

Bind Shell Backdoor, Configurazione della Password e Apache Tomcat AJP

Report di Risoluzione delle Criticità: Bind Shell Backdoor Detection

Passo 1: Verifica della Presenza della Backdoor con Netcat

Per determinare se una backdoor fosse in ascolto sulla porta 1524, è stato utilizzato netcat. Ecco il comando eseguito:

```
nc 192.168.50.101 1524
```

Questo comando tenta di stabilire una connessione TCP alla porta 1524 del server remoto con IP 192.168.50.101. Se la porta è aperta e in ascolto, questo è un forte indicatore di una possibile backdoor.

Passo 2: Identificazione del Processo

Per confermare la presenza di una backdoor e identificare il processo che stava ascoltando sulla porta sospetta, è stato utilizzato netstat:

```
sudo netstat -tulnp | grep 1524
```

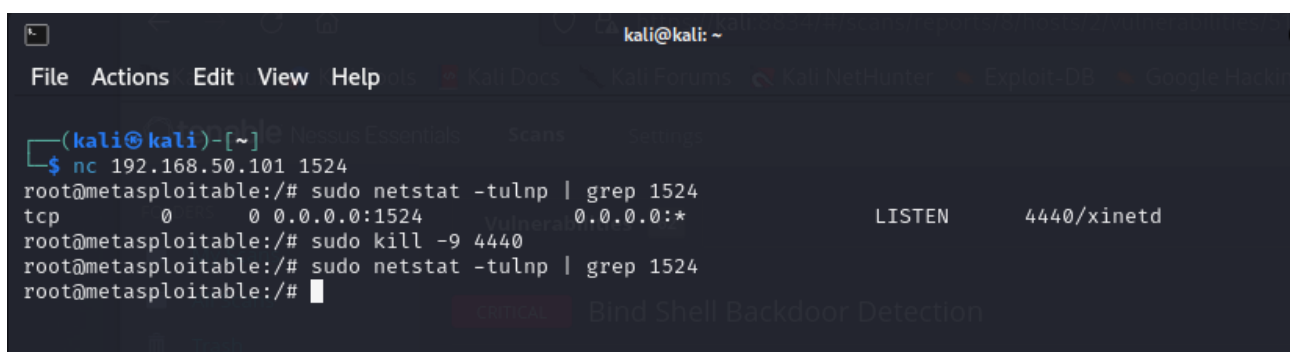
Questo comando mostra tutte le connessioni di rete attive e i processi in ascolto sulle porte, filtrando specificamente per la porta 1524.

Passo 3: Chiusura della Porta Sospetta

Una volta identificata la porta in ascolto (nel mio caso 4400), il processo associato è stato terminato utilizzando il comando kill:

```
sudo kill -9 4400
```

Questo comando termina forzatamente il processo con il PID 4400, chiudendo immediatamente la porta sospetta.



```
kali@kali: ~  
File Actions Edit View Help  
(kali@kali)-[~]  
$ nc 192.168.50.101 1524  
root@metasploitable:/# sudo netstat -tulnp | grep 1524  
tcp        0      0 0.0.0.0:1524          0.0.0.0:*          LISTEN      4440/xinetd  
root@metasploitable:/# sudo kill -9 4440  
root@metasploitable:/# sudo netstat -tulnp | grep 1524  
root@metasploitable:/#
```

Passo 4: Rimozione della Configurazione Dannosa

Successivamente, è stato modificato il file di configurazione `/etc/inetd.conf` per rimuovere la configurazione che consentiva la presenza della backdoor. Questo file è stato aperto con un editor di testo:

```
sudo nano /etc/inetd.conf
```

All'interno di questo file, è stata individuata e rimossa la seguente riga:

```
shell stream tcp nowait root /bin/sh sh -i
```

Questa riga configurava `inetd` per avviare una shell su richiesta via rete, creando una backdoor.

Passo 5: Verifica della Rimozione

Per assicurarsi che la backdoor fosse stata completamente rimossa, sono stati eseguiti nuovamente i controlli delle porte aperte e dei processi in ascolto:

```
sudo netstat -tulnp | grep 1524
```

Non è stato trovato nessun processo in ascolto sulla porta 1524, confermando che la backdoor era stata eliminata con successo.

Report di Risoluzione delle Criticità: Configurazione della Password del VNC Server

Passo 1: Accesso al Server

Per iniziare, ci siamo connessi al server Linux utilizzando SSH o direttamente tramite una console locale. Una volta connessi, abbiamo ottenuto privilegi elevati per eseguire le operazioni necessarie.

Passo 2: Elevazione ai Privilegi Root

Per elevare i privilegi, abbiamo utilizzato `sudo su` per diventare l'utente root: questo comando ci ha fornito i privilegi necessari per modificare le impostazioni del server VNC.

Passo 3: Impostazione della Password del VNC

Una volta ottenuti i privilegi di root, abbiamo impostato la password del server VNC utilizzando il comando `vncpasswd`. Questo comando configura la password che verrà utilizzata quando ci si connette al server VNC sulla porta 5900.

Il sistema ha richiesto di inserire e confermare la nuova password per il server VNC. Questa password sarà necessaria per qualsiasi connessione al server VNC.

Nota: È importante scegliere una password complessa e sicura per proteggere l'accesso al server.

```
The programs included with the Ubuntu system are free software;
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 su
[sudo] password for msfadmin:
root@metasploitable:/home/msfadmin# vnc
vncconnect vncpasswd vncserver
root@metasploitable:/home/msfadmin# vnc
vncconnect vncpasswd vncserver
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#
```

Report di Risoluzione delle Criticità: Apache Tomcat AJP Connector Request Injection (Ghostcat)

Passo 1: Navigazione nella Directory di Configurazione di Tomcat

Abbiamo navigato nella directory di configurazione di Tomcat, che solitamente si trova in `/etc/tomcat/` o `/usr/local/tomcat/conf/`.

```
cd /etc/tomcat/
```

Passo 2: Modifica del File di Configurazione `server.xml`

Il file di configurazione principale di Tomcat, `server.xml`, contiene le configurazioni del connettore AJP. Abbiamo aperto questo file con un editor di testo.

```
sudo nano server.xml
```

Passo 3: Commentare o Rimuovere il Connettore AJP

All'interno del file `server.xml`, abbiamo cercato la configurazione del connettore AJP. Solitamente, appare come segue:

```
<Connector port="8009" protocol="AJP/1.3" redirectPort="8443" />
```

Per disabilitare il connettore, abbiamo commentato questa linea aggiungendo `<!--` e `-->` attorno alla configurazione:

```
<!--  
<Connector port="8009" protocol="AJP/1.3" redirectPort="8443" />  
-->
```

In alternativa, la linea può essere completamente rimossa.

Motivo della Disabilitazione

Perché Disabilitare il Connettore AJP

Eliminazione del Vettore di Attacco: Disabilitando il connettore AJP, abbiamo eliminato il vettore di attacco utilizzato dalla vulnerabilità Ghostcat.

Semplicità: La disabilitazione del connettore è un metodo rapido ed efficace per mitigare la vulnerabilità, in attesa di un'eventuale patch o aggiornamento di sicurezza.

Quando Disabilitare il Connettore AJP:

Ambienti Non Dipendenti da AJP: Se il connettore AJP non è utilizzato nelle operazioni quotidiane del server, la disabilitazione è una soluzione rapida ed efficace per mitigare la vulnerabilità Ghostcat, come appunto in questo caso

Necessità di Mitigazione Immediata: Se è necessaria una protezione immediata e non è possibile aggiornare il software in tempi brevi, la disabilitazione è la soluzione più rapida.

```
<!-- 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"  
-->
```

