



S5/L5

VULNERABILITY SCAN E IMPLEMENTAZIONE
DI AZIONI DI RIMEDIO

Indice

- **Pagina 3:** Traccia dell'esercizio
- **Pagina 4-5:** Nessus Scanner
- **Pagina 6-7:** Apache Tomcat AJP Connector Request Injection (GhostCat)
- **Pagina 8-9:** Bind Shell Backdoor Detection
- **Pagina 10-11:** NFS Exported Share Information Disclosure
- **Pagina 12:** VNC Server 'password' Password
- **Pagina 13:** Conclusione

Traccia

Effettuare una scansione completa sul target Metasploitable.
Scegliete da un minimo di 2 fino ad un massimo di 4 vulnerabilità critiche / high e provate ad implementare delle azioni di rimedio.
Per dimostrare l'efficacia delle azioni di rimedio, eseguite nuovamente la scansione sul target e confrontate i risultati con quelli precedentemente ottenuti.

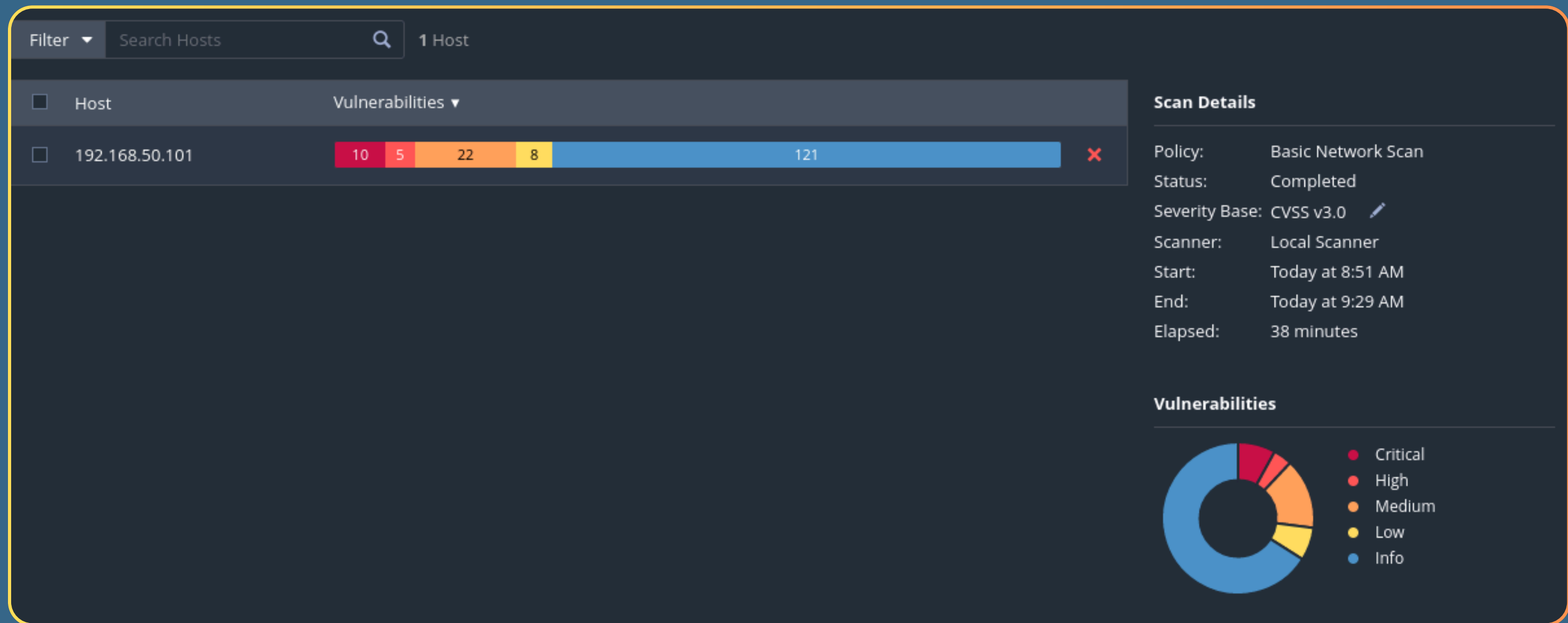
Nessus Scanner

Per la prova di oggi, si effettuerà un Vulnerability Scan della macchina Metasploitable con il programma **Nessus**.

Nessus è un vulnerability scanner che offre una vasta gamma di funzionalità tra cui scansioni automatiche e preimpostate, enumerazione e valutazioni di vulnerabilità note riscontrate e generazione di report dettagliati con [link a documentazione apposita](#).

Nessus ha un'interfaccia user friendly e abbastanza intuitiva.

Per lo scan dell'esercitazione, si utilizza la basic network scan: una scansione che andrà a scannerizzare le porte più comuni e utilizzate, insieme ai servizi a loro associate.



CRITICAL	9.8	9.0	134862	Apache Tomcat AJP Connector Request Injection (Ghostcat)
CRITICAL	9.8	-	51988	Bind Shell Backdoor Detection
CRITICAL	9.8	-	20007	SSL Version 2 and 3 Protocol Detection
CRITICAL	10.0	-	33850	Unix Operating System Unsupported Version Detection
CRITICAL	10.0*	5.1	32314	Debian OpenSSH/OpenSSL Package Random Number Generator Weakness
CRITICAL	10.0*	5.1	32321	Debian OpenSSH/OpenSSL Package Random Number Generator Weakness (SSL check)
CRITICAL	10.0*	5.9	11356	NFS Exported Share Information Disclosure
CRITICAL	10.0*	-	61708	VNC Server 'password' Password

Finita la scansione, è possibile vedere tutte le vulnerabilità riscontrate dal programma e classificate secondo il CVSS (Common Vulnerabilities Scoring System), che aiuta lo scanner ad associare un livello di rischio ben specifico a una data vulnerabilità.

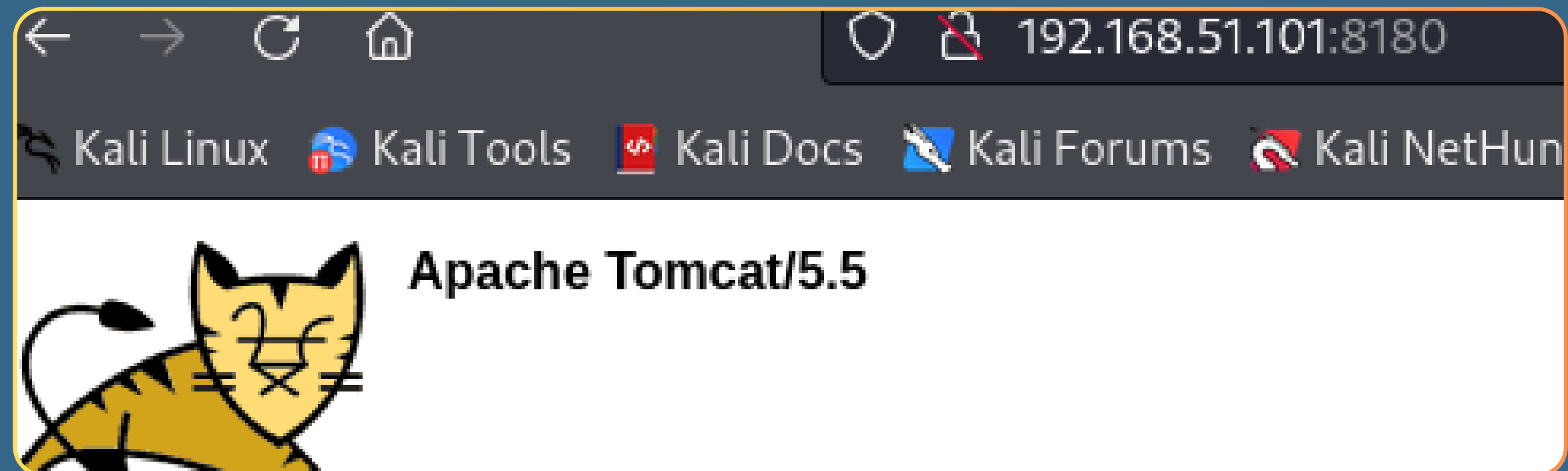
Andremo ad implementare in ordine, dall'alto verso il basso, delle azioni di rimedio sulle 4 vulnerabilità evidenziate.

Apache Tomcat AJP Connector Request Injection (GhostCat)

Questa vulnerabilità è stata riscontrata sulla porta 8009 della macchina target.

L'AJP Connector del servizio Apache Tomcat, potrebbe permettere ad un attaccante di leggere file di applicazioni web da un server vulnerabile e permetterebbe anche di ottenere l'accesso di esecuzione di comandi da remoto.

Possiamo verificare la versione di Tomcat attualmente installata su meta digitando l'IP della macchina target:8180 che è la porta a cui viene reindirizzato il servizio Tomcat.



Ci sono varie soluzioni al problema, in questo caso imposteremo l'AJP Connector in modo tale che richieda un'autorizzazione per essere usato.

Su meta ci sposteremo nella directory di tomcat, una volta lì apriremo il file server.xml che contiene le impostazioni di configurazione del servizio.

```
msfadmin@metasploitable: $ cd /etc/tomcat5.5
msfadmin@metasploitable:/etc/tomcat5.5$ ls
Catalina          context.xml       server-minimal.xml  tomcat-users.xml
catalina.policy   logging.properties  server.xml          web.xml
catalina.properties  policy.d         tomcat5.5
msfadmin@metasploitable:/etc/tomcat5.5$ sudo nano server.xml
[sudo] password for msfadmin:
```

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

```
'secretRequired="true" secret="macrokernel" />
```

Inserendo i parametri **secretRequired="true"** e **secret="microkernel"**, facciamo in modo che l'AJP connector richieda un'autorizzazione e che la password da inserire equivalga al parametro secret.

Questo andrà a mitigare la vulnerabilità

Bind Shell Backdoor Detection

Sulla porta 1524 è presente una backdoor che dà accesso ai comandi di root della macchina target tramite una shell. Per connettersi alla backdoor, basta utilizzare un software come netcat, inserendo l'IP e la porta interessati.

```
(kali@kali)-[~]  
$ nc 192.168.51.101 1524  
root@metasploitable:/#
```

Così abbiamo accesso di root. Per prima cosa dobbiamo individuare il processo che rende possibile il bindshell. Su meta utilizziamo la linea di comando **sudo lsof -i :1524** per trovare tutti i processi di rete attivi sulla porta 1524. Il comando lsof viene usato per ottenere informazioni sui file aperti in quel momento dai processi. Con l'opzione -i andiamo a chiedere informazioni riguardo i processi di rete e poi restringiamo ancora di più il campo aggiungendo la porta che ci interessa.


```
msfadmin@metasploitable:~$ sudo lsof -i :1524  
COMMAND  PID USER  FD  TYPE DEVICE SIZE NODE NAME  
inetd    4447 root   12u  IPv4  12053      TCP *:ingreslock (LISTEN)
```

In questo modo vediamo che il processo colpevole è il 4447 che è in ascolto sulla porta.

Usando il comando kill 4447 su meta, riusciamo ad eliminare il processo.


```
(kali@kali)-[~]  
$ nc 192.168.51.101 1524  
(UNKNOWN) [192.168.51.101] 1524 (ingreslock) : Connection refused
```

Questo però non basta a eliminare la backdoor definitivamente, ma la blocca per questo ciclo. Al riavvio della macchina, il processo riprenderà perciò dobbiamo creare una regola nel firewall per impedire l'accesso alla porta.

<input type="checkbox"/>		TCP	*	*	192.168.51.101	1524	*	none		Blocks connection to port 1524
--------------------------	---	-----	---	---	----------------	------	---	------	--	--------------------------------

Attivando questa regola nel firewall di PFSense, abbiamo bloccato tutto il traffico verso la porta 1524 della macchina meta.

```
(kali@kali)-[~]  
$ nc 192.168.51.101 1524  
(UNKNOWN) [192.168.51.101] 1524 (ingreslock) : Connection timed out
```

NFS Exported Share Information Disclosure

Network File Sharing permette di condividere directory e file con altri su una rete.

NFS è mal configurata sulla porta 2049 e permetterebbe a un potenziale attaccante di accedere alle directory condivise, leggerle e addirittura scriverci.

Configureremo due file di NFS : **Exports** e **Hosts.allow**.

```
GNU nano 2.0.7      File: /etc/exports      Modified
# /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)
```

Exports: File dove vengono specificate quali directories sono condivise dall'host con gli altri client e quali permessi vengono dati ai client. In questo caso specifico, vediamo che tutte le directories sono condivise, indicato con il segno /.

In questo file, specifichiamo che solo la directory /srv/nfs possa essere montata.

```
/srv/nfs *(rw,sync,no_root_squash)
```

Hosts.allow = File dove vengono specificati i client autorizzati ad accedere. Di default, chiunque può accedere all'NFS ma noi andremo a specificare che solo gli IP del network 192.168.51.0/24 possono accedere.

```
/etc/hosts.allow: list of hosts that are allowed to access the system.  
                  See the manual pages hosts_access(5) and hosts_options(5).
```

```
Example:    ALL: LOCAL @some_netgroup  
            ALL: .foobar.edu EXCEPT terminalserver.foobar.edu
```

```
If you're going to protect the portmapper use the name "portmap" for the  
daemon name. Remember that you can only use the keyword "ALL" and IP  
addresses (NOT host or domain names) for the portmapper, as well as for  
rpc.mountd (the NFS mount daemon). See portmap(8) and rpc.mountd(8)  
for further information.
```

```
sfd: 192.168.51.0/24
```

Con le nostre modifiche, il file sharing ora avviene solo tra i dispositivi nella rete interna 192.168.51.0/24.

VNC Server 'password' Password

Virtual Network Computing (VNC) è un servizio di accesso remoto usato principalmente per scopi amministrativi. La vulnerability scan ci segnala che sulla porta 5900, il servizio è protetto da una password debole e facilmente vulnerabile ad attacchi di tipo bruteforce.

Andremo a modificare la password per renderla più forte e più sicura, usando il comando `vncpasswd`.

```
msfadmin@metasploitable:~$ vncpasswd
Using password file /home/msfadmin/.vnc/passwd
VNC directory /home/msfadmin/.vnc does not exist, creating.
Password: _
```

Una volta cambiata la password, il sistema dovrebbe essere più sicuro.

Conclusione

Effettuiamo una nuova scansione con Nessus, per vedere se le vulnerabilità sono state effettivamente sanate.

Vulnerabilities					Total: 94
SEVERITY	CVSS V3.0	VPR SCORE	PLUGIN	NAME	
CRITICAL	9.8	-	20007	SSL Version 2 and 3 Protocol Detection	
CRITICAL	10.0	-	33850	Unix Operating System Unsupported Version Detection	
CRITICAL	10.0*	5.1	32314	Debian OpenSSH/OpenSSL Package Random Number Generator Weakness	
CRITICAL	10.0*	5.1	32321	Debian OpenSSH/OpenSSL Package Random Number Generator Weakness (SSL check)	
HIGH	8.6	5.2	136769	ISC BIND Service Downgrade / Reflected DoS	
HIGH	7.5	-	42256	NFS Shares World Readable	
HIGH	7.5	5.1	42873	SSL Medium Strength Cipher Suites Supported (SWEET32)	
HIGH	7.5	5.9	90509	Samba Badlock Vulnerability	

Le vulnerabilità segnalate prima non sono più riportate quindi le azioni di rimedio e mitigazione implementate hanno avuto esito positivo.