**PENGEMBANGAN WEB (TEORI)**

**LAPORAN EKSPERIMEN MENGENAI PROBLEM SERANGAN XSS DENGAN PENYELESAIAN MENGGUNAKAN SOLUSI ESCAPE OUTPUT**

*Laporan ini disusun untuk memenuhi tugas 1 mata kuliah Pengembangan Web (Teori)*



Disusun oleh kelompok B4:

**Asri Husnul Rosadi 221524035**

Faris Abulkhoir 221524040

Mahardika Pratama 221524044

Muhamad Fahri Yuwan 221524047

Najib Alimudin Fajri 221524053

Septyana Agustina 221524058

Sarah 221524059

Dosen Pengampu:

Joe Lian Min, M.Eng.

**JURUSAN TEKNIK KOMPUTER DAN INFORMATIKA**

**PROGRAM STUDI D4 TEKNIK INFORMATIKA**

**POLITEKNIK NEGERI BANDUNG**

**2024**

# **DAFTAR ISI**

[DAFTAR ISI i](#_Toc175499445)

[A. IDENTIFIKASI PROBLEM 1](#_Toc175499446)

[B. DESKRIPSI PROBLEM 1](#_Toc175499447)

[C. METODOLOGI EKSPERIMEN 1](#_Toc175499448)

[D. PELAKSANAAN EKSPERIMEN 2](#_Toc175499449)

[E. ANALISIS HASIL EKSPERIMEN 3](#_Toc175499450)

[F. KESIMPULAN 4](#_Toc175499451)

# **IDENTIFIKASI PROBLEM**

Serangan XSS adalah jenis kerentanan keamanan web di mana penyerang dapat menyisipkan dan mengeksekusi skrip berbahaya pada halaman web. Masalah ini sering terjadi ketika aplikasi web menampilkan data yang diterima dari pengguna tanpa melakukan escape atau encoding yang tepat. Escape output adalah teknik untuk mencegah XSS dengan mengganti karakter-karakter khusus dalam data yang akan ditampilkan dengan representasi yang aman.

# **DESKRIPSI PROBLEM**

Dalam eksperimen ini, terdapat dua halaman web:

* vulnerable.html: Halaman web yang tidak melakukan escape terhadap input pengguna, sehingga memungkinkan eksekusi skrip berbahaya.
* secure.html: Halaman web yang menggunakan metode escape output untuk mengamankan input pengguna sebelum menampilkannya, mencegah eksekusi skrip berbahaya.

Eksperimen ini bertujuan untuk membandingkan bagaimana kedua halaman menangani input yang mengandung kode JavaScript berbahaya, khususnya <script>alert('XSS!')</script>, yang dimasukkan melalui formulir input.

# **METODOLOGI EKSPERIMEN**

Langkah-langkah Eksperimen:

1. Persiapan Halaman Web:

* Buat dua halaman HTML: vulnerable.html dan secure.html.
* vulnerable.html akan menampilkan input langsung tanpa escape.
* secure.html akan menampilkan input setelah melakukan escape untuk mencegah eksekusi kode berbahaya.

1. Implementasi Kode:

* Pada vulnerable.html, tampilkan input pengguna langsung ke halaman tanpa escape.
* Pada secure.html, implementasikan fungsi escape untuk mengkonversi karakter-karakter khusus menjadi representasi aman sebelum menampilkan input.

1. Pengujian Kerentanan XSS:

* Uji kedua halaman dengan memasukkan input yang mengandung skrip berbahaya melalui formulir.
* Observasi apakah skrip berbahaya dijalankan pada halaman yang rentan dan dicegah pada halaman yang menggunakan escape output.

# **PELAKSANAAN EKSPERIMEN**

1. Implementasi dan Pengujian vulnerable.html:
2. Kode Implementasi:

|  |
| --- |
|  |

1. Pengujian

* Masukkan <script>alert('XSS!')</script> sebagai nilai parameter URL (misalnya ?name=<script>alert('XSS!')</script>).
* Klik tombol "Submit."
* Hasil: Pop-up alert "XSS!" muncul, menunjukkan bahwa skrip berbahaya berhasil dijalankan.

1. Implementasi dan Pengujian secure.html:
2. Kode Implementasi:

|  |
| --- |
|  |

1. Pengujian

* Masukkan <script>alert('XSS!')</script> sebagai nilai parameter URL (misalnya ?name=<script>alert('XSS!')</script>).
* Klik tombol "Submit."
* Hasil: Tidak ada pop-up alert yang muncul, dan skrip berbahaya ditampilkan sebagai teks biasa, menunjukkan bahwa metode escape berhasil mencegah eksekusi XSS.

# **ANALISIS HASIL EKSPERIMEN**

1. Hasil pada vulnerable.html:

Input berbahaya <script>alert('XSS!')</script> dieksekusi, menunjukkan bahwa halaman ini rentan terhadap serangan XSS karena tidak ada escape output yang diterapkan.

1. Hasil pada secure.html:

Input yang sama ditampilkan sebagai teks biasa tanpa eksekusi, menunjukkan bahwa metode escape output berhasil mencegah eksekusi skrip berbahaya.

1. Kelebihan dan Kekurangan:

Kelebihan menggunakan Escape Output:

* Mencegah eksekusi skrip berbahaya dengan mengkonversi karakter-karakter khusus menjadi representasi aman.
* Metode sederhana dan efektif untuk menghindari XSS pada data yang ditampilkan.

Kekurangan menggunakan Escape Output:

* Memerlukan pemahaman yang baik tentang karakter-karakter khusus dan bagaimana mereka harus di-escape.
* Perlu dipastikan bahwa escape dilakukan di semua tempat yang tepat di seluruh aplikasi.

# **KESIMPULAN**

Eksperimen ini menunjukkan bahwa tanpa escape output, aplikasi web rentan terhadap serangan XSS. Dengan menerapkan teknik escape output, input yang berpotensi berbahaya dapat ditampilkan dengan aman tanpa mengeksekusi skrip berbahaya. Teknik ini merupakan solusi efektif untuk melindungi aplikasi web dari kerentanan XSS.