

Pentest com Kali Linux





Instrutor: Vitor Mazuco

<http://facebook.com/vitormazuco>

Email: vitor.mazuco@gmail.com

WebSite: <http://vmzsolutions.com.br>

Damn Vulnerable Web Application (DVWA) é uma aplicação web em PHP/MySQL que é extremamente vulnerável. Seu principal objetivo é ser um auxílio para profissionais de segurança para testar suas habilidades e ferramentas em um ambiente legal, ajudar os desenvolvedores web a entender melhor os processos de segurança de aplicações web e ajudar tanto os alunos e professores a aprender sobre a segurança das aplicações web em um lugar controlado.

O objetivo do DVWA é praticar algumas das vulnerabilidades mais comuns da web, com vários níveis difíceis, com uma interface simples e direta. Tenha em atenção que existem vulnerabilidades documentadas e não documentadas com este software. Isso é intencional. Você é encorajado a tentar descobrir o maior número possível de problemas.

<http://192.168.1.163/dvwa/>

[Home](#)[Instructions](#)[Setup](#)[Brute Force](#)[Command Execution](#)[CSRF](#)[Insecure CAPTCHA](#)[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.



No menu à esquerda, selecione **SQL Injection**



Home

Instructions

Setup

Brute Force

Command Execution

CSRF

Insecure CAPTCHA

File Inclusion

SQL Injection

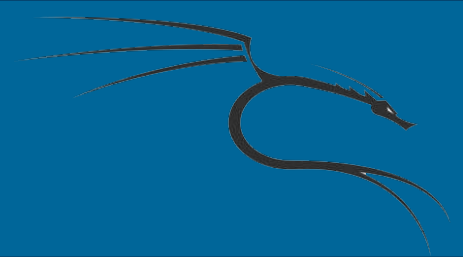
SQL Injection (Blind)

Vulnerability: SQL Injection

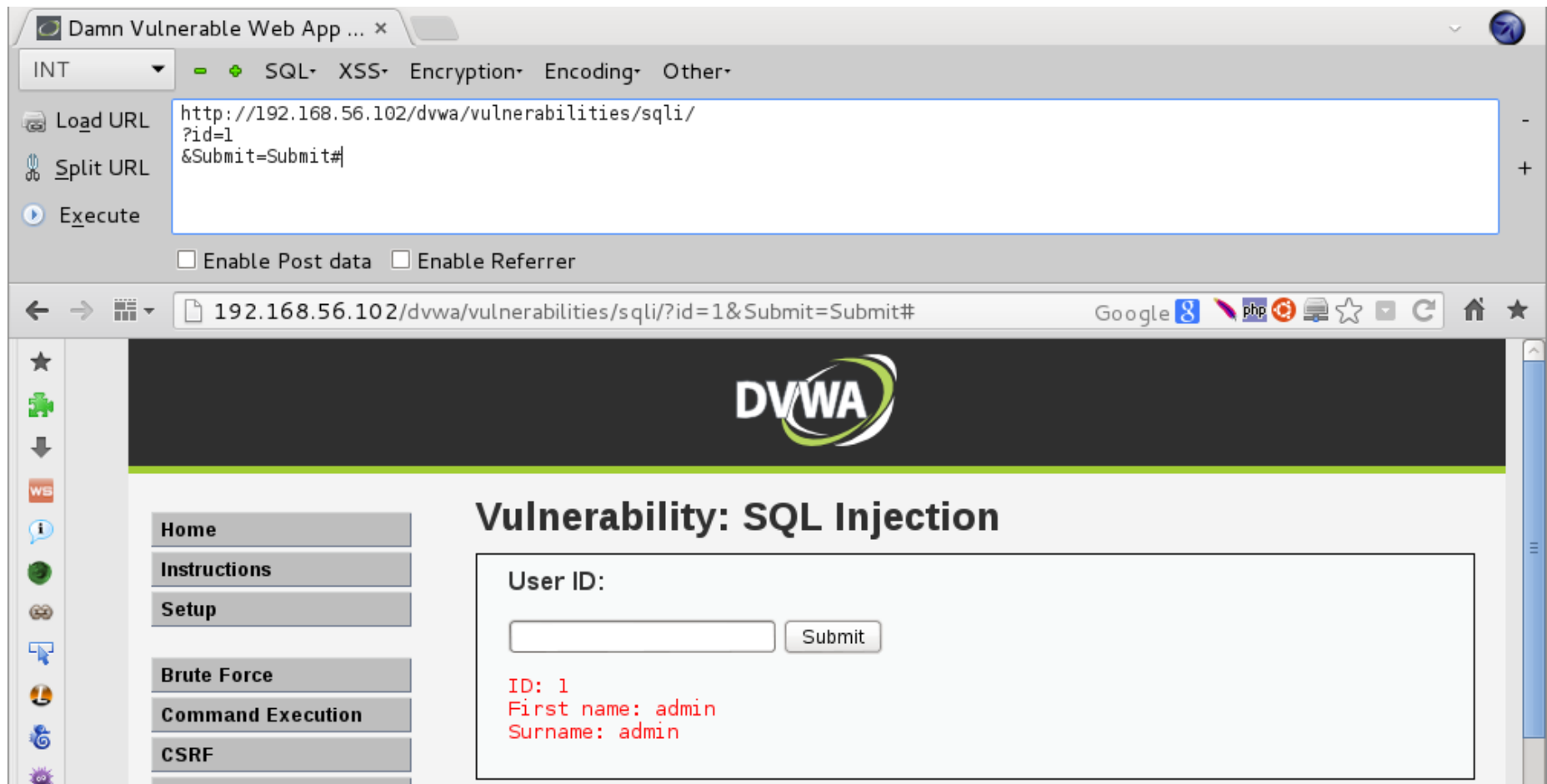
User ID:

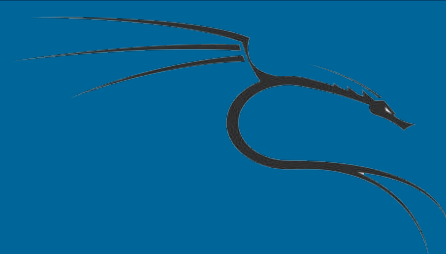
More info

<http://www.securiteam.com/securityreviews/5DP0N1P76E.html>
http://en.wikipedia.org/wiki/SQL_injection
<http://ferruh.mavituna.com/sql-injection-cheatsheet-oku/>
<http://pentestmonkey.net/cheat-sheet/sql-injection/mysql-sql-injection-cheat-sheet>



Digite um número na caixa de texto de **User ID** e clique em **Submit**.

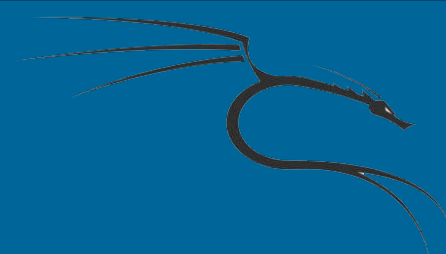




Vamos fazer uma modificação simples, alterar o valor do parâmetro **id** de 1 para 2 e Clique em **Execute**.

The screenshot shows the DVWA web application interface. At the top, there's a navigation bar with tabs: INT, SQL, XSS, Encryption, Encoding, and Other. The SQL tab is selected. Below the tabs, there's a section for URL manipulation with buttons for Load URL, Split URL, and Execute. The URL field contains: `http://192.168.56.102/dvwa/vulnerabilities/sqli/?id=2&Submit=Submit#`. Below the URL field, there are checkboxes for "Enable Post data" and "Enable Referrer". On the left side, there's a sidebar with a star icon and a list of links: Home, Instructions, Setup, Brute Force, Command Execution, CSRF, and Insecure CAPTCHA. The main content area is titled "Vulnerability: SQL Injection". It contains a form with a label "User ID:" and an input field. To the right of the input field is a "Submit" button. Below the input field, the results are displayed in red text: "ID: 2", "First name: Gordon", and "Surname: Brown".

Podemos ver que o parâmetro **id** corresponde à URL da página, portanto, usando a **Hackbar**, podemos tentar qualquer valor modificando o **ID** em vez de alterar o **ID** via URL. Isso é útil quando se testa um formulário com muitas entradas ou que redireciona para outras páginas, dependendo delas.



Substituímos um valor inválido, olha o que acontece.

