

Scansioniamo il sistema Metasploitable con **Nessus**.

<input type="checkbox"/>	Sev	CVSS ▼	VPR	Name	Family
<input type="checkbox"/>	CRITICAL	10.0 *	5.9	NFS Exported Share Information Disclosure	RPC
<input type="checkbox"/>	CRITICAL	10.0		Unix Operating System Unsupported Version Detection	General
<input type="checkbox"/>	CRITICAL	10.0 *		VNC Server 'password' Password	Gain a shell remotely
<input type="checkbox"/>	CRITICAL	9.8	9.0	Apache Tomcat AJP Connector Request Injection (Ghostcat)	Web Servers
<input type="checkbox"/>	CRITICAL	9.8		SSL Version 2 and 3 Protocol Detection	Service detection
<input type="checkbox"/>	CRITICAL	9.8		Bind Shell Backdoor Detection	Backdoors
<input type="checkbox"/>	HIGH	7.5	6.7	Samba Badlock Vulnerability	General
<input type="checkbox"/>	HIGH	7.5		NFS Shares World Readable	RPC

Risolviamo le vulnerabilità più **gravi**.

## Bind Shell Backdoor Detection

**CRITICAL** Bind Shell Backdoor Detection

**Description**  
A shell is listening on the remote port without any authentication being required. An attacker may use it by connecting to the remote port and sending commands directly.

**Solution**  
Verify if the remote host has been compromised, and reinstall the system if necessary.

**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 <a href="#">🔗</a>

Troviamo il processo relativo alla **porta**.

```
root@metasploitable:/home/msfadmin# netstat -tln | grep ":1524"
tcp        0      0 0.0.0.0:1524        0.0.0.0:*          LISTEN
root@metasploitable:/home/msfadmin# sudo lsof -i :1524
COMMAND PID USER   FD   TYPE DEVICE SIZE NODE NAME
xinetd  4455 root   12u  IPv4 12317      TCP *:ingreslock (LISTEN)
root@metasploitable:/home/msfadmin# _
```

Potremmo fermare il processo o disabilitare il servizio associato, ma comunque potrebbe ripartire al riavvio; è necessario quindi **eliminare** completamente la backdoor dal sistema.

```
GNU nano 2.0.7 File: /etc/inetd.conf
#<off># netbios-ssn      stream  tcp    nowait  root    /usr/sbin/tcpd  /usr/sbin/in.td$
telnet                  stream  tcp    nowait  telnetd  /usr/sbin/tcpd  /usr/sbin/in.tel$
#<off># ftp              stream  tcp    nowait  root    /usr/sbin/tcpd  /usr/sbin/in.ft$
tftp                   dgram  udp    wait    nobody   /usr/sbin/tcpd  /usr/sbin/in.tft$
shell                  stream  tcp    nowait  root    /usr/sbin/tcpd  /usr/sbin/in.rsh$
login                  stream  tcp    nowait  root    /usr/sbin/tcpd  /usr/sbin/in.rls$
exec                   stream  tcp    nowait  root    /usr/sbin/tcpd  /usr/sbin/in.re$
ingreslock stream tcp nowait root /bin/bash bash -i
```

Lo facciamo andando nel file di configurazione di **inetd** e cancellando l'ultima riga, responsabile dell'apertura della backdoor tramite la quale si potevano inviare comandi direttamente nella shell di metasploitable da remoto senza autenticazione.

## VNC Server 'password' Password

**CRITICAL** VNC Server 'password' Password

**Description**

The VNC server running on the remote host is secured with a weak password. Nessus was able to login using VNC authentication and a password of 'password'. A remote, unauthenticated attacker could exploit this to take control of the system.

**Solution**

Secure the VNC service with a strong password.

**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

La soluzione qui è molto semplice; basta modificare la password del servizio **vnc** con il comando da terminale **vncpasswd**, impostando una password sicura.

## NFS Exported Share Information Disclosure e NFS Shares World Readable

### CRITICAL NFS Exported Share Information Disclosure

#### Description

At least one of the NFS shares exported by the remote server could be mounted by the scanning host. An attacker may be able to leverage this to read (and possibly write) files on remote host.

#### Solution


Configure NFS on the remote host so that only authorized hosts can mount its remote shares.

#### Output

```
The following NFS shares could be mounted :
```

```
+ /
+ Contents of / :
- .
- ..
- bin
- boot
- cdrom
more...
```

To see debug logs, please visit individual host

Port ▲	Hosts
2049 / udp / rpc-nfs	192.168.50.101 

### HIGH NFS Shares World Readable

#### Description

The remote NFS server is exporting one or more shares without restricting access (based on hostname, IP, or IP range).

#### Solution

Place the appropriate restrictions on all NFS shares.

#### See Also


<http://www.tldp.org/HOWTO/NFS-HOWTO/security.html>

#### Output

```
The following shares have no access restrictions :
```

```
/ *
```

To see debug logs, please visit individual host

Port ▲	Hosts
2049 / tcp / rpc-nfs	192.168.50.101 

Analizziamo il file di configurazione di **nfs**.

```
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)
#

[ Read 10 lines ]
^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell
```

Eliminiamo la riga che permetteva l'accesso a **qualunque client da remoto** in scrittura e lettura con permessi di root; inoltre i cambiamenti effettuati sui file condivisi venivano scritti sincronamente sul disco fisico (sopra l'immagine del file di configurazione correttamente modificato). L'accesso andrà consentito soltanto a hosts specifici, sulle cartelle necessarie (non la root) e con i permessi minimi necessari.

## SSL Version 2 and 3 Protocol Detection

**Vulnerabilities** 79

**CRITICAL** SSL Version 2 and 3 Protocol Detection

**Description**

The remote service accepts connections encrypted using SSL 2.0 and/or SSL 3.0. These versions of SSL are affected by several cryptographic flaws, including:

- An insecure padding scheme with CBC ciphers.
- Insecure session renegotiation and resumption schemes.

An attacker can exploit these flaws to conduct man-in-the-middle attacks or to decrypt communications between the affected service and clients.

Although SSL/TLS has a secure means for choosing the highest supported version of the protocol (so that these versions will be used only if the client or server support nothing better), many web browsers implement this in an unsafe way that allows an attacker to downgrade a connection (such as in POODLE). Therefore, it is recommended that these protocols be disabled entirely.

NIST has determined that SSL 3.0 is no longer acceptable for secure communications. As of the date of enforcement found in PCI DSS v3.1, any version of SSL will not meet the PCI SSC's definition of 'strong cryptography'.

**Solution**

Consult the application's documentation to disable SSL 2.0 and 3.0.  
Use TLS 1.2 (with approved cipher suites) or higher instead.

Questo problema è presente sia sulla porta 25 dove è attivo il servizio smtp **postfix** e sulla porta 5432 dove è attivo il servizio **postgresql**.

```
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 -SSLv3 -TLSv1 -TLSv1.1_

</IfModule>

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

```
GNU nano 2.0.7      File: /etc/postgresql/8.3/main/postgresql.conf

max_connections = 100                # (change requires restart)
# Note: Increasing max_connections costs ~400 bytes of shared memory per
# connection slot, plus lock space (see max_locks_per_transaction). You might
# also need to raise shared_buffers to support more connections.
#superuser_reserved_connections = 3  # (change requires restart)
unix_socket_directory = '/var/run/postgresql' # (change requires restart)
unix_socket_group = ''               # (change requires restart)
unix_socket_permissions = 0777      # begin with 0 to use octal notation
                                     # (change requires restart)
#bonjour_name = ''                  # defaults to the computer name
                                     # (change requires restart)

# - Security and Authentication -

#authentication_timeout = 1min        # 1s-600s
#ssl = true                          # (change requires restart)
#ssl_ciphers = 'ALL:!ADH:!LOW:!EXP:!MD5:@STRENGTH' # allowed SSL ciphers
                                     # (change requires restart)
#password_encryption = on
#db_user_namespace = off

[ "tls" not found ]

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

Modificando i file di configurazione in modo tale da obbligare all'utilizzo del **tls** (aggiungendo la direttiva **SSLProtocol** su apache2 e commentando la riga che abilita l'**ssl** (ssl = true) costringendo pertanto ad utilizzare il **tls**, per postgresql) e rilanciando la scansione vediamo che il problema per postgresql è risolto, mentre per postfix, trattandosi di una versione molto vecchia che probabilmente non accetta direttive sulle versioni ssl, il consiglio è di **aggiornare** ad una versione più recente.

Effettuiamo nuovamente la scansione con Nessus.

<input type="checkbox"/> Sev	CVSS ▼	VPR	Name	Family
<input type="checkbox"/> CRITICAL	10.0		Unix Operating System Unsupported Version Detection	General
<input type="checkbox"/> CRITICAL	9.8	9.0	Apache Tomcat AJP Connector Request Injection (Ghostcat)	Web Servers
<input type="checkbox"/> CRITICAL	9.8		SSL Version 2 and 3 Protocol Detection	Service detection
<input type="checkbox"/> HIGH	7.5	5.9	Samba Badlock Vulnerability	General

Abbiamo risolto una buona parte dei problemi più gravi; si noti infine come per risolvere i rimanenti basterebbe semplicemente **aggiornare** i rispettivi software a versioni più recenti.