terra

Buyer's Guide • June 2026

# Agentic AI-Powered Pentesting

## A Security Leader's Framework for Evaluating Continuous, AI-Powered Pentesting Platforms

# Table of Contents



# Executive Summary

The pentesting market has reached an inflection point. Security leaders now face a structural problem that traditional penetration testing was never built to solve: fast-moving development teams adopting AI and shipping code continuously are dramatically expanding the attack surface.

31% of all breaches now start with software vulnerabilities, making them the leading entry point for attackers into enterprise systems. CISA's Known Exploited Vulnerabilities catalog reflects this acceleration: in 2025, CISA added 245 vulnerabilities to the catalog, a more than 30% increase over the prior two years' trend.

The window to act is closing, yet most enterprise pentesting programs still operate on a quarterly or annual cadence, producing point-in-time reports that begin aging the moment they are printed. The compounding gap between how fast environments change and how often they are validated forces security teams into one of two losing positions: move slowly and test deeply, or move quickly and test shallowly.

A second structural problem compounds the first. Most enterprise security programs treat the attack surface as three separate programs: network infrastructure tested by one vendor or team, web applications tested by another, and AI systems either left untested or handled through a separate red teaming engagement. The result is a web of fragmented vendor outputs, siloed findings that cannot be correlated across surfaces, and remediation delays that leave the most dangerous attack paths invisible. Attackers do not respect those boundaries. They chain weaknesses across network infrastructure, applications, and AI systems to move laterally and maximize impact. A pentesting program organized around how vendors sell, rather than how attackers operate, will always miss the chains that matter most.



 

AI-powered continuous pentesting was built to resolve this tradeoff. But as with any rapidly growing category, the market includes solutions ranging from impressive innovations to thinly relabeled tools. This guide is designed to help security leaders cut through the noise and understand what the category actually means, what separates credible platforms from surface-level automation, and what questions to ask before committing budget.

# What is Agentic AI-Powered Pentesting?

An agentic AI pentesting system can reason within an environment, discover attack surfaces, form hypotheses about exploitable paths, execute multi-step test sequences, evaluate results, and adjust its approach as it learns more about the target. Agentic systems can discover novel vulnerabilities, chain multiple smaller weaknesses into a full exploit path, test business logic, and adapt to application-specific behaviors not captured in any signature database.

# Approaches to AI-Powered Pentesting

| Dimension | Fully Autonomous AI Pentesting | Agentic AI-Powered with a Human-in-the-Loop |
|---|---|---|
| **Operating Model** | AI agents run without human involvement | AI agents run continuously, with human oversight built in for edge cases |
| **Core Appeal** | Continuous coverage, fast deployment, low-friction | Continuous coverage with expert validation, fast and deep deployment, and higher-trust output |
| **Finding Quality** | High volume output, high FP rate, and findings often need internal triage | High efficacy and accuracy findings,, no internal triage needed |
| **Business Context** | Struggles with workflow logic and org-specific nuance | Fully contextualized with the ability for humans to add context to the surface and prioritize what matters most |
| **Production Safety** | Unsafe for production environments | Human approvals for guardrailed actions ensure safety in production environments |
| **Assurance Model** | Maximum autonomy, minimal accountability | AI scale and speed, anchored by human judgment and compliance |
| **End Result** | Fast output, but noisy, risky, and more work for your team | Verified, context-aware findings that hold up to engineering and audit scrutiny |


 

# The Three Attack Surfaces Most Programs Treat in Isolation

Most enterprise pentesting programs are fragmented. Network infrastructure is assessed by a single team or vendor. Web and internal applications get tested by another. AI systems, if they get tested at all, are handled through a separate red-teaming engagement with its own scope, findings format, and reporting cycle. The result is three disconnected programs producing three sets of findings that no one is correlating.

That separation is becoming harder to manage as attackers move across multiple parts of an organization's attack surface during the same intrusion. A misconfiguration on a network service becomes the entry point. A chained authorization flaw in a web application becomes the escalation path. A prompt injection against an AI copilot with access to internal systems becomes the exfiltration method. None of those findings, in isolation, tells the full story.

## Web and Internal Applications

This is the surface that most agentic pentesting programs cover, but the quality of coverage varies significantly. Authenticated testing, multi-step workflow logic, business context, and API behavior require more than just AI agents. Sophisticated adversaries target the logic that connects systems, not just the endpoints those systems expose.

## External Network Infrastructure

Exposed services, misconfigured firewalls, unpatched network devices, and lateral movement paths from the perimeter inward remain a primary initial access vector. Network infrastructure testing is frequently siloed from application testing, so a finding at the network layer is never correlated with one at the application layer to reveal a combined exploit path. AI agents that can map exposed services, chain misconfigurations, and validate exploitability across the network environment close that gap.

## AI Systems and Agentic Workflows

AI assistants, copilots, agentic workflows, and LLM integrations have become embedded in office productivity suites, CRMs, code development environments, and customer-facing applications, all with access to the same inboxes, document repositories, databases, and internal systems your employees use.



 

Prompt injection, in which an attacker embeds malicious instructions into content that an AI system ingests and acts on, is a documented, production-scale risk. Real incidents have already demonstrated the consequences: CVE-2026-25724, discovered by Terra Security researchers, exposed a permission-control bypass vulnerability in Anthropic's Claude Code, leading to documented data exposure. A successful prompt injection into an agentic workflow results in unauthorized actions, not just leaked information.

Evaluating whether any platform formally tests AI-enabled applications, including prompt injection attack paths, jailbreak attempts, data exfiltration through AI workflows, and agent behavior manipulation, is now a baseline requirement for any enterprise security validation program.

# 8 Questions Every Buyer Should Ask When Evaluating AI-Powered Pentesting Platforms

These questions are designed to help you evaluate platforms in the market and build the internal case for what any modern pentesting program needs to deliver. They follow a deliberate sequence, where each question assumes you got a satisfactory answer to the one before it. A platform that cannot answer them clearly and specifically likely cannot deliver on what it promises.

## Question 1: How does the platform stay current with my environment as code ships and infrastructure changes?

A pentesting program that resets to zero with each engagement, re-mapping your attack surface from scratch, is not continuous. If a platform cannot maintain persistent context across testing cycles and automatically respond to changes, the window between engagements is an open invitation to adversaries.

| | |
|---|---|
| **Ask Specifically** | **Does coverage compound over time, or does each cycle restart from the same baseline? How does the platform detect and respond to a new deployment or a change in the authentication flow without you having to manually re-scope the engagement?** |

**What a strong answer looks like**: the platform maintains a living model of your environment that updates automatically as changes occur. Prior testing context informs new test cycles rather than


 

being discarded. When new code ships or a new endpoint appears, evaluation triggers without requiring a new scoping call.

## Question 2: Does the platform integrate with existing development and security workflows?

Most security teams have already invested in a toolchain: source control in GitHub or Azure DevOps, ticketing in Jira or ServiceNow, authentication through SSO and MFA, and CI/CD pipelines that merge and deploy code multiple times a day. A platform that sits outside that ecosystem creates operational friction that slows adoption and, in practice, means testing reverts to a periodic cadence regardless of what the vendor promises.

| | |
|---|---|
| **Ask Specifically** | **Which platforms does it integrate with natively, and what triggers a new test cycle? Does it push findings directly into your existing ticketing system, or does it require a separate workflow?** |

**What a strong answer looks like**: the platform has native integrations with major source control systems. Authentication is handled natively, so agents can test authenticated surfaces without a manual setup step each cycle. Findings route directly into the ticketing system your team already uses, with context attached so engineers can act without switching tools. The platform runs in parallel with your CI/CD pipeline and does not gate deployments.

## Question 3: Does the platform cover network, AI, and web attack surfaces, or test a single layer in isolation?

Most agentic pentesting platforms were built to focus on a single attack surface, leaving other attack surfaces to other tools, vendors, or separate red-teaming engagements. That model fragments your view of risk: each provider generates its own findings, formats, and reports, but no one is responsible for correlating them into the multi-step attack paths real adversaries execute. The result is an environment where a misconfigured network service, a web authorization flaw, and an AI prompt injection are each tested in isolation, even though attackers chain these weaknesses together to move laterally and exfiltrate data.

| | |
|---|---|
| **Ask Specifically** | **Does the platform chain vulnerabilities across multiple attack surfaces? Are findings from each surface correlated in a single view?** |



 

**What a strong answer looks like**: the vendor is explicit that the same platform continuously tests multiple layers, such as web applications, external network infrastructure, and AI systems, rather than limiting coverage to a single layer or requiring separate products and engagements for each. Findings from all covered surfaces appear in a unified view with shared context, audit trails, and remediation guidance oriented around breaking full attack chains, not just closing isolated issues on the one layer the tool happens to support best.

## Question 4: How does the platform find business logic vulnerabilities?

Vulnerabilities such as a minor authorization issue on one endpoint that chains into a critical path, when combined with a misconfiguration two steps downstream, are the ones sophisticated adversaries find and prioritize. They are also the ones most likely to represent real business impact rather than theoretical risk.

| | |
|---|---|
| Ask Specifically | Can you provide concrete examples of business logic findings from real-world environments? What does their methodology look like for stateful, multi-step testing across user roles and workflow sequences? |

**What a strong answer looks like**: the vendor walks through a specific, real example of a business logic finding, ideally one that required maintaining state across multiple user roles or endpoint interactions. They can explain the methodology used to produce it, not just describe the output.

## Question 5: Do all findings come with confirmed exploitability?

Many platforms surface theoretical risk without proving that any of these are actually exploitable in your specific environment. The result is a triage backlog that consumes engineering time without reducing actual exposure. Every hour a security engineer spends investigating a phantom vulnerability is an hour not spent on a real one.

| | |
|---|---|
| Ask Specifically | What is the platform's validation methodology? At what rate do findings survive validation before reaching your team? |


 

**What a strong answer looks like**: every critical finding includes a reproducible exploit path, demonstrated impact, and proof-of-concept steps your development team can verify without additional analysis. The vendor can describe the process and explain how unverified findings are filtered before they reach you.

## Question 6: What happens when an agent wants to take an intrusive action in your sensitive environment?

Fully autonomous agents follow exploit paths wherever they lead unless something stops them. An agent that autonomously triggers a destructive action because no guardrail existed, or because the guardrail was configurable rather than structural, is a liability.

| | |
|---|---|
| **Ask Specifically** | **How does the platform give our team full visibility and control over agent actions - including the ability to define scope, review commands before execution, and stop testing without causing disruption to production?** |

**What a strong answer looks like**: sensitive test chains require explicit human approval before execution. Rate limiting, scope enforcement, and prevention of destructive actions are built into the platform architecture. Every agent action is logged and auditable. The vendor can walk you through a specific example of an approval gate triggering.

## Question 7: How does the platform support remediation, and does it automatically perform retests?

The question now is whether the value stops at the finding or extends through to the fix. The most operationally valuable thing a pentesting platform can do is close the loop, not just open it.

| | |
|---|---|
| **Ask Specifically** | **Is remediation guidance specific to your architecture, framework, and technology stack, or is it generic? Does the platform automatically retest after a fix is deployed? Can you demonstrate what that retest workflow looks like?** |

**What a strong answer looks like**: remediation guidance is generated based on how the vulnerability manifests in the specific application, tailored to the team's actual tech stack. After a fix is deployed,


 

the platform automatically retests the attack path to confirm it is closed, not just that the vulnerability was addressed at the code level.

## Question 8: Are your reports accepted by compliance auditors without additional documentation or a separate engagement?

Compliance reviewers for highly regulated industries seek a qualified security professional to explain what was tested, how it was tested, and what was found. An automated report with no certified human sign-off raises immediate accountability questions that most compliance frameworks do not resolve in the vendor's favor.

| | |
|---|---|
| **Ask Specifically** | **Which frameworks are the reports accepted by? What makes that acceptance possible, meaning what human credentials and review process backs it?** |

**What a strong answer looks like**: reports are signed by certified pentesters and include a complete audit trail of every agent action, every human approval decision, and every validated finding.

# Terra Platform™: AI Scale with Expert Precision

Terra Platform is the Agentic Offensive Security platform that replaces point-in-time pentests with fully integrated continuous attack surface validation powered by a swarm of hundreds of trained AI agents paired with a Human-in-the-Loop.

Every meaningful change to code, configuration, or reachable attack surface triggers evaluation. Terra maintains full context across the testing lifecycle and proves what is actually exploitable as the environment evolves. Terra Platform™ is the only agentic Offensive Security platform to unify continuous pentesting across all three foundational attack surfaces under a single program:

**Web and Internal Application –** Terra's agents adapt to application workflows, authentication flows, and data handling to uncover critical vulnerabilities and chain multi-step exploit paths across user roles and endpoint interactions.

**External Network Infrastructure –** AI agents map exposed services, identify misconfigured network devices, and validate exploitability across your network environment, producing findings ranked by real business impact rather than raw severity scores.

**AI Red Teaming –** Terra tests your AI systems, copilots, agents, and LLM integrations against threats that prompt fuzzing cannot reach, including prompt injection, jailbreak attempts, unauthorized data exfiltration through AI workflows, and agent behavior manipulation.

All three surfaces are tested under the same agentic AI platform and Human-in-the-Loop model. Network, application, and AI findings appear in a single, connected view with a consistent audit trail. For security leaders currently running separate vendors or engagements for each surface, this means one platform, one scoping process, one finding format, and one compliance record.

## Terra's Approach Compared To Others

| | Manual Pentesting | Terra Platform™ | Fully Autonomous |
|---|---|---|---|
| **Testing Frequency** | Point-in-time Quarterly/annually | **Continuous** | Continuous |
| **Coverage** | Narrow limited by time/cost | **Wide and deep** | Wide but unvalidated |
| **False Positive Rate** | Varies | **Minimal** | High |
| **Business Logic Testing** | Yes | **Yes** | Varies |
| **Compliance Acceptance** | Accepted | **Accepted** | Partially accepted |
| **Production Safety** | High | **High** | Low |
| **Real-Time Findings** | No | **Yes** | Yes |
| **Cost Efficiency** | Low ROI | **High ROI** | Consumption-based |
| **Ease to Scale** | Low | **High** | High |
| **Human-in-the-Loop** | No | **Yes** | No |
| **Bring Your Own AI** | No | **Yes** | No |
| **On-Prem Support** | Yes | **Yes** | No |



 

# What Changes When You Run Terra

Before testing begins, human experts and AI agents establish the business and risk profile for your environment, evaluating authentication flows, business-logic dependencies, risk priorities, and organizational guardrails. These get embedded in each agent's operating parameters, so the agents know what they are working with before they start.

From that point forward, the coordinated swarm of AI agents handles what machines do best: comprehensive surface coverage across all three foundational attack surfaces, tireless execution across every attack vector, pattern recognition at scale, and reasoning across multi-step exploit chains that cross surface boundaries. The agents retain context from prior testing across all three surfaces, building a progressively more complete picture of exploitable risk as the environment evolves.

When agents encounter sensitive test paths, the platform routes decisions to a Human-in-the-Loop for safety protocol review before execution. This is structural to how Terra operates. This ensures production environments are never exposed to ungoverned destructive actions.

Every finding that reaches your team has passed through a double-validation layer and is confirmed exploitable. Your team receives a meaningful set of verified, exploitable findings, grounded in the business context. Remediation guidance is specific to your architecture and technology stack, and Terra automatically retests after fixes are deployed to confirm the attack path is actually closed.

# Total Cost of Ownership

The budget line item for traditional pentesting understates its true cost. Beyond the engagement fee, security and engineering teams absorb significant overhead that rarely appears in vendor comparisons: time spent re-scoping engagements, re-onboarding vendors, triaging unvalidated findings, and coordinating remediation without clear ownership.

A second layer of hidden cost compounds this for most enterprise security programs: the overhead of running three separate programs for application testing, network testing, and AI red teaming. Each requires its own vendor relationship, scoping and onboarding cycle, finding format, and compliance documentation. Security leaders who consolidate all three into a single Terra engagement report not only remove triage overhead but also eliminate the coordination burden of managing multiple offensive security vendors simultaneously. One platform, one audit trail, one remediation workflow.



 

> Terra's approach, combining their Agentic AI system with human oversight, gives the depth and scale a modern security organization needs in their pen test program while increasing accuracy and vulnerability validation.
>
> Yossi Yeshua
> *CISO, Riskified*

> Terra's agentic pentesting boosted our ROI by over 100%. Every dollar saved on repetitive tasks went straight into deeper, quality testing based on our business context.
>
> Kyle Kurdziolek
> *VP Security, BigID*

These are real engineering hours redirected away from productive security work, and real exposure windows that remain open while the administrative cycle completes. Terra customers report over 4x the value for money in half the time compared to traditional pentesting. That figure reflects what happens when triage overhead is removed from the equation.

With Terra, findings arrive validated and actionable, remediation guidance is specific to the team's actual stack, and retesting is automatic. The budget and time previously consumed by the process are redirected toward the deeper, continuous security validation the environment actually requires.

## Compliance as an Outcome

Every Terra report is signed by a certified pentester, with a complete audit trail of every agent action, every human approval decision, and every validated finding. The documentation required by auditors is generated automatically throughout the testing lifecycle, rather than assembled retroactively.

For security leaders building the internal case for program investment, this distinction matters. A pentesting program that cannot produce auditor-accepted documentation requires a separate reporting engagement, additional vendor spend, and extra coordination overhead whenever a compliance review occurs.

## The New Standard for Pentesting

Traditional point-in-time pentesting was built for a slower era. Fully autonomous AI testing offers speed, but without the validation, business context, and accountability that modern security programs require. The standard Terra set is continuous, agentic AI-powered security validation governed by an expert Human-in-the-Loop.



 

For security leaders, the selection criteria should be straightforward: choose the platform that covers web applications, network infrastructure, and AI systems under a single continuous program, produces findings that are correlated across all three surfaces, generates reports auditors will accept, and triggers evaluation the moment new code ships or the attack surface changes.

That is the standard.

Any platform that requires separate vendors, separate engagements, or separate audit trails for any of those three surfaces does not meet it.

# About Terra Security

Terra Security provides agentic AI-powered continuous penetration testing aligned to code changes and evolving attack surfaces, combining a swarm of trained AI agents with human supervision for safety and control. The company works with Fortune 500 organizations to ensure every attack surface is covered across the web, AI, internal apps, APIs, mobile, networks, and the cloud. For more information, visit terra.security.



 