W.A.P ATTACK

GUIDE

CLICK HERE

Welcome to WAP!

Learning made easy for `Web Application and Penitration`

Downloads →

ISO files ,PDF and many other goodies.

Start Learning →

Lets Begin the Hunger Games.

Linux Online →

Getting on with the basics.

About us →

Wanna Know more about us, click here!!

Welcome to WAP!

Learning made easy for `Web Application and Penitration`

XSS Attack →

ISO files ,PDF and many other goodies.

HTML Insertion →

Lets Begin the Hunger Games.

CSRF Attack →

Getting on with the basics.

PassWORD Attacks →

Wanna Know more about us, click here!!

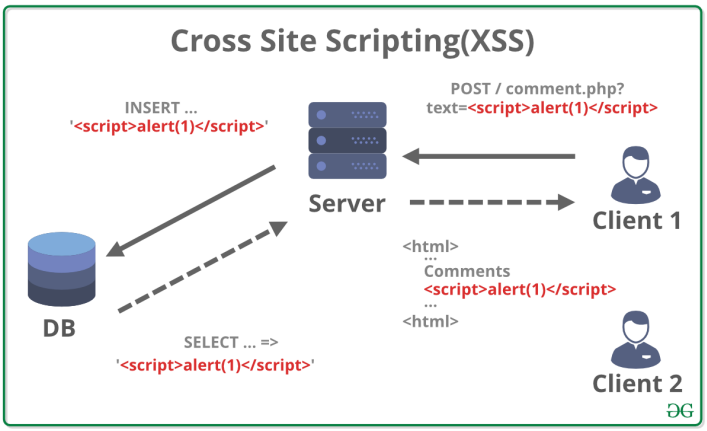
SQL Injection →

Wanna Know more about us, click here!!

XSS attack

EASY THAN YOU THINK

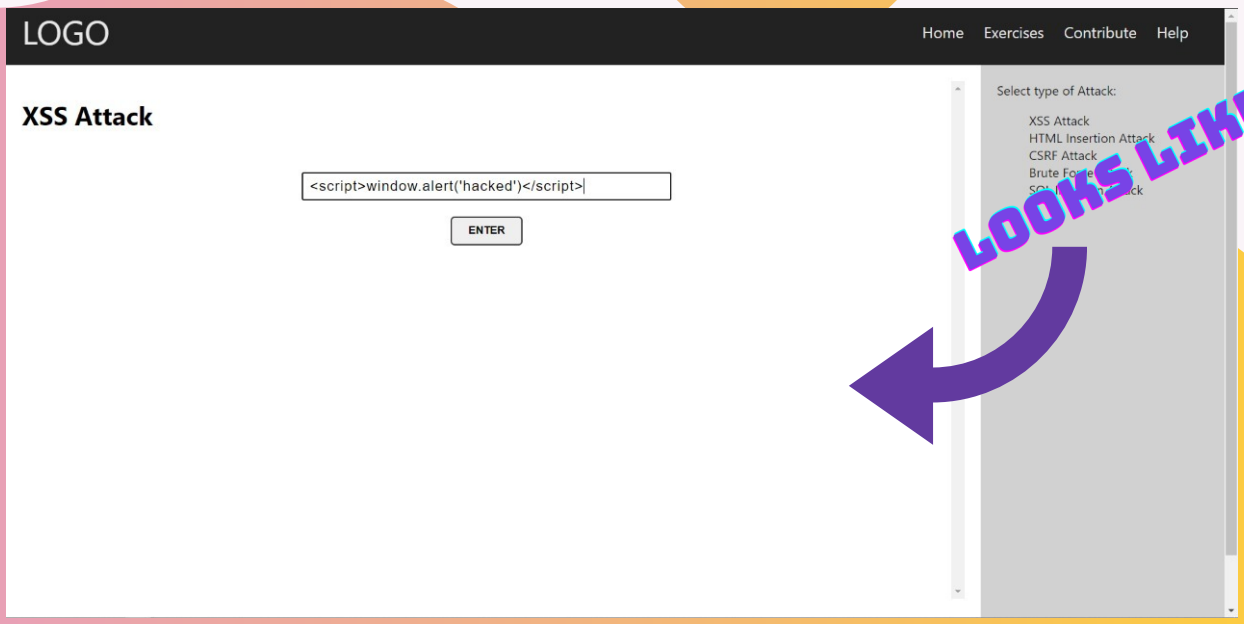
Cross-Site Scripting (XSS) attacks are a type of injection, in which malicious scripts are injected into otherwise benign and trusted websites. XSS attacks occur when an attacker uses a web application to send malicious code, generally in the form of a browser side script, to a different end user.



In general, effectively preventing XSS vulnerabilities is likely to involve a combination of the following measures: 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.

HOW TO?

1.Go to xss attack page



2.Type this command in the search bar and press enter.

```
<script>window.alert('hacked)</script>
```

3.You should see an alert window which means this was successful

HTML

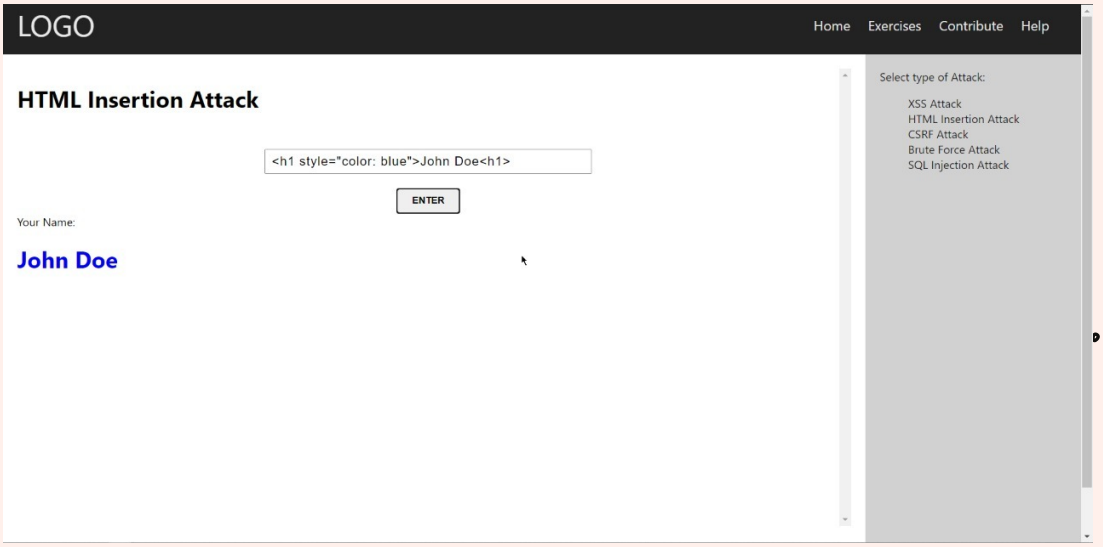
INSERTION ATTACK

HTML injection is a type of injection vulnerability that occurs when a user is able to control an input point and is able to inject arbitrary HTML code into a vulnerable web page. ... For example, malicious HTML code can be injected via the innerHTML JavaScript method, usually used to render user-inserted HTML code.

The most common way of detecting HTML injection is by looking for HTML elements in the incoming HTTP stream that contains the user input. A naïve validation of user input simply removes any HTML-syntax substrings (like tags and links) from any user-supplied text. To avoid false positives, the security mechanism that detects possible injections and protects the application should learn in what application context user input is allowed to contain HTML. Also, it should be able to stop HTML input if it learns that such text is pasted as-is in web page generated by vulnerable application components.

Instructions

1.Open the HTML insertion attack page



2.Now type the following in the search bar to de-face the website

```
<h1> style="color: blue">your name</h1>
```

3.Did you see your name in blue in the webpage ?

THIS IS

SQL

INJECTION ATTACK

Retrieving hidden data, where you can modify an SQL query to return additional results.

Subverting application logic, where you can change a query to interfere with the application's logic. UNION attacks, where you can retrieve data from different database tables.

Prevention Methods

1. Validate User Inputs.
2. Sanitize Data by Limiting Special Characters.
3. Enforce Prepared Statements and Parameterization.
4. Use Stored Procedures in the Database.
5. Actively Manage Patches and Updates.
6. Raise Virtual or Physical Firewalls.
7. Harden Your OS and Applications.

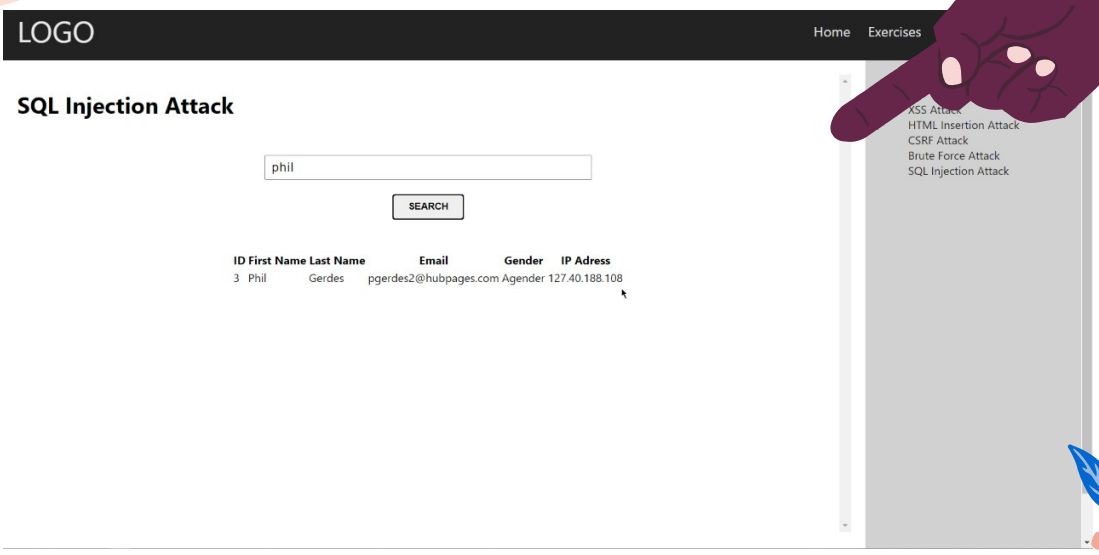
STEPS TO FOLLOW

Beginner friendly

Let do something that is easy and will let us see all the details in the database. when you type in the search bar you should receive a all the information related to the search

1. type the name 'phil' and press search

you should see some results down below . but what if you wanna see other people and the details? Without a valid name its a useless endeavor or is it ?



Okay, now to find all the information we will use sql injection

```
2.type this " ';-- "
```

Press enter and you should see almost all the results in the database as the will be dumped in the webpage

so whats happening?

```
String sql = "SELECT * FROM database WHERE name LIKE '% INPUT %';
```

now this command takes in name and cheackit but lets see what happnes when we add our code

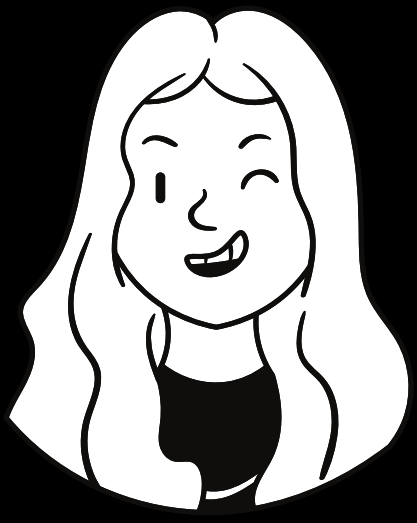
```
String sql = "SELECT * FROM database
WHERE name LIKE'%';-- %';
```

this `;` ends the statement that lets the user to enter another sql command.

FOR OTHER ATTACKS

REFFER TO RESEARCH DOCUMENTATION

Under sections Exercise



Raina Farooq

Isaac John Toppo

Abdeali Aliasgar Waseef