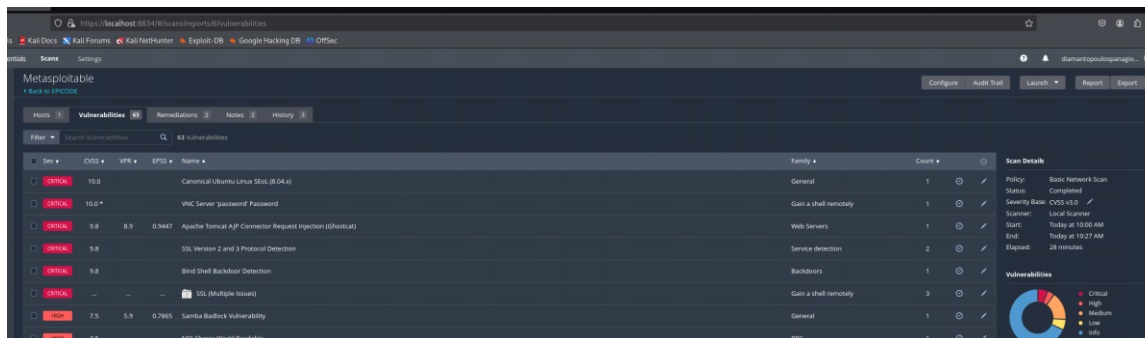


## Report di risoluzione vulnerabilità VNC password su Metasploitable

### Problema:

La scansione di vulnerabilità con Nessus ha rilevato che il server VNC su Metasploitable utilizzava una password debole e predefinita, precisamente la stringa "password". Questo rappresentava un rischio di sicurezza, in quanto una password troppo semplice facilita accessi non autorizzati.



### Azioni intraprese per la risoluzione:

Individuazione del processo VNC attivo tramite il comando:

```
ps aux | grep Xtightvnc
```

è stato identificato il processo Xtightvnc in esecuzione sul display :0, con PID 4601.

Terminazione del processo VNC attivo

Per permettere la modifica della password, è stato necessario terminare manualmente il processo VNC in esecuzione, eseguendo:

```
kill 4601
```

Questa azione ha interrotto il server VNC attivo sul display :0.

Cambio della password VNC

Con il server VNC fermo, è stata aggiornata la password usando il comando:

```
vncpasswd
```

Inserendo la nuova password sicura scelta: "Ottagono".

Questo comando ha aggiornato il file di configurazione della password cifrata, assicurando che il server VNC richieda la nuova password all'accesso.

## Riavvio del server VNC

Dopo la modifica della password, il server VNC è stato riavviato con:

```
vncserver :0
```

In questo modo il servizio è ripartito caricando la nuova configurazione, inclusa la password aggiornata.

```
msfadmin@metasploitable:~$ vncserver :0
xauth: creating new authority file /home/msfadmin/.Xauthority

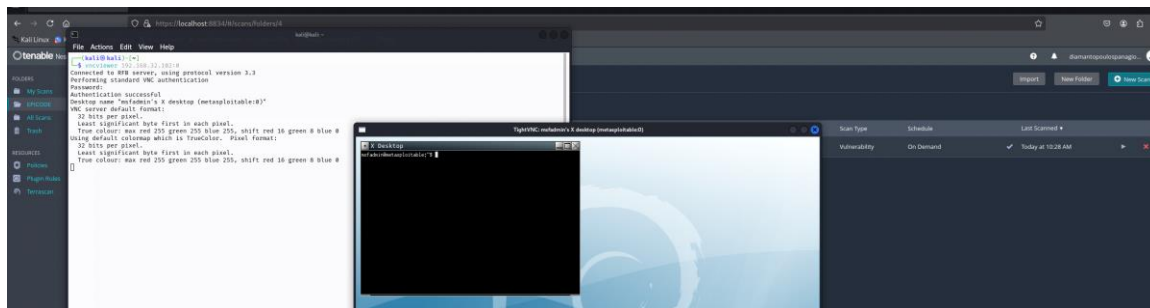
New 'X' desktop is metasploitable:0

Creating default startup script /home/msfadmin/.vnc/xstartup
Starting applications specified in /home/msfadmin/.vnc/xstartup
Log file is /home/msfadmin/.vnc/metasploitable:0.log
msfadmin@metasploitable:~$ _
```

## Risultato:

Testata la connessione da Kali a Metasploitable utilizzando il seguente comando:  
vncviewer 192.168.32.102:0

(Indirizzo IP Metasploitable 192.168.32.102 - Indirizzo IP Kali Linux 192.168.32.100)



La modifica della password ha risolto la vulnerabilità segnalata da Nessus. La password di accesso al server VNC non è più la predefinita e debole "password", ma una stringa personalizzata e più robusta, migliorando significativamente la sicurezza del sistema Metasploitable.

*Panagiotis Diamantopoulos*

## Report Mitigazione Vulnerabilità SSL Version 2 and 3 Protocol Detection

### Contesto

Durante la scansione di sicurezza con Nessus sulla macchina Metasploitable, è stata rilevata la presenza dei protocolli SSLv2 e SSLv3 abilitati sul servizio Apache HTTPS (porta 443). Questi protocolli sono noti per gravi vulnerabilità crittografiche, come il rischio di attacchi man-in-the-middle, padding oracle e la nota vulnerabilità POODLE.

### Obiettivo

Mitigare la vulnerabilità senza aggiornare il software né disabilitare il servizio Apache, mantenendo la funzionalità HTTPS e migliorando la sicurezza delle connessioni.

### Azioni intraprese

Accesso e verifica

Collegamento alla macchina Metasploitable e verifica dello stato del servizio Apache e delle porte in ascolto.

Abilitazione modulo SSL di Apache

Attivazione del modulo SSL per supportare HTTPS, se non già abilitato:

```
sudo a2enmod ssl
```

```
sudo /etc/init.d/apache2 restart
```

Abilitazione sito SSL

Abilitazione del sito SSL predefinito per Apache:

```
sudo a2ensite default-ssl
```

```
sudo /etc/init.d/apache2 restart
```

Disabilitazione dei protocolli SSLv2 e SSLv3

Modifica del file di configurazione SSL di Apache (tipicamente /etc/apache2/mods-available/ssl.conf o equivalente), aggiungendo/modificando la direttiva:

```
SSLProtocol all -SSLv2 -SSLv3
```

Questo comando indica ad Apache di accettare tutti i protocolli tranne SSLv2 e SSLv3, eliminando così i protocolli obsoleti e vulnerabili.

Configurazione delle suite di cifratura

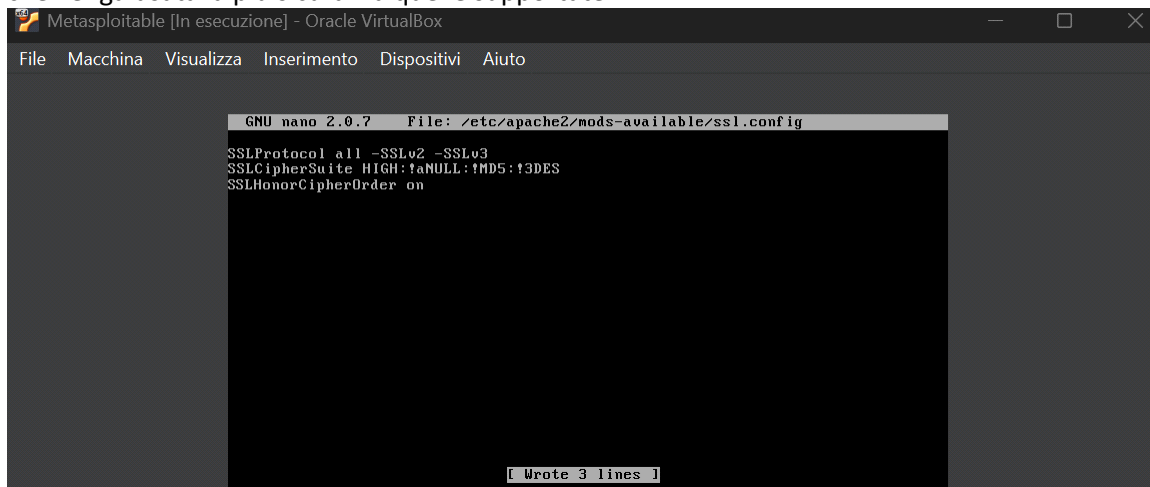
Per rafforzare ulteriormente la sicurezza, nel file di configurazione SSL sono state aggiunte le seguenti direttive:

*SSLCipherSuite HIGH:!aNULL:!MD5:!3DES*

*SSLHonorCipherOrder on*

SSLCipherSuite HIGH:!aNULL:!MD5:!3DES limita l'uso delle cifrature solo a quelle considerate forti, escludendo cifrature anonime, con MD5 e 3DES ritenute insicure.

SSLHonorCipherOrder on impone al server di scegliere la suite di cifratura preferita, assicurando che venga usata la più sicura fra quelle supportate.

A screenshot of a terminal window titled "Metasploitable [In esecuzione] - Oracle VirtualBox". The window shows a nano text editor editing the file "/etc/apache2/mods-available/ssl.config". The content of the file is as follows:

```
GNU nano 2.0.7 File: /etc/apache2/mods-available/ssl.config
SSLProtocol all -SSLv2 -SSLv3
SSLCipherSuite HIGH:!aNULL:!MD5:!3DES
SSLHonorCipherOrder on
```

At the bottom of the terminal, a status bar indicates "[ Wrote 3 lines ]".

Riavvio del servizio Apache

Per applicare tutte le modifiche è stato riavviato Apache con:

`sudo /etc/init.d/apache2 restart`

## Verifica della mitigazione

Da Kali Linux, tramite scansione Nmap specifica:

`nmap --script ssl-enum-ciphers -p 443 192.168.32.102`

È stata confermata la disabilitazione di SSLv2 e SSLv3, e la corretta applicazione delle suite di cifratura forti (unico protocollo attivo tcp 443 https)

```
(kali@kali)-[~]
$ nmap --script ssl-enum-ciphers -p 443 192.168.32.102
Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-17 13:31 EDT
Nmap scan report for 192.168.32.102
Host is up (0.0012s latency).

PORT      STATE SERVICE
443/tcp   open  https
MAC Address: 08:00:27:1A:5B:B2 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 38.70 seconds

(kali@kali)-[~]
$
```

## Conclusioni

La vulnerabilità legata all'uso di SSLv2 e SSLv3 è stata mitigata senza disabilitare il servizio Apache né effettuare aggiornamenti software.

Il servizio HTTPS rimane attivo e funzionante, garantendo comunicazioni cifrate sicure.

La configurazione delle suite di cifratura assicura l'uso di algoritmi robusti e protegge da downgrade verso cifrature deboli.

La soluzione adottata è una best practice di sicurezza, conforme agli standard attuali, che riduce significativamente il rischio di attacchi crittografici e man-in-the-middle.

- Panagiotis Diamantopoulos

## Mitigazione vulnerabilità: Debian OpenSSH/OpenSSL Package Random Number Generator Weakness

Durante la scansione iniziale eseguita con Nessus, sono state individuate due vulnerabilità critiche strettamente collegate tra loro:

Debian OpenSSH/OpenSSL Package Random Number Generator Weakness

Debian OpenSSL Package Random Number Generator Weakness (SSL check)

Entrambe derivano da un bug storico in alcune versioni di Debian (2006–2008), dove il generatore di numeri casuali (PRNG) di OpenSSL non generava sufficiente entropia. Questo ha causato la creazione di chiavi crittografiche deboli e prevedibili, potenzialmente soggette ad attacchi di tipo brute-force o man-in-the-middle.

La vulnerabilità interessava sia i servizi SSH (tramite chiavi host deboli), sia i servizi SSL (tramite certificati autofirmati non sicuri).

### Azioni correttive adottate

#### 1. Rigenerazione chiavi SSH vulnerabili

Per mitigare il rischio legato al servizio SSH, ho rigenerato manualmente le chiavi host RSA e DSA utilizzando i seguenti comandi:

```
sudo ssh-keygen -t rsa -f /etc/ssh/ssh_host_rsa_key
```

```
sudo ssh-keygen -t dsa -f /etc/ssh/ssh_host_dsa_key
```

Durante la procedura:

È stata confermata la sovrascrittura delle chiavi esistenti.

La passphrase è stata lasciata vuota, premendo semplicemente Invio quando richiesto.

Le chiavi ECDSA e ED25519 non sono state rigenerate perché non supportate dal sistema legacy in uso (errore: unknown key type).

Al termine, ho riavviato il servizio SSH per applicare i cambiamenti:

```
sudo /etc/init.d/ssh restart
```

```

Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /etc/ssh/ssh_host_rsa_key.
Your public key has been saved in /etc/ssh/ssh_host_rsa_key.pub.
The key fingerprint is:
7e:ca:93:c4:fe:3b:35:c3:73:73:d2:77:9d:dc:0e:38 root@metasploitable
msfadmin@metasploitable:~$ sudo ssh-keygen -t ecdsa -f /etc/ssh/ssh_host_ecdsa_k
ey
unknown key type ecdsa
msfadmin@metasploitable:~$ sudo ssh-keygen -t ecdsa -f /etc/ssh/ssh_host_ecdsa_k
ey
unknown key type ecdsa
msfadmin@metasploitable:~$ sudo ssh-keygen -t dsa -f /etc/ssh/ssh_host_dsa_key
Generating public/private dsa key pair.
/etc/ssh/ssh_host_dsa_key already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /etc/ssh/ssh_host_dsa_key.
Your public key has been saved in /etc/ssh/ssh_host_dsa_key.pub.
The key fingerprint is:
3f:4a:6c:76:d1:57:ec:12:b9:2c:ce:51:91:63:4c:cf root@metasploitable
msfadmin@metasploitable:~$ sudo /etc/init.d/ssh restart
* Restarting OpenBSD Secure Shell server sshd [ OK ]
msfadmin@metasploitable:~$

```

## 2. Rigenerazione certificato SSL autofirmato vulnerabile

Per la seconda vulnerabilità, relativa al servizio SSL, è stato identificato il certificato autofirmato predefinito presente in `/etc/ssl/certs/ssl-cert-snakeoil.pem`, potenzialmente generato in condizioni di bassa entropia.

Per rigenerarlo, ho utilizzato il comando:

```
sudo make-ssl-cert generate-default-snakeoil --force-overwrite
```

Questo ha prodotto una nuova coppia di certificati autofirmati (chiave privata + certificato pubblico) utilizzando un generatore di numeri casuali sicuro.

Successivamente, è stato riavviato il servizio Apache (se installato) per garantire che i nuovi certificati fossero attivi:

```
sudo /etc/init.d/apache2 restart
```

Per confermare l'avvenuta rigenerazione, ho controllato la data del certificato e il contenuto tramite:

```
sudo openssl x509 -in /etc/ssl/certs/ssl-cert-snakeoil.pem -text -noout
```

```
/etc/mysql/client-key.pem
/etc/mysql/client-cert.pem
/etc/mysql/ca-key.pem
find: /etc/lvm/backup: Permission denied
find: /etc/lvm/cache: Permission denied
find: /etc/lvm/archive: Permission denied
find: /etc/ssl/private: Permission denied
/etc/ssl/certs/ssl-cert-snakeoil.pem
find: /etc/unreal: Permission denied
msfadmin@metasploitable:~$ sudo find /etc -name "*.pem"
[sudo] password for msfadmin:
/etc/mysql/server-cert.pem
/etc/mysql/ca-cert.pem
/etc/mysql/client-req.pem
/etc/mysql/server-key.pem
/etc/mysql/server-req.pem
/etc/mysql/client-key.pem
/etc/mysql/client-cert.pem
/etc/mysql/ca-key.pem
/etc/ssl/certs/ssl-cert-snakeoil.pem
msfadmin@metasploitable:~$ sudo make-ssl-cert generate-default-snakeoil --force-
overwrite
msfadmin@metasploitable:~$ sudo /etc/init.d/apache2 restart
* Restarting web server apache2
msfadmin@metasploitable:~$ [ OK ]
```

## Esito della scansione finale

Dopo le azioni di mitigazione, è stata effettuata una nuova scansione Nessus.

Entrambe le vulnerabilità sono risultate risolte:

Le chiavi SSH deboli non sono più rilevate.

Il certificato SSL autofirmato è stato rigenerato con una nuova chiave sicura.

La macchina non presenta più chiavi o certificati generati con PRNG compromesso.

## Conclusioni

Le vulnerabilità critiche legate al generatore di numeri casuali debole in OpenSSL/OpenSSH su Debian sono state mitigate con successo.

Le azioni intraprese garantiscono un livello adeguato di sicurezza per le comunicazioni cifrate via SSH e SSL, eliminando il rischio di compromissione tramite chiavi previste o intercettazioni.



```
#shell stream tcp nowait root /bin/sh sh -i
```

In questo modo, la shell non verrà più attivata automaticamente al boot.

```
GNU nano 2.0.7      File: /etc/inetd.conf      Modified
#<off># netbios-ssn    stream  tcp    nowait  root    /usr/sbin/tcpd  /usr/sbin/in.telnetd
telnet               stream  tcp    nowait  root    /usr/sbin/tcpd  /usr/sbin/in.telnetd
#<off># ftp            stream  tcp    nowait  root    /usr/sbin/tcpd  /usr/sbin/in.ftpd
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
exec                 stream  tcp    nowait  root    /usr/sbin/tcpd  /usr/sbin/in.rexecd
#ingreslock stream tcp nowait root /bin/bash bash -i

[ Read 8 lines ]
^G Get Help  ^O WriteOut  ^R Read File ^V Prev Page ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^U Next Page ^_ UnCut Text ^T To Spell
```

### Esito finale:

Accesso remoto non autenticato bloccato

Nessun aggiornamento eseguito

Nessun servizio disabilitato

Mitigazione permanente rispettando tutti i vincoli