HORIZON3

Operationalizing CTEM: A Practical Playbook for Continuous Threat Exposure Management

How security leaders move from visibility to measurable risk reduction

WHITEPAPER

Most organizations will tell you they have a Continuous Threat Exposure Management (CTEM) initiative underway. They've expanded visibility, layered in prioritization, and invested in security tools that promise risk-based insight. From the outside, it looks like progress.

But if you press on outcomes like what's actually been reduced, what risk has been eliminated, and what would stop an attacker today, the answers get vague, fast. Because beneath all of that activity, very little has changed. In most environments, attackers are still discovering paths to exploitation and compromise.

Today's security teams remain buried in findings. Prioritization continues to lean heavily on scores and assumptions. And when leadership asks, "What actually matters right now?", the response is often a mix of educated guesses and partial truths. This isn't because CTEM is flawed. It's because most organizations haven't operationalized the parts that make it work.

CTEM introduced an important shift. It moved teams away from managing vulnerabilities as isolated issues and toward managing exposures as they exist in the real world. But in practice, many programs stall before they ever reach that point.

Most programs fall into a familiar pattern:

- → Discover more
- → Prioritize differently
- → Report more frequently

But they never answer the questions that actually matter:

- → What can an attacker do with this weakness?
- → Which exposures create real business risk?
- → Did fixing this change anything meaningful?

That gap between visibility and reality is where CTEM initiatives break down. Visibility alone doesn't reduce risk. It just makes the problem easier to see. The goal is not to see more. The goal is to understand what matters, act on it with confidence, and know that the action made a difference. That's what this playbook is designed to help you do.

Table of contents

What it actually takes to run CTEM 4

The capabilities you need 4

The people required to make it work 5

A reality check before you start 5

The CTEM operating loop 6

1. Discover exposure 6

2. Validate what's exploitable 6

3. Prioritize based on impact 7

4. Remediate with clarity and ownership 7

5. Verify fixes work 8

6. Repeat the loop 8

Running CTEM across your entire environment 9

External attack surface 9

Internal systems 10

Cloud and identity 10

Web applications 10

Third-party connections 11

Metrics that actually matter 12

Execution metrics 12

Validation metrics 12

Verification metrics 12

Outcome metrics 13

CTEM Maturity Model 14

From visibility to control 14

What it actually takes to run CTEM

CTEM isn't a product category, it's an operating model. And like any operating model, it only works when the pieces around it (data, people, and processes) are aligned and driving toward the same outcome. Most organizations already have many of those pieces in place. What they don't have is a way to connect them so they consistently produce measurable results.

The capabilities you need

Gartner® defines Continuous Threat Exposure Management as a five-phase process: Scope, Discover, Prioritize, Validate, and Mobilize. Those phases describe what a CTEM program does. To execute them consistently, organizations need a set of supporting capabilities that provide visibility into the environment, context around exposure, validation of exposure, mechanisms for remediation, and measurement of outcomes.

At a minimum a functioning CTEM program depends on 5 core capabilities.

1. **Visibility across your environment:** External assets, internal infrastructure, cloud workloads, identity systems, web applications, and third-party connections. You can't manage exposure if you don't understand where it exists.
2. **Exposure data:** Vulnerabilities, misconfigurations, identity risks, and data exposure signals. This is the raw input every program depends on.
3. **Validation:** The ability to test whether exposures are actually exploitable, and what happens if they are.
4. **Remediation workflows:** Clear, repeatable processes for getting issues into the hands of the teams that can fix them, and tracking progress to completion.
5. **Measurement:** Not just activity metrics, but indicators that tell you whether exposure is actually being reduced over time.

Most organizations are not starting from zero. They already have strong visibility and plenty of data. Many have ticketing systems and SLAs in place. What's missing is the ability to connect those pieces into a continuous loop that produces clarity and drives action.

The people required to make it work

Continuous Threat Exposure Management doesn't sit neatly within a single team. Security may own the program, but they don't own the environment.

That responsibility is distributed across the organization, often as follows:

- Security defines scope, drives validation, and tracks outcomes
- IT and infrastructure teams fix systems, apply patches, and maintain availability
- Cloud and DevOps teams manage configurations, access, and deployment pipelines
- Engineering teams own application and pipeline changes
- Leadership sets priorities and makes tradeoffs between risk and operational impact

This is often where coordination starts to stall because different teams operate on different timelines and often in silos. Fixes introduce risk, ownership is often unclear, and the same issue can touch multiple teams. Without shared ownership and clear accountability, remediation slows down or stops altogether.

Operationalizing CTEM is as much about coordination as it is about technology.

A reality check before you start

It's worth being honest about where most organizations are today.

You likely already have:

- Strong discovery capabilities: Vulnerability scanners, CSPM, EASM, and other tools generating continuous visibility
- Some form of prioritization: Risk scoring, threat intelligence, or RBVM platforms
- Established remediation processes: Ticketing systems, patching workflows, and SLAs

CTEM doesn't require replacing what you have. It requires connecting it through a loop that validates what matters, focuses remediation effort, and verifies that the outcome changed your exposure. Without that loop, CTEM becomes another layer of insight rather than a system for action. With it, it becomes a system that drives measurable improvement.

The CTEM Operating Loop

At its core, CTEM is not a set of activities. It's a system that runs continuously, connects each stage to the next, and produces outcomes you can measure. Most organizations already perform pieces of this loop. What's missing is consistency. And more importantly, the discipline to close it. The objective is straightforward. Identify what matters, fix it, prove it's fixed, and repeat.

1. Discover exposure

Every program starts with discovery. This is where signals from across the environment come together: vulnerabilities, misconfigurations, identity risks, and exposed assets. It all shows up here. Discovery produces volumes of data. It surfaces everything that could be a problem, without telling you what actually is. Left on its own, discovery creates too much noise, and noise slows everything down. The goal at this stage isn't precision. It's coverage. You need enough visibility to understand where exposure might exist, but not so much that decision-making becomes impossible.

2. Validate what's exploitable

This is where the loop moves from theory to reality. Validation answers a simple question. Can this weakness actually be exploited by an attacker? Instead of relying on severity scores or assumptions, you test:

- → Whether a known vulnerability can be exploited in your environment
- → Whether it can be chained with other weaknesses
- → How far an attacker can move once exploitation is achieved
- → What systems, data, or privileges become accessible

This step helps convert volume into something teams can act on. It also exposes something most teams don't see until it's too late. Not everything that looks urgent is exploitable, and some of the things that matter most don't look urgent at all. That's why prioritization based on scores alone often fails.

Validation changes how risk is understood and communicated. Instead of abstract descriptions, you're pointing to outcomes such as access gained, systems reached, and data exposed. Yet most organizations struggle here. Not because they don't understand the need, but because they don't have a reliable way to measure exploitable risk consistently. Without validation, prioritization stays inconsistent.

3. Prioritize based on impact

Once exposures are validated, prioritization becomes less subjective. You're no longer guessing which issues matter most. You're deciding based on what has already been proven.

What matters now is:

- → The business criticality of affected systems
- → The level of access gained
- → The ability to move laterally
- → The potential impact to data, operations, or customers

At this stage, the backlog should shrink. What remains is a smaller set of issues that actually matter. Not because they scored high, but because they're proven to create real risk. This is also where alignment starts to improve across teams. When priorities are grounded in evidence, decisions are easier to justify and harder to ignore.

4. Remediate with clarity and ownership

This is often where execution loses momentum. Not because teams don't know what to fix, but because execution breaks down. Findings don't map cleanly to owners and fixes can introduce operational risk, so work gets pushed behind other priorities. In some cases, teams don't trust the findings enough to act.

Therefore, effective remediation depends on clarity:

- → Who owns the issue
- → What needs to be done
- → Why it matters

Providing validation evidence changes how teams respond. When impact is clear, remediation becomes less of a debate and more of a decision. But even then, this step is not automatic. It requires coordination, communication, and follow-through across teams that are not always aligned. Without that, the loop breaks.

5. Verify fixes work

This is one of the most overlooked steps, and one of the most important. Fixing something doesn't mean the risk is gone. **But verification ensures:**

- → The exposure is actually removed
- → The attack path is no longer viable
- → No alternate path still exists

This requires retesting. Without verification that issues are resolved, organizations default to assumption. Over time, that erodes confidence in both the process and the results. Closing the loop means proving that the outcome changed, not just that an action was taken.

6. Repeat the loop

Environments don't stand still. New systems are deployed, configurations drift, identities change, and new vulnerabilities emerge. CTEM is not a project with an end state. It's a process that runs continuously. As a result, the loop should be executed:

- → On a regular cadence
- → After major changes
- → Following critical vulnerability disclosures
- → In response to incidents

Over time, consistency builds something most programs lack. Not just visibility, but a measurable reduction in exposure. You begin to see not only what's wrong, but whether it's improving. And that's what turns CTEM from a concept into an operating capability.

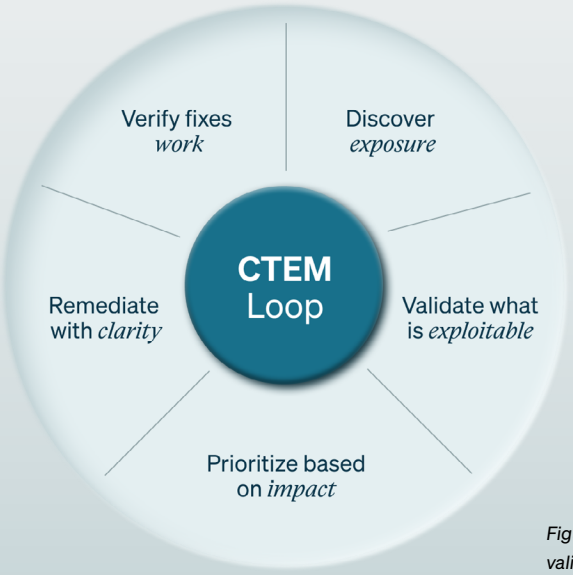


Figure 1: The CTEM operating loop: discover, validate, priotize, remediate, and verify

Running CTEM across your entire environment

The operating loop just discussed stays the same. What changes is knowing where exposure originates and how attackers could move through your environment. Most organizations aren't dealing with a single system. They're managing a mix of internal infrastructure, external assets, cloud environments, identity systems, web applications, and third-party connections.

The key point is that CTEM cannot be applied to only one part of the environment. The loop must be executed across every attack surface an attacker can reach. Otherwise, exposure simply migrates to the gaps between domains.

Security teams tend to treat the following domains as separate entities. Attackers don't. They move across them without friction. Applying the CTEM framework allows security teams to reflect that reality, not just in how exposure is discovered, but in how it is prioritized, validated, and remediated in the context of the entire attack surface.

The following are the types of exposures CTEM is designed to surface, validate, and prioritize across every domain of the environment.

External attack surface

This is where attacks often begin. Internet-facing systems, unmanaged assets, and exposed services define your entry points. Discovery will show you what's exposed but doesn't tell you what matters.

The real questions are:

- → Which external weaknesses can actually be exploited?
- → What happens next if they are?

The risk is not just external access. It's whether that access can be turned into internal compromise and in many environments it can. CTEM forces that question early, before an attacker answers it for you.

Internal systems

Once an attacker gains a foothold, the internal network becomes the path to deeper access. The question is not whether weaknesses exist. It's how far those weaknesses can take an attacker.

Here, the focus shifts more to movement:

- How easily can privileges be escalated?
- How far can an attacker move laterally?
- What critical systems or data become reachable?

Common issues include weak credentials, overprivileged accounts, misconfigured controls, and gaps in segmentation. On their own, these may not look urgent. But when chained together, they often lead directly to domain-level access or control of critical systems.

Cloud and identity

This is where many modern attacks actually succeed. In cloud environments, access is defined less by infrastructure and more by identity. Permissions, roles, and credentials determine what is reachable. That changes how risk needs to be evaluated.

A single misconfigured role or exposed credential may not look significant on its own. But in practice, it often opens the door to privilege escalation, cross-account access, or control of critical resources. These are not isolated issues. They are access paths.

Validation here should focus on what those paths enable:

- How privileges can be escalated
- How roles can be abused
- What systems or data become reachable

Without that, teams are left managing permissions in isolation, without understanding the impact.

Web applications

Web applications can exist on the internet, inside the corporate network, or in cloud environments. Regardless of where they reside, they often serve as a bridge to sensitive systems, data, and identities, making them a common target for attackers.

The focus should be on impact:

- → Can access controls be bypassed?
- → Can sensitive data be exposed?
- → Can the application be used to gain access to underlying systems or identities?

A vulnerability on its own does not define risk. What matters is what an attacker can achieve with it. CTEM brings that context by validating which web application weaknesses create meaningful exposure and contribute to real attack paths.

Third-party connections

Third-party risk is often managed through process. Assessments, attestations, questionnaires, and periodic reviews create the appearance of control. But they rarely answer the question that matters: What access do these third parties actually have, and what could they do with it?

In practice, third-party connections often introduce indirect access into internal systems, data, or identity infrastructure. That access is rarely tested.

CTEM brings this domain into scope by treating third-party access like any other exposure. It needs to be understood, prioritized, and validated in the context of real attack paths. Trust is not a control. Access management is.

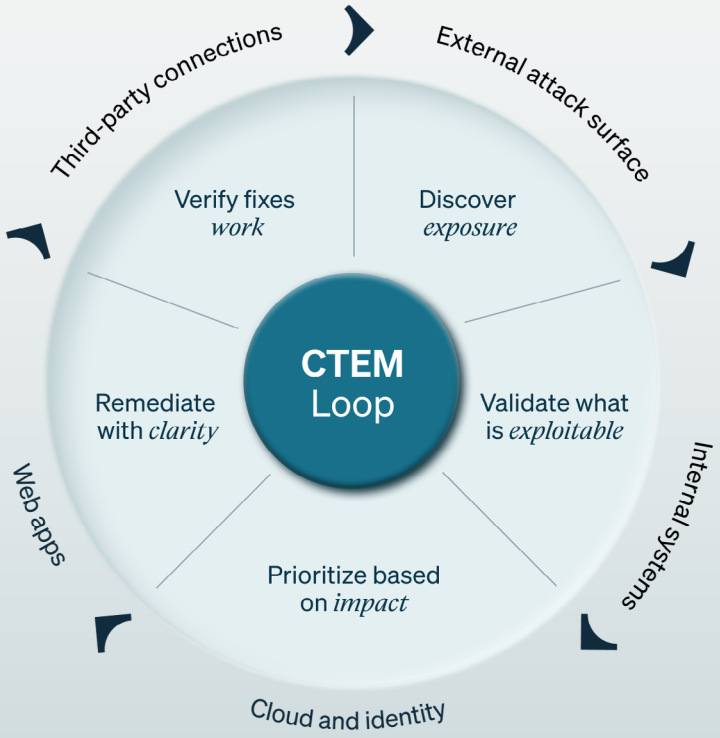


Figure 2: The CTEM operating loop applies across every attack surface, helping organizations continueously discover, validate, prioritize, remediate, and verify exposures in the context of real-world risk.

Metrics that actually matter

Metrics shape behavior. If you measure activity, you get activity. If you measure outcomes, you get progress. But most security programs still measure activity:

- Number of vulnerabilities
- Patch timelines
- Coverage rates
- Tickets closed

These are useful, but they don't tell you whether risk is actually going down. That's the problem. CTEM requires a different lens, one that reflects what an attacker can do, not just what exists in your environment.

Execution metrics

- Mean Time to Mitigate (MTTM)
- Mean Time to Remediate (MTTR)
- Recurrence Rate (ROR)

These metrics measure how effectively your organization is operating. They tell you whether issues are being addressed, how quickly, and whether the same problems keep coming back. But speed alone doesn't mean you're fixing the right things.

Validation metrics

- How quickly critical exposures are validated as real risk
- How much of your backlog is based on proof of risk
- How often attack paths are validated before remediation

These metrics measure how grounded your program is in reality. Without validation, teams spend time debating priority. With it, they can focus on what's already been proven. This is where consistency breaks down. Validation of risk is inconsistent, and as a result, prioritization never fully stabilizes. Also, if fixes are not verified, risk likely remains.

Verification metrics

- Percentage of remediated exposures that are re-tested
- Time to verify that a fix removed the risk
- Percentage of verified fixes that fully eliminate the attack path

These metrics measure whether remediation efforts are actually achieving their intended outcome. Closing a ticket does not necessarily mean risk has been reduced. Verification confirms that exposures have been removed, attack paths no longer exist, and remediation efforts produced the desired result. Without verification, organizations are left assuming fixes worked rather than proving they did.

Outcome metrics

- Reduction in exploitable paths
- Reduction in blast radius
- Decrease in high-impact exposures over time

These are the metrics that actually matter. They reflect whether your environment is becoming harder to exploit. This is also where leadership starts to pay attention. Not because the numbers are higher or lower, but because they finally connect to risk.

CTEM Maturity Model

CTEM maturity does not happen overnight. Organizations need to crawl first, walk next, and then run continuously. The goal is to move from assumptions about risk to evidence, and from isolated security activities to a repeatable loop for reducing real exposure.

Crawl

At this stage, security teams have visibility into vulnerabilities but limited understanding of which ones actually matter. Testing is often periodic and prioritization is driven by severity scores, compliance requirements, or the loudest stakeholder.

The first step is to establish the CTEM program loop: scope what matters, discover exposure, validate what can be exploited, prioritize based on impact, and begin verifying fixes.

Walk

At this stage, organizations begin executing the loop with more consistency. Security teams gain a clearer understanding of exploitability, how weaknesses get combined into attack paths, and their business impact. Prioritization improves because decisions are increasingly grounded in evidence instead of assumptions.

Here, progress becomes easier to measure, but execution is still uneven across teams, environments, and attack surfaces.

Run

At this stage, CTEM becomes continuous, validation is embedded into security workflows, remediation efforts are prioritized based on exploitability and impact, and fixes are continuously verified.

Most importantly, organizations can demonstrate that exposure is being reduced over time. The focus shifts from managing findings to operating a repeatable loop that continuously reduces attacker opportunity.

From visibility to control

Most security programs weren't built for the environment we're operating in today.

They were designed for a world where infrastructure was more predictable, changes happened less frequently, and periodic testing was enough to maintain confidence. That model doesn't hold anymore.

Today, environments are more complex and constantly changing. New systems are deployed, identities change, applications evolve, third-party connections expand, and new exposures emerge every day. Attackers don't wait for your next scan or your next report. They work with what's available, chaining together weaknesses that, on their own, didn't seem risky.

That's the gap Continuous Threat Exposure Management is meant to close. Not by introducing another framework, but by changing how security is executed day to day. It moves teams away from managing lists of findings and toward understanding how those findings translate into real risk. That's what this playbook is about. Not building a perfect program. Not chasing complete coverage. But putting a system in place that you can run consistently, with enough clarity to know whether you're making progress.

Because the question security leaders are being asked is changing. It's no longer just about how many vulnerabilities exist or how quickly they're patched. It's about understanding what would actually happen if someone tried to break in today, and whether the organization is in a better position than yesterday. CTEM, when it's operationalized, gives you a way to answer that question with confidence. And over time, it gives you the ability to change that answer.

If you're looking to operationalize CTEM in your environment, the key is connecting these steps into a repeatable process.

See how teams are putting this into practice with an integrated approach.

Learn more