

Picking and Configuring What You Scan

Wed, Aug 12, 4:00 PM EDT · <https://scottslab.io/posts/vulnerability-scan-targets-and-configuration>

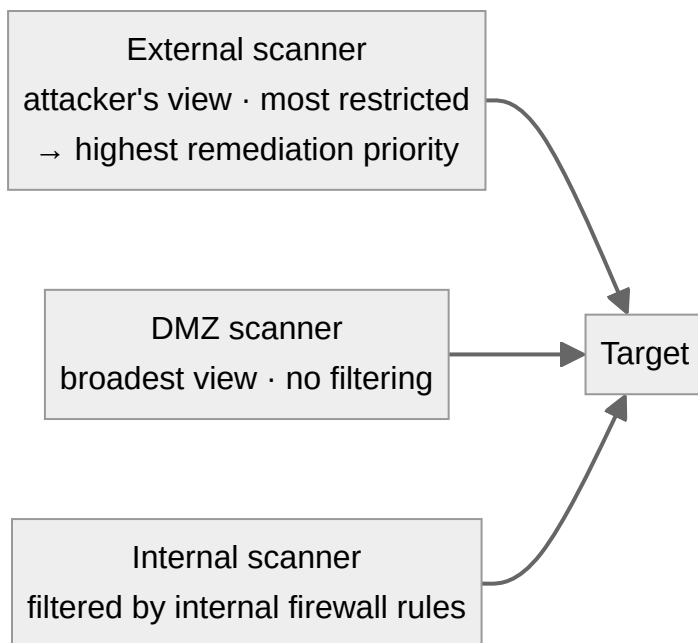


TL;DR

You can't scan what you haven't inventoried, so a program starts with an asset list (from a CMDB, config management, or a discovery scan) and prioritizes it by impact, likelihood, and criticality. Where you put the scanner changes what you see: a DMZ scanner sees everything, an external scanner sees the attacker's view (most restricted, highest remediation priority), an internal one sees through your firewall rules. And prefer credentialed scans (a read-only login) over agents or admin creds: depth without the extra baggage.

Every scan-config decision traces back to two questions: what am I scanning, and from where.

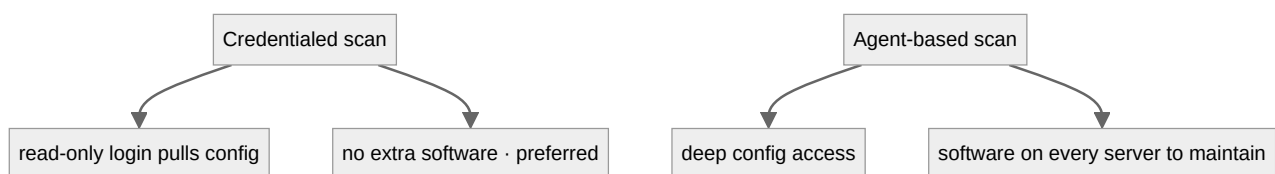
The "what" has a hard prerequisite people skip: an asset inventory. You cannot manage vulnerabilities on assets you don't know exist, so you pull the list from a CMDB, from config-management tooling, or, if you have neither, you run a lightweight discovery scan first just to build one. Then you prioritize, because scanning everything with equal weight wastes the effort. Three axes do it: impact (how sensitive is the data it stores, processes, or transmits), likelihood (how exposed is it: internet-facing or tucked behind a firewall, and which services are open), and criticality (how badly does it hurt if the system goes down). A public-facing box holding regulated data ranks above an internal wiki, and your scan schedule should say so.



Picking and Configuring What You Scan

The "from where" is scan perspective, and it's not a detail. The same target scanned from three places gives three different answers. A scanner in the DMZ has the broadest view, unfiltered by internal firewalls. An internal scanner sees what the internal firewall rules allow, which is the realistic picture for lateral-movement worries. And an external, internet-based scanner sees exactly what an outside attacker sees: the most restricted view, but the highest-priority findings, because those are exposed to the entire world. You want more than one perspective, because "invisible from the internet" and "invisible from the inside" are very different safety claims.

Then the mechanics. Combine scan types: network vulnerability scans for the infrastructure, application scans, and specialized web-app scans that actually probe for SQL injection and XSS. And choose credentialed over agent-based when you can: an agent gives deep config access but means running more software on every server, while a credentialed scan just logs in with a read-only account to read configuration. Read-only, not admin. Your scanner shouldn't be a domain-admin-shaped hole in the network waiting to be stolen.



Picking and Configuring What You Scan

A few Nessus specifics, because they're where scans quietly go wrong. Point it at IPs, hostnames, or an imported target file from your asset tool. Use the schedule tab to set frequency per system group, and scan the sensitive tiers more often. Leave assessment accuracy on normal, then decide whether potential false alarms show or hide. In production, enable safe checks and rate-limit so the scan

doesn't knock things over. Disable plugin families you don't run (no point scanning for Amazon Linux bugs across a Windows fleet) to speed things up. And remember that an active IPS on the scan path will distort the results: it blocks the scanner and hands you a rosier picture than reality.

Written by Scott S. — Contact: scott@scottschlangen.com — scottslab.io — <https://scottslab.io/posts/vulnerability-scan-targets-and-configuration>