# NETWORK SCANNING con NMAP

Nell'esercizio di oggi abbiamo visto da vicino nmap e i suoi comandi.

Scansione nmap–sT / Scansione nmap–sS / Scansione nmap -A

## Scansione nmap -sT

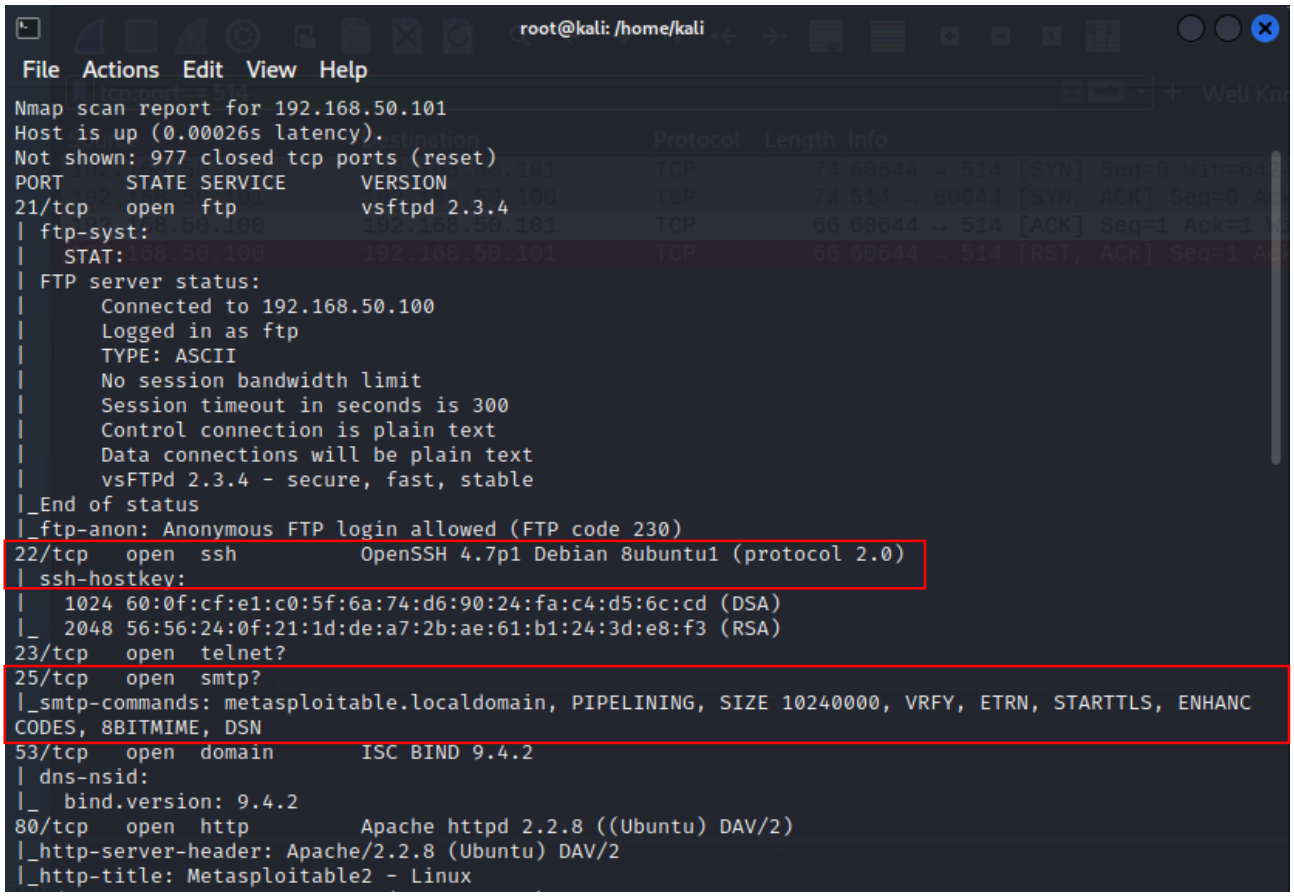
| PORT | STATE | SERVICE |
|---|---|---|
| 21 | OPEN | ftp |
| 22 | OPEN | Ssh |
| 23 | OPEN | telnet |
| 25 | OPEN | Smtp |
| 53 | OPEN | Domain |
| 80 | OPEN | http |
| 111 | OPEN | Rpcbind |
| 139 | OPEN | Netbios- ssn |
| 445 | OPEN | Microsoft- ds |
| 512 | OPEN | Exec |
| 513 | OPEN | Login |
| 514 | OPEN | shell |

## Scansione namap –sS

| PORT | STATE | SERVICE |
|---|---|---|
| 21 | OPEN | ftp |
| 22 | OPEN | Ssh |
| 23 | OPEN | telnet |
| 25 | OPEN | Smtp |
| 53 | OPEN | Domain |
| 80 | OPEN | http |
| 111 | OPEN | Rpcbind |
| 139 | OPEN | Netbios- ssn |
| 445 | OPEN | Microsoft- ds |
| 512 | OPEN | Exec |
| 513 | OPEN | Login |
| 514 | OPEN | shell |

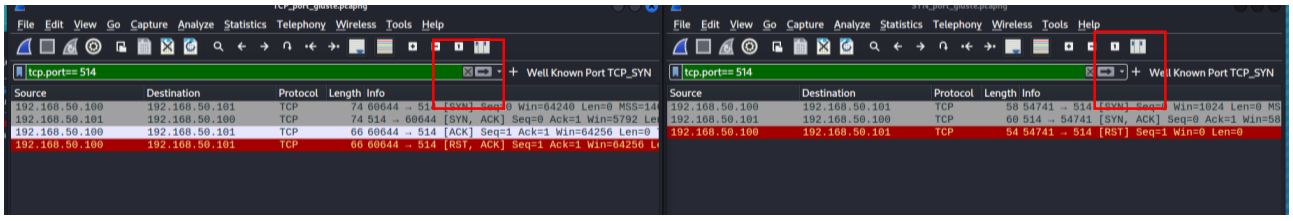
| INDIRIZZO SRC | INDIRIZZO DSC | SCAN | SERVIZI |
|---|---|---|---|
| 192.168.50.100 (linux) | 192.168.50.101 (meta) | Nmap –sT | 23 - 12 (well Know) |
| 192.168.50.100 (linux) | 192.168.50.101 (meta) | Nmap -sS | 23 - 12 (well Know) |
| 192.168.50.100 (linux) | 192.168.50.101 (meta) | (Sudo su) Nmap -A | 23 - 12 (well Know) |

## Scansione nmap -A




La scansione “nmap –A” fa una scansione molto piu approfondita. Essa ci permette di avere delle informazioni aggiuntive a discapito del tempo e della”discrezione”. Come possiamo leggere nel **man** di nmap essa ci restituisce sistema operativo, la versione, lo script scanning e il traceroute

## Wireshark




La differenza nelle due scansioni sono ben visibili all’interno dell’interfaccia di wireshark.

Sulla sinistra notiamo i tre passaggi del “3 hand shaking” (SYN >>SYN, HACK >> SYN)

Sulla destra invece abbiamo sempre tre passaggi ma con l’unica differenza nell’ultimo. La scansione “nmap –sS” ci mostra come alla fine il client restituisce un valore di reset (RST). Quest’ultimo non fa completare la comunicazione tra Client e Host. La scansione “nmap –sT” invece conclude il tutto mandando ACK finale e stabilendo una comunicazione tra Client e Host.