Pentest Tools

AI pentesting in 2026: why testing cadence decides who copes

Discover what 158 security practitioners told us about using AI for pentesting - where it works, where it doesn't, and where it makes the most sense

A 2026 survey from Pentest-Tools.com

Contents

Foreword: Where does automation fit into the craft of pentesting?

Penetration testing has always been context-dependent, creative work - a craft. And we believe it still is, in spite of commoditization through automation and the underlying compliance needs that push for it. The stories these pentesters shared with us ring as true today as they did when we first recorded them.

Precisely because **the ethical hacker mindset is deeply inquisitive and highly adaptable**, we wanted to hear from practitioners themselves how AI is changing the way they work so we can share with you the reality behind overblown headlines and apocalyptic scare tactics.

What we noticed is that **security practitioners sit between two conversations that rarely meet**.

- → In one, AI accelerates how fast they can find and report vulnerabilities.
- → In the other, the AI systems their organizations deploy introduce risks that need testing themselves.

Both conversations are speeding up at once, and the people who run security testing feel the pressure from each side.

Most research on this shift highlights how security leaders and buyers see things. We wanted to hear from **the people who do the testing**. So in June 2026, we surveyed **158 security practitioners who use AI-assisted tools** in their vulnerability assessment and validation work: penetration testers, security engineers, DevSecOps and AppSec professionals, consultants, and MSSP practitioners.

The results are remarkably consistent, across roles, testing frequencies, and organization sizes:

→ **AI has delivered on the discovery promise**. Practitioners can find more, faster.

→ **What AI hasn't delivered is proof**. The work of deciding which findings are real, which are exploitable, and which deserve a developer's attention hasn't shrunk. It has grown, because the volume feeding into it has expanded.

One survey respondent compressed the whole dataset into three sentences:

> An AI tool spits out 300 findings. I spent two days triaging, and 250 were junk - duplicate vulns, potential SQLi that is not exploitable, or AI-made-up CVEs that do not exist.
>
> I bought the tool to save time, but I did more manual work than before.

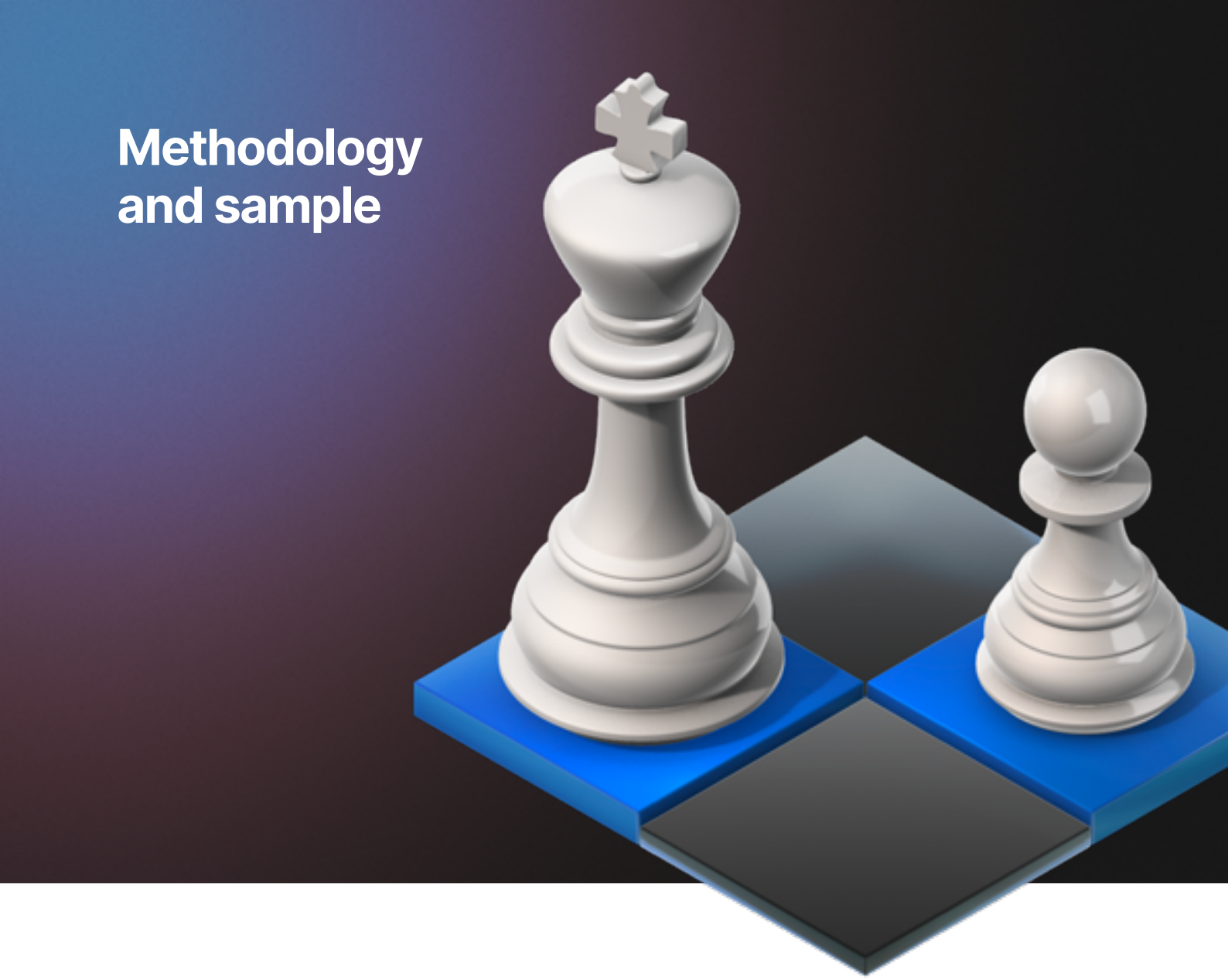
That's the pattern this report unpacks: **the time AI saves on discovery comes back as triage**. Trust erodes faster than it builds. **Business logic remains stubbornly human territory**. And the teams coping best aren't the ones with the newest tools - they're the ones who built their validation and prioritization muscle before the volume arrived.

Throughout this report, we pair the quantitative data with verbatim answers about where AI tools fail, what requires manual follow-up, what frustrates practitioners most, and what they wish

The numbers and quotes are here to help you locate your own team in the data, spot where your validation pipeline is most exposed, and decide what to strengthen first.

Adrian Furtuna
Pentest-Tools.com Founder & CEO

Methodology and sample



Pentest-Tools.com surveyed 158 security practitioners in June 2026 via an online questionnaire. All respondents were screened for using AI-assisted vulnerability assessment and validation tools in their work.

What we asked

The survey combined **12 substantive questions**:

- → where practitioners use AI in the pentest lifecycle
- → where the tools perform best
- → how often findings need manual validation
- → whether teams could absorb 500+ AI-generated findings from a single engagement

Primary role

Security engineer / blue team

23.4% (37 respondents)

Penetration tester / offensive security / red team

18.4% (29)

DevSecOps / AppSec

14.6% (23)

Consultant / MSP / MSSP

10.1% (16)

Security manager / CISO

7% (11)

Other roles responsible for offensive security

4.4% (7)

- → how stakeholder expectations are shifting
- → what drives AI pentesting software evaluation decisions
- → whether AI systems themselves are entering the testing scope.

Four open-ended questions captured:

- → where the tools fail
- → what requires follow-up
- → the biggest frustrations
- → the one thing practitioners wish AI could do.

How to read the numbers

Percentages are based on all 158 respondents unless otherwise noted.

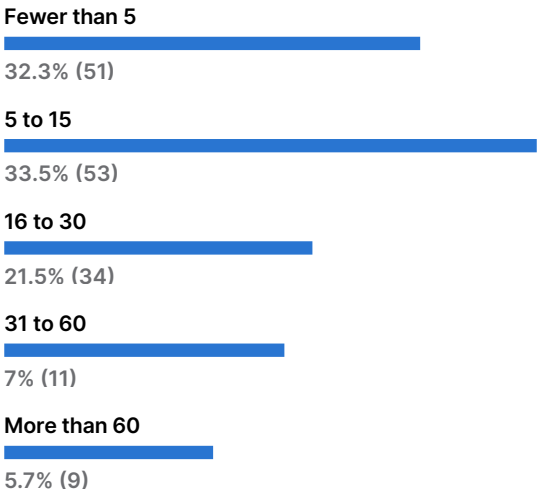
The manual validation statistic is based on the 147 respondents who have used AI tools for finding generation.

Where we segment by testing cadence, the highest-frequency groups are small, so we present those results as directional trends rather than firm conclusions.

Respondent quotes are lightly edited for spelling and grammar only; meaning is unchanged.

Organization size tested or secured

Testing cadence (pentests per month)



The five findings that capture the change

Five findings stood out. Together, they describe a discipline that has absorbed AI at the edges of the engagement and is now paying for it in the middle.

1 **Practitioners placed AI where mistakes are cheap to catch**

2 **The validation debt is nearly universal**

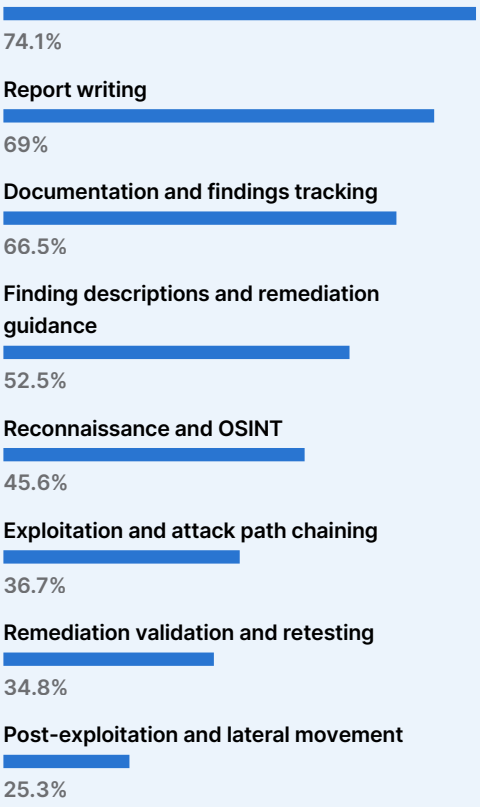
3 **Most teams cannot absorb the volume AI can generate**

4 **Hallucinated findings are the single biggest frustration**

5 **Practitioners buy signal quality before they buy speed**

1

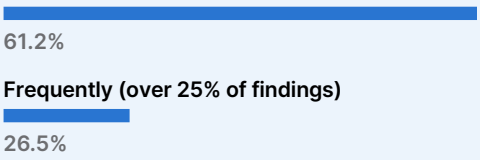
Where do you use AI in the pentest lifecycle?



AI usage is heaviest in vulnerability scanning and discovery (74.1%) and in the writing that follows: reports (69%), documentation and tracking (66.5%), finding descriptions with remediation guidance (52.5%). It drops sharply in the phases that require live judgment: exploitation and attack path chaining (36.7%), remediation validation and retesting (34.8%), post-exploitation (25.3%).

2

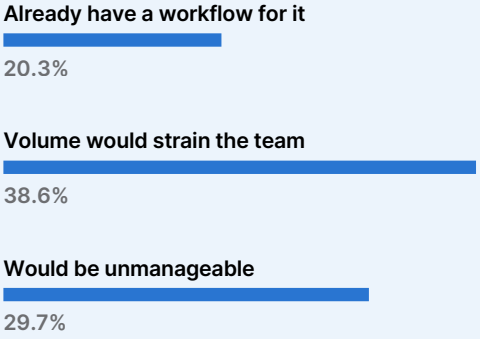
How often do AI-generated findings need significant manual validation?



Among the 147 respondents who have used AI tools to generate findings, 87.8% ran into findings that require significant manual validation - 61.2% sometimes (5-25% of findings), 26.5% frequently (more than a quarter of findings need rework). Roughly **1 in 4 practitioners reworks more than 25% of what AI tools produce**.

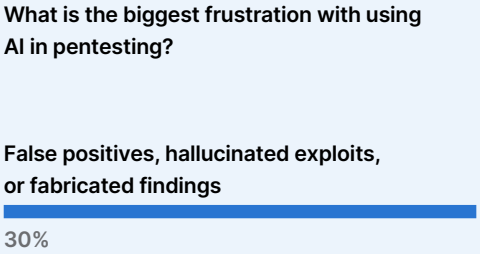
3

Could your team triage and validate 500+ AI-generated findings from a single engagement?



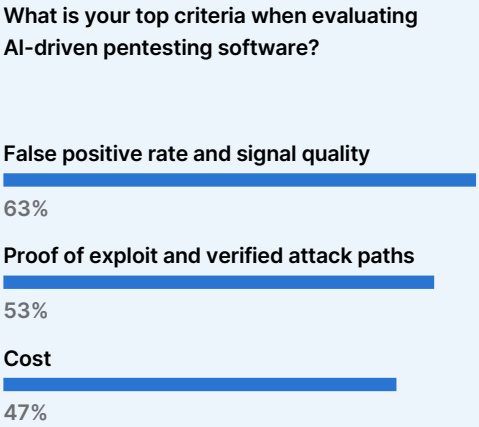
Asked whether their team could triage and validate more than 500 AI-generated vulnerability candidates from a single engagement, only 20.3% said they already have a workflow for it. 38.6% said the volume would strain the team; 29.7% said it would be unmanageable. **Nearly 7 in 10 teams could not confidently absorb the output** that current tooling can produce.

4



Roughly 30% of free-text responses (51 of 156) named **false positives, hallucinated exploits, or fabricated findings** as the biggest frustrations with using AI in pentesting - the most common answer by a wide margin, ahead of cost, integration, and every other complaint.

5



When evaluating AI-driven pentesting software, **false positive rate and signal quality** lead the criteria at 63%, with proof of exploit and verified attack paths second at 53%. Cost comes third at 47%. Two accuracy-related factors outrank price.

AI accelerated discovery

Volume went up

Validation became the bottleneck

Trust became conditional

These five findings work together

And because trust is conditional, practitioners now evaluate tools on proof, not promises.

> The biggest frustration is that the tools can generate a lot of findings, but not all of them are actionable. False positives, weak context, and unclear proof of impact still require a lot of manual review before the results can be trusted.

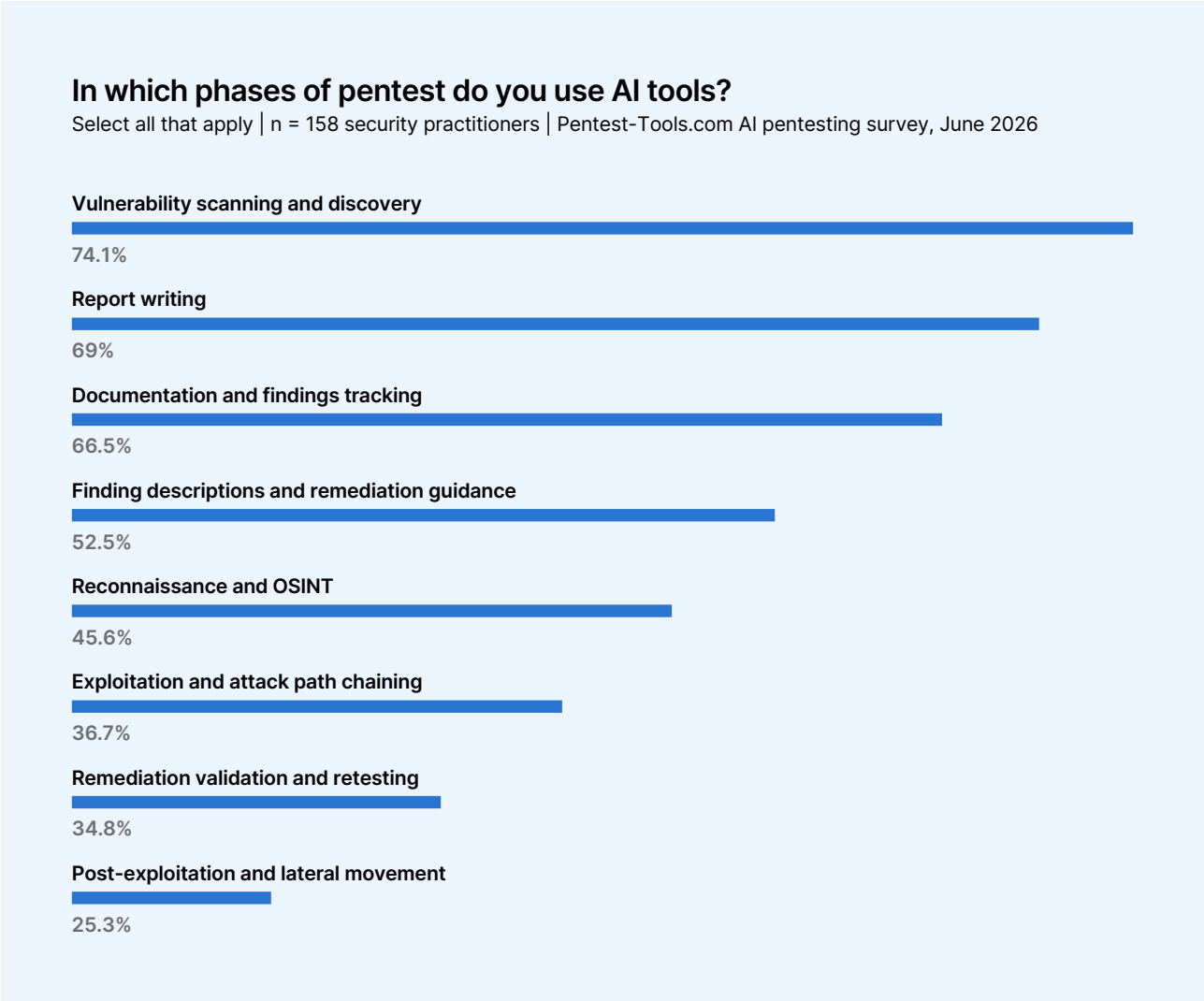
That's not an AI problem in isolation. It's what happens when discovery scales and validation doesn't - and validation is the part of the craft that resists automation most.

Where AI actually sits in the pentest workflow

To understand the triage problem, we need to start **where practitioners have actually placed AI in the engagement** - because the placement is deliberate, and it tells us what they've learned.

Heavy at the edges, light in the middle

AI usage by pentest phase, among all 158 respondents:

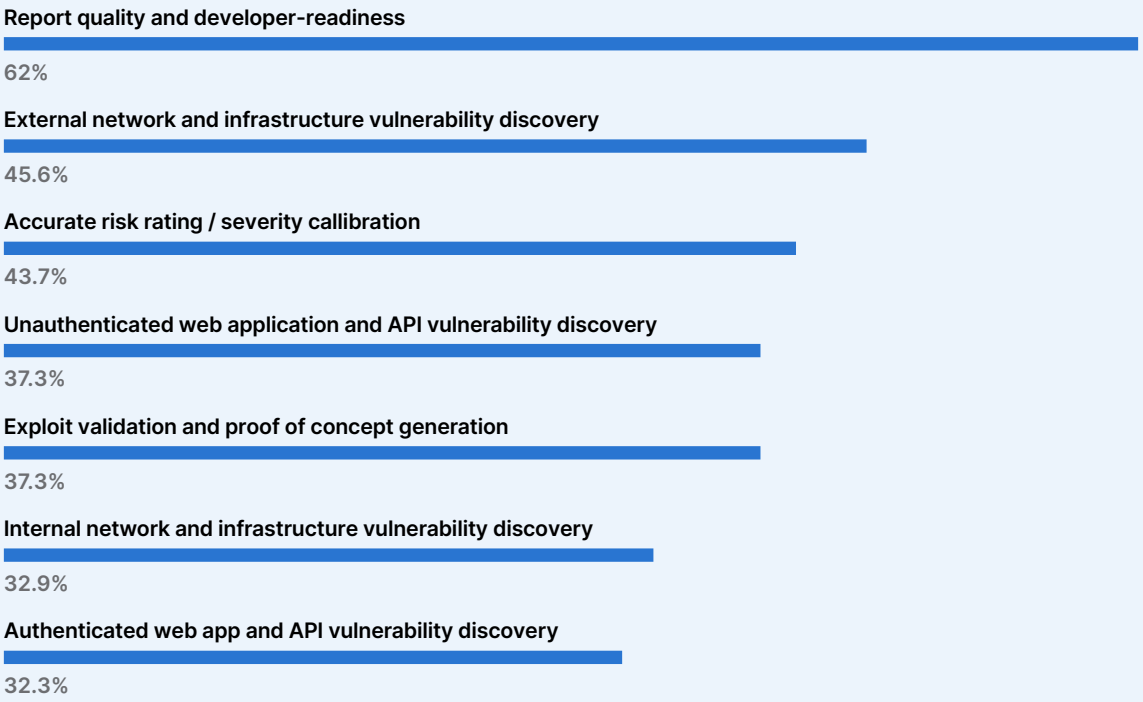


The shape is unmistakable. AI dominates the phases that bracket the engagement - the scanning before and the writing after - and thins out in the middle, **where a tester interacts with a live system and errors have consequences**.

Asked **where AI pentesting tools currently perform best**, practitioners confirmed the same shape from the other direction: report quality and developer-readiness first at 62%, accurate risk rating at 43.7%, exploit validation and proof of concept at 37.3%.

In which areas do you see pentesting tools currently perform best?

Select up to three | n = 158 security practitioners | Pentest-Tools.com AI pentesting survey, June 2026



The placement is the lesson

We don't read this as hesitancy or slow adoption. It's actually a **considered judgment about failure modes**.

A hallucinated sentence in a report draft costs a correction, but a hallucinated exploitation step against a production system has no safety net.

Practitioners have introduced AI where errors are cheap to catch and kept it out of the phases where errors are expensive.

> AI is strong at breadth and repetition, but humans still dominate in context, creativity, and exploiting the gaps between the rules.

The rest of this report explains what happens in the high-traffic stages where practitioners use AI most, and **why the middle of the engagement still belongs to the human tester**.

A hallucinated sentence in a report draft costs a correction, but a hallucinated exploitation step against a production system has no safety net.

The triage tax: where the time AI saved comes back (to bite)

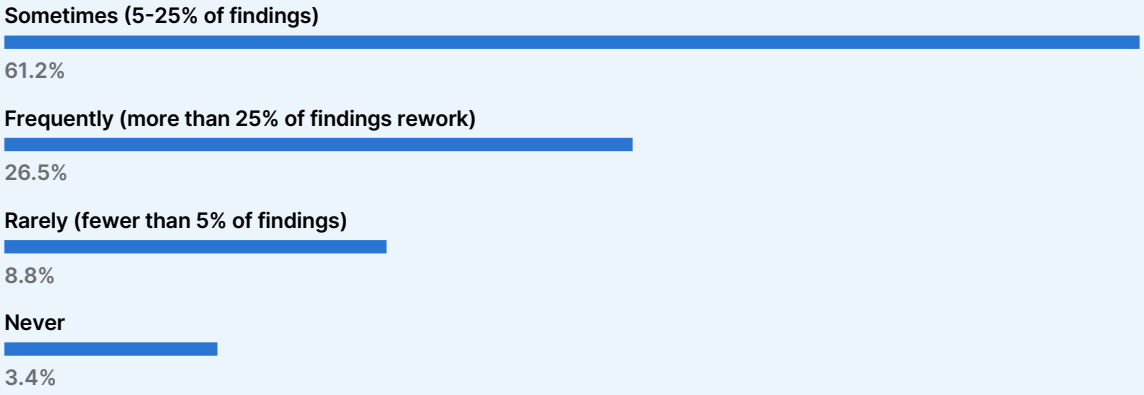
AI accelerates vulnerability discovery - that part of the promise holds up, and practitioners don't dispute it. What surprised many of them is where the work went afterward.

The validation debt accumulates in the highest-traffic stage

Among the 147 respondents who have used AI tools to generate findings, 87.8% run into findings that require significant manual validation: 61.2% sometimes (5-25% of findings) and 26.5% frequently (more than a quarter of findings need rework). Only 8.2% say this happens rarely, and 3.2% say never.

How frequently do AI-generated findings require significant manual validation effort?

Base: 147 respondents who have used AI tools for finding generation | n = 158 security practitioners | Pentest-Tools.com AI pentesting survey, June 2026



Scanning and discovery is also where practitioners use AI most (74.1%). The validation debt accumulates precisely where the traffic is heaviest.

Many practitioners expected AI to remove manual work. Instead, several described **buying a tool to save time and ending up spending that time on triage**. The effort moved downstream - from finding vulnerabilities to deciding which findings are real and which matter.

> It is the high volume of low quality or partially correct findings that still require significant manual validation.

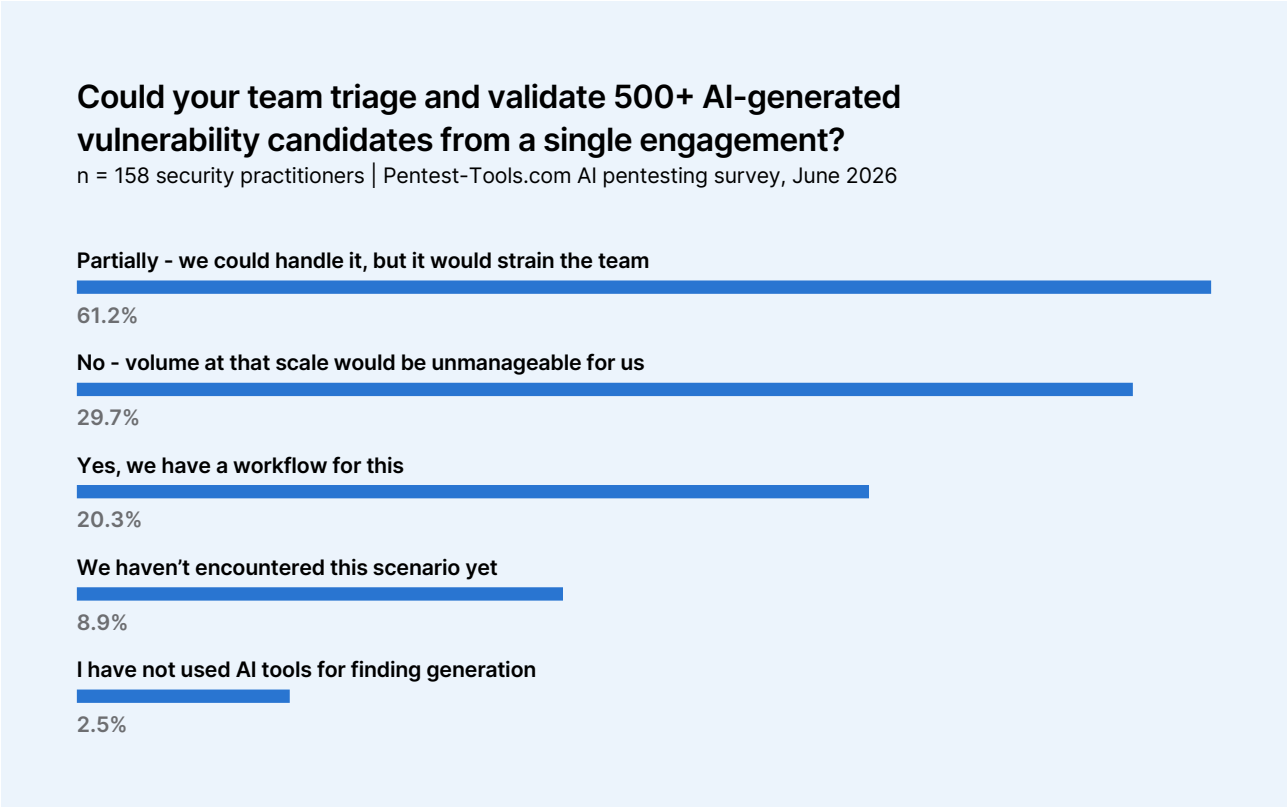
> Too many false positives and generic findings that require manual review, which reduces time savings and adds extra validation work.

Asked what type of finding most commonly requires manual follow-up, the answers clustered around four categories:

- → **business logic and contextual vulnerabilities** that need environmental knowledge
- → **false positives** flagged without verified exploitability
- → **complex exploit chains** where the AI can't determine whether a vulnerability is genuinely reachable
- → **documentation errors** - inconsistent risk ratings, wrong terminology, occasional hallucinations.

The 500-finding question

We asked the capacity question directly: if an AI tool generated more than 500 vulnerability candidates from a single engagement, could your team triage and validate them?



Nearly seven in ten teams sit in the strain-or-unmanageable zone. This is the number that should worry anyone who assumes more findings automatically translate into stronger security. Findings enter a pipeline, and for most teams, **the pipeline is already full**.

Nearly 7 in 10 teams could not confidently absorb the finding volume that current AI tooling can generate.

Cadence, not size, predicts who copes

Segmenting the data gave us the best practical insight: triage capacity reveals testing frequency more than company size.

Among teams running fewer than five tests per month, only 4% have a formal workflow for high-volume AI findings, and 55% describe the workload as unmanageable. **Teams that test more frequently are consistently better prepared** - more likely to have established workflows, less likely to feel overwhelmed.

Note: because our sample size for the highest-frequency groups is small, we treat this as a directional trend rather than a firm conclusion.

The logic is straightforward once you see it:

Frequent testing forces a team to build repeatable triage, prioritization, and retest habits, because ad-hoc handling collapses under a monthly rhythm.

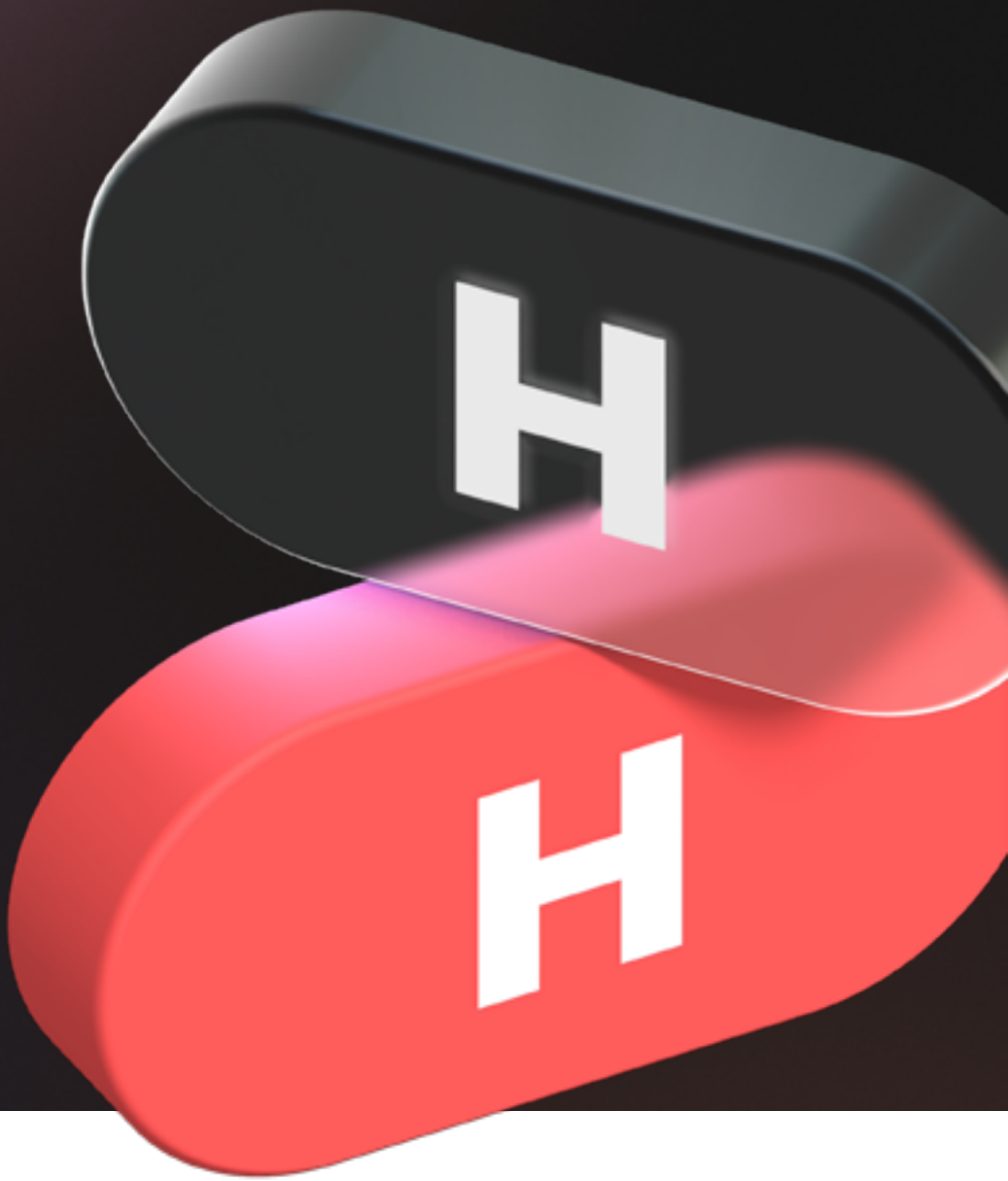
When the AI volume arrives, those teams route it into internal processes that already exist. Infrequent testers meet the same volume with no established workflow at all.

> I do not need a report with 1,000 vulns, because my org will never be able to actually tackle them.
> I need the ones that are more easily exploited and which target critical assets.

A hallucinated finding costs more than a missed one

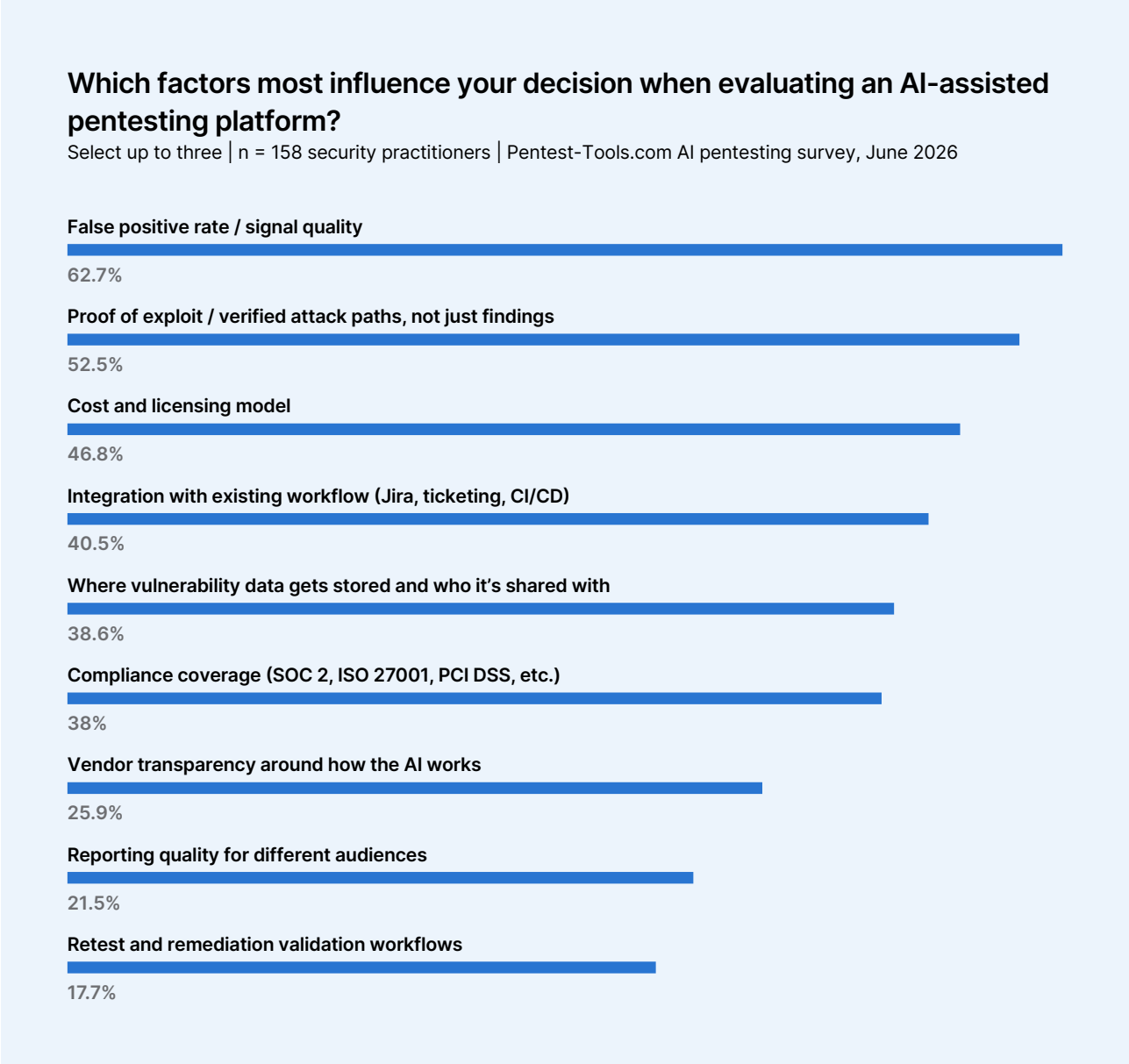
Validation workload is a capacity problem.

Hallucinated findings create a different kind of problem, though - they make teams question whether they can trust the output at all.



The most common frustration, by a wide margin

When we asked practitioners about their biggest frustration with the AI pentesting tools they've evaluated or purchased, roughly 30% of the responses (51 of 156) named false positizves, hallucinated exploits, or fabricated findings. Nothing else came close - not cost, not integration friction, not missing features.



The descriptions were specific:

- → AI-generated vulnerabilities that looked convincing but couldn't be reproduced
- → fabricated exploits and CVEs that don't exist

→ findings that were technically correct but ignored the organization's compensating controls

→ tools reporting success where there was none

> Big false positive rates and hallucinated exploits that waste time.

High volumes of low-confidence findings that require manual triage, combined with limited transparency into how conclusions were reached. Many tools generate impressive-looking output but provide insufficient evidence to support remediation decisions.

Why the cost of FPs compounds

A missed vulnerability is invisible until it matters. **A fabricated finding is immediately visible**, and it changes behavior.

Once practitioners catch a hallucinated finding, they stop extending the benefit of doubt to genuine ones. **Confidence becomes conditional** rather than assumed, and they start second-guessing whatever comes next.

It often takes one convincing hallucination to make a team much more cautious about what the tool produces afterward. From that point on, the team will likely pay a verification premium on the entire output, which is exactly the manual work they bought the tool to remove.

This is why the trust question and the capacity question are one and the same. Fabricated findings don't just waste the hours spent chasing them. They force re-verification of the honest findings too, and the triage tax rises across the board.

What humans still own: business logic and the attack path

We asked practitioners where **AI falls short, what still needs manual follow-up, and what they wish AI could do**. The number one answer every time? Understanding business logic. The second most cited gap was complex attack chaining and creativity.

The responses were specific, not vague complaints about AI limitations. Practitioners described exactly **where pattern matching stops and contextual reasoning has to take over**.

The coupon example

> AI pentesting tools can actually find SQL injections, but they won't understand 'this discount coupon should only work once per customer' and then also break the cash

> flow to abuse it.
>
> However, humans understand how the app is supposed to work. We find logic bugs like adding negative quantities to the cart is equal to free money, or changing the user ID in the URL to see other people's data, which doesn't require a signature.

This quote earns its length. A vulnerability like the coupon abuse produces no error, no signature, and no anomaly in the code. The application works exactly as intended. Recognizing that an attacker can abuse the workflow itself requires **understanding intent** - what the application is supposed to do - and that understanding still belongs to the human tester.

> They fail at understanding complex business logic flaws, where the application code runs without errors but the workflow itself is manipulated.
>
> How components are 'supposed to' interact, rather than how they necessarily do.
>
> AI is strong at breadth and repetition, but humans still dominate in context, creativity, and exploiting the gaps between the rules.

The chaining gap

The second recurring theme: **AI tools struggle to connect low-severity findings into realistic, multi-step attack paths.**

> AI tools still struggle with understanding business context, chaining findings into realistic attack paths, and

> judging which vulnerabilities actually matter in a specific environment. A human is better at validating impact, spotting edge cases, and explaining risk in a way the client can act on.

Individual findings are pattern-matchable. Finding the path between them requires a level of 'adversarial creativity' that respondents consistently say belongs to humans, not AI. This means you have to:

- → pivot through an environment
- → adapt when the obvious route is blocked
- → judge which combination actually reaches something valuable

Usage already reflects this

Recall our earlier look at where AI actually sits in the pentest workflow: usage is lowest in exploitation (36.7%) and post-exploitation (25.3%) - exactly the phases where understanding intent and chaining creatively matter most.

When we asked **what practitioners wish AI tools could do**, their answers mirrored the failures:

- → understand application workflows autonomously
- → distinguish theoretical risks from exploitable ones
- → chain findings into end-to-end attack paths
- → produce reproducible proofs of concept

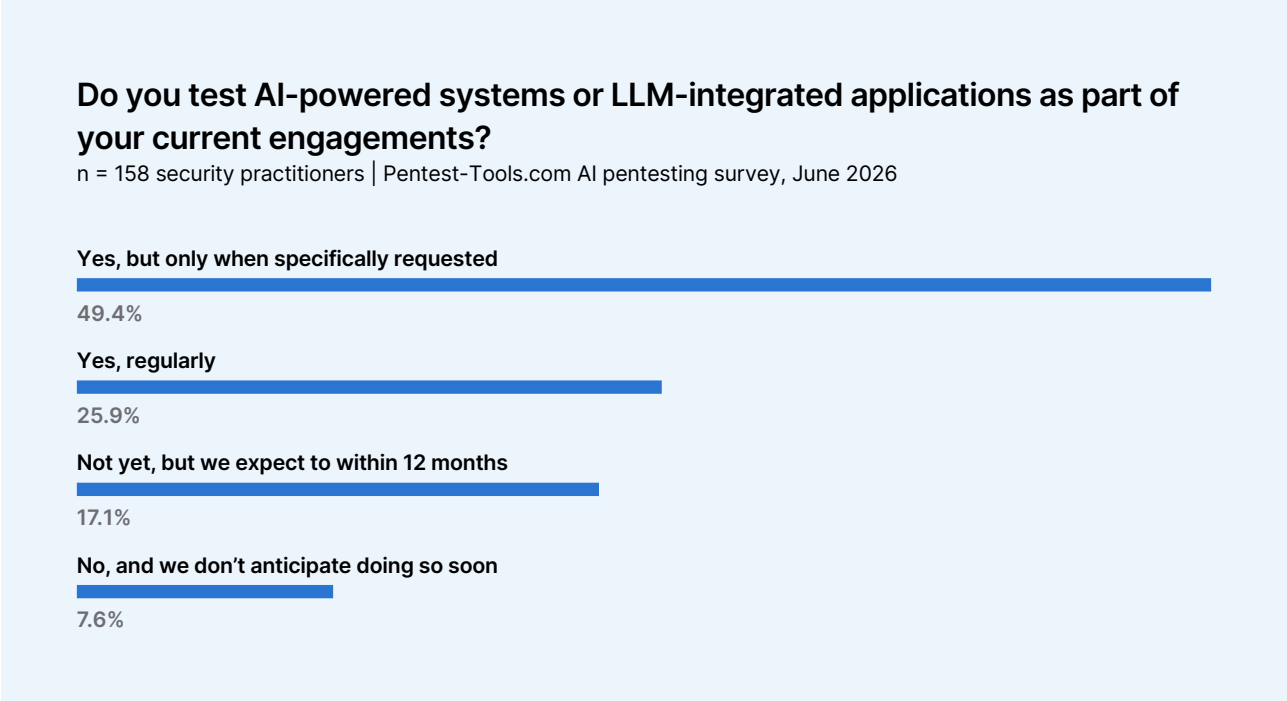
This wishlist is a map of the current boundary between machine pattern recognition and human judgment.

The scope keeps expanding: AI systems, shadow AI, and rising expectations

While practitioners absorb AI's output on the testing side, the systems they must test are changing too - and the people they answer to are noticing.

AI systems are slowly entering the standard scope

92.4% of surveyed practitioners already test AI systems or plan to within a year. Only 7.6% neither test them today nor anticipate doing so soon.



The most notable number here is the **49.4% who test AI systems only on request**.

AI system testing has arrived, but it hasn't yet become default scope - it's still triggered by someone asking. For most organizations, that means **the AI attack surface gets assessed reactively, not systematically**.

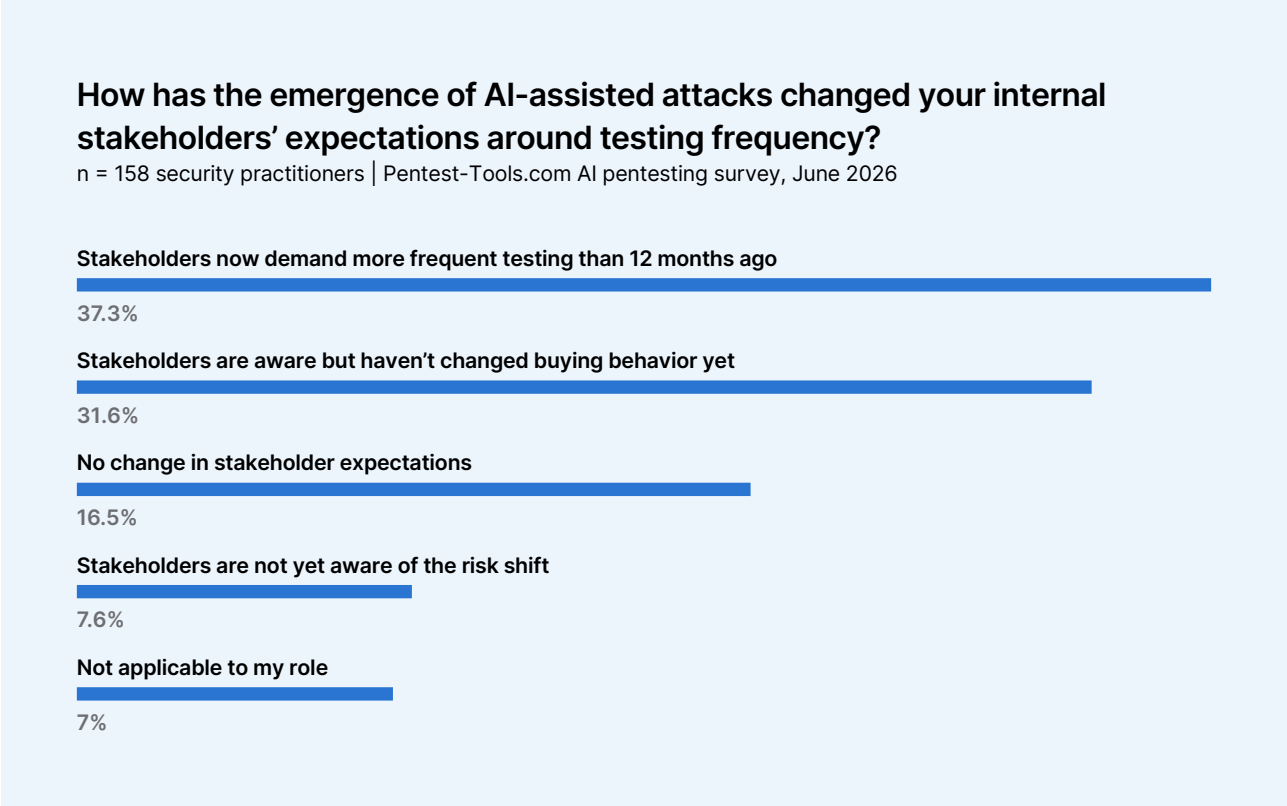
Shadow AI is on the radar, but not in the process

One in three practitioners (33.5%) already includes risks created by employees' unauthorized AI use in their assessment scope. Another 53.2% have discussed adding it but haven't formalized the process yet. Only 7.6% say it's not on their radar.

That 53.2% is the story: **shadow AI sits in the awkward middle** where nearly everyone acknowledges the risk and most haven't operationalized the response. **Formal coverage is catching up to awareness, but slowly**.

Stakeholders are moving faster than budgets

The pressure isn't only technical. 37.3% of respondents say internal stakeholders now demand more frequent testing than 12 months ago, driven by awareness of AI-assisted attacks. Another 31.6% say stakeholders are aware of the risk shift but haven't changed buying behavior yet.



The awareness is there; the decision cycle hasn't caught up. For practitioners, that means **demand for testing is likely to keep rising** while the resources to validate the results lag behind. This makes the triage tax we covered earlier even more urgent.

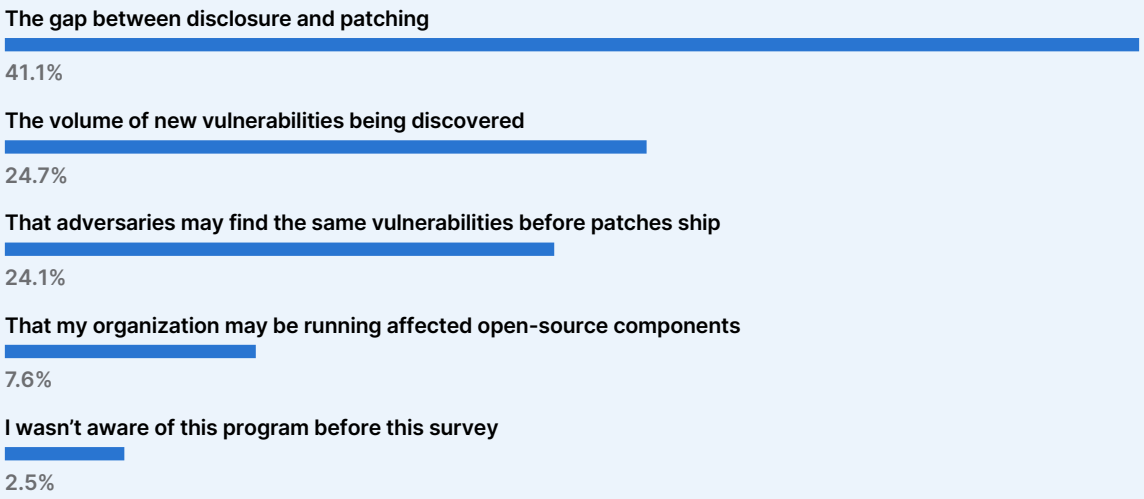
The disclosure-patching gap sharpens the stakes

We also asked about the wider ecosystem, since the largest AI models are shaping perceptions across the industry and across roles.

Frontier AI models can now identify vulnerabilities across major open-source projects at scale: one coordinated disclosure program referenced in the survey has disclosed over 1,500 AI-found vulnerabilities, with only around 97 patched at the time of fieldwork[1]. Asked which aspect concerns them most, 41.1% of practitioners pointed to the gap between disclosure and patching, and another 24.1% worried that adversaries may find the same vulnerabilities before patches ship.

Anthropic's coordinated vulnerability disclosure program has disclosed over 1,500 AI-found vulnerabilities across open-source projects, with only around 97 patched so far. Which aspect of this concerns you the most?

n = 158 security practitioners | Pentest-Tools.com AI pentesting survey, June 2026



1 Figures as presented to respondents in the survey questionnaire, drawn from Anthropic's coordinated vulnerability disclosure dashboard for Project Glasswing: 1,596 vulnerabilities disclosed across 281 open-source projects as of May 22, 2026, with 97 patched to Anthropic's knowledge. Source: red.anthropic.com/2026/cvd/ and "Project Glasswing: An initial update," anthropic.com/research/glasswing-initial-update (May 22, 2026). Figures reflect the dashboard snapshot available at the time of survey fieldwork (June 17-18, 2026) and have likely changed since.

Discovery has never been the real bottleneck, and practitioners know it.

Offensive security creates value when teams can reproduce a finding, confirm exploitability, prioritize it, and hand developers something they can act on.

AI changes how many findings enter that pipeline. It doesn't change how quickly organizations move through it, and the disclosure-patching gap is what that mismatch looks like at ecosystem scale.

What practitioners buy: signal quality before speed and price

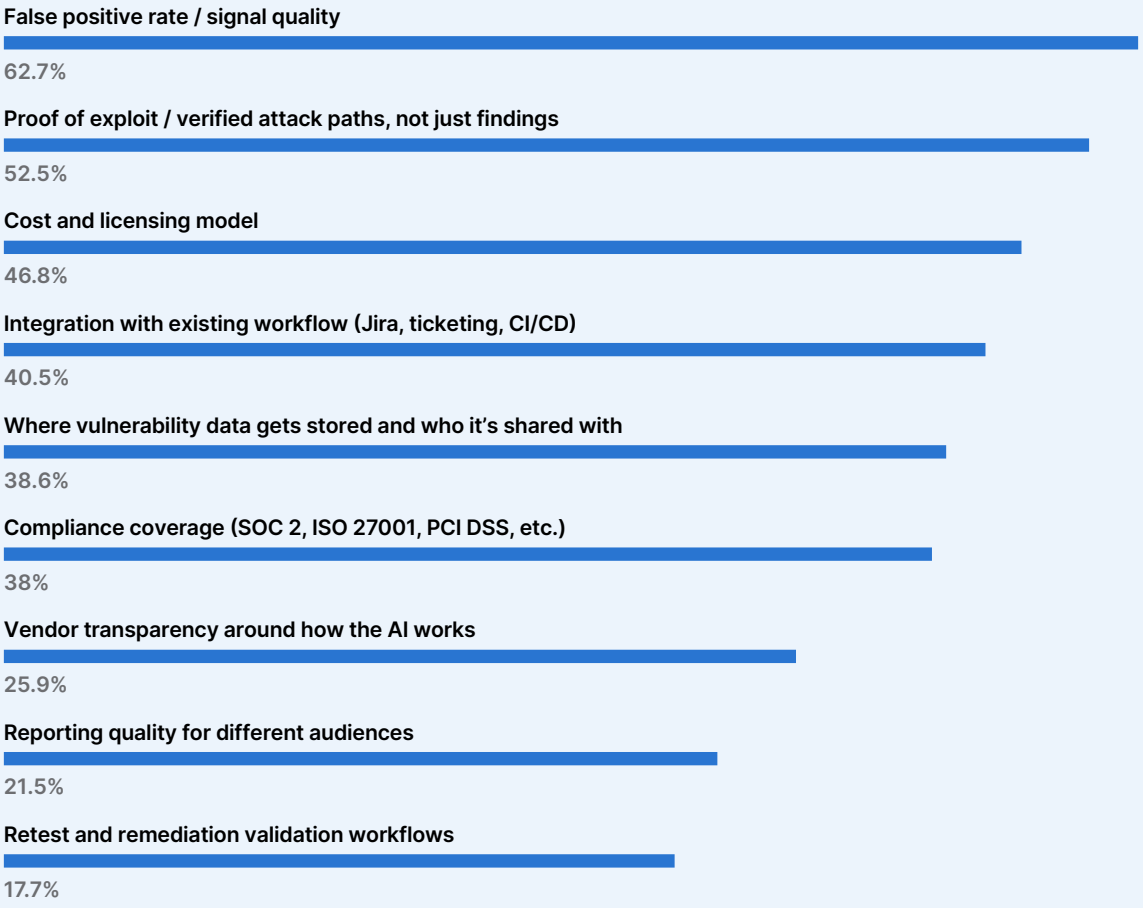
The evaluation criteria practitioners apply to AI-assisted pentesting platforms mirror the frustrations they described.

Asked which factors most influence their decision when evaluating a platform (selecting up to three), they ranked accuracy and proof above price:

- → False positive rate / signal quality: 63%
- → Proof of exploit / verified attack paths, not just findings: 53%
- → Cost and licensing model: 47%
- → Integration with existing workflow: 41%
- → Where vulnerability data gets stored and who it's shared with: 39%

Which factors most influence your decision when evaluating an AI-assisted pentesting platform?

Select up to three | n = 158 security practitioners | Pentest-Tools.com AI pentesting survey, June 2026



Two **accuracy-related factors outrank cost**. That's because practitioners have learned, often the hard way, that a cheap tool producing noise costs more than it saves. Every percentage point of false positive rate translates into **triage hours, and triage hours are the scarcest resource in the whole pipeline**.

The 39% who put data handling in their top three deserve a mention too. **Feeding vulnerability data into AI systems raises its own trust questions** - where the data goes, who can see it, and what the model does with it. For internal security teams, this multiplies compliance risks. For MSPs and MSSPs handling client data, this is a contractual constraint.

The buying criteria close the loop this report opened. Practitioners placed AI where errors are cheap, got buried in validation work where they used it most, learned that fabricated findings poison trust in everything else, and confirmed that the judgment-heavy phases still need humans.

So when they evaluate tools, they don't ask "how much can it find?", but "**how much of what it finds can I trust without redoing the work myself?**"

Closing the validation gap with evidence that holds up

The pattern in this survey isn't going to reverse. AI discovery volume will keep rising. Stakeholder demand for testing frequency will keep increasing too. What can change is the validation side - how fast a team can turn a candidate finding into a confirmed, evidence-backed, prioritized result.

This is where Pentest-Tools.com fits, and we'll be direct about it. What we've built is designed to shrink that validation gap, by making sure fewer junk findings enter the pipeline and the ones that remain arrive with proof attached.

Validation built into the workflow, not bolted on after

Most scanners stop at detection. They tell you something might be exposed, and leave the validation step (confirming whether the exposure is exploitable, under conditions an attacker would actually use) to the security team.

This is the gap vulnerability validation closes. It's the category of tools that confirm which detected issues are actually exploitable - by exploiting them safely, under your control, capturing evidence of what the exploit reached, and re-testing once a fix lands. That way you prioritize, fix, and report based on proven risk, not theoretical severity. Pentest-Tools.com combines detection and validation in one workflow, across web, network, API, and cloud - with evidence you can hand straight to developers, IT, leadership, or an auditor.

Proof you can point to: the Confirmed tag

Here's how that works in practice. When Pentest-Tools.com validates a finding, it marks it Confirmed - and it only earns that label by proving exploitability, using exploit modules written and maintained in-house by certified offensive security practitioners.

Anything that it can't prove stays labeled unconfirmed, so you're never guessing about the certainty behind a result.

Every Confirmed finding arrives with evidence attached: request and response pairs, extracted artifacts like usernames and passwords, and the option to replay the attack yourself. An ML Classifier filters out likely false positives before they reach you. The whole test stays under your control, which is precisely what lets you defend a finding to a developer, IT, leadership, or an auditor, without redoing the work first.

Sniper: Auto-Exploiter closes the validation step automatically for high-impact CVEs, returning request and response data, exploit traces, and supporting artifacts that hold up to client and auditor scrutiny. The output isn't a "potentially vulnerable Apache version". It's a confirmed exploitation path with the evidence attached - the "proof of exploit, not just findings" that 53% of surveyed practitioners named among their top buying criteria.

Less noise entering the pipeline

The survey's most common frustration is noise, so we'll speak to it directly.

The Machine Learning Classifier, built into the Website Vulnerability Scanner, cuts false positives by up to 50%. Fewer false positives entering the pipeline means the team's triage capacity gets spent on findings that matter.

In benchmark testing across 167 vulnerable environments, the Network Vulnerability Scanner ranked first for overall and remote detection accuracy, with near-perfect consistency between the detections it claims and the ones it actually makes.

And the Password Auditor identified valid credentials in 84% of test cases, compared to 15% for Hydra, the most common open-source alternative. That means "weak credential" findings come with proof of exploitability, not just a guess.

Built for the cadence that copes

The survey's clearest operational lesson: **teams that test frequently absorb AI volume**, and teams that test occasionally get buried. Frequent testing requires **strong processes** - scheduled

scans, diffs between runs, findings routed to where remediation already happens, retest artifacts captured automatically.

Vulnerability monitoring runs scheduled assessments against deployed systems, captures what changed between runs, and routes validated findings into Jira, Vanta, or wherever the team already manages remediation. Evidence is captured as a normal part of testing (including before-and-after artifacts on retest) so proving a fix held doesn't become a separate project.

Leadership and auditors receive artifacts, not assurances.

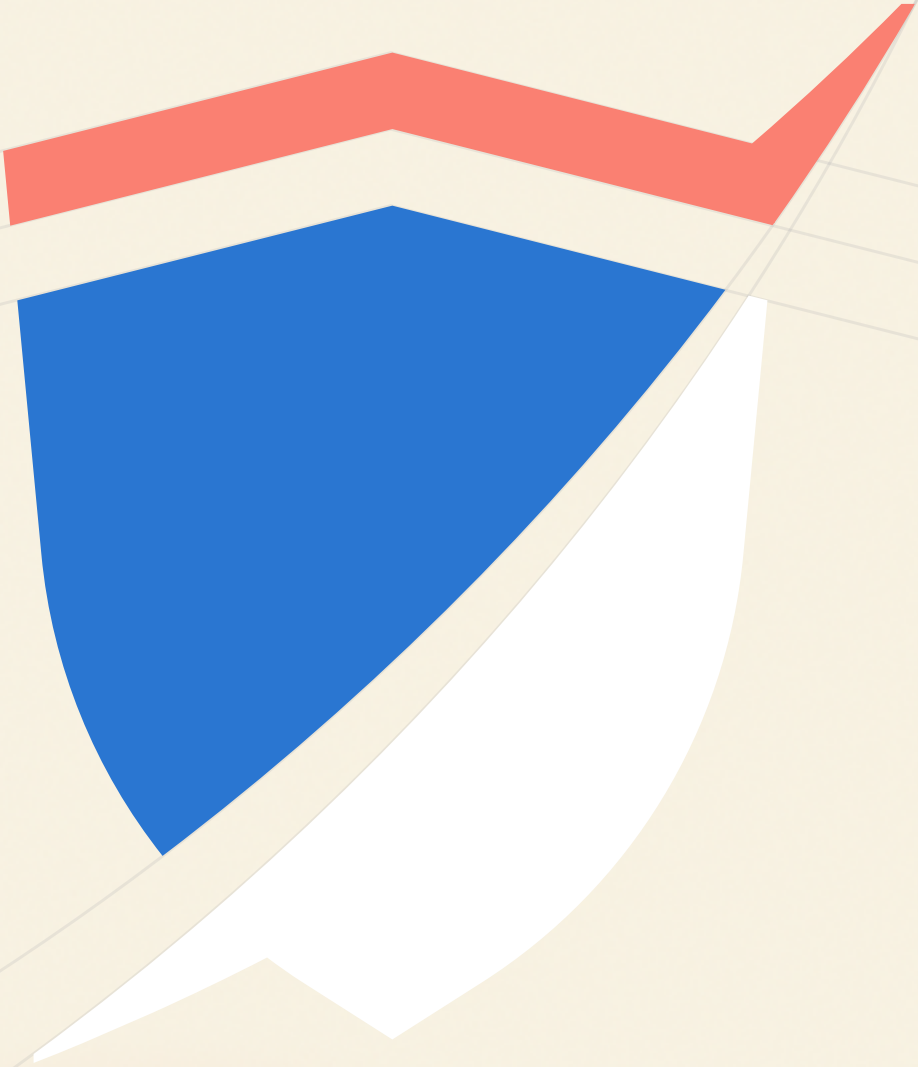
What this means in practice

For an internal security team, a normal week starts to look and feel different:

- → **Triage shrinks from days to hours**: false positives get filtered before a human sees them, so the queue stops looking like "300 findings, 250 junk" and starts holding findings worth someone's time.
- → **Validation stops being your manual job**: high-impact findings arrive already exploit-validated, with evidence attached, instead of "plausible but unverified" results you have to redo before anyone can act on them.
- → **Prioritization runs on proven risk**: you fix what's exploitable first, instead of working down a list ranked by theoretical severity and hoping the order is right.
- → **Retesting happens on schedule**: it produces before-and-after evidence. Proving a fix held is a by-product of the workflow, not a separate project.
- → **Handoffs stop being arguments**: developers get reproducible evidence, instead of a scanner line to dispute. Leadership and auditors receive artifacts, not assurances.

That’s the pattern. The tester keeps what only a tester can do - business logic, attack chaining, judgment calls. What disappears is the chore nobody signed up for: manually re-checking work the machine should have finished.

Explore more insights



Before you act on this report

The questions we hear most often fall into two groups: what the survey data does and doesn't support, and what to look for when evaluating tools in light of it.

We've kept them separate, because the answers come from different places - the first group from the data, the second from our experience building for this problem.

What the survey data can and can't tell you

Can AI replace our pentesters?

> Not on the evidence here. The survey's practitioners - the people using these tools daily - are unambiguous that business logic flaws, creative attack chaining, and contextual risk judgment remain human territory. The realistic model is division of labor: automation handles breadth, repetition, and evidence capture; testers handle intent, creativity, and the attack paths no pattern matcher sees. Teams get into trouble when they assign the wrong half to the wrong party.

Our team is small. We can't absorb hundreds of findings per engagement

> Neither can most teams - only 20.3% of survey respondents have a workflow for that volume. The data points away from building a bigger triage function and toward refusing unvalidated volume in the first place: fewer, confirmed findings beat a long list of candidates. A small team with 30 confirmed findings outperforms a large team with 500 unverified ones.

Should we be testing our own AI systems?

> The survey suggests you soon won't have a choice. 92.4% of practitioners either already test AI-powered systems or plan to within 12 months, and stakeholder awareness of AI-assisted attacks is already driving demand for more frequent testing.

The teams ahead of this are moving AI systems from “tested on request” to default scope - and putting shadow AI usage on the assessment checklist before an incident does it for them.

A small team with 30 confirmed findings outperforms a large team with 500 unverified ones.

With or without AI?
Choosing tooling in light of this

We tried an AI pentesting tool and drowned in false positives. Why would this be different?

That experience is the norm, not the exception - it’s the single most common frustration in this survey. The honest answer is that tools differ enormously on signal quality, and the difference is measurable.

Ask any vendor for benchmark data on detection accuracy, not just detection availability, and for validated findings with evidence attached rather than candidate findings. If they can’t show the difference between what they claim to detect and what they actually detect, that gap becomes your triage backlog.

Won’t more validation slow us down?

The survey data points the other way. The slow path is the current one: fast discovery followed by days of manual triage. Validation that happens automatically, at the moment of detection, with evidence attached, is what lets remediation start immediately. The teams coping best in this survey aren’t doing less validation. They’re doing it earlier, with processes instead of overtime.

How do we know AI-generated findings will satisfy a client or an auditor?

A finding satisfies scrutiny when it comes with evidence: proof the vulnerability existed, proof it was exploitable, proof of who validated it and how, and proof the fix held on retest.

The source of the finding matters less than the artifact trail behind it. If your current AI tooling produces findings without reproducible evidence, that's the gap to close before the next audit or client review - not after.

Vulnerability validation. Because detection alone proves nothing.

The story this survey tells isn't that AI failed pentesting.
If anything, AI succeeded - but only at one half of the job, and the half it conquered was never the bottleneck. Finding vulnerabilities has become cheap; proving them is getting more expensive.

Finding vulnerabilities has become cheap. Proving them hasn't.

No survey can tell you what to fix first in your own environment. What this one can tell you is where 158 of your peers feel the pressure: at the seam between what AI finds and what a human can prove. Start there.

Pentest-Tools.com

Security findings you can prioritize.
In web apps, network, and cloud.

Pentest-Tools.com is the only vulnerability validation product that combines detection and validation in one workflow across web, network, and cloud targets - with evidence practitioners can hand directly to developers, IT, leadership, and auditors.

It discovers security issues across **web, network, and cloud** targets and **validates** them automatically in the same workflow. When it proves exploitability, it marks the finding as Confirmed, backs it with **evidence** - such as valid credentials, file access indicators, and screenshots - and lets the practitioner replay the attack.

Unlike legacy scanners that stop at detection and leave practitioners to prove exploitability by hand, Pentest-Tools.com validates as it tests - keeping the practitioner in full control of every stage with deterministic capabilities and automating for efficiency, depth, and speed with AI only where it makes sense.

With Pentest-Tools.com in their workflow, security and IT teams spend less time on manual triage - and walk away with findings they can defend to leadership, auditors, developers, and IT, and a basis for prioritizing remediation on proven risk, not theoretical severity.

Discover more on **pentest-tools.com/product**

Pentest Tools

Europe, Romania, Bucharest
48 Bvd. Iancu de Hunedoara
support@pentest-tools.com
Pentest-Tools.com

Join our community on:
LinkedIn | Youtube | Reddit