

Laporan Vulnerabilities Keamanan

Kelompok:

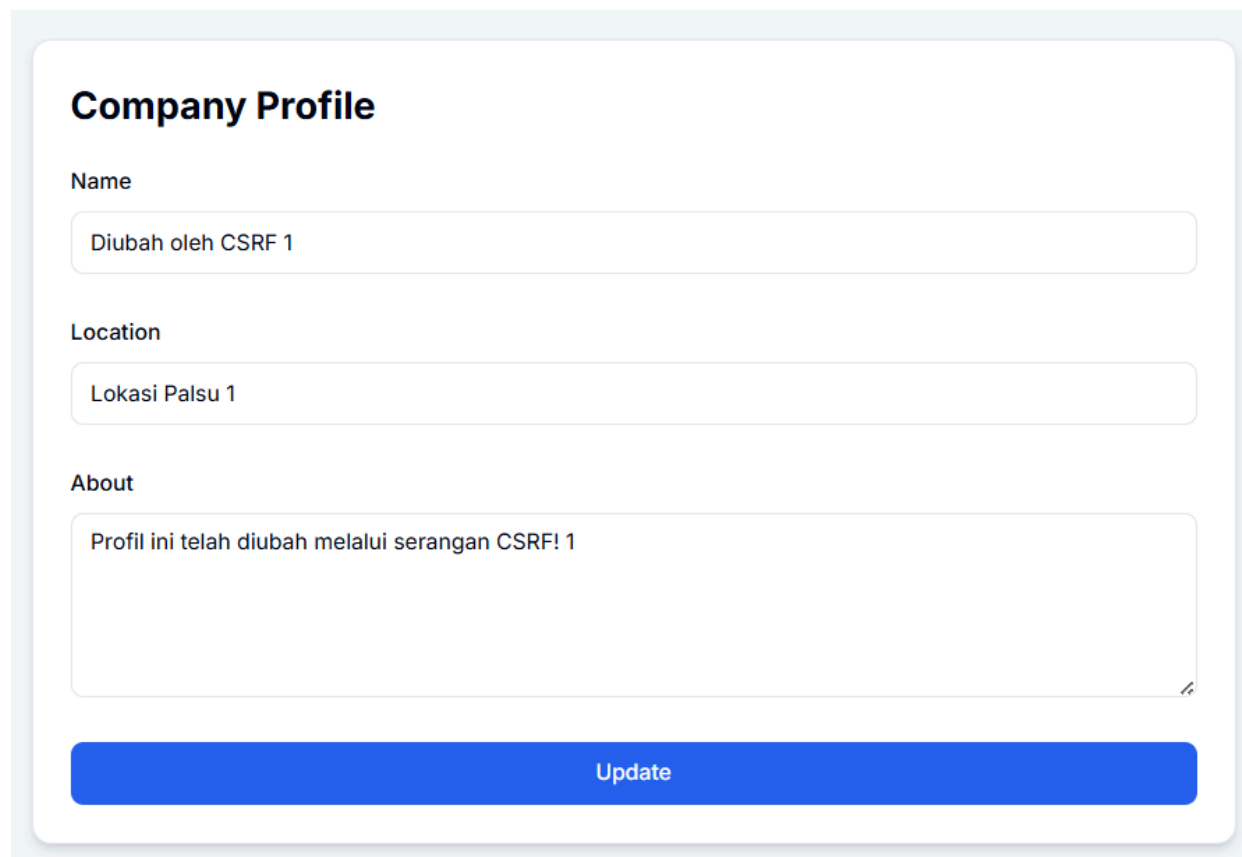
1. Farhan Raditya Aji (13522142)
2. Axel Santadi Warih (13522155)
3. Muhammad Dzaki Arta (13522149)

Berikut adalah hasil dari analisis Vulnerabilitas yang kami temukan:

1. Kerentanan Cross-Site Request Forgery (CSRF)

Aplikasi ini tidak mengimplementasikan token CSRF untuk pengiriman formulir, sehingga rentan dengan serangan csrf, seperti ilustrasi berikut ini:

- a. Company Berhasil Login ke LinkInPurry di port 8080



Company Profile

Name

Diubah oleh CSRF 1

Location

Lokasi Palsu 1

About

Profil ini telah diubah melalui serangan CSRF! 1

Update

- b. Ada Attacker yang membuat website dengan membuat form POST edit profile ke website LinkInPurry dengan isi data yang ingin diubah olehnya. Seperti contoh web Attacker dibawah ini:

Halaman untuk attack CSRF

Attack CSRF

- c. Ketika pengguna yang sudah login mengakses website attacker tersebut, browser secara otomatis menyertakan cookie sesi pengguna saat form tersubmit
- d. Akibatnya, permintaan perubahan data di server LinkInPurry dianggap valid karena memiliki cookie sesi yang aktif
- e. Profil company berhasil diubah oleh attacker tanpa sepengetahuan pengguna

Company Profile

Name

Diubah oleh CSRF 1

Location

Lokasi Palsu 1

About

Profil ini telah diubah melalui serangan CSRF! 1

Update

Maka Solusi Pencegahan yang Diimplementasikan:

1. **Token CSRF Unik**
 - a. Setiap form yang mengubah data (POST, PUT, DELETE) menyertakan token acak unik.

- b. Token dihasilkan oleh server, disimpan di sesi pengguna, dan disisipkan sebagai input tersembunyi di form (contoh: `<input type="hidden" name="csrf_token" value="RANDOM_TOKEN">`).
- c. Server memverifikasi token pada setiap permintaan untuk memastikan permintaan berasal dari aplikasi asli.

2. Validasi di Berbagai Level

- a. **Form**, token ditambahkan otomatis ke semua form sensitif (misalnya, form status aplikasi).
- b. **Controller**, permintaan POST divalidasi menggunakan `CSRFHandler::verifyToken()` untuk memastikan token cocok dengan sesi.
- c. **Router**, verifikasi token diterapkan secara terpusat untuk semua permintaan non-GET, mencegah kelalaian pengembang.

3. Dukungan AJAX dan Form

- a. Untuk form tradisional, token dikirim sebagai field tersembunyi.
- b. Untuk permintaan AJAX, token dikirim melalui header X-CSRF-TOKEN, memastikan perlindungan konsisten.

4. Keamanan Tambahan

- a. **Regenerasi Token**, token diperbarui setelah digunakan untuk mencegah penggunaan ulang (replay attack).
- b. **SameSite Cookie**, cookie sesi diatur dengan `SameSite=Strict` untuk mencegah pengiriman lintas situs.
- c. **Penanganan Error**, permintaan dengan token tidak valid ditolak dengan pesan error yang jelas (HTTP 400).

5. Berikut link commit perbaikan:

- a. Link : [fix\(docker\): error dockerfile configuration · sibobbbbbbb/if3110-tubes-2024-k01-08@c569388](https://github.com/sibobbbbbbb/if3110-tubes-2024-k01-08/@c569388)

Maka dengan solusi tersebut kerentanan CSRF ini dapat diatasi.

2. Tidak Ada Pembatasan Rate (Rate Limiting)

a. Executive Summary

- i. Kategori (OWASP A01:2021): Broken Access Control / Authentication
- ii. CWE-ID: 307 – Improper Restriction of Excessive Authentication Attempts
- iii. Risiko: Rendah–Sedang (bergantung pada kekuatan password user)

b. Deskripsi Vulnerability

Pada implementasi awal, endpoint `POST /auth/sign-in` tidak membatasi jumlah percobaan login. Ini membuka peluang serangan brute-force di mana penyerang dapat mencoba kombinasi email/password berulang kali tanpa hambatan.

c. Komponen Terdampak

- i. Route: POST /auth/sign-in
- ii. Controller: AuthController::renderAndHandleSignIn()
- iii. Belum ada pembatasan pada level aplikasi atau infrastruktur

d. Dampak

- i. Akses tidak sah: Penyerang dapat menebak password sampai berhasil.
- ii. Account takeover: Jika password lemah, akun user dapat diretas.
- iii. Enum user: Melalui perbedaan respons (valid vs invalid), akun email yang terdaftar bisa di-enumerate.

e. Langkah Reproduksi

- i. Kirim permintaan POST /auth/sign-in berulang kali dengan kombinasi email/password acak.
- ii. Anda akan mendapatkan response validasi (“email/password salah”) tanpa batas.
- iii. Coba ribuan permintaan otomatis; tidak ada penghalang.

f. Remediasi

- i. Implementasikan rate-limiting di level aplikasi dengan menggunakan middleware (RateLimitMiddleware) yang memblokir client (by IP + route) setelah N = 5 percobaan salah dalam periode 60s
- ii. Menampilkan flash message di UI agar user tahu berapa detik harus menunggu

g. Link ke commit

[feat\(auth\): add exponential backoff rate limiting to sign-in · sibobbbbbbb/if3110-tubes-2024-k01-08@9e0268d](https://github.com/sibobbbbbbb/if3110-tubes-2024-k01-08@9e0268d/feat(auth):_add_exponential_backoff_rate_limiting_to_sign-in_)

3. Penyimpanan Password yang Tidak Aman (Weak Password Complexity Enforcement)

a. Executive Summary

Walaupun aplikasi telah menggunakan password_hash() untuk menyimpan password, tidak ada mekanisme validasi untuk memastikan kompleksitas password. Akibatnya, user dapat memilih password yang sederhana—misalnya hanya terdiri dari 3-8 huruf alfabet—yang rentan terhadap serangan brute-force.

b. Kategori (OWASP A01:2021):

Broken Authentication / Broken Access Control

c. CWE-ID:

CWE-521 – Weak Password Requirements

d. Risiko:

Rendah–Sedang (bergantung pada pilihan password user). Jika user memilih password lemah, risiko kebocoran akun meningkat karena serangan brute-force bisa dilakukan dengan lebih mudah.

e. Deskripsi Vulnerability

Pada implementasi awal, meskipun fungsi `password_hash()` telah digunakan:

```
php
```

```
$hashed_password = password_hash($password, PASSWORD_DEFAULT);
```

tidak ada persyaratan standar minimal untuk kompleksitas password. Akibatnya, user dapat menggunakan password yang sangat sederhana dan mudah ditebak (misalnya 3-8 huruf alfabet), sehingga meningkatkan peluang brute-force attack.

f. Komponen Terdampak

- Form sign-up (baik untuk job seeker maupun company)
- Controller: AuthController
- Service: AuthService (method `createJobSeeker()` dan `createCompany()`)

g. Dampak

- Akses Tidak Sah: Password lemah memungkinkan penyerang melakukan serangan brute-force untuk mendapatkan akses ke akun user.
- Account Takeover: Jika password yang dipilih user mudah ditebak, risiko takeover akun menjadi signifikan.
- User Enumeration: Perbedaan respons dalam validasi password dapat digunakan untuk mengidentifikasi akun yang telah terdaftar.

h. Langkah Reproduksi

- a. Buka halaman sign-up pada aplikasi (misalnya untuk job seeker atau company).
- b. Isi form pendaftaran dengan data yang valid, tetapi gunakan password yang sederhana (misalnya "abcde").

- c. Submit form pendaftaran.
- d. Meskipun password tidak memenuhi kriteria kekuatan yang direkomendasikan, sistem akan tetap menerima password tersebut dan menyimpannya menggunakan `password_hash()`.

i. Link commit perbaikan

[feat\(auth\): improve password validation error handling on sign-up pages · sibobbbbbbb/if3110-tubes-2024-k01-08@37f9089](#)