Novee

The Definitive Buyer's Guide to AI Penetration Testing

The 8 Questions to Ask when Evaluating AI Pentesters in 2026

How to Use this Guide

This in-depth guide explains why AI penetration testing is top-of-mind for every CISO in 2026, and outlines key criteria for evaluating and selecting the right pentesting platform for your organization.

Inside, you'll find:

INTRODUCTION TO

AI Penetration Testing

Why AI Pentesting Matters Right Now to Security Teams

Security testing is under structural pressure, and industry practitioners are ready for a shift.

Over the past decade, software delivery has accelerated dramatically. CI/CD pipelines push code to production daily, or even sometimes hourly. Microservices multiply APIs, SaaS integrations expand trust boundaries, and AI coding assistants generate new functionality faster than security teams can manually review.

At the same time, attackers have adopted automation and AI. What once required highly skilled nation-state operators can now be partially replicated with intelligent tooling, and attacks from AI-equipped adversaries are on the rise.

While reducing the attack surface is still a critical component of security, the problem has moved beyond depth and breadth, and beyond signal-detection. According to the Zero Day Clock, the average TTE (time-to-exploit) of new vulnerabilities is 1.6 days*. Back in 2018, it was over 2 years.

Unlike quarterly security audits, attackers don't come knocking on a regular schedule. They probe continuously, spontaneously, and erratically.

Your environment changes constantly. Your adversaries probe constantly.
But your security validation is either ***too slow*** or ***too shallow.***

**Source: https://zerodayclock.com/*

Traditional penetration testing was built for a slower world: a team is scoped, access is provisioned, testing runs for weeks, and a static report is delivered. The moment the report is finalized, it begins to age.

On the flip side, automated scanners offer continuous coverage, but at the cost of shallow reasoning. They match signatures and flag known CVEs without understanding your business logic, context, or exploitability. Attackers today are "living off the land" – executing attacks within live, running systems.

It's no longer just about where you can be hit, but how *fast* you can get hit.

The result is a dangerous tradeoff:

Depth without scale
(manual pentesting)

vs

Scale without adversarial reasoning
(automated scanners)

Rather than resign themselves to this losing tradeoff, what security leaders actually need is **continuous adversarial validation**: offensive security that reasons like a real attacker and operates at machine speed.

Time is the enemy, so time-to-fix is the metric that matters; and only AI penetration testing can deliver deep, broad continuous coverage at speed.

That's why HackerOne cited: **67%** of hackers on their offensive security platform as having used AI to augment their work*

It's why: **34%** of survey respondents on Latio's 2026 AppSec Report listed "AI Pentesting" as the feature they're most excited about – the most of any category.**

And why Gartner reports that by 2028, more than: **60%** of enterprise penetration test programs will operate as continuous validation, replacing annual assessments as the primary proof of resilience.***

*Source: https://www.hackerone.com/blog/ai-security-trends-2025
**Source: https://www.latio.com/reports/2026-latio-application-security-report
***Source: The Future of Pen Testing Is Continuous Offensive Security Testing (Dhivya Poole, Carlos De Sola Caraballo, Mitchell Schneider, 6 March 2026, ID G00845606)

That is the promise of AI penetration testing. But not all AI pentesting platforms are built the same.

Before entrusting a platform to show you what attackers already know, it's important to understand what baseline technical competencies define this category.

WHAT EVERY

AI Pentesting Vendor Should Be Able to Do

1 Autonomous Surface Discovery

The system should be able to:

- Map exposed domains and subdomains
- Discover APIs and endpoints
- Identify authentication flows
- Enumerate user roles

! If a platform cannot independently explore your environment, it is not performing adversarial validation.

2 Multi-Step Attack Execution

Real-world exploitation often requires chaining vulnerabilities together.
The system should:

- Maintain state across sessions
- Execute multi-step workflows
- Adapt based on system feedback

! Single-step vulnerability detection is insufficient.

3 Exploit Validation

Findings should be:

- Reproducible
- Demonstrably exploitable
- Accompanied by proof-of-concept steps

! A list of potential weaknesses without validation is not offensive testing.

4 Safe Testing Controls

An enterprise-grade platform must:

- Avoid destructive payloads
- Prevent real data exfiltration
- Maintain clear audit trails
- Operate under strict authorization controls

! Security validation should reduce risk, not introduce it.

5 Actionable Reporting

At minimum, findings should include:

- Severity and impact
- Reproduction steps

! Without this, engineering teams cannot act efficiently.

EIGHT QUESTIONS

To Ask Your AI Pentesting Vendor

Below are **eight architectural questions to ask your AI penetration testing vendor**, to help you evaluate the best solution for your organization.

The right responses will separate surface-level automation and "AI-enhanced" manual penetration testing from true machine intelligence – the kind that can continuously uncover novel vulnerabilities, business logic flaws, and chained attack paths, using nothing but the same resources your attackers would have.

For ease, we've included the core capabilities, as outlined in Section 1, relevant to each question.

QUESTION 1 | Is This a Proprietary Model or an LLM Wrapper?

Core Capabilities on display:

1. Autonomous Surface Discovery
2. Multi-Step Attack Execution
3. Exploit Validation

Ask your vendor:

"Are you running a purpose-trained offensive security model, or orchestrating a frontier LLM?"

Many vendors rely heavily on frontier models trained for broad language tasks. While powerful, these models were not optimized for adversarial system interaction.

Offensive security is a stateful reasoning problem grounded in real environments: not something a textual predictive model can easily solve.

What to look for:

A purpose-built model trained on full attack trajectories — including failed attempts — reinforced by learning techniques that replicate how elite hackers probe, adapt, and eventually succeed.
This enables:

- Iterative hypothesis generation.
- Adaptive retry strategies.
- Stateful reasoning across multi-step workflows.
- Exploit chaining.

Bonus: Frontier LLMs are now offering their own built-in bug-hunters (think: Claude Code Security). But their capabilities are limited to scanning code produced by their own model. You want a flexible, versatile

model that can understand more than just code, and analyze live, running systems in production.

In short, **you want an AI (or team of AI agents) that mimics a team of elite hackers; a unified force of specialized agents, working in tandem to trace, resolve, and bypass exploitable source-to-sink chains.**

QUESTION 2 | Does Your Platform Continuously Learn And Build Adaptive, Persistent Intelligence about my Application?

Core Capabilities on display:

1. Autonomous Surface Discovery
2. Multi-Step Attack Execution
3. Exploit Validation

Ask your vendor:

"Does your platform maintain a living model of my application for repeated runs at exploit chains?"

Modern applications behave differently every time you test them; a vulnerability may only appear when a certain feature flag is enabled, or two roles interact in specific sequence, or even under specific load conditions. Therefore, real attackers must rely on more than intuition. They spend time in living, running systems, picking up a feel for how they work and leveraging historical builds to probe for weak points. Your AI penetration tester needs to be able to do the same.

What to look for:

- A dedicated memory layer that parses context from an agentic application mapper.
- Compounding coverage and understanding that deepens the longer the system runs.
- An application knowledge base (AKB) that includes every component of the application, capturing business logic and writing it to persistent memory.

Generic large language models lack systematic penetration testing methodology. They cannot explore application state spaces strategically, with persistent memory, and miss real exploitable flaws. That's where custom-built AI penetration tools stand out; they mimic how real attackers operate, but continuously.

QUESTION 3 | Is This "Humans-in-the-Loop" or True, Autonomous Continuous Attacker-Grade Reasoning?

Core Capabilities on display:

1. Autonomous Surface Discovery
2. Multi-Step Attack Execution
3. Exploit Validation

Ask your vendor:

"Is your system designed to operate continuously and autonomously as environments evolve?"

Traditional pentesting is episodic. Even some AI solutions merely automate the quarterly model. And "AI-enabled" penetration testing that requires humans in the loop is subject to the same periodic constraints

as fully manual penetration testing.

What to look for:

As code ships and environments change, so too do exploitable vulnerabilities. Get a solution that continuously:

- Re-evaluates attack paths.
- Generates new hypotheses.
- Tests for newly introduced weaknesses.

Security posture should reflect reality right now — not last quarter. Continuous validation aligns testing cadence with development velocity.

QUESTION 4 | Can it Autonomously Execute Multi-Step Exploit Chains and Address the AI-Enabled Attack Surface?

Core Capabilities on display:

1. Autonomous Surface Discovery

Ask your vendor:

"Can your system autonomously find multi-step vulnerabilities in-production, escalating severity across multiple endpoints? And does it take AI-enabled apps into account when reconnoitering the attack surface?"

Real attackers rarely exploit a single flaw in isolation. They:

- Combine access control weaknesses.
- Chain IDORs with privilege escalation.
- Abuse business logic across multiple endpoints.
- Escalate through stateful workflows.
- **Run attacks against AI-enabled applications.**
  - Think prompt injection, jailbreak attempts, data exfiltration, adversarial prompt generation, and manipulation of agent behavior.

Many automated tools detect individual vulnerabilities but cannot reason across steps. Others do not take AI tools into account.

What to look for:

A platform and model that maintains persistent context, enabling it to:

- Track authentication states.
- Switch user roles.
- Traverse multi-endpoint workflows.
- Chain vulnerabilities dynamically.

A complete chain of exploits is proof of verified, exploitable reachability – far more useful and actionable than a potential in-point flagged by a basic automated scanner.

And on the topic of exploitability:

QUESTION 5 | Do You Validate Exploitability or Just Report Risk?

Core Capabilities on display:

1. Exploit Validation
2. Actionable Reporting

Ask your vendor:

"Do you generate a working proof-of-concept for each critical finding?"

The industry suffers from alert fatigue. Many tools surface theoretical risk – known CVEs, configuration weaknesses, or pattern matches – without proving exploitability.

What to look for:

Validated, exploitable findings: critical vulnerabilities with:

- A reproducible exploit path.
- Demonstrated impact.
- Clear replication steps.

Always front-of-mind should be a mindset shift away from "potential risk" to "provable exploitation." There should be no theoretical findings and no false positives by design.

And design is key; namely, the intended methodology behind the design of the product:

QUESTION 6 | Can You Produce Meaningful Results Quickly – and show Full Coverage – Without Accessing Source Code?

Core Capabilities on display:

1. Exploit Validation
2. Safe Testing Controls
3. Actionable Reporting

Ask your vendor:

"Can you begin testing without my crown jewels, and get me actionable results fast?"

Real attackers most often don't receive privileged access; they start from zero knowledge, and work the problem as an outsider would.

> **A quick primer on penetration testing methodologies:**
> The level of upfront work to begin "AI Pentesting" depends on the type of solution. Penetration testing traditionally falls into three categories:
>
> **White Box:** The most transparent form of pentesting, to give pentesters every opportunity to leverage as many attack vectors as possible. You will be required to share full network credentials and system information.
>
> **Gray Box:** Limited information sharing, to give pentesters insight into severity of risk based on user access level. Usually involves sharing login credentials.
>
> **Black Box:** No information is provided at all, except for a domain name. The pentester will take care of the rest.

What to look for:

Testing should be able to meet real-world adversarial conditions. Best-in-class pentesters offer organizations the option to later expand into gray-box or white-box testing, but black-box capability demonstrates genuine reasoning depth.

This is about bringing offensive security results to board-level. You need actionable and clear **defensible coverage**; showing what was tested, not just what was found.

QUESTION 7 | Do You Close the Loop With Personalized Remediation and Retesting?

Core Capabilities on display:

1. Exploit Validation
2. Safe Testing Controls
3. Actionable Reporting

Ask your vendor:

"Are remediation steps tailored to my architecture? And do you automatically retest fixes?"

Many vendors stop at detection. They provide generic OWASP references and leave remediation to engineering teams.

What to look for:

A solution that integrates attack and defense in one loop:

- Findings include environment-specific remediation.
- Recommendations align with actual tech stack and configuration.
- After fixes are deployed, the system automatically retests.

The most in-depth threat report in the world is useless if it doesn't tell you how to fix the problem, and who should be assigned to fix it. This remediation guidance eliminates friction between security and engineering and accelerates risk reduction.

QUESTION 8 | Is This Actually Safe for my Production Environment?

Core Capabilities on display:

1. Safe Testing Controls

Ask your vendor:

"Is your platform safe to continuously run in production – does it get privileged access without guardrails?"

Running offensive operations of any kind demands strict controls.

What to look for:

A solution that enforces constraints at every level:

- Task-specific AI agents that are siloed in their capabilities.
- Customizable, configurable guardrails: rate-limiting, time-zone.

- restrictions, URL exclusion lists, and explicit destructive action prevention.
- A pre-test plan review offering the ability to audit and approve every test case before offensive action is taken.

The goal of any AI penetration testing solution should be to simulate the potentially harmful effects of an attack – not accidentally cause them.

IN SUMMARY...

Security testing is in need of a lateral thinking shift, and agentic AI penetration testing is the path forward. But only if it delivers:

1. Purpose-trained adversarial reasoning.
2. Multi-step exploit chaining.
3. Verified exploitability.
4. True black-box capability.
5. Continuous validation.
6. Personalized remediation with retesting.

Read on to find out how Novee does it.

NEXT STEPS:

Introducing Novee, the Leader in AI Penetration Testing

How Novee works, and why it leads the AI Pentesting category

Novee is the continuous offensive security platform that attacks your application the way real adversaries do, and feeds those discoveries directly back into defense.

It starts with true black-box testing and continuously uncovers novel vulnerabilities, business logic flaws, and chained attack paths. Built by veteran offensive security operators, Novee layers expert knowledge, application-specific context, agentic tooling, and a rigorous validation system on top of frontier models. The result is an AI penetration tester that adapts as your environment evolves, getting smarter over time. Every issue is validated and paired with precise, personalized fixes tailored to your architecture, tech stack, and business logic – so teams can reduce real risk as fast as attackers create it.

Our eight questions above are designed to help you determine whether an AI penetration testing platform can actually replicate attacker behavior, or whether it is simply automating existing scanning workflows.

Here's how Novee meets the criteria:

01

"Are you running a purpose-trained offensive security model, or orchestrating a frontier LLM?"

Rather than learning from static vulnerability descriptions, we built our own proprietary AI model, and trained it on full attack trajectories: sequences of reconnaissance, hypothesis generation, exploit attempts, environmental feedback, and iterative refinement.

In exploitation benchmarks involving real web attack scenarios, Novee's specialized model significantly outperformed frontier LLMs by over ~55% because it is optimized for stateful adversarial reasoning, rather than text prediction.

02

"Does your platform maintain a living model of my application for repeated runs at exploit chains?"

Traditional pentests restart from zero every engagement, but Novee builds persistent intelligence about your environment.

The system begins by autonomously mapping the attack surface – domains, APIs, endpoints, workflows, authentication states, and integrations – and then storing that information in a shared memory layer used by coordinated agents. New attacks build on previously discovered context; coverage compounds rather than resetting.

03

"Is your system designed to operate continuously and autonomously as environments evolve?"

Novee's architecture is designed for fully autonomous offensive reasoning.

A central AI orchestrator coordinates specialized sub-agents, allowing Novee to simulate the workflow of a human pentester, except running continuously, and exploring attack paths in parallel. Human experts remain valuable for strategic testing and complex edge cases, but the core offensive workflow runs autonomously.

04

“Can your system autonomously find multi-step vulnerabilities in-production, escalating severity across multiple endpoints?”

Novee’s reasoning model is specifically designed to discover and execute multi-step exploit chains, which combine small weaknesses across workflows, integrations, and access boundaries. We specialize in uncovering business logic flaws, stateful authorization bugs, chained injection vulnerabilities, and cross-system attack paths.

These are the classes of vulnerabilities most frequently missed by scanners, but often discovered by elite human pentesters.

05

“Do you generate a working proof-of-concept for each critical finding?”

Novee reports only validated vulnerabilities. Every finding includes:

- a working exploit or proof-of-concept
- reproduction steps
- evidence of real impact

Security teams receive a small set of high-confidence findings rather than thousands of alerts requiring manual triage.

06

“Can you begin testing without my crown jewels, and get me actionable results fast?”

Novee is the only AI Pentester on the market who can achieve true black-box pentesting: we don’t ask for your crown jewels up front. Given only a domain, the system autonomously performs infrastructure discovery, endpoint enumeration, API mapping, workflow reconstruction, and attack surface expansion.

Because deployment requires no integrations or source code access, organizations can start testing immediately and see meaningful findings within hours. As additional access is optionally granted (gray-box or white-box), Novee can add additional areas of focus to testing results, but **value is delivered immediately from the external attack surface.**

07

“Are remediation steps tailored to my architecture? And do you automatically retest fixes?”

Because the Novee both discovers and exploits vulnerabilities, it understands exactly how the issue manifests in the application, and generates personalized remediation guidance tailored to the customer’s architecture, frameworks, and business logic.

Once the issue is fixed, the system automatically retests the attack path to confirm the vulnerability has been eliminated, creating an attack-fix-verify loop.

08

“Is your platform safe to continuously run in production – does it get privileged access without guardrails?”

Novee’s system is designed to demonstrate exploitability without causing damage. Testing uses proof-of-concept payloads that validate vulnerabilities while avoiding destructive behavior or sensitive data extraction. And all tests begin with a preliminary report, showing users exactly which components will be tested, and how.

These protections allow the platform to operate continuously in production environments while maintaining safety.

How Novee Discovers Novel Vulnerabilities and Delivers Instant Remediation Guidance

The below examples – one research-focused, the other based on a real-world customer environment – show exactly how Novee operates, and how we achieve our mission: **uncovering novel vulnerabilities and delivering precise, personalized fixes tailored to your environment.**

Research: Discovering 16 New 0-Day Vulnerabilities in PDF Engines

Novee's research team demonstrated the platform's depth by discovering 16 previously unknown zero-day vulnerabilities across widely used PDF engines, using a 3-phased approach.

Phase 1: Seed

Identify and demonstrate attack paths. In one instance, we managed to enact a full Account Takeover (ATO) via DOM-based XSS in Apryse WebViewer.

Phase 2: Scent

Give Novee a scent to track. We taught the Novee AI hive what "real exploitable trust boundary violations" look like in each PDF ecosystem.

Phase 3: Exploit

Now Novee knows how to think like a hacker, so we let it act like one. Novee autonomously explored adjacent attack surface areas and identified additional vulnerabilities.

As a result, the Novee AI discovered **13 new exploitable vulnerabilities, in addition to the 3 found by our researchers.**

Faced with dynamic code paths and real trigger conditions, most tools would stop trying to find an exploit, or at best, guess where one might be found.

Novee AI doesn't give up, and it doesn't need to guess, because behind our team of human researchers is a hive of AI agents, specialized and designed to replicate their intuition, experience, and persistence.

AS REPORTED IN: Cyber Security News SECURITYWEEK CYBERSECURITY NEWS, INSIGHTS & ANALYSIS siliconANGLE

CUSTOMER CASE STUDY

JB Poindexter

"Our pen tests took weeks and consistently missed critical issues. Novee found them immediately and gave us instant remediation guidance. It showed us what we'd been missing."

JOHN BARROW, CISO,
JB POINDEXTER

JB Poindexter is a large U.S. manufacturer of truck bodies and equipment. Their environment blends operational technology, industrial systems, and software-enabled workflows. Downtime carries real operational and financial impact.

For their team, the shift to continuous adversarial validation meant:

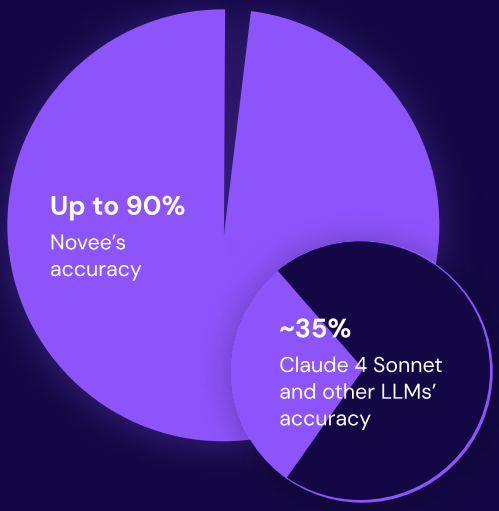
- Critical vulnerabilities surfaced immediately
- Exploitability validated, not theorized
- No waiting weeks for static PDF reports

Waiting weeks means running out the clock on exploitable vulnerabilities and ending up with nothing but an outdated report to show for it. Novee turns that downtime into remediation time.

DATA PROOF:

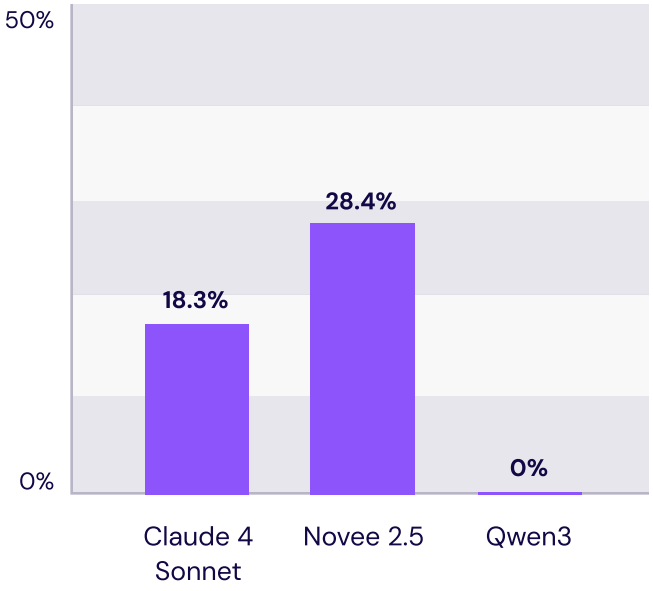
Purpose-Trained AI Models vs. Frontier

In live-browser exploit benchmarks, Novee's small, purpose-trained AI model achieved up to **90% accuracy,** outperforming Claude 4 Sonnet and other frontier LLMs **by over ~55%.**



Despite their impressive coding capabilities, frontier LLMs haven't been trained on the specific challenge of adversarial exploitation – and that specialized experience makes all the difference. Even in near-impossible scenarios, [the small reinforcement-trained model consistently outperformed frontier LLMs, while using its turns more efficiently](#).

MODEL	BASE DIFFICULTY	HARD DIFFICULTY
XSStrike (scanner)	79.0%	62.3%
Gemini 2.5 Pro	65.7%	49.4%
Claude 4 Sonnet	78.3%	64.7%
Novee 1: SFT only	87.0%	55.7%
Novee 1: SFT + RL	91.7%	90.0%



Adversarial exploitation

Even frontier models struggled on this task. Claude 4 Sonnet hit 18.3%. Novee 2.5, our most advanced model, reached 28.4% – **a 55% improvement.** The baseline Qwen3 model scored zero.

How Novee Can Help You

Novee was built by veteran offensive operators who distilled elite attacker tradecraft into a proprietary AI model, designed to think like a real adversary, and go to work for you. The result is an offensive security platform that discovers, maps, and exploits novel vulnerabilities with context and precision.

Novee clearly answers the question: **Can someone break into your system right now?**

Get a demo of the Novee platform, and be up and running in days with a continuous, validated report of novel vulnerabilities across your platform. And the guidance for how to fix them.

Novee

 | novee.security | LinkedIn