Challenge Write-up: Professor's Vault

Challenge Name: Professor's Vault **Category:** Web Exploitation (JWT) **Flag:** `SECE{M0n3y_H31st_15_0n!}` **Author:** Manus AI **Date:** January 30, 2026

1. Introduction

The "Professor's Vault" challenge presented a web application themed around the popular series *La Casa de Papel* (Money Heist). The goal was to gain access to the vault, which was restricted to the "Professor" role. The initial hint, "Forged What ..?", strongly suggested a vulnerability related to token forgery, specifically **JSON Web Tokens (JWT)**, a common pattern in modern web challenges.

2. Initial Reconnaissance and JWT Discovery

The application provided two main entry points: "JOIN THE HEIST" (registration) and "ENTER THE MINT" (login).

1. A new user, `testuser123`, was registered and logged in.
2. Upon successful login, the user was redirected to a profile page (`/profile`) which displayed the user's identity and role:
   - **Identity:** `testuser123`
   - **Role:** `hostage`
3. The profile page also contained a clear message: "❌ Vault access restricted to The Professor only."

Inspecting the browser's cookies revealed a session token in the form of a JWT:

```
token=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VybmFtZSI6InRlc3R1c2VyMTIzIiw
```

Decoding the payload section of the token revealed the following claims:

Claim	Value	Description
`username`	`testuser123`	The user's identity.
`role`	`hostage`	The user's access level.
`secret_plan`	`X0YGUAAwQPJ4chY9LPcIcgI8aQY8zelI9`	An encoded string, likely a secret or key.
`iat`	`1769764658`	Issued At timestamp.

The token's header indicated it was signed using the **HS256** (HMAC with SHA-256) algorithm, which requires a secret key for verification.

3. Vulnerability: Weak JWT Secret

The presence of the `secret_plan` claim, which appeared to be a Base64-encoded string, and the challenge's hint "Forged What ..?" suggested a vulnerability related to the JWT's secret key.

The initial attempt was to use the **"none" algorithm vulnerability**, but this was unsuccessful, indicating the server correctly validated the token's signature.

The next step was to attempt to **crack the secret key** using the known token and a wordlist. Given the theme of the challenge, a custom wordlist was created containing common terms from the *La Casa de Papel* series.

A Python script was used to test the signature against the wordlist:

```
# Snippet from the cracking script
secrets = [
    "professor", "theprofessor", "heist", "mint", "vault",
    "X0YGUAAwQPJ4chY9LPcIcgI8aQY8zelI9", "secret", "password", "123456",
    "lacasadepapel", "bella ciao", "dali", # ... and other terms
]
# ... (HMAC-SHA256 verification logic)
```

The script successfully identified the secret key: **`dali`**.

4. Exploitation: Forging the Professor's Token

With the secret key, a new JWT could be forged with elevated privileges. The goal was to change the `role` claim from `hostage` to the required **`Professor`** and the `username` to **`The Professor`** (matching the required identity from the profile page message).

The forged payload was constructed as follows:

Claim	Forged Value
`username`	`The Professor`
`role`	`Professor`
`secret_plan`	`X0YGUAAwQPJ4chY9LPcIcgI8aQY8zelI9`
`iat`	`1769764658`

This payload was then signed with the discovered secret key, `dali`, resulting in the final forged JWT:

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VybmFtZSI6IlRoZSBQcm9mZXNzb3IiLCJyb
```

5. Conclusion and Flag Retrieval

The forged JWT was set as the session cookie and the profile page was reloaded. The profile page now displayed the correct identity and role:

- **Identity:** `The Professor`
- **Role:** `Professor`

A new button, **"🏦 ACCESS THE VAULT"**, appeared. Clicking this button led to the final page, which contained the flag:

> ***CONGRATULATIONS!***
>
> ***`SECE{M0n3y_H31st_15_0n!}`***

This challenge serves as a practical example of a **JWT Weak Secret Key** vulnerability, highlighting the critical importance of using strong, high-entropy secrets for signing security tokens [1].

References

[1] OWASP Foundation. *JSON Web Token Cheat Sheet*. Available at: https://cheatsheetseries.owasp.org/cheatsheets/JSON_Web_Token_Cheat_Sheet.html

Not secure 52.205.96.89/vault

THE PROFESSOR'S VAULT

"Welcome, Professor. The gold is yours."

CONGRATULATIONS!

SECE{M0n3y_H31st_15_0n!}

You have successfully accessed The Professor's Vault!

The Master Plan (Encrypted)

```
XDYGQAAwQFJ4chY9LFcIoqI4aQY8zel19
```

The Professor always rotates his plans... and encodes them twice.
First rotation (ROT3), then encoding (Base64)
Reverse the process to reveal the true treasure...

HEIST STATUS: COMPLETE

- Mint infiltrated
- Security bypassed
- Vault accessed