

1. nmap -sn -PE <ip target>

Verificare se il target (la macchina **Metasploitable2**) è acceso e **raggiungibile** in rete, **senza fare una scansione delle porte**.

Utilizza un **ping ICMP Echo Request** (come il comando ping) per vedere se il sistema risponde.

```
(kali㉿kali)-[~]  
$ nmap -sn -PE 192.168.50.101  
Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-01 10:30 EDT  
Nmap scan report for 192.168.50.101  
Host is up (0.00022s latency).  
MAC Address: 08:00:27:CC:EB:61 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)  
Nmap done: 1 IP address (1 host up) scanned in 0.12 seconds
```

1) Il target è **attivo** e ha risposto al ping:

Host is up (0.00022s latency).

2) È stata anche rilevata la **scheda di rete virtuale**, segno che la macchina gira su VirtualBox:

MAC Address: 08:00:27:CC:EB:61 (PCS Systemtechnik/Oracle VirtualBox virtual NIC).

2. netdiscover -r <ip target>

Identificare host attivi nella rete locale 192.168.50.0/24 tramite pacchetti **ARP**, per rilevare IP, MAC e produttore della scheda.

```
Currently scanning: Finished! | Screen View: Unique Hosts
Trash
1 Captured ARP Req/Rep packets, from 1 hosts. Total size: 60
```

IP	At MAC Address	Count	Len	MAC Vendor / Hostname
192.168.50.101	08:00:27:cc:eb:61	1	60	PCS Systemtechnik GmbH

Host attivo trovato:

IP: 192.168.50.101

MAC: 08:00:27:cc:eb:61

Vendor: *PCS Systemtechnik GmbH* (VirtualBox)

Quindi la macchina **Metasploitable2** è attiva, correttamente visibile nella rete interna, e risponde alle richieste ARP.

3. crackmapexec smb <ip target>

Effettuare **enumerazione SMB** sul target per ottenere informazioni di sistema e sul protocollo SMB in uso.

```
(root@kali)-[/home/kali]
# crackmapexec smb 192.168.50.101
SMB 192.168.50.101 445 METASPLOITABLE [*] Unix (name:METASPLOITABLE) (domain:localdomain) (signing:False) (SMBv1:True)
```

Risultato ottenuto:

IP: 192.168.50.101

Porta: 445 (porta standard SMB)

Hostname: METASPLOITABLE

Sistema: Unix

Dominio: localdomain

SMB Signing: False : Potenziale vulnerabilità: comunicazioni non firmate

SMBv1: True : Protocollo vecchio e vulnerabile (es. EternalBlue)

SMB è attivo e utilizza **SMBv1**, noto per varie vulnerabilità critiche.

Signing disabilitato = un attaccante può **man-in-the-middle** il traffico SMB.

Il sistema è un **Unix/Linux** che emula SMB (tramite Samba), come previsto per Metasploitable2.

Altri protocolli: LDAP, WINRM, RDP, MSSQL, FTP, KERBEROS, NFS.

4. nmap <ip target> --top-ports 10 --open

Il comando ha l'obiettivo di scansionare le **10 porte TCP più comuni** sulla macchina con l'IP **192.168.50.101** (in questo caso, **Metasploitable2**) e mostrare **solo le porte aperte**. Questo ti aiuta a identificare rapidamente i servizi esposti su quella macchina.

```
(kali㉿kali)-[~]  
$ nmap 192.168.50.101 --top-ports 10 --open  
Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-01 12:21 EDT  
Nmap scan report for 192.168.50.101  
Host is up (0.00016s latency).  
Not shown: 3 closed tcp ports (reset)  
PORT      STATE SERVICE  
21/tcp    open  ftp  
22/tcp    open  ssh  
23/tcp    open  telnet  
25/tcp    open  smtp  
80/tcp    open  http  
139/tcp   open  netbios-ssn  
445/tcp   open  microsoft-ds  
MAC Address: 08:00:27:CC:EB:61 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)  
  
Nmap done: 1 IP address (1 host up) scanned in 0.16 seconds
```

Il comando ha restituito l'elenco delle porte **aperte** sulla macchina target, e i relativi **servizi** associati. Ogni porta aperta rappresenta una possibile **vulnerabilità** o un punto d'ingresso per un attacco.

5. nmap <ip target> -p- -sV --reason --dns-server <ns>

Sono state scansionate **tutte le porte TCP** (65535 porte) e **molte sono aperte**, indicando che il sistema è configurato in modo vulnerabile (tipico di Metasploitable2).

Servizi scoperti: Diversi servizi noti e vulnerabili sono in esecuzione su molte porte, tra cui **FTP, SSH, Telnet, HTTP, MySQL, VNC, NFS, RPC, PostgreSQL, e Samba**

```
(kali@kali)~$ nmap 192.168.50.101 -p- -sV --reason --dns-server ns
Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-01 12:30 EDT
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled. Try using --system-dns or specify valid servers with --dns-servers
Stats: 0:00:44 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 96.67% done; ETC: 12:31 (0:00:01 remaining)
Stats: 0:01:06 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 96.67% done; ETC: 12:32 (0:00:02 remaining)
Stats: 0:01:46 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 96.67% done; ETC: 12:32 (0:00:04 remaining)
Nmap scan report for 192.168.50.101
Host is up, received arp-response (0.000079s latency).
Not shown: 65505 closed tcp ports (reset)
PORT      STATE SERVICE      REASON      VERSION
21/tcp    open  ftp          syn-ack ttl 64 vsftpd 2.3.4
22/tcp    open  ssh          syn-ack ttl 64 OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp    open  telnet       syn-ack ttl 64 Linux telnetd
25/tcp    open  smtp         syn-ack ttl 64 Postfix smtpd
53/tcp    open  domain       syn-ack ttl 64 ISC BIND 9.4.2
80/tcp    open  http         syn-ack ttl 64 Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp   open  rpcbind      syn-ack ttl 64 2 (RPC #100000)
139/tcp   open  netbios-ssn  syn-ack ttl 64 Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp   open  netbios-ssn  syn-ack ttl 64 Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
512/tcp   open  exec         syn-ack ttl 64 netkit-rsh rexecd
513/tcp   open  login?       syn-ack ttl 64
514/tcp   open  shell        syn-ack ttl 64
1099/tcp  open  java-rmi     syn-ack ttl 64 GNU Classpath grmiregistry
1524/tcp  open  bindshell    syn-ack ttl 64 Metasploitable root shell
2049/tcp  open  nfs          syn-ack ttl 64 2-4 (RPC #100003)
2121/tcp  open  ftp          syn-ack ttl 64 ProFTPD 1.3.1
3306/tcp  open  mysql        syn-ack ttl 64 MySQL 5.0.51a-3ubuntu5
3632/tcp  open  distccd      syn-ack ttl 64 distccd v1 ((GNU) 4.2.4 (Ubuntu 4.2.4-1ubuntu4))
5432/tcp  open  postgresql   syn-ack ttl 64 PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp  open  vnc          syn-ack ttl 64 VNC (protocol 3.3)
6000/tcp  open  X11          syn-ack ttl 64 (access denied)
6667/tcp  open  irc          syn-ack ttl 64 UnrealIRCd
6697/tcp  open  irc          syn-ack ttl 64 UnrealIRCd
8009/tcp  open  ajp13        syn-ack ttl 64 Apache Jserv (Protocol v1.3)
8180/tcp  open  http         syn-ack ttl 64 Apache Tomcat/Coyote JSP engine 1.1
8787/tcp  open  drb          syn-ack ttl 64 Ruby DRb RMI (Ruby 1.8; path /usr/lib/ruby/1.8/drbb)
37755/tcp open  status       syn-ack ttl 64 1 (RPC #100024)
49756/tcp open  java-rmi     syn-ack ttl 64 GNU Classpath grmiregistry
50536/tcp open  mountd       syn-ack ttl 64 1-3 (RPC #100005)
58067/tcp open  nlockmgr     syn-ack ttl 64 1-4 (RPC #100021)
MAC Address: 08:00:27:CC:EB:61 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Service Info: Hosts: metasploitable.localdomain, irc.Metasploitable.LAN; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 150.58 seconds
```

L'output Nmap ti fornisce una mappa completa dei servizi vulnerabili su Metasploitable2.

ns: inserire DNS (8.8.8.8 ad esempio google) altrimenti nulla per usare un valore predefinito

6. us -mT -lv <ip target>:a -r 3000 -R 3

-mT: modalità **TCP scan**

-lv: interfaccia verbose (dettagli su ogni pacchetto)

192.168.50.101:a: target IP e range porte (tutte le porte da 0 a 65535)

-r 3000: velocità di scansione → 3000 pacchetti al secondo

-R 3: ripeti la scansione 3 volte (per ridurre falsi negativi)

```
(root@kali)-[/home/kali]
# us -mT -lv 192.168.50.101:a -r 3000 -R 3
adding 192.168.50.101/32 mode 'TCPscan' ports 'a' pps 3000
using interface(s) eth1
scanning 1.00e+00 total hosts with 1.97e+05 total packets, should take a little longer than 1 Minutes, 12 Seconds
TCP open 192.168.50.101:5900 ttl 64
TCP open 192.168.50.101:1524 ttl 64
TCP open 192.168.50.101:3632 ttl 64
TCP open 192.168.50.101:37755 ttl 64
TCP open 192.168.50.101:3306 ttl 64
TCP open 192.168.50.101:6697 ttl 64
TCP open 192.168.50.101:1099 ttl 64
TCP open 192.168.50.101:6667 ttl 64
TCP open 192.168.50.101:49756 ttl 64
TCP open 192.168.50.101:80 ttl 64
TCP open 192.168.50.101:2121 ttl 64
TCP open 192.168.50.101:8180 ttl 64
TCP open 192.168.50.101:6000 ttl 64
TCP open 192.168.50.101:22 ttl 64
TCP open 192.168.50.101:23 ttl 64
TCP open 192.168.50.101:512 ttl 64
TCP open 192.168.50.101:445 ttl 64
TCP open 192.168.50.101:53 ttl 64
TCP open 192.168.50.101:5432 ttl 64
TCP open 192.168.50.101:513 ttl 64
TCP open 192.168.50.101:8009 ttl 64
TCP open 192.168.50.101:514 ttl 64
TCP open 192.168.50.101:8787 ttl 64
TCP open 192.168.50.101:21 ttl 64
TCP open 192.168.50.101:58067 ttl 64
TCP open 192.168.50.101:50536 ttl 64
TCP open 192.168.50.101:2049 ttl 64
TCP open 192.168.50.101:139 ttl 64
TCP open 192.168.50.101:111 ttl 64
TCP open 192.168.50.101:25 ttl 64
sender statistics 2940.4 pps with 196608 packets sent total
listener statistics 196608 packets recieved 0 packets dropped and 0 interface drops
TCP open ftp[ 21] from 192.168.50.101 ttl 64
TCP open ssh[ 22] from 192.168.50.101 ttl 64
TCP open telnet[ 23] from 192.168.50.101 ttl 64
TCP open smtp[ 25] from 192.168.50.101 ttl 64
TCP open domain[ 53] from 192.168.50.101 ttl 64
TCP open http[ 80] from 192.168.50.101 ttl 64
TCP open sunrpc[ 111] from 192.168.50.101 ttl 64
TCP open netbios-ssn[ 139] from 192.168.50.101 ttl 64
TCP open microsoft-ds[ 445] from 192.168.50.101 ttl 64
TCP open exec[ 512] from 192.168.50.101 ttl 64
TCP open login[ 513] from 192.168.50.101 ttl 64
TCP open shell[ 514] from 192.168.50.101 ttl 64
TCP open rmiregistry[ 1099] from 192.168.50.101 ttl 64
TCP open ingreslock[ 1524] from 192.168.50.101 ttl 64
TCP open shilp[ 2049] from 192.168.50.101 ttl 64
TCP open scientia-ssdb[ 2121] from 192.168.50.101 ttl 64
```

Obiettivo: trovare tutte le **porte TCP aperte**.

Risultato: abbiamo trovato porte come 22 (SSH), 80 (HTTP), 3306 (MySQL), 5432 (PostgreSQL), 5900 (VNC), ecc.

```

TCP open      mysql[ 3306]      from 192.168.50.101  ttl 64
TCP open      distcc[ 3632]    from 192.168.50.101  ttl 64
TCP open      postgresql[ 5432] from 192.168.50.101  ttl 64
TCP open      winvnc[ 5900]    from 192.168.50.101  ttl 64
TCP open      x11[ 6000]      from 192.168.50.101  ttl 64
TCP open      irc[ 6667]      from 192.168.50.101  ttl 64
TCP open      unknown[ 6697]   from 192.168.50.101  ttl 64
TCP open      unknown[ 8009]   from 192.168.50.101  ttl 64
TCP open      unknown[ 8180]   from 192.168.50.101  ttl 64
TCP open      msgsrvr[ 8787]   from 192.168.50.101  ttl 64
TCP open      unknown[37755]   from 192.168.50.101  ttl 64
TCP open      unknown[49756]   from 192.168.50.101  ttl 64
TCP open      unknown[50536]   from 192.168.50.101  ttl 64
TCP open      unknown[58067]   from 192.168.50.101  ttl 64

```

6. `us -mU -lv <ip target>:a -r 3000 -R 3`

-mU: modalità **UDP scan**

Il resto dei parametri è uguale al comando TCP.

Obiettivo: trovare tutte le **porte UDP aperte**, che sono più difficili da scansionare perché:

UDP **non risponde** se la porta è chiusa → scansione più lenta e meno affidabile e molti firewall **filtrano il traffico UDP**

```

(root@kali)-[/home/kali]
# us -mU -lv 192.168.50.101:a -r 3000 -R 3
adding 192.168.50.101/32 mode 'UDPscan' ports 'a' pps 3000
using interface(s) eth1
scanning 1.00e+00 total hosts with 1.97e+05 total packets, should take a little longer than 1 Minutes, 12 Seconds
UDP open 192.168.50.101:2049  ttl 64
UDP open 192.168.50.101:111  ttl 64
UDP open 192.168.50.101:49155 ttl 64
UDP open 192.168.50.101:137  ttl 64
UDP open 192.168.50.101:53   ttl 64
UDP open 192.168.50.101:55491 ttl 64
UDP open 192.168.50.101:43894 ttl 64
sender statistics 2965.8 pps with 196635 packets sent total
listener statistics 21 packets recieved 0 packets dropped and 0 interface drops
UDP open      domain[ 53]      from 192.168.50.101  ttl 64
UDP open      sunrpc[ 111]     from 192.168.50.101  ttl 64
UDP open      netbios-ns[ 137]  from 192.168.50.101  ttl 64
UDP open      shilp[ 2049]     from 192.168.50.101  ttl 64
UDP open      unknown[43894]   from 192.168.50.101  ttl 64
UDP open      unknown[49155]   from 192.168.50.101  ttl 64
UDP open      unknown[55491]   from 192.168.50.101  ttl 64

```

Risultato: abbiamo trovato porte come 53 (DNS), 111 (RPC), 137 (NetBIOS), 2049 (NFS), ecc.

7. nmap -sS -sV -T4 <ip target>

Obiettivo:

Scoprire quali porte TCP sono aperte.

Scoprire quali servizi rispondono su queste porte.

Scoprire le **versioni dei software** per valutare potenziali vulnerabilità.

Farlo in modo veloce, grazie all'opzione **-T4**

```
(kali@kali)~$ nmap -sS -sV -T4 192.168.50.101
Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-02 06:16 EDT
Nmap scan report for 192.168.50.101
Host is up (0.00011s 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:CC:EB:61 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Service Info: Hosts: metasploitable.localdomain, irc.Metasploitable.LAN; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 52.48 seconds
```

Abbiamo scoperto che l'host 192.168.50.101:

È attivo (host up).

Ha **23 porte TCP aperte**, con molti servizi vulnerabili.

Sta eseguendo servizi vecchi, con versioni note per vulnerabilità critiche.

8. hping3 --scan known <ip target>

L'obiettivo è eseguire una **scansione delle porte TCP conosciute** (cioè le porte associate ai protocolli standard, come HTTP, SSH, FTP, ecc.) sull'host 192.168.50.101.

In particolare, hping3:

Invia pacchetti **TCP SYN** alle porte standard (es. 21, 22, 80, 443...).

Attende una risposta dal target.

Registra se le porte **rispondono** oppure **non rispondono**.

```
root@kali:~/home/kali# hping3 --scan known 192.168.50.101
Scanning 192.168.50.101 (192.168.50.101), port known
266 ports to scan, use -V to see all the replies
|-----|
|port|serv name| flags |ttl| id | win | len |
|-----|
All replies received. Done.
Not responding ports: (21 ftp) (22 ssh) (23 telnet) (25 smtp) (53 domain) (80 http) (111 sunrpc) (139 netbios-ssn) (445 microsoft-d) (512 exec) (513 login) (514 shell) (1099 rmiregistry) (1524 ingreslock) (2049 nfs) (2121 iprop) (3306 mysql) (3632 distcc) (5432 postgresql) (6000 x11) (6067 ircd) (6697 ircs-u)
```

Queste **porte "non rispondono"** ai pacchetti SYN che hping3 ha inviato.

In hping3, **nessuna risposta spesso significa che la porta è aperta o filtrata** (es. da un firewall).

Questo **non vuol dire che le porte siano chiuse** — anzi, molte di queste risultano **aperte** in Nmap.

hping non è solo in grado di inviare richieste di eco ICMP. Supporta anche i protocolli TCP, UDP, ICMP e RAW-IP, ha una modalità traceroute, la possibilità di inviare file tra canali coperti e molte altre funzionalità.

9. nc -nvz <ip target> 1-1024

Questo comando usa **Netcat (nc)** per effettuare una **scansione delle porte TCP da 1 a 1024** sul target 192.168.50.101, per verificare **quali sono aperte**.

Spiegazione degli argomenti:

-n=non risolvere i nomi DNS (più veloce)

-v=modalità **verbosa**, mostra i dettagli

-z= modalità **zero-I/O scan**, cioè non invia dati, serve solo per il port scanning

```
(kali㉿kali)-[~]  
$ nc -nvz 192.168.50.101 1-1024  
(UNKNOWN) [192.168.50.101] 514 (shell) open  
(UNKNOWN) [192.168.50.101] 513 (login) open  
(UNKNOWN) [192.168.50.101] 512 (exec) open  
(UNKNOWN) [192.168.50.101] 445 (microsoft-ds) open  
(UNKNOWN) [192.168.50.101] 139 (netbios-ssn) open  
(UNKNOWN) [192.168.50.101] 111 (sunrpc) open  
(UNKNOWN) [192.168.50.101] 80 (http) open  
(UNKNOWN) [192.168.50.101] 53 (domain) open  
(UNKNOWN) [192.168.50.101] 25 (smtp) open  
(UNKNOWN) [192.168.50.101] 23 (telnet) open  
(UNKNOWN) [192.168.50.101] 22 (ssh) open  
(UNKNOWN) [192.168.50.101] 21 (ftp) open
```

Risultati:

Port scan TCP semplice e rapido.

Utile per una verifica veloce, ma **non mostra i dettagli dei servizi** o versioni (come nmap -sV).

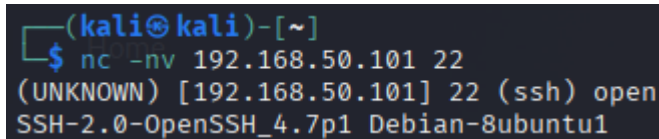
Può essere bloccato facilmente da firewall moderni (ma in un laboratorio come Metasploitable2 funziona benissimo).

10. nc -nv <ip target> 22

Questo comando serve a **connettersi manualmente alla porta 22 (SSH)** del target 192.168.50.101 usando Netcat (nc), in modo da vedere se:

La porta è **aperta**

Il servizio **risponde** e che **banner/versione espone**



```
(kali㉿kali)-[~]  
$ nc -nv 192.168.50.101 22  
(UNKNOWN) [192.168.50.101] 22 (ssh) open  
SSH-2.0-OpenSSH_4.7p1 Debian-8ubuntu1
```

La porta **22** è aperta.

Il server **SSH** ha risposto con il banner identificativo:

SSH-2.0-OpenSSH_4.7p1Debian-8ubuntu1

Usa il protocollo **SSH v2 (SSH-2.0)**

Esegue una vecchia versione di **OpenSSH (4.7p1)**

Basata su **Debian/Ubuntu**

11. nmap -sV <ip target>

L'opzione -sV dice a Nmap di:

effettuare il service/version detection, cioè identificare non solo **quali porte sono aperte**, ma anche **quali servizi le usano e in quale versione**.

Questo è **molto utile** perché ti permette di:

Trovare **versioni vulnerabili**

Identificare **servizi sospetti o backdoor**

Capire **quali exploit usare**

```
(kali@kali)-[~]
$ nmap -sV 192.168.50.101
Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-02 08:14 EDT
Nmap scan report for 192.168.50.101
Host is up (0.00019s 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:CC:EB:61 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Service Info: Hosts: metasploitable.localdomain, irc.Metasploitable.LAN; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 52.39 seconds
```

Nmap ha fornito:

Una mappa completa dei **servizi esposti**

Le versioni confermate di quei **servizi**

Spunti diretti per cercare exploit nei database come **Exploit-DB** o **MSF**

12. db_import <file.xml> (For Metasploit Framework)

Eeguire la scansione Nmap e salvare l'output in formato XML:

nmap -sV -oX metasploitable.xml 192.168.50.101 (-oX serve a salvare l'output in formato xml)

Questo genera un file chiamato **metasploitable.xml** sulla macchina Kali.

Avviare Metasploit:

msfconsole

Poi dentro Metasploit, **importiamo il file XML** che abbiamo appena creato:

db_import metasploitable.xml oppure scrivere tutto il path di dove si trova il file

in caso non riusciamo a connetterci al database dobbiamo usare il comando:

sudo systemctl start postgresql e poi **“status”** per verificare se attivo

```
msf6 > db_import metasploitable.xml
[*] Importing 'Nmap XML' data
[*] Import: Parsing with 'Nokogiri v1.13.10'
[*] Importing host 192.168.50.101
/usr/share/metasploit-framework/vendor/bundle/ruby/3.3.0/gems/recog-3.1.16/lib/recog/fingerprint/regexp_factory.rb:34: warning: nested repeat operator '+' and '?' was replaced with '*' in regular expression
[*] Successfully imported /home/kali/metasploitable.xml
msf6 > hosts

Hosts
=====

address      mac              name  os_name  os_flavor  os_sp  purpose  info  comments
-----
192.168.50.101 08:00:27:ccc:eb61 Linux         server

msf6 > services

Services
=====

host      port  proto  name      state  info
-----
192.168.50.101 21    tcp    ftp        open   vsftpd 2.3.4
192.168.50.101 22    tcp    ssh        open   OpenSSH 4.7p1 Debian Bubuntu protocol 2.0
192.168.50.101 23    tcp    telnet     open   Linux telnetd
192.168.50.101 25    tcp    smtp       open   Postfix smtpd
192.168.50.101 53    tcp    domain     open   ISC BIND 9.4.2
192.168.50.101 80    tcp    http       open   Apache httpd 2.2.8 (Ubuntu) DAV/2
192.168.50.101 111   tcp    rpcbind    open   2 RPC #100000
192.168.50.101 139   tcp    netbios-ssn open   Samba smbd 3.X - 4.X workgroup: WORKGROUP
192.168.50.101 445   tcp    netbios-ssn open   Samba smbd 3.X - 4.X workgroup: WORKGROUP
192.168.50.101 512   tcp    exec       open   netkit-rsh rshcd
192.168.50.101 513   tcp    login      open
192.168.50.101 514   tcp    shell      open   Netkit rshd
192.168.50.101 1099  tcp    java-rmi   open   GNU Classpath gswiregistry
192.168.50.101 1524  tcp    bindshell  open   Metasploitable root shell
192.168.50.101 2049  tcp    nfs        open   2-4 RPC #100003
192.168.50.101 2121  tcp    ftp        open   ProFTPD 1.3.11
192.168.50.101 3306  tcp    mysql      open   MySQL 5.0.51a-3ubuntu5
192.168.50.101 5432  tcp    postgresql open   PostgreSQL DB 8.3.0 - 8.3.7
192.168.50.101 5900  tcp    vnc        open   VNC protocol 3.3
192.168.50.101 6000  tcp    x11        open   access denied
192.168.50.101 6667  tcp    irc        open   UnrealIRCd
192.168.50.101 8009  tcp    ajp13      open   Apache Jserv Protocol v1.3
192.168.50.101 8180  tcp    http       open   Apache Tomcat/Coyote JSP engine 1.1

msf6 > search vsftpd 2.3.4

Matching Modules
=====

#  Name                                     Disclosure Date  Rank    Check  Description
--  -
0  exploit/unix/ftp/vsftpd_234_backdoor  2011-07-03      excellent No     VSFTPD v2.3.4 Backdoor Command Execution
```

Comandi usati: hosts e services

hosts:

Scopo: mostra gli host presenti nel database Metasploit.

Origine dei dati: importati da scansioni (es. nmap -oX file.xml + db_import).

Cosa mostra:

IP (address)

MAC address

Sistema operativo (os_name)

Scopo (es. server)

Nome host (se rilevato)

services:

Scopo: elenca i servizi attivi scoperti su ogni host.

Origine dei dati: da scansioni (es. Nmap con -sV) importate.

Cosa mostra:

Porta (port)

Protocollo (proto)

Nome del servizio (name)

Stato (state, tipicamente open)

Versione/dettagli (info)

13. nmap -f --mtu=512 <ip target>

il comando ha lo scopo di eseguire una **scansione "evasiva"** su Metasploitable2, utilizzando tecniche di **frammentazione dei pacchetti IP** per cercare di **eludere i sistemi IDS/IPS o firewall**.

-f: frammenta i pacchetti IP per renderli meno riconoscibili da firewall/IDS.

--mtu=512: imposta la dimensione del pacchetto a 512 byte (Multiple of 8); può influenzare come la frammentazione viene gestita.

Eludere i sistemi di difesa della rete che potrebbero bloccare o alterare le scansioni standard, inviando pacchetti frammentati che spesso passano inosservati.

```
(kali㉿kali)-[~]
$ nmap -f --mtu=512 192.168.50.101
Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-02 08:51 EDT
Nmap scan report for 192.168.50.101
Host is up (0.00029s 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:CC:EB:61 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Nmap done: 1 IP address (1 host up) scanned in 0.16 seconds
```

La macchina target (Metasploitable2) **non ha difese attive contro questa tecnica**.

Nmap ha identificato correttamente numerosi servizi su porte comuni: FTP, SSH, HTTP, SMB, NFS, MySQL, PostgreSQL, VNC, ecc.

14. masscan <network> -p80 --banners --source-ip <ip target>
<network>: è il range di IP da scansionare (es. 192.168.50.0/24)

-p80: esegue la scansione della **porta 80** (HTTP)

--banners: prova a **ottenere banner di servizio**, es. la versione del server web

--source-ip <IP>: **falsifica l'indirizzo IP sorgente** (spoofing)

Eeguire una **scansione veloce** su una rete bersaglio per la porta 80, **camuffando l'IP sorgente** e tentando di identificare i **banner HTTP** (versioni software, ecc.).

```
(root@kali)-[/home/kali]
# sudo masscan 192.168.50.101 -p80 --rate 1000 --banners --router-mac 08:00:27:ae:ce:1a
Starting masscan 1.3.2 (http://bit.ly/14GZzcT) at 2025-05-02 13:13:37 GMT
Initiating SYN Stealth Scan
Scanning 1 hosts [1 port/host]
^Cwaiting several seconds to exit...
^Cte: 0.00-kpps, 100.00% done, waiting -85-secs, found=0
```

abbiamo messo noi un MAC Address per risolvere i problemi con il protocollo ARP
(questo comando purtroppo non sono riuscito a farlo funzionare)

Informazioni riguardo il comando:

Questo è il più veloce scanner di porte Internet. Può scansionare l'intera rete Internet in meno di 6 minuti, trasmettendo 10 milioni di pacchetti al secondo.

Un'altra caratteristica di Masscan è che, oltre a rilevare porte aperte/chiuso, può anche acquisire semplici informazioni "banner". Il limite che deve affrontare è che Masscan dispone di un proprio stack TCP/IP.

Quando il sistema locale riceve un SYN-ACK dal target sondato, risponde con un pacchetto TST che interrompe la connessione prima che le informazioni banner possano essere acquisite. Il modo più semplice per evitare questo problema è assegnare a Masscan un indirizzo IP diverso:

```
Yeahhub.com
root@yeahhub:~# masscan 192.168.169.2/24 -p80 --banners --source-ip 192.168.169.150
Starting masscan 1.0.4 (http://bit.ly/14GZzcT) at 2018-10-17 18:56:26 GMT
-- forced options: -sS -Pn -n --randomize-hosts -v --send-eth
Initiating SYN Stealth Scan
Scanning 256 hosts [1 port/host]
Discovered open port 80/tcp on 192.168.169.138
rate: 0.00-kpps, 100.00% done, waiting -10-secs, found=1
```