**PENGEMBANGAN WEB (TEORI)**

**LAPORAN EKSPERIMEN CLICKJACKING PREVENTION: X-FRAME-OPTIONS VS CONTENT SECURITY POLICY (CSP)**

*Laporan ini disusun untuk memenuhi tugas 4 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

Sarah 221524059

Septyana Agustina 221524060

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](#_Toc177227684)

[A. IDENTIFIKASI PROBLEM 1](#_Toc177227685)

[B. DESKRIPSI PROBLEM 1](#_Toc177227686)

[C. METODOLOGI EKSPERIMEN 1](#_Toc177227687)

[D. PELAKSANAAN EKSPERIMEN 2](#_Toc177227688)

[E. ANALISIS HASIL EKSPERIMEN 3](#_Toc177227689)

# **IDENTIFIKASI PROBLEM**

Clickjacking adalah teknik serangan di mana penyerang menempatkan elemen UI transparan atau tersembunyi di atas halaman lain untuk mengelabui pengguna agar mengklik elemen yang tidak mereka niatkan. Untuk melindungi aplikasi web dari serangan clickjacking, dua header HTTP dapat digunakan: X-Frame-Options dan Content-Security-Policy (CSP). Tujuan eksperimen ini adalah untuk membandingkan efektivitas kedua metode tersebut dalam mencegah halaman web dimuat dalam iframe dari domain yang sama maupun berbeda.

# **DESKRIPSI PROBLEM**

Eksperimen ini mengevaluasi efektivitas dua metode pencegahan clickjacking:

1. X-Frame-Options Header:

DENY: Mencegah halaman dimuat dalam iframe dari semua sumber.

SAMEORIGIN: Hanya mengizinkan halaman dari domain yang sama untuk memuat halaman ini dalam iframe.

1. Content-Security-Policy (CSP) Header:

frame-ancestors 'none': Mencegah halaman dimuat dalam iframe dari semua sumber.

frame-ancestors 'self': Hanya mengizinkan halaman dari domain yang sama untuk memuat halaman ini dalam iframe.

# **METODOLOGI EKSPERIMEN**

**Variabel Terikat:** Efektivitas pencegahan clickjacking berdasarkan kemampuan mencegah halaman dimuat dalam iframe.

**Variabel Bebas:** Header HTTP yang digunakan (X-Frame-Options dan Content-Security-Policy).

**Metode Pengukuran:** Pengujian dilakukan dengan mencoba memuat halaman target (target.html) dalam iframe dari domain yang berbeda atau sama.

**Skenario Pengujian:**

**Pengujian dengan Header X-Frame-Options dan Content-Security-Policy:** Mengatur header dan mencoba memuat halaman target dalam iframe dari domain yang berbeda dan domain yang sama.

**Alat dan Teknologi:**

* + Server Express.js untuk menyajikan halaman dan mengatur header X-Frame-Options serta Content-Security-Policy.
  + HTML: Halaman index.html dan target.html sebagai halaman yang diuji untuk clickjacking.
  + Browser dengan Developer Tools untuk memeriksa header HTTP yang diterima dan mencoba memuat iframe, memastikan apakah halaman dilindungi dari clickjacking sesuai konfigurasi yang diterapkan.

**Langkah-langkah Eksperimen:**

1. **Setup Lingkungan:**

* Implementasi server Express.js untuk mengatur header HTTP yang diperlukan.
* Menyediakan halaman index.html (halaman utama) dan target.html (halaman target) di server.

1. **Implementasi Header:** Tambahkan header X-Frame-Options dan Content-Security-Policy pada respons dari server.
2. **Pengujian Pemuatan Iframe:** Coba memuat target.html dalam iframe dari domain yang sama dan berbeda untuk melihat apakah header mencegah pemuatan.

# **PELAKSANAAN EKSPERIMEN**

**Tahap 1: Setup Lingkungan**

* 1. **Setup Aplikasi:** Express.js diatur untuk menyajikan index.html dan target.html.
  2. **Implementasi Header:** Header X-Frame-Options dan Content-Security-Policy diatur untuk mengontrol pemuatan iframe.

**Tahap 2: Pengujian**

1. **Pengujian dengan X-Frame-Options dan Content-Security-Policy:**

Kirim permintaan menggunakan Postman ke tiga endpoint berbeda:

* 1. Memastikan halaman target tidak dapat dimuat dalam iframe dengan header yang diatur.
  2. Coba memuat halaman target dari domain yang berbeda untuk memverifikasi header.

# **ANALISIS HASIL EKSPERIMEN**

**Hasil Eksperimen:**

* 1. **X-Frame-Options: DENY:**
* Halaman target tidak dapat dimuat dalam iframe dari domain manapun. Ini berhasil mencegah clickjacking dari semua sumber.
  1. **Content-Security-Policy: frame-ancestors 'none':**
* Hasil yang sama dengan X-Frame-Options: DENY. Halaman target tidak dapat dimuat dalam iframe dari domain manapun.
  1. **X-Frame-Options: SAMEORIGIN:**
  + Halaman target dapat dimuat dalam iframe hanya jika domain iframe adalah domain yang sama. Efektif untuk mencegah clickjacking dari domain lain.
  1. **Content-Security-Policy: frame-ancestors 'self':**
  + Halaman target dapat dimuat dalam iframe hanya jika domain iframe adalah domain yang sama. Sama efektifnya dengan X-Frame-Options: SAMEORIGIN.

**Analisis Keseluruhan**

* + Kedua metode (X-Frame-Options dan Content-Security-Policy) efektif dalam mencegah clickjacking jika dikonfigurasi dengan benar.
  + X-Frame-Options: DENY dan Content-Security-Policy: frame-ancestors 'none' lebih ketat karena mencegah iframe dari semua sumber.
  + X-Frame-Options: SAMEORIGIN dan Content-Security-Policy: frame-ancestors 'self' memungkinkan pemuatan iframe dari domain yang sama, yang berguna dalam skenario di mana aplikasi perlu diizinkan memuat konten dari domain yang sama.

**Pros dan Cons**

**Pros:**

* + **X-Frame-Options**: Sederhana dan mudah diimplementasikan.
  + **Content-Security-Policy**: Lebih fleksibel dan menawarkan kontrol yang lebih granuler.

**Cons:**

* **X-Frame-Options**: Tidak mendukung kontrol berbasis domain yang kompleks.
* **Content-Security-Policy**: Konfigurasi lebih rumit dan mungkin memerlukan penyesuaian lebih banyak.

**Kesimpulan**

**X-Frame-Options** dan **Content-Security-Policy** keduanya efektif untuk melindungi terhadap clickjacking, dengan DENY dan frame-ancestors 'none' memberikan perlindungan yang lebih ketat dibandingkan SAMEORIGIN dan frame-ancestors 'self'. Pilihan antara keduanya tergantung pada kebutuhan spesifik aplikasi dan kebijakan keamanan yang diinginkan.