

# Open Source Security Tools for Information Technology Professionals

School of Professional Studies (SPS)

The City University of New York (CUNY)

Aron Trauring  
Adjunct Professor  
CEO, Zoteca

Class 7 November 7nd, 2005

# Class Agenda

**6:30-6:45** Article of the Week

**6:45-7:15** GUI IPTables and Smoothwall

**7:15-8:00** Port Scanning

**8:00-8:15** Break

**8:15-9:30** Nmap

## **Instructor**

Aron Trauring – Adjunct Professor, CUNY SPS / CEO, Zoteca

**Email:** [atrauring@zoteca.com](mailto:atrauring@zoteca.com)

**Personal Website:** <http://aronst.org/>

**Zoteca Corporate Website:** <http://www.zoteca.com/>

## GUI IPTables and Smoothwall

### Install and Configure Webmin

- webmin-firewall
- “Block all incoming traffic on external interface”
- “Enable firewall at boot time”

### Install and Configure Shorewall

- shorewall, webmin-shorewall
- Debian does not install default configuration files
- These may be found in `/usr/share/doc/shorewall/default-config/`

### Install and Configure Smoothwall

- Standalone security appliance
- Red and Green Zone

## Port Scanning

### TCP/UDP Ports

- 16 bits — 1-65,535
- 0-1,023 — reserved for common applications
- Also known as **well-known** ports
- Assigned by IANA — Internet Assigned Numbers Authority
- Originally run by one man, Jon Postel
- Assigned Well Known Ports <http://www.iana.org/assignments/port-numbers>

### Ephemeral (Dynamic, Private) Ports

- Every connection has two ends
- Server side is well-known port
- Client side is ephemeral assigned by IP stack
- Port taken for every connection
- Usually high end — 49152 through 65535 most common
- 32768-61000 — Linux
- 1024-4000 — Windows (problematic since it restricts number of connections)

### Exploiting Open Ports

- Illegitimate traffic can come in on legitimate open ports
- Port 80 (web) is particularly vulnerable
- Buffer overflow — overflow allocated data storage buffer and change the data that follows buffer in memory

### Buffer Overflow — Security Problem

- C subroutine pushes the return address onto the stack, so the subroutine knows where to return control to when it has finished
- When a dynamic buffer is allocated, the stack grows left by however big the buffer is.
- At the end of the function, the buffers are deallocated, everything pushed is popped, and a RET operation is called.
- This pops the return address off the stack and jumps there, shifting program control back to wherever the subroutine was called from.
- Overflow data will clobber the return address, overwriting part of it with the extra data.
- This changes where program control will go to continue execution when the subroutine has finished

- If random place in memory program “crashes” when the RET instruction attempts to jump control
- A technically inclined malicious user can ensure that the extra data points to a valid memory address, causing program control to be shifted to location of their choosing\
- There it can execute whatever arbitrary code the user has caused to be in that location with whatever privileges the currently executing program has

### **Buffer Overflow — History**

- Morris Worm (1988) — first Internet worm (exploited fingerd)
- Phrack (1996) — "Smashing the Stack for Fun and Profit"
- Coder Red Work (2001) – exploited IIS (which kept administrative privileges after binding port 80)
- SQLSlammer (2003)

### **Buffer Overflow — Programming Solutions**

- Choice of language — C and C++ do no memory checking; Java, Python, C# do
- Choice of safe libraries — particularly string handling routines
- Executable Space protection (kernel patches and processors)
- Stack Smashing protection (compilers)
- **Apply security patches!**

### **Buffer Overflow — Second-best Solutions**

- Close unnecessary ports
- Reduce the number of services running to only what is needed

### **Port Scanners**

- Poll TCP and UDP ports to see if an application answers back
- Simplest: telnet <IP address> <portno>

```
telnet microsoft.com 80
a<enter>
```

- Port scanner tools scan multiple ports, even ephemeral ports

### **Why Do Full Scan**

- Not all vendors stick to standards
- Malware may be hiding in high ports

## TCP Fingerprinting

- Each vendor implements TCP/IP stack for its own OS
- Response to queries might differ based on OS and version
- Useful information for attackers
- Not always accurate (borrowed code, embedded OSes)

## Port Scanning Considerations

- Network intensive so may degrade performance or shut computers down
- Legality not clear (is this considered an attempt to break in?)
- May be banned by ISP terms of service
- Don't do without permission
- iptables flag rules may block
- New iptables patch:

```
iptables -A INPUT -m psd -j DROP
```

## Uses for Port Scanning

- Network inventory — check for number of machines and IP addresses in use
- Network/server optimization — discover services that are unnecessary
- Discovering malware
- Discovering unauthorized and illicit services

## Malware

- Spyware — track user activities and report back to a central server
- Trojan Horse — back door for cracker, needs open port to communicate back (usually high port)
- Network Worms — open up ports as they spread from computer to computer

## Unauthorized Services

- P2P
- IM
- Employee installed

# Nmap

## Install

- nmap
- nmapfe
- ntop

## Nmap Command Line

- `nmap [options] [ip-range]`
- Single IP address — `192.168.1.1`
- List of IP addresses — `192.168.1.1,192.168.1.2`
- Range — `192.168.1.1-192.168.1.10`
- CIDR Notation — `192.168.1.0/24`

## Nmap Discovery Stage

- First find out which hosts in IP range are active
- Use different type of probes to avoid firewall restrictions and different types of devices
- Goal of probes is to solicit responses which demonstrate that an IP address is actually active
- Default probe: TCP ACK packet for port 80 and an ICMP Echo Request query to each target machine
- On LANs use an ARP scan
- Can increase possibility of discovery by using many types on different ports
- Particularly important in security audits

## -sL (List Scan)

- Lists each host of the network(s) specified, without sending any packets to the target hosts. By default,
- Nmap does reverse-DNS resolution on the hosts to learn their names
- Simple hostnames often give out much useful information give out.
- `fw.chi.playboy.com` is the firewall for the Chicago office of Playboy Enterprises.
- Nmap also reports the total number of IP addresses at the end.
- A good sanity check to ensure that you have proper IP addresses for your targets.
- If the hosts sport domain names you do not recognize, investigate further to prevent scanning the wrong company's network.

**-sP (Ping Scan)**

- Only performs a ping scan (host discovery)
- then print out the available hosts that responded to the scan.
- No further testing (such as port scanning or OS detection) is performed
- This is one step more intrusive than the list scan, and can often be used for the same purposes.
- It allows light reconnaissance of a target network without attracting much attention.
- Knowing how many hosts are up is more valuable to attackers than the list provided by list scan of every single IP and host name
- Used to count available machines on a network or monitor server availability.
- Often called a ping sweep: more reliable than pinging the broadcast address because many hosts do not reply to broadcast queries.
- When strict firewalls are in place between the source host running Nmap and the target network, using advanced techniques is recommended.
- Otherwise hosts could be missed when the firewall drops probes or their responses.

**-P0 (No ping)**

- This option skips the Nmap discovery stage altogether.
- Nmap attempts the requested scanning functions against every target IP address specified.
- Wasteful of network bandwidth but may be only option if up against a strong firewall

**-PS [portlist] (TCP SYN Ping)**

- This option sends an empty TCP packet with the SYN flag set.
- The default destination port is 80
- An alternate port can be specified as a parameter.
- A comma separated list of ports can even be specified (e.g. -PS22,23,25,80,113,1050,35000)
- Probes will be attempted against each port in parallel.

**-PA [portlist] (TCP ACK Ping)**

- Similar to the just-discussed SYN ping. The difference, as you could likely guess, is that the
- TCP ACK flag is set instead of the SYN flag.
- Remote hosts should always respond with a RST packet, disclosing their existence in the process.
- The reason for offering both SYN and ACK is to maximize the chances of bypassing firewalls.
- Netfilter/iptables `--syn` option might block SYN ping probes (-PS)
- ACK probe cuts right through these rules.
- Netfilter/iptables system `--state` option might block ACK probe
- SYN probe is more likely to work against stateful firewall

**-PU [portlist] (UDP Ping)**

- UDP ping, which sends an empty UDP packet to the given ports.
- If no ports are specified, the default is 31338.
- Upon hitting a closed port on the target machine, the UDP probe should elicit an ICMP port unreachable packet in return.
- This signifies to Nmap that the machine is up and available.
- Many other types of ICMP errors, such as host/network unreachable or TTL exceeded or lack of response indicative of a down or unreachable host. .
- Choose port highly unlikely to be in use, since if an open port is reached, most services ignore the empty packet and fail to return any response.
- Primary advantage is that it bypasses firewalls and filters that only screen TCP

**-PE; -PP; -PM (ICMP Ping Types)**

- Standard packets sent by the ubiquitous ping program.
- Many hosts and firewalls now block these packets, rather than responding as required by RFC 1122.
- Hence ICMP-only scans are rarely reliable enough against unknown targets over the Internet.
- For system administrators monitoring an internal network, they can be a practical and efficient approach.
- -PE option sends echo request.
- Timestamp and address mask queries are sent with the -PP and -PM respectively.
- Valuable when admins specifically block echo request packets while forgetting that other ICMP queries can be used for the same purpose.

**-PR (ARP Ping)**

- One of the most common Nmap usage scenarios is to scan an ethernet LAN.
- On most LANs the vast majority of IP addresses are unused at any given time.
- ARP scan puts Nmap and its optimized algorithms in charge of ARP requests.
- If it gets a response back, Nmap doesn't even need to worry about the IP-based ping packets since it already knows the host is up.
- ARP scans much faster and more reliably than IP-based scans.
- Is done by default when scanning ethernet hosts that Nmap detects are on a local ethernet network.

**-n (No DNS resolution)**

- Tells Nmap to never do reverse DNS resolution on the active IP addresses it finds.
- Since DNS is often slow, this speeds things up.

**-R (DNS resolution for all targets)**

- Tells Nmap to always do reverse DNS resolution on the target IP addresses.
- Normally this is only performed when a machine is found to be alive.

**Nmap Scan Types**

- Once discovery is done want to do port scanning
- Nmap provides many different type of scan types

“Experts understand the dozens of scan techniques and choose the appropriate one (or combination) for a given task.

Inexperienced users and script kiddies try to solve every problem with the default SYN scan. “

- Six port states recognized by Nmap

**1. open**

- An application is actively accepting TCP connections or UDP packets on this port.
- Often primary goal of port scanning.
- Each open port is an avenue for attack.
- Open ports also show services available for use on the network.

**2. closed**

- Accessible (it receives and responds to Nmap probe packets), but there is no application listening on it.
- Can be helpful in showing that a host is up on an IP address (host discovery, or ping scanning), and as part of OS detection.
- Because closed ports are reachable, it may be worth scanning later in case some open up.
- Administrators may want to consider blocking such ports with a firewall.

**3. filtered**

- Nmap cannot determine whether the port is open because packet filtering prevents its probes from reaching the port.
- The filtering could be from a dedicated firewall device, router rules, or host-based firewall software.
- These ports frustrate attackers because they provide so little information.
- Sometimes they respond with ICMP error messages such as type 3 code 13 (destination unreachable: communication administratively prohibited)
- Filters that simply drop probes without responding are far more common.
- Forces Nmap to retry several times just in case the probe was dropped due to network congestion rather than filtering.
- Slows down the scan dramatically.

#### 4. unfiltered

- A port is accessible, but Nmap is unable to determine whether it is open or closed.
- Only the ACK scan, which is used to map firewall rulesets, classifies ports into this state.
- Scanning unfiltered ports with other scan types such as Window scan, SYN scan, or FIN scan, may help resolve whether the port is open.

#### 5. open | filtered

- Nmap is unable to determine whether a port is open or filtered.
- This occurs for scan types in which open ports give no response.
- The lack of response could also mean that a packet filter dropped the probe or any response it elicited.
- Hence Nmap does not know for sure whether the port is open or being filtered.

#### 6. closed | filtered

- This state is used when Nmap is unable to determine whether a port is closed or filtered.

#### Caveats

- Hosts may be untrustworthy and send responses intended to confuse or mislead Nmap.
- Much more common are non-rfc-compliant hosts that do not respond as they should to Nmap probes.
- FIN, Null, and Xmas scans are particularly susceptible to this problem.
- Only one method may be used at a time, except that UDP scan (`-sU`) may be combined with any one of the TCP scan types.
- As a memory aid, port scan type options are of the form `-sC`, where C is a prominent character in the scan name, usually the first.
- By default, Nmap performs a SYN Scan

#### `-sS` (TCP SYN scan)

- SYN scan is the default and most popular scan option
- Can be performed quickly, scanning thousands of ports per second on a fast network not hampered by intrusive firewalls.
- SYN scan is relatively unobtrusive and stealthy, since it never completes TCP connections.
- Also works against any compliant TCP stack rather than depending on idiosyncrasies of specific platforms
- Allows clear, reliable differentiation between the open, closed, and filtered states.
- Often referred to as half-open scanning, because you don't open a full TCP connection.

- Send a SYN packet, as if you are going to open a real connection and then wait for a response.
- A SYN/ACK indicates the port is listening (open), while a RST (reset) is indicative of a non-listener.
- If no response is received after several retransmissions, the port is marked as filtered.
- The port is also marked filtered if an ICMP unreachable error (type 3, code 1,2, 3, 9, 10, or 13) is received.

#### **-sU (UDP scans)**

- UDP scanning is generally slower and more difficult than TCP so some security auditors ignore these ports.
- Exploitable UDP services are quite common and attackers certainly don't ignore the whole protocol.
- Can be combined with a TCP scan type such as SYN scan (-sS) to check both protocols during the same run.
- UDP scan works by sending an empty (no data) UDP header to every targeted port.
- If an ICMP port unreachable error (type 3, code 3) is returned, the port is closed.
- Other ICMP unreachable errors (type 3, codes 1, 2, 9, 10, or 13) mark the port as filtered.
- Occasionally a service will respond with a UDP packet, proving that it is open.
- If no response is received after retransmissions, the port is classified as open | filtered.

#### **-sN; -sF; -sX (TCP Null, FIN, and Xmas scans)**

- Exploit a subtle loophole in the TCP RFC 793 to differentiate between open and closed ports.
- Any packet not containing SYN, RST, or ACK bits will result in a returned RST if the port is closed and no response at all if the port is open.
- As long as none of those three bits are included, any combination of the other three (FIN, PSH, and URG) are OK.
- Null scan: Does not set any bits (tcp flag header is 0)
- FIN scan: Sets just the TCP FIN bit.
- Xmas scan: Sets the FIN, PSH, and URG flags, lighting the packet up like a Christmas tree.
- If a RST packet is received, the port is considered closed, while no response means it is open | filtered.
- The port is marked filtered if an ICMP unreachable error (type 3, code 1, 2, 3, 9, 10, or 13) is received.
- The key advantage to these scan types is that they can sneak through certain non-stateful firewalls and packet filtering routers.
- Another advantage is that these scan types are a little more stealthy than even a SYN scan.
- Not all systems follow RFC 793 to the letter.
- Another downside of these scans is that they can't distinguish open ports from certain filtered ones, leaving you with the response open | filtered.

**-sA (TCP ACK scan)**

- Never determines open (or even open | filtered) ports.
- It is used to map out firewall rulesets, determining whether they are stateful or not and which ports are filtered.
- The ACK scan probe packet has only the ACK flag set.
- When scanning unfiltered systems, open and closed ports will both return a RST packet.
- Nmap then labels them as unfiltered, meaning that they are reachable by the ACK packet
- Ports that don't respond, or send certain ICMP error messages back (type 3, code 1, 2, 3, 9, 10, or 13), are labeled filtered.

**-sW (TCP Window scan)**

- Exactly the same as ACK scan except that it exploits an implementation detail of certain systems to differentiate open ports from closed ones
- Examines the TCP Window field of the RST packets returned.
- On some systems, open ports use a positive window size (even for RST packets) while closed ones have a zero window.
- Relies on an implementation detail of a minority of systems out on the Internet, so you can't always trust it.

**-sO (IP protocol scans)**

- Allows you to determine which IP protocols (TCP, ICMP, IGMP, etc.) are supported by target machines.
- This isn't technically a port scan, since it cycles through IP protocol numbers rather than TCP or UDP port numbers.

**Advanced Scan Types**

- `--scanflags` — Allows you to design your own scan by specifying arbitrary TCP flags.
- `-sI (Idle)` — uses zombies

**Port Specification**

- By default, Nmap scans all ports up to and including 1024 as well as higher numbered ports listed in the `nmap-services` file for the protocol(s) being scanned.
- `-p <port ranges>` — Only scan specified ports
- `-p-` — Scan all ports
- You can specify a particular protocol by preceding the port numbers by T: or U:
- `-F` — you only wish to scan for ports listed in the `nmap-services` file which comes with nmap

**-sv (Version detection)**

- When doing vulnerability assessments (or even simple network inventories) you really want to know which mail and DNS servers and versions are running.
- Having an accurate version number helps dramatically in determining which exploits a server is vulnerable to.
- Version detection helps you obtain this information.
- After TCP and/or UDP ports are discovered version detection interrogates those ports to determine more about what is actually running.
- The nmap-service-probes database contains probes for querying various services and match expressions to recognize and parse responses.
- Nmap tries to determine the service protocol, application name, the version number, hostname, device type, the OS family and other miscellaneous details
- Most services don't provide all of this information.
- When Nmap cannot match responses to its database, it prints out a special fingerprint and a URL for you to submit to, if you know for sure what is running on the port.

**-sR (RPC grinder)**

- Used to determine the RPC program and version numbers.

**-o (Enable OS detection)**

- OS detection using TCP/IP stack fingerprinting.
- Nmap sends a series of TCP and UDP packets to the remote host and examines practically every bit in the responses.
- After performing dozens of tests Nmap compares the results to its nmap-os-fingerprints database of more than 1500 known OS fingerprints
- Prints out the OS details if there is a match.
- Includes a freeform textual description of the OS, and a classification: the vendor name, underlying OS, OS generation and device type
- If Nmap is unable to guess the OS of a machine, and conditions are good Nmap will provide a URL you can use to submit the fingerprint if you know (for sure) the OS running on the machine.
- OS detection enables several other tests which make use of information that is gathered during the process anyway.
- One of these is uptime measurement, which uses the TCP timestamp option (RFC 1323) to guess when a machine was last rebooted.
- Another is TCP Sequence Predictability Classification. This measures approximately how hard it is to establish a forged TCP connection against the remote host.

## Timing and Performance

- Performance one of Fyodor's highest development priorities
- Nmap utilizes parallelism and many advanced algorithms to accelerate scans
- User has ultimate control over how Nmap runs.
- Expert users carefully craft Nmap commands to obtain only the information they care about while meeting their time constraints.
- Omit non-critical tests
- Upgrade to the latest version of Nmap
- Optimize timing parameters

## Firewall/IDS Evasion and Spoofing

“Many Internet pioneers envisioned a global open network with a universal IP address space allowing virtual connections between any two nodes. This allows hosts to act as true peers, serving and retrieving information from each other. People could access all of their home systems from work, changing the climate control settings or unlocking the doors for early guests. This vision of universal connectivity has been stifled by address space shortages and security concerns. In the early 1990s, organizations began deploying firewalls for the express purpose of reducing connectivity. Huge networks were cordoned off from the unfiltered Internet by application proxies, network address translation, and packet filters. The unrestricted flow of information gave way to tight regulation of approved communication channels and the content that passes over them.”

- Nmap offers many features to help understand complex networks, and to verify that filters are working as intended.
- Supports mechanisms for bypassing poorly implemented defenses.
- One of the best methods of understanding your network security posture is to try to defeat it.
- Companies are increasingly monitoring traffic with intrusion detection systems (IDS).
- All of the major IDSs ship with rules designed to detect Nmap scans because scans are sometimes a precursor to attacks.
- Many of these products have morphed into intrusion prevention systems (IPS) that actively block traffic deemed malicious.
- Attackers with patience, skill, and the help of certain Nmap options can usually pass by IDSs undetected.

“Occasionally people suggest that Nmap should not offer features for evading firewall rules or sneaking past IDSs. They argue that these features are just as likely to be misused by attackers as used by administrators to enhance security. The problem with this logic is that these methods would still be used by attackers, who would just find other tools or patch the functionality into Nmap. Meanwhile, administrators would find it that much harder to do their jobs. Deploying only modern, patched FTP servers is a far more powerful defense than trying to prevent the distribution of tools implementing the FTP bounce attack.”

## Output

- Nmap offers several output formats
- Nmap provides options for controlling the verbosity of output as well as debugging messages.
- Output types may be sent to standard output or to named files, which Nmap can append to or clobber.
- Output files may also be used to resume aborted scans.

### Five Output Formats.

1. The default is called interactive output, and it is sent to standard output (stdout).
2. `-oN <filespec>` — Normal output, displays less runtime information and warnings since it is expected to be analyzed after the scan completes rather than interactively.
3. `-oX <filespec>` — XML output: can be converted to HTML, easily parsed by programs such as Nmap graphical user interfaces, or imported into databases.
4. `-oG <filespec>` — simple grepable output which includes most information for a target host on a single line (deprecated)
5. `-oS <filespec>` — sCRiPt KiDDi3 OutPUt (humor)

## Nlog

- Perl scripts to analyze log files
- Extensible

### Uses of Nmap

- Search for uncommon services
- Search for hidden web servers in embedded devices and workstations
- Search for unnecessary services on workstations
- Search for trojan horses (in high ports)
- Security audit from the outside