

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:

```
(kali㉿kali)-[~]
└─$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group de
len 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host proto kernel_lo
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP
efault qlen 1000
    link/ether 08:00:27:ad:25:87 brd ff:ff:ff:ff:ff:ff
    inet 192.168.20.3/24 brd 192.168.20.255 scope global noprefixroute eth0
        valid_lft forever preferred_lft forever
    inet6 fe80::a5b5:b2c:101:9a27/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
```

adesso visualizzo l'IP di metasploitable usando lo stesso comando

```
msfadmin@metasploitable:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc pfifo
    link/ether 08:00:27:cd:c5:7b brd ff:ff:ff:ff:ff:ff
    inet 192.168.20.2/24 brd 192.168.20.255 scope global eth0
    inet6 fe80::a00:27ff:fedc:c57b/64 scope link
        valid_lft forever preferred_lft forever
```

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

```
(kali㉿kali)-[~]
└─$ ping 192.168.20.2
PING 192.168.20.2 (192.168.20.2) 56(84) bytes of data.
64 bytes from 192.168.20.2: icmp_seq=1 ttl=64 time=3.31 ms
64 bytes from 192.168.20.2: icmp_seq=2 ttl=64 time=0.731 ms
64 bytes from 192.168.20.2: icmp_seq=3 ttl=64 time=0.670 ms
^C
— 192.168.20.2 ping statistics —
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 0.670/1.569/3.307/1.228 ms
```

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 pacchetti, 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, SYN/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-ssd  Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp   open  netbios-ssd  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

```
C:\Users\user>ipconfig

Windows IP Configuration

Ethernet adapter Local Area Connection:

    Connection-specific DNS Suffix  . : 
    Link-local IPv6 Address . . . . . : fe80::f5a8:f234:d94f:abe%11
    IPv4 Address. . . . . : 192.168.20.4
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . :
```

Entrambi i dispositivi sono sotto la stessa rete.

Adesso procedo con l'OS fingerprint con il codice usato in precedenza per la macchina metasploitable:

```
(root@kali)-[/home/kali]
# nmap -O 192.168.20.4
```

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:

```
49157/tcp open  unknown
MAC Address: 08:00:27:B6:80:C3 (Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Microsoft Windows 7|2008|8.1
OS CPE: cpe:/o:microsoft:windows_7::- cpe:/o:microsoft:windows_7::sp1 cpe:/o:microsoft:windows_server_2008::sp1 cpe:/o:microsoft:windows_server_2008:r2 cpe:/o:microsoft:windows_8 cpe:/o:microsoft:windows_8.1
OS details: Microsoft Windows 7 SP0 - SP1, Windows Server 2008 SP1, Windows Server 2008 R2, Windows 8, or Windows 8.1 Update 1
Network Distance: 1 hop
```

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