Engineer

Penetration test

an authorised attack that shows which flaws can be exploited

What it is

A penetration test, often shortened to pentest, is security testing in which the testers mimic real-world attacks to find ways around the security features of an application, a system or a network. NIST's Technical Guide to Information Security Testing and Assessment, SP 800-115, notes that this often means launching real attacks on real systems and data, with the tools and techniques that attackers use. The shorter definition in SP 800-53 Rev. 5, NIST's catalogue of security and privacy controls, adds that the assessors typically work under specific constraints, and that is what separates a test from an attack: it is authorised, bounded and agreed in writing before it starts.

What sets it apart from other security testing is the aim. A vulnerability scanner compares what it sees with descriptions of known flaws and reports what might be there. A penetration tester tries to use the flaw. SP 800-115 puts the difference plainly: scanners check only for the possible existence of a vulnerability, while the attack phase of a penetration test exploits it to confirm that it exists. Most tests also look for combinations, weaknesses on one or several systems that together give more access than any one of them would alone.

The OWASP Web Security Testing Guide (WSTG) notes that the technique also goes by the names black-box testing and ethical hacking. The team tests from the outside, without prior knowledge of the target's internal design, and behaves as an attacker would, although it is often given one or more valid user accounts.

Two choices shape every test. The first is the starting point. An external test begins outside the network with little more than the addresses in scope; an internal test begins inside it, usually with the access of an ordinary employee, and

tries to climb from there by privilege escalation. The second is who knows. SP 800-115 calls overt testing white hat testing: it runs with the knowledge and consent of the IT staff. Covert testing, or black hat testing, runs without the IT staff knowing but with the full knowledge and permission of upper management.

SP 800-115 divides the work into four phases, and adds in a footnote that this is one of many acceptable ways to group it. In planning, the rules are agreed, management approval is documented and the goals are set; the testing itself starts in discovery. Discovery gathers information about the target, scans it, and compares what it finds with vulnerability databases such as the US National Vulnerability Database and with the testers' own knowledge. The attack phase, which the guide calls "at the heart of any penetration test", tries to exploit each candidate flaw. When an exploit opens new access, the testers loop back into discovery from the new position. Reporting runs alongside all three and ends with a report that lists the vulnerabilities found, rates their risk and says how to fix them. Other methods split the same work differently: the Penetration Testing Execution Standard (PTES), which the WSTG lists among its reference methodologies, uses seven phases, from pre-engagement interactions to reporting.

Use the concept in a decision

Put "What is a penetration test?" to work in a real decision. We can help relate the concept to the system you are building or buying. Accounts, passwords and permissions arranged so people hold exactly the access their job needs, and leavers lose it that day.

Explore our services Get in touch →

Test the assumptions behind the term

Use the explanation and linked work to frame questions for your project. We can examine the assumptions with you before a tool is chosen. Wrote 1,000 RBAC rules to secure Linux servers — 1,000 access rules written to close a security gap

Wrote 1,000 RBAC rules to secure Linux servers Get in touch →

Why it matters

A scanner's report is a list of possibilities. Some of them cannot be exploited in your setup, and a scanner cannot see a flaw that nobody has described, or one that appears only when two harmless-looking weaknesses are used together. A penetration test answers a different question: what could an attacker actually do here, and how far could they get? SP 800-115 adds what else it can show: how well a system stands up to real attack patterns, how skilled an attacker would need to be, which further countermeasures would help, and whether the defenders notice the attack and respond.

That is why its findings carry weight with people who never read scanner output. A finding that says an outsider without an account read the customer table is evidence, not a probability. SP 800-115 describes penetration testing as an iterative process that uses minimal access to gain greater access, and a good report shows that chain step by step, so the fix can break it at its cheapest link.

The same realism makes it costly and risky. NIST calls it labour-intensive, says it needs great expertise, and warns that systems may be damaged or made inoperable during the test; experienced testers reduce that risk, but it can never be fully eliminated. Many of the things a tester does, such as attacking systems to compromise them, are activities an organisation's own policies normally forbid. The rules of engagement (ROE) exist for that reason: a document, agreed before the test begins, that gives the test team authority to carry out defined activities and nothing beyond them.

It is also one technique among several, not a verdict on its own. The WSTG observes that many teams make penetration testing their main way of checking a web application's security, and argues that it should be neither the main check nor the only one. Web applications are almost always bespoke, so testing them from outside is closer to research than to finding known flaws in standard products, and the test arrives late in the development life cycle, after the design and code are settled. When the testers have the finished application but no access to its source code, the guide adds, a penetration test is still clearly better than leaving the application untested.

Frequency follows from the cost. SP 800-115 suggests that, because of its high cost and potential impact, testing an organisation's network and systems once a year may be sufficient, with regular, lighter scanning in between. Some standards also ask for it directly: the WSTG notes that PCI DSS, the payment card industry's data security standard, defines its own penetration testing requirement, covering both external and internal testing.

What it looks like in practice

Most of a penetration test is decided before the first request is sent. Appendix B of SP 800-115 is a template for the rules of engagement, with sections for the purpose and scope, the people and the test schedule, how the parties communicate and when testing halts, the systems in scope and those excluded, the activities allowed and forbidden, how test data is handled, how results are

reported, and a page of signatures. A trimmed version for an external test of a web shop might read as follows. It was written for this article, and the names and addresses come from the ranges reserved for documentation.

```
RULES OF ENGAGEMENT: external test of shop.example.com

Purpose       Find which flaws an outsider can exploit, and how far
              each one leads.
Targets       203.0.113.10 to 203.0.113.20, ownership checked in WHOIS
Exclude list  203.0.113.15 (payment provider, not ours to test)
              Every third-party service the shop calls
Accounts      Two customer accounts and one support account
Schedule      3 to 14 November, 19:00 to 06:00 CET, outside trading peaks
Tester IPs    198.51.100.7 and 198.51.100.8 only
Allowed       Exploiting a confirmed flaw to show its impact; reading
              data that belongs to the test accounts; uploading one
              marked test file, removed at the end
Forbidden     Denial of service; changing or deleting customer data;
              social engineering; physical entry
Halt          Outage of the live shop, real customer data in view, or
              signs of a real intruder: stop, phone the security lead
Resume        Only on written word from the security lead
Test data     Sensitive; kept on the test team's laptops, deleted 30
              days after the final report
Reporting     Daily status note; final report within ten working days
Signed        Test team leader; chief information security officer
```

Each line answers a question that SP 800-115 raises. The guide says to verify that every public address in the target list belongs to the organisation before testing begins, and points to WHOIS registration data for that. It suggests testing an operational site in the evening rather than during peak business hours. It asks for the addresses the testers work from, so that administrators can tell test traffic from a real attack. It calls the list of systems that may not be touched the exclude list, and says the rules should give criteria for halting the test and a process for resuming it. It also asks what happens if the testers find an actual intruder: should testing stop, and when can it start again, on whose authority? At a minimum, the test team leader and senior management sign, stating that they understand the scope and its limits.

Under rules like these, the attack phase follows the pattern the WSTG gives for SQL injection. A scan of the shop’s search page fingerprints a possible flaw. The first evidence is a database error when a single quote is typed into the field. The tester then sends crafted input that changes the grammar of the query, and confirms the flaw only when the query returns data it should not. If that data

included the support account's session, the tester would sign in as support, look again from there, and the report would show the whole path: one unfiltered field, one leaked session, one support view of every order.

A test like this sits beside other checks, not in place of them. The write-up of a continuous security scanning programme from the founder's work under NextMariner describes static analysis, dynamic testing against a live instance and dependency checks running automatically, alongside written security audits that include a threat model and an assessment against the OWASP testing guide. The page gives the reason for that pairing: scanners find the classes of problem that someone has already described, and a threat model is where the problems specific to one platform get named.

Where it goes wrong

Testing without written authorisation. A verbal "go ahead" from a team lead does not cover attacks that company policy forbids, and it protects neither the testers nor the people who asked for the test. Write the rules of engagement first, and have the test team leader and senior management sign them before any traffic is sent.

Testing addresses that are not yours. A target list copied from an old diagram can include addresses that now belong to someone else, and shared platforms put other customers next to yours. Check that every public address in scope belongs to the organisation, as SP 800-115 asks, and put each third-party service on the exclude list by name.

Selling a scan as a penetration test. A scanner report with a new cover page lists possibilities, not proof. Ask for evidence of exploitation for every finding: what the tester did, what access it gave, and what the next step would have been.

Making it the only security testing. A single test sees a sample of the risks, at one moment, from outside. The WSTG places threat modelling in design, code review and source analysis during development, and penetration testing in the test stage that follows. Run all of them, and use the penetration test to confirm what the earlier checks could not.

Starting with no rule for stopping. If the testers take down a live service or find signs of a real intruder, and nobody has agreed who halts the test and who restarts it, people improvise under pressure. Put halt criteria, a call tree and a resume process in the rules of engagement.

Treating the results as ordinary data. A penetration test report is a map of what an adversary could exploit, and SP 800-115 says test results should be treated as sensitive. Agree how findings are gathered, sent, kept and destroyed before the test starts, and limit who receives the full report.

Closing findings without a re-test. A fix can be partial, or land in a different place from the flaw. The WSTG calls testing that tries to exploit vulnerabilities found in earlier reviews focused penetration testing, and notes that it shows whether specific flaws are actually fixed in the deployed code. Re-run the original exploit against every fix before the finding is closed.

Expecting a covert test to find everything. According to SP 800-115, covert testing examines the damage an adversary could cause; it does not test every control or find each vulnerability, and it usually gets in through the most basic weaknesses. Choose an overt test when the aim is coverage, and a covert one when the aim is to learn whether the defenders notice.

Further reading

- NIST SP 800-115: Technical Guide to Information Security Testing and Assessment — NIST's guide to planning and running security tests, with the four phases of a penetration test and a template for the rules of engagement.
- OWASP Web Security Testing Guide v4.2: Introduction — where penetration testing fits beside review and source analysis, and why it should not be the only technique.
- OWASP WSTG v4.2: Penetration Testing Methodologies — a short overview of PTES, the PCI DSS guidance, PTF, NIST SP 800-115 and OSSTMM.
- NIST SP 800-53 Rev. 5: Security and Privacy Controls for Information Systems and Organizations — NIST's catalogue of security and privacy controls, where control CA-8 covers penetration testing and its rules of engagement.

Glossary

Choose for your actual constraints

Bring the deployment path, the operational risks and the people on call. We can scope a change with checks, a recovery route and clear ownership.

See what it costs Get in touch →

Make the terminology practical

Ask how “What is a penetration test?” applies to your organisation. A concrete example from your current system is enough to start the discussion.

Explore our services

welcome@engineer.company +45 50 10 61 84 Copenhagen, Denmark

Take this page with you

Contact card

Carefully Crafted With Love by D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

Copyright © 2025 – present · Engineer ApS · All rights reserved.

Printed from https://engineer.company/glossary/penetration-test/