W12D4 - Analisi delle vulnerabilità e azioni di rimedio (Remediations)

INTRODUZIONE

Il presente report analizza alcune delle vulnerabilità trovate attraverso la scansione con Nessus (vedi i documenti "Scansione Inizio.pdf" e "Appendice A - Report di Scansione Nessus") e illustra alcuni dei metodi per sanarle (Remediations).

Giova sottolineare, innanzitutto, che ho effettuato delle scansioni sia con la VM Metasploitable su rete separata - interfacciata alla macchina Host tramite firewall PFSense - che con la VM Metasploitable nella stessa rete della macchina Host. Alcune delle vulnerabilità analizzate, in particolar modo "rlogin" e "rsh" legate al servizio "rexecd" sono state però rilevate solo quando la VM Metasploitable è stata posta nella stessa rete della macchina Host. Per questo motivo, ho utilizzato questa modalità lasciando al server DHCP il compito di dare il giusto IP alla VM.

Per un elenco completo delle vulnerabilità rilevate dopo le correzioni, e per i dettagli tecnici completi, si veda l'**Appendice B: Secondo Report di Scansione Nessus** allegato a questo documento.

~~~

SOMMARIO

Nel precedente report (W12D4 - Analisi delle vulnerabilità e azioni di rimedio) ho evidenziato come la VM Metasploitable fosse esposta a un rischio di compromissione estremamente elevato. La prima scansione aveva infatti rilevato un totale di 187 vulnerabilità, con rispettivamente 9 vulnerabilità Critical, 7 vulnerabilità High, 26 vulnerabilità Medium, 9 vulnerabilità Low e 136 vulnerabilità Info.

Oltre alla scansione, la seconda parte dell'esercitazione chiedeva poi di sanare alcune di queste vulnerabilità. Le prime quattro, preselezionate, riguardavano il server VNC, la presenza di una backdoor, un errore nella configurazione del server NFS e una suite di comandi remoti. Oltre queste, ho poi risolto **una vulnerabilità critica relativa ad Apache Tomcat**.

Tre delle cinque vulnerabilità sanate sono classificate Critical per la loro pericolosità. In base alla classificazione del rischio, quella più pericolosa riguardava l'utilizzo di una password debole per il servizio VNC, che apriva la porta alla possibilità, per un attaccante, di ottenere pieno accesso grafico e interattivo alla VM.

Altrettanto critiche le vulnerabilità relative alla presenza di un servizio di rete in ascolto su una porta della VM che consentiva all'attaccante, senza alcuna autenticazione, di connettersi a questa porta ed ottenere facilmente l'accesso diretto a una shell di comando e quella relativa al connettore AJP di Apache Tomcat che consentiva, qualora sfruttata, di leggere qualsiasi file sul sistema a cui il processo Tomcat aveva accesso in lettura.

A rendere particolarmente insidiose invece le vulnerabilità "rlogin" e "rsh" è la loro anzianità di servizio e il fatto che nessuna delle loro operazioni è crittografata. Sia le operazioni di login che quelle relative ai comandi remoti transitano in chiaro sulla rete. L'unica strada percorribile per sanare le vulnerabilità è stata quella di disattivare i due servizi.
~~~

VULNERABILITÀ VNC Server 'password' Password

La vulnerabilità VNC Server 'password' Password segnalata da Nessus fa parte delle vulnerabilità di livello Critical. Identificata dal plugin 61708 di Nessus, affligge il servizio VNC (Virtual Network Computing) in esecuzione sulla porta TCP 5900.

Nello specifico, il server VNC in esecuzione sulla VM Metasploitable è configurato per utilizzare una password debole ("password") per l'autenticazione. Essendo VNC un sistema di condivisione grafica del desktop che permette di controllare un computer a distanza, l'utilizzo di una password così debole dà la possibilità all'attaccante di ottenere pieno accesso grafico e interattivo alla VM. A seguire i dettagli tecnici:

• VNC Server 'password' Password

- **SYNOPSIS**: A VNC server running on the remote host is secured with a weak password
- **DESCRIPTION**: The VNC server running on the remote host is secured with a weak password. Nessus was able to login using VNC authentication and a password of 'password'. A remote, unauthenticated attacker could exploit this to take control of the system
- **SOLUTION**: Secure the VNC service with a strong password
- **RISK FACTOR**: Critical
- **CVSS v2.0 Base Score**: 10.0 (CVSS2#AV:N/AC:L/Au:N/C:C/I:C/A:C)
- **PLUGIN OUTPUT**: tcp/5900/vnc

Chiaramente la soluzione migliore prevede di configurare nuovamente il server VNC, così da sostituire la precedente password debole con una password più forte. Per ottenere questo risultato, ho innanzitutto **individuato il processo VNC in esecuzione** (ps aux | grep vnc) e, dopo averlo individuato, **l'ho stoppato** (kill PID).

Una volta accertatomi che il servizio fosse effettivamente stoppato, ho utilizzato il comando **sudo -H vncpasswd** per settare una nuova password. Ho utilizzato in questo caso una password forte, sanando la vulnerabilità. Per controllare l'effettiva scomparsa della vulnerabilità, ho riavviato il servizio VNC e ho effettuato una nuova scansione con Nessus. Come visibile nell'**Appendice B: Secondo Report di Scansione Nessus** allegato a questo documento, la vulnerabilità non è più presente.

```
msfadmin@metasploitable:/home/service$ ps aux | grep vnc
root      4890  0.1  0.5  14044 12076 ?        S    06:08   0:04 Xtightvnc :0 -d
esktop X -auth /root/.Xauthority -geometry 1024x768 -depth 24 -rfbwait 120000 -r
fbauth /root/.vnc/passwd -rfbport 5900 -fp /usr/X11R6/lib/X11/fonts/Type1/,/usr/
X11R6/lib/X11/fonts/Speedo/,/usr/X11R6/lib/X11/fonts/misc/,/usr/X11R6/lib/X11/fo
nts/75dpi/,/usr/X11R6/lib/X11/fonts/100dpi/,/usr/share/fonts/X11/misc/,/usr/shar
e/fonts/X11/Type1/,/usr/share/fonts/X11/75dpi/,/usr/share/fonts/X11/100dpi/ -co
/etc/X11/rgb
root      4904  0.0  0.0   2724  1184 ?        S    06:08   0:00 /bin/sh /root/.
vnc/xstartup
msfadmin  6671  0.0  0.0   3008   780 tty1     S+   06:50   0:00 grep vnc
```

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

VULNERABILITÀ Bind Shell Backdoor Detection

La vulnerabilità "Bind Shell Backdoor Detection" indica la presenza di un servizio di rete in ascolto su una porta della VM Metasploitable (in questo caso la 1524) che fornisce accesso diretto a una shell di comando, **senza alcuna richiesta di autenticazione**.

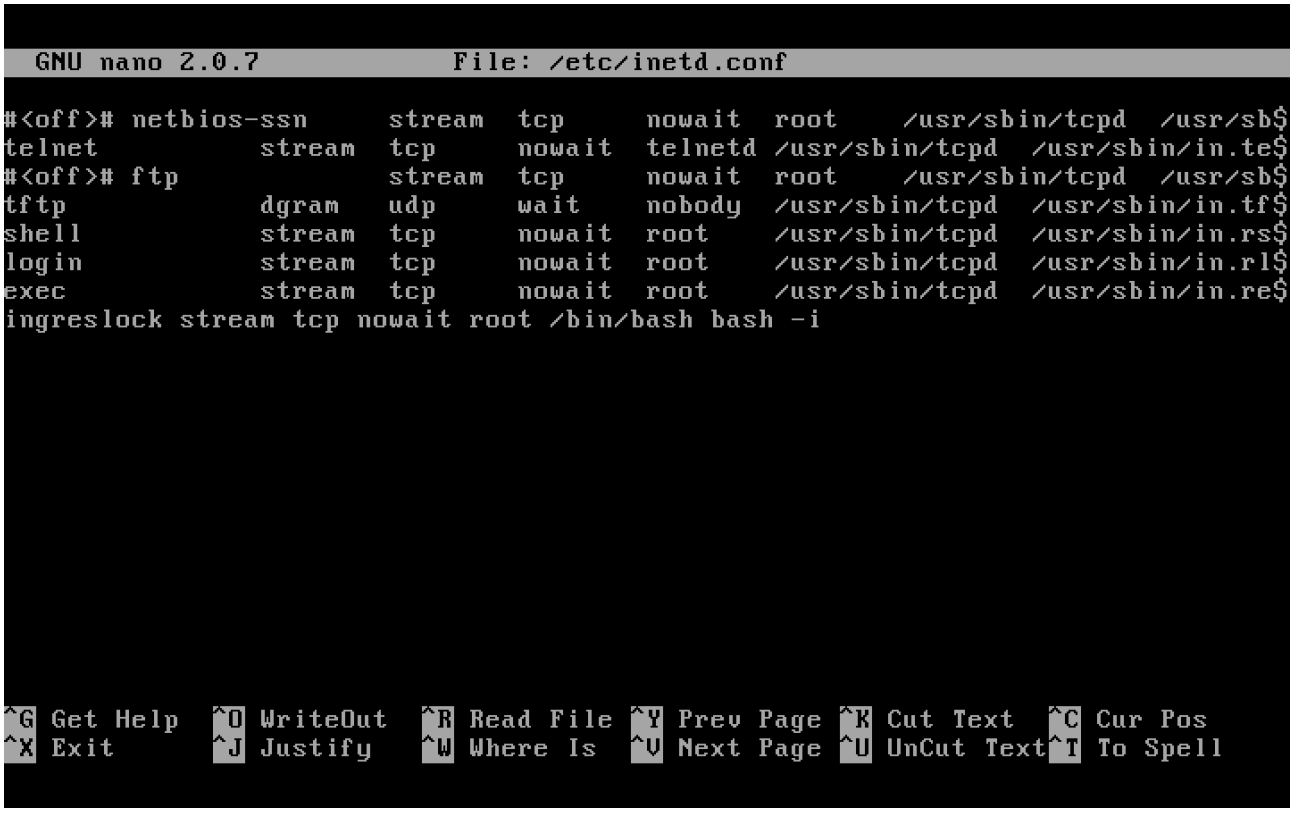
Un attaccante potrebbe quindi connettersi a questa porta, ottiene facilmente l'accesso diretto a una shell di comando sul sistema compromesso. In particolare, la shell in oggetto era configurata per ottenere i privilegi di root, dando quindi all'attaccante il controllo completo sulla macchina.

• 51988 - Bind Shell Backdoor Detection

- **SYNOPSIS**: The remote host may have been compromised
- **DESCRIPTION**: A shell is listening on the remote port without any authentication being required. An attacker may use it by connecting to the remote port and sending commands directly
- **SOLUTION**: Verify if the remote host has been compromised, and reinstall the system if necessary
- **RISK FACTOR**: Critical
- **CVSS v3.0 Base Score**: 9.8 (CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H)
- **PLUGIN OUTPUT**: tcp/1524/wild_shell

La risoluzione di questa vulnerabilità passa proprio dall'analisi del servizio in ascolto. Tramite il comando netstat ho rilevato che il processo associato alla vulnerabilità è xinetd, un applicativo per servizi di rete. La configurazione della backdoor aggiungeva quindi un'istruzione a xinetd, in modo da avviare il servizio (e la shell associata).

Dall'analisi del servizio in ascolto (processo xinetd sulla porta 1524), e incrociando le informazioni con il file /etc/services, ho identificato il servizio "**ingreslock**" come responsabile. Dopo una laboriosa ricerca, ho trovato la sua riga di configurazione all'interno del file legacy /etc/inetd.conf, confermando che xinetd utilizzava questa configurazione per avviare la shell. Commentando la riga di configurazione all'interno del file /etc/inetd.conf e riavviando il servizio xinetd, la vulnerabilità è stata sanata.



VULNERABILITÀ NFS Shares World Readable

La vulnerabilità NFS Shares World Readable affligge il server NFS (Network File System) della VM Metasploitable, e consiste **nell'esportazione senza restrizioni di una o più directory del filesystem**.

Nello specifico, questa vulnerabilità segnala che la directory radice (/) è fra quelle esportate senza restrizioni. Un attaccante può utilizzare questa vulnerabilità per accedere a file sensibili sulla macchina target, riuscendo così a leggerli, modificarli o cancellarli.

42256 - NFS Shares World Readable

- - **SYNOPSIS**: The remote NFS server exports world-readable shares
- - **DESCRIPTION**: The remote NFS server is exporting one or more shares without restricting
- access (based on hostname, IP, or IP range)
- - **SOLUTION**: Place the appropriate restrictions on all NFS shares
- - **RISK FACTOR**: Medium
- **CVSS v3.0 Base Score**: 7.5 (CVSS:3.0/AV:N/AC
- **PLUGIN OUTPUT**: tcp/2049/rpc-nfs

La chiusura della vulnerabilità NFS Shares World Readable passa dalla corretta configurazione delle esportazioni NFS. Modificando il file di configurazione /etc/exports ho fatto in modo che la directory radice (/) non fosse più esportata. Qualora ci fosse la necessità di esportare la directory radice, nel file di configurazione è possibile settare quali macchine possono esportare la directory.

Commentando la riga di configurazione nel file /etc/exports e riavviando il servizio di esportazione NFS con il comando **sudo exportfs -a**, ho rilevato che la vulnerabilità è stata sanata.

```
msfadmin@metasploitable:~$ sudo cat /etc/exports
[sudo] password for msfadmin:
# /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)
msfadmin@metasploitable:~$
```

VULNERABILITÀ rlogin Service Detection E rsh Service Detection

I servizi "rlogin" e "rsh", responsabili delle due vulnerabilità, fanno parte della stessa suite per il lancio di comandi remoti ("r-commands"). Nello specifico, "rlogin" consente a un utente di accedere a un sistema remoto, mentre "rsh" consente a un utente di eseguire comandi su un sistema remoto.

La vulnerabilità di "rsh" si lega alle stesse debolezze, già viste con "rlogin", relative all'autenticazione basata su IP (con bypass tramite IP Spoofing) e alla totale assenza di crittografia per i comandi e l'output. Questo rende tutto il traffico facilmente intercettabile. Il terzo componente degli "r-commands" è "rexecd", che ha le stesse funzioni di "rsh" ma utilizza esclusivamente nome utente e password per l'accesso. Anche in questo caso, però, l'intero processo di autenticazione è completamente in chiaro. Quindi nemmeno "rexecd" può essere considerato un servizio sicuro.

10205 - rlogin Service Detection

REFERENCE: CVE-1999-0651

- **SYNOPSIS**: The rlogin service is running on the remote host
- **DESCRIPTION**: The rlogin service is running on the remote host. This service is vulnerable since
- data is passed between the rlogin client and server in cleartext. A man-in-the-middle attacker can
- exploit this to sniff logins and passwords. Also, it may allow poorly authenticated logins without
- passwords. If the host is vulnerable to TCP sequence number guessing (from any network) or IP
- spoofing (including ARP hijacking on a local network) then it may be possible to bypass
- authentication. Finally, rlogin is an easy way to turn file-write access into full logins through
- the .rhosts or rhosts.equiv files.
- **SOLUTION**: Comment out the 'login' line in /etc/inetd.conf and restart the inetd process.
- Alternatively, disable this service and use SSH instead
- **RISK FACTOR**: High
- **CVSS v2.0 Base Score**: 7.5 (CVSS2#AV:N/AC:L/Au:N/C:P/I:P/A:P)
- **PLUGIN OUTPUT**: tcp/513/rlogin

10245 - rsh Service Detection

REFERENCE: CVE-1999-0651

- **SYNOPSIS**: The rsh service is running on the remote host
- **DESCRIPTION**: The rsh service is running on the remote host. This service is vulnerable since
- data is passed between the rsh client and server in cleartext. A man-in-the-middle attacker can
- exploit this to sniff logins and passwords. Also, it may allow poorly authenticated logins without
- passwords. If the host is vulnerable to TCP sequence number guessing (from any network) or IP
- spoofing (including ARP hijacking on a local network) then it may be possible to bypass
- authentication. Finally, rsh is an easy way to turn file-write access into full logins through
- the .rhosts or rhosts.equiv files
- **SOLUTION**: Comment out the 'rsh' line in /etc/inetd.conf and restart the inetd process.
- Alternatively, disable this service and use SSH instead
- **RISK FACTOR**: High
- **CVSS v2.0 Base Score**: 7.5 (CVSS2#AV:N/AC:L/Au:N/C:P/I:P/A:P)
- **PLUGIN OUTPUT**: tcp/514/rsh

A causa dell'obsolescenza e della pericolosità di questa suite di comandi, l'unica soluzione per mitigare la vulnerabilità è quella di disattivarli completamente. Essendo tutti questi servizi gestiti da **inetd,** per la disattivazione ho dovuto modificare la configurazione di inetd stesso. Nello specifico, ho commentato (escludendole quindi dall'esecuzione) le voci shell, login ed exec.

Dopo il riavvio del servizio inetd, le vulnerabilità risultano sanate.

VULNERABILITÀ 134862 - Apache Tomcat AJP Connector Request Injection (Ghostcat)

Come si evince dal nome, questa vulnerabilità riguarda il protocollo AJP (Apache JServ Protocol) all'interno di Tomcat. Nello specifico, il **connettore AJP** è abilitato per impostazione predefinita sulla porta 8009, ed è progettato per consentire a un server web front-end (come Apache HTTP Server) di comunicare efficientemente con un'istanza Tomcat back-end.

La vulnerabilità consente a un attaccante non autenticato di inviare richieste AJP appositamente create alla porta 8009. Questo difetto permette principalmente all'attaccante di leggere qualsiasi file sul sistema a cui il processo Tomcat ha accesso in lettura. In determinate circostanze, e se l'attaccante può scrivere file sul server, può evolvere in esecuzione di codice remoto.

134862 - Apache Tomcat AJP Connector Request Injection (Ghostcat)

REFERENCE: CVE-2020-1745
CVE-2020-1938

- **SYNOPSIS**: There is a vulnerable AJP connector listening on the remote host.
- **DESCRIPTION**: A file read/inclusion vulnerability was found in AJP connector. A remote,
- unauthenticated attacker could exploit this vulnerability to read web application files from a
- vulnerable server. In instances where the vulnerable server allows file uploads, an attacker could
- upload malicious JavaServer Pages (JSP) code within a variety of file types and gain remote code
- execution (RCE)
- **SOLUTION**: Update the AJP configuration to require authorization and/or upgrade the Tomcat
- server to 7.0.100, 8.5.51, 9.0.31 or later.
- **RISK FACTOR**: High
- **CVSS v3.0 Base Score**: 9.8 (CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H)
- **PLUGIN OUTPUT**: tcp/8009/ajp13

Sono fondamentalmente due le opzioni per mitigare questa vulnerabilità: la prima è aggiornare Apache Tomcat a una versione patchata che risolve la vulnerabilità. La seconda è quella di **disabilitare completamente il connettore**, qualora questo non serva. Vista l'impossibilità di aggiornare la VM Metasploitable, ho optato per la disattivazione del connettore.

Nella pratica, ho **modificato il file di configurazione server.xml** di Apache Tomcat così da cancellare l'intera riga relativa al connettore AJP. Questa azione impedisce a Tomcat di aprire e mettersi in ascolto su tale porta tramite il protocollo AJP, rendendo di fatto impossibile per un attaccante sfruttare la vulnerabilità.

Dopo il riavvio del servizio Apache Tomcat, la vulnerabilità è sanata. Come si può però vedere dall'Appendice B: Secondo Report di Scansione Nessus, allegato a questo documento, permangono delle vulnerabilità - anche critiche - relative ad Apache Tomcat.

~~~

ULTERIORI INFORMAZIONI

Per un elenco completo delle vulnerabilità rilevate sia prima che dopo le correzioni, si faccia riferimento alle Appendici A e B in allegato ai due report.
~~~