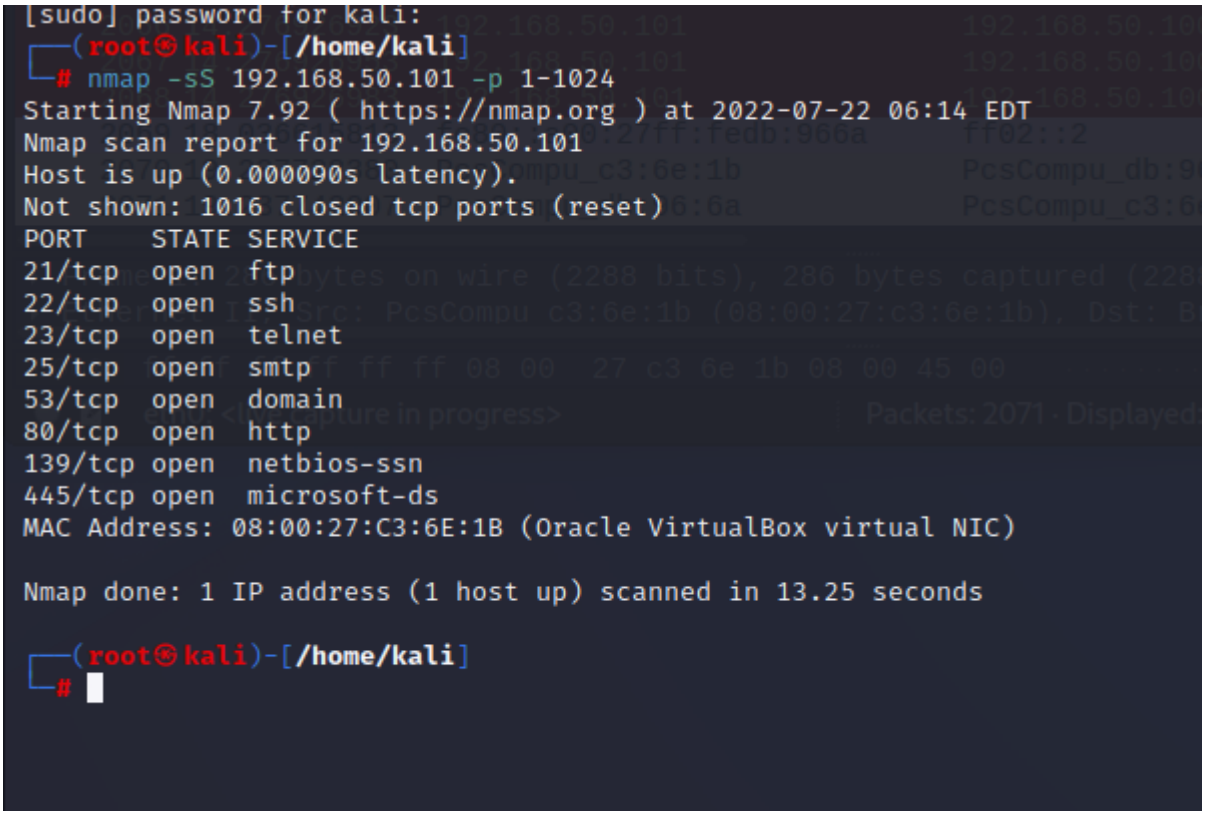
NMAP Federico Martella

Nmap è un software di port scanning, serve per individuare le porte aperte della macchina bersaglio. Oggi mostreremo una simulazione fra la macchina che scansiona Kali con indirizzo IP 192.168.50.100 e la macchina scansionata Metasploit con indirizzo IP 192.168.50.101 e andremo a prendere delle immagini da NMAP e WIRESHARK.
Vorrei dire due parole in breve su Wireshark per far comprendere l'utilizzo del software:
Wireshark è un software di analisi dei protocolli o packet sniffer che ci farà vedere più approfonditamente quali richieste fa NMAP alla macchina scansionata.

Nmap in Kali si usa dal prompt dei comandi è preinstallato su Kali.
La prima operazione da effettuare è entrare nel root di kali.
Ora analizzeremo i seguenti comandi:

"nmap -sS IP -p 1-1024" ,"nmap -sT IP -p 1-1024" e "nmap -A IP -p 1-1024".

Partiamo dal primo.
Con il comando "nmap -sS IP -p 1-1024" andiamo ad effettuare una scansione stealth molto meno invasiva rispetto al seconda e alla terza che andremo poi a fare.Il comando da utilizzare è -sS (SYN connect), sulla macchina bersaglio mettendo l'indirizzo IP della macchina. Le scansioni sono dirette alle porte 1-1024 (con il comando -p) cioè le porte più comuni.
La risposta sul prompt comandi sarà:



L’ immagine ci mostra le porte aperte sulla macchina bersaglio.

I servizi principali che andremo a spiegare sono FTP SSH TELNET SMTP DOMAIN HTTP NETBIOS:

FTP porta 21: File transfer protocol.
E’ il protocollo di livello applicativo per la trasmissione di file tra client e server.

SSH porta 22: Mette in comunicazione due host in maniera cifrata al contrario di Telnet che svolge lo stesso compito ma non criptando le informazioni. Ad esempio su due macchine con SSH/TELNET posso dare la possibilità ad uno di controllare l’altro.

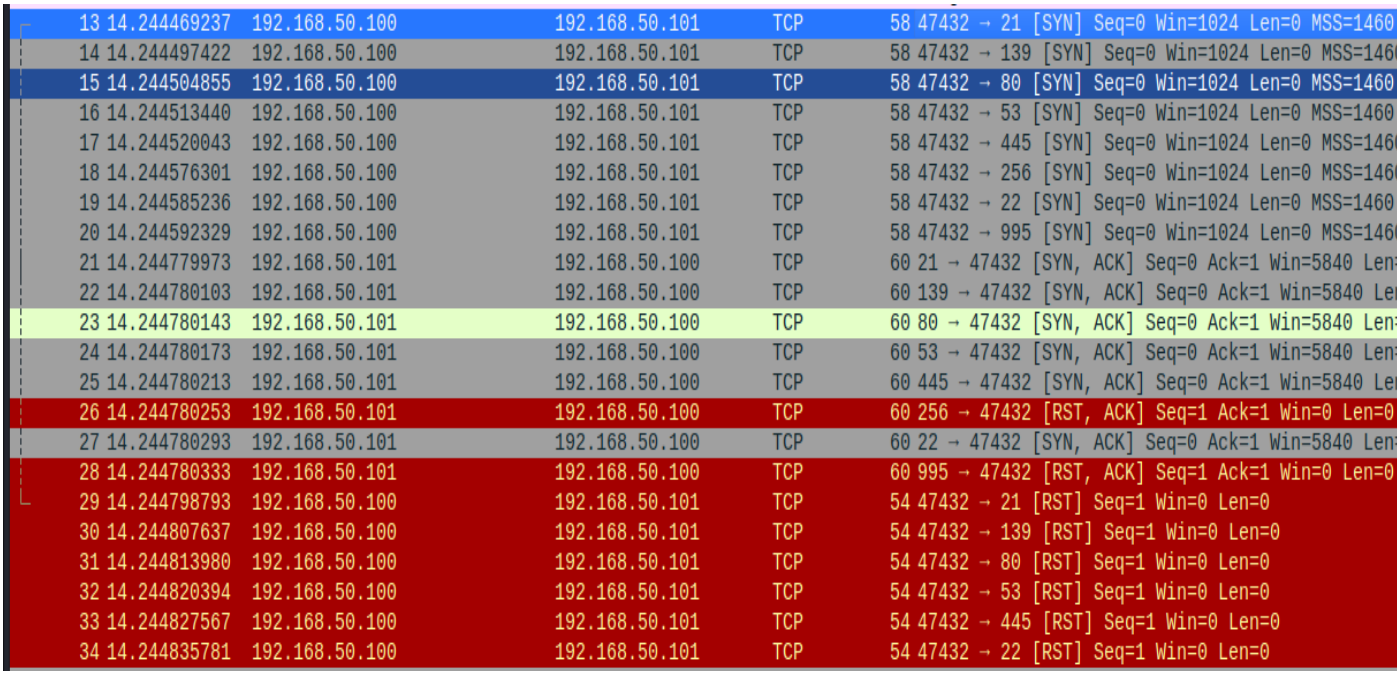
SMTP porta 25 : Simple mail transfer protocol, e’ il protocollo che ti fa mandare la mail al server mail. Protocollo tra client e server o tra server e server. Da non confondere con imap e pop3 che sono i protocolli per le mail quando arrivano al destinatario noi le mandiamo dalla nostra macchina prima al server con il protocollo SMPT sarà lui poi a spedirle al destinatario.

DOMAIN porta 53: dns serve per tradurre gli indirizzi di dominio in indirizzi IP, sui nostri computer solitamente il DNS e preimpostato dal gestore che abbiamo a casa ad esempio tim,wind,vodafone, ma si può anche sostituire ad esempio con il DNS di google 8.8.8.8

HTTP porta 80: Protocollo per la navigazione in internet non cifrata ora è meno usa perchè un po come telnet è nata la crittografia ed è più facile vedere scritto sulla barra di ricerca HTTPS che è lo stesso protocollo ma criptato.

NETBIOS-SSN porta 139: Protocollo per condividere le informazioni e le risorse in rete locale.

Adesso andiamo a vedere con il software wireshark come il comando dato su NMAP lavora sulla macchina bersaglio.



13	14.244469237	192.168.50.100	192.168.50.101	TCP	58 47432 → 21 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
14	14.244497422	192.168.50.100	192.168.50.101	TCP	58 47432 → 139 [SYN] Seq=0 Win=1024 Len=0 MSS=146
15	14.244504855	192.168.50.100	192.168.50.101	TCP	58 47432 → 80 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
16	14.244513440	192.168.50.100	192.168.50.101	TCP	58 47432 → 53 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
17	14.244520043	192.168.50.100	192.168.50.101	TCP	58 47432 → 445 [SYN] Seq=0 Win=1024 Len=0 MSS=146
18	14.244576301	192.168.50.100	192.168.50.101	TCP	58 47432 → 256 [SYN] Seq=0 Win=1024 Len=0 MSS=146
19	14.244585236	192.168.50.100	192.168.50.101	TCP	58 47432 → 22 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
20	14.244592329	192.168.50.100	192.168.50.101	TCP	58 47432 → 995 [SYN] Seq=0 Win=1024 Len=0 MSS=146
21	14.244779973	192.168.50.101	192.168.50.100	TCP	60 21 → 47432 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len
22	14.244780103	192.168.50.101	192.168.50.100	TCP	60 139 → 47432 [SYN, ACK] Seq=0 Ack=1 Win=5840 Le
23	14.244780143	192.168.50.101	192.168.50.100	TCP	60 80 → 47432 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len
24	14.244780173	192.168.50.101	192.168.50.100	TCP	60 53 → 47432 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len
25	14.244780213	192.168.50.101	192.168.50.100	TCP	60 445 → 47432 [SYN, ACK] Seq=0 Ack=1 Win=5840 Le
26	14.244780253	192.168.50.101	192.168.50.100	TCP	60 256 → 47432 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
27	14.244780293	192.168.50.101	192.168.50.100	TCP	60 22 → 47432 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len
28	14.244780333	192.168.50.101	192.168.50.100	TCP	60 995 → 47432 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
29	14.244798793	192.168.50.100	192.168.50.101	TCP	54 47432 → 21 [RST] Seq=1 Win=0 Len=0
30	14.244807637	192.168.50.100	192.168.50.101	TCP	54 47432 → 139 [RST] Seq=1 Win=0 Len=0
31	14.244813980	192.168.50.100	192.168.50.101	TCP	54 47432 → 80 [RST] Seq=1 Win=0 Len=0
32	14.244820394	192.168.50.100	192.168.50.101	TCP	54 47432 → 53 [RST] Seq=1 Win=0 Len=0
33	14.244827567	192.168.50.100	192.168.50.101	TCP	54 47432 → 445 [RST] Seq=1 Win=0 Len=0
34	14.244835781	192.168.50.100	192.168.50.101	TCP	54 47432 → 22 [RST] Seq=1 Win=0 Len=0

Essendo una scansione stealth il software NMAP manda un syn senza completare la three-way handshake (stretta di mano a tre vie del protocollo TCP).
Come possiamo vedere le porte aperte rispondo con un SYN ACK (ad esempio riga 27).
Una volta ricevuto il syn ack non c'è la risposta su wireshark da parte della macchina che sta effettuando la scansione questo perchè essendo una scansione stealth non vogliamo la chiusura del three-way handshake o più semplicemente della connessione.
La risposta della macchina che scansiona sarà [RST] reset che conclude la trasmissione.

Andiamo a vedere il prossimo comando:

”nmap -sT IP -p 1-1024”

A differenza del primo comando con -sT (TCP, connect) andiamo a effettuare il three-way handshake sulle porte che sono aperte sulla macchina scansionata. NMAP sul prompt comandi non darà nessuna differenza con il comando precedente come possiamo vedere in
figura:

```
┌──(root㉿kali)-[/home/kali]
└─# nmap -sT 192.168.50.101 -p 1-1024
Starting Nmap 7.92 ( https://nmap.org ) at 2022-07-22 06:18 EDT
Nmap scan report for 192.168.50.101
Host is up (0.00023s latency).
Not shown: 1016 closed tcp ports (conn-refused)
PORT    STATE SERVICE
21/tcp  open  ftp
22/tcp  open  ssh
23/tcp  open  telnet
25/tcp  open  smtp
53/tcp  open  domain
80/tcp  open  http
139/tcp open  netbios-ssn
445/tcp open  microsoft-ds
MAC Address: 08:00:27:C3:6E:1B (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 13.15 seconds

┌──(root㉿kali)-[/home/kali]
└─#
```

Possiamo notare che non c'è alcuna differenza fra i due comandi vedendo esclusivamente il prompt.
La differenza si nota su wireshark infatti le porte attive completano il three-way handshake come possiamo vedere in figura:

28	21.862966475	192.168.50.100	192.168.50.101	TCP	74 33104 → 143 [SYN] Seq=0 Win=64240 Len=0 MS
29	21.863003454	192.168.50.100	192.168.50.101	TCP	74 38680 → 25 [SYN] Seq=0 Win=64240 Len=0 MSS
30	21.863018591	192.168.50.100	192.168.50.101	TCP	74 44568 → 199 [SYN] Seq=0 Win=64240 Len=0 MS
31	21.863031708	192.168.50.100	192.168.50.101	TCP	74 50930 → 587 [SYN] Seq=0 Win=64240 Len=0 MS
32	21.863047266	192.168.50.100	192.168.50.101	TCP	74 34886 → 21 [SYN] Seq=0 Win=64240 Len=0 MSS
33	21.863061533	192.168.50.100	192.168.50.101	TCP	74 35144 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS
34	21.863077451	192.168.50.100	192.168.50.101	TCP	74 38908 → 111 [SYN] Seq=0 Win=64240 Len=0 MS
35	21.863098291	192.168.50.100	192.168.50.101	TCP	74 51334 → 443 [SYN] Seq=0 Win=64240 Len=0 MS
36	21.863124594	192.168.50.100	192.168.50.101	TCP	74 50210 → 445 [SYN] Seq=0 Win=64240 Len=0 MS
37	21.863139989	192.168.50.101	192.168.50.100	TCP	60 143 → 33104 [RST, ACK] Seq=1 Ack=1 Win=0
38	21.863140329	192.168.50.101	192.168.50.100	TCP	74 25 → 38680 [SYN, ACK] Seq=0 Ack=1 Win=579
39	21.863140379	192.168.50.101	192.168.50.100	TCP	60 199 → 44568 [RST, ACK] Seq=1 Ack=1 Win=0
40	21.863154319	192.168.50.100	192.168.50.101	TCP	74 54214 → 135 [SYN] Seq=0 Win=64240 Len=0 MS
41	21.863179429	192.168.50.100	192.168.50.101	TCP	66 38680 → 25 [ACK] Seq=1 Ack=1 Win=64256 Le

Come possiamo notare per la porta 25, in questo caso la porta del SMTP, wireshark ci fa vedere che il [SYN] [SYN,ACK] [ACK] è stato completato con successo.

Considerazione e differenze fra i due comandi:

-sS e -sT svolgono lo stesso compito ma in maniera differente il primo dei due comandi è molto più utilizzato essendo meno invasivo in quanto può bypassare l' IDS(intrusion detection system).
Un altra differenza è la velocità di esecuzione di -sS rispetto a -sT .
Il comando -sS necessita dei permessi di root mentre il comando -sT no.

Andiamo ora ad analizzare il terzo ed ultimo comando:

"nmap -A IP -p 1-1024":

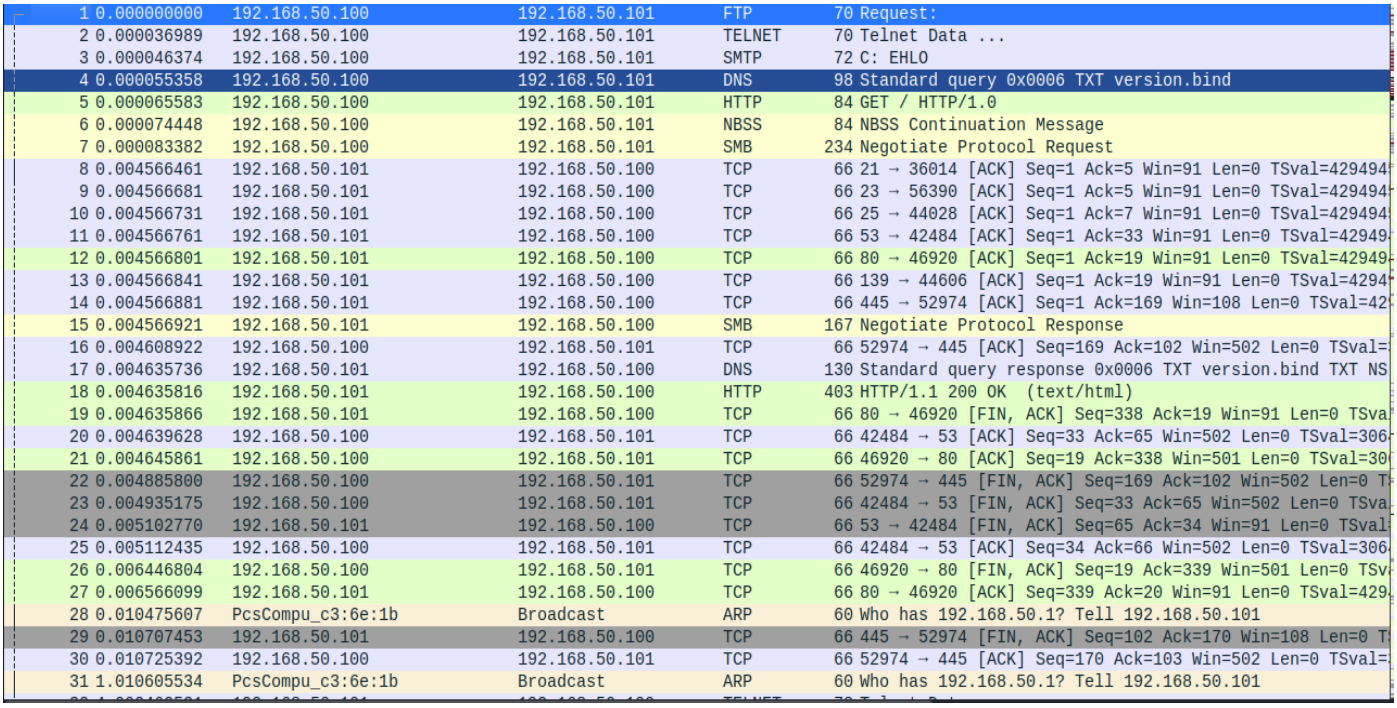
Questo comando provvede a fornire informazioni sul target come reverse DNS , OS (sistema operativo), tipo di device e mac address il -A (aggressive mode).
Come possiamo vedere nella figura sotto:

```
┌──(root㉿kali)-[/home/kali]
└─# nmap -A 192.168.50.101 -p 1-1024
Starting Nmap 7.92 ( https://nmap.org ) at 2022-07-22 06:31 EDT
Nmap scan report for 192.168.50.101
Host is up (0.00020s latency).
Not shown: 1016 closed tcp ports (reset)
PORT    STATE SERVICE     VERSION
21/tcp  open  ftp         ProFTPD 1.3.1
22/tcp  open  ssh         OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
| ssh-hostkey:
|   1024 60:0f:cf:e1:c0:5f:6a:74:d6:90:24:fa:c4:d5:6c:cd (DSA)
|_  2048 56:56:24:0f:21:1d:de:a7:2b:ae:61:b1:24:3d:e8:f3 (RSA)
23/tcp  open  telnet      Linux telnetd
25/tcp  open  smtp        Postfix smtpd
|_smtp-commands: metasploitable.localdomain, PIPELINING, SIZE 10240000, VRFY, ETRN, STARTTLS, ENHANCEDSTATUSCODES, 8BITMIME, DSN
53/tcp  open  domain      ISC BIND 9.4.2
| dns-nsid:
|_  bind.version: 9.4.2
80/tcp  open  http        Apache httpd 2.2.8 ((Ubuntu) PHP/5.2.4-2ubuntu5.10 with Suhosin-Patch)
|_http-server-header: Apache/2.2.8 (Ubuntu) PHP/5.2.4-2ubuntu5.10 with Suhosin-Patch
| http-methods:
|_  Potentially risky methods: TRACE
|_http-title: Site doesn't have a title (text/html).
139/tcp open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp open  netbios-ssn Samba smbd 3.0.20-Debian (workgroup: WORKGROUP)
MAC Address: 08:00:27:C3:6E:1B (Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6
OS details: Linux 2.6.9 - 2.6.33
Network Distance: 1 hop
Service Info: Host:  metasploitable.localdomain; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Host script results:
|_clock-skew: mean: 1h59m59s, deviation: 2h49m42s, median: 0s
|_smb2-time: Protocol negotiation failed (SMB2)
| smb-security-mode:
|   account_used: <blank>
|   authentication_level: user
|   challenge_response: supported
|_  message_signing: disabled (dangerous, but default)
|_nbstat: NetBIOS name: METASPLOITABLE, NetBIOS user: <unknown>, NetBIOS MAC: <unknown> (unknown)
| smb-os-discovery:
|   OS: Unix (Samba 3.0.20-Debian)
|   Computer name: metasploitable
|   NetBIOS computer name:
|   Domain name: localdomain
|   FQDN: metasploitable.localdomain
|_  System time: 2022-07-22T06:31:46-04:00

TRACEROUTE
HOP RTT     ADDRESS
1   0.20 ms 192.168.50.101
```

La figura mostra appunto non solo i comandi descritti sopra ma anche le versioni dei protocolli delle porte che sono aperte.
Abbiamo detto che il comando è molto più aggressivo andiamo a vedere in figura cosa accade su wireshark:



1	0.000000000	192.168.50.100	192.168.50.101	FTP	70	Request:
2	0.000036989	192.168.50.100	192.168.50.101	TELNET	70	Telnet Data ...
3	0.000046374	192.168.50.100	192.168.50.101	SMTP	72	C: EHLO
4	0.000055358	192.168.50.100	192.168.50.101	DNS	98	Standard query 0x0006 TXT version.bind
5	0.000065583	192.168.50.100	192.168.50.101	HTTP	84	GET / HTTP/1.0
6	0.000074448	192.168.50.100	192.168.50.101	NBSS	84	NBSS Continuation Message
7	0.000083382	192.168.50.100	192.168.50.101	SMB	234	Negotiate Protocol Request
8	0.004566461	192.168.50.101	192.168.50.100	TCP	66	21 → 36014 [ACK] Seq=1 Ack=5 Win=91 Len=0 TSval=429494
9	0.004566681	192.168.50.101	192.168.50.100	TCP	66	23 → 56390 [ACK] Seq=1 Ack=5 Win=91 Len=0 TSval=429494
10	0.004566731	192.168.50.101	192.168.50.100	TCP	66	25 → 44028 [ACK] Seq=1 Ack=7 Win=91 Len=0 TSval=429494
11	0.004566761	192.168.50.101	192.168.50.100	TCP	66	53 → 42484 [ACK] Seq=1 Ack=33 Win=91 Len=0 TSval=42949
12	0.004566801	192.168.50.101	192.168.50.100	TCP	66	80 → 46920 [ACK] Seq=1 Ack=19 Win=91 Len=0 TSval=42949
13	0.004566841	192.168.50.101	192.168.50.100	TCP	66	139 → 44606 [ACK] Seq=1 Ack=19 Win=91 Len=0 TSval=4294
14	0.004566881	192.168.50.101	192.168.50.100	TCP	66	445 → 52974 [ACK] Seq=1 Ack=169 Win=108 Len=0 TSval=42
15	0.004566921	192.168.50.101	192.168.50.100	SMB	167	Negotiate Protocol Response
16	0.004608922	192.168.50.100	192.168.50.101	TCP	66	52974 → 445 [ACK] Seq=169 Ack=102 Win=502 Len=0 TSval=
17	0.004635736	192.168.50.101	192.168.50.100	DNS	130	Standard query response 0x0006 TXT version.bind TXT NS
18	0.004635816	192.168.50.101	192.168.50.100	HTTP	403	HTTP/1.1 200 OK (text/html)
19	0.004635866	192.168.50.101	192.168.50.100	TCP	66	80 → 46920 [FIN, ACK] Seq=338 Ack=19 Win=91 Len=0 TSva
20	0.004639628	192.168.50.100	192.168.50.101	TCP	66	42484 → 53 [ACK] Seq=33 Ack=65 Win=502 Len=0 TSval=306
21	0.004645861	192.168.50.100	192.168.50.101	TCP	66	46920 → 80 [ACK] Seq=19 Ack=338 Win=501 Len=0 TSval=30
22	0.004885800	192.168.50.100	192.168.50.101	TCP	66	52974 → 445 [FIN, ACK] Seq=169 Ack=102 Win=502 Len=0 T
23	0.004935175	192.168.50.100	192.168.50.101	TCP	66	42484 → 53 [FIN, ACK] Seq=33 Ack=65 Win=502 Len=0 TSva
24	0.005102770	192.168.50.101	192.168.50.100	TCP	66	53 → 42484 [FIN, ACK] Seq=65 Ack=34 Win=91 Len=0 TSval
25	0.005112435	192.168.50.100	192.168.50.101	TCP	66	42484 → 53 [ACK] Seq=34 Ack=66 Win=502 Len=0 TSval=306
26	0.006446804	192.168.50.100	192.168.50.101	TCP	66	46920 → 80 [FIN, ACK] Seq=19 Ack=339 Win=501 Len=0 TSv
27	0.006566099	192.168.50.101	192.168.50.100	TCP	66	80 → 46920 [ACK] Seq=339 Ack=20 Win=91 Len=0 TSval=429
28	0.010475607	PcsCompu_c3:6e:1b	Broadcast	ARP	60	Who has 192.168.50.1? Tell 192.168.50.101
29	0.010707453	192.168.50.101	192.168.50.100	TCP	66	445 → 52974 [FIN, ACK] Seq=102 Ack=170 Win=108 Len=0 T
30	0.010725392	192.168.50.100	192.168.50.101	TCP	66	52974 → 445 [ACK] Seq=170 Ack=103 Win=502 Len=0 TSval=
31	1.010605534	PcsCompu_c3:6e:1b	Broadcast	ARP	60	Who has 192.168.50.1? Tell 192.168.50.101

Nella figura sopra possiamo vedere come il comando non si ferma solo al TCP, ma invia anche un pacchetto e si fa rispondere dalla macchina scansionata, le richieste sono molteplici ne ho isolata una per portarla alla vostra attenzione.

Ho isolato solo la porta 80 con il comando
“nmap -A IP -P 80”:

Da notare come il comando è identico cambia solamente il numero di porte scansionate infatti una volta che sappiamo quali porte sono aperte per andare ad avere dei risultati più puliti o guardare solo le porte che ci interessano possiamo cambiare il comando da -p 1-1024 ad una porta nello specifico, io l’ho fatto solo per dimostrare come agisce in questo caso NMAP e WIRESHARK.

Come prima immagine abbiamo NMAP con solo la porta 80:

```
┌──(root㉿kali)-[/home/kali]
└─# nmap -A 192.168.50.101 -p 80
Starting Nmap 7.92 ( https://nmap.org ) at 2022-07-22 08:42 EDT
Nmap scan report for 192.168.50.101
Host is up (0.00024s latency).

PORT   STATE SERVICE VERSION
80/tcp open  http    Apache httpd 2.2.8 ((Ubuntu) PHP/5.2.4-2ubuntu5.10 with Suhosi
n-Patch)
| http-methods:
|_  Potentially risky methods: TRACE
|_http-title: Site doesn't have a title (text/html).
|_http-server-header: Apache/2.2.8 (Ubuntu) PHP/5.2.4-2ubuntu5.10 with Suhosin-Patc
h
MAC Address: 08:00:27:C3:6E:1B (Oracle VirtualBox virtual NIC)
Warning: OSScan results may be unreliable because we could not find at least 1 open
 and 1 closed port
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6
OS details: Linux 2.6.9 - 2.6.33
Network Distance: 1 hop

TRACEROUTE
HOP RTT     ADDRESS
1   0.24 ms 192.168.50.101

OS and Service detection performed. Please report any incorrect results at https://
nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 21.81 seconds
```

Possiamo notare come NMAP fa la scansione solo ed esclusivamente della porta 80, ma comunque ci porta il mac address e il sistema operativo (OS).

Ora andiamo a vedere come si comporta NMAP su wireshark quando scansiona una sola porta.

Come possiamo notare il comando esegue il three handshake [SYN] RIGA 14 [SYN,ACK] RIGA 18, [ACK] RIGA 19.
Ma non si ferma a questo punto invia un pacchetto alla macchina scansionata(in questo caso essendo protocollo HTTP abbiamo visto che la prima richiesta è sempre un GET), la macchina che subisce la scansione risponde al pacchetto RIGA 26.
Ecco perché il comando è più aggressivo ed infatti i risultati di questo tipo di comando non sono solo le porte aperte, ma come detto in precedenza anche tutto quello che riguarda la versione il modello l'utilizzatore della macchina.