SAMPLE

Building Tomorrow's Cybersecurity Specialists

The Assumption's Keeper in the Machine Era

Aiman JoeHanz

Free sample: Introduction & Chapter 1

putrabooks.github.io

Buy the full book:

Carousell carousell.sg/p/building-tomorrow-s-cybersecurity-specialists-ebook-pdf-1471384847

Ko-fi

Payhip

Contents

Introduction

Chapter 1 — The Assumption

Chapter 2 — The Territories

Chapter 3 — The Path

Chapter 4 — The Working Day

Chapter 5 — The Assumption

Chapter 6 — The Burnout

Chapter 7 — The Conversation

Chapter 8 — The Instruments

Chapter 9 — The Year Whole

Chapter 10 — The Long Path

Introduction

Security is the one trade in this series whose failures are always public. A bridge that holds is invisible; a system that is not breached is invisible too, and the practitioner who kept it that way has no photograph to show. The visible events in this trade are the ones where something got through, which means the record of the work is almost entirely a record of its absences.

That makes the trade unusually hard to write about honestly. The books that do exist tend either toward the technical, where the reader learns the mechanics of an attack but not the judgment of the person defending against it, or toward the alarming, where the reader learns to be frightened but not to be useful. This book takes a third position. It assumes the reader is going to do this work, or is already doing it, and wants to know what distinguishes a practitioner worth trusting from one who is merely credentialed.

The distinction is not the toolkit. Toolkits are hireable, teachable, and obsolete in a season. The distinction is a habit of mind, and this book names it: the security specialist is the keeper of the assumption. Every system rests on beliefs about who can do what, from where, and under which conditions. Most of those beliefs are unstated. Some of them are false. The work is to find the false ones before somebody else does, and to say so in a form that gets the thing fixed.

The book follows the structure of the volumes beside it. An introduction to the trade and the two folklores that damage it. The territories where the work is done. The path into it. The working day. A chapter on the craft itself. Then the parts nobody trains for: the burnout, the conversation, the instruments, one composite year, and the long path. It names no product and no vendor, because a book that lists tools has a shelf life of about eighteen months and the judgment it describes should last a career.

One warning before you begin. This book will not teach you to hack. It will teach you to think about what you are defending, which is the part that is hard to buy and impossible to fake. A practitioner who has the thinking and not the tools will be useful within a month. A practitioner with the tools and not the thinking will be dangerous for years.

Who This Book Is For

This book is written for people who are deciding something, and it will be of least use to anyone who wants a procedure.

It is written first for the practitioner in the fourth or fifth year, who can already

do the work and has started to suspect that the work is not the whole job. Their technical reading is adequate. What they lack is a way of thinking about the estate as a set of assumptions with owners and dates, and about themselves as the person who holds those assumptions steady against an organisation that would rather not discuss them.

It is written second for the manager of practitioners who has never done the work. They inherit a function whose value is invisible and whose failures are public, and they are asked to make resourcing decisions with no language for the trade. The chapters on the working day, the burnout and the conversation are written with that reader in mind, because most failures of this function in real organisations are managerial rather than technical.

It is written third for the person considering the trade, who deserves an honest account rather than a recruitment pitch. The work is interesting. The hours are sometimes unreasonable. The trade attracts people who care more than their organisations do, and the usual reward for competence is more work of the same kind. That is worth knowing before spending five years finding out.

What the book does not contain is a checklist, deliberately. A checklist answers a question somebody else already asked. This trade is the practice of working out which question is being asked, and that cannot be reduced to an ordered sequence. The three commitments introduced in the first chapter are the closest thing to a method the book offers, and they are commitments rather than steps because they have to be applied to situations nobody has seen.

Chapter 1 — The Assumption

What the Trade Actually Keeps

Ask a security specialist what they do and you will get a job title back. Ask what they protect and you will get a list of systems. Ask what they are actually responsible for and the answer takes longer, because the honest answer is not a thing at all. It is a set of beliefs.

Every system runs on a stack of assumptions that nobody wrote down. That the person with the administrator account is who the account says they are. That the service on the internal network is not reachable from outside. That the library everyone depends on was reviewed by somebody. That the backup restores. That the vendor's update channel is not the vendor's own weakest link. That the person who left in March no longer has a working credential. Each assumption is doing structural work, holding up something real, and each one is invisible until it stops being true.

The trade exists because some of these assumptions are false right now. Not hypothetically, not in the abstract, but at this moment in the system you are responsible for. The practitioner's value is not that they own a scanner or hold a certification. It is that they have developed the habit of asking which load-bearing belief is quietly wrong, and the discipline to go and check rather than assume the answer is fine.

This is why the trade resists being reduced to a checklist. A checklist is a list of assumptions somebody else already tested. It is useful, and it belongs in the work, but it cannot be the whole of it, because the adversary is not working from your checklist. The adversary is looking for the belief you did not know you were relying on. The keeper's job is to be looking for it too, and to be looking first.

Two Folklores

Two stories about this trade circulate widely, and both of them make the work harder for the people actually doing it.

The first is the folk tale of the gatekeeper. In this story the security specialist is the person who says no: no to the feature, no to the vendor, no to the deadline, no to the exception. The role is understood as friction personified, a department whose product is refusal, and whose success is measured by how much of the organisation's ambition it managed to block. This tale is popular because it is partly earned. There are practitioners who behave this way, and there are organisations that reward it, because saying no is the only move that

carries no personal risk. If you refuse and nothing breaks, you look prudent. If you allow and something breaks, you look negligent. The incentive structure pushes toward refusal, and the folklore grows from that soil.

But the gatekeeper is a failed practitioner, not an archetype. Refusal is the cheapest available control and the least durable. A control that exists only as a person saying no is a control that disappears the moment that person is on leave, changes team, or is overruled by somebody more senior. Worse, refusal teaches the organisation to route around security rather than through it, and a team that has learned to route around you will do so silently and permanently. The practitioner worth keeping is the one who makes the safe path the easy path, and who spends their refusal budget on the two or three decisions that genuinely matter.

The second folklore is the tool buyer. In this story security is a procurement exercise: enough products, correctly layered, produce safety. The story is attractive because it is legible to a finance function, it produces a dashboard, and it can be delegated to somebody who does not need to understand the systems. It is also the story that has produced the most expensive failures in the field, because a stack of tools watching a system nobody understands is a stack of tools producing alerts nobody reads. Buying detection does not create understanding. It creates the appearance of it, which is worse, because the appearance suppresses the instinct to go and look.

Both folklores share a root error. They locate security in an attitude or in a purchase, when it actually lives in a set of beliefs about the system that are either true or false, and in the discipline of finding out which.

Deciding What Is Held

The specialist who understands what they are keeping becomes specific very quickly. They can tell you which assumptions in this estate are load-bearing, which are already broken, and which nobody has ever tested. They can point at the identity provider and say that everything above it is a decoration. They can point at the backup restore and say it has never been proven. They can point at the third-party integration and say the trust is mutual and undocumented.

That specificity is the whole craft, and it produces a definition that is narrower and more useful than the job description. The security specialist holds three things true.

First, name the adversary before you name the control. A control without a named adversary is a decoration. Authenticating users is not a control; authenticating users against a specific attempt to reuse stolen passwords is a control, and it will look different from the control that defends against a specific attempt to escalate privileges from a compromised service account. Until the adversary is named, the control cannot be evaluated, only purchased.

Second, assume the perimeter has already been crossed, and design for what happens after. This is not defeatism. It is arithmetic. Every estate of any size contains at least one path in, and the number of paths grows with every integration, every remote worker, every vendor connection, and every forgotten test environment left running. A design whose safety depends on nobody getting in is a design that has already failed. A design that assumes somebody is in, and asks what they can then reach, is a design that can be defended.

Third, make the safe path the easy path, because a control people route around is not a control. This is the clause that separates the practitioner from the auditor. It is possible to be entirely correct and entirely ineffective, and the difference is always whether the people doing the work found the correct behaviour easier than the incorrect one. If the compliant route takes four approvals and the non-compliant route takes one, you have built a training programme for non-compliance.

Where the Difficulty Actually Sits

The reason this trade is hard is not that the technology is obscure. Much of it is well documented and the fundamentals have been stable for decades. The difficulty is that the work requires holding two incompatible frames at once.

The first frame is probabilistic. The estate is large, the adversaries are many, and no defence is complete. Everything you do is a reduction of likelihood, never a guarantee, and the honest practitioner knows that any given day's work might be irrelevant to the breach that actually happens. This frame is uncomfortable to inhabit and easy to abandon, which is why so many practitioners drift toward either fatalism or its opposite, a brittle confidence in the current stack.

The second frame is particular. At the moment of an incident, one specific thing happened, in one specific place, for one specific reason, and the general statistics are worth nothing. Somebody's credential was reused, or a library

was compromised, or an engineer ran a command with the wrong context, and the response has to be exact rather than probabilistic. The practitioner has to move from thinking about populations to thinking about this instance, quickly, while being watched.

Holding both frames is the skill. The probabilistic frame without the particular produces paralysis and paperwork. The particular without the probabilistic produces whack-a-mole, where every incident is treated as unique and nothing is ever learned. The keeper of the assumption moves between them deliberately, using the statistics to decide where to look and the instance to decide what to do.

What Makes This Different From Its Neighbours

The trade is often confused with three neighbours, and the confusion is worth clearing because it changes what a practitioner practices.

It is not systems administration, though it depends on it. The administrator keeps the system running. The specialist asks what the running system allows somebody to do who should not be allowed to do it. The skills overlap heavily and the questions do not. An administrator who never asks the second question can be excellent at their job and leave the estate wide open.

It is not software engineering, though increasingly it is done in code. The engineer builds the thing. The specialist asks what the thing will do under an input its author did not imagine. This is why application security is its own discipline with its own instincts, and why an excellent engineer can be a poor reviewer of their own work in exactly the way that matters.

It is not compliance, though it is often reported as if it were. Compliance asks whether a documented control exists and operates as described. Security asks whether the control would actually stop the named adversary. These questions correlate weakly. An estate can be fully certified and entirely undefended, and the practitioners who discovered this the hard way are the reason the field now distinguishes the two in almost every serious conversation.

The last distinction is the one that determines whether a career in this trade goes well. The compliance question can be answered from a desk, with documents, by somebody who has never operated the system. The security question cannot. It requires touching the estate, forming a belief about how it actually behaves rather than how it is described, and being willing to be told that the belief was wrong.

The Habit Worth Building

If a reader takes one thing from this chapter, it should be a habit rather than a fact. When you encounter any claim about a system, ask which things would need to hold for the claim to stand, and then ask who last checked.

The claim might be that only the finance team can approve payments. What would have to be true? That the approval group is current, that the identity provider enforces it, that no service account can impersonate a member, that the payment system reads the group rather than a cached copy, that the cache invalidates, and that nobody outside the group has been granted the role directly and forgotten. Six beliefs, and most organisations have checked one.

This habit is not cynicism. It is the opposite. Cynicism assumes everything is broken and stops looking. The habit assumes nothing is known until it is checked, which is a far more productive posture, because it produces a list of things to verify rather than a mood. The practitioner who works this way becomes, over a few years, the person in the room who is quietly right about how the estate behaves, and that reputation is the only durable asset the trade offers.

What the Trade Is Not

The boundaries of this trade are badly drawn in public, and the confusion costs people years, so it is worth being blunt about what sits next to this work without being it.

A system administrator keeps a fleet running. Their knowledge of the estate is often deeper than any specialist's, and much of what a specialist needs they already have. What they are not asked to do is decide which assumptions the estate is allowed to rest on, or to argue for a budget against a risk nobody can see. The administrator fixes; the specialist asks why the thing became fixable. The distinction is not status, and a specialist who behaves as if it were will find very quickly that the administrator knows where every undocumented dependency lives.

An auditor examines evidence. There is real craft in it and real value, and a great many compliance roles involve serious technical judgment. But the auditor's job is to establish what is documented and true, against a standard someone else wrote. The specialist's job is to work out what is true when nobody wrote the standard yet, which is a different and much lonelier activity.

A developer builds features. The best security work in most organisations is done by developers who were never given the title, because they thought about who could reach their endpoint and what would happen if the input were hostile. Specialists who treat developers as the problem rather than the means will spend their careers losing to a distribution channel they cannot reach.

Information technology, in the corporate sense, runs services for staff. Overlap is total in small organisations, where one person does all of it and the title is aspirational. In large ones the confusion is expensive, because the person who owns the laptops and the person who owns the identity provider hold two halves of one question, and neither can answer it alone.

What the specialist holds that none of these hold is the adversary. Every other role in the list models a requirement, a user, a standard, or a failure. This one models a person who wants the system to do something its builders did not intend. That single orientation is what makes the work a trade rather than a duty, and it is what the rest of this chapter is about.

What Happens When an Assumption Breaks

The value of thinking in assumptions becomes visible in what happens afterwards, and the pattern is regular enough to be worth describing.

An assumption breaks quietly first. Somebody leaves, and the access they held stays. A service is migrated, and a network rule that was scoped to the old subnet now covers a wider one. A vendor changes a default, and a bucket becomes readable for a fortnight before anybody notices. None of this is dramatic. Each event is unremarkable in isolation, and each one removes a little of the reason the system was safe.

Then it breaks loudly, and the loudness is disproportionate to the change that caused it. The reason is structural: assumptions are what the other controls are calibrated against. Detection rules are written knowing who normally logs in. Segmentation is designed knowing which hosts talk to which. Rate limits are set knowing expected volumes. When the underlying assumption shifts, every control calibrated to it is now answering a slightly different question, and none of them announces that it has become less useful.

This is why a specialist's habit of listing assumptions is not documentation for its own sake. The list is how the trade notices a control going quietly out of calibration. An estate where nobody can say what the access model assumes is an estate where the alarms are answering last quarter's questions.

The recovery has a shape too. The first thing that returns is not confidence but a shorter list. Practitioners who have been through it tend to emerge with fewer, better-understood assumptions and a habit of dating each one, because an assumption that has no date is one nobody is reviewing. That shortened list is the actual deliverable of the experience, and it is what distinguishes someone who has been through an incident from someone who has read about one.

How to Read a Claim About Safety

A claim that a system is secure is not a fact and it is not quite a lie; it is a summary of assumptions, and learning to unpack one is among the most transferable skills in this trade.

The first thing to ask of any claim is what it quantifies. A statement that a control is in place is a statement about configuration. A statement that a risk has been reduced is a statement about a model. A statement that a certification has been achieved is a statement about an auditor's sampling. These are three different kinds of claim and they are routinely merged into one slide, which is why so many arguments about security end without anyone having been wrong.

The second thing to ask is what the claim is relative to. Every protection is protection against something, and the something is often left implicit because naming it would weaken the claim. A statement that data is encrypted is true and says nothing about whether the key management permits the operator to read the data. A statement that access is least-privilege is true relative to the previous state, which may itself have been generous. The useful question is always against which adversary, and a claim that cannot answer it is a description rather than a protection.

The third thing is the date. Claims are made about a moment and estates move, so a claim with no date is not evidence about the present. This sounds pedantic until you have watched an organisation make a decision in one quarter on the basis of a statement that was accurate in another, and the failure is so ordinary that it accounts for a large share of the findings in any mature review.

The fourth thing is who benefits from the claim being believed. This is not cynicism. A vendor, an internal team arguing for budget, and a practitioner who wants their work recognised all have reasons to state a true thing in a flattering frame. Knowing the interest does not make the claim false; it tells

you which parts were chosen and which were left out.

Practitioners who can perform this unpacking in conversation, gently and without appearing to cross-examine, are the ones whose judgment gets sought. The skill is not scepticism. It is the ability to hold a claim at the right resolution, so that it can be relied upon for exactly what it covers and no further.

The Assumption and the Trade

The trade is the keeper of the assumption. Not the assumption that people are good, or that systems are safe, but the specific load-bearing beliefs that a particular estate rests on, held deliberately enough that somebody notices when one of them stops being true.

Every chapter that follows is an elaboration of that sentence. The territories are the places where the beliefs differ. The path is how the habit gets built. The working day is what checking looks like at the scale of hours. The craft chapter is the mechanics of finding the false belief. And the harder chapters, the burnout and the conversation and the instruments, are about what happens to a person who has to say, repeatedly, that something believed to be true is not, and to do it in a way that produces change rather than defensiveness.

The assumption is what the trade keeps. Everything else is method.

ABOUT THE AUTHOR

Aiman JoeHanz writes the Building Tomorrow's People series. The cybersecurity volume takes as its subject the assumption, the thing every system believes about its own users until the day it is wrong. This book covers the threat model, the patch window, the incident call at two in the morning, the ransom negotiation, the auditor's checklist, and the defender's permanent disadvantage of needing to be right every time. It is written with sympathy for the trade that is only ever praised in retrospect.