Subject: Network Security Lab

Assignment: hacking web application using DVWA

Solution:

DVWA (damn vulnerable web app)

Damn Vulnerable Web Application, shorter DVWA, is a PHP/MySQL web application that is damn vulnerable.

The main goal of this pentesting playground is to aid penetration testers and security professionals to test their skills and tools. In addition it can aid web devs better understand how to secure web apps, but also to aid students/teachers to learn all about web app security and possible vulnerabilities.

SQL injection

SQL injection is a code injection technique that might destroy your database.

SQL injection is one of the most common web hacking techniques.

SQL injection is the placement of malicious code in SQL statements, via web page input.

SQL injection usually occurs when you ask a user for input, like their username/userid, and instead of a name/id, the user gives you an SQL statement that you will unknowingly run on your database.

How to prevent SQL injection?

Use prepared statements and parameterized queries

Parameterized Statements ensure that the parameters passed into the SQL Statements are treated safely.

Object-relational mapping - Most development teams prefer to use Object Relational Mapping frameworks to translate SQL result sets into code objects more seamlessly.

Escaping inputs - It is a simple way to protect against most SQL injection attacks. Many languages have standard functions to achieve this. You need to be aware while using escape characters in your code base where an SQL statement is constructed.

Xss attacks

Cross-Site Scripting (XSS) attacks are a type of injection in which vulnerabilities are typically found in Web applications.

XSS attacks occur when an attacker uses a web application to send malicious code, generally in the form of a browser side script, to a different end user.

XSS enables attackers to inject client-side scripts into Web pages viewed by other users.

A cross-site scripting vulnerability may be used by attackers to bypass access controls such as the same origin policy.

There are two types of XSS: persistent and non-persistent XSS. In persistent XSS, the malicious code is saved to the server in the database. Afterwards, the code is executed on the page. However, in non-persistent XSS, the attacker sends the injected malicious code to the server through an HTTP request.

What are the types of XSS attacks?

There are three main types of XSS attacks. These are: Reflected XSS, where the malicious script comes from the current HTTP request.

Stored XSS, where the malicious script comes from the website's database.

DOM-based XSS, where the vulnerability exists in client-side code rather than server-side code

How to prevent XSS attacks

Filter input on arrival: - At the point where user input is received, filter as strictly as possible based on what is expected or valid input.

Encode data on output: - At the point where user-controllable data is output in HTTP responses, encode the output to prevent it from being interpreted as active content. Depending on the output context, this might require applying combinations of HTML, URL, JavaScript, and CSS encoding.

Use appropriate response headers: - To prevent XSS in HTTP responses that aren't intended to contain any HTML or JavaScript, you can use the Content-Type and X-Content-Type-Options headers to ensure that browsers interpret the responses in the way you intend.

Content Security Policy: - As a last line of defense, you can use Content Security Policy (CSP) to reduce the severity of any XSS vulnerabilities that still occur.