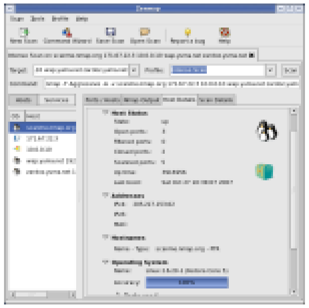
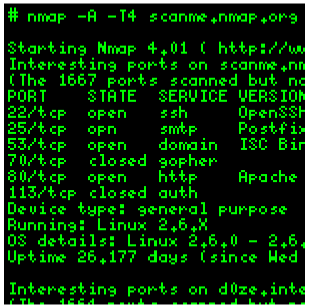


NMAP Free Security Scanner
Audit Your Network Now!
www.insecure.org/NMAP



```
# nmap -A -T4 scanme.nmap.org

Starting Nmap 4.01 ( http://ww
Interesting ports on scanme.nm
(The 1667 ports scanned but no
PORT    STATE  SERVICE VERSION
22/tcp  open   ssh     OpenSSH
25/tcp  opn    smtp    Postfix
53/tcp  open   domain  ISC Bin
70/tcp  closed gopher
80/tcp  open   http    Apache
113/tcp closed auth
Device type: general purpose
Running: Linux 2.6.X
OS details: Linux 2.6.0 - 2.6.
Uptime 26.177 days (since Wed

Interesting ports on d0ze.inte
```

Block and Slow Nmap with Firewalls

One of the best defensive measures against scanning is a well-configured firewall. Rather than simply obfuscate the network configuration, as some techniques described later do, well-configured firewalls can effectively block many avenues of attack.

Any decent firewall book emphasizes this cardinal rule: deny by default. Rather than trying to block suspected malicious traffic, block everything first, then specifically override that to allow essential traffic. It is much easier to overlook blocking something malicious than to accidentally explicitly allow the same. Additionally, failing to block bad traffic may not be noticed until it is exploited by an attacker, while failing to allow legitimate traffic is usually quickly discovered by the affected users. And they will keep reminding you until it is fixed.

Sponsors:

The two preceding reasons should be enough to convince anyone to go with deny-by-default, but there are other benefits as well. One is to slow down large scale reconnaissance from tools like Nmap. When an Nmap TCP SYN scan encounters a closed port, the target machine sends back a RST packet and that port's status is determined within the space of only one round-trip-time. That is under a quarter of a second, even across the world from my web server in California to an ISP in Moscow. If a firewall filters the port by dropping the probe, on the other hand, Nmap has to wait for a worst-case timeout before giving up. Nmap then makes several retransmissions just in case the packet was dropped by some router due to overcapacity rather than by a firewall rule. In large-scale scans, the difference can be quite significant. For example, a 1,000-port TCP SYN scan against a machine on my wireless network (**nmap -sS -T4 para**) takes only five seconds when all ports are open or closed. Filtering a dozen or so commonly exploited ports increases the scan time to 12 seconds. Moving to default-deny (filtering all ports except the five open ones) nearly triples the scan time to 33 seconds. A 28-second difference may not sound meaningful, but it can add up to extra days for large-scale scans.

Filtered ports are even more frustrating to attackers when the UDP protocol is used. When firewalling is not involved, virtually all systems respond with an ICMP port unreachable when Nmap probes a closed port. Open ports usually do not respond at all. So if a deny-by-default firewall drops a probe packet, Nmap cannot tell if the port is open or filtered. Retransmissions do not help here, as the port will never respond. Attackers must then resort to slower and much more conspicuous techniques such as Nmap version detection and SNMP community string brute forcing to make sense of the UDP ports.

To actually slow Nmap down, make sure the firewall is dropping the packets rather than responding with an ICMP error or TCP RST. Otherwise Nmap will run just as fast and accurately as if the ports were closed, though you still reap the benefit of blocking the probes. As an example of this distinction, the Linux iptables firewall offers the target actions DROP and REJECT. As the names imply, DROP does nothing beyond blocking the packet, while REJECT sends an error message back. The former is better for slowing down reconnaissance and is usually recommended, though REJECT can ease network trouble diagnosis by making it crystal clear that the firewall is blocking certain traffic.

Another tenet of firewalls is *defense in depth*. Even though ports are blocked by the firewall, make sure they are closed (no application is listening) anyway. Assume that a determined attacker will eventually breach the firewall. Even if they get through using a technique from Chapter 10, *Detecting and Subverting Firewalls and Intrusion Detection Systems*, the individual machines should be locked down to present a strong defense. This reduces the scope and damage of mistakes, which everyone makes on occasion. Attackers will need to find weaknesses in both the firewall and individual machines. A port scanner is pretty impotent against ports that are both closed and filtered. Using private address space (such as with network address translation) and additional firewalls provide even more protection.

Scan Proactively, Then Close or Block Ports and

Detect Nmap Scans

Fix Vulnerabilities

[Nmap | Sec Tools | Mailing Lists | Site News | About/Contact | Advertising | Privacy]

Custom Search