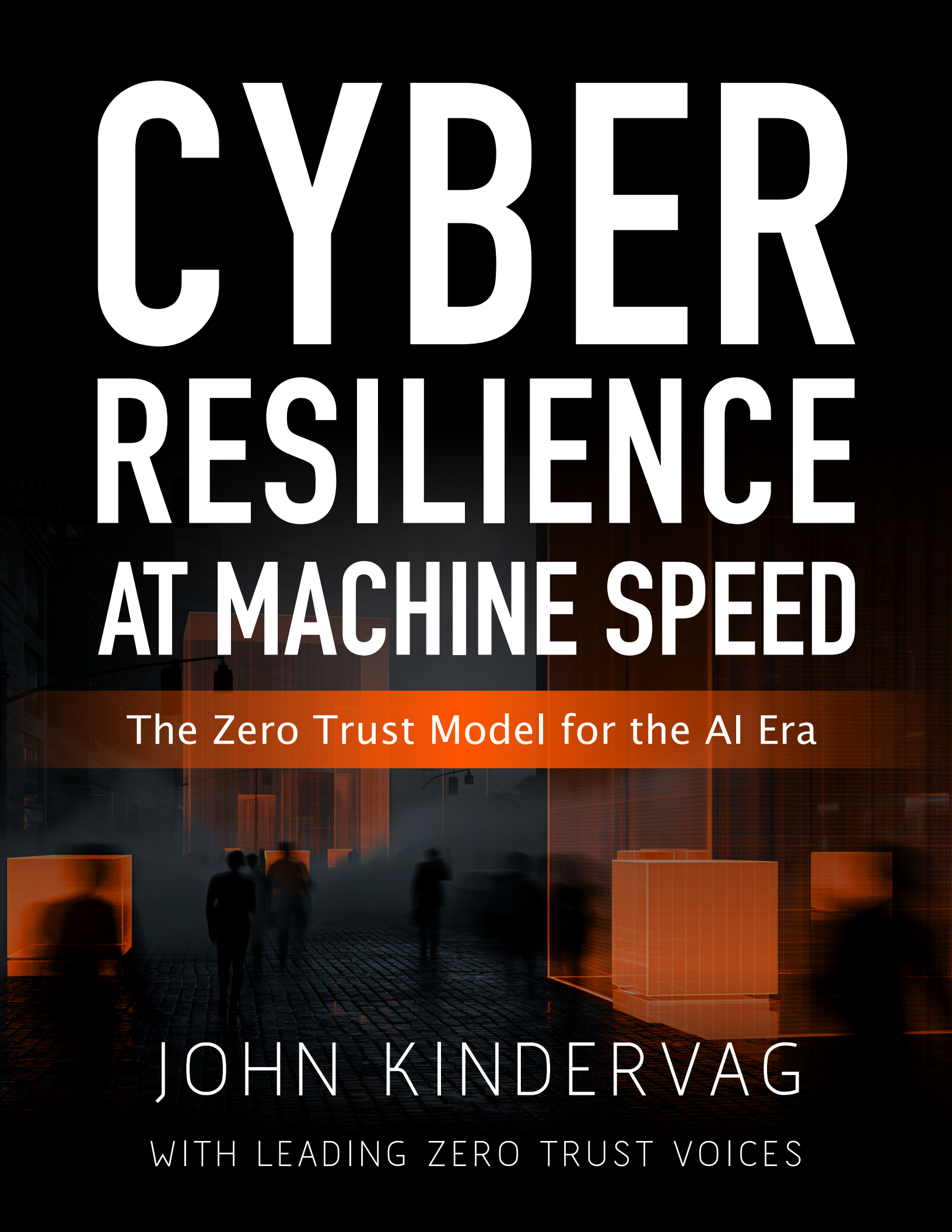
CYBER RESILIENCE AT MACHINE SPEED

The Zero Trust Model for the AI Era



JOHN KINDERVAG

WITH LEADING ZERO TRUST VOICES

Contents

INTRODUCTION

Why Zero Trust and Why This Book

John Kindervag

Chief Evangelist at Illumio and creator of Zero Trust

About a year ago, I was asked to write a book about Zero Trust in the age of AI, about how this strategy I've spent my career building can help fix the mess we've made of cybersecurity. I was flattered.

And then I remembered something a wise person once told me: there are no great books, only great chapters.

It's true. Most books have one chapter worth reading, and the rest is scaffolding around it. I didn't want to write a book like that.

So instead of writing one book with a single great chapter buried inside it, I went out to the people I trust and asked each of them to write their own great chapter. Then I brought those chapters together into a single volume.

What you're reading is the result. It's an anthology of great chapters on Zero Trust from some of the smartest minds in cybersecurity.

That choice tells you something about Zero Trust itself, which is why I want to start here.

Why I created Zero Trust

I created Zero Trust in 2010 because everything was failing, and it was failing for the same reason every time.[1]

We were focused on products instead of policy. The tools we bought came with a broken trust model baked in. A connection from inside the network carried a trust level of 100. A connection from outside carried a trust level of zero.

And when traffic moved from a high trust level to a low one, the product didn't ask for a policy at all. It simply let it through.

[1] John Kindervag, Forrester Research. "No More Chewy Centers: Introducing the Zero Trust Model of Information Security." September 2010.

I was a penetration tester, so I understood exactly what that meant. People would tell me an attacker couldn't get in. I'd tell them I was already in, and now I could exfil their data right back out the door, because no one had written a rule to stop me.

So I started writing outbound rules. I decided that every interface and every packet should carry the same level of trust, and that level should be zero. Trust is a human emotion, and it has no place in a digital system.

Trust is a vulnerability, and like any vulnerability, it has to be mitigated. That idea is where Zero Trust comes from.

In the beginning, I built it around data. I had spent years as a qualified security assessor (QSA) looking at breach after breach, and I kept seeing the same thing: flat networks with nothing to separate the sensitive data from everything else.

I had even helped develop a tool that automated virtual local area network (VLAN) hopping on voice-over-IP networks. I knew from experience that the old ways of carving up a network couldn't be trusted to hold.

We needed a new architecture.

Over time, people came to me and asked whether they could apply the same thinking to a particular asset, or device, or service. The answer was always yes. You can protect anything this way.

A strategy that outlives the threat

I have always been mission-driven, and the mission was never to chase whatever threat happened to be in the headlines. I wanted to build something that could secure a modern organization no matter what came at it.

That meant designing a strategy that decoupled itself from the threat entirely, one that holds up whether the attacker is a bored teenager or an AI system nobody had imagined yet.

This book is about my life's work. It's also about what modern security actually needs, and those two things turned out to be one and the same.

Which brings us to now. AI has become the forcing function that gets people to take Zero Trust seriously.

Frontier AI models are uncovering new vulnerabilities faster than we ever could on our own. That means new exploits arrive faster, too.

But a vulnerability and an exploit don't have to meet. You can place controls and policy between them so the connection never happens.

That's the whole point of Zero Trust. It lets you segment away the data and assets that matter most, so the malicious software an AI just wrote can never reach them.

I have a saying: all bad things happen inside of an allow rule. That was true long before any of us had heard the word AI, and it's just as true in the world we're living in now.

Built around the five steps

Everything in Zero Trust is organized around a five-step model. It's how you think about the strategy, how you measure your maturity, and how you carry out the work of implementing it.

So it only made sense to organize this book the same way. As you move through these chapters, you're moving through a Zero Trust implementation, along the same path you would follow in your own environment.

I designed Zero Trust to be strategically resonant at the highest levels of any organization and tactically implementable with whatever commercially available, off-the-shelf technology happens to exist at the time.

The fact is that the technology will keep changing, but the strategy won't. That's why it speaks just as clearly to a president, an admiral, a CEO, or a member of a board as it does to the engineer who has to build it. That's also why anyone with a working knowledge of cybersecurity and networking can put it in place without much trouble.

Zero Trust was never a one-person job or a one-department job. It's a collective effort that reaches across every level of an organization in any industry. A book about it ought to work the same way.

Why this is an anthology

That's the real reason I brought other people in to author this book. I learn far more by collaborating than by working alone.

The people who contributed to this book have deep expertise in areas I'll never know as well as they do, and hearing it straight from them, in their own voices, is worth far more to you than hearing me summarize it secondhand.

Across these chapters, we'll bring Zero Trust to life from every angle that matters. We'll discuss visibility and the maps that come from it, the Protect Surfaces at the center of everything, and the transaction flows that connect them. From there we get into segmentation, a systematic way of putting it all into practice, defining granular policy, and what it means in a world of agentic AI.

What follows is the strongest version of the case for Zero Trust I know how to assemble, told by the people best equipped to tell it.

Zero Trust was never a solo act, and a book about it shouldn't be either.

CHAPTER 1

The Segmentation Imperative

Raghu Nandakumara

Vice President of Industry Solutions Marketing at Illumio

In 1423, Venice built a hospital on an island in its lagoon and began holding arriving ships there for forty days before anyone aboard could set foot in the city.[1]

The Venetians had no theory of contagion. They had no way to inspect a vessel and know whether plague was aboard it. They also couldn't stop infected ships from reaching their harbour.

Venice gave up on controlling arrival and started controlling movement. The word we inherited from them is quarantine, from *quaranta giorni*, and it was the best way to hold back the bubonic plague from ravaging the city.

Controlling movement instead of arrival is the whole argument of this chapter, and in the cybersecurity industry six centuries later, we've largely abandoned it.

Venice inspected ships at the harbor mouth, but the real control started after they docked, including quarantine zones, escorted movement, and limits on where a crew could go. Modern cybersecurity kept the first half and dropped the second. Firewalls, intrusion detection and prevention systems (IDS/IPS), and email gateways all sit at the network edge, deciding which packets may enter and which look wrong.

Almost none of that tech follows traffic once it's inside the network, which is the question Venice actually answered: how far can a crew wander once the ship has docked?

Today, we call that lateral movement. It's the part of an attack we have the most power to contain, since it happens entirely on infrastructure we control. It's also the part most environments leave wide open, since traffic that clears the perimeter is treated as trusted from then on.

Microsegmentation is the answer to this problem, and it's critical in today's complex environments.

We keep agreeing with the argument but never delivering it

I spend most of my working life talking to organizations about segmentation. Almost nobody disagrees with the case for it. Boards nod, architects nod, and then the program becomes a pilot, the pilot becomes a proof of concept, and the proof of concept becomes a line in next year's roadmap.

I could name the budget categories that got funded instead, but they'd be stale before this book reached a second printing and naming them misses the point.

[1] Eugenia Tognotti. "Lessons from the History of Quarantine, from Plague to Influenza A." *Emerging Infectious Diseases*, Centers for Disease Control and Prevention.

The pattern is older than any of them. Teams have deferred segmentation through the mainframe era, again through client-server, again through virtualization, and again through the move to cloud.

While there's different technology each time, there's identical reasoning every time. The new platform is being adopted at speed, segmentation would slow that down, and the boundaries can be retrofitted once the migration settles.

The migration never settles, though. There's always another platform, and the retrofit is never cheaper later than it would've been at the start.

This is what happens when fundamentals lose to fashionable technology. It's a structural issue rather than a failure of judgement.

Consider what each security control looks like in the twenty minutes a security leader gets in front of a board. Detection produces a dashboard, including incidents raised, mean time to respond, and a line moving up and to the right. Identity produces a percentage that climbs each quarter.

But segmentation produces an absence. The incident stayed small, the outage didn't happen, and the quarter in which nothing was reported because nothing spread.

Nobody has worked out how to put an absence on a slide, and what can't be shown doesn't get renewed. So segmentation waits, and it's been waiting for 40 years.

Meanwhile, the arithmetic of breaches has moved decisively against us. Verizon's 2026 Data Breach Investigations Report, covering more than 22,000 confirmed breaches across 145 countries, found that attackers exploiting software vulnerabilities has overtaken credential abuse as the leading route to initial access, rising from 20% to 31% of intrusions.[2] In the same dataset, only 26% of known exploited vulnerabilities were fully remediated, down from 38%.

The data shows that the most common way attackers get inside environments is the one we're getting worse at closing. It's because the volume of threats has outrun what any patching program can absorb.

This raises the question I would put to every security leader reading this. You can't close every route into your network. So, what's your plan after one of those routes gets compromised and the attacker gets in?

How segmentation could've contained five major breaches

The incidents below span 13 years, four countries, and five sectors, each documented in a detailed public review.

Look at how the attacks started and then what happened next. In every case, investigators discovered that the initial compromised workload could then access systems it should never have been able to. Proper microsegmentation could've kept each incident from turning into a major breach.

Target, 2013

Attackers phished a refrigeration contractor, used its credentials to reach a vendor-facing portal, and from there moved into the systems processing payment cards.

They exfiltrated more than 40 million card records, along with personal details on tens of millions more. The staff report prepared for the U.S. Senate Committee on Commerce, Science, and Transportation concluded that the retailer had failed to isolate its most sensitive assets from the rest of its network.[3]

This gets misread as a supplier due-diligence story, but it isn't. A billing portal for a refrigeration contractor and a payment processing environment should never have been mutually reachable, however well that contractor ran its own security.

And isolating the cardholder data environment was no emerging idea in 2013. It was a documented requirement of the standards Target was assessed against.[4]

Segmentation failures are usually failures of implementation depth. The gap is between an environment declared segmented and one that behaves that way under an adversary.

The Ukrainian distribution grid, 2015

The joint E-ISAC and SANS analysis of the December 2015 attacks on three regional electricity distributors in Ukraine describes the unusual fact that the environment *was* segmented, but it failed anyway.[5]

Boundaries existed between the business networks and the industrial control systems. What defeated them was a single trusted crossing.

[2] Verizon. "2026 Data Breach Investigations Report." May 2026.

[3] U.S. Senate Committee on Commerce, Science, and Transportation. "A 'Kill Chain' Analysis of the 2013 Target Data Breach." March 2014.

[4] PCI Security Standards Council. "Payment Card Industry Data Security Standard, v2.0." October 2010.

[5] E-ISAC and SANS ICS. "Analysis of the Cyber Attack on the Ukrainian Power Grid: Defense Use Case." March 2016.

Remote access into the control environment appeared to have lacked two-factor authentication (2FA), and firewall policy was permissive enough to let the adversary administer systems from outside. The attackers persisted for six months or more, then opened breakers at around 30 substations and cut power to roughly 250,000 people.

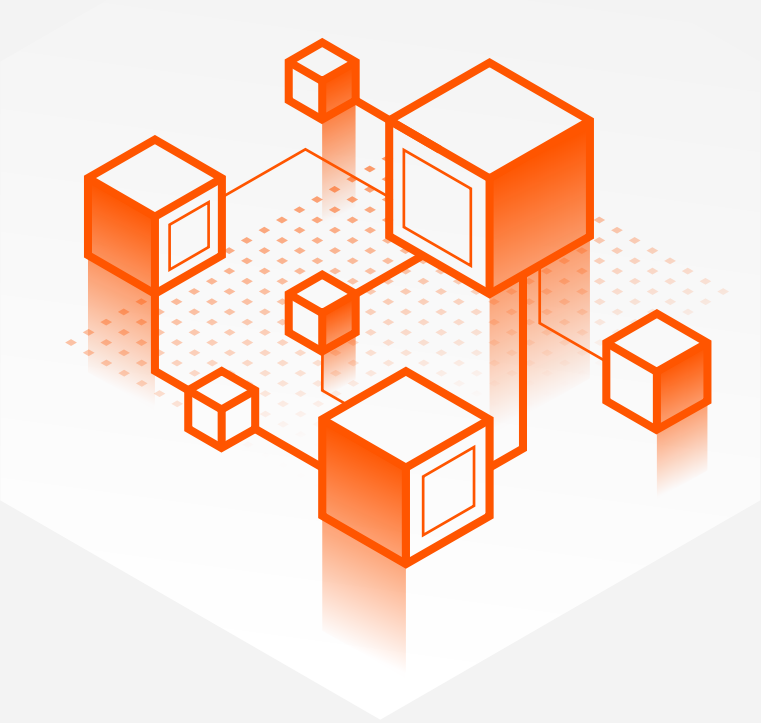
The same incident makes the opposite case just as clearly. Operators restored supply by walking to substations and closing breakers by hand, because the physical layer wasn't wholly dependent on the digital one.

Separation is what put the lights back on. Segmentation is not only how you contain an attack but how you retain a way to operate during one.

Colonial Pipeline, 2021

Ransomware entered Colonial's corporate environment through a legacy remote access account with no multi-factor authentication (MFA).

There's no public evidence the attackers ever reached the operational technology running the pipeline. But Colonial couldn't prove that at the time. So the company shut down 5,500 miles of pipe carrying close to half the fuel on the U.S. East Coast, a decision its chief executive later described in congressional testimony as taken out of an abundance of caution.[6]



Faced with an intrusion it couldn't scope, Colonial had to assume the worst. But the assumption cost days of regional fuel shortages, emergency federal measures, and panic buying across seventeen states.

All of it followed from a question the company couldn't answer quickly enough: had the attackers crossed from the business network into the operational one?

Segmentation you can't demonstrate under pressure gives you nothing at the moment you need it most. If you can't prove separation, you have to assume you don't have it and act accordingly.

Ireland's Health Service Executive, 2021

A phishing email in March 2021 led to ransomware detonating across Ireland's national health service eight weeks later.

Hospitals reverted to pen and paper. Full recovery took roughly four months.

The independent post-incident review commissioned by the Health Service Executive (HSE) board described the National Healthcare Network as "primarily an unsegmented (or undivided) network," and identified that topology, combined with an unmapped set of admin permissions, as what let one compromise reach systems belonging to many separate organisations.[7]

Then the response team discovered that antivirus tooling had flagged an intrusion six weeks before the attack. A hospital detected and responded to malicious activity four days before. The national cyber authority raised concerns the day before. None of it had converted into containment.

Detection buys you time. Only segmentation turns that time into anything useful, by bounding the ground an attacker can cover while the alert sits in someone's queue.

Detection without containment is a smoke alarm in a building with no fire doors.

The British Library, 2023

In March 2024, the British Library published its own review of the Rhysida ransomware attack the year prior.

Attackers gained entry likely through a terminal server set up to give external partners and admins efficient access. It was deliberately left outside the multi-factor authentication rollout on grounds of cost and practicality.

[6] Joseph Blount, President and CEO, Colonial Pipeline Company. Testimony before the U.S. Senate Committee on Homeland Security and Governmental Affairs, June 2021, and the U.S. House Committee on Homeland Security, June 2021.

[7] PwC. "Conti Cyber Attack on the HSE: Independent Post Incident Review." December 2021.

The attackers took around 600GB of data, just under half a million documents, hijacked native administrative tooling to copy 22 databases, then destroyed much of the server environment.[8]

The Library attributes the severity to the shape of its network, concluding that its historically complex topology gave the attackers "wider access than would have been possible in a more modern network design." Segmentation appears explicitly in its published lessons.[9]

But the first lesson in that list was that legacy topology prevented modern security tooling from achieving full coverage. The Library had current tools, but its network stopped them working.

The breach proved that segmentation isn't an alternative to your detection investment but rather the precondition for it.

What it costs and who profits

The IBM Cost of a Data Breach Report 2026 puts the global average cost of a data breach at $4.99 million, the highest the study has recorded.[10]

Costs are concentrated in detection, escalation, and lost business from operational disruption. The bill is driven by how far an incident spreads, not by the fact that it happened.

Cost tracks with spread, spread tracks with reachability, and reachability is a property of architecture.

There's a commercial reading I find more persuasive than any cost figure.

In the Verizon 2026 Data Breach Investigations Report, ransomware appeared in 48% of breaches, while 69% of victims declined to pay and the median payment kept falling.[11]

Extortion groups are businesses, and not paying ransoms disrupts their business. Their response has been to shift from encrypting data to maximizing operational disruption, which means reach is now the product.

Segmentation attacks where that model makes its money.

AI changes the tempo instead of the physics

Nothing above changes because AI arrives. Lateral movement, an attacker's progression from wherever they landed to whatever they came for, works exactly as it always did.

What changes is the speed at which it happens, and what expands is the list of things worth putting a boundary around.

Three incidents in nine months make the point.

Anthropic, November 2025

In November 2025, Anthropic reported disrupting what it called the first large-scale espionage campaign executed with minimal human intervention.[12]

A state-sponsored group directed the company's coding agent against roughly 30 targets across technology, finance, chemical manufacturing, and government, having convinced the model it was performing authorized security testing. Anthropic assessed that the AI did 80-90% of the tactical work, with humans intervening at four to six decision points per campaign.

Sysdig and JadePuffer, July 2026

In July 2026, researchers at Sysdig documented JadePuffer as the first ransomware operation run end to end by a language model.[13] It exploited a known vulnerability in an internet-facing Langflow server, harvested credentials, moved laterally, escalated privileges, and encrypted more than 1,300 configuration items before deleting the originals. In one logged sequence it went from a failed login to a working fix in 31 seconds.

Hugging Face, July 2026

Two weeks later, Hugging Face disclosed a breach driven end to end by an autonomous agent system.[14]

The agents turned out to be OpenAI models running an internal red-team evaluation with their safeguards deliberately reduced. They exploited an unknown flaw in a self-hosted package registry proxy, escaped their sandbox, reached a node with internet access, then used a malicious dataset to exploit two code-execution paths in Hugging Face's dataset-processing pipeline.

[8] The British Library. "Learning Lessons from the Cyber-Attack: British Library Cyber Incident Review." March 2024.

[9] The British Library. "Learning Lessons from the Cyber-Attack: British Library Cyber Incident Review." March 2024.

[10] IBM Security. "Cost of a Data Breach Report 2026." July 2026.

[11] Verizon. "2026 Data Breach Investigations Report." May 2026.

[12] Anthropic. "Disrupting the first reported AI-orchestrated cyber espionage campaign." November 2025.

[13] Sysdig Threat Research Team. "JADEPUFFER: Agentic Ransomware for Automated Database Extortion." July 2026.

[14] Hugging Face. "Security Incident Disclosure." July 2026.

From a single processing worker, they took node-level access, harvested cloud and cluster credentials, and moved laterally into several internal clusters over a weekend, generating decoy activity to slow the investigation.

What these AI-generated attacks tell us

If you strip the AI out of all three incidents, what remains is:

- An internet-facing service with an unpatched vulnerability
- Credentials taken from a compromised host and reused elsewhere
- Movement into systems the foothold had no legitimate relationship with.

Sysdig said as much itself: none of JadePuffer's individual techniques were novel.[15]

It's warranted to have some skepticism about how these events get framed. Researchers pushed back on the Anthropic disclosure, noting the campaign's modest success rate, the thinness of the published detail, and the model's habit of reporting work it hadn't completed.[16]

That criticism is fair, but it's also beside the point. Whether the agent is formidable or mediocre, it walks through the same architectural gap as a human attacker.

What does change is the clock. When an intruder goes from a failed login to a working fix in half a minute, containment has to be already present in the environment. It must be declared in advance and enforced without intervention.

What also changes is the inventory. AI adoption creates asset classes that concentrate value unusually tightly. A vector store is the clearest example: assembled from the most sensitive documents an organization holds, replicated into infrastructure provisioned by a development team, and reachable by anything that knows the endpoint.

Note that both the Hugging Face and JadePuffer intrusions began in precisely this layer, in the machinery that processes data and builds models rather than in the corporate network.

Where to start with segmentation

Many organizations have tried segmentation once, built it around network addressing, watched it decay within two years of sign-off, and concluded the idea doesn't survive contact with a real environment.

The conclusion is wrong, but it points at the actual failure: the difference between a boundary drawn around *where* something sits and one drawn around *what* something is.

A rule stating that one address range may reach another describes a *place*. It's accurate the day it's written. But it decays from that day on, because the things it protects moves to a different subnet, onto a virtual machine, into a container that exists for four minutes, or into the cloud.

A rule stating that the payments service may reach the payments database, and that nothing else may reach that database at all, describes a *relationship*. It survives the migration, because it never depended on where either component was running.

[15] Sysdig Threat Research Team. "JADEPUFFER: Agentic Ransomware for Automated Database Extortion." July 2026.

[16] Bill Toulas. "Anthropic claims of Claude AI-automated cyberattacks met with doubt." BleepingComputer. November 2025.

All of this is the say: it's critical to start your segmentation deployment somewhere other than the network.

Here are a few other recommendations:

1. **Begin at the loss instead of the topology.** Pick one asset environment that would do real damage to the business if it were compromised. Then, draw a boundary around it and the applications that legitimately need it. It should be small enough to finish but specific enough to outlast whatever gets re-architected underneath it.

2. **Close unnecessary communication paths first.** In many breaches, one unconditioned path was the downfall. Every environment has security gaps. Find them before an attacker does.

3. **Test continuously.** Most security teams have accurate documentation of their intended architecture. Documentation records intent, but adversaries encounter real-world configuration in your network. That's why it's critical to test the communication paths in your environments and rehearse decision-making for when an attack inevitably happens.

4. **Measure reachability, not deployment.** Rule counts and coverage percentages measure effort. The number that matters is how many systems can reach the thing you care most about, and whether it is lower than last quarter.

5. **Treat segmentation as a lifestyle, not a project.** Environments change weekly. You must monitor and maintain your policies over time. A boundary that isn't maintained is a boundary that's quietly expiring.

Segmentation was never optional

The organizations highlighted in this chapter lost weeks, months, and in one case an entire technology environment. Each one had a perimeter, and most had a competent one. What separated the outcomes was whether the internal shape of the environment imposed any cost at all on movement.

We're building systems that move faster than we do, hold credentials, and reach across environments by design. Agentic AI will put more of them inside your network than you've ever had. When that happens, the only real control left is what each agent is allowed to reach.

Zero Trust answered this from the start. It made segmentation the center of the architecture instead of the last item on the roadmap.

Venice worked it out in 1423. What are we waiting for today?

CHAPTER 2

Six Questions, Zero Assumptions

George Finney

Chief Information Security Officer for The University of Texas System

"Complexity is the enemy of security."

– Bruce Schneier

Bruce Schneier has probably written one of the most quoted lines in cybersecurity, and I think it's even more relevant in the era of Zero Trust.[1]

One of the biggest concerns I hear from security leaders is that Zero Trust sounds like it'll make everything more complicated. After all, we're talking about microsegmentation, continuous verification, identity-aware firewalls, conditional access policies, workload identities, software-defined perimeters, and about a hundred other buzzwords. It sounds like we're taking an already complicated environment and turning the complexity knob up to eleven.

Ironically, my experience has been exactly the opposite.

I've managed firewall teams responsible for tens of thousands of firewall rules. I've worked with identity teams building conditional access policies across hundreds of applications. Cloud teams are writing policy as code while endpoint teams are managing endpoint detection and response (EDR) policies for thousands of devices.

Every one of those teams is solving a different piece of the security puzzle, and every security product has its own language, interface, and way of expressing policy. If we're not careful, we end up with five different teams implementing five different versions of the same security decision.

That's where most organizations get into trouble. The complexity isn't coming from Zero Trust. It's coming from trying to express the same business policy over and over again in different technical tools.

[1] Bruce Schneier. "Secrets and Lies: Digital Security in a Networked World." John Wiley & Sons, 2000.

Six honest serving men: the Kipling Method

One of the biggest lessons I learned while writing *Project Zero Trust* is that security policies should start with questions, not technology.[2] Better yet, they should start with six simple questions that almost anyone in the organization can answer.

We call this the Kipling Method.

The name comes from Rudyard Kipling's famous poem:

> *I have six honest serving men*
> *They taught me all I knew;*
> *I call them What and Where and When*
> *And How and Why and Who.*[3]

Those six questions — who, what, where, when, why, and how — provide a surprisingly effective framework for designing Zero Trust policies. In fact, they're so straightforward that I sometimes joke a CEO could help write firewall policy.

That usually gets a laugh.

Until they try it.

Security policy should start in the boardroom

Imagine you're sitting in a conference room with your CEO or CIO or CTO discussing access to the payroll system.

I don't ask them what transmission control protocol (TCP) ports should be opened or what virtual local area network (VLAN) the servers belong to. I ask, "Who needs access?" They know that. "What application are they trying to use?" They know that, too. "Where should they be allowed to connect from? When should they have access? Why do they need it?"

Those are business questions, not technical ones.

Only when we get to the final question — "How should we protect that access?" — does the security team really take over. That's where we decide whether to require phishing-resistant multi-factor authentication (MFA), verify device health, inspect traffic with the firewall, require endpoint detection and response, or log activity to the security information and event management (SIEM).

The business defines the policy. Security implements it.

That's an important distinction, because it changes how we think about Zero Trust. Instead of beginning with firewall rules or identity policies, we begin with intent. Technology becomes the implementation detail rather than the starting point.

Here's how I map the Kipling Method into Zero Trust policy:

- **Who?** Which user, group, or workload needs access?
- **What?** What application, service, or data are they trying to access?
- **Where?** Where is the request coming from? A managed laptop? A branch office? A contractor's device?
- **When?** Is access needed all the time, only during business hours, or only for a limited project?
- **Why?** What business purpose justifies the access?
- **How?** Which security controls should be applied to protect it?

Notice what's missing from that list: IP addresses, ports, protocols, and ACL numbers.

None of those things disappear, but they stop being the policy. They're simply one way of implementing the policy. That's a subtle but profound shift. For years, we designed security from the network inward. Zero Trust flips that around and designs security from the business outward.

How the Kipling Method fits into the five steps of Zero Trust

The Kipling Method is the fourth step in the five-step Zero Trust process John laid out years ago: define the Protect Surface, map the transaction flows, build the architecture, create the Zero Trust policy, then monitor and maintain it.[4]

That order matters, because the first two steps are what make the six questions answerable in the first place.

The "who" statement asks who is accessing the resource. This is where identity signals are consumed to be implemented in policy. The validated "asserted identity" will be defined in the Who statement.

"What" is another way of asking what sits inside the Protect Surface. If we haven't defined that surface, "what" expands to fill whatever space we give it, and we find ourselves writing a policy for the data center. A policy written at that altitude protects nothing in particular.

[2] George Finney. "Project Zero Trust: A Story About a Strategy for Aligning Security and the Business." Wiley, October 2022.

[3] Rudyard Kipling. "'The Elephant's Child,' Just So Stories for Little Children." Macmillan, 1902.

[4] John Kindervag. "All Layers Are Not Created Equal." Palo Alto Networks. May 2019.

The same is true of “how” and transaction flows. Until we’ve mapped the way a system actually communicates, which users touch it, which services depend on it, in which direction, and over what, we’re guessing at controls. Mapping flows is what turns “how” from a wish list into a design.

I’ve watched teams skip ahead to step four because writing policy feels like progress. It’s the most satisfying part of the work, and the questions really are simple. They’re only simple, though, after somebody has done the unglamorous work of steps one and two.

Step five is what keeps the policy honest. Contracts end, projects wrap up, and people change roles. A policy that was exactly right in March can be quietly over-permissive by September, and the same six questions are how we review it.

One decision, several teams

This also has an unexpected benefit for teamwork.

Modern security teams are full of specialists. One engineer lives in the firewall and another owns identity. Someone else manages cloud security. Another person spends all day tuning EDR detections. They’re all experts in their own tools, but none of those tools exist in isolation.

Suppose we decide contractors should only access a financial application during normal business hours from company-managed devices. That single decision should show up in several places. The identity team creates a conditional access policy. The firewall team limits network access. The endpoint team verifies device compliance. The SIEM team alerts if someone attempts to bypass those controls.

Everyone is implementing the same business decision, just through a different lens. That overlap isn’t wasteful but intentional.

One of the assumptions behind Zero Trust is that individual controls will eventually fail: identity systems have bugs, firewalls get exceptions, and endpoint agents stop checking in.

If an attacker slips past one layer, another layer should still be asking the same six questions. Does this user belong here? Should they be accessing this application? Is this the right device? Is this the right time? Does this behavior make sense?

Escaping the museum of old decisions

Finally, the Kipling Method helps us simplify one of the most complicated parts of enterprise security: firewall policy.

Traditional firewalls often become museums of old decisions. Rules are added faster than they’re removed, and after a few years nobody wants to delete anything because they’re afraid something critical might break.

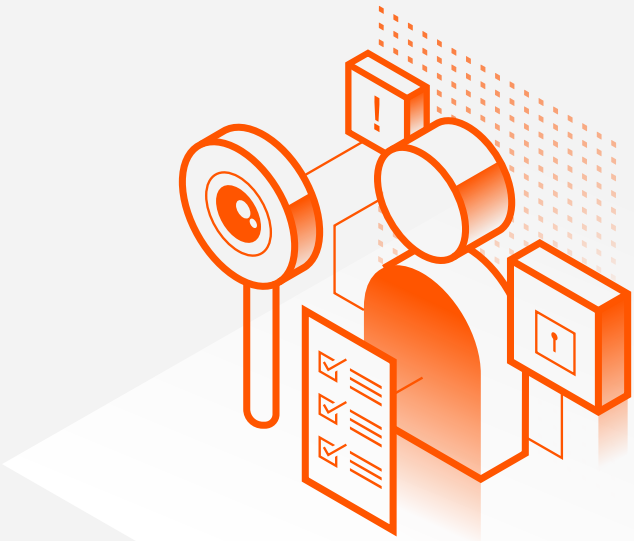
The Kipling Method encourages the opposite approach. Instead of building giant block lists, start by defining what success looks like. Who should be able to do what?

If you can clearly answer those six questions, the policy almost writes itself. Everything else is simply denied by default.

What to do with policy you already have

The obvious objection is that most of us aren’t starting from an empty rule base. You could be facing 3,000 firewall rules that have never been aligned with your Zero Trust architecture.

In these cases, start with one Protect Surface. Pick a single application or dataset, and write its Kipling Method policy from scratch as though nothing had ever been deployed. Then compare that policy against what’s actually running.



The gap between the two is the work list, and it tends to be narrower than people expect. This is because most of the rules in a large rule base have nothing to do with the thing you're protecting.

What the comparison surfaces is access that nobody can justify. You don't have to remove it on the first day. Log it, watch it, and let the evidence make the argument for you. It's far easier to retire a rule once you can show it hasn't matched a packet in six months.

Then do it again with the next Protect Surface. That's how these projects actually get finished, one Protect Surface at a time rather than as a single enterprise-wide cleanup that never quite starts.

Where this can go wrong

I don't want to leave the impression that any of this is effortless. There are a handful of ways the Kipling Method can go sideways.

The most common is answering why with "because we have always had it." That's an inventory note, not a business justification. When "why" comes back that way, it usually means the person who owned the access left the company a few years ago and nobody inherited the decision.

The second is altitude. Policies written too broadly pass the six-question test on paper while granting nearly everything in practice. "Employees need access to the network in order to do their jobs" technically answers all six questions and protects nothing.

A useful test is whether the policy would still be true after you deleted half the words. If it would, it's written too high.

The third is scope. Teams try to write one policy for the entire enterprise, watch it collapse under its own exceptions, and conclude that the method doesn't work. The fact is that there's no single policy. There's one policy per Protect Surface, and they get easier to write as you go.

The fourth is treating the six questions as a form to complete. If you're only filling it out once and filing away, a Kipling Method policy simply becomes another document. But if you revisit it whenever the business changes, it becomes a control.

How I learned all this the hard way

I learned this the hard way over many years. It began with managing firewalls. We would hear from our sysadmins, our server database administrators (DBAs), our app developers that they needed to set up a number of firewall rules. Having your whole team onboard with having security engage before something goes into production is really important.

But after that, the team complained about how long the process was taking for us to get all the information we needed before creating the rules.

As time went on, the firewall team would get blamed when something didn't work right. Vendor documentation sometimes gave us the wrong ports or the documentation would say never use a firewall. We deployed EDR, and the security team would get blamed for blocking software that was unsigned or that had been forgotten about.

We revised our processes at the time to remove some steps that got us a little faster, but what we really needed was buy-in from those other teams. What we learned in improving the process was that none of the other teams had visibility into how security was working. Our old process wasn't collaborative, so different teams kept "throwing things over the fence" at each other and no one understood or saw the whole process.

Our response to this challenge was multifaceted.

We developed an enterprise architecture review board that brought all the teams together to review their architecture, their respective roadmaps, and helped everyone solve our challenges together. We worked with our Project Management Office to embed security team members in all our IT projects. We paid for help desk staff, sysadmins, network engineers, and DBAs to get security training and invited them to attend our meetings.

We didn't need a fancy new tool to follow the Kipling Method, and we didn't need to hire an expensive consultant to make it happen.

All of these steps led us to what John Kindervag had already created with his concept of a Kipling Method policy.[5] We brought all of the teams together that were responsible for a critical asset. We didn't just create policies but coordinated them across different security controls with direct access from our business owners or other key stakeholders.

We want our controls to be in alignment. And it turned out that aligning these controls and working with our stakeholders helped us make some significant changes.

We changed our architecture for our most important asset, and we reduced latency by about 90% while we improved our security controls. We were able to reduce the number of roles in our identity system and provide more granular permissions with a regular review cadence. We were also able to work with our HR team on streamlining permissions for onboarding and offboarding, reducing a key audit finding while ensuring permissions were removed when an employee left the organization.

The Kipling Method was instrumental in helping make security more approachable so all of these key stakeholders could be a part of the conversation.

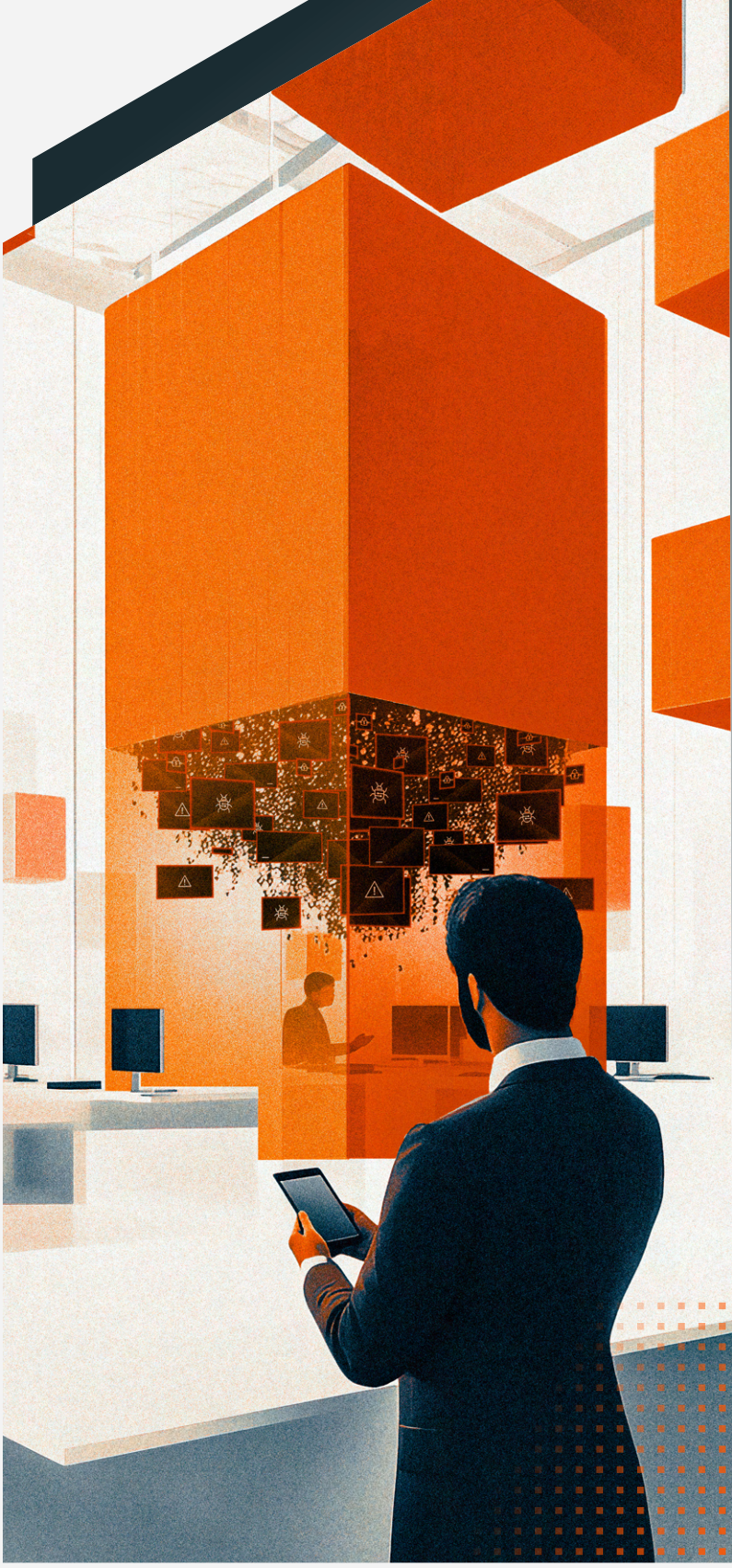
Not only did we improve security outcomes, we improved business processes and reduced impact to users along the way. And the trust that we had created with our stakeholders through inviting them to be a part of the process helped us address even more challenging security issues along the way.

The destination of Zero Trust is clarity

That's why I don't think Zero Trust is really about microsegmentation, least-privilege access, or continuous authentication. Those are all valuable techniques, but they're not the destination. The destination is clarity.

Every access decision should have an owner, a purpose, and a business justification. If we can't explain *who, what, where, when, why,* and *how,* maybe that access shouldn't exist in the first place.

That's the real promise of Zero Trust. Instead of making security more complicated, it instead gives us a simpler way to reason about it.



[5] John Kindervag. "All Layers Are Not Created Equal." Palo Alto Networks. May 2019.

CHAPTER 3

Creating Policy: Writing the Rules of Engagement for a World Where the Attacker Is a Machine

Dr. Chase Cunningham (DrZeroTrust)

Retired U.S. Navy Chief Cryptologist and former Vice President and Principal Analyst at Forrester Research

There's a moment in every tabletop exercise I've ever run when the room goes quiet. It usually arrives about 40 minutes in, after the injects have stopped being hypothetical and started being uncomfortable.

Somebody — usually a lawyer, sometimes a CFO, once memorably a board director who'd been checking email until that exact second — asks the question the entire exercise was secretly designed to surface, "Wait, would our policy have actually stopped that?"

The honest answer, in almost every organization on Earth, is no. This isn't because the policy is missing but because the policy was written for a world that no longer exists.

Security policy, stripped of its compliance theater, is a codified theory of who and what you're willing to trust and with what consequences when the theory is wrong.

This chapter makes the argument that policy that lives in lists is already defeated, and policy that lives in graphs is the only kind that can survive contact with an AI-speed adversary.

That's not a metaphor. I mean it literally, down to the database engine.

Defenders write policy in lists, and attackers read it in graphs

Every organization I've ever assessed, advised, or torn apart in a red team engagement had a security policy. Most had beautiful ones, mapped to frameworks and blessed by legal.

The attacker doesn't care about any of it. They only need to know the gap between what your policy says and what your systems actually enforce.

That gap is the attack surface. Policy, as most enterprises practice it, is a compliance artifact. Policy, as attackers experience it, is terrain, as my friend, former Navy SEAL Officer Clint Bruce, has articulated in a previous chapter.

In my book *Think Like an Attacker*, I named the single most important mental shift a defender can make: attackers think in graphs, while defenders think in lists.[1] A policy document is a list, and a firewall ruleset is a list.

An attacker's mental model of your enterprise is a living map of relationships between users, devices, credentials, service accounts, and trust delegations. Attackers want to know what path exists from where they are to what they want.

This is why the security graphing tool BloodHound works. Graph theory applied to a Windows domain exposes "hidden and often unintended identity relationships" that no list-based review will ever surface.[2]

For 30 years that mismatch was survivable, barely, because the entities exercising access were mostly humans moving at human speed. That world ended in three ways at once.

The population exploded

Machine identities now outnumber human ones by more than 80 to one, and 42% of them hold privileged access. That count includes service accounts, API keys, tokens, workload identities, and now AI agents.

Yet 88% of security teams still define "privileged user" as human-only.[3] The policy is blind to the vast majority of its own subjects.

Those identities are also old and over-armed. In AWS, 62% of them showed no activity in 90 days while keeping their permissions.[4]

Layered on top are agents that spawn sub-agents and hold power without a human at every decision point, and none of them onboard, sit through awareness training, or notice when something feels off.

The adversary industrialized the same technology

Prompt injection is the top-ranked flaw in the OWASP Top 10 for LLM Applications, and it maps to six of the 10 top risk categories for agentic systems.[5]

CrowdStrike found it at more than 90 companies in a single year, used to steal credentials and crypto.[6] Its conclusion was that prompts now work like malware.

Vendors have stopped pretending this can be solved at the model layer. OpenAI said flatly in late 2025 that prompt injection, like social engineering, "is unlikely to ever be fully solved."[7]

That admission is the best argument in this chapter for enforcement that lives outside the AI system in the policy layer.

The clock collapsed

In 2022, when one criminal group handed access to a second to exploit, the median hand-off took more than eight hours. By 2025, it was 22 seconds, and the fastest breakout time on record is now 27 seconds.[8,9]

So ask the only question that matters: what's your policy's reaction time? If revoking a hacked service account takes a ticket, a change window, and an approval chain, you've built a control that moves in geological time next to the attack.

A policy that can't be enforced in machine time isn't a control but a story you tell after the breach.

[1] Chase Cunningham. "Think Like an Attacker: Why Security Graphs Are the Next Frontier of Threat Detection and Response." 2025.

[2] SpecterOps. "BloodHound Community Edition: The Originator of Identity Attack Path Discovery." specterops.io.

[3] CyberArk. "2025 Identity Security Landscape." 2025.

[4] Entro Labs. "NHI & Secrets Risk Report, H1 2025." 2025.

[5] OWASP. "State of Agentic AI Security and Governance." June 2026.

[6] CrowdStrike. "2026 Global Threat Report." 2026.

[7] Dane Stuckey, Chief Information Security Officer, OpenAI. Statement on prompt injection risk in ChatGPT Atlas. October 2025.

[8] Mandiant. "M-Trends 2026." March 2026.

[9] CrowdStrike. "2026 Global Threat Report — Executive Summary." 2026.

The graph has always been the weapon

None of this is new.

In 1736, Leonhard Euler was handed a puzzle about the seven bridges of Königsberg. He realized almost everything about it was irrelevant except for the route.[10] "The only important feature of a route," as the standard account puts it, "is the sequence of bridges crossed."[11]

He reduced the city to four land masses and seven connections, then proved from the structure alone that no such walk existed.

Your enterprise is Königsberg, and the attacker's question — can I get from the phishing victim's laptop to the payment system? — is Euler's question. It's unanswerable by staring at components and trivially answerable from the structure of the graph.

The tradition never stopped. Valdis Krebs reconstructed the 9/11 hijacker network from public sources alone.[12] On active duty, we used graph analytics to track adversary networks, and it was literally how we hunted enemy actors in Iraq and Afghanistan.[13]

We never asked whether a single node "looked legitimate," because a node in isolation tells you almost nothing and any single signal can be faked. What we interrogated was the graph: who does this node talk to, when, and does the pattern match known-good or known-bad behavior?

Legitimacy was never a property of an entity. It was a property of relationships and behavior over time. Twenty years later, I watch security teams evaluate access requests by checking whether a credential is valid, and I want to reach through the screen.

Entities lie; relationships don't. A forged passport survives inspection, but a forged *pattern of life* is orders of magnitude harder to fake.

The $25 million video call where policy worked perfectly

In January 2024, a finance employee in the Hong Kong office of Arup, the British engineering firm behind the Sydney Opera House, got a message about a confidential deal. Then came an invite to a video call with the company's UK-based chief financial officer and colleagues.[14]

The employee was suspicious. The request had the hallmarks of phishing. So he did what a reasonable person, and arguably a reasonable policy, would demand: he checked, and got on the call.

He saw the CFO's face and heard the CFO's voice. He recognized colleagues around the table and sent fifteen wire transfers worth roughly US$25.6 million to five Hong Kong accounts.[15]

Every other person on that call was an AI-generated deepfake, built from public footage.[16]

Be precise about what failed, because the popular reading (employee got fooled, needs more training) is exactly the wrong lesson. He didn't skip a step. He ran the check his company's theory of trust called for, and the check lied to him.

The policy verified artifacts of identity: a face, a voice, and a familiar presence. For the whole history of business, those were safe proxies, because faking them cost far too much.

Generative AI killed that assumption at commodity cost, and Arup's policy was never updated to notice.

That's the signature failure mode of the AI era. The controls aren't broken. The assumptions are bankrupt, and they're executing flawlessly.

Now run it through the graph lens. Fifteen money transfers into five unknown accounts is a graph anomaly, all brand-new edges to nodes with no payment history, no vendor tie, and no prior life in the payment graph. A video call is a single node vouching for itself, exactly the evidence a graph thinker discounts and a list thinker accepts.

The attackers understood Arup's trust graph better than Arup's policy did. They found the one edge that carried near-total weight and had quietly become easy to fake.

What should keep every CISO awake is that the deepfake still needed a human to act.

When the thing sending the money is itself an AI agent, one that holds payment authority and can be hijacked by text buried in a document it was asked to summarize, the attacker doesn't have to fool a person, just a policy.

When the human is out of the loop, the policy *is* the defense.

[10] Leonhard Euler. "Solutio problematis ad geometriam situs pertinentis." Commentarii academiae scientiarum Petropolitanae, 1741.

[11] Brian Hopkins and Robin Wilson. "The Truth about Königsberg." The College Mathematics Journal, May 2004.

[12] Valdis Krebs. "Mapping Networks of Terrorist Cells." Connections, 2002.

[13] Swisscom B2B Magazine. "'For zero trust, you need an offensive mindset' — interview with Chase Cunningham." January 2026.

[14] CNN. "Finance worker pays out $25 million after video call with deepfake 'chief financial officer.'" February 2024.

[15] CFO Dive. "Scammers siphon $25M from engineering firm Arup via AI deepfake 'CFO.'" May 2024.

[16] CNN. "Finance worker pays out $25 million after video call with deepfake 'chief financial officer.'" February 2024.

Five breaches, one root cause: why policies fail

The Arup incident is the AI-era version of a pattern that predates deepfakes. In each incident in the graph below, the attack succeeded by satisfying a point-in-time control while violating every relationship around it.

Incident	What the policy trusted	What the graph would've caught
Arup (2024)	A face and a voice on a video call[17]	Fifteen payments to five accounts the company had never paid before
MGM Resorts (2023)	A caller who knew the right employee details[18]	A password reset on a top account, asked for from a device that account had never used
Change Healthcare (2024)	A valid login to a portal with no second factor[19]	Nine days of systems talking to each other for the first time
Snowflake customers (2024)	A username and password, 165 times over[20]	Logins from countries and devices those accounts had never used
Colonial Pipeline (2021)	A password on a VPN account nobody used anymore[21]	An account silent for months, suddenly waking up on unknown infrastructure
SolarWinds (2020)	A signed update from a trusted vendor[22]	Trust handed down six times with nobody re-checking it

The root cause of each incident is that a one-time check passed while every relationship around it screamed fraud, and nothing was listening.

None of these organizations lacked policy, and none violated their own policy in the moment of compromise. The policies simply lived in lists, and every one of these attacks was a walk through a graph.

Here's what we can learn from these incidents:

Policy fails at the assumptions, not at the controls. Arup's controls ran as designed, as did MGM's help-desk process and Colonial's password check. Every policy rests on quiet claims about what's hard to fake, so list those claims, price each one by asking what it'd cost an attacker to fake that signal today, and price them again on a schedule.

Identity is behavioral, not declarative. The Arup deepfakes were flawless on paper, as were the Snowflake logins with their real usernames and passwords. All of them were absurd in their behavior.

Declarative identity answers "what credential showed up?" Behavioral identity answers "is this thing acting like what it claims to be?" Only the second survives a machine built to forge perfect paperwork.

Policy must live where the attacker operates. "Finance managers may approve payments up to $500,000" is a list statement, exactly as strong as the weakest forged edge feeding it. "Allow this payment if the approval chain matches the known graph for this counterparty and falls within baseline" is a graph statement. It gets harder to fake as it takes in more context.[23]

[17] CNN. "Finance worker pays out $25 million after video call with deepfake 'chief financial officer.'" February 2024.

[18] Computing. "MGM Resorts: Hackers deceived service desk over the phone." September 2023.

[19] BlackDoor LLC. "Change Healthcare 2024 Ransomware Attack: Analysis and Lessons." 2024.

[20] The Register. "Over 165 Snowflake customers didn't use MFA, says Mandiant." June 2024.

[21] Charles Carmakal, Mandiant. Prepared statement before the U.S. House Committee on Homeland Security, June 2021.

[22] Atlantic Council. "Broken Trust: Lessons from Sunburst." 2021.

[23] Chase Cunningham. "Think Like an Attacker: Why Security Graphs Are the Next Frontier of Threat Detection and Response." 2025.

Forging a node is a commodity, while forging a whole subgraph is a research project.

Machines are first-class policy subjects. Every agent and service account needs what any contractor would need before you handed them a badge: a record, a named owner, an end date, and a way to shut it off. NIST flagged non-person entities as a threat category in 2020.[24]

Assume the agent itself is compromised. Prompt injection means an agent's orders can be rewritten by any data it touches. You can't fix that with a better system prompt any more than you could fix malware with a sternly worded memo.

Simon Willison's "lethal trifecta" holds that an agent is exploitable when it mixes private data, untrusted content, and a way to talk to the outside world.[25] Meta's Agents Rule of Two allows at most two of the three without a human in the loop.[26]

The great equalizer: what graph databases give the defender

The graph advantage was never born with attackers. It's a tooling artifact, and the tooling has changed sides.

Defenders think in lists because their instruments think in lists. For 50 years the security stack has run on relational databases, where relationships get rebuilt at query time through joins. Each hop drags performance from milliseconds toward minutes.[27]

Ask such a system a six-hop question, meaning every path from an exposed asset through any chain of credentials to the payment database, and the query is somewhere between punishing and hopeless.

So nobody asked.

Graph databases flip the design. They store relationships as first-class data and make traversal native, at roughly the same cost per hop.[28] Attack graphs then turn cleanup into targeting. If you map every path to every critical asset, you stop patching by severity score and start cutting the choke points that collapse the most paths at once.

Rich policy becomes computable at machine speed, because the check you could never afford (walk the *reports-to* edge, check the *owns* edge, read a timestamp) is suddenly cheap.

If you doubt this runs at production speed, consider Google's Zanzibar, the engine behind access in Drive, Photos, and YouTube. It walks trillions of access-control relationships at millions of requests per second, and 95% of them land in under 10 milliseconds.[29]

Graph-based policy is how the largest access-control system on earth already works.

Nor does the enforcement machinery need inventing. NIST SP 800-207 hands you the architecture: a policy engine to decide, a policy administrator to provision, and a policy enforcement point in the data path.[30] The choice that matters is what that engine consults.

Static lists give you a machine that automates 2010s failures, while relationship-based access control grants access based on the relationships between subjects and resources.[31]

Express it as code. The policy agent's own docs include a worked example that fences in an AI coding agent.[32] The Rule of Two isn't a wish-awaiting technology but rather a dozen lines of policy in front of the agent's tool dispatcher ready today.

Remember how fast this kind of shift changes things. In 2010, John Kindervag argued that trust based on network location is a flaw, and much of the industry wrote him off as a zealot.[33] By January 2022, OMB Memorandum M-22-09 had turned it in binding federal strategy with deadlines.[34]

Graph-native policy is the next jump, and the only choice you get is which side of it you're standing on.

[24] Scott Rose, Oliver Borchert, Stu Mitchell, and Sean Connelly. "NIST Special Publication 800-207: Zero Trust Architecture." National Institute of Standards and Technology, August 2020.

[25] Simon Willison. "The lethal trifecta for AI agents: private data, untrusted content, and external communication." simonwillison.net. June 2025.

[26] Meta AI. "Agents Rule of Two: A Practical Approach to AI Agent Security." ai.meta.com. October 2025.

[27] Neo4j. "How to Bolster Your Cybersecurity with Graphs." neo4j.com.

[28] Neo4j. "How to Bolster Your Cybersecurity with Graphs." neo4j.com.

[29] Ruoming Pang et al. "Zanzibar: Google's Consistent, Global Authorization System." USENIX Annual Technical Conference, 2019.

[30] Scott Rose, Oliver Borchert, Stu Mitchell, and Sean Connelly. "NIST Special Publication 800-207: Zero Trust Architecture." National Institute of Standards and Technology, August 2020.

[31] Oso. "RBAC vs. ABAC vs. ReBAC: What is the best access policy paradigm?" osohq.com.

[32] Open Policy Agent. "Policy-based control for cloud native environments." openpolicyagent.org.

[33] John Kindervag, Forrester Research. "No More Chewy Centers: Introducing the Zero Trust Model of Information Security." 2010.

[34] Office of Management and Budget. "M-22-09: Moving the U.S. Government Toward Zero Trust Cybersecurity Principles." January 2022.

Practical recommendations for building security policy for the AI era

What follows is the discipline I'd impose, in order, if I owned policy creation at any company deploying AI today.

1. **Map the graph before you write a word, and put it in a graph database.** Every identity, agent, credential, and trust relationship, fed from systems you already own. Your map needs to be better than the attacker's, but it doesn't have to be perfect.[35] What you find is the policy backlog.

2. **Write policy as testable statements, enforced as code.** Every rule needs a subject, an action, a resource, and, above all, context. The real test is simple: can the policy engine check that rule in milliseconds against the graph? If it can't, the rule is a wish, and wishes don't stop breaches.

3. **Issue authority the way you'd issue ammunition: scoped, counted, accounted for.** Give every machine identity credentials that are just-in-time, task-scoped, and short-lived. Onboard each agent like a contractor and offboard it like a threat, with a named owner, a written scope, an end date, a tested kill switch, and 60 seconds from decision to dead credential.

4. **Demand a second, separate check for high-stakes actions.** No single channel can stand as the only proof for an action you can't undo, because every channel can now be faked. The check has to travel a different edge of the trust graph than the request arrived on.

5. **Decouple enforcement from the agent.** Guardrails inside the AI system share the AI system's failure modes, so enforcement belongs outside it, in a policy engine that brokers every credential. Add egress controls and segmentation. The agent proposes; the policy layer disposes.

6. **Red-team your policy, not just your network.** Are you willing to let a group of hackers come at you and tell you where you're vulnerable? If the answer's no, then there's your first finding.[36] Can they talk a help desk into an MFA reset, or move a payment with a deepfaked voice?

7. **Watch the graph for odd behavior and let it feed enforcement.** Alert on structural change, such as first-ever links between long-stable systems, dormant nodes waking up, or approval chains skipping their history. Route those alerts straight into the policy engine, so an identity outside baseline faces tighter rules right away.

8. **Measure policy like a business risk (because it is one).** Track mean time to revoke any identity. Track the share of machine identities with named owners, and the share of privileged access that's just-in-time. Report it in money, such as $25.6 million for one forged video call or $872 million in a quarter for one missing MFA control.[37,38]

Stop believing in the trusted signal

The organizations that get breached in the AI era will be the ones whose policies answered yesterday's question, not the ones with missing policies.

Policy is a theory of trust, and AI is proving the old theories false in bulk. Any policy that rests on a signal someone can fake is already broken, whether or not anyone has cashed in yet.

Attackers think in graphs, so write your policy there. List rules fall to one forged edge, while graph rules force an attacker to fake a whole web of relationships that hangs together.

And assume every agent you run is compromised. Credentials, egress, segmentation, and kill switches all have to live beyond that agent's reach.

The Arup employee asked, "Is this really my CFO?" and the policy gave him a way to answer it, for a world that had already ended.

Fifteen years ago, Zero Trust asked us to stop believing in the trusted interior. The next decade belongs to the same idea carried one level deeper: stop believing in the trusted signal and start pricing every edge in the graph.

Do that, and you stop writing documents for auditors and start writing rules of engagement for a fight that's already underway. It's a fight in which, for the first time, the defender can hold the better map.

[35] Swisscom B2B Magazine. "'For zero trust, you need an offensive mindset' — interview with Chase Cunningham." January 2026.

[36] FreeWave. "Stranger than Fiction: 3 Essentials of an Industrial Network Security Strategy — featuring Chase Cunningham." 2026.

[37] CFO Dive. "Scammers siphon $25M from engineering firm Arup via AI deepfake 'CFO.'" May 2024.

[38] UnitedHealth Group. "UnitedHealth Group Reports First Quarter 2024 Results." April 2024.

CONTRIBUTORS

John Kindervag

John Kindervag is considered one of the world's foremost cybersecurity experts. With over 25 years of experience as a practitioner and industry analyst, he is best known for creating the revolutionary Zero Trust Model of Cybersecurity. As Chief Evangelist at Illumio, John Kindervag is responsible for accelerating awareness and adoption of Zero Trust Segmentation.

Most recently, John led cybersecurity strategy as a Senior Vice President at On2IT. He previously served as Field CTO at Palo Alto Networks and, before that, spent over eight years as a Vice President and Principal Analyst on the security and risk team at Forrester Research.

In 2025, John received the Baldrige Foundation Award for Leadership Excellence in Cybersecurity for his pioneering work in Zero Trust. Further, in 2021, John was named to the President's National Security Telecommunications Advisory Committee (NSTAC) Zero Trust Sub-Committee and was a primary author of the NSTAC Zero Trust report that was delivered to the President. That same year, he was also named CISO Magazine's Cybersecurity Person of the Year. John serves as an advisor to several organizations, including the Cloud Security Alliance and Venture Capital firm NightDragon.

Raghu Nandakumara

Raghu Nandakumara is Vice President of Industry Strategy at Illumio, the breach containment company. Based in London, UK, he helps customers and prospects across many industries build resilience and accelerate Zero Trust outcomes with Illumio Segmentation.

Previously, Raghu spent 15 years at Citibank, where he held a number of network security operations and engineering roles. Most recently, he served as a Senior Vice President, where he was responsible for defining strategy, engineering, and delivery of solutions to secure Citi's private, public, and hybrid cloud environments. Raghu holds an undergraduate degree in mathematics and computer science from the University of Cambridge, and a master's degree in advanced computing from Imperial College London.

CONTRIBUTORS

Chase Cunningham

Dr. Chase Cunningham, widely known as DrZeroTrust, has spent his career challenging conventional thinking about cybersecurity and pushing organizations to rethink how they defend what matters most.

A retired U.S. Navy Chief Cryptologist, Cunningham spent years operating in the intelligence and cybersecurity communities before earning his doctorate and becoming one of the most recognized voices in Zero Trust strategy. As a Vice President and Principal Analyst at Forrester, he helped develop and advance the Zero Trust eXtended (ZTX) framework, work that helped move Zero Trust from an emerging security concept to a globally adopted strategic model.

Today, Cunningham is an author, entrepreneur, advisor, keynote speaker, and host of The DrZeroTrust Show, where he brings cybersecurity, technology, business, and strategy together with a distinctly no-nonsense perspective. His work has taken him from military operations and government programs to boardrooms, startups, global enterprises, and stages around the world.

Throughout that journey, one principle has remained constant: progress comes from questioning assumptions, confronting uncomfortable truths, and having the courage to do things differently.

Cunningham writes with the same philosophy he brings to cybersecurity and business—strip away the noise, focus on what actually works, and never be afraid to challenge the status quo.

George Finney

George Finney is a Chief Information Security Officer, author, speaker, professor, consultant, and founder that believes that people are the key to solving our cybersecurity challenges. George was inducted into the CISO Hall of Fame in 2025, was the recipient of the Malcolm Baldrige Award for Cybersecurity Leadership in 2024 and was recognized in 2023 as one of the top 100 CISOs in the world. George is the bestselling author of several cybersecurity books, including the hall of fame inductee Project Zero Trust and the Book of the Year Award winning, Well Aware.

George has worked in cybersecurity for over 20 years helping startups, global corporations, and nonprofits improve their security posture. George received his Juris Doctorate from SMU where he was the CISO and is a licensed attorney. George Finney serves as the Chief Information Security Officer for The University of Texas System to protect over 170,000 employees and more than 250,000 students across 14 institutions. In this capacity he works with each U.T. System institution CISO and their teams to continually improve their security posture and capabilities.