cloud security alliance®

CISCO

Publication

Preparing Your Networks for the "AI Storm"

Hardening your network against AI-accelerated attacks

Acknowledgments

Author

Rich Mogull

Graphic Design

Stephen Smith

About the Sponsor

Cisco (NASDAQ: CSCO) is the worldwide technology leader that is revolutionizing the way organizations connect and protect in the AI era. For more than 40 years, Cisco has securely connected the world. With its industry leading AI-powered solutions and services, Cisco enables its customers, partners and communities to unlock innovation, enhance productivity and strengthen digital resilience.

Table of Contents

Introduction

Advances in AI models are transforming the relationship between attackers and defenders, undermining historic risk assumptions we used to develop our security programs. As shown with the release of models with advanced capabilities, such as Mythos and ChatGPT 5.5 (and detailed in The "AI Vulnerability Storm"), models are rapidly evolving and improving their ability to discover novel vulnerabilities, create new exploits, and engage in complex, multi-stage automated attacks. The past few years have shown us that the capabilities of today's frontier models will quickly be matched by widely available foundation models, and eventually local open-weight models.

The baseline risk models we used to build our security programs no longer hold true. Discovering new vulnerabilities and creating new exploits used to take months or even years, but now take hours. Those discoveries, and using them in attacks, used to require extensive technical skills, but those barriers are rapidly collapsing. The economics of cyber defense are changing; as the time and skill requirements for offense decrease, our historic defensive timelines are no longer effective, and we need to re-orient our programs.

While AI enables both attackers and defenders, it doesn't do so equally. As described in Core Collapse, there is a fundamental mathematical asymmetry: attackers get to focus all their efforts on a single problem (find a vulnerability *here*, develop an exploit for *this*, or attack *that* target), defenders face the combinatorial complexity of having to defend all resources from all attackers from all potential exploits for all time.

This is easily illustrated by looking at the impact of a single, newly-disclosed vulnerability. Attackers can use AI to rapidly develop an exploit and start automated attacks. Defenders, even with a patch in hand, have to identify every resource that needs to be patched, need to test it, and then need to deploy it without breaking running applications. It's a race running on different physics, no matter how good we are.

There are existing strategies that help rebalance the equation. Over time, as all new software is assessed and hardened before release using LLMs, we can expect the overall number of vulnerabilities in software to decline. And as defenders today, we can also add security boundaries to *increase the combinatorial complexity for successful attacks*.

Building resiliency to AI-empowered attackers spans all aspects of technology, and extends well beyond what security teams can manage themselves. We need to increase security barriers across all areas, from identities and workloads, to containers, APIs, and endpoints.

Networks are one of our most important tools to enforce these boundaries, but the network security, design, and operating models of most in-production, real-world networks are often insufficient. Despite years of advancement and investment, we still have too many flat networks, often rely on outdated hardware, and have network operational and security processes that aren't prepared for rapid response and automation. Something as simple as changing a firewall rule can take weeks or months in organizations with complex networks.

To increase attacker complexity (and costs), our networks should:

- Rapidly accelerate patching cadence and target patching on a continuous basis.
- Reduce blast radius through improved network segmentation and limited locations of high-value data exposure.
- Add multiple *serial, not parallel,* defensive layers to force attackers to bypass multiple boundaries. E.g. multiple firewalls at different layers within a single application stack.
- Be supported by responsive operational processes that can adapt as quickly as the threat environment changes.

The use of advanced AI capabilities for defensive purposes will eventually enable AI-proficient organizations to have timing competitive enough to withstand the attacker's speed of developing exploits. In the meantime, security boundaries will be the tool to slow down attacks and buy us the time to update deployments.

This paper provides guidance for network and network security administrators. We can't address our new resiliency requirements with network defenses alone, but increasing network security boundaries is one of the most effective places to start.

Modernizing Networks for Responsive Defense

As an industry, we have been modernizing our networks for over a decade, from improving our perimeter defenses to increasing network segmentation, and this has only accelerated in recent years. Migrating to the cloud pushed us even further, since cloud networks are software-defined by default, and software-defined networks make boundaries and segmentation far easier to create and change.

But we've faced real, legitimate challenges in updating these networks at scale. Many, if not most, physical networks are still relatively flat on the inside. Even cloud networks, which are highly flexible, easy to segment, and always software-defined, aren't always better, largely due to lift-and-shift migrations that copied the old datacenter designs. Zero Trust implementations too often have focused more on securing the workforce and campus edge, and less on datacenter and app-to-app (east-west) traffic. We don't necessarily own and manage our own networks, and face limitations to third-party observability.

Many organizations started on microsegmentation, but typically only for small subsets of their networks (due to implementation complexities). Much of our hardware is older, and difficult and slow to update. And our operational processes are deliberately methodical: outside the perimeter, we've never really needed to patch the internal network quickly, so our processes focused on the edge, with internal patching moving much more slowly.

None of this was wrong; it was a reasonable response to the risks we faced. Most of our recommendations in this paper are well-known and in progress. The problem is that AI changes the risks and the timelines. Not only the perimeter, but our internal network barriers become a critical expansion of security boundaries that are essential to stifling the attacker's AI advantages. Flat networks mean a larger blast radius when something gets popped. Slow, methodical change can't keep pace with exploit development that now moves in hours. And hardware you can't patch quickly, even if it's behind the firewalls, becomes a much greater liability.

The good news is that we already know what to do, but the new challenge is execution. We need to move at a rapid pace since the underlying risk assumptions we've used to justify historical priorities and timelines no longer apply. Our network security fundamentals of segmentation, patchability, and responsive operations can be highly effective in balancing the equation, but we now need to execute with greater granularity, in a prioritized order, on tight timelines.

Network engineering/operations and network security also need to work hand in hand to tackle these problems. The solutions span silos, requiring a combination of fundamental architectures, patching and vulnerability management, and security-specific defenses like NGFWs and WAF. The rest of this paper lays out the elements for modernization, then how to get there in phases.

The Building Blocks and Process

There are three dimensions to network modernization to respond to adversarial AI risks:

- **Operating Model/Processes:** Since we started by running wires to boxes and defending slow-to-change IP addresses, network and network security processes tend to be methodical, especially for hardware updates and structural changes. Those processes were built for environments with physical dependencies. Patching focused on the edge, with internal hardware secondary since it was lower risk. Most networks implemented one or two-layer defenses, focused on North-South traffic to defend from Internet threats. Cloud is more dynamic, but unevenly so, based on maturity, reliance on lift-and-shift, and hybrid dependencies. The shift is from *periodic projects to continuous operations*. Not just for patching and refreshes, but for segmentation itself: adding and keeping boundaries as we build, rather than as a one-and-done architecture. Think of it as two loops running in parallel:
    - **Keep the network patched and current**: the goal is continuous patching plus a prioritized refresh cadence, so you're never stuck defending something you can't fix. Historically this has been extremely challenging for network devices, which is why we recommend treating this as a new operational process and starting with external facing resources. This can and should be integrated into your *VulnOps* program. Described more in the Vulnerability Storm report, VulnOps treats vulnerability management like a DevOps-style program with continuous automation.
    - **Update segmentation and isolation**: move towards microsegmentation and continuously add and maintain multiple boundaries (North, South, East, and West) as applications change, implemented with a combination of topology and security defenses.

- **Infrastructure:** The network itself is a target, and we expect to see increasing volumes of vulnerabilities and exploits for network hardware and software used in automated attacks. This is already happening today; in the weeks we've been writing this paper there have been multiple campaigns targeting network infrastructure. Networks should be:
    - **Current**: updated hardware that is not end of life and is still supported.
    - **Patchable:** not only updated hardware, but integrated into a patching program capable of updating within hours, not weeks.
    - **Software-Defined:** As demonstrated so well in cloud, Software Defined Networks (SDN) are more agile and often default-deny (depending on platform e.g., Software Defined Perimeter (SDP)), with segmentation being the ground state instead of an add-on.
- **Architecture:** Architecture defines the boundaries that increase complexity and costs for attackers. We can't necessarily move instantly to every asset being firewalled at every level, but we build off these blocks:
    - **Perimeter security:** it's what we tend to be best at today, and that need doesn't go away. But, as you'll see, we need to align it with our more continuous operating model, and in some situations, we may want greater diversity of platforms and increased perimeter segmentation.
    - **Macrosegmentation:** better segmentation of business units, application stacks, and operating environments. The objective is to limit the blast radius to a single "stack" within the network.
    - **Microsegmentation:** Adding boundaries within the layers of an application stack, while also increasing boundaries between applications and services.
    - **Zero Trust:** combines those boundaries with identity and context-aware policy that's continuously verified, so access is granted per request instead of by network location. Trust stops being a function of an IP or a VLAN and becomes dynamic, least-privilege, and tied to identity and risk rather than topology. This includes technologies like the Software Defined Perimeter (SDP).

Underpinning all of this is visibility (what do we have where and how is it configured), attribution (who owns what and is responsible for the business and technical decisions), and, ideally, real-time telemetry. *We aren't downplaying the complexity of meeting these recommendations*, but modern technologies do make it more achievable, especially on a per-project basis.

We can also use the AI tools ourselves for continuous testing and improvement. Point your own LLMs at your network, provide them (safe) assessment tools, and use those results to see your strengths and weaknesses from the attacker's perspective.

All of this takes time, sometimes years, and AI-driven adversarial attacks have already started. We recommend taking a phased, prioritized, risk-based approach that can more realistically provide tangible results quickly, while still forming a good foundation to achieve long-term objectives.

Important point: the following phases can run in parallel, as part of a continuous operating process. You don't need to fully complete one phase before moving on to the next.

Phase 1: Prioritized, Focused Refresh and Hardening

Attackers will start with what they can reach, and as defenders, we care about what's going to hurt us most when it's compromised. Thus, the first step is identifying your external attack surface and aligning that to your organization's priorities.

For this phase, think "outside in, key assets first". You start by determining your Internet-facing attack surface, while also identifying your most important digital assets (the "crown jewels"). When determining the attack surface, we mean *everything*. Every device and service, *including all your security and network devices and services*. Anything and everything that touches the Internet and is potentially reachable from the outside. Don't forget assets like VPN servers, remote offices, edge routers, and cloud-based virtual security appliances.

Critical digital assets, or the "protect surface", are those applications and services that would most harm the business if they were compromised in a data breach or destructive attack. For these services, you want a complete map of connectivity and data flow. These are often business questions that the network and security teams can't answer themselves.

Then cross-correlate. For the given critical assets, from the outside in what networking devices are in line. Firewalls, WAF, routers, switches, VPNs (which might not be obvious)... all of it. Use this chart to determine your next actions:

Status	Recommended Action
Device End of Life (EoL)	Refresh hardware: rip and replace
Device in extended support	Replace as soon as feasible
Behind on patch level from current	Patch immediately
Device not deployed with HA (physical or virtual)	Add HA pair to support patching without downtime
Device does not support patching on the day of release due to process or hardware/software limitations	Add to a VulnOps program to support day of release patching, or refresh to hardware that supports day of release patching

There are also some tactical actions to take on all of these devices. And again, we don't assume these are easy, but they are highly valuable defenses that help prevent multiple, in-use attack techniques:

- Require MFA for management at a minimum, and admin should ideally be PAM-brokered.
- Disable management interfaces exposed to the Internet or the DMZ. Management interfaces include management access to devices, SDN controllers, and other management stations. They should reject all traffic from untrusted networks.
- Enable egress filtering. This helps reduce attacker command and control callbacks. Remember that this should include DNS-based filtering and not just IPs.
- If you use any automation scripts or code that touch network/security management, assess those for stored secrets, and use an LLM to assess them for security defects
- Put a WAF in front of all Internet-facing applications (device, virtual device, or cloud service)

This first phase focuses on hardening what we have, and ensuring we can keep it up to date moving forward. We remove high-risk network and security tooling that is most likely to be compromised itself. A security boundary is worthless if it becomes the entry point for the attack.

Double Up: Protecting OT and High Criticality Networks

Operational Technology/Industrial Control Systems may be difficult or impossible to patch with any regular cadence, if at all. While OT used to live in isolated environments, for decades we have increasingly connected them to IT networks, opening up routes from the Internet. They are prime targets, critical infrastructure, and poorly suited to defend from the rapid pace of AI discovered vulnerabilities, exploits, and attacks.

These networks should be disconnected from any route to the Internet. If that isn't possible, consider one-way technologies to collect telemetry like data diodes. If that isn't possible, use multiple (2 or more) inline next-generation firewalls from different vendors that are kept up to date so an attacker needs multiple zero day exploits to get to the OT network.

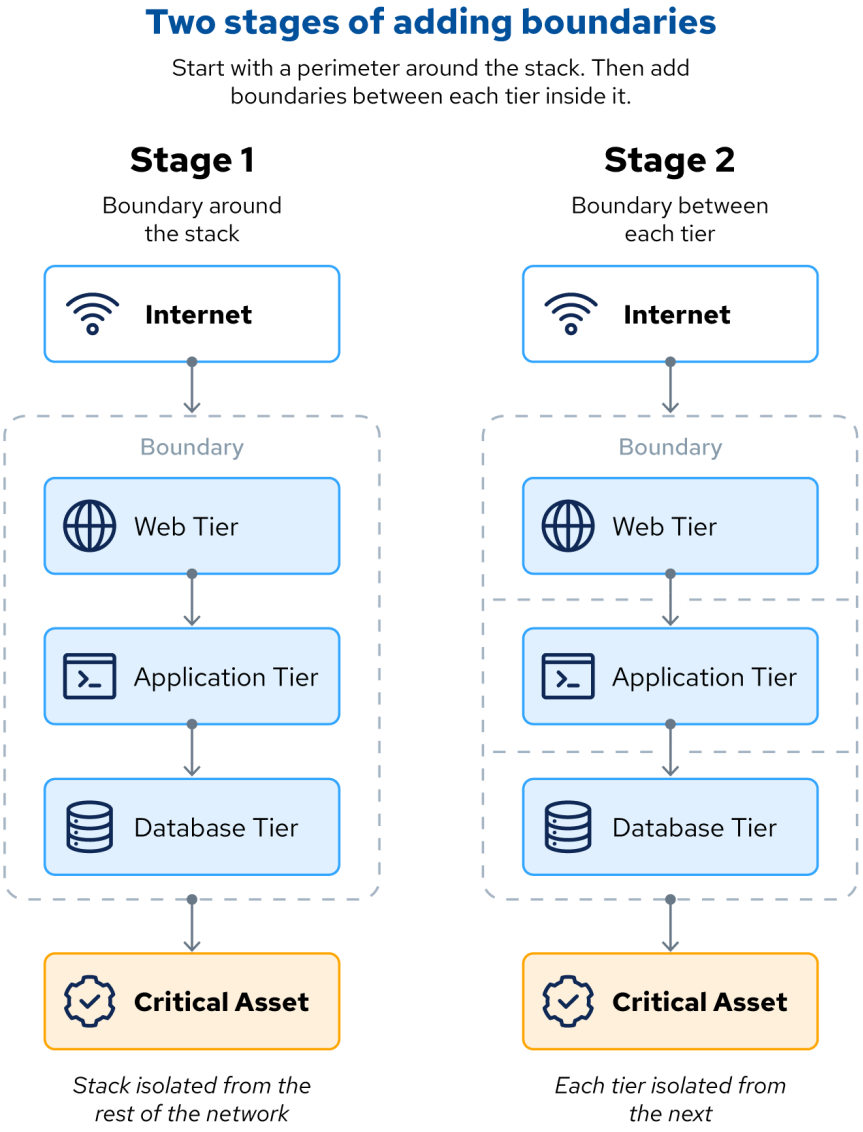
Phase 2: Add Prioritized, Focused Boundaries

Phase 1 focuses on hardening known assets and removing out-of-date hardware and software that can't be patched, or can't keep up with the pace of AI-driven patches. Phase 2 starts to add security boundaries to increase attacker combinatorial complexity and reduce the blast radius of successful attacks.

The first step is essentially macrosegmentation: isolate the stack so a compromise of less-valuable and less-defended resources on the same network don't become the easy path to the prioritized stack. There are multiple networking techniques you can apply here, including adding firewalls, isolating the subnet(s) and route(s), moving to an isolated virtual network, walling off using SDN controls, or jumping forward

into Zero Trust. Isolating from East-West attacks is the first priority, assuming you have already hardened against attacks from the North (the Internet) in Phase 1.

If you choose to isolate by using tooling within the host, that needs to operate at a low level, ideally outside the operating system (hypervisor) or a hardened agent that's externally managed so an attacker who does get onto the host can't simply disable the network security.



The second step, and one that will take more time and likely require new technologies, is to add additional *inline boundaries*. Our previous steps focus on reducing the easy path, but this step adds additional, real attacker costs. Each tier of an application should only talk to the tier above and the tier below, and only on the expected ports and protocols. SDN is the best option, and this pattern is often far easier to implement in cloud and fully virtualized environments.

Phase 3: Expand and Refine

Phases 1 and 2 are highly focused on rapidly protecting our most important assets. We replace out-of-date hardware and software, get everything up to date, isolate from less-defended resources, and then start adding security boundaries to increase attacker cost and complexity. Keep in mind, this is only the network and network security side of the equation; per our AI Vulnerability Storm Ready guidance, there are other parallel initiatives that are just as important.

Phase 3 is about extending and refining these defenses to the rest of our networks. We ideally move from flat networks, to macrosegmentation, to microsegmentation and then Zero Trust. Although moving directly to Zero Trust is ideal, it may not be as achievable in the short term. For macrosegmentation, we start walling off business units and trust zones.

Microsegmentation drives boundaries down to individual workloads and services, so each one only talks to what it actually needs. This can be slow and expensive at scale, and is usually implemented on a project-by-project basis. It can take years, it demands real flow and dependency mapping before you change anything, and it can break production if you rush it.

At CSA we define microsegmentation as part of Zero Trust, but to some network and security administrators they are distinct yet overlap to some degree. Microsegmentation can be implemented as network-only and isolates on a per-asset and per-application basis, with Zero Trust adding identity and context to connectivity decisions.

Zero Trust is where this all comes together. Once your boundaries are fine-grained and your policies follow workloads and identities instead of network location, you're authorizing access per connection. This is a significant barrier for attackers as has already been proven in the cloud. Watch your choke points as you go: identity providers and other consolidated control points become the highest-value targets precisely because everything trusts them, so they earn more scrutiny and more boundaries, not less.

During this phase, it’s absolutely critical to take a hard look at your technical management processes and interfaces. Networks require constant updates beyond just patching, and those management interfaces are a ripe target.

None of this is ever really "done," and that's the point. The networks that hold up under AI-driven attacks won't be the ones that finished a phase, they'll be the ones that made segmentation, patching, and visibility continuous, so every new application ships with its boundaries already in place, and every new exploit runs into a network built to cost the attacker more than it costs you. Incidents will still occur, and improving network telemetry also empowers incident response teams.

A Hard Transition, a Bright Future

None of the techniques in this paper are new, but the scale, scope, and timelines for implementation are different from how we have operated. Everything discussed is a foundational security principle, many of which are battle-tested in various environments. But we, as network and security professionals, must always balance friction with risks. Adding security controls adds cost and complexity, and can slow parts of the business down. Our recommendations aren't always easy, and certainly aren't free. We simply don't see any alternatives.

AI changes the fundamental risk assumptions we built our current practices around. It enables a wider range of lower-skilled attackers to operate closer to elite bug hunters and penetration testers. We can't rely on having weeks or months to patch the edge, and years to patch our internal infrastructure. We can't assume attackers will be unable to navigate our complex internal architectures. We can't expect that one unpatched VPN server that only that one branch office uses is safe because no attacker would care.

The math may favor offensive security today, but the path to balance the equation is clear. Eventually, all software will be inherently more secure before it's released as we fully integrate the same AI capabilities into software development and security assessments. As we modernize our networks and drive towards Zero Trust, we increase the number of security boundaries attackers need to bypass, increasing their combinatorial cost and complexity. When we can patch and update continuously, we narrow the time they have to operate.

It's this combination that shifts the math in our favor, and even small changes can bring large benefits. Update your operating model to be more responsive and integrate VulnOps. Focus on protecting your most critical assets first. Make sure you are on updated hardware and software that can be patched the day patches are released. Move as much as you can to SDNs, which include most virtual and cloud networks, since those make segmentation materially easier to implement and maintain. Start with East-West segmentation to isolate individual stacks, but drive towards increasing inline, North-South boundaries, and eventually land at Zero Trust and SDP.

Zero Trust or zero days, your choice.

The organizations that thrive in the AI era won't be those that eliminate every vulnerability. They'll be the ones that continuously reduce exploitable exposures faster than attackers can capitalize on it. Barriers buy time, and increase attacker costs. It's this combination of surface reduction, reaction speed, and security boundaries that creates true resiliency.