

Industrial Intrusion CTF

Task 3 Breach

Getting introduced to the ctf, we are presented with our ip and hint. (Note that the ip may change throughout the writeup due to the machine expiring.)

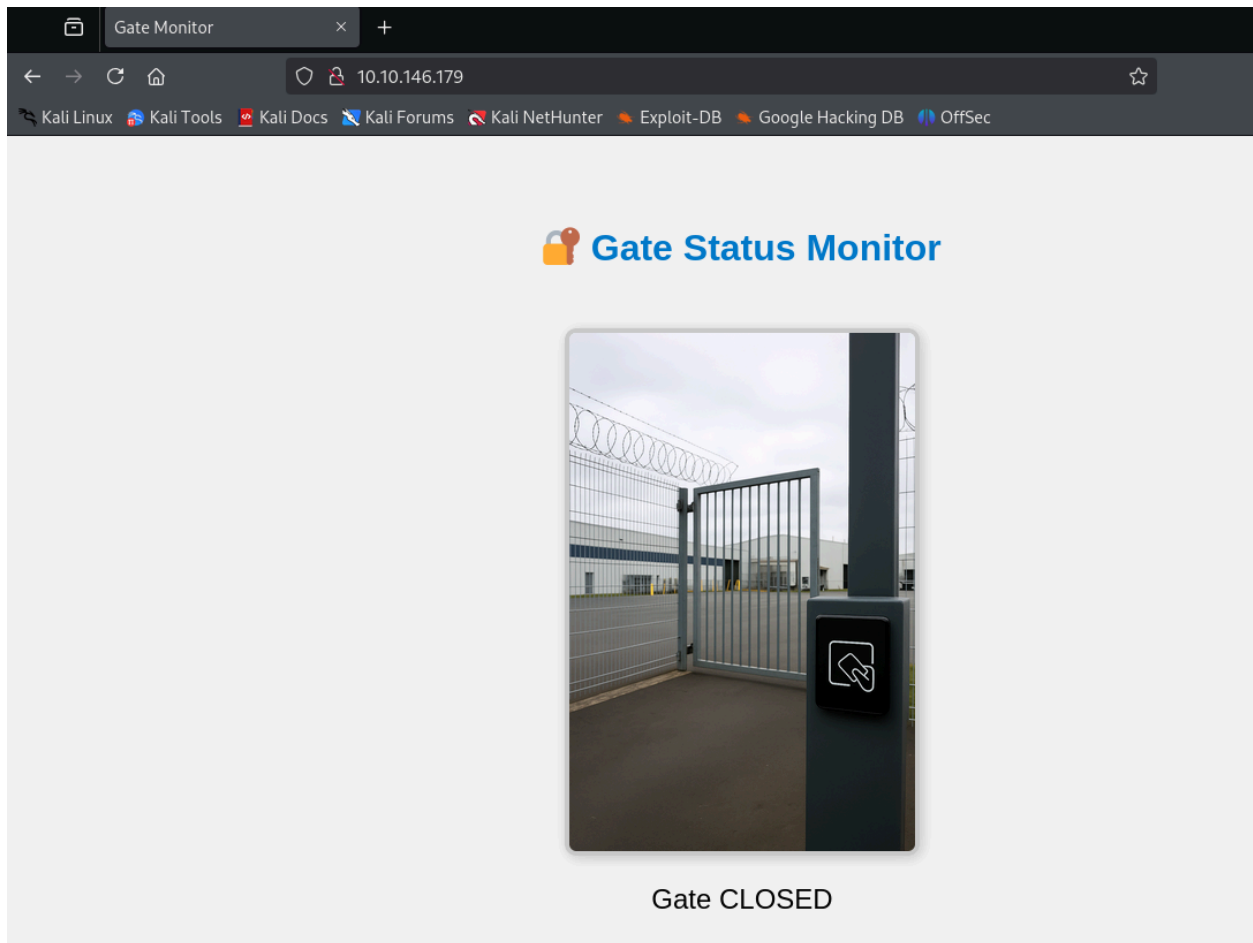
The screenshot displays the 'Active machines information' section at the top, showing the IP Address: 10.10.146.179. Below this, the 'Set up your virtual environment' section provides instructions on starting the AttackBox and Task Machines. The 'Attacker machine' status is 'Off' with a 'Start Kali Linux' button, and the 'Target machine' status is 'On'. A diagram shows a red box (Attacker) connected to a grey box (Target) by a green dashed arrow.

The main challenge area is titled 'EASY' and shows '0 Points'. It features a large orange gear icon in the center of a dark screen. To the right, a text box contains the following information:

- #1**
- This engagement aims to find a way to open the gate by bypassing the badge authentication system.
- The control infrastructure may hold a weakness: Dig in, explore, and see if you have what it takes to exploit it.
- Be sure to check all the open ports, you never know which one might be your way in!

Below the challenge area, a red text prompt says 'Answer the questions below'. A question asks 'What is the flag?'. The answer format is given as '***{*** _ _ _ _ _ _ _ _ }'. A 'Submit' button is located at the bottom right.

Taking a look at the webpage, it looks like we are locked out of the system.

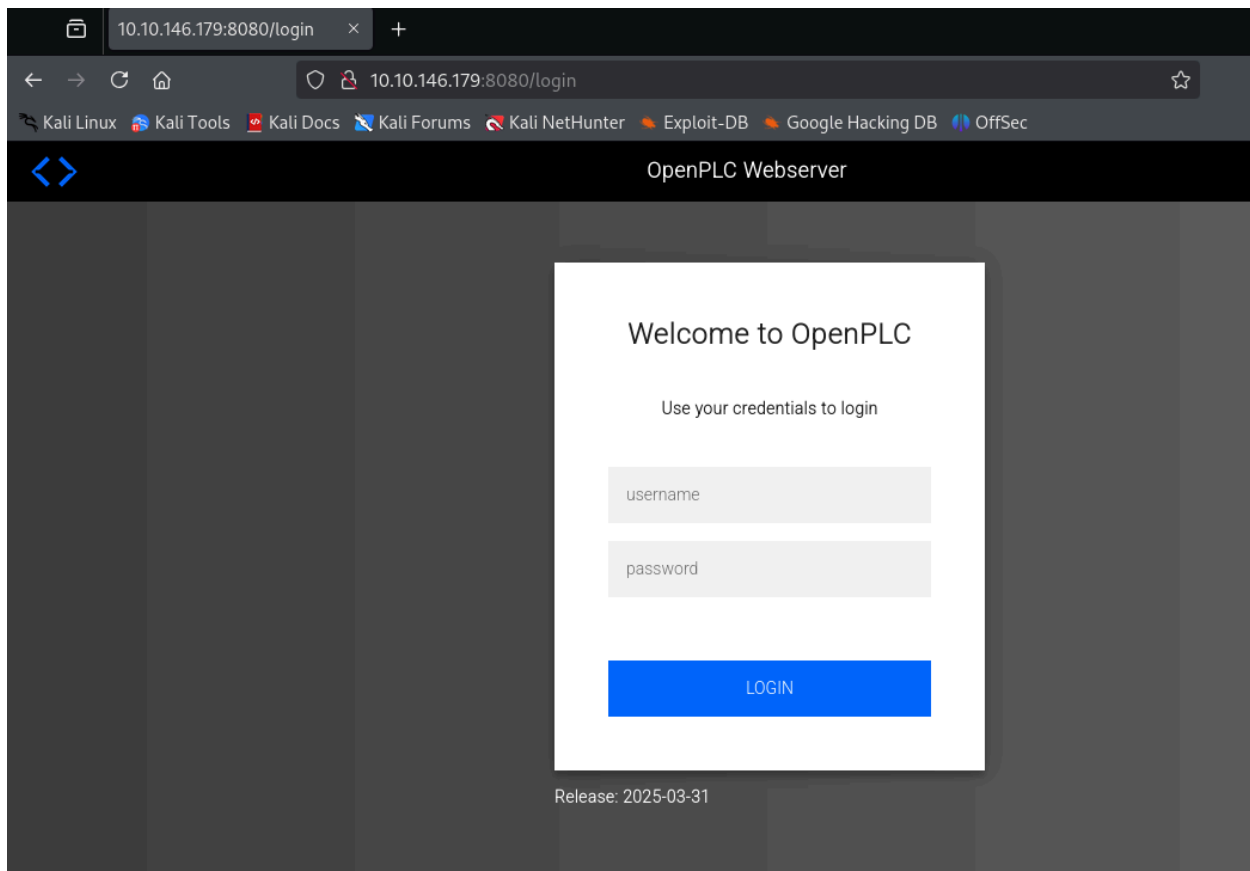


Running an nmap scan using `nmap 10.10.146.179 -sS -Pn -T5 -p-` to quickly see all open ports, we are presented with the following ports:

```
(kali㉿kali)-[~/Desktop]
$ nmap 10.10.146.179 -sS -Pn -T5 -p-
Starting Nmap 7.95 ( https://nmap.org ) at 2025-06-25 20:50 EDT
Warning: 10.10.146.179 giving up on port because retransmission cap hit (2).
Nmap scan report for 10.10.146.179
Host is up (0.10s latency).
Not shown: 65528 closed tcp ports (reset)
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
102/tcp   open  iso-tsap
502/tcp   open  mbap
1880/tcp  open  vsat-control
8080/tcp  open  http-proxy
44818/tcp open  EtherNetIP-2

Nmap done: 1 IP address (1 host up) scanned in 104.63 seconds
```

We can see that port 80 gives is a closed gate, so I then check port 8080 to see what's there and we are presented with a login page.

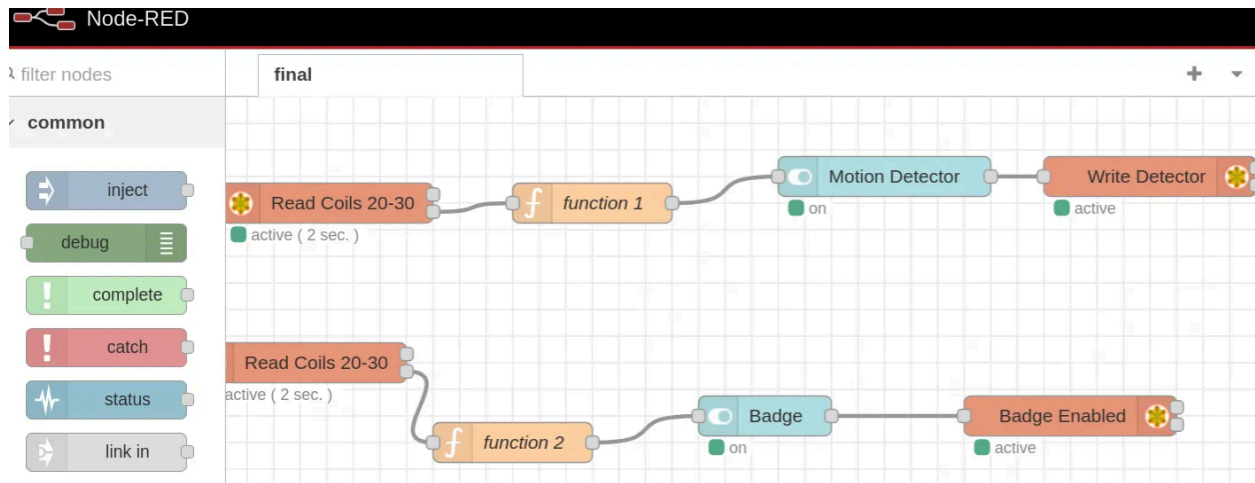


I then follow up with another nmap scan to gather more information.

I can see that port **1880** has html contents in it

trusion CTF

Port 1880 was hosting Node-RED, which showed us automation workflows.



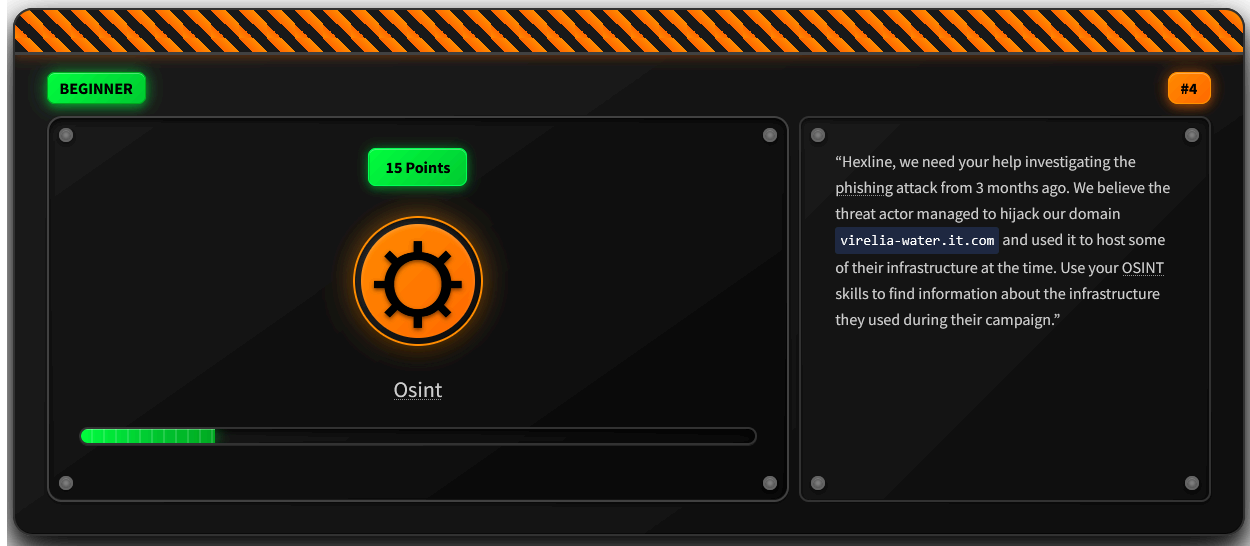
At this point, I had to go to <http://10.10.146.179:1880/ui> where there was a UI switch to disable both motion detector and badge. This allowed me to go back to the gate status monitor and see that it was opened and the flag provided.

What is the flag?

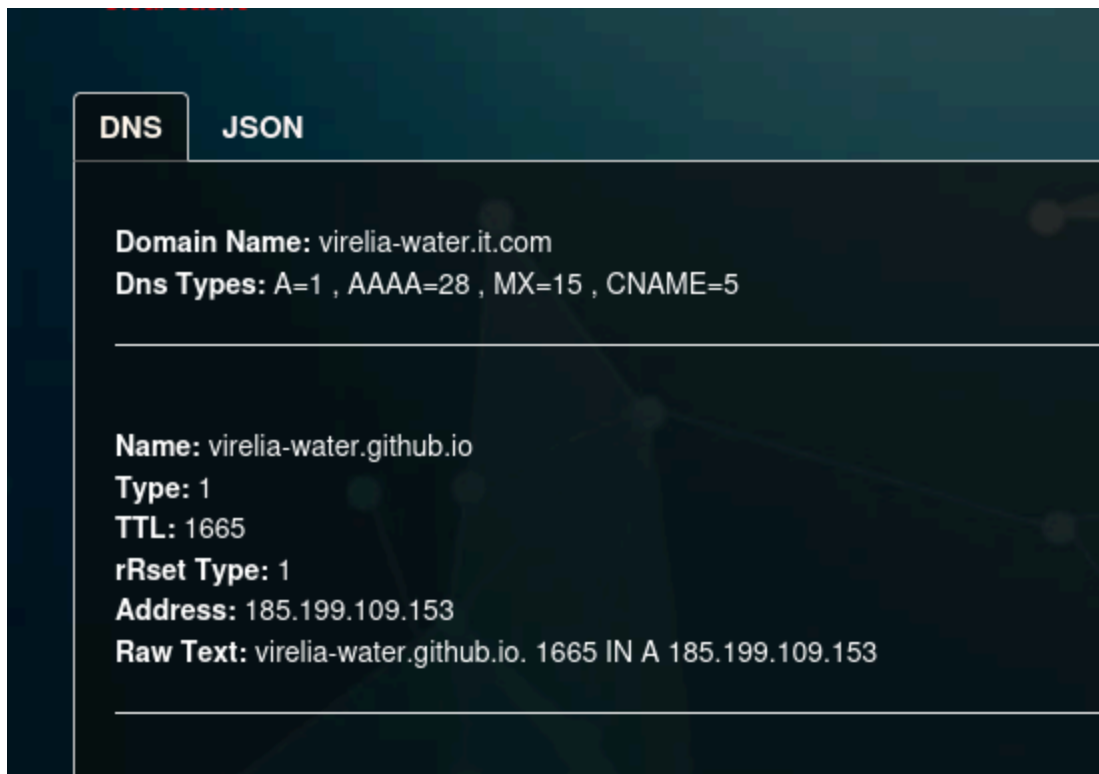
THM{s4v3_th3_d4t3_27_jun3}

✓ Correct Answer

Task 5 OSINT 1



DNS lookup shows that it is linked to a github pages

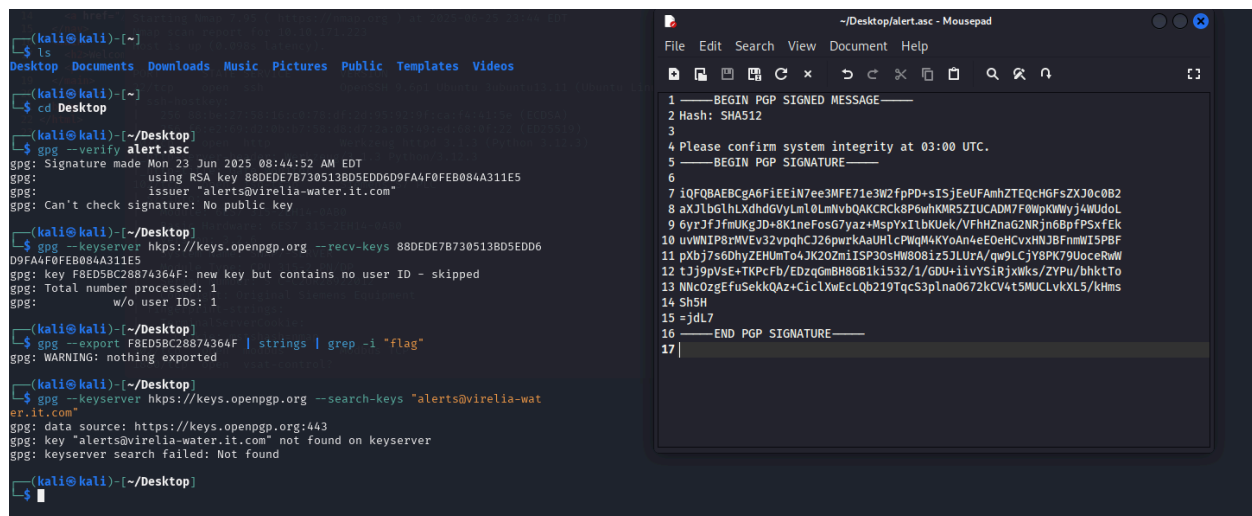


Scanning the github, I found that there was a file that has been removed during an investigation according to push history

```
mail-archives/ot-alerts/2025-06.html
2 + <html lang="en">
3 + <head>
4 +   <meta charset="UTF-8">
5 +   <meta name="robots" content="index, follow">
6 +   <title>OT Alerts Exceptions - June 2025</title>
7 +   <link rel="stylesheet" href="/styles.css">
8 + </head>
9 + <body>
10 +   <header><h1>OT Alerts Exception Report - June 2025</h1></header>
11 +   <nav>
12 +     <a href="/">Home</a>
13 +     <a href="/mail-archives/">Archives Home</a>
14 +     <a href="/policies/">Compliance Policies</a>
15 +   </nav>
16 +   <main>
17 +     <p>This page lists <em>exceptional</em> OT-Alert messages for June 2025 only. Routine alerts have been redacted.</p>
18 +     <div class="message">
19 +       <div class="hdr">
20 +         From: DarkPulse <alerts@virelia-water.it.com>;<br>
21 +         Date: Mon, 15 Jun 2025 02:15:00 +0000<br>
22 +         Subject: Scheduled OT Calibration
23 +       </div>
24 +       <pre>
25 + -----BEGIN PGP SIGNED MESSAGE-----
26 + Hash: SHA512
27 +
28 + Please confirm system integrity at 03:00 UTC.
29 + -----BEGIN PGP SIGNATURE-----
30 +
31 + iQFQBAEBCgA6FiEEiN7ee3MFE71e3W2fpPD+sISjEeUFamhZTEQcHGFsZXJ0c0B2
32 + aXJlbGhlLXdhdGVyLm10LmNvbQAKCRck8P6whKMR5ZIUcADM7F0wpKWwyj4wUdoL
33 + 6yrJfJfmUKgJD+8K1neFosG7yaz+MspyXIlbKUEk/VFhHZnaG2NRjn6BpfPSxfEk
34 + uvWNIP8rNVEv32vpqhCJ26pwrkAaUHLcPwqM4KY0An4eE0eHCvxHNJBfNmWI5PBF
35 + pXbj7s0DhyZEHUmTo4JK20ZmiISP30sHW80Bz5JLURa/qw9LCjY8PK70UoceRwW
36 + tJj9pVsE+TKPcFb/EDzqGmBH8GB1ki532/1/GDU+iiVYSiRjxwks/ZYPu/bhktTo
37 + NNcOzgEfuSekKQAz+CicLxwEcLQb219TqcS3pLna0672kCV4t5MUCLVxXL5/kHms
38 + Sh5H
39 + =jdL7
40 + -----END PGP SIGNATURE-----
41 +       </pre>
42 +     </div>
43 +   </main>
44 +   <footer>&copy; 2025 Virelia Water Control Facility</footer>
45 + </body>
46 + </html>
```

Since this was removed, I wanted to investigate the PGP Signature.

First I attempt to import the key to see if the flag may be somewhere in the key name.



```
(kali@kali)-[~]
$ ls
Desktop  Documents  Downloads  Music  Pictures  Public  Templates  Videos

(kali@kali)-[~]
$ cd Desktop

(kali@kali)-[~/Desktop]
$ gpg --verify alert.asc
gpg: Signature made Mon 23 Jun 2025 08:44:52 AM EDT
gpg:       using RSA key 88DEDE7B730513BD5EDD6D9FA4F0FEB084A311E5
gpg:       issuer "alerts@virelia-water.it.com"
gpg: Can't check signature: No public key

(kali@kali)-[~/Desktop]
$ gpg --keyserver htps://keys.openpgp.org --recv-keys 88DEDE7B730513BD5EDD6D9FA4F0FEB084A311E5
gpg: key F8ED5BC28874364F: new key but contains no user ID - skipped
gpg: Total number processed: 1
gpg:       w/o user IDs: 1

(kali@kali)-[~/Desktop]
$ gpg --export F8ED5BC28874364F | strings | grep -i "flag"
gpg: WARNING: nothing exported

(kali@kali)-[~/Desktop]
$ gpg --keyserver htps://keys.openpgp.org --search-keys "alerts@virelia-water.it.com"
gpg: data source: https://keys.openpgp.org:443
gpg: key "alerts@virelia-water.it.com" not found on keyserver
gpg: keyserver search failed: Not found

(kali@kali)-[~/Desktop]
$
```

```
~/Desktop/alert.asc - Mousepad
File Edit Search View Document Help

1 -----BEGIN PGP SIGNED MESSAGE-----
2 Hash: SHA512
3
4 Please confirm system integrity at 03:00 UTC.
5 -----BEGIN PGP SIGNATURE-----
6
7 iQFQBAEBCgA6FiEEIN7ee3MFE71e3W2fpPD+sISjEeUFamhZTEQcHGFSZXJ0c0B2
8 aXJlbGhlLXdhbGVyLm0LmNvbQAKCRCK8P6whKMRSZIUcADM7F0WpKWMyj4WUdoL
9 6yr3FjfmUKg3D+8K1neFosG7yaz+MspYxILbKUEk/VFhHZnaG2NRjn6BpfP5xfEk
10 uvWNTP8rMVEv32vpqhCJ26pwrkAaUHLcPWqM4KYoaAn4eEOeHcvxHNJ8FnmWISPF
11 pXbj7s6DhyZEHUmTo4JK2OZmiISP30sHW808iz5JLurA/qw9LCjY8PK79UoceRwW
12 tJj9pVsE+TKPcFb/EDzqGmBH8GB1kIS32/1/GDU+liivYSirjxwks/ZYPu/bhktTo
13 NNC0zgEfusEkkQAz+CicLXwEclQb219TqcS3plna0672kCV4t5MUCLvKXLS/kHms
14 ShSH
15 =jdl7
16 -----END PGP SIGNATURE-----
17
```

That led me to a dead end with a fake key and the email not giving me anything either.

Next, I take the signature and try to decrypt it using base64 decoding.

Simply enter your data then push the decode button.

i For encoded binaries (like images, documents, etc.) use the file upload form a little further down on this page.

☐ Decode each line separately (useful for when you have multiple entries).

< DECODE > Decodes your data into the area below.

[0x]P[0x][0x][0x]
◆[0x][0x](S[0x][0x]^m[0x][0x]hYLD[0x][0x]alerts@virelia-water.it.com◆
[0x][0x][0x][0x]◆[0x]>[0x]Q[0x]*[0x]P
wt2Xc[G]GqA[0x]cQ-\$+1Q/K[p]Pyl=j([0x]-[0x])
G4EeEv[0x]D[0x]Jf[0x][u]=J-(J[0x]y[0x]2pv[0x]<[0x]'G'u.w[0x][0x]5>+Js,c[0x]4[0x][0x]◆3([0x]p[0x]SKY:%x[0x][0x]yJ[0x]G

For an alternate approach, I wanted to check the website certificate to see all the subdomains for this website.

Industrial Intrusion CTF

| Recipe | | Input |
|---|--|------------------------------|
| From Hex <div> Delimitter None </div> | | 54484d7b5375357373737d |
| | | Output THM{Su5sss} |

Sure enough, it looks like we have found our flag. Now it was time to check:

BEGINNER #4

15 Points

Osint

Answer the questions below

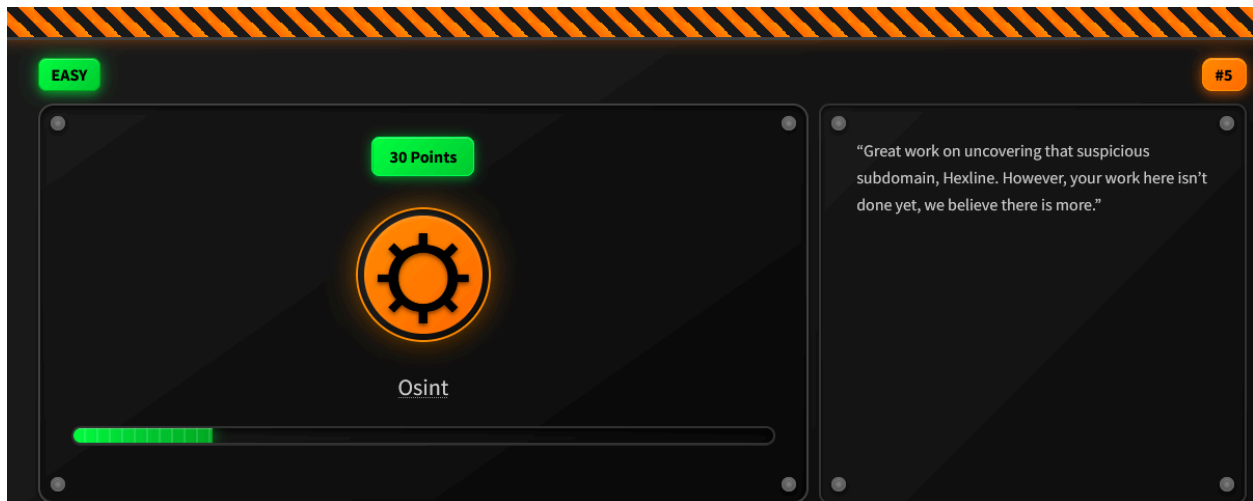
What is the flag?

THM{Su5sss} ✓ Correct Answer

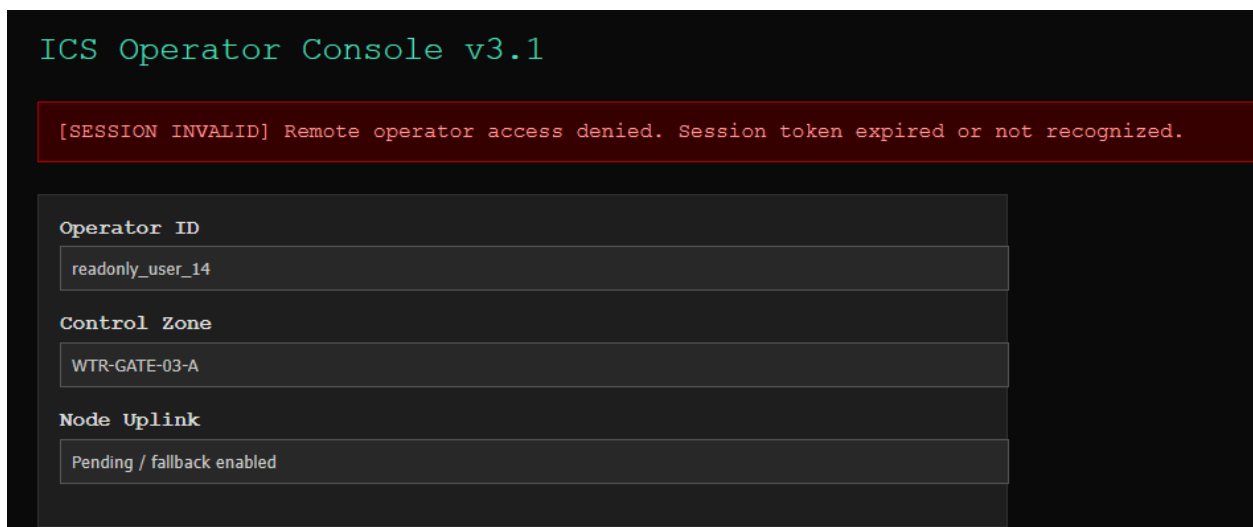
We have found the flag!

Task 6 OSINT 2

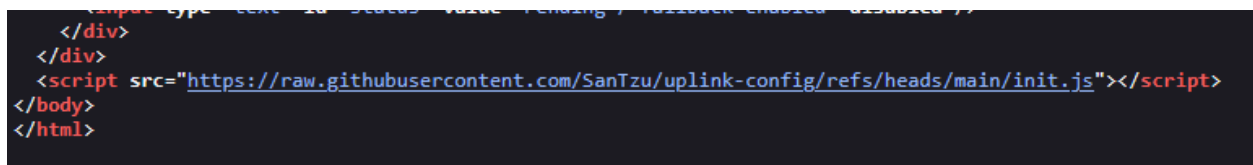
Our next task was to keep snooping around



I went back to the certificate subdomain list and saw <https://stage0.virelia-water.it.com/> which I decided to visit.



Looking at the source code, I saw that there was a link to a source script which looked interesting



The script looked like:

```
var beacon = {  
  session_id: "0-TX-11-403",  
  fallback_dns: "uplink-fallback.virelia-water.it.com",  
  token: "JBSWY3DPEBLW64TMMQQQ=="  
};
```

This was interesting because the initial error message on the website said that there was an error with the token. I then decided to do a dns lookup for the fallback_dns `uplink-fallback.virelia-water.it.com`

Domain Name: uplink-fallback.virelia-water.it.com

Dns Types: TXT=16

Name: uplink-fallback.virelia-water.it.com

Type: 16

TTL: 1799

rRset Type: 16

Strings: eyJzZXNzaW9uljoVC1DTJEtMTcyliwiZmxhZyl6IIRITXt1cGxpbmRfY2hhbm5lbF9jb25maXJtZW9In0=

Raw Text: uplink-fallback.virelia-water.it.com. 1799 IN TXT

"eyJzZXNzaW9uljoVC1DTJEtMTcyliwiZmxhZyl6IIRITXt1cGxpbmRfY2hhbm5lbF9jb25maXJtZW9In0="

This led me to see that there was something which seemed to be base64 encrypted, so I decrypted the string.

Simply enter your data then press the decode button.

```
eyJzZXNzaW9uljoiVC1DTjEtMTcyliwiZmxhZyl6IIRITXt1cGxpbmtfY2hhbm5lbF9jb25maXJtZWRR9In0=
```

 For encoded binaries (like images, documents, etc.) use the file upload form a little further down on this page.

AUTO-DETECT  Source character set. Detected: UTF-8

☐ Decode each line separately (useful for when you have multiple entries).

 **Live mode OFF** Decodes in real-time as you type or paste (supports only the UTF-8 character set).

< DECODE > Decodes your data into the area below.

```
{"session":"T-CN1-172","flag":"THM{uplink_channel_confirmed}"}
```

It looks like I have found a flag! Now I just confirmed that it would be for this task.

What is the flag?

THM{uplink_channel_confirmed}

✓ Correct Answer

Task 7 OSINT 3

This task seems to be related to the path I was going down with the first OSINT task and the PGP signing.



I then started my investigation at the github commit with the PGP signing

<https://github.com/virelia-water/compliance/commit/6d355c02e0e08525712fbd720695acd0450a067a>

```
1 file changed +46 -0 lines changed
mail-archives/ot-alerts/2025-06.html
8 + </head>
9 + <body>
10 + <header><h1>OT Alerts Exception Report - June 2025</h1></header>
11 + <nav>
12 + <a href="/">Home</a>
13 + <a href="/mail-archives/">Archives Home</a>
14 + <a href="/policies/">Compliance Policies</a>
15 + </nav>
16 + <main>
17 + <p>This page lists <em>exceptional</em> OT-Alert messages for June 2025 only. Routine alerts have been redacted.</p>
18 + <div class="message">
19 + <div class="hdr">
20 + From: DarkPulse <alerts@virelia-water.it.com><br>
21 + Date: Mon, 15 Jun 2025 02:15:00 +0000<br>
22 + Subject: Scheduled OT Calibration
23 + </div>
24 + <pre>
25 + -----BEGIN PGP SIGNED MESSAGE-----
26 + Hash: SHA512
27 +
28 + Please confirm system integrity at 03:00 UTC.
29 + -----BEGIN PGP SIGNATURE-----
30 +
31 + iQFQBACBgA6FiEEiN7ee3MFE71e3w2fpPD+sISjEeUFAMhZTEQcHGfSfZXJ0c0B2
32 + aXJlbGhlLXdhdG9yLm0LmNvbQAKCRCK8P6whKMR5ZIUCADM7F0wpKWwyj4wUdoL
33 + 6yrJfJfmUKgJD+8K1neFosG7yaz+MspYXIlbKUek/VFhHZnaG2NRjn6BpfPSxfEk
34 + uvWNIP8rMVEv32vpqhCJ26pwrkAaUhlcPwQm4KY0An4eE0eHCvxHNJBfNmWI5PBF
35 + pXbj7s6DhyZEHUmTo4JK20ZmiISP30sHW808iz5JLURa/qw9LCjY8PK79UoceRwW
36 + tJj9pVsE+TKPcFb/EDzQ6mBH8GB1ki532/1/GDU+iivYSirjxwks/ZYPu/bhktTo
37 + NNcOzgEfuSekKQAz+CicLxwEcLQb219TqcS3plna0672kCV4t5MUCLvkXL5/kHms
38 + Sh5H
39 + =jdL7
40 + -----END PGP SIGNATURE-----
41 + </pre>
42 + </div>
43 + </main>
44 + <footer>&copy; 2025 Virelia Water Control Facility</footer>
45 + </body>
46 + </html>
```

Now having that open, I would like to check if any of the keyserver have any keys related to DarkPulse

```
(kali㉿kali)-[~/Downloads]
$ gpg --keyserver keyserver.ubuntu.com --search-keys "DarkPulse"
gpg: data source: http://185.125.188.26:11371
(1)      Ghost (THM{h0pe_th1s_k3y_doesnt_le4d_t0_m3}) <solstice.tech.ops@gmail
.
      DarkPulse (THM{h0pe_th1s_k3y_doesnt_le4d_t0_m3}) <alerts@virelia-wate
r Output
      2048 bit RSA key F8ED5BC28874364F, created: 2025-06-23
Enter number(s), N)ext, or Q)uit > █
```

Well, it looks like I have found the flag I was looking for.

What is the flag?

THM{h0pe_th1s_k3y_doesnt_le4d_t0_m3}

✓ Correct Answer


Task 24 Reverse Engineering "Auth"

This task requires us to reverse engineer an authentication

EASY

#23

30 Points



Reversing

ZeroTrace intercepts a stripped-down authentication module running on a remote industrial gateway. Assembly scrolls across glowing monitors as she unpacks the logic behind the plant's digital checkpoint.

[Files \(materials\)](#)

10.10.253.80 9005

With the zip file downloaded, I unzipped it and decompiled it using ghidra. There we can see that the main gist of the program is that it asks for an unlock code and will output the flag to me if it is the correct input.


```

int iVar1;
char *pcVar2;
undefined8 uVar3;
size_t sVar4;
FILE *__stream;
long in_FS_OFFSET;
undefined8 local_168;
undefined8 local_160;
undefined8 local_158 [8];
char flag [264];
long local_10;

local_10 = *(long *) (in_FS_OFFSET + 0x28);
local_160 = 0xefcdab8967452301;
printf("[?] Enter unlock code: ");
pcVar2 = fgets((char *)local_158, 0x40, stdin);
if (pcVar2 == (char *)0x0) {
    fwrite("Error reading input\n", 1, 0x14, stderr);
    uVar3 = 1;
}
else {
    sVar4 = strcspn((char *)local_158, "\r\n");
    *(undefined1 *) ((long)local_158 + sVar4) = 0;
    sVar4 = strlen((char *)local_158, 0x40);
    if (sVar4 == 8) {
        local_168 = local_158[0];
        transform(&local_168, 8);
        iVar1 = memcmp(&local_168, &local_160, 8);
        if (iVar1 == 0) {
            __stream = fopen("flag.txt", "r");
            if (__stream == (FILE *)0x0) {
                perror("fopen");
                uVar3 = 1;
            }
            else {
                pcVar2 = fgets(flag, 0x100, __stream);
                if (pcVar2 == (char *)0x0) {
                    fwrite("Error reading flag\n", 1, 0x13, stderr);
                }
                else {
                    printf("[+] Access Granted! Flag: %s", flag);
                }
                fclose(__stream);
                uVar3 = 0;
            }
        }
        else {
            puts("[!] Access Denied!");
            uVar3 = 1;
        }
    }
}

```

There is a `transform()` function as well which seems to XOR values with the hex value of 0x55

```
void transform(long param_1,ulong param_2)
{
    undefined8 local_10;

    for (local_10 = 0; local_10 < param_2; local_10 = local_10 + 1) {
        *(byte *)(local_10 + param_1) = *(byte *)(local_10 + param_1) ^ 0x55;
    }
    return;
}
```

Looking at the main function closely now, we can see that our input should be 8 characters long. Also, we see that there is `local_160 = 0xefcdab8967452301;` set at the top of our program, which happens to also be 8 characters. My understanding is that this variable gets put through the transform function so that each character is XORed by 0x55, and then that is compared to our input. Therefore, we have to perform the transformation:

0x01 XOR 0x55 = 0x54

0x23 XOR 0x55 = 0x76

0x45 XOR 0x55 = 0x10

0x67 XOR 0x55 = 0x32

0x89 XOR 0x55 = 0xdc

0xab XOR 0x55 = 0xfe

0xcd XOR 0x55 = 0x98

0xef XOR 0x55 = 0xba

To input the code into the program, I would need to use the following code

```
python -c "import sys; sys.stdout.buffer.write(b'\x54\x76\x10\x32\xdc\xfe\x98
```

```
\xba\n')" | nc 10.10.253.80 9005
```

```
(kali㉿kali)-[~/Desktop]
$ python -c "import sys; sys.stdout.buffer.write(b'\x54\x76\x10\x32\xdc\xfe\x98\xba\n')" | nc 10.10.253.80 9005
[?] Enter unlock code: [+] Access Granted! Flag: THM{Simple_tostart_nice_done_mwww}
```

This seems to have been successful and the flag was provided.

What is the content of flag.txt?

THM{Simple_tostart_nice_done_mwww}

✓ Correct Answer

Task 25 Reverse Engineering "Access Granted"

The screenshot shows a CTF task interface with a dark theme and orange accents. At the top, there's a header with orange and black diagonal stripes. The task is labeled 'EASY' in a green box and '#24' in an orange box. The main area is divided into two panels. The left panel, titled 'Reversing', features a large orange gear icon and a green '30 Points' badge. Below the icon is a progress bar with several green segments. The right panel contains the task description: 'ZeroTrace intercepts a suspicious HMI login module on the plant floor. Reverse the binary logic to reveal the access key and slip past digital defences.' Below the description is a link 'Files (materials)' and the IP address '10.10.253.80 9009'.

This task provides files, which I unpack and decompile with ghidra.

```

undefined8 main(void)
{
    int iVar1;
    long in_FS_OFFSET;
    char local_38 [40];
    long local_10;

    local_10 = *(long *)(in_FS_OFFSET + 0x28);
    setvbuf(stdout,(char *)0x0,2,0);
    setvbuf(stdin,(char *)0x0,2,0);
    printf("Enter the password : ");
    read(0,local_38,0x1f);
    printf("\nprocessing...");
    iVar1 = strcmp(pass,local_38,10);
    if (iVar1 == 0) {
        puts("Access Granted!");
        print_flag();
    }
    else {
        puts("\nWrong Password!");
    }
    if (local_10 != *(long *)(in_FS_OFFSET + 0x28)) {
        /* WARNING: Subroutine does not return */
        __stack_chk_fail();
    }
    return 0;
}

```

What I initially see is that there is a string compare of 10 characters between our input and a `pass` variable. So I check what the variable contains.

| | | | | | | | | | |
|----------|--------------|--|-----------------|-------------------|--|--|--|----------|----------------------------------|
| -- -- | | | | | | | | | |
| | | | | <code>pass</code> | | | | XREF[2]: | Entry Point(*), main:001013b1(*) |
| 00104010 | 69 6e 64 | | <code>ds</code> | "industrial" | | | | | |
| | 75 73 74 | | | | | | | | |
| | 72 69 61 ... | | | | | | | | |

It seems like the string "industrial" is stored there, which is a 10 character string. So if I start my password with "industrial" it should be successful. I then try it out:

```
(kali㉿kali)-[~/Desktop]
$ echo "industrial" | nc 10.10.194.199 9009
Enter the password :
processing ... Access Granted!
THM{s0meth1ng_inthe_str1ng_she_knows}

(kali㉿kali)-[~/Desktop]
$ echo "industrialrevolution" | nc 10.10.194.199 9009
Enter the password :
processing ... Access Granted!
THM{s0meth1ng_inthe_str1ng_she_knows}
```

It seems to work and gives me the flag.


What is the content of the file flag.txt?

THM{s0meth1ng_inthe_str1ng_she_knows}

✓ Correct Answer

Final result

With the above flags, I was able to rank #521

| Rank | Username | [Task 1] #1 | [Task 2] #1 | [Task 3] #1 | [Task 4] #1 | [Task 5] #1 | [Task 6] #1 | [Task 7] #1 |
|------|--|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| 521 |  g00fyg00bers | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |