CloudSEK Research Team CTF - Write-Up

Name: V Anand

Email: anandrao843@gmail.com
Phone: 8105859451

Resume Link:
https://docs.google.com/document/d/10Q4wnYd6p9XJZr3HH9nlaf7YRG-zO5soGLwnoAjLxLw/edit?usp=sharing

Date: 07 December 2025

Challenge 1 – Nitro Automation

Category: Web Exploitation / Automation

Difficulty: Medium

Flag: **ClOuDsEk_ReSeArCH_tEaM_CTF_2025{ab03730caf95ef90a440629bf12228d4}**

Overview:

This challenge required automating a time-bound transformation of a randomly generated string. The task must be completed before the timer expires, making manual execution impossible.

Reconnaissance:

- GET /task returns a random string inside HTML.

- POST /submit expects CSK__<Base64(reversed_string)>__2025.

Transformation Steps:

1. Reverse the string

2. Base64 encode

3. Wrap with CSK__{payload}__2025

Python Automation was used to fetch → process → submit rapidly.

Final Flag:

ClOuDsEk_ReSeArCH_tEaM_CTF_2025{ab03730caf95ef90a440629bf12228d4}

Challenge 2 – Feedback Portal (XXE)

Category: Web Security (XML External Entity Injection)

Difficulty: Medium

Flag: **ClOuDsEk_ReSeArCH_tEaM_CTF_2025{b3e0b6d2f1c1a2b4d5e6f71829384756}**

Overview:

Client-side JS generated XML using unsanitized user input. XXE injection allowed file-reading on server.

Exploit:

A malicious XML payload defined an entity pointing to file:///flag.txt.

Using:

```
<!DOCTYPE xxe [ <!ENTITY flag SYSTEM "file:///flag.txt"> ]>

<name>&flag;</name>
```

Server responded with the file contents.

Final Flag:

ClOuDsEk_ReSeArCH_tEaM_CTF_2025{b3e0b6d2f1c1a2b4d5e6f71829384756}

Challenge 3 – Triangle: Break the Trinity

Category: Web Security / Authentication Bypass

Difficulty: Medium

Flag: **ClOuDsEk_ReSeArCH_tEaM_CTF_2025{474a30a63ef1f14e252dc0922f811b16}**

Overview:

A multi-step authentication system with username, password, and three OTPs. A hidden developer comment revealed unremoved .bak files.

Key Findings:

- login.php.bak revealed credentials admin/admin.

- google2fa.php.bak showed a critical vulnerability:

```
if (otp_generated == user_input)  // loose comparison!
```

This allowed PHP type juggling, enabling bypass using:

```
otp1=true, otp2=true, otp3=true
```

Final Flag:

ClOuDsEk_ReSeArCH_tEaM_CTF_2025{474a30a63ef1f14e252dc0922f811b16}

Challenge 4 – Strike Bank (APK + JWT Forgery)

Category: Mobile Security / Web Exploitation

Difficulty: Medium

Flag: **ClOuDsEk_ReSeArCH_tEaM_CTF_2025{ccf62117a030691b1ac7013fca4fb685}**

Overview:

The APK's BeVigil scan exposed:

- Username: tuhin1729

- Password: 123456

- JWT secret (Base64): c3RyIWszYjRua0AxMDA5JXN1cDNyIXMzY3IzNw==

Decoded: str!k3b4nk@1009%sup3r!s3cr37

Logged into employee portal → captured JWT → forged admin JWT due to insecure HS256 implementation.

Final Flag:

ClOuDsEk_ReSeArCH_tEaM_CTF_2025{ccf62117a030691b1ac7013fca4fb685}