

Apa itu XSS: Percobaan sederhana Menunjukkan Penyerangan Cross-Site Scripting

Oleh: Reza Fadlillah Ardi

Universitas Pelita Bangsa

rezaardi0501@gmail.com

Pendahuluan

Apa itu XSS dan Mengapa Berbahaya?

Cross-Site Scripting (XSS) merupakan salah satu jenis masalah keamanan di web yang sering terjadi, yang memberi kesempatan kepada penyerang untuk menanamkan kode berbahaya (biasanya berupa JavaScript) ke halaman web yang diakses oleh orang lain. Serangan ini berlangsung ketika sebuah aplikasi web gagal untuk memeriksa atau menyaring input dari pengguna dengan baik, sehingga skrip berbahaya dapat berjalan di browser korban.

Dampak dari serangan XSS bisa sangat serius, antara lain:

- Pencurian Data: Informasi seperti cookie, token sesi, atau detail login pengguna bisa diambil oleh penyerang.
- Pengubahan Tampilan: Tampilan halaman web dapat dimodifikasi untuk menyebarkan informasi palsu atau perangkat lunak berbahaya.
- Pengalihan ke Situs Berbahaya: Korban bisa tanpa sadar dialihkan ke situs yang melakukan penipuan.

Berdasarkan OWASP (Open Web Application Security Project), XSS selalu masuk dalam daftar 10 ancaman utama bagi keamanan aplikasi web selama bertahun-tahun, menjadikannya risiko yang nyata yang harus dikenali oleh pengembang dan pengelola sistem.

Tujuan Artikel

Artikel ini memiliki tujuan untuk:

- Mempelajari pengertian dasar XSS serta kategorinya (Reflected, Stored, dan DOM-based XSS).
- Melakukan percobaan sederhana untuk menunjukkan cara kerja XSS dengan menggunakan formulir HTML dan PHP yang memiliki kerentanan.
- Mengevaluasi efek dari serangan serta cara menanggulanginya melalui tindakan pengamanan yang sesuai.

Target Pembaca Artikel

- Pengembang situs yang ingin mengetahui tentang kerentanan XSS serta metode untuk menghindarinya.
- Mahasiswa Teknologi Informasi atau Keamanan Siber yang sedang mempelajari hacking etis dan keselamatan aplikasi.
- Pengelola Sistem yang bertugas menjaga keamanan situs web perusahaan.

Pembahasan Utama

Konsep Dasar XSS

Cross-Site Scripting (XSS) merupakan salah satu jenis kerentanan di website yang paling sering ditemukan, yang memungkinkan penyerang untuk memasukkan kode JavaScript berbahaya ke dalam sebuah halaman web. Ketika seseorang membuka halaman itu, skrip jahat akan dijalankan di browser mereka.

Tipe-Tipe XSS:

1. Reflected XSS

- Skrip berbahaya dikirim melalui tautan (seperti dalam parameter `?search=<script>alert(1)</script>`).
- Sering kali digunakan dalam serangan penipuan.
- Contoh:

`http://target.com/search?query=<script>alert("XSS")</script>`

2. Stored XSS

- Skrip ini disimpan dalam basis data (misalnya di bagian komentar) dan akan dijalankan setiap kali halaman diakses.
- Lebih berbahaya karena dapat mempengaruhi banyak pengguna.

3. DOM-based XSS

- Terjadi karena adanya perubahan pada DOM yang dilakukan oleh JavaScript tanpa melibatkan server.
- Contoh:

```
document.write("<img src='x' onerror='alert(1)'>");
```

Eksperimen Sederhana: Reflected XSS

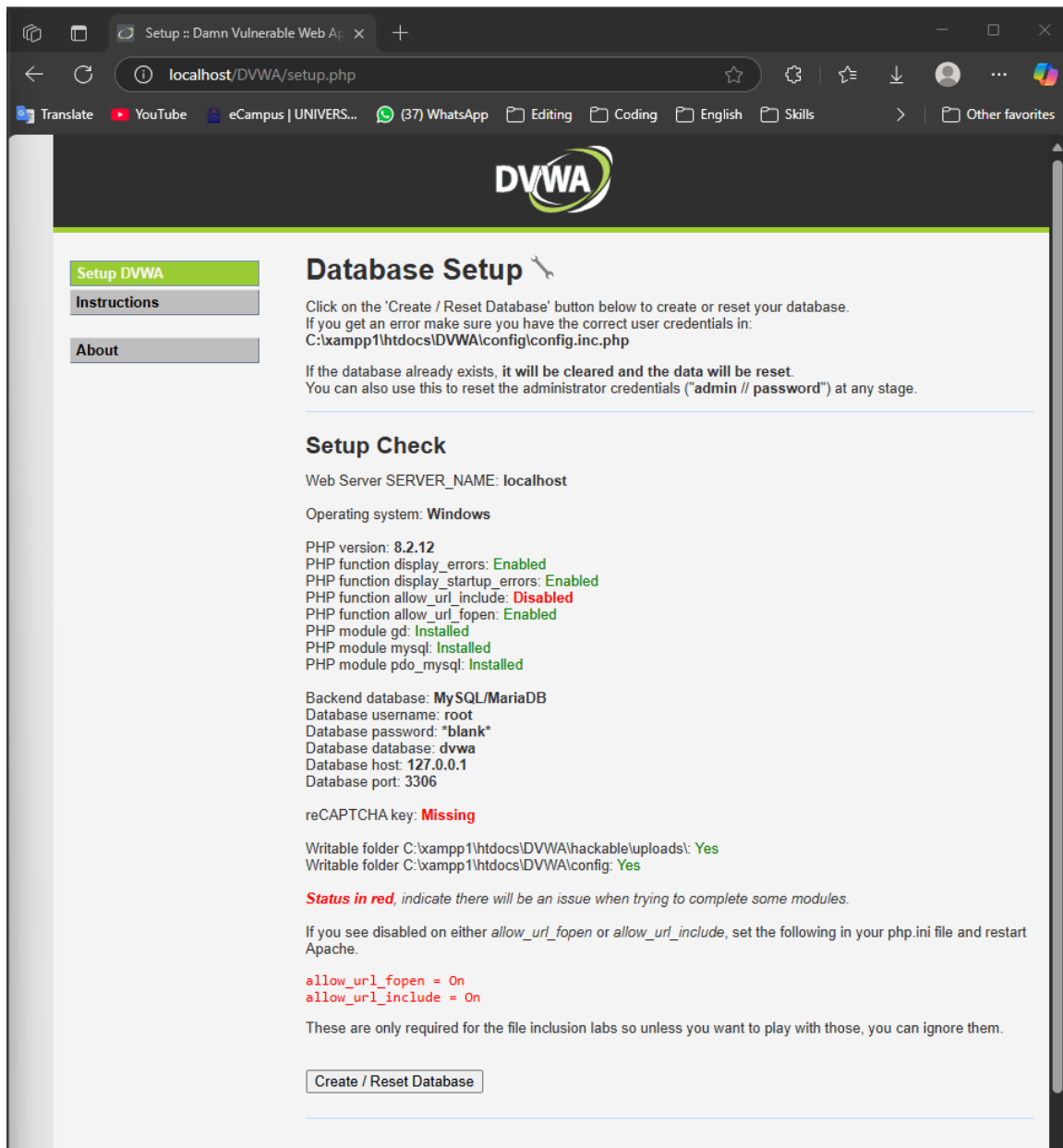
Tools Software Percobaan:

- Browser (Chrome/Firefox dengan DevTools)
- Server Lokal (XAMPP, Live Server Extension di VS Code)

Langkah Percobaan:

1. Buka XAMPP, dan run MySQL dan Apache server.
2. Download Damn Vulnerable Web Application(DVWA) dalam bentuk zip, extract, rename menjadi "DVWA", dan pindahkan ke folder C/xampp/htdocs.
3. Buka file *config*, rename file yang ada didalamnya menjadi *config.inc.php*. Lalu, buka file config dengan notepad dan ubah user menjadi 'root' dan password ''.
- 4.

5. Buka browser dan masukkan URL 'localhost/DVWA/setup.php', scroll ke bawah dan klik 'create/reset DB'



The screenshot shows a web browser window with the address bar displaying 'localhost/DVWA/setup.php'. The browser's address bar also shows 'Setup :: Damn Vulnerable Web App'. The page has a dark header with the DVWA logo. On the left, there is a sidebar with three links: 'Setup DVWA' (highlighted in green), 'Instructions', and 'About'. The main content area is titled 'Database Setup' with a wrench icon. It contains instructions on how to create or reset the database, including the path 'C:\xampp\htdocs\DVWA\config\config.inc.php'. Below this, there is a 'Setup Check' section that lists various system and PHP configurations. The 'reCAPTCHA key' is listed as 'Missing' in red. At the bottom, there is a button labeled 'Create / Reset Database'.

Setup DVWA

Database Setup

Click on the 'Create / Reset Database' button below to create or reset your database.
If you get an error make sure you have the correct user credentials in:
C:\xampp\htdocs\DVWA\config\config.inc.php

If the database already exists, it will be cleared and the data will be reset.
You can also use this to reset the administrator credentials ("admin // password") at any stage.

Setup Check

Web Server SERVER_NAME: localhost

Operating system: **Windows**

PHP version: **8.2.12**
PHP function display_errors: **Enabled**
PHP function display_startup_errors: **Enabled**
PHP function allow_url_include: **Disabled**
PHP function allow_url_fopen: **Enabled**
PHP module gd: **Installed**
PHP module mysql: **Installed**
PHP module pdo_mysql: **Installed**

Backend database: **MySQL/MariaDB**
Database username: **root**
Database password: ***blank***
Database database: **dvwa**
Database host: **127.0.0.1**
Database port: **3306**

reCAPTCHA key: **Missing**

Writable folder C:\xampp\htdocs\DVWA\hackable\uploads\: **Yes**
Writable folder C:\xampp\htdocs\DVWA\config: **Yes**

Status in red, indicate there will be an issue when trying to complete some modules.

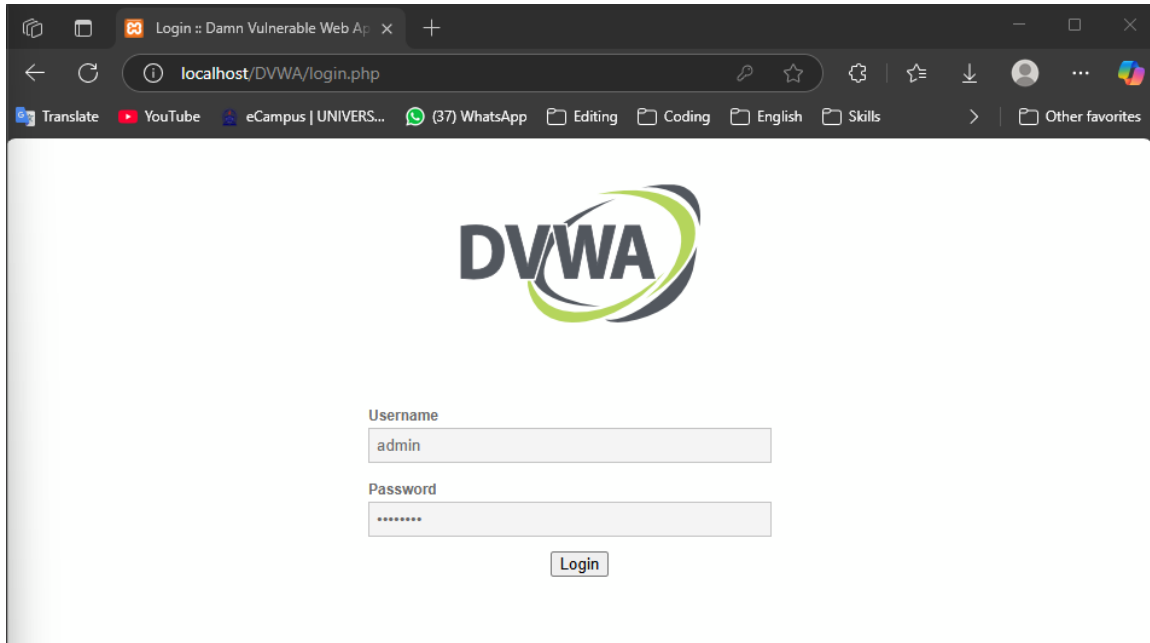
If you see disabled on either `allow_url_fopen` or `allow_url_include`, set the following in your php.ini file and restart Apache.

`allow_url_fopen = On`
`allow_url_include = On`

These are only required for the file inclusion labs so unless you want to play with those, you can ignore them.

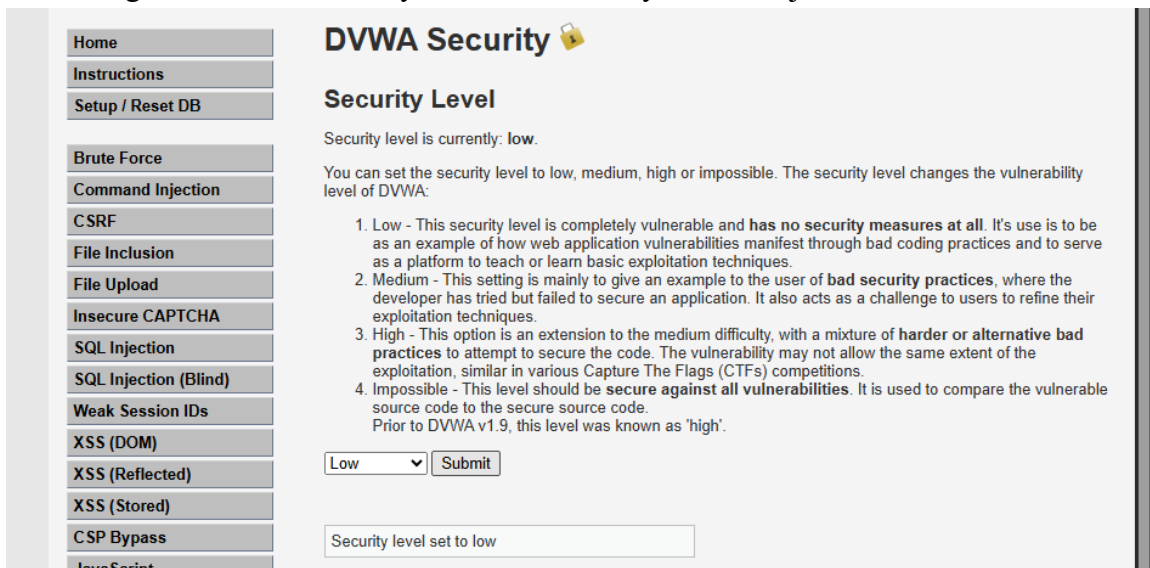
Create / Reset Database

6. Pergi ke 'localhost/DVWA/login.php', masuk dengan username 'admin' dan password 'password'.



A screenshot of a web browser showing the DVWA login page. The browser's address bar displays 'localhost/DVWA/login.php'. The page features the DVWA logo at the top center. Below the logo, there are two input fields: 'Username' with the text 'admin' and 'Password' with masked characters '*****'. A 'Login' button is positioned below the password field. The browser's tab is titled 'Login :: Damn Vulnerable Web App'.

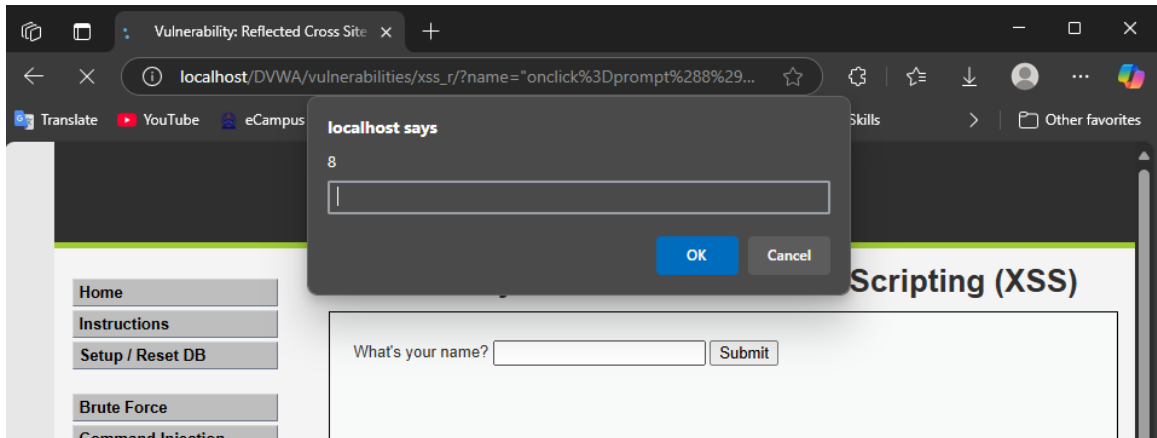
7. Pergi ke 'DVWA Security' dan ubah security level menjadi low



A screenshot of the 'DVWA Security' page. The left sidebar contains a menu with options: Home, Instructions, Setup / Reset DB, Brute Force, Command Injection, CSRF, File Inclusion, File Upload, Insecure CAPTCHA, SQL Injection, SQL Injection (Blind), Weak Session IDs, XSS (DOM), XSS (Reflected), XSS (Stored), CSP Bypass, and JavaScript. The main content area is titled 'DVWA Security' with a lock icon. Below the title, the 'Security Level' section states: 'Security level is currently: low.' It explains that the security level can be set to low, medium, high, or impossible, and that it changes the vulnerability level of DVWA. A list of four levels is provided: 1. Low (completely vulnerable), 2. Medium (bad security practices), 3. High (harder or alternative bad practices), and 4. Impossible (secure against all vulnerabilities). At the bottom, there is a dropdown menu set to 'Low' and a 'Submit' button. Below the button, a text box displays 'Security level set to low'.

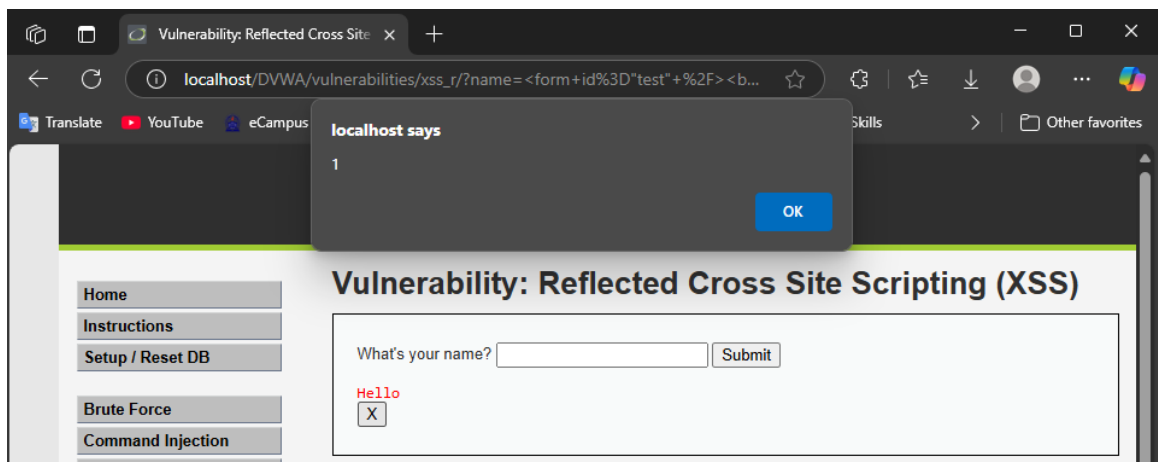
8. Jalankan 5 Cross Site Scripting Queries

1. “onclick=prompt(8)><svg/onload=prompt(8)>”@x.y

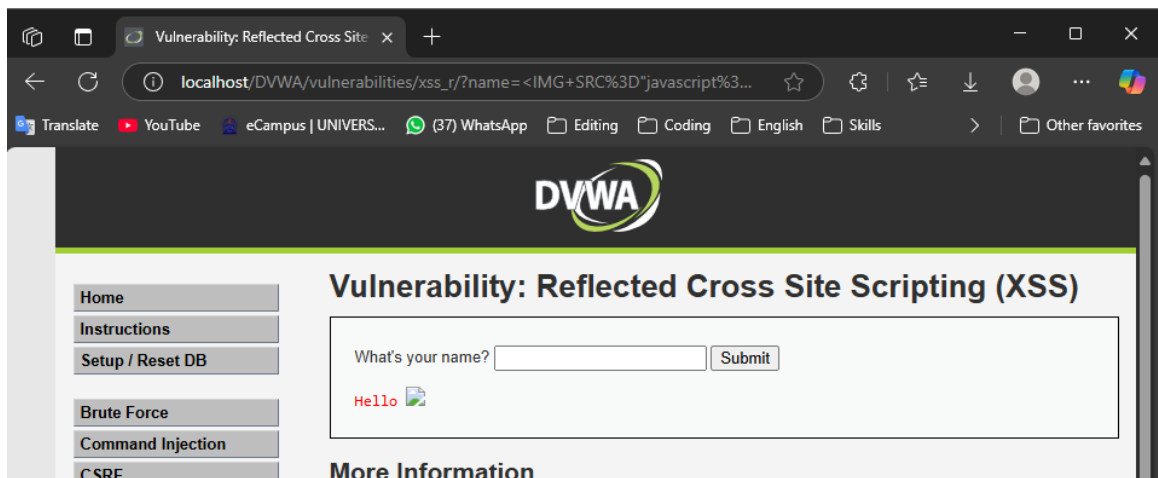


2. <form id="test" /><button

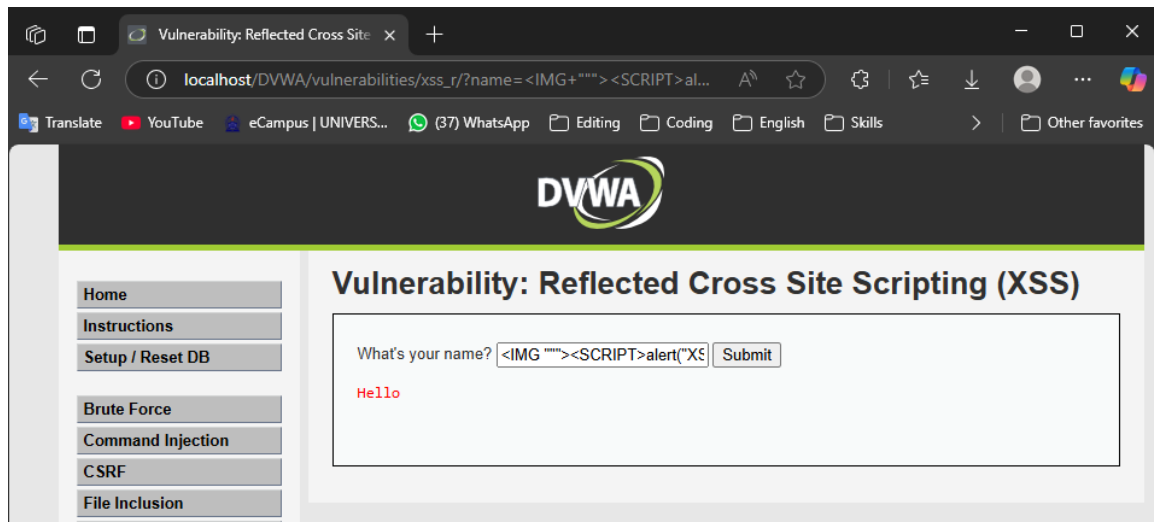
form="test"formaction="javascript:javascript:alert(1)">X



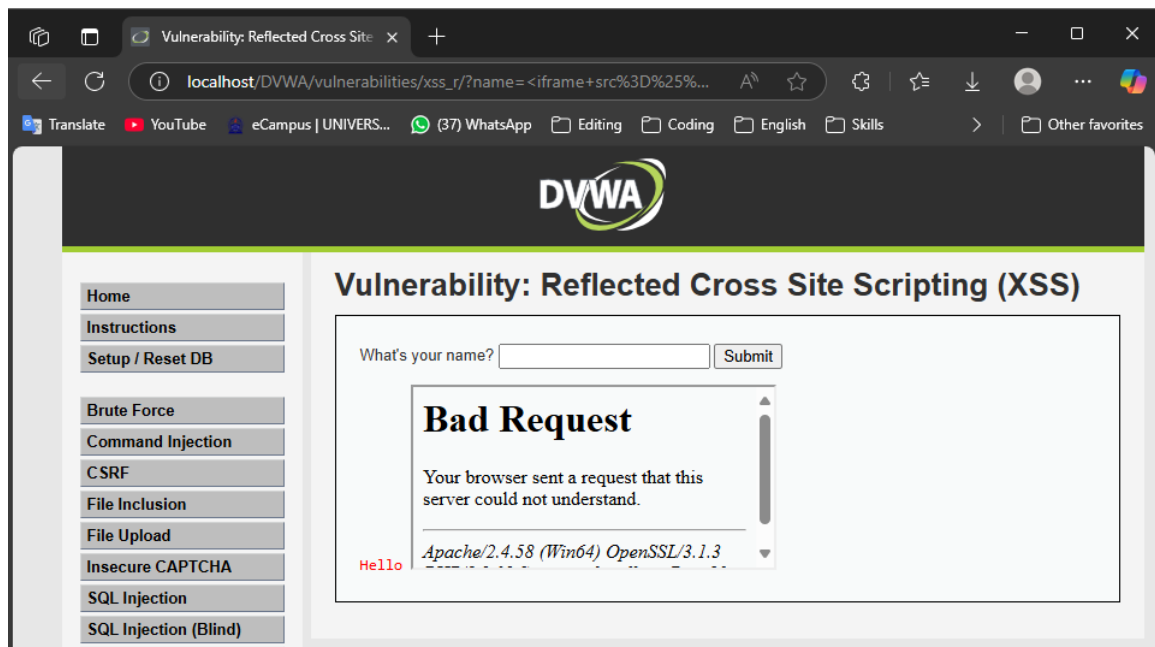
3.



4. `<SCRIPT>alert("XSS")</SCRIPT>`



5. `<iframe src=%(scriptlet)s<`



Kesimpulan

1. XSS merupakan bahaya serius yang dapat mengambil informasi, merubah tampilan situs web, atau mengarahkan pengguna ke halaman berbahaya.
2. Proyek ini menunjukkan risiko serangan XSS dan menyoroti pentingnya validasi dan sanitasi input yang tepat untuk mengamankan aplikasi web.

Dengan memahami konsep XSS, kita dapat menciptakan situs web yang lebih aman dan melindungi pengguna dari serangan yang berbahaya.

Referensi

OWASP. (2023). *"Cross-Site Scripting (XSS)"*. <https://owasp.org/www-community/attacks/xss/>

PortSwigger. (2023). *"What is XSS?"*. <https://portswigger.net/web-security/cross-site-scripting>

GitHub. (2024). *"XSS Attack Simulation"*. [100rabhhh/XSS-Attack-Simulation](https://github.com/100rabhhh/XSS-Attack-Simulation)