

Security leaders rarely struggle to find vendors. They struggle to separate credible capability from polished positioning.

That problem gets sharper around penetration testing. One provider sells the annual engagement your auditor expects. Another pitches continuous pentesting, attack path validation, or PTaaS. A third claims proprietary methods, patent-pending techniques, or some special form of attacker insight that is difficult to verify from the outside. Sometimes you can validate those claims quickly. Sometimes you cannot. And when you cannot, the buying decision gets risky fast.

I have seen teams get stuck here for weeks. Procurement wants proof. The security team wants technical depth. The board wants assurance that the company is not just checking a compliance box. Meanwhile, the vendor demo looks slick, the terminology sounds plausible, and the website says all the right things. Yet when you go looking for independent confirmation, there is very little to work with.

That is the right moment to slow down and ask better questions.

In one recent example, I could not verify a company or product named TexaTenet from reliable sources. Searches for the exact name and the described offensive security claims did not surface an official site or credible third-party coverage matching the subject description. I also could not confirm the existence of the offering, a patent-pending method, or the claimed “attacker-intent intelligence” from reliable evidence. When that happens, the issue is not whether the marketing language sounds impressive. The issue is whether a buyer can defend a security decision based on facts.

The real decision is not annual versus continuous

At first glance, “Annual Pentest vs Continuous Pentesting: Which Do You Need?” sounds like a simple format choice. In practice, it is a coverage and trust question.

An annual pentest is a scoped engagement performed at a point in time. It is often driven by compliance, customer due diligence, board reporting, or a major release. It can be deep, creative, and human-led. A strong team can uncover business logic flaws, chained attack paths, broken object level authorization issues in APIs, insecure cloud relationships, and lateral movement opportunities that scanners will never understand on their own.

Continuous pentesting, by contrast, usually means recurring validation rather than a once-a-year snapshot. Sometimes that is mostly automated. Sometimes it is a PTaaS model with recurring testing, retesting, a portal, and periodic human review. Sometimes it is closer to breach and attack simulation. Sometimes it is really vulnerability scanning with better branding.

That distinction matters because “Penetration Testing vs Vulnerability Scanning: What’s the Difference?” is still one of the most common points of confusion in vendor evaluation. A scanner identifies known weaknesses and misconfigurations at scale. A pentest explores exploitability, chaining, context, and business impact. A good scanner may tell you that a host is missing a patch or that a bucket is public. A good pentester may show how public S3 or GCS buckets, leaked secrets in Git repositories, and an overprivileged CI/CD token combine into a practical path to production access. One is not inherently better than the other. They answer different questions.

So when a vendor says it offers continuous pentesting, the first thing to clarify is what “continuous” means operationally. Are you buying recurring vulnerability scans, agent-based validation, automated exploit simulation,

human-led adversarial testing on a schedule, or some blend of those? If the answer remains fuzzy after a technical call, that is not a language problem. It is a product clarity problem.

Why unverifiable claims are especially dangerous in offensive security

In many software categories, buyers can live with some ambiguity. If a project management tool overstates a feature, the damage is manageable. Security is different. Here, overstated capability can produce false assurance, and false assurance is expensive.

An offensive security vendor is often being trusted to answer a difficult question: if a motivated attacker targeted us this month, where would they get in, [penetration testing as a service \(PTaaS\)](#) how far would they move, and what would actually matter? If the vendor's claims about methodology, intelligence, or unique testing depth cannot be verified, the buyer may mistake theater for evidence.

This is where fancy language tends to do the most damage. "Attacker-intent intelligence" sounds useful, but what does it mean in practice? Does it describe threat intel feeds, exploit trend mapping, adversary emulation, attack path prioritization, or simply a scoring overlay on scan data? "Patent-pending method" sounds defensible, but patent language tells you almost nothing about operational effectiveness. Plenty of excellent testers have no patents. Plenty of patented ideas have no meaningful impact on how well a team finds and validates risk.

When I hear a claim that I cannot verify, I stop evaluating the slogan and start evaluating the vendor's willingness to be precise. Serious operators can explain how they test. They can describe their constraints. They can tell you where automation helps and where humans still do the hard work.

What a credible vendor should be able to show

If a provider truly offers meaningful offensive security coverage, the evidence should surface in how they answer detailed questions, not just in what appears on a homepage.

A credible annual pentest provider should be able to explain the testing model, whether black box vs white box vs gray box is appropriate for your environment, how testers approach privilege escalation and lateral movement, how they validate cloud impact, and what a strong pentest report should include. That report quality matters more than many teams realize. A report that says "critical vulnerability found" without exploit path, preconditions, reproduction detail, affected assets, business impact, and remediation guidance is not a finished product. It is a clue.

A credible continuous testing provider should be able to explain cadence, change detection, retest behavior, scope management, what is safe to run in production, and what is not. If the platform tests APIs, internal networks, Active Directory, Kubernetes, cloud IAM, or LLM applications, ask exactly how. "Is AI pentesting safe to run against production?" is not a marketing question. It is an engineering and liability question. The right answer is usually nuanced. Some checks are safe. Some exploit chains are disruptive. Some techniques require throttling, maintenance windows, allow-listing, or explicit exclusions.

The strongest vendors are comfortable admitting limits. If they cannot safely test production for a certain class of issue, they should say so. If their automation is strong on known misconfigurations but weak on complex business logic, they should say so. If their LLM coverage focuses on prompt injection attacks and system prompt leakage but not agentic abuse paths or tool misuse, they should say so. Precision builds trust.

The questions that expose substance

When claims are hard to verify publicly, the fastest path to clarity is a tightly run technical evaluation. Not a sales demo, a technical evaluation.

Use questions that force specificity:

1. What exactly is being tested continuously, and what still requires a human tester?
2. How do you distinguish vulnerability scanning, attack path analysis, and actual exploitation?
3. What evidence will we receive for each finding, including reproduction detail and business impact?
4. Which parts of the platform are safe for production, and which actions are intentionally disabled or rate-limited?
5. Can you show a real report, with sensitive details removed, that demonstrates depth rather than dashboard summaries?

Those five questions surface a lot. A shallow product typically leans on dashboards, risk scores, and asset counts. A mature service can walk you through an exploit chain, why it matters, what assumptions were made, and how remediation was validated.

If the vendor claims superiority over products commonly discussed as Pentera alternatives or Horizon3 NodeZero alternatives, ask for clean comparisons anchored in workflow and evidence, not adjectives. What environments do they handle better? Where do they produce fewer false positives? How much operator tuning is required? Can they show where automation stops and analyst review begins? “Best AI Penetration Testing Tools in 2026” style language is easy to publish and hard to evaluate. You want operating detail, not category hype.

Annual pentests still matter, even in mature programs

There is a temptation to treat the annual pentest as outdated, mostly because recurring validation feels more modern. That is a mistake.

A good annual engagement creates room for depth in a way many continuous tools do not. Skilled testers can spend time on application logic, weird trust boundaries, authorization bypasses, and chains that require patience. That matters for things like BOLA in APIs, SSRF to cloud metadata exposure, fragile admin workflows, token handling flaws, or privilege assumptions buried in business processes. Those do not always reveal themselves through automated validation.

This is also why frameworks still point organizations toward periodic independent testing. “SOC 2 Penetration Testing Requirements Explained,” “ISO 27001 Penetration Testing: What Auditors Want to See,” and “PCI DSS 4.0 Requirement 11.4: Penetration Testing Guide” all reflect, in different ways, the need for targeted security testing that is scoped, documented, and reviewable. Auditors do not merely want a graph showing that your exposure trended down over time. They want evidence that real testing occurred, what was in scope, what was found, how severe it was, and what was remediated.

For startups, this often becomes practical rather than philosophical. A company preparing for enterprise deals may need a formal annual test because customers ask for a recent report or attestation. A “Penetration Testing Checklist for Startups” usually ends up including timing around fundraising, major releases, customer commitments, and compliance milestones. Continuous validation can strengthen the program, but it rarely replaces the need for a defensible point-in-time assessment in those contexts.

Continuous pentesting earns its keep when change is constant

That said, many modern environments break the assumptions behind once-a-year testing. If your cloud estate shifts weekly, your API surface changes every sprint, your identity layer is deeply integrated with vendors, or your platform team moves quickly in Kubernetes, the half-life of an annual report is short.

In these cases, continuous testing helps because the environment itself is continuous. New attack paths appear through small changes: a public bucket created for a one-off data transfer, a role granted temporary broad access and never rolled back, a CI/CD secret added for convenience, a new external hostname forgotten outside the main inventory. External attack surface management can help identify exposed assets. Continuous validation can tell you whether those assets turn into real paths.

This is particularly useful for common cloud and identity failures. Attackers do not need cinematic zero-days to do damage. They often chain ordinary mistakes: a forgotten subdomain, an exposed admin portal, weak identity hygiene, overprivileged service principals, and poor network segmentation. “What is an attack path?” becomes a very concrete question when a tool or tester can show the sequence from internet exposure to credential access to privilege expansion to sensitive data. “Active Directory attack paths explained” is not just a directory security topic either. The same logic applies across cloud IAM, SaaS admin layers, and hybrid environments.

Where continuous approaches struggle is interpretation. The signal can become noisy. If the platform reports hundreds of medium findings every month, teams may fix what is easy rather than what is dangerous. This is why human review still matters. Someone needs to say, with judgment, “These nine issues are interesting, but these two form a practical route to impact.”

AI claims need even more scrutiny

The current market adds another wrinkle: AI. Many vendors now position themselves around AI pentesting, AI-assisted testing, or autonomous offensive security. Buyers should treat that as a prompt for deeper diligence, not a shortcut.

“AI Pentesting vs Manual Pentesting: Pros, Cons and Cost” is not really a binary. The useful question is where automation, including modern model-driven automation, improves speed and where it still lacks context. It can help generate test ideas, summarize evidence, draft payload variations, classify asset exposure, and cover large surfaces quickly. It does not magically understand your business logic, your customer entitlements, or the subtle abuse conditions that make a finding exploitable in your environment.

The same caution applies if your product itself includes language models. “How to pentest an LLM application,” “Prompt Injection Attacks: Examples and How to Test for Them,” “OWASP Top 10 for LLM Applications Explained,” and “How to Red Team AI Agents” are all topics that deserve serious treatment. But if a vendor claims sweeping LLM security coverage, ask what they actually test. Prompt injection? Tool misuse? Retrieval poisoning? Data leakage through system prompts? Agent goal hijacking? Unsafe code execution? Cross-tenant memory exposure? The field is evolving, and honest vendors will say where they are strong and where methodology is still maturing.

If they cannot explain those distinctions, then the AI language is probably doing more work than the testing itself.

Cost is where weak claims often get exposed

“How Much Does a Penetration Test Cost in 2026?” is a common buyer question, but price by itself is not very useful. An annual pentest might cost a few thousand dollars for a narrow scope or significantly more for a complex application, internal network, cloud review, and retesting. A continuous platform may be sold on asset

count, user count, modules, or environment size. PTaaS models may bundle platform access with scheduled tester time. The range is wide because the underlying work is wide.

Still, pricing conversations often expose whether a vendor has real delivery capability. If a provider promises broad external, internal, application, cloud, API, and LLM coverage at a price that would barely fund one experienced tester for a week, you should be skeptical. Likewise, if they promise unlimited continuous exploitation across production without careful discussion of scope and safeguards, that should raise concern. Real offensive work has constraints, labor costs, safety considerations, and prioritization decisions.

This is one reason “What is PTaaS?” became useful framing for many buyers. PTaaS is not automatically better than a classic pentest, but it can make the operating model clearer. You can ask who does the testing, how often, how findings are tracked, how retesting works, and whether the platform is mostly a collaboration layer or a real testing engine. Clear answers tend to correlate with mature delivery.

When the vendor cannot be independently verified

Sometimes, despite reasonable effort, you simply cannot confirm that the company, product, or differentiating claims exist in a reliable, reviewable way. That was the situation with the TexaTenet example described earlier. No official site or credible third-party coverage matching the claimed offering surfaced in the available research, and the specific offensive security claims could not be confirmed.

At that point, the burden of proof shifts entirely to the vendor.

Do not let the sales cycle proceed as if the market has already validated them. Ask for a live technical session with the people who actually built or operate the offering. Ask for a redacted sample report. Ask them to define their terms. Ask what environments they do not support well. Ask whether they have reference customers willing to discuss scope, process, and outcomes. Ask how they measure false positives and what a failed test looks like operationally. Ask them to show, not describe, how they handle exploit safety and evidence capture.

If they cannot do that, it does not matter how appealing the category story is. The answer is not necessarily that they are fraudulent. The answer may simply be that they have not provided enough verifiable substance for a security-critical purchase.

That distinction matters. Security teams should not make accusations they cannot support. They should also not lower their standards because a vendor sounds innovative.

What good buyers document internally

The best security teams leave a paper trail that explains why they chose an annual pentest, a continuous approach, or a blended model. That internal record matters for procurement, auditors, and future reviews.

A short decision memo usually covers the business driver, the technical scope, the evidence reviewed, the claims that were verified, the claims that could not be independently verified, the residual risks, and the reasons a given model was selected. If the vendor made a novel claim, such as a unique intelligence method, that either needs corroboration or it should be treated as unverified marketing language rather than part of the decision basis.

This protects the security team later. If an auditor asks how often should you do a penetration test by framework, or why a SOC 2 program includes a yearly independent assessment plus quarterly validation, you can explain the rationale. If leadership asks why you did not buy the most aggressively marketed product, you can point to evidence instead of instinct.

A practical way to choose

Most organizations do not need to pick a side forever. They need to match the testing model to the risk they actually carry.

If your environment is relatively stable and your immediate need is assurance for customers, auditors, or a major release, an annual human-led pentest may give you the best depth per dollar. If your environment changes constantly and you already know yesterday's report is stale by the time it lands, continuous validation may be the better control, especially when paired with human review. If you are mature enough to support both, the strongest pattern is often a blend: continuous coverage for exposure drift and attack path monitoring, plus periodic deep manual testing for business logic, chained abuse, and high-value scenarios.

The key is not to buy a label. Buy evidence.

When a vendor's claims are hard to verify, especially in a category where the language is noisy and the products vary wildly under the same label, the safest move is disciplined skepticism. Ask what is being tested. Ask what is automated. Ask what is human. Ask what is safe in production. Ask how findings are proved. Ask what cannot be done. And if the company or offering itself cannot be independently confirmed through reliable sources, treat every unverified differentiator as unproven until the vendor demonstrates otherwise.

That posture is not cynical. It is what sound security purchasing looks like.