Realzione sulla scansione delle porte con `nmap` su Metasploitable e Windows10 pro

Introduzione

L'obiettivo dell'esercitazione di oggi, era usare la macchina Kali Linux per eseguire delle scansioni su porte utilizzando il tool `nmap` (già precedentemente installato sulla Kali), avendo come target la Metasploitable e Windows10 pro.

Che tipi di scansioni bisogna eseguire?

- OS fingerprint (per entrambe le macchine)
- Syn Scan
- TCP connect
- Version detection

È importante conoscere gli indirizzi IP di entrambe le macchine per poter effettuare queste scansioni.

Per individuare gli indirizzi IP delle macchine su cui vogliamo fare lo scan delle porte, esistono diversi metodi:

- Aprire dalla Virtualbox entrambe le macchine ed eseguire i seguenti comandi:
  - `ipconfig` su Windows
  - `ifconfig` su Metasploitable
- Dalla Kali eseguire il comando `sudo nmap -sn 192.168.50.0/24` che elenca gli host vivi con gli IP che fanno parte di quella LAN (nel mio caso `192.168.50.X`)

Una volta che abbiamo a disposizione gli indirizzi IP corretti delle due macchine, iniziamo la scansione usando `nmap`.

Scansione Os fingerprint

OS Fingerprinting è una scansione che viene effettuata per identificare il tipo di sistema operativo che è installato sulla macchina target.

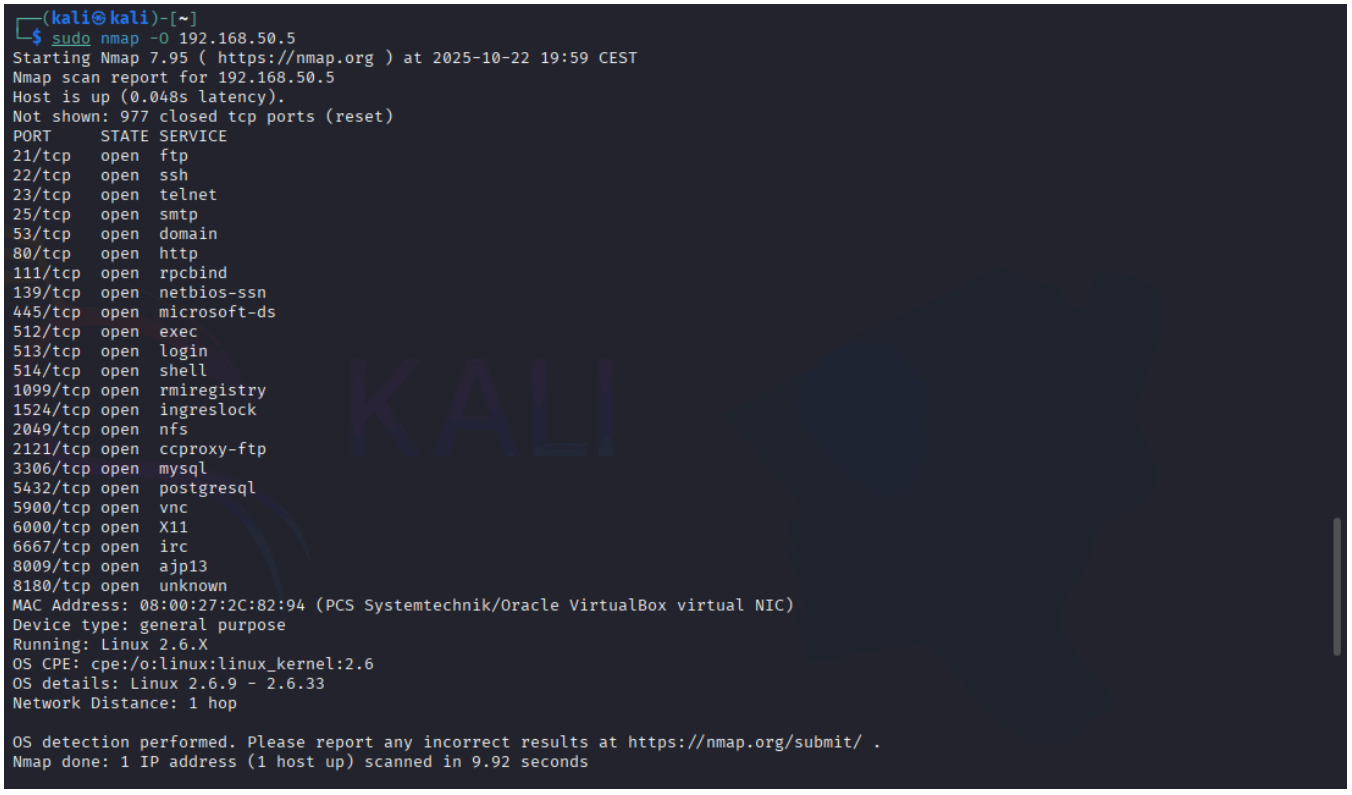
Il comando deve essere usato in questo modo:

```
sudo nmap -O <target>
```

Andremo ad eseguire questo comando sulla Kali, e lo faremo per entrambe le macchine, per vedere la distinzione tra i due sistemi operativi installati.

OS Fingerprinting per Metasploitable

```
sudo nmap -O 192.168.50.5
```



Lanciando il comando si può vedere come, nelle ultime righe, viene indicato:

- il tipo di device
- il sistema operativo
- versione del kernel
- uptime approssimativo

OS Fingerprinting per Windows10 pro

La stessa cosa accade se lanciamo lo stesso comando, prendendo come target l'indirizzo IP di Windows:

```
sudo nmap -O 192.168.50.6
```

Vediamo l'output con la seguente immagine:

Anche qui vediamo, nelle ultime righe:

- il tipo di device
- il sistema operativo
- versione del kernel
- uptime approssimativo

SYN Scan

Eseguiremo il comando SYN scan solo sulla Metasploitable.

Cosa fa esattamente? Esegue una scansione "half-open", inviando i pacchetti SYN e attendendo risposte SYN/ACK.

- Se riceve SYN/ACK, vuol dire che la porta è aperta
- Se riceve RST, vuol dire che la porta è chiusa

È più veloce e meno rilevabile.

Vediamo come funziona nel concreto:

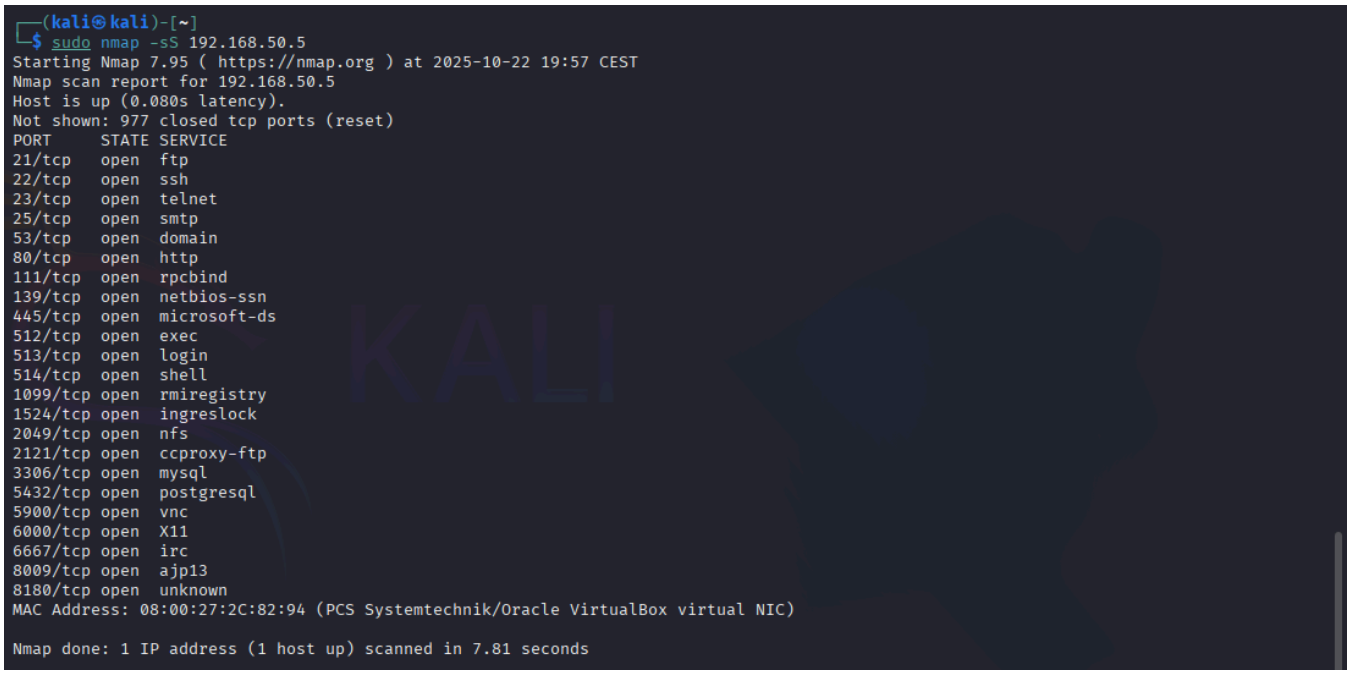
```
sudo nmap -sS <target>
```

Questo è il comando `nmap` da lanciare dalla Kali.

Nel mio caso, il comando è il seguente:

```
sudo nmap -sS 192.168.50.5
```

Vediamo di seguito l'output e lo analizziamo:



Quello che possiamo vedere è una sorta di tabella che raccoglie i dati sulle porte:

- colonna `PORT` → indica il numero della porta e il tipo (tcp, udp ect)
- colonna `STATE` → indica lo stato della porta, se è `open` oppure `closed`
- colonna `SERVICE` → indica il servizio attivo in quella porta (smtp, telnet, http ect)

TCP Connect Scan

La scansione TCP connect, utilizza una chiamata di sistema connect() che completa tutta la 3-way handshake TCP.

È più facile da eseguire, in quanto non richiede privilegi speciali.

La eseguiremo sempre sulla Metasploitable

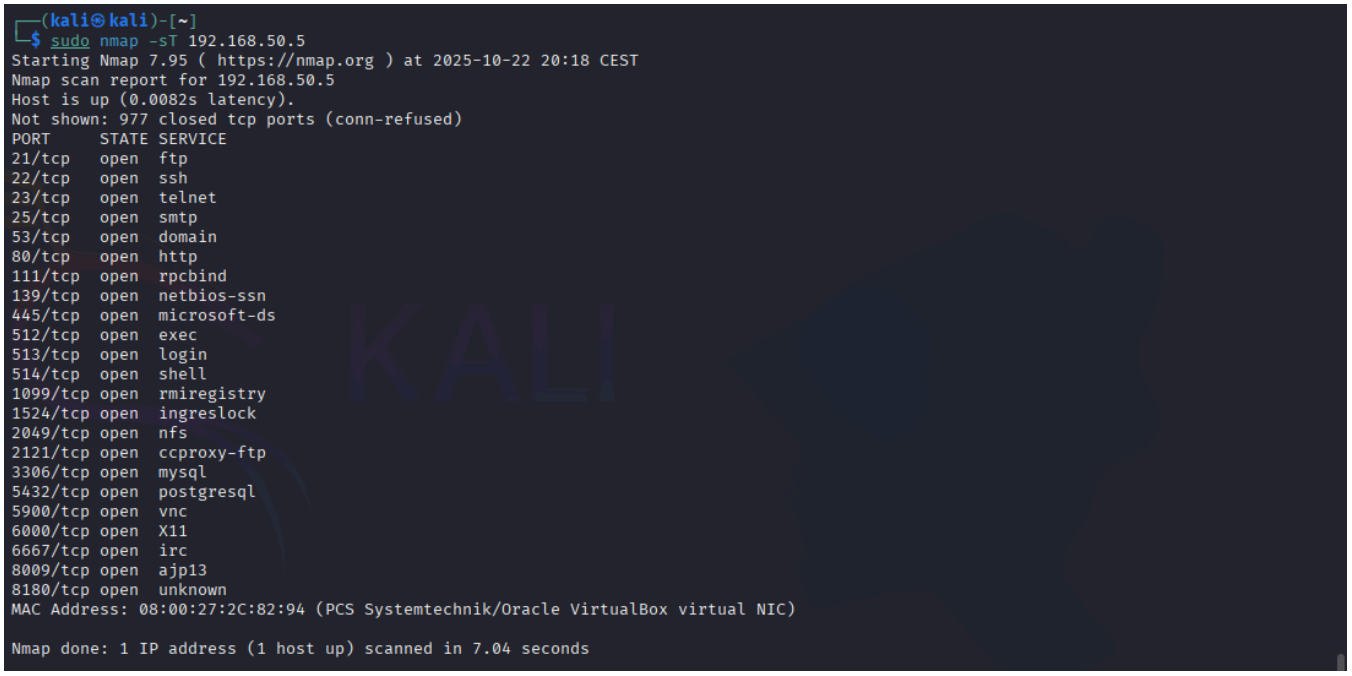
Viene lanciato con il seguente comando, da Kali:

```
sudo nmap -sT <target>
```

Nel mio caso:

```
sudo nmap -sT 192.168.50.5
```

Vediamo l'output:



Come possiamo vedere, l'output è praticamente identico a quello della SYN Scan. Come mai?

Riassumiamo nella seguente tabella le differenze:

Caratteristiche	SYN Scan	TCP Connect Scan
Tipo di connessione	parziale (half-open)	Completa (full 3-way handshake)
Rumorosità	Basso profilo (meno rilevabile)	Più evidente nei log
Richiede permessi root?	Sì	No
Velocità	Più veloce	Più lento
Visibilità sui log del target	Difficile da rilevare	facile da rilevare

Quindi cosa vediamo?

- I risultati sulle porte aperte/chiuse sono identici
- la differenza principale è nel modo in cui si ottengono, non nel risultato finale:
    - SYN è più stealth e rapido
    - TCP Connect è più semplice ma facilmente loggabile e meno efficiente

Version detection

Version Detection viene usato per identificare i servizi e le versioni su porte aperte.

Dopo aver identificato le porte aperte, invia richieste mirate per capire:

- Quale servizio è in esecuzione (es. http, ssh, ftp ect)
- La versione esatta del software (es. Apache 2.4.58, OpenSSH 8.2p1)

Usa un database di firme e risposte tipiche per confrontare con i dati ricevuti. Può anche riconoscere banner applicativi, protocolli particolari e, a volte, il nome del dispositivo.

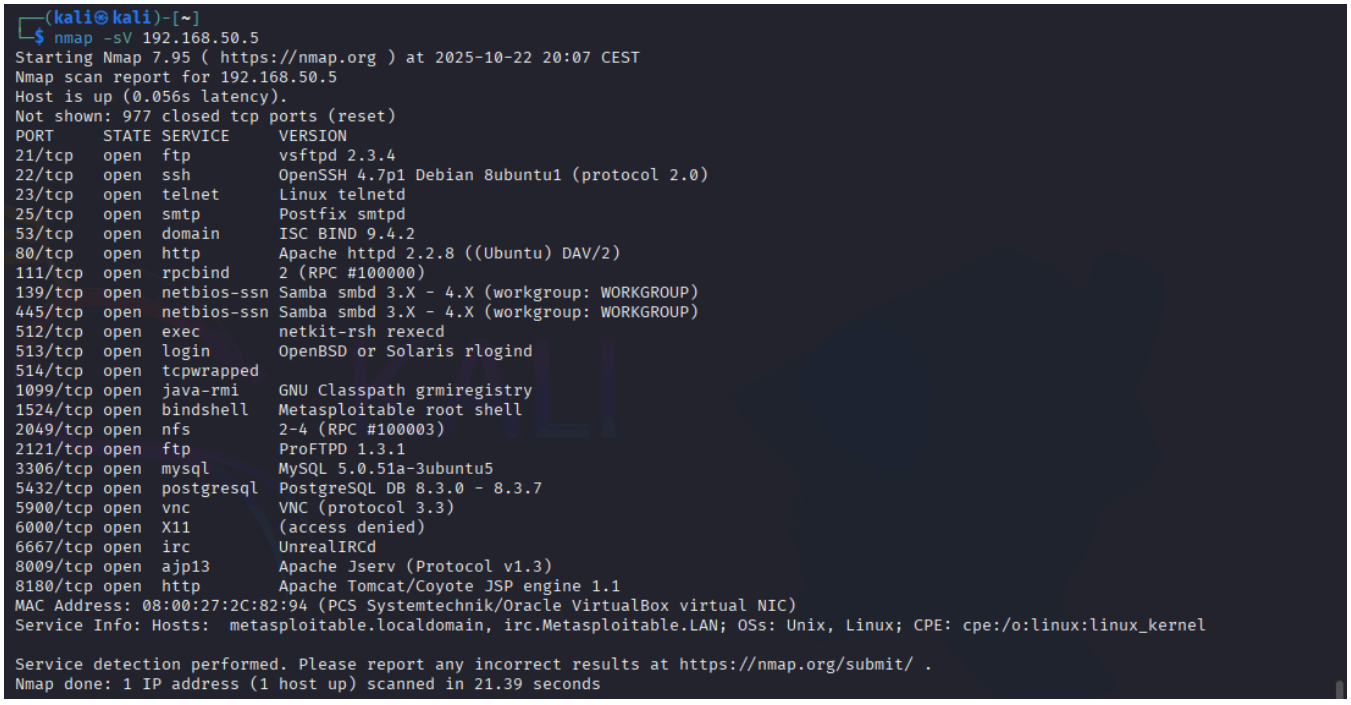
Ecco come deve essere lanciato il comando su Linux: ù

```
sudo nmap -sV <target>
```

Nel mio caso:

```
sudo nmap 192.168.50.5
```

Di seguito analizziamo l'output:



Come ci aspettavamo, l'output mostra chiaramente:

- `PORT` → le porte scansionate
- `STATE` → lo stato della porta
- `SERVICE` → il servizio attivo sulla porta
- `VERSION` → la versione del servizio attivo sulla porta

Conclusioni

La campagna di scansione ha permesso di identificare le porte aperte, i servizi esposti e le versioni software presenti sui sistemi analizzati, oltre ad ottenere indicazioni utili sul sistema operativo tramite fingerprinting.

Le scansioni SYN (`-sS`) si sono rivelate più rapide e meno evidenti nei log rispetto alle scansioni TCP connect (`-sT`), mentre i risultati sullo stato delle porte (aperte/chiuse/filtrate) sono risultate sostanzialmente coerenti tra le due metodologie; le differenze principali risiedono nella visibilità dell'attività nei log e nel requisito di privilegi.

L’identificazione delle versioni (`-sV`) ha confermato versioni datate di alcuni servizi che rappresentano potenziali rischi di vulnerabilità se non aggiornati; dove possibile, si raccomanda un controllo approfondito tramite tool e script NSE mirati.

L’OS fingerprinting (`-O`) ha fornito utili indicazioni sul tipo di sistema e sulla sua exposability, ma i risultati vanno interpretati con cautela quando sono presenti firewall o meccanismi di evasione.