

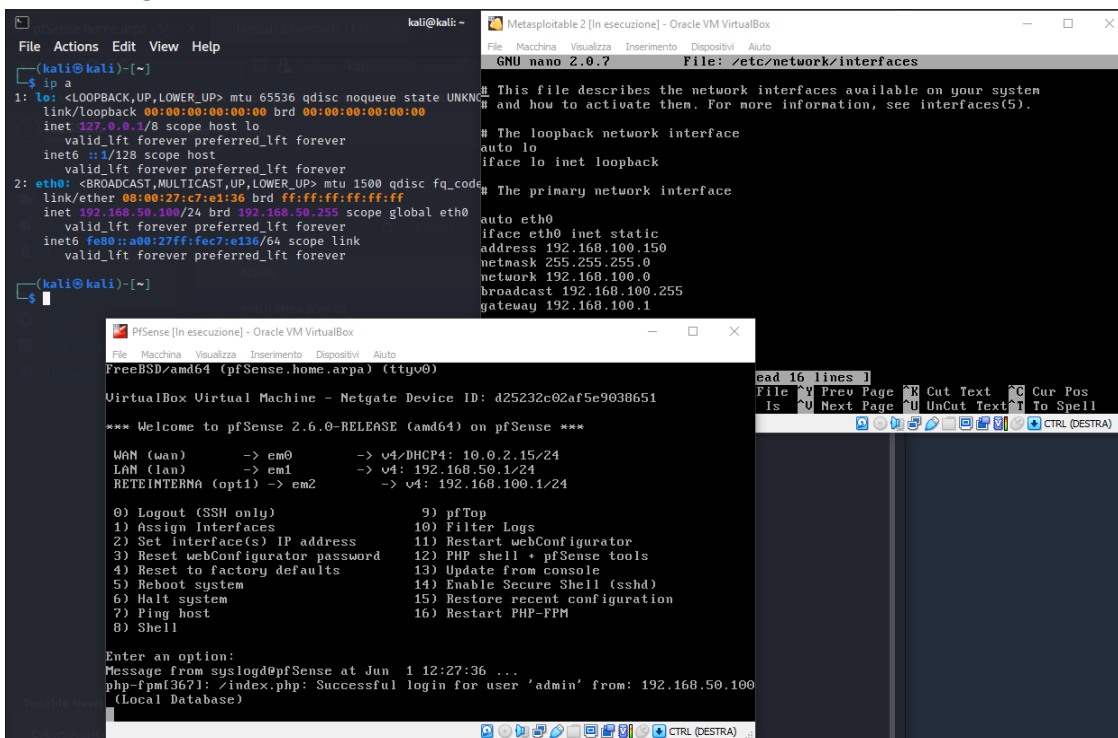
Metasploit Remediation

Obiettivo: Effettuare una scansione delle criticità sulla macchina Metasploitable e risolvere alcune delle criticità presenti.

Strumenti Utilizzati:

- **Kali Linux** (usato per effettuare le scansioni e utilizzare gli exploit di Metasploit);
- **Metasploitable** (macchina bersaglio);
- **pfSense** (usato come router e necessariamente come firewall);
- **Nessus** (utilizzato come vulnerability scanner).

1) Configurazione Indirizzi IP



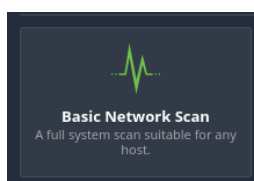
Per cominciare il progetto ho configurato gli indirizzi IP di **Kali** e **Metasploitable** con reti diverse facendoli connettere tra loro grazie a **PFsense** utilizzato come Router.

2) Avvio di Nessus e inizio della scansione

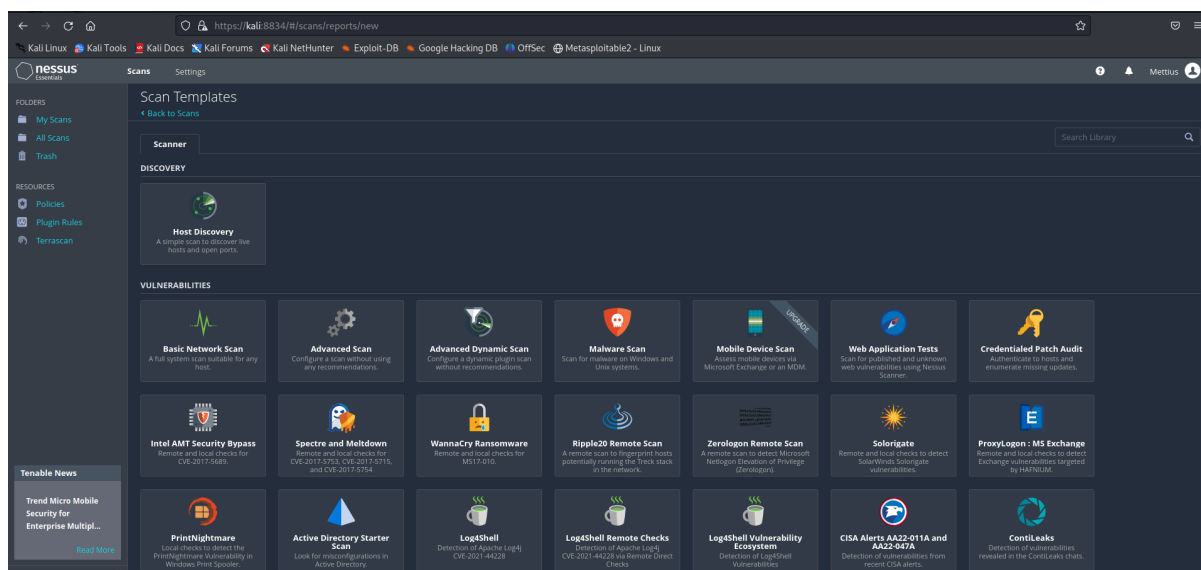
Avendo installato precedentemente **Nessus** sulla macchina Kali, ho avviato il servizio con:

sudo systemctl start nessusd.service

ed aprendo su browser il programma con <https://kali:8834> ho avviato la scansione base:



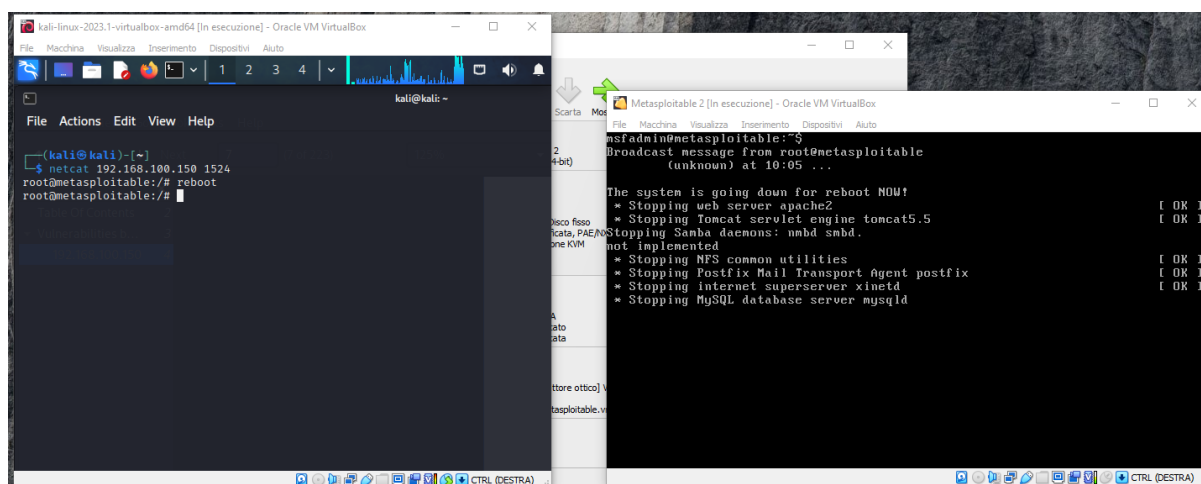
Schermata Principale di Nessus:



Dalla scansione vengono fuori diverse vulnerabilità, cosa ovvia trattandosi **Metasploitable** di una macchina appositamente vulnerabile. L'esercizio richiede di risolvere particolari vulnerabilità che tenterò di sanare.

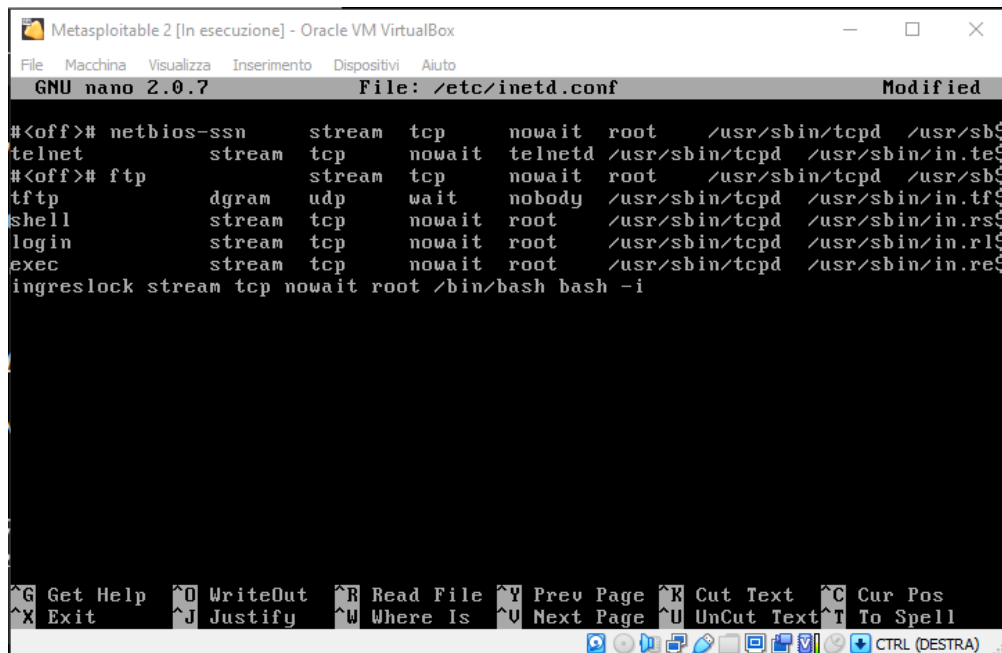
3) Bind Shell Backdoor Detection

Il primo problema che sono andato a risolvere è quello della presenza di una *backdoor* che sfrutta la porta **1524/tcp** e permette a chiunque sia in grado di connettersi alla **Metasploitable**, di entrare direttamente come *root*. Nel mio caso mi sono connesso alla macchina bersaglio tramite il tool di Kali **Netcat**, ovvero mi sono connesso in remoto digitando semplicemente l'IP di **Metasploitable**.



Una volta entrato come *root* ho utilizzato un comando semplice come *reboot*, riavviando la macchina, ma altre persone con una vulnerabilità del genere potrebbero fare molti più danni come usare un **keylogger** per rubare le password degli account della macchina controllata.

Soluzione: per risolvere il problema sono entrato con l'editor *nano* nel percorso */etc/inetd.conf* cancellando l'ultima riga (figura sotto). Questa è una stringa di codice che crea la backdoor collegando chiunque si connetta alla shell di default della macchina.



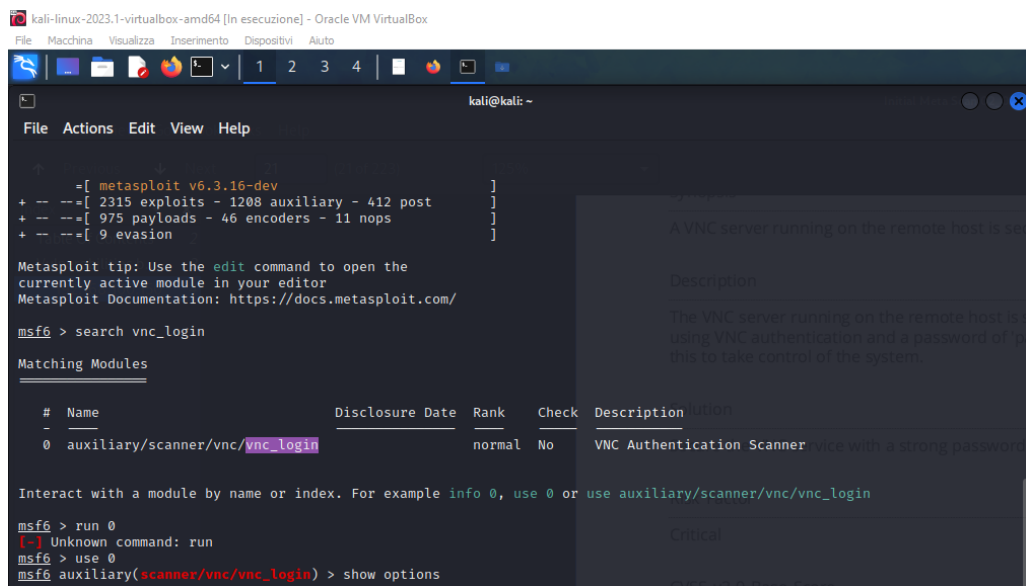
```
Metasploitable 2 [In esecuzione] - Oracle VM VirtualBox
File  Macchina  Visualizza  Inserimento  Dispositivi  Aiuto
GNU nano 2.0.7      File: /etc/inetd.conf      Modified

#<off># netbios-ssn      stream  tcp     nowait  root    /usr/sbin/tcpd  /usr/sbin/tcpd
telnet      stream  tcp     nowait  telnetd /usr/sbin/tcpd  /usr/sbin/in.telnetd
#<off># ftp      stream  tcp     nowait  root    /usr/sbin/tcpd  /usr/sbin/ftpd
tftp      dgram   udp     wait    nobody   /usr/sbin/tcpd  /usr/sbin/in.tftpd
shell      stream  tcp     nowait  root    /usr/sbin/tcpd  /usr/sbin/in.rsh
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

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

4) VNC Server 'password' Password

I **VNC** sono applicazioni che permettono il controllo in remoto di una macchina. In questo caso la criticità è la debolezza della password (è utilizzata "password" per l'appunto) che permette un facile accesso al server **VNC**. Utilizzando la *msfconsole* da **Kali** (comando *msf6* da riga di comando) e cercando il modulo che si riferiva alla login del server ho potuto simulare il framework di **Metasploit** con attacco al server **VNC** settando l'*host*, lo username "root" e usando una lista di password predefinite.



```
kali-linux-2023.1-virtualbox-amd64 [In esecuzione] - Oracle VM VirtualBox
File  Macchina  Visualizza  Inserimento  Dispositivi  Aiuto

kali@kali: ~
File  Actions  Edit  View  Help

-=[ metasploit v6.3.16-dev ]-
+ -- --=[ 2315 exploits - 1208 auxiliary - 412 post ]-
+ -- --=[ 975 payloads - 46 encoders - 11 nops ]-
+ -- --=[ 9 evasion ]-

Metasploit tip: Use the edit command to open the
currently active module in your editor
Metasploit Documentation: https://docs.metasploit.com/

msf6 > search vnc_login

Matching Modules
=====
#  Name                                     Disclosure Date  Rank  Check  Description
--  -
0  auxiliary/scanner/vnc/vnc_login          normal          No    VNC Authentication Scanner

Interact with a module by name or index. For example info 0, use 0 or use auxiliary/scanner/vnc/vnc_login

msf6 > run 0
[!] Unknown command: run
msf6 > use 0
msf6 auxiliary(scanner/vnc/vnc_login) > show options
```

```
Module options (auxiliary/scanner/vnc/vnc_login):

  Name          Current Setting      Required  Description
  ----          -
  BLANK_PASSWORDS false                    no        Try blank passwords for all users
  BRUTEFORCE_SPEED 5                      yes       How fast to bruteforce, from 0 to 5
  DB_ALL_CREDS     false                  no        Try each user/password couple stored in the current data
  base
  DB_ALL_PASS      false                  no        Add all passwords in the current database to the list
  DB_ALL_USERS     false                  no        Add all users in the current database to the list
  DB_SKIP_EXISTING none                   no        Skip existing credentials stored in the current database
  (Accepted: none, user, user@realm)
  PASSWORD         /usr/share/metasploit-framework/ no        The password to test
  PASS_FILE        data/wordlists/vnc_passwords.txt no        File containing passwords, one per line
  Proxies          no                    no        A proxy chain of format type:host:port[,type:host:port][,
  ...]
  RHOSTS           yes                   yes       The target host(s), see https://docs.metasploit.com/docs
  /using-metasploit/basics/using-metasploit.html
  RPORT            5900                 yes       The target port (TCP)
  STOP_ON_SUCCESS  false                yes       Stop guessing when a credential works for a host
  THREADS          1                    yes       The number of concurrent threads (max one per host)
  USERNAME         <BLANK>              no        A specific username to authenticate as
  USERPASS_FILE    no                   no        File containing users and passwords separated by space,
  one pair per line
  USER_AS_PASS     false                no        Try the username as the password for all users
  USER_FILE        no                   no        File containing usernames, one per line (NCR/CNC/AC)
  VERBOSE          true                 yes       Whether to print output for all attempts

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

msf6 auxiliary(scanner/vnc/vnc_login) > set rhosts 192.168.100.150
rhosts => 192.168.100.150
msf6 auxiliary(scanner/vnc/vnc_login) > set USERNAME root
USERNAME => root
msf6 auxiliary(scanner/vnc/vnc_login) > run

[*] 192.168.100.150:5900 - 192.168.100.150:5900 - Starting VNC login sweep
[*] 192.168.100.150:5900 - No active DB -- Credential data will not be saved!
[*] 192.168.100.150:5900 - 192.168.100.150:5900 - Login Successful: :password
[*] 192.168.100.150:5900 - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/vnc/vnc_login) > run

[*] 192.168.100.150:5900 - 192.168.100.150:5900 - Starting VNC login sweep
[*] 192.168.100.150:5900 - No active DB -- Credential data will not be saved!
[*] 192.168.100.150:5900 - 192.168.100.150:5900 - LOGIN FAILED: :password (Incorrect: Authentication failed)
[*] 192.168.100.150:5900 - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/vnc/vnc_login) >
```

Soluzione: come si vede dalla figura, ho fatto prima un login al server con le impostazioni base (ovvero “root”, “password”) riuscendo ad entrare. Per risolvere il problema sono andato con l’editor di testo *nano* all’interno di */.vnc/passwd* modificando la password a mano e ritentando il login ma stavolta fallendo. La password in questa specifica Metasploitable è facilmente decriptabile essendo cifrata con il *3DES*, ormai in disuso da anni a favore del *Advanced Encryption Standard*. La soluzione sarebbe aggiornare il sistema o cambiarlo completamente a favore di nuove tecnologie altrimenti un semplice programma trovabile anche su github può facilmente decifrare la password **VNC**.

5) Unix Operating System Unsupported Version Detection

Collegandomi alla soluzione precedente un’altra vulnerabilità è quella del Sistema Operativo non supportato. La soluzione sarebbe di aggiornare la macchina **Metasploitable** ma come per le versioni troppo vecchie di **Windows** (ad esempio 98, XP, 7 etc.) che hanno raggiunto la “*end of life*”, non c’è più supporto e perciò non è più possibile scaricare aggiornamenti.

6) NFS Exported Share Information Disclosure

Il **Network File System** consente a un client di collegarsi tramite rete a un server dove possono essere situate cartelle condivise, come ad esempio file che possono servire ai dipendenti di un’azienda. In questo caso però la criticità sta nel fatto che chiunque può instaurare delle cartelle all’interno di **Metasploitable** con il pericolo di entrare all’interno del sistema e creare quella che si chiama “*Privilege Escalation*”, cioè ottenere i privilegi che sono in genere preclusi ad altri utenti (come quelli di *root*).

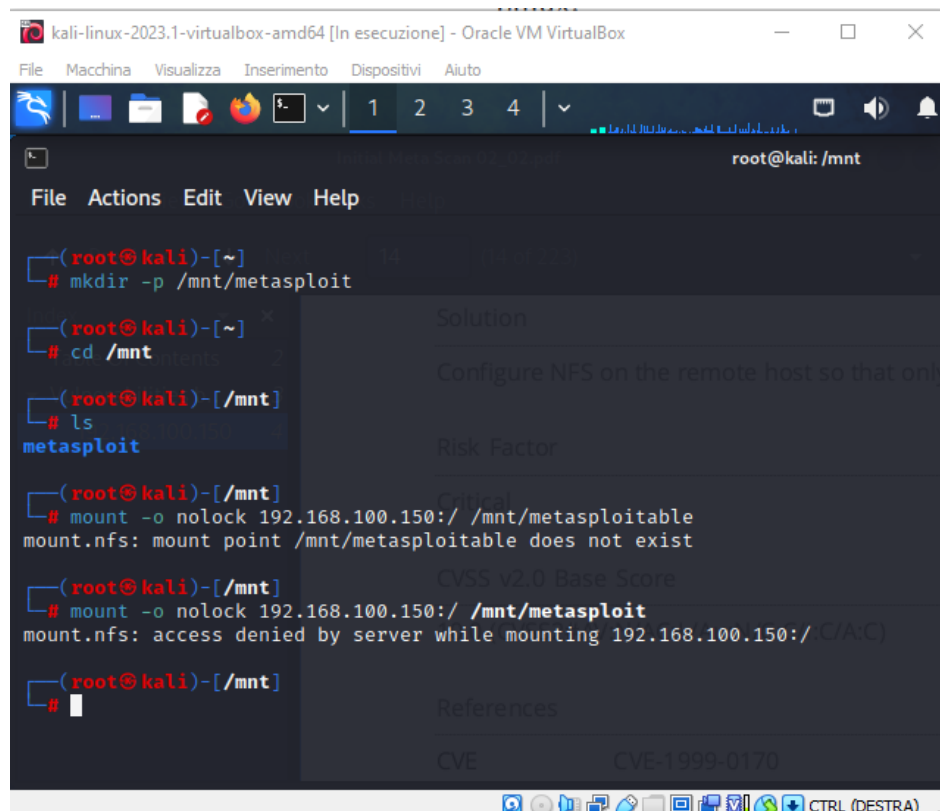
Ho provato infatti a montare una cartella creata su **Kali** direttamente nella directory *root* di **Metasploitable** e successivamente controllando con *ls* se ci fossi riuscito.

```
root@kali: /mnt
File Actions Edit View Help
root@kali:~# cd /mnt
root@kali:~/mnt# ls -al
.  ..  metasploitable
root@kali:~/mnt# mount -o nolock 192.168.100.150:/ /mnt/metasploitable
root@kali:~/mnt# cd metasploitable
root@kali:~/mnt/metasploitable# ls -la
total 124
drwxr-xr-x 21 root root 4096 Jun 1 09:39 .
drwxr-xr-x 3 root root 4096 Jun 1 11:36 ..
-rw-r--r-- 1 root root 0 Jun 1 06:27 '}'
drwxr-xr-x 2 root root 4096 May 13 2012 bin
drwxr-xr-x 3 root root 4096 Apr 28 2010 boot
lrwxrwxrwx 1 root root 11 Apr 28 2010 cdrom -> media/cdrom
drwxr-xr-x 2 root root 4096 Apr 28 2010 dev
drwxr-xr-x 94 root root 4096 Jun 1 10:42 etc
drwxr-xr-x 6 root root 4096 Apr 16 2010 home
drwxr-xr-x 2 root root 4096 Mar 16 2010 initrd
lrwxrwxrwx 1 root root 32 Apr 28 2010 initrd.img -> boot/initrd.img-2.6.24-16-server
drwxr-xr-x 13 root root 4096 May 13 2012 lib
drwxr-xr-x 2 root root 16384 Mar 16 2010 lost+found
drwxr-xr-x 4 root root 4096 Mar 16 2010 media
drwxr-xr-x 3 root root 4096 Apr 28 2010 mnt
-rw-r--r-- 1 root root 26009 Jun 1 10:43 nohup.out
drwxr-xr-x 2 root root 4096 Mar 16 2010 opt
drwxr-xr-x 2 root root 4096 Apr 28 2010 proc
drwxr-xr-x 13 root root 4096 Jun 1 10:43 root
drwxr-xr-x 2 root root 4096 May 13 2012 sbin
drwxr-xr-x 2 root root 4096 Apr 28 2010 srv
drwxr-xr-x 2 root root 4096 Apr 28 2010 sys
-rw-r--r-- 1 root root 0 Jun 1 09:39 'T+','$'\003'
drwxrwxrwt 4 root root 4096 Jun 1 10:43 tmp
drwxr-xr-x 12 root root 4096 Apr 28 2010 usr
drwxr-xr-x 14 root root 4096 Mar 17 2010 var
lrwxrwxrwx 1 root root 29 Apr 28 2010 vmlinuz -> boot/vmlinuz-2.6.24-16-server
root@kali:~/mnt/metasploitable# ls root
Desktop  reset_logs.sh  vnc.log
root@kali:~/mnt/metasploitable# cat root/.ssh/authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAABIWAAQEApmGJFZNl0bMNALQx7M6SGoi4KnmJ6PVxpbpG70ShHqldJkcteZ2dPFSBw76IUIPR00h+WBV0x1c6iPL/0zUYFHyFKAz1e6/
SteoweG1j2q0ffdomVhvXXv5Jga5FWwYB8R0QxsOWMTQTYSeBa6X6e777GVKCDLYgZSo8Wmr5JXln/Tw7XotowHr8FEGVw2Z1krU3Zo9Bzp0e0ac2U+qUG1Z1u/WwgtLZ
s5/D9IyHrWocvQPE+kcP+J2mt4y1uA73KqoXfdw5oGUKxdF09f1nu20wkJ0c+Vw8Vw7bwkrf+1Rg10Mg1J5Cs4WocvXsXovcNnbALTp3w= msfadmin@metasploitable
```

Chiunque possa accedere alla macchina in questo modo può copiare la “*authorized_keys*” di *root* per poter loggare con tutti i suoi privilegi.

Soluzione: Per impedire ad esterni di poter creare cartelle all’interno della **Metasploitable** sono andato con l’editor di testo nano all’interno di **/etc/exports** eliminando l’ultima riga contenente settaggi pericolosi come **rw** che permette la lettura e scrittura di file e cartelle e **no_root_squash** che permette a qualsiasi utente di accedere a qualunque file come se fosse root quando non potrebbe avere tali privilegi.

```
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)
#
#
# * (rw, sync, no_root_squash, no_subtree_check)
```



```
(root@kali)-[~]
# mkdir -p /mnt/metasploit

(root@kali)-[~]
# cd /mnt

(root@kali)-[/mnt]
# ls
metasploit

(root@kali)-[/mnt]
# mount -o nolock 192.168.100.150:/ /mnt/metasploitable
mount.nfs: mount point /mnt/metasploitable does not exist

(root@kali)-[/mnt]
# mount -o nolock 192.168.100.150:/ /mnt/metasploit
mount.nfs: access denied by server while mounting 192.168.100.150:/C/A/C

(root@kali)-[/mnt]
#
```

7) rexecd Service Detection

L'ultima vulnerabilità da analizzare (tra quelle proposte dalla traccia) è quella del processo **rexecd**. Nelle varie scansioni che ho effettuato questa criticità non è mai stata trovata, neanche con una scansione base completa o una avanzata di tutte le porte e che comprendesse solo i *plugin* inerenti a questa criticità stessa (ho cercato sulla wiki di tenable quale *plugin* rilevasse il problema). Anche su internet non ho trovato nulla che riguardasse questa criticità sulla **Metasploitable** utilizzata e la soluzione che posso azzardare è quella di un falso positivo nella scansione della slide oppure potrebbe trattarsi di un bug presente su versioni diverse della Metasploitable utilizzata nella mia prova.

8) Altre Vulnerabilità

L'esercizio era incentrato sul risolvere delle vulnerabilità predefinite nella quale non era obbligatorio utilizzare un **Firewall**. Ho fatto una prima prova senza di esso, utilizzando quindi **PFsense** come un semplice router per collegare le due reti di **Kali** e **Metasploitable** e ho riportato le soluzioni "manuali" a tali criticità in questo report.

Successivamente ho voluto creare una nuova macchina **Metasploitable** e provare a creare delle policy sul firewall di **PFsense** in modo da ridurre ulteriormente, oltre alle vulnerabilità scelte dall'esercizio, le restanti di livello *Critico* e *Alto*.

Ho settato le policy soltanto con le porte riguardanti le vulnerabilità di livello *Critico* e il risultato della scansione, come si può vedere dal report, indica che molti problemi riscontrati dallo scan iniziale di **Nessus**, anche quelli di livello *Alto* e *Medio*, sono stati risolti lasciando soltanto un **Rischio Residuo**, che sarà l'azienda cliente a decidere se mantenerlo o fare un ulteriore secondo intervento.

Firewall / Rules / RETEINTERNA

The changes have been applied successfully. The firewall rules are now reloading in the background.
Monitor the filter reload progress.

Floating WAN LAN RETEINTERNA

Rules (Drag to Change Order)											
<input type="checkbox"/>	States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
<input type="checkbox"/>	✗ 0/0 B	IPv4 TCP	192.168.50.100	*	192.168.100.150	22 (SSH)	*	none			
<input type="checkbox"/>	✗ 0/0 B	IPv4 TCP	192.168.50.100	*	192.168.100.150	5900 (VNC)	*	none			
<input type="checkbox"/>	✗ 0/0 B	IPv4 TCP	192.168.50.100	*	192.168.100.150	5432	*	none			
<input type="checkbox"/>	✗ 0/0 B	IPv4 UDP	192.168.50.100	*	192.168.100.150	2049	*	none			
<input type="checkbox"/>	✗ 0/0 B	IPv4 TCP	192.168.50.100	*	192.168.100.150	5432	*	none			
<input type="checkbox"/>	✗ 0/0 B	IPv4 TCP	192.168.50.100	*	192.168.100.150	25 (SMTP)	*	none			
<input type="checkbox"/>	✗ 0/0 B	IPv4 TCP	192.168.50.100	*	192.168.100.150	1524	*	none			
<input type="checkbox"/>	✗ 0/0 B	IPv4 TCP	192.168.50.100	*	192.168.100.150	8009	*	none			
<input type="checkbox"/>	✗ 0/0 B	IPv4 TCP	192.168.50.100	*	192.168.100.150	80 (HTTP)	*	none			

Add Add Delete Save Separator

9) Conclusioni

L'utilizzo di un **Firewall** è una scelta ottimale per bloccare l'accesso a quelle porte vulnerabili della macchina sulla quale abbiamo riscontrato problemi grazie al software **Nessus**; è importante però anche intervenire manualmente su pericoli come backdoor o bug perchè potrebbero essere sfruttati da chi riesce a bypassare il **Firewall**.

Un ulteriore consiglio è quello di tenere macchine e software sempre aggiornati all'ultima versione, il costo di un nuovo software è quasi sicuramente inferiore al prezzo da pagare per una falla nella sicurezza e alla conseguente perdita o appropriazione di dati sensibili o privati da parte di criminali.