

# Progetto della settimana su nmap

La differenza principale tra le due scansioni è che il metodo SYN ci mostra solo se le porte sono aperte senza creare rumore e dunque di difficile rintracciamento, invece la scansione TCP essendo completa cioè che esegue il 3 way handshake crea più rumore ma crea anche un canale di comunicazione.

Nel primo scan che andremo a fare useremo il metodo

-sS: questo metodo è meno invasivo rispetto ad sT. Con questo metodo Nmap non completa il 3-way-handshake, ma chiude la comunicazione inviando un pacchetto RST(reset). Tuttavia, riesce a recuperare informazioni sullo stato della porta. Utile in quanto genera meno “rumore” a livello di rete.

The image shows a Kali Linux terminal window on the left and a Wireshark network traffic capture on the right. The terminal window displays the output of an Nmap SYN scan (-sS) performed on the IP address 192.168.50.101. The output shows that all 1024 TCP ports are open. The Wireshark window shows the captured network traffic, including the SYN packets sent by Nmap and the corresponding RST (reset) packets received from the target IP address. The Wireshark interface includes a packet list, packet details, and packet bytes panes.

No.	Time	Source	Destination	Protocol	Length	Info
3	0.072241042	192.168.50.100	192.168.50.101	TCP	58	40207 → 995 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
4	0.072273284	192.168.50.100	192.168.50.101	TCP	58	40207 → 199 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
5	0.072278414	192.168.50.100	192.168.50.101	TCP	58	40207 → 135 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
6	0.072284183	192.168.50.100	192.168.50.101	TCP	58	40207 → 587 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
7	0.072289089	192.168.50.100	192.168.50.101	TCP	58	40207 → 80 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
8	0.072294018	192.168.50.100	192.168.50.101	TCP	58	40207 → 993 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
9	0.072300838	192.168.50.100	192.168.50.101	TCP	58	40207 → 113 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
10	0.072306461	192.168.50.100	192.168.50.101	TCP	58	40207 → 443 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
11	0.072312965	192.168.50.100	192.168.50.101	TCP	58	40207 → 53 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
12	0.072321097	192.168.50.100	192.168.50.101	TCP	58	40207 → 23 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
13	0.072604862	192.168.50.101	192.168.50.100	TCP	60	995 → 40207 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
14	0.072604981	192.168.50.101	192.168.50.100	TCP	60	199 → 40207 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
15	0.072605021	192.168.50.101	192.168.50.100	TCP	60	135 → 40207 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
16	0.072605062	192.168.50.101	192.168.50.100	TCP	60	587 → 40207 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
17	0.072605102	192.168.50.101	192.168.50.100	TCP	60	80 → 40207 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0
18	0.072605142	192.168.50.101	192.168.50.100	TCP	60	993 → 40207 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
19	0.072605183	192.168.50.101	192.168.50.100	TCP	60	113 → 40207 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
20	0.072605223	192.168.50.101	192.168.50.100	TCP	60	443 → 40207 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
21	0.072638027	192.168.50.100	192.168.50.101	TCP	54	40207 → 80 [RST] Seq=1 Win=0 Len=0
22	0.072661058	192.168.50.101	192.168.50.100	TCP	60	53 → 40207 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0
23	0.072661128	192.168.50.101	192.168.50.100	TCP	60	23 → 40207 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0
24	0.072665186	192.168.50.100	192.168.50.101	TCP	54	40207 → 53 [RST] Seq=1 Win=0 Len=0
25	0.072671169	192.168.50.100	192.168.50.101	TCP	54	40207 → 23 [RST] Seq=1 Win=0 Len=0

A sinistra dopo aver scritto il comando nmap, osserviamo le porte e il loro stato(in questo caso sono tutte aperte).

Una breve descrizione dei servizi:

- **ftp**: la sigla sta per “file transfer protocol”. Indica il protocollo che si usa per trasferire i file verso un server e viceversa, uno dei protocolli più usati sono HTTP e HTTPS che usiamo per richiamare i siti web.
- **Ssh**: è l’acronimo di Secure Socket Shell, definizione usata per indicare un protocollo che fornisce agli amministratori di rete un modo sicuro per accedere a un computer remoto.
- **Telnet**: è un protocollo di rete, utilizzato tramite interfaccia a riga di comando per fornire all'utente sessioni di login remoto.
- **Smtip**: “protocollo semplice di trasferimento di posta”. L’SMTIP è specificamente responsabile dell'invio e dell'inoltro di e-mail da un mittente a un destinatario.
- **Domain**: associa il nome di un dominio ad un indirizzo IP.
- **http**: è un linguaggio di testo che consente la comunicazione tra client e server attraverso internet.
- **Netbioss-ssn**: è l'acronimo di Network basic input output system ed è utilizzato in Windows per la condivisione di file e stampanti.
- **Exect**: sostituisce l'intero contenuto corrente del processo con un nuovo programma.
- **Login**: è la procedura di identificazione che permette di essere riconosciuti da un’applicazione.

- **Shell:** La shell è l'interprete dei comandi di un sistema operativo.

**-sT**: il seguente scan che facciamo è il metodo più invasivo perché completa il passaggio 3way handshake, cioè controlla se una port è aperta o meno e recupera informazioni sul servizio in ascolto stabilendo un canale.

The image shows a Kali Linux terminal window on the left and a Wireshark network capture window on the right. The terminal window displays the execution of an Nmap SYN scan on the target IP 192.168.50.101. The output shows that 1012 TCP ports are closed and 23 are open, including services like ftp, ssh, telnet, smtp, domain, http, rpcbind, netbios-ssn, microsoft-ds, exec, login, and shell. The Wireshark window shows the captured network traffic, including ARP requests and SYN packets from the scanner to the target. The packet list on the right shows a series of SYN packets (No. 3 to 23) and their corresponding responses (ACK, RST, or SYN/ACK). The packet details pane shows the structure of the Ethernet II, Internet Protocol, and Transmission Control Protocol (TCP) packets. The packet bytes pane shows the raw data of the captured packets.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	PcsCompu_db:96:6a	Broadcast	ARP	42	Who has 192.168.50.101? Tell 192.168.50.100
2	0.000410345	PcsCompu_c6:2e:25	PcsCompu_db:96:6a	ARP	60	192.168.50.101 is at 08:00:27:c6:2e:25
3	0.036072861	192.168.50.100	192.168.50.101	TCP	74	34790 → 110 [SYN] Seq=0 Win=64240 Len=0 MSS=1460
4	0.036101677	192.168.50.100	192.168.50.101	TCP	74	51096 → 445 [SYN] Seq=0 Win=64240 Len=0 MSS=1460
5	0.03611759	192.168.50.100	192.168.50.101	TCP	74	60488 → 21 [SYN] Seq=0 Win=64240 Len=0 MSS=1460
6	0.036121322	192.168.50.100	192.168.50.101	TCP	74	59064 → 22 [SYN] Seq=0 Win=64240 Len=0 MSS=1460
7	0.036130335	192.168.50.100	192.168.50.101	TCP	74	60096 → 256 [SYN] Seq=0 Win=64240 Len=0 MSS=1460
8	0.036139556	192.168.50.100	192.168.50.101	TCP	74	39620 → 139 [SYN] Seq=0 Win=64240 Len=0 MSS=1460
9	0.036149377	192.168.50.100	192.168.50.101	TCP	74	53518 → 199 [SYN] Seq=0 Win=64240 Len=0 MSS=1460
10	0.036157167	192.168.50.100	192.168.50.101	TCP	74	58852 → 23 [SYN] Seq=0 Win=64240 Len=0 MSS=1460
11	0.036165725	192.168.50.100	192.168.50.101	TCP	74	38240 → 135 [SYN] Seq=0 Win=64240 Len=0 MSS=1460
12	0.036173437	192.168.50.100	192.168.50.101	TCP	74	47556 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460
13	0.036502024	192.168.50.101	192.168.50.100	TCP	60	110 → 34790 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
14	0.036502119	192.168.50.101	192.168.50.100	TCP	74	445 → 51096 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0
15	0.036502156	192.168.50.101	192.168.50.100	TCP	74	21 → 60488 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0
16	0.036502192	192.168.50.101	192.168.50.100	TCP	74	22 → 59064 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0
17	0.036502229	192.168.50.101	192.168.50.100	TCP	60	256 → 60096 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
18	0.036502264	192.168.50.101	192.168.50.100	TCP	74	139 → 39620 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0
19	0.036502300	192.168.50.101	192.168.50.100	TCP	60	199 → 53518 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
20	0.036502335	192.168.50.101	192.168.50.100	TCP	74	23 → 58852 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0
21	0.036526005	192.168.50.100	192.168.50.101	TCP	66	51096 → 445 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSv
22	0.036533282	192.168.50.100	192.168.50.101	TCP	66	60488 → 21 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSv
23	0.036537665	192.168.50.100	192.168.50.101	TCP	66	59064 → 22 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSv

Anche in questo caso osserviamo che quasi tutte le porte sono aperte.

Osserviamo che alla riga 20 ci è data la risposta SYN/ACK che sta a dire che la porta è aperta.

In questo terzo caso invece utilizzando lo scan -A osserviamo che è molto più aggressivo e invasiva rispetto agli altri due perché stabilisce una connessione completa e identifica la versione del servizio attivo su quella determinata porta, scannerizza lo script, il sistema OS e la sua versione in uso

The image shows a Kali Linux terminal window on the left and a Wireshark network capture window on the right. The terminal window displays the output of an Nmap -A scan of 192.168.50.101. The scan results show that the host is up, has 1012 closed TCP ports, and is running various services including FTP, SSH, Telnet, SMTP, DNS, and HTTP. The Wireshark window shows a network capture on interface eth0, displaying a list of packets. The selected packet is an ARP request (No. 1) from 192.168.50.101 to the broadcast address ff:ff:ff:ff:ff:ff. The packet details show the Ethernet II header and the ARP request structure.

```
(root@kali)~  
# nmap -A 192.168.50.101 -p 1-1024  
Starting Nmap 7.92 ( https://nmap.org ) at 2022-07-22 10:32 EDT  
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled. Try using --system-dns or specify valid servers with --dns-servers  
Nmap scan report for 192.168.50.101  
Host is up (0.00030s latency).  
Not shown: 1012 closed tcp ports (reset)  
PORT      STATE SERVICE      VERSION  
21/tcp    open  ftp          vsftpd 2.3.4  
|_ftp-anon: Anonymous FTP login allowed (FTP code 230)  
|_ftp-syst:  
|_STAT:  
|_FTP server status:  
|_  Connected to 192.168.50.100  
|_  Logged in as ftp  
|_  TYPE: ASCII  
|_  No session bandwidth limit  
|_  Session timeout in seconds is 300  
|_  Control connection is plain text  
|_  Data connections will be plain text  
|_  vsFTPd 2.3.4 - secure, fast, stable  
|_End of status  
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) DAV/2)  
|_http_server_header: Apache/2.2.8 (Ubuntu) DAV/2  
|_http_title: Metasploitable2 - Linux  
111/tcp   open  rpcbind      2 (RPC #100000)  
|_rpcinfo:  
|_  program version port/proto service  
|_  100000 2 111/tcp rpcbind  
|_  100000 2 111/udp rpcbind  
|_  100003 2,3,4 2049/tcp nfs  
|_  100003 2,3,4 2049/udp nfs  
|_  100005 1,2,3 54037/tcp mountd  
|_  100005 1,2,3 56874/udp mountd  
|_  100021 1,3,4 37621/tcp nlockmgr  
|_  100021 1,3,4 55950/udp nlockmgr
```

Wireshark capture details for the selected packet (No. 1):

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	PcsCompu_db:96:6a	Broadcast	ARP	42	who has 192.168.50.101? Tell 1

Frame 1: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface eth0, id 0  
Ethernet II, Src: PcsCompu\_db:96:6a (08:00:27:db:96:6a), Dst: Broadcast (ff:ff:ff:ff:ff:ff)  
Address Resolution Protocol (request)

Packet details:

```
0000 ff ff ff ff ff ff 08 00 27 db 96 6a 08 06 00 01 .....j....  
0010 08 00 06 04 00 01 08 00 27 db 96 6a c0 a8 32 64 .....j...2d  
0020 00 00 00 00 00 00 c0 a8 32 65 .....2e
```

```

root@kali: ~
File Actions Edit View Help
| 100000 2 111/udp rpcbind
| 100003 2,3,4 2049/tcp nfs
| 100003 2,3,4 2049/udp nfs
| 100005 1,2,3 54037/tcp mountd
| 100005 1,2,3 56874/udp mountd
| 100021 1,3,4 37621/tcp nlockmgr
| 100021 1,3,4 55950/udp nlockmgr
| 100024 1 40646/udp status
| 100024 1 45877/tcp status
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)
512/tcp open exec netkit-rsh rshcd
513/tcp open login?
514/tcp open shell Netkit rshd
MAC Address: 08:00:27:C6:2E:25 (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:
| smb-os-discovery:
| OS: Unix (Samba 3.0.20-Debian)
| Computer name: metasploitable
| NetBIOS computer name:
| Domain name: localdomain
| FQDN: metasploitable.localdomain
|_ System time: 2022-07-22T10:33:31-04:00
|_ smb2-time: Protocol negotiation failed (SMB2)
|_ clock-skew: mean: 2h00m03s, deviation: 2h49m43s, median: 2s
| smb-security-mode:
| account_used: guest
| authentication_level: user
| challenge_response: supported
|_ message_signing: disabled (dangerous, but default)
|_ nbstat: NetBIOS name: METASPLOITABLE, NetBIOS user: <unknown>, NetBIOS MAC: <unknown> (unknown)

TRACEROUTE
HOP RTT ADDRESS
1 0.30 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 75.28 seconds

```