NMAP

SCANNING TECHNIQUES

Barbagallo Valerio
Da Lozzo Giordano
Mellini Giampiero

scanning techniques	nmap usage	TCP	UDP	Anonymity level / IDS evasion	Firewall evasion	root privileges
HOST DISCOVERY						
Ping scan	-sP			low	low	
List scan	-sL			high	-	
TCP Syn ping	-PS [portlist]	X		medium	medium	
TCP Ack ping	-PA [portlist]	X		medium	medium	
IP protocol ping	-PO [protocol list]	X		medium	medium	X
UDP ping	-PU [portlist]		X	medium	medium	X
ARP ping	-PR			high	high	
Skip host discovery	-PN					
PORT SCANNING						
TCP syn scan	-sS	X		medium	low	X
TCP connect scan	-sT	X		low	low	
UDP scan	-sU		X	medium	medium	X
TCP Null, FIN, Xmas scan	-sN; -sF; -sX	X		medium	medium	X
TCP ack scan	-sA	X		medium	medium	X
TCP Window scan	-sW	X		medium	medium	X
TCP Maimon scan	-sM	X		medium	medium	X
Custom TCP scan	--scanflags <TCP flags>	X		-	-	-
Idlescan	-sI <zombie-host[:probe-port]>	X		high	medium	X
IP protocol scan	-sO			medium	medium	X
FTP bounce scan	-b < username:password@server:port >	X		high	medium	
INTERESTING OPTIONS						
versioning	-sV			low	-	
OS detection	-O			low	-	X
timing	-T <0-5>			low-high	-	
fragment packets	-f			low	medium	X
decoy scan	-D <decoy1[,decoy2][,ME]>			high	-	X
port range	-p <from-to>			-	-	
spoof source address	-S <ip address>			high	high	X
spoof MAC address	--spoof-mac <mac addr or vendor name>			high	medium	X

PORT SCANNING TECHNIQUES

1) TCP syn scan

attacker	→	SYN	→	target
attacker	←	SYN+ACK	←	target
attacker	→	RST	→	target
open port				

attacker	→	SYN	→	target
attacker	←	RST+ACK	←	target
closed port				

- The default scansion when you have root privileges
- Fast and efficient

error0@pinguino:~$ sudo nmap -sS 192.168.44.170
Starting Nmap 4.50 (http://insecure.org) at 2008-01-08 10:28 CET
Interesting ports on 192.168.44.170:
Not shown: 1708 closed ports
PORT STATE SERVICE
902/tcp open iss-realsecure-sensor
5432/tcp open postgres
8009/tcp open ajp13
*Nmap done: 1 IP address (1 host up) scanned in **0.218 seconds***

error0@pinguino:~$ sudo nmap -sS -sV 192.168.44.170
Starting Nmap 4.50 (http://insecure.org) at 2008-01-08 10:29 CET
Interesting ports on 192.168.44.170:
Not shown: 1708 closed ports
PORT STATE SERVICE VERSION
902/tcp open ssl/vmware-auth VMware GSX Authentication Daemon 1.10 (Uses VNC)
5432**/tcp open http **Apache Tomcat/Coyote JSP engine 1.1
8009/tcp open ajp13?
Service detection performed. Please report any incorrect results at http://insecure.org/nmap/submit/ .
*Nmap done: 1 IP address (1 host up) scanned in **41.234 seconds***

- In this example you can notice how much the option –sV can be useful to understand the service on the target machine
- But –sV take much more time than a simple TCP syn scan

2) TCP connect scan

attacker	→	SYN	→	target
attacker	←	SYN+ACK	←	target
attacker	→	ACK	→	target
attacker	→	RST+ACK	→	target
open port				

attacker	→	SYN	→	target
attacker	←	RST+ACK	←	target
closed port				

- This is the default scansion when you don’t have the root privileges
- Compared to the TCP syn scan, this produces the same result, but in a bit more time

3) UDP scan

attacker	→	UDP	→	target
open | filtered port				

attacker	→	UDP	→	target
attacker	←	ICMP port unrecheable	←	target
closed port				

- The major problem of this scansion is the long time that it requires

error0@pinguino:~$ sudo nmap -sV -sU 79.9.230.67
Starting Nmap 4.50 (http://insecure.org) at 2008-01-07 20:29 CET
Interesting ports on host67-230-dynamic.9-79-r.retail.telecomitalia.it (79.9.230.67):
Not shown: 1485 closed ports
PORT STATE SERVICE VERSION
67/udp open|filtered dhcps
1900/udp open|filtered UPnP
4672/udp open|filtered rfa
Service detection performed. Please report any incorrect results at http://insecure.org/nmap/submit/ .
*Nmap done: 1 IP address (1 host up) scanned in **1550.092 seconds***

- In this example is very interesting to notice how Nmap make the DNS reverse name resolution

4) TCP Null, FIN, and Xmas scans

- These scansions are very similar. They can bypass some non-stateful firewall
- In this example we discuss about the Fin scan

attacker	→	FIN	→	target
open | filtered port				

attacker	→	FIN	→	target
attacker	←	RST+ACK	←	target
closed port				

root@pinguino:/home/error0# iptables -A INPUT -m state --state
ESTABLISHED,RELATED,INVALID -j ACCEPT
root@pinguino:/home/error0# iptables -L
Chain INPUT (policy DROP)
target prot opt source destination
ACCEPT 0 -- anywhere anywhere state
INVALID,RELATED,ESTABLISHED
Chain FORWARD (policy ACCEPT)
target prot opt source destination
Chain OUTPUT (policy ACCEPT)
target prot opt source destination

- Here is our configuration of Netfilter, the linux firewall

error0@pinguino:~$ sudo nmap -sS -sV -p 5431-5435 172.16.36.1
Starting Nmap 4.50 (http://insecure.org) at 2008-01-08 17:03 CET
Interesting ports on 172.16.36.1:
PORT STATE SERVICE VERSION
5431/tcp filtered unknown
5432/tcp filtered postgres

5433/tcp filtered unknown
5434/tcp filtered unknown
5435/tcp filtered unknown
Service detection performed. Please report any incorrect results at http://insecure.org/nmap/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 3.274 seconds

error0@pinguino:~$ sudo nmap -sF -sV -p 5431-5435 172.16.36.1
Starting Nmap 4.50 (http://insecure.org) at 2008-01-08 17:04 CET
Interesting ports on 172.16.36.1:
PORT STATE SERVICE VERSION
5431/tcp closed unknown
5432/tcp open|filtered postgres
5433/tcp closed unknown
5434/tcp closed unknown
5435/tcp closed unknown
Service detection performed. Please report any incorrect results at http://insecure.org/nmap/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 6.389 seconds

- This Nmap output show how a normal TCP syn scan detects all the ports as filtered, while the TCP fin scan can bypass the firewall

5) TCP ack scan

attacker	→	ACK	→	target
attacker	←	RST	←	target
open | closed port				

attacker	→	ACK	→	target
filtered port				

- This is used to map out firewall rulesets, determining whether they are stateful or not and which ports are filtered or unfiltered.
- We did this test in the context of the example above, in fact we know that Netfilter is a stateful firewall

error0@pinguino:~$ sudo nmap -sA -sV -p 5431-5435 172.16.36.1
Starting Nmap 4.50 (http://insecure.org) at 2008-01-08 17:03 CET
Interesting ports on 172.16.36.1:
PORT STATE SERVICE VERSION
5431/tcp filtered unknown
5432/tcp filtered postgres
5433/tcp filtered unknown
5434/tcp filtered unknown
5435/tcp filtered unknown
Service detection performed. Please report any incorrect results at http://insecure.org/nmap/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 3.201 seconds

6) TCP window scan

- This is exactly the same as ACK scan except that it exploits an implementation detail of certain systems to differentiate open ports from closed ones, rather than always printing unfiltered when a RST is returned, by examining the TCP window field

7) Custom TCP scan

- With this type of scansion you can repeat any of the scansion that Nmap already knows
- You can specify a TCP scan type (such as -sA or -sF). That base type tells Nmap how to interpret responses

```
error0@pinguino:~$ sudo nmap -sF  192.168.0.4
Starting Nmap 4.50 ( http://insecure.org ) at 2008-01-09 18:52 CET
Interesting ports on 192.168.0.4:
Not shown: 1710 closed ports
PORT    STATE         SERVICE
902/tcp open|filtered iss-realsecure-sensor
Nmap done: 1 IP address (1 host up) scanned in 1.559 seconds
error0@pinguino:~$ sudo nmap --scanflags FIN  192.168.0.4
Starting Nmap 4.50 ( http://insecure.org ) at 2008-01-09 18:52 CET
Interesting ports on 192.168.0.4:
Not shown: 1710 closed ports
PORT    STATE    SERVICE
902/tcp filtered iss-realsecure-sensor
Nmap done: 1 IP address (1 host up) scanned in 1.520 seconds
```

- Let your creative juices flow, while evading intrusion detection systems whose vendors simply paged through the Nmap man page adding specific rules!

8) Idlescan

- This advanced scan method allows for a truly blind TCP port scan of the target (meaning no packets are sent to the target from your real IP address)

attacker	→	SYN+ACK	→	zombie
attacker	←	RST	←	zombie
IPID identification				

- Step 1: choose a “zombie” and probe for its current IP identification (IPID) number

attacker	→	SYN	→	target
zombie	←	SYN+ACK	←	target
zombie	→	RST	←	target
IPID increment				

attacker	→	SYN	→	target
zombie	←	SYN+ACK	←	target
IPID doesn’t increment				

- Step 2: the attacker send a spoofed packet from the “zombie” to the targegt

attacker	→	SYN+ACK	→	zombie
attacker	←	RST	←	zombie
IPID increate by 2 since step 1: **open port**				

attacker	→	SYN+ACK	→	zombie
attacker	←	RST	←	zombie
IPID only increrased by 1: **closed port**				

- Step 3: probe zombie IPID again

```
error0@pinguino:~$ sudo nmap -P0 -p 134-136 -sI 172.16.36.128 172.16.36.129
Starting Nmap 4.50 ( http://insecure.org ) at 2008-01-10 01:59 CET
Idle scan using zombie 172.16.36.128 (172.16.36.128:80); Class: Incremental
Interesting ports on 172.16.36.129:
PORT    STATE         SERVICE
134/tcp closed|filtered ingres-net
```

135/tcp open msrpc
136/tcp closed|filtered profile
MAC Address: 00:0C:29:B8:C2:3C (VMware)
Nmap done: 1 IP address (1 host up) scanned in 3.324 seconds

- Remember to find an host whose connection is in idle state
- And to use the –P0 option to prevent Nmap from sending the initial ping to the target machine. This slows the scan time, but ensures that no packets are sent to the target from your real IP

9) IP protocol scan

attacker	→	UDP	→	target
attacker	←	ICMP port unreachable	←	target
UDP open				

attacker	→	UDP	→	target
UDP open | filtered				

attacker	→	Echo request	→	target
attacker	←	Echo reply	←	target
ICMP open				

attacker	→	Echo request	→	target
ICMP open | filtered				

- …and so on. There is a different way to interrogate the target for each protocol
- IP protocol scan 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

error0@pinguino:~$ sudo nmap -sO 192.168.0.4
Starting Nmap 4.50 (http://insecure.org) at 2008-01-09 17:26 CET
Interesting protocols on 192.168.0.4:
Not shown: 250 closed protocols
PROTOCOL STATE SERVICE
1 open icmp
2 open|filtered igmp
6 open tcp
17 open udp
136 open|filtered udplite
255 open|filtered unknown
Nmap done: 1 IP address (1 host up) scanned in 2.642 seconds

10) FTP bounce scan

attacker	→	PORT	→	FTP server
attacker	←	200 PORT command ok	←	FTP server
attacker	→	LIST	→	FTP server
target	←	SYN	←	FTP server
target	→	SYN+ACK	→	FTP server
target	←	ACK	←	FTP server
attacker	←	Transfer complete	←	FTP server
open port				

attacker	→	PORT	→	FTP server
attacker	←	200 PORT command ok	←	FTP server
attacker	→	LIST	→	FTP server
target	←	SYN	←	FTP server
target	→	RST	→	FTP server
attacker	←	Connection refused	←	FTP server
closed port				

- The FTP bounce attack uses standard FTP functionality
- The FTP bounce attack is a well positioned TCP port scan through a firewall. **FTP is a commonly available application through a packet-filtering device**
- this attack uses an **FTP server in passive mode**

*nmap -v -b **anonymous:anon@192.168.0.7** 192.168.0.5*
Starting nmap 3.81 (http://www.insecure.org/nmap/) at 2005-04-23 20:37 EDT
*Resolved ftp bounce attack **proxy** to 192.168.0.7 (192.168.0.7).*
Attempting connection to ftp://anonymous:anon@192.168.0.7:21
Connected:Login credentials accepted by ftp server!
Initiating TCP ftp bounce scan against 192.168.0.5 at 20:37
Discovered open port 6969/tcp on 192.168.0.5
Discovered open port 135/tcp on 192.168.0.5
Discovered open port 139/tcp on 192.168.0.5
Discovered open port 445/tcp on 192.168.0.5
Scanned 1663 ports in 9 seconds via the Bounce scan.
Host 192.168.0.5 appears to be up ... good.
Interesting ports on 192.168.0.5:
(The 1659 ports scanned but not shown below are in state: closed)
PORT STATE SERVICE
135/tcp open msrpc
139/tcp open netbios-ssn
445/tcp open microsoft-ds
6969/tcp open acmsoda
MAC Address: 00:11:43:43:A8:34 (Dell (WW Pcba Test))
Nmap finished: 1 IP address (1 host up) scanned in 20.602 seconds
Raw packets sent: 2 (68B) | Rcvd: 1 (46B)

- The FTP bounce attack is interesting, but it's probably not going to work with contemporary FTP servers. If you need to scan through a firewall, you may have better luck with idlescan