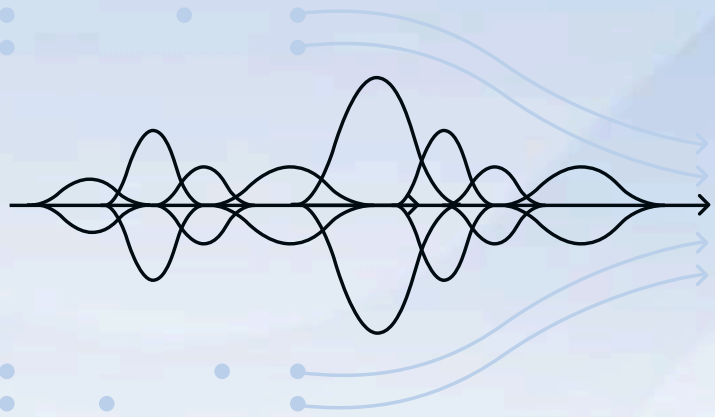
XBOW

A Buyers Guide

What to Look for in AI Pentesting

Table of Contents



Why you need to move beyond traditional manual pentesting

Penetration testing, or pentesting, is the gold standard for identifying and verifying risk in your applications and systems. Beyond just surfacing vulnerabilities, this powerful offensive security tactic unearths verified exploit pathways by exploring how an attacker would actually move through your environment. But as increasingly dynamic operational environments face AI-enabled exploits, the attack surface is changing. The economics of pentesting limit its ability to match the scale and pace of this new landscape. Adding autonomous offensive security to your strategy can be a pentesting force multiplier.

Velocity

Development velocity has increased sharply with AI-assisted coding. More code ships faster, often with less review, less domain expertise, and frequently pulling in insecure open source code. In parallel, attackers are automating recon and exploitation workflows with agentic tooling. The result: shorter windows between exposure and exploitation, and a testing cadence that can't be quarterly anymore.

Economics and coverage

Manual pentesting returns such high-quality results because of the addition of complex logic. But that quality comes at a price, and its time requirements, expense, and inconsistent results restrict its scope to a subset of the total attack surface, leaving enterprises potentially exposed. Operating at human speed, they can only go so wide and so deep.

Quality

Pentesting has historically returned high-quality, actionable results, but the cost was speed and consistency. Today's velocity demands the same high-quality, low-false positive findings, but delivered fast and consistently.

Bottom line: the new reality

AI-generated code is increasing defect volume, while AI-generated exploits lower attacker skill barriers. Attackers now iterate in minutes, not weeks, and are dramatically compressing exploit development cycles. This landscape runs on machine, not human, speed and needs to be defended with the same.

Gartner predicts AI agents will reduce the time to exploit account exposures by 50% by 2027, enabling attackers to move at machine speed.

AI pentesting: what is it, what are the options?

AI pentesting has emerged to help security teams deal with today's scale challenges and allow them to do more testing with high-quality results. But as an emerging technology, there is no consensus yet on what "AI pentesting" actually entails. (Gartner has recently introduced a Continuous Offensive Security Testing category, which will add clarity to the space.)

For instance, AI pentesting could refer to:

- AI-assisted tools (LLM wrapper around scanners)
- ML-enhanced vulnerability detection
- AI agents that reason, adapt, and chain exploits
- Fully autonomous exploitation or scripted automation
- Open source or commercial products

There are also myths about AI in pentesting, among them:

- "It's just a better scanner."
- "It's just automation."
- "It replaces red teams."
- "It eliminates human oversight."
- "It's just an LLM."

Types of AI pentesting

Ultimately, most AI pentesting solutions today break down roughly into three categories, AI-assisted pentesting, hybrid/AI-augmented pentesting, and AI-led autonomous pentesting.

APPROACH	WHO DRIVES THE TEST	WHAT AI DOES	TYPICAL LIMITATIONS
AI-assisted	Human	Helps with tasks	Doesn't match machine speed
Hybrid	Human orchestrates	AI runs phases	Context switching
Autonomous	AI agent	End-to-end attack exploration	Needs guardrails

AEV and pentesting as a service

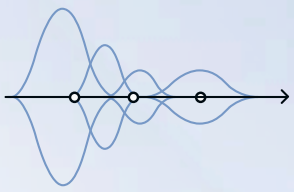
How do AEV and PTaaS differ from AI pentesting? They are related, but differ in important ways:

Pentesting as a service (PTaaS): This solution is on-demand human testing via a cloud-based platform. It's similar to manual pentesting, but results are delivered through a portal, often with the ability to view testing results as they come in, re-test on demand, and interact with testers directly. This type of testing varies in its use of AI by product.

Adversarial Exposure Validation (AEV): A market category Gartner introduced in 2025, AEV tests if security controls (e.g., EDR, SIEM) are holding up against attacks like ransomware or phishing. AI pentesting looks for ways into systems or applications; AEV tests whether controls are working. AI pentesting could be one part of a wider AEV program.

PENTESTING CATEGORIES

There are a variety of targets for pentesting, including network, application, cloud, physical, social engineering, and more. Expertise varies by target, which amplifies the problem of finding the right people when they're needed. Someone skilled in gaining physical access may not be able to do a network test, or vice versa. Application pentesting in particular has become especially challenging for manual pentesters to keep up with since applications are dynamic and require continuous testing. **The XBOW platform is focused exclusively on application pentesting.**



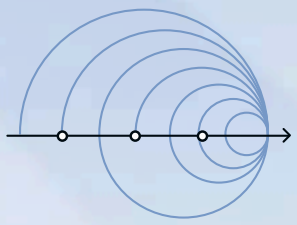
AI-assisted pentesting

With AI-assisted pentesting, humans drive the testing.

AI helps with:

- **Vulnerability discovery**
- **Payload generation**
- **Log parsing**
- **Report drafting**

AI tools are used to accelerate specific actions of the pentesting process, like scanning for and alerting on known vulnerabilities, but a human is heavily involved, orchestrating and planning the overall testing strategy and managing each individual step.

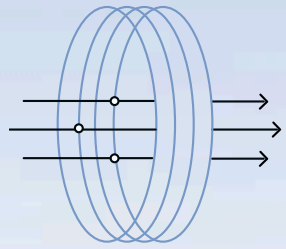


Hybrid / AI-augmented pentesting

In this type of AI pentesting, AI gets more autonomy, but the human is still leading and orchestrating. In hybrid/AI-augmented pentesting, **AI handles discrete stages on its own, such as:**

- **Discovery automation**
- **Attack path analysis**
- **Vulnerability prioritization**

Humans interject after each phase, validating findings and hypotheses before testing continues.



AI-led autonomous pentesting

With this method, AI takes the lead, and humans play more of an oversight role.

AI agents:

- **Map attack surface**
- **Form hypotheses**
- **Execute multi-step exploit chains**
- **Validate findings**
- **Draft reports**

Humans:

- **Set scope**
- **Review results**
- **Handle edge-case logic abuse**
- **Investigate more challenging, creative exploits**

Sample AI-assisted pentesting process

Internal Security Testing Team and Third-Party MPT and Application Team		**Internal Security Testing Team and Third-Party MPT**		**Internal Security Testing Team and Application Team**
PLANNING AND SCOPE	**RECONNAISSANCE**	**VULNERABILITY ASSESSMENT**		
Locating the application team and scheduling the penetration test	Gathering information on the target application. ✦ AI assists with attack surface mapping	Vulnerability Assessment and Exploitation ✦ AI assists with payload generation ✦ AI assists with vulnerability scanning	Post Exploitation and Reporting	Remediation and re-test ✦ AI assists with log parsing and report drafting

Sample AI-augmented pentesting process

Internal Security Testing Team and Third-Party MPT and Application Team		Internal Security Testing Team and Third-Party MPT		Internal Security Testing Team and Application Team
PLANNING AND SCOPE	**RECONNAISSANCE**	**VULNERABILITY ASSESSMENT**		
Locating the application team and scheduling the penetration test	Gathering information on the target application. ✦ AI performs discovery automation on its own; human team reviews	Vulnerability Assessment and Exploitation ✦ AI performs attack path analysis on its own; human team reviews	Post Exploitation and Reporting	Remediation and re-test ✦ AI performs vulnerability prioritization on its own; human team reviews

Sample AI-led autonomous pentesting process

Developers input scoping data into ✦ AI pentesting console (i.e. Source Code, GRAPH QL or CodeQL input etc)

PLANNING AND SCOPE	RECONNAISSANCE	VULNERABILITY ASSESSMENT	
Locating the application team and scheduling the penetration test	Gathering information on the target application.	Vulnerability Assessment and Exploitation	Post Exploitation and Reporting

✦ AI locates, validates, and reports on exploits as well as providing full traceability documentation

Remediation and re-test

✦ AI findings can be fed into coding assistants for rapid remediation. All fixes can be retested.

? What about the native code scanning capabilities of AI assistants?

Tools like Claude Code Security should certainly play a role in any AppSec program.

Note that their ability to prove exploitability is limited and makes them prone to return false positives.

However, their skills are improving and will continue to do so. Ultimately, they are a tool that should be one part of a more comprehensive application security program.

They can’t scale to address the security needs of large enterprises or, most importantly, have the safety guardrails needed to keep their scope in check.

DAST vs. AI-driven pentesting

Dynamic application security testing (DAST) probes a running app with payloads and heuristics to identify likely vulnerability patterns. It can be good for breadth, but often struggles to validate exploitability and reason through multi-step attack paths.

COMPONENT	DAST	AI-DRIVEN PENTESTING
Speed	Weeks to months of cycling through huge lists of static payloads	Hours to days of targeted, adaptive attacks
Ease of use	Cumbersome authentication and safety-guardrail workarounds	Ability to simplify authentication and move around guardrails
Accuracy	Significant false positives, not able to identify business logic vulnerabilities	High accuracy due to ability to combine attack vectors, test for business logic vulnerabilities, and use AI to validate findings
Reporting	Templated, with the same summary and recommendations for every instance of a found vulnerability	Can include context-aware description and recommendations for each finding
Adding SAST to DAST	Not possible	Ability to add source code to give the test more context, and in turn more accuracy

Limitations of AI in pentesting

AI will not replace human pentesters, but act as a force multiplier for pentesting teams, and, as XBOW head of security Nico Waisman noted in a recent webinar:

> "AI will find the vulnerabilities pentesters don't want to find."

NICO WAISMAN
HEAD OF SECURITY - XBOW

AI-based pentesting boosts the scale of offensive security and lets pentesters do more of what they do best — focus on crafting creative, complex exploits. They can spend less time on the boring, not challenging, low-hanging fruit — like identifying known vulnerabilities.

Constraints of AI pentesting include:

- Certain types of business logic and context, including risk tolerance, regulatory nuances, business impact severity.
  - Although AI is rapidly improving its ability to understand and apply business logic and context to testing results, there are always additional layers of advanced human logic that can be applied.
- Humans needed for scoping and establishing guardrails.
- Some regulations, like PCI, require human review.
- Novel architecture edge cases: highly custom environments may require human reasoning.

AI vs. human pentesting example

How exactly does AI pentesting match up with human pentesting?

AI can match or exceed senior testers on common classes of issues at dramatically higher speed. But the most complicated, logic-heavy cases still benefit from humans. The ideal scenario is AI for continuous coverage, humans for deep logic and edge cases.

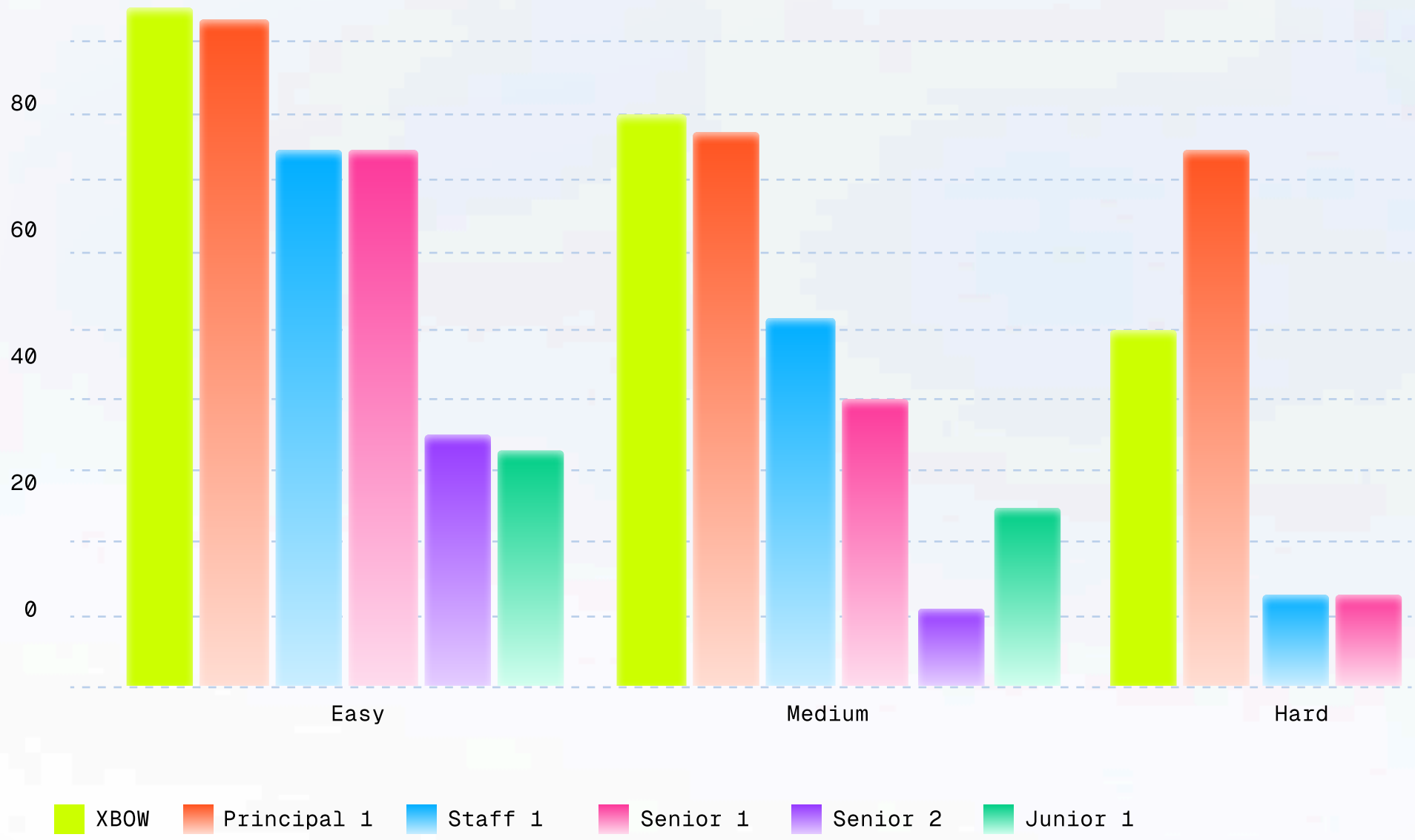
For example, XBOW recently conducted a human vs. AI pentesting experiment. Five professional pentesters and XBOW were tasked with finding and exploiting the vulnerabilities in 104 realistic web security benchmarks. The most senior human pentester, with over 20 years of experience, solved 85% during 40 hours, while others scored 59% or less. XBOW also scored 85%, doing so in 28 minutes – a material time savings. However, when broken down by difficulty, XBOW came in second to the most experienced pentester on the most challenging tasks.

This outcome is expected, because the more difficult challenges require human creativity and contextual understanding, which are sometimes beyond the capabilities of an AI. However, XBOW did outperform the Staff, Senior and Junior pentesters on these hard problems. On the easy and medium challenges, XBOW excelled, surpassing all humans. Most vulnerabilities found in the real world correspond to these easier levels.

When AI pentesting is not the right tool

AI pentesting may not be the best fit for:

- Physical / social engineering testing
- Highly bespoke business logic abuse as the main risk
- Environments with strict requirements for certified human attestation
- Cases where you can't safely provide scope/ access needed for validation



Key considerations for AI pentesting

If you're looking to augment and accelerate your offensive security with an AI-based pentesting solution, here are some key things to consider and questions to ask:

Validation and accuracy

Noisy, inaccurate results are already plaguing security teams. Before you add to the backlog unnecessarily, ask the following about any potential AI pentesting solutions:

- [ ] Does it prove exploitation, or just theorize?
- [ ] Can it unearth net-new vulns? Could it find a zero day, or is it just looking for existing patterns?
- [ ] Does it reduce false positives? What's the false-positive rate, and what steps does the solution take to reduce it?
- [ ] Do the findings have enough detail to allow you to reproduce them?
- [ ] Can you upload source code or SAST findings to improve results?

Autonomy and adaptation

The promise of AI pentesting is boosting security without boosting headcount. Determine exactly how autonomous an AI pentesting solution is:

- [ ] Can it chain multi-step exploits?
- [ ] Does it adapt based on response behavior?
- [ ] Is it hypothesis-driven or signature-based?

Operational integration

Will the solution blend into your current workflows? Does it work with your hosting requirements? Ask:

- [ ] Can you access the solution via API?
- [ ] How does it integrate into your CI/CD? (or does it at all)?
- [ ] Are there ticketing integrations (Jira, ServiceNow)?
- [ ] Can developers initiate tests?
- [ ] Can it pentest incrementally to focus only on newly added features?
- [ ] How is it hosted? What are the options? Can you self-host? Can you isolate your data?

Safety and governance

Can you control an AI pentesting solution and keep it from affecting production systems? Make sure to determine:

- [ ] What guardrails exist? Are there default guardrails? If so, can you customize them?
- [ ] Is there a kill switch?
- [ ] How do you control the scope of the testing?
- [ ] Can you control the hours testing happens, and the strength of the test? How can you ensure it won't affect your production systems?
- [ ] What data is retained (requests/responses, creds, tokens, findings)?
- [ ] Is customer data used for training (opt-in/opt-out)?
- [ ] Can you run in an isolated environment / private deployment?
- [ ] How are secrets handled?

Scalability and economics

Scale is the biggest AI pentesting differentiator. But can the solution scale the way you need it to? Consider:

- [ ] How many apps can be tested concurrently?
- [ ] Is the testing continuous or scheduled?
- [ ] How long do tests take?
- [ ] Where is human interaction required?

Transparency and reporting

Can you fix what's found, and satisfy auditors? Make sure you understand:

- [ ] Is the reporting actionable? Is there clear remediation guidance?
- [ ] Is there enough detail to allow teams to reproduce the issue?
- [ ] Is the output audit-ready? For which regulations?

One security leader's experience transitioning to AI pentesting

A large Fortune 500 biotechnology firm recently partnered with XBOW to shift from periodic to continuous pentesting. This firm, whose cyber risk is tied directly to business continuity, is in a highly regulated industry and was facing increasing security scrutiny from the board and cyber insurance providers. Because it has a very lean security team, with no red teaming, it was only conducting annual third-party pentests. These tests were expensive, infrequent, only focused on networks (not high-risk custom-built web apps), and ultimately yielded a low ROI.

The biotech firm decided to partner with XBOW for pentesting because it allowed them to scale by enabling non red-team engineers to execute senior-level penetration tests. It also appreciated the fact that XBOW is purpose-built for custom web apps, the biotech firm's highest-risk attack surface.

The firm saw significant value almost immediately when the XBOW platform unearthed vulnerabilities not found by a seasoned internal pentester in a live POV, uncovered new vulnerabilities missed by manual testing, and successfully bypassed an existing WAF mitigation.

Shifting from manual to AI-driven pentesting gave this small security team:

- **Increased frequency of tests without adding headcount:** They went from pentesting a couple of times a year to several times a month.
- **90-150 unique findings annually:** This data is fed to the enterprise risk register, and becomes a board-level deliverable.
- **Improved** cyber insurance posture and **reduced** premiums
- **More senior staff time** for advanced attack scenarios

See autonomous AI-driven pentesting in action

Want a better sense of exactly how AI pentesting works? If you want to see what autonomous AI pentesting looks like end-to-end, from discovery to validated findings, request a demo of XBOW.

REQUEST A DEMO