Laporan Tugas
IF4053 - Keamanan Perangkat Lunak

Disusun oleh:

Manuella Ivana Uli Sianipar	13521051
Nyoman Ganadipa Narayna	13522066
Dhafin Fawwaz Ikramullah	13522084

PROGRAM STUDI SARJANA TEKNIK INFORMATIKA

SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA

INSTITUT TEKNOLOGI BANDUNG

2025

I. Deskripsi Proyek

LinkinPurry, sebuah platform lowongan kerja untuk agen O.W.C.A., dibangun menggunakan PHP murni, JavaScript vanilla, dan CSS. Fitur utamanya mencakup posting pekerjaan dengan rich text editor (Quill.js) dan unggah CV/video.

Mengingat *tech stacks* yang fundamental (tanpa framework modern), keamanan aplikasi ini sangat bergantung pada implementasi manual yang cermat untuk validasi input di sisi server, sanitasi output sebelum render HTML, dan penanganan sesi yang aman, karena proteksi bawaan dari framework tidak tersedia. Mengandalkan validasi JavaScript di klien saja tidaklah cukup.

Repositori proyek: https://github.com/Labpro-21/if3110-tubes-2024-k01-08

II. Kerentanan dan Referensi Commit

1. CSRF pada Form

Telah ditemukan celah keamanan Cross-Site Request Forgery (CSRF) pada beberapa endpoint form, yaitu:

- Form sign in (/auth/sign-in)
- Form sign up job seeker (/auth/sign-up/job-seeker) dan company (/auth/sign-up/company)
- Form company profile (/company/profile)
- Form apply pekerjaan (/jobs/<job id>/apply)
- Form create pekerjaan (/company/jobs/create)
- Form edit pekerjaan (/company/jobs/<job id>/edit)
- Form view applications (/company/jobs/<job id>/applications)

Form di atas tidak memiliki validasi token CSRF, sehingga berpotensi dieksploitasi oleh pihak tidak bertanggung jawab untuk mengirimkan permintaan palsu atas nama user. Tanpa perlindungan CSRF, aplikasi berisiko membiarkan data form dikirim dari sumber luar tanpa izin pengguna.

Perbaikan yang dilakukan:

- Penambahan proses generate token pada saat form dirender.
- Validasi token pada request POST.
- Respons *401 Unauthorized* jika token tidak valid atau tidak tersedia.

Referensi commit:
https://github.com/DhafinFawwaz/if3110-tubes-2024-k01-08/commit/3089262b21ef60d3de3cfff70179e467f68c25de

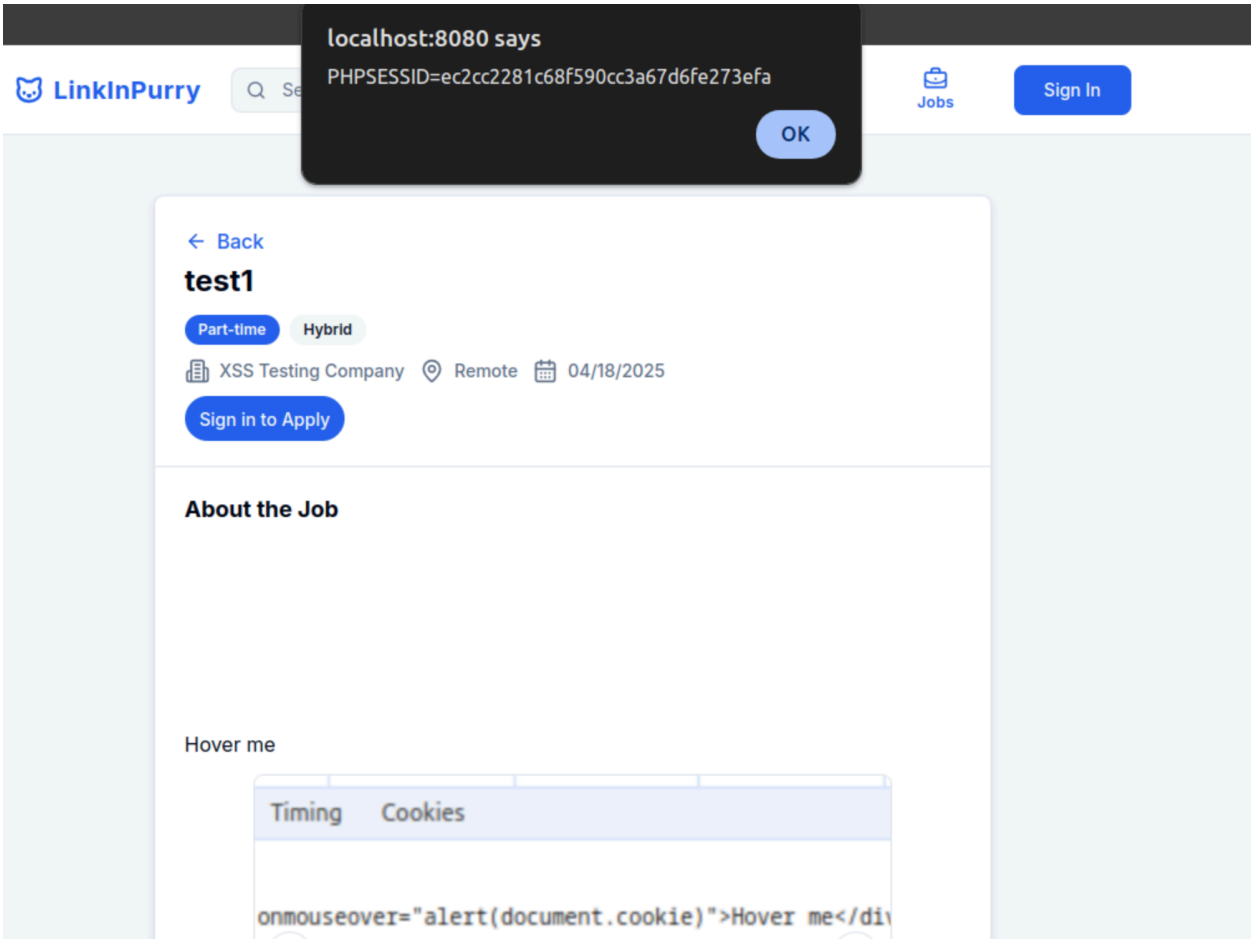
2. *XSS* pada Komponen *Rich Text Editor*

Rich text editor memang kerapkali menjadi makanan empuk untuk *attack XSS*. Web application ini menggunakan rich text editor, sehingga kami curiga bahwa ini dapat dijadikan sasaran untuk mencoba XSS.

Setiap halaman dengan rich text editor memiliki vulnerability ini yaitu:

- Form Create Job (/company/jobs/create)
- Form Edit Job (/company/jobs/100015/edit)

Kami mencoba beberapa sintaks HTML (dengan intensi XSS) sebagai input di dalam rich text pada front end aplikasi, namun hasilnya nihil, output berhasil tersanitasi. Lebih lanjut, saat kami mencoba untuk mencoba POST *directly* ke API, ternyata XSS masih dapat dilakukan. Eksplorasi ini kami dapat bukan hanya *random guessing* tetapi dengan melihat code yang dipakai, yaitu tidak mensanitasi output dari backend.



Gambar 1. Terjadinya *Cross Site Scripting*

Dari observasi *codebase* dan percobaan tersebut kami menyimpulkan bahwa input hanya disanitasi ketika di *client* dan akan dikirim kepada *server*. Sehingga apabila kami langsung

menaruhnya ke *server* melalui API, maka payload tidak tersanitasi sehingga XSS terjadi saat render html.

Perbaikan yang dilakukan:

- Dilakukan sanitasi disaat menampilkan rich text ke layar.

Commit:
https://github.com/DhafinFawwaz/if3110-tubes-2024-k01-08/commit/9eb29c12ea6f3d9bc0d6ba38c83e04dd4a407d05

3. **Storage Exhaustion**

   Ditemukan celah keamanan pada proses upload file yang memungkinkan terjadinya Storage Exhaustion. Ini dapat terjadi karena file yang diupload sangat panjang (> 200 karakter) sehingga terjadi rollback pada database akibat validasi nama file. Namun, file yang diupload tetap masuk ke dalam database. Hal ini menyebabkan file tetap memenuhi ruang penyimpanan walaupun data terkait tidak tercatat di database.

   Halaman yang memiliki celah ini yaitu:

   - Form Apply Job (/jobs/<job id>/apply)

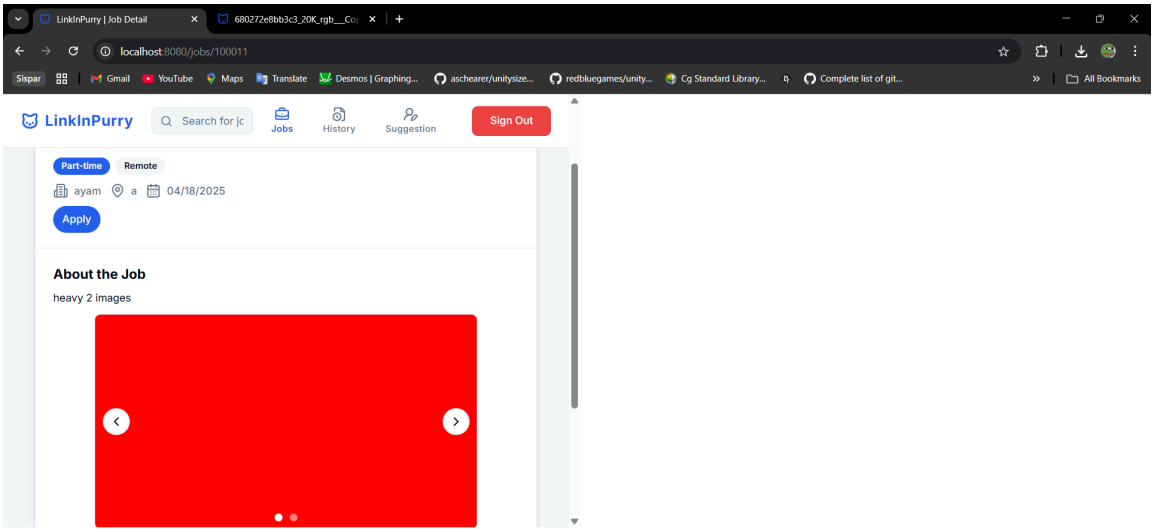
   Perbaikan yang dilakukan:

   - Penambahan proses pembersihan dengan pemotongan nama file menjadi maksimal 100 karakter ketika terjadi kegagalan insert ke database.

   Referensi commit:
   https://github.com/DhafinFawwaz/if3110-tubes-2024-k01-08/commit/3209ff4ba7a38df6f04f123fddfb9be0a6897f01

4. **Client-side Denial of Service (Image Bomb)**

   Terdapat celah keamanan pada pembuatan job baru saat mengupload file image dengan dimension yang sangat besar, misalnya dengan ukuran 20000px × 20000px. Server tidak mengecek ukuran dimensi dari image yang diupload. Ini menyebabkan browser client menjadi menjadi ngelag dan tidak dapat melanjutkan aktivitasnya dengan mudah saat browser merender image tersebut. Dengan perangkat yang lebih lemah bisa jadi browser akan freeze atau bahkan crash. Berikut ini contoh client side yang freeze karena merender gambar merah denga ukuran 20000px × 20000px.



Gambar 2. Bukti Terjadi

Halaman yang memiliki celah ini yaitu:

- Form Create Job (/company/jobs/create)
- Form Edit Job (/company/jobs/100015/edit)

Perbaikan yang dilakukan:

- Dilakukan validasi file dengen mengecek dimension dari file image.

Referensi commit:
https://github.com/DhafinFawwaz/if3110-tubes-2024-k01-08/commit/aa90fa41812914d4c65c4bf56d7fcb1c82ca4d38

5. Improper File Upload Validation

Ditemukan celah keamanan saat mengupload file csv dan video saat mengapply job dan saat mengupload multiple file images saat membuat dan mengedit job. Ini dapat dilakukan dengan membuat file misalnya hacker.php atau hacker.pdf, lalu menyamarkannya dengan merename filenya menjadi hacker.png. Lalu upload file tersebut pada form. Server tidak mengecek apakah file yang diupload benar. Tidak ada exploit berbahaya yang ditemukan untuk celah ini.

Halmaan yang memiliki celah ini yaitu:

- Form Apply Job (/jobs/<job_id>/apply)

Perbaikan yang dilakukan:

- Dilakukan validasi file dengen mengecek MIME TYPE dari file.

Referensi commit:

https://github.com/DhafinFawwaz/if3110-tubes-2024-k01-08/commit/ccf53dca962681043181062f05a58f001040d73f

CSRF
https://github.com/DhafinFawwaz/if3110-tubes-2024-k01-08/commit/c754c634d2330cb9d55df0d6c89a1346e884d2b0
/auth/sign-in

```
<html>
<body>
    <form action="http://localhost:8080/auth/sign-in" method="POST">
        <input type="hidden" name="email" value="attack@example.com">
        <input type="hidden" name="password" value="attack">
        <input type="submit" value="Send CSRF!">
    </form>
</body>
</html>
```

CSRF
https://github.com/DhafinFawwaz/if3110-tubes-2024-k01-08/commit/c754c634d2330cb9d55df0d6c89a1346e884d2b0
/auth/sign-up/job-seeker (pake session job seeker)
/auth/sign-up/company (pake session company)

```
<html>
<body>
  <form action="http://localhost:8080/auth/sign-up/job-seeker" method="POST">
    <input type="hidden" name="email" value="attacker@example.com">
    <input type="hidden" name="password" value="attack">
    <input type="submit" value="Send CSRF!">
  </form>

  <script>
    document.forms[0].submit();
  </script>
</body>
</html>
```

CSRF
https://github.com/DhafinFawwaz/if3110-tubes-2024-k01-08/commit/c754c634d2330cb9d55df0d6c89a1346e884d2b0
/company/profile

```
<html>
  <body>
    <form action="http://localhost:8080/company/profile" method="POST">
      <input type="hidden" name="name" value="name">
      <input type="hidden" name="location" value="location">
```

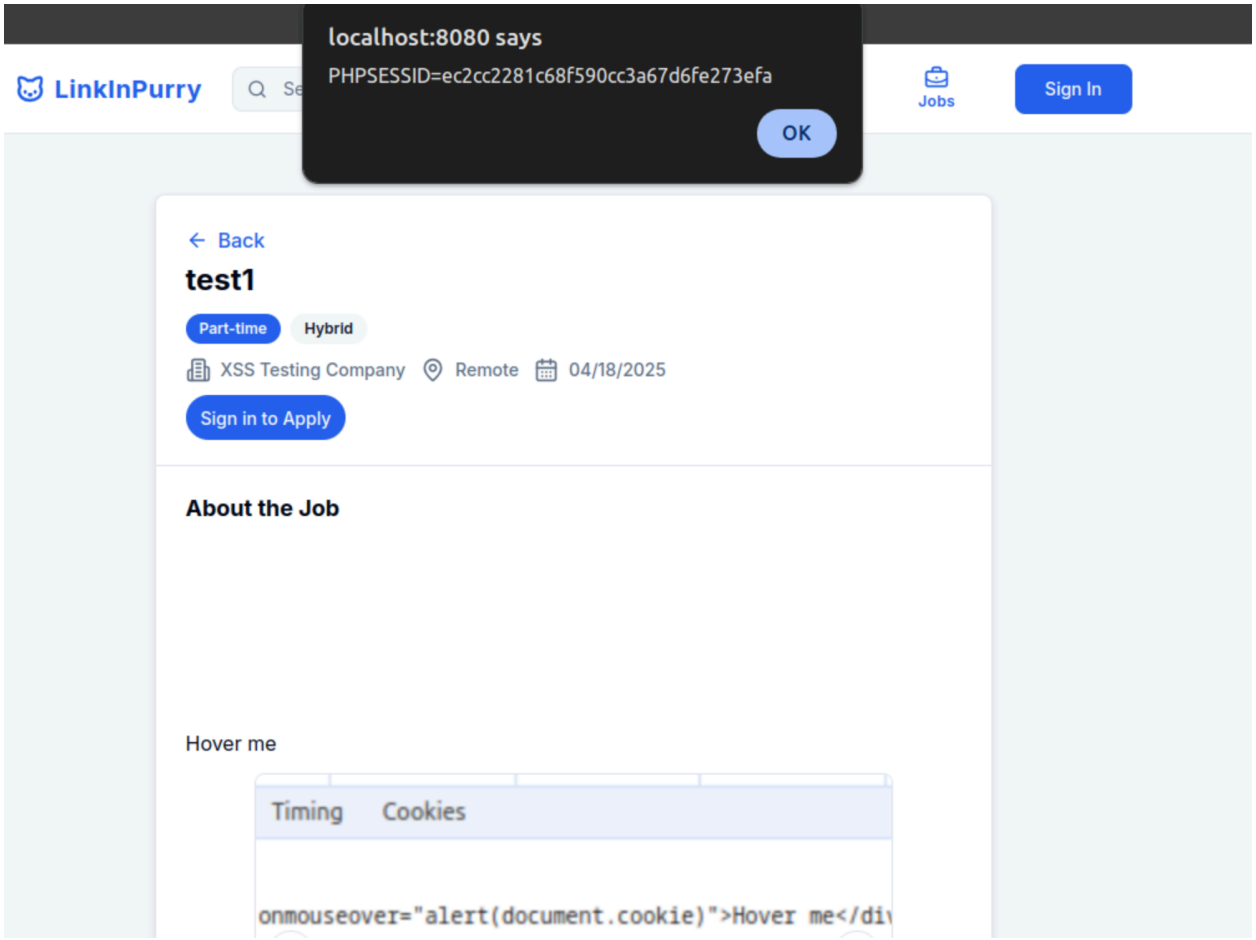
```
    <input type="hidden" name="about" value="about">
    <input type="submit" value="Send CSRF!">
  </form>

  <script>
    document.forms[0].submit();
  </script>
 </body>
</html>
```

XSS
https://github.com/DhafinFawwaz/if3110-tubes-2024-k01-08/commit/fc3bb69206e2a44768d9a0bbb4f5975032b0daf3
Jika payload pada multipart/form-data dengan http POST ke endpoint berikut (gunakan curl/postman misalnya)
http://localhost:8080/company/jobs/create
http://localhost:8080/company/jobs/100015/edit
Untuk payload description contohnya

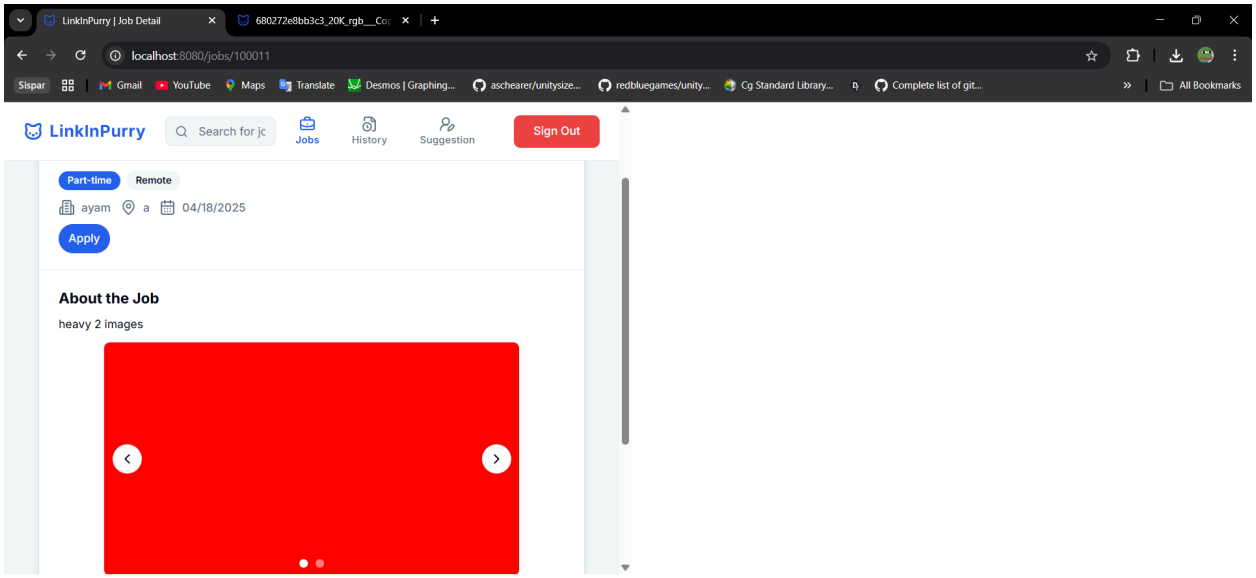
```
<p><svg onload=”alert(document.cookie)”></svg></p>
```

Hacker bisa mendapatkan credentials user dan menggunakan nya untuk login sebagai user lain..



Storage Exhaustion
https://github.com/DhafinFawwaz/if3110-tubes-2024-k01-08/commit/3209ff4ba7a38df6f04f123fddfb9be0a6897f01
Buat file dengan filename yang panjangnya 200 karakter. Upload file png pada form ke endpoint http://localhost:8080/jobs/100015/apply
Yang terjadi adalah row sql yg diinsert telah dirollback karena error panjang nama file, tetapi filenya belum dihapus. Dengan mengupload file yang length filenamenya misal 200 an karakter sudah cukup karena ada tambahan prefix dari path url dan tambahan lagi untuk prefix uniqueid nya.

Client-side Denial of Service (with Image Bomb)
https://github.com/DhafinFawwaz/if3110-tubes-2024-k01-08/commit/aa90fa41812914d4c65c4bf56d7fcb1c82ca4d38
Upload file image dengan size file kecil tetapi memiliki image dimension yang besar misalnya 20000 x 20000. Kinerja dari browser client yang akan menampilkan gambar tersebut akan menjadi sangat lambat. Misal resize window dan semuanya jadi lambat. Kadang beberapa yang diklik memberikan response dalam waktu yang cukup lama.



Improper File Upload Validation
https://github.com/DhafinFawwaz/if3110-tubes-2024-k01-08/commit/ccf53dca962681043181062f05a58f001040d73f
Gunakan file lain misal hacker.php atau hacker.pdf, rename jadi png. Upload ke halaman yang ada formnya misalnya pada endpoint
http://localhost:8080/company/jobs/create
http://localhost:8080/company/jobs/100015/edit
http://localhost:8080/jobs/100015/apply
Namun ini tidak bisa diexploit dan tidak berdampak berat pada website.