Last 4 of Us vs FIRST 2020 - 1:0

# ~~whoami~~ whoarewe

We are a new team. This was our first CTF as a team.

For 50% of the team, this was their first CTF, ever.

We plan to join FIRST this year. Not a member yet.

We have never been to a FIRST conference. It was our first. (Sorry)

As you could have guessed, some of us like PS4.

And most of us hate steganos :D

Our daily job is to reverse IoT malware, and analyze IoT exploits, vulnerabilities

And yes, our company's name is really a killer dog.

And yes, these are slides, not Python code, so we use AI instead of ML.

The team – through the eyes of this CTF

Albert Zsigovits
@albertzsigovits
🇭🇺

Forensics/PCAP expert

Dorka Palotay
@pad0rka
🇭🇺

Our ICS/Scada expert
Even though she did not know it.

Filip Savin
double agent with zero social media presence
🇱🇹

Expert in EVERYTHING
Seriously

Zoltan Balazs
@zh4ck
🇭🇺

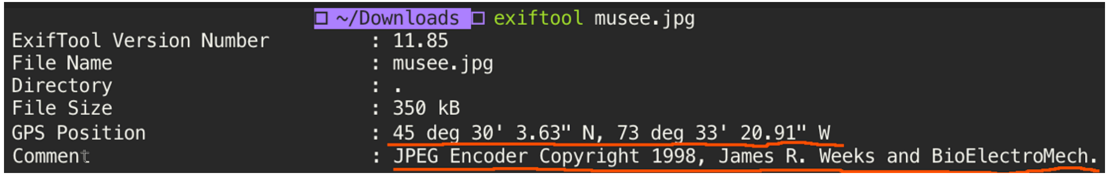
Expert in trolling in Mattermost

Expert in bossing around the others to document their findings

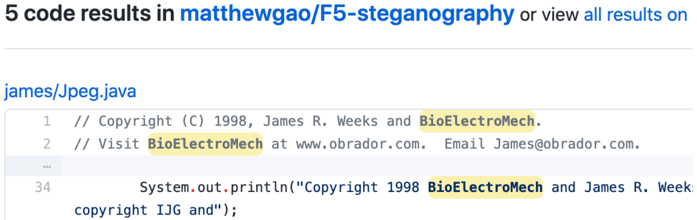
Follow us on Twitter ….

CUJO AI

Albert – The Museum challenge



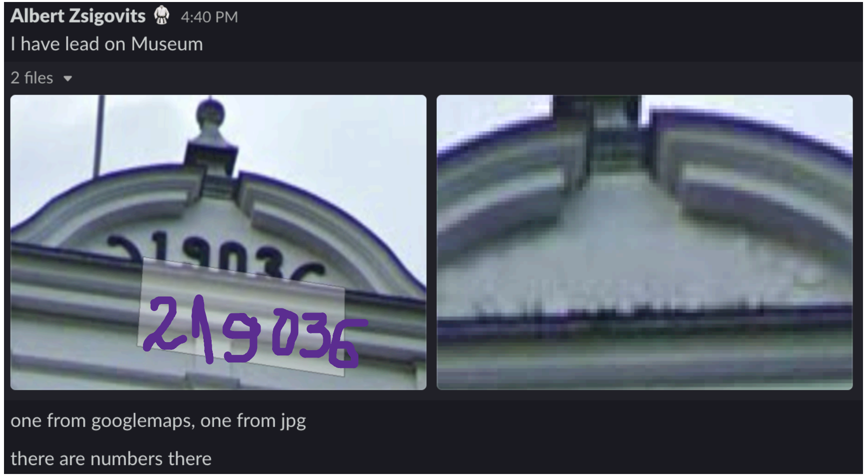
```
□ ~/Downloads □ exiftool musee.jpg
ExifTool Version Number         : 11.85
File Name                       : musee.jpg
Directory                       : .
File Size                       : 350 kB
GPS Position                    : 45 deg 30' 3.63" N, 73 deg 33' 20.91" W
Comment                         : JPEG Encoder Copyright 1998, James R. Weeks and BioElectroMech.
```



5 code results in matthewgao/F5-steganography or view all results on

james/Jpeg.java

```
// Copyright (C) 1998, James R. Weeks and BioElectroMech.
// Visit BioElectroMech at www.obrador.com.  Email James@obrador.com.
...
        System.out.println("Copyright 1998 BioElectroMech and James R. Week
copyright IJG and");
```



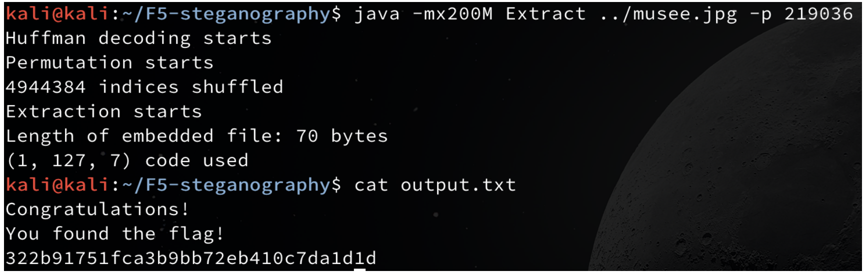
Albert Zsigovits 4:40 PM

I have lead on Museum

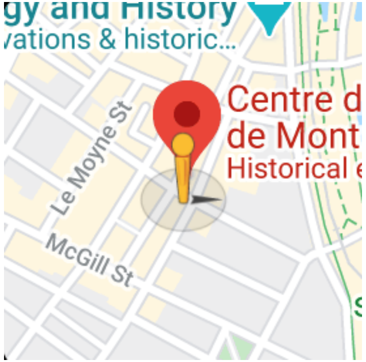
2 files ▾

one from googlemaps, one from jpg

there are numbers there

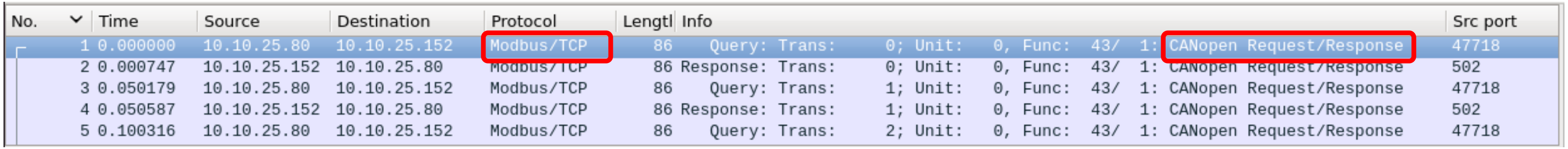


```
kali@kali:~/F5-steganography$ java -mx200M Extract ../musee.jpg -p 219036
Huffman decoding starts
Permutation starts
4944384 indices shuffled
Extraction starts
Length of embedded file: 70 bytes
(1, 127, 7) code used
kali@kali:~/F5-steganography$ cat output.txt
Congratulations!
You found the flag!
322b91751fca3b9bb72eb410c7da1d1d
```

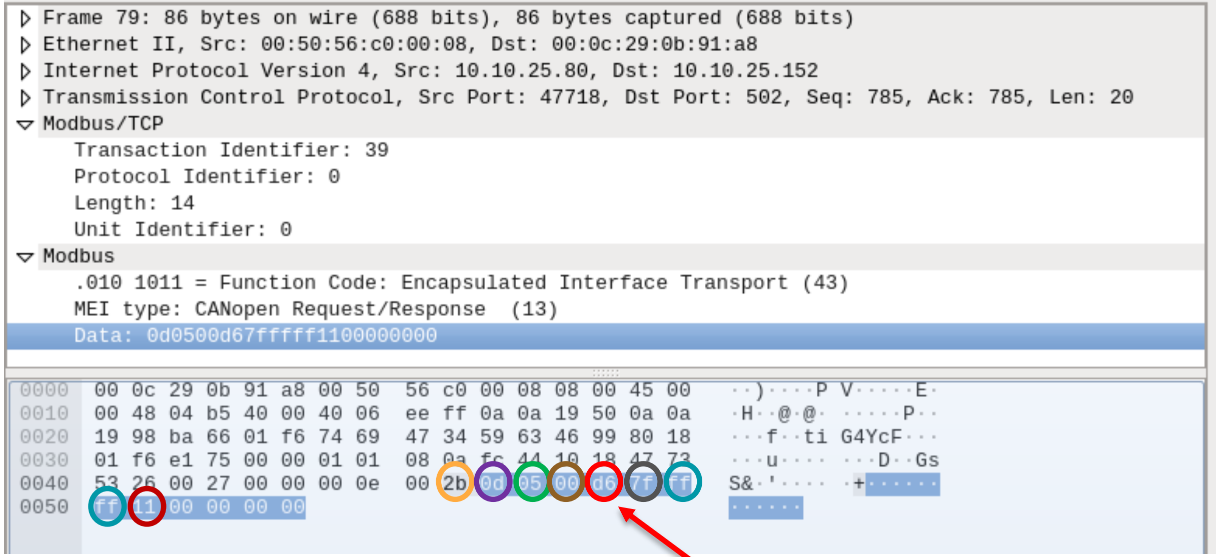


file
strings
xxd
uudeview
scalpel
foremost
openstego
stego.net
stegsnow
steghide
stegosuite
stegdetect
stegbreak
zsteg
stegsolver
steganabra
JavaStegano
F5-Steganography

Dorka – Weird Modbus Traffic



```
No.    Time       Source        Destination   Protocol    Lengtl Info                                                                    Src port
     1 0.000000   10.10.25.80   10.10.25.152  Modbus/TCP     86     Query: Trans:    0; Unit:   0, Func:  43/  1: CANopen Request/Response   47718
     2 0.000747   10.10.25.152  10.10.25.80   Modbus/TCP     86 Response: Trans:    0; Unit:   0, Func:  43/  1: CANopen Request/Response   502
     3 0.050179   10.10.25.80   10.10.25.152  Modbus/TCP     86     Query: Trans:    1; Unit:   0, Func:  43/  1: CANopen Request/Response   47718
     4 0.050587   10.10.25.152  10.10.25.80   Modbus/TCP     86 Response: Trans:    1; Unit:   0, Func:  43/  1: CANopen Request/Response   502
     5 0.100316   10.10.25.80   10.10.25.152  Modbus/TCP     86     Query: Trans:    2; Unit:   0, Func:  43/  1: CANopen Request/Response   47718
```



```
▷ Frame 79: 86 bytes on wire (688 bits), 86 bytes captured (688 bits)
▷ Ethernet II, Src: 00:50:56:c0:00:08, Dst: 00:0c:29:0b:91:a8
▷ Internet Protocol Version 4, Src: 10.10.25.80, Dst: 10.10.25.152
▷ Transmission Control Protocol, Src Port: 47718, Dst Port: 502, Seq: 785, Ack: 785, Len: 20
▽ Modbus/TCP
    Transaction Identifier: 39
    Protocol Identifier: 0
    Length: 14
    Unit Identifier: 0
▽ Modbus
    .010 1011 = Function Code: Encapsulated Interface Transport (43)
    MEI type: CANopen Request/Response  (13)
    Data: 0d0500d67fffff1100000000

0000  00 0c 29 0b 91 a8 00 50  56 c0 00 08 08 00 45 00   ··)····P V·····E·
0010  00 48 04 b5 40 00 40 06  ee ff 0a 0a 19 50 0a 0a   ·H··@·@· ·····P··
0020  19 98 ba 66 01 f6 74 69  47 34 59 63 46 99 80 18   ···f··ti G4YcF···
0030  01 f6 e1 75 00 00 01 01  08 0a fc 44 10 18 47 73   ···u···· ···D··Gs
0040  53 26 00 27 00 00 00 0e  00 2b 0d 05 00 d6 7f ff   S&·'···· ·+······
0050  ff 11 00 00 00 00                                  ······
```

Function code (0x2B)

MEI type (0x0D)

Protocol control

Reserved field

Network ID

Node ID (0x7F)

Index (0xFF, 0xFF)

Sub-index

Solution: D6

Challenge: Find the protocol description

CiA 309-2

Interfacing CANopen with TCP/IP

Part 2: Modbus/TCP mapping

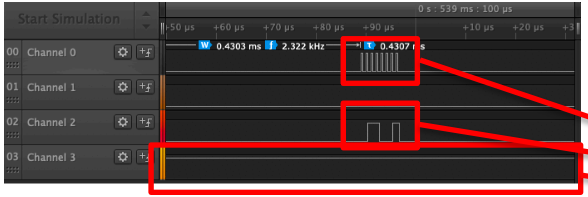
Table 1: Command codes used in the sub-index field

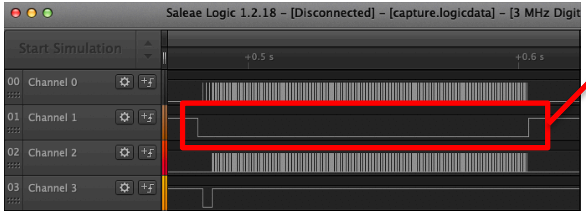
Code	Command
00_h	Reserved
01_h	GATEWAY_INITIALIZATION
02_h to 03_h	Reserved
04_h	START_ALL_NODES
05_h	PRE_OP_ALL_NODES
06_h	STORE_CONFIGURATION
07_h	Reserved
08_h	RESTORE_CONFIGURATION
09_h to 10_h	Reserved
11_h	STOP_ALL_NODES

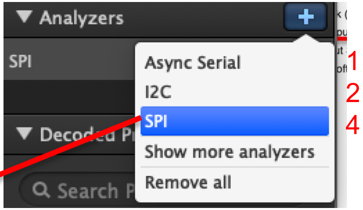
CUJOAI

Filip – PLC firmware injection (1/2)

1. Google "logicdata file" -> Saleae Logic soft
2. Load capture, map channels

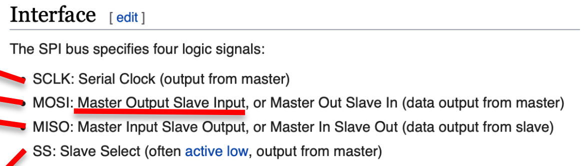






1 channel
2 channels
4 channels

Wikipedia on "SPI" (4 channels)

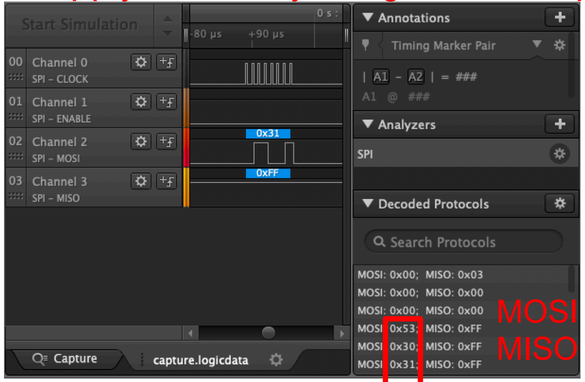


Interface [edit]

The SPI bus specifies four logic signals:

- SCLK: Serial Clock (output from master)
- MOSI: Master Output Slave Input, or Master Out Slave In (data output from master)
- MISO: Master Input Slave Output, or Master In Slave Out (data output from slave)
- SS: Slave Select (often active low, output from master)

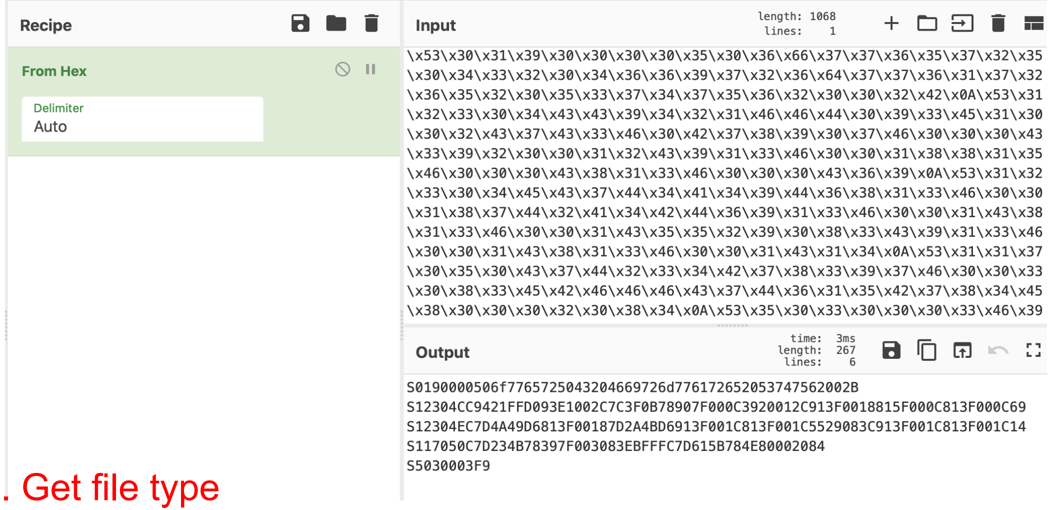
3. Apply SPI analyzer, get data, export to CSV



```
MOSI: 0x00;  MISO: 0x03
MOSI: 0x00;  MISO: 0x00
MOSI: 0x00;  MISO: 0x00
MOSI: 0x53;  MISO: 0xFF
MOSI: 0x30;  MISO: 0xFF
MOSI: 0x31;  MISO: 0xFF
```

MOSI data when MISO = 0xFF

4. Cyber chef from hex, save to file



From Hex

Delimiter: Auto

Input

length: 1068
lines: 1

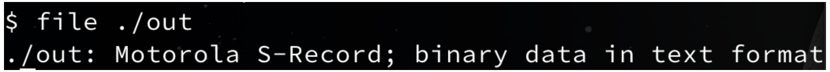
```
\x53\x30\x31\x39\x30\x30\x30\x30\x35\x30\x36\x66\x37\x37\x36\x35\x37\x32\x35
\x30\x34\x33\x32\x30\x34\x36\x36\x39\x37\x32\x36\x64\x37\x37\x36\x31\x37\x32
\x36\x35\x32\x30\x35\x33\x37\x34\x37\x35\x36\x32\x30\x30\x32\x42\x0A\x53\x31
\x32\x33\x30\x34\x43\x43\x39\x34\x32\x31\x46\x46\x44\x30\x39\x33\x45\x31\x30
\x30\x32\x43\x37\x43\x33\x46\x30\x42\x37\x38\x39\x30\x37\x46\x30\x30\x30\x43
\x33\x39\x32\x30\x30\x31\x32\x43\x39\x31\x33\x46\x30\x30\x31\x38\x38\x31\x35
\x46\x30\x30\x30\x43\x38\x31\x33\x46\x30\x30\x30\x43\x36\x39\x0A\x53\x31\x32
\x33\x30\x34\x45\x43\x37\x44\x34\x41\x34\x39\x44\x36\x38\x31\x33\x46\x30\x30
\x31\x38\x37\x44\x32\x41\x34\x42\x44\x36\x39\x31\x33\x46\x30\x30\x31\x43\x38
\x31\x33\x46\x30\x30\x31\x43\x35\x35\x32\x39\x30\x38\x33\x43\x39\x31\x33\x46
\x30\x30\x31\x43\x38\x31\x33\x46\x30\x30\x31\x43\x31\x34\x0A\x53\x31\x31\x37
\x30\x35\x30\x43\x37\x44\x32\x33\x34\x42\x37\x38\x33\x39\x37\x46\x30\x30\x33
\x30\x38\x33\x45\x42\x46\x46\x46\x43\x37\x44\x36\x31\x35\x42\x37\x38\x34\x45
\x38\x30\x30\x30\x32\x30\x38\x34\x0A\x53\x35\x30\x33\x30\x30\x30\x33\x46\x39
```

Output

time: 3ms
length: 267
lines: 6

```
S0190000506f776572504320466972d776172652053747562002B
S12304CC9421FFD093E1002C7C3F0B78907F000C3920012C913F0018815F000C813F000C69
S12304EC7D4A49D6813F00187D2A4BD6913F001C813F001C5529083C913F001C813F001C14
S117050C7D234B78397F003083EBFFFC7D615B784E80002084
S5030003F9
```

5. Get file type



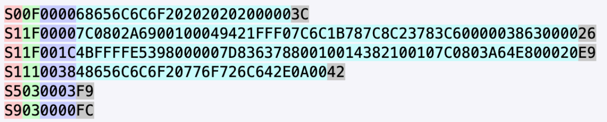
```
$ file ./out
./out: Motorola S-Record; binary data in text format
```

CUJO AI

Filip – PLC firmware injection (2/2)

If the integer 120 was passed as an argument to the function in the firmware stub, what would the function return?

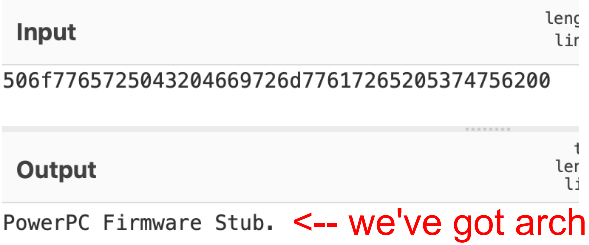
6. Google "motorola s-record" -> Wikipedia on S-REC file format, example:



```
S00F000068656C6C6F202020202000003C
S11F00007C0802A6900100049421FFF07C6C1B787C8C23783C6000003863000026
S11F001C4BFFFE5398000007D83637880010014382100107C0803A64E800020E9
S111003848656C6C6F20776F726C642E0A0042
S5030003F9
S9030000FC
```

niice, we've got something like this, S0 is header string

7. Cyber chef header from hex



Input

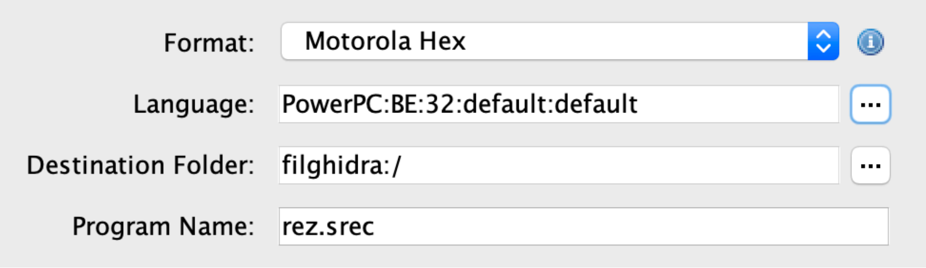
```
506f7765725043204669726d77617265205374756200
```

Output

```
PowerPC Firmware Stub.
```

<-- we've got arch!

8. Load our S-REC file into ghidra, with PPC arch:



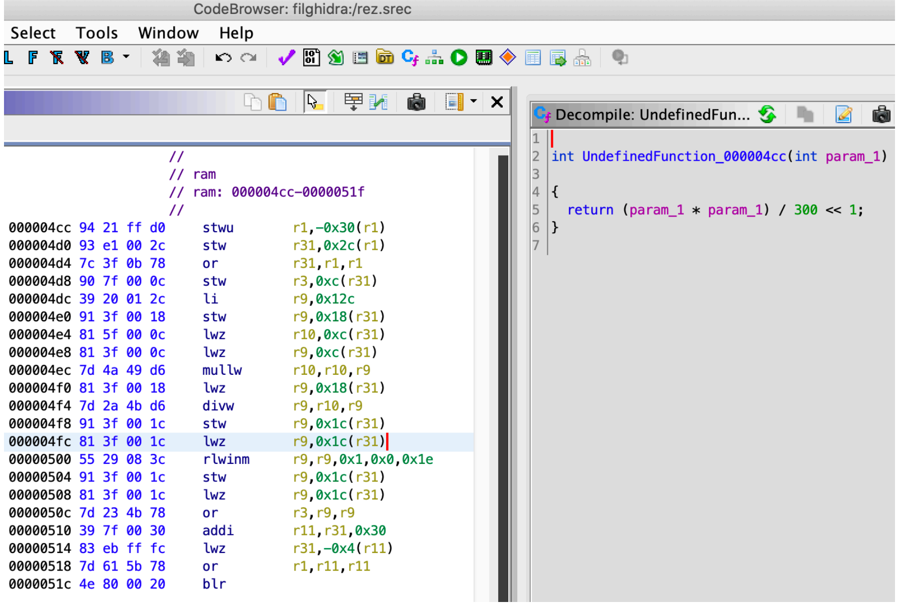
Format: Motorola Hex

Language: PowerPC:BE:32:default:default

Destination Folder: filghidra:/

Program Name: rez.srec

9. Decompile:



```
                        //
                        // ram
                        // ram: 000004cc-0000051f
                        //
000004cc 94 21 ff d0     stwu       r1,-0x30(r1)
000004d0 93 e1 00 2c     stw        r31,0x2c(r1)
000004d4 7c 3f 0b 78     or         r31,r1,r1
000004d8 90 7f 00 0c     stw        r3,0xc(r31)
000004dc 39 20 01 2c     li         r9,0x12c
000004e0 91 3f 00 18     stw        r9,0x18(r31)
000004e4 81 5f 00 0c     lwz        r10,0xc(r31)
000004e8 81 3f 00 0c     lwz        r9,0xc(r31)
000004ec 7d 4a 49 d6     mullw      r10,r10,r9
000004f0 81 3f 00 18     lwz        r9,0x18(r31)
000004f4 7d 2a 4b d6     divw       r9,r10,r9
000004f8 91 3f 00 1c     stw        r9,0x1c(r31)
000004fc 81 3f 00 1c     lwz        r9,0x1c(r31)
00000500 55 29 08 3c     rlwinm     r9,r9,0x1,0x0,0x1e
00000504 91 3f 00 1c     stw        r9,0x1c(r31)
00000508 81 3f 00 1c     lwz        r9,0x1c(r31)
0000050c 7d 23 4b 78     or         r3,r9,r9
00000510 39 7f 00 30     addi       r11,r31,0x30
00000514 83 eb ff fc     lwz        r31,-0x4(r11)
00000518 7d 61 5b 78     or         r1,r11,r11
0000051c 4e 80 00 20     blr
```

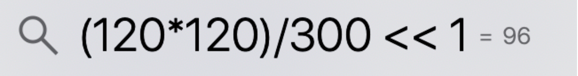
```

int UndefinedFunction_000004cc(int param_1)

{
  return (param_1 * param_1) / 300 << 1;
}

```

10. Result = **96**

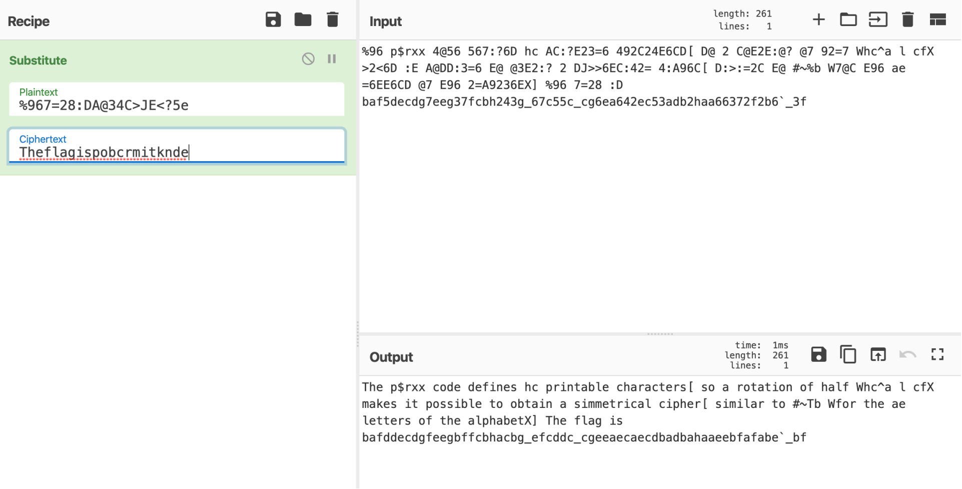


(120*120)/300 << 1 = 96

CUJOAI

Zoltan, @zh4ck – X/2 salad, a.k.a Half Caesar

I am the crypto guy, who instead of realizing this is ROT-47, goes all the way down and manually solves the challenge as a case sensitive substitution cipher.



Plaintext
`%967=28:DA@34C>JE<?5e`

Ciphertext
`Theflagispobcrmitknde`

```
length: 261
 lines:   1
```

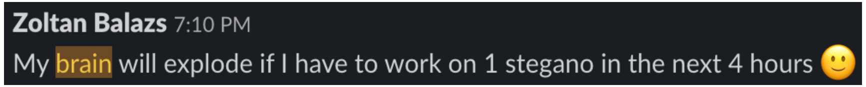
```
%96 p$rxx 4@56 567:?6D hc AC:?E23=6 492C24E6CD[ D@ 2 C@E2E:@? @7 92=7 Whc^a l cfX
>2<6D :E A@DD:3=6 E@ @3E2:? 2 DJ>>6EC:42= 4:A96C[ D:>:=2C E@ #~%b W7@C E96 ae
=6EE6CD @7 E96 2=A9236EX] %96 7=28 :D
baf5decdg7eeg37fcbh243g_67c55c_cg6ea642ec53adb2haa66372f2b6`_3f
```

```
  time:   1ms
length:   261
 lines:     1
```

```
The p$rxx code defines hc printable characters[ so a rotation of half Whc^a l cfX
makes it possible to obtain a simmetrical cipher[ similar to #~Tb Wfor the ae
letters of the alphabetX] The flag is
bafddecdgfeegbffcbhacbg_efcddc_cgeeaecaecdbadbahaaeebfafabe`_bf
```

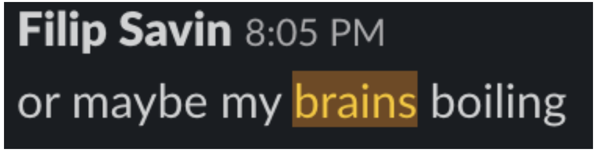
I am also the guy who when can't solve a challenge, instead of researching the topic, starts trolling the opponents with fake hints and trolls the organizers for looking bored.

Blood, sweat and tears



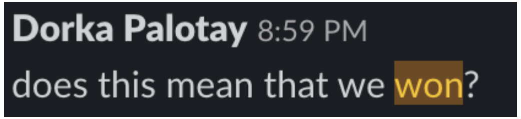
Zoltan Balazs 7:10 PM

My brain will explode if I have to work on 1 stegano in the next 4 hours 🙂



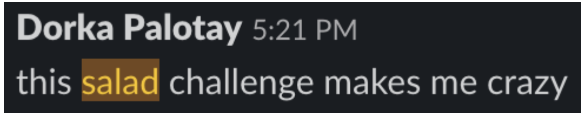
Filip Savin 8:05 PM

or maybe my brains boiling



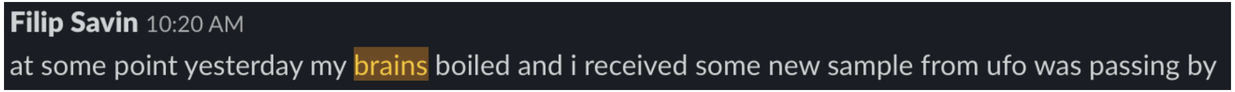
Dorka Palotay 8:59 PM

does this mean that we won?



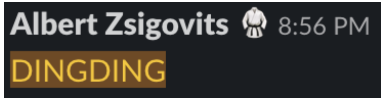
Dorka Palotay 5:21 PM

this salad challenge makes me crazy



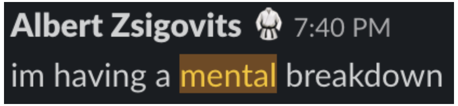
Filip Savin 10:20 AM

at some point yesterday my brains boiled and i received some new sample from ufo was passing by



Albert Zsigovits 🥋 8:56 PM

DINGDING

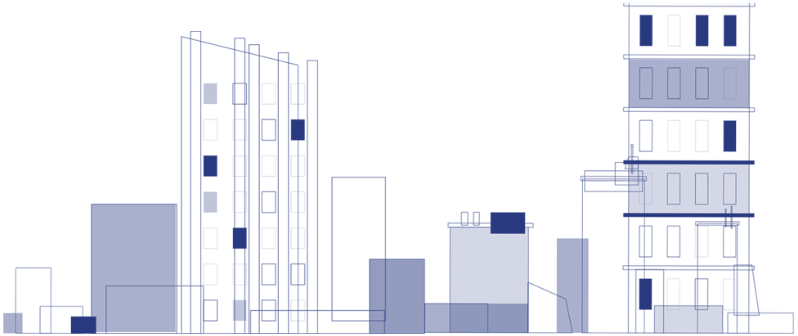


Albert Zsigovits 🥋 7:40 PM

im having a mental breakdown

We invite everyone to solve this STEGANO challenge

Hint: there is no solution



CUJO AI