

# Dokumentasi: OWASP Top 10–2021

## TryHackMe Room

Jonathan Jethro

May 5, 2025

## Contents

<b>1</b>	<b>Pendahuluan</b>	<b>2</b>
<b>2</b>	<b>Broken Access Control</b>	<b>2</b>
2.1	Deskripsi . . . . .	2
2.2	Langkah-langkah . . . . .	2
2.3	Flag . . . . .	2
<b>3</b>	<b>Cryptographic Failures</b>	<b>3</b>
3.1	Deskripsi . . . . .	3
3.2	Langkah-langkah . . . . .	3
3.3	Flag . . . . .	3
<b>4</b>	<b>Insecure Design</b>	<b>3</b>
4.1	Deskripsi . . . . .	3
4.2	Langkah-langkah . . . . .	4
4.3	Flag . . . . .	4
<b>5</b>	<b>Security Misconfiguration</b>	<b>4</b>
5.1	Deskripsi . . . . .	4
5.2	Langkah-langkah . . . . .	4
5.3	Flag . . . . .	5
<b>6</b>	<b>Vulnerable and Outdated Components</b>	<b>5</b>
6.1	Deskripsi . . . . .	5
6.2	Langkah-langkah Exploit Lab . . . . .	5
6.3	Flag . . . . .	6
<b>7</b>	<b>Identification and Authentication Failures</b>	<b>6</b>
<b>8</b>	<b>Data Integrity Failures</b>	<b>6</b>
<b>9</b>	<b>Security Logging and Monitoring Failures</b>	<b>8</b>

# 1 Pendahuluan

OWASP (Open Worldwide Application Security Project) Top 10 adalah daftar sepuluh risiko keamanan aplikasi web paling kritis yang diidentifikasi oleh komunitas keamanan. TryHackMe menyediakan ruang latihan interaktif untuk memahami dan mengeksploitasi setiap kerentanan ini secara praktis.

## 2 Broken Access Control

### 2.1 Deskripsi

Broken Access Control terjadi ketika pembatasan akses pengguna tidak diterapkan dengan benar, memungkinkan pengguna yang tidak berwenang untuk mengakses fungsi atau data yang seharusnya dibatasi.

### 2.2 Langkah-langkah

1. Deploy mesin dan akses `http://MACHINE_IP`.
2. Login dengan kredensial:
  - Username: `noot`
  - Password: `test1234`
3. Ubah parameter `note_id` di URL untuk mengakses catatan pengguna lain.

### 2.3 Flag

`flag{fivefourthree}`

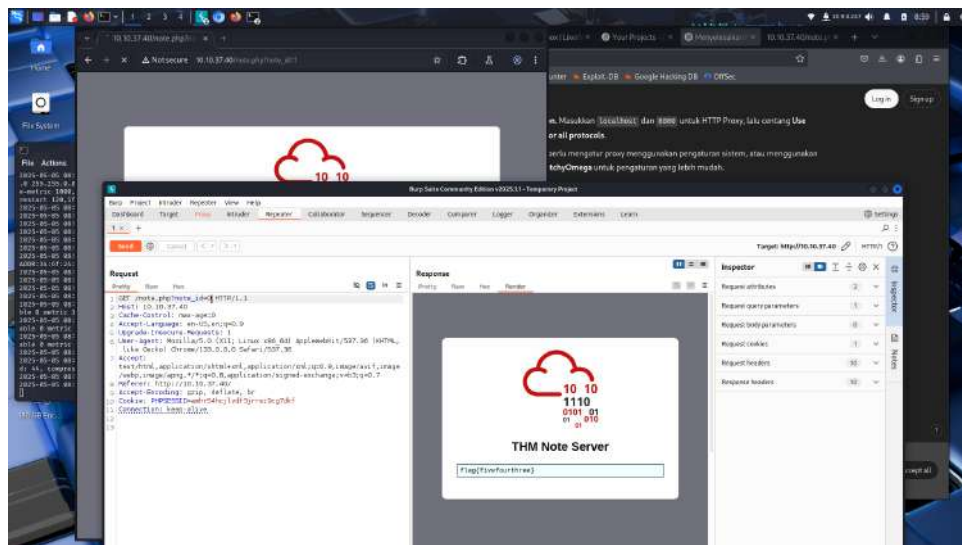


Figure 1: Gambar berhasil mendapatkan flag pada task Broken Access Control

## 3 Cryptographic Failures

### 3.1 Deskripsi

Cryptographic Failures mengacu pada kelemahan dalam implementasi kriptografi yang dapat menyebabkan kebocoran data sensitif.

### 3.2 Langkah-langkah

1. Akses aplikasi di `http://MACHINE_IP:81/`.
2. Periksa kode sumber halaman login untuk menemukan petunjuk direktori sensitif.
3. Temukan direktori `/assets` dan file `webapp.db`.
4. Ekstrak hash password admin dari database.
5. Gunakan alat seperti `crackstation` untuk memecahkan hash.
6. Login sebagai admin untuk mendapatkan flag.

### 3.3 Flag

THM{Yzc2YjdkMjE5N2VjMzNhOTE3NjdiMjdl}

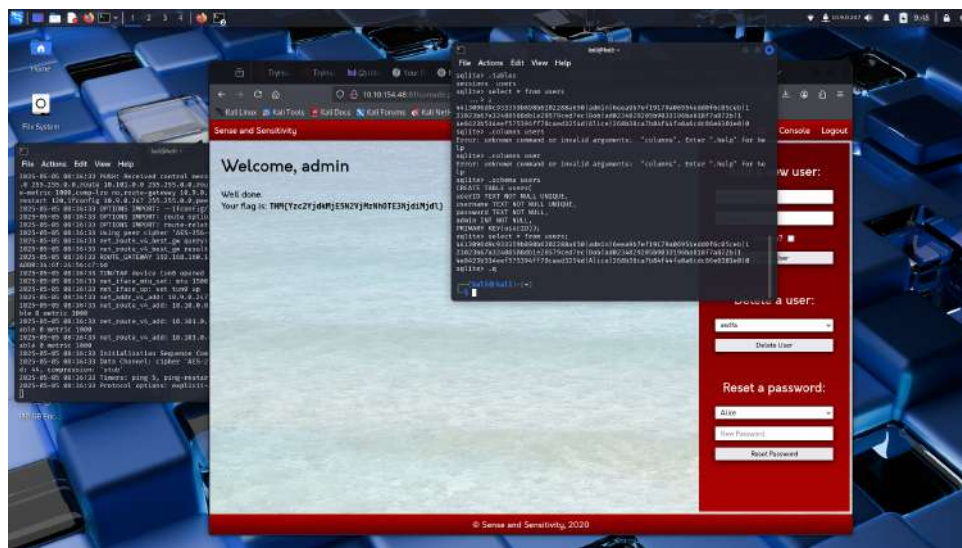


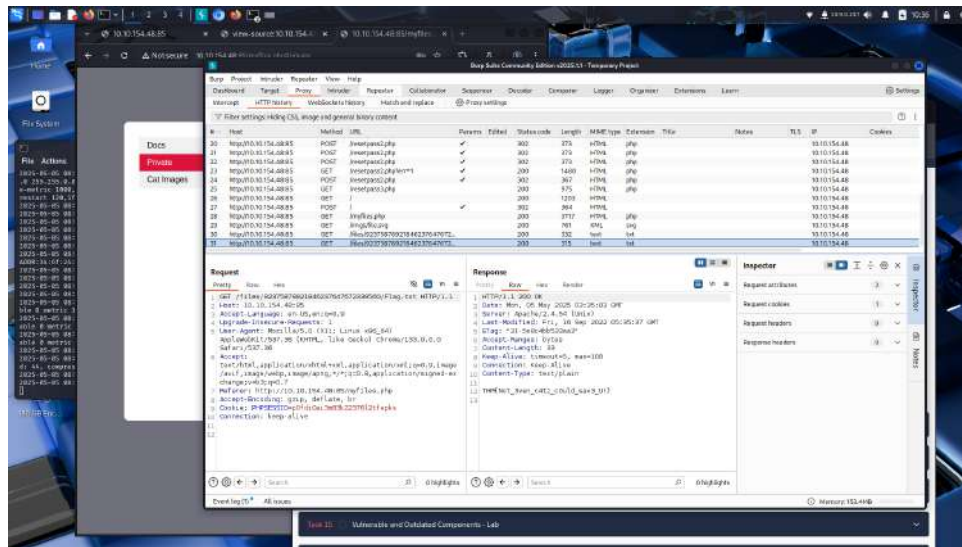
Figure 2: Gambar berhasil mendapatkan flag pada task Cryptographic Failures

## 4 Insecure Design

### 4.1 Deskripsi

Insecure Design mencakup kelemahan dalam desain aplikasi yang memungkinkan eksploitasi oleh penyerang.

1. Akses aplikasi dan coba reset password untuk pengguna **joseph**.
2. Gunakan metode brute force untuk menebak pertanyaan keamanan atau parameter lainnya.

$$\text{THM}\{\text{Not\_3ven\_c4tz\_c0uld\_sav3\_U!}\}$$


## 5 Security Misconfiguration

Security Misconfiguration terjadi ketika pengaturan keamanan tidak dikonfigurasi dengan benar, memungkinkan akses tidak sah atau informasi sensitif terekspos.

1. Akses konsol Werkzeug di `http://MACHINE_IP:86/console`.
2. Jalankan kode Python untuk membaca file sistem:

```
1 import os print(os.popen("ls -l").read())
2
```

## 5.3 Flag

THM{Just\_a\_tiny\_misconfiguration}

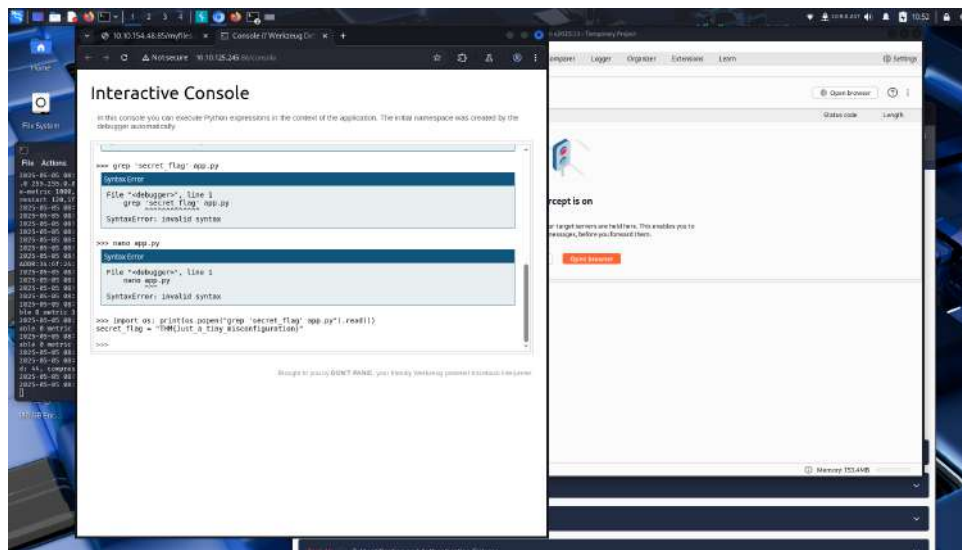


Figure 4: Gambar berhasil mendapatkan flag pada task Security Misconfiguration

## 6 Vulnerable and Outdated Components

### 6.1 Deskripsi

Kerentanan ini muncul ketika aplikasi menggunakan pustaka atau komponen yang usang atau rentan. Komponen seperti framework, pustaka, dan server environment jika tidak diperbarui dapat dieksploitasi oleh penyerang.

### 6.2 Langkah-langkah Exploit Lab

1. Deploy lab dari TryHackMe dan buka alamat IP mesin `http://MACHINE_IP:84/`.
2. Gunakan alat seperti `exploit.db` untuk mencari eksploitasi versi khusus.
3. Lakukan search relevan seperti `bookstore` setelah ditemukan versi yang dapat digunakan seperti (Online Book Store 1.0 Unauthenticated Remote Code Execution) yang diketahui rentan terhadap Remote Code Execution (RCE) CVE-2017-5638.
4. Cari exploit publik untuk kerentanan tersebut, misalnya dari Exploit-DB atau GitHub:

```
1 python 47887.py
```

5. Jalankan exploit menggunakan Python atau Burp Suite repeater. Contoh (jika menggunakan script Python):

```
1 python 47887.py http://MACHINE_IP:84 whoami
```

6. Jika berhasil, kamu akan mendapat respons command shell dari server target.

7. Jalankan perintah untuk mencari flag, misalnya:

```
1 cat /home/flag.txt
```

## 6.3 Flag

THM{But\_its\_n0t\_my\_f4ult!}

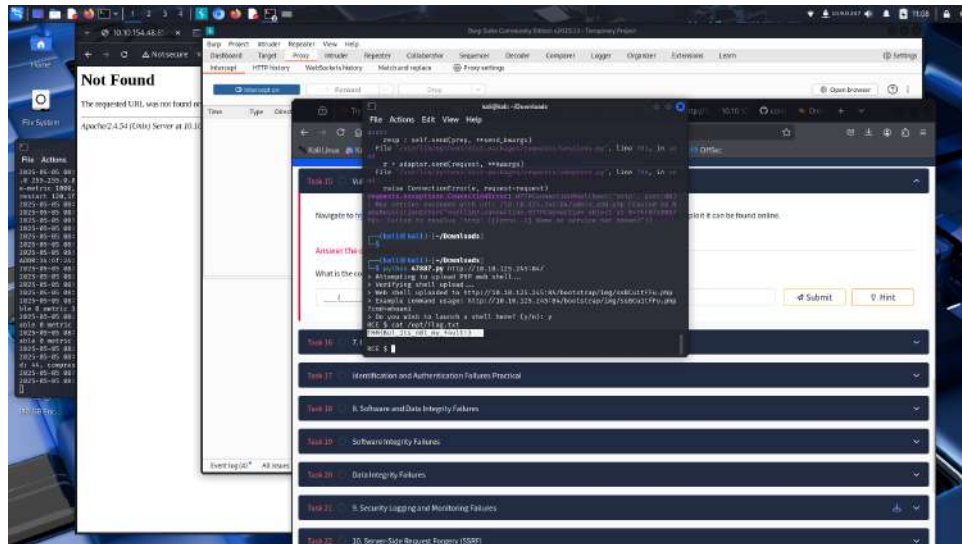


Figure 5: Gambar berhasil mendapatkan flag pada task Vulnerable and Outdated Components

## 7 Identification and Authentication Failures

**Deskripsi:** Kegagalan identifikasi dan autentikasi terjadi ketika aplikasi tidak mengelola kredensial secara aman, termasuk kata sandi lemah, penyimpanan yang tidak aman, dan tidak adanya MFA.

**Langkah-langkah:**

- Deploy lab Tryhackme dan buka alamat IP mesin `http://MACHINE_IP:8088`)
- daftar menggunakan nama "darren" dan "arthur".
- jika berhasil maka bendera akan ditemukan di kedua akun saat dibuka

Flag Darren: `fe86079416a21a3c99937fea8874b667`

Flag Arthur: `d9ac0f7db4fda460ac3edeb75d75e16e`

## 8 Data Integrity Failures

### Deskripsi

Data Integrity Failures terjadi ketika data yang dapat diubah oleh klien (misalnya cookie) dipercaya secara berlebihan oleh server tanpa verifikasi integritas. Contoh umum adalah pemalsuan atau manipulasi JWT (JSON Web Token) tanpa validasi tanda tangan yang benar, memungkinkan penyerang mengakses informasi yang tidak seharusnya.

## Studi Kasus: JWT None Algorithm Vulnerability

1. Akses aplikasi web pada `http://MACHINE_IP:8089`.
2. Lakukan login menggunakan kredensial:
  - Username: `guest`
  - Password: `guest`
3. Setelah berhasil login, buka Developer Tools (F12), lalu navigasikan ke tab **Storage** atau **Application** tergantung browser yang digunakan.
4. Temukan cookie dengan nama: `jwt-session`
5. Salin nilai token JWT. JWT terdiri dari tiga bagian yang dipisahkan oleh tanda titik (`.`), yaitu:
  - Header
  - Payload
  - Signature
6. Decode bagian **header** dan **payload** menggunakan Base64 decoder online atau secara manual. Header asli terlihat seperti:

```
1 {  
2   "typ": "JWT",  
3   "alg": "HS256"  
4 }
```

7. Modifikasi header menjadi:

```
1 {  
2   "typ": "JWT",  
3   "alg": "none"  
4 }
```

8. Modifikasi payload menjadi:

```
1 {  
2   "username": "admin"  
3 }
```

9. Encode ulang header dan payload ke dalam Base64 (tanpa padding karakter `"="` jika tidak diperlukan) dan gabungkan keduanya dengan titik (`.`) di antara keduanya dan **hilangkan bagian signature**, sehingga JWT menjadi:

`<base64(header)>.<base64(payload)>.`

Contoh hasil manipulasi token:

`eyJ0eXAiOiJKV1QiLCJhbGciOiJIub251In0.eyJ1c2VybWVtZSI6ImFkbWluIn0.`

10. Gantikan nilai cookie `jwt-session` dengan JWT yang telah dimodifikasi.



11. Refresh halaman. Jika berhasil, aplikasi akan mengenali pengguna sebagai admin.
12. Bendera yang ditampilkan:

THM{Dont\_take\_cookies\_from\_strangers}

## Flag

THM{Dont\_take\_cookies\_from\_strangers}

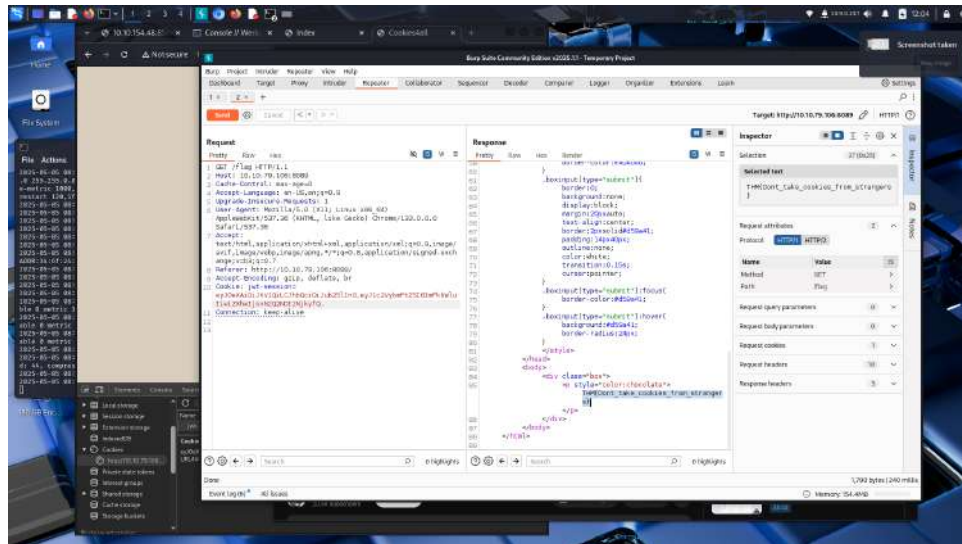


Figure 6: Gambar berhasil mendapatkan flag pada task Data Integrity Failures

## Kesimpulan

Kerentanan ini terjadi karena aplikasi mempercayai JWT dengan algoritma none tanpa memverifikasi tanda tangan. Implementasi JWT yang aman harus selalu memverifikasi tanda tangan menggunakan secret key dan tidak mengizinkan algoritma none, bahkan jika dikirim oleh klien.

## 9 Security Logging and Monitoring Failures

**Deskripsi:** Ketika aplikasi gagal mencatat dan memantau aktivitas penting, hal ini memperlambat respons terhadap insiden keamanan.

### Langkah-langkah:

- Download Task Files yang tersedia di soal
- Audit pencatatan login, logout, dan kesalahan autentikasi.
- Pastikan monitoring aktif terhadap log.
- Identifikasi IP address penyerang

1 IP address penyerang: 49.99.13.16

2



- Analisis serangan yang digunakan oleh penyerang

```
1 penyerang sedang melakukan serangan brute force
2
```

- Verifikasi kebijakan retensi dan keamanan log.

## Server-Side Request Forgery (SSRF)

### Deskripsi

SSRF terjadi ketika aplikasi web yang rentan dapat dipaksa oleh penyerang untuk membuat permintaan HTTP ke sumber daya internal atau eksternal, menggunakan kredensial atau akses yang dimiliki server aplikasi. Ini memungkinkan penyerang mengakses sistem internal, metadata cloud, atau mengeksfiltrasi data sensitif seperti API key.

### Langkah-langkah Eksploitasi SSRF

1. Akses aplikasi web di `http://MACHINE_IP:8087`.
2. Jelajahi halaman utama dan klik tombol "Unduh Lanjutkan".
3. Perhatikan bahwa parameter `server` mengarah ke `secure-file-storage.com`.
4. Ubah parameter `server` menjadi alamat dari AttackBox.
5. Di AttackBox, buka terminal dan jalankan perintah berikut untuk menangkap permintaan masuk:

```
1 nc -lvp 80
2
```

6. Reload halaman yang menggunakan parameter server yang telah dimodifikasi.
7. Cek output dari Netcat untuk melihat apakah permintaan diterima dan apakah terdapat **API key** yang terekspos.
8. Catat flag dari respon permintaan:

```
THM{Hello_Im_just_an_API_key}
```

### Flag

```
THM{Hello_Im_just_an_API_key}
```

### Catatan Tambahan

Terdapat area admin yang hanya dapat diakses oleh `localhost`. Dengan menggunakan SSRF, kita dapat mengarahkan permintaan ke `127.0.0.1` untuk mencoba mengakses area tersebut secara tidak langsung.