conversation

1. **Do the discovery first.** Never price before rung 2. You are guessing otherwise.
2. **Anchor on their value, not your effort.** "You said this is costing about [X] per month" is the anchor.
3. **Present options at different price points** — Book 4's three options, which also lets them choose rather than accept or reject.
4. **Say the number and stop talking.** The silence after a price is uncomfortable and filling it costs money.
5. **Do not discount reflexively.** If you must move, take something out rather than lowering the price for the same scope — otherwise your original number was fiction.
6. **Get it in writing.** Every time, at every scale.

Real Example

Two ways of answering "what would this cost?"

Weak: "It'd probably be about ten days, so around [figure], but it depends on a few things."

Better: "Based on what you described — the three days a month your team spends on this and the audit finding you need to close — I'd approach it three ways. [Option A] addresses the immediate finding, roughly [figure]. [Option B] also fixes the recurring time cost, roughly [figure]. [Option C] is the full rebuild, roughly [figure], and I don't think you need it yet. I'd recommend B."

The second version anchors on their stated cost, offers a choice, recommends clearly, and rules out the most expensive option — which is more persuasive than any argument for it would have been.

Case Study: The Cheapest Client

A pattern almost everyone who has worked independently recognizes.

The client who negotiated hardest on price is the one who requests the most additions, questions the most decisions, is slowest to pay, and is least satisfied at the end.

This is not coincidence. Someone optimizing primarily for cost is, by definition, not primarily valuing the outcome — and a client who does not value the outcome cannot be satisfied by achieving it.

The practical consequence: pricing is a filter as much as a number. Pricing appropriately removes a category of client you do not want, which is worth more than the revenue you decline.

Reflection

- Are you pricing your time or your outcomes?
- Have you ever taken work you knew you should not have?
- What would you need to see to walk away from something lucrative?

Journal

Write about work you took that you should have declined. What was the signal you noticed and ignored?

Exercises

1. Write down your walk-away signals before you need them.
2. Practise saying a price and then stopping. Out loud.
3. Write your positioning in one sentence: who you help, with what, specifically.

AI Prompts

- "Help me structure three pricing options for this engagement: [description and what the client said it is costing them]. Anchor on their stated cost, not my hours."
- "Here is my positioning statement. Is it specific enough to be easy to refer, or does it describe everyone in my field?"
- "What contract terms should be in writing before I start work of this type?"

Books

- *Never Split the Difference* — Chris Voss
- *The Win Without Pitching Manifesto* — Blair Enns. Short, sharp, and specifically about not competing on price.

Checklist

- I do discovery before pricing
- I anchor on their value, not my hours
- I present options at different levels
- I have written walk-away signals
- Everything is in writing before work starts

Common Mistakes

- Pricing before discovery
- Underpricing as a strategy, which attracts the worst clients
- Discounting the price without removing scope
- Filling the silence after saying a number
- Starting before terms are agreed

Key Takeaways

- Value pricing needs rung 2; without it you default to hours
- Price is a signal and a filter, not only a number
- Resistance to documentation is the information
- Specific positioning travels further than broad positioning

Next Mission

Book 9: the long game, and how not to stop playing it.

Appendix: Kickoffs, Scripts, and the Ladder

REFERENCE

Everything from this book in one place.

The Discovery Ladder

- 1 VERIFY what is actually in place today
- 2 ANCHOR what does "good" look like to them
- 3 LAY OUT three options, including do nothing
- 4 UNCOVER surface objections before they go quiet
- 5 END a specific next step, a named owner, a date

Skip a rung and the engagement fails there, not where you notice it.

The Discovery Questions

VERIFY

Can I see the current setup myself?
When did this last work as intended?
Who actually performs this today?

ANCHOR

What does "good" look like, specifically?
How would you know this had worked?
What's it worth to you to fix?

UNCOVER

What's the part of this you're least comfortable with?
If this goes wrong in six months, what went wrong?
Who on your team is going to hate this?
What haven't I asked about that I should have?

END

Who owns the next step, and by when?
Who else needs to be comfortable before this moves?

The Priority Question

"What made this a priority now?"

Does more diagnostic work than any other question.

The Room Map

DECIDER: signs off, controls budget
INFLUENCER: shapes the decider – rarely has the title
USER: lives with the outcome – usually not consulted
BLOCKER: loses something if this works – engage, don't route around

"Who else needs to be comfortable with this before it moves forward?"

The Scope Sentence

"Yes, we can do that – here's what it does to the timeline. Want me to write it up?"

The Three Nos

THE TRADE "I can do that if we move [X].
Which matters more?"
THE ALTERNATIVE "I wouldn't recommend that. Here's
what I'd do instead."
THE PLAIN NO "That's not something I can do well,
and I'd rather tell you now."

The Kickoff Document

```
OBJECTIVE          success, in their words
IN SCOPE           specifically
OUT OF SCOPE       specifically – does the most work
DECISION MAKER     name
OTHER STAKEHOLDERS the room map
KEY RISKS
CADENCE
DEFINITION OF DONE
```

The Cadence Update

```
STATUS:           on track / at risk / blocked
DONE THIS WEEK:
NEXT WEEK:
ANYTHING I NEED FROM YOU:
FLAGS:            or "nothing"
NEXT UPDATE:      date
```

Send it even when it is boring.

The Engagement Rhythm

```
KICKOFF  ->  UPDATES  ->  MID-POINT  ->  HANDOVER  ->  FOLLOW-UP
                                     (redirect          (30-60 days,
                                     cheaply)           unprompted)
```

Walk-Away Signals

```
resistance to documentation    <- the clearest one
no named decision-maker
won't give access to verify
blames every previous provider
pressure to start before terms
scope expanding during the sales conversation
```

A Closing Note

The shift this book describes is small to state and hard to make: stop being the person who does what was asked. Become the person who finds out what is actually needed.

It is uncomfortable at first, because doing what was asked is safe. Nobody blames you for delivering the request. Except that they do — when the request was wrong and the implementation fails, the blame lands on the implementation, and that is you.

I learned this repairing phones, where the customer's diagnosis was frequently wrong and doing the wrong job well cost me the relationship.

I learned it again running a business, where the most expensive decisions were the ones I made without asking enough questions. And I learned it a third time walking away from a million dollars because somebody would not write anything down.

None of that required expertise I did not already have. It required asking one more question than felt necessary, and being willing to hear an answer I did not want.

*Verify before you propose. Ask what they're least comfortable with.
Write it down.*

Connect

[linkedin.com/in/efrenlsalazar](https://www.linkedin.com/in/efrenlsalazar) | wakeupxpodcast.com

Questions for the author. Two gaps, both from your consulting and solution-engineering work: (1) **a real engagement where the presented problem turned out not to be the real one** — the phone-repair analogy carries Chapter 1, but a professional example would land harder with the audience this book is for; (2) **a time you said no and it strengthened the relationship**, which Chapter 4 currently argues without showing. Also still open: whether The Discovery Ladder keeps its name or reverts to V.A.L.U.E. — noted in the front matter, and a one-line decision to make.

Asking Better Questions

Whether it's a podcast guest or a room full of stakeholders, the consultant mindset starts the same way: listen first.



Podcast guest promo — Tolgay Kizilelma, PhD



On-stage interview, Museum of Science and Curiosity

Greg McCord — The Fractional CISO Case

Greg McCord is CISO at Lightcast and founder of McCord Keystone Advisory, where he helps startups who can't afford a full-time CISO build security into their product from the beginning — before they're big enough to be scrambling to fix it later. Met at RSA Conference, his path into the field started nowhere near a keyboard.

From Video Games to Defense

Greg grew up in Indiana and didn't touch a computer until he was an adult. What pulled him in was video games — he loved playing them, then started tearing them apart to understand how they worked. The turning point was a simple realization: if he could do that, somebody else could do the same thing to whatever he built. That's what flipped his curiosity from playing to defending — active duty military and defense contract work in Germany, then Pacific Life leading a GRC program, then building the security program at Lightcast.

Don't Die on the Vine for One Risk

His sharpest advice for anyone aiming at a CISO seat: technical risk language doesn't land with a CEO. When leadership asks "are we secure, are we safe," a specific critical vulnerability isn't the answer they need — you have to translate risk into the business outcome they actually care about, or the conversation goes nowhere.

Watch the full episode: youtu.be/IGDTBvpQ0-8

Greg's Advice, In Practice

Pick your lane before you pick your certs.

Attacker, defender, or audit/compliance — figure out which pulls you first. The cert path should follow that choice, not lead it.

Certs get your foot in the door. What you do with the work gets you the job.

Degrees and certifications open the conversation — they don't close the deal on their own.

Build security in at the beginning, not as a retrofit.

His whole advisory practice exists because fixing security after a product ships to MVP is always harder than building it in from day one.

Take your own mental health seriously.

He's writing a book about his own nonlinear path partly to say this directly: the job is hard, the path is rarely straight, and reaching out for support is not optional.

Most jobs come from people, not postings.

He was found through a shared LinkedIn post from a friend — including how he ended up on this very podcast. His advice: show up at the nonprofit and community group meetings, not just the job boards.

REFLECTION

Are you translating your work into language your stakeholders actually care about, or into language that only makes sense to you? Who could you ask to check?

GUEST SPOTLIGHT

Larry Switzer — Sell the Data, Not the Lock-In

Larry Switzer runs North American sales for Ziggiz, a cyber data platform originally funded by DARPA to help under-resourced hospitals survive AI-era attacks. His clearest value here isn't the product — it's how honestly he talks about hype.

Decouple Your Data From the Vendor

His central pitch to companies: stop buying one platform to cover endpoint, firewall, and SIEM together. "Deal with your data first and then pick what you want." Vendor lock-in isn't inevitable — it's a structural choice most companies default into without realizing there's another way.

Honest Pushback Against AI Hype

Asked directly about fully autonomous, sub-minute-response AI security operations centers, he doesn't hedge: he doesn't believe they exist yet, no matter what's being marketed. "It defies the laws of physics" given how most current tools actually pull and process data. Worth noting because he sells into this exact market — and still says it plainly.

The Real AI Skill Is Knowing What to Ask

His clearest AI observation: two people can use the identical tool on identical data and get wildly different results, purely because of how they framed the question. His conclusion isn't that AI replaces critical thinking — it's that AI makes critical thinking more valuable, not less.

Watch the full episode: youtu.be/f7b5JyffUHS

GUEST SPOTLIGHT

Larry's Advice, In Practice

Watch where the money actually goes, not where the noise is.

His advice for staying current: follow top venture firms and see who they're funding — that tells you more about real direction than press releases do.

Training is not the same as doing.

He's direct about this: get engaged with practitioner communities and get hands-on, because classroom training alone doesn't transfer the same way.

A high failure rate is compatible with being excellent.

"A baseball player can hit .300 and be a superstar. Life is about what you do when you fail."

Legacy companies often can't copy your business model even if they want to.

Changing how you charge customers can cannibalize existing revenue — which is exactly why a new, differently-structured competitor can outmaneuver a much bigger one.

REFLECTION

Where in your own work are you relying on the tool instead of the question you're asking it? What would change if you got sharper about the ask, not the tool?

GUEST SPOTLIGHT

John Cuellar — Bootstrapped Beats Sexy

John Cuellar is CEO and co-founder of Cascade Clarity, a fractional-CTO and enterprise AI governance practice. He's built and sold multiple companies across three decades — and his clearest lesson comes from directly comparing his first business against his second.

Zero Capital to \$10 Million

His first company, Harbor Technology Group, was built on pure profit with no outside investment, and sold to Mellon Financial six years later for about \$10M. The deal closed October 5, 2001, weeks after 9/11, with a buying committee whose office overlooked the Twin Towers. His lesson from that whole run, stated flatly: "you can't go broke making money."

The VC-Funded Company Made Less, Per Dollar

His second venture, Synchr, raised roughly \$130M in venture funding over 15+ years and never scaled to what investors expected. His own comparison: "I made 10X more on the first one versus the product business... even though product is more sexy." Not an argument against raising capital — a direct data point about what bootstrapped, profitable growth actually returned him.

You'll Have to Manage the Agent, Not Just the Person

His clearest framing on AI's effect on leadership: "you have to have the skill now to manage the non-human actors in your organization." An agent doesn't get a performance review the way a person does — but someone still has to measure whether it actually did the job.

Watch the full episode: youtu.be/-Tt2XKA1-jw

John's Advice, In Practice

Bad organizational data breaks automation before AI ever gets a chance.

Watching Dell run flawless cross-department workflows taught him that accurate data about who reports to whom is the real bottleneck — not the technology layered on top of it.

Trying is not failing.

"There is really no failure, because trying is not failing. If you tried, you did not fail, period."

Relationships outlast any single deal.

A broken-down van trip through Colombia and Ecuador is literally how he met the business partner who now runs his company's front-end sales — he called the guy back six months later and it turned into a real partnership.

Go past the chat window — learn the tools underneath it.

His advice for anyone job-hunting or changing roles: learn a bit of how tools like Claude Code actually work, not just the consumer chat interface, and use that to automate real tasks like resume tailoring or job-post evaluation.

REFLECTION

Is there a relationship in your own life that started as an inconvenience — a delay, a breakdown, a detour — the way John's did? What came out of it?

BE OUR NEXT GUEST

Your Story Belongs on Wake Up X

If The Consultant Mindset gave you something worth using, you already have something worth sharing. The Wake Up X podcast exists to put real, unpolished career and business stories in front of people who need to hear them — and every guest so far started exactly where you are now: reading, building, and figuring it out in public.

You might be a fit if you:

Broke into tech, cybersecurity, or business without the traditional path

Built something — a lab, a business, a portfolio, a community — and can talk about how

Have a failure or turning point you're willing to tell straight, not polished

Are mentoring, leading, or building something worth other people knowing about

HOW TO APPLY

Share your story at wakeupxpodcast.com/guest, or connect directly on LinkedIn at linkedin.com/in/efrenlsalazar and mention this book.

Being a guest is also visibility — the same principle Book 7 makes the case for.

Acknowledgments

THE PEOPLE WHO MADE THIS POSSIBLE

None of Wake Up X happened alone, including The Consultant Mindset. Every book in this library, the podcast, and the free library exist because specific people showed up for a kid nobody expected much from. This is the record of who they are.

THE EARLY YEARS

Miss England, the 6th-grade teacher who stayed after class, unpaid, to teach the fundamentals. Sister Cora and Lewis Bair of Computers for Kids, who handed a teenager a key and a broken machine and let the work speak. Jose Luis Bravo, Angel Martinez, Isaac Murrilo, and Alex De Haro — the friends and cousins who translated the world, gave hope, drove him where he needed to go, and carried him before there was anything to carry him toward.

THE BUSINESS YEARS

Rojelio Salazar, his cousin and first mentor, RNS Computer Repairs — the reason it is called Wake Up X. Kyle Erni, his first business partner. Erick and Pancho Arauza, co-founders of The Cellphone Fix. Danny Maya, of Maya Traditional Mexican Restaurant, Taqueria Maya, and Salt and Water, who pushed him in front of a camera for the first time.

THE CAREER YEARS

Igor, who gave him the vocabulary for work he already knew how to do. Nate, who he mentored into a full-time role alongside his own. Ken Kerrick, opponent turned friend in the ISACA Sacramento election. Alvin Jue and David De los Reyes, who believed in the middle of it, not just at the finish.

THE PODCAST

Every guest who trusted an unproven show with their story, including Gustavo Mastroianni, Cathy Stephenson, Greg McCord, Aaron Portnoy, Sana Talwar, Jorge Avila, Anil Reddy, Devin Prasad, Larry Switzer, Louis Kemp, Mohana Balakrishnan, Eddie Lopez, and Michael Fierros — full episodes at wakeupxpodcast.com.

COMMUNITY PARTNERS

ISACA Sacramento and ISC2 Sacramento, The Cellphone Fix, A.N.U.L Technologies, AISOC, AspireByte, and PodGrabber — the organizations building alongside this movement.

AND

His parents, who arrived with nothing, worked since childhood, and gave away whatever was left over. Nothing about the first version of this story predicted the version you are holding now.

Continue the Journey

EVERY BOOK LEADS SOMEWHERE ELSE

The Consultant Mindset is one part of a 12-book system. If it helped, the next step is one page away — take whichever layer matches where you are standing.

THE LIBRARY

Layer 1 — The Story

I Just Have To Wake Up — why any of this exists.

Layer 2 — The Mindset

Comfort Is Expensive — the real cost of playing it safe.

Layer 3 — The Blueprint

DECODE: The Wake Up X Playbook — zero experience to executive thinking. Book 0.

Layer 4 — The Playbooks

1 Break In · 2 Build It · 3 Think Like a Cybersecurity Professional · 4 Business First · 5 Lead Without a Title · 6 AI for IT & Cybersecurity · 7 Build Your Brand · 8 The Consultant Mindset · 9 Level Up

FIND US EVERYWHERE

Podcast & free library: wakeupxpodcast.com · Spotify: open.spotify.com/show/28GZnRBrIV2efHOhZdiftt · Apple Podcasts: podcasts.apple.com/us/podcast/id1792821643 · YouTube: youtube.com/@Efren.wakeupx · LinkedIn: linkedin.com/in/efrenlsalazar · Substack: substack.com/@wakeupxefren

IN GOOD COMPANY

ISACA Sacramento, ISC2 Sacramento, The Cellphone Fix, A.N.U.L Technologies, AISOC, AspireByte, and PodGrabber — the partners and collaborators building alongside the movement. See the full list at wakeupxpodcast.com/community.html.

Get DECODE free, apply to be a guest, or read the story behind the movement — all of it starts at wakeupxpodcast.com.

WAKE UP X

SOLUTIONS



Efren L. Salazar

FOUNDER · HOST · CYBERSECURITY PROFESSIONAL · EDUCATOR · COMMUNITY BUILDER

STAY CONNECTED

Podcast & Blueprint Library:	wakeupxpodcast.com
LinkedIn:	linkedin.com/in/efrenlsalazar
YouTube:	youtube.com/@Efren.wakeupx
Be a Guest:	wakeupxpodcast.com/guest

THE WAKE UP X PUBLISHING SYSTEM

BOOK 9

Level Up

Long-term growth, burnout, legacy, and mentoring others

The long game, and how not to stop playing it

THE
WAKEUPX
SOLUTIONS

EFREN L. SALAZAR

Founder & Host, Wake Up X | IT Operations Manager & Solution Engineer
Government & Regulatory Partner, ISACA Sacramento | CompTIA Security+

wakeupxpodcast.com

Educational content — not professional, legal, financial, or career-placement advice.
Full disclaimer in DECODE, the foundation of this library.

Before You Begin

HOW TO USE THIS BOOK

This book is about steering a career over decades and lasting long enough for it to compound. It is the last book in the library. It is the one that decides whether the other nine matter.

Where this book sits

Two books come first — the memoir I Just Have To Wake Up and the mindset book Comfort Is Expensive — before this ten-book library. Each one is short on purpose. Pick up the one that matches the problem in front of you right now.

- n **Book 0 — DECODE.** The foundation. The map of the whole path.
- n **Book 1 — Break In.** Landing your first role in IT or cybersecurity.
- n **Book 2 — Build It.** Home labs and a portfolio that prove you can do the work.
- n **Book 3 — Think Like a Cybersecurity Professional.** Detection, triage, and incident response thinking.
- n **Book 4 — Business First.** Finance, sales, GRC, and why the business hired you.
- n **Book 5 — Lead Without a Title.** Influence, EQ, and mentoring before anyone promotes you.
- n **Book 6 — AI for IT & Cybersecurity.** Using AI as an actual working tool, not a novelty.
- n **Book 7 — Build Your Brand.** LinkedIn, content, and being findable.
- n **Book 8 — The Consultant Mindset.** Discovery, solution selling, and stakeholder management.
- n **Book 9 — Level Up.** Long-term growth, burnout, legacy, and mentoring others. **you are here**

Book 0 is the foundation. *DECODE: The Wake Up X Playbook* holds the six-level path, the Curiosity Compass, and the core frameworks. Nothing in any other book contradicts it. If a term here is unfamiliar, DECODE defines it.

What is in this book

Part One — Steering

1. The Career Is Longer Than You Think and Shorter Than You Plan For
2. Approach, Environment, or Goal

Part Two — Lasting

3. Burnout, and Why It Is Usually Structural
4. Staying Employable for Twenty Years

Part Three — What It Was For

5. What Are You Building?
6. Leaving It Open Behind You
7. Appendix: The Review, and the Whole Library

How to actually use it

This book is different from the other nine. Those are about capability. This one is about direction and endurance, and it will be more useful to you at year three than at month one.

If you are just starting, read Chapters 1 and 6 now and come back to the rest when the questions become real. If you have been at this a while and something feels off without being nameable, this is the book.

A note on Chapter 3. It deals with burnout, and it takes the position that most causes are structural rather than personal. If you are currently in a bad place, that chapter is not a substitute for talking to someone. It is a framework for diagnosing an environment, not for treating a person.

I am writing this the way I would say it to you sitting across a table — not lecturing a room. Take what works, remix the rest into your own version, and leave what does not fit.

PART

ONE

Steering

The Career Is Longer Than You Think and Shorter Than You Plan For

MISSION

To give you a way to steer over decades instead of reacting year by year.

Objectives

- Run a real review on a real schedule
- Distinguish activity from movement
- Stop accumulating a career by default

Story

I did not plan any of this.

I started in IT at thirteen with a donated computer that had no mouse. I did not speak English until around sixteen. I earned my high school diploma as an adult, nearly thirty, after being told as a child that I could not learn.

At no point in that sequence did I have a career plan. I had curiosity, and I had a series of things that seemed interesting or necessary, and I followed them.

What I want to be honest about is that this worked, and it also cost me. There were years I spent on things that were not moving me anywhere, and I did not notice because I was busy and busy feels like progress.

There were decisions I made emotionally, at moments of frustration, that I would have made differently with a week's distance.

The thing I did not have — and what this chapter is — is a habit of stopping to look.

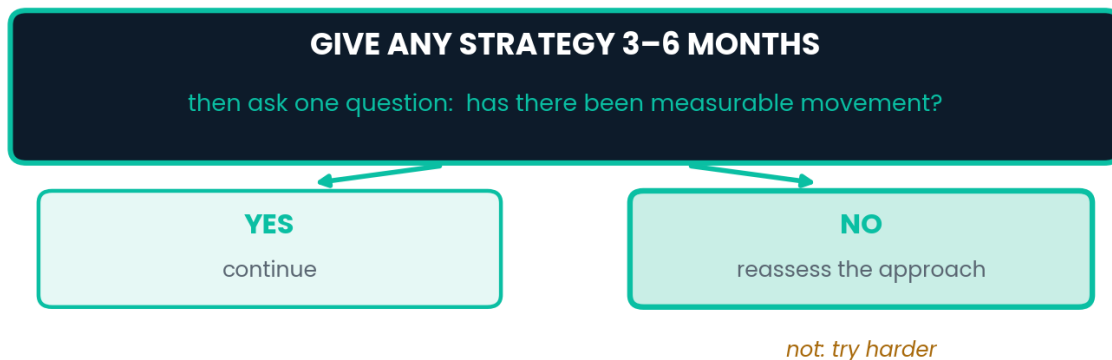
Why It Matters

Most careers are not chosen. They are accumulated. One opportunity accepted after another until you look up ten years later somewhere you never decided to go.

Sometimes that is fine. Frequently it is not, and by then the cost of correcting is high. You have a salary you rely on, a specialization you did not choose, and an identity built around work you did not select.

The alternative is not a rigid twenty-year plan. Nobody has ever successfully executed one, and the people who claim to did not. It is a review habit.

Framework: The Quarterly Review



Approach, environment, or goal — diagnose which before you act.

Shorter and you react to noise. Longer and you lose years to a method that stopped working.

The review. Three to six months, one question, and the answer determines whether you continue or reassess.

Run a business review on yourself, the way a company reviews its own performance.

Give any goal or strategy a real timeframe — **three to six months**. At the end of that window, ask one question: **has there been measurable movement?**

If yes, continue. If no, that is not a signal to try harder. It is a signal to reassess the approach.

The window is deliberate. Shorter and you are reacting to noise — nothing meaningful moves in six weeks. Longer and you will spend years on an approach that stopped working in month four. Most people run no review at all and then make a large emotional decision at year three.

What to Review

- **Career.** Am I closer to the role or the situation I want than I was three months ago? What is the specific evidence?
- **Skills.** What can I do now that I could not do then? Name it concretely. "I learned more about cloud" is not an answer.
- **Finances.** Is the trajectory what I intended?

- **Education.** Is the certification or course actually moving anything, or is it a way to feel productive without changing?
- **Relationships and network.** Who do I know now that I did not? Book 1's contact ledger makes this reviewable.
- **Health and energy.** The one everyone omits, and the one that ends careers early.

Activity Versus Movement

The distinction that makes the review useful.

Activity: three certifications this year, forty applications sent, two courses completed, a lot of hours.

Movement: a different role, a different level of responsibility, a capability you did not have, a relationship that produced an opportunity.

Activity is easy to accumulate and it feels like progress. Movement is what the review is measuring. Someone with three certifications and no change in role has not progressed — they have been busy, and being busy is how people spend five years without noticing.

This is not an argument against certifications. It is an argument for noticing when the thing you are doing has stopped producing the result you started it for.

Step-by-Step Guide: Running Your First Review

Ninety minutes, once a quarter. Put it in the calendar as a recurring event.

1. **Write down where you are,** factually. Role, level, capabilities, situation.
2. **Compare to your last review.** First time, compare to a year ago from memory.
3. **For each area, ask: what moved?** Evidence, not feeling.
4. **Separate activity from movement.** Honestly.
5. **For anything that did not move, ask why** — Chapter 2 gives you the three categories.
6. **Decide one thing to change,** not five.
7. **Write down what you expect to see by the next review.** This is the part that makes the next review possible.

That last step is what turns a review into a series rather than an event.

Real Example

Someone reviews their year honestly for the first time.

Activity: two certifications, roughly two hundred applications, a completed course, a great deal of studying.

Movement: none. Same role, same level, same salary, no new relationships, no interviews past first round.

The honest conclusion is uncomfortable and useful: the approach is not working, and more of it will not fix it. Two hundred applications producing no interviews is not a volume problem — Book 1's argument about gates applies exactly.

The change: stop applying at volume, start running T.R.A.C.K. properly on six roles a month, and go to a chapter meeting.

Next quarter: eleven applications, four conversations with humans, two interviews. Less activity, actual movement.

Case Study: The Five-Year Drift

The failure this chapter is designed to prevent.

Someone takes a role that is fine. Not exciting, but fine. The team is pleasant, the work is manageable, the pay is adequate.

Year two: still fine. Year three: they notice they have not learned anything new in a while, but a project comes up and they are busy. Year four: fine. Year five: they look for a new role and discover their skills are five years old, their network is entirely internal, and the market has moved.

Nothing bad happened. That is exactly the problem — nothing happened at all, and there was no mechanism that would have surfaced it.

A quarterly review catches this at month six, when the correction is small.

Reflection

- What have you actually accomplished in the last three months, as opposed to been busy with?
- When did you last stop and look at the direction rather than the work?
- Is there something you have been doing for a year that has produced nothing?

Journal

Write down where you were three years ago and where you are now. Then ask whether you chose the difference or accumulated it.

Exercises

1. Run your first review this week. Ninety minutes, six areas, honest answers.
2. Put the next one in your calendar as a recurring event.
3. Separate one thing you are currently doing into activity or movement.

AI Prompts

- "Here is what I did professionally in the last three months. Help me separate activity from movement, and be blunt about which is which."
- "I have been doing [approach] for [time] with no result. Ask me diagnostic questions to work out whether the problem is my approach, my environment, or my goal."
- "Help me design a quarterly self-review covering career, skills, network, and energy, with questions that need evidence rather than feelings."

Books

- *So Good They Can't Ignore You* — Cal Newport
- *Designing Your Life* — Burnett and Evans. Practical, and specifically about steering rather than planning.
- *Atomic Habits* — James Clear

Podcast Episodes

Wake Up X episodes with guests further along. Listen for whether their path was designed or accumulated — most will say accumulated, and most will name a moment where they stopped and looked.

Checklist

- I run a review every three to six months
- I distinguish activity from movement
- My review needs evidence, not feelings
- I write down what I expect to see by the next one
- The next review is in my calendar

Common Mistakes

- No review at all, then a large emotional decision at year three

- Confusing activity with movement
- Reviewing feelings rather than evidence
- Deciding five changes and doing none

Key Takeaways

- Give every strategy three to six months and a measurable test
- No movement means reassess the approach, not try harder
- Activity accumulates easily; movement is what the review measures
- Careers are accumulated by default and steered only deliberately

Next Mission

When the review says nothing moved, what you actually change.

Approach, Environment, or Goal

MISSION

To diagnose correctly when something is not working, before you act.

Objectives

- Distinguish the three causes of no movement
- Fix the approach before changing the environment
- Recognize when the environment genuinely cannot be fixed

Why It Matters

When the review says nothing moved, people act — usually by changing jobs, occasionally by changing fields, sometimes by adding another certification.

Frequently they change the wrong thing, and arrive somewhere new with the same result, because the cause travelled with them.

Framework: The Three Causes

Approach

You are doing the right thing in the wrong way.

Signal: the goal still appeals to you, other people achieve it, and your method is one you chose without much thought.

Example: you are applying to jobs but never networking. Same goal, different method. Book 1 exists entirely for this case.

Fix this first. It is the most common answer and the cheapest to change. Changing your environment when the approach was the problem simply relocates the problem.

Environment

You are doing everything right and this place has no path for you.

Signal: you are performing well, you have asked directly about progression and received vague answers, people at your level have not moved in years, and the roles you want do not exist here.

This is real, and it is more common than people admit. Individual effort cannot overcome a structural ceiling. No amount of visibility fixes an organization with no next role in it.

The trap: staying one more year, repeatedly, because leaving is uncomfortable and the discomfort of staying arrives slowly. Five years of that costs more than any single bad job change ever could.

Goal

You wanted this because of how it looked from outside, and now that you can see it clearly, you do not want it.

Signal: you have achieved intermediate steps and felt nothing, you avoid the work rather than the difficulty, and you notice relief rather than disappointment when something delays it.

This is not failure. Recognizing it early is the entire benefit of running reviews. The cost of discovering it at year eight instead of year two is enormous.

Professional Explanation

Diagnosing Honestly

The three causes are easy to confuse, because each has a comfortable and an uncomfortable version.

It is comfortable to conclude the environment is the problem, because it needs nothing of you. It is uncomfortable to conclude the approach is the problem, because it means you have been doing it wrong.

It is comfortable to conclude the goal is still right and you just need to try harder. It is uncomfortable to conclude you do not want it, because you have told people you do.

The check: **for each cause, ask what conclusion would be inconvenient for you.** That is usually the one worth examining hardest.

The Order

Diagnose in this order, because of cost:

1. **Approach first.** Cheapest to change, most commonly the cause. Give a genuinely different approach three to six months before concluding it is the environment.
2. **Environment second.** Expensive but survivable. Before leaving, test it: ask directly about progression, in a conversation not an implication, and see what the answer is.
3. **Goal last.** Most expensive to change and most important to get right. Do not conclude this in a bad month.

The exception: if the environment is actively harmful — the value conflict, unfairness, and no-control conditions in Chapter 3 — do not spend six months optimizing your approach inside it. Some environments cannot be out-worked, and the review's purpose is partly to notice when you are in one.

Testing the Environment Before Leaving

Before concluding a place has no path, do this once:

Ask your manager directly: *"I'd like to move toward [specific thing]. What would I need to demonstrate for that to be realistic here, and roughly what timeframe?"*

The answer is highly useful. A specific answer with a timeframe means there is a path. A vague answer, twice, means there is not — and you have your diagnosis without guessing.

Most people never ask. They infer, wrongly in both directions.

Real Example

Three people with the same symptom — no progress in eighteen months.

The first is applying at volume with a generic résumé and no networking. Approach problem. Fixing it costs nothing and produces interviews within two months.

The second is doing everything correctly — visible, documented, well-regarded, asking directly. There is no senior role at their organization and there will not be. Environment problem. They leave, and the same behaviour produces movement immediately elsewhere.

The third achieved the certification and the first analyst role and felt nothing. They realize they liked the idea of the field more than the work, and what they actually enjoy is the teaching and explaining part. Goal problem. They move toward training and enablement, and are much happier.

Same symptom, three causes, three completely different correct actions. Acting without diagnosing would have been wrong for at least two of them.

Case Study: The Move That Did Not Help

The pattern that makes diagnosis worth the effort.

Someone frustrated by lack of progress changes jobs. New company, small pay increase, fresh start.

Eighteen months later, the same frustration. They conclude the industry is the problem.

What actually travelled with them was the approach: still not visible, still not asking for what they wanted, still assuming good work would be noticed, still no network outside their immediate team.

Changing environment fixes an environment problem. It does nothing at all for an approach problem, and it costs a year of ramp-up to learn that.

Reflection

- Which of the three causes would be most inconvenient for you to conclude?
- Have you asked directly about progression, in a real conversation?
- Have you given a genuinely different approach three to six months?

Journal

Write down what you would do if you knew for certain the problem was your approach. Then write what you would do if you knew for certain it was the environment. Compare which one you would rather be true — that preference is worth examining.

Exercises

1. Diagnose your current situation against the three causes.
2. Ask the direct progression question this month.
3. If it is the approach, change one specific thing and give it a quarter.

Templates: The Diagnostic Questions

APPROACH

Do other people achieve this with a different method?
Did I choose my method, or default into it?
Have I genuinely tried something different for 3 months?

ENVIRONMENT

Does the role I want exist here?
Have people at my level moved in the last two years?
Have I asked directly, and what was the answer?

GOAL

Did achieving the last milestone feel like anything?
Am I avoiding the work, or the difficulty?
Was I relieved the last time this got delayed?

AI Prompts

- "Here is my situation: [description]. Ask me diagnostic questions to determine whether this is an approach, environment, or goal problem. Do not tell me your conclusion until you have asked at least five."
- "I want to ask my manager directly about progression. Help me phrase it so it is a real question rather than an implication that I am leaving."
- "I think my goal may have changed. Ask me questions that would distinguish 'this is hard' from 'I do not want this.'"

Books

- *Designing Your Life* — Burnett and Evans
- *So Good They Can't Ignore You* — Cal Newport
- *Range* — David Epstein. Relevant to concluding a goal has changed without treating it as failure.

Checklist

- I diagnose before acting
- I fix approach before changing environment
- I have asked directly about progression at least once
- I do not conclude goal problems in a bad month
- I know which conclusion would be inconvenient for me

Common Mistakes

- Changing environment when the approach was the cause
- Never asking directly, then inferring
- Concluding the goal is wrong during a bad quarter
- Staying one more year, repeatedly

Key Takeaways

- Approach, environment, or goal — diagnose before acting
- Approach is cheapest and most common; fix it first
- Ask directly about progression; a vague answer twice is your answer
- The inconvenient conclusion is usually the one worth examining

PART

TWO

Lasting

Burnout, and Why It Is Usually Structural

MISSION

To help you recognize burnout early and diagnose its cause correctly.

Objectives

- Recognize the symptoms before they become a crisis
- Understand that most causes are structural
- Know what actually helps

Why It Matters

This industry burns people out at a high rate. Security especially — always-on expectations, incidents at inconvenient hours, and a job where success is invisible and failure is extremely visible.

The people with the best careers are frequently not the most talented. They are the ones who were still there at year fifteen, still curious, still building. Compounding needs continuity, and continuity is the thing burnout takes.

Professional Explanation

What It Actually Looks Like

Burnout is not being tired. Tired is fixed by a weekend. Burnout presents as:

- **Cynicism** about work you used to care about
- **Doing the minimum** and feeling nothing about it
- **Dread** on Sunday that starts earlier each week
- **Reduced capability** at things that used to be easy
- **Withdrawal** from colleagues you used to like
- **Loss of curiosity**, which in this field is the clearest signal of all

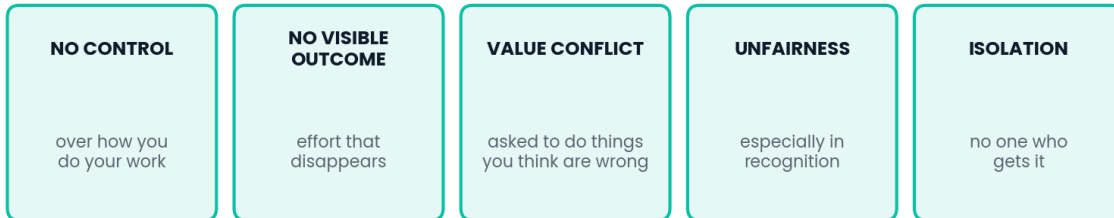
The cynicism usually comes first. It is usually misread as maturity — *"I've just seen how it really works now."* Sometimes that is true. Frequently it is the first symptom wearing a respectable disguise.

The curiosity signal is worth watching specifically. Everyone in this field started because something was interesting. When that goes, something is wrong. It is not a character

development.

What Causes It

WHAT ACTUALLY CAUSES BURNOUT



Four of five are structural, not personal. "Manage your stress better" is the wrong intervention.

Five causes. Four of them are structural, which is why the usual advice does not work.

Rarely workload alone. The reliable causes are:

- **No control** over how you do your work
- **No visible outcome** — effort that disappears
- **Value conflict** — being asked to do things you think are wrong
- **Unfairness** — especially in recognition
- **Isolation**

Notice that four of the five are structural rather than personal. Which means "manage your stress better" is usually the wrong intervention. The right one is changing the condition, and sometimes that means changing the environment.

This matters because the standard advice — sleep more, exercise, set boundaries — treats burnout as a personal resilience failure. Those things help, and they cannot fix a value conflict or an unfairness problem.

You cannot out-discipline a bad environment. People spend years trying, and the years do not come back. If you take one thing from this book, take this.

What Actually Helps

- **Boundaries that are stated, not implied.** People cannot respect a boundary they do not know exists. "I don't check messages after seven" said once beats hoping people notice.

- **Something outside work that you are also good at.** When your entire identity is your job, a bad quarter at work is a bad quarter at being a person.
- **Relationships in the field.** Isolation makes everything heavier. This is one of the strongest arguments for chapter involvement, and one nobody makes.
- **Saying no earlier.** Most burnout is built from yeses that should have been noes six months prior. Book 8's three forms apply directly.
- **Visible outcomes.** If your work disappears, make some of it visible — Book 5's success file is partly a burnout intervention.
- **Actually leaving when the environment is the cause.** No amount of resilience fixes a by design bad situation.

Real Example

Someone in a SOC role experiencing all the symptoms.

The personal-resilience reading: they need better work-life balance, more sleep, a hobby.

The structural reading, on examination: they have no control over which alerts they work, their tuning suggestions have been declined repeatedly, the queue is unmanageable by design, and two colleagues with less tenure were promoted without explanation.

That is no-control, no-visible-outcome, and unfairness — three of the five, all structural.

A hobby does not fix that. What fixes it is either changing the conditions — which needs the conversation about tuning authority, and Book 4's framing to make the case — or leaving.

Telling this person to sleep more is not just unhelpful; it is a diagnosis that locates the problem in them.

Case Study: The Cynicism That Was Real

Worth acknowledging honestly, because the framing above can be applied too broadly.

Sometimes the cynicism is accurate. Sometimes an organization genuinely is dysfunctional, the security programme genuinely is theatre, and the person noticing it is correct rather than burned out.

The distinguishing question is not whether the observation is true. It is **whether you still care**.

Someone who sees dysfunction clearly and is still trying to fix it is not burned out; they are clear-eyed and frustrated. Someone who sees it and has stopped trying, and feels nothing about that, is burned out.

The second state is where careers end quietly. It is the point at which the environment question stops being optional.

Reflection

- Which of the five causes is present in your current situation?
- Are you still curious about this field?
- Have you been treating a structural problem with personal discipline?

Journal

Write about the last time you were genuinely interested in something at work. If it was a long time ago, that is information rather than a judgment.

Exercises

1. Check your situation against the five causes. Name which are present.
2. State one boundary out loud to the person who needs to hear it.
3. If your work is invisible, make one piece of it visible this month.

Templates: The Honest Check

Do I still find any of this interesting?
Which of the five causes are present right now?
Is my cynicism accurate, and do I still care?
What have I been trying to out-discipline?
What would have to change for this to be sustainable?
Is that change available here?

Run this at each quarterly review, not only when things are bad.

AI Prompts

- "Here are the five structural causes of burnout: no control, no visible outcome, value conflict, unfairness, isolation. Ask me questions about my situation to identify which are present."
- "Help me phrase a boundary to my team clearly and without apology."
- "I want to make a case for more control over how my work is prioritized. Help me structure it as a proposal rather than a complaint."

Books

- *Burnout: The Secret to Unlocking the Stress Cycle* — Nagoski and Nagoski
- *Extreme Ownership* — Willink and Babin, with the caveat that ownership does not mean absorbing a broken environment indefinitely
- *Deep Work* — Cal Newport

Podcast Episodes

Wake Up X episodes where guests discuss difficult periods. Listen for whether the resolution was personal or structural — it is nearly always structural.

Resources

- Your local chapter, as an isolation intervention as much as a professional one
- Anyone in the field who will talk honestly. Book 5's personal board, seat five.

Checklist

- I know which of the five causes are present for me
- I state boundaries rather than implying them
- I have something outside work I am also good at
- I have relationships in the field outside my employer
- I am not trying to out-discipline a structural problem

Common Mistakes

- Treating cynicism as wisdom
- Treating a structural problem with personal discipline
- Waiting for it to resolve itself
- Isolation, which makes every other cause heavier

Key Takeaways

- Burnout is not tiredness; loss of curiosity is the clearest signal
- Four of the five causes are structural
- You cannot out-discipline a bad environment
- The question is not whether the cynicism is accurate, but whether you still care

Next Mission

The practical side of lasting: staying employable across decades.

Staying Employable for Twenty Years

MISSION

To keep you valuable as the technology, the market, and you all change.

Objectives

- Build breadth deliberately rather than accidentally
- Identify what stays valuable as tools change
- Avoid the specialization trap

Why It Matters

The technology you know now will be a lot different in ten years. Some of what you are expert in will be irrelevant. This has happened in every decade of this industry and there is no reason to expect an exception.

What determines whether that is a crisis or a non-event is what else you built.

Specialists are the ones getting laid off right now. Breadth across networking, security, IT, programming fundamentals, and hands-on AI — plus business literacy on top — is what actually protects a career.

I want to be precise about this, because it is easy to misread as an argument against depth. It is not. Depth in one area plus working competence across several, plus the ability to speak to the business, is the combination. Depth alone is exposed. Breadth alone is shallow.

Professional Explanation

The Specialization Trap

Specializing is rewarded early. You become the person who knows the thing, you are valuable, you are asked for.

The trap has two parts.

First, the thing can become irrelevant. Products get replaced, approaches get superseded, whole categories get absorbed into platforms. Deep expertise in something that no longer matters is worth very little. The transition is painful because you are starting from a lower position than your title suggests.

Second, deep specialists are expensive and narrow. In a reduction, the calculation is cold: this person costs a great deal and does one thing, and we can buy that thing as a service. That calculation is not a judgment of quality.

The protection is not to avoid depth. It is to have depth *and* the ability to do adjacent things competently, *and* to speak the business language Book 4 teaches.

What Stays Valuable

Independent of tools and platforms:

- **Understanding how systems fail** and how to diagnose from symptoms — Book 2's break sessions
- **Judgment about what matters** under uncertainty — Book 3
- **Being able to explain technical things to non-technical people** — Book 4
- **Accountability** — being someone who owns outcomes
- **Relationships** — Book 1's ledger, Book 7's visibility
- **The ability to learn something new quickly**, which is itself a trainable skill

Notice how many of these are covered by other books in this library. That is the design. The library is not nine topics; it is one argument delivered nine ways.

Deliberate Breadth

Breadth acquired accidentally is thin. Breadth acquired deliberately is a moat.

A practical approach: **one deep area, three working areas, one emerging area.**

- **Deep:** your lane — your Curiosity Compass lane from DECODE, inside your career current from Book 2. The thing you are genuinely expert in.
- **Working:** three adjacent areas where you are competent enough to be useful, understand the vocabulary, and can work with specialists without needing translation.
- **Emerging:** one thing you are actively learning that may matter in three years. Right now, for most people, that is AI as an operational tool — Book 6.

Review this at your quarterly review. The emerging area should change every couple of years; if it has not, you have stopped scanning.

On Certifications, Honestly

Certifications are useful for three things: getting past résumé filters, structuring your learning, and satisfying a requirement someone else has set.

They are not useful for showing capability to anyone who can assess capability directly, and they do not substitute for movement — Chapter 1's distinction applies.

The pattern to avoid: collecting certifications as a way of feeling productive while nothing changes. Three certifications and the same role is activity, not movement.

Framework: The Employability Review

Add this to your quarterly review, once a year.

DEEP AREA:	what am I genuinely expert in? is it still relevant in five years?
WORKING AREAS:	three adjacent things I can be useful in which is weakest?
EMERGING AREA:	what am I learning that may matter in 3 years? when did I last change this?
BUSINESS FLUENCY:	can I make a case in business terms? have I done it recently?
NETWORK:	how many people outside my employer would take my call?
TRANSFERABLE:	if my current tools vanished, what remains?

That last question is the whole chapter. Answer it honestly once a year.

Real Example

Two people, fifteen years in, same starting point.

The first specialized deeply in one vendor's platform. He is genuinely excellent at it and was well paid for a decade. The platform is being replaced across the industry. His expertise transfers partially, his title does not, and he is competing for roles a level below where he was, against people who are cheaper.

The second went deep on networking and security, kept working competence in systems administration and cloud, learned enough scripting to automate, and spent years learning to talk to business stakeholders. Less exceptional in any single dimension.

When the market shifted, the first had to rebuild. The second adjusted.

Neither made a mistake exactly. One had a fragile position and did not know it, because fragility is invisible until it is tested.

Case Study: What the Layoff Conversation Looks Like

Uncomfortable and worth knowing, because it explains what to build.

The decision is made quickly, by people with incomplete information, under time pressure. What they can see: cost, what the person visibly does, what other leaders say about them, and how replaceable the function is.

The protected position is: reasonable cost, visibly does several things, other leaders know who you are, and the function needs judgment and accountability rather than execution of a specified task.

Every element of that is buildable, and most of them are covered in Books 4, 5, and 7. None of them is about being the best engineer.

Reflection

- If your primary technology disappeared, what would remain?
- When did you last change your emerging area?
- How many people outside your employer would take your call?

Journal

Write down what you were expert in five years ago. How much of it still matters? That ratio, projected forward, is your exposure.

Exercises

1. Complete the employability review honestly.
2. Identify your weakest working area and spend a quarter on it.
3. Name your emerging area and what you will do about it this month.

AI Prompts

- "Here is my skill profile: [description]. Identify where I am fragile — over-specialized, or missing adjacent competence that would make me hard to replace."
- "What adjacent skills would make someone with [my depth area] much harder to replace, and why?"
- "I want to build working competence in [area] without going deep. What is the minimum useful level, and how would I know I have reached it?"

Books

- *Range* — David Epstein. The case for breadth, with evidence.
- *So Good They Can't Ignore You* — Cal Newport
- *Antifragile* — Nassim Taleb. Relevant to why fragility is invisible until tested.

Checklist

- I have one deep area, three working areas, one emerging area
- My emerging area has changed in the last two years
- I can make a business case, and have done recently
- I know how many people outside my employer would take my call
- I have answered the "if my tools vanished" question honestly

Common Mistakes

- Depth alone, which is fragile
- Breadth alone, which is shallow
- Collecting certifications as a substitute for movement
- Never changing the emerging area
- Assuming your current expertise will remain relevant

Key Takeaways

- One deep area, three working areas, one emerging area
- What stays valuable: diagnosis, judgment, explanation, accountability, relationships, learning speed
- Deep specialists are expensive and narrow, which is a calculation not a judgment
- Fragility is invisible until it is tested

PART

THREE

What It Was For

What Are You Building?

MISSION

To make you answer the question deliberately rather than inherit someone else's answer.

Objectives

- Name what you are actually optimizing for
- Check your decisions against it
- Stop running a definition of success you did not choose

Why It Matters

At some point the question stops being "what is my next role" and becomes "what am I building." Most people never make that transition explicitly, and so they keep optimizing for the next rung of a ladder they never chose to climb.

Framework: The Five Things People Build



None is superior. The damage comes from running someone else's version by default.

None is superior. The damage comes from running someone else's version by default.

Mastery. Depth, craft, being genuinely excellent at a thing. The satisfaction is in the work itself.

Freedom. Options, income that does not need permission, the ability to walk away. The satisfaction is in not being trapped.

Impact. Changing how something works, at scale. The satisfaction is in the difference made.

Security. Stability, predictability, a life that does not wobble. The satisfaction is without anxiety.

For other people. A family, a community, a next generation that starts further ahead than you did. The satisfaction is in what it enables for someone else.

None of these is superior. What causes damage is running someone else's version by default.

The person optimizing for freedom while chasing a title designed for someone optimizing for security will be unhappy at every rung. Will not understand why, because by every external measure they are succeeding.

Professional Explanation

How to Tell Which One You Are

Not by what you say. By what you have actually chosen when the choices conflicted.

- When you turned something down, what were you protecting?
- When you accepted something that looked worse on paper, what did it give you?
- What do you resent spending time on, and what does that resentment protect?
- When you imagine year ten, what are you doing, and who is around?

The answers are usually more consistent than people expect, and frequently different from what they would have claimed.

Checking Decisions Against It

Once you have named it, the test is simple: **look at your last three significant decisions and ask whether they match.**

Most people find at least one that does not. That is not a crisis; it is information. The point is to notice the drift while it is one decision rather than five years.

They Change

What you are building at twenty-five is frequently not what you are building at forty. Circumstances change, people arrive, priorities move.

The failure is not changing. The failure is changing without noticing, and continuing to optimize for a thing you stopped wanting — which is exactly what the quarterly review is for.

Real Example

Someone optimizing for freedom, without having named it, takes a senior role at a large organization.

More money, better title, a real step up by any external measure. Also: less control over their time, a longer commute, more meetings, and a structure they cannot change.

They are unhappy within a year and cannot explain why, because everything they can point to improved. Friends and family are confused. They conclude something is wrong with them.

Nothing is wrong with them. They optimized for security and status while building for freedom, and the two conflicted in a way that was invisible until they were inside it.

Having named it in advance would not have made the decision for them. It would have made the trade-off visible, which is all anyone can ask.

Case Study: The Success That Did Not Feel Like Anything

A pattern worth recognizing because it is a signal rather than an ingratitude.

Someone reaches a goal they worked toward for years — the role, the salary, the recognition — and feels essentially nothing. A day of satisfaction, then normal.

The usual reading is that they are unable to be satisfied. Occasionally true. More often the goal belonged to someone else's definition — a parent's, an industry's, a peer group's — and achieving it did not deliver anything because it was never actually theirs.

This is the strongest available signal that the goal question from Chapter 2 needs revisiting. It is worth taking seriously rather than dismissing as a mood.

Reflection

- Which of the five are you actually building?
- Do your last three significant decisions match it?
- Whose definition of success have you been running?

Journal

Write one sentence: **what am I building, and for whom?** Then look at your last three significant decisions and ask whether they match. Sit with any that do not.

Exercises

1. Write the sentence. One sentence, no hedging.
2. Test it against your last three decisions.
3. Ask someone who knows you well what they think you are optimizing for. The gap is instructive.

Templates: The Building Questions

When I turned something down, what was I protecting?
When I took something that looked worse on paper,
what did it give me?
What do I resent spending time on?
In year ten, what am I doing and who is around?

ONE SENTENCE: I am building _____ for _____.

Do my last three decisions match it?

AI Prompts

- "Here are three significant career decisions I made and why: [descriptions]. What do they suggest I am actually optimizing for, as opposed to what I say?"
- "Ask me questions that would distinguish whether I am building for mastery, freedom, impact, security, or other people."
- "I achieved something I worked toward for years and felt nothing. Ask me questions to explore whether the goal was mine."

Books

- *Designing Your Life* — Burnett and Evans
- *The Psychology of Money* — Morgan Housel. Specifically on defining enough for yourself rather than inheriting a definition.
- *Four Thousand Weeks* — Oliver Burkeman

Checklist

- I can state what I am building in one sentence
- My last three decisions match it
- I know whose definition I have been running
- I revisit this at least annually

Common Mistakes

- Never asking the question explicitly
- Running an inherited definition

- Optimizing for external markers that conflict with what you actually want
- Changing what you want and continuing to optimize for the old thing

Key Takeaways

- Five things people build; none is superior
- The damage comes from running someone else's version by default
- Test it against decisions, not against intentions
- Achieving something and feeling nothing is a signal, not ingratitude

Leaving It Open Behind You

MISSION

To close the library where it started.

Story

I was placed in special education classes as a child and told, repeatedly, that I could not learn. I believed it for a long time, because when you are a child and adults tell you what you are, you tend to take their word for it.

I was a farm-worker family's kid. I did not speak English well. The system had decided what I was.

In sixth grade, a teacher named Miss England stayed after class to teach me the fundamentals nobody else had bothered to teach me. She did not have to. It was not part of her job, and I was not an easy student.

She gave me hope when I had none.

I did not learn English until around sixteen. I earned my high school diploma as an adult, nearly thirty. I started in IT at thirteen with a donated computer that had no mouse, so I learned the whole machine through keyboard shortcuts because it was the only way in.

Everything after that — the businesses, the operations work, the podcast, these books — came later. And none of it happens without a teacher who decided I was worth extra time.

I have never been able to tell her what it meant.

Why It Matters

There is somebody right now who has been told they are not smart enough, or that they started too late, or that people like them do not do this kind of work.

Some of them are being told it by teachers. Some by family. Most of them, eventually, by themselves.

You now know things that person needs to know. You do not need to be an expert. You need to be one step ahead and willing to turn around.

Professional Explanation: On Failure

I have failed more than a hundred times. Probably more than five hundred.

Businesses that did not work. A first employee I hired on a recommendation without a real interview, who was wrong for what the business actually needed. My failure, not his, because I did not vet or set expectations. Decisions I would take back. Years spent on things that went nowhere.

Every one of them is in what I know now.

Failure is not the opposite of success. It is part of it. If you take one thing from this library to whoever you help, take that one — because the reason most people stop is that they believe failing means they were wrong to start.

The people who do not make it are almost never the ones who failed most. They are the ones who stopped.

Framework: What This Library Was Actually For

Ten books, and one argument delivered ten ways.

- **Book 0** gave you the map.
- **Book 1** got you in the door by reducing a stranger's risk.
- **Book 2** made your capability visible by building and breaking things.
- **Book 3** taught judgment rather than tools.
- **Book 4** connected your work to how the business survives.
- **Book 5** showed that influence precedes authority.
- **Book 6** made AI a tool rather than a crutch.
- **Book 7** made you findable before you felt ready.
- **Book 8** shifted you from executing requests to diagnosing problems.
- **Book 9** is about steering, lasting, and what it was for.

The frameworks are borrowed. T.R.A.C.K., V.A.L.U.E., the Discovery Ladder, the four gates — all remixed from sales, marketing, and leadership thinking that already existed, deliberately dragged into a field that does not naturally reach for them.

That is not a weakness of the material. It is the entire idea. Cybersecurity and IT are unusually isolated disciplines, and importing what other fields already know is the cheapest advantage available.

Remix them again into your own version. I would rather you did.

Real Example

I started in IT at thirteen with a donated computer and no mouse. I did not speak English until around sixteen. I got my high school diploma at nearly thirty, after being told as a child that I could not learn.

I am now an IT operations manager and solution engineer, a founder, a partner with my local ISACA chapter, and I host a podcast where I get to ask people how they did it.

None of that was planned at thirteen, or at thirty. All of it came from staying curious and not stopping.

If you are early and it feels impossible, that is not evidence that it is. **It is what early feels like.**

Case Study: The Four Minutes

Someone posts a beginner question in a community. Awkwardly phrased. The kind that usually gets a link and a slightly impatient tone.

The typical response: "just google it."

The response that matters: a direct answer to the actual question, one sentence about the thing they will hit next, and an offer to answer follow-ups.

Four minutes.

For the person receiving it, that is frequently the difference between continuing and quitting. Not because the information was unavailable, but because someone treated them as worth answering.

I know this because I was that person, repeatedly. The people who answered me properly are the reason I kept going, and most of them have no idea.

Reflection

- n Who was your Miss England? Have you told them?
- n Who is one step behind you right now?
- n What would you have needed to hear when you were starting?

Journal

Write the message you wish someone had sent you when you were starting. Then find someone to send it to.

Exercises

1. Answer one beginner question properly this week. Not a link. A real answer.
2. Make one introduction between two people who should know each other.
3. Tell someone who helped you what it meant. Do it before you cannot.

Project

Commit to one hour a month, for six months, helping people earlier on the path than you. Six hours total. Then look at what it did to your own understanding — and to theirs.

Checklist

- I have someone I am actively helping
- I share my failures, not only my results
- I have told someone who helped me what it meant
- I have stopped treating failure as evidence I should not have started

Key Takeaways

- Failure is part of success; the ones who do not make it are the ones who stopped
- You do not need to be an expert — one step ahead and willing to turn around
- Somebody did this for you, or you needed them to
- Remix the frameworks into your own version

The Last Page

Wake up. Start where you are.

Build one thing. Break it. Write it down. Send the message to the stranger. Ask the question you are afraid is stupid. Say the number and stop talking. Own the bad day out loud. Give the twenty-minute talk. Fix the profile this afternoon.

Run the review. Notice when nothing has moved. Change the approach before you change the environment, and change the environment when it genuinely is the environment.

And when someone earlier on the path than you asks a question — turn around and answer it properly.

That is the whole thing. Ten books, one idea.

Somebody made a path for you, or you built one where there was none.

Either way, leave it open behind you.

Connect

Tell me your story — I mean that literally.

[linkedin.com/in/efrenlsalazar](https://www.linkedin.com/in/efrenlsalazar)

The podcast: **wakeupxpodcast.com**

Appendix: The Review, and the Whole Library

REFERENCE

Everything from this book, plus the library at a glance.

The Quarterly Review

Give any strategy 3-6 months, then ask:
has there been measurable movement?

YES -> continue

NO -> reassess the approach (not: try harder)

REVIEW AREAS

career	am I closer than 3 months ago? evidence?
skills	what can I do now that I couldn't?
finances	is the trajectory what I intended?
education	is this moving anything, or feeling productive?
network	who do I know now that I didn't?
energy	the one everyone omits

ACTIVITY vs MOVEMENT

activity	= certifications, applications, hours
movement	= different role, new capability, an opportunity

The Three Causes

APPROACH	right thing, wrong method – fix this FIRST
ENVIRONMENT	doing everything right, no path here
GOAL	wanted it from outside; don't now

Diagnose in that order. The inconvenient conclusion is usually the one worth examining.

The Diagnostic Questions

APPROACH

Do others achieve this with a different method?
Did I choose my method, or default into it?
Have I genuinely tried something different for 3 months?

ENVIRONMENT

Does the role I want exist here?
Have people at my level moved in two years?
Have I asked directly – and what was the answer?

GOAL

Did the last milestone feel like anything?
Am I avoiding the work, or the difficulty?
Was I relieved when it got delayed?

The Five Causes of Burnout

no control	over how you do your work
no visible outcome	effort that disappears
value conflict	asked to do things you think are wrong
unfairness	especially in recognition
isolation	

Four of five are structural.
You cannot out-discipline a bad environment.

The Employability Review — annually

DEEP AREA:	still relevant in five years?
WORKING AREAS:	three adjacent; which is weakest?
EMERGING AREA:	when did I last change it?
BUSINESS FLUENCY:	made a case recently?
NETWORK:	who outside my employer takes my call?
TRANSFERABLE:	if my tools vanished, what remains?

What Are You Building

MASTERY	depth, craft, excellence
FREEDOM	optionality, income without permission
IMPACT	changing how something works
SECURITY	stability, a life that doesn't wobble
FOR OTHERS	family, community, a next generation

ONE SENTENCE: I am building _____ for _____.
Do my last three decisions match it?

The Library at a Glance

0	DECODE	the map
1	BREAK IN	reduce a stranger's risk
2	BUILD IT	build, break, document, explain
3	THINK LIKE A PRO	judgment, not tools
4	BUSINESS FIRST	money, risk, and translation
5	LEAD WITHOUT A TITLE	influence precedes authority
6	AI FOR IT & CYBER	a tool, not a crutch
7	BUILD YOUR BRAND	findable before you're ready
8	CONSULTANT MINDSET	diagnose, don't prescribe
9	LEVEL UP	steer, last, and reach back

The Frameworks, One Line Each

THE FOUR GATES	database, skim, risk, conversation
T.R.A.C.K.	tailor, research, approach, contribute, keep
BUILD-BREAK-DOC-EXPLAIN	the project loop
THE RISK SENTENCE	if / exploits / then / likelihood
THE TRIAGE LOOP	six questions, every alert
THE VALUE TRANSLATION	what breaks, what it costs, the fix, do nothing
V.A.L.U.E.	validate, ask, listen, underscore, engage
THE FIVE SOURCES	competence, reliability, generosity, clarity, composure
THE VERIFICATION GATE	embarrassed? unprovided specifics? sign it?
THE THREE LAYERS	findable, credible, known
THE DISCOVERY LADDER	verify, anchor, lay out, uncover, end
THE QUARTERLY REVIEW	3-6 months, one question

Questions for the author. This book carries more of your material than any other, and it still has gaps: (1) **a period where you nearly stopped** — Chapter 3 argues about burnout by design but has no moment of yours in it; (2) **someone you have mentored and where they ended up** — this is the third book to flag it. It is the single most valuable outstanding addition across the whole library, because it closes the loop Miss England opens; (3) whether the quarterly review is something you actually run, and what it has changed for you. Chapter 1 would be much stronger with your own version.

The Next Generation, Watching

Leveling up eventually means standing next to the people coming up behind you — proof the loop this book describes actually closes.



Museum of Science and Curiosity, with the group



Museum of Science and Curiosity, event photo

Obese to Obsessed

Leveling up isn't only a career. It's who's carrying you there.

Every framework in this library assumes you have the energy, focus, and stamina to use it. That assumption isn't free. For years I ran a business, worked full-time in IT, and built a podcast while carrying weight that was slowly taking all three away from me. Leveling up your title means nothing if you don't level up the body and mind carrying you through the climb.



Same person, two different chapters of the same rebuild

THE STORY

100 Pounds, 7 Months, No Shortcuts

I didn't fix this with a program I bought. I fixed it the same way I fixed a broken laptop or a failing business process: I got honest about the real problem, I built a structure I could actually repeat, and I stopped doing it alone. In seven months, training naturally with no shortcuts, I lost 100 pounds. The number isn't the point of this chapter. The rebuild underneath it is.

The Rebuild Loop

1. Name It

Stop calling it "being busy." Name the actual pattern you're stuck in, out loud, to someone.

2. Structure It

Turn the goal into a repeatable weekly system, not a one-time burst of motivation.

3. Witness It

Bring in people who see your effort and hold you to it. Willpower alone runs out.

4. Repeat It

The plan doesn't need to be perfect. It needs to survive a bad week and still run.

5. Become It

At some point the identity flips from "trying to change" to "this is who I am now."

THE COMMUNITY

Obese to Obsessed

What started as a personal rebuild became a small community and a brand: Obese to Obsessed, under Efren's Vizion. Not a diet plan. A standing reminder that the same discipline that rebuilds a body rebuilds a career, a business, or a bad year — and that none of it has to be done alone.



Launching the Obese to Obsessed shirt, Efren's Vizion

REFLECTION

Where in your career or life are you running on willpower alone instead of a system? Who could you bring in to witness it?

IN THE FIELD

One Team, One Goal

Leveling up rarely feels like a solo achievement in the moment — it looks like standing next to the people who were in the room with you when it happened. Some milestones are worth a photo just to prove you were actually there.



A team milestone, marked in person

BE OUR NEXT GUEST

Your Story Belongs on Wake Up X

If Level Up gave you something worth using, you already have something worth sharing. The Wake Up X podcast exists to put real, unpolished career and business stories in front of people who need to hear them — and every guest so far started exactly where you are now: reading, building, and figuring it out in public.

You might be a fit if you:

Broke into tech, cybersecurity, or business without the traditional path

Built something — a lab, a business, a portfolio, a community — and can talk about how

Have a failure or turning point you're willing to tell straight, not polished

Are mentoring, leading, or building something worth other people knowing about

HOW TO APPLY

Share your story at wakeupxpodcast.com/guest, or connect directly on LinkedIn at linkedin.com/in/efrenlsalazar and mention this book.

Being a guest is also visibility — the same principle Book 7 makes the case for.

Acknowledgments

THE PEOPLE WHO MADE THIS POSSIBLE

None of Wake Up X happened alone, including Level Up. Every book in this library, the podcast, and the free library exist because specific people showed up for a kid nobody expected much from. This is the record of who they are.

THE EARLY YEARS

Miss England, the 6th-grade teacher who stayed after class, unpaid, to teach the fundamentals. Sister Cora and Lewis Bair of Computers for Kids, who handed a teenager a key and a broken machine and let the work speak. Jose Luis Bravo, Angel Martinez, Isaac Murrilo, and Alex De Haro — the friends and cousins who translated the world, gave hope, drove him where he needed to go, and carried him before there was anything to carry him toward.

THE BUSINESS YEARS

Rojelio Salazar, his cousin and first mentor, RNS Computer Repairs — the reason it is called Wake Up X. Kyle Erni, his first business partner. Erick and Pancho Arauza, co-founders of The Cellphone Fix. Danny Maya, of Maya Traditional Mexican Restaurant, Taqueria Maya, and Salt and Water, who pushed him in front of a camera for the first time.

THE CAREER YEARS

Igor, who gave him the vocabulary for work he already knew how to do. Nate, who he mentored into a full-time role alongside his own. Ken Kerrick, opponent turned friend in the ISACA Sacramento election. Alvin Jue and David De los Reyes, who believed in the middle of it, not just at the finish.

THE PODCAST

Every guest who trusted an unproven show with their story, including Gustavo Mastroianni, Cathy Stephenson, Greg McCord, Aaron Portnoy, Sana Talwar, Jorge Avila, Anil Reddy, Devin Prasad, Larry Switzer, Louis Kemp, Mohana Balakrishnan, Eddie Lopez, and Michael Fierros — full episodes at wakeupxpodcast.com.

COMMUNITY PARTNERS

ISACA Sacramento and ISC2 Sacramento, The Cellphone Fix, A.N.U.L Technologies, AISOC, AspireByte, and PodGrabber — the organizations building alongside this movement.

AND

His parents, who arrived with nothing, worked since childhood, and gave away whatever was left over. Nothing about the first version of this story predicted the version you are holding now.

Continue the Journey

EVERY BOOK LEADS SOMEWHERE ELSE

Level Up is one part of a 12-book system. If it helped, the next step is one page away — take whichever layer matches where you are standing.

THE LIBRARY

Layer 1 — The Story

I Just Have To Wake Up — why any of this exists.

Layer 2 — The Mindset

Comfort Is Expensive — the real cost of playing it safe.

Layer 3 — The Blueprint

DECODE: The Wake Up X Playbook — zero experience to executive thinking. Book 0.

Layer 4 — The Playbooks

1 Break In · 2 Build It · 3 Think Like a Cybersecurity Professional · 4 Business First · 5 Lead Without a Title · 6 AI for IT & Cybersecurity · 7 Build Your Brand · 8 The Consultant Mindset · 9 Level Up

FIND US EVERYWHERE

Podcast & free library: wakeupxpodcast.com · Spotify: open.spotify.com/show/28GZnRBrIV2efHOHzdiftt · Apple Podcasts: podcasts.apple.com/us/podcast/id1792821643 · YouTube: youtube.com/@Efren.wakeupx · LinkedIn: linkedin.com/in/efrenIsalazar · Substack: substack.com/@wakeupxefren

IN GOOD COMPANY

ISACA Sacramento, ISC2 Sacramento, The Cellphone Fix, A.N.U.L Technologies, AISOC, AspireByte, and PodGrabber — the partners and collaborators building alongside the movement. See the full list at wakeupxpodcast.com/community.html.

Get DECODE free, apply to be a guest, or read the story behind the movement — all of it starts at wakeupxpodcast.com.

WAKE UP X

SOLUTIONS



Efren L. Salazar

FOUNDER · HOST · CYBERSECURITY PROFESSIONAL · EDUCATOR · COMMUNITY BUILDER

STAY CONNECTED

Podcast & Blueprint Library:	wakeupxpodcast.com
LinkedIn:	linkedin.com/in/efrenlsalazar
YouTube:	youtube.com/@Efren.wakeupx
Be a Guest:	wakeupxpodcast.com/guest

THE
WAKEUPX
MOVEMENT

STOP
SURVIVING.
START
BUILDING.



DISCIPLINE
LEARNING
RISK
PEOPLE
PURPOSE
FREEDOM

COMFORT IS EXPENSIVE

THE REAL COST OF PLAYING IT SAFE

MINDSET • MOVEMENT • OPPORTUNITY • IMPACT

EFREN L. SALAZAR

FOUNDER & CEO | WAKE UP X SOLUTIONS

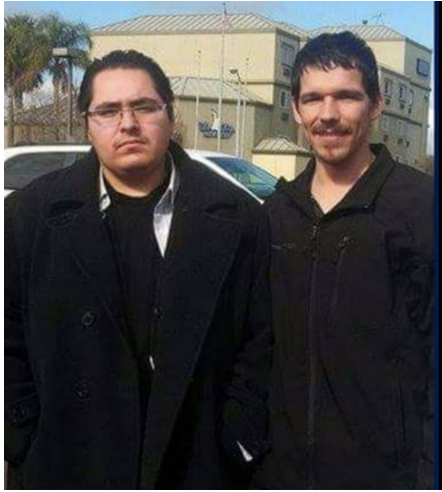
Comfort Is Expensive

The Real Cost of Playing It Safe

EFREN L. SALAZAR

Founder and CEO, Wake Up X Solutions · Host, The Wake Up X Podcast
Government & Regulatory Partner, ISACA Sacramento

My cousin. My first mentor. RNS Computer Repairs.



You told me I was lazy.
You were right.

But you saw strategic,
not weak — a gift,
not a flaw.

Rojelio Salazar

Nobody believed in me from day one. Nobody but you.

You would call, every time, and say it straight: "Wake up, man. Let's get the mission running." That call is why this movement carries your words in its name.

You lived that first, at RNS Computer Repairs — your name on your own sign. I'm not building this to repay you. I'm building it because it was always ours.

I miss you every day. This book belongs to you.

**Thank you for believing in me when no one else would.
Rest easy, cousin — we're living this dream together.**

COMFORT IS EXPENSIVE

The Real Cost of Playing It Safe

Copyright © 2026 Efren L. Salazar. All rights reserved.

Published by Wake Up X Solutions, Sacramento, California.

Part of the Wake Up X library: two gateway books plus the ten-book career library, Book 0 through Book 9. All of it free at www.wakeupxpodcast.com.

No part of this book may be reproduced or distributed in any form without the written permission of the author, except for brief quotations used in reviews or educational settings with attribution.

This is a work of nonfiction. The events described are recounted from the author's memory and to the best of his recollection. Certain dates are approximate. Some names have been withheld out of respect for the privacy of the people involved.

Educational content. Not professional, legal, financial, or career-placement advice. The views expressed are the author's own and do not represent those of any employer, client, or professional association.

First edition, version 1.0 · August 2026

www.wakeupxpodcast.com

Contents

- Read This First
- Where This Book Fits
- 01 Where I Started
- 02 The People Who Carried Me
- 03 What Staying Safe Cost Me
- 04 I Built My Own Door
- 05 The Four Words
- 06 How I Think Now
- 07 What I Have Built So Far
- 08 What I Am Building For You
- The Point of All of It
- Read Next
- Stay Connected
- About the Author

Read This First

Comfort is the most expensive thing I have ever paid for.

Let me be straight with you before you turn the page. This is not a get rich story. I did not find a shortcut. What I can give you is the truth about what it cost me to stay safe, what it took to leave, and what I found on the other side.

I was put in ESL classes, then special education. Teachers looked at me and said out loud that I could not learn. I did not speak English well until I was about sixteen. I had to read things ten or twenty times before they stuck. I did not get my high school diploma until I was almost thirty. By then I had already built and run real businesses without one.

Today I am the founder and CEO of Wake Up X Solutions. I host The Wake Up X Podcast, I partner with ISACA Sacramento's Government and Regulatory team, and I give away a library of ten books for free. Nothing about the first version of me predicted the second one.

Never let your current title decide your future potential.

So if a voice in your head says your starting point rules you out, I would like to stand here as proof that it does not.

One more thing. I did not do this alone. I am going to name every person who carried me, because the story where one guy pulls himself up by himself is a lie that keeps people stuck. Somebody always goes first for you. Your job is to notice who, thank them, and then be that person for someone else.

Where This Book Fits

This is one of four layers. You can start anywhere, but they were built to hand off to each other.

Wake Up X is not a pile of separate PDFs. It is one system with four entry points, and this book is the second layer.

Layer one is my memoir, *I Just Have To Wake Up*. That is the full life story, every person who carried me, and the whole timeline in order. If you want to know why any of this exists, start there.

Layer two is the book in your hands. This is the mindset. What comfort actually costs, how to leave it, how to think bigger, and why you turn around and help people once you are through.

Layer three is DECODE: The Wake Up X Playbook, which the library itself calls Book 0. That is the map, from zero experience to executive thinking, no degree required.

Layer four is Books 1 through 9, each aimed at one specific problem. Breaking in. Building proof. Judgment. Business. Leadership. AI. Brand. Consulting. Lasting. Book 0 plus those nine is the whole career library, ten books in total.

Story explains why. Mindset explains how you think. The blueprint explains what to do. The playbooks handle the specific problem you have today.

Four Layers, One System

Start anywhere. Every layer points to the next.

LAYER 1 • THE STORY

I Just Have To Wake Up

Why Wake Up X exists. The full life story, the people, the timeline.



LAYER 2 • THE MINDSET

YOU ARE HERE

Comfort Is Expensive

How to leave the comfort zone, think bigger, and give it back.



LAYER 3 • THE BLUEPRINT

DECODE: The Wake Up X Career Blueprint

The career map. Zero experience to executive thinking.



LAYER 4 • THE PLAYBOOKS

1 Break In • 2 Build It • 3 Think Like a Pro • 4 Business First • 5 Lead Without a Title

6 AI for IT & Cyber • 7 Build Your Brand • 8 The Consultant Mindset • 9 Level Up

Four layers, one system. This book is layer 2.

Choose your path

You do not have to read everything, and you definitely do not have to read it in order. Pick the path that matches where you are standing right now.

01 Break into tech. No experience yet and you need a first role. DECODE, then Break In, then Build It.

02 Break into cybersecurity. Already working in IT and you want to move across. Book 0, DECODE, then Think Like a Cybersecurity Professional, then Build It.

03 Sharpen your security and AI skills. You are already in and you want to get better. Think Like a Cybersecurity Professional, then AI for IT and Cybersecurity, then Level Up.

04 Business, consulting, and brand. You want to lead, consult, or be found. Business First, Lead Without a Title, Build Your Brand, then The Consultant Mindset.

05 Just the story. You want to know why any of this exists. I Just Have To Wake Up, then this book.

06 Give me everything. You are not sure which one fits. Take the whole library and start wherever you like.

Every one of these is free at www.wakeupxpodcast.com. That site is the hub. Everything else points back to it.

CHOOSE YOUR PATH

Pick the Path That Fits

You do not have to read everything, or read it in order.

BREAK INTO TECH

No experience yet, need a first role.

[DECODE](#) › [Break In](#) › [Build It](#)

BREAK INTO CYBER

Already in IT, want to move across.

[DECODE](#) › [Think Like a Pro](#) › [Build It](#)

SHARPEN SECURITY & AI

You are in, want to get sharper.

[Think Like a Pro](#) › [AI for IT & Cyber](#) › [Level Up](#)

BUSINESS, CONSULTING & BRAND

Want to lead, consult, or be found.

[Business First](#) › [Lead Without a Title](#) › [Build Your Brand](#) › [Consultant Mindset](#)

JUST THE STORY

You want to know why any of this exists.

[I Just Have To Wake Up](#) › [Comfort Is Expensive](#)

GIVE ME EVERYTHING

Not sure which fits? Take the whole library.

[All ten books, free, in any order you like](#)

No book is a dead end. Every path leads somewhere else. Free at wakeupxpodcast.com.

Where I Started

They told me I could not learn. I decided to prove them wrong.

My parents came here from Mexico as farm workers. No money. No house lined up. No English. They had both been working since they were five and six years old. My father only reached the third grade, and the part that stays with me is that he loved school. He loved math. He never got the chance. My mother was very young when she had me, a kid raising a kid, and she carried a load nobody that age should carry.

They built a life from nothing and then gave away whatever was left to help other people. Sometimes they got taken advantage of for it. Watching that took away every excuse I could have made for myself.

School did not go well. I was moved down a level, then down another, until I was in the most basic classes the school had. More than one teacher told me straight to my face that I could not learn. Not that I was behind. Not that I needed more time. That I could not. That was one of the lowest points of my life. It is also the moment something hardened in me. I remember thinking, I am never coming back to this place. Being told you cannot learn and actually not being able to learn are two very different things. It took me years to understand the school had mixed them up.

Nobody gave me a manual for school, for English, or for life. So I did to my own life what I later learned to do to computers. Watch it closely. Find the pattern. Take it apart. Put it back together well enough to use.

The neighborhood

I grew up in West Sacramento in the 1990s, when gang activity was at its peak. Friends went to prison. Friends overdosed. Friends died. Here is the part that should worry you, and it has nothing to do with danger. After enough years it stopped looking like tragedy to me. It started looking like a normal life.

That is what comfort does. It makes anything feel normal. You can get comfortable in a rough neighborhood, a dead end job, or a low opinion of yourself. A place does not have to be good to feel familiar.

How comfort actually traps you

It happens in four steps, and you rarely notice any of them while they are happening.

- 01 The known bad thing feels quieter. Whatever you are living with is at least predictable. The unknown is not.
- 02 Familiar starts to feel normal. Not good. Just normal. That is the switch flipping.
- 03 Normal becomes identity. You stop saying this is my situation and start saying this is just who I am.

04 Identity becomes a ceiling. Once it is who you are, you stop testing the roof at all.

The known bad thing is quieter than the unknown one. That is the entire reason people stay.

I want to say something about the people I grew up with, because everyone who was not there writes them off. Almost every one of them had a good heart. They lived badly and got judged for their tattoos and how they carried themselves, but the hearts were good. They used to tell me, you see the inside of us, not the outside. I still carry that. It is the same skill I use now in rooms full of executives. I read people.

PART 2

The People Who Carried Me

Nobody makes it alone. Somebody always goes first for you.

I was the only kid in my house. No brothers, no sisters. I was so quiet I barely spoke, and I could not speak the language everyone around me used. My older cousins translated the world for me. So the people who raised me next to my parents were friends and cousins. I am not writing a word about success without naming them.

Miss England

Sixth grade. She stayed after class on her own time, unpaid, and taught me the basics. Letters. Sounds. Reading. One on one. Nobody made her do it.

She is also the first adult in a classroom who told me that everybody learns differently. Written down it looks small. To me it was the first time anyone suggested I was not broken. If one teacher giving a little extra time can change a whole life, then everything I have built on mentorship is not a theory. I am the proof.

Rojelio Salazar

My cousin. He ran his own shop, RNS Computer Repairs. His own initials on his own sign. Understand what that looked like to a kid like me. Everyone I knew either worked for somebody else or worked outside the law. Rojelio did neither. He fixed computers and he answered to himself.

That is what people miss when they ask how someone from where I am from ends up running businesses. It was not ambition. It was that I saw it done once, by someone with my last name. One shop, one sign, and owning something stopped being a fantasy. He was the first person who ever sat down and showed me how business actually works, from the ground up. He drove an hour and a half just to check on me. He called every day. He believed in me before I believed in myself. He told me to get out of the streets, that he had already been through it, and to get into my business because I was already smart enough.

Jose Luis Bravo

My first friend and still my best friend. We were neighbors. Most kids drifted away. He stayed. We fought all the time because we knew exactly how to get under each other's skin, which is its own kind of love.

He got a job before I did, and for years people compared me to him. That guy is doing better than you. You should be doing the same. It hurt when I was young and it pushed me when I got older, until one day the thought finally landed. Stop comparing me to anybody. Look at what I can do.

I used to beg him to get me in where he worked. He said he had me, and then never followed through. Everyone expects me to be bitter. I am the opposite. He told me what they wanted and left the rest to me, and he was right to. I was not ready. If he had handed me that job I would have settled and stopped there.

Take the hint somebody gives you and be grateful. Do not expect them to carry the whole thing. Angel Martinez

Angel gave me hope when I had none. We were in the special classes together in seventh and eighth grade. He knew I could not speak English or stand up for myself, and he decided that was his problem too. He translated for me. He taught me the language. He spoke for me until I could speak for myself.

When teachers came at me for speaking Spanish, because Spanish was all I had, Angel pushed back. He told them that is all he knows, so if you want to teach him, teach him, let us do it together. He was younger than me and he acted like my big brother.

Isaac Murillo

Isaac was my first marketing department, years before I knew marketing was a job. He talked me up everywhere he went. Efren knows computers. Take your laptop to Efren. A big reason I believed I could make something of it is that someone else was already saying it out loud.

Alex De Haro

My cousin Alejandro, who everyone calls Alex. More brother than cousin. He pulled me under his wing in high school and kept me safe. He checked on me constantly. I had no car and no money, so he drove me where I needed to go and covered what I could not. He also gave me the line I still say to myself before I walk into any room. I used to tell him I was ugly and he was the good looking one. He said it is not how you look. It is never how you look. It is how you talk. That has held up in front of every executive I have ever met.

I was a puzzle nobody had put together yet. Every one of these people handed me a piece.

PART 3

What Staying Safe Cost Me

Every year I played it safe, I paid for it somewhere else.

Let me show you the receipts instead of talking in slogans.

I asked to get into a computer class in high school and got turned down because my grades were not good enough. Teacher after teacher told me I would never do it. I was not smart enough. I did not have the grades. The safe move was to believe them, and for a while I did. That belief cost me years I will never get back.

Then I tried Job Corp, because it looked like the easier road to a diploma. They promised three months. After I signed up I found out none of my credits transferred and it would take four years. The place felt less like a school and more like a holding pen, full of the exact environment I was trying to escape. I dropped out. Even the safe choice sent me a bill. Later, United Airlines gave me my first corporate job. Three months in, they fired me. I want to be clear about why, because the lazy version is that I was not good enough. That is not what happened. It was a communication problem, and it came from the top down. Nobody gave me the tools. Nobody gave me a structure to work inside. I was expected to perform in a system that was never explained to me.

I do not say that to let myself off the hook. I say it because it taught me how to lead. If your people do not know what the structure is, then the structure does not exist. Everything falls apart when nothing is organized, and that lands on whoever is running it.

Communication is not a soft skill. Communication is the structure.

Then Google showed me the opposite, through a contract with Astreya Partners in Mountain View. I started on a Friday, which is the funniest possible day to start there. Music. Events. Game systems everywhere. Someone said hey, it is Friday, let us go get some beers, and my honest reaction was, how are we drinking at work?

They provided everything. Good food, drinks, and a ride home so nobody drove. This was before Uber, so they put you in a taxi. It felt like an adult playground.

But the perks were not what got me. It was the structure holding them up. How organized it was. How friendly it was. How much thought went into making the place work. I had come from a folding cart in a hair salon, and here was a company that had designed the whole experience of working there.

Put those two jobs side by side and you have the lesson. One had no structure and it cost me the job. The other had structure everywhere, and that is why people gave it everything. That comparison was the first real thing I ever owned. Not money. A picture of what good looks like.

PART 4

I Built My Own Door

Nobody was going to open one, so I stopped waiting.

It started with a donated computer when I was thirteen. An old Windows machine with no working mouse. Most people would have called it broken. I taught myself to run the whole thing from the

keyboard. The start menu, the command prompt, all of it. Nobody showed me. I just refused to let a missing part be the reason I stopped.

That machine, not a class and not a certificate, is where my career actually started.

The job nobody wanted

I volunteered first, because there was no paycheck available to me. I volunteered until the Calling Teen Center in West Sacramento put me on payroll as a dishwasher and a janitor. I want that order stated plainly, because people skip the unpaid part when they tell these stories, and it is the most important part.

Dishes and floors. Not glamorous. But it put me inside a building that also ran a program called Computers for Kids, led by a Sac State professor named Lewis Bair. That is the whole argument for taking the job nobody wants. It puts you in the building. Computers for Kids had a simple idea. Families who could not afford a computer should have one anyway. Broken machines came in, volunteers rebuilt them, and a family walked out with one that worked. I asked Lewis if I could stay after my shift and learn, and he said yes.

Understand what that meant to a kid who had been told he could not learn. Nobody there asked about my grades. Nobody checked what track the school put me on. They handed me a broken machine and a screwdriver and let the work answer for me.

Nobody asked for my grades. They handed me a broken machine and let the work speak.

I started at the help desk, sitting with people who had never touched a keyboard. A lot of them were parents and grandparents from my own community, embarrassed about it. I knew that feeling from the other side of the desk, so I never made anybody feel slow. I have kept that rule in every room since.

I stayed from 2008 to 2013, mostly as a volunteer, and came back in September 2013 as their IT Manager. The kid who came in washing dishes ended up running the technology for the program that taught him.

Around the same time a nun named Sister Cora handed me a key and a small room where I taught English and computer basics to my community. She did not ask for a resume either. She handed me a key.

The business nobody gave me permission to start

By sixteen, neighbors were leaving computers on my doorstep. Twenty dollars for one repair, forty for another, some days close to a hundred and fifty. People knocked on my bedroom window to drop off machines. I was a teenager working dish shifts by day and running a business out of my front room at night.

I had no business name then. People just knew me as Efren the tech. The reason I had customers at all is Kyle Erni, who was fixing computers in West Sacramento before me. When Kyle left the

state, his customers came to me. Later he came back and we ran it together as CompuFix Sacramento.

When the market crashed and there was no work left in California, I moved to Salt Lake City in 2011 to find some, and the company came with me as CompuFix Utah. I ran it while working full days as a line cook to cover rent.

That kitchen is worth telling honestly. I am Mexican American, first generation, and I assumed that would count for something with coworkers from the same background. Instead they made sure I knew I did not belong. I was told more than once that I was not important enough and that I was never above them.

I had no comeback and I did not go looking for one. What I told myself was that maybe I did not have whatever they thought they had, but I had something else. I was willing to keep learning. I used it as fuel and never let them see it land.

Growing up I never fully fit the American side or the Mexican side. Standing outside both showed me a pattern I have seen in every room since. People push each other down to look taller instead of pulling each other up.

If almost nobody is willing to lift others up, that gap is exactly where you can make a difference.

When Kyle moved on, I took the business over completely and renamed it Compu-Fix 2u. A folding cart inside a hair salon and a phone number. Kyle and I are still in touch and he is a successful engineer now. Neither of us ended up where that folding cart would have predicted.

The partner who changed everything

A message on MySpace turned my repair work into a real company. Erick Arauza reached out with a simple idea. Open a store together. I had already fixed a computer for his cousin without ever meeting him. That one repair put my name in front of him.

We built The Cellphone Fix, and later his brother Pancho Arauza joined us. Seven years. Three locations at the peak. Customers driving hours to reach us. Two of those locations are still open and still making money today. I ran the technical side as Chief Technology Officer.

It worked because we were opposites. I handled the technology. Erick handled sales, and more importantly he taught me sales. Everything I know about selling and communicating traces back to him. Pancho was the one who would look at me and say stop being scared, you have the skills, just go do it.

The right partner covers the half of the business you are worst at. Stop looking for a mirror.

We closed one location, and not because business was bad. It was a bad contract. That one mistake pushed me to learn agreements, policy, and compliance, and it is the reason I moved into that side of the field at all. I do not tell that story as a warning. I tell it as a lesson I paid for.

Danny Maya and the camera

I got decent at marketing partly by watching my friend Danny Maya. Danny co-founded Maya Traditional Mexican Restaurant and Taqueria Maya, and he is a partner in Salt and Water. He was already talking to a camera for his own business while I was still hiding behind one.

He kept telling me it was my turn on the other side of the lens. You have a good personality. You are humble. You are honest. You do not lie to people. I kept saying I do not know. He pushed until I finally recorded something. It was not good. But I did it, and the only thought afterward was keep pushing yourself.

There is no podcast without that. Danny is also the clearest example of how everything in my life gets built. Friend first, business second. I am now director of marketing across his companies, and nobody ever posted a job for it. I have never gotten anything important through an application form.

PART 5

The Four Words

The break did not come when I was failing. It came when I was winning.

By 2016 I was a co-founder and CTO with three locations, a team, and customers who drove hours to use us. By any outside measure I had already made it out.

That is the year Rojelio died. I was twenty-six.

Sit with the order of those two facts, because it is the most important thing in this book. The crisis did not come because I was broke. It came while I was doing well and could not tell you what any of it was for. I had a real business and no purpose attached to it. Losing him took the floor out from under me. He was the one person who reached out and took my hand back when there was nothing to be proud of yet. I went into a depression afterward and felt like I had lost everything, and there was a lot of pain inside our family in that same stretch. It is still hard to talk about.

I am telling you anyway, because the polished version of a success story helps nobody, and because somebody reading this is standing in that exact spot right now. Then one ordinary day I was simply done being stuck. The words that went through my head were not a slogan somebody wrote for me. There were four of them.

I just have to wake up.

That is where the name comes from. Everything with Wake Up X on it was built after 2016. After him. The business existed before. The mission did not.

If you are waiting to hit bottom before you change, understand that plenty of people never hit bottom. They stay level forever, at a height that was never the goal.

PART 6

How I Think Now

The mindset came first. The results came after.

Erick handed me a book called The 10X Rule by Grant Cardone, and I took one idea from it. Whatever goal you set, multiply it by ten. Not because you will always hit it, but because aiming ten times higher changes what you are willing to try.

Aim for a hundred dollars a day and you stop at a hundred. Aim way past it and even falling short leaves you somewhere you never would have reached.

Now let me correct how most people repeat that idea. This is not about wanting more money. If money is the only reason you are here, you are looking at it wrong. You can make money anywhere. Real estate. Lending. Sales. The money is available. That was never the hard part.

If you are here only for the money, you can get money anywhere. What you cannot get anywhere is a purpose.

Here is what the mindset actually looks like, based on living it.

- 01 Treat your situation like a puzzle, not a verdict. Watch it closely. Find the pattern. Take it apart. Put it back together well enough to use.
- 02 Go where you are the least impressive person in the room. Not to feel small, but to borrow a higher standard. Every skill I have came from somebody who had it first.
- 03 Be willing to be a beginner again. I left a business I built and went back to entry level on purpose. Starting at the bottom was not a step back. It was a plan.
- 04 Build structure before anything else. Everything falls apart if nothing is organized. Systems survive bad weeks. Motivation does not.
- 05 Compete with your own last version. I spent years being compared to other people. The day that stopped mattering is the day everything started moving.

None of the five are about money. The money follows the thinking.

Rebuilding when nobody is watching

I learned this by rebuilding myself when everything else was already on fire. I lost a hundred pounds in seven months, training naturally, no shortcuts, while I was depressed. I have gained some of it back since, and I am telling you that on purpose. A story that only shows the win is useless to you. The number is not the point. The method is, and it is the same method I use on a stuck business or a stalled career.

- 01 Name it. Stop calling it being busy. Say the real problem out loud to another person.
- 02 Structure it. Turn the goal into a weekly routine instead of a burst of motivation.
- 03 Let people see it. Bring in people who notice the effort and hold you to it. Willpower alone runs out.
- 04 Repeat it. The plan does not need to be perfect. It needs to survive a bad week and still run.
- 05 Become it. At some point it stops being something you are trying and becomes who you are.

That rebuild turned into a brand, Obese to Obsessed. Not a diet plan. A reminder that the discipline that rebuilds a body will rebuild a career, and that none of it has to be done alone.

Consistency is the whole trick. It is boring, which is exactly why people go looking for a shortcut instead.

PART 7

What I Have Built So Far

From a folding cart to a room full of executives.

I want to lay this out plainly, not to impress you, but so you can see what came out the other side of everything in the last six parts. Almost every one of these started with somebody telling me no.

01 Three businesses under one roof. Compu-Fix 2u out of a folding cart, then The Cellphone Fix for seven years and three locations, then Wake Up X Solutions, founded the same month the shop closed.

02 Chief Technology Officer with no degree. No diploma at the time either. I ran the systems, the applications, the security, and the risk for a company with a team and real revenue.

03 Marketing that pays for itself. I raised online marketing return by 46 percent across the shop and Danny's restaurant, and I am now director of marketing across his companies.

04 One of five ambassadors out of more than fifteen thousand. At the healthcare organization where I work, I was selected as one of five enterprise ambassadors from a workforce of over fifteen thousand people.

05 Operations manager and solution architect. I started back at entry level as a field engineer in 2022. By February 2025 I was leading operations and designing solutions.

06 A partnership, not a title. I volunteered at ISACA Sacramento, then ran for Director of Communications against a CISO with more than twenty years of executive experience. I lost with 43.5 percent of the vote and gained a friend in Ken Kerrick, who won. Today I partner with the chapter's Government and Regulatory team, and plan to run for a leadership seat again.

07 A podcast that opens doors. I could not pay for mentors, so I offered exposure instead. My first guest was Gustavo Mastroianni, who had been watching my LinkedIn and was impressed by an old marketing video from my repair shop days. I asked, he said yes, let us do it.

08 A diploma and a certification, on my terms. I finished high school at almost thirty and earned CompTIA Security+, not because I needed them to succeed, but to close the chapter myself.

09 More than fifteen people hired. The same blueprint I wrote down has helped more than fifteen people get jobs. That is not a claim. That is a record.

10 A free library. Ten books, given away, not sold.

Almost every one of these started with somebody telling me no. The people who pushed me there

Nate was a contractor with me at the healthcare organization. I saw more in him than he saw in himself and I would not let it go. I kept pushing him and handed him everything I knew, and we both ended up hired full time.

Igor was the tech one and I was the tech two and the lead, and neither of us ever cared about those numbers. He is from Ukraine and holds a cybersecurity degree. I already knew the work. What he gave me was the vocabulary I had never been taught.

He is also the one who broke my stubbornness about credentials. He told me it is just a piece of paper, you already have the skills, what are you going to lose. So I went and finished my diploma in a few months and went after Security+. In return I taught him sales, communication, and interviews, and handed him my connections. He got a cybersecurity analyst job. We built each other.

Surround yourself with people who use your own advice on you. That is how you find out you were not taking it.

And a man named David De los Reyes wrote a testimonial for me, which proves the whole argument of this book. We have never met in person. Not once. The entire relationship happened over LinkedIn, and he went on to introduce me to his CEO, John, a serial entrepreneur out of Oakland. You will always need new people.

PART 8

What I Am Building For You

Ten books. A podcast. A community. All of it free.

There is a habit in my field I set out to break. Some people guard what they know. Ask them how they got in and the door quietly shuts, because helping you might cost them their edge.

I watched enough of that to reach a conclusion. The gap is not talent or intelligence or work ethic. It is access. Somebody who already made the jump you are trying to make, willing to tell you honestly how they did it. Most people never get to meet the person one step ahead of them.

I did not invent the answer either. I learned it at Computers for Kids and I have been running it ever since. A free library instead of free computers. Mentorship instead of after shift lessons. Older people learning technology for the first time, exactly like the ones I sat with at that center.

Be the lift you never got.

Here is what to actually do, in the order it worked for me.

01 Get in the building. I washed dishes to be inside a building with a computer lab. The title was never the point. Being there was.

02 Ask the direct question. I asked Lewis Bair if I could stay after my shift and learn. He said yes. Almost nobody asks, which is exactly why asking works.

03 Build something people can see. Every client, job, and opportunity I have come from building something visible and then showing up to back it up.

04 Trade value for mentorship, not money. I could not pay mentors, so I paid them in exposure. Work out what you have that they want.

05 Close your own loop. I finished my diploma at almost thirty, not because I needed it, but to stop carrying it.

06 Hold the door. The moment you get through, somebody behind you is standing where you were. That is not charity. That is the entire point.

The library, and why each book exists

I could not find the map when I needed it, so I wrote the one I wish somebody had handed me at nineteen, at twenty-six, and again at thirty-five. People call these secrets. They are not secrets. They are things nobody bothers to say out loud. Start with the problem you actually have.

BOOK 0 DECODE: The Wake Up X Playbook

Zero experience to executive thinking. No degree required.

Start here if you do not know where to start, or if you have been trying for a year and nothing is moving. This is the whole system in one place.

BOOK 1 Break In

How to land your first role, from application to offer to your first ninety days. Read this if you are applying and hearing nothing back. Most people are not losing on skill. They are losing on process.

BOOK 2 Build It

Home labs and a portfolio that prove you can do the work.

Read this if you have no experience and keep being told you need experience. A build breaks that loop. A donated computer with no mouse is where my whole career started. BOOK 3 Think Like a Cybersecurity Professional

Judgment, triage, and response. It is a judgment problem, not a tool problem. Read this if you know the tools but freeze when something real happens. Tools are teachable in a weekend. Judgment is what gets you kept.

BOOK 4 Business First

Money, risk, and why the business hired you.

Read this if you are technical and still getting overlooked. I learned this the hard way when a company refused to pay me and I had to teach myself the rules from scratch. BOOK 5 Lead Without a Title

Influence, feedback, and mentoring before anyone promotes you.

Read this if you are waiting on a title to start leading. Leading is a behavior, and people notice it long before HR does.

BOOK 6 AI for IT and Cybersecurity

Using AI as a working tool, not a novelty.

Read this if you are either scared of AI or trusting it too much. Both are expensive. BOOK 7 Build Your Brand

Being findable. Visibility gets you the opportunity you are not qualified for yet. Read this if you are good and nobody knows it. A testimonial from a man I have never met opened a door for me. That is not luck. That is being findable.

BOOK 8 The Consultant Mindset

Discovery, scope, and working with people who hold the budget.

Read this if you want to consult or run your own thing. I lost a location over a contract I did not read carefully enough, and that mistake is why this book exists.

BOOK 9 Level Up

Long term growth, burnout, and mentoring others.

Read this if you already got in and it is starting to cost you. Lasting is its own skill.

If one of these helps you, send it to somebody who needs it more than you did.

Start with the problem you have, not with page one.

The Point of All of It

I do not tell this story so you feel something about me. I tell it so you trust that the advice was earned under real pressure and not borrowed from a classroom.

I was told I could not learn. I learned anyway. I was told I was not important enough. I built something that made the question irrelevant. Nobody gave me an opportunity, so I made one, and then spent the rest of my career making them for other people.

I want to name one more person, because this is being written in 2026 and he is in it right now. Alvin Jue believed in me through this whole stretch, and more than that, he is there on the days when it feels like none of it is working. Everybody shows up for the finished version. Very few people stand next to you in the middle of it, when the numbers are flat and the doubt is loud. That is what a friend actually is.

And my parents are still here to see all of it. They knew the kid who was around the wrong people. They know the man who mentors now. They tell me they are surprised, and I understand why. Nothing about the first version of me predicted this one.

My own parents did not see it coming. It happened anyway.

It does not matter what certificate you hold or where you are starting from. If you have hunger, ambition, and the will to keep showing up, you can build something that matters. Always go for it. Do not give up. Stay focused. Surround yourself with people smarter than you. Teach yourself, and if you do not have the resources, go find them. There are no excuses left now that a mentor is one message away.

Comfort is expensive. Pay attention to what yours is charging you.

Securing Tomorrow. Empowering Today.

If you want the longer version of this story, it is in my memoir, *I Just Have To Wake Up*.

Read Next

Pick the one that matches where you are standing right now.

If you want the whole story

Read I Just Have To Wake Up. Everything in this book is compressed. The memoir is the full version, in order, with every person named and every year accounted for. Miss England, Sister Cora, Lewis Bair, Kyle Erni, Erick and Pancho, Danny Maya, Igor, Nate, Ken Kerrick, Jose Luis, Angel, Isaac, Alex, Alvin, and my parents. If a story in here made you want the rest of it, that is where the rest of it is.

If you are ready to move

Read DECODE: The Wake Up X Playbook, listed as Book 0 in the library. This book is about how to think. DECODE is about what to do, step by step, with no degree and no connections required. Mindset without a map turns into motivation that fades by Friday.

If you already know your problem

Go straight to one of the nine playbooks. If nobody is answering your applications, that is Break In. If you have no experience to point at, that is Build It. If you are technical and still overlooked, that is Business First. Do not read them in order. Read the one that matches the wall in front of you.

All of it is free at www.wakeupxpodcast.com. Nothing is held back for a paid tier.

Stay Connected

This book is one part of a bigger mission. The podcast, the free library, and a community built for anyone who was ever told no.

Podcast & Free Library: www.wakeupxpodcast.com

LinkedIn: [linkedin.com/in/efrenlsalazar](https://www.linkedin.com/in/efrenlsalazar)

YouTube: [youtube.com/@Efren.wakeupx](https://www.youtube.com/@Efren.wakeupx)

Be a Guest on the Show: wakeupxpodcast.com/guest

About the Author

Efren L. Salazar is the founder and CEO of Wake Up X Solutions and the host of The Wake Up X Podcast. He partners with ISACA Sacramento's Government and Regulatory team, and he holds CompTIA Security+.

He was born to Mexican farm-working parents in the Sacramento area, was placed in ESL and then special education classes, and was told by teachers that he could not learn. He fixed his first computer at thirteen, a donated machine with no working mouse, and ran a repair business out of his front room as a teenager.

He spent five years at Computers for Kids, first as a volunteer and eventually as their IT Manager, then co-founded The Cellphone Fix and served as Chief Technology Officer across seven years and three locations. He finished his high school diploma at close to thirty, after having already built and run multiple businesses without one.

He now works in enterprise IT operations and solution architecture, where he was selected as one of five enterprise ambassadors from a workforce of more than fifteen thousand. Through Wake Up X he gives away a library of ten books at no cost, mentors people trying to break into the field, and has helped more than fifteen people get hired.

He lives in the Sacramento area.

Securing Tomorrow. Empowering Today.

Acknowledgments

THE PEOPLE WHO MADE THIS POSSIBLE

None of Wake Up X happened alone, including Comfort Is Expensive. Every book in this library, the podcast, and the free library exist because specific people showed up for a kid nobody expected much from. This is the record of who they are.

THE EARLY YEARS

Miss England, the 6th-grade teacher who stayed after class, unpaid, to teach the fundamentals. Sister Cora and Lewis Bair of Computers for Kids, who handed a teenager a key and a broken machine and let the work speak. Jose Luis Bravo, Angel Martinez, Isaac Murrilo, and Alex De Haro — the friends and cousins who translated the world, gave hope, drove him where he needed to go, and carried him before there was anything to carry him toward.

THE BUSINESS YEARS

Rojelio Salazar, his cousin and first mentor, RNS Computer Repairs — the reason it is called Wake Up X. Kyle Erni, his first business partner. Erick and Pancho Arauza, co-founders of The Cellphone Fix. Danny Maya, of Maya Traditional Mexican Restaurant, Taqueria Maya, and Salt and Water, who pushed him in front of a camera for the first time.

THE CAREER YEARS

Igor, who gave him the vocabulary for work he already knew how to do. Nate, who he mentored into a full-time role alongside his own. Ken Kerrick, opponent turned friend in the ISACA Sacramento election. Alvin Jue and David De los Reyes, who believed in the middle of it, not just at the finish.

THE PODCAST

Every guest who trusted an unproven show with their story, including Gustavo Mastroianni, Cathy Stephenson, Greg McCord, Aaron Portnoy, Sana Talwar, Jorge Avila, Anil Reddy, Devin Prasad, Larry Switzer, Louis Kemp, Mohana Balakrishnan, Eddie Lopez, and Michael Fierros — full episodes at wakeupxpodcast.com.

COMMUNITY PARTNERS

ISACA Sacramento and ISC2 Sacramento, The Cellphone Fix, A.N.U.L Technologies, AISOC, AspireByte, and PodGrabber — the organizations building alongside this movement.

AND

His parents, who arrived with nothing, worked since childhood, and gave away whatever was left over. Nothing about the first version of this story predicted the version you are holding now.

Continue the Journey

EVERY BOOK LEADS SOMEWHERE ELSE

Comfort Is Expensive is one part of a 12-book system. If it helped, the next step is one page away —take whichever layer matches where you are standing.

THE LIBRARY

Layer 1—The Story

I Just Have To Wake Up —why any of this exists.

Layer 2—The Mindset

Comfort Is Expensive —the real cost of playing it safe.

Layer 3—The Blueprint

DECODE: The Wake Up X Playbook —zero experience to executive thinking. Book 0.

Layer 4—The Playbooks

1 Break In · 2 Build It · 3 Think Like a Cybersecurity Professional · 4 Business First · 5 Lead Without a Title · 6 AI for IT & Cybersecurity · 7 Build Your Brand · 8 The Consultant Mindset · 9 Level Up

FIND US EVERYWHERE

Podcast & free library: wakeupxpodcast.com · Spotify: open.spotify.com/show/28GZnRBrIV2efHOhZdiftt · Apple Podcasts: podcasts.apple.com/us/podcast/id1792821643 · YouTube: youtube.com/@Efren.wakeupx · LinkedIn: linkedin.com/in/efrenlsalazar · Substack: substack.com/@wakeupxefren

IN GOOD COMPANY

ISACA Sacramento, ISC2 Sacramento, The Cellphone Fix, A.N.U.L Technologies, AISOC, AspireByte, and PodGrabber —the partners and collaborators building alongside the movement. See the full list at wakeupxpodcast.com/community.html.

Get DECODE free, apply to be a guest, or read the story behind the movement —all of it starts at wakeupxpodcast.com.

WAKE UP X

SOLUTIONS



Efren L. Salazar

Founder & Host, The Wake Up X Solutions

Government & Regulatory Partner, ISACA Sacramento · CompTIA Security+

STAY CONNECTED

Podcast & Blueprint Library:	wakeupxpodcast.com
LinkedIn:	linkedin.com/in/efrenlsalazar
YouTube:	youtube.com/@Efren.wakeupx
Be a Guest:	wakeupxpodcast.com/guest

◆ THE WAKE UP X MOVEMENT · FREE FIELD GUIDE

DECODE

The Wake Up X Career Blueprint

Zero Experience to Executive Thinking. No Degree Required.

You don't need a degree, a connection, or a lucky break. You need the map. This is the exact path from complete beginner to executive-level thinking.

ENTRY

FIT

BREADTH

DIFFERENTIATION

OWNERSHIP

EXECUTIVE

THE
WAKEUPX
SOLUTIONS

Efren L. Salazar

Host, Wake Up X Podcast · Founder & Chairman, The Wake Up X Solutions
ISACA Sacramento Board · CompTIA Security+

SECURING TOMORROW. EMPOWERING TODAY.

wakeupxpodcast.com

BEFORE YOU READ

What This Is, and Isn't

This library is educational content, built from lived experience, publicly available research, and the stories of Wake Up X podcast guests. It reflects one person's path and judgment, not universal rules.

It is not professional legal, financial, tax, or accounting advice, and it is not a guarantee of employment, income, or certification outcomes. Salary figures, growth statistics, and industry data are cited from named public sources at the time of writing and can change — verify anything time-sensitive before acting on it.

Lab instructions, scripts, and technical configurations are for learning in a personal or isolated lab environment. Test and verify before applying anything to a production system or a work environment. Tool, vendor, and certification names belong to their respective owners and are referenced for education only.

For anything involving business formation, contracts, taxes, or employment law, talk to a licensed professional in your jurisdiction. This library will point you toward the right questions to ask — it can't replace the person qualified to answer them for your specific situation.

SHARE FREELY

You're encouraged to share this library freely, in full, with attribution.
Please don't sell it.

THE WAKE UP X MOVEMENT

Advocate. Community Builder.

I host a podcast. I'm a partner with ISACA Sacramento. I've built a community for people trying to break into this field the way I did, with no map and no connections. What I haven't done yet is stand on a big conference stage, and I'm not going to pretend otherwise.

This library, the podcast, and the community are the real record. The stages are next — and when that happens, this page gets updated with the real photos, not before.

WHERE TO FIND THE REAL RECORD

The podcast: wakeupxpodcast.com · The community: wakeupxpodcast.com/voices
Connect directly: [linkedin.com/in/efrenlsalazar](https://www.linkedin.com/in/efrenlsalazar)

AS SEEN AT

More From the Field

Real moments with the ISACA Sacramento community at the REIGNITE conference — the rooms where this movement actually happens.



ISACA Sacramento REIGNITE conference



ISACA Sacramento REIGNITE conference

BEFORE YOU START

Why I'm Showing You Any of This

Every photo and story in this library exists for one reason: not to prove I'm special, but to prove the opposite. I didn't have a blueprint when I started. No mentor handed me a plan, no degree opened a door, no connection got me in early. I built the plan because I needed one and it didn't exist yet.

So when you see a storefront, a conference badge, or a shirt with a brand on it, don't read it as "he had something you don't." Read it as evidence: an unfinished-feeling path, worked over years, one unglamorous rep at a time, actually adds up to something. That's the only claim this library makes on my behalf. Everything else in it is for you — your first ninety days, your first lab, your first post, your first team.

THE ONLY RULE THAT MATTERS

If a framework in this library came from somewhere else — sales, marketing, leadership — that's on purpose. Take it, remix it, make it yours. This was never about copying me. It's about you not having to start from zero the way I did.

Networking & GRC Foundations

Two pieces of language every technical career eventually needs: the seven layers that all networking builds on, and the standards that turn security work into something the business side can trust.

The OSI Model — all seven layers

7	Application	<i>HTTP, DNS</i>
6	Presentation	<i>TLS, encoding</i>
5	Session	<i>sessions</i>
4	Transport	<i>TCP, UDP</i>
3	Network	<i>IP, routing</i>
2	Data Link	<i>MAC, switching</i>
1	Physical	<i>cables, Wi-Fi</i>

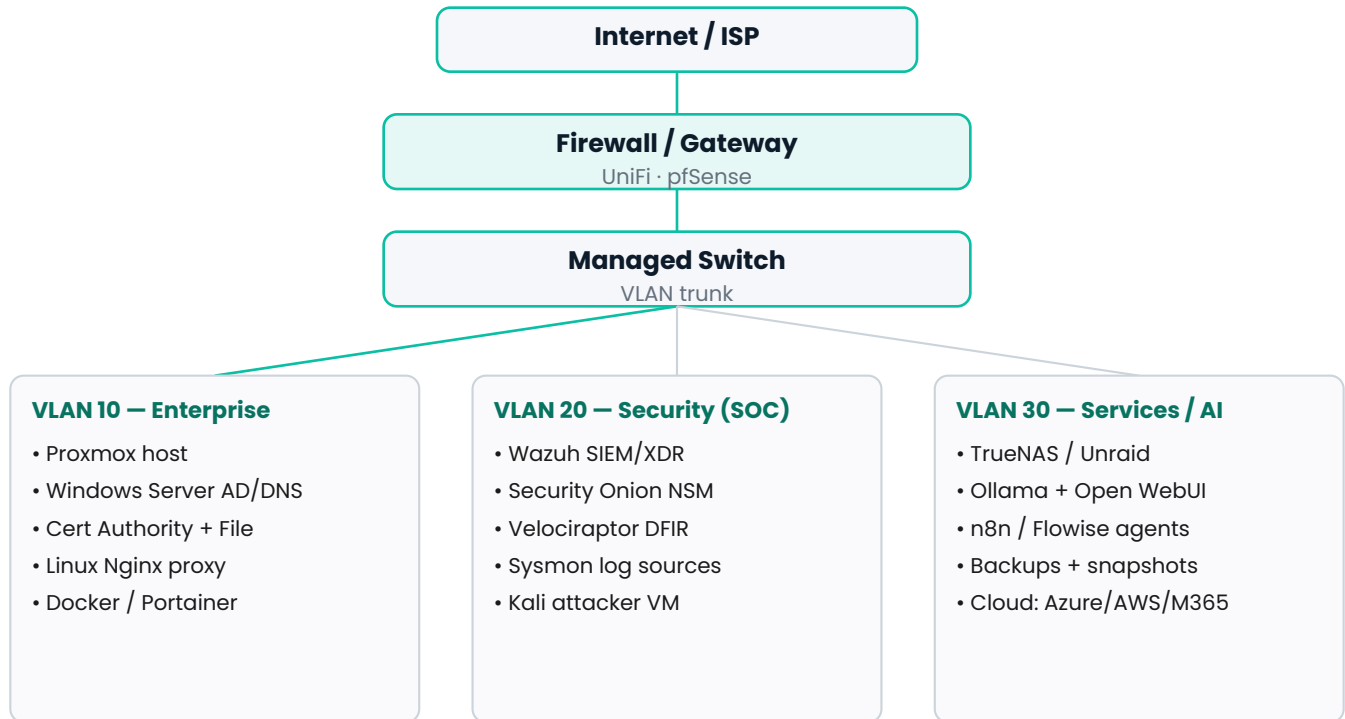
"All People Seem To Need Data Processing" — the classic memory trick, bottom to top.

GRC Standards at a Glance

PCI DSS 4.0.1	Protecting cardholder / payment data
HIPAA	U.S. healthcare data privacy & security
GDPR	EU personal-data privacy rights
SOC 2	Service-provider trust & controls
ISO/IEC 27001:2022	Information security management (ISMS)
NIST CSF 2.0	Govern, Identify, Protect, Detect, Respond, Recover
CIS Controls v8.1	Prioritized, practical security baseline

One Diagram to Build Toward

A realistic segmented home lab — the kind you can defend in an interview. Start with one mini-PC and grow into it. Every track in this book's home-lab curriculum maps to a box below.



More Acronyms & a Resume Bullet Bank

A few more terms worth knowing cold, plus a formula and stealable bullets for turning your lab work into something a resume can carry.

Extended Glossary

APT

Advanced Persistent Threat

CA

Certificate Authority

CIA Triad

Confidentiality, Integrity, Availability

CVE

Common Vulnerabilities & Exposures

DFIR

Digital Forensics & Incident Response

EDR / XDR

Endpoint / Extended Detection & Response

IAM

Identity & Access Management

IDS / IPS

Intrusion Detection / Prevention

MFA

Multi-Factor Authentication

NAT

Network Address Translation

OSI

Open Systems Interconnection

PKI

Public Key Infrastructure

RBAC

Role-Based Access Control

SOC

Security Operations Center

SSO

Single Sign-On

VPN

Virtual Private Network

ZTNA

Zero Trust Network Access

PoLP

Principle of Least Privilege

Resume Bullet Bank — steal and tailor

Formula: action verb + what you built/the tool + measurable result or business value.

Deployed a Wazuh SIEM across a 3-host lab, alerting on brute-force logins in under 10 seconds.

Built a segmented network (3 VLANs, firewall, reverse proxy) isolating a SOC from enterprise services.

Stood up Active Directory with Group Policy and a Certificate Authority for 50+ simulated users.

Automated a recurring task with Python/PowerShell, cutting a 30-minute job to under 2 minutes.

WHERE SHOULD YOU START?

Click a level below to jump straight to the part of this blueprint built for you right now.

ENTRY-LEVEL

Never worked in tech or cybersecurity yet, or still finding your footing in your first year.

Start at Level 1: The Basics of Breaking In >>

MID-LEVEL

You're in a role already and ready to add the skills that make you rare, not just qualified.

Start at Level 4: The Multiplier Skills >>

EXECUTIVE-LEVEL

Already leading, or aiming for it, and want a repeatable way to decide and lead under pressure.

Start at Level 6: Think Like an Executive >>

Prefer to read start to finish? Just turn the page, the blueprint is built as a complete path either way.

KEEP GOING

This blueprint pulls directly from conversations with real cybersecurity and tech leaders on the Wake Up X podcast. If you want the full, unedited version of any story referenced here:

- ▶ Subscribe on YouTube: youtube.com/@Efren.wakeupx
- ▶ Listen on Spotify and Apple Podcasts
- ▶ Everything else: wakeupxpodcast.com

WHY THIS BLUEPRINT EXISTS

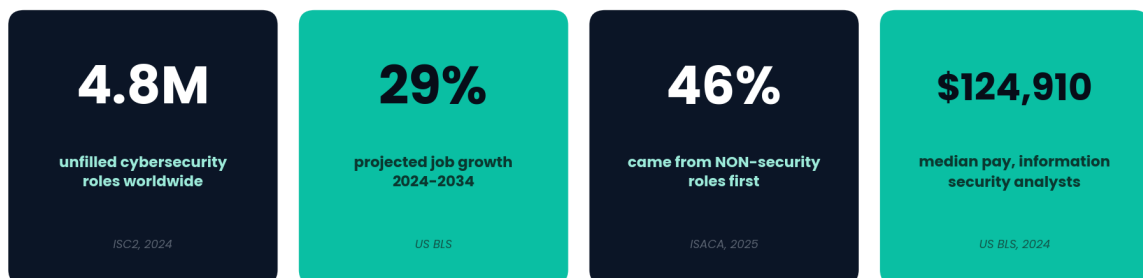
You don't need a degree. You don't need connections. You don't need a lucky break. You need the map, and almost nobody hands it to beginners for free. This is that map.

The numbers behind the opportunity

Before any advice, here's the business case for this field, in real numbers from primary sources, not hype.

THE OPPORTUNITY, IN NUMBERS

The demand is real, the pay is real, and almost half of the people doing this job started somewhere else

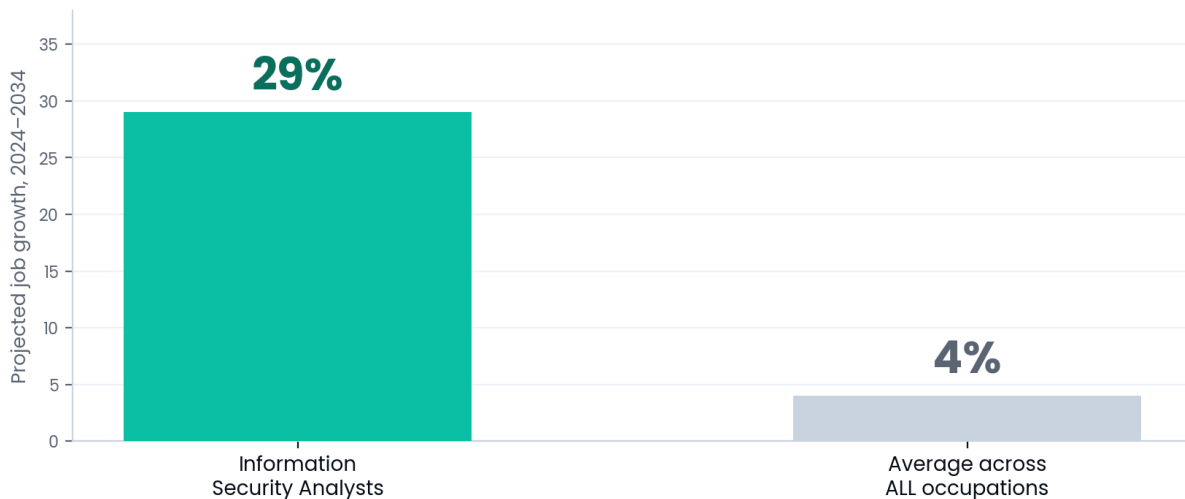


Sources: ISC2 Cybersecurity Workforce Study 2024 · US Bureau of Labor Statistics · ISACA State of Cybersecurity 2025

The Opportunity, In Numbers

- **Cybersecurity's real gap has shifted from headcount to skills.** ISC2's 2025 study found 95% of teams face a skills gap, up from 44% in 2024, against a global workforce of about 5.5 million. Building real skills matters more than chasing an empty seat.
- **29% projected growth for information security analysts between 2024 and 2034**, per the US Bureau of Labor Statistics, versus roughly 4% average across all occupations. That's about seven times the typical rate.
- **\$124,910 median pay** for information security analysts, US BLS 2024 data.
- **46% of current cybersecurity professionals came from non-security roles first** (ISACA, 2025). Nearly half of this industry is career changers. You are not late, and you are not an outsider.

GROWING 7x FASTER THAN THE AVERAGE JOB



Source: US Bureau of Labor Statistics Occupational Outlook Handbook

Growing 7x faster than the average job

One more number worth sitting with: the World Economic Forum found only **14% of organizations say they have the skilled people they need**. The demand is not theoretical. The gap is structural, and it's the reason a blueprint like this matters more than another certification ad.

A path, not a pep talk

Most career advice in tech and cybersecurity comes from one of two places: someone who broke in twenty years ago and forgot how confusing it actually is, or someone selling a certification course. This is neither.

I'm Efren L. Salazar, founder and host of The Wake Up X Solutions, a weekly business, cybersecurity, and AI podcast based in Sacramento, California. Every episode is an honest, unscripted conversation with someone who made it into tech, from newcomers to founders

to security leaders, asking them exactly how they did it. This blueprint is what happens when you take all of those conversations and compress them into one usable map.

I worked my way up from entry-level jobs into technology without a degree, without early access to the industry, and without anyone handing me the path in advance. So this is the map I didn't get. There's a full chapter on my own story later, not because it's the point, but because you deserve to know the advice here was actually earned, not borrowed.

The mission behind this blueprint

Wake Up X is a movement, not just a podcast. The whole reason it exists is simple: **most people never get to meet the person who's one step ahead of them.**

That's the gap. Not talent, not intelligence, not work ethic. Access. Somebody who already made the jump you're trying to make, willing to tell you honestly how they did it.

So the promise of this blueprint is the same as the promise of the show:

- ▶ **Build the connections you were never given.** You do not need connections to belong in this field. You build them.
- ▶ **Real advice, zero gatekeeping.** Nothing here is held back for a paid tier.
- ▶ **Built for every stage of the climb,** entry-level to executive to founder.
- ▶ **Closing the gap, one story at a time.**

Everything in here is tested against two things: what actually worked for the guests on the show, and what actually worked in the work I've done and the teams I've led. Where I reference a guest's story, you can go hear the full, unscripted conversation. This document is the blueprint. The podcast is the library behind it.

Nothing here is original, and that's the point

Almost every idea in this blueprint has been said before, by someone else, somewhere else. The four beliefs in Part Two, the sales conversation framework, the discipline of documenting your work, none of it was invented in a vacuum. What made it useful wasn't the invention. It was the remix.

Cybersecurity and IT are unusually isolated fields. The work rewards people who can sit alone with a problem, and that same instinct often keeps people away from sales, communication, marketing, and business thinking, skills that belong to a completely different world. I noticed that gap early: a field full of sharp technical minds who'd never been taught how to talk about their own value. So I took what already worked in sales, marketing, and leadership, and bridged it into cybersecurity and IT, on purpose, because nobody else was doing it for this specific audience.

You should do the same thing with everything in this blueprint. Don't just copy it. Take the frameworks, the tactics, the mindset, and bend them into your own shape. Every influencer, author, and mentor worth listening to is doing exactly that: passing along the same core truths, filtered through their own experience. That remix is where your own voice actually comes from.

How to use this blueprint

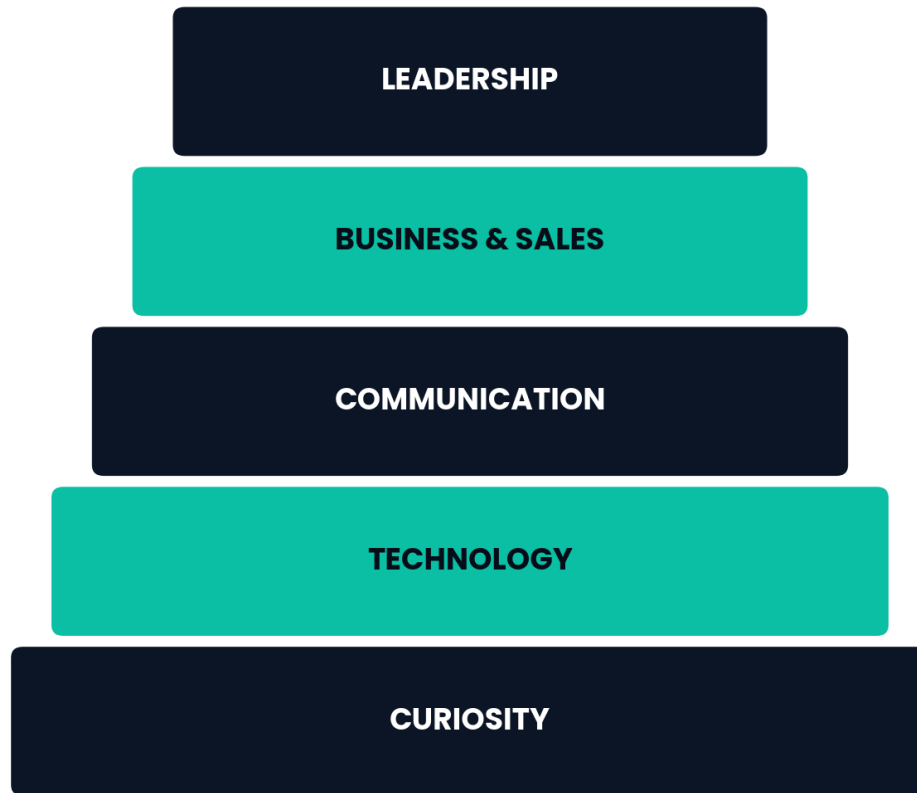
Read it in order the first time. It's built as a path: basics, then fit, then breadth, then differentiation, then business sense, then leadership, then a system to keep yourself honest about progress. Come back to individual sections once you're actually in a role and need the next level.

The Career Value Stack

Before the six levels, here's the underlying shape of what you're actually building. Every level in this blueprint adds one more layer to a stack, and each layer only holds weight because of the one underneath it.

- ▶ **Curiosity** is the foundation. It's the willingness to poke at something you don't understand yet.
- ▶ **Technology** sits on top of curiosity. It's the actual technical skill, whichever lane you land in.
- ▶ **Communication** sits on top of technology. Skill nobody can understand or trust is skill that stays invisible.
- ▶ **Business & Sales** sits on top of communication. Once people understand you, the next question is whether what you do connects to something they actually value.
- ▶ **Leadership** is the ceiling, and it's not a title. It's what happens when the other four layers are strong enough that people start asking you to lead them.

THE CAREER VALUE STACK



Each layer stands on the one below it — curiosity is the foundation, leadership is the ceiling

The Career Value Stack

Skipping a layer doesn't make you faster. It makes the stack wobble. A lot of technically brilliant people get stuck for years because they built technology skill on a thin curiosity base, or leadership ambition on a thin communication base. This blueprint is organized to build the stack in order, on purpose.

Let's start with the whole map.

01

PART ONE

THE PATH, START TO FINISH

Six levels, one map

Before the details, see the whole shape of where this is going. Every level builds on the one before it. Skipping a level doesn't speed you up, it just means you'll have to double back later.

- ▶ **Level 1 — Entry.** You understand the basic landscape of tech and cybersecurity well enough to know what you don't yet know.
- ▶ **Level 2 — Fit.** You know which lane actually matches how you think, not just which job posting sounded good.
- ▶ **Level 3 — Breadth.** You've cross-trained enough adjacent fundamentals that you're never the person who says "that's not my area."
- ▶ **Level 4 — Differentiation.** You've added the sales, communication, and business-translation skills that make a specialist rare instead of replaceable.
- ▶ **Level 5 — Ownership.** You understand how business actually works, budgets, contracts, risk, well enough to make decisions that hold up.
- ▶ **Level 6 — Executive Thinking.** You can lead, decide, and communicate at the level of the room you want to be in, before you have the title that says you belong there.

This isn't a beginner's book. Every level in this map applies whether you haven't started yet, you're five years into a technical role, or you're already leading a team and trying to think like the executives above you. Read the levels that match where you are, then keep the rest as the map for where you're headed.

THE WAKE UP X SIX-LEVEL PATH

Every level builds on the one before it – skipping one just means doubling back later



The Wake Up X Six-Level Path

TRY THIS ON FOR SIZE

Two people start the same week in IT support. One spends two years only learning ticket-resolution shortcuts. The other spends the same two years also learning basic networking, watching how their manager makes decisions, and practicing explaining technical problems in plain language to non-technical coworkers. Same job, same time. Only one of them is actually climbing the six levels.

Reflect before you move on

- ▶ Which of the six levels are you actually operating at right now, honestly, not where you wish you were?
- ▶ What's the one level you've been skipping past without realizing it?

02

PART TWO

THE MINDSET

The four beliefs underneath every level

Frameworks and certifications matter, but none of them stick without the mindset underneath them. These four beliefs show up, in some form, in almost every guest conversation on the show, regardless of their lane or level.

1. Create your own opportunity. If nobody will give you a chance, build something that makes them notice: a home lab, a project, a volunteer role, a small business. Proof beats permission. Waiting for someone to open a door is a strategy with a very long, very uncertain timeline. Building something real, even small, gives you a door you can open yourself, and it's usually the thing that makes the "official" doors open faster too.

2. Consistency beats intensity. One hour a day for a year outperforms a heroic weekend you never repeat. This is the least exciting belief on this list and the most reliable one. Nobody burns out from an hour a day. People burn out from trying to cram a year of growth into a single unsustainable sprint, and then quietly quit when the sprint ends.

3. Your network is your net worth. Most doors open because of a person, not a posting. Job boards are where opportunities go to get buried under three hundred other applicants. A real relationship, even a thin one, skips the line almost every time. This belief gets its own full framework later in this book, the Connection Flywheel, because it's too important to leave as a one-line slogan.

4. Learn in public. Share what you're studying. It attracts mentors and opportunities before you feel ready. Nobody feels ready. The people who look confident online are just the ones who posted anyway.

"If nobody wants to give me the opportunity, I am going to make my own opportunity."

EFREN L. SALAZAR, HOST, WAKE UP X

Books that shaped this mindset

- ▶ **Can't Hurt Me**, David Goggins, on building tolerance for discomfort as a trainable skill, not a personality trait
- ▶ **Atomic Habits**, James Clear, on why consistency compounds and intensity doesn't
- ▶ **The 10X Rule**, Grant Cardone, on setting targets big enough that "create your own opportunity" stops being a suggestion and starts being the only option
- ▶ **Ego Is the Enemy**, Ryan Holiday, on staying teachable once you've made some progress

TRY THIS ON FOR SIZE

Two people apply to the same fifty jobs over three months. One waits by the inbox. The other spends that same window building a small home-lab project, posting about it twice a week, and messaging three people a week who work in the field. By month three, the second person has had eleven real conversations and two referrals. The first person has had four rejection emails and a lot of silence. Same fifty applications. Different mindset entirely.

Reflect before you move on

- ▶ Which of the four beliefs do you already live by, and which one do you avoid?
- ▶ What's one thing you could "learn in public" this week, even in a small way?

03

PART THREE

WHERE THIS BLUEPRINT CAME FROM

One chapter on my path, then back to yours

I'm including this chapter because I want you to know exactly what "starting from nothing" looked like for me, not so you feel anything about me, but so you trust that the rest of this book was earned the hard way and actually works.

I grew up in a farm-working, immigrant family. Money and resources were both tight. Teachers placed me in special education classes and, for years, told me directly that I couldn't learn. I didn't speak English as a first language and didn't really pick it up until around sixteen. I got my high school diploma as an adult, almost thirty years old.

Looking back, most of what I actually did in those years was social engineer and reverse engineer my own life. Nobody handed me a manual for how school worked, how English worked, or how a technology career worked, so I did to my own circumstances what a good analyst does to an unfamiliar system: watch closely, find the pattern, take it apart, and rebuild it well enough to use it. That instinct, treat the thing you don't understand as solvable instead of as a verdict on you, is the actual origin of every framework in this book.

It didn't start as a business method. It started as survival, and only later became a system I could teach.

One teacher changed the trajectory before any of that got resolved. In sixth grade, Miss England stayed after class on her own time and taught me fundamentals, letters, sounds, reading, one on one. She's the reason I believe the mentorship model behind Wake Up X actually works: one person, willing to give a little extra time, can change the entire shape of someone's future.

At thirteen, my family received a donated computer, an early-1990s Windows PC with no working mouse. I learned to run the whole thing from the keyboard alone. That, not a class or a certification, was the actual start of my technology career. From there, I started fixing computers and phones for anyone who'd let me, despite the language barrier.

Around that same time, I started volunteering with an organization called Computers for Kids. The advertised deal was simple: put in the hours, walk away with a free computer. I didn't care about the computer. I wanted to learn how the whole thing worked, so I stayed long after the incentive stopped mattering to me.

I started at the bottom of that, essentially help-desk work, sitting with people and walking them through the basics of a computer they'd never really used. My cousin was right there with me through a lot of it. But the longer I stayed, the more I noticed I wasn't just interested in the fix in front of me. I wanted to know how the whole operation ran underneath it, how people got signed up, how the workflow moved, what was slow, what nobody had gotten around to improving yet, not because I thought I knew better than anyone there, but because I couldn't turn that question off. Eventually that curiosity moved me from help desk into a supervisor role, running more of the administrative side.

I trace that instinct straight back to my parents. They came into this country as farm workers with no money, no housing lined up, and no English, and they still built a life and spent whatever they had left over helping everyone around them, family and strangers both, sometimes getting taken advantage of for exactly that generosity. Watching that removed any excuse I could ever make for myself. If they could do that with nothing, I had no right to sit on what little I did have. That's the real root of why I care more about lifting the people around me than about being the smartest person in a room, and years later, building Wake Up X felt less like starting something new and more like continuing something they'd already taught me.

To Computers for Kids: thank you for the door you opened for a kid who just wanted to understand how things worked. That opportunity is still paying forward today.

That repair work turned into a real business. I met my eventual co-founder through a chance referral, a stranger's cousin brought me a broken computer, and we built a cell phone repair company together: I ran the technical side, he ran sales. It worked because we were opposites, not mirrors, a lesson that shows up again in Part Nine of this book.

Along the way I failed more times than I can count, got taken advantage of at least once badly enough to push me toward governance and compliance work, and eventually built The Wake Up X Solutions out of everything that path taught me.

One book reframed all of it for me: The 10X Rule, by Grant Cardone. It took the isolation, the language barrier, and the being told I couldn't learn, and turned all of it from something I was hiding into fuel. That's really the point of telling you any of this: not the specific details of my story, but the proof that the beliefs and frameworks in this book were built under real pressure, not in a classroom.

Belonging nowhere, and choosing to uplift anyway

One job during that same stretch was at a Mexican restaurant. I'm Mexican American, first generation, and I figured that would count for something there. Instead, coworkers who were Mexican themselves made me feel like I didn't belong. I was told, more than once: you're not important enough, and even if you think you're better than us, you're never more and above than us. We're the real, traditional Mexicans, the hard workers. You're not.

I didn't have a comeback for that in the moment, and I didn't go looking for one. What I told myself instead was: maybe I don't have the same potential they think they have, but I have something different, dedication, and I'm willing to learn and extend my knowledge. I used it as motivation to move forward, and faster, not to prove anyone wrong to their face. I never let them see it land. Part of that was understanding, even then, that this kind of thing was going to keep happening in different rooms for the rest of my life. Part of it was already knowing it wasn't going to be the thing that decided my future. These obstacles are always going to happen. They don't define what comes after them.

Growing up, I never fully fit the "American" side or the "Mexican" side either. I've noticed the same pattern more than once since, in Latino and Mexican spaces I've worked in: a tendency to push each other down to look better by comparison, instead of pulling each other up. Sitting outside both identities gave me a strange kind of clarity about it. I decided a long time ago that if almost everyone around me was willing to belittle, and almost nobody was willing to uplift, that gap was exactly where I could make a difference.

Wake Up X, the mentoring, Computers for Kids before it, all of it comes out of that same decision: be the uplift I didn't get.

The library that rebuilt my thinking

- ▶ **Resilience and mindset:** Can't Hurt Me, You Are a Badass, Unfu*k Yourself, The 5 Second Rule, Ego Is the Enemy, Mind Hacking, On Fire, The Power of One More
- ▶ **Habits and mastery:** Atomic Habits, The Power of Habit, So Good They Can't Ignore You
- ▶ **Sales and motivation:** The 10X Rule, Sell or Be Sold, Be Obsessed Or Be Average, The Closer's Survival Guide, How to Get and Stay Motivated, all Grant Cardone
- ▶ **Marketing:** This Is Marketing, Magnetic Marketing, Traffic Secrets, Dotcom Secrets, Influence
- ▶ **Leadership:** Extreme Ownership, Leadership Strategy and Tactics, Dare to Lead, The Score Takes Care of Itself, The 7 Habits of Highly Effective People, Good to Great, Traction, The Advantage, A Higher Loyalty
- ▶ **Money and finance:** Rich Dad Poor Dad, The Psychology of Money, The Art of Spending Money, The Millionaire Next Door, The Millionaire Fastlane, Think and Grow Rich, The Total Money Makeover
- ▶ **Communication and negotiation:** Never Split the Difference, How to Win Friends and Influence People, How to Talk to Anyone
- ▶ **Strategy and influence:** The Art of War, 48 Laws of Power, Thinking Fast and Slow, Algorithms to Live By
- ▶ **The one technical crossover:** Social Engineering, because the human layer is the one most technical people ignore

Almost none of these are cybersecurity books. That's on purpose. The technical material is everywhere. This list is the part almost nobody in this field reads, and it's exactly the gap this blueprint exists to close.

I still run all of it at once, a full-time role as an IT operations and solution engineer, The Wake Up X Solutions, and a partnership with ISACA Sacramento, because the mission doesn't pause for a schedule. If a chapter like this one reads as tidy, understand it isn't. It's early mornings, recorded conversations edited between meetings, and treating the parts of the week that look like networking, mentoring, and community-building as real work, not a hobby squeezed in around a job.

If you're reading this, Miss England: thank you for not giving up when everyone else already had. Everything after this page exists because of that one decision.

If any part of my path sounds like yours, the rest of this book is what I built to make your version faster than mine was.

Reflect before you move on

- ▶ What's the one thing you'd need to believe about yourself to start, even without a clear path yet?
- ▶ Who is your version of Miss England, someone who gave you a chance when they didn't have to?

04

PART FOUR

LEVEL 1 — THE BASICS OF BREAKING IN

What "cybersecurity" actually means

If you're starting from zero, here's the confusion almost everyone runs into: tech, IT, and cybersecurity aren't one field. They're a cluster of related but distinct disciplines, and most job postings don't explain the difference.

- ▶ **IT** keeps systems running: help desk, support, networking, systems administration.
- ▶ **Cybersecurity** protects those systems, and splits further into offense (finding weaknesses before attackers do), defense (monitoring and responding to threats), and governance (making sure the organization follows the rules it's supposed to follow).
- ▶ **Cloud and DevOps** build and run the infrastructure everything else sits on.
- ▶ **GRC (Governance, Risk, and Compliance)** is the bridge between the technical work and the legal, regulatory, and business side of the organization.

Nobody hands you this map on day one. Most people, guests on the show included, spend their first year or two figuring it out by trial and error. You don't have to.

Two lanes people underestimate: Cloud and GRC

Most beginners default toward help desk or offensive security because those are the most visible entry points online. Two other lanes deserve just as much attention.

- ▶ **Cloud is where almost everything is moving.** AWS, Azure, and Google Cloud fundamentals are now relevant to nearly every IT and security role, not just dedicated cloud engineers. Even a basic understanding of how cloud environments are structured and secured makes you noticeably more employable.
- ▶ **GRC is where the business and technical worlds actually meet.** It's less flashy than offensive security, but it's one of the fastest-growing lanes, and it rewards exactly the communication and business skills this blueprint spends so much time on. If you're naturally more comfortable in conversations than in a terminal, don't overlook this lane just because it's quieter online.

TRY THIS ON FOR SIZE

A career changer with no technical background assumes cybersecurity only means hacking, and nearly gives up before starting because that path feels impossibly far away. A short conversation with a GRC professional reframes it entirely: her background in compliance and policy work from an unrelated industry was already ninety percent of the skill set GRC roles actually need. She was already qualified for a lane she didn't know existed.

Your first ninety days, concretely

- **Weeks 1-4:** Learn the landscape above. Pick one free resource (TryHackMe, a CompTIA study guide, a YouTube fundamentals series) and go through it fully before starting a second one.
- **Weeks 5-8:** Build a home lab. A single virtual machine you break and fix yourself teaches more than a month of passive video watching.
- **Weeks 9-12:** Start showing your work publicly, a LinkedIn post about what you learned, a small write-up, a project on GitHub. This is what starts conversations with people already in the field.



Your First Ninety Days

Entry points that actually work

- Help desk and technical support roles, still one of the most reliable ways into IT with no prior experience
- Internal transfers, moving from a non-technical role into IT within the same company, using the trust you've already built
- Volunteer or nonprofit IT work, real experience without needing a paid role first
- A home lab plus a documented project, proof of skill when you don't yet have a job title to point to

Build a real home lab

You don't need to buy hardware to start. The standard, current path is: install a free virtualization platform, run more than one operating system on the same machine, and break things on purpose so you learn how to fix them.

- **Virtualization platforms.** VirtualBox and VMware Workstation Player are both free and are the standard starting point for running virtual machines on your own computer.

- **Operating systems to practice on.** A Windows Server virtual machine to learn domains, Active Directory, and DNS, the systems that run inside almost every mid-size and large company. A Linux distribution like Ubuntu for general systems practice. Kali Linux specifically if your lane leans toward offensive security, since it comes preloaded with the standard tools.
- **Networking fundamentals to actually touch, not just read about.** Set up a small virtual network between two or three machines and practice how they discover and talk to each other. Understanding DNS, how a name like a website turns into an address a computer can use, is one of the single most valuable fundamentals in this entire field, technical or not.
- **Project management tools.** Even a free tool like Trello or a basic ticketing system teaches you to think in workflows, priorities, and status, the exact skill that separates someone who does tasks from someone who runs a project.

Document everything you do in this lab. A short write-up of what you built and what broke is worth more on a resume than a certification with no story behind it, because it proves you can actually do the thing.

Nine lab projects that actually go on a resume

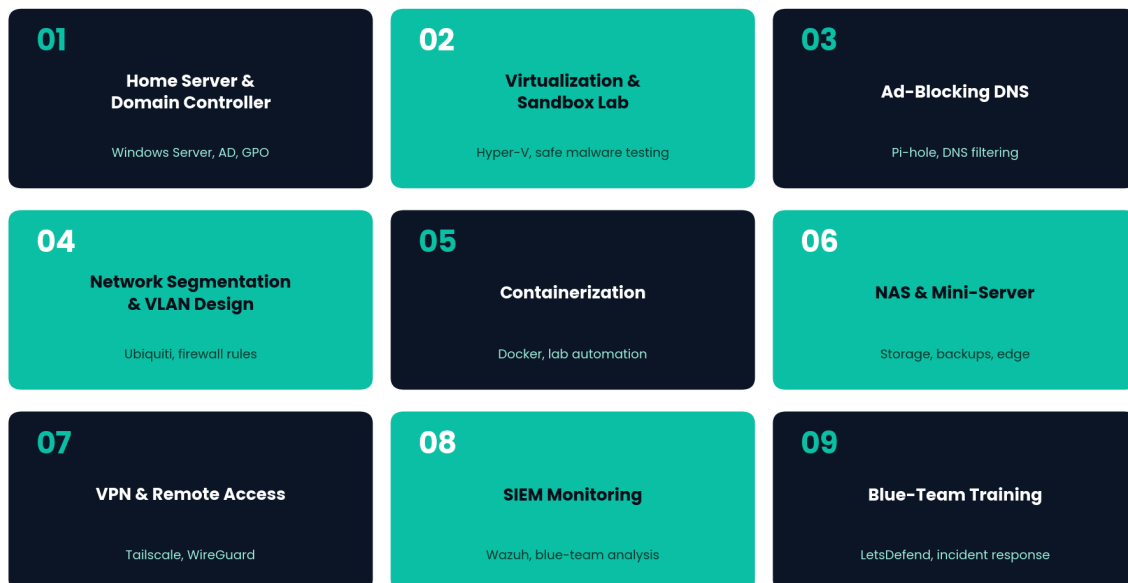
This isn't a hypothetical list. This is the exact set of builds in my own home lab, in rough order of difficulty. Each one maps to a real job skill and each one gives you a documented project to talk about in an interview.

- **1. Home Server & Domain Controller.** A full Windows Server lab with Active Directory and Group Policy, simulating enterprise user and device management, access control, and administrative delegation. This is the single most transferable lab on the list, almost every mid-size company runs on it.
- **2. Virtualization & Sandbox Lab.** A home VDI environment and sandbox using Hyper-V to test cybersecurity tools, malware behavior, and system configurations safely. Learn to break things without breaking anything real.
- **3. Ad-Blocking DNS (Pi-hole).** A network-wide DNS sinkhole blocking ads and malicious domains across the entire home network. Teaches DNS security and filtering in a way you can actually see working.
- **4. Network Segmentation & VLAN Design.** A VLAN-based segmented network using Ubiquiti routers and access points with custom firewall rules. This is real network architecture, not a diagram in a textbook.
- **5. Containerization with Docker.** Lightweight test environments for app deployment, network tool testing, and lab automation. Docker skill is now assumed in almost every infrastructure and DevOps conversation.

- **6. NAS & Mini-Server Builds.** A NAS plus a small mini-server (a ZimaBoard or similar) for centralized storage, backups, and edge-computing experiments. Teaches storage, redundancy, and backup strategy, the unglamorous skills that save companies.
- **7. VPN & Remote Access.** A Tailscale (WireGuard-based) VPN enabling secure remote access to servers, virtual labs, and endpoints. Remote access done correctly is a genuine security competency.
- **8. SIEM Monitoring (Wazuh).** A Wazuh SIEM deployed across endpoints to collect logs, monitor security events, and practice blue-team analysis. This is the closest a home lab gets to real SOC work.
- **9. Blue-Team Training (LetsDefend).** Hands-on incident-response simulations and network-traffic analysis to build genuine detection and SOC skills against realistic scenarios.

NINE LAB PROJECTS THAT GO ON A RESUME

Start with one. Document it. Let the list build itself over a year.



Nine Lab Projects That Go On a Resume

Nobody starts with all nine. Start with one, document it using the Portfolio Project Documentation Wheel from Part Eight, and let the list build itself over a year.

Automate your lab with n8n

Once you have a few of these running, add **n8n**, a free, self-hostable workflow automation tool. It's one of the highest-leverage additions to any home lab, because it turns a

collection of separate projects into a connected system.

- ▶ **Automate alerting.** Pipe SIEM alerts into a Slack, Discord, or email notification instead of checking a dashboard manually.
- ▶ **Automate reporting.** Schedule a weekly summary of what your lab detected, exactly like a real security team produces for leadership.
- ▶ **Automate the boring parts.** Backup checks, uptime monitoring, certificate expiration warnings, all the things that get forgotten until they break.
- ▶ **Practice integration thinking.** Wiring tools together is a genuine engineering skill, and it's the same instinct behind the AI-literacy section in Part Six.

Automation is also the clearest possible proof of the "be more valuable than your title" principle. Anyone can do a task. The person who automates the task and documents the time saved is speaking the language of the business.

Certifications: a path, not a collection

Certifications open doors, but only when they're earned with purpose, tied to a role you're actually aiming for, not collected because they exist. Employers can tell the difference between someone who studied for a real reason and someone chasing letters after their name.

The order that consistently works, based on both the podcast guests and the current state of the field:

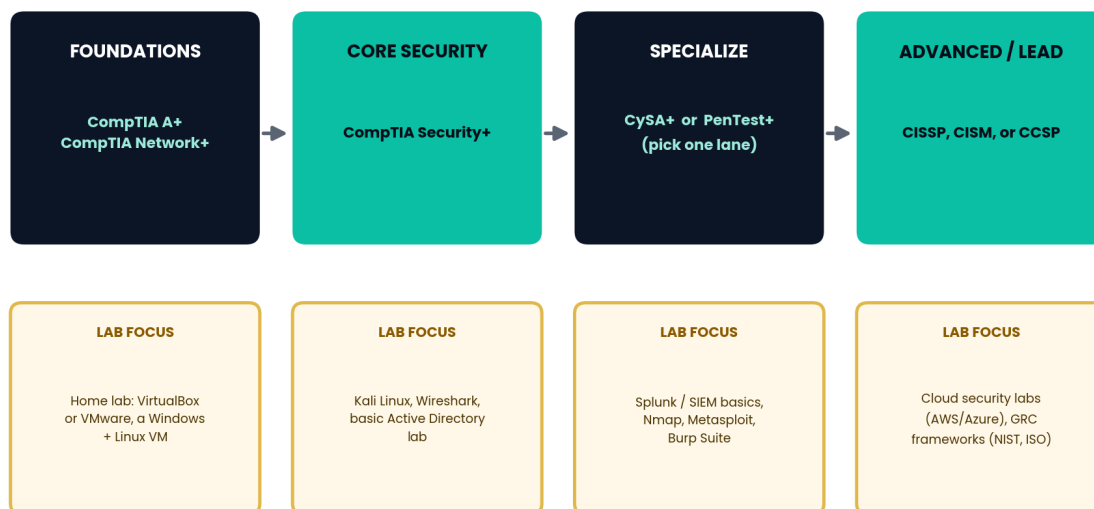
- ▶ **ISC2 CC (Certified in Cybersecurity)** if you want a free, genuinely entry-level starting point. It won't carry a resume by itself, but it costs nothing and proves you started.
- ▶ **CompTIA A+ and Network+** for general IT and networking fundamentals, if you're coming in with no technical background at all.
- ▶ **CompTIA Security+** as the real front door into cybersecurity. This is the most commonly requested entry-level cybersecurity certification in the industry right now, and it's the one that shows up in the most job postings.
- ▶ From there, branch based on your lane: **CySA+** if you're leaning defensive and analyst-track, **PenTest+** if you're leaning offensive.
- ▶ Further out: **CISSP, CISM, or CCSP** for leadership and governance-track roles.

One thing nobody tells beginners about the advanced certs: CISSP and CISM both require roughly five years of relevant paid work experience before you can be fully certified, not just before you should attempt them. You can sit the CISSP exam early and hold "Associate of ISC2" status until you've earned the experience, but you cannot shortcut the clock. Plan them as a year-five goal, not a year-one goal.

It's also worth being honest about a broken part of this industry: surveys have found employers requesting certifications that require five or more years of experience for roles advertised as entry-level. If you see that, it's usually a badly written job posting, not a real bar. Apply anyway.

CERTIFICATION & LAB ROADMAP

A path with purpose, not a collection — pick certs tied to the role you actually want



Certification & Lab Roadmap

It is not recommended to try to collect every certification available. A smaller set of certifications tied clearly to the role you want, backed by real lab work and documented projects, will always beat a long list that doesn't add up to a clear direction.

Creators and resources worth your time

You don't have to figure this out with silence in the background. These are the channels and shows I actually recommend, split by what they're good for.

For technical skill:

- **NetworkChuck** (youtube.com/@networkchuck) — the best beginner-friendly entry point for IT, networking, Linux, and cloud. If you're starting from zero, start here.
- **David Bombal** (youtube.com/@davidbombal) — deep networking and cybersecurity content, plus a lot of honest career-path interviews with people already in the field.
- **John Hammond** (youtube.com/@_johnhammond) — hands-on malware analysis, CTF walkthroughs, and offensive security. Go here once the fundamentals are in place.

- ▶ **Darknet Diaries** — real stories of hacks, breaches, and the people behind them. It won't teach you a skill directly, but it teaches you how this industry actually thinks, which is its own kind of fluency.
- ▶ **TryHackMe and LetsDefend** — guided, structured labs when you want more direction than building everything from scratch. LetsDefend in particular is built around blue-team incident response.

For leadership and business thinking:

- ▶ **Grant Cardone** (youtube.com/@grantcardone) — sales, obsession, and setting targets big enough to force action.
- ▶ **Tony Robbins** (youtube.com/@tonyrobbinslive) — psychology, state management, and decision-making under pressure.
- ▶ **School of Hard Knocks** (youtube.com/@theschoolofhardknocks) — street-level interviews about money, careers, and what people actually earn. Uncomfortable and useful.

In memory of Dr. Eric Cole

Dr. Eric Cole (youtube.com/@drericcole) passed away on May 19, 2026, at 56. He was a former CIA hacker, a Presidential Commissioner on Cybersecurity, a Wall Street Journal bestselling author, and, for a lot of us, the person who made cybersecurity leadership make sense.

His core message is the one this entire blueprint is built around: **security is a business problem, not just a technical one**. He spent thirty years teaching people to translate technical risk into language executives could act on, which is exactly the Communication and Business layers of the Career Value Stack.

His channel and his books, including *Cyber Crisis* and *Online Danger*, are still up and still worth your time. If you want to understand how to think like a security executive rather than just work like a security technician, start with him.

Reflect before you move on

- ▶ Which certification on the roadmap actually ties to a role you want, versus one you'd just be collecting?
- ▶ Have you actually built anything in a home lab yet, or only watched videos about it?

05

PART FIVE

LEVEL 2 — FIND YOUR LANE

The Wake Up X Curiosity Compass

Once you understand the landscape, the next question is which part of it actually fits how you think. This is the tool I built after watching too many talented people burn out in roles that didn't match their natural disposition. It gives you one primary lane, not a hedge, because clarity moves you forward faster than nuance does.

Fix. You want something broken to work again, fast. Help desk, support, systems administration.

Build. You'd rather construct something from nothing than inherit someone else's mess. Network architecture, infrastructure, engineering.

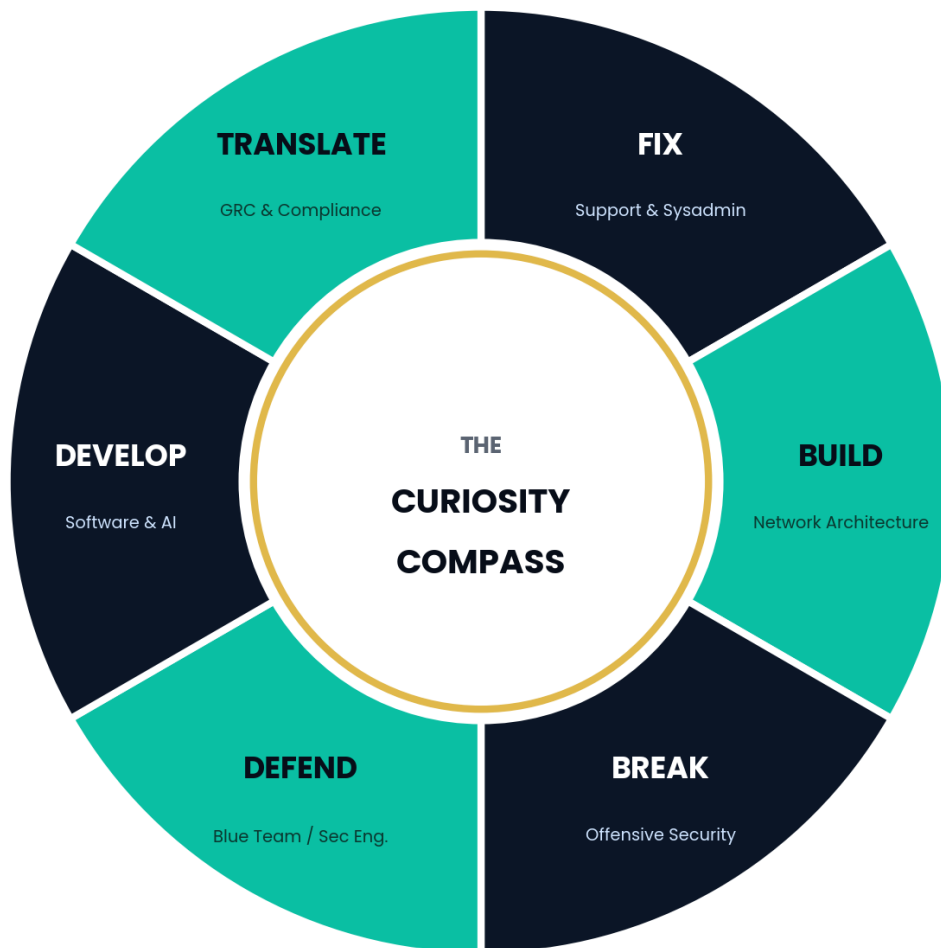
Break. You're driven to find the weakness before anyone else does. Offensive security and research.

Defend. You'd rather watch for the intrusion and catch it in the act. Security engineering, blue team.

Develop. You want to make the system do something new. Software, automation, applied AI.

Translate. You're the person who can sit with lawyers, executives, and engineers and make sure everyone understands the same risk. GRC and compliance specifically.

FIND YOUR LANE



The Wake Up X Curiosity Compass

Find your lane: a diagnostic that actually resolves

Answer these four questions honestly. Each answer points somewhere. Whichever lane shows up most is your starting lane.

1. Something just broke and everyone's watching. What's your instinct?

- Get it working again as fast as possible → **Fix**
- Figure out exactly why it broke before touching anything → **Defend**
- Ask how someone could make this break on purpose → **Break**

2. You've got a free weekend and no obligations. What are you building?

- ▶ A system, network, or setup that didn't exist before → **Build**
- ▶ A script or app that automates something annoying → **Develop**
- ▶ A cleaner process or document that gets everyone aligned → **Translate**

3. Which room would you rather be in?

- ▶ Hands on a keyboard, deep in a technical problem → **Fix, Build, Break, or Develop**
- ▶ In front of stakeholders, explaining risk in plain language → **Translate**
- ▶ Watching dashboards, hunting for something that's off → **Defend**

4. The tiebreaker. After a hard day, what do you want to be true?

- ▶ "I fixed it." → **Fix**
- ▶ "I built it." → **Build**
- ▶ "I found the hole before anyone else did." → **Break**
- ▶ "I caught the intrusion." → **Defend**
- ▶ "I made something new." → **Develop**
- ▶ "I made sure everyone understood the risk." → **Translate**

If questions 1 through 3 point in different directions, question 4 wins. It's the one people answer with their gut instead of their resume.

Your lane is a starting point, not a life sentence. Most careers move across two or three of these over time. Pick the one that fits you now, go deep enough to be genuinely good at it, and let the Pivot Panel below handle it if you picked wrong.

Real insight from the podcast, one guest per lane

- **Fix — James Landsaw, Help Desk to CISSP.** He started at the bottom of the ticket queue. Proof that entry-level support is a real professional foundation, not a waiting room.
- **Build — Devin Prasad, Network Architect.** Chose disciplined, fundamentals-first growth over the fastest paycheck.
- **Break — Aaron Portnoy, Security Researcher.** An offensive mindset built from adolescence into a full research career.
- **Defend — Cathy Stephenson, Deputy CISO track.** From state prison IT to the Deputy CISO track. Proof the door opens from places you'd never guess.
- **Develop — Jorge Avila, Founder, Paleo Tech.** Built a company from a non-traditional background using applied technical and AI skill.
- **Translate — Gustavo Mastroianni, CISO.** A governance- and relationship-driven path to executive security leadership, including the moments that almost ended his career.

Also on the show: **Greg McCord** (CISO and founder), **Sana Talwar** (AI security), **Anil Reddy** (founder, ARK Strategies), and **Larry Switzer** (Ziggiz), among others.

Every one of these is a full episode. This section is the summary. The podcast is where you hear exactly how they did it, in their own words, unscripted.

The Pivot Panel

Picked a lane and it's not clicking? Use this to course-correct instead of quitting the field entirely.

- **Fix feels too repetitive:** you may be a Build person who hasn't had the chance to design anything yet.
- **Build feels too slow:** you may be a Develop person who wants to see software ship faster than infrastructure allows.
- **Break feels isolating:** you may be a Defend person who wants the same instincts applied to protection.
- **Defend feels too reactive:** you may be a Break person better suited to hunting the problem than waiting for it.
- **Develop feels disconnected from people:** you may be a Translate person who needs the business-facing side of the work.
- **Translate feels too far from technical work:** you may be a Fix or Build person who needs more hands-on time first.

Connect With Every Guest

Every guest referenced in this library is a real person with a real story, worth following beyond these pages. Watch the full episode, connect on LinkedIn, say the podcast sent you.

James Landsaw

Help Desk to CISSP

[linkedin.com/in/jameslandsaw](https://www.linkedin.com/in/jameslandsaw)
youtu.be/u9ZqnE8G8fU

Gustavo Mastroianni

CISO, Schools Insurance Authority

[linkedin.com/in/gustavo-mastroianni](https://www.linkedin.com/in/gustavo-mastroianni)
youtu.be/RxQIV7VWdnw

Mohana Balakrishnan, CISSP

CTO, Schools Insurance Authority

[linkedin.com/in/mohana-b-24012016](https://www.linkedin.com/in/mohana-b-24012016)
youtu.be/MZJdZRsF4KY

Devin Prasad

Network Architect

[linkedin.com/in/devin-p-98791355](https://www.linkedin.com/in/devin-p-98791355) · devinprasad.net
youtu.be/AP0Uizvp9IU

Aaron Portnoy

Chief Product Officer, Mind Guard

[linkedin.com/in/aaronportnoy](https://www.linkedin.com/in/aaronportnoy)
youtu.be/OC8PqZXJHC4

Cathy Stephenson

Deputy CISO track

[linkedin.com/in/cathy-stephenson-6b54a026](https://www.linkedin.com/in/cathy-stephenson-6b54a026)
youtu.be/PU3tAlAKjGs

Jorge Avila

Founder, Paleo Tech Group

[linkedin.com/in/jorgeavilapmp](https://www.linkedin.com/in/jorgeavilapmp)
youtu.be/ycR7h1BR5v4

Greg McCord

CISO and Founder

[linkedin.com/in/gregmccord](https://www.linkedin.com/in/gregmccord)
youtu.be/IGDTBvpQ0-8

Sana Talwar

Product & AI Security

[linkedin.com/in/sana-talwar](https://www.linkedin.com/in/sana-talwar)
youtu.be/SnGIMUTG8v0

Anil Reddy

Founder, ARK Strategies

[linkedin.com/in/anil-reddy-kondakrindi-1933891](https://www.linkedin.com/in/anil-reddy-kondakrindi-1933891)
youtu.be/IOI5IK_UezA

Larry Switzer

Sales Leader, Ziggiz

[linkedin.com/in/larry-switzer-24a539](https://www.linkedin.com/in/larry-switzer-24a539)
youtu.be/f7b5JyffUHs

Louis Kemp

CTO, ANUL Technologies

[linkedin.com/in/louis-kemp-iv24](https://www.linkedin.com/in/louis-kemp-iv24) · anultechnologies.com
youtu.be/0KT3Vdgdcci

Kevin Bali

Guest Speaker

[linkedin.com/in/kevinbali](https://www.linkedin.com/in/kevinbali)
youtu.be/1axMImSsAWo

Zankar Vaishnav

Guest Speaker

[linkedin.com/in/zankarv](https://www.linkedin.com/in/zankarv)
youtu.be/1axMImSsAWo

Jesse Telles

Guest Speaker

[linkedin.com/in/jesse-telles](https://www.linkedin.com/in/jesse-telles)
youtu.be/1axMImSsAWo

Harmanjot Singh

Guest Speaker

[linkedin.com/in/harmanjotsingh7](https://www.linkedin.com/in/harmanjotsingh7)
youtu.be/1axMImSsAWo

Solomon Lee

Guest Speaker

[linkedin.com/in/solomon-lee-962304192](https://www.linkedin.com/in/solomon-lee-962304192)
youtu.be/1axMImSsAWo

Ryan Doctor

IT Service & Incident Management

[linkedin.com/in/ryandocor](https://www.linkedin.com/in/ryandocor)
youtu.be/tjC2-sdFnJw

John Cuellar

CEO & Co-Founder, Cascade Clarity

[linkedin.com/in/cuellar](https://www.linkedin.com/in/cuellar) · cascadeclearity.ai
youtu.be/-Tt2XKA1-jw

Michael Fierros

AspireByte

[linkedin.com/company/aspirebyte](https://www.linkedin.com/company/aspirebyte) · aspirebyte.carrrd.co
Episode not yet released

SHOUTOUTS

The Community Behind the Community

Not everyone who shaped this movement sat behind a microphone. These are people from the ISACA Sacramento community and beyond who showed up, organized, and made the rooms this library talks about actually happen.

- Jim L.
- Megan Rutlen
- Alvin Jue
- Eddie Lopez
- Cody Fultz
- Austin Schlocker
- Stacy Ditta
- Alexander Knapp
- Khadija Hersi
- Joe Santos
- Mario Ruiz
- Priscilla Valentin
- José Páez
- Moamen Ramadan
- Lainie van Doornewaard
- Bradley Ponce
- Bryan Curnutt
- Corinna Martinez
- Yashwini Kamdar
- Tim Swaney
- Ethan Ng

THANK YOU

If your name belongs here and isn't, or belongs in the Guest Directory instead, reach out at wakeupxpodcast.com — this list gets updated as the community grows.

ISACA Sacramento's Rising Cohort

DECODE calls out "The Next Generation, ISACA Sacramento" as proof there's no single right background. Here's who they actually are, in their own words.

Irene Shaju — MIS & Business Analytics, Sac State

International transfer student, graduating spring.

Got into cybersecurity because she loves coding — and because she sees AI as too big a shift to go unchecked. Her framing: cybersecurity is the discipline that keeps a rapidly advancing technology on a leash, while still being genuinely fun to work in.

Harmanjot Singh — Computer Science, Sac State

Schooled in India for 15 years, returned to the US in 2020.

Was torn between CS and MIS after his sophomore year; a mentor told him to stick with CS. The real turning point was discovering the CyberCorps Scholarship for Service — a federal program that pointed him straight at cybersecurity. Already Security+ and ISC2 certified.

Jared Holloway — Management Information Systems, Sac State

Senior, former foster youth.

Resilience isn't a footnote in his story, it's the center of it. His first internship, at Golden One, gave him the foot in the door he needed. Post-graduation, he's aiming to bridge cybersecurity and project management.

Still Figuring It Out, Out Loud

Zankar Vaishnav — Management Information Systems, Sac State

Transfer student, first year, joined ISACA his first semester.

Got pulled toward cybersecurity through a campus tech-support job, shadowing managers doing defensive work behind the scenes. Calls cybersecurity "a vast ocean" — the scale of it is what drew him in, not what scared him off.

Kevin Bali — Business Administration, Sac State

President, ISACA student chapter and MIS Association.

Deliberately combining MIS, business analytics, HR, AI management, and a criminal justice minor toward one goal: digital forensics work that makes communities safer. His sharpest line: "if it's not the people you work with that you enjoy, then it's for nothing."

Solomon Lee — Computer Science, Sac State

From Chicago, graduating this December.

Pivoted from software engineering into cyber after winning a scholarship. Pushes back on generic "don't follow your passion" advice — his point is you won't sustain the grind on a job you don't actually want. Currently building a SIEM lab on Azure from zero prior cloud experience.

Jesse Telles — Software Engineering Cybersecurity

Drawn from architecture and systems design into security.

What pulled him toward cyber was fascination with how components interact — which led straight to supply-chain security: how automating system-building creates vulnerabilities with effects that cascade across every connected system.

Reflect before you move on

- ▶ Which lane did the Curiosity Compass point you toward, and does it match what you expected?
- ▶ If you picked wrong, which pivot fits you better?

06

PART SIX

LEVEL 3 — CROSS-TRAINING

Specialists get hired. Cross-trained people get promoted.

Whatever lane you picked, don't stop there. Being a specialist gets you a job. Cross-training gets you the better offers, the promotions, and the mobility to move between roles when the market shifts. It's also the single best protection against being replaced, someone with real breadth across five areas is far harder to lay off than someone who only does one narrow thing well.

- ▶ **Networking fundamentals.** Understanding how data moves makes you more useful in every adjacent role, even if you never take a networking job.
- ▶ **Hardware fundamentals.** Knowing what's physically happening under the software builds a troubleshooting instinct pure software people often lack.
- ▶ **Programming fundamentals.** You don't need to be a developer. You need enough to automate your own work and understand what you're asking a developer to build.
- ▶ **Risk, vulnerability, and compliance basics.** Even a Build or Develop person becomes far more valuable once they understand how their work gets audited and regulated.
- ▶ **AI literacy.** Learn to use it, verify its output, and never paste anything sensitive into it. Leverage, not a crutch.

AI literacy, in practice

Every level in this field now touches AI in some form, so treat it as its own cross-training area, not an afterthought.

- ▶ **Use it to accelerate study, not replace it.** Ask it to quiz you on Security+ objectives, explain a concept three different ways, or generate practice scenarios. Never let it be the only source you learn a concept from.
- ▶ **Use it to draft, then verify.** A first-pass resume bullet, a first-pass incident report, a first-pass script, all reasonable. Submitting any of it without checking the details yourself is not.
- ▶ **Never paste sensitive data into a public tool.** Client information, credentials, proprietary code, and real vulnerability details don't belong in a prompt box you don't control.
- ▶ **Learn basic prompt structure.** Being specific about role, context, format, and constraints gets dramatically better output than a vague one-line ask, and this skill itself is becoming a hiring differentiator.
- ▶ **Use it as a translator.** Ask it to turn a technical finding into a one-paragraph, non-technical summary for a stakeholder. This is one of the fastest ways to practice the Communication layer of the Career Value Stack.

TRY THIS ON FOR SIZE

Two candidates apply for the same mid-level security analyst role. One only knows security tools. The other can read a network diagram, explain a vulnerability to a non-technical executive, and write a basic script to automate a report. The hiring manager picks the second one every time, not because the first candidate lacks skill, but because the second one already speaks three departments' language.

Reflect before you move on

- ▶ Outside your main lane, which of the five cross-training areas are you weakest in right now?
- ▶ What's one small step you could take on that area this month?

07

PART SEVEN

LAND THE ROLE

Applying, interviewing, and following up like it matters

Skills get you qualified. This part of the process is what actually gets you hired, and it's the part almost nobody teaches.

The T.R.A.C.K. Application System

Most applications fail before the resume is even read, because the system behind them was weak. Run every serious application through five steps, in this order:

- **Tailor.** Match the resume and cover message to this exact role, not a generic version sent everywhere. Mirror the language in the job posting itself.
- **Research.** Know the team, the product, and something recent about the company before you go any further. This is what makes step three possible.
- **Approach.** Message someone actually on the team, not just a recruiter, with a short and specific note. Do this **before** you apply, so your name is already familiar when the application lands. Even a brief reply is worth more than a cold submission.
- **Contribute the application.** Submit through the official channel, and reference the conversation from step three if you had one. A referral, when you can get one, moves you past the resume pile entirely.
- **Keep following up.** One follow-up after the interview within twenty-four hours, one more if you hear nothing by the timeline they gave you. Then let it go and put the energy into the next one.

THE T.R.A.C.K. APPLICATION SYSTEM



The T.R.A.C.K. Application System

Most people only do the "submit" step, then wonder why a strong resume gets no response. The other four are what separate an application from a candidacy. The order matters: research feeds the approach, and the approach is what makes the application

land warm instead of cold.

Telling your story with the STAR method. Whether it's an interview question or a LinkedIn post, structure your examples the same way every time: **Situation** (what was going on), **Task** (what you were responsible for), **Action** (what you specifically did), **Result** (what changed because of it, ideally with a number attached). Vague answers lose to specific ones every time, even when the underlying work was similar.

The follow-up nobody does. After an interview, send a short, specific thank-you within twenty-four hours, referencing one real detail from the conversation, not a generic template. If you haven't heard back in the timeframe they gave you, one polite follow-up email is expected, not pushy. Silence doesn't read as respectful, it usually just reads as forgettable.

The V.A.L.U.E. conversation, for negotiating and for selling yourself. This applies whether you're negotiating an offer, pitching a client, or making the case for a promotion.

- ▶ **Validate** the other person's position or need first, before making your own case.
- ▶ **Ask** questions before you pitch anything, so you actually know what they care about.
- ▶ **Listen** for what they say matters most, then speak directly to that.
- ▶ **Underscore** your value with specifics, not adjectives. A number beats an opinion.
- ▶ **Engage** them in the next step directly, don't leave the next move ambiguous.

What hiring managers are actually looking for

Employers consistently say the same thing in different words: they can teach a specific tool, but they can't easily teach curiosity, communication, or reliability.

- ▶ **What actually stands out:** documenting your own projects, explaining your thinking clearly, and showing evidence of solving a real problem, even a small one, beats one more certification on a resume.
- ▶ **If you're on the hiring side yourself one day:** vet technical claims directly instead of trusting a recommendation alone.
- ▶ **Set clear expectations and terms in writing** before day one, every time.
- ▶ **Hire for complementary skills on a team,** not five people who are all good at the exact same thing.

Everything is negotiable

Almost nobody treats negotiation as a life skill. Most people file it under "salary conversation" or "car dealership" and leave it there. In reality it shows up everywhere: in relationships, in business, in real estate, in IT and cybersecurity contracts, even in how you navigate faith and family expectations. The only real exception is a shelf price at a retail store. Almost every other interaction has more room in it than people assume, and the people who do best in business, and in life, are usually the ones who never stopped testing where that room was.

This is one of the biggest key topics in this whole book, not because negotiation is complicated, but because almost nobody is taught to see it everywhere it actually applies.

If you want to go deeper than this book can cover, *Never Split the Difference* by Chris Voss and *Getting to Yes* by Fisher and Ury are two of the most respected, widely recommended starting points, one from a former FBI hostage negotiator, one from the Harvard Negotiation Project. Both show up constantly on "best negotiation books" lists for a reason.

Reflect before you move on

- ▶ Does your resume tell one clear story with the STAR method, or is it still a list of duties?
- ▶ When did you last follow up after an interview or application, and did you actually do it?

08

PART EIGHT

LEVEL 4 — THE MULTIPLIER SKILLS

What makes a specialist irreplaceable

Cross-training builds your foundation. This is what makes you genuinely rare on top of it: the ability to sell, communicate, and translate technical work into business language.

A Fix person becomes rare once they can also sell and communicate clearly. A Build person becomes rare once they can translate their architecture to a non-technical executive. A Develop person becomes rare once they can explain, in dollars and hours, why their automation matters.

This is the layer that decides whether you stay a good technical hire your whole career, or get pulled into the room where decisions get made.

Be more valuable than your title

Your job title is the floor, not the ceiling. I currently work as a desktop engineer, but that title doesn't cover project management, solution engineering, cross-team communication, executive meetings, or leadership work, and I do all of it anyway, because I took the initiative to learn alongside the network team, the cybersecurity team, the infrastructure team, and the applications team instead of staying inside my own lane. A specialist in a single narrow position is the easiest person to replace. Someone who's built real foundations across many areas, even without formal authority over any of them, is much harder to let go.

Case study: the communication hub nobody else built. I work inside a large enterprise spread across multiple cities and states, where sites had no way to communicate with each other, everything ran through disconnected phone calls with no shared structure. Nobody else had bridged that gap. So I built the bridge myself: one connection point per site, welcoming every location to join, until it grew into a real, standing communication network across the whole organization. Nobody assigned me that project. I saw a structural problem that mattered to the business and solved it without being asked, which is exactly the kind of initiative that gets remembered at review time.

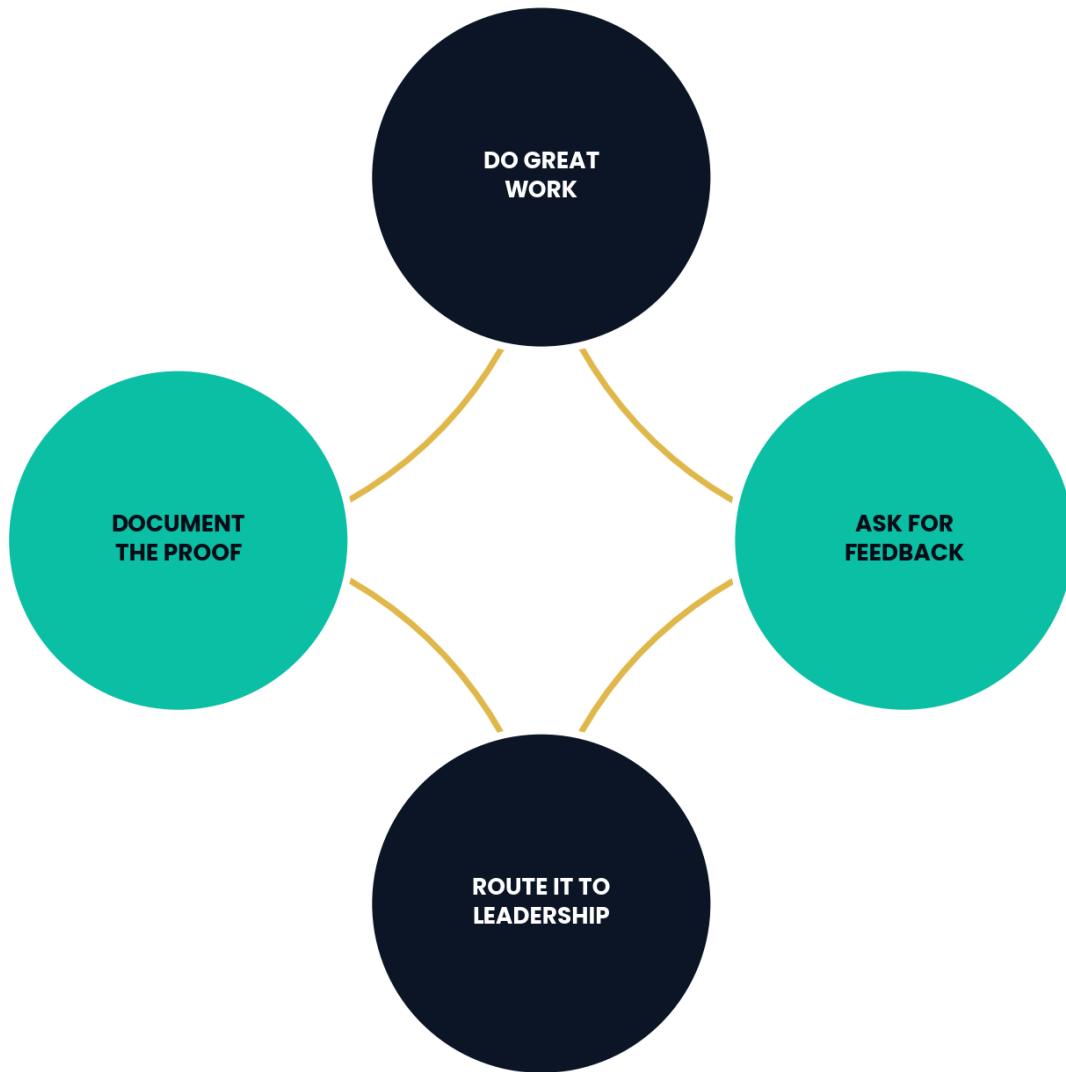
Case study: building structure where there was none. At another point, I walked into a site with no real organization, just phone calls with no system behind them. There was a ticketing tool, but it wasn't being used as one. I introduced staff members to each other, ran real meetings, and worked directly with CEOs and stakeholders to build an actual structured process out of the chaos. That's the same instinct as the communication hub: find the gap nobody's addressing, and become the person who closes it.

The reputation loop: make your value visible

Doing great work quietly is not a strategy. Treat your own career the way a business treats its five-star reviews.

- ▶ **Ask for feedback directly**, right after you've done something well, while it's still fresh for the other person.
- ▶ **Route that feedback to the people who matter**, your supervisor, your manager, leadership, not just to yourself. Genuine praise from someone you helped, sent directly to your chain of command, carries far more weight than you describing your own work.
- ▶ **Document every success as you go**: the project, the approval, the email from leadership confirming it. Anyone can claim they did good work. Very few people can actually prove it with a paper trail. That proof is what separates you in a review, a promotion case, or a future interview.
- ▶ Think of it as a two-way exchange: help someone genuinely, and they often become a future reference for you, the same way you may become one for them.

THE REPUTATION LOOP



Make your value impossible to overlook

The Reputation Loop

The Portfolio Project Documentation Wheel

The reputation loop tells you to document your work. Here's exactly what a documented project should capture, every time, so it's ready to become a resume line, an interview story, or a case for a raise without any scrambling later.

- ▶ **The problem.** What was broken, missing, or inefficient, in plain language.
- ▶ **The constraint.** What made it hard: time, budget, politics, missing information.

- ▶ **The action.** What you specifically did, step by step, in your own words.
- ▶ **The result.** What changed, ideally with a number: time saved, incidents reduced, dollars protected.
- ▶ **The proof.** The email, the approval, the screenshot, the metric, whatever makes it verifiable.

Five minutes after finishing any real project, fill in those five lines somewhere you'll actually find again. A year from now, "I think I fixed something like that once" turns into a fully-formed STAR story in under a minute.

Try this

Pick one thing you built or fixed recently. Explain it in two sentences to someone with zero technical background, focused entirely on what it means for them: time saved, risk reduced, money protected. If you can't yet, that's your next skill to build, not a failure.

Reflect before you move on

- ▶ Are you more valuable than your current title suggests, and can you prove it on paper?
- ▶ Who have you helped recently that you could ask for feedback, routed to the right person?

09

PART NINE

LEVEL 5 — THINK LIKE AN OWNER

Business sense every technical person needs, even if you never start a company

You don't need to run a business to think like the person who does. Every one of these lessons came out of real business ownership, and every one applies whether you're an employee, a contractor, or a founder.

► **Partner with your opposite, not your mirror.** If someone would just duplicate a skill set you already have, hire them, don't make them a co-founder. Complementary skills are what make a partnership work. This is the exact lesson from the cell phone repair business I described in Part Three.

► **Interview a business partner the way you'd vet a hire.** I've had bad business partners for the same reason I had a bad hire: trusting a referral or a good feeling instead of actually checking. Now, before I partner with anyone, I sit down and ask directly what their goals are, what they think the mission is, and where they see this going. If you wouldn't hand someone a role on your team without testing their claims first, don't hand them half your business on a handshake either.

► **Customer service and reliability are a real competitive advantage.** Anyone can learn the technical skill. Fewer people treat every client or coworker like a person instead of a transaction. Even in a support or help-desk role, exceptional service is one of the fastest, most repeatable ways to stand out, and it costs nothing but attention.

► **Almost everything is negotiable, except retail.** If you're talking directly to a business owner or stakeholder, there's usually room to negotiate. A shelf price at a store is the one exception. There's more on this in the negotiation section ahead.

► **Get it in writing, every time.** I once turned down a one-million-dollar offer. The number alone should have made it the easiest yes of my life. But the other side, including their own lawyer, wouldn't put a single thing in a signed contract. They wanted a verbal agreement, funded with my own money going in first. Walking away from that felt reckless in the moment, not smart, that kind of money doesn't come around twice, and everything in me wanted to find a way to make the verbal version work. I said no anyway, because a verbal agreement protects no one, no matter how friendly the room feels while you're in it. It stung for a long time afterward, wondering what that deal could have become. It only turned into a lesson later, once I saw how many people I know who did sign something that wasn't real, and lost far more than a missed opportunity. Whether the deal in front of you is worth a hundred dollars or a million, the rule doesn't change.

- ▶ **Vet skill claims yourself.** My first hire came through a good friend's recommendation, and he sounded sharp and technical in the interview, enough that I didn't push much further than that. What the business actually needed at that point wasn't more technical depth, it was someone who could sit across from a customer and represent us well. I found that out the hard way, after the fact, because I hadn't set clear expectations or payment terms in writing going in either. That mismatch cost real time and real money before I admitted what had gone wrong. A friend's word is not a substitute for testing the claim yourself, no matter how much you trust the person making the introduction.
- ▶ **Separate personal and business finances from day one,** even for a side hustle.
- ▶ **Never go into business with family.** Use a favor-based arrangement instead if you want to help a family member professionally. Money and blood mixed badly enough times that this is close to a hard rule.
- ▶ **Build a real team as early as you can.** A solo operator hits a ceiling fast. Everything gets easier once there's a team, even a small one, that you trust.
- ▶ **Failure is not the opposite of success. It's the tuition.** I've failed in business more times than I can count, more than a hundred, maybe more than five hundred. What matters is what you extract from it.

Business and money books worth your time

- ▶ **Rich Dad Poor Dad, The Millionaire Next Door, The Psychology of Money**, for how to actually think about money, not just make it
- ▶ **Traction** and **The Advantage**, for running a team or a company with real structure
- ▶ **Good to Great**, for the difference between a good business and a lasting one
- ▶ **The Total Money Makeover**, for getting personal finances in order before scaling anything

Employed or independent? Run the real numbers

At some point almost everyone in this field wonders whether to go independent. Here's the honest comparison, with real data instead of hype.

What the employed path pays. US Bureau of Labor Statistics puts the median for information security analysts at **\$124,910** (2024). Predictable, with benefits, retirement matching, and paid leave attached.

What the independent path pays. This is where the numbers get noisy, so treat any single figure with suspicion:

- ▶ Cybersecurity consultant averages differ hugely by source: roughly **\$112,000** (Indeed), **\$132,000** (ZipRecruiter), **\$157,000** (Glassdoor). Same job title, three very different pictures.
- ▶ Freelance cybersecurity specialists commonly bill **\$90 to \$170 per hour**, with senior specialists going higher.
- ▶ Independent contractor cybersecurity roles average around **\$133,000**, with the middle of the market between roughly \$111,000 and \$150,000.

The trap nobody warns you about. Take an hourly rate, multiply by 2,080 hours, and you get a number that feels life-changing. It's fiction. Industry benchmark data consistently shows independent consultants actually bill somewhere between **900 and 1,200 hours a year**, roughly 150 to 220 billable days. Use **1,000 hours** as a conservative planning figure.

The missing hours don't vanish, they go into writing proposals that get no reply, chasing invoices, doing your own admin and taxes, business development, professional development, and the dead gaps between contracts.

The honest formula. Before you quit anything, calculate:

- ▶ **Your target take-home income**, plus

- ▶ **The cost of replacing what employment gave you:** health insurance, retirement contributions, paid time off, and
- ▶ **Real business overhead:** liability insurance, accounting, software, legal, and
- ▶ **Self-employment taxes,** then
- ▶ **Divide by 1,000 billable hours,** not 2,080.

That result is your floor rate. Not your asking price, your floor. Below it, you are subsidizing your client out of your own pocket.

TRY THIS ON FOR SIZE

Someone earning \$125,000 as an employee sees consultants billing \$150 an hour and does the math: $\$150 \times 2,080 = \$312,000$. They quit. In year one they bill 850 hours, because half their time went to finding work. That's \$127,500 gross, minus roughly \$12,000 in health insurance, minus self-employment tax, minus software and insurance and accounting. They cleared less than they made as an employee, with none of the security. The rate was never the problem. The billable-hours assumption was.

What actually makes the independent path work. The same things this whole blueprint has been building toward: a pipeline that comes from the Connection Flywheel, proof from the Portfolio Documentation Wheel, and the ability to sell and translate value from the Multiplier Skills. Going independent doesn't replace those skills. It's the exam for them.

10

PART TEN

STAYING IN THE GAME

What getting fired taught me about rights, HR, and never being a victim of it again

Before any of this became a business, it was a string of odd jobs, and not a graceful one. I got fired more than once. I got laid off. I quit jobs where I was doing real work and getting overworked for it with nothing back, at IT jobs and in restaurants both, places that had no real interest in what I actually brought to them. I took contract work and gig work on purpose instead of chasing something full-time, because I needed to keep building experience even while everything else around me stayed unstable.

There was one firing in particular that changed the direction of everything. I told myself, right then, that I would never again be dependent on someone else's decision to hand me an opportunity. I wasn't going to be a victim of it. I was going to figure it out myself and build my own way instead of waiting for someone to give me one. That decision is the actual starting point of The Wake Up X Solutions, more than any business plan was.

It's also where my understanding of employee rights, HR, and compliance actually comes from. Not a course. Getting overworked and undervalued enough times that I went and learned exactly what I was and wasn't owed, and what a company can and can't do. Everything in this section comes out of that.

Growth, promotions, and layoffs

Breaking in is the beginning, not the finish line. Tech and cybersecurity have both been through real layoff waves, so staying valuable once you're in matters just as much as getting in.

- ▶ **Performance reviews aren't a formality.** Walk in with your own documentation ready: projects completed, feedback received, measurable results. Don't rely on your manager to remember everything you did all year.
- ▶ **Asking for a raise or a promotion is a business case, not a favor.** Bring the specific value you've added, ideally in dollars, time saved, or risk reduced, and connect it directly to what the next level of responsibility requires.
- ▶ **If a layoff happens, breadth is what protects you next time.** The multiplier skills and cross-training in this blueprint aren't just for getting hired, they're what makes you the harder person to cut when a company reduces headcount, because you're never doing just one replaceable thing.
- ▶ **Keep your documentation and network current even when things feel stable.** The worst time to start building your reputation and your contact list is after you've already lost the role. The best time was the whole time you had it.

Build a Personal Board of Directors

No real executive makes every decision alone. Build your own small, informal board, three to five people who fill different roles:

- ▶ **A technical mentor**, further along in your specific lane, who can sanity-check skill decisions.
- ▶ **A business mentor**, who understands career and compensation strategy beyond just the technical path.
- ▶ **A peer**, at roughly your level, who you trade honest feedback with regularly.
- ▶ **Someone outside the field entirely**, who keeps you grounded and reminds you why any of this matters beyond the job.

You don't need permission to form this board. Most people never ask anyone to formally join, they just consistently go to the same few people, and the relationship becomes real through repetition.

TRY THIS ON FOR SIZE

A round of layoffs hits a mid-size company. Two employees with similar tenure are affected. One only ever did the specific tasks in their job description, quietly and competently, for four years. The other spent that same time cross-training into adjacent skills, documenting their wins, and staying visible in a local professional chapter. The first person spends five months searching cold. The second has three warm conversations lined up within a week, because the network and the documented track record were both already built before they needed either one.

Reflect before you move on

- ▶ If you lost your role tomorrow, how many warm conversations could you start within a week?
- ▶ Who's currently on your personal board of directors, even informally?

11

PART ELEVEN

LEVEL 6 — THINK LIKE AN EXECUTIVE

Leadership starts before the title

Ownership, accountability, and sound judgment are behaviors you can practice at your current level, regardless of title. Title tends to follow the behavior, not the other way around.

- ▶ Ownership over your work and its outcomes, not just your assigned tasks
- ▶ Accountability when something goes wrong, before anyone asks
- ▶ Emotional intelligence in how you deliver hard news
- ▶ Coaching and mentoring the person one step behind you
- ▶ Delegation, trusting others with real responsibility instead of holding onto everything
- ▶ Strategic thinking, connecting today's task to the larger goal it serves

“Never let your current title determine your future potential.”

EFREN L. SALAZAR

Leadership books worth building a shelf around

- ▶ **Extreme Ownership**, Jocko Willink and Leif Babin, on accountability that starts with you, always
- ▶ **Dare to Lead**, Brené Brown, on the emotional intelligence side of leading people
- ▶ **The 7 Habits of Highly Effective People**, Stephen Covey, still one of the most complete frameworks for personal and professional judgment
- ▶ **The Score Takes Care of Itself**, Bill Walsh, on standards over shortcuts

The Executive Dashboard

Executives track a handful of numbers that tell them, at a glance, whether things are actually healthy. Build your own personal version:

- ▶ **What moved this month** that you can point to directly
- ▶ **What's at risk** if nothing changes in the next quarter
- ▶ **Who you helped**, and whether they know it was you
- ▶ **What you learned** that you didn't know ninety days ago

Four honest answers to those four questions, checked monthly, will tell you more about your real trajectory than a job title ever will.

The Wake Up X Decision Filter

Executives aren't distinguished by having all the answers. They're distinguished by having a repeatable way to decide under pressure with incomplete information. Run any significant decision through these five questions before you commit:

- ▶ **Impact.** What actually changes if I make this decision, and for whom?
- ▶ **Reversibility.** Can this be undone if it's wrong, or is this a one-way door?
- ▶ **Data.** Am I deciding on real information, or on assumption and urgency?
- ▶ **Stakeholders.** Who else is affected, and have I actually asked them, not just informed them?
- ▶ **Timing.** Does this decision need to happen now, or does urgency just feel that way?

THE WAKE UP X DECISION FILTER

Run any significant decision through all five gates before you commit



One-way door + real impact + thin data = slow down. Reversible + low impact = move fast.

The Wake Up X Decision Filter

A one-way door decision with real impact and thin data deserves more time than pressure will make you feel like you have. A reversible, low-impact decision deserves speed, not a committee.

TRY THIS ON FOR SIZE

You're offered a partnership on a project that could double your income, but the other side won't put anything in writing and wants to move on a handshake by end of week. Run it through the filter: high impact, low reversibility once your own money or reputation is in, thin data on their intentions, real stakeholders (your own business or family) who haven't weighed in, and urgency that's coming from them, not from the facts. That's not a hard decision once you slow down enough to actually look at it.

Reflect before you move on

- ▶ Think of a recent decision you rushed. Which of the five filter questions would have slowed you down in the right way?
- ▶ Where in your current role are you already leading, regardless of title?

12

PART TWELVE

THE QUARTERLY REVIEW

Give your strategy a real deadline

Businesses review performance every quarter. Most people never do that for their own career, and then wonder why nothing's moved in two years.

Set a real timeframe, three to six months, for any goal: a job search, a certification, a promotion, a business idea. If there's no measurable movement in that window, that's your signal to reassess the strategy, not to grind harder at the same approach.

Ask yourself every quarter:

- ▶ What actually moved in the last three months, and what didn't?
- ▶ Is the lack of movement a sign I need to work harder, or a sign my approach is wrong?
- ▶ What would I do differently if I were starting this goal today, knowing what I know now?
- ▶ Am I treating my finances, education, and soft skills with the same seriousness a business treats its own numbers?

Your Career Health Score

Rate yourself honestly, one to five, on each of the five areas below, every quarter. Track the trend, not just the snapshot.

- ▶ **Skill growth.** Did you learn something new and provable this quarter?
- ▶ **Network strength.** Did you add or deepen real relationships this quarter?
- ▶ **Documentation.** Is your portfolio and resume current with your most recent wins?
- ▶ **Financial position.** Are you saving, or at least aware of where the money goes?
- ▶ **Energy and sustainability.** Are you building this at a pace you can actually keep up for years?

A low score in one area for one quarter isn't a crisis. The same low score three quarters running is the signal this whole framework exists to catch.

Reflect before you move on

- ▶ What's your honest Career Health Score right now, area by area?
- ▶ What's the one number you'd be most embarrassed to say out loud, and why?

13

PART THIRTEEN

REAL VOICES, REAL ROOMS

Why this isn't theory

Every framework in this blueprint exists because I tested it against real conversations with real people who actually did each part of this path. The Curiosity Compass alone represents six full episodes' worth of career histories. This blueprint is the summary. The podcast is where you hear the whole story, in their own words, including the parts that didn't go as planned.

If you want to hear how a help desk technician actually became a CISSP, how a security researcher's curiosity started well before it had a paycheck attached, or how a governance-minded professional worked their way to CISO, those conversations are already recorded and waiting for you. New episodes go up regularly, and every one of them gets run through the same lens as this book: what actually worked, not what sounds good in theory.

Build your network in person, not just online

Everything in this blueprint gets easier once you're in a room with real people doing the work. Local chapters are one of the most underused resources in this entire field, and most of them welcome newcomers on purpose.

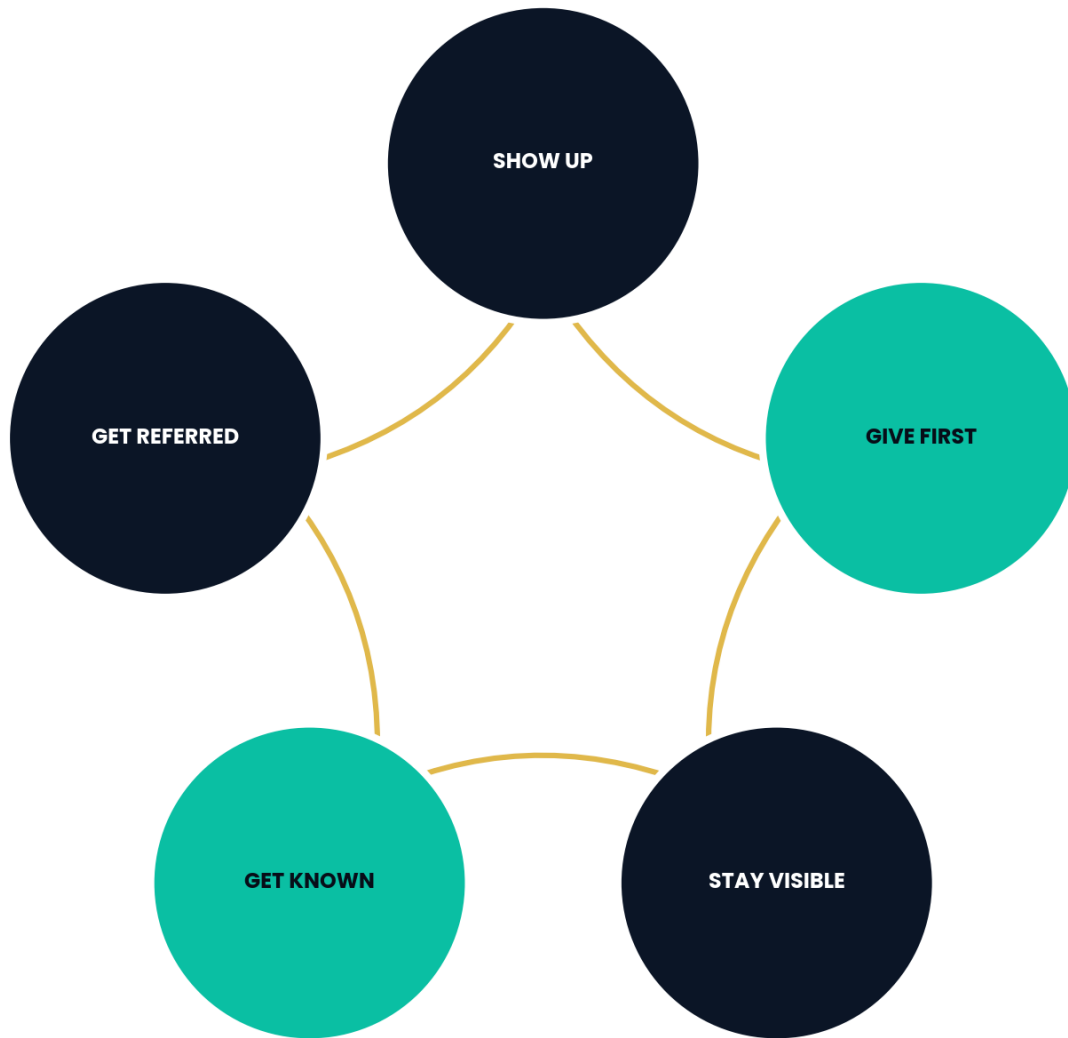
- ▶ **ISACA and ISC2 local chapters** for governance, risk, compliance, and general security community, in nearly every major metro area.
- ▶ **PMI (Project Management Institute) chapters**, valuable even for technical people, since project management is one of the multiplier skills from Part Eight.
- ▶ **Larger national events**, RSA Conference and DEF CON among them, where the density of people and conversations in a few days can outpace months of networking online.

The Connection Flywheel

Networking that works isn't a series of one-off asks. It's a loop that compounds:

- ▶ **Show up.** Same chapter, same event, more than once. Consistency is what makes you recognizable.
- ▶ **Give first.** Answer a question, make an introduction, share a resource, before you need anything back.
- ▶ **Stay visible.** Post what you're learning. Comment genuinely on what others share.
- ▶ **Get known.** After enough turns of the loop, people start associating your name with your lane.
- ▶ **Get referred.** Referrals are the natural output of being known, not a separate thing you have to chase.

THE CONNECTION FLYWHEEL



Momentum compounds — each turn makes the next one easier

The Connection Flywheel

I partner with ISACA Sacramento myself, and it's one of the clearest examples I have of this flywheel in action: showing up consistently to a local chapter turned into real relationships, mentorships, and opportunities that no cold application ever could have produced. Find the chapter or event nearest you, show up more than once, and let the relationships compound.

Reflect before you move on

- ▶ Which guest story from this blueprint felt closest to your own situation, and why?
- ▶ What's one local chapter or event near you that you've been meaning to actually attend?

14

PART FOURTEEN

TAKE YOUR NEXT STEP

You have the mindset, the map, and the moves

Curiosity gets you in the door. Cross-training and multiplier skills make you valuable. Business understanding creates impact. Leadership opens opportunities. None of it requires a traditional path, a degree, or connections you don't already have.

Stay connected to the rest of this movement:

- ▶ Subscribe to Wake Up X on YouTube: youtube.com/@Efren.wakeupx
- ▶ Follow on LinkedIn: linkedin.com/in/efrenlsalazar
- ▶ Listen on Spotify and Apple Podcasts
- ▶ Explore the full library at wakeupxpodcast.com
- ▶ Have a story that belongs on the show? wakeupxpodcast.com/guest

I'd genuinely like to hear your story too. Add me on LinkedIn and let's talk, this blueprint only works if it turns into a conversation, not a one-way download.

"I finally know what to do. I have a plan. And I'm not doing this alone."

WHAT EVERY READER SHOULD FEEL, WELCOME TO THE MOVEMENT

GLOSSARY

Plain-language definitions for every term in this blueprint

- ▶ **A+, Network+, Security+** — CompTIA's entry-level certifications, in that order, for general IT, networking, and cybersecurity fundamentals.
- ▶ **Active Directory (AD)** — Microsoft's system for managing users, computers, and permissions across a network. Foundational knowledge for almost any IT or security role.
- ▶ **Blue Team** — The defensive side of cybersecurity: monitoring, detecting, and responding to threats.
- ▶ **Red Team / Offensive Security** — The attacking side: finding weaknesses before real attackers do, with permission.
- ▶ **CISO** — Chief Information Security Officer, the most senior security leadership role in most organizations.
- ▶ **CISSP, CISM, CCSP** — Advanced, experience-required certifications for security leadership, management, and cloud security respectively.
- ▶ **Cloud (AWS, Azure, GCP)** — The major providers of on-demand computing infrastructure that most modern companies run on instead of, or alongside, their own physical servers.
- ▶ **CySA+, PenTest+** — CompTIA's mid-level certifications for defensive analysis and offensive testing, the natural next step after Security+.
- ▶ **DNS (Domain Name System)** — The system that translates a website name into the address a computer actually uses to find it. One of the most foundational networking concepts in this field.
- ▶ **DevOps** — The practice of combining software development and IT operations, usually with heavy automation, to ship and run software faster and more reliably.
- ▶ **GRC (Governance, Risk, and Compliance)** — The lane that bridges technical security work with legal, regulatory, and business requirements.
- ▶ **Home lab** — A personal setup, often just virtual machines on your own computer, used to practice real skills hands-on.
- ▶ **Kali Linux** — A Linux distribution preloaded with offensive security tools, standard for anyone practicing penetration testing.
- ▶ **n8n** — A free, self-hostable workflow automation tool. In a home lab, it connects your separate projects into one system: alerting, reporting, and scheduled checks.

- ▶ **NIST, ISO 27001** — Widely used governance and security frameworks that GRC professionals work with regularly.
- ▶ **Pi-hole** — A network-wide DNS sinkhole that blocks ads and malicious domains for every device on a network. A common and genuinely useful first home-lab project.
- ▶ **SIEM (Security Information and Event Management)** — A tool that collects and analyzes security data across a network, central to most Blue Team work. Wazuh is a free, open-source option well suited to a home lab.
- ▶ **Tailscale / WireGuard** — A modern, simple way to build a secure private network for remote access to your own servers and labs.
- ▶ **VLAN (Virtual LAN)** — A way of splitting one physical network into separate isolated segments, so a compromised device can't reach everything else.
- ▶ **STAR Method** — Situation, Task, Action, Result. The structure for telling any professional story clearly.
- ▶ **T.R.A.C.K.** — Tailor, Research, Apply, Connect, Keep following up. This blueprint's system for actually landing a role.
- ▶ **V.A.L.U.E.** — Validate, Ask, Listen, Underscore, Engage. This blueprint's framework for negotiating and selling yourself.
- ▶ **Vulnerability** — A weakness in a system that could be exploited. Finding and fixing these is the core of most security work.
- ▶ **Zero-day** — A vulnerability that's being exploited before the vendor has had a chance to fix it, among the most serious findings in the field.

QUICK REFERENCE: EVERY FRAMEWORK IN ONE PLACE

Bookmark this section. Once you've read the full blueprint once, this is the page to come back to.

- ▶ **The Six-Level Path** — Entry, Fit, Breadth, Differentiation, Ownership, Executive Thinking. The overall map.
- ▶ **The Career Value Stack** — Curiosity, Technology, Communication, Business & Sales, Leadership. What's actually being built at every level.
- ▶ **The Curiosity Compass** — Fix, Build, Break, Defend, Develop, Translate. Find your primary lane.
- ▶ **The T.R.A.C.K. Application System** — Tailor, Research, Apply, Connect, Keep following up. How to actually get hired.
- ▶ **The STAR Method** — Situation, Task, Action, Result. How to tell any professional story.
- ▶ **The V.A.L.U.E. Conversation** — Validate, Ask, Listen, Underscore, Engage. How to negotiate or sell yourself.
- ▶ **The Reputation Loop** — Do great work, ask for feedback, route it to leadership, document the proof.
- ▶ **The Portfolio Project Documentation Wheel** — Problem, constraint, action, result, proof. Document every real project.
- ▶ **The Wake Up X Decision Filter** — Impact, reversibility, data, stakeholders, timing. How to decide under pressure.
- ▶ **The Career Health Score** — Skill growth, network strength, documentation, financial position, energy. Rate yourself every quarter.
- ▶ **The Connection Flywheel** — Show up, give first, stay visible, get known, get referred. How real networking compounds.
- ▶ **The Personal Board of Directors** — A technical mentor, a business mentor, a peer, and someone outside the field entirely.
- ▶ **The Executive Dashboard** — What moved, what's at risk, who you helped, what you learned. Track it monthly.

Every one of these is meant to be remixed into your own version, not copied exactly. That's the whole philosophy behind this blueprint.

KEEP GOING

This blueprint is free, and it's meant to be shared. If it gave you a clearer map than the one you started with, the best next step is simple: follow along, and pass it to one person who needs it.

BE OUR NEXT GUEST

Your Story Belongs on Wake Up X

If DECODE gave you a map, you already have something worth sharing. The Wake Up X podcast puts real, unpolished career and business stories in front of people who need to hear them — and every guest so far started exactly where you are now.

You might be a fit if you:

Broke into tech, cybersecurity, or business without the traditional path

Built something — a lab, a business, a portfolio, a community — and can talk about how

Have a failure or turning point you're willing to tell straight, not polished

Are mentoring, leading, or building something worth other people knowing about

HOW TO APPLY

Share your story at wakeupxpodcast.com/guest, or connect directly on LinkedIn at linkedin.com/in/efrenlsalazar and mention DECODE.

Acknowledgments

THE PEOPLE WHO MADE THIS POSSIBLE

None of Wake Up X happened alone, including DECODE. Every book in this library, the podcast, and the free library exist because specific people showed up for a kid nobody expected much from. This is the record of who they are.

THE EARLY YEARS

Miss England, the 6th-grade teacher who stayed after class, unpaid, to teach the fundamentals. Sister Cora and Lewis Bair of Computers for Kids, who handed a teenager a key and a broken machine and let the work speak. Jose Luis Bravo, Angel Martinez, Isaac Murrilo, and Alex De Haro — the friends and cousins who translated the world, gave hope, drove him where he needed to go, and carried him before there was anything to carry him toward.

THE BUSINESS YEARS

Rojelio Salazar, his cousin and first mentor, RNS Computer Repairs — the reason it is called Wake Up X. Kyle Erni, his first business partner. Erick and Pancho Arauza, co-founders of The Cellphone Fix. Danny Maya, of Maya Traditional Mexican Restaurant, Taqueria Maya, and Salt and Water, who pushed him in front of a camera for the first time.

THE CAREER YEARS

Igor, who gave him the vocabulary for work he already knew how to do. Nate, who he mentored into a full-time role alongside his own. Ken Kerrick, opponent turned friend in the ISACA Sacramento election. Alvin Jue and David De los Reyes, who believed in the middle of it, not just at the finish.

THE PODCAST

Every guest who trusted an unproven show with their story, including Gustavo Mastroianni, Cathy Stephenson, Greg McCord, Aaron Portnoy, Sana Talwar, Jorge Avila, Anil Reddy, Devin Prasad, Larry Switzer, Louis Kemp, Mohana Balakrishnan, Eddie Lopez, and Michael Fierros — full episodes at wakeupxpodcast.com.

COMMUNITY PARTNERS

ISACA Sacramento and ISC2 Sacramento, The Cellphone Fix, A.N.U.L Technologies, AISOC, AspireByte, and PodGrabber — the organizations building alongside this movement.

AND

His parents, who arrived with nothing, worked since childhood, and gave away whatever was left over. Nothing about the first version of this story predicted the version you are holding now.

Continue the Journey

EVERY BOOK LEADS SOMEWHERE ELSE

DECODE is one part of a 12-book system. If it helped, the next step is one page away — take whichever layer matches where you are standing.

THE LIBRARY

Layer 1 — The Story

I Just Have To Wake Up — why any of this exists.

Layer 2 — The Mindset

Comfort Is Expensive — the real cost of playing it safe.

Layer 3 — The Blueprint

DECODE: The Wake Up X Playbook — zero experience to executive thinking. Book 0.

Layer 4 — The Playbooks

1 Break In · 2 Build It · 3 Think Like a Cybersecurity Professional · 4 Business First · 5 Lead Without a Title · 6 AI for IT & Cybersecurity · 7 Build Your Brand · 8 The Consultant Mindset · 9 Level Up

FIND US EVERYWHERE

Podcast & free library: wakeupxpodcast.com · Spotify: open.spotify.com/show/28GZnRBrIV2efHOHzdiftt · Apple Podcasts: podcasts.apple.com/us/podcast/id1792821643 · YouTube: youtube.com/@Efren.wakeupx · LinkedIn: linkedin.com/in/efrenlsalazar · Substack: substack.com/@wakeupxefren

IN GOOD COMPANY

ISACA Sacramento, ISC2 Sacramento, The Cellphone Fix, A.N.U.L Technologies, AISOC, AspireByte, and PodGrabber — the partners and collaborators building alongside the movement. See the full list at wakeupxpodcast.com/community.html.

Get DECODE free, apply to be a guest, or read the story behind the movement — all of it starts at wakeupxpodcast.com.

WAKE UP X

SOLUTIONS



Efren L. Salazar

FOUNDER · HOST · CYBERSECURITY PROFESSIONAL · EDUCATOR · COMMUNITY BUILDER

STAY CONNECTED

Podcast & Blueprint Library:	wakeupxpodcast.com
LinkedIn:	linkedin.com/in/efrenlsalazar
YouTube:	youtube.com/@Efren.wakeupx
Be a Guest:	wakeupxpodcast.com/guest

THE WAKE UP X SOLUTIONS

STARTING
FROM NOTHING
BUILT TO
SOLVE EVERYTHING

MENTORSHIP
DISCIPLINE
HARD WORK
LEADERSHIP
PURPOSE
IMPACT

ONE KID. ONE COMPUTER. ONE DECISION.
EVERYTHING CHANGED.

THE WAKE UP X MOVEMENT

I Just Have To Wake Up

The Cousin Who Woke Me Up, and Everything I Built After

EFREN L. SALAZAR

Founder and CEO, Wake Up X Solutions · Host, The Wake Up X Podcast
Government & Regulatory Partner, ISACA Sacramento

FOR ROJELIO SALAZAR

My cousin. My first mentor. My best friend.

He believed in me before I believed in myself.

This book is dedicated to my cousin, Rojelio Salazar, who ran RNS Computer Repairs and put his own name on the sign before I knew that was something a person like me was allowed to do.

He was the first person who ever took the time to really uplift me. Not just encourage me. Actually sit down and show me how business works, from the ground all the way up. Nobody else in my life was doing that. He was there when nobody else was willing to help me out.

He would drive about an hour and a half just to come see me. Just to check on me. He called every day, and we would talk about nothing and everything, the new tech, whatever was going on. We were a good team. A partner and a best friend at the same time.

He was always looking after his family, making sure everybody was good. That is where I learned that dedication to your family comes first. Your family may not always see the bigger picture of the life you are trying to build. But the ones who take that small moment to appreciate you and support you anyway, those are the ones you owe. Give it back to them, and give it back to the community they came from.

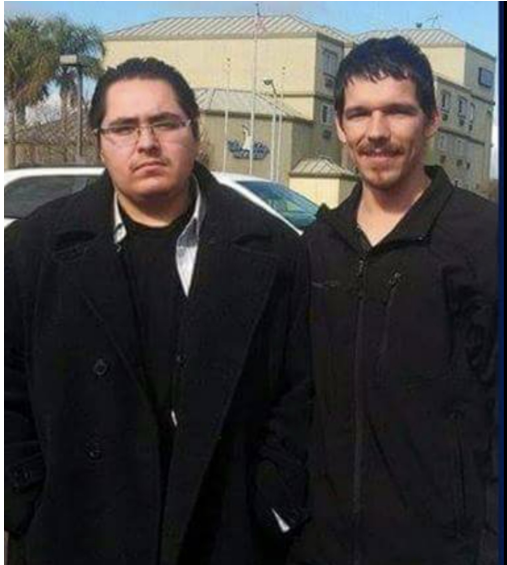
His passing was one of the hardest moments of my life. I went into a depression after it. I felt like I had lost everything, and there was a lot of internal trauma in our family in that same stretch. That was when a lot of things stopped going right, and it is still hard for me to talk about, even now.

So I want to give a huge appreciation to him, and to all of his family too. Our whole family has always been like that.

If it were not for him, I would not be the person I am right now. I would not have had that kind of mentor, that education, or those resources. He made all of this possible. He is the reason it is called Wake Up X, because he is the one who woke me up out of my own reality.

I got out. I got into my business. I did not stop.

RNS Computer Repairs



Rojelio Salazar

You told me I was lazy.
You were right.

But you saw what nobody
else did. Not lazy —
strategic. You called it
a gift, not a flaw.

Nobody believed in me from day one. Nobody but you.

When the whole world said I was crazy, you called anyway — every time. "Wake up, man. Let's get the mission running." That call is why this entire movement carries your words in its name.

I'm not building this to pay a debt. I'm building it because it was always ours — the same fire, the same mission, the same belief that we could help people nobody else would fight for.

You lived that first. RNS Computer Repairs. Your name, on your own sign, before anyone gave you permission to put it there.

I miss you every single day. This book — this whole library — belongs to you.

**Thank you for believing in me when no one else would.
Rest easy, cousin. We're living our dream, together, right now.**

I JUST HAVE TO WAKE UP

The Cousin Who Woke Me Up, and Everything I Built After

Copyright © 2026 Efren L. Salazar. All rights reserved.

Published by Wake Up X Solutions, Sacramento, California.

No part of this book may be reproduced or distributed in any form without the written permission of the author, except for brief quotations used in reviews or educational settings with attribution.

This is a work of nonfiction. The events described are recounted from the author's memory and to the best of his recollection. Certain dates are approximate. Some names have been withheld out of respect for the privacy of the people involved.

The views expressed here are the author's own and do not represent those of any employer, client, or professional association.

First edition, version 1.0 · August 2026

Part of the Wake Up X library: two gateway books plus the ten-book career library, Book 0 through Book 9. All of it free at www.wakeupxpodcast.com.

www.wakeupxpodcast.com

Contents

- 01 The Verdict
- 02 Miss England
- 03 The Computer With No Mouse
- 04 The Ones Who Were There First
- 05 The Janitor Who Wanted to Understand
- 06 Locked Out
- 07 Where the Standard Came From
- 08 Street Heart
- 09 Belonging Nowhere, and Choosing to Uplift Anyway
- 10 Two Doors I Walked Through Early
- 11 My Name on the Banner
- 12 The Cellphone Fix
- 13 I Just Have To Wake Up
- 14 The Book That Turned the Volume Up
- 15 The Corporate Classroom
- 16 Nobody Handed Me the Opportunity
- 17 ISACA and the Community
- 18 The Rebuild Nobody Saw
- 19 The Mission

Where This Book Fits

This is layer one of four. It is the why. Everything else is the how.

Wake Up X is not a pile of separate PDFs. It is one system with four entry points, and you are holding the first one.

This book is the story. Where I came from, who carried me, what it cost, and why any of the rest exists. It is the only one of the four that is about a life rather than a method.

Layer two is Comfort Is Expensive. That is the mindset version, short and direct: what staying comfortable actually charges you, how to leave, and why you turn around and hold the door once you are through. If you want the lessons without the full biography, start there instead.

Layer three is DECODE: The Wake Up X Playbook, which the library lists as Book 0. The map from zero experience to executive thinking, no degree required.

Layer four is Books 1 through 9, each aimed at one specific problem you might be facing this week. Book 0 plus those nine is the whole career library, ten books in total.

Story explains why. Mindset explains how you think. The blueprint explains what to do. The playbooks handle the problem in front of you today.

THE ECOSYSTEM

Four Layers, One System

Start anywhere. Every layer points to the next.

LAYER 1 · THE STORY

YOU ARE HERE

I Just Have To Wake Up

Why Wake Up X exists. The full life story, the people, the timeline.



LAYER 2 · THE MINDSET

Comfort Is Expensive

How to leave the comfort zone, think bigger, and give it back.



LAYER 3 · THE BLUEPRINT

DECODE: The Wake Up X Career Blueprint

The career map. Zero experience to executive thinking.



LAYER 4 · THE PLAYBOOKS

1 Break In · 2 Build It · 3 Think Like a Pro · 4 Business First · 5 Lead Without a Title

6 AI for IT & Cyber · 7 Build Your Brand · 8 The Consultant Mindset · 9 Level Up

Four layers, one system. This book is layer 1.

Choose Your Path

You do not have to read everything, and you do not have to read it in order.

If you only want the story, this book is the whole thing. But if reading it makes you want the method underneath it, here is where to go depending on where you are standing right now.

Break into tech. No experience yet and you need a first role. Start with Book 0, DECODE, then Break In, then Build It.

Break into cybersecurity. Already working in IT and you want to move across. Book 0, DECODE, then Think Like a Cybersecurity Professional, then Build It.

Sharpen your security and AI skills. You are already in and you want to get better. Think Like a Cybersecurity Professional, then AI for IT and Cybersecurity, then Level Up.

Business, consulting, and brand. You want to lead, consult, or be found. Business First, Lead Without a Title, Build Your Brand, then The Consultant Mindset.

Give me everything. You are not sure which fits. Take the whole library and start wherever you like.

All of it is free at www.wakeupxpodcast.com. That site is the hub, and everything points back to it.

CHOOSE YOUR PATH

CHOOSE YOUR PATH

Pick the Path That Fits

You do not have to read everything, or read it in order.

BREAK INTO TECH

No experience yet, need a first role.

[DECODE](#) › [Break In](#) › [Build It](#)

BREAK INTO CYBER

Already in IT, want to move across.

[DECODE](#) › [Think Like a Pro](#) › [Build It](#)

SHARPEN SECURITY & AI

You are in, want to get sharper.

[Think Like a Pro](#) › [AI for IT & Cyber](#) › [Level Up](#)

BUSINESS, CONSULTING & BRAND

Want to lead, consult, or be found.

[Business First](#) › [Lead Without a Title](#) › [Build Your Brand](#) › [Consultant Mindset](#)

JUST THE STORY

You want to know why any of this exists.

[I Just Have To Wake Up](#) › [Comfort Is Expensive](#)

GIVE ME EVERYTHING

Not sure which fits? Take the whole library.

[All ten books, free, in any order you like](#)

No book is a dead end. Every path leads somewhere else. Free at wakeuxpodcast.com.

Before You Start

TAKE WHAT WORKS. BUILD WHAT'S YOURS.

I want to be clear about something before you go any further. Everything in this book comes from years of learning, listening, reading, studying, experimenting, and paying attention to what actually worked in my life. I've listened to countless audiobooks, read books, studied frameworks, learned from speakers, collected quotes, and explored different blueprints for success.

But I didn't follow every blueprint exactly. I took the parts that made sense to me. I kept the lessons that actually helped. I tested them in my own life, in my own environment, and then I combined them into something that worked for me. That is what I want you to do with this book.

Don't copy my life. Build your own blueprint. Not everything in here will work for you, and that's okay. You may read one chapter and find one idea that changes how you think. You may find another where only a paragraph stands out. You may disagree with some of it. You may need to modify things to fit your career, your family, your finances, your environment, or your goals.

I'm not here to promise you that this framework will magically transform everyone's life. What I can tell you is that it worked for me. It was built through experience, mistakes, setbacks, and years of putting different pieces together. These aren't ideas I pulled out of thin air. They are lessons I gathered, tested, adapted, and implemented on my own journey.

There were also times in my life when I was surrounded by difficult environments and influences. Those experiences became part of my story and shaped how I viewed choices, consequences, survival, and opportunity. But this book isn't about those environments. It's about what I learned from them, and what I chose to do with those lessons.

My goal is to give you the pieces. You decide how to put them together. Take what resonates. Test it. Adjust it. Improve it. Throw away what doesn't work. Then build something that belongs to you.

Because the ultimate goal was never to become me. It's to become the best version of yourself, and to create a blueprint that works for your life.

THE METHOD

Build Your Own Blueprint

Take what works. Throw away what doesn't.

1

TAKE THE HIGHLIGHTS

Read widely. Keep the ideas that actually land for you.

2

TEST IT IN YOUR LIFE

Not in theory. In your job, your budget, your week.

3

ADJUST IT

Fit it to your career, family, finances, environment.

4

DISCARD WHAT FAILS

If it does not work for you, it is not your blueprint.

5

COMBINE THE REST

The pieces that survive become your own system.

6

BECOME THE BEST VERSION

The goal was never to become me. It was to become you.

Six steps. Use them on this book first.

THE ROAD SO FAR

Not a Straight Line

It never is.

- 
- Age 10** ● Taking clocks apart just to see how they worked.
 - 6th grade** ● Miss England stays after class. "Everybody learns differently."
 - Age 13** ● A donated PC with no mouse. Learns to run it all by keyboard.
 - Age 16** ● Neighbors knocking on the window. Repairs out of the front room.
 - 2008** ● Job Corp promised 3 months, told 4 years. Drops out.
 - 2008–2013** ● Computers for Kids — volunteer, then their IT Manager.
 - 2011** ● Moves to Salt Lake City. CompuFix Utah.
 - 2014–2021** ● The Cellphone Fix — CTO: infrastructure, apps, cyber, risk.
 - 2016** ● Rojelio dies. Four words: I just have to wake up.
 - Sep 2021** ● Founds Wake Up X Solutions, the month the shop closes.
 - 2022–2023** ● Two contracts, back at entry level. On purpose.
 - 2023–now** ● Ops lead, then operations manager and solution architect.
 - Now** ● Security+. ISACA Sacramento partner. A library given away free.

Not a straight line. It never is.

CHAPTER 1

The Verdict

They told me I could not learn. I decided that was a problem to solve, not a sentence to serve.

I grew up in a farm-working, immigrant family in the Sacramento area. Money was tight. Resources were tighter. English was not my first language, and I did not really speak it comfortably until I was around sixteen.

School handled that the way schools handle a lot of kids who do not fit the standard mold. I was placed in ESL classes, then special education. Teachers told me directly, more than once, that I could not learn. Not that I was behind. Not that I needed more time. That I could not.

Here is what I understood about myself even as a kid, even without anyone ever putting a name to it: things that came easy to other people took me longer. I would have to read something ten or twenty times before it actually landed. I have never been formally diagnosed with anything, but I believe I am dyslexic. Nobody ever tested me for it. They just moved me down a level, and then down another one, until I was in the most basic classes the school had.

That was one of the most discouraging moments of my life, and it is also the moment something hardened in me. I remember thinking: I am never coming back to this place. Being told you cannot learn and actually not being able to learn are two completely different things, and it took me years to understand that the school had confused the two.

Nobody handed me a manual for how school worked, how English worked, or how any of this was supposed to go. So I did to my own life what I would later learn to do to systems for a living. Watch it closely. Find the pattern. Take it apart. Rebuild it well enough to use.

Never let your current title determine your future potential.

That instinct did not start as a business method. It started as survival. It became a system I could teach only much later, once I understood what I had actually done.

CHAPTER 2

Miss England

One teacher stayed after class. That is the whole reason any of this exists.

In sixth grade, a teacher named Miss England stayed after class on her own time and taught me the fundamentals. Letters, sounds, reading, one on one. Nobody was making her do it. Nobody was paying her for it.

She is the one who told me something no other teacher ever had: everybody learns differently. That sentence sounds small written down. It was not small to me. It was the first time an adult in a classroom suggested that the problem was not that I was broken. It was that nobody had found the

right way to reach me yet.

I had people around me outside of school who noticed things too. They would tell me when a teacher was being unfair, or point out that I was being singled out for being a little different. I am not telling you this to make myself the victim of it. I am telling you because that support, small as it sometimes was, is part of why I did not fully give up on myself.

If you're reading this, Miss England: thank you for not giving up when everyone else already had.

If one teacher giving a little extra time with no reason to can change the entire shape of a kid's future, then the mentorship model I have built everything else around is not a theory. I lived it first.

CHAPTER 3

The Computer With No Mouse

The start of a technology career, and it did not look like one.

The curiosity showed up early. By ten I was taking clocks apart and wiring speakers into them just to see what would happen.

At thirteen, my family received a donated computer. An early-90s Windows machine with no working mouse. Most people would have called that broken and moved on. I taught myself to run the entire thing from the keyboard alone. The start menu, the command prompt, all of it. Nobody showed me how. I just refused to let a missing part be the reason I stopped.

That machine, not a class and not a certification, is where the career actually started. From there I began fixing computers and phones for anyone who would let me, language barrier and all.

CHAPTER 4

The Ones Who Were There First

Before any mentor, any partner, and any teacher, there were four people who carried me.

I was the only kid in my house. No brothers, no sisters. I was quiet to the point of silence, I did not speak English, and my older cousins had to translate the world for me. So the people who ended up raising me alongside my parents were friends and cousins, and I am not finishing this book without naming them.

Jose Luis Bravo was my first friend and he is still my best friend. We were neighbors. I came from a farm town and most kids drifted away, and he was the one who stayed. We fought constantly because we knew exactly how to get under each other's skin, which is its own kind of closeness.

He got a job before I did, and for years I was measured against him. My parents, everybody around me: that guy is doing better than you, you should be doing the same. It stung when I was young and it drove me when I got older, until one day the thought landed as: stop comparing me to anybody.

Look at what I can do.

I used to ask him to get me in where he worked. He would say yeah, man, I got you, and then not do it. Here is the part people expect me to be bitter about, and I am not. He gave me the hint. He told me what they required, and the rest was on me. That was correct. I was not ready. I was insecure, I was not confident, and I was not responsible enough yet for a real commitment, and if he had handed it to me I would have settled for less. I am glad that door never opened. Take the hint somebody gives you and be grateful for it. Do not expect them to carry the whole thing.

Nothing was ever given to me, and that is exactly why I appreciate the smallest gesture.

Angel Martinez gave me hope when I had none. We were in the special classes together in seventh and eighth grade. He knew I could not speak English, could not communicate, and could not stand up for myself, and he decided that was his problem too. He translated for me. He taught me English. He talked to people on my behalf until I could do it myself.

When teachers came down on me for speaking Spanish, because that was genuinely all I had, Angel was the one who spoke up. He would tell them that is all he knows, so if you want to teach him, then teach him, let us teach him together. He was younger than me and he acted like my older brother. He advocated for me before I knew the word advocate.

Isaac Murrilo was my first marketing department, years before I knew that was a job. He was my number one supporter for the computer repair work and he advocated for me everywhere he went. Efren knows computers. Get your laptop fixed by Efren. If you need this, he can do it. He is a big part of why I started believing I could make something out of it. The rest of his story is his own to tell, but I am not leaving his name out of mine.

Then there is my cousin Alejandro, who everyone calls Alex De Haro. He is more like my brother. He took me under his wing in high school and made sure I was safe. He checked on me constantly, no matter what. I did not have transportation or money, so he drove me where I needed to go and covered what I could not. He was my backbone, and he is still my mentor.

He also gave me the line I still repeat to myself in every room I walk into. I used to tell him I was ugly, that he was the good-looking one. He said it is not how you look. It is never how you look. It is how you talk. That has held up in front of every executive I have ever met.

I was a messed-up puzzle that had not been put together yet. Every one of these people handed me a piece.

CHAPTER 5

The Janitor Who Wanted to Understand

If nobody will open a door for you, walk through one that is already open.

I started out volunteering. Not for a paycheck, because there was not one. I volunteered until the Calling Teen Center in West Sacramento put me on payroll as a dishwasher and a janitor. That is the actual order, and I want it stated plainly, because people skip past the unpaid part when they tell these stories.

Dishes and floors. Not glamorous. But it put me inside a building that also ran a program called Computers for Kids, led by a professor from Sac State named Lewis Bair. That is the whole argument for taking the job nobody wants. It puts you inside the building.

Computers for Kids was a nonprofit with a simple, powerful idea behind it. Families who could not afford a computer should still have one. Donated machines came in broken, outdated, or wiped. Volunteers rebuilt them. Then a family that had never owned a computer walked out with one that worked.

The deal for volunteers was just as simple. Put in a set number of hours, walk away with a free computer of your own. I was not really there for the computer. I asked Lewis directly if I could learn on the side, after my shift, and he made it happen. He gave me time after clock-out to sit with the equipment and actually understand it.

Understand what that meant for a kid who had been told he could not learn. Nobody at C4K asked for my grades. Nobody looked at what track I had been placed in at school. They handed me a broken machine and a screwdriver and let the work speak. It was the first place in my life where the only question was whether I could figure it out. I started at essentially help-desk level, walking people through machines they had never really used. A lot of them were parents and grandparents in my own community, sitting down at a keyboard for the first time in their lives, embarrassed about it. I knew that feeling from the other side of the desk. I had been the kid in the room who did not understand what everyone else seemed to. So I never once made anybody feel slow, and I have kept that rule in every room since.

The longer I stayed, the more I wanted to understand how the whole operation ran underneath. Not because I thought I knew better than anyone, but because I could not turn the question off. That curiosity moved me from help desk into a supervisor role running the administrative side, and eventually into managing the technology for the program itself.

Around the same time, a nun named Sister Cora gave me a key and a small space where I taught English and computer literacy to people in the local Latino community. She did not ask me for a resume either. She handed me a key. I was already teaching kids at the center, and it made sense to bring the same thing to people in my own community who needed it. Sister Cora, thank you for trusting a kid with a room. By sixteen, neighbors were leaving computers on my doorstep. Twenty dollars for one repair, forty for another, some days close to a hundred and fifty. People would knock on my bedroom window at home to drop off a machine or pick one up, enough of it that some neighbors got curious about what was actually going on in that house. It was never anything but computer repair. I was a teenager working janitorial and dish shifts by day and quietly running a business out of my own front room by night.

If nobody wants to give me the opportunity, I'm going to make my own opportunity.

I want to be clear about how long that door stayed open, because people assume it was a summer. I stayed with Computers for Kids from September 2008 to April 2013. Almost five years, most of it as a volunteer and an intern, and then in September 2013 I came back as their Information Technology Manager. The kid who came in washing dishes ended up running the technology for the program that taught him.

Everything I do now is that program repeated at a larger scale. A free library instead of free computers. Mentorship instead of after-shift lessons. The elderly learning technology for the first time, exactly like the ones I sat with at that center. I did not invent the Wake Up X model. I learned it at Computers for Kids and I have been running it ever since.

Computers for Kids, more than any class or certification, is the real foundation everything after it was built on. Lewis Bair, thank you for the door you opened for a kid who just wanted to understand how things worked. That opportunity is still paying forward today.

CHAPTER 6

Locked Out

Two doors closed on me before I was twenty. Both of them taught me the same thing.

I tried to get into a computer class in high school and got rejected because my grades were not good enough. More than one teacher told me some version of the same thing. You will never do it. You are not smart enough. You do not have the grades. Those moments made something click. I do not care what anybody says. I do not care what anybody tells me. I will do it anyway, my own way. So I started teaching myself, reverse-engineering how things worked, one book at a time.

A few years later I tried another way in through Job Corp. I had been about a year from finishing my high school diploma, and they told me if I transferred in, I could finish in about three months. After I signed up, I found out none of my credits transferred. It would have taken four more years. On top of that disappointment, the place itself felt less like a school and more like a holding pen, full of the same kind of environment I had been trying to get away from in the first place, just concentrated and made worse. I dropped out. That was around 2008. Utah was still three years away. I did not get my actual high school diploma until I was close to thirty, and when I finally did, it was not because I needed it to succeed. By then I had already built and run more than one real business without it. I finished it to close that chapter on my own terms.

CHAPTER 7

Where the Standard Came From

My parents arrived with nothing and gave most of it away anyway.

My family was born and raised in Mexico, and they came into this country as farm workers with no money, no housing lined up, and no English. They had been working since they were five and six

years old. That is not an exaggeration. That was the arrangement.

My father only ever reached the third grade in Mexico. That was the highest he got. The part that stays with me is that he loved school. He loved math. He loved learning. He never got the chance. When people ask where my drive came from, that is the honest answer. I was handed an opportunity he wanted his whole life and never received. My mother was very young when she had me. A kid raising a kid, basically. She has told me how hard that was, and I want to say plainly that she did an amazing job carrying a load nobody that age should have to carry.

Like any family, there were conflicts and hard moments I saw growing up. I am not going to pretend otherwise. What I did with them was simple. I looked at the good parts of my parents and kept those. I looked at the hard parts and decided those were things I would not repeat. That is how I look at life in general now. There is always good and there is always bad, in every situation and every person.

They built a life with nothing and then spent whatever was left over helping the people around them, family and strangers both. Sometimes they got taken advantage of for exactly that generosity. Watching that removed every excuse I could have made for myself. If they could do that with nothing, I had no right to sit on what little I did have.

CHAPTER 8

Street Heart

Almost everyone I knew had a good heart. They were just in a bad lifestyle.

I grew up in West Sacramento in the 1990s, when gang activity was at its peak. That was the environment. Friends went to prison. Friends overdosed. Friends died. I felt like I was fighting almost every day, physically or mentally, and I want to be honest about what that does to you. It stops looking like tragedy and starts looking like the ordinary shape of a life. For a while, it did to me.

Here is the part people get wrong about where I am from. Almost everyone I knew had a good heart. They were in a bad lifestyle, and they got judged for their tattoos and the way they carried themselves, but the hearts were good. My people used to tell me: you see the inside of us, not the outside. That stayed with me, and it is the same skill I use now in rooms full of executives. I read people. I can usually tell when somebody just wants help, or just wants to be seen.

Some of the older men around me told me to get out while I still could. Others told me the opposite. Then there was my cousin, Rojelio Salazar.

Rojelio ran his own computer shop. RNS Computer Repairs. His name on it, his business, his own thing built from nothing, and I want you to understand what that looked like to me at that age. Everybody I knew worked for somebody else or worked outside the law. Rojelio did neither. He fixed computers, he put his own initials on the sign, and he answered to himself.

That is the part people miss when they ask how a kid from where I am from ends up running businesses. It is not that I was ambitious. It is that I saw it done. One person in my family, one shop, one name over the door, and suddenly the thing was not theoretical anymore. RNS Computer Repairs is the reason I ever believed a business was something a person like me could own instead of just work for.

He was humble about it, and he was always lifting up the people around him. He never made me feel like a kid hanging around. He talked to me like somebody who was going to run his own shop one day, before there was any evidence of that at all. And then he told me so plainly.

Get out of the streets. I went through this already and it is not worth it. Get into your business. You're already smart enough. Keep going and don't stop.

I did not know it then, but that sentence had a second half coming, and it would not arrive for another fifteen years.

CHAPTER 9

Belonging Nowhere, and Choosing to Uplift

Anyway The gap I found is exactly where I decided to stand.

From 2008 to 2011 I was in Sacramento, riding out a market crash that took California's work with it. The repairs slowed down. There were no jobs to apply for. I stuck it out for three years before I accepted that the work was not coming back, and in 2011 I moved to Salt Lake City, Utah, to go find some.

I took the business with me. By then it had a name, CompuFix Sacramento, built with Kyle Erni as my partner. When I moved, it became CompuFix Utah. Same company, same hands, new state on the sign. I ran it while working full days as a line cook to pay rent, and I stayed through the end of 2012.

The restaurant job is the part worth telling honestly. I am Mexican American, first generation, and I figured that would count for something in a kitchen full of people from the same background. Instead, coworkers who were Mexican themselves made sure I knew I did not belong. I was told more than once that I was not important enough. That even if I thought I was better, I was never more and above them. That they were the real, traditional Mexicans, the hard workers, and I was not.

I had no comeback in the moment, and I did not go looking for one. What I told myself instead was this. Maybe I did not have the same potential they believed they had, but I had something different. Dedication, and a willingness to learn and to extend what I knew. I used it as motivation to move faster, not to prove anyone wrong to their face. I never let them see it land.

Part of that was understanding, even then, that this would keep happening in different rooms for the rest of my life. Part of it was already knowing it was not going to be the thing that decided my future. Obstacles like that are always going to happen. They do not define what comes after them.

Growing up, I never fully fit the American side or the Mexican side. Sitting outside both gave me a strange kind of clarity about a pattern I have watched repeat in room after room since. A tendency to push each other down to look better by comparison, instead of pulling each other up.

So I made a decision a long time ago. If almost everyone around me was willing to belittle, and almost nobody was willing to uplift, then that gap was exactly where I could make a difference. Everything I have built since comes out of the same choice: be the uplift I did not get.

CHAPTER 10

Two Doors I Walked Through Early

One of them fired me. One of them showed me what a real company looks like.

Before any of the corporate work that stuck, there were two rooms I got into early, and both of them taught me something I still use.

United Airlines was my first corporate IT job. Three months in, I got fired. I want to be precise about why, because the easy version of this story is that I was not good enough. That is not what happened. It was a communication failure, and it ran downward, not upward. Nobody gave me the right resources. Nobody gave me the structure to work inside of. I was expected to perform in a system that had never been explained to me.

I do not say that to take myself off the hook. I say it because it is a leadership lesson and I have never forgotten it. If your people do not know what the structure is, the structure does not exist. Everything falls down when nothing is organized. That is on whoever is running it.

Communication is not a soft skill. It is the whole structure.

The other door was Google, through a contract with Astreya Partners, in Mountain View, California. Summer of 2013, on the audio and video engineering side, at the corporate campus.

I started on a Friday, which is the funniest possible day to start at Google. There were DJs. Music. Events. Game systems everywhere. I walked in and thought I had misread the whole thing, and then somebody said hey, it's Friday, let's go get some beers. My honest reaction was: how are we going to drink at work?

But that was Google. They provided everything. High-end food, drinks, rides home so nobody drove, all of it free. This was before Uber, so they handed you a taxi. It was an adult playground.

What actually amazed me was not the perks. It was the structure underneath them. How organized it was. How friendly it was. How much thought had gone into making it work. I had come from folding carts in hair salons and repair counters, and here was a company that had engineered the entire experience of working there.

Put those two jobs side by side and you get the whole lesson. One had no structure and it cost me the job. The other had structure everywhere you looked and it is why people gave it everything they had. I have built every business, every team, and every framework since with that comparison in my

head.

CHAPTER 11

My Name on the Banner

A folding cart in a hair salon, and my name on the banner.

I came back to Sacramento around 2013. Kyle wanted to go do something different, which he had every right to do. So I took over the full business myself and renamed it one last time. CompuFix Sacramento, then CompuFix Utah, now Compu-Fix 2u. One company, three names, and for the first time the whole thing was mine.

It was a mobile setup. Hardware and software upgrades, Mac and PC repairs, virus removal, factory unlocks, free diagnostics, run out of whatever borrowed space I could get. For a while that meant a folding cart set up inside a hair salon in West Sacramento. No funding, no lease, no sign over a door. A phone number, a banner, and a habit of showing up wherever a client would let me work.

I want to give Kyle Erni his credit here, because the story does not start with me. When I was a teenager, he was the one fixing computers in West Sacramento. He was the innovator. When he left the state, his customers started coming to me, and that is how I inherited a reputation I had not built yet. Back then I had no company and no name. People just knew me as Efren the tech, the guy who fixed computers. Later he came back and we built CompuFix Sacramento together.

So the sequence looks like this. The skill came first. The customers came second. The name came third. Full ownership came last, and it took years. Most people try to run that in reverse. They want the brand, the office, and the title before they have anything anyone actually needs.

Compu-Fix 2u never really stopped either. Even after the next thing started, I kept the mobile side running alongside it.

Kyle and I are still in each other's lives, and he is a successful engineer now. Neither of us ended up where a folding cart and a borrowed corner would have predicted. There is one job from those years I want on the record, because it taught me something about myself I would not have learned any easier way. I spent a stretch in the Bay Area on a contract for NetApp doing IAM, identity and access management. Access

removals, provisioning, credentials. I did not have a name for that discipline at the time. I just picked it up.

They fired me. I made a lot of mistakes, and I am not going to dress that up as unfair, because it was not. But the feedback I got on the way out is something I have carried ever since. I was the fastest learner there. Faster than anyone. I just moved so fast that I broke things behind me.

That was the day I found out I had an actual skill. Not a good attitude, not effort. A specific, measurable ability to absorb something new faster than the people around me. Getting fired is what

showed it to me, and once I could name it, I could aim it. The rest of that stretch was pure ground work. Flyers outside Walmart. Door to door. Cold calling. Email marketing. Standing in plazas handing out paper to strangers. That is where I learned to sell, and I learned it because nobody was going to sell for me. At the same time I was teaching the basics of computers to the Latino community through FISH, a Catholic program, and still working part-time at the Calling Teen Center. Emails, what to click, what not to click, how the internet actually worked. I did not call it security awareness then. That is exactly what it was.

Compu-Fix 2u. A folding cart, a banner, and a phone number. CompuFix Sacramento, then Utah, then finally mine.

CHAPTER 12

The Cellphone Fix

March 2014 to September 2021. CTO of a company nobody would have called a tech company.

The repair work turned into a real company through a message. This was when MySpace was still what everybody used, and Erick Arauza reached out to me on it with a proposition. There was already a connection neither of us knew about: I had fixed a computer for his cousin without knowing whose it was.

The proposition was simple: open a store together. As the business grew, his brother Pancho joined us. We built The Cellphone Fix, Computer and Cellphone Repairs into one of the most recognized repair shops in the Sacramento area, three locations at its peak, with customers driving hours to use it. I kept running mobile services the whole time.

It worked because we were opposites, not mirrors. Erick ran sales. He taught me almost everything I know now about sales, communication, and getting comfortable outside my own comfort zone, and I have never stopped giving him credit for that. The lesson underneath it, that the right partner covers the half of the business you are worst at, still shows up in everything I teach.

I had the other half. I was CTO of that company, and I want to be specific about what that actually meant, because people hear a title and picture a corner office. It meant all of the computer infrastructure. The applications. Cybersecurity. Risk. I was technical and I was implementation, which is to say I designed it and then I was the one who built it, because there was nobody else to hand it to.

That is the part of my resume most people miss when they see a repair shop. I was doing infrastructure, application management, security, and risk years before I had a certification or a vocabulary for any of it. It was not a job title somebody granted me. It was work the business needed and I was the only one who could do it.

March 2014 to September 2021. Seven and a half years, and I learned the trade behind the trade. Marketing, content creation, video production, website development, community engagement. It is the exact foundation everything I build now stands on. We had a third location in Folsom for a while

that we ended up closing. Not because business was bad, but because of a contract we should have understood better going in. Along the way I failed more times than I can count and got taken advantage of at least once badly enough that it pushed me straight toward governance and compliance work. I do not tell that part as a warning. I tell it as a curriculum.

Two locations are still open and profitable today. I eventually stepped away, not because the business was failing, but because I felt the pull toward something new. The Cellphone Fix, Computer & Cellphone Repair. The team that built one of Sacramento's most recognized repair shops.

CHAPTER 13

I Just Have To Wake Up

Four words, in 2016, from the worst year I had.

Rojelio died in 2016. I was twenty-six years old, already running The Cellphone Fix, already what most people would have called successful.

That is the part I did not expect. I thought I had already made it out. I had a business with my name attached, locations, a team, customers driving hours to reach us. And losing him still took the floor out from under me, because I lost the one person who had reached out and taken my hand back when there was nothing to be proud of yet. For a long time afterward I was afraid to let anyone else get that close, in case I lost them too. I drifted. I got quiet. I was never a bad kid. I was just around bad people once, and then later I was a man with a real business and no idea what it was actually for. That is the honest version of success nobody puts on a poster. The revenue was fine. The shops were open. And I was standing in the middle of all of it wondering what the point of any of it was.

Until one ordinary day, I was simply done being stuck. And the sentence that went through my head was not a slogan somebody wrote for me. It was four words.

I just have to wake up.

That is it. That is the moment. Everything I have built since carries that name because of him.

I have thought many times about the fact that his shop was RNS Computer Repairs and mine became CompuFix, and then Compu-Fix 2u, and then The Cellphone Fix.

Different names, same trade, one family. He saw the beginning of it and not the rest. Everything with the Wake Up X name on it was built after 2016. After him. That is not a coincidence, and I have stopped pretending it is. The business existed before. The mission did not.

So I will say it here, in writing, where it lasts: Rojelio, I got out. I got into my business. I did not stop.

CHAPTER 14

The Book That Turned the Volume Up

Multiply the goal by ten and watch what you become willing to attempt.

Waking up gave me a decision. What turned that decision into a direction was a book. Erick put Grant Cardone's The 10X Rule in my hands, and I took one idea from it. Whatever goal you set, multiply it by ten. Not because you will always reach it, but because aiming ten times higher changes what you are willing to attempt in the first place.

Set out to make a hundred dollars a day and the instinct is to stop at a hundred. Aim far past it, and even falling short leaves you somewhere you would never otherwise have stood.

It did something else too. It took the isolation, the language barrier, and the years of being told I could not learn, and it turned all of it from something I was hiding into fuel. Almost nothing I read in those years was about cybersecurity, and that was on purpose. The technical material is everywhere. The books on resilience, habits, sales, marketing, leadership, money, negotiation, and influence are the part almost nobody in this field reads. Closing that gap is the reason the library I built later exists at all.

CHAPTER 15

The Corporate Classroom

I left my own business to learn what it could not teach me alone.

I went back into the corporate world deliberately at the entry level. Field engineering for Dell, HP, and Lenovo. Imaging, printers, driving from site to site. Starting at the bottom again was not a setback. It was a strategy, and I want to be honest that it was also frightening. I had been out of corporate so long I assumed I was outdated. I knew hardware. I did not know networking. I was scared of Excel.

What I did have was the business half almost nobody at that level had. Selling, communicating, understanding an agreement before signing it. That is what actually moved me through rooms.

Somewhere in there a company decided not to pay me for work I had already done. That is the whole origin of my interest in governance and compliance. Not a class. Getting stiffed, and realizing that if I did not learn my own legal footing, contract work would keep happening to me instead of for me.

I came back to corporate in June 2022 as a Dell enterprise deployment lead on contract. Six months. Then another contract, an enterprise deployment technician and IT operation lead in Sacramento, another six months. Then in June 2023 it finally turned into a full-time role at a large healthcare organization, working overnight operations lead in Roseville. In February 2025 that became an operations manager and solution architect role in Yuba City.

Two contracts and a year to convert. I am putting the dates in because people see a title and assume it arrived. It did not. I got there on referrals I never asked for. People vouched for me, which is its own quiet lesson about how the work travels.

At that healthcare organization I was later selected as one of five enterprise ambassadors out of more than fifteen thousand employees. I mention it because of the shape of it. I was told as a kid that I could not learn. Fifteen thousand people, and they picked five.

That is where I met Nate. We were both contractors, and I could see something in him he could not see in himself yet. So I kept pushing. I handed him everything I had figured out about follow-up, communication, and positioning yourself so the decision is easy for the person making it. We both got hired full-time.

That is the part I am proudest of, and it is the reason the whole platform exists. What I gave Nate was not a favor. It was a repeatable system, and it has since helped more than fifteen people get hired.

At that new position I met Igor. On paper I was the tech two and the lead, and Igor was the tech one under me. I never once ran it that way in my head. I saw a partnership, and I would tell anyone starting out to do the same. Some of the smartest people you will ever work with sit below you on an org chart, and some sit above you and have nothing to teach you. The title tells you almost nothing.

Here is the specific thing Igor gave me, and it is smaller than people expect. Verbiage. I already knew the work. I had been doing the substance of it for years, under different names, in shops and businesses and other people's buildings. What I did not have was the vocabulary the industry uses for it. Igor handed me the proper wording for things I could already do.

That sounds minor. It is not. There is a whole population of capable people locked out of this field for no reason except that nobody taught them what the industry calls the thing they already know how to do. I was one of them, and I have never forgotten how thin the barrier turned out to be.

Igor is from Ukraine. I am Mexican American. Different countries, same story underneath, and he was the one who pointed that out. We traded. I gave him sales, communication, interviewing, system administration, and the social-engineering side of dealing with people. He gave me the language.

He also did something nobody else had managed. He got me to stop being stubborn about credentials. I had spent years saying I did not need a diploma or a certification because I had already built businesses without them. He did not argue with that. He just said I would need it later, that it was one piece of paper, that I already had everything else. Then he pointed out I was giving him advice I refused to take myself. So I took it. I finished my high school diploma in about three months, because it turned out I had most of the credits sitting there the whole time. I told almost nobody. Then I went after Security+.

Igor went and got his certification and his bachelor's, and I handed him every connection I had. He is a cybersecurity analyst now. That role could have been mine, and the only thing standing between me and it was credentials. I am not bitter about that. It is the clearest proof I have that the paper is worth getting, and watching him pass me is still one of the proudest things in my career.

I want to see somebody get better than me. I want to see somebody do better than me. That's the whole mission.

It is also why titles carry so little weight with me. A tech one can outrun a tech two. Somebody with no certifications can out-think somebody holding ten. I am deliberately open about my own credentials, because of how many people those credentials would have disqualified. No bachelor's. No master's. For a long time, not even a high school diploma, and I built and ran real businesses anyway.

For a long time I resisted the idea that I needed any formal schooling to make it in this field. But the deeper I got into cybersecurity, the clearer it became that a bachelor's and eventually a master's are going to matter for where this is headed. I have CompTIA Security+ so far, with more certifications in progress, and I have stopped pretending the degree question is settled. My point is not that school does not matter. It is that your title does not decide your ceiling.

What actually pulled me into this field was not a class. It was watching how many people around me were getting manipulated by scams and phishing, and wanting to understand and stop that. I had always been fascinated by the hacker mindset, the psychology of why people do what they do online, and the social-engineering side of all of it. Cybersecurity turned out to be the one place where my IT skills, my sales instincts, my communication, and that psychology all fit together instead of competing with each other.

THE THROUGHLINE

Why cybersecurity was the only field that could hold all of it.

CHAPTER 16

Nobody Handed Me the Opportunity

Build in public long enough and the room starts to recognize you before you introduce yourself.

I left The Cellphone Fix in September 2021 and founded Wake Up X Solutions the same month. There was no gap year, no savings runway, no plan B. One thing closed and the next one opened on the same page of the calendar.

The marketing work was real work, not a side project. I ran digital marketing for The Cellphone Fix and for Maya Traditional Mexican Restaurant, and moved online marketing ROI up 46 percent. Campaigns, email, social, website optimization. Danny Maya did something else for me first, and it is the reason there is a podcast at all. When I started getting decent at marketing, he was already doing it, already talking in front of a camera for his restaurant. Watching him do that is part of what pulled me toward marketing. I was the analytical one behind the camera. He was the front-end person who knew how to communicate.

He kept telling me it was time for me to be on the other side of the lens. You have a great personality. You are humble. You are realistic. You do not lie to people. I kept saying I do not know,

man. He pushed and pushed until I finally broke and did it. It was not a good video. But I did it, and the thought afterward was just: keep pushing yourself. Danny Maya is worth explaining, because it is the clearest example of how everything in my life actually gets built. He co-founded Maya Traditional Mexican Restaurant and Taqueria Maya, and he is a partner in Salt and Water. He was a close friend first. The business came second. I am now director of marketing across all of it. Nobody put out a job posting. A friendship turned into work because both people kept showing up for each other over years.

That is the whole pattern. I have never once gotten anything important through an application form. Before that I ran a marketing company called The Wake Up Agency. Working with business clients, I kept running into the same gap over and over. Almost nobody understood how exposed they actually were, or what compliance and

cybersecurity even meant for them. That pattern is a big part of what pulled me fully into the field.

The podcast started almost as a joke. An experiment I half-expected to fail. Everybody told me the same thing. There are already a thousand podcasts, why would you start another one? I did not do it for anybody else at first. I did it because I could not get into the cybersecurity industry through the normal doors, and nobody was handing me mentors, so I decided to go find my own and offer something in return that was not money. Instead of paying a mentor for their time, I gave them a platform. Exposure, promotion, a recording they could share.

The first guest was Gustavo Mastroianni. He had been watching my LinkedIn for a while, and what caught his attention was not a certification or a title. It was an old marketing video from my business days. He was impressed by work I had done years earlier for a repair shop.

So I asked him straight out. I am planning to start a podcast, would you be interested? He said yes, let us do it. No pitch deck, no audience, no proof it would work. One person willing to bet on somebody who had built something visible.

It worked better than I expected. People started following the show, asking how to break into cybersecurity, asking about the guests. That is when I realized this could be bigger than something just for me. It could be something I built for everybody else standing where I used to stand. That is the reason it grew into a real brand: a website, YouTube, Spotify, Apple Podcasts, and eventually e-books and articles on top of the show itself. I did not get a referral into this industry, and nobody handed me a career. Every job, every client, every certification came from building something visible enough that people could find it, and then showing up in person to back it up. Visibility never replaced the work. It is what let the work get seen. Those are different things, and people confuse them constantly.

There is one moment I point to when somebody asks whether it actually works. A project post I had put out ended up on a screen at a national conference. A stranger walked up to me pointing at it like she already knew who I was. She did not, not really. The post had done that work in advance, before I ever walked into the room. That is the whole argument I make to anyone starting out. Build it where people can find it, and keep building it after it stops feeling impressive. Most people quit at exactly the point where it starts compounding.

Network With Efren Salazar. The podcast built to trade exposure for mentorship.

CHAPTER 17

ISACA and the Community

I never stopped showing up. I just did it with a larger purpose.

The pattern that started at the Calling Teen Center never really changed. Walk in, volunteer, be useful, say yes.

That is exactly how I found ISACA Sacramento. I came in as a volunteer. Not a board member, not a speaker, not anybody with a title. I set up rooms. I broke down rooms. I worked the check-in table and talked to whoever walked through the door, and I asked for nothing in return, because I was not there to collect anything. I was there because that is what I do when I want to understand a room.

Volunteering is the most underrated move in any industry, and I will keep saying it until people get tired of hearing it from me. Nobody screens volunteers. Nobody asks a volunteer for a degree, a certification, or ten years of experience. You just show up, do the unglamorous part, and suddenly you are inside the room you were trying to get invited to.

That volunteering turned into Chair of Communications, and then into a partnership with the chapter's Government and Regulatory team, with plans to run for a leadership seat again.

Then I did something that surprised a lot of people. I ran for Director of Communications at ISACA Sacramento against an accomplished CISO with more than twenty years of executive leadership and a reputation that spoke for itself. On one side, all the titles. On the other side, a technician, an entrepreneur, a community builder, and a podcast host who had started as a volunteer and worked his way into the room.

What set my campaign apart was simple. I did not just talk about building a communications platform. I had already built one, with my own hands, out of real conversations with CISOs, CTOs, founders, students, and people just starting out. My vision was just as clear: connect ISACA Sacramento to the Bay Area and Southern California into one statewide network, bring in students from local universities, and give members more visibility for the work they do.

Out of 46 votes, I earned 43.5 percent.

I did not win the seat, and I am completely at peace with that. A regular tech professional stepped into the arena with a seasoned executive and walked away with nearly half the vote. That is proof that hard work, consistency, relationships, and service still matter more than a line on a resume.

I walked away with something better than a title, too. Ken Kerrick, who won the role, turned out to have a lot in common with me. Both of us from immigrant working families. What started as a race became a friendship, and I have already invited him onto the show to tell his own origin story. We were never there to compete. We were there to learn from each other and grow the community. Different paths, same mission. I may have been the lower-level tech in the room, but I was already

thinking like a leader. Every one of those doors opened because I said yes before I felt ready. An ISACA REIGNITE conference. Showing up, badge on, talking to whoever walked through the door. The ISACA Sacramento Director of Communications race. 43.5 percent against a twenty-year CISO, and a friendship on the other side of it.

CHAPTER 18

The Rebuild Nobody Saw

100 pounds in 7 months, fixed the same way I fix everything else.

Every framework I teach assumes you have the energy, focus, and stamina to actually use it. That assumption is not free, and for years I was not holding up my end of it. I was running a business, working full-time in IT, and building a podcast, all while carrying weight that was slowly taking something from all three. Leveling up your title means nothing if you do not level up the body and mind carrying you through the climb. I did not fix it with a program I bought. I fixed it the same way I fix a broken laptop or a failing business process. I got honest about the actual problem, built a structure I could repeat, and stopped trying to do it completely alone. Training naturally, with no shortcuts, I lost 100 pounds in seven months.

I will be straight with you about the rest of it. I gained some of it back. I am telling you that on purpose, because the number was never the point and I am not going to sell you a before-and-after photo. What those seven months proved is that when I put dedication and full effort behind a structure, the structure holds. That is a fact about me I get to keep regardless of what the scale says in any given year.

The rebuild underneath it is the part that transfers. Consistency, not talent, is the thing that actually moves your life. People will doubt you, and there will be seasons that knock you down. A strong enough mindset does not mean you never get knocked down. It means you get back up and keep the structure running anyway.

What started as a personal rebuild became a small community and a brand: Obese to Obsessed, under Efren's Vizion. Not a diet plan. A standing reminder that the same discipline that rebuilds a body rebuilds a career, a business, or a bad year, and that none of it has to be done alone.

Obese to Obsessed, Efren's Vizion. The brand built from the rebuild.

THE FRAMEWORK

The same five steps rebuild a body, a business, or a bad year.

CHAPTER 19

The Mission

Education. Community. Access. For everyone who was told no.

There is a habit in this industry I set out to break. Some people guard what they know. Ask them how they got in and the door quietly closes, because helping you might cost them their edge.

I watched enough of that to reach a conclusion. The field does not need more specialists keeping score. It needs places where the answer is simply handed over. That is the whole gap, and it is not talent, intelligence, or work ethic. It is access. Somebody who already made the jump you are trying to make, willing to tell you honestly how they did it. Most people never get to meet the person who is one step ahead of them.

Our community is big and our people are even bigger, but nobody is there to connect it and bridge it together.

So the platform became more than a podcast. It is a community with real training resources, courses, and a free library. It reaches people at every stage. The elderly learning technology for the first time, young people finding their footing, professionals looking for permission to start over.

I have never fully fit neatly into the American side or the Mexican side of my own identity, and for a long time that felt like something to explain away. I do not anymore. I am Mexican American, Chicano, and proud of both halves of that. Not to show off, but because showing your values and your work openly is different from pretending to be something you are not. Be proud of what you are and proud of what you are doing. My goals from here are specific, and I am writing them down so they can be held to. Cybersecurity consultant. Founder of a Managed Security Service Provider. Community leader bridging Sacramento with the Bay Area and Southern California, running training programs that close the gap between where people are and where this industry needs them to be.

And I still run all of it at once. A full-time role in IT operations and solution engineering, The Wake Up X Solutions, and an ongoing role with ISACA Sacramento, because the mission does not pause for a schedule. If any of this reads as tidy, understand that it is not. It is early mornings, conversations recorded and edited between meetings, and treating the parts of the week that look like networking, mentoring, and community- building as real work, not a hobby squeezed in around a job.

THE GOALS

Written down, because a goal you won't say out loud isn't a goal yet.

THE ECOSYSTEM

Every one of these started with nobody giving permission.

The Point of All of It

I did not tell you this story so you would feel something about me. I told it so you would trust that the advice was earned under real pressure, not borrowed from a classroom. I was told I could not learn. I learned anyway. I was told I was not important enough. I built something that made the question irrelevant. I was never given the opportunity, so I made one, and then spent the rest of my career making them for other people. It does not matter what certification you hold or where you are

starting. If you have hunger, ambition, and the will to keep showing up, you can build something that matters. Do not let anybody's opinion become your ceiling. Control what you can control, and let the rest go.

And if any part of this path sounds like yours, everything I have built exists to make your version faster than mine was.

I want to end on one small thing that proves the whole point. A man named David De los Reyes wrote a testimonial for me, and here is the funny part. We have never met in person. Not once. The entire relationship happened over LinkedIn.

Efren is highly technical and has the ability to go deep on complex subjects, but he also has a way of making technical conversations feel approachable and easy to follow. What stands out most is the initiative he brings to everything he does.

David De los Reyes, via LinkedIn

He then introduced me to his CEO, John, a serial entrepreneur out of Oakland, California. A great person, and a door I would never have found on my own.

That is what I mean about connections and resources. You will always need new people. LinkedIn is one of the biggest connections you should go for, and I say that as somebody whose first podcast guest, first testimonial, and several real opportunities all came from a platform where I had never shaken anyone's hand.

I also want to name Alvin Jue, because this is being written in 2026 and he is in it right now. He believed in me through this whole stretch, and more importantly he is there on the days when it feels like none of it is working. Everybody sees the finished version. Very few people are standing next to you in the middle of it, when the numbers are flat and the doubt is loud, telling you to keep going.

That is what a friend actually is. Not somebody who shows up for the announcement. Somebody who helps you carry it before there is anything to announce. Alvin, thank you. That is the whole model of this book in one relationship: people helping each other out, on purpose, without keeping score.

My parents are still alive, and they have watched the whole arc of it. They knew the kid who was around the wrong people. They know the man who mentors now. They tell me they are surprised, and I understand why. Nothing about the first version of me predicted this one.

If you are reading this from the version of your life that nobody would predict anything good from, that is the part I want you to take. My own parents did not see it coming. It happened anyway.

Securing Tomorrow. Empowering Today. Where To Go From Here This was the story. The method is a separate set of books, and they are free.

Everything you just read is the reason the library exists. I could not find the map when I needed it, so I wrote the one I wish somebody had handed me at nineteen, at twenty- six, and again at

thirty-five. It is Book 0 plus Books 1 through 9, ten books in total, and none of it costs anything.

Let me tell you exactly why I made DECODE. I made it so you would not have to live inside your own head the way I did. I overthought everything for years, because there was no blueprint and no one to ask. So I took what was in my head and put it on the page as a full blueprint, from beginner to executive thinking.

These are the same things I have been telling people privately for years. People call them secrets. They are not secrets, they are just things nobody bothers to say out loud. The people I have shared them with have gone and gotten jobs with them. That is not a claim, that is the record.

What I saw was a gap. No growth, no ladder, no one handing anything down. So instead of guarding it, I wrote it down and gave it away. Let us work together and figure it out.

Start with DECODE: The Wake Up X Playbook, which the library itself lists as Book 0. It is the field guide for going from zero experience to executive thinking with no degree required. From there, take whichever of the nine fits the problem you are actually facing right now.

There is one more book between this one and DECODE, and it is the one I would hand you next. Comfort Is Expensive: The Real Cost of Playing It Safe is the mindset layer. It takes everything you just read about my life and pulls out the part that transfers to yours: what staying somewhere safe actually charges you, the five habits of thought that changed my results, the loop I use to rebuild anything, and the six moves that got me through doors nobody opened for me. It is short on purpose. You can read it in an evening.

The memoir tells you why. Comfort Is Expensive tells you how to think. DECODE tells you what to do. The nine playbooks handle whatever specific wall you hit along the way.

THE FREE LIBRARY

BOOK 0 DECODE: The Wake Up X Playbook. Zero experience to executive thinking. Book 1 Break In. Getting through the door when nobody is holding it open.

Book 2 Build It. Turning a skill into something that exists.

Book 3 Think Like a Cybersecurity Professional.

Book 4 Business First. Why the technology is never the whole job.

Book 5 Lead Without a Title.

Book 6 AI for IT and Cybersecurity.

Book 7 Build Your Brand.

Book 8 The Consultant Mindset.

Book 9 Level Up.

All of it is at wakeupxpodcast.com. Take what works. Build what's yours.

About the Author

Efren L. Salazar is the founder and CEO of Wake Up X Solutions and the host of The Wake Up X Podcast. He partners with ISACA Sacramento's Government and

Regulatory team. He was born to Mexican farm-working parents in the Sacramento area, was placed in ESL and special education classes, and was told by teachers that he could not learn. He has never been formally diagnosed, but he believes he is dyslexic.

He fixed his first computer at thirteen, a donated machine with no working mouse, and ran a repair business out of his front room as a teenager. He spent almost five years at Computers for Kids, first as a volunteer and eventually as their IT Manager. He went on to co-found The Cellphone Fix, serving as Chief Technology Officer across seven years and three locations.

He now works in enterprise IT operations and solution architecture for a large healthcare organization, where he was selected as one of five enterprise ambassadors out of more than fifteen thousand employees. He holds CompTIA Security+ and finished his high school diploma at close to thirty, after having already built and run multiple businesses without it.

He partners with ISACA Sacramento's Government and Regulatory team, and plans to run for a chapter leadership seat again. Through Wake Up X he gives away a library of career and cybersecurity books at no cost, mentors people trying to break into the field, and has helped more than fifteen people get hired. He also runs the fitness brand Obese to Obsessed and serves as director of marketing for Danny Maya's businesses: Maya Traditional Mexican Restaurant, Taqueria Maya, and Salt and Water.

He lives in the Sacramento area. He is still in touch with Kyle Erni, his first business partner.

Securing Tomorrow. Empowering Today. Stay Connected

This book is one part of a bigger mission. If any of this story reached you, here is where to find the rest of it. The podcast, the free library, and the community built for anyone who was ever told no.

Podcast & Free Library: <https://www.wakeupxpodcast.com>

Connect on LinkedIn: <https://www.linkedin.com/in/efrenlsalazar>

YouTube: <https://www.youtube.com/@Efren.wakeupx>

Be a Guest on the Show: <https://wakeupxpodcast.com/guest.html>

Acknowledgments

THE PEOPLE WHO MADE THIS POSSIBLE

None of Wake Up X happened alone, including I Just Have To Wake Up. Every book in this library, the podcast, and the free library exist because specific people showed up for a kid nobody expected much from. This is the record of who they are.

THE EARLY YEARS

Miss England, the 6th-grade teacher who stayed after class, unpaid, to teach the fundamentals. Sister Cora and Lewis Bair of Computers for Kids, who handed a teenager a key and a broken machine and let the work speak. Jose Luis Bravo, Angel Martinez, Isaac Murrilo, and Alex De Haro —the friends and cousins who translated the world, gave hope, drove him where he needed to go, and carried him before there was anything to carry him toward.

THE BUSINESS YEARS

Rojelio Salazar, his cousin and first mentor, RNS Computer Repairs —the reason it is called Wake Up X. Kyle Erni, his first business partner. Erick and Pancho Arauza, co-founders of The Cellphone Fix. Danny Maya, of Maya Traditional Mexican Restaurant, Taqueria Maya, and Salt and Water, who pushed him in front of a camera for the first time.

THE CAREER YEARS

Igor, who gave him the vocabulary for work he already knew how to do. Nate, who he mentored into a full-time role alongside his own. Ken Kerrick, opponent turned friend in the ISACA Sacramento election. Alvin Jue and David De los Reyes, who believed in the middle of it, not just at the finish.

THE PODCAST

Every guest who trusted an unproven show with their story, including Gustavo Mastroianni, Cathy Stephenson, Greg McCord, Aaron Portnoy, Sana Talwar, Jorge Avila, Anil Reddy, Devin Prasad, Larry Switzer, Louis Kemp, Mohana Balakrishnan, Eddie Lopez, and Michael Fierros —full episodes at wakeupxpodcast.com.

COMMUNITY PARTNERS

ISACA Sacramento and ISC2 Sacramento, The Cellphone Fix, A.N.U.L Technologies, AISOC, AspireByte, and PodGrabber —the organizations building alongside this movement.

AND

His parents, who arrived with nothing, worked since childhood, and gave away whatever was left over. Nothing about the first version of this story predicted the version you are holding now.

Continue the Journey

EVERY BOOK LEADS SOMEWHERE ELSE

I Just Have To Wake Up is one part of a 12- book system . If it helped , the next step is one page away —take whichever layer matches where you are standing .

THE LIBRARY

Layer 1— The Story

I Just Have To Wake Up —why any of this exists.

Layer 2 — The Mindset

Comfort Is Expensive —the real cost of playing it safe.

Layer 3 — The Blueprint

DECODE The Wake Up X Playbook —zero experience to executive thinking . Book 0.

Layer 4 — The Playbooks

1 Break In · 2 Build It · 3 Think Like a Cybersecurity Professional · 4 Business First · 5 Lead Without a Title · 6 AI for IT & Cybersecurity · 7 Build Your Brand · 8 The Consultant Mindset · 9 Level Up

FIND US EVERYWHERE

Podcast & free library: wakeupxpodcast.com · Spotify: open.spotify.com/show/28GZnRBrIV2efHOhZdiftt · Apple Podcasts: podcasts.apple.com/us/podcast/id1792821643 · YouTube: youtube.com/@Efren.wakeupx · LinkedIn: linkedin.com/in/efrenIsalazar · Substack: substack.com/@wakeupxefren

IN GOOD COMPANY

ISACA Sacramento, ISC2 Sacramento, The Cellphone Fix, A.N.U.L Technologies, AISOC, AspireByte, and PodGrabber —the partners and collaborators building alongside the movement. See the full list at wakeupxpodcast.com/community.html.

Get DECODE free, apply to be a guest, or read the story behind the movement —all of it starts at wakeupxpodcast.com.

WAKE UP X

SOLUTIONS



Efren L. Salazar

Founder & Host, The Wake Up X Solutions

Government & Regulatory Partner, ISACA Sacramento · CompTIA Security+

STAY CONNECTED

Podcast & Blueprint Library:	wakeupxpodcast.com
LinkedIn:	linkedin.com/in/efrenlsalazar
YouTube:	youtube.com/@Efren.wakeupx
Be a Guest:	wakeupxpodcast.com/guest