

# NMAP

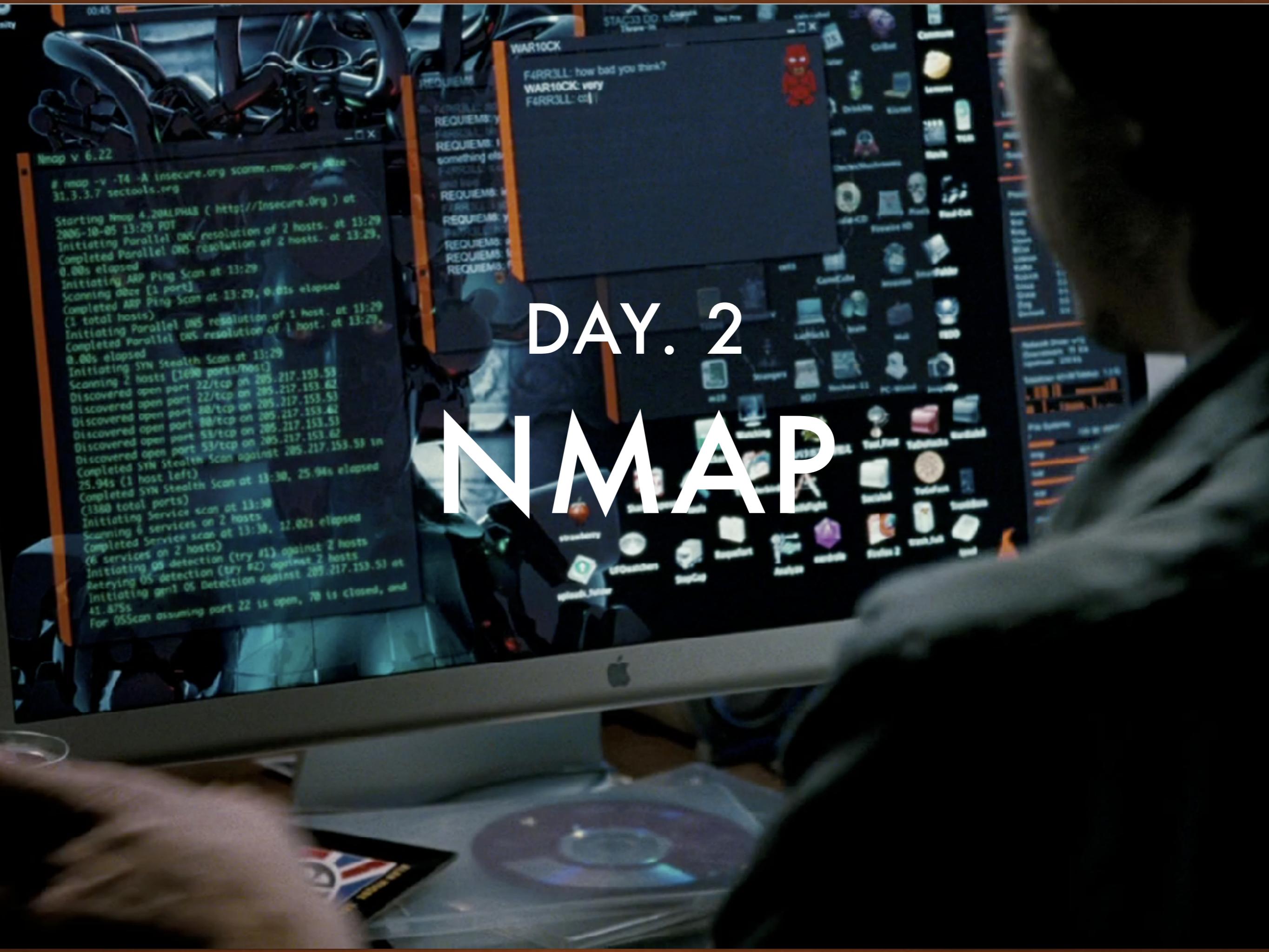
```
* Welcome to CityPower Grid Rerouting *
Authorised Users only!
New users MUST notify Sys/Ops.
login:

[ mobile] rcr ebx, 1
[ mobile] bsr ecx, ec
[ mobile] shrd ebx, e
[ mobile] chrd axx, a
[ mobile] [ mobile]
nmap -v -SS -O 10.2.2.2
Starting nmap 0.2.54BETA25
Insufficient responses for TCP sequencing (3), OS
accurate results on 10.2.2.2:
Ports scanned but not shown below are in
State Service
open ssh
No exact OS matches for host
Nmap run completed -- 1 IP address (1 host up) scann
sshnuke 10.2.2.2 -rootpw...210H0101.. successful.
Connecting to 10.2.2.2:ssh ... successful.
Attempting to exploit SSHv1 CRC32
Resetting root password to "210H0101"
System open: Access Level <9>
SSH 10.2.2.2 -l root
root@10.2.2.2's password: [REDACTED]
```

RTF COMM  
ACCESS GRANT

# DAY. 2

# NMAP



# AGENDA DAY 2

- Advanced Scanning Options
  - Firewall/ IDS Evasion and spoofing
  - Timing and Performance Options
  - IPv6 Scanning
- Output Options
- Debugging and Troubleshooting

# FIREWALL/IDS EVASION AND SPOOFING



# NMAP ADVANCED OPTIONS

## FIREWALL/IDS EVASION

- Firewall, IDS/IPS can make mapping a network exceedingly difficult.
- Nmap offers many features to help understand these complex networks, and to verify that filters are working as intended. It even supports mechanisms for bypassing poorly implemented defenses.
- Launch an FTP bounce scan, idle scan, fragmentation attack, or try to tunnel through one of your own proxies.

# FIREWALL/IDS EVASION

## WINDOWS FIREWALL VS ZONE ALARM

```
root@kali:~# nmap 192.168.2.103
Starting Nmap 7.12 ( https://nmap.org ) at 2016-08-06 00:01 EDT
Stats: 0:00:03 elapsed; 0 hosts completed (0 up), 1 undergoing ARP Ping Scan
ARP Ping Scan Timing: About 100.00% done; ETC: 00:01 (0:00:00 remaining)
Nmap scan report for 192.168.2.103
Host is up (0.036s latency).
Not shown: 993 filtered ports
PORT      STATE SERVICE
135/tcp    open  msrpc
139/tcp    open  netbios-ssn
445/tcp    open  microsoft-ds
554/tcp    open  rtsp
902/tcp    open  iss-realsecure
912/tcp    open  apex-mesh
2869/tcp   open  icslap
3389/tcp   open  ms-wbt-server
5357/tcp   open  wsdapi
10243/tcp  open  unknown
MAC Address: 00:0C:29:83:ED:69 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 40.75 seconds
root@kali:~# nmap 192.168.2.103
Starting Nmap 7.12 ( https://nmap.org ) at 2016-08-06 00:01 EDT
Nmap scan report for 192.168.2.103
Host is up (0.00043s latency).
All 1000 scanned ports on 192.168.2.103 are filtered
MAC Address: 00:0C:29:83:ED:69 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 34.42 seconds
root@kali:~#
```

# PORT SCANNING TECHNIQUES

## TCP NULL, FIN, AND XMAS SCANS

- -sN; -sF; -sX (TCP NULL, FIN, and Xmas scans)
- These three scan types (even more are possible with the --scanflags option described in the next section) exploit a subtle loophole in the TCP RFC to differentiate between open and closed ports.
- When scanning systems compliant with this RFC text, 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.

# PORt SCANNING TECHNIQUES

## TCP ACK SCAN

- -sA (TCP ACK scan)
- This scan is different than the others discussed so far in that it 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 (unless you use --scanflags). 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, but whether they are open or closed is undetermined. Ports that don't respond, or send certain ICMP error messages back (type 3, code 0, 1, 2, 3, 9, 10, or 13), are labeled filtered.

# PORt SCANNING TECHNIQUES

## TCP ACK SCAN

```
bash-3.2# nmap -sA -Pn 192.168.176.226

Starting Nmap 7.00 ( https://nmap.org ) at 2016-07-27 21:41 WIB
Nmap scan report for 192.168.176.226
Host is up (0.00010s latency).
All 1000 scanned ports on 192.168.176.226 are unfiltered
MAC Address: 00:0C:29:BA:E6:DD (VMware)

Nmap done: 1 IP address (1 host up) scanned in 13.24 seconds
bash-3.2#
```

# PORt SCANNING TECHNIQUES

## FTP BOUNCE SCAN

- -b FTP relay host (FTP bounce scan) .
- This scan is different than the others discussed so far in that it 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 (unless you use --scanflags). 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, but whether they are open or closed is undetermined. Ports that don't respond, or send certain ICMP error messages back (type 3, code 0, 1, 2, 3, 9, 10, or 13), are labeled filtered.

# BASIC SCANNING

## PORT SCANNING TECHNIQUES

```
bash-3.2# nmap -Pn -p21 -b anonymous:test@192.168.176.159 127.0.0.1

Starting Nmap 7.00 ( https://nmap.org ) at 2016-07-27 17:27 WIB
Nmap scan report for raiser (127.0.0.1)
Host is up.

PORT      STATE SERVICE
21/tcp    open  ftp

Nmap done: 1 IP address (1 host up) scanned in 9.11 seconds
bash-3.2# nmap -Pn -p22 -b anonymous:test@192.168.176.159 192.168.176.225

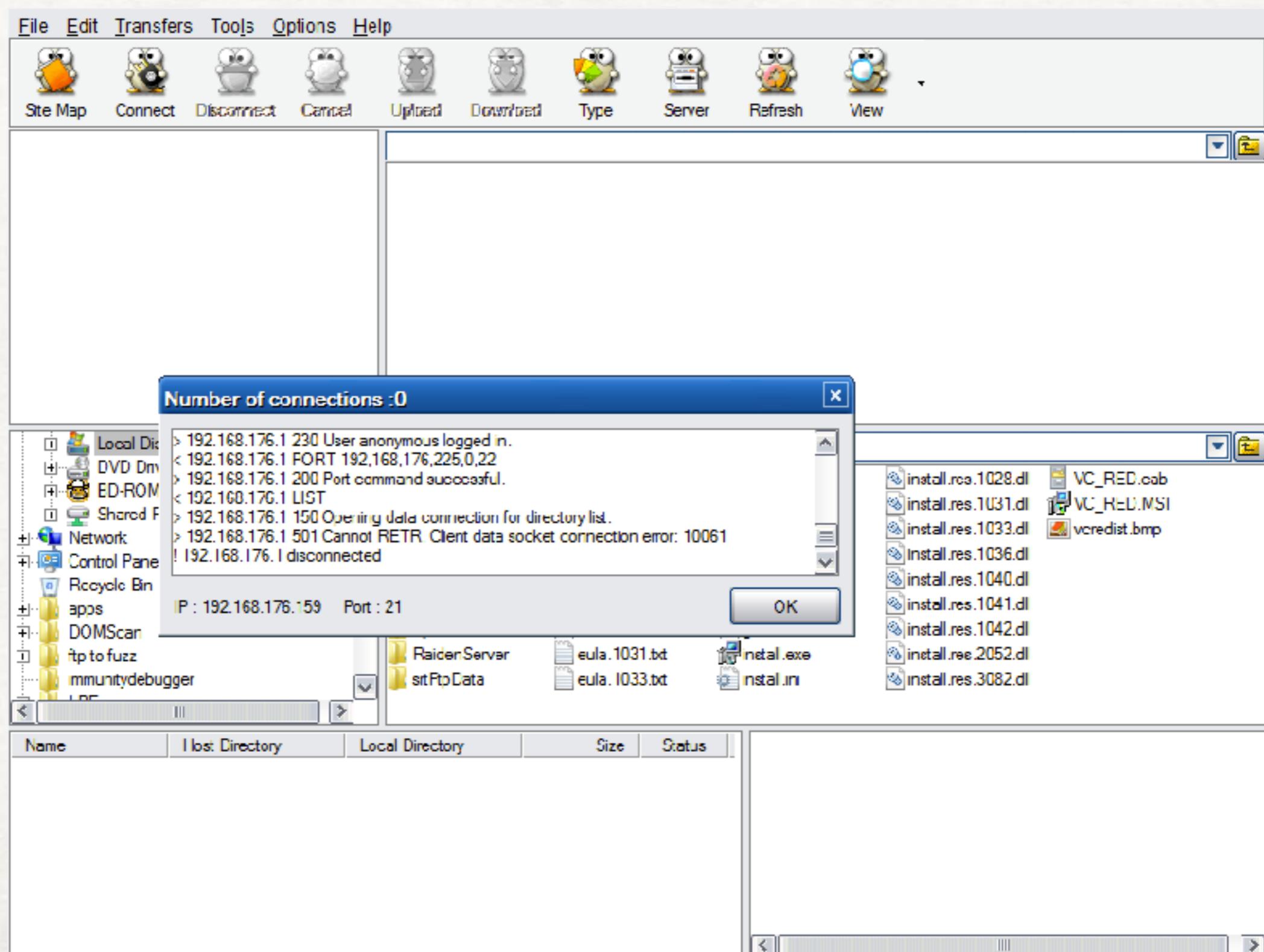
Starting Nmap 7.00 ( https://nmap.org ) at 2016-07-27 17:27 WIB
Nmap scan report for 192.168.176.225
Host is up.

PORT      STATE SERVICE
22/tcp    open  ssh

Nmap done: 1 IP address (1 host up) scanned in 10.13 seconds
bash-3.2# █
```

# BASIC SCANNING

## PORT SCANNING TECHNIQUES



# FIREWALL/IDS EVASION

## FRAGMENT PACKETS

- -f : fragment packets

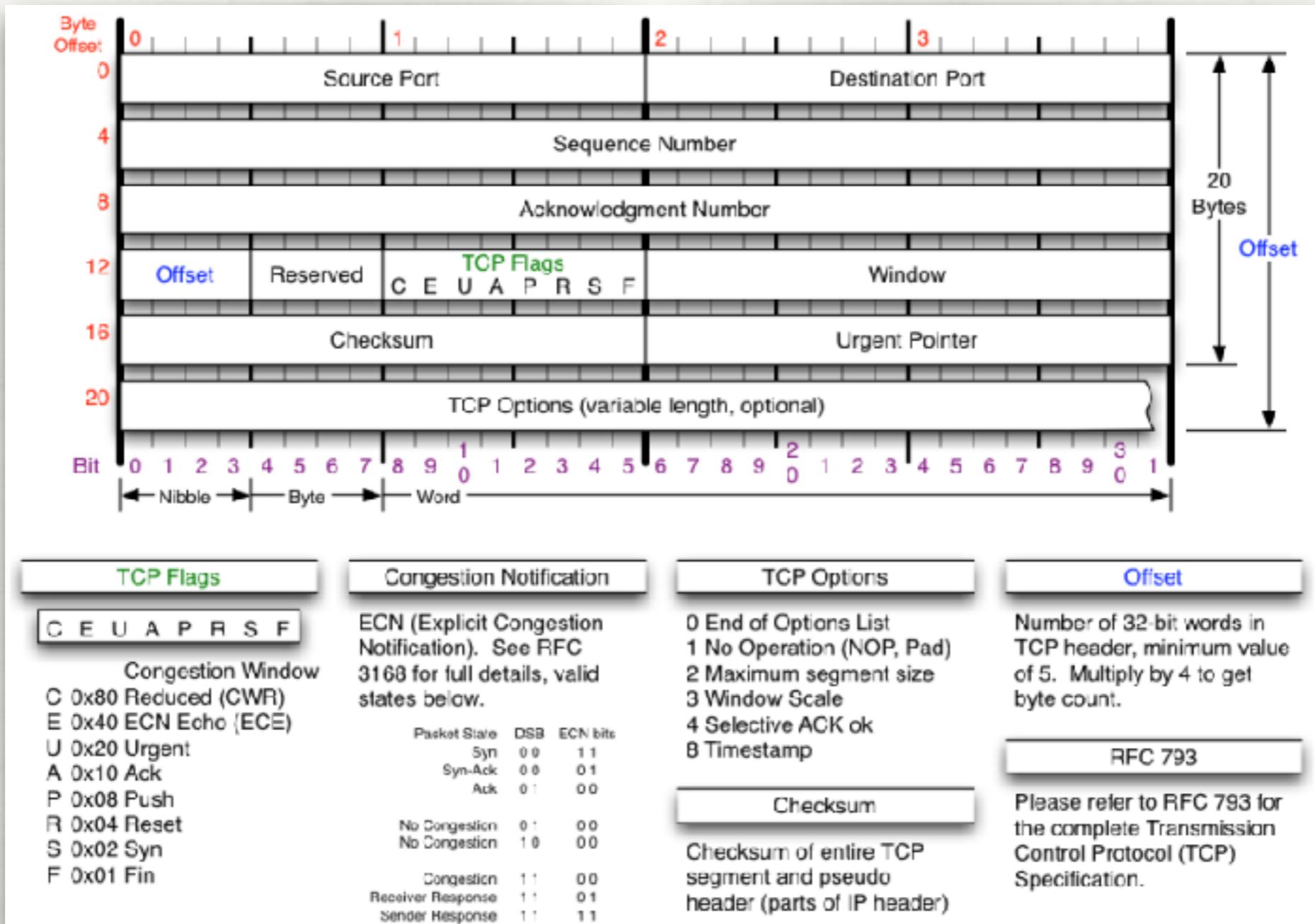
The -f option causes the requested scan (including ping scans) to use tiny fragmented IP packets.

The idea is to split up the TCP header over several packets to make it harder for packet filters, intrusion detection systems, and other annoyances to detect what you are doing.

Specify this option once, and Nmap splits the packets into eight bytes or less after the IP header. So a 20-byte TCP header would be split into three packets. Two with eight bytes of the TCP header, and one with the final four. Of course each fragment also has an IP header.

# FIREWALL/IDS EVASION

## FRAGMENT PACKETS



# FIREWALL/IDS EVASION

## FRAGMENT PACKETS

```
root@kali:~# nmap -PS -p135 192.168.2.103 | Expression... Clear Apply Save
Starting Nmap 7.12 (https://nmap.org) at 2016-08-05 23:26 EDT
Nmap scan report for 192.168.2.103
Host is up (0.00063s latency).
PORT      STATE SERVICE
135/tcp    open  msrpc
MAC Address: 00:0C:29:83:ED:69 (VMware)
Nmap done: 1 IP address (1 host up) scanned in 13.16 seconds
root@kali:~# nmap -f -PS -p135 192.168.2.103
Starting Nmap 7.12 (https://nmap.org) at 2016-08-05 23:27 EDT interface 0
Nmap scan report for 192.168.2.103
Host is up (0.00073s latency).
PORT      STATE SERVICE
135/tcp    open  msrpc
MAC Address: 00:0C:29:83:ED:69 (VMware)
Nmap done: 1 IP address (1 host up) scanned in 13.09 seconds
root@kali:~#
```

# FIREWALL/IDS EVASION

## FRAGMENT PACKETS

- Install Wireshark
- Copy VM

# FIREWALL/IDS EVASION

## WIRESHARK CAPTURE

No.	Time	Source	Destination	Protocol	Length	Info
91	17.69413000	192.168.2.104	192.168.2.103	TCP	58	38416→135 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
94	17.69577400	192.168.2.104	192.168.2.104	TCP	60	135→38416 [SYN, ACK] Seq=0 Ack=1 Win=0 LS2 Len=0 MSS=1460
95	17.69579800	192.168.2.104	192.168.2.103	TCP	54	38416→135 [RST] Seq=1 Win=0 Len=0

Protocol: TCP (6)

- ↳ Header checksum: 0x6930 [validation disabled]
- Source: 192.168.2.104 (192.168.2.104)
- Destination: 192.168.2.103 (192.168.2.103)
- [Source GeoIP: Unknown]
- [Destination GeoIP: Unknown]

↳ Transmission Control Protocol, Src Port: 38416 (38416), Dst Port: 135 (135), Seq: 0, Len: 0

Source Port: 38416 [38416]  
Destination Port: 135 (135)  
[Stream index: 15]  
[TCP Segment Len: 0]  
Sequence number: 0 (relative sequence number)  
Acknowledgment number: 0  
Header Length: 24 bytes

↳ .... 0000 c0c0 0010 = Flags: 0x002 (SYN)  
Window size value: 1024  
[Calculated window size: 1024]  
Checksum: 0xc6a2 [validation disabled]  
Urgent pointer: 0  
Options: (1 bytes), Maximum segment size

0000 20 0c 29 83 ed 59 00 0c 29 6e 88 83 08 00 45 00  
0010 20 2c 92 7c 00 20 39 06 69 80 c0 e8 02 68 c0 a5  
0020 c2 67 96 10 00 87 8e 71 22 5b 03 00 00 00 00 02  
0030 c4 00 c6 a2 00 00 02 04 05 b4 ..)..... }n....E.  
.....9. io...h..  
.g.....q 'l....'  
.....

# FIREWALL/IDS EVASION

## WIRESHARK CAPTURE

No.	Time	Source	Destination	Protocol	Length	Info
9	4.253785000	192.168.2.104	192.168.2.103	IPv4	42	Fragmented IP protocol (proto=TCP 5, off=0, ID=d1ae) [Reassembled in #1]
10	4.254017000	192.168.2.104	192.168.2.103	IPv4	42	Fragmented IP protocol (proto=TCP 5, off=8, ID=d1ae) [Reassembled in #1]
11	4.254215000	192.168.2.104	192.168.2.103	TCP	42	61811->135 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
14	4.255509000	192.168.2.103	192.168.2.104	TCP	60	135->61811 [SYN, ACK] Seq=0 Ack=1 Win=892 Len=0 MSS=1460
15	4.255536000	192.168.2.104	192.168.2.103	TCP	54	61811->135 [RST] Seq=1 Win=0 Len=0

[Destination port: Unknown]  
↳ [3 IPv4 Fragments (24 bytes): #9(8), #10(8), #11(8)]  
\* Transmission Control Protocol, Src Port: 61811 (61811), Dst Port: 135 (135), Seq: 0, Len: 0  
Source Port: 61811 (61811)  
Destination Port: 135 (135)  
[Stream Index: 2]  
[TCP Segment Len: 0]  
Sequence number: 0 (relative sequence number)  
Acknowledgment number: 0  
Header Length: 24 bytes  
↳ .... 0000 0000 0010 = Flags: 0x002 (SYN)  
Window size value: 1024  
[Calculated window size: 1024]  
\* Checksum: 0x0664 [validation disabled]  
Urgent pointer: 0  
↳ Options: [4 bytes], Maximum segment size  
0000 00 0c 29 83 ed 69 00 0c 29 0e 88 83 08 00 45 00 ..)....i... )n....E.  
0010 00 1c d1 ae 00 02 25 05 08 0c c0 ab 02 68 c0 a1 .....%>....h..  
0020 02 67 05 64 06 00 02 04 05 b4 ..q.d.... ...

# FIREWALL/IDS EVASION

## FRAGMENT PACKETS

--mtu (using the specified Maximum Transmission Unit)

Or you can specify your own offset size with the --mtu option.

Don't also specify -f if you use —mtu.

If your host OS is causing problems, try the **--send-eth** option to bypass the IP layer and send raw ethernet frames.

# FIREWALL/IDS EVASION

## FRAGMENT PACKETS

```
root@kali:~# nmap --mtu 16 -PS -p135 192.168.2.103
Starting Nmap 7.12 ( https://nmap.org ) at 2016-08-05 23:48 EDT
Nmap scan report for 192.168.2.103
Host is up (0.00096s latency).
PORT      STATE SERVICE
135/tcp    open  msrpc
MAC Address: 00:0C:29:83:ED:69 (VMware)
Nmap done: 1 IP address (1 host up) scanned in 13.09 seconds
root@kali:~#
```

[redacted]

```
ts)

[redacted]
pe:ip:data]
(00:0c:29:6e:88:83), Dst: VMware_83:ed:69 (00:0c:29:83:ed:69)
```

# FIREWALL/IDS EVASION

## WIRESHARK CAPTURE

No.	Time	Source	Destination	Protocol	Length	Info
47	14.149739000	192.168.2.104	192.168.2.103	IPv4	50	Fragmented IP protocol (p=ctc-TCP 0, off=0, ID=0269) [Reassembled in #48]
48	14.149861000	192.168.2.104	192.168.2.103	TCP	42	4002034-35 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
51	14.151012000	192.168.2.103	192.168.2.104	TCP	60	135>43233 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460
52	14.151030000	192.168.2.104	192.168.2.103	TCP	51	43233>135 [RST] Seq=1 Win=0 Len=0

Header length: 20 bytes  
Differentiated Services field: 0x00 (0x0P 0x00: Default; ION: 0x00: Not-ECN Capable Transport))  
Total Length: 36  
Identifier: 0x0269 (517)  
Flags: 0x01 (More Fragments)  
Fragment offset: 0  
Time to live: 17  
Protocol: TCP (6)  
Header checksum: 0xe6e0 (validation disabled)  
Source: 192.168.2.104 (192.168.2.104)  
Destination: 192.168.2.103 (192.168.2.103)  
[Source GeoIP: Unknown]  
[Destination GeoIP: Unknown]  
Reassembled IPv4 in Frame: 48  
Data (16 bytes)  
Data: 00e100014a601e9f0000000050020400  
Length: 161

# FIREWALL/IDS EVASION

## NMAP

- Fragmentation is only supported for Nmap's raw packet features, which includes TCP and UDP port scans (except connect scan and FTP bounce scan) and OS detection. Features such as version detection and the Nmap Scripting Engine generally don't support fragmentation because they rely on your host's TCP stack to communicate with target services.

# FIREWALL/IDS EVASION

## DECOY

- -D <decoy1>[,<decoy2>][,ME][,...] (Cloak a scan with decoys)

Causes a decoy scan to be performed, which makes it appear to the remote host that the host(s) you specify as decoys are scanning the target network too.

It is generally an effective technique for hiding your IP address.

Thus their IDS might report 5–10 port scans from unique IP addresses, but they won't know which IP was scanning them and which were innocent decoys.

You can also use RND to generate a random, non-reserved IP address, or RND:<number> to generate <number> addresses.

# FIREWALL/IDS EVASION

## DECOY

- Decoys are used in the initial ping scan (using ICMP, SYN, ACK, or whatever), during the actual port scanning phase and also used during remote OS detection (-O).
- Decoys do not work with version detection or TCP connect scan
- Using too many decoys may slow your scan and potentially even make it less accurate. Also, some ISPs will filter out your spoofed packets, but many do not restrict spoofed IP packets at all.

# FIREWALL/IDS EVASION

## DECOY SCANS

```
root@kali:~# nmap -D RND:3 192.168.2.103
Starting Nmap 7.12 ( https://nmap.org ) at 2016-08-06 00:18 EDT
Nmap scan report for 192.168.2.103
Host is up (0.0058s latency).
Not shown: 985 closed ports
PORT      STATE    SERVICE          LENGTH INFO
PORT      STATE    SERVICE          LENGTH INFO
135/tcp   open     msrpc           76      Standard query 0xb3e2 A dns.msftncsi.com
139/tcp   open     netbios-ssn     76      Standard query 0xb3e2 A dns.msftncsi.com
445/tcp   open     microsoft-ds    92      Standard query response 0xb3e2 A 131.107.255.2
554/tcp   open     rtsp            76      Standard query 0x1b3c AAAA dns.msftncsi.com
902/tcp   open     iss-realsecure 92      Standard query response 0xb3e2 A 131.107.255.2
912/tcp   open     apex-mesh       76      Standard query 0xb3e2 A 131.107.255.2
1025/tcp  filtered NFS-or-IIS
1026/tcp  open     LSA-or-nterm
1027/tcp  open     IIS
1028/tcp  open     unknown
1029/tcp  open     ms-lsa
1030/tcp  open     iad1
2869/tcp  open     seicslab
5357/tcp  open     wsdapi
10243/tcp open     unknown
MAC Address: 00:0C:29:83:ED:69 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 17.25 seconds
root@kali:~# [REDACTED]
ip:udp:dns]
```

# FIREWALL/IDS EVASION

## WIRESHARK CAPTURE

The screenshot shows a Wireshark capture window with the following details:

- Title Bar:** Intel(R) PRO/1000 MT Network Connections \Device\NPF\_{FDE253EE-2C3B-4AA1-B1EB-C5A1D51D0292} [Wireshark 1.8.6 (SVN Rev 48142 from /trunk-1.8)]
- Menu Bar:** File Edit View Go Capture Analyze Statistics Telephony Tools Internets Help
- Toolbar:** Standard icons for opening files, saving, zooming, and filtering.
- Filter Bar:** Shows the current filter: `in.1st_host == 192.168.2.103`.
- Table View:** The main pane displays a list of network frames. The columns are: No., Time, Source, Destination, Protocol, Length, Info. The list contains approximately 450 entries, mostly TCP connections between 192.168.2.103 and itself, with various protocols like http, http-alt, ms-wbt-server, ms-wbt-client, and various port mappings (e.g., pptp, epmap, imaps).
- Frame Details:** A detailed view of frame 402 is shown at the bottom left, showing the raw hex and ASCII data.
- Bottom Status Bar:** Shows the file path (File \ Temp\wresh), packet count (Packets: 2451), and profile (Profile: Default).

# FIREWALL/IDS EVASION

## DECOY

This can be defeated through router path tracing, response-dropping, and other active mechanisms.

Note that the hosts you use as decoys should be up or you might accidentally SYN flood your targets. Also it will be pretty easy to determine which host is scanning if only one is actually up on the network.

Use IP addresses instead of names (so the decoy networks don't see you in their nameserver logs).

# FIREWALL/IDS EVASION

## SPOOF SOURCE ADDRESS

- -S <IP\_Address> (Spoof source address)

In some circumstances, Nmap may not be able to determine your source address (Nmap will tell you if this is the case). In this situation, use -S with the IP address of the interface you wish to send packets through.

Another possible use of this flag is to spoof the scan to make the targets think that someone else is scanning them

The -e option and -Pn are generally required for this sort of usage.

# FIREWALL/IDS EVASION

## SPOOF SOURCE ADDRESS

```
root@kali:~# nmap -e eth0 -S 192.168.2.199 192.168.2.183
WARNING: If -S is being used to fake your source address, you may also have to use -e <interface>[!<interface>] if you are using it to specify your real source address, you can ignore this warning.
Starting Nmap 7.12 ( https://nmap.org ) at 2016-08-06 32:06 EDT
Nmap scan report for 192.168.2.103
Host is up (ping) with mac 00:0c:29:83:ed:69 [eth0].
Nmap scan report for 192.168.2.103
Host is up (ping) with mac 00:0c:29:83:ed:69 [eth0].
Not shown: 2984 closed ports
PORT      STATE    SERVICE
135/tcp    open     msrpc
139/tcp    open     netbios ssn
445/tcp    open     microsoft ds
554/tcp    open     rtsp
802/tcp    open     iss realsecure
912/tcp    open     apex mesn
1025/tcp   filtered NFS-or-IIS
1026/tcp   open     LSA-or-ntern
1027/tcp   open     IIS
1028/tcp   open     unknown
1029/tcp   open     [4ms-lsa] on interface 0
1030/tcp   open     iad1
2869/tcp   open     icslap
3389/tcp   open     ms-wbt-server
3397/tcp   open     wsdapi
10243/tcp  open     unknown
MAC Address: 00:0C:29:83:ED:69 (VMware)
Nmap done: 1 IP address (1 host up) scanned in 15.71 seconds
root@kali:~#
```

# FIREWALL/IDS EVASION

## SPOOF SOURCE ADDRESS

No.	Time	Source	Destination	Protocol	Length	Info	New Column
1002	53.907361000	192.168.2.199	192.168.2.103	TCP	60	58923 > blackjack [SYN] Seq=0 Win=1024 blackjack	
1003	53.907361000	192.168.2.199	192.168.2.103	TCP	60	58923 > auth [SYN] Seq=0 Win=1024 Len=auth	
1007	53.907361000	192.168.2.199	192.168.2.103	TCP	60	58923 > http-alt [SYN] Seq=0 Win=1024 http-alt	
1008	53.907361000	192.168.2.199	192.168.2.103	TCP	60	58923 > telnet [SYN] Seq=0 Win=1024 L telnet	
1009	53.907361000	192.168.2.199	192.168.2.103	TCP	60	58923 > mysql [SYN] Seq=0 Win=1024 Len=mysql	
1010	53.907361000	192.168.2.199	192.168.2.103	TCP	60	58923 > domain [SYN] Seq=0 Win=1024 L domain	
1011	53.907361000	192.168.2.199	192.168.2.103	TCP	60	58923 > https [SYN] Seq=0 Win=1024 Len=https	
1012	53.907361000	192.168.2.199	192.168.2.103	TCP	60	58923 > ppulp [SYN] Seq=0 Win=1024 Len=ppulp	
1019	53.910988000	192.168.2.199	192.168.2.103	TCP	60	58923 > rtsp [SYN] Seq=0 Win=1024 Len=rtsp	
1020	53.910988000	192.168.2.199	192.168.2.103	TCP	60	58923 > pop3 [SYN] Seq=0 Win=1024 Len=pop3	
1021	53.910989000	192.168.2.199	192.168.2.103	TCP	60	58923 > smux [SYN] Seq=0 Win=1024 Len=smux	
1025	53.911599000	192.168.2.199	192.168.2.103	TCP	60	58923 > rfb [SYN] Seq=0 Win=1024 Len=rfb	
1026	53.911600000	192.168.2.199	192.168.2.103	TCP	60	58923 > netbios-ssn [SYN] Seq=0 Win=1 netbios-ssn	
1027	53.911600000	192.168.2.199	192.168.2.103	TCP	60	58923 > cpmmp [SYN] Seq=0 Win=1024 L cpmmp	
1028	53.911601000	192.168.2.199	192.168.2.103	TCP	60	58923 > smtp [SYN] Seq=0 Win=1024 Len=smtp	
1029	53.911601000	192.168.2.199	192.168.2.103	TCP	60	58923 > microsoft_db [SYN] Seq=0 Win=microsoft db	
1030	53.911601000	192.168.2.199	192.168.2.103	TCP	60	58923 > h22jhostcall [SYN] Seq=0 Win=h22jhostcall	
1031	53.911601000	192.168.2.199	192.168.2.103	TCP	60	58923 > ftp [SYN] Seq=0 Win=1024 Len=ftp	
1032	53.911602000	192.168.2.199	192.168.2.103	TCP	60	58923 > imaps [SYN] Seq=0 Win=1024 Len=imaps	
1033	53.911602000	192.168.2.199	192.168.2.103	TCP	60	58923 > rup [SYN] Seq=0 Win=1024 Len=rup	
1043	53.912426000	192.168.2.199	192.168.2.103	TCP	60	58923 > ssh [SYN] Seq=0 Win=1024 Len=ssh	
1044	53.912427000	192.168.2.199	192.168.2.103	TCP	60	58923 > imap [SYN] Seq=0 Win=1024 Len=imap	
1045	53.912427000	192.168.2.199	192.168.2.103	TCP	60	58923 > sunrpc [SYN] Seq=0 Win=1024 L sunrpc	
1046	53.912427000	192.168.2.199	192.168.2.103	TCP	60	58923 > ms_wbt_server [SYN] Seq=0 Win=ms_wbt_server	
1047	53.912427000	192.168.2.199	192.168.2.103	TCP	60	58923 > submission [SYN] Seq=0 Win=10 submission	
1048	53.912427000	192.168.2.199	192.168.2.103	TCP	60	58923 > pop3s [SYN] Seq=0 Win=1024 L pop3s	
1055	53.915546000	192.168.2.199	192.168.2.103	TCP	60	58923 > nfs [SYN] Seq=0 Win=1024 Len=nfs	
1057	53.915546000	192.168.2.199	192.168.2.103	TCP	60	58923 > oracle_db [SYN] Seq=0 Win=oracle_db https	
1058	53.915546000	192.168.2.199	192.168.2.103	TCP	60	58923 > rxapi [SYN] Seq=0 Win=1024 Len=rxapi	
1061	53.915784000	192.168.2.199	192.168.2.103	TCP	60	58923 > ewall [SYN] Seq=0 Win=1024 L ewall	

# FIREWALL/IDS EVASION

## USE SPECIFIED INTERFACE

- -e <interface> (Use specified interface)

Tells Nmap what interface to send and receive packets on. Nmap should be able to detect this automatically, but it will tell you if it cannot.

# FIREWALL/IDS EVASION

## SPECIFY INTERFACE

```
root@kali:~# nmap -e eth0 -S 192.168.2.199 192.168.2.183
WARNING: If -S is being used to fake your source address, you may also have to use -e <interface>[!<interface>] if you are using it to specify your real source address, you can ignore this warning.
Starting Nmap 7.12 ( https://nmap.org ) at 2016-08-06 32:06 EDT
Nmap scan report for 192.168.2.103
Host is up (ping) with latency 0.0013s (0.0013s latency).
Not shown: 2984 closed ports
PORT      STATE    SERVICE
135/tcp    open     msrpc
139/tcp    open     netbios ssn
445/tcp    open     microsoft ds
554/tcp    open     rtsp
902/tcp    open     iss realsecure
912/tcp    open     apex mesn
1025/tcp   filtered NFS-or-IIS
1026/tcp   open     LSA-or-ntern
1027/tcp   open     IIS
1028/tcp   open     unknown
1029/tcp   open     [closed (14ms-lsa)] on interface 0
1030/tcp   open     iad1
2869/tcp   open     icslap
3389/tcp   open     ms-wbt-server
3397/tcp   open     wsdapi
10243/tcp  open     unknown
MAC Address: 00:0C:29:83:ED:69 (VMware)
Nmap done: 1 IP address (1 host up) scanned in 15.71 seconds
root@kali:~#
```

# FIREWALL/IDS EVASION

## USE SPOOF SOURCE PORT NUMBER

- `--source-port <portnumber>; -g <portnumber>`

Nmap will send packets from define port.

Most scanning operations that use raw sockets, including SYN and UDP scans, support the option completely.

The option notably doesn't have an effect for any operations that use normal operating system sockets, including DNS requests, TCP connect scan, version detection, and script scanning

# FIREWALL/IDS EVASION

## SPOOF SOURCE PORT

```
root@kali:~# nmap --source-port 53 192.168.2.103 [x] [x] [x] ?  
Starting Nmap 7.12 ( https://nmap.org ) at 2016-08-06 02:37 EDT  
Nmap scan report for 192.168.2.103  
Host is up (0.0016s latency).  
Not shown: 984 closed ports  
PORT      STATE     SERVICE  
135/tcp    open      msrpc  
139/tcp    open      netbios-ssn  
445/tcp    open      microsoft-ds  
554/tcp    open      rtsp  
902/tcp    open      iss-realsecure  
912/tcp    open      apex-mesh  
1025/tcp   filtered NFS-or-IIS  
1026/tcp   open      LSA-or-nterm  
1027/tcp   open      IIS  
1028/tcp   open      unknown  
1029/tcp   open      ms-lsa  
1030/tcp   open      iad1  
2869/tcp   open      icslap  
3389/tcp   open      ms-wbt-server  
5357/tcp   open      wsddapi  
10243/tcp  open      unknown  
MAC Address: 00:0C:29:83:ED:69 (VMware)  
Nmap done: 1 IP address (1 host up) scanned in 16.89 seconds  
root@kali:~#
```

# FIREWALL/IDS EVASION

## WIRESHARK CAPTURE

File: [i:J:\Juel--192.168.2.103](#) Expression... Clear Apply Save

No.	Time	Source	Destination	Protocol	Length	Info	New Column
domain47_2325210	192.168.2.104	192.168.2.103	TCP	60	domain > nzbjndstcarisc [SYN] Seq=0 Win=1024 Len=0	53	
domain47_2325220	192.168.2.104	192.168.2.103	TCP	60	domain > wap-vcal-s [SYN] Seq=0 Win=1024 Len=0 MSS=1460	53	
domain47_2325220	192.168.2.104	192.168.2.103	TCP	60	domain > 10566 [SYN] Seq=0 Win=1024 Len=0 MSS=1460	53	
domain47_2325220	192.168.2.104	192.168.2.103	TCP	60	domain > 2002 [SYN] Seq=0 Win=1024 Len=0 MSS=1460	53	
domain47_2330020	192.168.2.104	192.168.2.103	TCP	60	domain > 20828 [SYN] Seq=0 Win=1024 Len=0 MSS=1460	53	
domain47_2330030	192.168.2.104	192.168.2.103	TCP	60	domain > c13 software 1 [SYN] Seq=0 Win=1024 Len=0	53	
domain47_2330030	192.168.2.104	192.168.2.103	TCP	60	domain > smc-fhttp [SYN] Seq=0 Win=1024 Len=0 MSS=1	53	
domain47_2330030	192.168.2.104	192.168.2.103	TCP	60	domain > 121/1 [SYN] Seq=0 Win=1024 Len=0 MSS=1460	53	
domain47_2330030	192.168.2.104	192.168.2.103	TCP	60	domain > 1998 [SYN] Seq=0 Win=1024 Len=0 MSS=1460	53	
domain47_2330040	192.168.2.104	192.168.2.103	TCP	60	domain > caids-sensor [SYN] Seq=0 Win=1024 Len=0 M	53	
domain47_2330040	192.168.2.104	192.168.2.103	TCP	60	domain > writesrv [SYN] Seq=0 Win=1024 Len=0 MSS=1	53	
domain47_2336720	192.168.2.104	192.168.2.103	TCP	60	domain > veritas-phx [SYN] Seq=0 Win=1024 Len=0 M	53	
domain47_2336720	192.168.2.104	192.168.2.103	TCP	60	domain > mtpq [SYN] Seq=0 Win=1024 Len=0 MSS=1460	53	
domain47_2336720	192.168.2.104	192.168.2.103	TCP	60	domain > apf_3052 [SYN] Seq=0 Win=1024 Len=0 MSS=1	53	
domain47_2336720	192.168.2.104	192.168.2.103	TCP	60	domain > finger [SYN] Seq=0 Win=1024 Len=0 MSS=146	53	
domain47_2336730	192.168.2.104	192.168.2.103	TCP	60	domain > ph [SYN] Seq=0 Win=1024 Len=0 MSS=1460	53	
domain47_2336730	192.168.2.104	192.168.2.103	TCP	60	domain > 19999 [SYN] Seq=0 Win=1024 Len=0 MSS=1460	53	
domain47_2336730	192.168.2.104	192.168.2.103	TCP	60	domain > 880 [SYN] Seq=0 Win=1024 Len=0 MSS=1460	53	
domain47_2336730	192.168.2.104	192.168.2.103	TCP	60	domain > 7920 [SYN] Seq=0 Win=1024 Len=0 MSS=1460	53	
domain47_2336730	192.168.2.104	192.168.2.103	TCP	60	domain > backup-express [SYN] Seq=0 Win=1024 Len=0	53	
domain47_2345940	192.168.2.104	192.168.2.103	TCP	60	domain > doom [SYN] Seq=0 Win=1024 Len=0 MSS=1460	53	
domain47_2345950	192.168.2.104	192.168.2.103	TCP	60	domain > 1010 heart [SYN] Seq=0 Win=1024 Len=0 MSS=	53	
domain47_2345950	192.168.2.104	192.168.2.103	TCP	60	domain > sumit_ig [SYN] Seq=0 Win=1024 Len=0 MSS=	53	
domain47_2372790	192.168.2.104	192.168.2.103	TCP	60	domain > als3 callback [SYN] Seq=0 Win=1024 Len=0	53	
domain47_2375270	192.168.2.104	192.168.2.103	TCP	60	domain > rsh-spx [SYN] Seq=0 Win=1024 Len=0 MSS=14	53	
domain47_2375270	192.168.2.104	192.168.2.103	TCP	60	domain > boinc-client [SYN] Seq=0 Win=1024 Len=0 M	53	
domain47_2375280	192.168.2.104	192.168.2.103	TCP	60	domain > 32779 [SYN] Seq=0 Win=1024 Len=0 MSS=1460	53	
domain47_2379400	192.168.2.104	192.168.2.103	TCP	60	domain > corbaloc [SYN] Seq=0 Win=1024 Len=0 MSS=1	53	
domain47_2379410	192.168.2.104	192.168.2.103	TCP	60	domain > funk-dialout [SYN] Seq=0 Win=1024 Len=0 M	53	
domain47_2379410	192.168.2.104	192.168.2.103	TCP	60	domain > geniuslm [SYN] Seq=0 Win=1024 Len=0 MSS=1	53	
domain47_2379410	192.168.2.104	192.168.2.103	TCP	60	domain > 55056 [SYN] Seq=0 Win=1024 Len=0 MSS=1460	53	
domain47_2379410	192.168.2.104	192.168.2.103	TCP	60	domain > 50800 [SYN] Seq=0 Win=1024 Len=0 MSS=1460	53	

Frame 411: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0  
Ethernet II, Src: vmware\_6e:88:83 (00:0c:29:6e:88:83), Dst: vmware\_83:ed:69 (00:0c:29:83:ed:69)

```
0000  00 0c 29 6e 88 00 00 0c 29 6e 88 81 08 00 45 00  ..).1.. )n....F.
0010  00 2c 96 4f 00 00 30 06 6e 5d c0 a8 02 58 c0 a8  ..,.0..0. n]....n.
0020  02 67 00 35 00 87 d2 1f 22 15 00 00 00 00 60 02  .q.5.... .....
0030  04 00 19 16 00 00 02 04 05 h4 00 00  ..... ....
```

File: C:\Users\juel\appData\Local\Temp\wiresh... | Packets: 2527 Displayed: 2093 Marked: 0 Dropped: 0 | Profile: Default

# FIREWALL/IDS EVASION

## APPEND DATA TO SENT PACKETS

--data <hex string>

This option lets you include binary data as payload in sent packets.

--data 0xdeadbeef and --data \xCA\xFE\x09

--data-string <string>

This option lets you include a regular string as payload in sent packets.

--data-string "Scan test"

# FIREWALL/IDS EVASION

## APPEND DATA

```
root@kali:~# nmap --data 0xabadabba 192.168.2.103
Starting Nmap 7.12 ( https://nmap.org ) at 2016-08-06 02:45 EDT
Nmap scan report for 192.168.2.103
Host is up (0.00064s latency).
Not shown: 984 closed ports
PORT      STATE SERVICE
135/tcp    open  msrpc
139/tcp    open  netbios-ssn
445/tcp    open  microsoft-ds
554/tcp    open  rtsp
902/tcp    open  iss-realsecure
912/tcp    open  apex-mesh
1025/tcp   filtered NFS-or-IIS
1026/tcp   open  LSA-or-nterm
1027/tcp   open  IIS
1028/tcp   open  unknown
1029/tcp   open  ms-lsa
1030/tcp   open  iad1
2869/tcp   open  icslap
3389/tcp   open  ms-wbt-server
5357/tcp   open  wsddapi
10243/tcp  open  unknown
MAC Address: 00:0C:29:83:ED:69 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 17.35 seconds
root@kali:~#
```

# FIREWALL/IDS EVASION

## WIRESHARK CAPTURE

No.	Time	Source	Destination	Protocol	Length	Info
124	41.170588000	192.168.2.103	192.168.2.102	TCP	60	1481→8895 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
132	44.505255000	192.168.2.104	192.168.2.103	NDS	62	NDS Continuation Message
133	44.505385000	192.168.2.104	192.168.2.103	RPC	62	Continuation
134	44.505450000	192.168.2.104	192.168.2.103	TCP	62	60933→135 [SYN] Seq=0 Win=1024 Len=4 MSS=1460
135	44.505531000	192.168.2.104	192.168.2.103	TCP	62	60933→113 [SYN] Seq=0 Win=1024 Len=4 MSS=1460
136	44.505580000	192.168.2.104	192.168.2.103	POP	62	C:\253\2\2\253\2\2
137	44.505657000	192.168.2.104	192.168.2.103	TCP	62	[TCP segment of a reassembled PDU]
-----						
.... .0. .... = Urgent: Not set						
.... ...0 .... = Acknowledgment: Not set						
.... .... 0... = Push: Not set						
.... .... ..0.. = Reset: Not set						
► .... .... ..1. = Syn: Set						
.... .... ..0 = -in: Not set						
window size value: 1024						
[Calculated window size: 1024]						
* Checksum: 0x2301 [validator disabled]						
[Good Checksum: False]						
[Bad Checksum: False]						
Urgent pointer: 0						
* Options: (4 bytes), Maximum segment size						
► Maximum segment size: 1460 bytes						
► [SEQ/ACK analysis]						
* Data (1 bytes)						
Data: abbaabba						
[Length: 4]						
0000	00 0c 29 83 ed 69 00 0c 29 6e 88 83 08 00 45 00	..).i.. )n....E.				
0010	00 30 c4 85 00 00 27 06 49 23 c0 a8 02 68 c0 c8	.0..... I#...h..				
0020	02 67 ee 05 00 87 0c 76 38 89 00 00 00 00 02	.g.....lv 8.....`				
0030	04 00 23 01 00 00 02 04 05 b4 ab ba ab ba	..#.....				

# FIREWALL/IDS EVASION

## APPEND DATA TO SENT PACKETS

--data-length <number>

This option tells Nmap to append the given number of random bytes to most of the packets it sends, and not to use any protocol-specific payloads.

TCP packets are generally 40 bytes and ICMP echo requests are just 28. Some UDP ports and IP protocols get a custom payload by default.

--data-length 0 for no random or protocol-specific payloads.

# FIREWALL/IDS EVASION

## APPEND DATA

```
root@kali:~# nmap --data-length 55 192.168.2.103
Starting Nmap 7.12 ( https://nmap.org ) at 2016-08-06 02:54 EDT
Nmap scan report for 192.168.2.103
Host is up (0.0017s latency).Session... Clear Apply Save
Not shown: 984 closed ports
PORT      STATE SERVICE          | Protocol | Length | Info
135/tcp    open  msrpc           | POP      | 113     | C: \230\2255\260\255\37
139/tcp    open  netbios-ssn     | NBSS    | 113     | NBSS Continuation Message
445/tcp    open  microsoft-ds    | SSL      | 113     | Continuation Data
554/tcp    open  rtsp            | TCP      | 113     | 56803→80 [SYN] Seq=0 Win
902/tcp    open  iiss-realsecure | TCP      | 113     | [TCP segment of a reasse...
912/tcp    open  apex-mesh       | TELNET   | 113     | Telnet Data ...
1025/tcp   filtered NFS-or-IIS | TCP      | 113     | 56803→8888 [SYN] Seq=0 Win
1026/tcp   open  LSA-or-nterm   | PPTP    | 113     | Unknown control type (42)
1027/tcp   open  IIS             | TCP      | 113     | 56803→8080 [SYN] Seq=0 Win
1028/tcp   open  unknown         | SMUX    | 113     |
1029/tcp   open  ms-lsa          | TCP      | 60      | 199→56803 [RST, ACK] Seq=0 Win
1030/tcp   open  iadl             | TCP      | 113     | 56803→1025 [SYN] Seq=0 Win
2869/tcp   open  icslap          | RTSP    | 113     | Continuation
3389/tcp   open  ms-wbt-server  |
5357/tcp   open  wsdapi          |
10243/tcp  open  unknown         |
MAC Address: 00:0C:29:83:ED:69 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 17.85 seconds
```

# FIREWALL/IDS EVASION

## WIRESHARK CAPTURE

No.	Time	Source	Destination	Protocol	Length	Info
8	2.914864000	192.168.2.104	192.168.2.103	POP	113	C: \230\2255\26
9	2.915078000	192.168.2.104	192.168.2.103	NBSS	113	NBSS Continuation
10	2.915164000	192.168.2.104	192.168.2.103	SSL	113	Continuation Data
11	2.915216000	192.168.2.104	192.168.2.103	TCP	113	56803→80 [SYN] S
12	2.915287000	192.168.2.104	192.168.2.103	TCP	113	[TCP segment of ]
13	2.915392000	192.168.2.104	192.168.2.103	TELNET	113	Telnet Data ...
14	2.915460000	192.168.2.104	192.168.2.103	TCP	113	56803→8888 [SYN]
15	2.915522000	192.168.2.104	192.168.2.103	PPTP	113	Unknown control
16	2.915586000	192.168.2.104	192.168.2.103	TCP	113	56803→8080 [SYN]
17	2.915645000	192.168.2.104	192.168.2.103	SMUX	113	
20	2.917954000	192.168.2.103	192.168.2.104	TCP	60	199→56803 [RST, ]
21	2.920077000	192.168.2.104	192.168.2.103	TCP	113	56803→1025 [SYN]
22	2.920188000	192.168.2.104	192.168.2.103	RTSP	113	Continuation

[Time since reference or first frame: 2.915392000 seconds]

Frame Number: 13

Frame Length: 113 bytes (904 bits)

Capture Length: 113 bytes (904 bits)

[Frame is marked: False]

[Frame is ignored: False]

[Protocols in frame: eth:ethertype:ip:tcp:telnet]

[Coloring Rule Name: TCP SYN/FIN]

[Coloring Rule String: tcp.flags & 0x02 || tcp.flags.fin == 1]

Ethernet II Src: VMware\_60-00-02 (00:0c:29:60:00:02) Dst: VMware\_02-ed-60 (00:0c:29:02:ed:60)

# FIREWALL/IDS EVASION

## SPECIFIED IP OPTIONS

--ip-options <S|R [route]|L [route]|T|U ... >; --ip-options <hex string>

For example, \x01\x07\x04\x00\*36\x01 is a hex string containing 36 NULL bytes.

--ttl <value> (Set IP time-to-live field)

Sets the IPv4 time-to-live field in sent packets to the given value.

# FIREWALL/IDS EVASION

## RANDOMIZE TARGET HOST

--randomize-hosts

Tells Nmap to shuffle each group of up to 16384 hosts before it scans them. This can make the scans less obvious to various network monitoring systems, especially when you combine it with slow timing options.

# FIREWALL/IDS EVASION

## RANDOMIZE HOSTS

```
root@kali:~# nmap -PS -sn --randomize-hosts 192.168.2.100-254
Starting Nmap 7.12 ( https://nmap.org ) at 2016-08-06 03:20 EDT [semiblinded P
Nmap scan report for 192.168.2.107           113 Telnet Data ...
Host is up (0.26s latency).  TCP             113 56803-8888 [SYN] Seq=0 Win=1024
MAC Address: A4:E1:E8:9A:89:65 (Unknown)    113 Unknown control type (42369)
Nmap scan report for 192.168.2.108           113 56803-8080 [SYN] Seq=0 Win=1024
Host is up (0.26s latency).  SMUX            113
MAC Address: 00:1C:7B:75:A3:EC (Castlenet Technology) RST, ACK] Seq=1 Ack=
Nmap scan report for 192.168.2.103           113 56803-1025 [SYN] Seq=0 Win=1024
Host is up (0.00019s latency).  RTSP           113 Continuation
MAC Address: 00:0C:29:83:ED:69 (VMware)
Nmap scan report for 192.168.2.100
Host is up (0.00013s latency).
MAC Address: 20:C9:D0:DB:93:9F (Apple)
Nmap scan report for 192.168.2.102
Host is up (0.26s latency).
MAC Address: 28:CF:DA:00:B1:B1 (Apple)
Nmap scan report for 192.168.2.104
Host is up.
Nmap done: 155 IP addresses (6 hosts up) scanned in 29.10 seconds
root@kali:~#
```

# FIREWALL/IDS EVASION

## SPOOF MAC ADDRESS

--spoof-mac <MAC address, prefix, or vendor name>

Asks Nmap to use the given MAC address for all of the raw ethernet frames it sends. This option implies --send-eth to ensure that Nmap actually sends ethernet-level packets.

If it is simply the number 0, Nmap chooses a completely random MAC address for the session

Valid --spoof-mac argument examples are Apple, 0, 01:02:03:04:05:06, deadbeefcafe, 0020F2, and Cisco

# FIREWALL/IDS EVASION

## SPOOF MAC ADDRESS

```
root@kali:~# nmap -p135 --spoof-mac 01:02:03:04:05:06 192.168.2.103
:25:15.485403000 EDT
[0000000000 seconds]
Starting Nmap 7.12 ( https://nmap.org ) at 2016-08-06 03:25 EDT
[0000000000 seconds]
Spoofing MAC address 01:02:03:04:05:06 (No registered vendor)
Nmap scan report for 192.168.2.103
Host is up:(0.00048s latency).
PORT      STATE SERVICE
135/tcp    open  msrpc
MAC Address: 00:0C:29:83:ED:69 (VMware)
[bits]
Nmap done: 1 IP address (1 host up) scanned in 13.06 seconds
root@kali:~# █
[!type:ip:tcp]
[FIN]
[flags & 0x02 || tcp.flags.fin == 1]
```

# FIREWALL/IDS EVASION

## WIRESHARK CAPTURE

No.	Time	Source	Destination	Protocol	Length	Info
2	0.695192000	192.168.2.104	192.168.2.103	TCP	58	55093→135 [SYN] Seq=0 Win=102
3	0.696207000	192.168.2.103	192.168.2.104	TCP	60	135→55093 [SYN, ACK] Seq=0 Ack=1
9	3.711587000	192.168.2.103	192.168.2.104	TCP	60	[TCP Retransmission] 135→55093

Encapsulation type: Ethernet (1)  
Arrival Time: Aug 6, 2016 03:25:15.484388000 EDT  
[Time shift for this packet: 0.000000000 seconds]  
Epoch Time: 1470468315.484388000 seconds  
[Time delta from previous captured frame: 0.695192000 seconds]  
[Time delta from previous displayed frame: 0.000000000 seconds]  
[Time since reference or first frame: 0.695192000 seconds]  
Frame Number: 2  
Frame Length: 58 bytes (464 bits)  
Capture Length: 58 bytes (464 bits)  
[Frame is marked: False]  
[Frame is ignored: False]  
[Protocols in frame: eth:ethertype:ip:tcp]  
[Coloring Rule Name: TCP SYN/FIN]  
[Coloring Rule String: tcp.flags & 0x02 || tcp.flags.fin == 1]

► Ethernet II, Src: Woonsang\_04:05:06 (01:02:03:04:05:06), Dst: Vmware\_83:ed:69 (00:0c:29:83:ed:69)  
▼ Internet Protocol Version 4, Src: 192.168.2.104 (192.168.2.104), Dst: 192.168.2.103 (192.168.2.103)  
Version: 4  
0000 00 0c 29 83 ed 69 01 02 03 04 05 06 08 00 45 00 ..).i.....E.  
0010 00 2c 8a 8d 00 00 2e 06 7c 1f c0 a8 02 68 c0 a8 .,.....|. ....h..  
0020 02 67 d7 35 00 87 74 3b d1 7d 00 00 00 00 60 02 .g.5..t;.}....`.  
0030 04 00 f0 90 00 00 02 04 05 b4 ..... .

# FIREWALL/IDS EVASION

## RELAY TCP CONNECTIONS THROUGH A CHAIN OF PROXIES

--proxies <Comma-separated list of proxy URLs>

Asks Nmap to establish TCP connections with a final target through supplied chain of one or more HTTP or SOCKS4 proxies. Proxies can help hide the true source of a scan or evade certain firewall restrictions, but they can hamper scan performance by increasing latency.

proto://host:port (Valid protocols are HTTP and SOCKS4.)

```
#nmap --proxies socks4://202.44.56.78:8090 -p80 -sV -vv google.com
```

**Warning: this feature is still under development and has limitations.** It is implemented within the nsock library and thus has no effect on the ping, port scanning and OS discovery phases of a scan. Only NSE and version scan benefit from this option so far—other features may disclose your true address. SSL connections are not yet supported, nor is proxy-side DNS resolution (hostnames are always resolved by Nmap).

# TIMING AND PERFORMANCE



# NMAP ADVANCED OPTIONS

## TIMING AND PERFORMANCE

- Many Nmap features have configurable timing options.
- Timing options can be used to speed up or slow down scanning operations depending on needs

# TIMING AND PERFORMANCE

## TIMING TEMPLATES

- -T<0-5>: Set timing template (higher is faster)

Template	Name	Notes
-T0	Paranoid	Extremely slow
-T1	Sneaky	Useful for avoiding intrusion detection
-T2	Polite	Unlikely to interfere with the target system
-T3	Normal	This is the default timing template
-T4	Aggressive	Produces faster results on local networks
-T5	Insane	Very fast and aggressive scan

# TIMING AND PERFORMANCE

## TIMING TEMPLATES

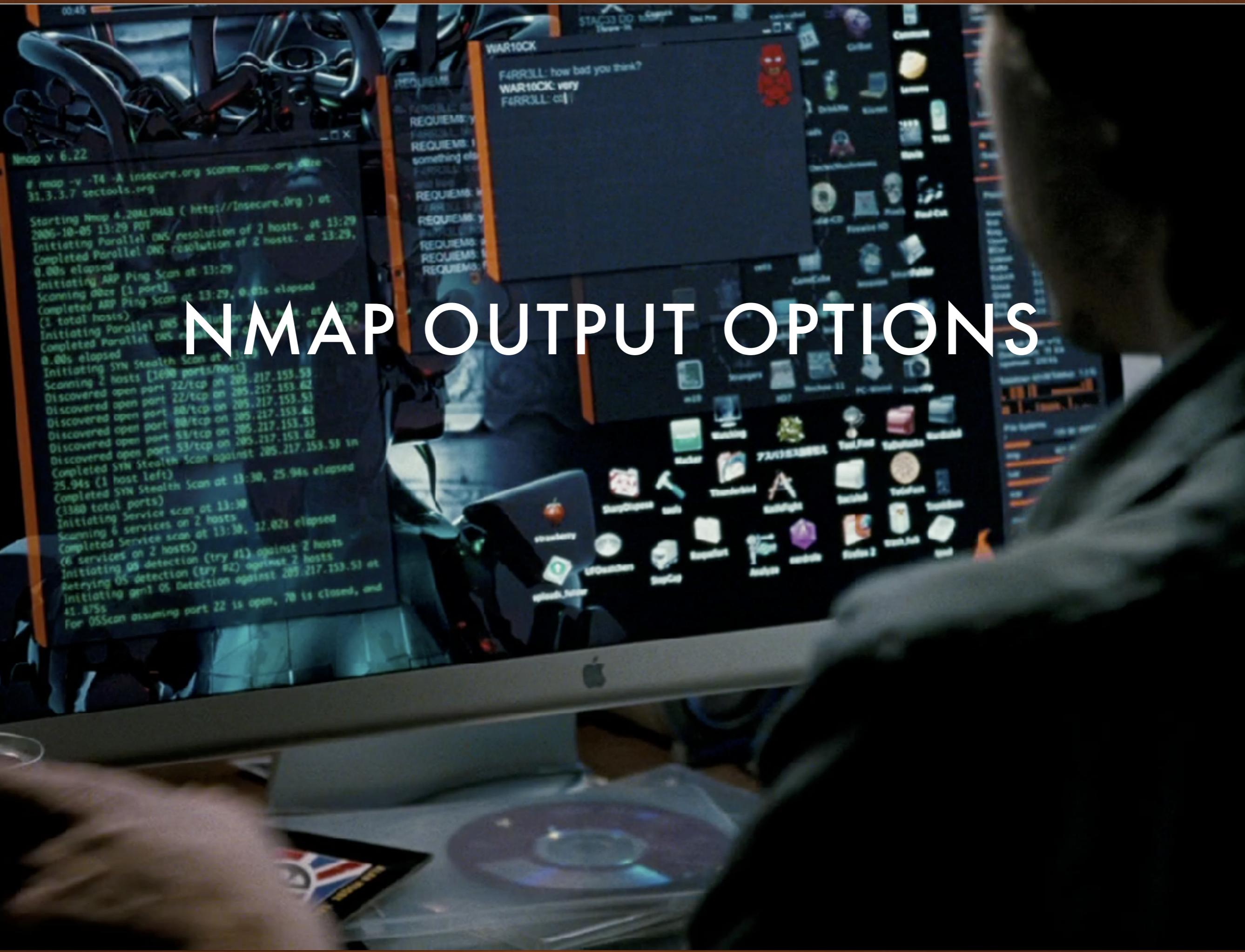
```
root@kali:~/Desktop/result# nmap -T4 192.168.2.103
Starting Nmap 7.12 ( https://nmap.org ) at 2016-08-07 03:02 EDT
Nmap scan report for 192.168.2.103
Host is up (0.0013s latency).
Not shown: 984 closed ports
PORT      STATE    SERVICE
135/tcp    open     msrpc
139/tcp    open     netbios-ssn
445/tcp    open     microsoft-ds
554/tcp    open     rtsp
902/tcp    open     iss-realsecure
912/tcp    open     apex-mesh
1025/tcp   filtered NFS-or-IIS
1026/tcp   open     LSA-or-nterm
1027/tcp   open     IIS
1028/tcp   open     unknown
1039/tcp   open     sbl
1042/tcp   open     afrog
2869/tcp   open     icslap
3389/tcp   open     ms-wbt-server
5357/tcp   open     wsdapi
10243/tcp  open     vfork
MAC Address: 00:0C:29:83:ED:69 (VMware)
Nmap done: 1 IP address (1 host up) scanned in 15.38 seconds
```

# TIMING AND PERFORMANCE

## OTHER OPTIONS

- `--max-retries` `numtries` (Specify the maximum number of port scan probe retransmissions) .
- `--host-timeout` `time` (Give up on slow target hosts) .
- `--scan-delay` `time`; `--max-scan-delay` `time` (Adjust delay between probes) .

# NMAP OUTPUT OPTIONS



# NMAP OUTPUT

Nmap offers several options for creating formatted output.

1. Interactive/stdout Output
2. Save Output to a Text File
3. Save Output to XML File
4. Grepable Output
5. Output All Supported File Types
6. sl<rIpt kIddi3 Output

# NMAP OUTPUT

## INTERACTIVE OUTPUT

Interactive output is what Nmap prints to the stdout stream, which usually appears on the terminal window you executed Nmap from.

# NMAP OUTPUT

## INTERACTIVE OUTPUT

```
root@kali:~# nmap 192.168.2.103

Starting Nmap 7.12 ( https://nmap.org ) at 2016-08-06 05:55 EDT
Nmap scan report for 192.168.2.103
Host is up (0.0017s latency).
Not shown: 984 closed ports
PORT      STATE    SERVICE
135/tcp    open     msrpc
139/tcp    open     netbios-ssn
445/tcp    open     microsoft-ds
554/tcp    open     rtsp
902/tcp    open     iss-realsecure
912/tcp    open     apex-mesh
1025/tcp   filtered NFS-or-IIS
1026/tcp   open     LSA-or-nterm
1027/tcp   open     IIS
1028/tcp   open     unknown
1029/tcp   open     ms-lsa
1030/tcp   open     iadl
2869/tcp   open     icslap
3389/tcp   open     ms-wbt-server
5357/tcp   open     wsdapi
10243/tcp  open     unknown
MAC Address: 00:0C:29:83:ED:69 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 15.42 seconds
root@kali:~#
```

# NMAP OUTPUT

## NORMAL OUTPUT

-oN: Normal Output

It is similar to interactive output, except that notes which lose relevance once a scan completes are removed.

It is assumed that the file will be read after Nmap completes, so estimated completion times and new open port alerts are redundant to the actual completion time and the ordered port table.

Since output may be saved a long while and reviewed among many other logs, Nmap prints the execution time, command-line arguments, and Nmap version number on the first line.

# NMAP OUTPUT

## NORMAL OUTPUT

```
root@kali:~# nmap 192.168.2.103 -oN /root/Desktop/result/nmap103.txt

Starting Nmap 7.12 ( https://nmap.org ) at 2016-08-06 05:58 EDT
Nmap scan report for 192.168.2.103
Host is up (0.0028s latency).
Not shown: 984 closed ports
PORT      STATE    SERVICE
135/tcp    open     msrpc
139/tcp    open     netbios-ssn
445/tcp    open     microsoft-ds
554/tcp    open     rtsp
902/tcp    open     iss-realsecure
912/tcp    open     apex-mesh
1025/tcp   filtered NFS-or-IIS
1026/tcp   open     LSA-or-nterm
1027/tcp   open     IIS
1028/tcp   open     unknown
1029/tcp   open     ms-lsa
1030/tcp   open     iad1
2869/tcp   open     icslap
3389/tcp   open     ms-wbt-server
5357/tcp   open     wsdapi
10243/tcp  open     unknown
MAC Address: 00:0C:29:83:ED:69 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 16.29 seconds
```

# NMAP OUTPUT

## NORMAL OUTPUT

```
Open ▾  nmap103.txt ~/Desktop/result Save     
# Nmap 7.12 scan initiated Sat Aug 6 05:58:15 2016 as: nmap -oN /root/Desktop/result/nmap103.txt 192.168.2.103  
Nmap scan report for 192.168.2.103  
Host is up (0.0028s latency).  
Not shown: 984 closed ports  
PORT      STATE    SERVICE  
135/tcp    open     msrpc  
139/tcp    open     netbios-ssn  
445/tcp    open     microsoft-ds  
594/tcp    open     rdp  
932/tcp    open     ciss-realsecure  
912/tcp    open     apex-mosn  
1925/tcp   filtered NFS-or-IIS  
1926/tcp   open     LSA or nterm  
1927/tcp   open     IIS  
1928/tcp   open     unknown  
1929/tcp   open     ns-lsa  
1930/tcp   open     _adl  
2869/tcp   open     _cslap  
3389/tcp   open     ms-wbt-server  
5357/tcp   open     wsdapi  
19243/tcp  open     unknown  
MAC Address: 00:0C:29:83:E0:69 (VMware)  
  
# Nmap done at Sat Aug 6 05:58:32 2016      1 IP address (1 host up) scanned in 16.29 seconds
```

# NMAP OUTPUT

## XML OUTPUT

-oX: XML Output

XML, the extensible markup language

The XML format includes more information than the others and is extensible enough that new features can be added without breaking existing programs that use it. It can be parsed by standard XML parsers, which are available for all popular programming languages

# NMAP OUTPUT

## XML OUTPUT

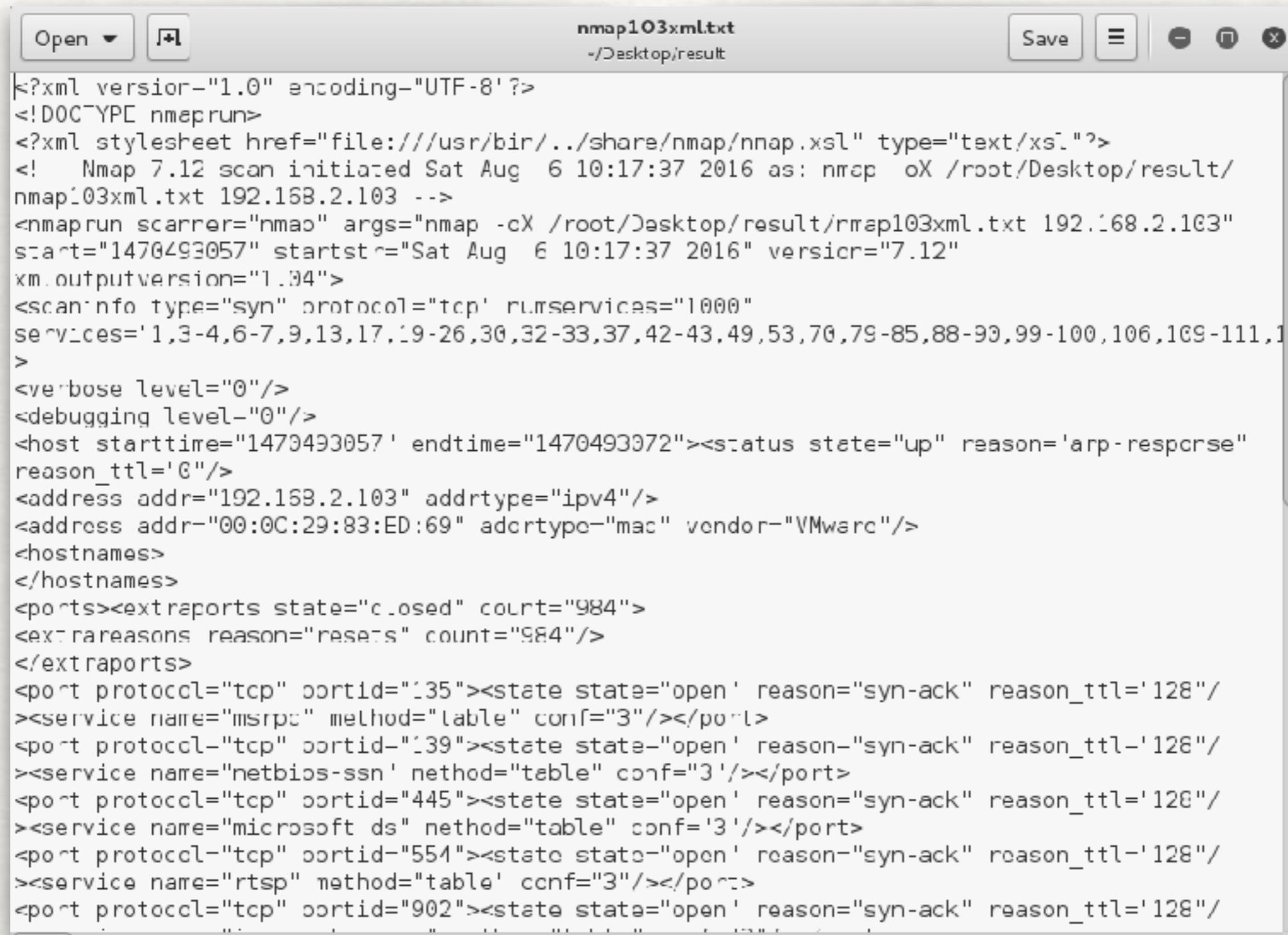
```
root@kali:~# nmap 192.168.2.103 -oXt /root/Desktop/result/nmap103xml.txt

Starting Nmap 7.12 ( https://nmap.org ) at 2016-08-06 10:17 EDT
Nmap scan report for 192.168.2.103
Host is up (0.0012s latency).
Not shown: 984 closed ports
PORT      STATE    SERVICE
135/tcp    open     msrpc
139/tcp    open     netbios-ssn
445/tcp    open     microsoft-ds
554/tcp    open     rtsp
902/tcp    open     iss-realsecure
912/tcp    open     apex-mesh
1025/tcp   filtered NFS-or-IIS
1026/tcp   open     LSA-or-nterm
1027/tcp   open     IIS
1028/tcp   open     unknown
1029/tcp   open     ms-lsa
1030/tcp   open     iad1
2869/tcp   open     icslap
3389/tcp   open     ms-wbt-server
                   "selected (815 bytes)
5357/tcp   open     wsddapi
10243/tcp  open     unknown
MAC Address: 00:0C:29:83:ED:69 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 15.77 seconds
root@kali:~#
```

# NMAP OUTPUT

## XML OUTPUT



The screenshot shows a text editor window with the following details:

- Title Bar:** nmap103xml.txt  
-/Desktop/result
- Buttons:** Open, Save, Minimize, Maximize, Close.

The content of the file is the XML output from an Nmap scan. The XML structure includes:

- Scan Information:** Version 1.0, encoding UTF-8, Nmap 7.12 scan initiated Sat Aug 6 10:17:37 2016.
- Service Scan Details:** Scan type: syn, protocol: tcp, ports: 1,3-4,6-7,9,13,17,19-26,30,32-33,37,42-43,49,53,70,79-85,88-93,99-100,106,109-111,113.
- Host Details:** Host is up, reason: arp-response, reason\_ttl: 0.
- Address Details:** Address 192.168.2.103 (IPv4) and MAC address 00:0C:29:83:ED:69 (VMware).
- Port Scanning Results:** Ports 135, 139, 445, 554, 902 are open. Services identified include msrpc, netbios-ssn, microsoft ds, rtsp, and a table service.

```
<?xml version="1.0" encoding="UTF-8'?>
<!DOCTYPE nmaprun>
<?xmlstylesheet href="file:///usr/share/nmap/nmap.xsl" type="text/xsl"?>
<! Nmap 7.12 scan initiated Sat Aug 6 10:17:37 2016 as: nmap -oX /root/Desktop/result/nmap103xml.txt 192.168.2.103 -->
<nmaprun scanner="nmap" args="nmap -oX /root/Desktop/result/nmap103xml.txt 192.168.2.103" start="1470493057" startstr="Sat Aug 6 10:17:37 2016" version="7.12" xmlobjectversion="1.04">
<scaninfo type="syn" protocol="tcp" runservices="1000" services='1,3-4,6-7,9,13,17,19-26,30,32-33,37,42-43,49,53,70,79-85,88-93,99-100,106,109-111,113'>
<verbose level="0"/>
<debugging level="0"/>
<host starttime="1470493057" endtime="1470493072"><status state="up" reason='arp-response' reason_ttl='0'/>
<address addr="192.168.2.103" addrtype="ipv4"/>
<address addr="00:0C:29:83:ED:69" addrtype="mac" vendor="VMware"/>
<hostnames>
</hostnames>
<ports><extrareports state="closed" count="984">
<extrareasons reason="resets" count="984"/>
</extrareports>
<port protocol="tcp" portid="135"><state state="open" reason="syn-ack" reason_ttl='128'/><service name="msrpc" method="table" conf="3"/></port>
<port protocol="tcp" portid="139"><state state="open" reason="syn-ack" reason_ttl='128'/><service name="netbios-ssn" method="table" conf="3"/></port>
<port protocol="tcp" portid="445"><state state="open" reason="syn-ack" reason_ttl='128'/><service name="microsoft ds" method="table" conf="3"/></port>
<port protocol="tcp" portid="554"><state state="open" reason="syn-ack" reason_ttl='128'/><service name="rtsp" method="table" conf="3"/></port>
<port protocol="tcp" portid="902"><state state="open" reason="syn-ack" reason_ttl='128'/>
```

# NMAP OUTPUT

## GREPABLE OUTPUT

-oG: Grepable Output

This output format is covered last because it is deprecated.

It is a simple format that lists each host on one line and can be trivially searched and parsed with standard Unix tools such as grep, awk, cut, sed, diff, and Perl.

# NMAP OUTPUT

## GREPABLE OUTPUT

```
root@kali:~# nmap -oGt /root/Desktop/result/nmap103grep.txt  
  
Starting Nmap 7.12 ( https://nmap.org ) at 2016-08-06 21:10 EDT  
Nmap scan report for 192.168.2.103  
Host is up (0.00076s latency).  
Not shown: 984 closed ports  
PORT      STATE    SERVICE  
135/tcp    open     msrpc  
139/tcp    open     netbios-ssn  
445/tcp    open     microsoft-ds  
554/tcp    open     rtsp  
902/tcp    open     iss-realsecure  
912/tcp    open     apex-mesh  
1025/tcp   filtered NFS-or-IIS  
1026/tcp   open     LSA-or-nterm  
1027/tcp   open     IIS  
1028/tcp   open     unknown  
1029/tcp   open     ms-lsa  
1030/tcp   open     iad1  
2869/tcp   open     icslap  
3389/tcp   open     ms-wbt-server  
5357/tcp   open     wsdapi  
10243/tcp  open     unknown  
MAC Address: 00:0C:29:83:ED:69 (VMware)  
  
Nmap done: 1 IP address (1 host up) scanned in 16.29 seconds  
root@kali:~#
```

# NMAP OUTPUT

## GREPABLE OUTPUT

```
Open ▾  nmap103grep.txt ~/Desktop/result Save     
# Nmap 7.12 scan initiated Sat Aug 6 21:10:34 2016 as: nmap -oG /root/Desktop/result/nmap103grep.txt 192.168.2.103  
Host: 192.168.2.103 () Status: Up  
Host: 192.168.2.103 () Ports: 135/open/tcp//msrpc///, 139/open/tcp//netbios-ssn///, 445/open/tcp//microsoft-ds///, 554/open/tcp//rtsp///, 902/open/tcp//iss-realsecure///, 912/open/tcp//apex-mesh///, 1025/filtered/tcp//NFS-or-IIS///, 1026/open/tcp//LSA-or-nterm///, 1027/open/tcp//IIS///, 1028/open/tcp//unknown///, 1029/open/tcp//ms-lsa///, 1030/open/tcp//iad1///, 2869/open/tcp//icslap///, 3389/open/tcp//ms-wbt-server///, 5357/open/tcp//wsdapi///, 10243/open/tcp//unknown/// Ignored State: closed (984)  
# Nmap done at Sat Aug 6 21:10:49 2016 -- 1 IP address (1 host up) scanned in 16.29 seconds
```

# NMAP OUTPUT

## ALL OUTPUT

-oA: All Output

The -oA parameter saves the output of a scan in text, grepable, and XML formats.

# NMAP OUTPUT

## ALL OUTPUT

```
root@kali:~/Desktop/result# nmap -oA nmap103-baru 192.168.2.103
```

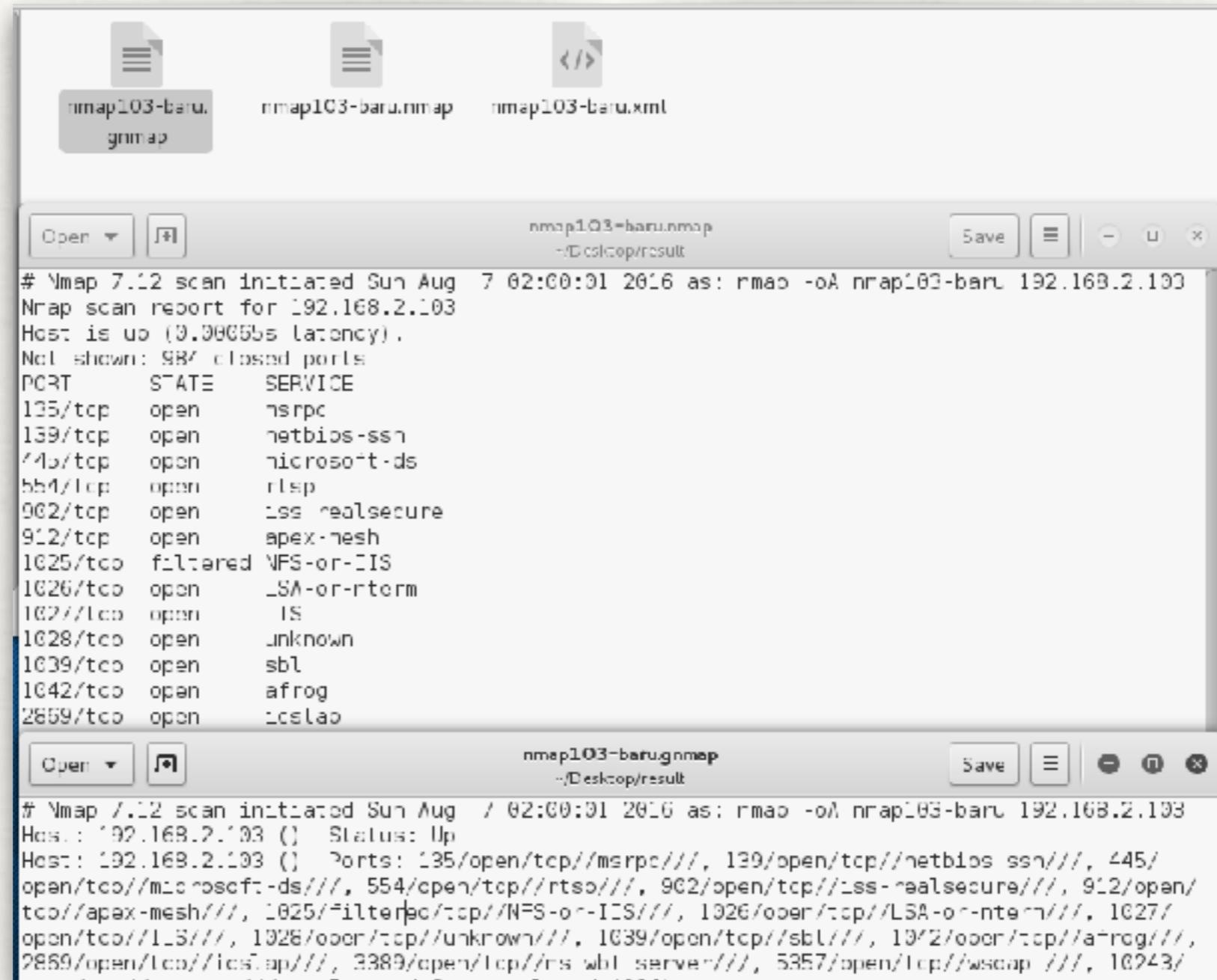
```
Starting Nmap 7.12 ( https://nmap.org ) at 2016-08-07 02:00 EDT
Nmap scan report for 192.168.2.103
Host is up (0.00065s latency).
Not shown: 984 closed ports
PORT      STATE    SERVICE
135/tcp    open     msrpc
139/tcp    open     netbios-ssn
445/tcp    open     microsoft-ds
554/tcp    open     rtsp
902/tcp    open     iss-realsecure
912/tcp    open     apex-mesh
1025/tcp   filtered NFS-or-IIS
1026/tcp   open     LSA-or-nterm
1027/tcp   open     IIS
1028/tcp   open     unknown
1039/tcp   open     sbl
1042/tcp   open     afrog
2869/tcp   open     icslap
3389/tcp   open     ms-wbt-server
5357/tcp   open     wsdapi
10243/tcp  open     unknown
MAC Address: 00:0C:29:83:ED:69 (VMware)
```

```
Nmap done: 1 IP address (1 host up) scanned in 15.41 seconds
```

```
root@kali:~/Desktop/result# █
```

# NMAP OUTPUT

## ALL OUTPUT



The screenshot shows a desktop interface with a file manager window open. The window displays three files: 'nmap103-baru.nmap' (selected), 'nmap103-baru.xml', and 'nmap103-baru.gnmap'. Below the files is a terminal-like window showing the raw Nmap scan output for host 192.168.2.103.

**nmap103-baru.nmap**

```
# Nmap 7.12 scan initiated Sun Aug 7 02:00:01 2016 as: nmap -oA nmap103-baru 192.168.2.103
Nmap scan report for 192.168.2.103
Host is up (0.00055s latency).
Not shown: 987 closed ports
PORT      STATE    SERVICE
135/tcp    open     msrpc
139/tcp    open     netbios-ssn
445/tcp    open     microsoft-ds
554/tcp    open     rtsp
902/tcp    open     lss-realsecure
912/tcp    open     apex-mesh
1025/tcp   filtered NFS-or-CIFS
1026/tcp   open     LSA-or-rterm
1027/tcp   open     IS
1028/tcp   open     unknown
1039/tcp   open     sbl
1042/tcp   open     afrog
2859/tcp   open     lcs-lao
```

**nmap103-baru.gnmap**

```
# Nmap 7.12 scan initiated Sun Aug 7 02:00:01 2016 as: nmap -oA nmap103-baru 192.168.2.103
Host: 192.168.2.103 () Status: Up
Host: 192.168.2.103 () Ports: 135/open/tcp//msrpc///, 139/open/tcp//netbios ssn///, 445/
open/tcp//microsoft-ds///, 554/open/tcp//rtsp///, 902/open/tcp//lss-realsecure///, 912/open/
tcp//apex-mesh///, 1025/filter/ec/tcp//NFS-or-CIFS///, 1026/open/tcp//LSA-or-rterm///, 1027/
open/tcp//IS///, 1028/open/tcp//unknown///, 1039/open/tcp//sbl///, 1042/open/tcp//afrog///,
2859/open/tcp//lcs-lao///, 3389/open/tcp//rs-wbt-server///, 5357/open/tcp//wssoap ///, 10243/
```

# NMAP OUTPUT

## \$CRIPT KIDDIE OUTPUT

-oS: \$crIpT kIddI3 OuTPut

Script kiddie output is like interactive output, except that it is post-processed to better suit the 'l33t HaXXorZ!

# NMAP OUTPUT

## SCRIPT KIDD13 OUTPUT

```
root@kali:~# nmap -Pn 192.168.2.103 >/root/Desktop/result/nmap103skript.txt

Starting Nmap 7.12 ( https://nmap.org ) at 2016-08-07 00:21 EDT
Nmap scan report for 192.168.2.103
Host is up (0.0011s latency).
Not shown: 984 closed ports
PORT      STATE    SERVICE
135/tcp    open     msrpc
139/tcp    open     netbios-ssn
445/tcp    open     microsoft-ds
554/tcp    open     rtsp
902/tcp    open     iss-realsecure
912/tcp    open     apex-mesh
1025/tcp   filtered NFS-or-IIS
1026/tcp   open     LSA-or-nterm
1027/tcp   open     IIS
1028/tcp   open     unknown
1029/tcp   open     ms-lsa
1030/tcp   open     iad1
2869/tcp   open     icslap
3389/tcp   open     ms-wbt-server
                  "grep.txt" selected (748 bytes)
5357/tcp   open     wsdapi
10243/tcp  open     unknown
MAC Address: 00:0C:29:83:ED:69 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 15.44 seconds
root@kali:~#
```

# NMAP OUTPUT

## \$CRIPT KIDD13 OUTPUT

```
Open ▾  nmap103skript.txt ~/Desktop/result Save   
$tart|ng nMap 7.12 ( HtTPz://NmAp.0rg ) at 2016-08-07 00:21 eDT  
NmaP scAn r3p0rt f0r 192.168.2.103  
H0$t !s up (0.0011S lAt3Ncy).  
not $h0Wn: 984 CloSed p0rTs  
P0rt ST4Te S3RV!C3  
135/tCp Open m$rpC  
139/tcp opeN neTb!0z-s$n  
445/tcp open M!croSoft -Ds  
554/tcp 0p3n rt$p  
902/tcp open i$s-reals3cur3  
912/Tcp op3n apex-M3sh  
1025/tcP filtered nFz-0r-I|s  
1026/tcp opEn LS4-or-nt3rM  
1027/tcp Open 1Iz  
1028/Tcp open unknoWn  
1029/tcp op3N mz-l$A  
1030/tcp 0pEn !ad1  
2869/tCp 0p3n Ic$lAp  
3389/Tcp op3n mz-wbT-$3rvEr  
5357/TCp 0pEn w$daP1  
10243/tcp Open unknoWn  
M4C addr3ss: 00:0c:29:83:3D:69 (VMwar3)  
Nmap d0ne: 1 iP aDdR3$s (1 hoSt up) scanned in 15.44 SEC0ndz
```

# NMAP OUTPUT

## CREATE HTML OUTPUT

- Nmap does not have an option for saving scan results in HTML, however it is possible to convert XML output to HTML automatically.
- An Nmap XML output file usually contains a reference to an XSL stylesheet called nmap.xsl that describes how the transformation takes place.

```
<?xmlstylesheet href="/usr/share/nmap/nmap.xsl" type="text/xsl"?>  
  
<?xmlstylesheet href="http://nmap.org/svn/docs/nmap.xsl"  
type="text/xsl"?>
```

- Using xslproc to generate xml output to html.

# NMAP OUTPUT

## CREATE HTML OUTPUT

```
root@kali:~# cd Desktop/result/  
root@kali:~/Desktop/result# xsltproc nmap103.xml -o nmap103.html  
root@kali:~/Desktop/result#
```

The screenshot shows a web browser window titled "Nmap Scan Report - Scanned at Sat Aug 6 10:17:37 2016 – Iceweasel". The address bar contains the URL "file:///root/Desktop/result/nmap103.html". The page content is a scan report generated by Nmap, with a dark header bar containing the title. Below the header, there is a navigation bar with links to "Most Visited", "Offensive Security", "Kali Linux", "Kali Docs", "Kali Tools", "Exploit-DB", and "Aircrack-ng".

# NMAP OUTPUT

## CREATE HTML OUTPUT

Nmap Scan Report – Scanned at Sat Aug 6 10:17:37 2016 – Iceweasel

Nmap Scan Report - Sca... +

file:///root/Desktop/result/nmap103.html ▾ C Google

Most Visited ▾ Offensive Security Kali Linux Kali Docs Kali Tools Exploit-DB Aircrack-ng

### Nmap Scan Report - Scanned at Sat Aug 6 10:17:37 2016

Scan Summary | 192.168.2.103

#### Scan Summary

Nmap 7.12 was Initiated at Sat Aug 6 10:17:37 2016 with these arguments:  
`nmap -oX /root/Desktop/result/nmap103xml.txt 192.168.2.103`

Verbosity: 0; Debug level 0

Nmap done at Sat Aug 6 10:17:52 2016; 1 IP address (1 host up) scanned in 15.77 seconds

## 192.168.2.103

#### Address

- 192.168.2.103 (Ipv4)
- 00:0C:29:83:ED:69 - VMware (mac)

Go to top

#### Ports

The 984 ports scanned but not shown below are in state: **closed**

Toggle Closed Ports

Toggle Filtered Ports

# NMAP DEBUGGING AND TROUBLESHOOTING



# NMAP DEBUGGING

## VERBOSITY

-v (Increase verbosity level) .

Increases the verbosity level, causing Nmap to print more information about the scan in progress. Open ports are shown as they are found and completion time estimates are

provided when Nmap thinks a scan will take more than a few minutes. Use it twice or more for even greater verbosity: -vv, or give a verbosity level directly, for example -v3..

# NMAP DEBUGGING VERBOSITY

```
root@kali:~/Desktop/result# nmap -v 192.168.2.103
Starting Nmap 7.12 ( https://nmap.org ) at 2016-09-07 01:43 EDT
Initiating ARP Ping Scan at 01:43
Scanning 192.168.2.103 [1 port]
Completed ARP Ping Scan at 01:43, 0.02s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host at 01:43
Completed Parallel DNS resolution of 1 host at 01:43, 0.00s elapsed
Initiating SYN Stealth Scan at 01:43
Scanning 192.168.2.103 [1000 ports]
Discovered open port 139/tcp on 192.168.2.103
Discovered open port 554/tcp on 192.168.2.103
Discovered open port 135/tcp on 192.168.2.103
Discovered open port 3389/tcp on 192.168.2.103
Discovered open port 5357/tcp on 192.168.2.103
Discovered open port 1027/tcp on 192.168.2.103
Discovered open port 1028/tcp on 192.168.2.103
Discovered open port 10243/tcp on 192.168.2.103
Discovered open port 145/tcp on 192.168.2.103
Discovered open port 912/tcp on 192.168.2.103
Discovered open port 902/tcp on 192.168.2.103
Discovered open port 1039/tcp on 192.168.2.103
Discovered open port 2869/tcp on 192.168.2.103
Discovered open port 1042/tcp on 192.168.2.103
Discovered open port 1026/tcp on 192.168.2.103
Completed SYN Stealth Scan at 01:43, 2.32s elapsed (1000 total ports)
Nmap scan report for 192.168.2.103
Host is up (0.00068s latency).      result
Not shown: 984 closed ports
PORT      STATE SERVICE
135/tcp    open  msrpc
139/tcp    open  netbios-ssn
445/tcp    open  microsoft-ds
554/tcp    open  rtsp
902/tcp    open  iss-realsecure
912/tcp    open  apex-mesh
1025/tcp   filtered  netgear-1_0_S
```

# NMAP DEBUGGING

## VERBOSITY

```
root@kali:~/Desktop/result# nmap -v3 -p 135-500 192.168.2.103

Starting Nmap 7.12 ( https://nmap.org ) at 2016-08-07 01:44 EDT
Initiating ARP Ping Scan at 01:44
Scanning 192.168.2.103 [1 port]
Completed ARP Ping Scan at 01:44, 0.01s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 01:44
Completed Parallel DNS resolution of 1 host. at 01:44, 13.01s elapsed
DNS resolution of 1 IPs took 13.01s. Mode: Async [#: 1, OK: 0, NX: 0, DR: 1, SF:
0, TR: 3, CN: 0]
Initiating SYN Stealth Scan at 01:44
Scanning 192.168.2.103 [366 ports]
Discovered open port 139/tcp on 192.168.2.103
Discovered open port 135/tcp on 192.168.2.103
Discovered open port 445/tcp on 192.168.2.103
Completed SYN Stealth Scan at 01:44, 1.11s elapsed (366 total ports)
Nmap scan report for 192.168.2.103
Host is up, received arp-response (0.00057s latency).
Scanned at 2016-08-07 01:44:15 EDT for 14s
Not shown: 363 closed ports
Reason: 363 resets
PORT      STATE SERVICE      REASON
135/tcp    open  msrpc        syn-ack ttl 128
139/tcp    open  netbios-ssn   syn-ack ttl 128
445/tcp    open  microsoft-ds  syn-ack ttl 128
MAC Address: 00:0C:29:83:ED:69 (VMware)

Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 14.20 seconds
Raw packets sent: 430 (18.904KB) | Rcvd: 367 (14.680KB)
```

# NMAP DEBUGGING

## DEBUGGING

-d (Increase debugging level) .

Debugging output is useful when a bug is suspected in Nmap, or if you are simply confused as to what Nmap is doing and why. As this feature is mostly intended for developers.

-d1-9

# NMAP DEBUGGING

## DEBUGGING

```
root@kali:~/Desktop/result# nmap -d 192.168.2.103

Starting Nmap 7.12 ( https://nmap.org ) at 2016-08-07 01:48 EDT
PORTS: Using top 1000 ports found open [TCP:1000, UDP:0, SCTP:0]
----- Timing report -----
hostgroups: min 1, max 100000
rtimeouts: init 1000, min 100, max 10000
max-scan-delay: TCP 1000, UDP 1000, SCTP 1000
parallelism: min 0, max 0
max-retries: 10, host-timeout: 0
min-rate: 0, max-rate: 0
-----
Initiating ARP Ping Scan at 01:48
Scanning 192.168.2.103 [1 port]
Packet capture filter (device eth0): arp and arp[18:4] = 0x000C296E and arp[22:2] = 0x8883
Completed ARP Ping Scan at 01:48, 0.00s elapsed (1 total hosts)
Overall sercing rates: 733.14 packets / s, 30791.79 bytes / s.
mass_rdns: Using DNS server 192.168.2.1
Initiating Parallel DNS resolution of 1 host. at 01:48
mass_rdns: 13.00s 0/1 [#: 1, CK: 0, NK: 0, DR: 0, SF: 0, TR: 0]
Completed Parallel DNS resolution of 1 host. at 01:48, 13.00s elapsed
DNS resolution of 1 IPs took 13.00s. Mode: Asyrc [#: 1, OK: 0, NX: 0, DR: 1, SF: 0, TR: 3, CN: 0]
Initiating SYN Stealth Scan at 01:48
Scanning 192.168.2.103 [1000 ports]
Packet capture filter (device eth0): dst host 192.168.2.104 and (icmp or icmp6 or (tcp or udp or sctp) and (src host 192.168.2.103)))
Discovered open port 3389/tcp on 192.168.2.103
Discovered open port 135/tcp on 192.168.2.103
Discovered open port 554/tcp on 192.168.2.103
Discovered open port 445/tcp on 192.168.2.103
Discovered open port 1026/tcp on 192.168.2.103
```

# NMAP DEBUGGING

## DEBUGGING

```
root@kali:~/Desktop/result# nmap -c9 192.168.2.103

Starting Nmap 7.12 ( https://nmap.org ) at 2016-08-07 01:48 EDT
[fetchfile found /usr/bin/.../share/nmap/rmap-services]
PORTS: Using top 1000 ports of cndcpent (TCP:1000, UDP:0, SCTP:0)
[fetchfile found /usr/bin/.../share/nmap/rmap.xls]
The max # of sockets we are using is: 0
----- Timing report -----
hostgroups: min 1, max 100000
rtt-timeouts: init 1000, min 100, max 10000
max-scan-delay: TCP 1000, UDP 1000, SCTP 1000
parallelism: min 0, max 0
max-retries: 10, host-timeout: 0
min-rate: 0, max-rate: 0
-----
[fetchfile found /usr/bin/.../share/nmap/rmap-payloads]
Initiating ARP Ping Scan at 01:48
Scanning 192.168.2.103 [1 port]
Packet capture filter [device eth0]: arp and arp[18:4] = 0x000C296E and arp[22:2]
]c = 0x8883
SENT (0.0398s) ARP who-has 192.168.2.103 tell 192.168.2.104
**TIMING STATS** (0.0400s): IP, probes active/freshports/left/retry_stack/customizing/retranswait/onbench, cwnd/ssthresh/celay, timeout/srtt/rttvar/
Groupstats (1/1 incomplete): 1/*/*/*/*/* 10.00/75/* 200000/ 1/ 1
192.168.2.103: 1/0/0/1/0/0 10.00/75/0 200000/-1/-1
Current sending rates: 836.12 packets / s, 35117.06 bytes / s.
Overall sending rates: 836.12 packets / s, 35117.06 bytes / s.
RCVD (0.0403s) ARP reply 192.168.2.103 is-at 00:0C:29:83:ED:69
Found 192.168.2.103 in incomplete hosts list.
ultrascan_host_probe_update called for machine 192.168.2.103 state UNKNOWN -> H0
SI_UP (trynum 0 time: 611)
Timeout vals: srtt: -1 rttvar: -1 to: 200000 delta 513 --> srtt: 513 rttvar: 500
0 to: 100000
Timeout vals: srtt: -1 rttvar: -1 to: 200000 delta 513 ==> srtt: 513 rttvar: 500
0 to: 100000
Changing ping technique for 192.168.2.103 to ARP
```

# NMAP DEBUGGING

## REASON

--reason (Host and port state reasons) .

Shows the reason each port is set to a specific state and the reason each host is up or down. This option displays the type of the packet that determined a port or hosts state.

For example, A RST packet from a closed port or an echo reply from an alive host. The information Nmap can provide is determined by the type of scan or ping. The SYN scan and SYN

ping (-sS and -PS) are very detailed, but the TCP connect scan (-sT) is limited by the implementation of the connect system call. This feature is automatically enabled by the debug option (-d). and the results are stored in XML log files even if this option is not specified.

# NMAP DEBUGGING REASON

```
root@kali:~/Desktop/result# nmap -reason 192.168.2.103
Starting Nmap 7.12 ( https://nmap.org ) at 2016-08-07 01:52 EDT
Nmap scan report for 192.168.2.103
Host is up, receives arp-response (0.00036s latency).
Not shown: 905 closed ports
Reason: 985 resets.txt      nmap103skript.txt      nmap103xml.txt
PORT      STATE      SERVICE      REASON
135/tcp    open       ts3cc       syn-ack ttl 128
139/tcp    open       netbios-ssr  syn-ack ttl 128
445/tcp    open       microsoft-ds syn-ack ttl 128
554/tcp    open       rtsp        syn-ack ttl 128
902/tcp    open       iss-realsecure syn-ack ttl 128
912/tcp    open       spex-nesh   syn-ack ttl 128
1025/tcp   filtered  NFS-or-IIS  no-response
1026/tcp   open       SA-or-nterm syn-ack ttl 128
1027/tcp   open       IIS         syn-ack ttl 128
1028/tcp   open       Unknown    syn-ack ttl 128
1039/tcp   open       sbc        syn-ack ttl 128
1042/tcp   open       strog      syn-ack ttl 128
2869/tcp   open       icslap     syn-ack ttl 128
5357/tcp   open       wsdaps    syn-ack ttl 128
10243/tcp  open       Unknown    syn-ack ttl 128
MAC Address: 00:0C:29:B3:ED:69 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 15.43 seconds
root@kali:~/Desktop/result#
```

# NMAP DEBUGGING

## PERIODIC RESULT

--stats-every time (Print periodic timing stats) .

Periodically prints a timing status message after each interval of time. The time is a specification of the kind described in the section called "TIMING AND PERFORMANCE"; so for example, use --stats-every 10s to get a status update every 10 seconds. Updates are printed to interactive output (the screen) and XML output.

# NMAP DEBUGGING

## PERIODIC RESULT

```
root@kali:~/Desktop/result# nmap --stats-every 5s 192.168.2.103
Starting Nmap 7.12 ( https://nmap.org ) at 2016-08-07 02:06 EDT
Stats: 0:00:05 elapsed; 0 hosts completed (0 up), 1 undergoing ARP Ping Scan
Parallel DNS resolution of 1 host. Timing: About 0.00% done
Stats: 0:00:10 elapsed; 0 hosts completed (0 up), 1 undergoing ARP Ping Scan
Parallel DNS resolution of 1 host. Timing: About 0.00% done
Stats: 0:00:15 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 98.67% done; ETC: 02:07 (0:00:00 remaining)
Nmap scan report for 192.168.2.103
Host is up (0.00094s latency).
Not shown: 984 closed ports
PORT      STATE    SERVICE
135/tcp    open     msrpc
139/tcp    open     netbios-ssn
445/tcp    open     microsoft-ds
554/tcp    open     rtsp
902/tcp    open     iss-realsecure
912/tcp    open     apex-mesh
1025/tcp   filtered NFS-or-IIS
1026/tcp   open     LSA-or-nterm
1027/tcp   open     IIS
```

# NMAP DEBUGGING

## PACKET TRACE

--packet-trace (Trace packets and data sent and received) .

Causes Nmap to print a summary of every packet sent or received. This is often used for debugging, but is also a valuable way for new users to understand exactly what Nmap is doing under the covers.

To avoid printing thousands of lines, you may want to specify a limited number of ports to scan, such as -p20-30. If you only care about the goings on of the version detection subsystem, use --version-trace instead.

If you only care about script tracing, specify --script-trace. With --packet-trace, you get all of the above.

# NMAP DEBUGGING

## PACKET TRACE

```
root@kali:~/Desktop/result# nmap --packet-trace --script-trace -p105-140 192.168.2.103

Starting Nmap 7.12 ( https://nmap.org ) at 2016-08-07 02:15 EDT
SENT (0.0581s) ARP who has 192.168.2.103 tell 192.168.2.104
RCVD (0.0586s) ARP reply 192.168.2.103 is-at 00:0C:29:83:ED:59
NSOCK< INFO [0.0660s] nsock_ioc_new(): nsock_ioc_new (IOID #1)
NSOCK< INFO [0.0670s] nsock_connect_udp(): UDP connection requested to 192.168.2.1:53 (IOID #1) EID 6
NSOCK< INFO [0.0670s] nsock_read(): Read request from IOID #1 [192.168.2.1:53] (timeout: 1s) EID 19
NSOCK< INFO [0.0670s] nsock_write(): Write request for 44 bytes to IOID #1 EID 27 [192.168.2.1:53]
NSOCK< INFO [0.0670s] nsock_trace_handler_callback(): Callback: CONNECT SUCCESS for EID 0 [192.168.2.1:53]
NSOCK< INFO [0.0670s] nsock_trace_handler_callback(): Callback: WRITE SUCCESS for EID 27 [192.168.2.1:53]
NSOCK< INFO [0.2790s] nsock_trace_handler_callback(): Callback: READ SUCCESS for EID 19 [192.168.2.1:53] (143 bytes)
NSOCK< INFO [0.2790s] nsock_read(): Read request from IOID #1 [192.168.2.1:53] (timeout: 1s) EID 34
NSOCK< INFO [0.4060s] nsock_write(): Write request for 44 bytes to IOID #1 EID 43 [192.168.2.1:53]
NSOCK< INFO [0.4060s] nsock_trace_handler_callback(): Callback: WRITE SUCCESS for EID 43 [192.168.2.1:53]
NSOCK< INFO [0.4140s] nsock_trace_handler_callback(): Callback: READ SUCCESS for EID 34 [192.168.2.1:53] (137 bytes)
NSOCK< INFO [0.4140s] nsock_read(): Read request from IOID #1 [192.168.2.1:53] (timeout: 1s) EID 53
NSOCK< INFO [0.8070s] nsock_write(): Write request for 44 bytes to IOID #1 EID 59 [192.168.2.1:53]
NSOCK< INFO [0.8070s] nsock_trace_handler_callback(): Callback: WRITE SUCCESS for EID 59 [192.168.2.1:53]
NSOCK< INFO [0.87910s] nsock_trace_handler_callback(): Callback: READ SUCCESS for EID 58 [192.168.2.1:53] (137 bytes)
NSOCK< INFO [0.87910s] nsock_read(): Read request from IOID #1 [192.168.2.1:53] (timeout: 1s) EID 66
NSOCK< INFO [0.87910s] nsock_ioc_delete(): nsock_ioc_delete (IOID #1)
NSOCK< INFO [0.87910s] nevent_delete(): nevent_delete on event #66 (type READ)
SENT (13.0729s) TCP 192.168.2.104:52279 > 192.168.2.103:135 S ttl=39 id=44830 ipLen=44 seq=3656263236 win=1024 <mss 1460>
RCVD (13.0730s) TCP 192.168.2.104:52279 > 192.168.2.103:139 S ttl=54 id=57096 ipLen=44 seq=0656263236 win=1024 <mss 1460>
SENT (13.0930s) TCP 192.168.2.104:52279 > 192.168.2.103:137 S ttl=79 id=13094 ipLen=44 seq=3656263236 win=1024 <mss 1460>
SENT (13.0732s) TCP 192.168.2.104:52279 > 192.168.2.103:130 S ttl=46 id=33004 ipLen=44 seq=0656263236 win=1024 <mss 1460>
SHNI (13.0934s) TCP 192.168.2.104:52279 > 192.168.2.103:136 S .11=71 id=40576 ipLen=44 seq=3656263236 win=1024 <mss 1460>
SENT (13.0735s) TCP 192.168.2.104:52279 > 192.168.2.103:140 S ttl=38 id=45164 ipLen=44 seq=0656263236 win=1024 <mss 1460>
RCVD (13.0743s) TCP 192.168.2.103:140 > 192.168.2.134:52279 RA ttl=128 inc=9795 ipLen=40 seq=0 win=0
SENT (14.1760s) TCP 192.168.2.104:52268 > 192.168.2.103:136 S ttl=47 id=39908 ipLen=44 seq=065625773 win=1024 <mss 1460>
SHNI (14.1762s) TCP 192.168.2.104:52283 > 192.168.2.103:138 S .11=49 id=47507 ipLen=44 seq=3656325773 win=1024 <mss 1460>
SENT (14.1764s) TCP 192.168.2.104:52268 > 192.168.2.103:137 S ttl=54 id=42152 ipLen=44 Pseq=065625773 win=1024 <mss 1460>
SEN1 (14.1766s) TCP 192.168.2.104:52283 > 192.168.2.103:139 S TTL=38 id=52216 ipLen=44 seq=3656325773 win=1024 <mss 1460>
SENT (14.1767s) TCP 192.168.2.104:52268 > 192.168.2.103:135 S TTL=54 id=42522 ipLen=44 seq=3656325773 win=1024 <mss 1460>
RCVD (14.1764s) TCP 192.168.2.103:136 > 192.168.2.134:52280 RA TTL=128 inc=9795 ipLen=40 seq=0 win=0<br>...
RCVD (14.1766s) TCP 192.168.2.103:138 > 192.168.2.131:52280 RA TTL=128 inc=9797 ipLen=40 seq=0 win=0<br>...
RCVD (14.1767s) TCP 192.168.2.103:137 > 192.168.2.134:52280 RA TTL=128 inc=9798 ipLen=40 seq=0 win=0
```

# NMAP DEBUGGING

## OTHER OPTIONS

--open (Show only open (or possibly open) ports) .

Specify --open to only see hosts with at least one open, open filtered, or unfiltered port, and only see ports in those states.

--iflist (List interfaces and routes) .

Prints the interface list and system routes as detected by Nmap. This is useful for debugging routing problems or device mischaracterization (such as Nmap treating a PPP connection as ethernet).

# NMAP DEBUGGING

## OTHER OPTIONS

```
root@kali:~/Desktop/result# nmap --open 192.168.2.103
Starting Nmap 7.12 ( https://nmap.org ) at 2016-08-07 02:23 EDT
Nmap scan report for 192.168.2.103
Host is up (0.00030s latency).
Not shown: 984 closed ports, 1 filtered port
PORT      STATE SERVICE
135/tcp    open  msrpc
139/tcp    open  netbios-ssn
445/tcp    open  microsoft-ds
554/tcp    open  rtsp
902/tcp    open  iss-realsecure
912/tcp    open  apex-mesh
1026/tcp   open  LSA-or-nterm
1027/tcp   open  IIS
1028/tcp   open  unknown
1039/tcp   open  sbl
1042/tcp   open  afrog
2869/tcp   open  icslap
3389/tcp   open  ms-wbt-server
5357/tcp   open  wsdapi
10243/tcp  open  unknown
MAC Address: 00:0C:29:83:ED:69 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 15.41 seconds
```

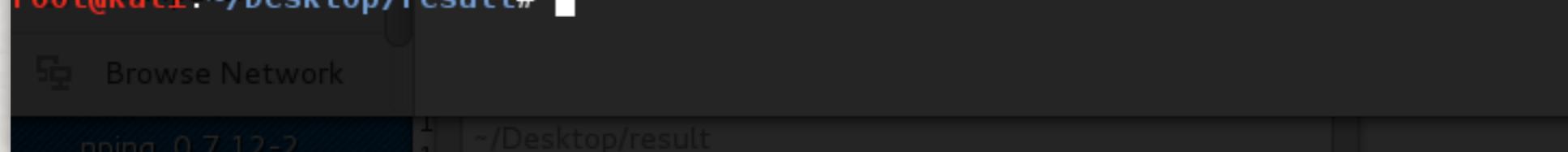
# NMAP DEBUGGING

## OTHER OPTIONS

```
root@kali:~/Desktop/result# nmap --iflist 192.168.2.103 </>
Starting Nmap 7.12 ( https://nmap.org ) at 2016-08-07 02:24 EDT
*****INTERFACES*****
gnmap
DEV (SHORT) IP/MASK          TYPE    UP MTU   MAC
eth0 (eth0) 192.168.2.104/24  ethernet up 1500  00:0C:29:6E:88:83
eth0 (eth0) fe80::20c:29ff:fe6e:8883/64  ethernet up 1500  00:0C:29:6E:88:83
lo  (lo)    127.0.0.1/8       loopback up 65536
lo  (lo)    ::1/128          loopback up 65536

*****ROUTES*****
DST/MASK          DEV METRIC GATEWAY
192.168.2.0/24   eth0 0
0.0.0.0/0        eth0 1024 192.168.2.1
::1/128          lo  0
fe80::20c:29ff:fe6e:8883/128 lo  0
::1/128          lo  256
fe80::/64         eth0 256
ff00::/8          eth0 256

root@kali:~/Desktop/result#
```



The screenshot shows the Nmap interface running in a terminal window. At the top, there's a network browser titled "Browse Network". Below it, the command history shows the execution of "nmap --iflist 192.168.2.103". The main area displays the output of the Nmap command, listing network interfaces and routes.

# NMAP MISCELLANEOUS OPTIONS



# NMAP MISCELLANEOUS OPTIONS

## IPV6 SCAN

-6 (Enable IPv6 scanning) .

Nmap has IPv6 support for its most popular features. Ping scanning, port scanning, version detection, and the Nmap Scripting Engine all support IPv6. The command syntax is the same as usual except that you also add the -6 option. Of course, you must use IPv6 syntax if you specify an address rather than a hostname.

An address might look like 3ffe:7501:4819:2000:210:f3ff:fe03:14d0, so hostnames are recommended. The output looks the same as usual, with the IPv6 address on the "interesting ports" line being the only IPv6 giveaway.

While IPv6 hasn't exactly taken the world by storm, it gets significant use in some (usually Asian) countries and most modern operating systems support it. To use Nmap with IPv6, both the source and target of your scan must be configured for IPv6.

# NMAP DEBUGGING

## IPV6 SCAN

```
root@kali:~/Desktop/result# nmap -6 -vv fe80::20c:29ff:fe6e:8883
Starting Nmap 7.12 ( https://nmap.org ) at 2016-08-07 02:47 EDT
Initiating Parallel DNS resolution of 1 host. at 02:47
Completed Parallel DNS resolution of 1 host. at 02:47, 13.00s elapsed
Initiating SYN Stealth Scan at 02:47
Scanning fe80::20c:29ff:fe6e:8883 [1000 ports]
Completed SYN Stealth Scan at 02:47, 0.01s elapsed (1000 total ports)
Nmap scan report for fe80::20c:29ff:fe6e:8883
Host is up, received localhost-response (0.0000050s latency).
All 1000 scanned ports on fe80::20c:29ff:fe6e:8883 are closed because of 1000 resets
```

```
Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 13.06 seconds
  Videos  Raw packets sent: 1000 (64.000KB) | Rcvd: 2000 (124.000KB)
```

```
root@kali:~/Desktop/result#
```



Floppy Disk



Computer

# NMAP MISCELLANEOUS OPTIONS

## AGGRESSIVE SCAN

-A (Aggressive scan options) .

This option enables additional advanced and aggressive options. Presently this enables OS detection (-O), version scanning (-sV), script scanning (-sC) and traceroute (--traceroute).. More features may be added in the future. The point is to enable a comprehensive set of scan options without people having to remember a large set of flags.

However, because script scanning with the default set is considered intrusive, you should not use -A against target networks without permission. This option only enables features, and not timing options (such as -T4) or verbosity options (-v) that you might want as well. Options which require privileges (e.g. root access) such as OS detection and traceroute will only be enabled if those privileges are available.

# NMAP DEBUGGING

## AGGRESSIVE SCAN

```
root@kali:~/Desktop/result# nmap -A 192.168.2.103 |nmap --nmap103-format
```

Starting Nmap 7.02 ( https://nmap.org ) at 2016-08-07 02:50 EDT  
Nmap scan report for 192.168.2.103  
Host is up (0.0003/6s latency).  
Not shown: 984 closed ports

PORT	STATE	SERVICE	VERSION
135/tcp	open	msrpc	* Microsoft Windows RPC
139/tcp	open	netbios-ssn	Microsoft Windows 98 netbios-ssn
445/tcp	open	microsoft-ds	Microsoft Windows 13 microsoft-ds
554/tcp	open	rtsp	
902/tcp	open	ssl/vmware-auth	VMware Authentication Daemon 1.1.0 (Uses VNC, SOAP)
912/tcp	open	vmware-auth	VMware Authentication Daemon 1.0 (Uses VNC, SOAP)
1025/tcp	filtered	NFS or IIS	
1026/tcp	open	msrpc	Microsoft Windows RPC
1027/tcp	open	msrpc	Microsoft Windows RPC
1028/tcp	open	msrpc	Microsoft Windows RPC
1039/tcp	open	msrpc	Microsoft Windows RPC
1042/tcp	open	msrpc	Microsoft Windows RPC
2869/tcp	open	http	Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
3389/tcp[open filtered]	tcpwrapped		

|\_ssl-cert: Subject: commonName=WIN-REINUERD0NW  
|\_Not valid before: 2016-04-10T15:11:00  
|\_Not valid after: 2016-10-13T15:11:00  
|\_ssl-date: 2016-08-07 00:52:45+00:00; 0s from scanner time.  
5057/tcp open http -/Desktop/result Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)  
|\_http-server-header: Microsoft HTTPAPI httpd 2.0  
|\_http-title: Service unavailable Other Documents  
10243/tcp open http Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)  
|\_http-server-header: Microsoft HTTPAPI httpd 2.0  
|\_http-title: Not Found  
MAC Address: 00:0C:29:83:ED:69 (VMware)  
Device type: general purpose  
Running: Microsoft Windows 7 Pro 6.1.7601 SP1  
OS CPE: cpe:/o:microsoft:windows-7::cpe:/o:microsoft:windows-7::sc1:cpe:/o:microsoft:windows-server-2008-sp1:cpe:/o:microsoft:windows-8:cpe:/o:microsoft:windows-8.1/ Ignored State: closed (984)

# NMAP MISCELLANEOUS OPTIONS

## OTHER OPTIONS

**-V; --version** (Print version number) .

Prints the Nmap version number and exits.

**-h; --help** (Print help summary page) .

Prints a short help screen with the most common command flags. Running Nmap without any arguments does the same thing.

**--resume filename** (Resume aborted scan) .

Cancel NMAP by pressing ctrl-C. Restarting the whole scan from the beginning may be undesirable. Fortunately, if normal (-oN) or grepable (-oG) logs were kept, the user can ask Nmap to resume scanning with the target it was working on when execution ceased. Simply specify the --resume option and pass the normal/grepable output file as its argument. No other arguments are permitted, as Nmap parses the output file to use the same ones specified previously.



# REFERENCE

- NMAP Manual
- NMAP Online book (free content) - <https://nmap.org/book/>