

CONSEGNA W9D2

L'obiettivo dell'esercitazione è quello di eseguire tre tipi diversi di scansioni utilizzando il prompt "nmap" nei confronti della metasploitable ovvero:

- La scansione SYN
- La scansione TCP
- La scansione switch -A

Facoltativo: evidenziare la differenza tra la scansione completa TCP e la scansione SYN intercettando le richieste inviate dalla macchina sorgente con Wireshark.

Svolgimento

Con un ambiente preparato in precedenza, con la kali configurata con ip 192.168.50.100 che tramite rete interna comunicava con una metasploitable con ip 192.168.50.101. Facciamo dunque un ping per vedere se le due macchine comunicano. Dopo esserci accertati che le macchine comunicano tra di loro iniziamo a completare le richieste fatte la traccia. Partiamo con il primo comando:

- `sudo nmap -sS 192.168.50.101 -p 1-1024`

Questo comando ci permette di effettuare una scansione inviando pacchetti SYN per rilevare le porte aperte senza però completare il three-way handshake (è una sequenza di 3 messaggi tra client e server per stabilire una connessione TCP). Questa è una scansione molto veloce e tra le tre che ci chiede l'esercizio è la meno rilevabile.

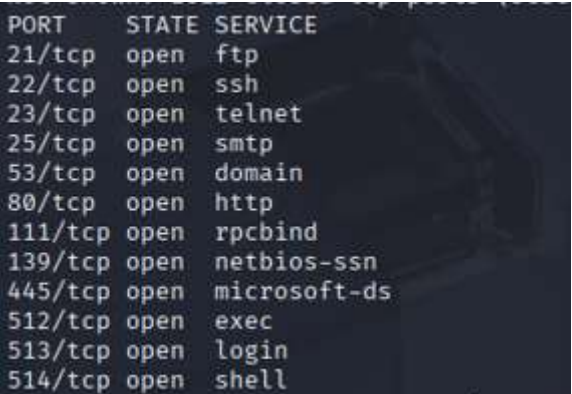
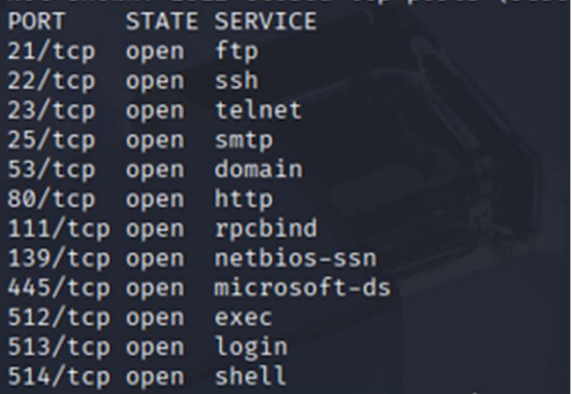
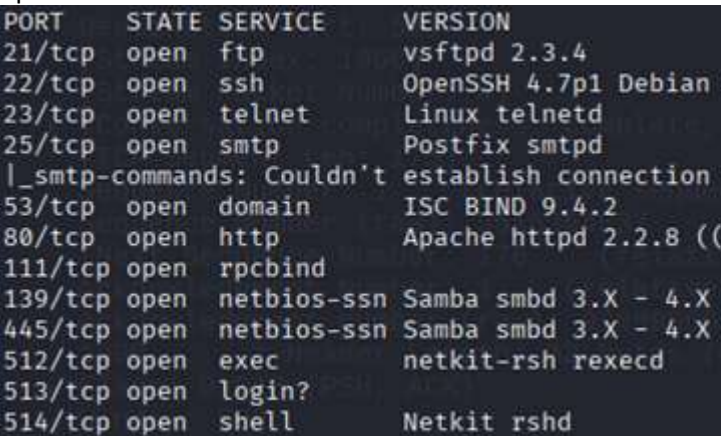
- `sudo nmap -sT 192.168.50.101 -p 1-1024`

Questo comando ci permette di effettuare una scansione completa con ogni porta, a differenza della precedente è un po' più lenta e anch'essa è facilmente rilevabile da sistemi di difesa

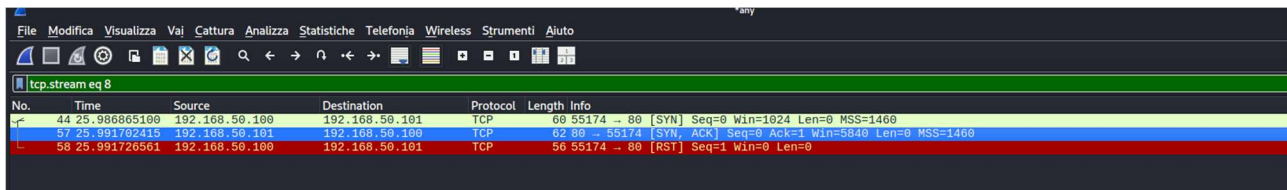
- `sudo nmap -A 192.168.50.101 -p 1-1024`

Questo comando permette di fare una scansione completa, rilevando ad esempio il sistema operativo, le varie versioni dei servizi presenti all'interno della macchina nella quale siamo entrati. Questa scansione è molto lenta.

Di seguito come richiesto dalla traccia una tabella che confronta le tre scansioni mettendo in evidenza il modo e il risultato ottenuto da ognuna.

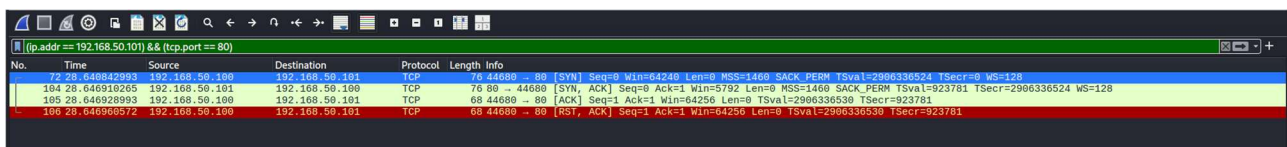
Fonte dello scan	Target dello scan	Tipologia di scan	Metodologia	Risultati ottenuti
Kali (192.168.50.100)	Metasploit (192.168.50.101)	-sS	Stealth, non completa la connessione TCP.	12 Porte aperte  <pre> PORT STATE SERVICE 21/tcp open ftp 22/tcp open ssh 23/tcp open telnet 25/tcp open smtp 53/tcp open domain 80/tcp open http 111/tcp open rpcbind 139/tcp open netbios-ssn 445/tcp open microsoft-ds 512/tcp open exec 513/tcp open login 514/tcp open shell </pre>
Kali (192.168.50.100)	Metasploit (192.168.50.101)	-sT	Connessione completa con ogni porta.	12 Porte aperte  <pre> 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 111/tcp open rpcbind 139/tcp open netbios-ssn 445/tcp open microsoft-ds 512/tcp open exec 513/tcp open login 514/tcp open shell </pre>
Kali (192.168.50.100)	Metasploit (192.168.50.101)	-A	Scansione aggressiva ma che genera molto traffico quindi facilmente rintracciabile.	Oltre alle porte, rilevate le varie versioni, il sistema operativo  <pre> PORT STATE SERVICE VERSION 21/tcp open ftp vsftpd 2.3.4 22/tcp open ssh OpenSSH 4.7p1 Debian 23/tcp open telnet Linux telnetd 25/tcp open smtp Postfix smtpd _smtp-commands: Couldn't establish connection 53/tcp open domain ISC BIND 9.4.2 80/tcp open http Apache httpd 2.2.8 ((Ubuntu)) 111/tcp open rpcbind 139/tcp open netbios-ssn Samba smbd 3.X - 4.X 445/tcp open netbios-ssn Samba smbd 3.X - 4.X 512/tcp open exec netkit-rsh rexecd 513/tcp open login? 514/tcp open shell Netkit rshd </pre>

Nella figura sottostante possiamo vedere la comunicazione forzata che avviene tramite il primo prompt e leggendo i pacchetti tramite Wireshark ci possiamo rendere conto come la connessione TCP non venga completata per far sì che sia meno facile da intercettare, e lo notiamo perché invece del classico SYN-SYN,ACK-ACK troviamo SYN-SYN,ACK-RST che dimostra come la sequenza del tree-way handshake sia incompleta.



No.	Time	Source	Destination	Protocol	Length	Info
44	25.986865100	192.168.50.100	192.168.50.101	TCP	60	55174 → 80 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
57	25.991782415	192.168.50.101	192.168.50.100	TCP	62	80 → 55174 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1460
58	25.991726561	192.168.50.101	192.168.50.101	TCP	56	55174 → 80 [RST] Seq=1 Win=0 Len=0

Nella figura sottostante vediamo la comunicazione forzata che avviene tramite il secondo prompt e tramite lettura possiamo vedere come la connessione TCP anche in questo caso viene forzata la chiusura in quanto nell'ultimo scambio è presente il RST. Questa è una precauzione da parte di chi effettua lo scan per cercare di lasciare meno quanto meno tracce è possibile in quanto prima stabilisce una connessione finita come possiamo vedere dallo scambio SYS, SYS/ACK, SYS per poi non comunicare nulla forzando il RST.



No.	Time	Source	Destination	Protocol	Length	Info
72	28.648842993	192.168.50.100	192.168.50.101	TCP	76	44680 → 80 [SYN] Seq=0 Win=1024 Len=0 MSS=1460 SACK_PERM TSval=2906336524 TSecr=0 WS=128
104	28.646918265	192.168.50.101	192.168.50.100	TCP	76	80 → 44680 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1460 SACK_PERM TSval=923781 TSecr=2906336524 WS=128
105	28.646928993	192.168.50.100	192.168.50.101	TCP	68	44680 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=2906336530 TSecr=923781
106	28.646969572	192.168.50.100	192.168.50.101	TCP	68	44680 → 80 [RST, ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=2906336530 TSecr=923781