Esercizio S5_L2_Enumeration

CONSEGNA:

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

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

E la seguente sul target Windows:

- OS fingerprint.

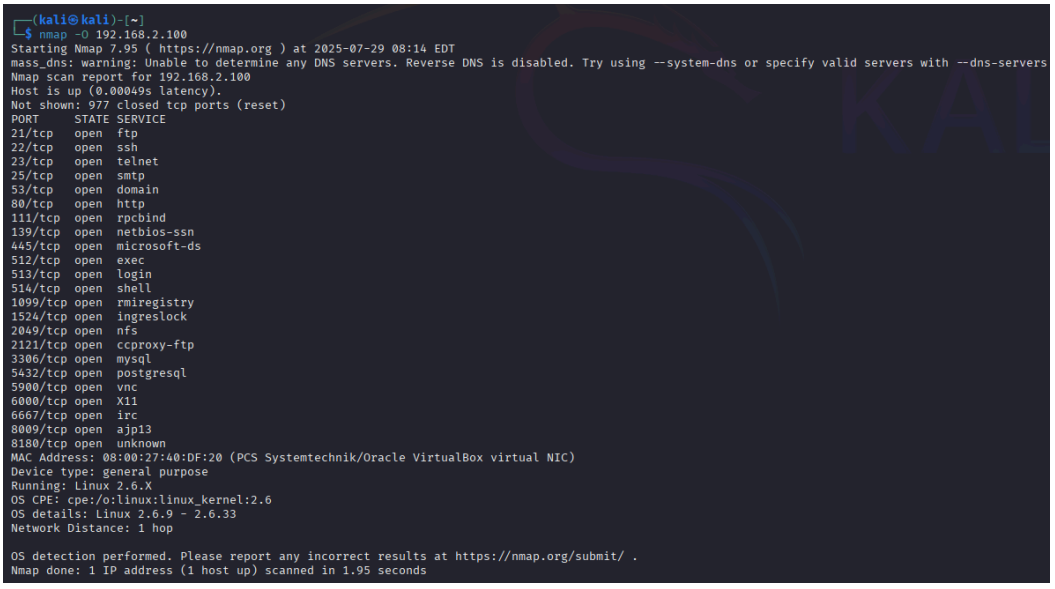
A valle delle scansioni è prevista la produzione di un report contenente le seguenti info (dove disponibili):

- IP.
- Sistema Operativo.
- Porte Aperte.
- Servizi in ascolto con versione.

SCANNING METASPLOIABLE

nmap -O 192.168.2.100

Iniziamo dunque con la scansione volta a scovare il tipo di OS utilizzato dalla macchina che andiamo a scansionare.



A seguito della scansione otteniamo in output la versione del sistema operativo: Linux 2.6.9 - 2.6.33.

Nmap ottiene le informazioni tramite una tecnica chiamata **OS fingerprinting attivo**, basata sull'invio di pacchetti TCP/IP e ICMP costruiti in modi specifici, per osservare come il sistema target risponde. Ogni sistema operativo implementa lo **stack TCP/IP** in modo leggermente diverso, e queste differenze possono essere usate per dedurre il tipo di sistema.

Alcuni dei parametri analizzati da Nmap sono:

- **TTL (Time To Live)** iniziale nei pacchetti di risposta
- **Window Size TCP** (dimensione della finestra)
- **Gestione dei flag TCP anomali** (es. FIN, URG, PSH su porte chiuse)
- **Risposte ICMP (Echo e Timestamp)** e comportamento nei confronti dei pacchetti frammentati

Nmap confronta queste risposte con un **database interno di firme note** per identificare il sistema operativo più probabile. Se i parametri corrispondono in modo sufficientemente accurato, Nmap fornisce un'identificazione precisa.

Esempio:
Se un sistema risponde con:

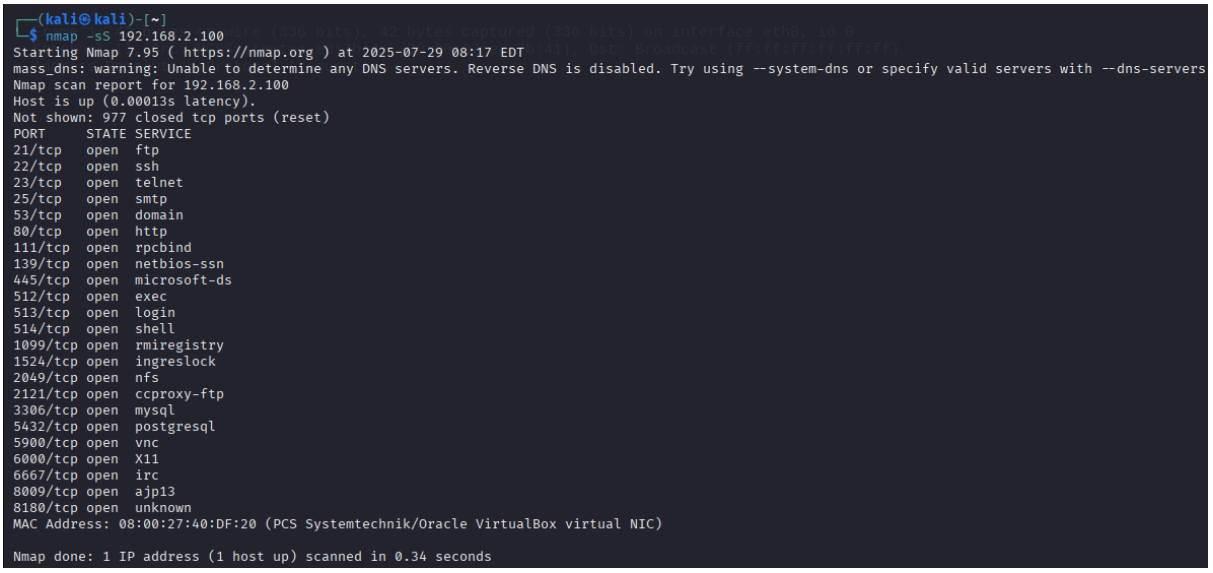
- TTL = 64
- Window Size = 5840
- MSS = 1460
- Risposte ICMP attive

Nmap probabilmente lo classificherà come **Linux**.

Nmap non legge quindi direttamente il nome dell'OS: lo deduce analizzando come si comporta a livello di rete.

nmap -sS 192.168.2.100

Come da consegna ho poi eseguito il comando per scansionare i servizi tramite SYN.

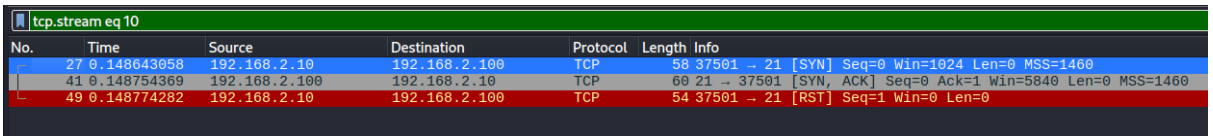


Al contempo ho aperto wireshark per analizzare il traffico di rete grazie al quale posso portare un esempio della scansione SYN.

Nell'immagine successiva possiamo notare infatti uno scambio three-way handshake che inizia dalla macchina Kali con ip 192.168.2.10:37501 verso la metasploitable 192.168.2.100:21.

Il servizio risponde al SYN della kali con un SYN-ACK informando quindi nmap della disponibilità del servizio sulla porta 21.

A quel punto nmap interrompe il tentativo di connessione mandando un messaggio RST (Reset). Lo stesso processo viene ovviamente ripetuto per tutti i servizi che nmap va a scansionare e riporterà i risultati in output al termine dell'analisi.



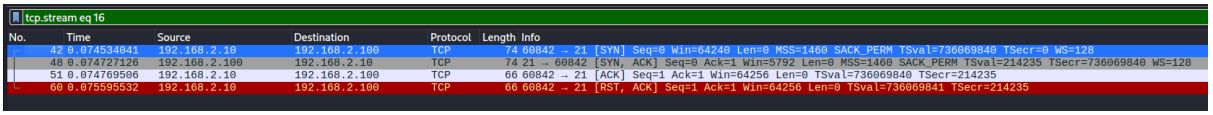
nmap -sT 192.168.2.100

```
┌──(kali㉿kali)-[~]
└─$ nmap -sT 192.168.2.100
Starting Nmap 7.95 ( https://nmap.org ) at 2025-07-29 08:34 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
Nmap scan report for 192.168.2.100
Host is up (0.00049s latency).
Not shown: 977 closed tcp ports (conn-refused)
PORT     STATE SERVICE
21/tcp   open  ftp
22/tcp   open  ssh
23/tcp   open  telnet
25/tcp   open  smtp
53/tcp   open  domain
80/tcp   open  http
111/tcp  open  rpcbind
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
512/tcp  open  exec
513/tcp  open  login
514/tcp  open  shell
1099/tcp open  rmiregistry
1524/tcp open  ingreslock
2049/tcp open  nfs
2121/tcp open  ccproxy-ftp
3306/tcp open  mysql
5432/tcp open  postgresql
5900/tcp open  vnc
6000/tcp open  X11
6667/tcp open  irc
8009/tcp open  ajp13
8180/tcp open  unknown
MAC Address: 08:00:27:40:DF:20 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 0.17 seconds
```

Analizzando, per analogia, la stessa porta (21) utilizzando lo scan TCP completo (Nmap -sT), notiamo che stavolta i pacchetti scambiati sono quattro:

- Il client invia un pacchetto SYN.
- Il server risponde con SYN-ACK.
- Il client completa il three-way handshake con un ACK.
- Infine, la connessione viene chiusa tramite un pacchetto RST-ACK.



In pratica, possiamo osservare che durante questo tipo di scansione la connessione con il servizio viene effettivamente stabilita (tramite il completamento del three-way handshake) e interrotta solo dopo, con un reset.

Questa è la principale differenza rispetto allo scan SYN (Nmap -sS), in cui la connessione non viene mai completata, rendendo lo scan più stealth.

nmap -sV 192.168.2.100

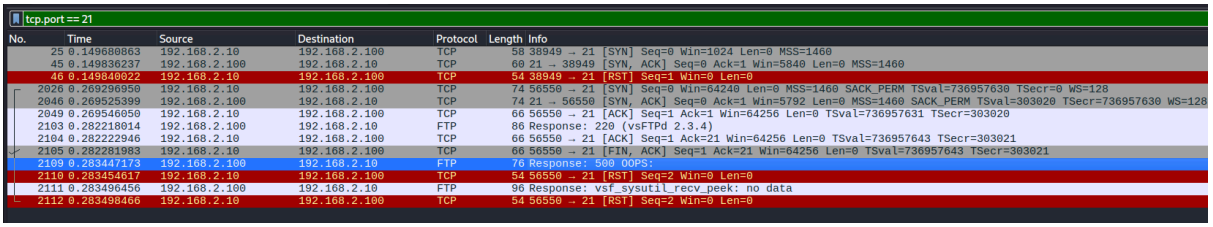
Il prossimo comando utilizzato è servito per la scansione delle versioni dei servizi attivi.

```
┌──(kali㉿kali)-[~]
└─$ nmap -sV 192.168.2.100
Starting Nmap 7.95 ( https://nmap.org ) at 2025-07-29 08:49 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
Nmap scan report for 192.168.2.100
Host is up (0.00056s 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: 08:00:27:40:DF:20 (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 52.55 seconds
```

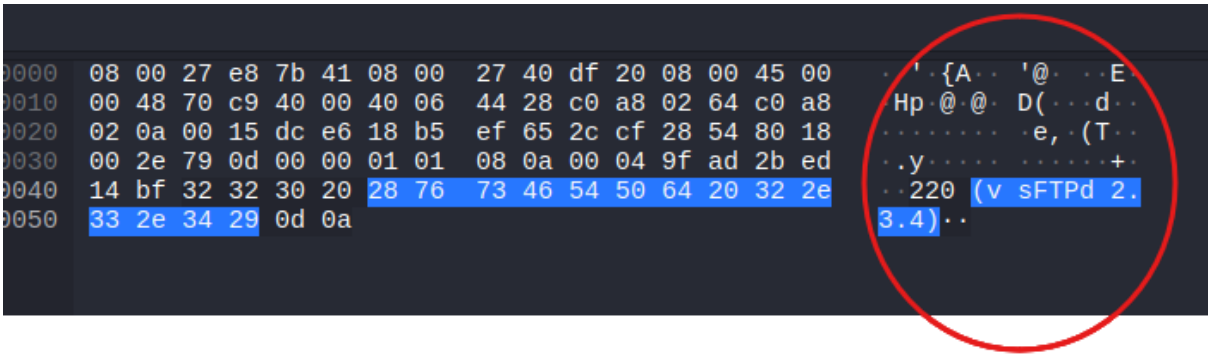
Nmap invia probe specifici a ciascuna porta aperta usando il suo database `nmap-service-probes`, cercando di far rispondere i servizi in modo identificativo (banner, stringhe, protocollo). Le risposte vengono analizzate tramite pattern matching per determinare nome, versione, tipo di dispositivo, sistema operativo e, se presente, l'identificatore CPE.

L'output mostra per ogni porta:
- Il nome del servizio,
- La versione identificata,
- Eventuali dettagli aggiuntivi come sistema operativo o build.



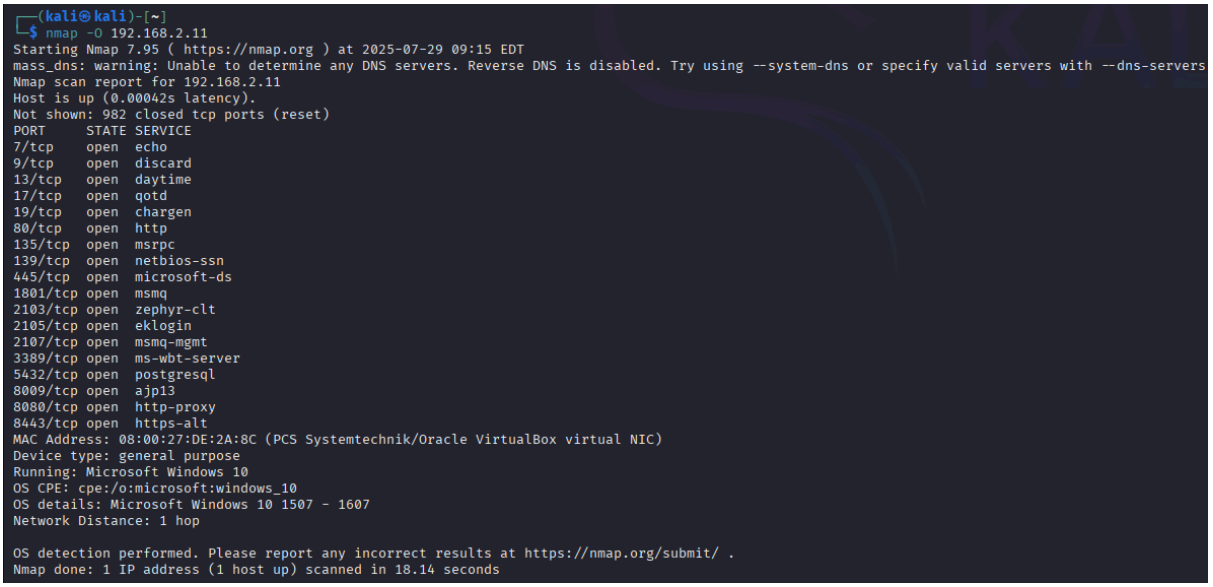
Prestando attenzione alla lista qui sopra, filtrata sulla porta tcp 21, notiamo che il pacchetto 2103 contiene una risposta da parte della metasploitable tramite servizio FTP dove viene comunicata la versione del servizio (vsFTPd 2.3.4).

Nel dettaglio del payload:



SCANNING WINDOWS

nmap -O 192.168.2.11



Confrontando i risultati delle scansioni Nmap eseguite su due target distinti (Metasploitable e Windows 10), emergono differenze significative:

*Il sistema **Metasploitable** espone volontariamente numerosi servizi, molti dei quali obsoleti e vulnerabili (es. **Telnet**, **vsFTPd**, **Apache 2.2**, **Samba 3.x**). È un ambiente di laboratorio progettato per essere compromesso, ideale per test e simulazioni di attacco.*

*Al contrario, **Windows 10** mostra una configurazione tipica di un sistema reale, con servizi nativi Microsoft (es. **RPC**, **NetBIOS**, **MSMQ**) e un'esposizione di rete più limitata. Tuttavia, sarebbe comunque necessario indagare sulle versioni dei servizi attivi, per verificare l'eventuale presenza di vulnerabilità note o configurazioni deboli.*

*Notiamo comunque alcuni **servizi generalmente considerati critici**, tra cui:*

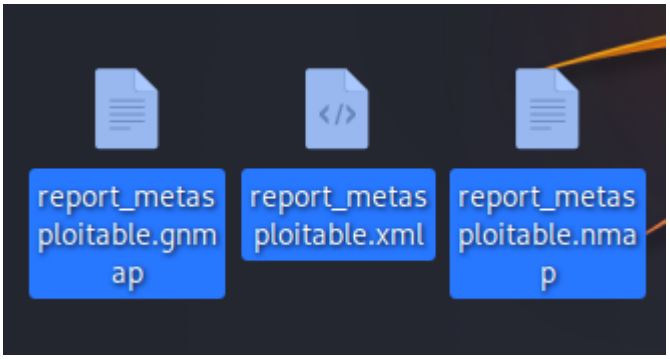
- ***Porta 139 (netbios-ssn)** – usata per condivisione file, rischio elevato se esposta*
- ***Porta 445 (microsoft-ds)** – associata al protocollo SMB, vettore di attacchi famosi come EternalBlue*
- ***Porta 3389 (RDP)** – protocollo di desktop remoto, obiettivo comune di brute force e vulnerabilità*
- ***Porte 7, 9, 13, 17, 19** – servizi legacy come Echo, Discard, Daytime, QOTD, Chargen; non necessari e potenzialmente sfruttabili per attacchi DoS.*

REPORT CON NMAP

Report Metasploitable

```
┌──(kali㉿kali)-[~/Desktop]
└─$ nmap -sS -sV -O -oA report_metasploitable 192.168.2.100
Starting Nmap 7.95 ( https://nmap.org ) at 2025-07-29 09:35 EDT
```

L'output di tale comando salverà i risultati delle scansioni in 3 file differenti:



Il file .nmap è un semplicissimo file di testo, quello con estensione .gnmap è invece un pochino più grezzo e con una formattazione poco consona alla lettura.
Il terzo file invece è in formato XML, il che consente di trasformarlo in un html tramite "xsltproc" restituendoci un report molto gradevole.

Qualora si volesse generare solamente il file XML possiamo utilizzare direttamente il parametro -oX:

```
┌──(kali㉿kali)-[~/Desktop]
└─$ nmap -sS -sV -O --script vuln -oX metasploitable_report.xml 192.168.2.100
Starting Nmap 7.95 ( https://nmap.org ) at 2025-07-29 09:49 EDT
```

Ho quindi colto l'occasione per generare il report contenente altresì una scansione delle vulnerabilità per renderlo il più completo possibile.

Una volta generato il report possiamo convertirlo con xsltproc in un file html tramite il seguente comando:

sudo xsltproc /usr/share/nmap/nmap.xsl *nomefile.xml* -o *nomefile.html*

```
┌──(kali㉿kali)-[~/Desktop]
└─$ sudo xsltproc /usr/share/nmap/nmap.xsl metasploitable_report.xml -o metasploitable_report_vuln.html
```

Di seguito possiamo notare le prime righe del report generato in formato html aperto tramite browser.

192.168.2.100

Address

- 192.168.2.100 (ipv4)
- 08:00:27:40:DF:20 - PCS Systemtechnik/Oracle VirtualBox virtual NIC (mac)

Ports

The 977 ports scanned but not shown below are in state: **closed**

- 977 ports replied with: **reset**

Port		State (toggle closed [0] | filtered [0])	Service	Reason	Product
21	tcp	open	ftp	syn-ack	vsftpd
	ftp-vsftpd-backdoor	VULNERABLE: vsFTPd version 2.3.4 backdoor State: VULNERABLE (Exploitable) IDs: BID:48539 CVE:CVE-2011-2523 vsFTPd version 2.3.4 backdoor, this was reported on 2011-07-04. Disclosure date: 2011-07-03 Exploit results: Shell command: id Results: uid=0(root) gid=0(root) References: https://github.com/rapid7/metasploit-framework/blob/master/modules/exploits/unix/ftp/vsftpd_234_backdoor.rb https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2011-2523 https://www.securityfocus.com/bid/48539 http://scarybeastsecurity.blogspot.com/2011/07/alert-vsftpd-download-backdoored.html			
22	tcp	open	ssh	syn-ack	OpenSSH
23	tcp	open	telnet	syn-ack	Linux telnetd
25	tcp	open	smtp	syn-ack	Postfix smtpd
	smtp-vuln-cve2010-4344	The SMTP server is not Exim: NOT VULNERABLE			
53	tcp	open	domain	syn-ack	ISC BIND
80	tcp	open	http	syn-ack	Apache httpd

Essendo una Metasploitable2 notiamo dunque che nel report sono state raccolte, oltre alle varie info generiche, anche una lista di tutte le vulnerabilità potenzialmente sfruttabili per ciascun servizio.

1099	tcp	open	java-rmi	syn-ack	GNU Classpath grmiregistry		
	rmi-vuln-classloader	VULNERABLE: RMI registry default configuration remote code execution vulnerability State: VULNERABLE Default configuration of RMI registry allows loading classes from remote URLs which can lead to remote code execution. References: https://github.com/rapid7/metasploit-framework/blob/master/modules/exploits/multi/misc/java_rmi_server.rb					
1524	tcp	open	bindshell	syn-ack	Metasploitable root shell		
2049	tcp	open	nfs	syn-ack		2-4	RPC #100003
2121	tcp	open	ftp	syn-ack	ProFTPD	1.3.1	
3306	tcp	open	mysql	syn-ack	MySQL	5.0.51a-3ubuntu5	
5432	tcp	open	postgresql	syn-ack	PostgreSQL DB	8.3.0 - 8.3.7	
	ssl-poodle	VULNERABLE: SSL POODLE information leak State: VULNERABLE IDs: BID:70574 CVE:CVE-2014-3566 The SSL protocol 3.0, as used in OpenSSL through 1.0.1i and other products, uses nondeterministic CBC padding, which makes it easier for man-in-the-middle attackers to obtain cleartext data via a padding-oracle attack, aka the "POODLE" issue. Disclosure date: 2014-10-14 Check results: TLS_RSA_WITH_AES_128_CBC_SHA References: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-3566 https://www.imperialviolet.org/2014/10/14/poodle.html https://www.securityfocus.com/bid/70574 https://www.openssl.org/~bodo/ssl-poodle.pdf					

Report Metasploitable

Procediamo dunque a creare un report anche per la macchina windows:

```
┌──(kali㉿kali)-[~/Desktop]
└─$ nmap -sS -sV -O --script vuln -oX windows_report.xml 192.168.2.11
Starting Nmap 7.95 ( https://nmap.org ) at 2025-07-29 10:13 EDT
```

Convertiamo nuovamente il file xml in html:

```
┌──(kali㉿kali)-[~/Desktop]
└─$ sudo xsltproc /usr/share/nmap/nmap.xsl  windows_report.xml -o windows_report.html
[sudo] password for kali:
```

E anche in questo caso abbiamo ottenuto un report comprensivo di eventuali vulnerabilità:

192.168.2.11

Address

- 192.168.2.11 (ipv4)
- 08:00:27:DE:2A:8C - PCS Systemtechnik/Oracle VirtualBox virtual NIC (mac)

Ports

The 982 ports scanned but not shown below are in state: **closed**

- 982 ports replied with: **reset**

Port		State (toggle closed [0] | filtered [0])	Service	Reason	Product
7	tcp	open	echo	syn-ack	
9	tcp	open	discard	syn-ack	
13	tcp	open	daytime	syn-ack	Microsoft Windows International daytime
17	tcp	open	qotd	syn-ack	Windows qotd
19	tcp	open	chargen	syn-ack	
80	tcp	open	http	syn-ack	Microsoft IIS httpd
	http-csrf	Couldn't find any CSRF vulnerabilities.			
	http-dombased-xss	Couldn't find any DOM based XSS.			
	http-stored-xss	Couldn't find any stored XSS vulnerabilities.			
	http-server-header	Microsoft-IIS/10.0			
135	tcp	open	msrpc	syn-ack	Microsoft Windows RPC
139	tcp	open	netbios-ssn	syn-ack	Microsoft Windows netbios-ssn
445	tcp	open	microsoft-ds	syn-ack	Microsoft Windows 7 - 10 microsoft-ds
1801	tcp	open	msmq	syn-ack	
2103	tcp	open	msrpc	syn-ack	Microsoft Windows RPC
2105	tcp	open	msrpc	syn-ack	Microsoft Windows RPC
2107	tcp	open	msrpc	syn-ack	Microsoft Windows RPC
3389	tcp	open	ms-wbt-server	syn-ack	Microsoft Terminal Services
5432	tcp	open	postgresql	syn-ack	
8009	tcp	open	ajp13	syn-ack	Apache Jserv
8080	tcp	open	http	syn-ack	Apache Tomcat/Coyote JSP engine
	http-csrf	Couldn't find any CSRF vulnerabilities.			
	http-dombased-xss	Couldn't find any DOM based XSS.			
	http-stored-xss	Couldn't find any stored XSS vulnerabilities.			
	http-slowloris-check	VULNERABLE: Slowloris DOS attack State: LIKELY VULNERABLE IDs: CVE:CVE-2007-6750 Slowloris tries to keep many connections to the target web server open and hold them open as long as possible. It accomplishes this by opening connections to the target web server and sending a partial request. By doing so, it starves the http server's resources causing Denial Of Service. Disclosure date: 2009-09-17 References: http://ha.ckers.org/slowloris/ https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-6750			

CONCLUSIONI:

L'attività ha permesso di eseguire una serie di scansioni di rete mirate su due ambienti distinti: ***Metasploitable2****, progettato per essere vulnerabile, e un* ***host Windows 10****, configurato in modo più realistico.*

Tramite l'uso combinato di ***Nmap****,* ***Wireshark*** *e strumenti di reportistica, è stato possibile:*

- *Identificare il sistema operativo e le versioni dei servizi attivi*
- *Analizzare il comportamento delle connessioni TCP (SYN e TCP Connect)*
- *Effettuare una scansione di vulnerabilità sui servizi rilevati*
- *Generare un report HTML leggibile tramite xsltproc*

Questa attività ha evidenziato l'importanza delle scansioni nella fase di ***ricognizione di rete****, nonché la necessità di* ***limitare l'esposizione dei servizi*** *e* ***tenere aggiornati i sistemi*** *per evitare vulnerabilità note.*