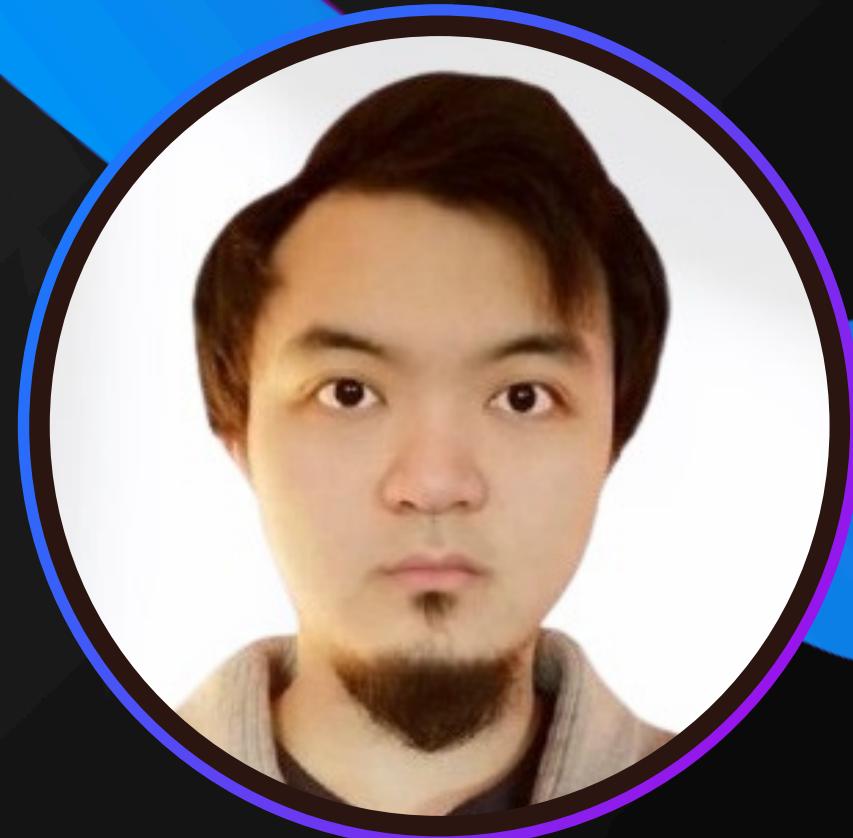




# S5\_L5 PROJECT

*Presented by: Zhongshi Liu*

10/5/2024



# Riassunto del Progetto

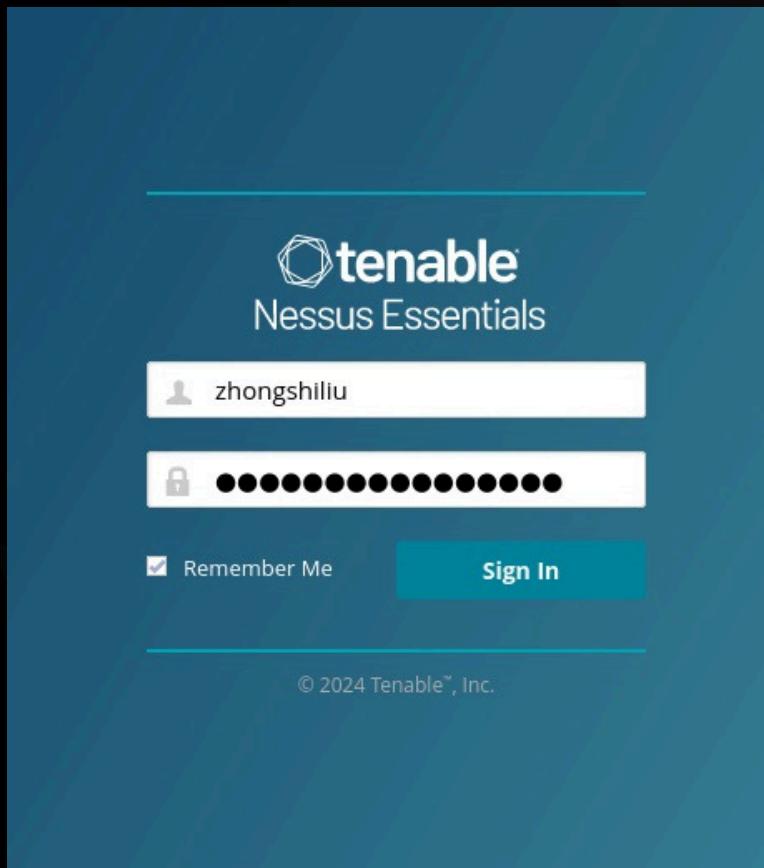
Il progetto è suddiviso in tre parti: la fase di scansione con Nessus, l'analisi e la risoluzione delle vulnerabilità critiche individuate, e infine la valutazione finale.

- Nella prima parte del progetto, verranno illustrati i procedimenti per eseguire la scansione con Nessus.
- Nella seconda fase, suddivisa in due parti, analisi e risoluzione, selezioneremo quattro delle vulnerabilità critiche individuate durante la scansione. Valuteremo il livello di criticità di ciascuna, insieme al relativo punteggio CVSS (Common Vulnerability Scoring System), e forniremo una descrizione di ciascuna vulnerabilità, insieme alle relative soluzioni. Nella parte dedicata alla risoluzione, ci concentreremo sull'eliminazione delle vulnerabilità individuate nella fase di analisi, presentando passaggi dettagliati per risolvere ciascun problema identificato.
- Infine, nell'ultima parte del progetto, eseguiremo una nuova scansione del sistema con Nessus per verificare se le vulnerabilità sono state correttamente risolte.

# Procedimenti per eseguire la scansione con Nessus

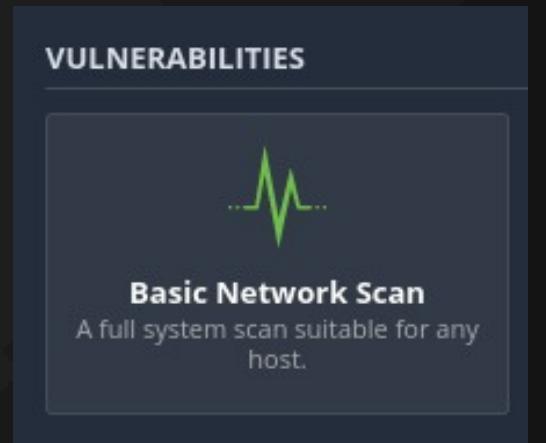
- Per avviare la scansione, useremo la macchina virtuale Kali Linux come client per eseguire il tool Nessus e la macchina Metasploitable 2 come target.
- Per avviare il servizio Nessus su Kali, utilizziamo il comando `sudo systemctl start nessusd.service` e accediamo alla sua interfaccia grafica tramite browser all'URL <https://kali:8834>.

```
(kali㉿kali)-[~/Desktop]$ sudo systemctl start nessusd.service
```



# Procedimenti per eseguire la scansione con Nessus

- Una volta sulla pagina principale di Nessus, clicchiamo sul pulsante "New Scan" nell'angolo in alto a destra del browser e selezioniamo "Basic Network Scan", che fornisce una scansione con le funzioni di base predefinite.
- Nella sezione "Basic", inseriamo l'IP target della macchina Metasploitable. Successivamente, nella sezione "Discovery", selezioniamo l'opzione "Port Scan (All Ports)". Infine, nella sezione "Assessment", selezioniamo "Scan for All Web Vulnerabilities (Complex)" per ottenere una scansione il più completa possibile.



1)

The 'BASIC' tab is selected. Under the 'General' section, the 'Name' field contains 'My scan of Meta' and the 'Targets' field contains '192.168.50.101'.

2)

The 'DISCOVERY' tab is selected. The 'Scan Type' dropdown is set to 'Port scan (all ports)'.

3)

The 'ASSESSMENT' tab is selected. The 'Scan Type' dropdown is set to 'Scan for all web vulnerabilities (complex)'.

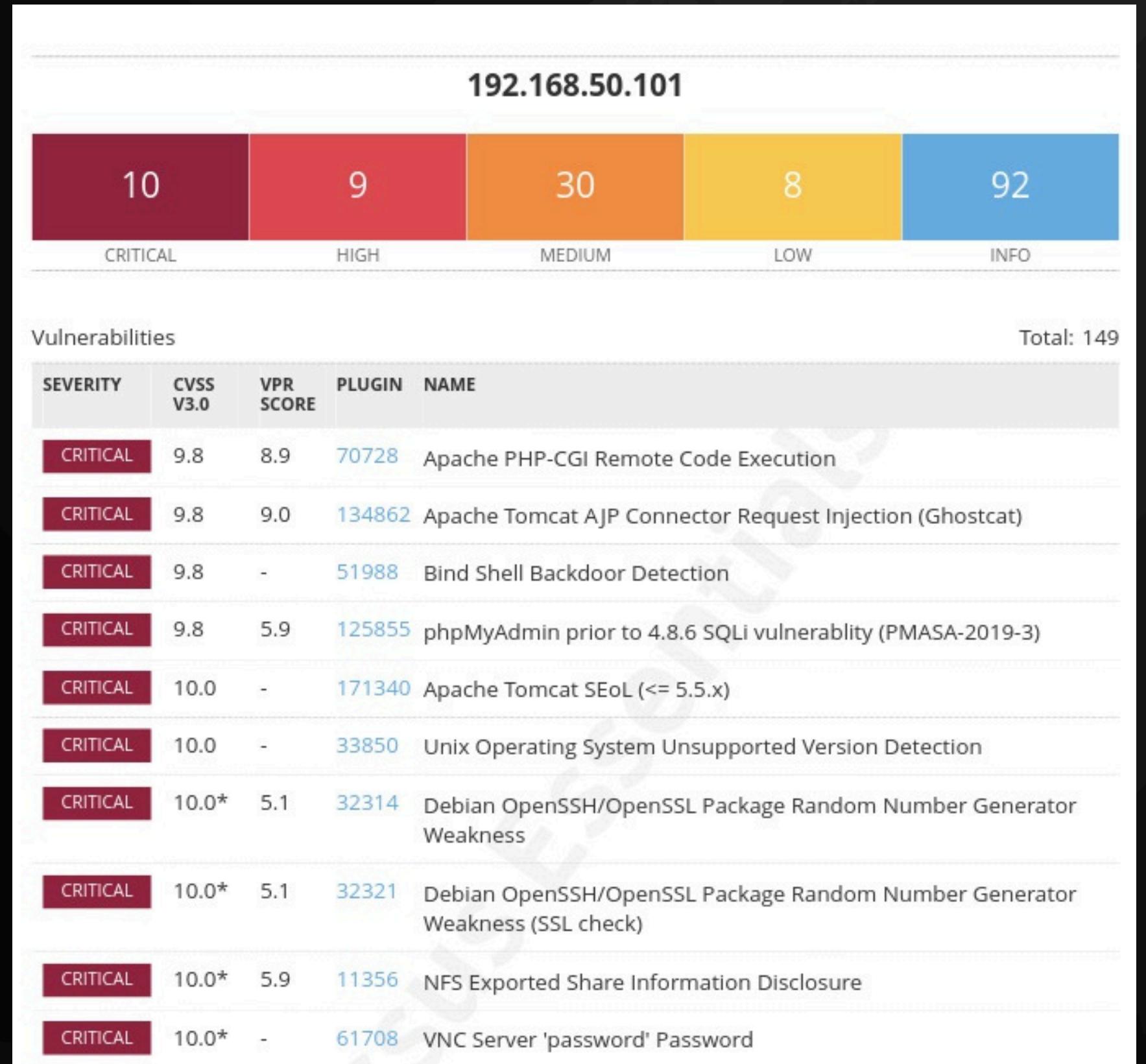


# Risultati della scansione



| Vulnerabilities 96              |          |        |       |   |
|---------------------------------|----------|--------|-------|---|
| Filter ▾ Search Vulnerabilities |          |        |       | 96 Vulnerabilities                                  |
| □                               | Sev ▾    | CVSS ▾ | VPR ▾ | Name ▲  |
| □                               | CRITICAL | 10.0 * | 5.9   | NFS Exported Share Information Disclosure           |
| □                               | CRITICAL | 10.0   |       | Unix Operating System Unsupported Version Detection |
| □                               | CRITICAL | 10.0 * |       | VNC Server 'password' Password                      |
| □                               | CRITICAL | 9.8    |       | Bind Shell Backdoor Detection                       |
| □                               | MIXED    | ...    | ...   | Apache Tomcat (Multiple Issues)                     |
| □                               | MIXED    | ...    | ...   | Phpmyadmin (Multiple Issues)                        |
| □                               | CRITICAL | ...    | ...   | SSL (Multiple Issues)                               |
| □                               | MIXED    | ...    | ...   | PHP (Multiple Issues)                               |
| □                               | HIGH     | 8.3    |       | CGI Generic SQL Injection (blind)                   |
| □                               | HIGH     | 7.5    | 5.9   | Samba Badlock Vulnerability                         |
| □                               | HIGH     | 7.5 *  |       | CGI Generic Command Execution                       |
| □                               | HIGH     | 7.5 *  |       | CGI Generic Remote File Inclusion                   |
| □                               | HIGH     | 7.5    |       | NFS Shares World Readable                           |

# Report generato da Nessus



# **Analisi - Vulnerabilità Critiche**

# VNC Server 'password' Password

CVSS v2.0: Fattore di rischio: Critico Punteggio: 10.0

- Descrizione:

Il server VNC in esecuzione sull'host remoto è protetto con una password debole. Nessus è stato in grado di effettuare l'accesso utilizzando l'autenticazione VNC e una password di 'password'. Un attaccante remoto e non autenticato potrebbe sfruttare ciò per prendere il controllo del sistema.

- Soluzione:

Proteggere il servizio VNC con una password robusta.

| Output   |                |
|--|----------------|
| Nessus logged in using a password of "password". |                |
| To see debug logs, please visit individual host  |                |
| Port ▲   | Hosts          |
| 5900 / tcp / vnc                                 | 192.168.50.101 |

# VNC Server 'password' Password

- Azione di Rimedio:

Andiamo a modificare la password del server VNC utilizzando il terminale di Metasploitable. Utilizzando il comando "sudo su", otteniamo i privilegi di amministratore.

```
msfadmin@metasploitable:~$ sudo su  
[sudo] password for msfadmin:  
root@metasploitable:/home/msfadmin# _
```

Successivamente, con il comando "vncpasswd", procediamo a cambiare la password predefinita "password" con una più sicura, nel nostro caso "LZSAdm1n". È importante notare che la password deve essere composta da non più di otto caratteri.

```
root@metasploitable:/home/msfadmin# vncpasswd  
Using password file /root/.vnc/passwd  
Password:  
Verify:  
Would you like to enter a view-only password (y/n)? n
```

# Bind Shell Backdoor Detection

CVSS v2.0: Fattore di rischio: Critico Punteggio: 9.8

- Descrizione:

Una shell è in ascolto sulla porta remota senza richiedere alcuna autenticazione. Un attaccante potrebbe utilizzarla collegandosi alla porta remota e inviando comandi direttamente.

- Soluzione:

Verifica se l'host remoto è stato compromesso e, se necessario, reinstalla il sistema.

**Output**

```
Nessus was able to execute the command "id" using the
following request :

This produced the following truncated output (limited to 10 lines) :
----- snip -----
root@metasploitable:/# uid=0(root) gid=0(root) groups=0(root)
root@metasploitable:/#
----- snip -----
```

To see debug logs, please visit individual host

| Port ▲                  | Hosts          |
|-------------------------|----------------|
| 1524 / tcp / wild_shell | 192.168.50.101 |

# Bind Shell Backdoor Detection

- Azione di Rimedio:

Questa vulnerabilità evidenzia la presenza di una backdoor che ascolta sulla porta [1524](#). Per affrontare questo problema, abbiamo due opzioni:

- A. Chiudere la porta.
- B. Applicare regole firewall per bloccare il traffico verso la porta.

## Soluzione A - Chiusura della Porta:

Nel terminale di Metasploitable, possiamo verificare lo stato della porta usando il comando netstat con i parametri –tulpn

- t: Mostra solo le connessioni TCP.
- u: Mostra solo le connessioni UDP.
- l: Mostra solo le porte in ascolto.
- p: Mostra il PID (Process ID) e il nome del programma associato a ciascuna connessione.
- n: Mostra gli indirizzi IP e numeri di porta in formato numerico, invece di risolverli in nomi simbolici.

# Bind Shell Backdoor Detection

```
root@metasploitable:~# sudo netstat -tulnp | grep 1524
tcp        0      0 0.0.0.0:1524          0.0.0.0:*                  LISTEN
4526/xinetd
root@metasploitable:~# sudo kill 4526
```

L'output del comando `sudo netstat -tulnp | grep 1524` mostrerà le informazioni riguardanti le connessioni di rete in ascolto sulle porte UDP e TCP, filtrando quelle che contengono il numero di porta 1524 e mostrando anche il PID e il nome del programma associato, se presente.

Con il comando `sudo kill "PID"`, terminiamo il processo.

Vrifichiamo la chiusura della porta:

```
(kali㉿kali)-[~]
$ sudo nmap -sS -p 1524 192.168.50.101
[sudo] password for kali:
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-05-10 07:20 EDT
Nmap scan report for 192.168.50.101
Host is up (0.00074s latency).

PORT      STATE SERVICE
1524/tcp  closed  ingreslock
MAC Address: 08:00:27:53:4C:A3 (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 13.22 seconds
```

# Bind Shell Backdoor Detection

## Soluzione B - Applicare regole firewall per bloccare il traffico verso la porta.:

Iniziamo accendendo e configurando la macchina virtuale PfSense.

prima di creare il firewall, proviamo a effettuare il ping tra Kali e Metasploitable per verificare la connessione.

Ora possiamo creare il nostro firewall con le seguenti regole:

Azione: Blocco;

Interfaccia: LAN;

Protocollo: TCP;

Sorgente: any;

Destinazione: 192.168.50.101 (Metasploitable);

Intervallo di porta di destinazione: 1524.

```
(kali㉿kali)-[~]
$ ping 192.168.50.101
PING 192.168.50.101 (192.168.50.101) 56(84) bytes of data.
64 bytes from 192.168.50.101: icmp_seq=1 ttl=63 time=1.60 ms
64 bytes from 192.168.50.101: icmp_seq=2 ttl=63 time=1.32 ms
64 bytes from 192.168.50.101: icmp_seq=3 ttl=63 time=1.29 ms
64 bytes from 192.168.50.101: icmp_seq=4 ttl=63 time=1.44 ms
64 bytes from 192.168.50.101: icmp_seq=5 ttl=63 time=1.35 ms
64 bytes from 192.168.50.101: icmp_seq=6 ttl=63 time=1.55 ms
64 bytes from 192.168.50.101: icmp_seq=7 ttl=63 time=3.31 ms
64 bytes from 192.168.50.101: icmp_seq=8 ttl=63 time=1.29 ms
64 bytes from 192.168.50.101: icmp_seq=9 ttl=63 time=1.47 ms
64 bytes from 192.168.50.101: icmp_seq=10 ttl=63 time=1.44 ms
64 bytes from 192.168.50.101: icmp_seq=11 ttl=63 time=1.56 ms
^C
— 192.168.50.101 ping statistics —
11 packets transmitted, 11 received, 0% packet loss, time 10017ms
rtt min/avg/max/mdev = 1.290/1.601/3.306/0.549 ms
```

**Firewall: Rules: Edit**

**Edit Firewall rule**

|                        |   |
|------------------------|---|
| Action                 | <input type="button" value="Block"/> <input checked="" type="checkbox"/> Block<br>Choose what to do with packets that match the criteria specified below.<br>Hint: the difference between block and reject is that with reject, a packet (TCP RST or ICMP port unreachable for UDP) is returned to the sender, whereas with block the packet is dropped silently. In either case, the original packet is discarded. |
| Disabled               | <input type="checkbox"/> <input checked="" type="checkbox"/> Disable this rule<br>Set this option to disable this rule without removing it from the list.   |
| Interface              | <input type="button" value="LAN"/> <input checked="" type="button" value="LAN"/><br>Choose on which interface packets must come in to match this rule.  |
| Protocol               | <input type="button" value="TCP"/> <input checked="" type="button" value="TCP"/><br>Choose which IP protocol this rule should match.<br>Hint: in most cases, you should specify TCP here.   |
| Source                 | <input type="checkbox"/> <input checked="" type="checkbox"/> not<br>Use this option to invert the sense of the match.<br>Type: <input type="button" value="any"/> Address: <input type="text" value=""/> / <input type="button" value="Advanced"/> - Show source port range   |
| Destination            | <input type="checkbox"/> <input checked="" type="checkbox"/> not<br>Use this option to invert the sense of the match.<br>Type: <input type="button" value="Single host or alias"/> Address: <input type="text" value="192.168.50.101"/> / <input type="button" value="Advanced"/>   |
| Destination port range | from: <input type="button" value="(other)"/> <input type="text" value="1524"/><br>to: <input type="button" value="(other)"/> <input type="text" value="1524"/><br>Specify the port or port range for the destination of the packet for this rule.<br>Hint: you can leave the 'to' field empty if you only want to filter a single port  |
| Log                    | <input checked="" type="checkbox"/> <input type="checkbox"/> Log packets that are handled by this rule<br>Hint: the firewall has limited local log space. Don't turn on logging for everything. If you want to do a lot of logging, consider using a remote syslog server (see the Diagnostics: System logs: Settings page).  |
| Description            | <input type="text" value=""/> You may enter a description here for your reference.  |

# Bind Shell Backdoor Detection

Il firewall creato blocca tutte le connessioni in arrivo da qualsiasi indirizzo verso la porta di Metasploitable.

|  |     |   |   |                |      |   |      |  |
|--|-----|---|---|----------------|------|---|------|--|
|  | TCP | * | * | 192.168.50.101 | 1524 | * | none |  |
|--|-----|---|---|----------------|------|---|------|--|

Effettuiamo un test tramite il terminale di Kali con nmap. Possiamo constatare che la porta 1524 ha l'ingresso bloccato.

```
(kali㉿kali)-[~]
└─$ sudo nmap -sS -Pn -p 1524 192.168.50.101
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-05-10 10:29 EDT
Nmap scan report for 192.168.50.101
Host is up.

PORT      STATE      SERVICE
1524/tcp  filtered  ingreslock

Nmap done: 1 IP address (1 host up) scanned in 15.07 seconds
```

# NFS Exported Share Information Disclosure

CVSS v2.0: Fattore di rischio: Critico Punteggio: 10.0

- Descrizione:

Almeno una delle condivisioni NFS esportate dal server remoto potrebbe essere montata dall'host di scansione. Un attaccante potrebbe sfruttare questo per leggere (e eventualmente scrivere) file sull'host remoto.

- Soluzione:

Configurare NFS sull'host remoto in modo che solo gli host autorizzati possano montare le sue condivisioni remote.

**Output**

```
The following NFS shares could be mounted :  
+ /  
+ Contents of / :  
- .  
- ..  
- bin  
- boot  
- cdrom  
- dev  
- etc  
- home  
- initrd  
- initrd.img  
- lib  
- lost+found  
- media  
- mnt  
- nohup.out  
- opt  
- proc  
- root  
- sbin  
- srv  
- sys  
- tmp  
- usr  
- var  
- vmlinuz  
less...  
To see debug logs, please visit individual host  


| Port ▲               | Hosts          |
|----------------------|----------------|
| 2049 / udp / rpc-nfs | 192.168.50.101 |


```

# NFS Exported Share Information Disclosure

- Azione di Rimedio:

Per risolvere questo problema abbiamo due potenziali soluzione :

- A. modificare i file di configurazione in modo tale da restringere la cartelle condivisibili ed allo stesso tempo limitare le macchine che possono accedere.
- B. Creare delle FW rules addizionali per gestire le connessioni.

## Soluzione A – Modificare i File di Configurazione:

Per risolvere questa vulnerabilità, possiamo modificare i file di configurazione per limitare le cartelle condivisibili e controllare l'accesso alle macchine autorizzate. NFS può essere configurato attraverso tre file principali:

1. [`/etc\(exports`](#): Questo file contiene l'elenco dei volumi che possono essere condivisi. Qui possiamo specificare le directory da esporre e configurare le autorizzazioni di accesso per utenti e gruppi.
2. [`/etc/hosts.allow`](#): Questo file consente di specificare quali computer sulla rete sono autorizzati ad accedere alle condivisioni NFS. Possiamo definire regole basate su indirizzi IP, nomi host o reti IP.
3. [`/etc/hosts.deny`](#): Questo file consente di specificare quali computer sulla rete sono esclusi dall'accesso alle condivisioni NFS. Possiamo utilizzarlo per bloccare l'accesso da determinati indirizzi IP o reti.

# NFS Exported Share Information Disclosure

1) /etc/exports: Dal terminale di Metasploitable, modifichiamo il file di configurazione "exports". Qui apportiamo modifiche ai permessi di scrittura e lettura nella directory condivisibile, impostandoli su sola lettura (Read-only).

```
GNU nano 2.0.7          File: /etc/exports

# /etc/exports: the access control list for filesystems which may be exported
#               to NFS clients. See exports(5).
#
# Example for NFSv2 and NFSv3:
# /srv/homes      hostname1(rw,sync) hostname2(ro,sync)
#
# Example for NFSv4:
# /srv/nfs4        gss/krb5i(rw,sync,fsid=0,crossmnt)
# /srv/nfs4/homes  gss/krb5i(rw,sync)
#
# *(ro,sync,no_root_squash,no_subtree_check)
```

2) /etc/hosts.allow: : Dal terminale di Metasploitable, possiamo modificare il file di configurazione "hosts.allow". Qui possiamo inserire gli indirizzi IP o le reti che desideriamo consentire l'accesso dopo il blocco generale "ALL:".

```
GNU nano 2.0.7          File: /etc/hosts.allow          Modified

# /etc/hosts.allow: list of hosts that are allowed to access the system.
#                   See the manual pages hosts_access(5) and hosts_options(5).
#
# Example:   ALL: LOCAL @some_netgroup
#             ALL: .foobar.edu EXCEPT terminalserver.foobar.edu
#
# If you're going to protect the portmapper use the name "portmap" for the
# daemon name. Remember that you can only use the keyword "ALL" and IP
# addresses (NOT host or domain names) for the portmapper, as well as for
# rpc.mountd (the NFS mount daemon). See portmap(8) and rpc.mountd(8)
# for further information.
#
ALL: [REDACTED]
```

# NFS Exported Share Information Disclosure

3) /etc/hosts.deny: : Dal terminale di Metasploitable, modifichiamo il file di configurazione "hosts.deny". Qui possiamo inserire gli indirizzi IP o le reti che desideriamo negare l'accesso dopo il blocco generale "ALL:".

```
GNU nano 2.0.7           File: /etc/hosts.deny           Modified

# /etc/hosts.deny: list of hosts that are _not_ allowed to access the system.
#                   See the manual pages hosts_access(5) and hosts_options(5).
#
# Example:      ALL: some.host.name, .some.domain
#               ALL EXCEPT in.fingerd: other.host.name, .other.domain
#
# If you're going to protect the portmapper use the name "portmap" for the
# daemon name. Remember that you can only use the keyword "ALL" and IP
# addresses (NOT host or domain names) for the portmapper, as well as for
# rpc.mountd (the NFS mount daemon). See portmap(8) and rpc.mountd(8)
# for further information.
#
# The PARANOID wildcard matches any host whose name does not match its
# address.
#
# You may wish to enable this to ensure any programs that don't
# validate looked up hostnames still leave understandable logs. In past
# versions of Debian this has been the default.
# ALL: [REDACTED]
```

# NFS Exported Share Information Disclosure

Soluzione B – Applicare regole del firewall per bloccare gli indirizzi IP non desiderati verso la porta

Creiamo una regola firewall per bloccare gli indirizzi IP non desiderati in ingresso verso la porta 2049. Nell'esempio che ho fornito, ho bloccato tutte le connessioni in ingresso provenienti da qualsiasi indirizzo verso la porta 2049 di Metasploitable. Tuttavia, in pratica, possiamo creare delle regole firewall che bloccano solo gli indirizzi specificati.

|                          |  |  |     |   |   |                |      |   |      |  |  |
|--------------------------|--|--|-----|---|---|----------------|------|---|------|--|--|
| <input type="checkbox"/> |  |  | TCP | * | * | 192.168.50.101 | 2049 | * | none |  |  |
|--------------------------|--|--|-----|---|---|----------------|------|---|------|--|--|

Effettuiamo un test tramite il terminale di Kali con nmap. Possiamo constatare che la porta 2049 e' filtered.

```
(kali㉿kali)-[~]
$ sudo nmap -sS -Pn -p 2049 192.168.50.101
[sudo] password for kali:
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-05-12 10:29 EDT
Nmap scan report for 192.168.50.101
Host is up.

PORT      STATE      SERVICE
2049/tcp  filtered  nfs

Nmap done: 1 IP address (1 host up) scanned in 2.16 seconds
```

# Debian OpenSSH/OpenSSL Package Random Number Generator Weakness

CVSS v2.0: Fattore di rischio: Critico Punteggio: 10.0

- Descrizione:

La chiave host SSH remota è stata generata su un sistema Debian o Ubuntu che contiene un bug nel generatore di numeri casuali della sua libreria OpenSSL.

Il problema è dovuto a un pacchettizzatore Debian che ha rimosso praticamente tutte le fonti di entropia nella versione remota di OpenSSL.

Un attaccante può facilmente ottenere la parte privata della chiave remota e utilizzarla per decifrare la sessione remota o configurare un attacco man-in-the-middle.

- Soluzione:

Considera tutto il materiale crittografico generato sull'host remoto come indovinabile. In particolare, il chiave SSH dovrebbe essere rigenerato.

| Output  |                |
|---|----------------|
| No output recorded.                             |                |
| To see debug logs, please visit individual host |                |
| Port ▲  | Hosts          |
| 22 / tcp / ssh                                  | 192.168.50.101 |

# Debian OpenSSH/OpenSSL Package Random Number Generator Weakness

- Azione di Rimedio:

La soluzione più diretta è la rigenerazione delle chiavi SSH utilizzando un metodo sicuro. Per farlo, possiamo utilizzare il comando `ssh-keygen`.

Apriamo il terminale sul nostro Metasploitable e inseriamo il seguente comando: `ssh-keygen -t rsa -b 4096`. In questo comando:

- t** specifica il tipo di algoritmo da utilizzare per la generazione delle chiavi (in questo caso, RSA);
- b** specifica la lunghezza, in bit, della chiave generata.

```
msfadmin@metasploitable:~$ ssh-keygen -t rsa -b 4096
```

Il comando `ssh-keygen` ci chiederà dove salvare la chiave. Possiamo premere semplicemente Invio per accettare la posizione predefinita oppure scegliere di salvarla in un percorso personalizzato.

```
Enter file in which to save the key (/home/msfadmin/.ssh/id_rsa):  
/home/msfadmin/.ssh/id_rsa already exists.  
Overwrite (y/n)? y
```

# Debian OpenSSH/OpenSSL Package Random Number Generator Weakness

Possiamo anche optare per proteggere la chiave con una passphrase. Nel mio caso, l'ho salvata come "SonoLZS".

Dopo aver completato questi passaggi, il comando ssh-keygen genererà la coppia di chiavi SSH per noi.

```
msfadmin@metasploitable:~$ ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/home/msfadmin/.ssh/id_rsa):
/home/msfadmin/.ssh/id_rsa already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/msfadmin/.ssh/id_rsa.
Your public key has been saved in /home/msfadmin/.ssh/id_rsa.pub.
The key fingerprint is:
88:bb:b7:a2:d5:55:5a:e9:25:4f:13:4e:96:d3:4a:6b msfadmin@metasploitable
msfadmin@metasploitable:~$
```

# Conclusione

L'implementazione delle azioni di rimedio ha svolto un ruolo chiave nel potenziare la sicurezza del sistema, riducendo il rischio di violazioni e garantendo l'integrità delle informazioni. La scansione successiva ha confermato un effettivo miglioramento della sicurezza.



| Filter ▾                 |          |        |       | Search Vulnerabilities                              | 96 Vulnerabilities |         |  |  |
|--------------------------|----------|--------|-------|---|--------------------|---------|--|--|
| <input type="checkbox"/> | Sev ▾    | CVSS ▾ | VPR ▾ | Name ▾  | Family ▾           | Count ▾ |  |  |
| <input type="checkbox"/> | CRITICAL | 10.0   |       | Unix Operating System Unsupported Version Detection | General            | 1       |  |  |
| <input type="checkbox"/> | MIXED    | ...    | ...   | Apache Tomcat (Multiple Issues)                     | Web Servers        | 4       |  |  |
| <input type="checkbox"/> | MIXED    | ...    | ...   | Phpmyadmin (Multiple Issues)                        | CGI abuses         | 4       |  |  |
| <input type="checkbox"/> | MIXED    | ...    | ...   | PHP (Multiple Issues)                               | CGI abuses         | 3       |  |  |
| <input type="checkbox"/> | HIGH     | 8.3    |       | CGI Generic SQL Injection (blind)                   | CGI abuses         | 1       |  |  |
| <input type="checkbox"/> | HIGH     | 7.5    | 5.9   | Samba Badlock Vulnerability                         | General            | 1       |  |  |
| <input type="checkbox"/> | HIGH     | 7.5 *  |       | CGI Generic Command Execution                       | CGI abuses         | 1       |  |  |
| <input type="checkbox"/> | HIGH     | 7.5 *  |       | CGI Generic Remote File Inclusion                   | CGI abuses         | 1       |  |  |
| <input type="checkbox"/> | HIGH     | 7.5    |       | NFS Shares World Readable                           | RPC                | 1       |  |  |