**PENGEMBANGAN WEB (TEORI)**

**LAPORAN EKSPERIMEN MENGENAI PROBLEM SERANGAN SESSION FIXATION DENGAN PENYELESAIAN MENGGUNAKAN SOLUSI SECURE COOKIE**

*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

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](#_Toc175601025)

[**A.** **IDENTIFIKASI PROBLEM** 1](#_Toc175601026)

[**B.** **DESKRIPSI PROBLEM** 1](#_Toc175601027)

[**C.** **METODOLOGI EKSPERIMEN** 1](#_Toc175601028)

[**D.** **PELAKSANAAN EKSPERIMEN** 3](#_Toc175601029)

[**E.** **ANALISIS HASIL EKSPERIMEN** 4](#_Toc175601030)

# **IDENTIFIKASI PROBLEM**

Dalam pengembangan aplikasi web, sesi pengguna yang tidak dilindungi dengan baik dapat menjadi target serangan session fixation. Serangan ini terjadi ketika penyerang memaksa pengguna untuk menggunakan ID sesi tertentu yang sudah diketahui atau ditetapkan oleh penyerang sebelumnya. Jika ID sesi yang sama digunakan oleh pengguna dan diterima oleh server, penyerang dapat mengambil alih sesi pengguna dan mendapatkan akses tidak sah ke akun pengguna.

# **DESKRIPSI PROBLEM**

Cookie sesi sering kali digunakan untuk menyimpan ID sesi yang digunakan oleh server untuk mengidentifikasi pengguna. Jika cookie sesi tidak dilindungi dengan atribut keamanan yang memadai, seperti HttpOnly, Secure, dan SameSite, maka penyerang dapat dengan mudah memanipulasi atau mencuri cookie tersebut. Tanpa perlindungan yang memadai, aplikasi web rentan terhadap serangan *session fixation* yang dapat mengakibatkan pencurian akun atau data pengguna.

# **METODOLOGI EKSPERIMEN**

1. **Desain Eksperimen**

Eksperimen ini berfokus pada penerapan atribut keamanan pada cookie sesi untuk mencegah serangan *session fixation*. Pengujian dilakukan dengan mengamati perilaku aplikasi web sebelum dan sesudah penerapan atribut keamanan pada cookie.

1. **Variabel Eksperimen**

* Variabel Bebas (Independent Variable):
* Penerapan atribut keamanan pada cookie (HttpOnly, Secure, dan SameSite).
* Variabel Terikat (Dependent Variable):
* Kemampuan penyerang untuk mengambil alih sesi pengguna melalui serangan *session fixation*.
* Variabel Terkendali (Controlled Variables):
* Lingkungan pengujian: Server aplikasi web dijalankan dengan konfigurasi yang sama di kedua kondisi, kecuali pada pengaturan atribut cookie.
* Atribut cookie lainnya dipertahankan sama untuk memastikan bahwa perbedaan yang diamati hanya disebabkan oleh penerapan atribut keamanan.

1. **Prosedur Eksperimen**
2. Tahap Persiapan:

* Menyiapkan server aplikasi web menggunakan Express.js dengan sesi yang dikelola oleh middleware express-session.
* Membuat dua versi server: satu tanpa atribut keamanan pada cookie sesi, dan satu dengan atribut HttpOnly, Secure, dan SameSite diaktifkan.

1. Pengujian Tanpa Atribut Keamanan:

* Akses endpoint /login untuk membuat sesi pengguna tanpa atribut keamanan pada cookie.
* Coba manipulasi sesi dengan menambahkan cookie sesi palsu melalui *developer tools*.
* Amati apakah aplikasi mengenali ID sesi yang telah dimanipulasi.

1. Penerapan Atribut Keamanan:

* Modifikasi server untuk menambahkan atribut HttpOnly, Secure, dan SameSite pada cookie sesi.
* Ulangi langkah yang sama dengan mengakses endpoint /login dan coba manipulasi sesi.
* Amati apakah aplikasi menolak ID sesi yang telah dimanipulasi.

1. **Alat dan Bahan**
2. Alat:

* Server Aplikasi: Server Express.js.
* Browser atau Alat Uji API: Untuk mengakses endpoint aplikasi dan memverifikasi status sesi.

1. Bahan:

* Kode Sumber: Kode JavaScript untuk aplikasi web dengan dan tanpa penerapan atribut keamanan pada cookie.

1. **Pengukuran dan Pengamatan**
2. Pengukuran:

* Apakah sesi tetap dapat diambil alih oleh penyerang setelah manipulasi ID sesi.

1. Pengamatan:

* Perbedaan dalam respons server sebelum dan sesudah penerapan atribut keamanan pada cookie.

1. **Analisi Data**

Data yang dikumpulkan akan dianalisis untuk mengevaluasi efektivitas penerapan atribut keamanan pada cookie dalam mencegah serangan *session fixation*. Analisis akan mencakup perbandingan kemampuan penyerang dalam mengambil alih sesi sebelum dan sesudah penerapan atribut keamanan.

# **PELAKSANAAN EKSPERIMEN**

1. **Tahap Persiapan**

Eksperimen dilakukan dalam dua tahap utama: pengujian sebelum dan sesudah penerapan atribut keamanan pada cookie sesi. Server aplikasi dijalankan pada port yang sama (port 3000) untuk kedua tahap pengujian, dengan perbedaan hanya pada konfigurasi cookie.

1. **Pengujian Tanpa Expiration Time**
2. Menjalankan Server:

* Server aplikasi dijalankan menggunakan kode tanpa atribut keamanan pada cookie sesi.

node server-before.js

1. Login Pengguna

* Akses endpoint /login untuk membuat sesi pengguna tanpa atribut keamanan pada cookie..
* Coba manipulasi ID sesi melalui *developer tools* dan periksa apakah aplikasi mengenali ID sesi palsu

1. **Pengujian Setelah Penerapan Atribut Keamanan**
2. Menjalankan Server dengan Atribut Keamanan:

* Server aplikasi dimodifikasi untuk menambahkan atribut HttpOnly, Secure, dan SameSite pada cookie sesi, lalu dijalankan.

node server-after.js

1. Login Pengguna

* Akses endpoint /login untuk membuat sesi pengguna dengan cookie yang aman.
* Coba manipulasi ID sesi dengan cara yang sama dan periksa apakah aplikasi menolak ID sesi palsu.

1. **Dokumentasi dan Pengumpulan Data**

Selama pelaksanaan eksperimen, hasil-hasil pengujian dicatat, termasuk respons server terhadap manipulasi ID sesi sebelum dan sesudah penerapan atribut keamanan.

# **ANALISIS HASIL EKSPERIMEN**

1. **Hasil Pengamatan**

* Pengujian Tanpa Atribut Keamanan:
* Sesi dapat diambil alih oleh penyerang setelah manipulasi ID sesi melalui *developer tools*.
* Pengujian Setelah Penerapan Expiration Time:
* Aplikasi menolak ID sesi palsu, dan penyerang tidak dapat mengambil alih sesi pengguna.

1. **Analisis Pros dan Cons**

**Pros:**

1. Keamanan Meningkat:

* Penerapan atribut HttpOnly, Secure, dan SameSite pada cookie berhasil mencegah serangan *session fixation*, meningkatkan keamanan aplikasi web..

1. Perlindungan Sesi yang Lebih Baik:

* Dengan atribut keamanan yang tepat, sesi pengguna lebih terlindungi dari manipulasi eksternal.

**Cons:**

1. Kompatibilitas dan Implementasi:

* Beberapa browser lama mungkin tidak mendukung SameSite, sehingga memerlukan perhatian tambahan dalam implementasi.

1. **Kesimpulan**

Eksperimen ini menunjukkan bahwa penerapan atribut keamanan pada cookie sesi secara efektif mencegah serangan *session fixation*, meningkatkan keamanan aplikasi web dengan mengurangi risiko pengambilalihan sesi oleh pihak yang tidak sah. Meskipun ada beberapa tantangan terkait kompatibilitas, manfaat keamanannya jauh lebih besar, terutama dalam aplikasi yang menangani informasi sensitif atau digunakan di lingkungan yang berisiko tinggi.