

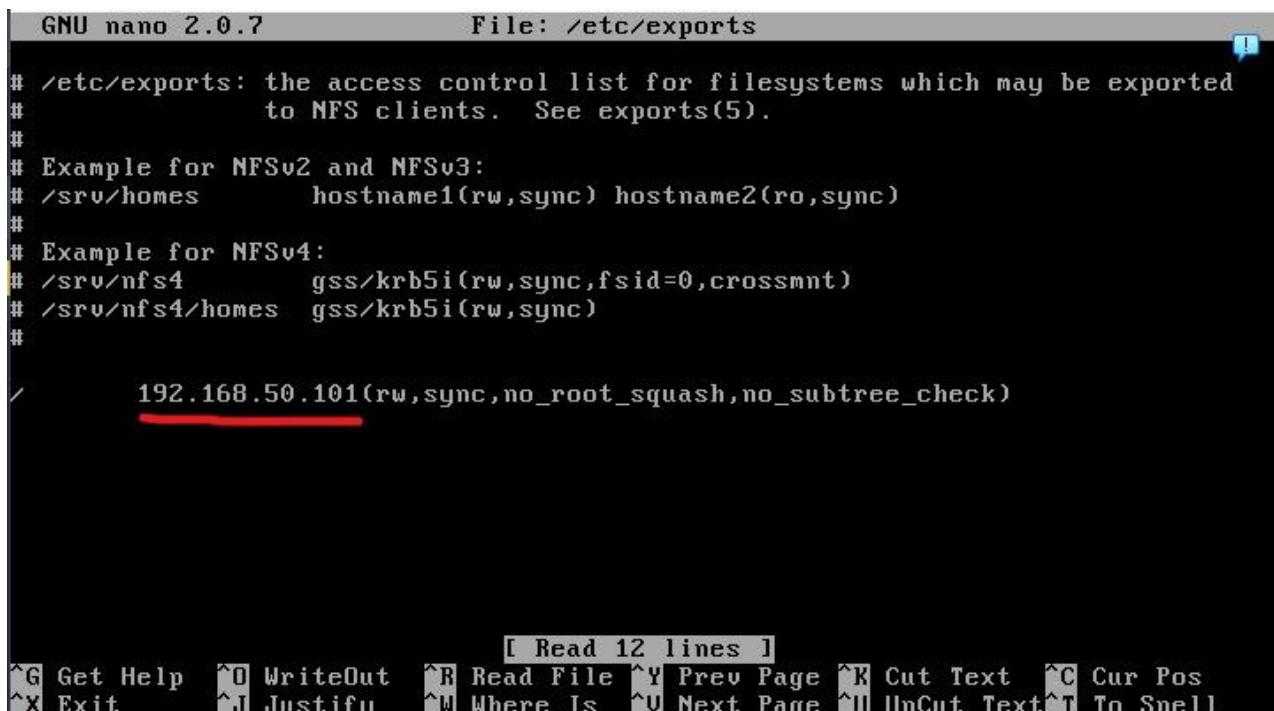
REMEDIATION METASPOITABLE

- *Soluzione: 11356 - NFS Exported Share Information Disclosure*

La causa di tale vulnerabilità era che il file system NFS, essendo in condivisione aperta, avrebbe potuto scatenare l'entrata di un utente malintenzionato, il quale può andare a leggere, o addirittura scrivere, file su host. Pertanto andiamo a togliere i permessi ed attribuirli solo ed esclusivamente all'host remoto, in questo caso Meta.

Entriamo nel file exports e andiamo a cambiare i privilegi del NFS. Per entrare nel file basterà usare il comando:

`sudo nano /etc/exports`



```
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)
#
/      192.168.50.101(rw,sync,no_root_squash,no_subtree_check)

[ Read 12 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  ^U Next Page ^U UnCut Text ^T To Spell
```

La figura in alto mostra già la risoluzione del problema. L'unica differenza trovata con il file prima di modificarlo era un * al posto dell'indirizzo IP segnato. L'asterisco consentiva a tutti i client di poter accedere al file system, ma inserendo solo l'indirizzo IP di Meta in questo caso il permesso è consentito solo ed esclusivamente all'Host Remoto.

Effettuiamo una scansione per vedere se abbiamo ovviato al problema, inserendo solo la porta che ci interessa: 2049

Nessus Essentials / Folder: X

https://kali:8834/#/scans/reports/11/hosts/2/vulnerabilities

Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec

There's an error with your feed. Click here to view your license information.

nessus
Essentials

Scans Settings

123456789101112131415161718192021222324252627282930313233343536373839404142434445464748495051525354555657585960616263646566676869707172737475767778798081828384858687888990919293949596979899100

NFS test / 192.168.50.101

Configure Audit Trail Launch Report Export

Vulnerabilities 52

Filter Search Vulnerabilities 52 Vulnerabilities

Sev	Score	Name	Family	Count		
CRITICAL	10.0	Unix Operating System Unsupported Version Detection	General	1		
CRITICAL	10.0 *	VNC Server 'password' Password	Gain a shell remotely	1		
CRITICAL	9.8	Apache Tomcat AJP Connector Request Injection (Ghostcat)	Web Servers	1		
CRITICAL	...	SSL (Multiple Issues)	Gain a shell remotely	3		
MIXED	...	SSL (Multiple Issues)	Service detection	3		
HIGH	7.5	Samba Badlock Vulnerability	General	1		
MIXED	...	SSL (Multiple Issues)	General	26		
MIXED	...	ISC Bind (Multiple Issues)	DNS	5		
MEDIUM	6.5	TLS Version 1.0 Protocol Detection	Service detection	2		
MEDIUM	5.9	SSL DROWN Attack Vulnerability (Decrypting RSA with Obsolete and Weakened eNcryption)	Misc.	1		
MIXED	...	SSH (Multiple Issues)	Misc.	6		
MIXED	...	HTTP (Multiple Issues)	Web Servers	3		

Host Details

IP: 192.168.50.101
MAC: 08:00:27:7B:21:1D
OS: Linux Kernel 2.6 on Ubuntu 8.04 (hardy)
Start: Today at 5:54 AM
End: Today at 6:16 AM
Elapsed: 22 minutes
KB: [Download](#)

Vulnerabilities

Critical

High

Medium

Low

Info

- *Soluzione: 61708 - VNC Server 'password' Password*

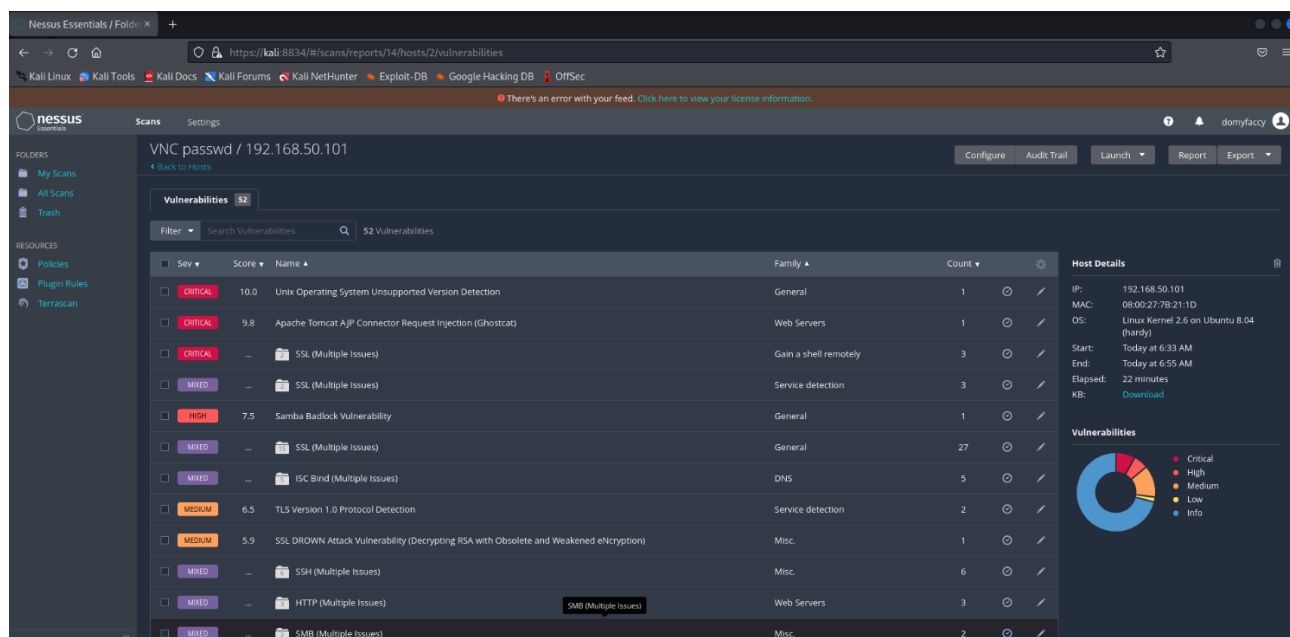
In questo caso invece la soluzione era di andare a cambiare la password del servizio VNC con una più sicura. Pertanto basta entrare in root su meta con il comando <<sudo su>> e successivamente eseguire il comando:

vncpasswd

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

Questo ci permetteva di inserire una nuova password con un massimo di 8 caratteri. Subito dopo il comando (grazie alle azioni da root) ci permetteva direttamente di inserire la nuova password che però **NON VISIBILE!**

Facciamo un test di scansione veloce ma solo sulla porta che ci interessa inerente a tale vulnerabilità, ovvero la porta 5900 e come si può notare dalla figura sottostante la vulnerabilità non appare.



- *Soluzione: 51988 - Bind Shell Backdoor Detection*

Quest'ultima vulnerabilità è una porta aperta sulla quale un utente malintenzionato senza alcuna autenticazione potrebbe mettersi in ascolto, collegandosi e inviando comandi diretti. Per evitarlo creiamo un firewall che blocchi pacchetti in entrata su questa porta. Per farlo usiamo il comando:

sudo iptables -I INPUT -p tcp --dport 1524 -j DROP

```
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.
msfadmin@metasploitable:~$ sudo loadkeys it
[sudo] password for msfadmin:
Loading /usr/share/keymaps/it.map.bz2
msfadmin@metasploitable:~$ sudo iptables -I INPUT -p tcp --dport 1524 -j DR
OP
msfadmin@metasploitable:~$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target      prot opt source                destination            tcp dpt:ingreslock
DROP        tcp  --  anywhere              anywhere
Chain FORWARD (policy ACCEPT)
target      prot opt source                destination
Chain OUTPUT (policy ACCEPT)
target      prot opt source                destination
msfadmin@metasploitable:~$
```

Dopo aver mandato il comando possiamo analizzare il firewall inserito grazie a "iptables", il quale è un programma di utilità per lo spazio utente che consente a un amministratore di sistema di configurare le regole del filtro dei pacchetti IP del firewall; il comando da usare è <<sudo iptables -L>>, di fatti ci viene mostrato il firewall appena inserito (sottolineato in arancio) con le istruzioni di seguito.

Andiamo a fare un'ultima analisi di scansione con Nessus in questo caso dell'intero sistema in modo da vedere effettivamente se tutte le vulnerabilità prese in questione sono state eliminate.

The screenshot shows the Nessus Essentials web interface. The main heading is "Scansione Finale / 192.168.50.101". Below it, a table lists vulnerabilities. The table has columns for Severity (Sev), Score, Name, Family, and Count. The vulnerabilities are categorized by severity: Critical (red), High (orange), Medium (yellow), Low (green), and Info (blue). The 'Host Details' panel on the right shows the following information:

- IP: 192.168.50.101
- MAC: 08:00:27:7B:21:1D
- OS: Linux Kernel 2.6 on Ubuntu 8.04 (hardy)
- Start: Today at 7:21 AM
- End: Today at 7:51 AM
- Elapsed: 30 minutes
- KB: [Download](#)

The 'Vulnerabilities' pie chart shows the distribution of severity levels: Critical (red), High (orange), Medium (yellow), Low (green), and Info (blue).

Sev	Score	Name	Family	Count
CRITICAL	10.0	Unix Operating System Unsupported Version Detection	General	1
CRITICAL	...	SSL (Multiple Issues)	Gain a shell remotely	3
MIXED	...	SSL (Multiple Issues)	Service detection	3
MIXED	...	Apache Tomcat (Multiple Issues)	Web Servers	3
MIXED	...	Web Server (Multiple Issues)	Web Servers	3
HIGH	7.5	Samba Badlock Vulnerability	General	1
MIXED	...	SSL (Multiple Issues)	General	27
MIXED	...	ISC Bind (Multiple Issues)	DNS	5
MEDIUM	6.5	TLS Version 1.0 Protocol Detection	Service detection	2
MEDIUM	5.9	SSL DROWN Attack Vulnerability (Decrypting RSA with Obsolete and W	Misc.	1
MIXED	...	SSH (Multiple Issues)	Misc.	6
INFO	...	HTTP (Multiple Issues)	Web Servers	5

Ecco fatto! Per vedere le nuove vulnerabilità, apri il file [ScansioneFinale.pdf](#)