Write-Up JOINTS UGM

ganyuwangy

4leph

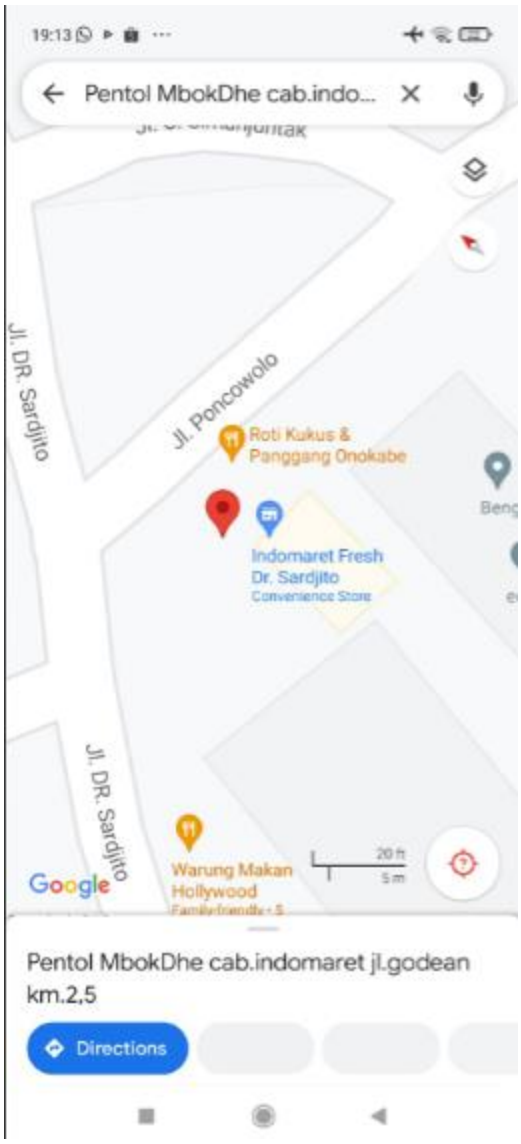
DAFTAR ISI

OSINT

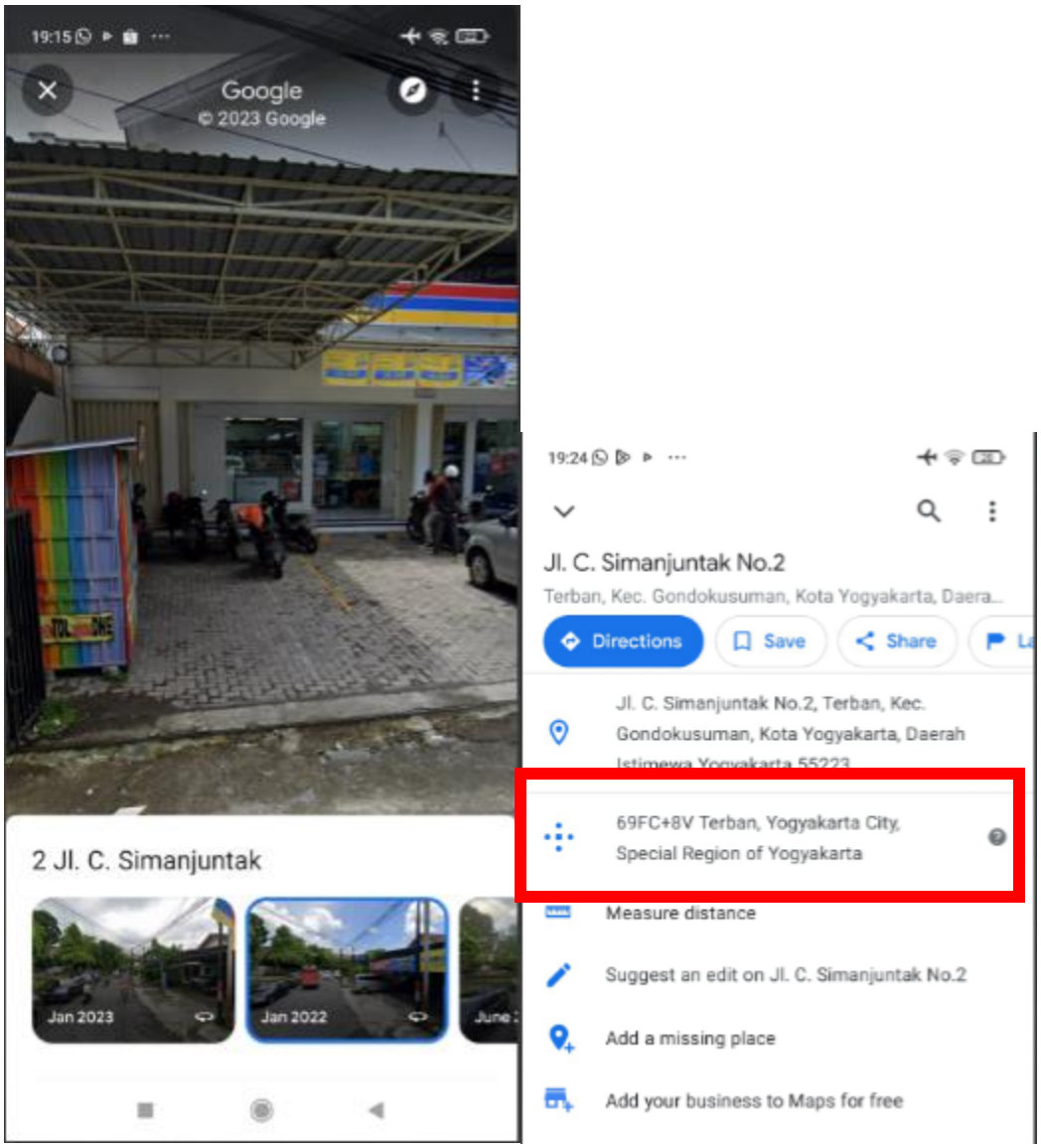
- whereIsThis(100 pts)

Diberikan file gambar where is this.jpg.

Dengan clue “Pentol MbokDhe” lokasi dapat ditemukan dengan mudah di google maps.



Untuk memastikan lokasi yang pas pada soal, maka digunakan fitur street view 360



Pada google maps tertera Pluscode: 69FC+8V dan Kelurahan: Terban. Jawaban dalam huruf kapital.

FLAG

JCTF2023{69FC+8V_TERBAN}

CRYPTO

- Easy CBC (100 pts)

Diberikan file python challenge.py dan file gambar out.bmp.

```
└──$ cat challenge.py

# !pip install certifi==2021.10.8
```

```
# !pip install cffi==1.15.0
# !pip install cryptography==36.0.2
# !pip install Pillow==9.0.1
# !pip install wincertstore==0.2
import os
from cryptography.hazmat.primitives.ciphers import Cipher, algorithms, modes
from cryptography.hazmat.backends import default_backend
import PIL.Image as Image

class CBCEncryption:
    def __init__(self, key, iv):
        self.cipher = Cipher(algorithms.AES(key), modes.CBC(iv), backend=default_backend())
        self.encryptor = self.cipher.encryptor()

    def encrypt(self, image):
        return self.encryptor.update(image)

    def finalize_encrypt(self):
        return self.encryptor.finalize()

def EncryptImage(encryption, image, output):
    output = output + '.bmp'
    image = Image.open(image)
    image.save('temp.bmp')
    with open('temp.bmp', 'rb') as reader:
        with open(output, 'wb') as writer:
            image_data = reader.read()
            header, body = image_data[:54], image_data[54:]
            body += b'\x35' * (16 -(len(body) % 16))
```

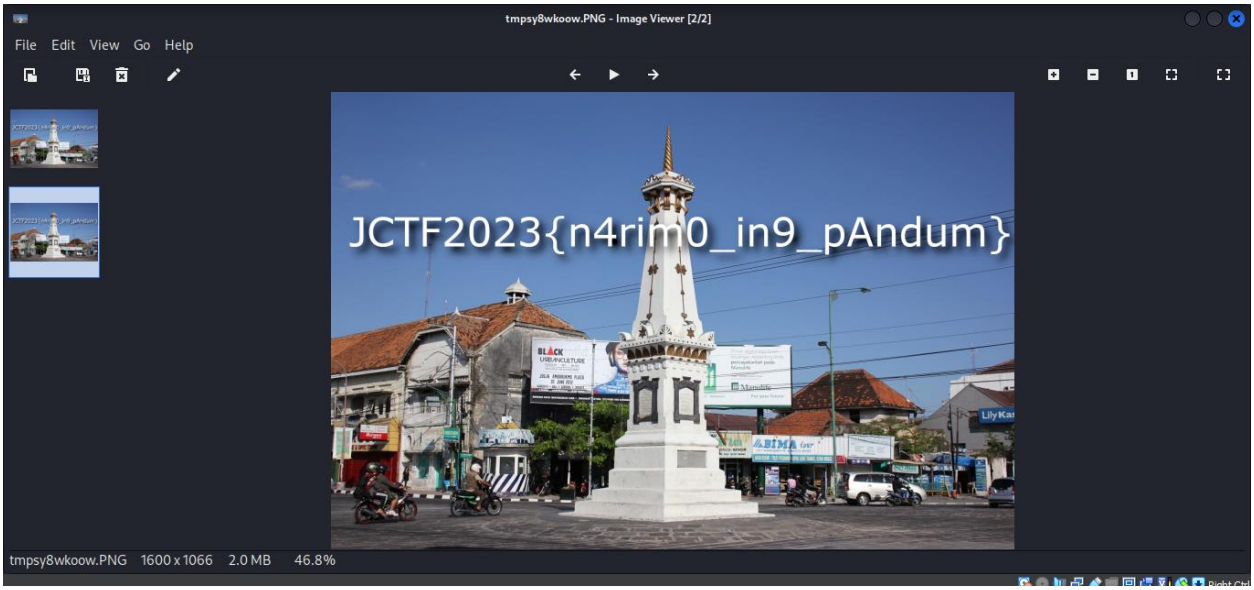
```
        body = encryption.encrypt(body) + encryption.finalize_encrypt()
        writer.write(header + body)
        writer.close()
        reader.close()
    os.remove('temp.bmp')


def main():
    key = b'JOINTSCTF2023'
    key = key.ljust(32, b'\x35')

    iv = key[:16]
    iv = bytearray(iv)
    for i in range(16):
        iv[i] = iv[i] ^ 0x35
    iv = bytes(iv)

    AesCbc = CBCEncryption(key, iv)
    EncryptImage(encryption=AesCbc, image='flag.jpg', output='out')


if __name__ == '__main__':
```

Dengan google.fu, ditemukan skrip https://github.com/user163/image-encryption untuk mendekripsi file gambar dari AES-CBC. Berikut modifikasi source code untuk mendapat flag.

```
┌──$ cat script.py
from cryptography.hazmat.primitives.ciphers import Cipher, algorithms, modes
from cryptography.hazmat.backends import default_backend
import PIL.Image as Image


key = b'JOINTSCTF2023'
```

```
key = key.ljust(32, b'\x35')


iv = key[:16]
iv = bytearray(iv)
for i in range(16):
    iv[i] = iv[i] ^ 0x35
iv = bytes(iv)


cipher = Cipher(algorithms.AES(key), modes.CBC(iv), backend=default_backend())


with open('out.bmp', 'rb') as reader:
    image_data = reader.read()
    encrypted_data = image_data[54:]
    decryptor = cipher.decryptor()
    decrypted_data = decryptor.update(encrypted_data) + decryptor.finalize()


with open('decrypted.bmp', 'wb') as writer:
    header = image_data[:54]
    writer.write(header + decrypted_data)


Image.open('decrypted.bmp').show()
```

Berikut gambar yang berhasil didekripsi



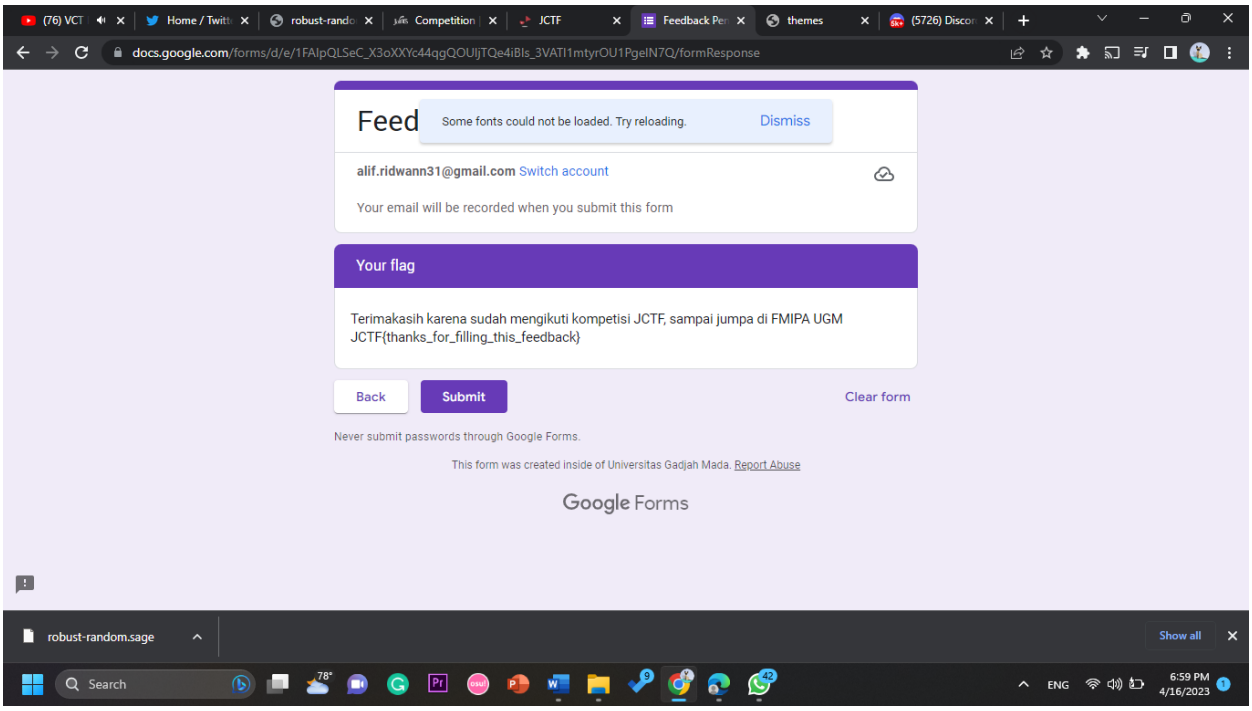
FLAG

JCTF2023{n4rim0_in9_pAndum}

MISC

- FEEDBACK (100 pts)

Dengan mengisi feedback pada google form, flag didapatkan dengan mudah.



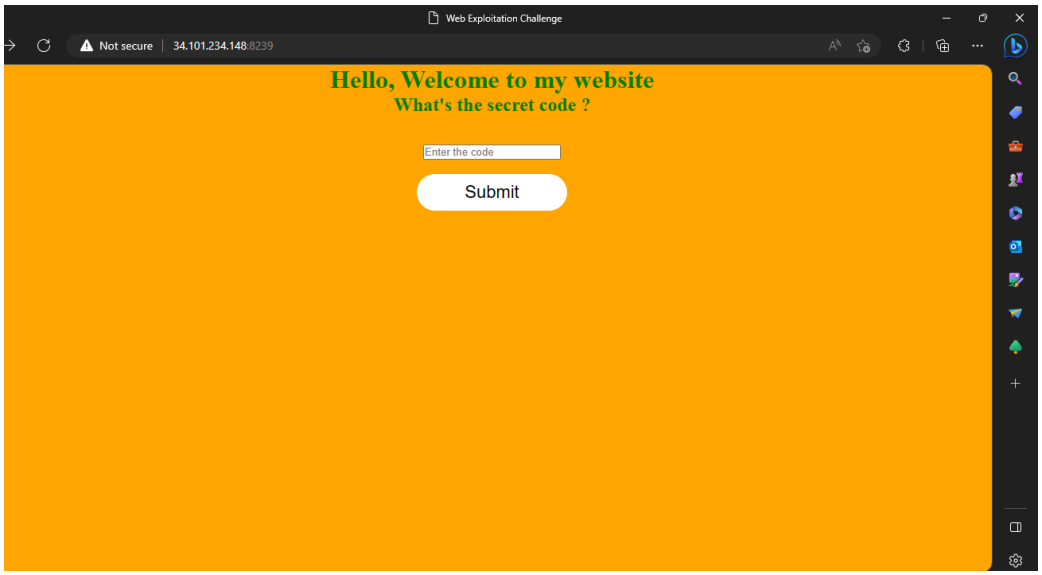
FLAG

JCTF{thanks_for_filling_this_feedback}

WEB

- Vision (100 pts)

Diberikan IP Address dan port 34.101.234.148:8239.



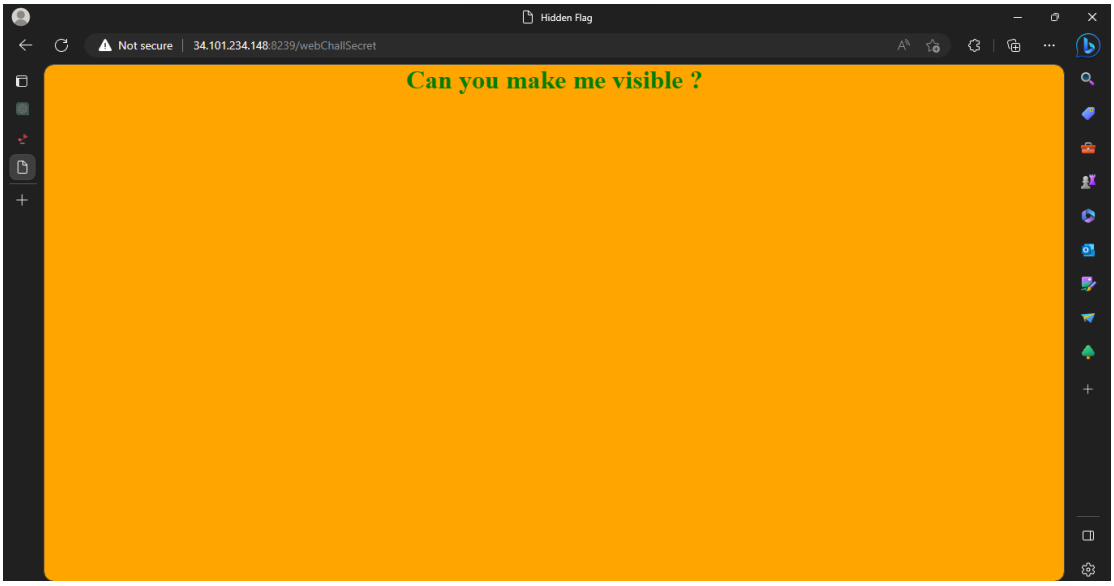
Untuk mencari informasi, diawali mengecek page source dari web tersebut.

```
└──$ curl 34.101.234.148:8239


<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
    <title>Web Exploitation Challenge</title>
    <link rel="stylesheet" href="views/style.css">
</head>
<body>
    <h1>Hello, Welcome to my website</h1>
    <h2>What's the secret code ?</h2> <br> <br>
    <input type="text" placeholder="Enter the code" id="userInput"> <br> <br>
    <button type="submit" class="btn" onclick="inputCode()">Submit</button>
    <h1 id="message"> </h1>
    <img src="image/clue.png" alt="clue"/>
    <div class="popup" id="popup">
        <h2>Congratulation</h2>
        <img src="image/flag.png"> <br><br>
        <a href="/webChallSecret"> Next </a>
    </div>
    <style>
        body{background-color: orange; text-align: center; color: green;}
    </style>

    <script>

        let popup = document.getElementById("popup");
        function inputCode() {
            let input= document.getElementById("userInput").value;
            let message = document.querySelector("#message")
            if(input == "mantapujiwa"){
                popup.classList.add("showPopup");
                message.innerHTML = "Your code is right!";
            }
```

```
            else{
                message.innerHTML = "Your code is wrong!";
            }
        }
    </script>
</body>
</html>
```

Ditemukan kunci dari input yaitu **mantapujiwa**, dan masuk ke halaman selanjutnya. Ditemukan gambar pada source file css, berupa potongan puzzle gambar yang apabila disatukan menjadi flag. Gambar terlihat sangat jelas, flag mudah didapat.



FLAG

JCTF2023{s0_e4sy_w3b_3xPl0itation}