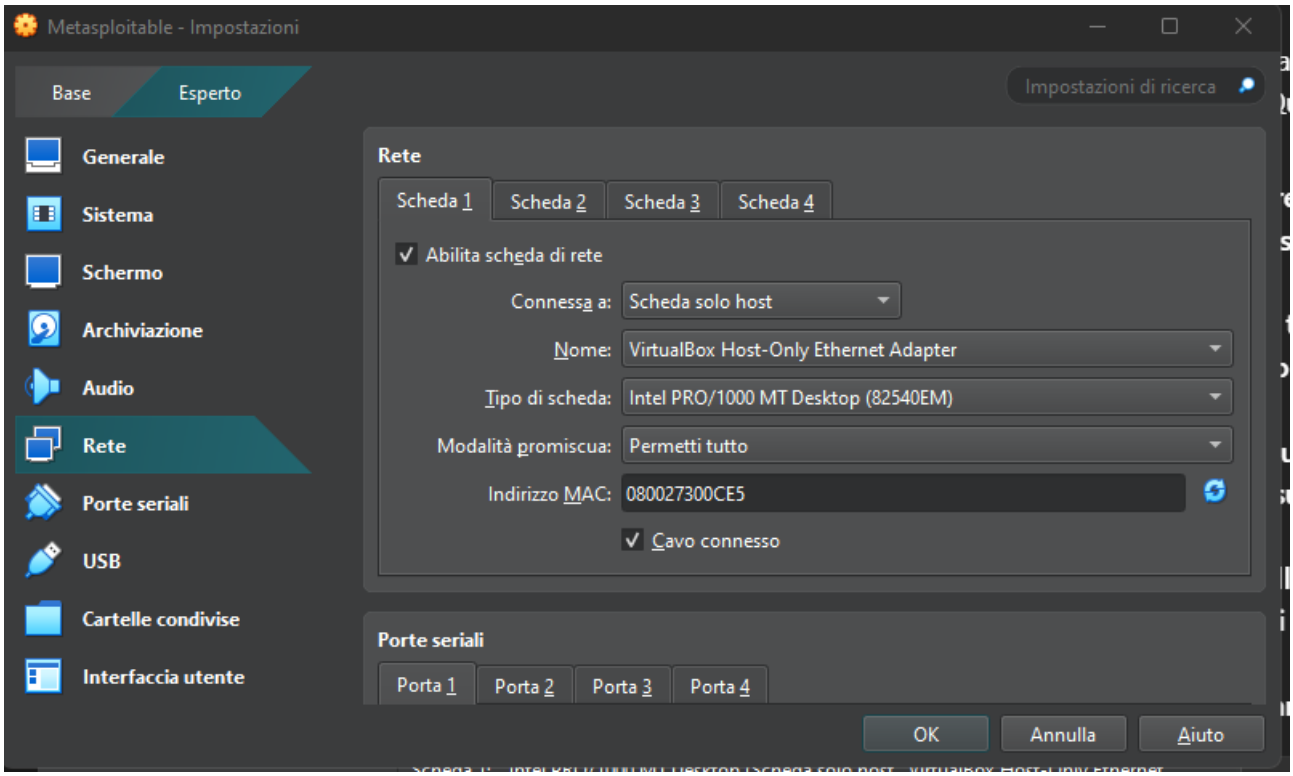
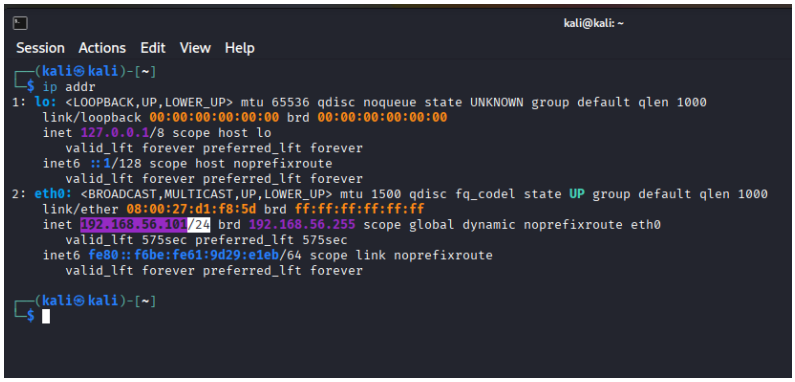
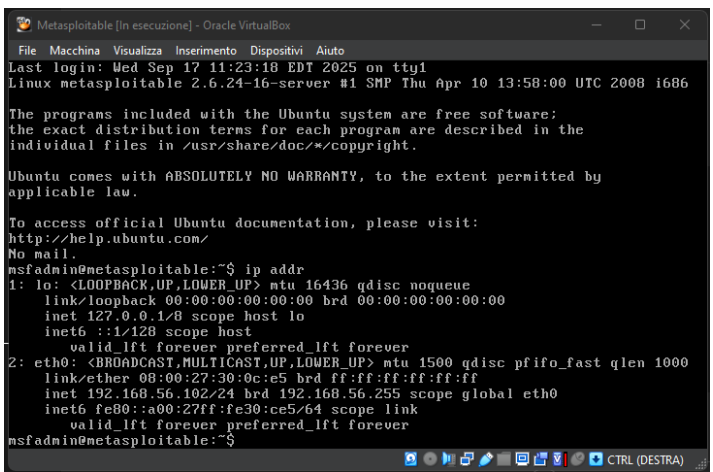
W10D4 – FRANCESCO MONTALTO

INTRO:
Per l'esercitazione ho configurato Kali (attaccante) e Metasploitable (target) sulla stessa rete "Host-Only". Tra i motivi, anche perché tra gli indirizzi IP sono fissi o assegnati da un DHCP interno alla rete virtuale, rendendo riproducibili i test. Inoltre, le comunicazioni rimangono interne all'host, riducendo latenza variabile e problemi di routing che potrebbero alterare i risultati delle scansioni.



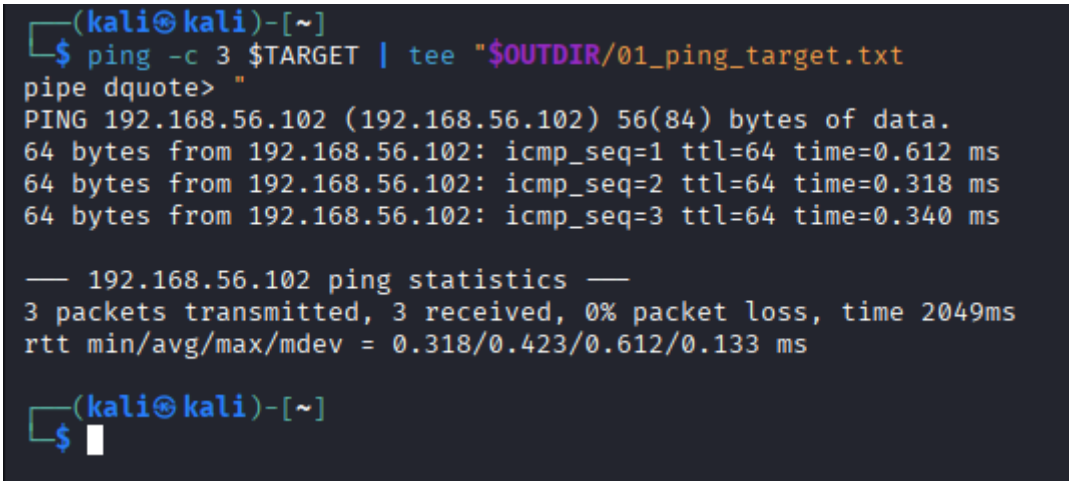
Ho verificato gli IP di entrambe le VM.





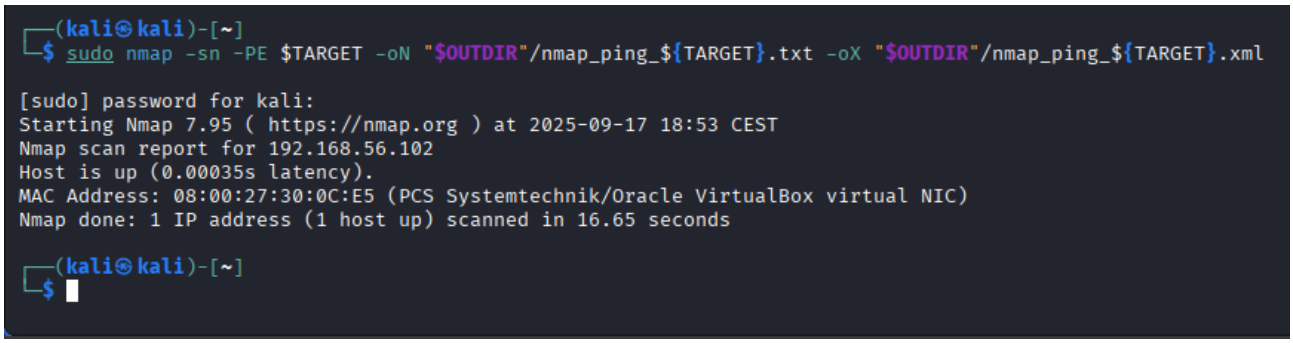
Ho centralizzato gli output ed usato variabili per non sbagliare IP, in modo anche da avere dei comandi più veloci. Ora siamo pronti ad utilizzare i 15 strumenti per la raccolta informazioni

Ho verificato le connettività.



COMANDI

1. nmap -sn -PE <target>



Ho eseguito una scansione di discovery con Nmap per verificare se l'host `192.168.56.102` è raggiungibile sulla rete. Il comando non cerca porte aperte: si limita a capire se la macchina è "up" e raccogliere qualche informazione di base (es. indirizzo MAC). Output salvati sia in formato testo che XML per documentazione.

Come ho strutturato il comando:

-sudo= esegue il comando con privilegi elevati; necessario per alcune tecniche di discovery e per ottenere informazioni a basso livello (es. MAC).

-nmap= lo scanner usato.

-sn = "no port scan": disabilita il controllo delle porte; Nmap esegue solo host discovery (ping/ARP ecc.).

-PE = usa ICMP Echo Request (ping standard) come metodo di discovery. Se il target risponde a ICMP, viene segnato come "up".

$TARGET = variabile che contiene l'indirizzo IP scansionato (qui 192.168.56.102).

-oN "$OUTDIR"/nmap_ping_${TARGET}.txt = salva l'output in formato leggibile (normal) nella cartella $OUTDIR con nome nmap_ping_192.168.56.102.txt.

-oX "$OUTDIR"/nmap_ping_${TARGET}.xml = salva lo stesso output in formato XML (utile per parsing automatico o report strutturati).

Output estratto:

-Nmap scan report for 192.168.56.102= host identificato.

-Host is up (0.00035s latency). = host risponde al ping, latenza molto bassa.

-MAC Address: 08:00:27:30:0C:E5 (PCS Systemtechnik/Oracle VirtualBox virtual NIC) = macchina in esecuzione su VirtualBox (Metasploitable probabile).

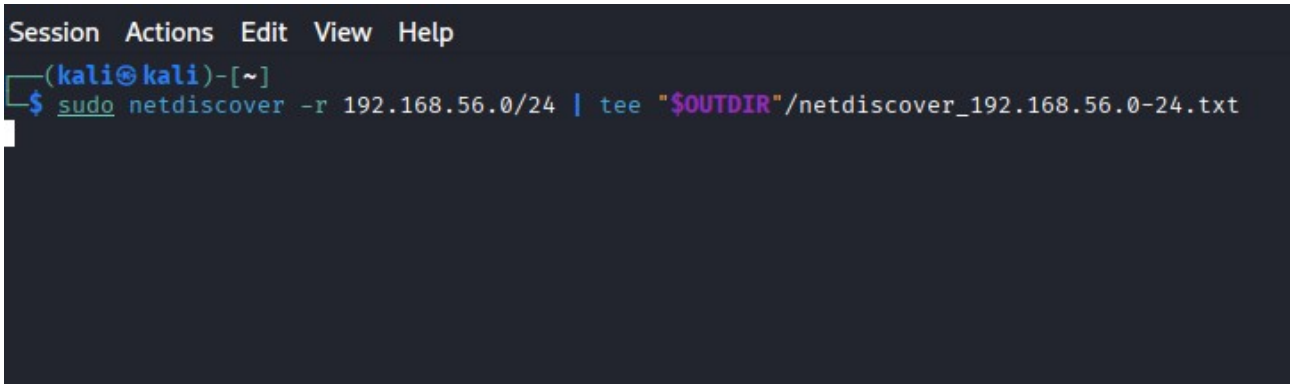
-Nmap done: 1 IP address (1 host up) scanned in 16.65 seconds = scansione

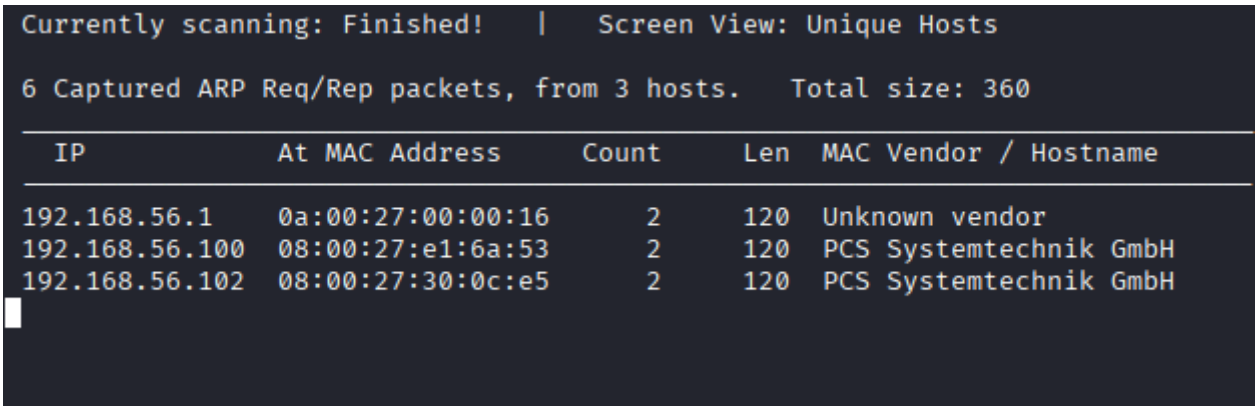
Considerazioni finali:

-L’host 192.168.56.102 è raggiungibile e risponde a ICMP Echo: sappiamo quindi che è attivo e presente sulla subnet.

-Poiché è stato usato solo discovery, non sappiamo ancora quali servizi o porte siano esposte: questa è solo la prima conferma di “esistenza” e posizione nella rete.

2. netdiscover -r <target>





Ho effettuato una scansione ARP della sottorete 192.168.56.0/24 con Netdiscover per individuare velocemente i dispositivi attivi sulla LAN. L'output è stato salvato con tee per conservare una copia testuale nel folder di output.

Come ho strutturato il comando:

-netdiscover = strumento automatico per discovery su reti locali basato su ARP (utile quando ICMP è bloccato o per rilevare VM).

- -r 192.168.56.0/24 = intervallo di rete da scansionare (da .1 a .254 nella subnet /24).

- | (pipe)= passa l'output di netdiscover al comando successivo.

- tee "$OUTDIR"/netdiscover_192.168.56.0-24.txt =salva l'output su file e lo mostra a schermo; $OUTDIR è la cartella di destinazione usata per i file del report.

Output estratto:

-Currently scanning: Finished! | Screen View: Unique Hosts = scansione completata.

-6 Captured ARP Req/Rep packets, from 3 hosts. = sono stati osservati 6 pacchetti ARP, provenienti da 3 host distinti.

Elenco host rilevati:

-192.168.56.1 — MAC 0a:00:27:00:00:16 = Vendor: Unknown vendor (probabile gateway o host di rete).

-192.168.56.100 — MAC 08:00:27:e1:6a:53 = Vendor: PCS Systemtechnik GmbH (Oracle VirtualBox virtual NIC).

-192.168.56.102 — MAC 08:00:27:30:0c:e5 = Vendor: PCS Systemtechnik GmbH (Oracle VirtualBox virtual NIC).

Considerazioni finali:

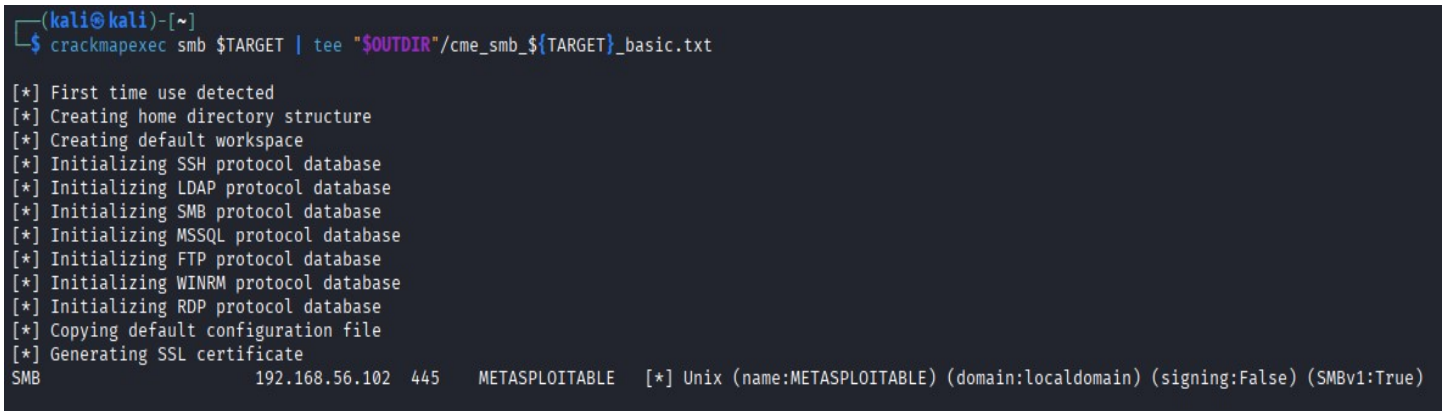
-Rilevate tre macchine nella subnet.

-192.168.56.102 è confermato come VM

-192.168.56.100 è la macchina dell'attaccante o un'altra VM utile per pivoting/test.

-192.168.56.1 potrebbe essere il gateway o un dispositivo di rete; merita verifica prima di lanciarci su exploit per evitare di testare dispositivi di infrastruttura non coinvolti.

3. crackmapexec <target>



Ho usato crackmapexec per un controllo rapido del servizio SMB sull'host 192.168.56.102. Lo strumento riporta se la porta SMB (445) è attiva, il nome della macchina, la presenza di SMBv1 e lo stato del signing. L'output è stato salvato con tee per documentare il risultato.

Come ho strutturato il comando:

-crackmapexec =tool di post-exploitation e discovery per protocolli di rete (SMB, SSH, etc.), usato per enumerazione rapida e test di autenticazione.

-smb =modulo di crackmapexec che si concentra sul protocollo SMB (porta 445).

-$TARGET =variabile che contiene l'indirizzo IP del bersaglio (qui 192.168.56.102).

-| (pipe) = inoltrare l'output al comando successivo.

-tee "$OUTDIR"/cme_smb_${TARGET}_basic.txt =salva una copia dell'output su file nella cartella di destinazione ($OUTDIR) e lo mostra a schermo.

Output estratto:

-Rilevata presenza del servizio=SMB 192.168.56.102 445 METASPLOITABLE.

-Nome host= METASPLOITABLE.

-Dominio= localdomain.

-signing: False = SMB signing non attivo (riduce integrità e protezione contro alcuni attacchi man-in-the-middle).

-SMBv1: True = supporto per SMBv1 abilitato

Considerazioni finali:

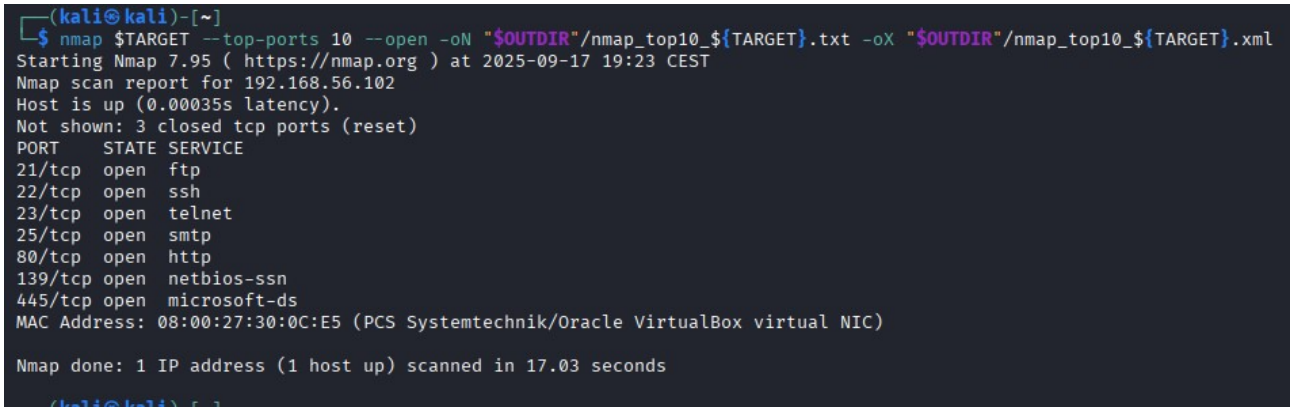
-Host con SMB attivo su porta 445=presenza di una superficie d’attacco significativa.

-Abilitazione di SMBv1 è un fattore di rischio elevato= SMBv1 è vulnerabile a exploit storici e facilmente sfruttabile in ambienti di laboratorio come Metasploitable.

-SMB signing disabilitato= implica che la comunicazione SMB non è protetta contro alcune manipolazioni; peggiora il profilo di sicurezza.

-Il nome METASPLOITABLE= conferma che si tratta della macchina target del laboratorio, quindi risultati coerenti con gli obiettivi dell’esercitazione.

4. nmap <target> –top-ports 10 –open



Ho eseguito una scansione rapida sulle 10 porte più usate per verificare quali servizi importanti siano esposti dall'host 192.168.56.102. L'output è stato salvato sia in formato testo leggibile che in XML per documentazione e analisi automatica.

Come ho strutturato il comando:

-nmap = lo scanner usato per sondare porte e servizi.

-$TARGET = variabile contenente l'IP bersaglio (qui 192.168.56.102).

- -top-ports 10 = scansiona le 10 porte più comuni (veloce, utile per uno snapshot iniziale).

- -open = mostra solo le porte che risultano aperte; nasconde quelle chiuse/filtrate per semplificare l'output.

-oN "$OUTDIR"/nmap_top10_${TARGET}.txt = salva l'output in formato leggibile (normal) nella cartella $OUTDIR.

-oX "$OUTDIR"/nmap_top10_${TARGET}.xml = salva lo stesso output in formato XML per eventuale parsing/report automatico.

Output estratto:

-Host is up (0.00035s latency). = host raggiungibile.

-Not shown: 3 closed tcp ports (reset) = alcune porte chiuse non mostrate.

-Porte aperte rilevate:

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: 08:00:27:30:0C:E5 (PCS Systemtechnik/Oracle VirtualBox virtual NIC) = host su VirtualBox.

Considerazioni finali:

-L’host espone numerosi servizi di rete classici= web (80), mail (25), SSH (22), FTP (21), Telnet (23) e servizi Windows/SMB (139/445).

-Presenza di Telnet (porta 23) è un campanello d’allarme=protocollo non cifrato e tipicamente usato in macchine con scarse protezioni ; facile obiettivo in laboratorio.

-SMB e NetBIOS aperti = confermano superfici d’attacco tipiche di Metasploitable (condivisioni, credenziali deboli, exploit noti).

-FTP e SMTP = possono esporre file o informazioni utili per escalation o ricognizione.

-Questo risultato conferma che l’host è ricco di vettori d’attacco = la priorità è enumerare ogni servizio per capire versioni, credenziali e condivisioni accessibili.

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

```
┌──(kali㉿kali)-[~]
└─$ nmap $TARGET -p- -sV --reason --dns-servers 8.8.8.8 -oN "$OUTDIR"/nmap_allports_sV_reason_${TARGET}.txt -oX "$OUTDIR"/nmap_allports_sV_reason_${TARGET}.
xml
Starting Nmap 7.95 ( https://nmap.org ) at 2025-09-17 19:25 CEST
Nmap scan report for 192.168.56.102
Host is up, received arp-response (0.00017s 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 Netkit rshd
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/drb)
47438/tcp open  java-rmi    syn-ack ttl 64 GNU Classpath grmiregistry
49315/tcp open  mountd      syn-ack ttl 64 1-3 (RPC #100005)
58361/tcp open  nlockmgr    syn-ack ttl 64 1-4 (RPC #100021)
58811/tcp open  status      syn-ack ttl 64 1 (RPC #100024)
MAC Address: 08:00:27:30:0C:E5 (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 146.02 seconds
```

Ho eseguito una scansione completa di tutte le porte (-p-) con rilevamento delle versioni (-sV) sul target 192.168.56.102. Ho chiesto a Nmap di mostrare la motivation dello stato di porta (--reason) e di usare 8.8.8.8 come DNS per eventuali reverse lookup. Output salvati in testo e XML per il report.

Come ho strutturato il comando:

-nmap = lo scanner.

-$TARGET = IP bersaglio (qui 192.168.56.102).

- -p- = scansiona tutte le 65535 porte TCP (non solo le top/quelle comuni).

- -sV = version detection: prova a identificare il servizio e la sua versione su ciascuna porta aperta.

- --reason = mostra il motivo per cui Nmap ha classificato lo stato di una porta (es. syn-ack, reset, ttl, ecc.).

- --dns-servers 8.8.8.8 = forza l’uso del server DNS specificato per risolvere nomi; utile per risultati coerenti di reverse lookup quando il resolver locale è inaffidabile.

- -oN ... -oX ... = salva rispettivamente in formato leggibile e in XML strutturato nella cartella $OUTDIR.

Output estratto:

Porte e servizi importanti rilevati (con versione quando disponibile):

-21/tcp = vsftpd 2.3.4 (FTP)

-22/tcp = OpenSSH 4.7p1 (SSH)

-23/tcp = telnetd (Telnet non cifrato)

-25/tcp = Postfix smtpd (SMTP)

-53/tcp = ISC BIND 9.4.2 (DNS)

-80/tcp = Apache httpd 2.2.8 (Ubuntu) DAV/2 (web)

-111/tcp = rpcbind (RPC)

-139/tcp e 445/tcp = Samba/Samba smbd 3.x - 4.x (NetBIOS/SMB)

-512/514/1099/1524 ecc. = netkit rshd/rexecd/login/shell, java-rmi, bindshell Metasploitable root shell (servizi remoti insicuri)

-2049/tcp = nfs

-3306/tcp = MySQL 5.0.51a

-5432/tcp = PostgreSQL DB 8.3.0 - 8.3.7

-5900/tcp = VNC (protocol possibly open)

-6667/6697 = UnrealIRCd (IRC server)

-8009/8180/8787/... = Tomcat / JRuby / Apache JServ / diverse app server (varie app web dinamiche)

-Molte altre porte RPC (mountd, nlockmgr, status, etc.) indicate come aperte = superficie estesa.

Info addizionali:

-MAC Address: 08:00:27:30:0C:E5 (Oracle VirtualBox virtual NIC) = VM confermata.

-Service Info: metasploitable.localdomain, inc.Metasploitable.LAN; Oss = Unix, Linux

-Tempo di scansione= 146 s (scansione completa e ricca di version detection).

Considerazioni finali:

L’host espone molti servizi obsoleti e insicuri (vsftpd 2.3.4, OpenSSH molto datato, Samba 3.x, MySQL 5.0, PostgreSQL 8.3, Apache 2.2.8, ecc.). Questi rappresentano vettori d’attacco noti e facilmente sfruttabili in ambiente di laboratorio come Metasploitable.

6. us -mT -Iv <target>:a -r 3000 -R 3 && us -mU -Iv <target>:a -r 3000 -R 3

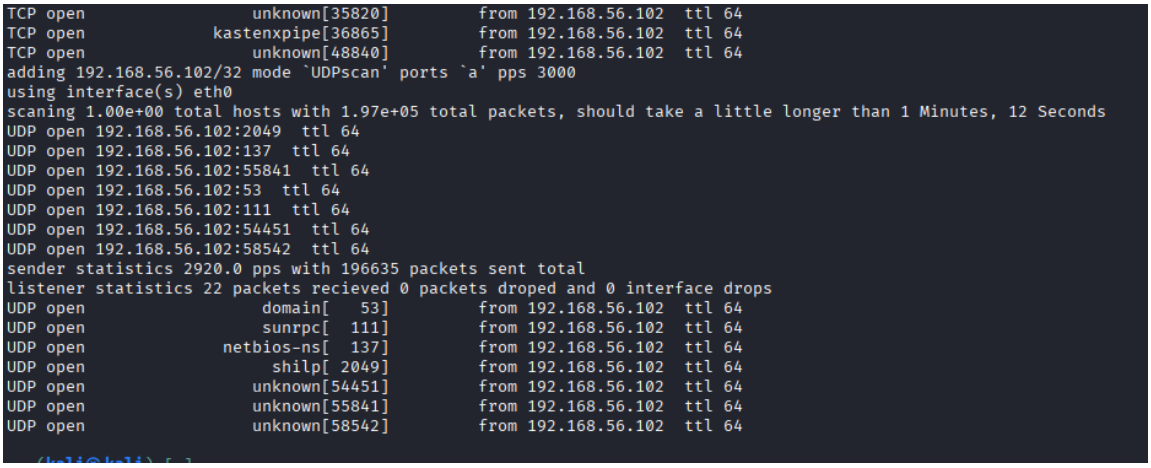
```
┌──(kali㉿kali)-[~]
└─$ sudo apt update && sudo apt install unicornscan -y

Ign:1 http://http.kali.org/kali kali-rolling InRelease
Ign:2 https://packages.microsoft.com/repos/code stable InRelease
Ign:1 http://http.kali.org/kali kali-rolling InRelease
Ign:2 https://packages.microsoft.com/repos/code stable InRelease
Ign:1 http://http.kali.org/kali kali-rolling InRelease
Ign:2 https://packages.microsoft.com/repos/code stable InRelease
0% [Working]
```

```
┌──(kali㉿kali)-[~]
└─$ sudo unicornscan -mT -Iv ${TARGET}:a -r 3000 -R 3 && sudo unicornscan -mU -Iv ${TARGET}:a -r 3000 -R 3

[sudo] password for kali:
adding 192.168.56.102/32 mode `TCPscan' ports `a' pps 3000
using interface(s) eth0
scaning 1.00e+00 total hosts with 1.97e+05 total packets, should take a little longer than 1 Minutes, 12 Seconds
TCP open 192.168.56.102:25  ttl 64
TCP open 192.168.56.102:8787  ttl 64
TCP open 192.168.56.102:3632  ttl 64
TCP open 192.168.56.102:5900  ttl 64
TCP open 192.168.56.102:514  ttl 64
TCP open 192.168.56.102:6697  ttl 64
TCP open 192.168.56.102:513  ttl 64
TCP open 192.168.56.102:22  ttl 64
TCP open 192.168.56.102:512  ttl 64
TCP open 192.168.56.102:3306  ttl 64
TCP open 192.168.56.102:1524  ttl 64
TCP open 192.168.56.102:8009  ttl 64
TCP open 192.168.56.102:36865  ttl 64
TCP open 192.168.56.102:35820  ttl 64
TCP open 192.168.56.102:48840  ttl 64
TCP open 192.168.56.102:2121  ttl 64
TCP open 192.168.56.102:2049  ttl 64
TCP open 192.168.56.102:53  ttl 64
TCP open 192.168.56.102:445  ttl 64
TCP open 192.168.56.102:6000  ttl 64
TCP open 192.168.56.102:80  ttl 64
TCP open 192.168.56.102:33384  ttl 64
TCP open 192.168.56.102:8180  ttl 64
TCP open 192.168.56.102:1099  ttl 64
TCP open 192.168.56.102:6667  ttl 64
TCP open 192.168.56.102:139  ttl 64
```

```
TCP open 192.168.56.102:5432  ttl 64
TCP open 192.168.56.102:111  ttl 64
TCP open 192.168.56.102:21  ttl 64
TCP open 192.168.56.102:23  ttl 64
sender statistics 2941.8 pps with 196608 packets sent total
listener statistics 196608 packets recieved 0 packets droped and 0 interface drops
TCP open                     ftp[   21]         from 192.168.56.102  ttl 64
TCP open                     ssh[   22]         from 192.168.56.102  ttl 64
TCP open                  telnet[   23]         from 192.168.56.102  ttl 64
TCP open                    smtp[   25]         from 192.168.56.102  ttl 64
TCP open                  domain[   53]         from 192.168.56.102  ttl 64
TCP open                    http[   80]         from 192.168.56.102  ttl 64
TCP open                  sunrpc[  111]         from 192.168.56.102  ttl 64
TCP open             netbios-ssn[  139]         from 192.168.56.102  ttl 64
TCP open            microsoft-ds[  445]         from 192.168.56.102  ttl 64
TCP open                    exec[  512]         from 192.168.56.102  ttl 64
TCP open                   login[  513]         from 192.168.56.102  ttl 64
TCP open                   shell[  514]         from 192.168.56.102  ttl 64
TCP open            rmiregistry[ 1099]         from 192.168.56.102  ttl 64
TCP open              ingreslock[ 1524]         from 192.168.56.102  ttl 64
TCP open                   shilp[ 2049]         from 192.168.56.102  ttl 64
TCP open           scientia-ssdb[ 2121]         from 192.168.56.102  ttl 64
TCP open                   mysql[ 3306]         from 192.168.56.102  ttl 64
TCP open                  distcc[ 3632]         from 192.168.56.102  ttl 64
TCP open              postgresql[ 5432]         from 192.168.56.102  ttl 64
TCP open                   winvnc[ 5900]         from 192.168.56.102  ttl 64
TCP open                      x11[ 6000]         from 192.168.56.102  ttl 64
TCP open                      irc[ 6667]         from 192.168.56.102  ttl 64
TCP open                  unknown[ 6697]         from 192.168.56.102  ttl 64
TCP open                  unknown[ 8009]         from 192.168.56.102  ttl 64
TCP open                  unknown[ 8180]         from 192.168.56.102  ttl 64
TCP open                  msgsrvr[ 8787]         from 192.168.56.102  ttl 64
TCP open                  unknown[33384]         from 192.168.56.102  ttl 64
```



Ho installato e usato Unicornscan come scansore ad alta velocità alternativo a Nmap per confermare l’elenco di porte aperte sul target 192.168.56.102. Ho lanciato prima la scansione TCP (mode T) e poi la UDP (mode U) per avere un secondo punto di vista sui servizi esposti. Unicornscan è più “aggressivo” e parallelo, quindi molto rapido ma anche più rumoroso**.**

Come ho strutturato il comando:

-unicornscan = il tool di scanning.

- -mT / -mU = modalità TCP o UDP.

- -I = modalità non interattiva / informazioni di base (qui usato con -v per verbose).

- -v = output verbose (più dettagli a schermo).

-${TARGET}:a - a = tutte le porte (scan “all”).

- -r 3000 = rate di invio 3000 pacchetti per secondo (alta velocità).

- -R 3 = ripete il probe fino a 3 volte per affidabilità.

Output estratto:

-TCP (alcune delle porte riportate):

21 (ftp)

22 (ssh)

23 (telnet)

25 (smtp)

53 (domain)

80 (http)

111 (rpcbind)

139 (netbios-ssn)

445 (microsoft-ds / SMB)

512/513/514 (rexec/rlogin/rsh/login/shell family)

1099 (java-rmi)

1524 (ingreslock / metasp root shell)

2049 (nfs)

3306 (mysql)

5432 (postgresql)

5900 (vnc)

6667 (irc), 6697 (irc SSL)

8009, 8180, 8787, 33384, 35820, 36865, 48840 (app/web servers / unknown services)

UDP (estratto):

53 (domain)

111 (sunrpc)

137 (netbios-ns)

2049 (shilp / spesso NFS-related)

-porte UDP alte= 54451, 55841, 58542 (mostrate come unknown)

-Statistiche= alto throughput (2900 pps) e grandi volumi di pacchetti inviati/ricevuti; TTL 64 tipico di host Linux.

Considerazioni finali:

-I risultati confermano e ampliano quanto trovato con Nmap= la macchina espone molti servizi, inclusi servizi obsoleti e non cifrati (telnet, rsh/rexec, servizi RPC, SMB, MySQL/Postgres datati).

-Unicornscan ha rilevato anche molte porte applicative “non standard” (high ports) che possono nascondere pannelli o backdoor presenti nella VM Metasploitable.

-Le porte UDP rilevate (53, 111, 137, 2049 e altre high-ports) sono importanti perché molte vulnerabilità su servizi UDP sono meno visibili con scan TCP standard.

-L’uso di un tool ad alta velocità conferma che non si tratta di falsi positivi dovuti a scansioni lente: la superficie è realmente estesa.

-In un ambiente reale questa entità di servizi rappresenterebbe una criticità molto alta; qui è attesa, ma utile per esercitazione.

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

```
┌──(kali㉿kali)-[~]
└─$ nmap -sS -sV -T4 $TARGET -oN "$OUTDIR"/nmap_sS_sV_T4_${TARGET}.txt -oX "$OUTDIR"/nmap_sS_sV_T4_${TARGET}.xml
Starting Nmap 7.95 ( https://nmap.org ) at 2025-09-17 19:53 CEST
Nmap scan report for 192.168.56.102
Host is up (0.00020s 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       OpenBSD or Solaris rlogind
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: 08:00:27:30:0C:E5 (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 29.26 seconds
```

Ho lanciato una scansione SYN (stealth) con rilevamento versione e timing accelerato (T4) su 192.168.56.102. L’obiettivo è ottenere rapidamente un elenco affidabile dei servizi con le versioni rilevate per indirizzare le fasi successive di enumerazione e ricerca exploit.

Come ho strutturato il comando:

-nmap = lo scanner di rete usato.

-$TARGET = variabile contenente l’IP bersaglio (qui 192.168.56.102).

- -sS = SYN scan: invia SYN e interpreta le risposte (syn-ack = open). Richiede solitamente privilegi per inviare pacchetti raw.

- -sV = version detection: tenta di identificare il servizio e la versione su ogni porta aperta.

- -T4 = timing template 4 (aggressivo): velocizza lo scan ma aumenta rumore e probabilità di essere rilevati.

- -oN ... -oX ... = salva output in formato leggibile e XML nella cartella $OUTDIR.

Output estratto:

-Host up = scansione completata in 29.26 secondi.

-Porte aperte e versioni principali (da screenshot):

21/tcp = vsftpd 2.3.4

22/tcp = OpenSSH 4.7p1

23/tcp = telnetd (Linux telnetd)

25/tcp = Postfix smtpd

53/tcp = ISC BIND 9.4.2

80/tcp = Apache httpd 2.2.8 (Ubuntu) DAV/2

111/tcp = rpcbind 2

139/tcp e 445/tcp = Samba smbd 3.x - 4.x (workgroup: WORKGROUP)

512/513/514 = netkit-rsh/rexecd/login/shell family (servizi remoti non cifrati)

1099/tcp = java-rmi (classpath / rmiregistry)

1524/tcp =bindshell / Metasploitable root shell

2049/tcp = nfs (RPC)

2121/ftp = ProFTPD 1.3.1 (altro ftp)

3306/tcp = MySQL 5.0.51a-3ubuntu5

5432/tcp = PostgreSQL DB 8.3.0 - 8.3.7

5900/tcp =VNC (protocol 3.3)

6667/tcp = UnrealIRCd

8009/8180/8787/... = Tomcat/Jserv/varie app server (diverse applicazioni web)

Considerazioni finali:

L’host espone numerosi servizi datati e vulnerabili: presenza di servizi come vsftpd 2.3.4, Samba 3.x, MySQL 5.0, Apache 2.2.8 e servizi di autenticazione non cifrati (telnet, rsh, rexec) è tipica di Metasploitable e indica un’alta probabilità di exploit riusciti in ambiente di test.

La superficie d’attacco è ampia; in un contesto reale questo equivale a rischio critico.

8. hping3 –scan known <target

```
┌──(kali㉿kali)-[~]
└─$ sudo hping3 --scan known -S -V $TARGET 2>&1 | tee "$OUTDIR"/hping3_scan_known_${TARGET}.txt
Scanning 192.168.56.102 (192.168.56.102), port known
266 ports to scan, use -V to see all the replies
+----+----------+--------+---+-----+-----+-----+
|port| serv name |  flags |ttl| id  | win | len |
+----+----------+--------+---+-----+-----+-----+
using eth0, addr: 192.168.56.101, MTU: 1500
    1 tcpmux     : ..R.A ...   64     0     0    46
    2 nbp        : ..R.A ...   64     0     0    46
    4 echo       : ..R.A ...   64     0     0    46
    6 zip        : ..R.A ...   64     0     0    46
    7 echo       : ..R.A ...   64     0     0    46
    9 discard    : ..R.A ...   64     0     0    46
   11 systat     : ..R.A ...   64     0     0    46
   13 daytime    : ..R.A ...   64     0     0    46
   15 netstat    : ..R.A ...   64     0     0    46
   17 qotd       : ..R.A ...   64     0     0    46
   19 chargen    : ..R.A ...   64     0     0    46
   20 ftp-data   : ..R.A ...   64     0     0    46
   21 ftp        : .S..A ...   64     0  5840    46
   22 ssh        : .S..A ...   64     0  5840    46
   23 telnet     : .S..A ...   64     0  5840    46
   25 smtp       : .S..A ...   64     0  5840    46
   37 time       : ..R.A ...   64     0     0    46
   43 whois      : ..R.A ...   64     0     0    46
   49 tacacs     : ..R.A ...   64     0     0    46
   53 domain     : .S..A ...   64     0  5840    46
   67 bootps     : ..R.A ...   64     0     0    46
   68 bootpc     : ..R.A ...   64     0     0    46
   69 tftp       : ..R.A ...   64     0     0    46
   70 gopher     : ..R.A ...   64     0     0    46
   79 finger     : ..R.A ...   64     0     0    46
```

```
   80 http       : .S..A ...   64     0  5840    46
   88 kerberos   : ..R.A ...   64     0     0    46
  102 iso-tsap   : ..R.A ...   64     0     0    46
  104 acr-nema   : ..R.A ...   64     0     0    46
  106 poppassd   : ..R.A ...   64     0     0    46
  110 pop3       : ..R.A ...   64     0     0    46
  111 sunrpc     : .S..A ...   64     0  5840    46
  113 auth       : ..R.A ...   64     0     0    46
  119 nntp       : ..R.A ...   64     0     0    46
  123 ntp        : ..R.A ...   64     0     0    46
  135 epmap      : ..R.A ...   64     0     0    46
  137 netbios-ns : ..R.A ...   64     0     0    46
  138 netbios-dgm: ..R.A ...   64     0     0    46
  139 netbios-ssn: .S..A ...   64     0  5840    46
  143 imap2      : ..R.A ...   64     0     0    46
  161 snmp       : ..R.A ...   64     0     0    46
  162 snmp-trap  : ..R.A ...   64     0     0    46
  163 cmip-man   : ..R.A ...   64     0     0    46
  164 cmip-agent : ..R.A ...   64     0     0    46
  174 mailq      : ..R.A ...   64     0     0    46
  177 xdmcp      : ..R.A ...   64     0     0    46
  179 bgp        : ..R.A ...   64     0     0    46
  199 smux       : ..R.A ...   64     0     0    46
  209 qmtp       : ..R.A ...   64     0     0    46
  210 z3950      : ..R.A ...   64     0     0    46
  213 ipx        : ..R.A ...   64     0     0    46
  319 ptp-event  : ..R.A ...   64     0     0    46
  320 ptp-general: ..R.A ...   64     0     0    46
  345 pawserv    : ..R.A ...   64     0     0    46
  346 zserv      : ..R.A ...   64     0     0    46
  369 rpc2portmap: ..R.A ...   64     0     0    46
```

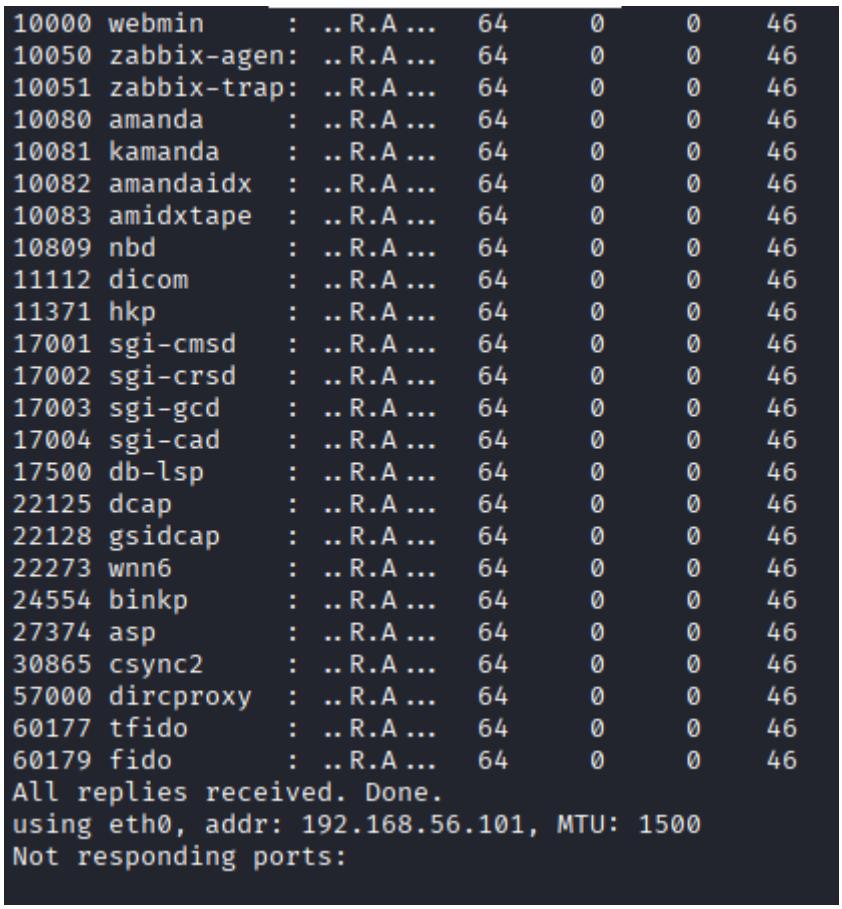
```
 370 codaauth2  : ..R.A...   64     0     0    46
 371 clearcase  : ..R.A...   64     0     0    46
 389 ldap       : ..R.A...   64     0     0    46
 427 svrloc     : ..R.A...   64     0     0    46
 443 https      : ..R.A...   64     0     0    46
 444 snpp       : ..R.A...   64     0     0    46
 445 microsoft-d: .S..A...   64     0  5840    46
 464 kpasswd    : ..R.A...   64     0     0    46
 465 submissions: ..R.A...   64     0     0    46
 487 saft       : ..R.A...   64     0     0    46
 500 isakmp     : ..R.A...   64     0     0    46
 512 exec       : .S..A...   64     0  5840    46
 513 login      : .S..A...   64     0  5840    46
 514 shell      : .S..A...   64     0  5840    46
 515 printer    : ..R.A...   64     0     0    46
 517 talk       : ..R.A...   64     0     0    46
 518 ntalk      : ..R.A...   64     0     0    46
 520 route      : ..R.A...   64     0     0    46
 538 gdomap     : ..R.A...   64     0     0    46
 540 uucp       : ..R.A...   64     0     0    46
 543 klogin     : ..R.A...   64     0     0    46
 544 kshell     : ..R.A...   64     0     0    46
 546 dhcpv6-clie: ..R.A...   64     0     0    46
 547 dhcpv6-serv: ..R.A...   64     0     0    46
 548 afpovertcp : ..R.A...   64     0     0    46
 554 rtsp       : ..R.A...   64     0     0    46
 563 nntps      : ..R.A...   64     0     0    46
 587 submission : ..R.A...   64     0     0    46
 607 nqs        : ..R.A...   64     0     0    46
 623 asf-rmcp   : ..R.A...   64     0     0    46
 628 qmqp       : ..R.A...   64     0     0    46
 631 ipp        : ..R.A...   64     0     0    46
```

```
 636 ldaps      : ..R.A...   64     0     0    46
 646 ldp        : ..R.A...   64     0     0    46
 655 tinc       : ..R.A...   64     0     0    46
 706 silc       : ..R.A...   64     0     0    46
 749 kerberos-ad: ..R.A...   64     0     0    46
 750 kerberos4  : ..R.A...   64     0     0    46
 751 kerberos-ma: ..R.A...   64     0     0    46
 752 passwd-serv: ..R.A...   64     0     0    46
 754 krb-prop   : ..R.A...   64     0     0    46
 775 moira-db   : ..R.A...   64     0     0    46
 777 moira-updat: ..R.A...   64     0     0    46
 779 moira-ureg : ..R.A...   64     0     0    46
 783 spamd      : ..R.A...   64     0     0    46
 853 domain-s   : ..R.A...   64     0     0    46
 871 supfilesrv : ..R.A...   64     0     0    46
 873 rsync      : ..R.A...   64     0     0    46
 989 ftps-data  : ..R.A...   64     0     0    46
 990 ftps       : ..R.A...   64     0     0    46
 992 telnets    : ..R.A...   64     0     0    46
 993 imaps      : ..R.A...   64     0     0    46
 995 pop3s      : ..R.A...   64     0     0    46
1080 socks      : ..R.A...   64     0     0    46
1093 proofd     : ..R.A...   64     0     0    46
1094 rootd      : ..R.A...   64     0     0    46
1099 rmiregistry: .S..A...   64     0  5840    46
1127 supfiledbg : ..R.A...   64     0     0    46
1178 skkserv    : ..R.A...   64     0     0    46
1194 openvpn    : ..R.A...   64     0     0    46
1210 predict    : ..R.A...   64     0     0    46
```

```
1236 rmtcfg      : ..R.A...   64     0     0    46
1313 xtel        : ..R.A...   64     0     0    46
1314 xtelw       : ..R.A...   64     0     0    46
1352 lotusnote   : ..R.A...   64     0     0    46
1433 ms-sql-s    : ..R.A...   64     0     0    46
1434 ms-sql-m    : ..R.A...   64     0     0    46
1524 ingreslock  : .S..A...   64     0  5840    46
1645 datametrics:  ..R.A...   64     0     0    46
1646 sa-msg-port:  ..R.A...   64     0     0    46
1649 kermit      : ..R.A...   64     0     0    46
1677 groupwise   : ..R.A...   64     0     0    46
1701 l2f         : ..R.A...   64     0     0    46
1812 radius      : ..R.A...   64     0     0    46
1813 radius-acct:  ..R.A...   64     0     0    46
2000 cisco-sccp  : ..R.A...   64     0     0    46
2049 nfs         : .S..A...   64     0  5840    46
2086 gnunet      : ..R.A...   64     0     0    46
2101 rtcm-sc104  : ..R.A...   64     0     0    46
2102 zephyr-srv  : ..R.A...   64     0     0    46
2103 zephyr-clt  : ..R.A...   64     0     0    46
2104 zephyr-hm   : ..R.A...   64     0     0    46
2119 gsigatekeep:  ..R.A...   64     0     0    46
2121 iprop       : .S..A...   64     0  5840    46
2135 gris        : ..R.A...   64     0     0    46
2401 cvspserver  : ..R.A...   64     0     0    46
2430 venus       : ..R.A...   64     0     0    46
2431 venus-se    : ..R.A...   64     0     0    46
2432 codasrv     : ..R.A...   64     0     0    46
2433 codasrv-se  : ..R.A...   64     0     0    46
```

```
2583 mon         : ..R.A...   64     0     0    46
2600 zebrasrv    : ..R.A...   64     0     0    46
2601 zebra       : ..R.A...   64     0     0    46
2602 ripd        : ..R.A...   64     0     0    46
2603 ripngd      : ..R.A...   64     0     0    46
2604 ospfd       : ..R.A...   64     0     0    46
2605 bgpd        : ..R.A...   64     0     0    46
3689 daap        : ..R.A...   64     0     0    46
2606 ospf6d      : ..R.A...   64     0     0    46
2607 ospfapi     : ..R.A...   64     0     0    46
2608 isisd       : ..R.A...   64     0     0    46
2628 dict        : ..R.A...   64     0     0    46
2792 f5-globalsi:  ..R.A...   64     0     0    46
2811 gsiftp      : ..R.A...   64     0     0    46
2947 gpsd        : ..R.A...   64     0     0    46
3050 gds-db      : ..R.A...   64     0     0    46
3130 icpv2       : ..R.A...   64     0     0    46
3205 isns        : ..R.A...   64     0     0    46
3260 iscsi-targe:  ..R.A...   64     0     0    46
3306 mysql       : .S..A...   64     0  5840    46
3389 ms-wbt-serv:  ..R.A...   64     0     0    46
3493 nut         : ..R.A...   64     0     0    46
3632 distcc      : .S..A...   64     0  5840    46
3690 svn         : ..R.A...   64     0     0    46
4031 suucp       : ..R.A...   64     0     0    46
4094 sysrqd      : ..R.A...   64     0     0    46
4190 sieve       : ..R.A...   64     0     0    46
4353 f5-iquery   : ..R.A...   64     0     0    46
4369 epmd        : ..R.A...   64     0     0    46
4373 remctl      : ..R.A...   64     0     0    46
4460 ntske       : ..R.A...   64     0     0    46
```

```
4500 ipsec-nat-t: ..R.A...   64     0     0    46
4557 fax        : ..R.A...   64     0     0    46
4559 hylafax    : ..R.A...   64     0     0    46
4569 iax        : ..R.A...   64     0     0    46
4691 mtn        : ..R.A...   64     0     0    46
4899 radmin-port: ..R.A...   64     0     0    46
4949 munin      : ..R.A...   64     0     0    46
5060 sip        : ..R.A...   64     0     0    46
5061 sip-tls    : ..R.A...   64     0     0    46
5222 xmpp-client: ..R.A...   64     0     0    46
5269 xmpp-server: ..R.A...   64     0     0    46
5308 cfengine   : ..R.A...   64     0     0    46
5353 mdns       : ..R.A...   64     0     0    46
5432 postgresql : .S..A...   64     0  5840    46
5555 rplay      : ..R.A...   64     0     0    46
5556 freeciv    : ..R.A...   64     0     0    46
5666 nrpe       : ..R.A...   64     0     0    46
5667 nsca       : ..R.A...   64     0     0    46
5671 amqps      : ..R.A...   64     0     0    46
5672 amqp       : ..R.A...   64     0     0    46
5680 canna      : ..R.A...   64     0     0    46
5683 coap       : ..R.A...   64     0     0    46
5684 coaps      : ..R.A...   64     0     0    46
6000 x11        : .S..A...   64     0  5840    46
6001 x11-1      : ..R.A...   64     0     0    46
6002 x11-2      : ..R.A...   64     0     0    46
6003 x11-3      : ..R.A...   64     0     0    46
6004 x11-4      : ..R.A...   64     0     0    46
6005 x11-5      : ..R.A...   64     0     0    46
6006 x11-6      : ..R.A...   64     0     0    46
6007 x11-7      : ..R.A...   64     0     0    46
6346 gnutella-sv: ..R.A...   64     0     0    46
6347 gnutella-rt: ..R.A...   64     0     0    46
6379 redis      : ..R.A...   64     0     0    46
6444 sge-qmaster: ..R.A...   64     0     0    46
6445 sge-execd  : ..R.A...   64     0     0    46
6446 mysql-proxy: ..R.A...   64     0     0    46
6514 syslog-tls : ..R.A...   64     0     0    46
6566 sane-port  : ..R.A...   64     0     0    46
6667 ircd       : .S..A...   64     0  5840    46
6696 babel      : ..R.A...   64     0     0    46
6697 ircs-u     : .S..A...   64     0  5840    46
7000 bbs        : ..R.A...   64     0     0    46
7001 afs3-callba: ..R.A...   64     0     0    46
7002 afs3-prserv: ..R.A...   64     0     0    46
7003 afs3-vlserv: ..R.A...   64     0     0    46
7004 afs3-kaserv: ..R.A...   64     0     0    46
7005 afs3-volser: ..R.A...   64     0     0    46
7007 afs3-bos   : ..R.A...   64     0     0    46
7008 afs3-update: ..R.A...   64     0     0    46
7009 afs3-rmtsys: ..R.A...   64     0     0    46
7100 font-servic: ..R.A...   64     0     0    46
8021 zope-ftp   : ..R.A...   64     0     0    46
8080 http-alt   : ..R.A...   64     0     0    46
8081 tproxy     : ..R.A...   64     0     0    46
8088 omniorb    : ..R.A...   64     0     0    46
8140 puppet     : ..R.A...   64     0     0    46
8990 clc-build-d: ..R.A...   64     0     0    46
9098 xinetd     : ..R.A...   64     0     0    46
9101 bacula-dir : ..R.A...   64     0     0    46
9102 bacula-fd  : ..R.A...   64     0     0    46
9103 bacula-sd  : ..R.A...   64     0     0    46
9418 git        : ..R.A...   64     0     0    46
9667 xmms2      : ..R.A...   64     0     0    46
9673 zope       : ..R.A...   64     0     0    46
```



Ho lanciato una scansione SYN con hping3 sulle porte “note” del target 192.168.56.102. Lo scopo era verificare la risposta TCP del sistema usando un approccio di packet-crafting diverso da Nmap/Unicornscan, per confermare i servizi realmente raggiungibili e il tipo di reply (utile per capire se la porta è open, closed o filtrata).

Come ho strutturato il comando:

-hping3 = tool di packet-crafting e scanning (più manuale e “basso livello” rispetto a Nmap).

- --scan known = usa una lista predefinita di porte “note/standard” (quick scan su porte comuni anziché tutte le 65k).

- -S = invia pacchetti con il flag TCP SYN (simula l’inizio di una connessione, come nel SYN scan).

- -V = verbose: mostra informazioni estese su ogni risposta ricevuta.

-$TARGET = IP bersaglio (192.168.56.102).

-2>&1 = redirige lo stderr nello stdout (così tutto viene catturato):
Ogni processo Unix ha tre file descriptor standard:
0 = stdin (input dall’utente o da un file)
1 = stdout (output “normale”, quello che si vede sullo schermo)
2 = stderr (messaggi d’errore / diagnostici, separati da stdout)
Separarli è utile perché gli errori non si mischiano ai dati “puliti”. Ma quando si vuole catturare tutto li si deve unire.
Ho così appreso che:

> : reindirizza stdout in un file (sovrascrive).
cmd > out.txt = solo stdout va in out.txt; gli errori rimangono sul terminale.

2> : reindirizza stderr in un file.
cmd 2> err.txt = solo errori vanno in err.txt.

2>&1 = duplica stderr verso dove punta stdout in quel momento.

- | tee "$OUTDIR"/hping3_scan_known_${TARGET}.txt = salva l’output su file e lo mostra a schermo.

Output estratto:

-Riga iniziale: using eth0, addr: 192.168.56.101, MTU: 1500 = interfaccia/sorgente usata per lo scan.

-Tabella delle porte con colonne tipo: |port|serv_name|flags|ttl|id|win|len| = mostra il comportamento delle risposte per ogni porta.

-Porte che hanno risposto con flag indicanti apertura o risposta utile (esempi rilevanti dal log):
21 ftp = risposta (SYN = SYN/ACK tipico)
22 ssh
23 telnet
25 smtp
53 domain
80 http
111 rpcbind
139 netbios-ssn / 445 microsoft-ds (SMB)
512/513/514 (rsh/login/rexec family)
1099 java-rmi
1524 (bindshell/Metasploitable indicator)
2049 nfs
3306 mysql = segnalata come rispondente
5432 postgresql
5900 vnc
6667/6697 irc
numerose high-ports/servizi applicativi (8009, 8180, 8787, 33384, 35820, 36865, 48840, 54451, 55841, 58542 ecc.) riportate come “open/rispondenti”.

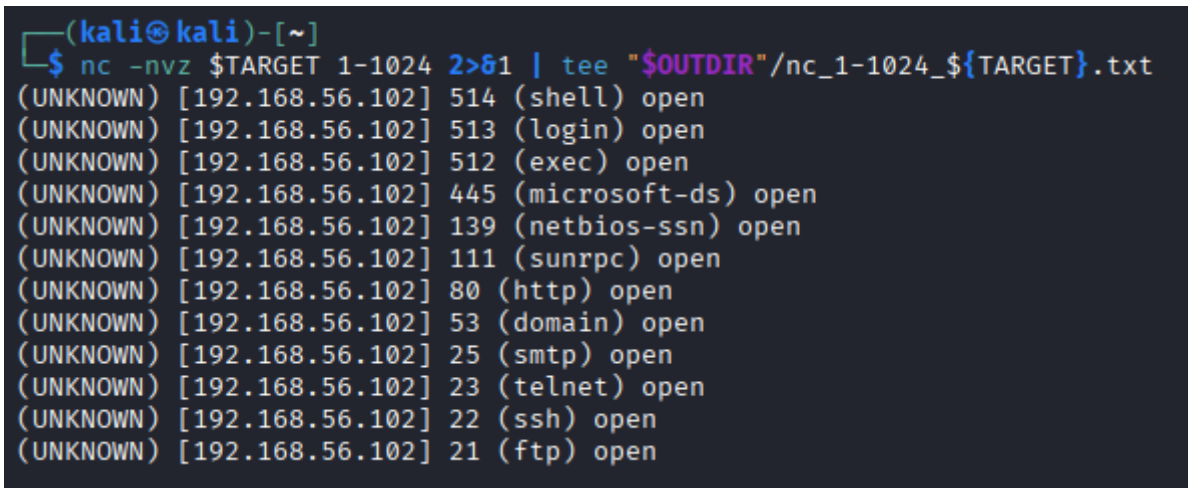
All replies received. Done. = scan concluso con tutte le risposte attese.

Considerazioni finali:

-Hping3 conferma i risultati precedenti (Nmap / Unicornscan): la macchina 192.168.56.102 espone molti servizi, inclusi servizi obsoleti e non cifrati (telnet, rsh, rexec), database datati, SMB, servizi RPC/NFS e vari app server.

-Il pattern di risposte TCP (SYN = SYN/ACK vs SYN= RST) permette di distinguere porte aperte da chiuse o filtrate: in questo caso molte porte hanno risposto in modo attendibile, quindi non si tratta di falsi positivi dovuti a un singolo tool.

9. nc -nvz <target> 1-1024



Ho usato netcat (nc) in modalità “scan” per verificare rapidamente quali porte da 1 a 1024 rispondono sul target 192.168.56.102. L’output è stato registrato con tee per conservare la prova testuale mentre lo vedevo a schermo.

Come ho strutturato il comando:

-nc = netcat, un “coltellino svizzero” per connessioni TCP/UDP e banner grabbing.

- -n = non risolve nomi DNS; usa solo indirizzi IP (più veloce e evita delay di reverse lookup).

- -v = verbose; mostra dettagli su connessioni aperte/chiuse.

- -z = zero-I/O mode: non apre una sessione interattiva, si limita a sondare se la porta accetta connessioni (scan).

-$TARGET = indirizzo IP del bersaglio (qui 192.168.56.102).

-1-1024 = intervallo di porte da testare (porte “well-known”); utile per una prima occhiata rapida.

-2>&1 = unisce stderr a stdout (così anche eventuali messaggi di errore finiscono nel file e nella pipe, come abbiamo visto prima).

- | tee "$OUTDIR"/nc_1-1024_${TARGET}.txt = mostra l’output a schermo e lo salva contemporaneamente in file dentro $OUTDIR.

Output estratto:

-Dall’output postato emergono queste porte aperte:
21 = ftp (open)
22 = ssh (open)
23 = telnet (open)
25 = smtp (open)
53 = domain/DNS (open)
80 = http (open)
111 = sunrpc/rpcbind (open)
139 =netbios-ssn (open)
445 =microsoft-ds / SMB (open)
512/513/514 =services rsh/rexec/login/shell (open)
altri

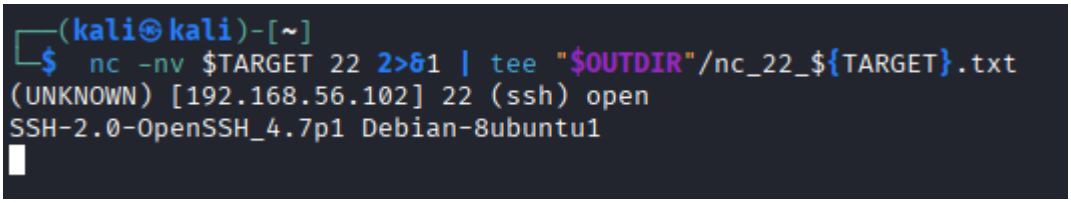
-Netcat mostra anche tra parentesi il nome del servizio risolto via /etc/services, e la parola open per le porte raggiungibili.

Considerazioni finali:

-Conferma pratica: le porte identificate coincidono con quelle già viste via Nmap/Unicornscan/hping3 = quindi non sono falsi positivi.

-Netcat è diretto: se una porta accetta la TCP handshake, nc -z la segnala come open = utile per verifica rapida.

10. nc -nv <target> 22



Ho usato nc per connettermi direttamente alla porta SSH (22) del target e leggere il banner modulare che il servizio restituisce. Questo conferma non solo che SSH è in ascolto, ma anche quale implementazione/versione risponde: è un'informazione utile per selezionare controlli o exploit in ambiente di laboratorio.

Come ho strutturato il comando:

-nc = netcat, semplice tool per aprire connessioni TCP/UDP.

- -n = evita risoluzioni DNS (usa solo IP), più veloce e pulito.

- -v = verbose: mostra il banner e i dettagli della connessione.

-$TARGET = indirizzo IP del bersaglio (es. 192.168.56.102).

-22 = porta da contattare (SSH).

-2>&1 = unisce stderr a stdout così tutto va nel pipe e poi nel file.

- | tee "$OUTDIR"/nc_22_${TARGET}.txt = duplica l'output.

Output estratto:

Questo banner = SSH-2.0-OpenSSH_4.7p1 Debian-8ubuntu1

Indica che il server risponde come OpenSSH versione 4.7p1 (build Debian).

Considerazioni finali:

Il banner conferma che SSH è attivo e che il servizio è una versione datata di OpenSSH. Versioni così vecchie possono contenere vulnerabilità o supportare algoritmi deprecati; in ambiente di laboratorio (Metasploitable) è normale trovare software volutamente vecchio e vulnerabile.

11. nmap -sV <target>

```
┌──(kali㉿kali)-[~]
└─$ nmap -sV $TARGET -oN "$OUTDIR"/nmap_sV_${TARGET}.txt -oX "$OUTDIR"/nmap_sV_${TARGET}.xml
Starting Nmap 7.95 ( https://nmap.org ) at 2025-09-17 20:13 CEST
Nmap scan report for 192.168.56.102
Host is up (0.00020s 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       OpenBSD or Solaris rlogind
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: 08:00:27:30:0C:E5 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Service Info: Hosts:  metasploitable.localdomain, irc.Metasploitable.LAN; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel
```

Ho eseguito una scansione con rilevamento versione (-sV) sull’host 192.168.56.102 per identificare quali servizi sono in ascolto e, quando possibile, quale versione software risponde. Anche qui, output salvati in formato testo leggibile e XML per documentazione.

Come ho strutturato il comando:

-nmap = lo strumento di scansione.

- -sV = version detection: Nmap cerca di identificare il servizio e la versione estraendo banner o interrogando i servizi.

-$TARGET = variabile con l’IP bersaglio (qui 192.168.56.102).

- -oN "$OUTDIR"/nmap_sV_${TARGET}.txt = salva l’output “human readable” in file .txt nella cartella di output.

- -oX "$OUTDIR"/nmap_sV_${TARGET}.xml = salva lo stesso output in XML strutturato (utile per parsing e report automatici).

Output estratto:

-Dall’output emergono le seguenti porte:
21/tcp = vsftpd 2.3.4
22/tcp = OpenSSH 4.7p1
23/tcp = telnetd (Linux telnetd)
25/tcp = Postfix smtpd
53/tcp = ISC BIND 9.4.2
80/tcp = Apache httpd 2.2.8 (Ubuntu) DAV/2
111/tcp = rpcbind 2
139/tcp & 445/tcp = Samba smbd 3.x - 4.x (workgroup: WORKGROUP)
512/513/514 = servizi rlogin/rsh/rexec/login (netkit rshd / rexecd)
1099/tcp = java-rmi (rmiregistry / classpath)
1524/tcp = bindshell (Metasploitable root shell indicator)
2049/tcp = NFS
2121/tcp = ProFTPD 1.3.1
3306/tcp = MySQL 5.0.51a
5432/tcp = PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp = VNC (protocol 3.3)
6667/tcp = UnrealIRCd
8009/8180/8787/… = vari app server (Tomcat, JRuby, Apache JServ, ecc.)

Considerazioni finali:

-L’host espone numerosi servizi obsoleti e potenzialmente vulnerabili: la presenza di vsftpd 2.3.4, Samba 3.x, Apache 2.2.8, MySQL/Postgres vecchi, Telnet e servizi rsh indica un sistema intenzionalmente insicuro (Metasploitable).

-Questi servizi forniscono numerosi vettori di attacco: accesso anonimo FTP, share SMB, database non aggiornati, pannelli/app server vulnerabili e demoni non cifrati.

-Priorità: SMB/FTP/Telnet/servizi remoti non cifrati -> DB -> App servers. In un ambiente reale questo sarebbe classificato come rischio critico**.**

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

```
┌──(kali㉿kali)-[~]
└─$ sudo systemctl start postgresql
[sudo] password for kali:

┌──(kali㉿kali)-[~]
└─$ sudo msfdb init
[i] Database already started
[+] Creating database user 'msf'
[+] Creating databases 'msf'
[+] Creating databases 'msf_test'
[+] Creating configuration file '/usr/share/metasploit-framework/config/database.yml'
[+] Creating initial database schema

┌──(kali㉿kali)-[~]
└─$
```

```
┌──(kali㉿kali)-[~]
└─$ sudo systemctl status postgresql --no-pager
● postgresql.service - PostgreSQL RDBMS
     Loaded: loaded (/usr/lib/systemd/system/postgresql.service; disabled; preset: disabled)
     Active: active (exited) since Wed 2025-09-17 20:16:11 CEST; 1min 19s ago
 Invocation: 171c2a2098a1450ebe80537aaf7f71f8
    Process: 1645 ExecStart=/bin/true (code=exited, status=0/SUCCESS)
   Main PID: 1645 (code=exited, status=0/SUCCESS)
   Mem peak: 1.8M
        CPU: 14ms
```

```
[*] Importing 'Nmap XML' data
[*] Import: Parsing with 'Nokogiri v1.14.5'
[*] Importing host 192.168.56.102
/usr/share/metasploit-framework/vendor/bundle/ruby/3.3.0/gems/recog-3.1.21/lib/recog/fingerprin
nd '?' was replaced with '*' in regular expression
[*] Successfully imported /home/kali/progetto/nmap_allports_sV_reason_192.168.56.102.xml

Hosts
=====

address         mac                name  os_name  os_flavor  os_sp  purpose  info  comments
-------         ---                ----  -------  ---------  -----  -------  ----  --------
192.168.56.102  08:00:27:30:0c:e5        Linux                     server

Services
========

host            port   proto  name         state  info
----            ----   -----  ----         -----  ----
192.168.56.102  21     tcp    ftp          open   vsftpd 2.3.4
192.168.56.102  22     tcp    ssh          open   OpenSSH 4.7p1 Debian 8ubuntu1 protocol 2.0
192.168.56.102  23     tcp    telnet       open   Linux telnetd
192.168.56.102  25     tcp    smtp         open   Postfix smtpd
192.168.56.102  53     tcp    domain       open   ISC BIND 9.4.2
192.168.56.102  80     tcp    http         open   Apache httpd 2.2.8 (Ubuntu) DAV/2
192.168.56.102  111    tcp    rpcbind      open   2 RPC #100000
192.168.56.102  139    tcp    netbios-ssn  open   Samba smbd 3.X - 4.X workgroup: WORKGROUP
192.168.56.102  445    tcp    netbios-ssn  open   Samba smbd 3.X - 4.X workgroup: WORKGROUP
192.168.56.102  512    tcp    exec         open   netkit-rsh rexecd
192.168.56.102  513    tcp    login        open
192.168.56.102  514    tcp    shell        open   Netkit rshd
192.168.56.102  1099   tcp    java-rmi     open   GNU Classpath grmiregistry
```

```
┌──(kali㉿kali)-[~]
└─$ msfconsole -q -x "db_import $OUTDIR/nmap_allports_sV_reason_${TARGET}.xml; hosts; services; exit"
[*] Importing 'Nmap XML' data
[*] Import: Parsing with 'Nokogiri v1.14.5'
[*] Importing host 192.168.56.102
/usr/share/metasploit-framework/vendor/bundle/ruby/3.3.0/gems/recog-3.1.21/lib/recog/fingerprint/regexp_factory.rb:34: warning: nested repeat operator '+' a
nd '?' was replaced with '*' in regular expression
[*] Successfully imported /home/kali/progetto/nmap_allports_sV_reason_192.168.56.102.xml

Hosts
=====

address         mac                name  os_name  os_flavor  os_sp  purpose  info  comments
-------         ---                ----  -------  ---------  -----  -------  ----  --------
192.168.56.102  08:00:27:30:0c:e5        Linux                      server

Services
========

host            port   proto  name         state  info
----            ----   -----  ----         -----  ----
192.168.56.102  21     tcp    ftp          open   vsftpd 2.3.4
192.168.56.102  22     tcp    ssh          open   OpenSSH 4.7p1 Debian 8ubuntu1 protocol 2.0
192.168.56.102  23     tcp    telnet       open   Linux telnetd
192.168.56.102  25     tcp    smtp         open   Postfix smtpd
192.168.56.102  53     tcp    domain       open   ISC BIND 9.4.2
192.168.56.102  80     tcp    http         open   Apache httpd 2.2.8 (Ubuntu) DAV/2
192.168.56.102  111    tcp    rpcbind      open   2 RPC #100000
192.168.56.102  139    tcp    netbios-ssn  open   Samba smbd 3.X - 4.X workgroup: WORKGROUP
192.168.56.102  445    tcp    netbios-ssn  open   Samba smbd 3.X - 4.X workgroup: WORKGROUP
192.168.56.102  512    tcp    exec         open   netkit-rsh rexecd
192.168.56.102  513    tcp    login        open
```

```
192.168.56.102  514    tcp    shell        open   Netkit rshd
192.168.56.102  1099   tcp    java-rmi     open   GNU Classpath grmiregistry
192.168.56.102  1524   tcp    bindshell    open   Metasploitable root shell
192.168.56.102  2049   tcp    nfs          open   2-4 RPC #100003
192.168.56.102  2121   tcp    ftp          open   ProFTPD 1.3.1
192.168.56.102  3306   tcp    mysql        open   MySQL 5.0.51a-3ubuntu5
192.168.56.102  3632   tcp    distccd      open   distccd v1 (GNU) 4.2.4 (Ubuntu 4.2.4-1ubuntu4)
192.168.56.102  5432   tcp    postgresql   open   PostgreSQL DB 8.3.0 - 8.3.7
192.168.56.102  5900   tcp    vnc          open   VNC protocol 3.3
192.168.56.102  6000   tcp    x11          open   access denied
192.168.56.102  6667   tcp    irc          open   UnrealIRCd
192.168.56.102  6697   tcp    irc          open   UnrealIRCd
192.168.56.102  8009   tcp    ajp13        open   Apache Jserv Protocol v1.3
192.168.56.102  8180   tcp    http         open   Apache Tomcat/Coyote JSP engine 1.1
192.168.56.102  8787   tcp    drb          open   Ruby DRb RMI Ruby 1.8; path /usr/lib/ruby/1.8/drb
192.168.56.102  47438  tcp    java-rmi     open   GNU Classpath grmiregistry
192.168.56.102  49315  tcp    mountd       open   1-3 RPC #100005
192.168.56.102  58361  tcp    nlockmgr     open   1-4 RPC #100021
192.168.56.102  58811  tcp    status       open   1 RPC #100024
```

```
┌──(kali㉿kali)-[~]
└─$ for f in "$OUTDIR"/*.xml; do echo "db_import $f" >> "$OUTDIR"/import.rc; done

┌──(kali㉿kali)-[~]
└─$ sed -n '1,200p' "$OUTDIR"/import.rc
db_import /home/kali/progetto/nmap_allports_sV_reason_192.168.56.102.xml
db_import /home/kali/progetto/nmap_ping_192.168.56.102.xml
db_import /home/kali/progetto/nmap_sS_sV_T4_192.168.56.102.xml
db_import /home/kali/progetto/nmap_sV_192.168.56.102.xml
db_import /home/kali/progetto/nmap_top10_192.168.56.102.xml

┌──(kali㉿kali)-[~]
```

```
┌──(kali㉿kali)-[~]
└─$ msfconsole -r "$OUTDIR"/import.rc

Metasploit tip: The use command supports fuzzy searching to try and
select the intended module, e.g. use kerberos/get_ticket or use
kerberos forge silver ticket


 ________________________________________
/ it looks like you're trying to run a \
\ module                               /
 ----------------------------------------
  \
   \
      __
     /  \
     |  |
     @  @
     |  |
     || |/
     || ||
     |\_/|
     \___/


       =[ metasploit v6.4.84-dev                          ]
+ -- --=[ 2,547 exploits - 1,309 auxiliary - 1,683 payloads     ]
+ -- --=[ 432 post - 49 encoders - 13 nops - 9 evasion          ]

Metasploit Documentation: https://docs.metasploit.com/
The Metasploit Framework is a Rapid7 Open Source Project

[*] Processing /home/kali/progetto/import.rc for ERB directives.
```

```
Metasploit Documentation: https://docs.metasploit.com/
The Metasploit Framework is a Rapid7 Open Source Project

[*] Processing /home/kali/progetto/import.rc for ERB directives.
resource (/home/kali/progetto/import.rc)> db_import /home/kali/progetto/nmap_allports_sV_reason_192.168.56.102.xml
[*] Importing 'Nmap XML' data
[*] Import: Parsing with 'Nokogiri v1.14.5'
[*] Importing host 192.168.56.102
/usr/share/metasploit-framework/vendor/bundle/ruby/3.3.0/gems/recog-3.1.21/lib/recog/fingerprint/regexp_factory.rb:34: warning: nested repeat operator '+' a
nd '?' was replaced with '*' in regular expression
[*] Successfully imported /home/kali/progetto/nmap_allports_sV_reason_192.168.56.102.xml
resource (/home/kali/progetto/import.rc)> db_import /home/kali/progetto/nmap_ping_192.168.56.102.xml
[*] Importing 'Nmap XML' data
[*] Import: Parsing with 'Nokogiri v1.14.5'
[*] Importing host 192.168.56.102
[*] Successfully imported /home/kali/progetto/nmap_ping_192.168.56.102.xml
resource (/home/kali/progetto/import.rc)> db_import /home/kali/progetto/nmap_sS_sV_T4_192.168.56.102.xml
[*] Importing 'Nmap XML' data
[*] Import: Parsing with 'Nokogiri v1.14.5'
[*] Importing host 192.168.56.102
[*] Successfully imported /home/kali/progetto/nmap_sS_sV_T4_192.168.56.102.xml
resource (/home/kali/progetto/import.rc)> db_import /home/kali/progetto/nmap_sV_192.168.56.102.xml
[*] Importing 'Nmap XML' data
[*] Import: Parsing with 'Nokogiri v1.14.5'
[*] Importing host 192.168.56.102
[*] Successfully imported /home/kali/progetto/nmap_sV_192.168.56.102.xml
resource (/home/kali/progetto/import.rc)> db_import /home/kali/progetto/nmap_top10_192.168.56.102.xml
[*] Importing 'Nmap XML' data
[*] Import: Parsing with 'Nokogiri v1.14.5'
[*] Importing host 192.168.56.102
[*] Successfully imported /home/kali/progetto/nmap_top10_192.168.56.102.xml
msf >
```

```
msf > hosts

Hosts
=====

address         mac                name  os_name  os_flavor  os_sp  purpose  info  comments
-------         ---                ----  -------  ---------  -----  -------  ----  --------
192.168.56.102  08:00:27:30:0c:e5        Linux                     server

msf > services
Services
========

host            port   proto  name          state  info
----            ----   -----  ----          -----  ----
192.168.56.102  21     tcp    ftp           open   vsftpd 2.3.4
192.168.56.102  22     tcp    ssh           open   OpenSSH 4.7p1 Debian 8ubuntu1 protocol 2.0
192.168.56.102  23     tcp    telnet        open   Linux telnetd
192.168.56.102  25     tcp    smtp          open   Postfix smtpd
192.168.56.102  53     tcp    domain        open   ISC BIND 9.4.2
192.168.56.102  80     tcp    http          open   Apache httpd 2.2.8 (Ubuntu) DAV/2
192.168.56.102  111    tcp    rpcbind       open   2 RPC #100000
192.168.56.102  139    tcp    netbios-ssn   open   Samba smbd 3.X - 4.X workgroup: WORKGROUP
192.168.56.102  445    tcp    microsoft-ds  open   Samba smbd 3.X - 4.X workgroup: WORKGROUP
192.168.56.102  512    tcp    exec          open   netkit-rsh rexecd
192.168.56.102  513    tcp    login         open   OpenBSD or Solaris rlogind
192.168.56.102  514    tcp    shell         open   Netkit rshd
192.168.56.102  1099   tcp    java-rmi      open   GNU Classpath grmiregistry
192.168.56.102  1524   tcp    bindshell     open   Metasploitable root shell
192.168.56.102  2049   tcp    nfs           open   2-4 RPC #100003
192.168.56.102  2121   tcp    ftp           open   ProFTPD 1.3.1
192.168.56.102  3306   tcp    mysql         open   MySQL 5.0.51a-3ubuntu5
192.168.56.102  3632   tcp    distccd       open   distccd v1 (GNU) 4.2.4 (Ubuntu 4.2.4-1ubuntu4)
```

Ho avviato il servizio PostgreSQL, inizializzato il database di Metasploit (msfdb) e importato i file XML di Nmap nella base dati di Metasploit. In questo modo le informazioni delle scansioni (host, porte, servizi, versioni) sono centralizzate nel DB del framework e possono essere consultate e usate da moduli/metasploit per automatizzare enumerazioni e prove successive.

Come ho strutturato il comando:

A) Avvio di PostgreSQL
-sudo = privilegi amministrativi per gestire servizi.
-systemctl start postgresql =avvia il servizio PostgreSQL richiesto da Metasploit per memorizzare dati (hosts, servizi, credenziali).

B) Inizializzazione del DB di Metasploit
msfdb init = crea l'utente DB (msf), le tabelle iniziali, imposta database.yml in /usr/share/metasploit-framework/config/ e genera lo schema. Messaggi tipo "[+] Creating databases 'msf'" indicano che il DB è pronto.

C)Importazione singola
-msfconsole = avvia l'interfaccia Metasploit.
-q = quiet: riduce output iniziale.
-x "..." = esegue la lista di comandi passati e poi esce; qui db_import <file> importa l'XML, hosts e services

mostrano gli oggetti importati.
-db_import = legge l’XML di Nmap e popola le tabelle hosts, services, vulns (se presenti) del DB Metasploit.

D) Creazione di un file resource per importare più XML
-for ...; do echo "db_import $f" >> import.rc; done = costruisce automaticamente uno script import.rc che contiene una riga db_import per ogni XML nella cartella.
-sed -n '1,200p' = mostra le prime 200 righe per verificare il file creato.
-msfconsole -r import.rc = avvia msfconsole ed esegue il file resource, importando in batch tutti gli XML.

Output estratto:

-Messaggi di successo tipo: [+] Successfully imported /home/kali/progetto/xxx.xml.

-msf > hosts = mostra l’host importato (es. 192.168.56.102 con MAC e OS).

-msf > services = elenca tutte le porte/servizi importati (ftp, ssh, telnet, smtp, http, smb, mysql, ecc.) con stato/version info prese dagli XML.

-Conferma che Nmap XML è stato parsato correttamente (usa Nokogiri/recog per fingerprinting).

Considerazioni finali:

Centralizzare i risultati in Metasploit trasforma semplici output di scansione in un database interrogabile: si possono cercare host per servizio, eseguire moduli scanner/auxiliary che leggono dal DB, tracciare credenziali, e collegare risultati a moduli exploit in modo ripetibile.

13. nmap -f –mtu=512 <target>

Ho lanciato una scansione Nmap usando frammentazione dei pacchetti IP (-f) e forzando una MTU (unità di trasmissione massima) di 512 byte (--mtu 512).
Una MTU la dimensione massima in byte di un pacchetto di dati che può essere inviato in una rete in un unico frame, senza subire frammentazione.
L'idea è testare se frammentare i pacchetti può eludere filtri semplici o cambiare il comportamento di apparati intermedi; i risultati sono stati salvati, anche qui, in .txt e .xml per documentazione.

Come ho strutturato il comando:

-nmap = lo scanner usato.

- -f = fragment packets: chiede a Nmap di spezzare i pacchetti IP in più frammenti. Alcuni firewall/IDS semplici non riescono a riassemblarli correttamente e quindi non vedono la scansione.

- --mtu 512 = forza la dimensione massima dell'unità di trasmissione (MTU) a 512 byte per ogni frammento; impatta la dimensione dei frammenti prodotti.

-$TARGET = indirizzo IP bersaglio (qui la VM 192.168.56.102).

- -oN "$OUTDIR"/nmap_fragment_mtu512_${TARGET}.txt = salva l'output in formato testo leggibile nella cartella $OUTDIR.

- -oX "$OUTDIR"/nmap_fragment_mtu512_${TARGET}.xml = salva l'output anche in XML per parsing e archiviazione.

Output estratto:

-Host raggiungibile = Host is up.

-Elenco porte trovate (stesse principali delle scansioni precedenti)=
ftp (21)
ssh (22)
telnet (23)
smtp (25)
http (80)
rpcbind (111)
netbios/SMB (139/445)
mysql (3306)
postgresql (5432)
vnc (5900)
vari servizi applicativi (Tomcat, JRuby, ecc.)

Considerazioni finali:

-La frammentazione non ha nascosto l'host né impedito l'individuazione delle porte: Nmap ha comunque rilevato le stesse porte aperte. In pratica l'evade-based scan con -f/--mtu a volte può funzionare contro dispositivi mal configurati, ma non è una soluzione magica universale.

-In questo laboratorio (rete semplice / VM) i pacchetti frammentati sono stati riassemblati o gestiti correttamente dall'host/gateway, quindi la scansione rimane efficace.

14. masscan <network> -p80 –banners –source-ip <target>

```
┌──(kali㉿kali)-[~]
└─$ ip -o -4 addr show | awk '{print $2, $4}'
lo 127.0.0.1/8
eth0 192.168.56.101/24

┌──(kali㉿kali)-[~]
└─$ ip route get 192.168.56.102
192.168.56.102 dev eth0 src 192.168.56.101 uid 1000
    cache

┌──(kali㉿kali)-[~]
└─$ ping -c 3 192.168.56.102
PING 192.168.56.102 (192.168.56.102) 56(84) bytes of data.
64 bytes from 192.168.56.102: icmp_seq=1 ttl=64 time=0.396 ms
64 bytes from 192.168.56.102: icmp_seq=2 ttl=64 time=0.347 ms
64 bytes from 192.168.56.102: icmp_seq=3 ttl=64 time=0.359 ms

── 192.168.56.102 ping statistics ──
3 packets transmitted, 3 received, 0% packet loss, time 2029ms
rtt min/avg/max/mdev = 0.347/0.367/0.396/0.020 ms
```

```
┌──(kali㉿kali)-[~]
└─$ sudo nmap -Pn -p1-1024 -T4 --min-rate 200 -sV 192.168.56.102 -oN "$OUTDIR"/nmap_target_1-1024_${TARGET}.txt -oX "$OUTDIR"/nmap_target_1-1024_${TARGET}.
xml
Starting Nmap 7.95 ( https://nmap.org ) at 2025-09-18 11:48 CEST
Nmap scan report for 192.168.56.102
Host is up (0.00014s latency).
Not shown: 1012 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       OpenBSD or Solaris rlogind
514/tcp open  shell       Netkit rshd
MAC Address: 08:00:27:30:0C:E5 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Service Info: Host:  metasploitable.localdomain; 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 28.94 seconds
```

Ho verificato prima la connettività e la configurazione IP locale verso il target (192.168.56.102) (comandi ip, ip route, ping) per accertarmi che la rete sia raggiungibile; poi ho eseguito una scansione rapida e mirata delle porte da 1 a 1024 con Nmap includendo rilevamento versioni (-sV) e un rate/timing più aggressivo per ottenere i risultati in tempi contenuti. Gli output sono stati salvati in testo e XML per la documentazione.

Come ho strutturato il comando:

-ip -o -4 addr show | awk '{print $2, $4}'
ip -o -4 addr show = mostra le interfacce IPv4 in formato one-line per ognuna.
| awk '{print $2, $4}' = mantiene solo nome interfaccia e indirizzo/netmask (es. eth0 192.168.56.101/24): controllo rapido della tua IP sorgente.

-ip route get 192.168.56.102 = restituisce la route che il kernel userà per raggiungere il target (interfaccia e IP sorgente effettivi). Utile per verificare che il traffico esca dall'interfaccia prevista (evita sorprese con più interfacce).

ping -c 3 192.168.56.102 = tre echo request per confermare latenza e perdita pacchetti; conferma che l'host risponde a livello ICMP.

sudo nmap -p1-1024 -T4 --min-rate 200 -sV 192.168.56.102 -oN "$OUTDIR"/nmap_target_1-1024_${TARGET}.txt -oX "$OUTDIR"/nmap_target_1-1024_${TARGET}.xml
sudo = privilegi per alcune tecniche più affidabili (es. raw sockets).
nmap = scanner.
p1-1024 = scansiona le porte "well-known" (1..1024). È una scansione mirata, più veloce che scansionare tutte le 65535.
T4 = timing aggressivo
--min-rate 200 = impone almeno 200 pacchetti al secondo; accelera lo scan su host locali/VM.
-sV = rilevamento versione per ogni porta aperta.
-oN / -oX = salva output in testo leggibile e XML per tracciare le evidenze.

Output estratto:

-IP locale/interfaccia= eth0 192.168.56.101/24.

-Route verso target= 192.168.56.102 dev eth0 src 192.168.56.101.

-ping 3/3 ricevuti, RTT 0.35 ms = connettività stabile e bassa latenza (VM sulla stessa rete).

-Scansione Nmap (porte 1–1024) = porte aperte rilevate (conferma degli elementi già raccolti):
21 ftp = vsftpd 2.3.4
22 ssh =OpenSSH 4.7p1
23 telnet = telnetd
25 smtp = Postfix
53 domain = BIND 9.4.2
80 http = Apache 2.2.8 (DAV/2)
111 rpcbind
139/445 netbios/SMB = Samba smbd 3.x
512/513/514/1099/1524/... = servizi remoti e bind shell indicatori Metasploitable
3306 mysql, 5432 postgresql, 5900 vnc, 8009/8180/... app servers ecc.

Considerazioni finali:

La verifica di rete (ip/route/ping) dimostra che non ci sono problemi di instradamento: i pacchetti verso il target passano dall'interfaccia prevista. Questo rende affidabili i risultati della scansione.