The five stages of a pentest.

Every serious penetration test runs through the same five stages. First, recon: gather information about the target. Second, scanning: find the services and weak points. Third, exploitation: try to use those weak points to get in or cause harm. Fourth, post-exploitation: work out how much damage a real attacker could do from there. Fifth, reporting: turn it all into something your team can act on. The well-known frameworks (PTES, OSSTMM, NIST SP 800-115) are all versions of this same shape.

HOW IT WORKS

01 Stage 1: Reconnaissance

The tester gathers information about the target, either from a distance (passive) or with light touching (active). The output is a map of what is in scope and worth attacking.

Passive recon uses public sources: WHOIS records, DNS history, search results, social media, archived pages, code repos, certificate logs. The target sees nothing.

Active recon touches the target: pings, banner grabs, web crawling, light port probing. It shows up in logs but does not break anything.

For application jobs this stage is short, because you hand over a URL and a login. For external network or red team jobs, it is one of the longest stages.

02 Stage 2: Scanning and enumeration

The tester maps the target's exposed surface in detail. The output is a list of services, technologies, accounts, endpoints, and known weak spots to look into.

Typical work:

- Port scanning to list services.
- Working out each service's exact version.
- Scanning for known CVEs against those versions.
- Crawling the web app to find endpoints.
- Probing the login and access checks.
- Listing configuration details.

SOURCES

[1] NIST SP 800-115 Technical Guide to Information Security Testing

[2] PTES Penetration Testing Execution Standard

[3] OSSTMM Open Source Security Testing Methodology Manual

[4] OWASP Web Security Testing Guide

This is where automated scanners run, as one tool among many. Their output feeds the next stage; it is not the deliverable. The tester reads the scanner output by hand, throws out false alarms, and decides what to chase.

03 Stage 3: Exploitation

The tester tries to use the weak spots from stage 2 to get in, do something they should not, or reach data they should not see. The output is confirmed weaknesses, with evidence.

For each possible finding, the question is simple: can this really be exploited in normal conditions, or is it only a theory?

Real exploitation includes:

- Building payloads tuned to the target's actual setup, not generic ones.
- Chaining several small findings into one bigger one.
- Testing business-logic flaws no scanner can find.
- Checking login and access limits with more than one account.
- Recording evidence (request, response, screenshot, video) as each finding is confirmed.

If a finding cannot be reliably reproduced, it does not go in the report.

04 Stage 4: Post-exploitation

Once the tester has access or impact, they work out the real blast radius: what an attacker who got this far could actually do, what data they could reach, what else they could move to, and how long the access would last.

For application jobs, this stays inside the app: what data, which other users, which admin features. For network or red team jobs, it extends to moving sideways, staying in, and reaching your real crown jewels.

The rules of engagement matter most here. The tester does not actually steal production data, change important records, or leave a backdoor behind. The blast radius is documented, not carried out. Your incident-response team should still treat it as serious.

05 Stage 5: Reporting

The findings are written up for two readers: the executive who needs the business risk, and the engineers who have to fix things.

A good report has:

- An executive summary in plain words.
- A scope and method section.
- The findings, each with a title, severity, the affected part, evidence, real-world impact, and how to fix it.
- A fix plan ordered by risk and effort.
- Appendices with raw evidence and references.

Reporting also includes the debrief: a meeting where the tester walks you through the report, answers questions, and agrees the next steps. The retest of fixes is sometimes called a sixth stage; we fold it into reporting, because it produces an updated report. See pentest report formats for more.

06 Which frameworks describe this?

Three come up often:

- NIST SP 800-115 (Technical Guide to Information Security Testing). The US government's reference. It uses four phases: planning, discovery, attack, reporting.
- PTES (Penetration Testing Execution Standard). A community framework that goes deeper into each phase, especially intelligence gathering and threat modelling.
- OSSTMM (Open Source Security Testing Methodology Manual). A more formal framework with measurable security metrics.

The words differ; the shape is the same. Do not get stuck on which framework name is in a proposal, as long as the work covers the five stages above.

All five stages, every engagement.

securelayer7.net/learn/pentest/pentest-methodology-stages

Open online