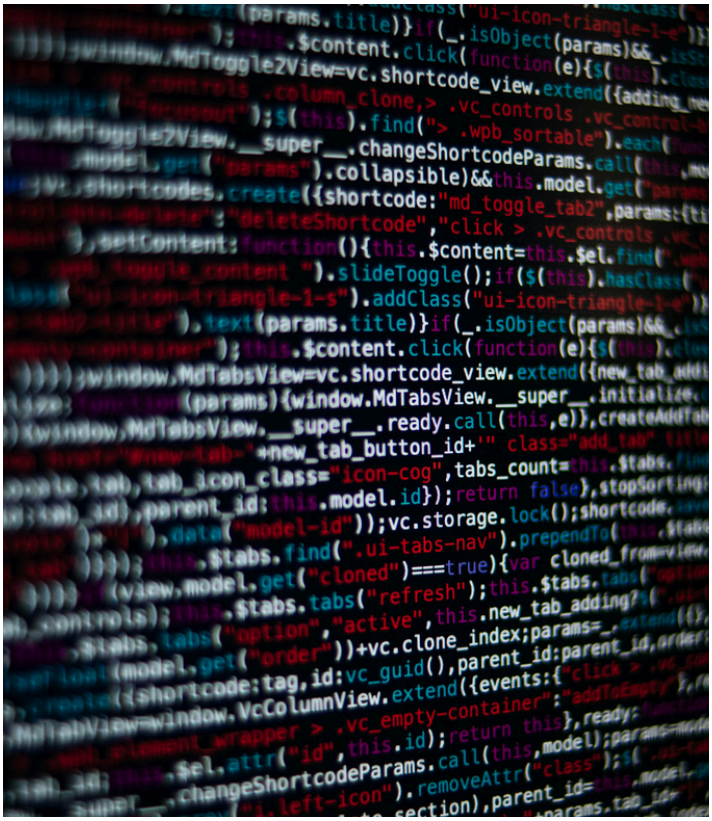
PRAKTIKUM 7

BAC and Business Logic Vulner

Prepared by Kayla Putri Maharani 5026231158 PAI B



OVER ---- VIEW

Praktikum 7 berfokus pada pemahaman dan eksploitasi kelemahan *Broken Access Control* serta *Business Logic Vulnerabilities* pada aplikasi web. Fokus utamanya adalah menganalisis bagaimana kontrol akses yang lemah dapat dilewati dan bagaimana logika bisnis aplikasi dapat dimanipulasi sehingga menghasilkan perilaku yang tidak sesuai desain.

PRAKTIKUM 7 ---- REPORT

Daftar Isi

Horizontal Privilege Escalation

Flag : PAI25{h0r1z0n741_pr1v11363_35c4141710n_1d0r_3hh3hh3}

Deskripsi Soal

1. Aplikasi menyediakan form register dan login. Setelah login, user diarahkan ke halaman profil dengan URL yang mengandung parameter ID dalam bentuk string yang terlihat seperti hasil encoding.
2. Tugas yang harus dilakukan yaitu mengakses profil user lain tanpa otorisasi yang benar dengan teknik Horizontal Privilege Escalation untuk mendapatkan flag di profil admin.
3. Horizontal di sini artinya saya tetap sebagai user biasa, tetapi mencoba mengakses data milik user lain pada level hak akses yang sama, misalnya admin atau user lain, dengan memanipulasi identitas target.

Analisis Logika

1. Setelah registrasi dan login, aplikasi mengarahkan ke halaman profil dengan URL http://195.85.19.90:5000/profile/MTM0
2. String MTM0 terlihat bukan angka biasa, tetapi lebih mirip Base64. Ketika dicek, MTM0 adalah Base64 dari angka 134.
   Artinya
   - Aplikasi menyimpan ID asli user di server
   - Lalu menampilkan ID tersebut dalam bentuk Base64 di URL
   - Tidak ada bukti bahwa server memeriksa apakah ID yang diakses sesuai dengan user yang sedang login
3. Ini adalah pola klasik IDOR Insecure Direct Object Reference
   Objek user diidentifikasi hanya berdasarkan ID di URL tanpa pengecekan tambahan seperti
   - Apakah ID tersebut memang milik session saat ini
   - Apakah role atau user yang login punya hak akses ke ID tersebut

Langkah Pengerjaan

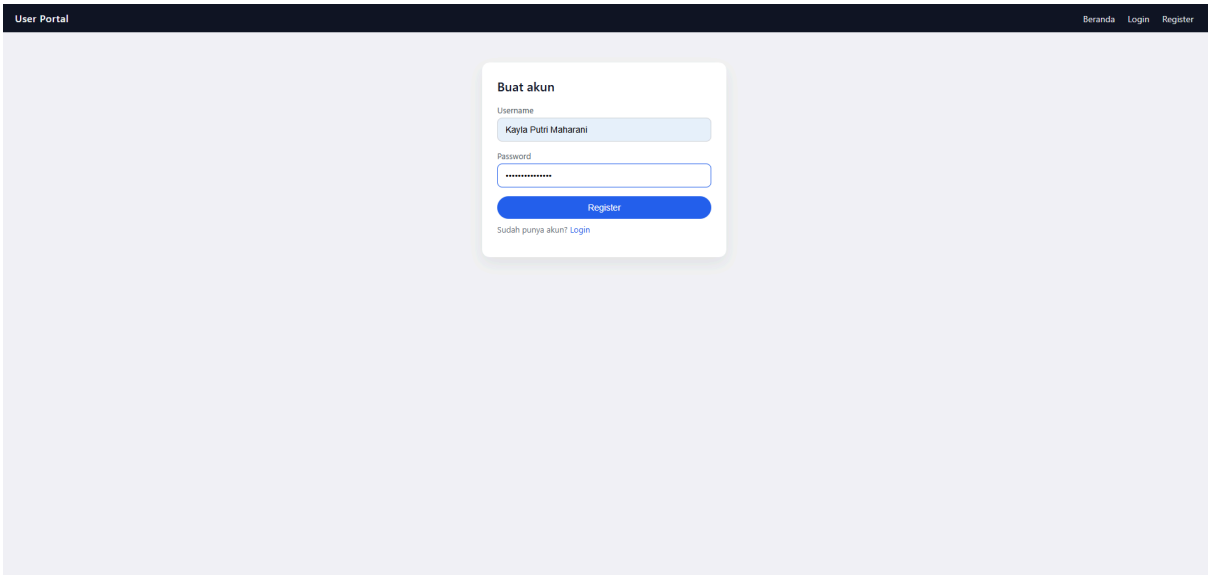
1. Register akun dan login sampai tiba di halaman profil
   Contoh hasil login URL profil saya
   http://195.85.19.90:5000/profile/MTM0
   ID asli saya 134 yang di encode menjadi MTM0
2. Menguji encoding dengan mengganti ID
   Saya menghitung beberapa nilai Base64 dari angka kecil, misalnya
   - 1 menjadi MQ==
   - 2 menjadi Mg==
   - 3 menjadi Mw==
3. Lalu saya mencoba membuka
   http://195.85.19.90:5000/profile/MQ==
4. Halaman profil yang muncul bukan lagi akun saya, tetapi akun lain yaitu admin. Pada halaman tersebut terdapat field khusus
   SECRET TOKEN. Nilai dari field ini adalah flag :
   PAI25{h0r1z0n741_pr1v11363_35c4141710n_1d0r_3hh3hh3}
5. Logika di sisi server kemungkinan besar seperti ini
   - Server menerima request ke endpoint /profile/{encoded_id}
   - Server melakukan decode Base64 menjadi ID numerik
   - Server langsung mengambil data user dengan ID tersebut dari database
   - Server menampilkan data tanpa cek apakah ID itu sama dengan user yang sedang login

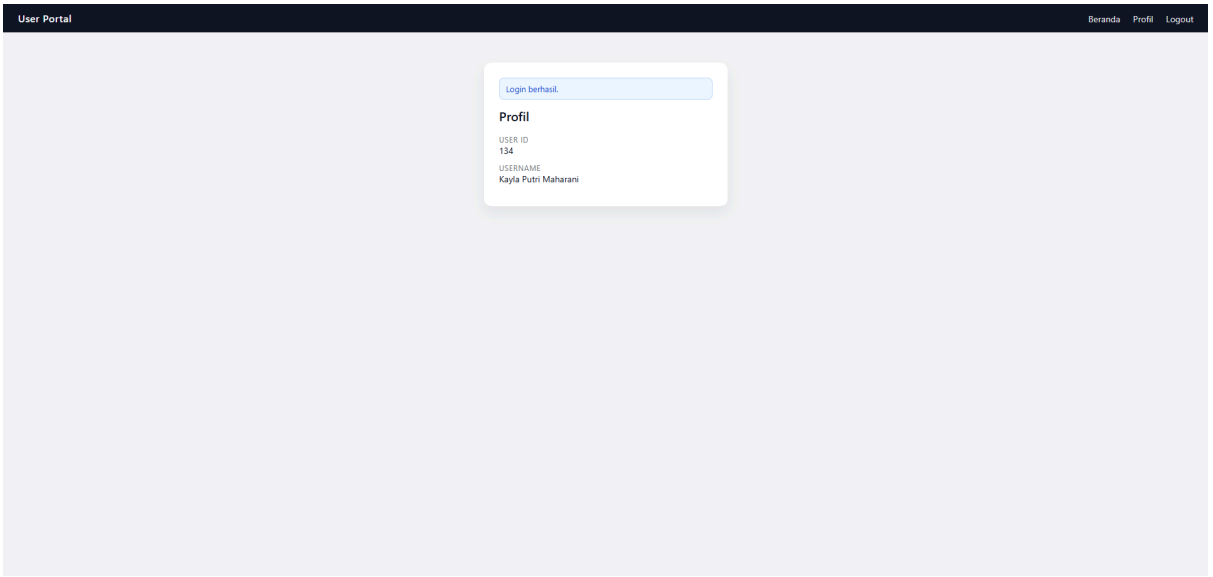
   Jadi, siapa pun yang sudah login dapat mengakses profil user lain hanya dengan mengganti nilai ID yang di encode.

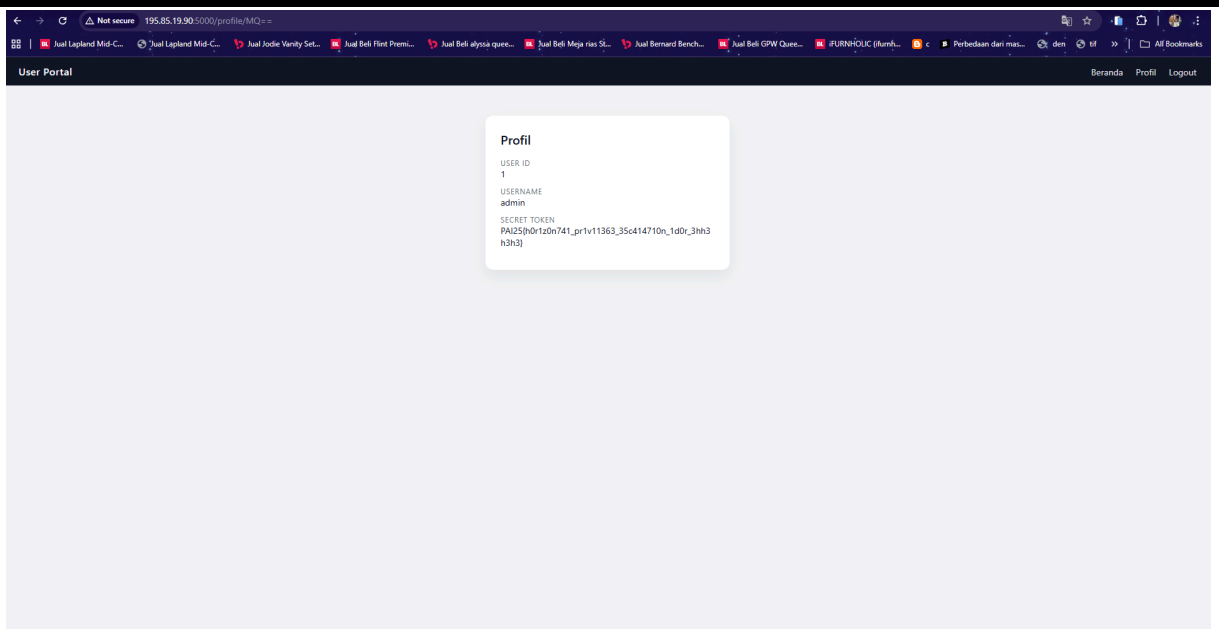
Kesimpulan

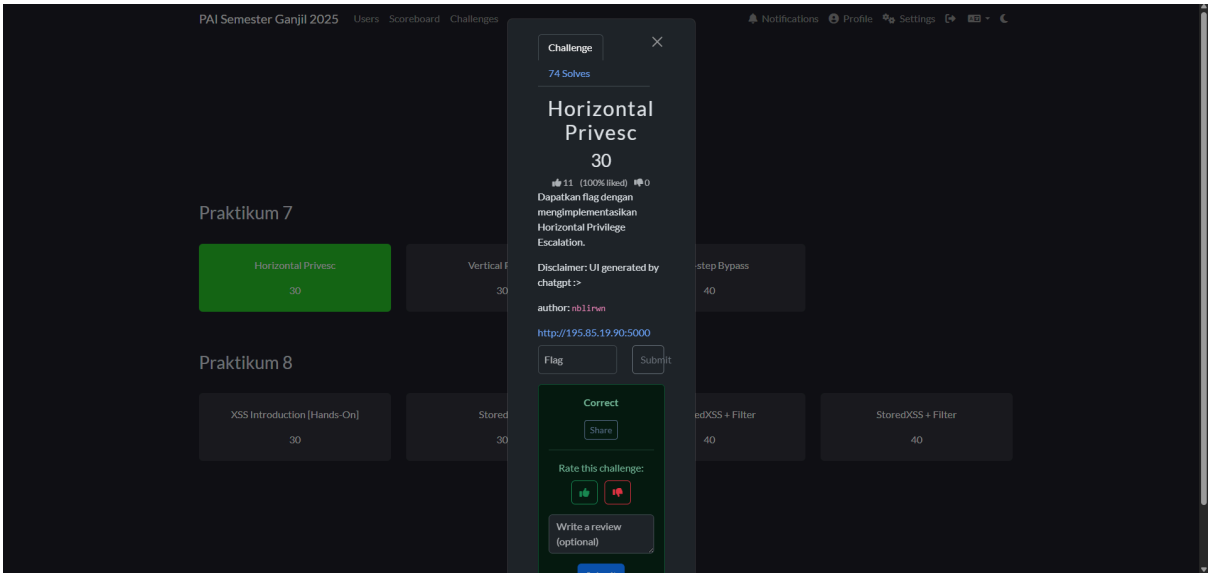
Kelemahan utama yaitu aplikasi tidak menerapkan pengecekan akses berbasis session. Ketika suatu endpoint mengizinkan user mengubah ID objek secara langsung di URL tanpa validasi tambahan, ini menjadi IDOR dan memungkinkan Horizontal Privilege Escalation seperti yang terjadi di sini.

Lampiran Step






Not secure
195.85.19.90:5000/profile/MQ==
User Portal
Beranda
Profil
Logout
Profil
USER ID
1
USERNAME
admin
SECRET TOKEN
PAI25{h0r1z0n741_pr1v11363_35c414710n_1d0r_3hh3h3h3}


PAI Semester Ganjil 2025
Users
Scoreboard
Challenges
Notifications
Profile
Settings
Praktikum 7
Horizontal Privesc
30
Praktikum 8
XSS Introduction [Hands-On]
30
StoredXSS + Filter
40
Challenge
74 Solves
Horizontal Privesc
30
11 (100% liked) 0
Dapatkan flag dengan mengimplementasikan Horizontal Privilege Escalation.
Disclaimer: UI generated by chatgpt :>
author: nbliriwn
http://195.85.19.90:5000
Flag
Submit
Correct
Share
Rate this challenge:
Write a review (optional)

Vertical Privilege Escalation

Flag : PAI25{v3r71c41_pr1v11363_35c414710n_c00k13_r013}

Deskripsi Soal

Aplikasi Simple Panel memiliki dua role utama

- user
- admin

Halaman admin /admin seharusnya hanya bisa diakses oleh role admin. Setiap user baru akan memiliki role default user di dashboard.

Tugas yang harus dilakukan yaitu meningkatkan hak akses dari user biasa menjadi admin dengan memanfaatkan kelemahan penyimpanan role di cookie.

Analisis Logika

1. Setelah register dan login, di dashboard terlihat informasi
    - Role : user
    - Jika mencoba akses /admin muncul pesan Access denied
2. Dengan membuka Developer Tools lalu ke Application dan Cookies, ditemukan tiga nilai penting
    - role = user
    - username = "Kayla Putri Maharani"
    - session = <session_token>
3. Masalah logis di sini

    Informasi sensitif berupa role user disimpan di cookie pada sisi klien. Aplikasi tampaknya langsung mempercayai nilai role dari cookie tanpa validasi ulang dengan database di sisi server. Dengan kata lain server menggunakan sesuatu seperti role = $_COOKIE["role"] untuk menentukan hak akses user.Ini adalah contoh Broken Access Control pada level vertikal. Saya bisa mengubah role sendiri dari user menjadi admin.

Langkah Pengerjaan

1. Login dengan akun user biasa
   Dashboard menampilkan role user. Akses ke /admin masih ditolak.
2. Buka DevTools -> Tab Application -> Bagian Cookies -> Edit nilai cookie role
   - Sebelumnya
     role = user
   - Diubah menjadi
     role = admin
3. Setelah halaman di refresh
   - Badge di dashboard berubah menjadi admin
   - Tombol atau menu menuju admin console muncul
4. Ketika tombol tersebut di klik, atau langsung mengunjungi /admin
   - Halaman admin console berhasil dibuka
   - Di dalamnya terdapat ADMIN TOKEN yang merupakan flag
     PAI25{v3r71c41_pr1v11363_35c414710n_c00k13_r013}

Logika di Sisi Server

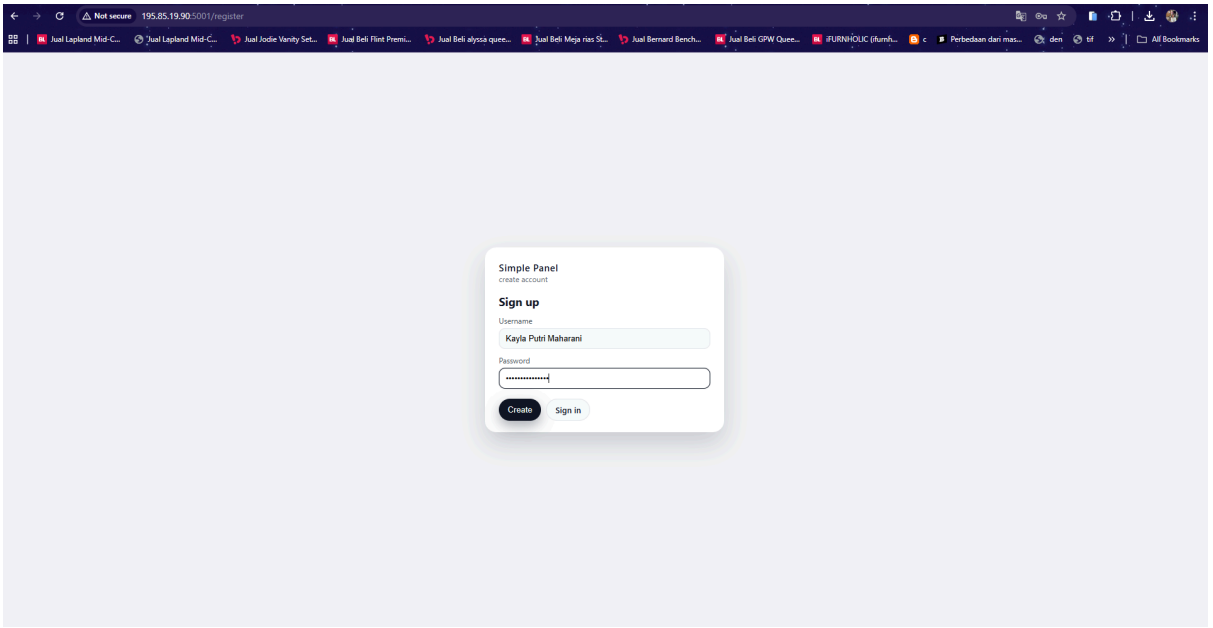
- Perilaku aplikasi ini mengindikasikan pola seperti pada setiap request, aplikasi membaca cookie role
  role = $_COOKIE["role"]
- Untuk menentukan apakah user boleh mengakses halaman admin aplikasi hanya melakukan pemeriksaan sederhana seperti
  - Jika role sama dengan admin maka izinkan
  - Jika bukan admin maka tolak
- Tidak ada validasi
  - Apakah role yang tertulis di cookie sesuai dengan yang ada di database
  - Apakah perubahan nilai role di cookie terjadi secara sah melalui proses internal sistem

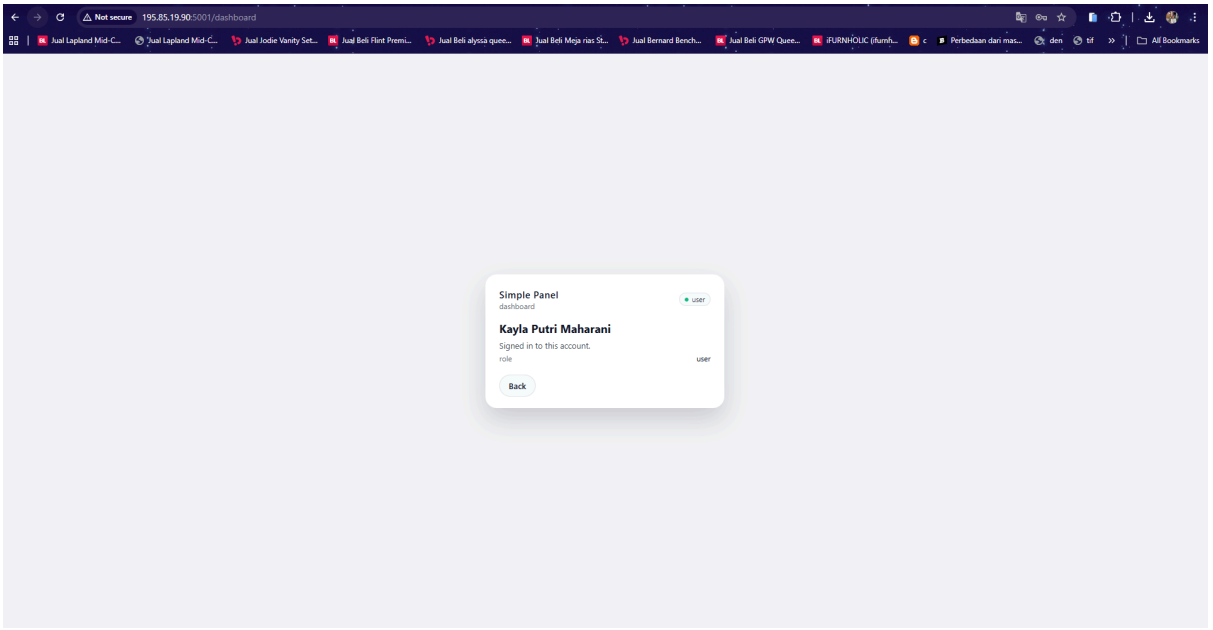
Akibatnya User bisa langsung memodifikasi nilai role di cookie dan otomatis memperoleh hak admin tanpa proses administrasi apa pun.

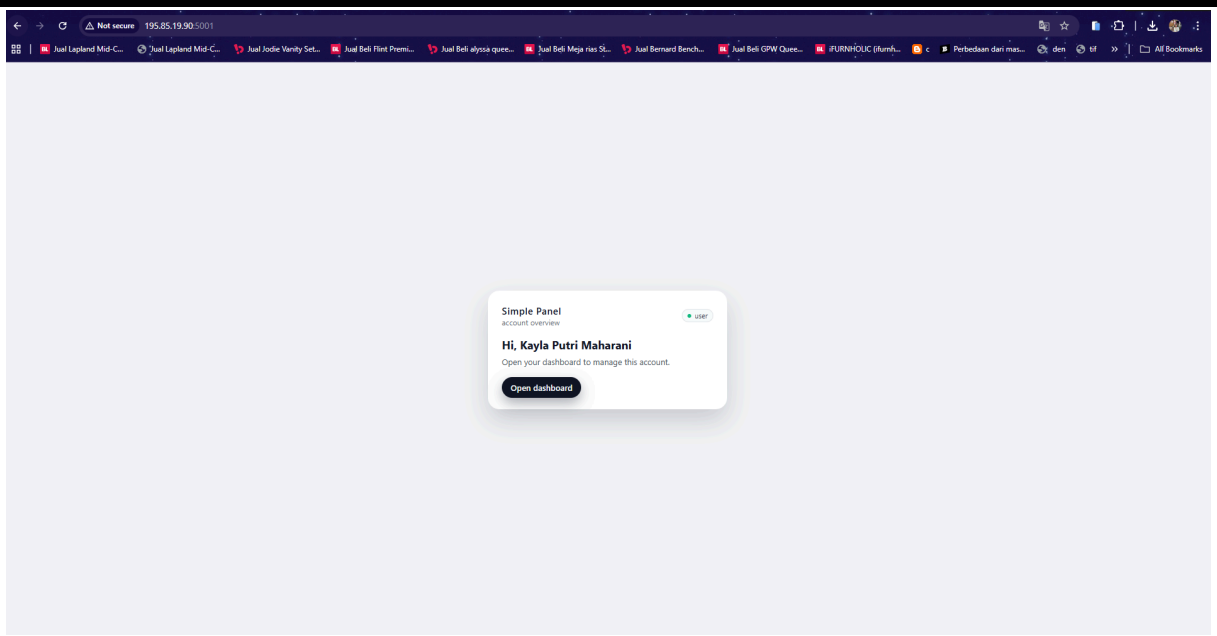
Kesimpulan

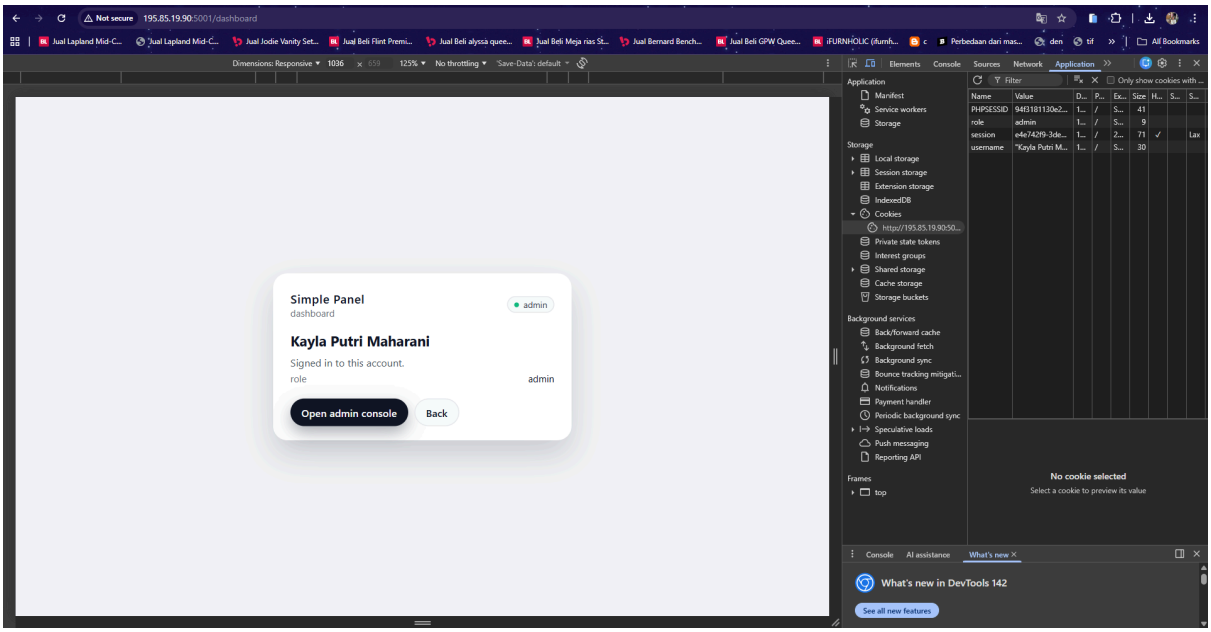
Vertical Privilege Escalation terjadi karena logika otorisasi hanya bergantung pada nilai cookie yang dikendalikan klien. Ini sangat berbahaya karena hak akses tertinggi bisa diperoleh hanya dengan mengedit data di browser, tanpa kompromi kredensial admin asli.

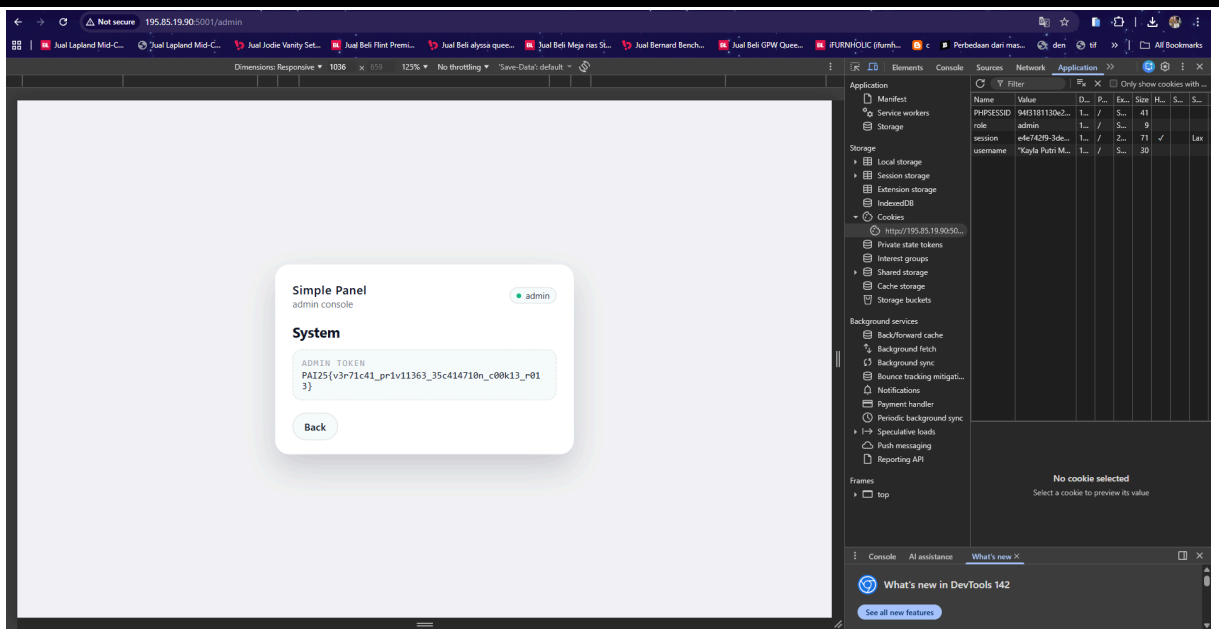
Lampiran Step

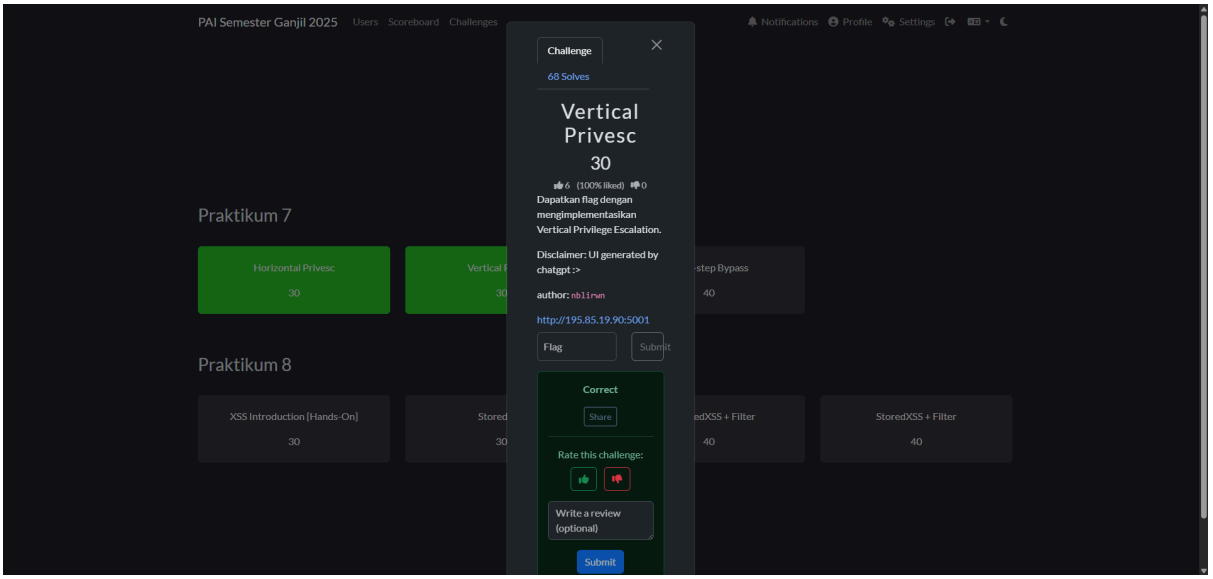





Not secure
195.85.19.90:5001
Simple Panel
account overview
user
Hi, Kayla Putri Maharani
Open your dashboard to manage this account.
Open dashboard


Not secure
195.85.19.90:5001/dashboard
Simple Panel
dashboard
admin
Kayla Putri Maharani
Signed in to this account.
role
admin
Open admin console
Back
Elements
Console
Sources
Network
Application
No cookie selected
Select a cookie to preview its value
What's new in DevTools 142
See all new features


Not secure 195.85.19.90:5001/admin
Simple Panel
admin console
admin
System
ADMIN TOKEN
PAI25{v3r71c41_pr1v11363_35c414710n_c00k13_r013}
Back
Application
Manifest
Service workers
Storage
Storage
Local storage
Session storage
Extension storage
IndexedDB
Cookies
Private state tokens
Interest groups
Shared storage
Cache storage
Storage buckets
Background services
Back/forward cache
Background fetch
Background sync
Bounce tracking mitigati...
Notifications
Payment handler
Periodic background sync
Speculative loads
Push messaging
Reporting API
Frames
top
No cookie selected
Select a cookie to preview its value
What's new in DevTools 142
See all new features


PAI Semester Ganjil 2025
Users
Scoreboard
Challenges
Notifications
Profile
Settings
Praktikum 7
Horizontal Privesc
30
Praktikum 8
XSS Introduction [Hands-On]
30
Challenge
68 Solves
Vertical Privesc
30
6 (100% liked) 0
Dapatkan flag dengan mengimplementasikan Vertical Privilege Escalation.
Disclaimer: UI generated by chatgpt :>
author: nbliwn
http://195.85.19.90:5001
Flag
Submit
Correct
Share
Rate this challenge:
Write a review (optional)
Submit

Multi step Process Bypass

Flag encoded :UEFJMjV7NTFtcDEzX211MTcxXzU3M3BfcHlwYzM1NV9ieXA0NTVfdjE0X2YxMzFkX3gxeDF9

Flag decoded : PAI25{simple_multi_step_process_bypass_via_field_x1x1}

Deskripsi Soal

Aplikasi Acme Membership menerapkan proses upgrade akun melalui beberapa tahap

1. User mengirim request upgrade
2. Admin melakukan review
3. Admin melakukan approve atau reject

Secara normal, user tidak bisa langsung menjadi admin karena harus menunggu tindakan admin di sisi backend.

Tugas yang harus dilakukan yaitu memanfaatkan kelemahan logika bisnis untuk melewati seluruh proses tersebut dan langsung mengubah role menjadi admin.

Analisis Logika

1. Setelah login sebagai user biasa, dashboard menampilkan
   - Current role : user
   - Status upgrade : pending atau belum diapprove
2. Ketika tombol Request upgrade ditekan saya mengamati tab Network di DevTools.
   Hasilnya :
   - Tidak tampak request POST yang mengirim data penting dari UI
   - Hal ini mengindikasikan bahwa proses sesungguhnya mungkin hanya terjadi di endpoint backend tertentu yang biasanya dipanggil oleh admin, bukan user
3. Dari potongan kode backend yang diberikan pada soal, ditunjukkan bahwa terdapat endpoint upgrade.php yang mengubah role berdasarkan parameter tertentu ketika menerima request POST.
   Contoh logika sederhana yang mungkin digunakan
   - Jika menerima POST dengan
     upgrade = 1
     dan
     accept = 1
     maka role user diupdate menjadi admin.

4. Di sinilah muncul Business Logic Vulnerability
    - Backend tidak memverifikasi siapa yang memanggil endpoint upgrade tersebut
    - Backend hanya mengecek isi parameter POST
    - Tidak ada pembatasan bahwa parameter tersebut hanya boleh dikirim oleh role admin atau melalui panel admin

Artinya, selama saya sudah login sebagai user dan memiliki session yang sah, saya bisa langsung menembak endpoint upgrade.php dengan payload yang benar dan memancing logika upgrade berjalan seolah olah sudah di approve admin.

Langkah Pengerjaan

Untuk mengeksploitasi kelemahan ini, saya menggunakan JavaScript fetch dari console browser.

1. Buka halaman aplikasi dalam kondisi sudah login sebagai user biasa
   Dashboard masih menampilkan Current role : user
2. Buka DevTools ->Tab Console
3. Jalankan kode berikut

```
fetch("upgrade.php", {
  method: "POST",
  headers: { "Content-Type": "application/x-www-form-urlencoded" },
  body: "upgrade=1&accept=1"
});
```

4. Alasan teknis kenapa saya memakai kode tersebut :
    a. Endpoint yang dituju
       upgrade.php
       Dari konteks soal dan potongan backend, endpoint ini adalah tempat logika upgrade diproses.
    b. Method yang digunakan
       method: "POST"
       Potongan backend menunjukkan bahwa logika upgrade membaca nilai dari parameter POST bukan GET, misalnya
       $_POST["upgrade"] dan $_POST["accept"]
    c. Header yang diberikan
       Content-Type: application/x-www-form-urlencoded
       Ini adalah format standar form web biasa.
       Dengan header ini, body akan dikirim dalam bentuk pasangan key value seperti yang biasanya dikirim oleh form HTML.

d. Body yang dikirim
   body: "upgrade=1&accept=1"
   Dari logika yang diberikan di soal, kondisi upgrade ke admin terjadi ketika
   - upgrade bernilai 1 menandakan ada permintaan upgrade
   - accept bernilai 1 menandakan permintaan tersebut disetujui

   Jadi kode ini meniru request yang seharusnya dikirim oleh admin ketika menekan tombol Approve pada panel admin.Bedanya request ini justru dikirim langsung oleh user biasa melalui console tanpa melalui UI admin.
e. Mengapa fetch di console cukup untuk menjadi admin
   - Saat saya menjalankan fetch di console, request dikirim menggunakan session dan cookie yang aktif di browser
   - Backend melihat request POST ke upgrade.php dengan parameter benar
   - Backend tidak memeriksa apakah pengirimnya admin atau user biasa
   - Backend hanya melihat payload valid lalu menjalankan logika mengubah role user menjadi admin di database

   Dengan kata lain saya memaksa backend menjalankan jalur logika final proses upgrade tanpa mengikuti alur resmi multi step di UI.

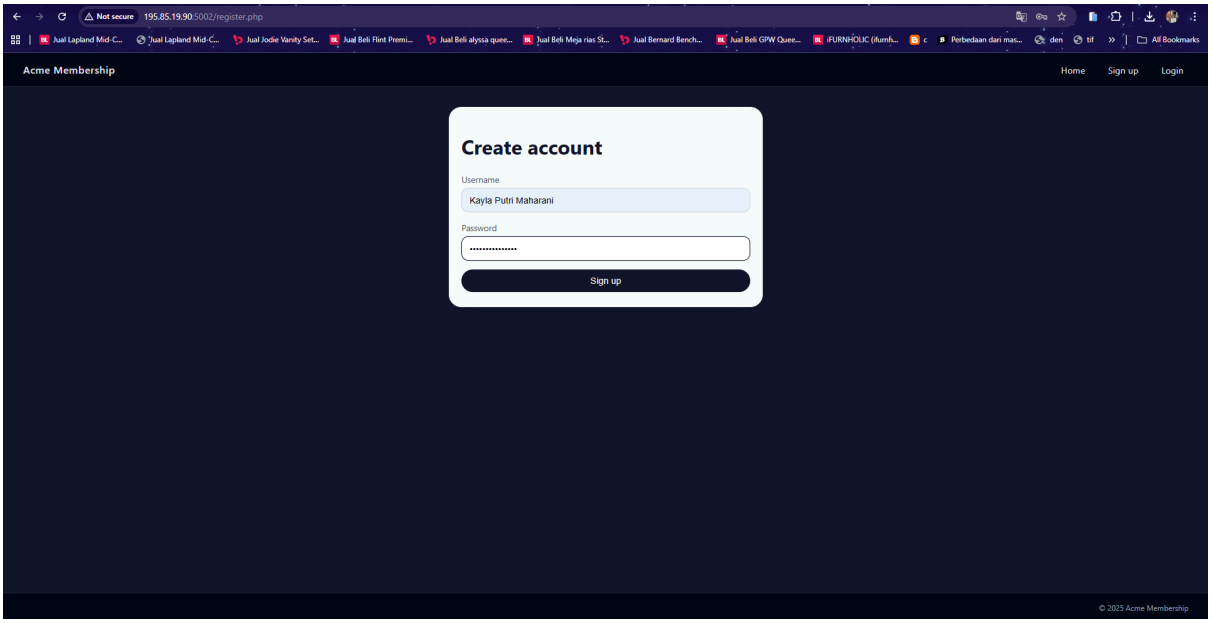
5. Setelah menjalankan kode fetch, saya refresh halaman dashboard
   - Current role berubah menjadi admin
   - Panel admin menjadi dapat diakses
6. Setelah menjalankan kode fetch, saya refresh halaman dashboard
   - Current role berubah menjadi admin
   - Panel admin menjadi dapat diakses

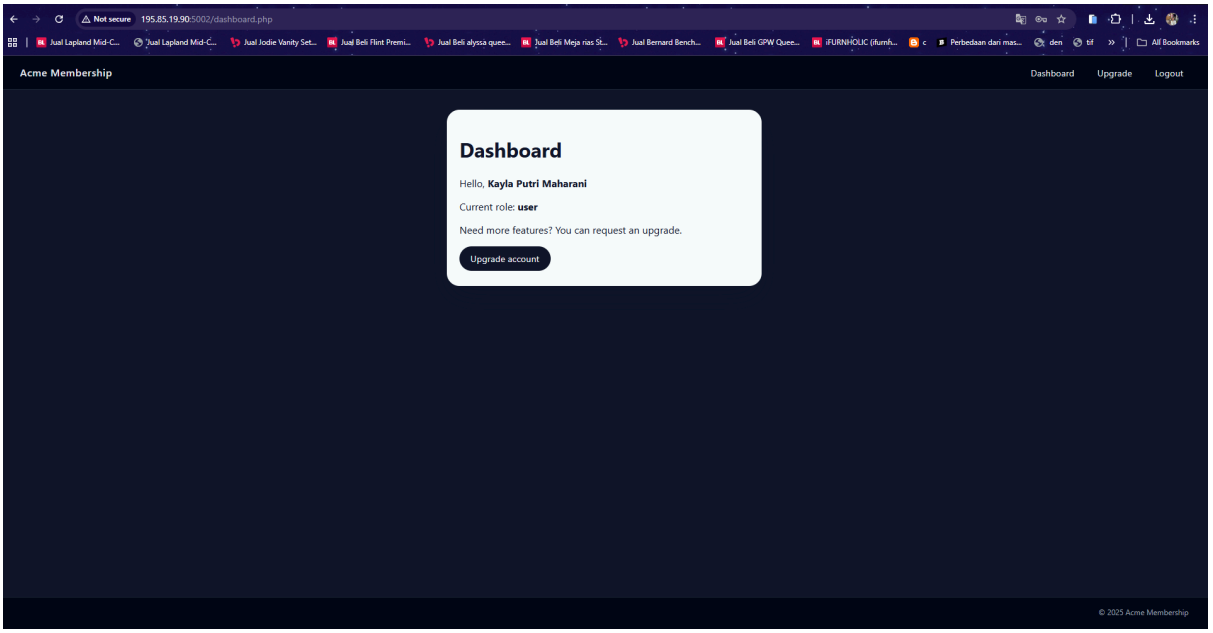
   Pada halaman admin, flag muncul dalam bentuk string yang masih di encode
   UEFJMjV7NTFtcDEzX211MTcxXzU3M3BfcHlwYzM1NV9ieXA0NTVfdjE0X2YxMzFkX3gxeDF9
7. Langkah selanjutnya adalah melakukan decode terhadap string tersebut
   - String itu adalah Base64
   - Setelah di decode hasilnya
     PAI25{simple_multi_step_process_bypass_via_field_x1x1}.
     Ini adalah flag final dari challenge.

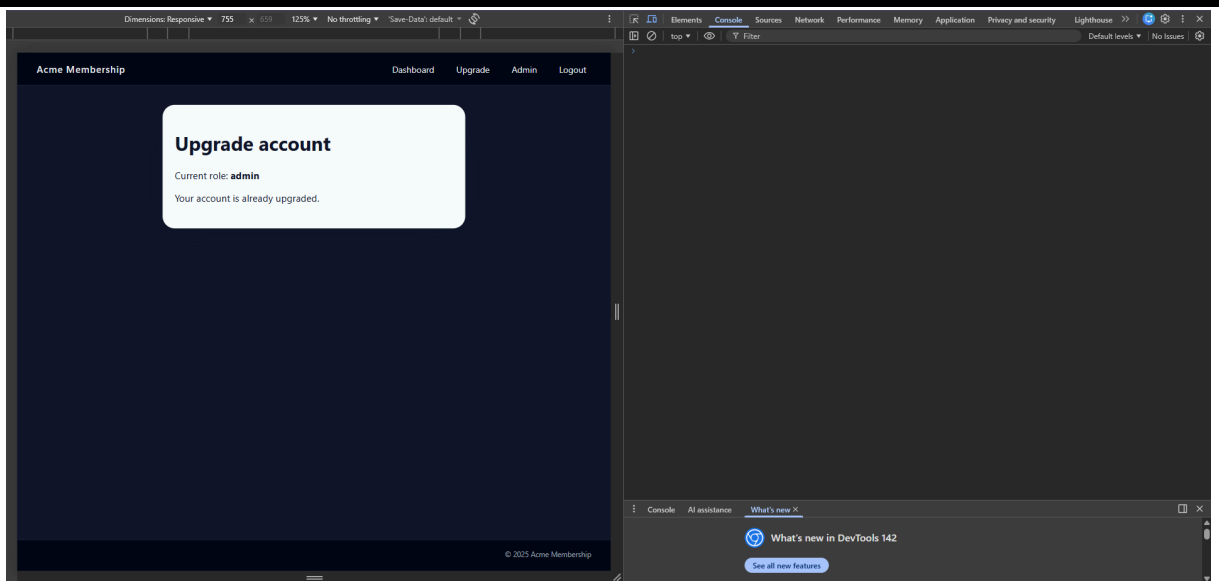
Kesimpulan

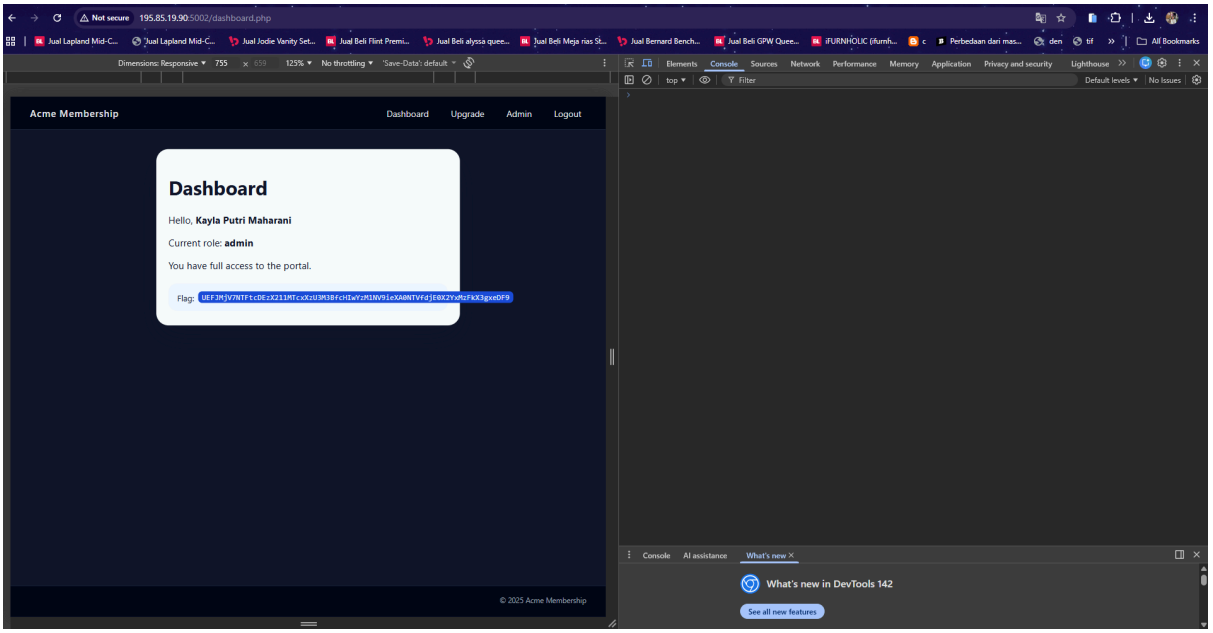
Kelemahan utama pada skenario ini adalah tidak adanya kontrol yang membedakan request yang sah dari admin dengan request manipulasi yang dikirim oleh user biasa. Logika bisnis yang seharusnya menjamin bahwa hanya admin yang boleh menyetujui upgrade akun tidak diikuti oleh implementasi teknis yang tepat di backend.

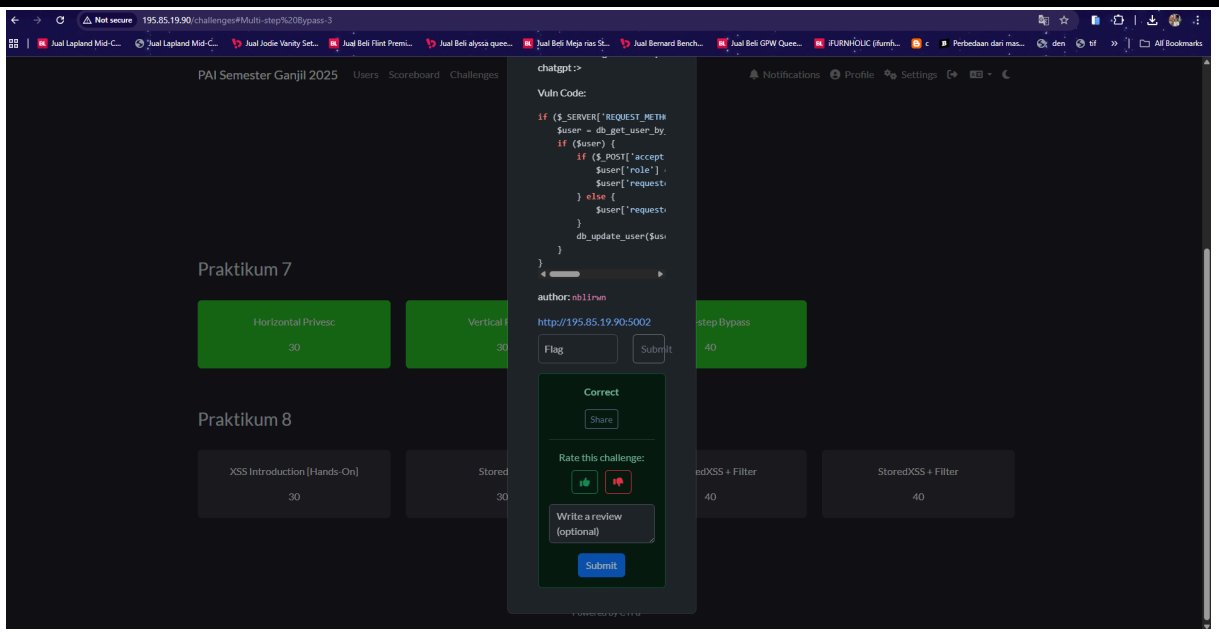
Lampiran Step






Acme Membership
Dashboard
Upgrade
Admin
Logout
Upgrade account
Current role: admin
Your account is already upgraded.
© 2025 Acme Membership
What's new in DevTools 142
See all new features


Not secure
195.85.19.90:5002/dashboard.php
Acme Membership
Dashboard
Upgrade
Admin
Logout
Dashboard
Hello, Kayla Putri Maharani
Current role: admin
You have full access to the portal.
Flag: UEFJMjV7NTFtcDEzX211MTcxXzU3M3BfcHIwYzM1NV9leXA0NTVfdjE0X2YwM2FkX3gxeDF9
© 2025 Acme Membership
What's new in DevTools 142
See all new features


Not secure 195.85.19.90/challenges#Multi-step%20Bypass-3
PAI Semester Ganjil 2025
Users Scoreboard Challenges
Notifications Profile Settings
chatgpt :>
Vuln Code:
author: nblirwn
http://195.85.19.90:5002
Flag
Submit
Correct
Share
Rate this challenge:
Write a review (optional)
Submit
Praktikum 7
Horizontal Privesc
30
40
Praktikum 8
XSS Introduction [Hands-On]
30
40
StoredXSS + Filter
40