



THCon Pre-Challenge

0xf4b

04/04/2024



- **Fabien PERIGAUD - 0xf4b**
- **Reverse team tech lead**
- **Challenges enthousiast**

The challenge

- **Four steps**

- Step 0: Steganography
- Step 1: Web
- Step 2: Reverse
- Step 3: Harder Reverse

- **Increasing difficulty**

- Well... unless you don't know web!

THRON 2024

TOULOUSE HACKING CONVENTION

PRE CHALLENGE

HOW TO PLAY

One should seek mail addresses of the following form: `thc-2024-flag-(large-random-string-flag)@m0rgan.net`.

Please report your achievements by sending a mail to the given addresses.

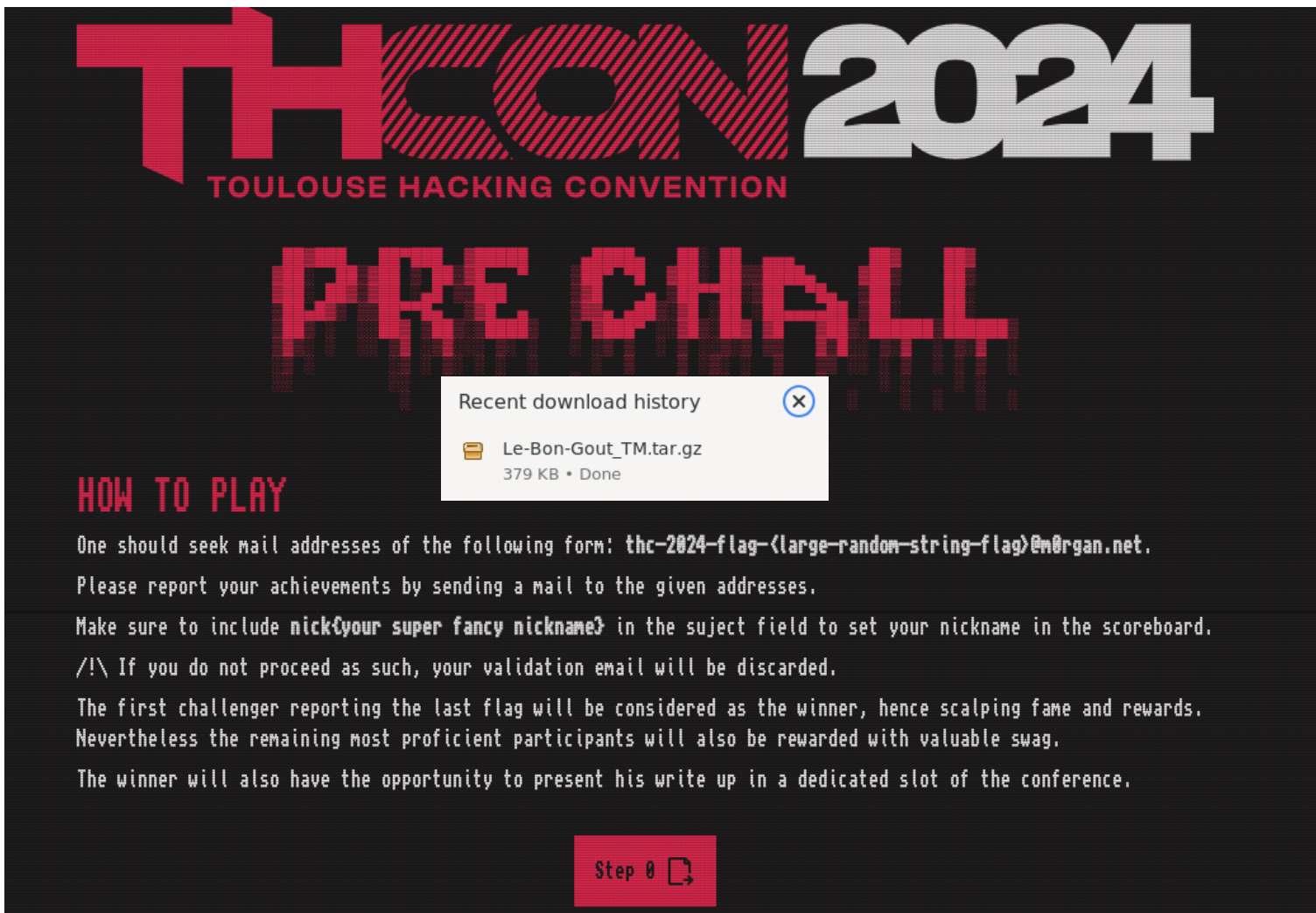
Make sure to include `nick{your super fancy nickname}` in the subject field to set your nickname in the scoreboard.

/!\ If you do not proceed as such, your validation email will be discarded.

The first challenger reporting the last flag will be considered as the winner, hence scalping fame and rewards. Nevertheless the remaining most proficient participants will also be rewarded with valuable swag.

The winner will also have the opportunity to present his write up in a dedicated slot of the conference.

Step 0



THRON 2024

TOULOUSE HACKING CONVENTION

PRE CHALL

Recent download history

Le-Bon-Gout_TM.tar.gz
379 KB • Done

HOW TO PLAY

One should seek mail addresses of the following form: `thc-2024-flag-(large-random-string-flag)@m0rgan.net`.
Please report your achievements by sending a mail to the given addresses.

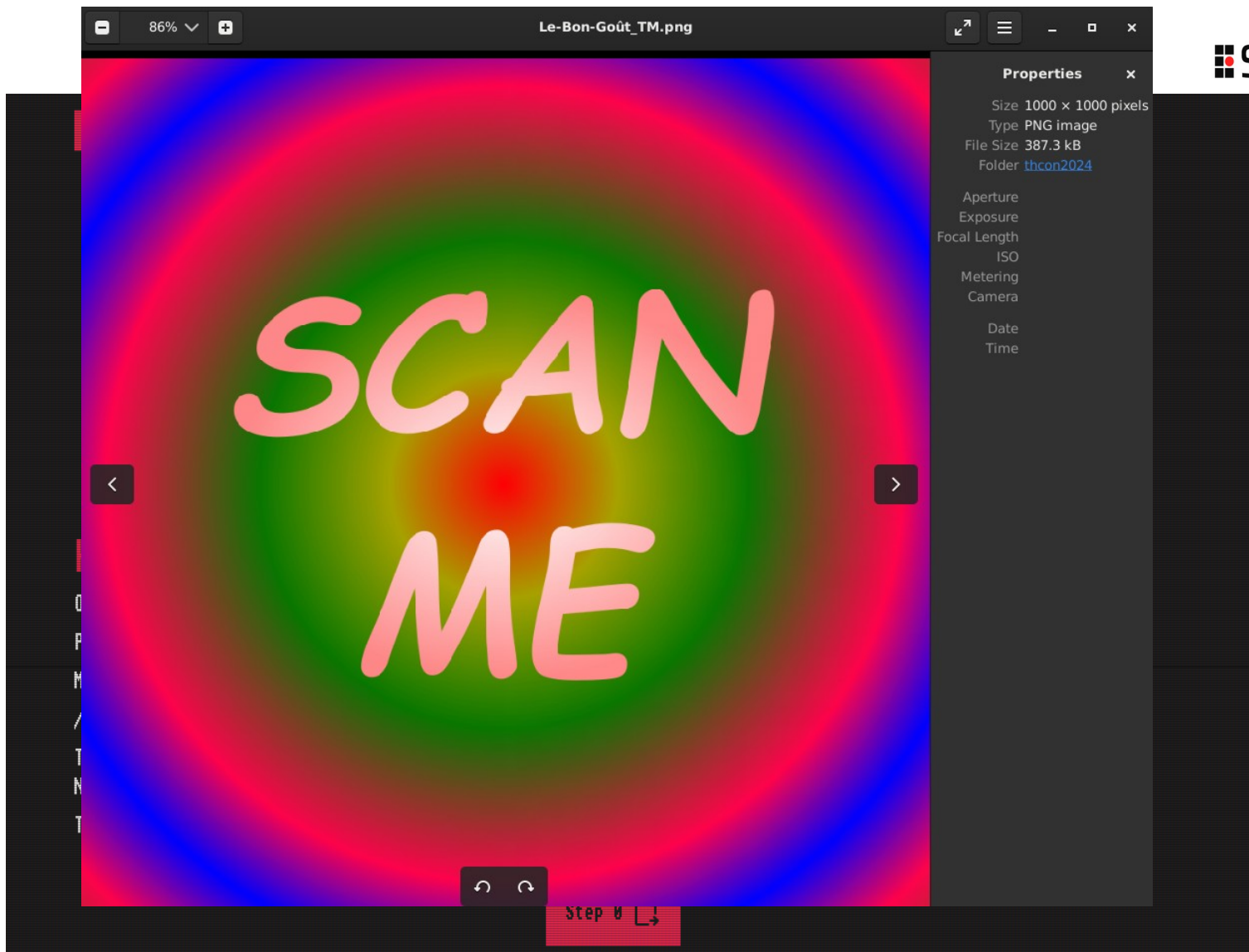
Make sure to include `nick{your super fancy nickname}` in the sujet field to set your nickname in the scoreboard.
/!\ If you do not proceed as such, your validation email will be discarded.

The first challenger reporting the last flag will be considered as the winner, hence scalping fame and rewards.
Nevertheless the remaining most proficient participants will also be rewarded with valuable swag.

The winner will also have the opportunity to present his write up in a dedicated slot of the conference.

Step 0

Step 0



Step 0

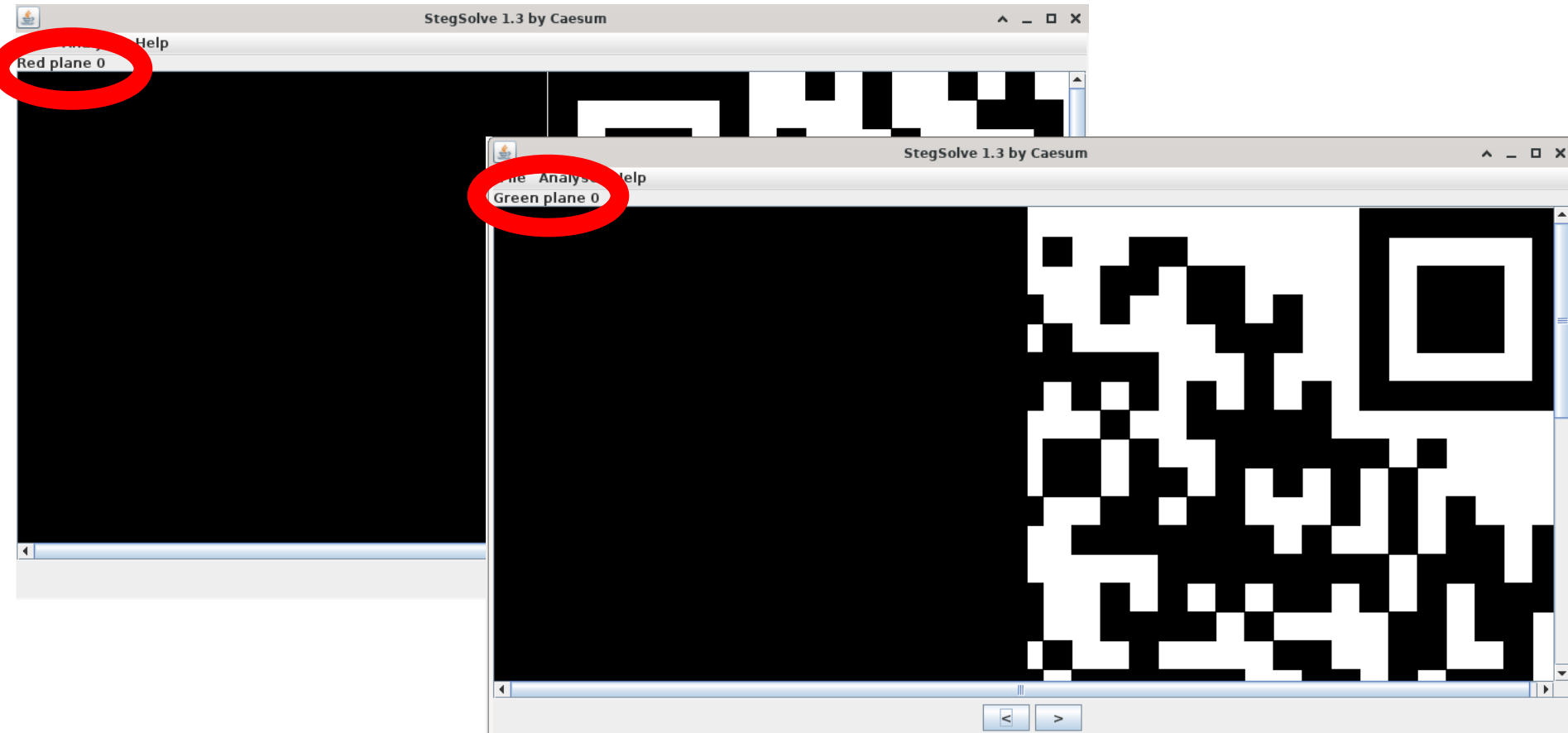
STEGRANOGRAPHY



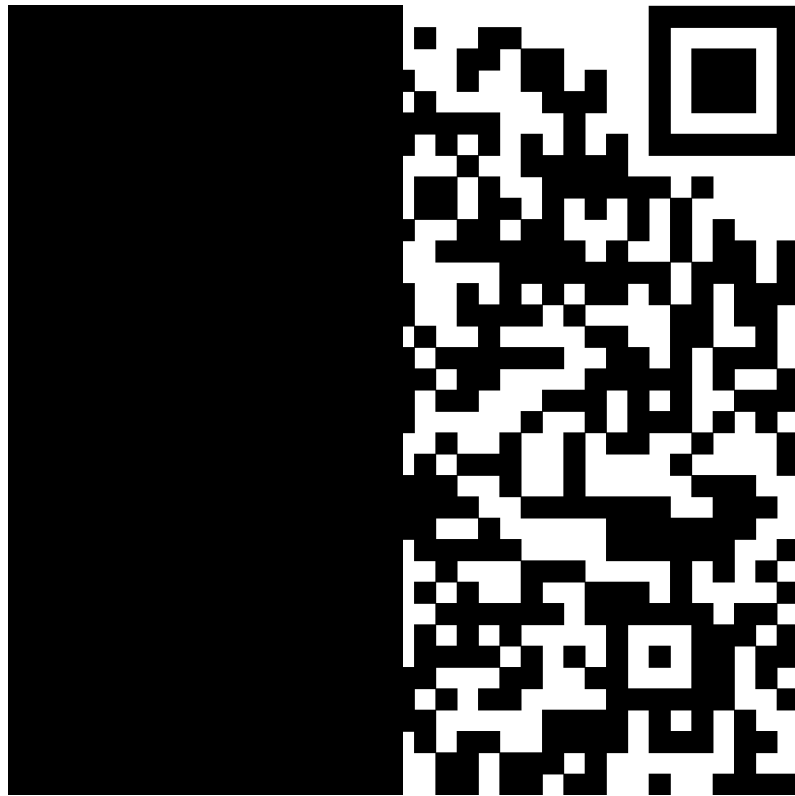
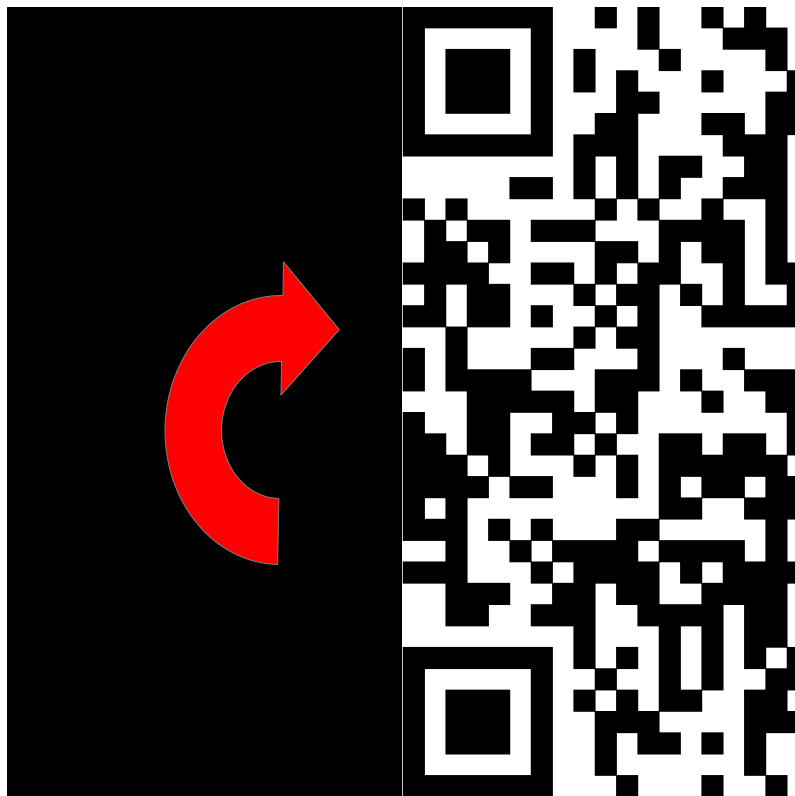
Step 0 → StegSolve



Step 0 → StegSolve



Step 0



Step 0 - Done



← → ↻ 🏠 🔒 thcon-2024.m0rgan.net/zL9EgdswmP

Congrats, this means completion of step 0

flag thc-2024-flag-efs1csztiwc2aqentuzlxurfceelv41bv1tzor4gf2v9778wxf9jztx6zxi@m0rgan.net

Step 1 [here](#) (auto-signed certificate: 2804082853f87ff9a5cfdbfaedd99988b6f53261efe8aa384b678f35c9cf6825)

Step 1

Login to GraphMin

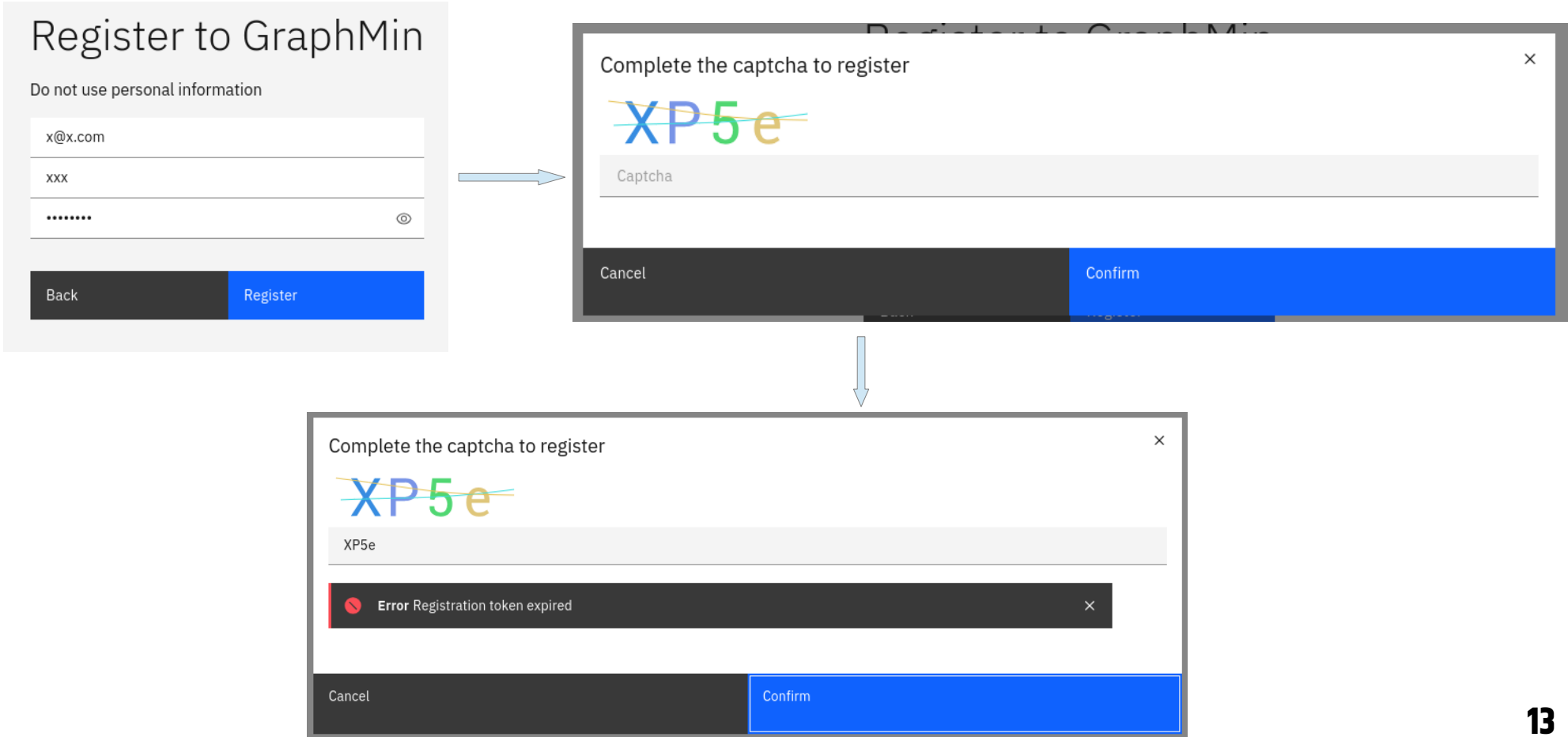


Register

Login

```
<body data-sveltekit-preload-data="hover">
  <div style="display: contents">
    <script>
      {
        sveltekit.worthnav = {
          base: new URL(".", location).pathname.slice(0, -1),
          env: {"PUBLIC_GRAPHQL_URL": "https://51.105.240.10/graphql"}
        }
      }
    </script>
  </div>
</body>
```

Step 1 - Register



Step 1 - Register

75	https://51.105.240.10	POST	/graphql	✓	200	3046	JSON	✓	51.105.240.10	08:29:51.3 Ap
76	https://51.105.240.10	POST	/graphql	✓	200	379	JSON	✓	51.105.240.10	08:29:56.3 Ap

Request

```
1 POST /graphql HTTP/1.1
2 Host: 51.105.240.10
3 Content-Length: 556
4 Sec-Ch-Ua: "Not:A-Brand";v="99", "Chromium";v="112"
5 Accept: application/graphql+json, application/json
6 Content-Type: application/json
7 Sec-Ch-Ua-Mobile: ?0
8 Authorization: Bearer undefined
9 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
  Chrome/112.0.5615.50 Safari/537.36
10 Sec-Ch-Ua-Platform: "Linux"
11 Origin: https://51.105.240.10
12 Sec-Fetch-Site: same-origin
13 Sec-Fetch-Mode: cors
14 Sec-Fetch-Dest: empty
15 Referer: https://51.105.240.10/register
16 Accept-Encoding: gzip, deflate
17 Accept-Language: en-US,en;q=0.9
18 Connection: close
19
20 {
  "operationName": "register",
  "query":
    "mutation register($email: String!, $name: String!, $password: String!) {\n  register(email: $e
    mail, name: $name, password: $password) {\n    ... on BaseError {\n      __typename\n      mess
    age\n    }\n    ... on ZodError {\n      __typename\n      fieldErrors {\n        message\n
    }\n      }\n    ... on MutationRegisterSuccess {\n      __typename\n      data {\n        captc
    ha\n        registrationToken\n      }\n    }\n    __typename\n  }\n}\n",
  "variables": {
    "email": "x@x.com",
    "name": "xxx",
    "password": "xxxxxxxxx"
  }
}
```

Response

```
1 HTTP/1.1 200 OK
2 Server: nginx/1.25.4
3 Date: Wed, 03 Apr 2024 06:29:51 GMT
4 Content-Type: application/json; charset=utf-8
5 Content-Length: 2751
6 Connection: close
7 X-Powered-By: Express
8 access-control-allow-origin: https://51.105.240.10
9 vary: Origin
10 access-control-allow-credentials: true
11
12 {
  "data": {
    "register": {
      "__typename": "MutationRegisterSuccess",
      "data": {
        "captcha":
          "<svg xmlns='http://www.w3.org/2000/svg' width='200' height='50' viewBox='0,0,200,
          50'><path d='M9 13 C94 31,104 27,186 15' stroke='\\#b9e141' fill='\\#none'><path d='\\M
          21 30 C90 6,79 47,191 33' stroke='\\#7aeab2' fill='\\#none'><path fill='\\#5ce4c2' d='\\M
          46,78 44,19L46,33 39,71L46,46 39,84Q44,98 42,14 42,79 43,42L42,80 43,43L42,82 43,45Q40,59
          44,70 37,72 44,70L37,70 44,67L37,81 44,78Q32,81 44,62 30,10 41,53L30,23 41,66L30,11 41,5
          4Q27,52 38,57 27,52 32,04L27,55 32,07L27,46 14,53L32,85 14,53L32,97 32,15L32,89 32,07Q32,
          86 36,77 34,25 38,55L34,30 38,59L34,35 38,65Q35,72 40,39 38,67 40,39L38,65 40,37L38,56 40
          ,28Q41,54 40,39 43,41 39,23L43,27 39,09L43,27 39,08Q45,27 38,05 46,22 35,92L46,18 35,87L4
          6,10 14,46L51,49 14,47L51,62 44,18L46,75 44,152'></path fill='\\#45e293' d='\\M101,50 4,3
          0L107,39 30,39L107,93 35,47L108,26 35,66L109,04 30,27L116,19 4,36L121,34 4,23L128,48 30,2
          6L129,54 35,72L129,55 35,57L130,40 30,35L136,04 4,30L142,00 4,41L132,24 44,08L127,39 44,0
          5L119,52 16,52L118,87 12,89L118,73 12,91L118,10 16,44L110,04 44,06L105,34 44,17L95,68 4,3
          0L101,62 4,412'></path fill='\\#5151ea' d='\\M94,34 4,42L94,34 44,23L88,91 44,19L88,85 26
          ,54L69,59 26,61L69,46 44,06L64,21 44,20L64,22 4,40L69,58 4,37L69,63 22,42L88,78 22,24L88,
          90 4,36L94,21 4,282'></path fill='\\#75dfaa' d='\\M164,91 36,23L164,85 36,17L164,87 36,19
          Q164,84 34,37 163,48 33,25L163,54 33,31L163,59 33,36Q162,16 32,17 158,39 31,32L158,31 31,
          24L158,36 31,29Q153,26 30,29 158,46 28,27L150,42 28,23L150,26 28,07Q147,58 26,17 147,58 2
          2,78L147,52 22,71L147,53 22,73Q147,61 19,20 158,69 16,63L150,72 16,67L150,72 16,66Q153,65
```

Step 1 - Register

75	https://51.105.240.10	POST	/graphql	✓	200	3046	JSON	✓	51.105.240.10	08:29:51.3 Ap
76	https://51.105.240.10	POST	/graphql	✓	200	379	JSON	✓	51.105.240.10	08:29:56.3 Ap

Request

Pretty Raw Hex

```
1 POST /graphql HTTP/1.1
2 Host: 51.105.240.10
3 Content-Length: 556
4 Sec-Ch-Ua: "Not:A-Brand";v="99", "Chromium";v="112"
5 Accept: application/graphql+json, application/json
6 Content-Type: application/json
7 Sec-Ch-Ua-Mobile: ?0
8 Authorization: Bearer undefined
9 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
  Chrome/112.0.5615.50 Safari/537.36
10 Sec-Ch-Ua-Platform: "Linux"
11 Origin: https://51.105.240.10
12 Sec-Fetch-Site: same-origin
13 Sec-Fetch-Mode: cors
14 Sec-Fetch-Dest: empty
15 Referer: https://51.105.240.10/register
16 Accept-Encoding: gzip, deflate
17 Accept-Language: en-US,en;q=0.9
18 Connection: close
19
20 {
  "operationName": "register",
  "query":
    "mutation register($email: String!, $name: String!, $password: String!) {\n  regist
    mail, name: $name, password: $password) {\n    ... on BaseError {\n      __typename
    age\n    }\n    ... on ZodError {\n      __typename\n      fieldErrors {\n        m
    }\n    }\n    ... on MutationRegisterSuccess {\n      __typename\n      data {\n
    ha\n      registrationToken\n    }\n    }\n    __typename\n  }\n}\n",
  "variables": {
    "email": "x@x.com",
    "name": "xxx",
    "password": "xxxxxxxx"
  }
}
```

Response

Pretty Raw Hex Render

```
1 HTTP/1.1 200 OK
2 Server: nginx/1.25.4
3 Date: Wed, 03 Apr 2024 06:29:51 GMT
4 Content-Type: application/json; charset=utf-8
5 Content-Length: 2751
6 Connection: close
7 X-Powered-By: Express
8 access-control-allow-origin: https://51.105.240.10
9 vary: Origin
```

Request

Pretty Raw Hex

```
1 POST /graphql HTTP/1.1
2 Host: 51.105.240.10
3 Content-Length: 406
4 Sec-Ch-Ua: "Not:A-Brand";v="99", "Chromium";v="112"
5 Accept: application/graphql+json, application/json
6 Content-Type: application/json
7 Sec-Ch-Ua-Mobile: ?0
8 Authorization: Bearer undefined
9 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
  Chrome/112.0.5615.50 Safari/537.36
10 Sec-Ch-Ua-Platform: "Linux"
11 Origin: https://51.105.240.10
12 Sec-Fetch-Site: same-origin
13 Sec-Fetch-Mode: cors
14 Sec-Fetch-Dest: empty
15 Referer: https://51.105.240.10/register
16 Accept-Encoding: gzip, deflate
17 Accept-Language: en-US,en;q=0.9
18 Connection: close
19
20 {
  "operationName": "captcha",
  "query":
    "mutation captcha($captcha: String!, $registrationToken: String!) {\n  captcha(captcha: $ca
    ptcha, registrationToken: $registrationToken) {\n    ... on BaseError {\n      __typename\n
    message\n    }\n    ... on MutationCaptchaSuccess {\n      __typename\n    }\n    _t
    ypeName\n  }\n}\n",
  "variables": {
    "captcha": "uHWS",
    "registrationToken": "c1tdg5y3e0g65lmpqstmrj136"
  }
}
```

Response

Pretty Raw Hex Render

```
1 HTTP/1.1 200 OK
2 Server: nginx/1.25.4
3 Date: Wed, 03 Apr 2024 06:29:56 GMT
4 Content-Type: application/json; charset=utf-8
5 Content-Length: 86
6 Connection: close
7 X-Powered-By: Express
8 access-control-allow-origin: https://51.105.240.10
9 vary: Origin
10 access-control-allow-credentials: true
11
12 {
  "data": {
    "captcha": {
      "__typename": "BaseError",
      "message": "Registration token expired"
    }
  }
}
```

Step 1 – break captcha!

- First try with Tesseract → fail

- SVG format!

- Just remove the lines...

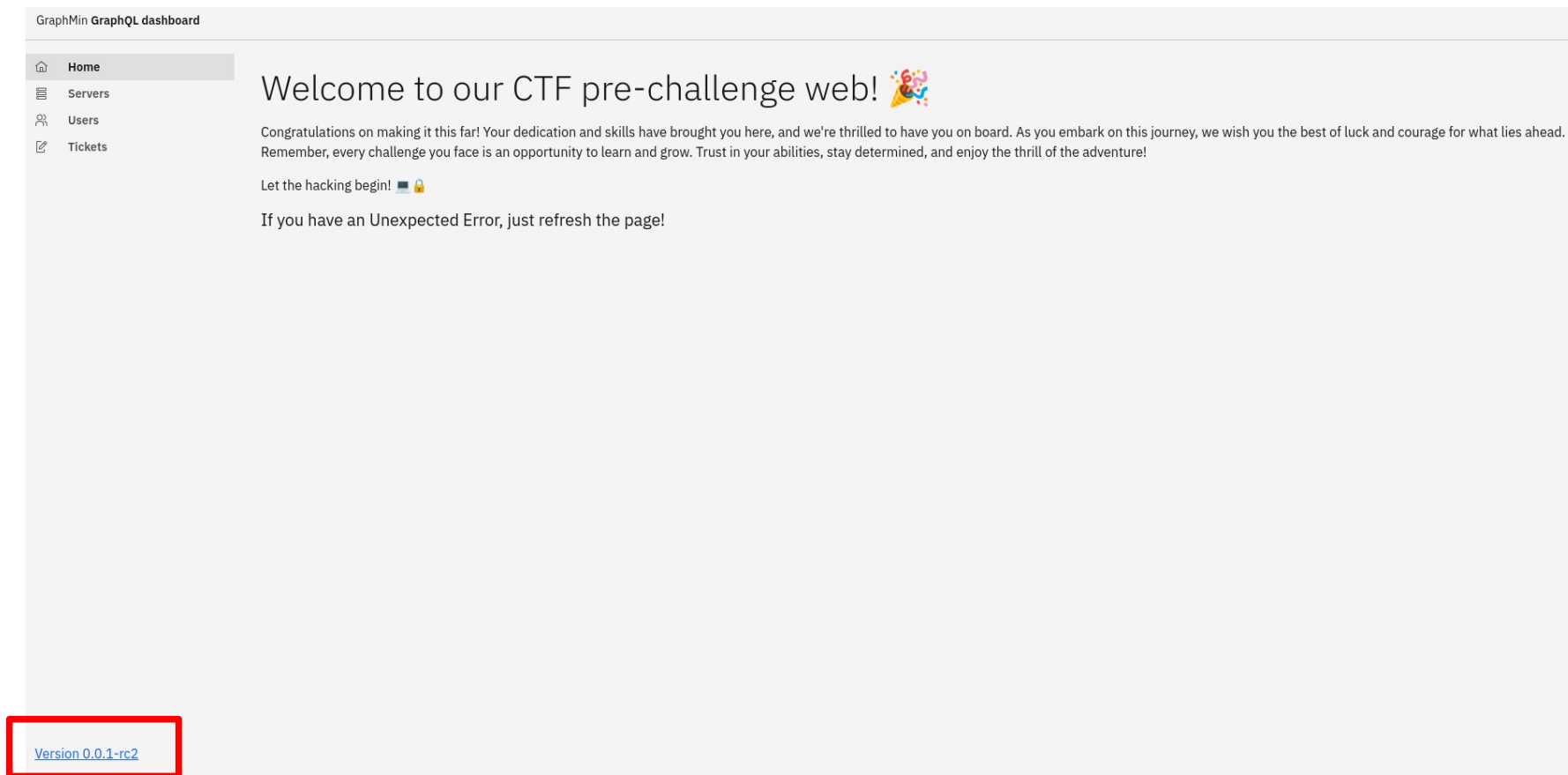
- Now Tesseract works :)

- Send register request, break captcha, send confirmation

- Account created!

```
<svg xmlns="http://www.w3.org/2000/svg" width="200" height="50" viewBox="0,0,200,50"> == $0
<path fill="#7894e7" d="M81.26 28.16L71.52 28.26L71.53 44.24L66.05 44.14L66.02 4.30L81.23 4.28L81.31 4.3
6087.62 4.30 91.17 7.58L91.16 7.58L91.28 7.70Q94.71 10.87 94.71 16.23L94.82 16.34L94.73 16.25Q94.68 21.6
1 91.14 24.86L91.17 24.89L91.26 24.99Q87.60 28.12 81.23 28.12L81.29 28.19ZM71.52 8.65L71.37 23.90L81.33
24.01L81.24 23.93Q85.34 23.97 87.33 21.80L87.42 21.89L87.28 21.75Q89.34 19.63 89.34 16.32L89.38 16.37L8
9.27 16.26Q89.24 12.92 87.23 10.71L87.33 10.80L87.29 10.76Q85.32 8.59 81.30 8.59L81.34 8.63L71.53 8.66
Z"/></path>
<path fill="#dfc67a" d="M159.86 44.67L159.82 44.63L159.99 44.80Q153.54 44.72 149.97 40.64L149.90 40.57L1
49.97 40.65Q146.33 36.50 146.33 29.86L146.34 29.87L146.37 28.70L146.42 28.74Q146.36 22.31 150.03 18.14L1
50.10 18.21L150.13 18.23Q153.78 14.04 159.09 14.04L159.04 13.99L159.13 14.00Q164.98 14.00 167.95 17.69L1
68.04 17.78L168.05 17.79Q170.94 21.40 170.94 27.53L170.88 27.47L170.87 30.30L151.97 30.33L151.87 30.45L1
51.87 30.45Q151.98 35.02 153.95 37.79L153.98 37.82L153.79 37.63Q155.92 40.56 159.96 40.56L159.94 40.54L1
59.82 40.42Q162.62 40.48 164.68 39.70L164.77 39.79L164.72 39.74Q166.77 38.95 168.25 37.58L168.26 37.59L1
70.44 41.17L170.35 41.08Q168.70 42.54 166.10 43.59L166.13 43.63L166.16 43.65Q163.60 44.75 159.94 44.75ZM
159.13 18.32L159.13 18.32L158.98 18.17Q156.14 18.15 154.30 20.32L154.42 20.44L154.45 20.47Q152.50 22.54
152.09 25.93L152.22 26.07L152.25 26.18L165.63 26.22L165.55 25.70L165.45 25.59Q165.44 22.36 163.86 20.26L
163.88 20.28L163.89 20.29Q162.32 18.20 159.01 18.20Z"/></path>
<path fill="#50d772" d="M112.30 26.94L107.77 26.40L110.08 4.29L130.06 4.33L130.11 9.17L114.57 9.02L113.4
4 20.64L113.33 20.53Q114.59 19.55 116.15 18.94L116.30 19.09L116.27 19.05Q117.87 18.48 119.87 18.46L119.8
7 18.45L119.67 18.26Q125.20 18.20 128.33 21.72L128.44 21.83L128.35 21.74Q131.51 25.28 131.51 31.41L131.5
3 31.43L131.49 31.39Q131.56 37.44 128.34 41.08L128.43 41.16L128.31 41.05Q125.21 44.80 119.06 44.80L118.9
5 44.68L119.07 44.80Q113.99 44.78 110.62 42.02L110.46 41.85L110.47 41.87Q107.26 39.26 107.39 33.90L107.3
3 33.83L107.43 33.71L112.34 33.76L112.25 33.66Q112.21 36.88 114.08 38.67L114.17 38.76L114.21 38.80Q115.9
7 40.48 118.95 40.48L118.95 40.47L119.02 40.54Q122.46 40.56 124.34 38.16L124.19 38.00L124.33 38.14Q126.1
9 35.71 126.19 31.53L126.12 31.46L126.23 31.57Q126.08 27.62 124.18 25.26L124.24 25.33L124.25 25.34Q122.2
9 22.93 118.90 22.93L118.93 22.96L118.90 22.92Q115.76 22.96 114.34 23.92L114.30 23.88L114.32 23.89Q113.0
1 24.97 112.33 26.96L112.21 26.84Z"/></path>
<path fill="#378ce1" d="M30.40 4.30L39.76 19.95L48.93 4.30L55.40 4.25L42.92 24.15L55.81 44.20L49.39 44.2
1L39.75 28.30L30.18 44.12L23.74 44.18L36.69 24.16L23.95 4.30L30.37 4.27Z"/></path>
<path d="M13 27 C82 24,115 27,197 16" stroke="#40ddd" fill="none"/></path>
<path d="M8 4 C104 16,109 21,197 22" stroke="#e7c459" fill="none"/></path>
</svg>
```

Step 1 – Logged in



Step 1 – Logged in

GraphMin GraphQL dashboard

app

Version 0.0.1-rc2

Changelogs Initial release
Added users
Added servers
Added tickets

API

Version 0.0.1-rc3

Changelogs New mutation upsertUserCert to add or update user certificate

[Version 0.0.1-rc2](#)

Step 1 : Tickets

←

→

↺

🏠

🌐 https://51.105.240.10/tickets/3

🔴

⚙️

👤

📄

👤

⋮

GraphMin GraphQL dashboard

🏠 Home

📁 Servers

👥 Users

📄 Tickets

New feature

CLOSED

2/11/2024, 3:27:55 AM opened by Whidix

I would like to have a new feature on the web server

Whidix

2/11/2024, 4:53:24 AM

Hey, as a team leader, I would like to have the possibility to sign my team's public keys to authenticate them

Admin

2/11/2024, 4:23:24 PM

Hello, I am the admin, the feature will be available soon, I will take care of it

Whidix

2/11/2024, 6:55:00 PM

Ok, thank you

Admin

3/1/2024, 10:27:55 AM

Hey, the new feature is available, you can now sign your team's public keys through the web interface; you just need to upload the pubKey

Whidix

3/1/2024, 10:29:55 AM

Thank you

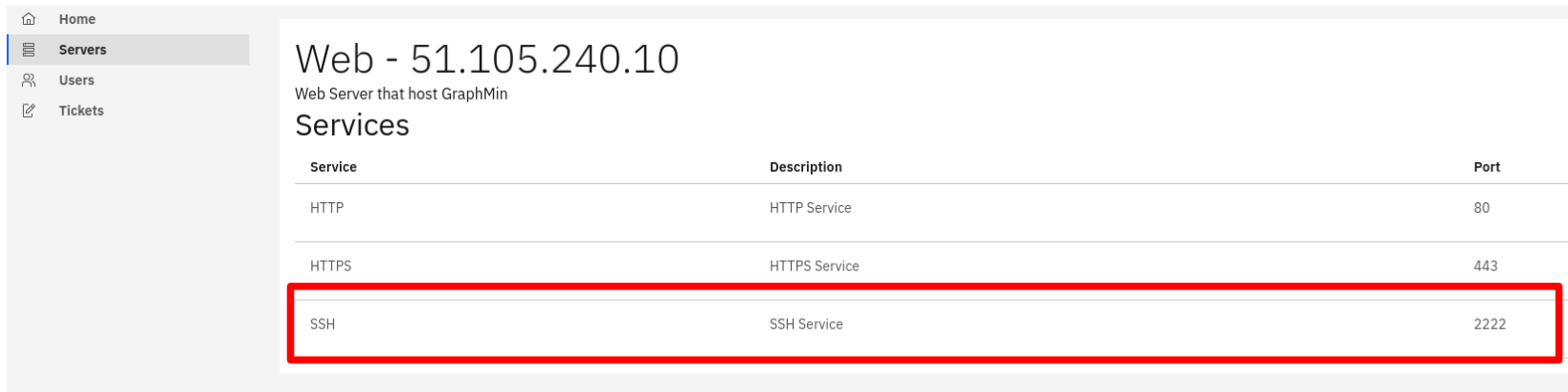
Message

📤

Step 1

■ Clear goal:

- Understand how the upsertUserCert mutation works
- Upload a public key
- Login to SSH



The screenshot shows a web management interface. On the left is a sidebar with navigation links: Home, Servers (selected), Users, and Tickets. The main content area displays information for a server named 'Web - 51.105.240.10', described as 'Web Server that host GraphMin'. Below this, a section titled 'Services' contains a table with three rows: HTTP (port 80), HTTPS (port 443), and SSH (port 2222). The SSH row is highlighted with a red rectangle.

Service	Description	Port
HTTP	HTTP Service	80
HTTPS	HTTPS Service	443
SSH	SSH Service	2222

Step 1

120 https://51.105.240.10 POST /graphql ✓ 200 368 JSON ✓ 51.105.240.10 08:3

Request

Pretty Raw Hex

```
1 POST /graphql HTTP/1.1
2 Host: 51.105.240.10
3 Cookie: token=s5mqn9tygjbkojrcpygho
4 Content-Length: 328
5 Sec-Ch-Ua: "Not:A-Brand";v="99", "Chromium";v="112"
6 Accept: application/graphql+json, application/json
7 Content-Type: application/json
8 Sec-Ch-Ua-Mobile: ?0
9 Authorization: Bearer s5mqn9tygjbkojrcpygho
10 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/112.0.5615.50 Safari/537.36
11 Sec-Ch-Ua-Platform: "Linux"
12 Origin: https://51.105.240.10
13 Sec-Fetch-Site: same-origin
14 Sec-Fetch-Mode: cors
15 Sec-Fetch-Dest: empty
16 Referer: https://51.105.240.10/users/1
17 Accept-Encoding: gzip, deflate
18 Accept-Language: en-US,en;q=0.9
19 Connection: close
20
21 {
  "operationName": "UserCert",
  "query":
    "query UserCert($userId: ID!) {\n  userCert(userId: $userId) {\n    ... on BaseError {\n      __typename\n      message\n    }\n    ... on QueryUserCertSuccess {\n      __typename\n      data {\n        publicKey\n        id\n      }\n    }\n  }\n  __typename\n  }\n}\n",
  "variables": {
    "userId": "1"
  }
}
```

Response

Pretty Raw Hex Render

```
1 HTTP/1.1 200 OK
2 Server: nginx/1.25.4
3 Date: Wed, 03 Apr 2024 06:32:04 GMT
4 Content-Type: application/json; charset=utf-8
5 Content-Length: 75
6 Connection: close
7 X-Powered-By: Express
8 access-control-allow-origin: https://51.105.240.10
9 vary: Origin
10 access-control-allow-credentials: true
11
12 {
  "data": {
    "userCert": {
      "__typename": "BaseError",
      "message": "Not authorized"
    }
  }
}
```

Step 1

Request

```
1 POST /graphql HTTP/1.1
2 Host: 51.105.240.10
3 Cookie: token=s5mqn9tygjbkojrcpygho
4 Content-Length: 325
5 Sec-Ch-Ua: "Not:A-Brand";v="99", "Chromium";v="112"
6 Accept: application/graphql+json, application/json
7 Content-Type: application/json
8 Sec-Ch-Ua-Mobile: ?0
9 Authorization: Bearer s5mqn9tygjbkojrcpygho
10 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
    Chrome/112.0.5615.50 Safari/537.36
11 Sec-Ch-Ua-Platform: "Linux"
12 Origin: https://51.105.240.10
13 Sec-Fetch-Site: same-origin
14 Sec-Fetch-Mode: cors
15 Sec-Fetch-Dest: empty
16 Referer: https://51.105.240.10/tickets/21
17 Accept-Encoding: gzip, deflate
18 Accept-Language: en-US,en;q=0.9
19 Connection: close
20
21 {
  "operationName": "upsertUserCert",
  "query":
    "mutation upsertUserCert($pubKey: String!) {\n  upsertUserCert(pubKey: $pubKey) {\n    ... on B
    aseError {\n      __typename\n      message\n    }\n    ... on BaseError {\n      __typename\n      message\n    }\n  }\n}",
  "variables": {
    "pubKey": "ssh-rsa AAAAAA"
  }
}
```

Response

```
1 HTTP/1.1 200 OK
2 Server: nginx/1.25.4
3 Date: Wed, 03 Apr 2024 06:37:46 GMT
4 Content-Type: application/json; charset=utf-8
5 Content-Length: 301
6 Connection: close
7 X-Powered-By: Express
8 access-control-allow-origin: https://51.105.240.10
9 vary: Origin
10 access-control-allow-credentials: true
11
12 {
  "data": {
    "upsertUserCert": {
      "__typename": "BaseError",
      "message":
        "Command failed: ssh-keygen -s /keys/domain -I 23 -n thc2k24 -V +52w -O no-port-forwarding
        -O no-x11-forwarding -O no-agent-forwarding -O no-user-rc -O no-pty 23.pub\ndo_ca_sign: una
        ble to open \"23.pub\": No such file or directory\r\n"
    }
  }
}
```

Step 1 - Finally

Home

Servers

Users

Tickets

Contact information 

Email address

user123@gmail.com 

Name

user123

Keys 

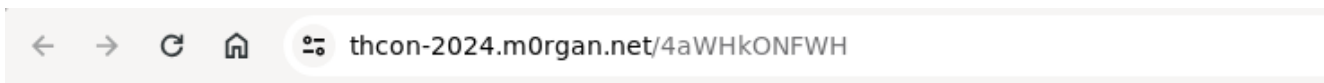
Password

Signed Public Key

ssh-rsa-cert-v01@openssh.com
AAAAHHNzaC1yc2EtY2VydC12MDFAb3BlbnNzaC5jb20AAAAgo8yNPd0no04ndyrm8lpKYN6A1PzxFIHgDFC16APeIXgAAAADAQABAAABgQC04L0fjmBw2UtDq5WrILW8JgQjEIPi1212ul8yWeUBRJ72oa6y
tlecerf@FRL187 

Step 1 - Done

- Login with ssh
- Got next URL



Congrats, this means completion of step 1

flag thc-2024-flag-aun9dwzu6wxd7spcg5k59pj6wnanmy0kasohy kz5q9qbwi yocdyl26axazwg@m0rgan.net

Step 2 [here](#)

Step 2

- **Windows binary**
- **“Usage: %s hex0 hex1 ... hex11”**
 - Takes 12 hex arguments
- **Arguments are used for**
 - Handle and loop start (hex0)
 - Resource import (hex1 / hex2)
 - Imports resolution (hex10 / hex11)
- **Arguments 3 to 9 are not used**
- **In the end, decrypt and load a payload**

Step 2 – hex0 / hex1 / hex2

- hex0 == 0

```
hModule = GetModuleHandleW((LPCWSTR)args[0]);
```

```
for ( m = args[0]; m <= 11; ++m )  
    v37[m] = args[m];
```

- Resources handling

```
for ( j = 0; j < 4; ++j )  
    Type[j] = (unsigned __int8)(args[1] >> (8 * j));  
hResInfo = FindResourceW(hModule, (LPCWSTR)LOWORD(args[2]), (LPCWSTR)Type);
```

C++

 Copier

```
HRSRC FindResourceW(  
    [in, optional] HMODULE hModule,  
    [in]           LPCWSTR lpName,  
    [in]           LPCWSTR lpType  
);
```

- lpName == "RAW" → hex1 == 574152
- lpType == 101 → hex2 == 65

Step 2 – Imports handling

■ Custom imports by hash

```
LoadResource = (__int64 (__fastcall *) (HMODULE, HRSRC))import_by_hash("kernel32", 0x5C4B3BD, args[10], args[11]);
```

■ Constraints

- $\text{Hex10} < 2^{10}$
- $\text{Hex11} < 2^{15}$

■ Simple algorithm

```
int64 __fastcall hash_build(const char *import_name, unsigned int hex10, int hex11)
{
    double v3; // xmm1_8
    unsigned int hash; // [rsp+20h] [rbp-38h]
    size_t i; // [rsp+28h] [rbp-30h]
    size_t v7; // [rsp+40h] [rbp-18h]

    v7 = len(import_name, 0x32uLL);
    hash = hex10;
    if ( (double)(int)hex10 > pow_0((double)(int)hex10, v3) || (double)hex11 > pow_0((double)hex11, (double)(int)hex10) )
        exit(1);
    for ( i = 0LL; i < v7; ++i )
        hash += (import_name[i] + hex11 * hash) & 0FFFFFFF;
    return hash;
}
```

Step 2 – Imports handling (2)

- We know the expected import name...
- We know the output hash...
- Let's bruteforce!

```
$ time pypy bf1.py  
(53, 30864)
```

```
real 0m1.261s
```

- **hex10 == 35**
- **hex11 == 7890**

Step 2 – Arguments found!

- **"C:\Users\user\Desktop\Dread-Loader.exe" 0 574152 65 3 4
5 6 7 8 9 35 7890**
 - Executes a payload
 - You can reverse it (Rust...)!
 - ... or just run it?

Step 2 - Network

■ Listen to traffic

207	28.194121905	192.168.56.114	192.168.56.1	DNS	86	Standard query 0xc4be A vulnerable_satellite.thcon
208	29.198868819	192.168.56.114	192.168.56.1	DNS	86	Standard query 0xc4be A vulnerable_satellite.thcon
209	30.202131004	192.168.56.114	192.168.56.1	DNS	86	Standard query 0xc4be A vulnerable_satellite.thcon

■ Fake DNS!

23	4.796455043	192.168.56.114	192.168.56.1	DNS	86	Standard query 0x8a0b A vulnerable_satellite.thcon
24	4.796963181	192.168.56.1	192.168.56.114	DNS	128	Standard query response 0x8a0b A vulnerable_satellite.thcon A 192.168.56.1
25	4.798580670	192.168.56.114	192.168.56.1	TCP	66	58904 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM

■ Listen on 80?

Step 2 - Done

92	71.311206978	192.168.56.114	192.168.56.1	DNS	86 Standard query 0xeb2e A vulnerable_satellite.thcon
93	71.311499253	192.168.56.1	192.168.56.114	DNS	128 Standard query response 0xeb2e A vulnerable_satellite.thcon A 192.168.56.1
94	71.313177285	192.168.56.114	192.168.56.1	TCP	66 58921 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
95	71.313205285	192.168.56.1	192.168.56.114	TCP	66 80 → 58921 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM WS=128
96	71.313571096	192.168.56.114	192.168.56.1	TCP	54 58921 → 80 [ACK] Seq=1 Ack=1 Win=262656 Len=0
97	71.313950137	192.168.56.114	192.168.56.1	HTTP	119 GET / HTTP/1.1
98	71.313965823	192.168.56.1	192.168.56.114	TCP	54 80 → 58921 [ACK] Seq=1 Ack=66 Win=64256 Len=0
99	71.314854094	192.168.56.1	192.168.56.114	TCP	210 80 → 58921 [PSH, ACK] Seq=1 Ack=66 Win=64256 Len=156 [TCP segment of a reassembled PDU]
100	71.314890491	192.168.56.1	192.168.56.114	TCP	4434 80 → 58921 [PSH, ACK] Seq=157 Ack=66 Win=64256 Len=4380 [TCP segment of a reassembled PDU]
101	71.314920213	192.168.56.1	192.168.56.114	HTTP	75 HTTP/1.0 200 OK (text/html)
102	71.315093525	192.168.56.114	192.168.56.1	TCP	54 58921 → 80 [ACK] Seq=66 Ack=4559 Win=262656 Len=0
103	71.315284211	192.168.56.114	192.168.56.1	TCP	54 58921 → 80 [FIN, ACK] Seq=66 Ack=4559 Win=262656 Len=0
104	71.315296680	192.168.56.1	192.168.56.114	TCP	54 80 → 58921 [ACK] Seq=4559 Ack=67 Win=64256 Len=0
105	71.316322634	192.168.56.114	192.168.56.1	DNS	86 Standard query 0x5c52 A vulnerable_satellite.thcon
106	71.316603131	192.168.56.1	192.168.56.114	DNS	128 Standard query response 0x5c52 A vulnerable_satellite.thcon A 192.168.56.1
107	71.317115124	192.168.56.114	192.168.56.1	TCP	66 58922 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
108	71.317133121	192.168.56.1	192.168.56.114	TCP	66 80 → 58922 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM WS=128
109	71.317228404	192.168.56.114	192.168.56.1	TCP	54 58922 → 80 [ACK] Seq=1 Ack=1 Win=262656 Len=0
110	71.317351670	192.168.56.114	192.168.56.1	HTTP	172 POST / HTTP/1.1
111	71.317359876	192.168.56.1	192.168.56.114	TCP	54 80 → 58922 [ACK] Seq=1 Ack=119 Win=64128 Len=0
112	71.317621162	192.168.56.1	192.168.56.114	TCP	252 80 → 58922 [PSH, ACK] Seq=1 Ack=119 Win=64128 Len=198 [TCP segment of a reassembled PDU]

Frame 110: 172 bytes on wire (1376 bits), 172 bytes captured (1376 bits) on interface vboxnet0, 1
Ethernet II, Src: PcsCompu_3e:2f:c0 (08:00:27:3e:2f:c0), Dst: 0a:00:27:00:00:00 (0a:00:27:00:00:00)
Internet Protocol Version 4, Src: 192.168.56.114, Dst: 192.168.56.1
Transmission Control Protocol, Src Port: 58922, Dst Port: 80, Seq: 1, Ack: 1, Len: 118
Hypertext Transfer Protocol
POST / HTTP/1.1
x-c2-url: https://thcon-2024.m0rgan.net/4ns9LHLgJi\r\n
accept: */*\r\n
host: vulnerable_satellite.thcon\r\n\r\n
[Full request URI: http://vulnerable_satellite.thcon/]
[HTTP request 1/1]
[Response in frame: 113]

0000 0a 00 27 00 00 00 08 00 27 3e 2f c0 08 00 45 00 ...'.....'>/...E.
0010 00 9e cc 5b 40 00 80 06 3c 3a c0 a8 38 72 c0 a8 ...[0...<...8r..
0020 38 01 e6 2a 00 50 24 c2 4c ca 05 81 ee 7d 50 18 8...*.PS: L....}P.
0030 04 02 a2 b5 00 00 50 4f 53 54 20 2f 20 48 54 54PO ST / HTT
0040 50 2f 31 2e 31 0d 0a 78 2d 63 32 2d 75 72 6c 3a P/1.1..x -c2-url:
0050 20 68 74 74 70 73 3a 2f 2f 74 68 63 6f 6e 2d 32 https:// /thcon-2
0060 30 32 34 2e 6d 30 72 67 61 6e 2e 6e 65 74 2f 34 024.m0rg an.net/4
0070 6e 73 39 4c 48 4c 67 4a 69 0d 0a 61 63 63 65 70 ns9LHLgJ i...accep
0080 74 3a 20 2a 2f 2a 0d 0a 68 6f 73 74 3a 20 76 75 t: */*.. host: vu
0090 6c 6e 65 72 61 62 6c 65 5f 73 61 74 65 6c 6c 69 lnerable_satelli
00a0 74 65 2e 74 68 63 6f 6e 0d 0a 0d 0a te.thcon

thcon-2024.m0rgan.net/4ns9LHLgJi

Congrats, this means completion of step 2

flag thc-2024-flag-jwrwwijfo58vsj8okmfmyr0lx1lbadpee4dxdctqvetjwna6dvajqwjpbyk@m0rgan.net

Step 3 [here](#)

Step 3 – All cries

```
fab@x:/tmp$ tar xvJf all-cries.tar.xz
all-cries/
all-cries/flag.enc
all-cries/README.PLZ
all-cries/this-is-no-xoreaxeaxeax.elf
```

```
fab@x:/tmp$ cd all-cries/
```

```
fab@x:/tmp/all-cries$ cat README.PLZ
```

The 4 bytes key alphabet is the printable 95 ASCII characters
[127 downto 33].

Have fun

```
fab@x:/tmp/all-cries$ ./this-is-no-xoreaxeaxeax.elf
```

```
usage(): ./mov `perl -e 'print "\xAA\xBB\xCC\xDD"'` in.bin.enc out.bin
e.g. use a 4 bytes passphrase
```

Step 3

- References to “xoreaxeaxeax” and “mov”
- Points to “movfuscator”
- Highly obfuscated binary to decrypt flag.enc
- 4 ascii characters passphrase

```
if ( stack_argc < 4 )
{
    puts("usage(): ./mov `perl -e 'print \"\\xAA\\xBB\\xCC\\xDD\"'` in.bin.enc out.bin\n e.g. use a 4 bytes passphrase\n");
    exit(1LL);
}
if ( count_args(argv[1]) < 4 )
{
    puts("usage(): ./mov `perl -e 'print \"\\xAA\\xBB\\xCC\\xDD\"'` in.bin.enc out.bin\n e.g. use a 4 bytes passphrase\n");
    exit(1LL);
}
v7 = (_BYTE *)argv[1];
key[0] = *v7;
key[2] = v7[1];
key[4] = v7[2];
key[6] = v7[3];
puts("Hi my good wanderer ~/ That is damn movfuscated\n");
set_SIGINT(2, ((__int64)SIGINT_handler));
v8 = open((const char *)argv[2], 0, 0);
if ( v8 < 0 )
{
    puts("Failed to open input file\n");
    exit(1LL);
}
fd_input = v8;
v9 = lseek(v8, 0LL, 2u);
if ( v9 == -1 )
{
    puts("Failed to lseek input file to the end\n");
    exit(1LL);
}
file_size[0] = v9;
file_size[1] = 0LL;
if ( lseek(fd_input, 0LL, 0) == -1 )
{
    puts("Failed to lseek input file begin\n");
    exit(1LL);
}
if ( mmap(0xCAFE0000uLL, file_size[0], 1uLL, 0x12uLL, fd_input, 0LL) != 3405643776LL )
{
    puts("Failed to mmap input file\n");
    exit(1LL);
}
```

```

58 if ( mmap(0xCAFE0000uLL, file_size[0], 1uLL, 0x12uLL, fd_input, 0LL) != 3405643776LL )
59 {
60     puts("Failed to mmap input file\n");
61     exit(1LL);
62 }
63 v11 = open((const char *)argv[3], 578, v10 ^ 0x1A4u);
64 if ( v11 < 0 )
65 {
66     puts("Failed to open output file\n");
67     exit(1LL);
68 }
69 fd_output = v11;
70 if ( lseek(v11, file_size[0] - 1LL, 0) == -1 )
71 {
72     puts("Failed to lseek output file to begin\n");
73     exit(1LL);
74 }
75 if ( write(fd_output, (const char *)&unk_426200, 1uLL) == -1 )
76 {
77     puts("Failed to write to output file\n");
78     exit(1LL);
79 }
80 if ( mmap(0x42420000uLL, file_size[0], 3uLL, 0x11uLL, fd_output, 0LL) != 1111621632 )
81 {
82     puts("Failed to mmap output file\n");
83     exit(1LL);
84 }
85 ((void (*)(void))loc_401485)();
86 goodboy = returned_r8 == 0xACED;
87 munmap(fd_input, file_size[0]);
88 munmap(fd_output, file_size[0]);
89 close(fd_input);
90 close(fd_output);
91 if ( goodboy )
92 {
93     puts("Aced it ! \\°/\n");
94     exit(0LL);
95 }
96 puts("Thou shall Halt and Catch Fire /!\\\n");
97 exit(1LL);
98 }

```

About 31000 mov
instructions...

```
.text:0000000004014FF loc_4014FF: ; DATA XREF: .text:loc_4014FF10
.text:0000000004014FF ; .text:00000000040D4BB10
.text:0000000004014FF mov rax, offset loc_4014FF
.text:000000000401506 mov r8, offset unk_426320
.text:00000000040150D mov rbx, [r8]
.text:000000000401510 mov qword_426330, rax
.text:000000000401518 mov qword_426340, rbx
.text:000000000401520 mov rax, 0
.text:000000000401527 mov rbx, rax
.text:00000000040152A mov rdx, rax
.text:00000000040152D mov r8, offset qword_426330
.text:000000000401534 mov r9, offset qword_426340
.text:00000000040153B mov cl, 1
.text:00000000040153D mov al, [r8]
.text:000000000401540 mov bl, [r9]
.text:000000000401543 mov rsi, offset unk_426400
.text:00000000040154A mov byte ptr [rsi+rax], 0
.text:00000000040154E mov [rsi+rbx], cl
.text:000000000401551 mov cl, [rsi+rax]
.text:000000000401554 mov al, [r8+1]
.text:000000000401558 mov bl, [r9+1]
.text:00000000040155C mov rsi, offset unk_426400
.text:000000000401563 mov byte ptr [rsi+rax], 0
.text:000000000401567 mov [rsi+rbx], cl
.text:00000000040156A mov cl, [rsi+rax]
.text:00000000040156D mov al, [r8+2]
.text:000000000401571 mov bl, [r9+2]
.text:000000000401575 mov rsi, offset unk_426400
.text:00000000040157C mov byte ptr [rsi+rax], 0
.text:000000000401580 mov [rsi+rbx], cl
.text:000000000401583 mov cl, [rsi+rax]
.text:000000000401586 mov al, [r8+3]
.text:00000000040158A mov bl, [r9+3]
.text:00000000040158E mov rsi, offset unk_426400
.text:000000000401595 mov byte ptr [rsi+rax], 0
.text:000000000401599 mov [rsi+rbx], cl
.text:00000000040159C mov cl, [rsi+rax]
.text:00000000040159F mov al, [r8+4]
.text:0000000004015A3 mov bl, [r9+4]
.text:0000000004015A7 mov rsi, offset unk_426400
.text:0000000004015AE mov byte ptr [rsi+rax], 0
.text:0000000004015B2 mov [rsi+rbx], cl
```

Step 3 - Litterature

The M/o/Vfuscator

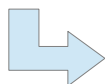
Turning 'mov' into a soul-crushing RE nightmare

{ domas, @xoreaxeaxeax

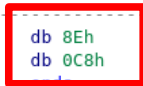
Step 3 - Blocks

- All blocks are sequentially executed
- Big loop
 - Forced exception in the end, signal handler restarts the loop

```
.text:0000000004014AC segv_handler proc near ; DATA
.text:0000000004014AC mov rsp, init_rsp
.text:0000000004014B4 retn
.text:0000000004014B4 segv_handler endp
.text:0000000004014B4
.text:0000000004014B5 ; -----
.text:0000000004014B5 ;
.text:0000000004014B5 loc_014B5: ; CODE
.text:0000000004014B5 mov rdi, 0Bh
.text:0000000004014B5 lea rsi, segv_handler
.text:0000000004014BC call set_SIGINT
.text:0000000004014C4 mov init_rsp, rsp
.text:0000000004014C9 mov rdi, 4
.text:0000000004014D1 lea rsi, sigill_handler
.text:0000000004014E0 call set_SIGINT
.text:0000000004014E5 sigill_handler: ; DATA
.text:0000000004014E5 mov rsp, init_rsp
```



```
.text:000000000424EB0 mov r8, offset aced
.text:000000000424EB7 mov r8, [r8+r15*8]
.text:000000000424EBB mov r8, [r8+r15*8]
.text:000000000424EBB ; -----
.text:000000000424EBF db 8Eh
.text:000000000424EC0 db 0C8h
.text:000000000424EC0 _text
.text:000000000424EC0
```



```
.text:0000000004014E5 sigill_handler: ; DATA XREF: .
.text:0000000004014E5 mov rsp, init_rsp
.text:0000000004014ED mov rax, 0
.text:0000000004014F4 mov r8, offset counter
.text:0000000004014FB mov [r8+r15*8], rax
.text:0000000004014FF
.text:0000000004014FF block_start_14ff: ; DATA XREF: .
.text:0000000004014FF ; .text:0000000
.text:0000000004014FF mov rax, offset block_start_14ff ;
.text:000000000401506 mov r8, offset enabled_block
.text:00000000040150D mov rbx, [r8]
.text:000000000401510 mov scratch_reg1, rax
.text:000000000401518 mov scratch_reg2, rbx
.text:000000000401520 mov rax, 0
.text:000000000401527 mov rbx, rax
.text:00000000040152A mov rdx, rax
.text:00000000040152D mov r8, offset scratch_reg1
```



- Mechanism to “enable” some blocks

Step 3 - Blocks

```
.text:0000000000413409 loc_413409: ; DATA XREF: .text:00000000004123EAro
.text:0000000000413409 ; .text:loc_413409io
.text:0000000000413409 mov rax, offset loc_413409
.text:0000000000413410 mov r8, offset enabled_block
.text:0000000000413417 mov rbx, [r8]
.text:000000000041341A mov scratch_reg1, rax
.text:0000000000413422 mov scratch_reg2, rbx
.text:000000000041342A mov rax, 0
.text:0000000000413431 mov rbx, rax
.text:0000000000413434 mov rdx, rax
.text:0000000000413437 mov r8, offset scratch_reg1
.text:000000000041343E mov r9, offset scratch_reg2
.text:0000000000413445 mov cl, 1
.text:0000000000413447 mov al, [r8]
.text:000000000041344A mov bl, [r9]
.text:000000000041344D mov rsi, offset eq_test_array
.text:0000000000413454 mov byte ptr [rsi+rax], 0
.text:0000000000413458 mov [rsi+rbx], cl
.text:000000000041345B mov cl, [rsi+rax]
.text:000000000041345E mov al, [r8+1]
.text:0000000000413462 mov bl, [r9+1]

.text:0000000000412C62 mov [r10+r15*8+2], al
.text:0000000000412C67 mov al, [r8+r15*8+3]
.text:0000000000412C6C mov bl, [r9+r15*8+3]
.text:0000000000412C71 mov rsi, offset carry_arrays
.text:0000000000412C78 mov rsi, [rsi+rcx*8]
.text:0000000000412C7C mov dl, [rsi+rax]
.text:0000000000412C7F mov rsi, offset add_array
.text:0000000000412C86 mov rsi, [rsi+rcx*8]
.text:0000000000412C8A mov al, [rsi+rax]
.text:0000000000412C8D mov rsi, offset carry_arrays
.text:0000000000412C94 mov rsi, [rsi+rax*8]
.text:0000000000412C98 mov cl, [rsi+rbx]
.text:0000000000412C9B mov rsi, offset add_array
.text:0000000000412CA2 mov rsi, [rsi+rax*8]
.text:0000000000412CA6 mov al, [rsi+rbx]
.text:0000000000412CA9 mov rsi, offset add_array
.text:0000000000412CB0 mov rsi, [rsi+rcx*8]
.text:0000000000412CB4 mov cl, [rsi+rdx]
.text:0000000000412CB7 mov [r10+r15*8+3], al
.text:0000000000412CBC mov al, [r8+r15*8+4]
.text:0000000000412CC1 mov bl, [r9+r15*8+4]
.text:0000000000412CC6 mov rsi, offset carry_arrays
```

Start of a basic block

→ checks if block address matches the “enabled_block”

If match → R15 = 0

Else → R15 = 1

Every “register” access is indexed by R15

Step 3 – Segv handler

- Segv handler to end execution
- Last block checks a value and dereference a “magic” (in r8) depending on the result

```
.text:000000000424EB0
.text:000000000424EB7
.text:000000000424EBB
.text:000000000424EBB ; -----
.text:000000000424EBF
.text:000000000424EC0
.text:000000000424EC0 _text
; .....
```

```
mov     r8, offset aced
mov     r8, [r8+r15*8]
mov     r8, [r8+r15*8]
```

```
db 8Eh
db 0C8h
ends
```

Step 3 - Arrays

■ Arrays of 256 arrays of 256 bytes

<code>.data:0000000000457E00</code>	<code>off_457E00</code>	<code>dq offset unk_458600</code>	<code>; DATA_XREF: .text:0000000000401612+o</code>	<code>.data:0000000000458D00</code>	<code>X_array_7</code>	<code>db 0</code>	<code>; DATA_XREF: .data:0000000000457E38+o</code>
<code>.data:0000000000457E08</code>		<code>dq offset unk_458700</code>	<code>; .text:0000000000401907+o ...</code>	<code>.data:0000000000458D01</code>		<code>db 1</code>	
<code>.data:0000000000457E10</code>		<code>dq offset unk_458800</code>		<code>.data:0000000000458D02</code>		<code>db 2</code>	
<code>.data:0000000000457E18</code>		<code>dq offset unk_458900</code>		<code>.data:0000000000458D03</code>		<code>db 3</code>	
<code>.data:0000000000457E20</code>		<code>dq offset unk_458A00</code>		<code>.data:0000000000458D04</code>		<code>db 4</code>	
<code>.data:0000000000457E28</code>		<code>dq offset unk_458B00</code>		<code>.data:0000000000458D05</code>		<code>db 5</code>	
<code>.data:0000000000457E30</code>		<code>dq offset unk_458C00</code>		<code>.data:0000000000458D06</code>		<code>db 6</code>	
<code>.data:0000000000457E38</code>		<code>dq offset unk_458D00</code>		<code>.data:0000000000458D07</code>		<code>db 7</code>	
<code>.data:0000000000457E40</code>		<code>dq offset unk_458E00</code>		<code>.data:0000000000458D08</code>		<code>db 0</code>	
<code>.data:0000000000457E48</code>		<code>dq offset unk_458F00</code>		<code>.data:0000000000458D09</code>		<code>db 1</code>	
<code>.data:0000000000457E50</code>		<code>dq offset unk_459000</code>		<code>.data:0000000000458D0A</code>		<code>db 2</code>	
<code>.data:0000000000457E58</code>		<code>dq offset unk_459100</code>		<code>.data:0000000000458D0B</code>		<code>db 3</code>	
<code>.data:0000000000457E60</code>		<code>dq offset unk_459200</code>		<code>.data:0000000000458D0C</code>		<code>db 4</code>	
				<code>.data:0000000000458D0D</code>		<code>db 5</code>	
				<code>.data:0000000000458D0E</code>		<code>db 6</code>	
				<code>.data:0000000000458D0F</code>		<code>db 7</code>	
				<code>.data:0000000000458D10</code>		<code>db 0</code>	
				<code>.data:0000000000458D11</code>		<code>db 1</code>	
				<code>.data:0000000000458D12</code>		<code>db 2</code>	

■ These are lookup tables for a “AND” operation

■ Ex: $7 \text{ AND } 0xE \rightarrow \text{array}[7][0xE] \rightarrow 6$

Step 3 – Identify all arrays

- **AND: 0x457E00**
- **OR: 0x447600**
- **XOR: 0x468600**
- **ADD: 0x426500**
 - Carrys: 0x436D00
- **NOT (boolean): 0x447500**

Step3 – Add 64bits

```
.text:000000000412E2F      mov     r8, offset scratch_reg1
.text:000000000412E36      mov     r9, offset scratch_reg2
.text:000000000412E3D      mov     r10, offset scratch_reg1
.text:000000000412E44      mov     al, [r8+r15*8]
.text:000000000412E48      mov     bl, [r9+r15*8]
.text:000000000412E4C      mov     rsi, offset carry_arrays
.text:000000000412E53      mov     rsi, [rsi+rcx*8]
.text:000000000412E57      mov     dl, [rsi+rax]
.text:000000000412E5A      mov     rsi, offset add_array
.text:000000000412E61      mov     rsi, [rsi+rcx*8]
.text:000000000412E65      mov     al, [rsi+rax]
.text:000000000412E68      mov     rsi, offset carry_arrays
.text:000000000412E6F      mov     rsi, [rsi+rcx*8]
.text:000000000412E73      mov     cl, [rsi+rbx]
.text:000000000412E76      mov     rsi, offset add_array
.text:000000000412E7D      mov     rsi, [rsi+rax*8]
.text:000000000412E81      mov     al, [rsi+rbx]
.text:000000000412E84      mov     rsi, offset add_array
.text:000000000412E8B      mov     rsi, [rsi+rcx*8]
.text:000000000412E8F      mov     cl, [rsi+rdx]
.text:000000000412E92      mov     [r10+r15*8], al
.text:000000000412E96      mov     al, [r8+r15*8+1]
.text:000000000412E9B      mov     bl, [r9+r15*8+1]
.text:000000000412EA0      mov     rsi, offset carry_arrays
.text:000000000412EA7      mov     rsi, [rsi+rcx*8]
.text:000000000412EAB      mov     dl, [rsi+rax]
.text:000000000412EAE      mov     rsi, offset add_array
.text:000000000412EB5      mov     rsi, [rsi+rcx*8]
.text:000000000412EB9      mov     al, [rsi+rax]
.text:000000000412EBC      mov     rsi, offset carry_arrays
.text:000000000412EC3      mov     rsi, [rsi+rax*8]
.text:000000000412EC7      mov     cl, [rsi+rbx]
.text:000000000412ECA      mov     rsi, offset add_array
.text:000000000412ED1      mov     rsi, [rsi+rax*8]
.text:000000000412ED5      mov     al, [rsi+rbx]
.text:000000000412ED8      mov     rsi, offset add_array
.text:000000000412EDF      mov     rsi, [rsi+rcx*8]
.text:000000000412EE3      mov     cl, [rsi+rdx]
.text:000000000412EE6      mov     [r10+r15*8+1], al
.text:000000000412EEB      mov     al, [r8+r15*8+2]
.text:000000000412EF0      mov     bl, [r9+r15*8+2]
.text:000000000412EF5      mov     rsi, offset carry_arrays
```

Beginning of a 64bits add

(here, only 2 bytes added)

Step3 – IF

```
.text:00000000004016D7 mov bl, [r9+r15+4]
.text:00000000004016DC mov rsi, offset eq_test_array
.text:00000000004016E3 mov byte ptr [rsi+rax], 0
.text:00000000004016E7 mov [rsi+rbx], cl
.text:00000000004016EA mov cl, [rsi+rax]
.text:00000000004016ED mov al, [r8+r15+5]
.text:00000000004016F2 mov bl, [r9+r15+5]
.text:00000000004016F7 mov rsi, offset eq_test_array
.text:00000000004016FE mov byte ptr [rsi+rax], 0
.text:0000000000401702 mov [rsi+rbx], cl
.text:0000000000401705 mov cl, [rsi+rax]
.text:0000000000401708 mov al, [r8+r15+6]
.text:000000000040170D mov bl, [r9+r15+6]
.text:0000000000401712 mov rsi, offset eq_test_array
.text:0000000000401719 mov byte ptr [rsi+rax], 0
.text:000000000040171D mov [rsi+rbx], cl
.text:0000000000401720 mov cl, [rsi+rax]
.text:0000000000401723 mov al, [r8+r15+7]
.text:0000000000401728 mov bl, [r9+r15+7]
.text:000000000040172D mov rsi, offset eq_test_array
.text:0000000000401734 mov byte ptr [rsi+rax], 0
.text:0000000000401738 mov [rsi+rbx], cl
.text:000000000040173B mov cl, [rsi+rax]
.text:000000000040173E mov [r10+r15], cl ; counter == 0x11
.text:0000000000401742 mov rax, 0
.text:0000000000401749 mov r8, offset if_result_buf
.text:0000000000401750 mov al, [r8+r15]
.text:0000000000401754 mov rbx, r15
.text:0000000000401757 mov rsi, offset or_arrays
.text:000000000040175E mov rsi, [rsi+rax*8]
.text:0000000000401762 mov bl, [rsi+rbx] ; if res || r15
.text:0000000000401765 mov rax, offset block_17f4 ; if false
.text:000000000040176C mov r8, offset enabled_block
.text:0000000000401773 mov [r8+rbx*8], rax
.text:0000000000401777 mov rax, 0
.text:000000000040177E mov rbx, rax
.text:0000000000401781 mov r8, offset if_result_buf
.text:0000000000401788 mov r9, offset if_result_buf
.text:000000000040178F mov al, [r8+r15]
.text:0000000000401793 mov rsi, offset bool_not_array
.text:000000000040179A mov bl, [rsi+rax]
.text:000000000040179D mov [r9+r15], bl
.text:00000000004017A1 mov rax, 0
.text:00000000004017A8 mov r8, offset if_result_buf
.text:00000000004017AF mov al, [r8+r15]
.text:00000000004017B3 mov rbx, r15
.text:00000000004017B6 mov rsi, offset or_arrays
.text:00000000004017BD mov rsi, [rsi+rax*8]
.text:00000000004017C1 mov al, [rsi+rbx]
.text:00000000004017C4 mov r15b, al
.text:00000000004017C7 mov rax, offset block_d4e8 ; if true
.text:00000000004017CE mov r8, offset enabled_block
.text:00000000004017D5 mov [r8+r15*8], rax
.text:00000000004017D9 mov rax, 1
.text:00000000004017E0 mov rbx, r15
.text:00000000004017E3 mov rsi, offset or_arrays
.text:00000000004017EA mov rsi, [rsi+rax*8]
.text:00000000004017EE mov al, [rsi+rbx]
.text:00000000004017F1 mov r15b, al
```

Depending on test, set “enabled block” to a block address

Step 3 – Strategy

- **“Patterns” can be recognized**
- **Deobfuscate all the things!**
 - Create labels for each block
 - Rename some variables
 - Identify all operations
- **From 31000 mov instructions...**
 - ... to ~500 lines of pseudo assembly

Step 3 – (not so dirty) Deobfuscator

```
38 for l in f:
39     l = l.rstrip()
40     ll = l.split(b"\t")
41     if len(ll) != 3:
42         continue
43     addr = int(ll[0].split(b":")[0],16)
44     taddr = b"0x%x" % addr
45     if b"mov    rax" in ll[2] and taddr in ll[2]:
46         #print("XXX","New block %x" % addr)
47         labels[addr] = b"label_%03x" % block_num
48         block_num += 1
49         blocks[block_addr] = block_ins
50         block_addr = addr
51         block_ins = []
52     block_ins.append(ll)
53 blocks[block_addr] = block_ins
54
```

Step 3 – (not so dirty) Deobfuscator

```
38 for l in f:
39     l = l.rstrip()
40     ll = l.split(b"\t")
41     if len(ll) != 3:
42         continue
43     addr = int(ll[0].split(b":")[0],16)
44     taddr = b"0x%x" % addr
45     if b"mov    rax" in ll[2] and taddr in ll[2]:
46         #print("XXX","New block %x" % addr)
47         labels[addr] = b"label_%03x" % block_num
48         block_num += 1
49         blocks[block_addr] = block_ins
50         block_addr = addr
51         block_ins = []
52     block_ins.append(ll)
53     blocks[block_addr] = block_ins
54
119 #XOR
120 for b in blocks:
121     if starts = []
122     if stops = []
123     in_if = False
124     for i in range(len(blocks[b])):
125         curr = blocks[b]
126         if curr[0] is None:
127             continue
128         if i < (len(blocks[b])-6) and b"mov    r8" in curr[i][2]:
129             if b"mov    r9" in curr[i+1][2]:
130                 if b"mov    r10," in curr[i+2][2]:
131                     if b"mov    rsi,0x468600" in curr[i+5][2]:
132                         v1 = label(curr[i][2].split(b"r8,")[1])
133                         v2 = label(curr[i+1][2].split(b"r9,")[1])
134                         v3 = label(curr[i+2][2].split(b"r10,")[1])
135                         if_starts.append([i,v1,v2,v3])
136                         in_if = True
137                     if in_if and b"mov    BYTE PTR [r10+r15*1],al" in curr[i][2]:
138                         if_stops.append(i)
139                         in_if = False
140     assert(len(if_starts) == len(if_stops))
141     for v in range(len(if_starts)-1, -1, -1):
142         blocks[b] = blocks[b][:if_starts[v][0]] + [[None,b"XOR",if_starts[v][1],if_starts[v][2],if_starts[v][3]]] + blocks[b][if_stops[v]+1:]
143
```

Step 3 – (not so dirty) Deobfuscator

```
38 for l in f:
39     l = l.rstrip()
40     ll = l.split(b"\t")
41     if len(ll) != 3:
42         continue
43     addr = int(ll[0])
44     taddr = b"0x%x" %
45     if b"mov    rax":
46         #print("XXX",
47             labels[addr]
48             block_num +=
49             blocks[block_
50             block_addr =
51             block_ins = []
52     block_ins.append(
53     blocks[block_addr] =
54
609 for b in blocks:
610     print(" %s" % label(b).decode())
611     for i in blocks[b]:
612         if i[0] is None:
613             if i[1] == b"MOV":
614                 if i[2] == b"enable blk":
615                     print("\tJMP %s" % (i[3].decode()))
616                 else:
617                     print("\tMOV %s, %s" % (i[2].decode(), i[3].decode()))
618             elif i[1] == b"MOVR":
619                 if i[2] == b"enable blk":
620                     print("\tJMP %s" % (i[3].decode()))
621                 else:
622                     print("\tMOV %s, %s" % (i[2].decode(), i[3].decode()))
623             elif i[1] == b"MOVRX":
624                 print("\tMOV %s, %s" % (i[2].decode(), i[3].decode()))
625             elif i[1] == b"MOVB":
626                 print("\tMOV.B %s, %s" % (i[2].decode(), i[3].decode()))
627             elif i[1] == b"MOVRX":
628                 print("\tMOV (%s) %s, %s" % (i[4].decode(), i[2].decode(), i[3].decode()))
629             elif i[1] == b"MOVB":
630                 print("\tMOV.B (%s) %s, %s" % (i[4].decode(), i[2].decode(), i[3].decode()))
631             elif i[1] == b"MOVB2":
632                 print("\tMOV.B [%s], %s" % (i[2].decode(), i[3].decode()))
633             elif i[1] == b"MOVB2":
634                 print("\tMOV.B [%s], %s" % (i[2].decode(), i[4].decode(), i[3].decode()))
635             elif i[1] == b"MOVBADD":
636                 print("\tMOV.B [%s+%s], %s" % (i[3].decode(), i[2].decode(), i[4].decode()))
637             elif i[1] == b"MOVBADD2":
638                 print("\tMOV.B %s, [%s+%s]" % (i[2].decode(), i[3].decode(), i[4].decode()))
639             elif i[1] == b"MOVB2":
640                 print("\tMOV.B [%s], [%s+%s]" % (i[2].decode(), i[3].decode(), i[4].decode()))
641             elif i[1] == b"MOVBX":
642                 print("\tMOV.B (%s) %s, [%s]" % (i[4].decode(), i[2].decode(), i[3].decode()))
643             elif i[1] == b"MOVBX2":
644                 print("\tMOV.B (%s) [%s], %s" % (i[4].decode(), i[3].decode(), i[2].decode()))
645             elif i[1] == b"NOP":
646                 continue
647             elif i[1] == b"ADD":
648                 print("\tADD %s, %s, %s" % (i[4].decode(), i[2].decode(), i[3].decode()))
649             elif i[1] == b"XOR":
650                 print("\tXORB %s, %s, %s" % (i[4].decode(), i[2].decode(), i[3].decode()))
651             elif i[1] == b"GOODBOY":
652                 print("\tGOODBOY")
653             elif i[1] == b"BADBOY":
654                 print("\tBADBOY")
655             elif i[1] == b"IFEQ":
656                 if (len(i[4]) == 2):
657                     print("\tIF (%s == %s) GOTO %s ELSE %s" % (i[3].decode(), i[2].decode(), i[4][1].decode(), i[4][0].decode()))
658                 else:
659                     print("\tIF (%s == %s) GOTO %s" % (i[3].decode(), i[2].decode(), i[4][0].decode()))
660             else:
661                 print(i)
662         else:
663             print(i)
664
],if_starts[v][3]] + blocks[b][if_stops[v]+1:]
```

Step 3 – (not so dirty) Deobfuscator

```
$ python simp.py listing | head -n 40
_start
    MOV counter, 0x0
_label_000
    MOV var_000, 0x11
    IF (var_000 == counter) GOTO label_01a ELSE label_001
_label_001
    MOV var_000, 0x0
    MOV (0x0) var_010, var_000
    ADD var_010, var_010, var_010
    MOV var_011, input_key
    ADD var_010, var_010, var_011
    MOV.B var_03b, [var_010]
    MOV var_000, 0x0
    MOV var_013, 0x20
    MOV var_001, 0x0
    MOV var_012, 0x0
_label_002
    IF (counter == var_001) GOTO label_004 ELSE label_003
_label_003
    ADD var_012, var_012, var_013
    MOV var_014, 0x1
    ADD var_001, var_001, var_014
    JMP label_002
_label_004
    MOV var_013, var_012
    ADD var_015, var_000, var_000
    ADD var_013, var_013, var_015
    MOV var_015, weird
    ADD var_013, var_013, var_015
    MOV.B (r8) var_00e, [var_013]
    XORB var_00f, var_03b, var_00e
    MOV var_000, 0x0
    MOV var_017, 0x20
    MOV var_002, 0x0
    MOV var_016, 0x0
_label_005
    IF (counter == var_002) GOTO label_007 ELSE label_006
_label_006
    ADD var_016, var_016, var_017
    MOV var_018, 0x1
```

Step 3 – Big picture

- **AES Sbox**
- **“Hardcoded” key schedule**
 - 4 bytes secret input used to alter key schedule
- **CTR mode**
- **Weird # of rounds (17 ?)**
- **A hash is computed during the deciphering of the file**
 - And compared to a hardcoded one
- **Let's bruteforce!**

Step 3 – Bruteforce

- Took a few hours
- Python executed using pypy
- 4 instances running

```
fab@sawfish: ~/chalthcon2024/step3/all-cries 107x27
167, 79, 201, 132, 176, 199, 254, 78, 40, 121, 228, 49, 29, 149, 116, 179, 1, 210, 43, 184, 227, 63, 159,
191, 78, 106, 164, 153, 173, 9, 10, 119, 229, 178, 154, 151, 7, 196, 1, 48, 217, 16, 170, 96, 111, 37, 76,
180, 150, 106, 27, 176, 93, 112, 52, 9, 163, 118, 228, 154, 69, 184, 255, 223, 168, 7, 111, 103, 252, 78, 2
36, 87, 16, 124, 75, 156, 228, 144, 27, 91, 198, 251, 152, 171, 78, 241, 49, 161, 37, 124, 27, 69, 76, 197,
235, 3, 16, 242, 17, 31, 166, 139, 89, 33, 114, 145, 250, 64, 154, 159, 155, 68, 236, 41, 14, 180, 52, 65,
177, 70, 98, 48, 61, 110, 147, 253, 146, 149, 182, 136]
16 [165, 28, 227, 200, 159, 4, 204, 249, 251, 249, 50, 119, 0, 149, 232, 11]
127

127
126
125
124
123
122
121
120
119
118
117
116
115
114
113
112
111
110

fab@sawfish: ~/chalthcon2024/step3/all-cries 108x27
69, 38, 126, 242, 254, 215, 38, 28, 209, 33, 231, 133, 57, 182, 40, 198, 3, 180, 21, 252, 26, 10, 188, 167,
79, 201, 132, 176, 199, 254, 78, 40, 121, 228, 49, 29, 149, 116, 179, 1, 210, 43, 184, 227, 63, 159, 191, 78
106, 164, 153, 173, 9, 10, 119, 229, 178, 154, 151, 7, 196, 1, 48, 217, 16, 170, 96, 111, 37, 76, 180, 150
106, 27, 176, 93, 112, 52, 9, 163, 118, 228, 154, 69, 184, 255, 223, 168, 7, 111, 103, 252, 78, 236, 87, 1
6, 124, 75, 156, 228, 144, 27, 91, 198, 251, 152, 171, 78, 241, 49, 161, 37, 124, 27, 69, 76, 197, 235, 3, 1
6, 242, 17, 31, 166, 139, 89, 33, 114, 145, 250, 64, 154, 159, 155, 68, 236, 41, 14, 180, 52, 65, 177, 70, 9
8, 48, 61, 110, 147, 253, 146, 149, 182, 136]
16 [165, 28, 227, 200, 159, 4, 204, 249, 251, 249, 50, 119, 0, 149, 232, 11]
126

127
126
125
124
123
122
121
120
119
118
117
116
115
114
113
112
111
110

fab@sawfish: ~/chalthcon2024/step3/all-cries 107x28
247, 210, 228, 148, 123, 170, 39, 155, 65, 119, 14, 63, 189, 51, 18, 45, 128, 254, 223, 59, 175, 33, 251, 9
5, 51, 136, 16, 66, 122, 152, 139, 24, 45, 205, 5, 117, 54, 98, 21, 237, 15, 32, 123, 149, 33, 37, 170, 88,
66, 29, 85, 196, 124, 245, 180, 22, 190, 57, 248, 174, 42, 156, 40, 180, 36, 149, 96, 122, 66, 175, 105, 1
77, 169, 38, 126, 242, 254, 215, 38, 28, 209, 33, 231, 133, 57, 182, 40, 198, 3, 180, 21, 252, 26, 10, 188,
167, 79, 201, 132, 176, 199, 254, 78, 40, 121, 228, 49, 29, 149, 116, 179, 1, 210, 43, 184, 227, 63, 159,
191, 78, 106, 164, 153, 173, 9, 10, 119, 229, 178, 154, 151, 7, 196, 1, 48, 217, 16, 170, 96, 111, 37, 76,
180, 150, 106, 27, 176, 93, 112, 52, 9, 163, 118, 228, 154, 69, 184, 255, 223, 168, 7, 111, 103, 252, 78, 2
36, 87, 16, 124, 75, 156, 228, 144, 27, 91, 198, 251, 152, 171, 78, 241, 49, 161, 37, 124, 27, 69, 76, 197,
235, 3, 16, 242, 17, 31, 166, 139, 89, 33, 114, 145, 250, 64, 154, 159, 155, 68, 236, 41, 14, 180, 52, 65,
177, 70, 98, 48, 61, 110, 147, 253, 146, 149, 182, 136]
16 [165, 28, 227, 200, 159, 4, 204, 249, 251, 249, 50, 119, 0, 149, 232, 11]
125

127
126
125
124
123
122
121
120
119
118
117
116
115
114
113
112
111
110

fab@sawfish: ~/chalthcon2024/step3/all-cries 108x28
106, 164, 153, 173, 9, 10, 119, 229, 178, 154, 151, 7, 196, 1, 48, 217, 16, 170, 96, 111, 37, 76, 180, 150
106, 27, 176, 93, 112, 52, 9, 163, 118, 228, 154, 69, 184, 255, 223, 168, 7, 111, 103, 252, 78, 236, 87, 1
6, 124, 75, 156, 228, 144, 27, 91, 198, 251, 152, 171, 78, 241, 49, 161, 37, 124, 27, 69, 76, 197, 235, 3, 1
6, 242, 17, 31, 166, 139, 89, 33, 114, 145, 250, 64, 154, 159, 155, 68, 236, 41, 14, 180, 52, 65, 177, 70, 9
8, 48, 61, 110, 147, 253, 146, 149, 182, 136]
16 [165, 28, 227, 200, 159, 4, 204, 249, 251, 249, 50, 119, 0, 149, 232, 11]
124

127
126
125
124
123
122
121
120
119
118
117
116
115
114
113
112
111
110
109
108
```

Step 3 - Finally

```
FOUND FOR KEY 34 32 36 36
```

```
fab@sawfish:~/chal/thcon2024/step3/all-cries$ time ./this-is-no-xoreaxeaxeax.elf 4266 flag.enc flag.dec
Hi my good wanderer °/ That is damn movfuscated
Aced it ! \°/

real    0m16.447s
```

```
$ file flag.dec
```

```
flag.dec: JPEG image data, JFIF standard 1.01, resolution (DPI),
density 300x300, segment length 16, comment: "thc-2024-flag-
2kv0iayavqir6ybnfnipcryc6cr5r22zvmsnmys7eye6fgilk1qjlnsxyeb@m0rgan.net
", progressive, precision 8, 200x152, components 3
```





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



<https://twitter.com/synacktiv>



<https://synacktiv.com>