



Is your CTEM program ready for the AI era?

What AI changed on offense, where each stage of a CTEM program breaks, and what each stage needs now. For CISOs and security architects. October 2026.

Summary

- AI has changed the economics of offense. Finding a vulnerability, writing a working exploit and chaining it into an attack path now take hours and very little money, and much of that work runs without a person in the loop.
- Most CTEM programs still depend on people for analysis at every stage: scoping, triage, exploitability, prioritization and remediation planning. Faster tools speed that work up, but people remain the limit.
- Each stage needs facts that are produced continuously and checked against your real environment: whether a finding is exploitable on this resource right now, which attack paths it sits on, which changes would close those paths, and proof that they did.
- A program built on assisted analysts keeps human throughput as its ceiling against attackers who no longer have one. Adding analysts or faster tools does not close that gap. Moving the analysis itself into continuous, verified intelligence does.

What changed on offense

The attacker's costs have fallen at every step.

- **More vulnerabilities.** More than 86,000 CVEs were published in 2026 by late October, against about 48,000 in all of 2025. AI is surfacing old bugs in batches: one Firefox release fixed 271 bugs found by a single AI model, and the Linux kernel security list went from two or three reports a week to five to ten a day.

- **Smarter chaining.** In the GTG-1002 campaign that Anthropic disclosed in November 2025, AI ran an estimated 80 to 90 percent of the operation on its own, with people stepping in at a handful of decision points.
- **Faster execution.** The fastest observed breakout from initial access to lateral movement is 27 seconds (CrowdStrike 2026 Global Threat Report). Mandiant's average time to exploit fell from 32 days in 2021 and 2022 to 5 days in 2023; in 2024, exploitation came on average a day before a patch was available. Exploitation is now the most common way into a breach, at 31 percent of initial access (Verizon DBIR 2026).
- **Cheaper exploits.** Anthropic reported that Claude Mythos Preview wrote a working Linux kernel exploit in about half a day, for under \$1,000 (April 2026). Google's threat intelligence group has since reported the first zero-day exploit it assesses, with high confidence, was developed with AI.

"Current models are already better vulnerability researchers than I am, and in a year, they will likely be better than everyone." (Nicholas Carlini, Anthropic, [un]prompted 2026)

The industry's first response, faster patching, more AI tooling and broader discovery, is necessary. It is not enough on its own. Nearly every CTEM program to date uses people as the analysis layer: analysts triage scanner output, teams judge exploitability, engineers pick the remediation. Making each step faster does not remove the step.

Five stages, one broken assumption

Gartner's CTEM framework has five stages: scoping, discovery, prioritization, validation and mobilization. Each one assumes that people will do the analysis.

Scoping decides what the program covers and what matters to the business. In most programs it is a workshop and a spreadsheet, revisited once a year. Cloud environments change daily, so the scope is out of date soon after it is written. Scope needs to be a live model of your environment and its business context, updated as the environment changes.

Discovery works. Scanners find CVEs by matching package versions on every resource, and their output grows with CVE volume multiplied by the size of your estate. The problem is everything after discovery: triage, correlation and analysis were sized for a fraction of today's volume. Spending more on faster discovery solves a problem that is already automated.

Prioritization runs on proxies that are failing at the same time. - CVSS scores theoretical severity with no knowledge of your environment. FIRST measured that a policy of fixing everything rated 7 or above spends 57 percent of remediation effort to cover 82 percent of exploited CVEs, and only about 4 percent of that effort lands on CVEs that were ever exploited. - EPSS estimates the global probability of exploitation, not whether the conditions are met on your resource. - KEV worked as a late filter because working exploits were expensive to build. When AI makes an exploit cheap for almost any CVE, "exploited in the wild" stops separating the dangerous from the rest, just as volume peaks. About 31 percent of the last decade's CVEs already have public proof-of-concept code (VulnCheck), and AI is closing the gap for the rest. - Runtime context from cloud security tools (is the package loaded, is it reachable) works at the package level, not at the level of the conditions a CVE needs. A package can be loaded and reachable while the CVE's prerequisites are still not met, and those findings are the hardest false positives to clear.

What prioritization needs is a verdict for your environment: is this CVE exploitable on this resource, given its actual code, configuration, deployment, network and runtime state? A global score cannot answer that.

Validation has four common approaches, and none gives continuous coverage at CVE scale. - Breach and attack simulation tests whether defenses detect techniques. It is useful, but it fires attacks, so it runs on a schedule and on a limited scope. - Manual penetration testing is definitive but happens once or twice a year. - Automated penetration testing runs more often, but to avoid production risk it usually targets staging or a copy of production, and drift between the two weakens the result. - Simulation and digital-twin models reflect how the environment was configured, not how it is running.

None of them answers, continuously and without touching production: are the prerequisites for this CVE met on this resource right now, and is the path to it really open?

Mobilization defaults to patching because the information needed to do anything else is not in the workflow, and patching is slow. The median time to fix vulnerabilities on CISA's exploited list rose to 43 days (Verizon DBIR 2026). In application security, the median organization fixes 16.8 percent of its static-analysis findings, and findings older than 90 days are much less likely to ever be fixed (Semgrep, 2026). When AI-speed discovery adds findings faster than patching can close them, the backlog becomes permanent. Patching is also often the most expensive option, and it is chosen without looking at the alternatives.

Every exploitable CVE sits on an attack path, and the whole path is where you can act. Changes on the way in (network routes, security group rules, access policies) can remove an attacker's ability to reach the CVE. Changes on the way out (IAM scope, data store permissions, segmentation) limit

what an attacker gains. One network control that sits on 30 attack paths across 15 CVEs is worth far more than 15 separate patch tickets, and a ticket-per-CVE workflow never sees that.

Assisted analysts or a continuous intelligence layer

There are two ways to apply AI to a CTEM program, and they lead to different places.

AI-assisted analysis. Analysts get AI summaries and suggestions, and each step runs faster. The structure does not change: people still do the analysis, and their throughput is still the limit.

A continuous intelligence layer. Verification, attack-path analysis, risk ranking and mitigation options are produced continuously in the background. People and their agents start from decisions that are already analyzed and backed by evidence. Human judgment moves from doing the analysis to making the call.

This does not mean AI everywhere. AI earns its place where it turns unstructured advisories and research into structured exploitation conditions. Verdicts, checks and path scoring should be deterministic, because they need to be correct and repeatable, not creative. AI where intelligence is generated, deterministic logic where it must be trusted.

How to modernize each stage

Each stage needs something different. Here is what each one needs, and the use cases that deliver it with Defendermate: run today by skills in your own agents, and soon by Defendermate's agents ([join early access](#)).

Scoping and discovery

What it needs: a live, read-only model of your cloud and your existing findings, with business context, that updates as the environment changes. Your current scanners and tools stay; their findings are deduplicated per resource and joined to that model, rather than adding another scanner.

- **Unified findings.** For security and cloud engineers. One view of every finding, deduplicated per resource and joined to a live model of your environment. Done when you connect.

Prioritization

What it needs: environment-specific exploitability verdicts, ranked by real attack paths to what matters.

- **Exploitability triage.** For vulnerability management analysts and security engineers. Within minutes: whether a CVE is exploitable in your environment, which assets it puts at risk, and the evidence. Done by skills in any agent.

Validation

What it needs: proof on production at the scope you choose (a CVE, a resource or a full path) without firing anything, with evidence for each step.

- **Safe attack path validation.** For red teams and vulnerability management leads. Prove whether a path is really open before engineering time goes into it, as often as you need. Done by the [exploitability validation agent](#), coming soon.
- **Adversary-informed modeling.** For red teams and threat intelligence teams. Add the techniques that matter to you, and every path and ranking updates to match. Done by the [attack technique agent](#), coming soon.

Mobilization

What it needs: mitigations delivered as code, a clear answer on whether an available fix is complete and safe, changes made under your own policy, and re-verification that each path is closed.

- **Exposure mitigation.** For cloud security engineers. Mitigations as code, scoped to what makes the CVE exploitable, deployed under your policy. Done by the [exposure mitigation agent](#), coming soon.
- **Remediation orchestration.** For platform teams and SREs. Know when a safe fix exists, roll it out with rollback, and prove the path closed. Done by the [patch management agent](#), coming soon.
- **Detection coverage.** For detection engineers. See which detections would fire on your real attack paths, and close the gaps. Done by the [detection engineering agent](#), coming soon.

Across the program

- **Continuous compliance.** For GRC and compliance engineers. Control evidence that stays current, with SLAs measured on what is exploitable. Supports FedRAMP continuous

monitoring and PCI DSS work. Done by the [control monitoring agent](#), coming soon.

- **Evidence-backed exceptions.** For GRC and compliance engineers. Deviation requests and risk acceptances drafted with verified proof attached. Done by the [exceptions agent](#), coming soon.
- **Ask your own AI.** For anyone triaging exposure. Ask Claude, ChatGPT or Copilot "is this CVE exploitable here?" over MCP, and get answers with evidence. Done by any agent, over MCP. See [Built for agents](#).

Two more properties make this workable for a real team.

Your agents, your workflow. Security, cloud, SRE and IT teams already work with AI assistants. The intelligence layer should be open to the agents they use or build, not only to one vendor's interface, so each team can run the stages its own way.

A policy gate on every change. When agents act on production, every change should pass a policy you set, and every action should leave an audit trail of what was verified, when, and what was decided.

Defendermate connects once, read-only, then analyzes, plans and acts in a loop, with every change under your policy and re-verified. [Try it free](#) or [talk to us](#).

Three pressures that will not wait

Remediation deadlines now follow exploitability, and they are measured in days. In June 2026, CISA's Binding Operational Directive 26-04 replaced the federal known-exploited and internet-facing remediation directives with deadlines set by risk: whether the asset is exposed, whether the vulnerability is known to be exploited, whether exploitation can be automated, and how much control it gives an attacker. The tightest tier is three days, with a check for compromise. CISA's reason: attackers' "use of AI may further narrow the time defenders have to react between patch release and possible exploitation." The directive binds federal agencies, and CISA urges everyone else to follow it. A deadline set by exploitability needs exploitability facts about your own environment, not a severity score.

AI is producing a patch wave that capacity alone cannot absorb. The UK's National Cyber Security Centre has told organizations to prepare for a "vulnerability patch wave": put perimeter technology first, update by default, and prioritize by risk. Remediation is already falling behind. In the 2026 Verizon DBIR, only 26 percent of known-exploited vulnerabilities were fully remediated, down from 38 percent, and the median time to fully patch them rose to 43 days, from 32. More people will

not close that gap. Knowing which fixes matter on your estate, and which exposures you can contain before a patch ships, will.

Insurers price what they can see, and policy terms now penalize old, unpatched CVEs. Carriers scan policyholders from the outside; Coalition tells brokers that critical findings left open can lead to contingencies that must be resolved before renewal. Chubb's Neglected Software Exploit endorsement, where it applies, shifts more of the loss to the policyholder, in steps, for vulnerabilities left unpatched more than 45 days after publication. At-Bay's 2026 claims report traces 87 percent of ransomware claims to remote access. A record of what was verified exploitable, when, and what was done about it is the evidence you will want at renewal and after an incident.

Sources

- NVD, CVEs published January 1 to October 25, 2026 and in 2025, excluding rejected (queried October 25, 2026)
- Mozilla, Firefox 150 release, April 21, 2026 (271 bugs found by Claude Mythos Preview)
- Willy Tarreau on Linux kernel security reports, via LWN, March 2026
- Anthropic, disclosure of the GTG-1002 campaign, November 2025
- CrowdStrike, 2026 Global Threat Report
- Mandiant, "How Low Can You Go? An Analysis of 2023 Time-to-Exploit Trends", October 2024; Google Cloud, M-Trends 2026
- Verizon, 2026 Data Breach Investigations Report, May 2026
- Anthropic, Claude Mythos Preview, April 2026
- Google Threat Intelligence Group, "Adversaries Leverage AI for Vulnerability Exploitation, Augmented Operations, and Initial Access", May 11, 2026
- Nicholas Carlini, "Black-hat LLMs", [un]prompted 2026
- FIRST, EPSS remediation-strategy measurement, October 2023
- VulnCheck, State of Exploitation: A Decade
- Semgrep, Remediation at Scale, 2026
- CISA, Binding Operational Directive 26-04, "Prioritizing Security Updates Based on Risk", June 10, 2026
- UK National Cyber Security Centre, Ollie Whitehouse, "Preparing for a 'vulnerability patch wave'", May 1, 2026
- Coalition Help Center, "Accessing mid-term security alerts in the Broker Platform"
- Chubb, Cyber Insurance Products, Neglected Software Exploit endorsement
- At-Bay, 2026 InsurSec Report, April 22, 2026