REPORT S3/L1

Scansione dei servizi on Nmap

Traccia

Si richiede allo studente di effettuare le seguenti scansioni sul target Metasploitable:

- OS fingerprint.
- Syn Scan.
- TCP connect - trovate differenze tra i risultati della scansioni TCP connect e SYN?
- Version detection.

E la seguente sul target Windows:

- OS fingerprint.

Svolgimento

Controlliamo le impostazioni di rete delle macchine di Kali e Metasploitable in modo che entrambe si trovino sulla stessa rete. Assegniamo i seguenti indirizzi alle macchine:

- 192.168.20.4 per Kali Linux
- 192.168.20.5 per Metasploitable2

Eseguiamo i comandi visti a lezione per scansionare il S.O. Metasploitable2 da Kali Linux.

Per l'OS fingerprint lanciamo il comando: *sudo nmap –O 192.168.20.5*

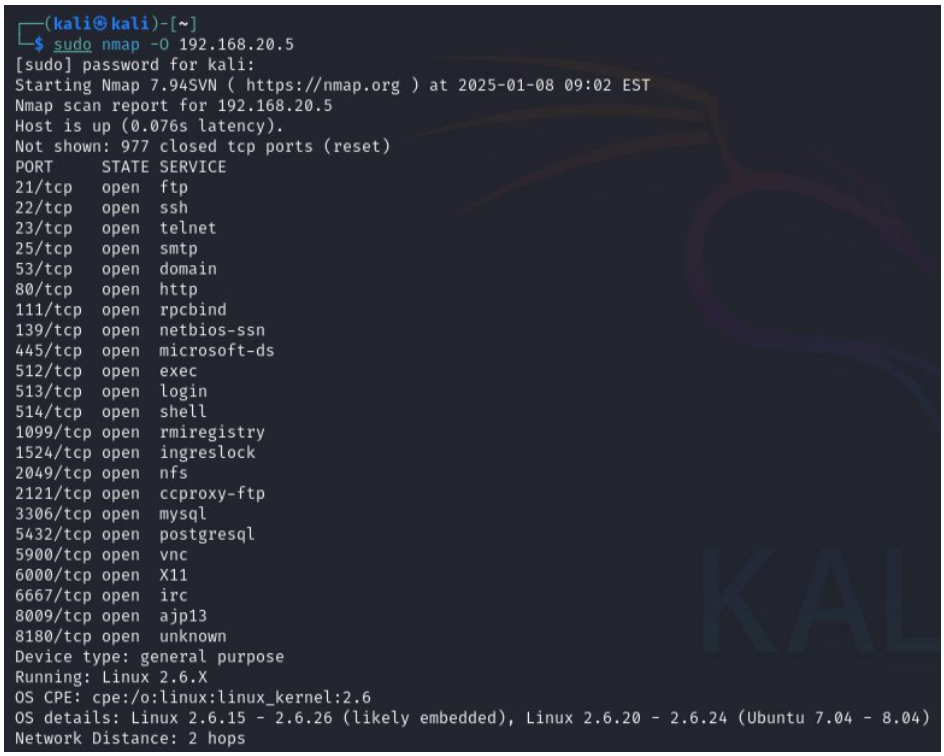


Figura 1. OS Fingerprint di Metasploitable2

Possiamo vedere che alla terz'ultima riga del risultato possiamo leggere **OS CPE: ....: linux_kernel:2.6** che è esattamente il kernel di Metasploitable2.

Per il syn scan lanciamo: sudo *nmap-sS 192.168.20.5*

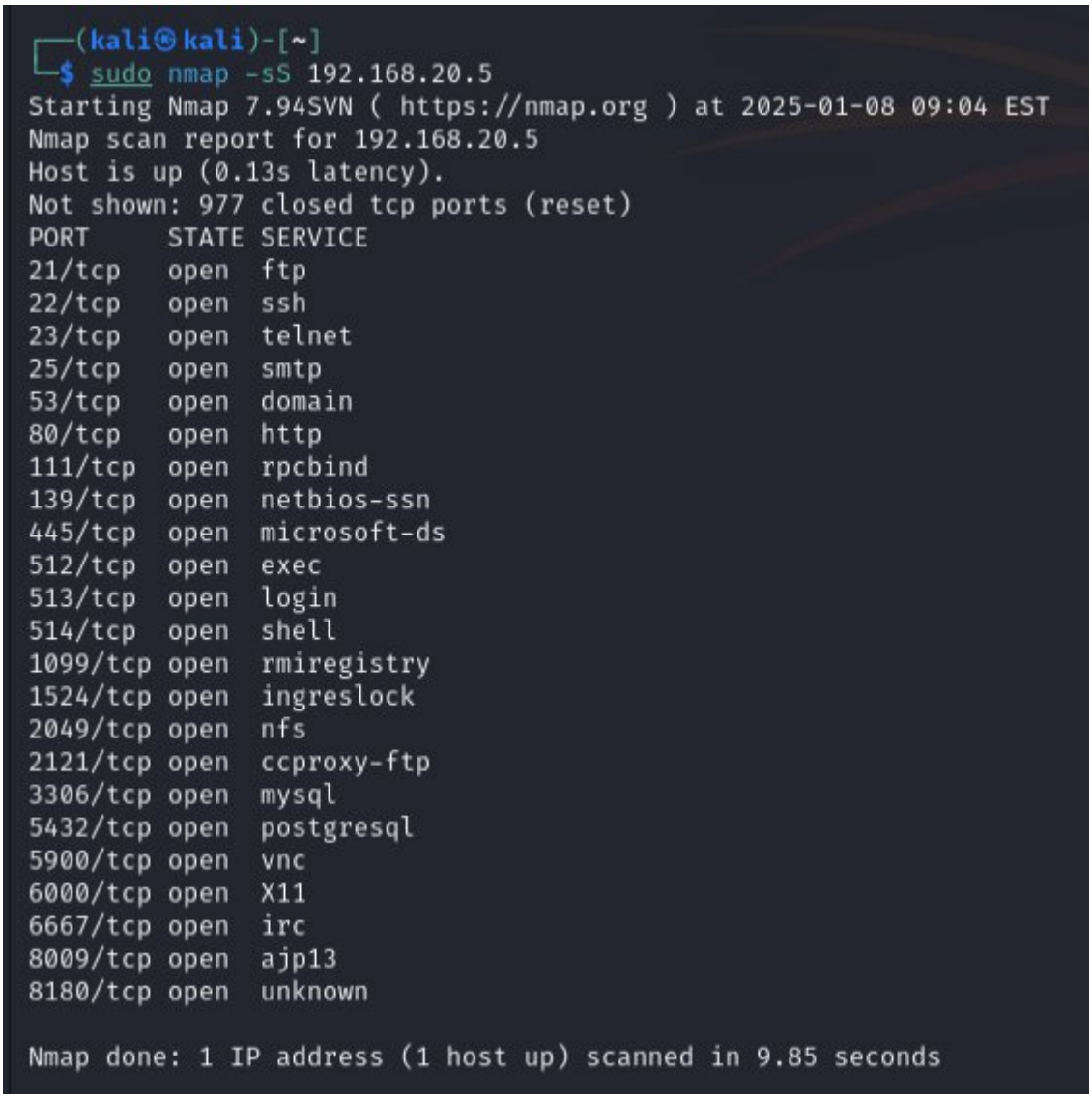


Figura 2. Syn Scan di Metasploitable2

Per il TCP connect lanciamo: sudo *nmap-sT 192.168.20.5*

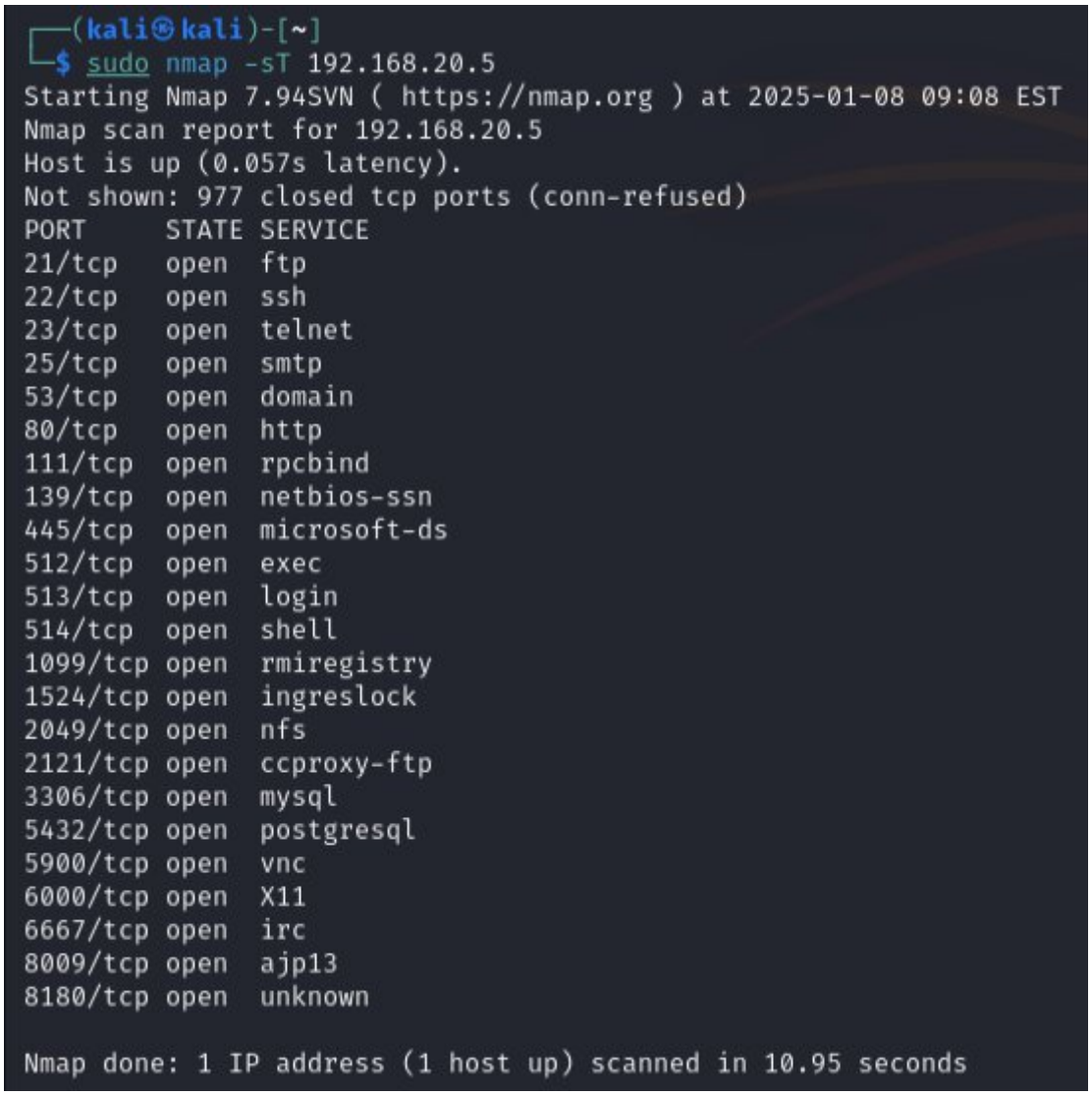


Figura 3. TCP connect su Metasploitable2

Dopo l’esecuzione di questo comando non notiamo differenze nel risultato con il precedente.

Per il version detection lanciamo: sudo *nmap-sV 192.168.20.5*

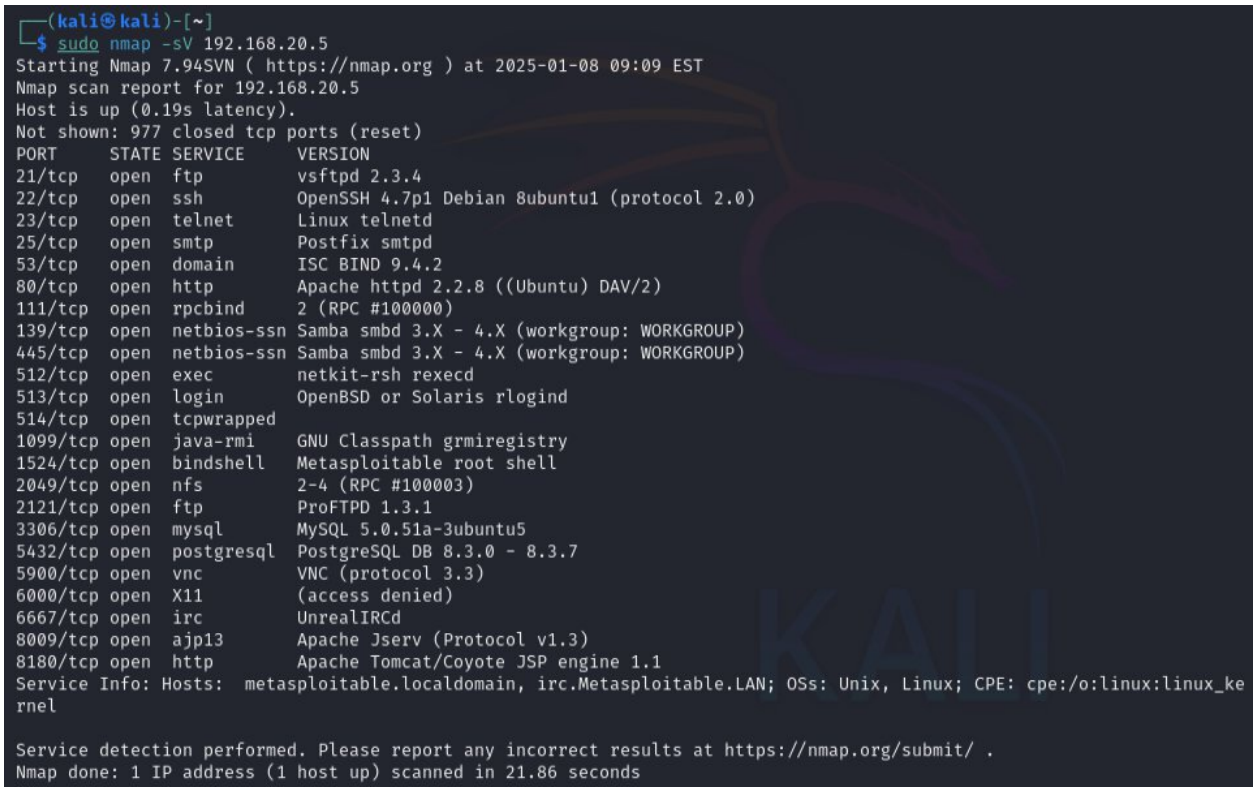


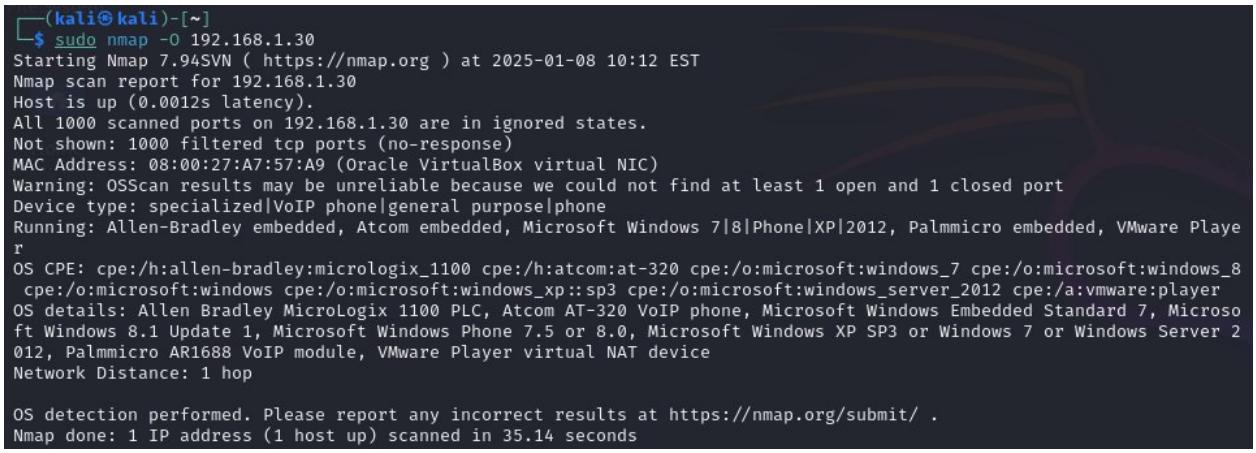
Figura 4. Version detection su Metasploitable2

Per eseguire il version detection su windows lanciamo la VM di Windows 10 e anche su questo sistema impostiamo manualmente l'indirizzo IP in modo che Kali e Windows siano sulla stessa rete.

Assegniamo rispoettivamente:

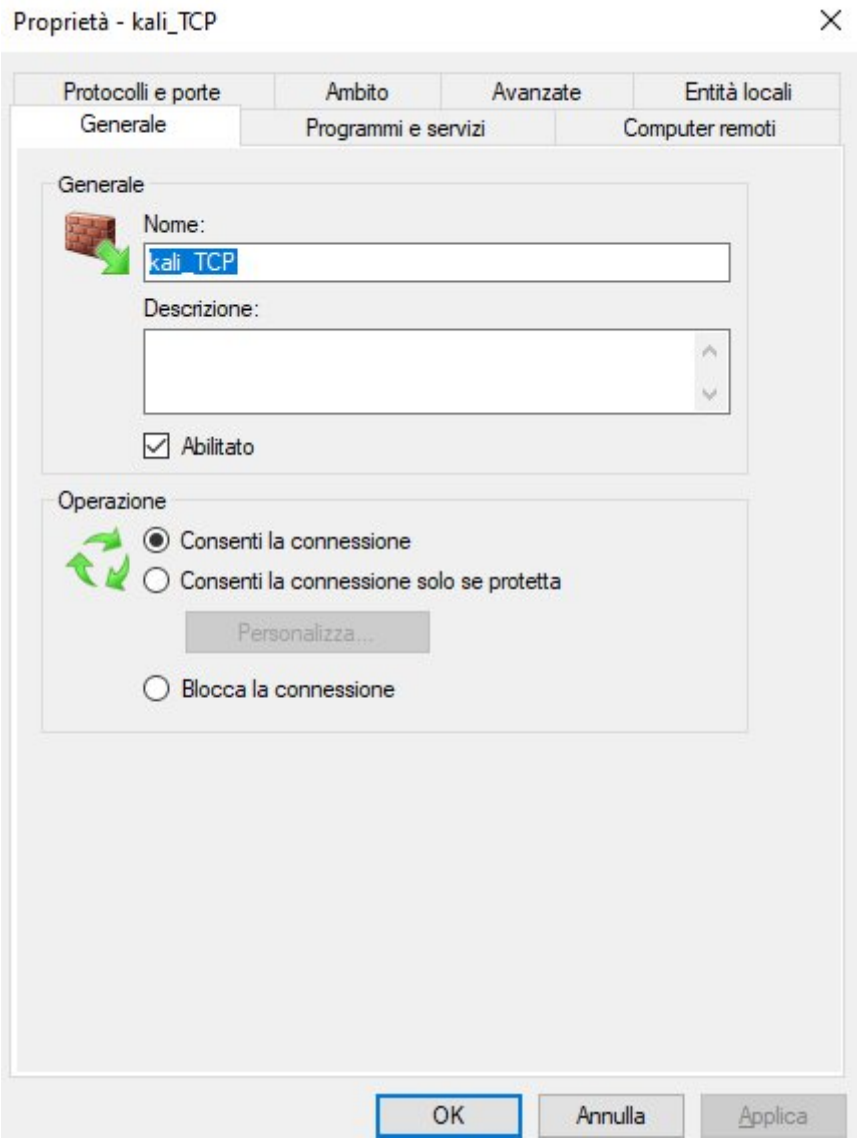
- 192.168.1.50 a Kali
- 192.168.1.30 a Windows

Proviamo il comando sudo *nmap-sV 192.168.1.30*, otteniamo come risposta l'errore riportato in figura:



Questo ci fa capire che, anche se i due sistemi sono nella stessa rete, Windows non è "raggiungibile". Proviamo ad aggiungere delle regole firewall su Windows così da permettere la comunicazione con il protocollo TCP, che viene utilizzato da nmap.

Impostiamo la regola sia in entrata che in uscita, come in figura:



A questo punto ripetiamo il comando ed otteniamo:

```
┌──(kali㉿kali)-[~]
└─$ sudo nmap -O 192.168.1.30
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-01-08 10:16 EST
Nmap scan report for 192.168.1.30
Host is up (0.0045s latency).
Not shown: 997 filtered tcp ports (no-response)
PORT    STATE SERVICE
135/tcp open  msrpc
139/tcp open  netbios-ssn
445/tcp open  microsoft-ds
MAC Address: 08:00:27:A7:57:A9 (Oracle VirtualBox virtual NIC)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: general purpose
Running (JUST GUESSING): Microsoft Windows 2019|10 (90%)
OS CPE: cpe:/o:microsoft:windows_10
Aggressive OS guesses: Microsoft Windows Server 2019 (90%), Microsoft Windows 10 1909 (88%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 1 hop

OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 21.71 seconds
```

Possiamo vedere che l’identificazione del S.O. è andata a buon fine al 90%.