

## Report – M3W12D5 – Fase di remediation

### Introduzione:

Obbiettivo di questa esercitazione è individuare tramite l'utilizzo del tool Nessus le vulnerabilità della macchina Metasploitable2, tra le critiche sceglierne un certo numero e implementare delle azioni di rimedio, infine effettuare una seconda scansione per verificarne l'efficacia.

### Remediation:

In questa fase dell'esercitazione verranno scelte le vulnerabilità, verranno approfondite e capite, poi si proporranno ed eseguiranno delle azioni per porvi rimedio.

1.

CRITICAL

10.0

10.0

156016

Apache Log4Shell RCE detection via Path Enumeration (Direct Check HTTP)

**Approfondimento:** la libreria Log4j è affetta da molte vulnerabilità che permettono l'esecuzione di codice da remoto, un metodo semplice e diretto consiste nell'aggiornamento ad una versione più recente nella quale queste vulnerabilità sono state corrette.

**Remediation:** iniziamo verificando la versione di Log4j installata su Meta con il comando:

- `sudo find / -name 'log4j*'`

```
msfadmin@metasploitable:~$ sudo find / -name 'log4j*'
[sudo] password for msfadmin:
/usr/share/java/log4j-1.2.15.jar
/usr/share/java/log4j-1.2.jar
/usr/share/doc/libcommons-launcher-java/examples/example/src/etc/log4j.xml
```

Dopodiché bisogna scaricare il pacchetto dal sito con il comando:

- `wget https://downloads.apache.org/logging/log4j/2.3.2/apache-log4j-2.3.2-bin.tar.gz`

poi i file vanno estratti dall'archivio e sostituiti a quelli presenti in /usr/share/java.

Sfortunatamente Metasploitable2 essendo progettata per essere una macchina vulnerabile non consente di effettuare aggiornamenti.

```
msfadmin@metasploitable:~$ sudo wget https://downloads.apache.org/logging/log4j/2.3.2/apache-log4j-2.3.2-bin.tar.gz
--13:53:51-- https://downloads.apache.org/logging/log4j/2.3.2/apache-log4j-2.3.2-bin.tar.gz
=> 'apache-log4j-2.3.2-bin.tar.gz'
Resolving downloads.apache.org... 127.0.0.1
Connecting to downloads.apache.org|127.0.0.1|:443... failed: Connection refused.
```

2.

CRITICAL

10.0\*

5.9

11356

NFS Exported Share Information Disclosure

**Approfondimento:** NFS (network file system) è un sistema per accedere ad un computer remoto come se fosse locale e visualizzare, caricare/scaricare o condividere file.

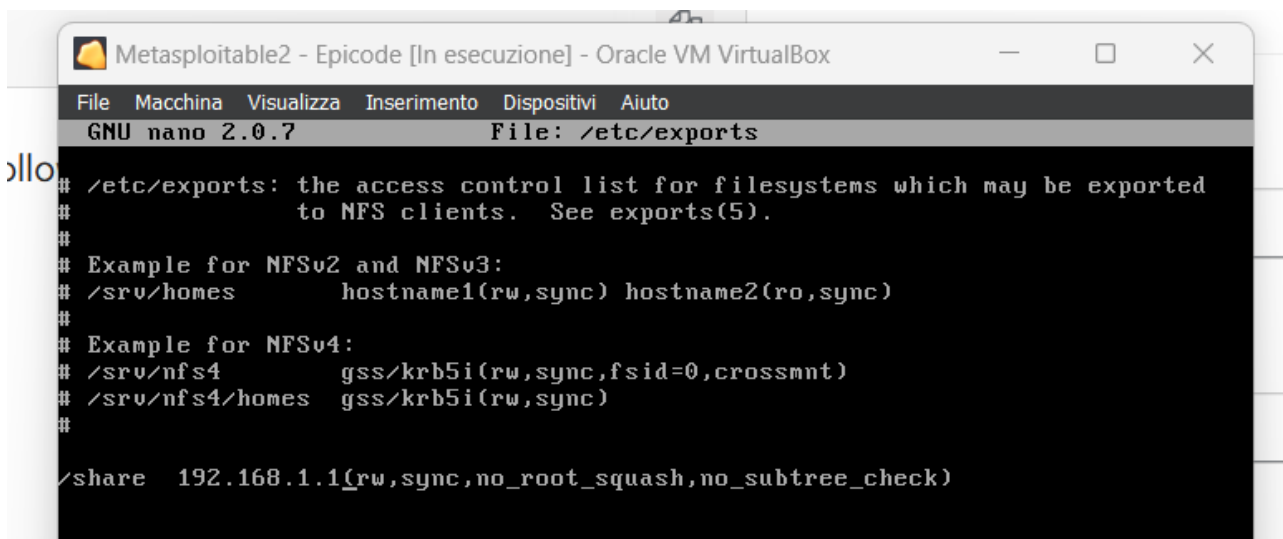
Se il servizio non è configurato correttamente, l'attaccante può essere in grado di visualizzare e quindi tentare di sfruttare questa vulnerabilità per caricare/scaricare file a cui non dovrebbe avere accesso.

La soluzione consiste nella configurazione corretta del servizio in modo da consentire l'accesso solo ad utenti autorizzati.

**Remediation:** La configurazione può essere modificata editando il file `/etc/exports` per esempio con il comando:

- `sudo nano /etc/exports`

Notiamo che viene condiviso l'intero file system, quindi possiamo editare il percorso e limitarlo ad una sola cartella, inoltre invece di consentire l'accesso a chiunque con “\*” lo limitiamo ad un host specifico, in questo caso 192.168.1.1 di pfsense anche se ha poco sense (ha!) giusto a scopo didattico.



The screenshot shows a window titled "Metasploitable2 - Epicode [In esecuzione] - Oracle VM VirtualBox". Inside, the GNU nano 2.0.7 text editor is open to the file `/etc/exports`. The file content is as follows:

```
# /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)
#
/share 192.168.1.1(rw,sync,no_root_squash,no_subtree_check)
```

3.

CRITICAL

10.0\*

-

61708

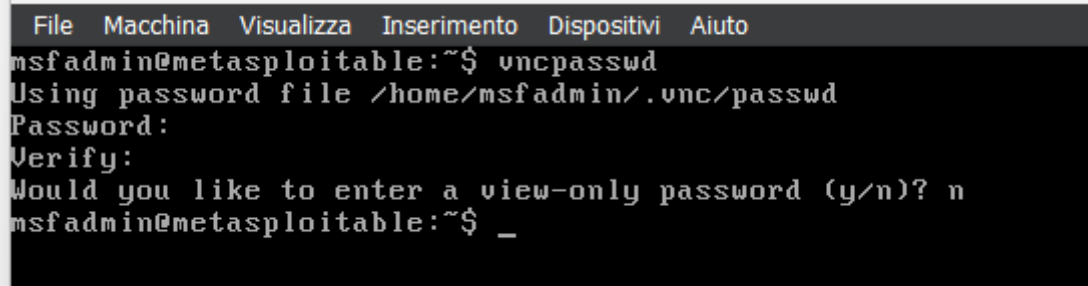
VNC Server 'password' Password

**Approfondimento:** Il server VNC è protetto da una password debole e questo può consentire un facile accesso ad un attaccante, siccome i VNC (virtual network computing) sono software per la connessione remota dove un server invia lo stream del proprio desktop al client e quest'ultimo può operare come se fosse davanti al computer stesso, visualizzando file e eseguendo codice.

**Remediation:** Se la password è debole la soluzione più diretta consiste nel cambiarla in una più forte come ad esempio: +A9h`8\*

Per cambiare la password dobbiamo editare il file di configurazione con il comando:

- vncpasswd



```
File Macchina Visualizza Inserimento Dispositivi Aiuto
msfadmin@metasploitable:~$ vncpasswd
Using password file /home/msfadmin/.vnc/passwd
Password:
Verify:
Would you like to enter a view-only password (y/n)? n
msfadmin@metasploitable:~$ _
```

Successivamente possiamo aggiungere una regola al firewall iptables per bloccare il traffico in entrata sulla porta 5900 con il comando:

- sudo iptables -A INPUT -p tcp --dport 5900 -j DROP

Un altro metodo potrebbe essere quello di generare chiavi pubblica e privata così che per l'attaccante sia quasi impossibile penetrare la sicurezza, per esempio ho trovato questa guida: <https://ubuntuforums.org/showthread.php?t=383053>

4.

CRITICAL

9.8

-

51988

Bind Shell Backdoor Detection

**Approfondimento:** Sulla porta 1524 è in ascolto una shell che non richiede nessuna autenticazione e può essere sfruttata per ottenere completo accesso alla macchina.

**Remediation:** Si potrebbe aggiungere una regola al firewall, sia su iptables che pfsense. Approfondendo la ricerca ho scoperto che il servizio attivo sulla porta è un xinetd (super server daemon), questo ascolta le richieste in arrivo e in base alla porta utilizzata per la connessione lancia il servizio appropriato.

Quindi un'altra soluzione consiste nel modificare le impostazioni di xinetd per disattivare la shell, per esempio usando il comando:

- `sudo nano /etc/inetd.conf`

```
GNU nano 2.0.7      File: /etc/inetd.conf
#<off># netbios-ssn    stream  tcp    nowait  root    /usr/sbin/tcpd  /usr/sbin/in.telnetd
telnet               stream  tcp    nowait  telnetd /usr/sbin/tcpd  /usr/sbin/in.telnetd
#<off># ftp           stream  tcp    nowait  root    /usr/sbin/tcpd  /usr/sbin/in.tftpd
tftp                 dgram  udp    wait    nobody   /usr/sbin/tcpd  /usr/sbin/in.tftpd
shell                stream  tcp    nowait  root    /usr/sbin/tcpd  /usr/sbin/in.rshd
login                stream  tcp    nowait  root    /usr/sbin/tcpd  /usr/sbin/in.rlogind
rexec                 stream  tcp    nowait  root    /usr/sbin/tcpd  /usr/sbin/in.rexecd
ingreslock stream tcp nowait root /bin/bash bash -i
```

Il servizio colpevole della creazione della shell è l'ultimo, che quindi provvedo a cancellare.

5.

CRITICAL

9.8

-

20007

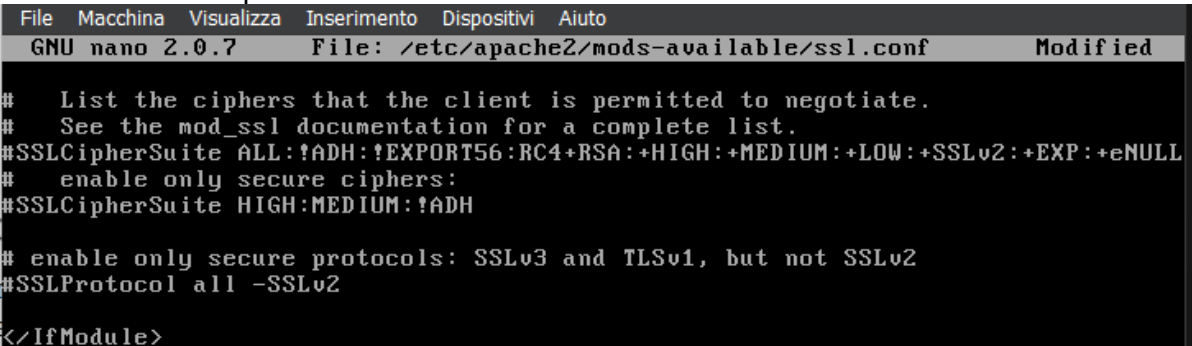
SSL Version 2 and 3 Protocol Detection

**Approfondimento:** Il servizio remoto utilizza i protocolli SSL v2 o v3 per criptare le comunicazioni, tuttavia questi hanno delle note debolezze permettendo ad un attaccante di decrittare le comunicazioni o iniziare un attacco man-in-the-middle.

**Remediation:** Viene consigliata la disattivazione di questi due protocolli a favore di TLS 1.2 o più recente.

Per fare ciò editiamo il file di configurazione con il comando:

- `sudo nano /etc/apache2/mods-available/ssl.conf`



```
File  Macchina  Visualizza  Inserimento  Dispositivi  Aiuto
GNU nano 2.0.7  File: /etc/apache2/mods-available/ssl.conf  Modified

# List the ciphers that the client is permitted to negotiate.
# See the mod_ssl documentation for a complete list.
#SSLCipherSuite ALL:!ADH:!EXPORT56:RC4+RSA:+HIGH:+MEDIUM:+LOW:+SSLv2:+EXP:+eNULL
# enable only secure ciphers:
#SSLCipherSuite HIGH:MEDIUM:!ADH

# enable only secure protocols: SSLv3 and TLSv1, but not SSLv2
#SSLProtocol all -SSLv2

</IfModule>
```

decommentiamo la riga e aggiungiamo anche v3 in modo da far scegliere solo TLS

```
# enable only secure protocols: TLSv1, but not SSLv2 or SSLv3_
SSLProtocol all -SSLv2 -SSLv3
```