



# 42 Cyber Workshop

Mika, Fred, Mike, Frank



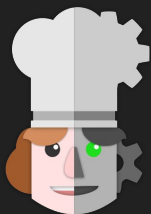
# Table of contents

- Encodings
- Crypto
- Web
- Binary Exploitation
- Misc

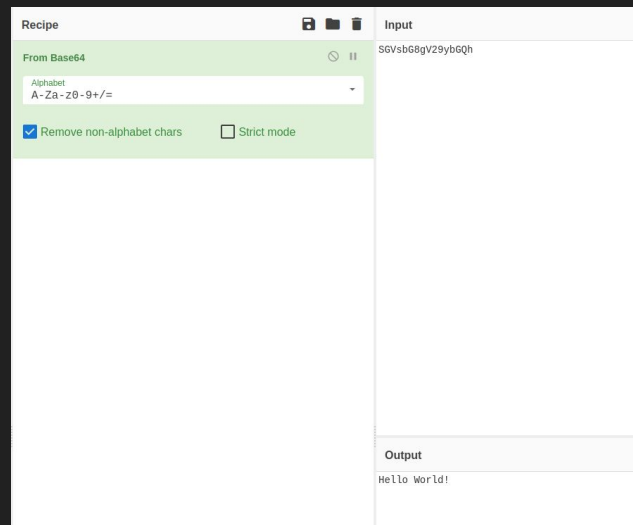
# Encodings

- Representation of data
  - 13 in decimal (base 10) is 0b1101 in binary (base 2) and 0xd in hexadecimal (0-f, base 16)
- Most common:
  - Base64
  - Hex (base 16)
- Resources:
  - Cyberchef: <https://gchq.github.io/CyberChef/> (Demo time)

Best tool :



# CyberChef

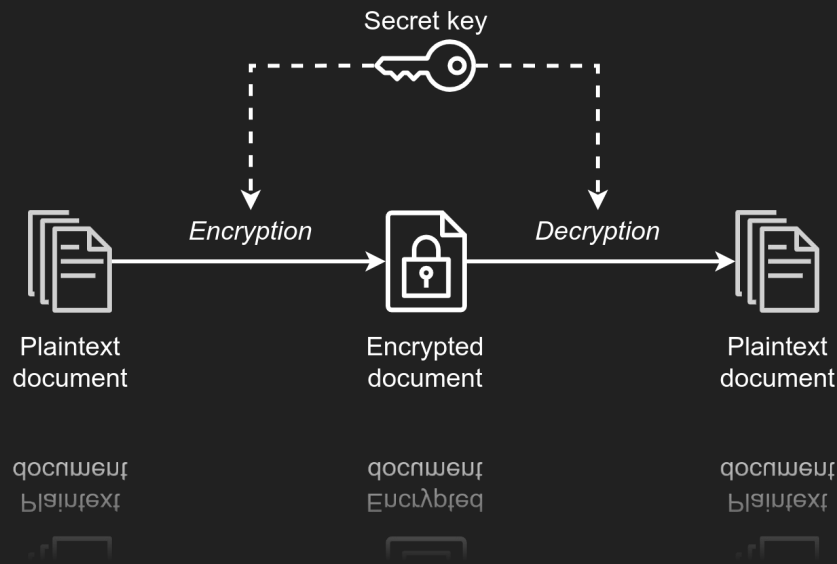


# Misc - OSINT

- Open Source Intelligence
- Find information on the internet
  - google search (with google dorks, advanced search)
  - Google maps, e.g. street view
  - reverse image search
  - social media
  - metadata of public images
- Tools: <https://osintframework.com/>, google dorks

# Crypto - Symmetric encryption

- The encryption key is the same as the decryption key
  - $m = c(\text{key})$
  - $m$ : message
  - $c$ : ciphertext
  - $\text{key}$ : encryption key
  - example: Caesar, xor, AES, DES

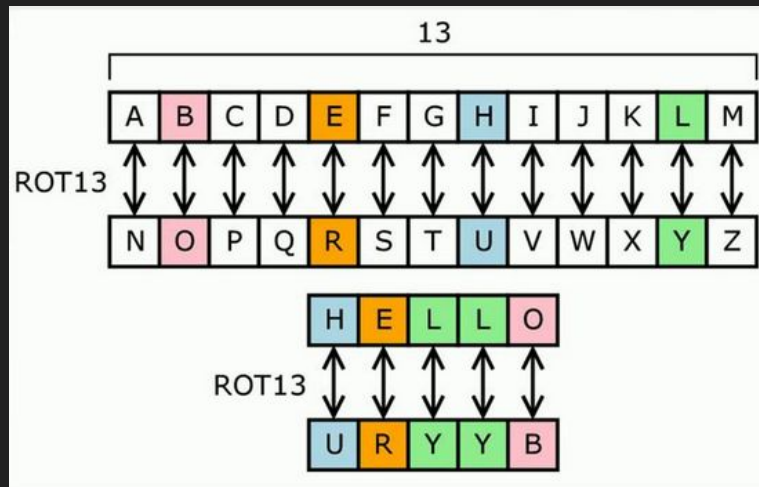


# Crypto - Asymmetric encryption

- The encryption key is the same as the decryption key
  - $c = \text{pubK}(m)$
  - $m = \text{privK}(c)$
  - $m$ : message
  - $c$ : ciphertext
  - $\text{pubK}$ : public key
  - $\text{privK}$ : private key
  - Examples: RSA, ECC

# Crypto - Substitution Ciphers

- Substitution Cipher
  - Caesar cipher being the most famous one
  - Can easily be cracked with frequency analysis.
  - e.g. “e” is the most common letter in English
- Vigenère Cipher
  - A substitution cipher with a password
- Resources:
  - <https://www.dcode.fr/> (demo time)



|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A B C D E F G H I J K L M N O P Q R S T U V W X Y Z |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| A   | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
| B   | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A |
| C   | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B |
| D   | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C |
| E   | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D |
| F   | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E |
| G   | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F |
| H   | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G |
| I   | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H |
| J   | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I |
| K   | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J |
| L   | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K |
| M   | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L |
| N   | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M |
| O   | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N |
| P   | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O |
| Q   | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P |
| R   | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q |
| S   | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R |
| T   | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S |
| U   | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T |
| V   | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U |
| W   | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V |
| X   | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W |
| Y   | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X |
| Z   | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y |

cipher VVVRBACP

key COVERCOVER...

plaintext THANKYOU

In encrypting plaintext, the cipher letter is found at the intersection of the column headed by the plaintext letter and the row indexed by the key letter. To decrypt ciphertext, the plaintext letter is found at the head of the column determined by the intersection of the diagonal containing the cipher letter and the row containing the key letter.



# Crypto - RSA (asymmetric)

- Encryption:  $c = m^e \bmod n$ 
  - $n=p*q$ ,  $e$  public key, is usually send along the message
  - $c$  - ciphertext
  - $m$  - plaintext message
- Decryption:  $m = c^d \bmod n$ 
  - $d = (p-1)*(q-1) = d$  private key. ( $p$  and  $q$  are two large prime numbers chosen by the user)
  - $n = p*q$
- Security
  - The security relies that  $n$  (which is public) cannot be factored, and  $p$  and  $q$  can not be found.
  - If  $n$  is factored,  $p$  and  $q$  can be found, and  $d$  can be calculated.

# Web - General

- basically everything that has to do with websites
  - websites are being used by everyone, everywhere
  - a lot of elements in web traffic can cause security issues
- most relevant protocol in web context: HTTP
  - A protocol is a set of rules on how to communicate
  - It can be compared to the rules of sentence construction in natural language
- web security = very extensive field
  - many different attack vectors and things to consider

# Web - HTTP - HyperText Transfer Protocol

- Method of communication between the client (e.g. browser) and the web-server
- The Client requests, and the server answers



Example of a HTTP GET request

- Several different HTTP methods:
  - **GET**: typically to fetch resources like html or javascript resources
  - **POST**: typically to send data to the web-server, e.g. when submitting credentials or uploading a new profile picture.
  - **PUT, DELETE, OPTIONS** etc.

# Web - Basics



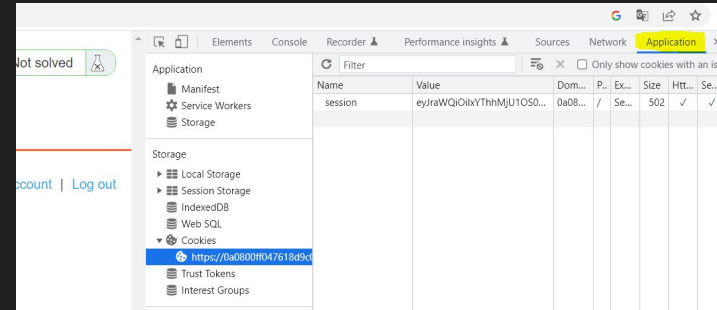
- Developer tools
  - Inspector → view HTML
  - Console → interactive JS console
  - Debugger → view files
  - Network → view requests
  - Style Editor → view CSS
  - Storage → Cookies

Demo time :)

# Web - Cookies

- to keep track of user sessions
- if someone logs in, they receive a cookie in the response
  - every subsequent request of the user with that cookie is recognized as logged in

```
Response
1 HTTP/1.1 302 Found
2 Location: /my-account
3 Set-Cookie: session=eyJraWQiOiJhOTJhM2Q0ZS1kYWI1LTQ0ZjMtOTViNS1lZDlkZmNiMmU4ZDgiLCJhbGciOiJSUzI1NiJ9.eyJpc3MiOiJwb3J0c3dpZ2dldiIsInN1YiI6ImdpZw5lciIsImV4cCI6MTYzMTAyNTUsImX0i6bnVudG4hbmVudG93a6A3L3p15zcIfGwYVVeHhMZH9P4A1PYQL_Skbv2hPZnGbuyl9CemAvssZ4tnfWAS38bUw1d8qyxq4S8f2wL7dvvGyQ6iJoLoB9vaYGObsUCRFaGjPJmYxzslntEScAJhdq00-6jch5DUBpL2vJhcBEbhvzwN1B23ERVY-MQ9X3aiXvWmsNgCSUVp4NnyDZF7oJQ3110jtJhacjOZaXf1FL3fORHmfzvwJb3uIsvSMFPLMb-q1LgQXzZYU-AROhwG9xzAgcD-QTp9-aYuJBZfZmSoY-38jBp6EbwPME7NpJQ; Secure; HttpOnly; SameSite=None
4 Connection: close
5 Content-Length: 0
```



# Web - Cookies: JSON Web Tokens

- Structure of JWTs: (1) Header, (2) Payload, (3) Signature
- Many potential weaknesses in implementation of JWTs
- Resources: [jwt.io](https://jwt.io)

**Request**

```
1 POST /login HTTP/1.1
2 Host: 0a1600f203d9bfbcc0ceb31300f60063.web-security-academy.net
3 Cookie: session=
4 Content-Length: 68
5 Cache-Control: max-age=0
6 Sec-Ch-Ua: " Not A;Brand";v="99", "Chromium";v="96"
7 Sec-Ch-Ua-Mobile: ?0
8 Sec-Ch-Ua-Platform: "Linux"
9 Upgrade-Insecure-Requests: 1
10 Origin: https://0a1600f203d9bfbcc0ceb31300f60063.web-security-academy.net
11 Content-Type: application/x-www-form-urlencoded
12 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.4472.164 Safari/537.36
13 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
14 Sec-Fetch-Site: same-origin
15 Sec-Fetch-Mode: navigate
16 Sec-Fetch-User: ?1
17 Sec-Fetch-Dest: document
18 Referer: https://0a1600f203d9bfbcc0ceb31300f60063.web-security-academy.net/login
19 Accept-Encoding: gzip, deflate
20 Accept-Language: en-GB,en-US;q=0.9,en;q=0.8
21 Connection: close
22
23 csrf=JLzJaDLZfK20wHXfVnwRXRQZpd7KZFvk&username=wiener&password=peter
```

**Response**

```
1 HTTP/1.1 302 Found
2 Location: /my-account
3 Set-Cookie: session=eyJraWQiOiJhOTJhM2Q0ZS1kYWI1LTQ0ZjMtOTViNS1lZDlkZmNiMmU4ZDgiLCJhbGciOiJSUzI1NiJ9.eyJpc3MiOiJwb3J0c3dpZ2dldiIsInN1YiI6IndpZmVlciiIsImV4cCI6MTY3MTAyNTU5MX0.WvnU0g4h8W1dVBI53Zka6AJ-po6GNNuq2J3S8ACu74-jjP15zciFgWYVVVehMZHP4A1PYQL_Skbv2hPZnGbuyl9CemAvssZ4tnRWA53Bbu1d8qyxq4X58f2WL7dwwGyQ6iJoLoB9vaYQbsUCRFeaGjPJmYxzs1n_tESCAJhdq00-6jch5DUBpL2vJHcBEbhvzwN1B23ERVY-MQ9X3aiXvWmsNgCSUVP4NnYDZF7oJQ3110jtJhac0jOzaXf1FL3fORHmfzvwJb3uIsvSMFPLmb-q1LgQXzZYU-AR0HwG9xzAgcD-QTP9-aYuJBZfZmS0y-3BjBp6EbwPME7NpJQ; Secure; HttpOnly; SameSite=None
4 Connection: close
5 Content-Length: 0

JWT String @ [Signature verification failed]

eyJraWQiOiJhOTJhM2Q0ZS1kYWI1LTQ0ZjMtOTViNS1lZDlkZmNiMmU4ZDgiLCJhbGciOiJSUzI1NiJ9.eyJpc3MiOiJwb3J0c3dpZ2dldiIsInN1YiI6IndpZmVlciiIsImV4cCI6MTY3MTAyNTU5MX0.WvnU0g4h8W1dVBI53Zka6AJ-po6GNNuq2J3S8ACu74-jjP15zciFgWYVVVehMZHP4A1PYQL_Skbv2hPZnGbuyl9CemAvssZ4tnRWA53Bbu1d8qyxq4X58f2WL7dwwGyQ6iJoLoB9vaYQbsUCRFeaGjPJmYxzs1n_tESCAJhdq00-6jch5DUBpL2vJHcBEbhvzwN1B23ERVY-MQ9X3aiXvWmsNgCSUVP4NnYDZF7oJQ3110jtJhac0jOzaXf1FL3fORHmfzvwJb3uIsvSMFPLmb-q1LgQXzZYU-AR0HwG9xzAgcD-QTP9-aYuJBZfZmS0y-3BjBp6EbwPME7NpJQ

Header
{
  "kid": "a92a3d4e-dab5-44f3-95b5-ed9dfcb2e8d8",
  "alg": "RS256"
}

Payload
{
  "iss": "portswigger",
  "sub": "wiener",
  "exp": 1671025591
}
```

# Web - SQL Injection

*Using apostrophes ('), quotation marks (") or backslashes (\) to break out of a string and manipulate the actual SQL.*



abc' UNION SELECT username,  
password FROM users; --

Website

SELECT name, type FROM products WHERE name =  
'\$userInput';

Database

Select name, type FROM products WHERE name = 'abc'  
UNION SELECT username, password FROM users; -- '

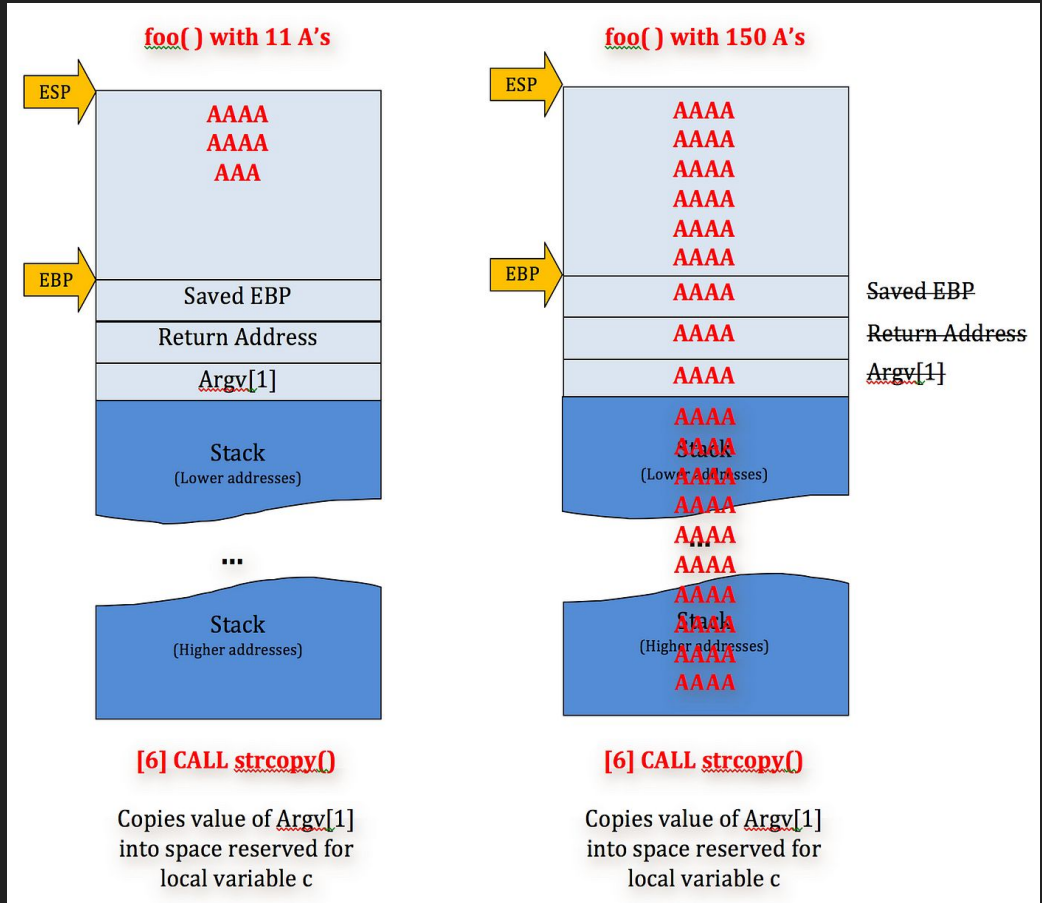
# Web - XSS - Cross-Site Scripting

- Client-side code injection
  - Execute code (javascript) in the browser of the victim.
  - Goal: Steal cookie, or session of the victim
- Stored XSS
  - An attacker is able to inject javascript through e.g. a comment in a forum. The text is persistent.
- Reflected XSS:
  - An attacker is able to inject javascript through a volatile method. E.g. through a HTTP parameter. The attacker has to send a link to a victim in order to exploit him



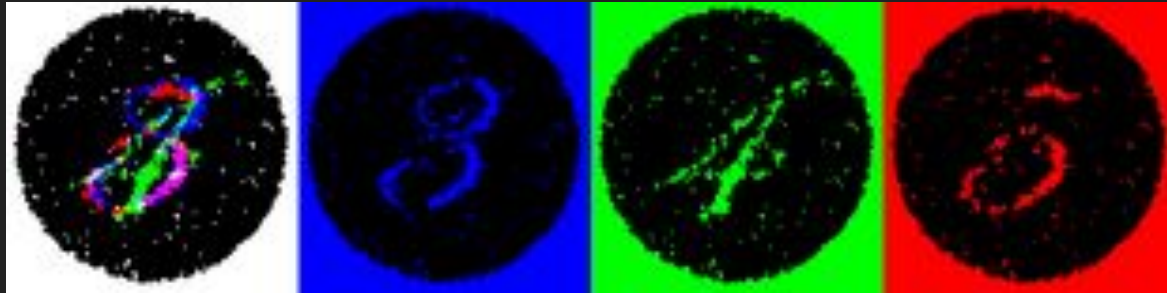
# Pwn - Buffer Overflow

- Out of buffer range write
- Able to overwrite values on the stack
- Integrity of the stack gets corrupted and variables can be overwritten.
- More advanced attacks can be used to execute arbitrary code.



# Misc - Steganography

- Hiding things in images
  - Metadata (e.g. time the image was created, author, size etc.)
  - Actual image (The actual pixel values)



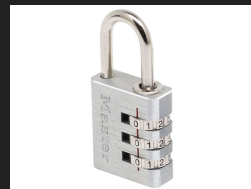
# Misc - Password cracking

- Try every combination (Brute force)
  - e.g. try 000 - 999 for a 3-digit code
  - Takes time
  - Trying all possible combinations increases exponentially with password length
- Can be done with password list - called a dictionary attack
  - Instead of all possible combinations, try the most likely ones.
  - Try weak passwords first, e.g. admin, password. 12345, 123456 ...
  - Create custom dictionaries by e.g. crawling the website and compose them yourself.
  - Tools: cewl, crunch



# Misc - Brute Forcing

- Online attacks
  - Launch the attack on an website or program itself
  - Tools: hydra, crackmapexec
- Offline
  - Extract the hash or a token that lets you crack the password independently of the application
  - john the ripper, hashcat



Good luck!

## Want more of this?

- hack.lu is happening at the moment!
- Join our discord server ([discord.letzpwn.lu](https://discord.letzpwn.lu))
  - Play CTFs together
  - Currently playing: hack.lu CTF
- Qualifier LCSC CTF from 1st of February-31st of March
  - For more info follow us on linkedin, twitter or X (@letzpwnasbl)