

## Risoluzione di alcune criticità trovate con Nessus

**Apache Tomcat AJP Connector Request Injection (Ghostcat):** È stata rilevata una vulnerabilità di lettura/inclusione di file nel connettore AJP. Un utente malintenzionato remoto e non autenticato potrebbe sfruttare questa vulnerabilità per leggere i file delle applicazioni Web da un server vulnerabile. Nei casi in cui il server vulnerabile consenta il caricamento di file, un utente malintenzionato potrebbe caricare codice JavaServer Pages (JSP) dannoso all'interno di una varietà di tipi di file e ottenere l'esecuzione di codice in modalità remota (RCE).

Andiamo a commentare questo pezzo di codice nel file server.xml così disabilitiamo il servizio AJP connector, poi effettuiamo un restart di Tomcat. Un'altra soluzione sarebbe quella di aggiungere delle credenziali di autenticazione nel campo "requiredSecret attribute"

```
GNU nano 2.0.7      File: server.xml

<!-- Define a SSL HTTP/1.1 Connector on port 8443 -->
<!--
<Connector port="8443" maxHttpHeaderSize="8192"
    maxThreads="150" minSpareThreads="25" maxSpareThreads="75"
    enableLookups="false" disableUploadTimeout="true"
    acceptCount="100" scheme="https" secure="true"
    clientAuth="false" sslProtocol="TLS" />
-->

<!-- Define an AJP 1.3 Connector on port 8009 -->
<!--Connector port="8009"
    enableLookups="false" redirectPort="8443" protocol="AJP/1.3" />--$

<!-- Define a Proxied HTTP/1.1 Connector on port 8082 -->
<!-- See proxy documentation for more information about using this. -->
<!--
<Connector port="8082"
    maxThreads="150" minSpareThreads="25" maxSpareThreads="75"
    enableLookups="false" acceptCount="100" connectionTimeout="20000"

^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  ^U UnCut Text ^T To Spell
```

**Bind Shell Backdoor Detection:** Una shell è in ascolto sulla porta remota senza che sia richiesta alcuna autenticazione. Un utente malintenzionato può utilizzarlo collegandosi alla porta remota e inviando direttamente i comandi. Collegandoci in ascolto tramite il comando "netcat" possiamo verificare che riusciamo ad entrare nella macchina senza autenticarci sulla porta 1524.

```
(niko@kali)~  
$ netcat 192.168.50.101 1524  
root@metasploitable:/# echo ciao  
ciao  
root@metasploitable:/# ls  
bin  
boot  
cdrom  
dev  
etc  
home  
initrd  
initrd.img  
lib  
lost+found  
media  
mnt  
nohup.out  
opt  
proc  
root  
sbin  
srv  
sys  
tmp  
usr  
var  
vmlinuz  
root@metasploitable:/# whoami  
root  
root@metasploitable:/#
```

Da qui killiamo il processo relativo alla porta 1524

```
(niko@kali)-[~]  
$ netcat 192.168.50.101 1524  
root@metasploitable:/# fuser -k -n tcp 1524  
1524/tcp:          4312  6129
```

Facciamo una scansione per verificare che non compaia la porta ed effettivamente non compare più.

```
(niko@kali)-[~]  
$ nmap 192.168.50.101  
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-06-28 10:22 CEST  
Nmap scan report for 192.168.50.101  
Host is up (0.00094s latency).  
Not shown: 983 closed tcp ports (conn-refused)  
PORT      STATE SERVICE  
22/tcp    open  ssh  
25/tcp    open  smtp  
53/tcp    open  domain  
80/tcp    open  http  
111/tcp   open  rpcbind  
139/tcp   open  netbios-ssn  
445/tcp   open  microsoft-ds  
1099/tcp  open  rmiregistry  
2049/tcp  open  nfs  
2121/tcp  open  ccproxy-ftp  
3306/tcp  open  mysql  
5432/tcp  open  postgresql  
5900/tcp  open  vnc  
6000/tcp  open  X11  
6667/tcp  open  irc  
8009/tcp  open  ajp13  
8180/tcp  open  unknown  
  
Nmap done: 1 IP address (1 host up) scanned in 0.21 seconds
```

**VNC Server 'password' Password:** Il server VNC in esecuzione sull'host remoto è protetto con una password debole. Nessus è riuscito ad accedere utilizzando l'autenticazione VNC e una password "password". Un utente malintenzionato remoto e non autenticato potrebbe sfruttare questa situazione per assumere il controllo del sistema.

Con il comando “vncpasswd” ci verrà chiesto di inserire una password ma questo non risolverà il problema perchè connettendoci a vnc tramite la porta 5900 ci darà automaticamente permessi di root, quindi per cambiare effettivamente la password ho utilizzato il comando “sudo” e modificato la password contenuta all'interno del file /.vnc/passwd

```
msfadmin@metasploitable:/etc$  
msfadmin@metasploitable:/etc$  
msfadmin@metasploitable:/etc$ vncpasswd  
Using password file /home/msfadmin/.vnc/passwd  
VNC directory /home/msfadmin/.vnc does not exist, creating.  
Password:  
Warning: password truncated to the length of 8.  
Verify:  
Would you like to enter a view-only password (y/n)? n  
msfadmin@metasploitable:/etc$  
msfadmin@metasploitable:/etc$
```

**NFS Exported Share Information Disclosure:** Almeno una delle share NFS esportate dal server remoto potrebbe essere montata dall'host di scansione. Un utente malintenzionato potrebbe essere in grado di sfruttare questo per leggere (ed eventualmente scrivere) file sull'host remoto.

IMPORTANTE: tenere presente che i privilegi di root non erano necessari per montare le condivisioni remote poiché la porta di origine per montare le condivisioni era superiore a 1024.

Facciamo un check dei servizi network che sono avviati con il comando "rpcinfo"

```
msfadmin@metasploitable:~$ rpcinfo -p 192.168.50.101
  program vers  proto   port
  100000    2   tcp    111  portmapper
  100000    2   udp    111  portmapper
  100024    1   udp   45626  status
  100024    1   tcp   54657  status
  100003    2   udp    2049  nfs
  100003    3   udp    2049  nfs
  100003    4   udp    2049  nfs
  100021    1   udp   40492  nlockmgr
  100021    3   udp   40492  nlockmgr
  100021    4   udp   40492  nlockmgr
  100003    2   tcp    2049  nfs
  100003    3   tcp    2049  nfs
  100003    4   tcp    2049  nfs
  100021    1   tcp   36996  nlockmgr
  100021    3   tcp   36996  nlockmgr
  100021    4   tcp   36996  nlockmgr
  100005    1   udp   44613  mountd
  100005    1   tcp   37123  mountd
  100005    2   udp   44613  mountd
  100005    2   tcp   37123  mountd
  100005    3   udp   44613  mountd
  100005    3   tcp   37123  mountd
msfadmin@metasploitable:~$
```

Utilizzando il comando "showmount" vediamo che tutto il filesystem è montabile/scrivibile (/\*)

```
msfadmin@metasploitable:~$ showmount -e 192.168.50.101
Export list for 192.168.50.101:
/ *
msfadmin@metasploitable:~$
```

Se facessi "service rpcbind start" potrei montare il network filesystem senza credenziali.

Per evitare ciò dobbiamo modificare i permessi di chi può accedere alle share condivise.

Ci sono tre file che posso modificare:

- /etc/hosts.allow: contiene gli host che hanno accesso alle cartelle condivise
- /etc/hosts.deny: contiene gli host che non hanno accesso alle cartelle condivise
- /etc/exports: contiene la lista delle cartelle condivisibili

Vado a bloccare l'ip della macchina kali sul servizio nfs

```
GNU nano 2.0.7      File: /etc/hosts.deny      Modified
# for further information.
#
# The PARANOID wildcard matches any host whose name does not match its
# address.
#
# You may wish to enable this to ensure any programs that don't
# validate looked up hostnames still leave understandable logs. In past
# versions of Debian this has been the default.
# ALL: PARANOID

nfs : 192.168.50.100

^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
```

Aggiungo anche una regola sul firewall per bloccare i pacchetti tcp/udp sulla porta 2049 della metasploitable.

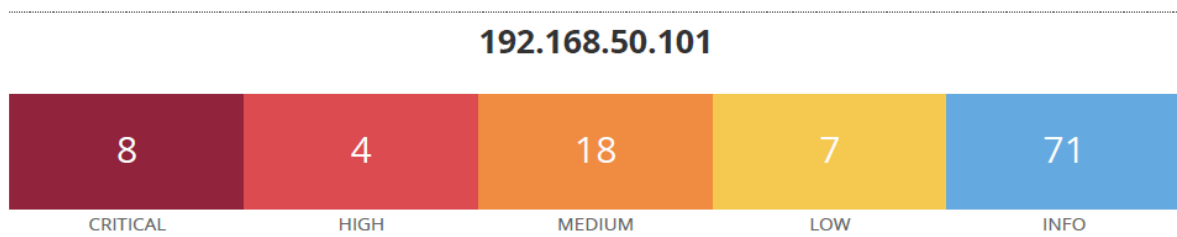
Rules (Drag to Change Order)											
<input type="checkbox"/>	States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
<input checked="" type="checkbox"/>	✓ 1/213 KiB	*	*	*	LAN Address	80	*	*		Anti-Lockout Rule	
<input type="checkbox"/>	✗ 0/0 B	IPv4 TCP/UDP	*	*	192.168.50.101	2049	*	none			
<input type="checkbox"/>	✓ 76/181 KiB	IPv4 *	LAN subnets	*	*	*	*	none		Default allow LAN to any rule	
<input type="checkbox"/>	✓ 0/0 B	IPv6 *	LAN subnets	*	*	*	*	none		Default allow LAN IPv6 to any rule	

**Unix Operating System Unsupported Version Detection:** La soluzione è quella di aggiornare ad una versione OS di UNIX che sia supportata attualmente.

**Samba Badlock Vulnerability:** Un utente malintenzionato man-in-the-middle in grado di intercettare il traffico tra un client e un server che ospita un database SAM può sfruttare questa falla per forzare un downgrade del livello di autenticazione, che consente l'esecuzione di chiamate di rete Samba arbitrarie nel contesto dell'utente intercettato, ad esempio visualizzare o modificare dati sensibili di sicurezza nel database di Active Directory (AD) o disabilitare servizi critici.

La soluzione è quella di fare l'upgrade a Samba version 4.2.11 / 4.3.8 / 4.4.2 o successive.

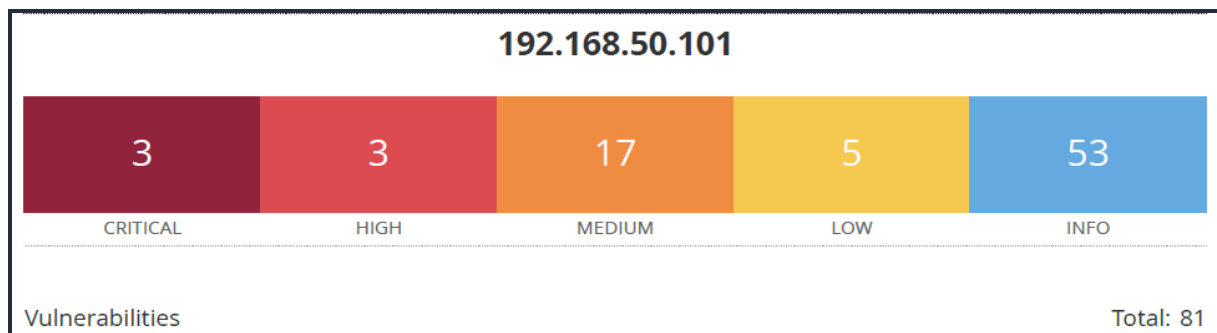
## REPORT INIZIALE



Vulnerabilities

Total: 108

## REPORT FINALE



Vulnerabilities

Total: 81

Alla fine di queste soluzioni le criticità sono diminuite come si può vedere in modo approfondito dal file di report lasciato in allegato.