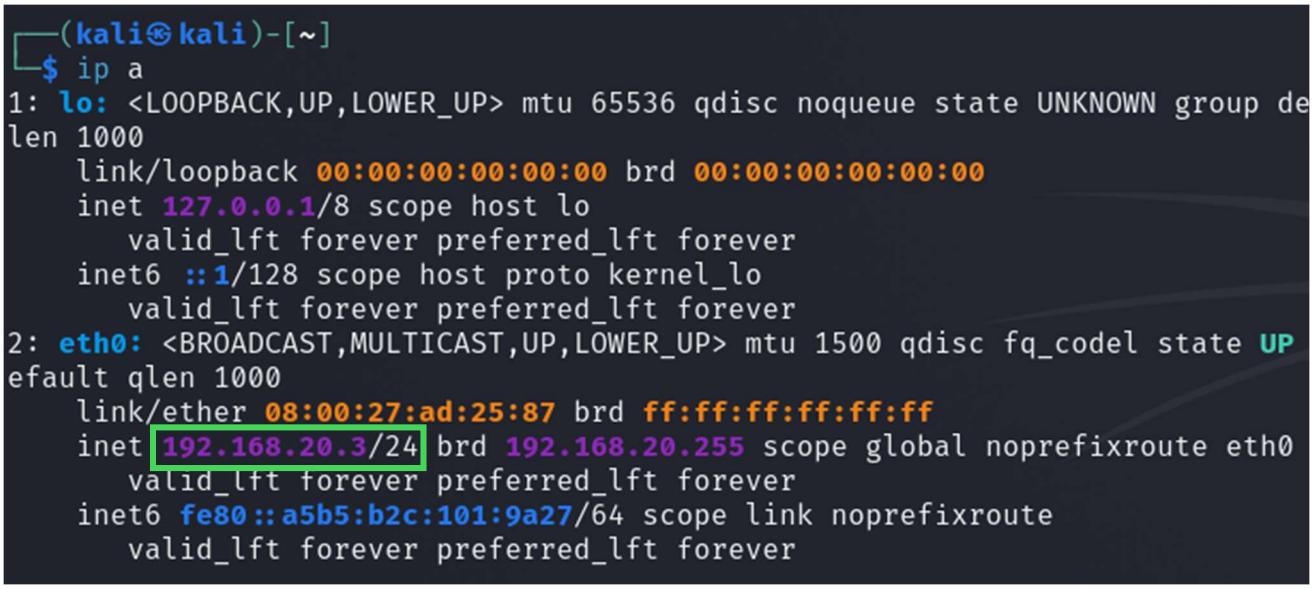
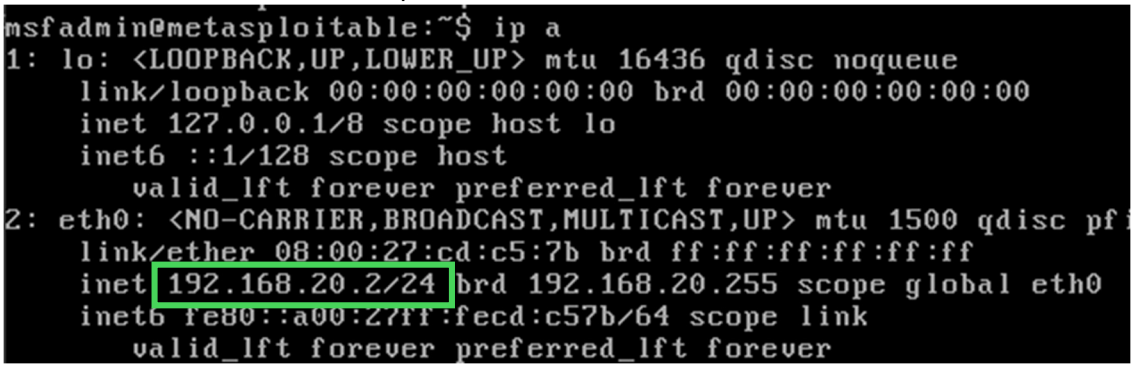
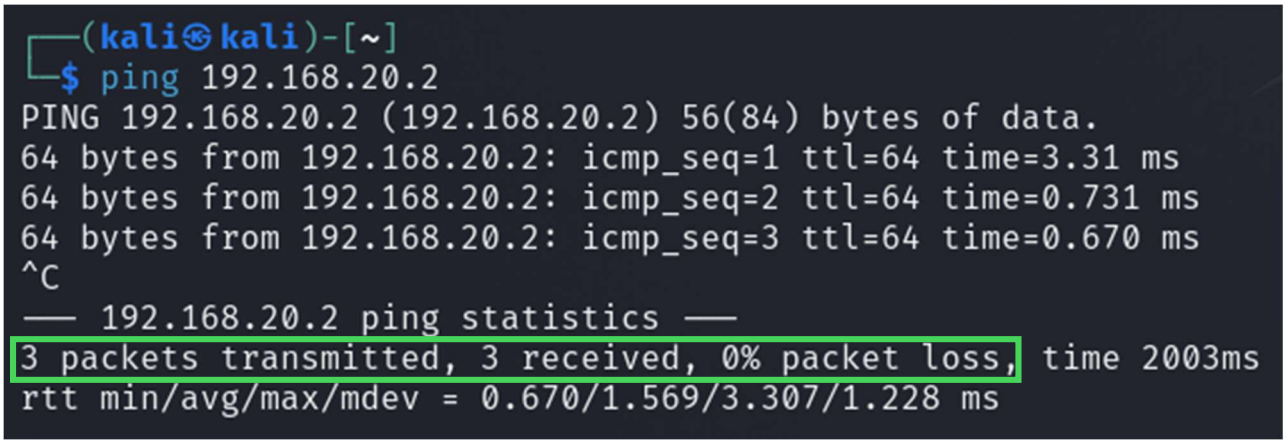
Come primo passo, per eseguire una corretta scansione, metto tutte le macchine sulla stessa rete interna.
Successivamente visualizzo e se necessario modifico gli indirizzi IP in modo da avere degli indirizzi che stanno sotto la stessa rete.
Visualizzo l'IP di kali:



adesso visualizzo l'IP di metasploitable usando lo stesso comando



provo a pingare metasploitable da kali per vedere se c'è comunicazione:



il ping ha dato esito positivo quindi le due macchine sono collegate correttamente sulla stessa rete.

Adesso provo a scovare il sistema operativo della macchina metasploitable con l' OS fingerprint dando i privilegi alla macchina kali perché nmap utilizza pacchetti RAW per inviare richieste personalizzate e analizzare le risposte.
I pacchetti RAW sono dei pacchetti creati e manipolati da nmap a un livello del protocollo di rete (basso livello) per avere il pieno controllo su essi.
Successivamente, dopo aver dato a kali i privilegi, il comando per eseguire OS fingerprint è:

```
┌──(root㉿kali)-[/home/kali]
└─# nmap -O 191.168.20.2_
```

Restituendo in Output le porte aperte seguite da questa stringa che indica le informazioni del sistema operativo target:

```
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6
OS details: Linux 2.6.9 - 2.6.33
Network Distance: 1 hop
```

La consegna richiede di eseguire altri due tipi di scansione ovvero Syn scan e TCP connect.
Per eseguire il Syn scan si usa il seguente comando:

```
┌──(root㉿kali)-[/home/kali]
└─# nmap -sS 192.168.20.2 _
```

Dove -sS sta per "SYN Stealth" e 192.168.20.2 è l'indirizzo IP della macchina target.
Il comando restituirà in output tutte le porte analizzate senza completare il 3-way-handshake inviando solo richieste di SYN, SYN/ACK e reset a fine comunicazione:

```
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-01-08 09:51 EST
Nmap scan report for 192.168.20.2
Host is up (0.00071s latency).
Not shown: 977 closed tcp ports (reset)
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
1099/tcp open  rmiregistry
1524/tcp open  ingreslock
2049/tcp open  nfs
2121/tcp open  ccproxy-ftp
3306/tcp open  mysql
5432/tcp open  postgresql
5900/tcp open  vnc
6000/tcp open  X11
6667/tcp open  irc
8009/tcp open  ajp13
8180/tcp open  unknown
MAC Address: 08:00:27:CD:C5:7B (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 13.93 seconds
```

successivamente ho eseguito la scansione TCP connect usando il comando:

```
┌──(root㉿kali)-[/home/kali]
└─# nmap -sT 192.168.20.2
```

dove -sT sta per “scanTCP” e 192.168.20.2 è l’indirizzo IP della macchina target
il comando restituirà in output una lista delle porte aperte usando il protocollo TCP completo quindi seguendo tutte le fasi del 3-way-handshake, risultando più invasivo e meno nascosto rispetto al SYN Scan

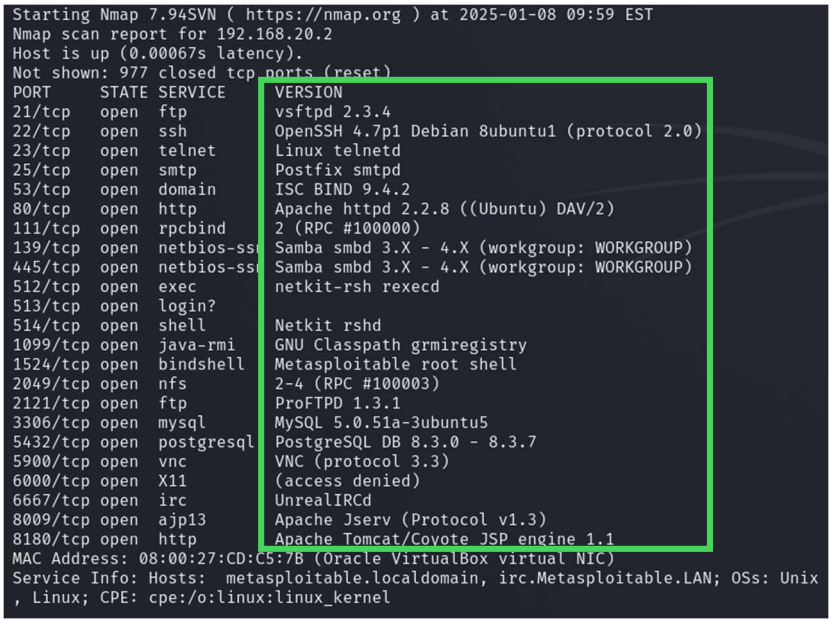
```
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-01-08 09:55 EST
Nmap scan report for 192.168.20.2
Host is up (0.00062s latency).
Not shown: 977 closed tcp ports (conn-refused)
PORT     STATE SERVICE
21/tcp   open  ftp
22/tcp   open  ssh
23/tcp   open  telnet
25/tcp   open  smtp
53/tcp   open  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
1099/tcp open  rmiregistry
1524/tcp open  ingreslock
2049/tcp open  nfs
2121/tcp open  ccproxy-ftp
3306/tcp open  mysql
5432/tcp open  postgresql
5900/tcp open  vnc
6000/tcp open  X11
6667/tcp open  irc
8009/tcp open  ajp13
8180/tcp open  unknown
MAC Address: 08:00:27:CD:C5:7B (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 13.21 seconds
```

La differenza tra i due scan sta appunto nel tipo di protocollo usato per inviare i paccehtti, come detto, il SYN Scan non completa il 3-way-handshake inviando solo richieste di tipo SYN, SYN/ACK e reset per interrompere la comunicazione, mentre il TCP Connect esegue un 3-way-handshake completo inviando le normali richieste di SYN, STN/ACK e ACK per eseguire la connessione e lo scambio dei pacchetti.

La successiva richiesta è quella di eseguire uno scan della versione dei servizi in esecuzione sulle porte aperte (Version Detection) e quindi per fare ciò, ho usato il comando:

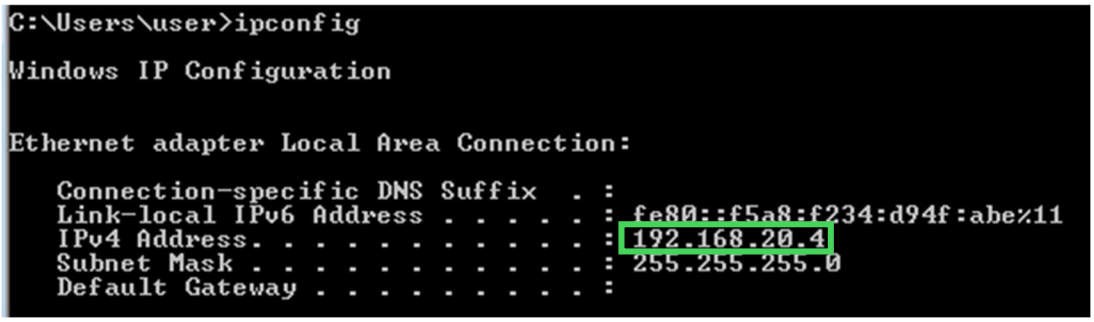
```
┌──(root㉿kali)-[/home/kali]
└─# nmap -sV 192.168.20.2
```

Dove -sV sta per “Scan Version” e 192.168.20.2 è l’indirizzo IP della macchina target.
Questo comando restituirà in output una lista delle porte aperte seguite dalla versione del servizio in esecuzione sulla rispettiva porta:

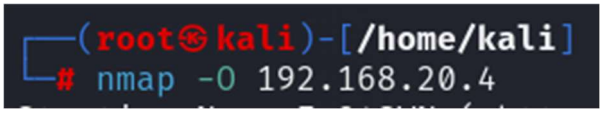


```
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-01-08 09:59 EST
Nmap scan report for 192.168.20.2
Host is up (0.00067s latency).
Not shown: 977 closed tcp ports (reset)
PORT     STATE SERVICE     VERSION
21/tcp   open  ftp         vsftpd 2.3.4
22/tcp   open  ssh         OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp   open  telnet      Linux telnetd
25/tcp   open  smtp        Postfix smtpd
53/tcp   open  domain      ISC BIND 9.4.2
80/tcp   open  http        Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp  open  rpcbind     2 (RPC #100000)
139/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
512/tcp  open  exec        netkit-rsh rexecd
513/tcp  open  login?
514/tcp  open  shell       Netkit rshd
1099/tcp open  java-rmi    GNU Classpath grmiregistry
1524/tcp open  bindshell   Metasploitable root shell
2049/tcp open  nfs         2-4 (RPC #100003)
2121/tcp open  ftp         ProFTPD 1.3.1
3306/tcp open  mysql       MySQL 5.0.51a-3ubuntu5
5432/tcp open  postgresql  PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp open  vnc         VNC (protocol 3.3)
6000/tcp open  X11         (access denied)
6667/tcp open  irc         UnrealIRCd
8009/tcp open  ajp13       Apache Jserv (Protocol v1.3)
8180/tcp open  http        Apache Tomcat/Coyote JSP engine 1.1
MAC Address: 08:00:27:CD:C5:7B (Oracle VirtualBox virtual NIC)
Service Info: Hosts:  metasploitable.localdomain, irc.Metasploitable.LAN; OSs: Unix
, Linux; CPE: cpe:/o:linux:linux_kernel
```

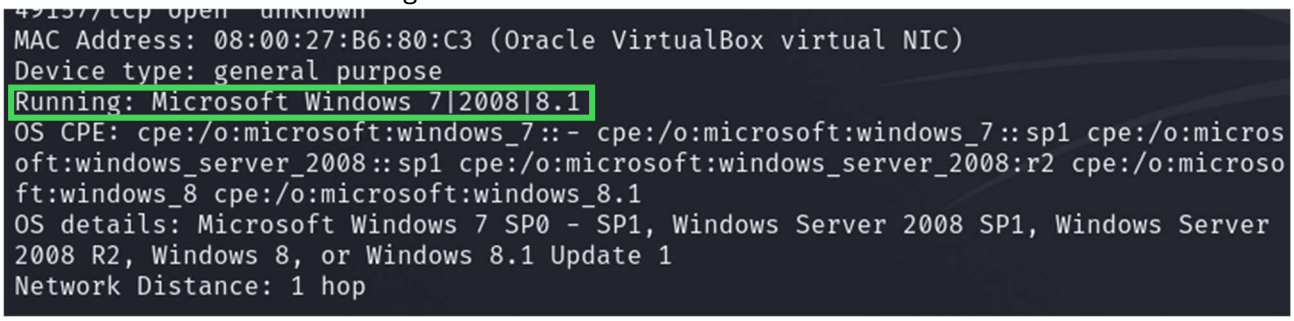
La parte finale di questo esercizio richiedeva di eseguire un OS fingerprint su una macchina target con sistema operativo windows.
Dopo aver settato le schede per avere la macchina kali e la macchina windows sulla stessa rete, sono andato a visualizzare l'indirizzo IP di windows



Entrambi i dispositivi sono sotto la stessa rete.
Adesso procedo con l'OS fingerprint con il codice usato in precedenza per la macchina metasploitable:



Dove 192.168.20.4 è l'indirizzo IP della macchina Windows.
Avremo come output una lista di porte aperte o sconosciute e a seguito delle stringhe che danno informazioni sulla macchina target:



possiamo notare come la rilevazione del sistema operativo è andata a buon fine.