

How a Pen Test Actually Runs: Discovery, Attack, and Leaving No Trace

Fri, Aug 14, 9:00 AM EDT · <https://scottslab.io/posts/how-a-penetration-test-runs>



TL;DR

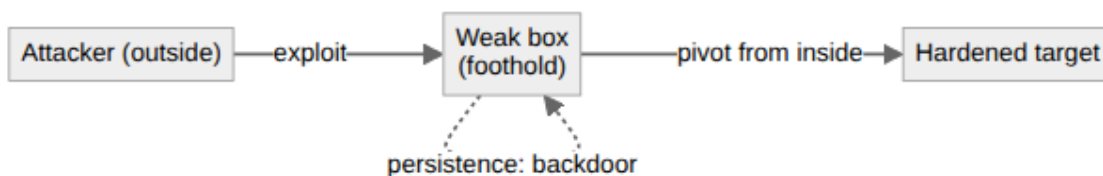
NIST models a pentest as an iterative loop between discovery and attack, not a straight line. You recon, you break in, and what you find inside sends you back to recon. Discovery is reconnaissance (passive and active, OSINT, footprinting, even war driving and war flying). The attack phase is gain access → escalate privileges → move laterally, with pivoting (using a weak box as a foothold to reach hard targets) and persistence (backdoors that survive the original hole). Then you clean up everything and restore what you touched.

A real engagement isn't a single shot; NIST frames it as a loop between discovery and attack that you run over and over. You learn something, you use it to get in, and getting in reveals new things to learn. So you drop back to discovery from a better vantage point. The map you draw from outside is nothing like the one you draw from a foothold inside.

Discovery is reconnaissance, and it comes in flavors. Passive recon watches without touching: OSINT from public sources, footprinting the organization's exposed surface. It leaves no trace. Active recon actually probes, which is louder but richer. It even goes physical: war driving to find wireless networks from a car, or war flying to do the same from a drone. The goal throughout is to build the attack surface before you attack it.



The attack phase is the part people picture, and it has a shape: gain access through some weakness, escalate privileges from whatever you landed as to something powerful, then move laterally to reach what you actually came for. Two techniques carry a lot of the weight. Pivoting is using a system you've already popped, often a soft, low-value box, as a foothold to reach the hardened targets you couldn't touch from outside. The weak machine becomes your inside vantage point. Persistence is installing a way back in that doesn't depend on the original exploit: a backdoor. So when the hole you came through gets patched, you're still there.



The part amateurs skip is the ending. A professional pentest cleans up after itself: remove the backdoors, delete the tools and artifacts, and restore every modified system to its pre-test state. You were simulating an attacker, not becoming a liability. Leaving persistence mechanisms or altered configs behind turns your security assessment into the very thing it was supposed to find. Test, document, and put everything back the way you found it.