root@localhost:~$ echo "b477l3 0f l337"

BATTLE_OF_1337

CTF 2022

BATTLE OF 1337 OFFICIAL WRITEUP

Writeup By : Trailbl4z3r

Trailbl4z3r

TABLE OF CONTENT

INTRO

Write-up produced for educational purposes only. I do not promote or encourage illegal hacking activity without written permission in general. I believe that ethical hacking, information security and cyber security should be familiar subject to anyone using digital information and computer.

There will always be more than one method to capture a flag in CTF. Methods shown here is just my method, there may be better or quicker or cleaner method. Challenges that are included in this write-up may also include challenges that I did not manage to solve in the competition or event period, but at the time of writing, I have the solution to the challenge, hence it is added in this write-up.

For the easiness of reading, some of my thought processes, struggles, and xxx are omitted in this write-up. When solving a challenge, typically we try many different paths and dead ends before finding the right path to solve it, this write-up will contain only the right paths that I found.

This write-up is made with blood and tears, Gotta Catch 'Em All™.

REVERSE ENGINEERING

SIMPLIFY

A simple crackme challenge, open the file in Ghidra, and then can see the disassembled code of the main function which validates if the user input is the flag.

To get the flag, just solve a few simple arithmetic.

```
undefined8 main(undefined8 param_1,undefined8 param_2)

{
  undefined8 in_R9;
  long in_FS_OFFSET;
  int local_20;
  uint local_1c;
  uint local_18;
  uint local_14;
  long local_10;

  local_10 = *(long *)(in_FS_OFFSET + 0x28);
  printf("Enter code: ");
  __isoc99_scanf("%i-%i-%i-%i",&local_20,&local_1c,&local_18,&local_14,in_R9,param_2);
  if ((((local_20 + local_14 * 3 == 18044) && (local_1c * local_18 * 3 == 5174190)) &&
      (local_20 == 1010)) && (local_18 * 49363 + local_14 == 63683948)) {
    printf("Correct code! The flag is %i-%i-%i-%i\n",0x3f2,(ulong)local_1c,(ulong)local_18,
           (ulong)local_14);
  }
  else {
    puts("Wrong code..");
  }
  if (local_10 != *(long *)(in_FS_OFFSET + 0x28)) {
                    /* WARNING: Subroutine does not return */
    __stack_chk_fail();
  }
  return 0;
}
```

Local_20 = 1010

Local_14 = (18044 – 1010) / 3 = 5678

Local_18 = (63683948 – 5678) / 49363 = 1290

Local_1c = 5174190 / 3 / 1290 = 1337

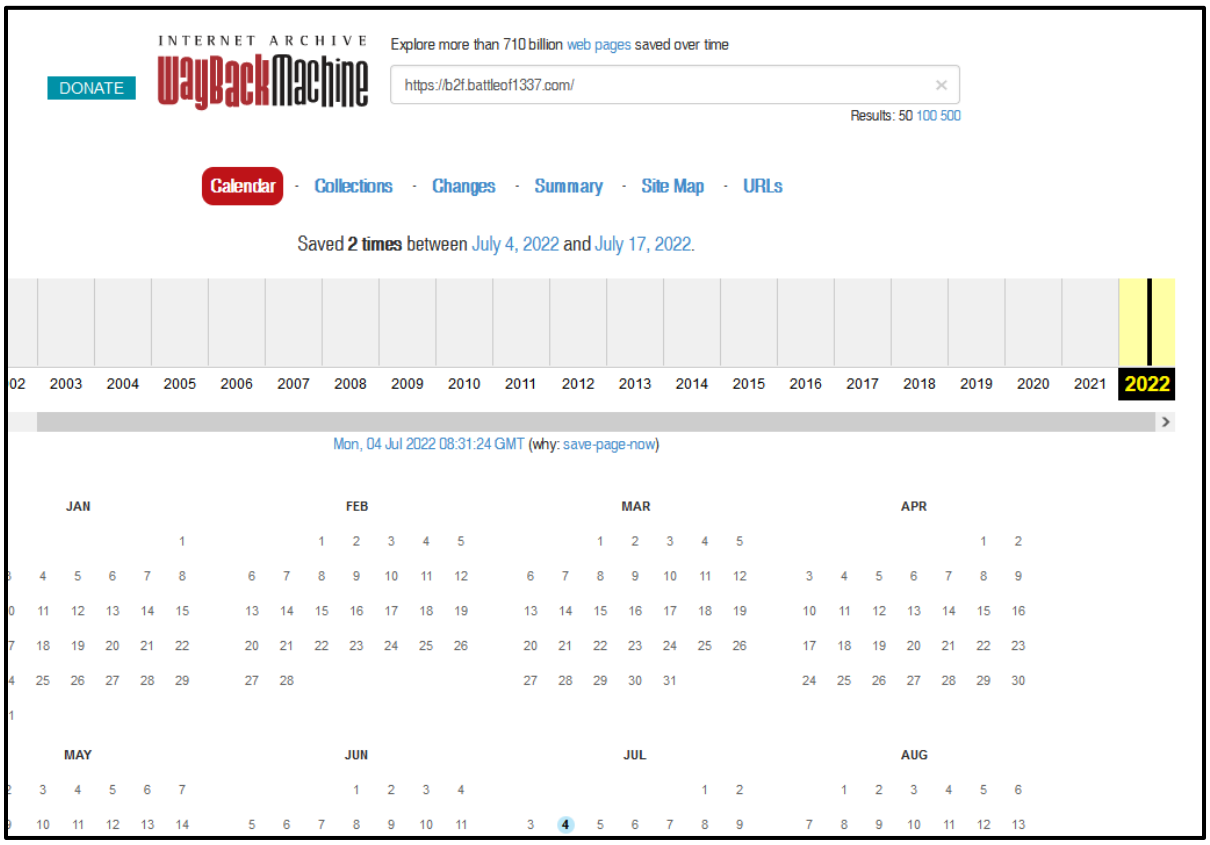
```
root@kali:/media/sf_VM_Shared/bo1337# ./crackme
Enter code: 1010-1337-1290-5678
Correct code! The flag is 1010-1337-1290-5678
root@kali:/media/sf_VM_Shared/bo1337#
```

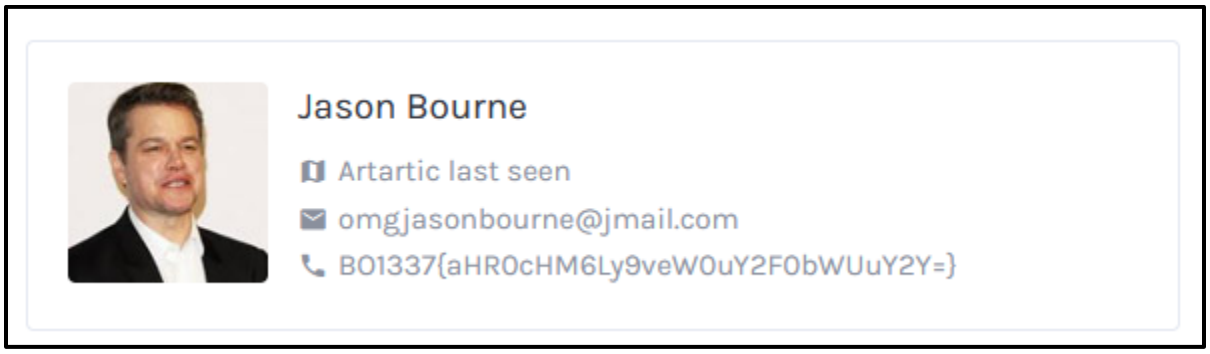
FLAG: BO1337{1010-1337-1290-5678}

OSINT

BACK TO THE FUTURE

I managed to solve this one under 1 minute when the CTF started, go check the recording. *wink* The title and the description directly point to Wayback Machine. Throw in the URL, pick the snapshot on July 4th, ???, profit!

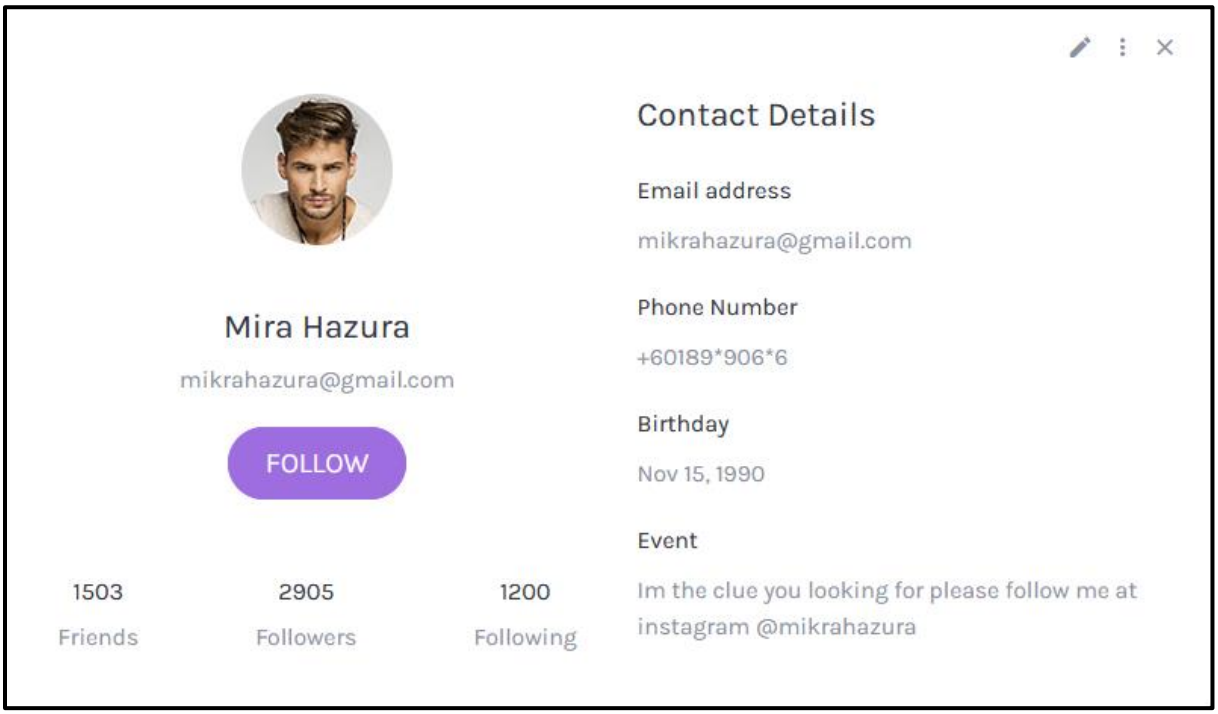




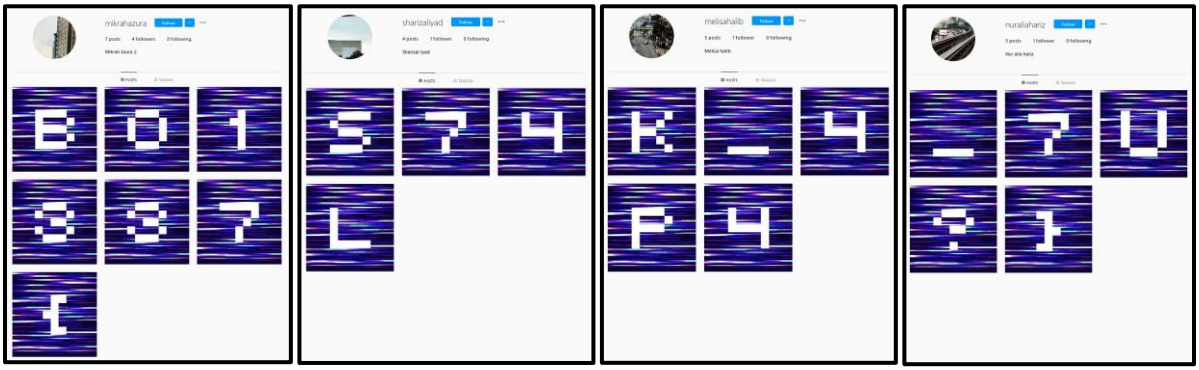
FLAG: BO1337{aHR0cHM6Ly9veW0uY2F0bWUuY2Y=}

1GRAM

Back to the current page or stay in the past but click on the website. No matter who you click, a new profile will be popped up, asking u to follow the guy on Instagram. Don't worry, flag can be obtained without following the guy.



His posts contain the start of the flag, one of the pictures tagged several other users. One of the users contain next part of the flag in their posts, rinse and repeat.

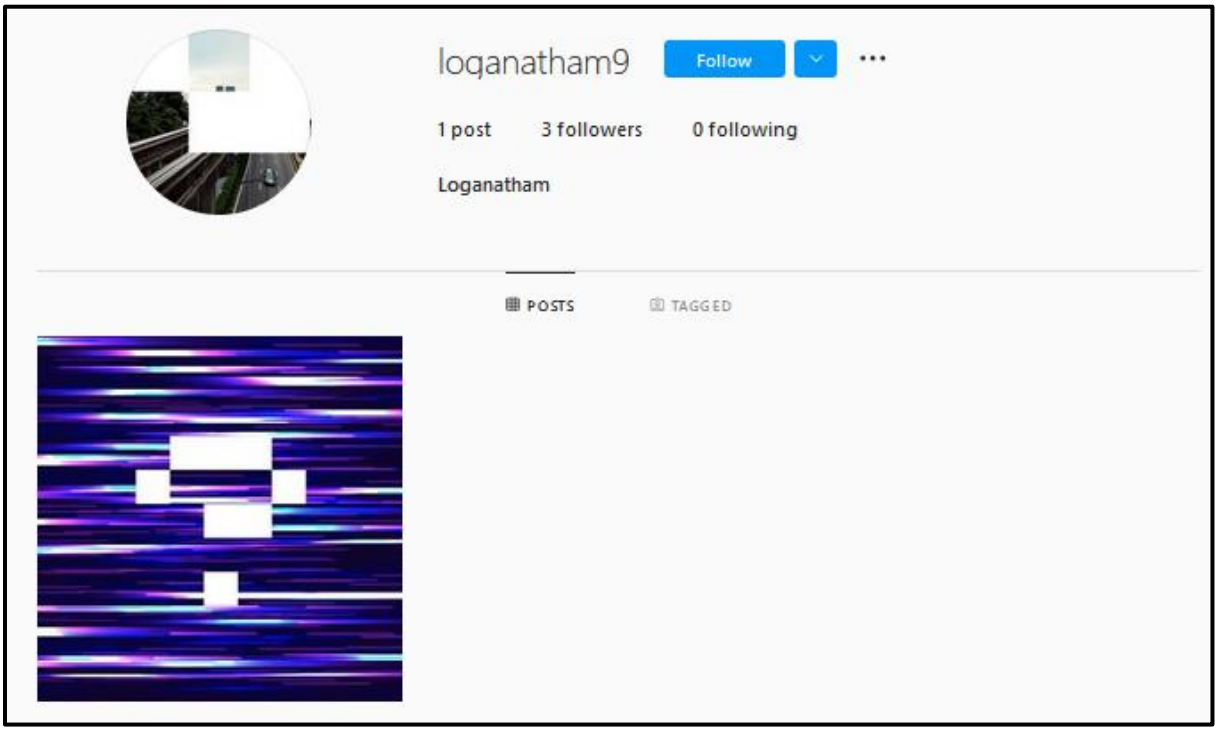


Flag: BO1337{S74LK_4P4_7U?}

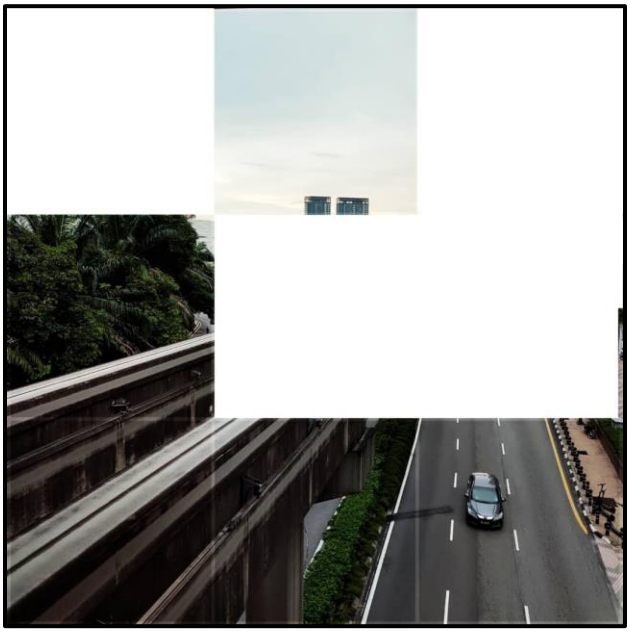
SNAP

Thanks to this challenge, I now know the whole Monorail route of Malaysia.

From the last Instagram account in the previous OSINT challenge, one of the pictures still tagged with another account. Going to that account only have one question mark picture, with the question "dimanakah saya?", and a very sus display picture.



Extract the Display Picture:



And it's time for a treasure hunt.

From the half-censored display picture, the rail can be identified that it is monorail. MRT, LRT, ETS rail doesn't look like that. The picture is either taken by someone on a monorail or at the monorail station. Then, it is Google Map time.

After virtually touring the whole Monorail line in Malaysia, the exact spot can be identified that it is from the Imbi Monorail Station.

FLAG: BO1337{Imbi}

NET

SEMERAH PADI

PCAP file given, description said to find out what is downloaded. Open the file in Wireshark, follow then extract the only file downloaded through HTTP (conveniently named Flag).

2022-07-01 13:24:52	192.168.0.36	54578	192.168.0.37	31337	TCP	54578 → 31337 [ACK]
2022-07-01 13:24:52	192.168.0.36	54578	192.168.0.37	31337	HTTP 192.168.0.37:31337	GET /Flag HTTP/1.1
2022-07-01 13:24:52	192.168.0.37	31337	192.168.0.36	54578	TCP	31337 → 54578 [ACK]
2022-07-01 13:24:52	192.168.0.37	31337	192.168.0.36	54578	TCP	31337 → 54578 [PSH,
2022-07-01 13:24:52	192.168.0.37	31337	192.168.0.36	54578	TCP	31337 → 54578 [PSH,
2022-07-01 13:24:52	192.168.0.37	31337	192.168.0.36	54578	TCP	31337 → 54578 [ACK]
2022-07-01 13:24:52	192.168.0.36	54578	192.168.0.37	31337	TCP	54578 → 31337 [ACK]
2022-07-01 13:24:52	192.168.0.36	54578	192.168.0.37	31337	TCP	54578 → 31337 [ACK]
2022-07-01 13:24:52	192.168.0.36	54578	192.168.0.37	31337	TCP	54578 → 31337 [ACK]
2022-07-01 13:24:52	192.168.0.37	31337	192.168.0.36	54578	TCP	31337 → 54578 [PSH,
2022-07-01 13:24:52	192.168.0.36	54578	192.168.0.37	31337	TCP	54578 → 31337 [ACK]
2022-07-01 13:24:52	192.168.0.37	31337	192.168.0.36	54578	TCP	31337 → 54578 [PSH,
2022-07-01 13:24:52	192.168.0.36	54578	192.168.0.37	31337	TCP	54578 → 31337 [ACK]
2022-07-01 13:24:52	192.168.0.37	31337	192.168.0.36	54578	HTTP	HTTP/1.0 200 OK

Open the file and will see bunch of Lorem Ipsum, filler text. When counting all the characters in the text, something weird is seen:

2c	,	0.01%	|
2d	-	0%	
2e	.	1.95%	||
2f	/	0.01%	|

A "/", Lorem Ipsum got no "/", imposter found. CTRL+F:

```
porttitor hts/psei.o/WzUetp:/atbncm5NAv Lectus
```

Two Column Transposition Cipher. Oh, it means split the text from the middle into half, then combine them up together:

```
hts/psei.o/WzUetp:/atbncm5NAv

hts/psei.o/WzUe
tp:/atbncm5NAv

h t s / p s e i . o / W z U e
 t p : / a t b n c m 5 N A v

https://pastebin.com/5WNzAUve
```

And you thought this is over? No, now got 2 more file:

```
text 0.05 KB
1. https://ufile.io/bo0cubra
2. https://ufile.io/g6xqrbl2
```

First file is good music, second file is noise. I will be using one of these files as my new alarm and I won't tell you which one. Open the second file in Audacity, View Spectrogram, Spectrogram Settings, Max Frequency -> 16000 Hz.

FLAG: BO1337{2878f7b0f8deea26a66d642ebe045620efc43091}

STREAMLINE

Challenge Description hinted that a password bruteforce will be needed. *Insert potato PC meme here*. However, the challenge only provided a JPG file. Steghide extract on the JPG file without password, a handshake.cap is found.

Inside this handshake.cap contain 4 EAPOL keys needed to bruteforce the password.

2022-06-28 15:51:31	ae:0b:fb:d7:e5:a2	XiaomiCo_1c:53:05	EAPOL	Key (Message 1 of 4)
2022-06-28 15:51:39	ae:0b:fb:d7:e5:a2	XiaomiCo_1c:53:05	EAPOL	Key (Message 1 of 4)
2022-06-28 15:51:39	XiaomiCo_1c:53:05	ae:0b:fb:d7:e5:a2	EAPOL	Key (Message 2 of 4)
2022-06-28 15:51:39	ae:0b:fb:d7:e5:a2	XiaomiCo_1c:53:05	EAPOL	Key (Message 3 of 4)
2022-06-28 15:51:39	XiaomiCo_1c:53:05	ae:0b:fb:d7:e5:a2	EAPOL	Key (Message 4 of 4)

Convert the file using hcxpcapngtool and hcxhashtool into the format ready to be bruteforced. SSID given in the handshake.cap file.

```
hcxpcapngtool -o handshake.2200 handshake.cap
hcxhashtool -i handshake.2200 -o handshakeandeap.2200 --essid="Ridzuan Wifi"
```

Last step, run Hashcat to bruteforce the file and witness your computer turn into a nuclear reactor. After Hashcat hit the temperature limited, ask hint from admin to reduce the number of characters needed to bruteforce.

```
8bd0546009847e110a1783ca4a7d1b11:ae0bfbd7e5a2:a89ced1c5305:Ridzuan Wifi:oyen@9367

Session..........: hashcat
Status...........: Cracked
Hash.Mode........: 22000 (WPA-PBKDF2-PMKID+EAPOL)
Hash.Target......: handshakeandeap.2200
Time.Started.....: Wed Jul 20 11:56:09 2022 (2 secs)
Time.Estimated...: Wed Jul 20 11:56:11 2022 (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Mask.......: oyen?s?d?d?d?d [9]
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........:   107.6 kH/s (11.34ms) @ Accel:8 Loops:256 Thr:512 Vec:1
Speed.#3.........:     6317 H/s (6.60ms) @ Accel:2 Loops:32 Thr:128 Vec:1
Speed.#*.........:   113.9 kH/s
Recovered........: 1/1 (100.00%) Digests
Progress.........: 258048/330000 (78.20%)
Rejected.........: 0/258048 (0.00%)
Restore.Point....: 135168/330000 (40.96%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:0-1
Restore.Sub.#3...: Salt:0 Amplifier:0-1 Iteration:0-1
Candidate.Engine.: Device Generator
Candidates.#1....: oyen*3718 -> oyen"8884
Candidates.#3....: oyen-6439 -> oyen#0918
Hardware.Mon.#1..: Temp: 75c Util: 97% Core:1746MHz Mem:3504MHz Bus:16
Hardware.Mon.#3..: N/A

Started: Wed Jul 20 11:56:05 2022
Stopped: Wed Jul 20 11:56:13 2022
```

FLAG: BO1337{oyen@9367}

MISC

RAYQUAZA

Download the file, file the file

```
root@kali:/media/sf_VM_Shared/bo1337# file 1337Pikachu
1337Pikachu: Game Boy Advance ROM image: "POKEMON EMER" (BPEE01, Rev.00)
```

GameBoy! Download an emulator online, VisualBoyAdvance is used for this case. Load the GBA file, play the game, get the flag.

Challenge Description hinted that the flag can be obtained from one of the NPC.

FLAG: BO1337{91091b84a367c97a93eb7b5ba35e850e}

HEIHAWRU

Given a text file with a long lyric repeated twice, but each time encrypted with different cipher.

The first repetition of the lyric encrypted with ROT23, meanwhile the second part is encrypted with ROT3.

At the end of the text file, this string is given:

```
5:6:2 6:1:1 31:3:1 15:3:3 15:3:3 43:4:1 27:2:1 32:3:1 33:1:1 41:3:1 38:3:4 24:2:2 10:5:4
41:5:3 45:6:3 35:1:1 15:3:3 1:3:3 36:2:2 34:1:1 45:2:3 21:2:2 17:1:2 11:4:2
```

This is book cipher, for each block of text, the first number represent the line number, second number represent the nth word, and lastly the third number represent the nth alphabet in the word.

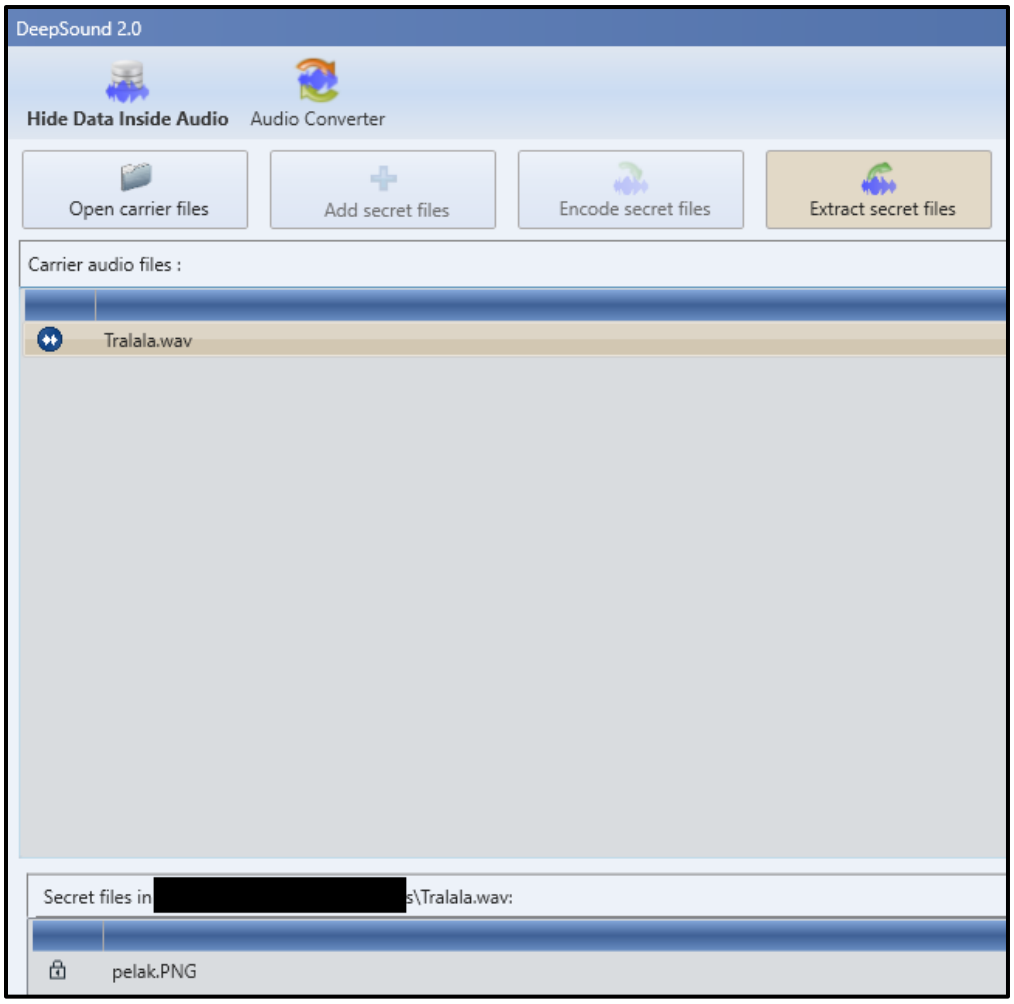
Solve the book cipher and get the flag.

FLAG: BO1337flAgisdarK3noKluai

GUNTHER

Another song was given, and this one is catchy. Never mind the previous song that I said I was going to set as alarm, this now is my new alarm.

Use DeepSound the extract the flag from the song. That's it.



Check the extracted picture, found the flag.

FLAG: BO1337{9d6382bf597a3014a8472762fedce888}

FLY

I didn't actually solve this in the CTF, but now that I know the solution, I am adding this in the write-up also. This is first of the two challenges that I didn't solve but included in this write-up.

Given the following string

```
623;0;724;0;124;0;126;0;635;0;618;0;125;0;257;0;288;0;876;0;163;0;047;0;023;{0;045;0;871;0;873;0;090;0;086;0;072;0;002;0;006;0;016;0;202;0;172;0;880;0;865;0;803;0;851;00;}
```

The code given above are IATA Flight code, to solve this, google the code and take the first letter of their respective 2-letter code.

COMPANY NAME	COUNTRY / TERRITORY	2-LETTER CODE	ACCOUNTING CODE (PAX)	AIRLINE PREFIX CODE
Bulgaria Air JSC	Bulgaria	FB	623	623
SWISS International Air Lines Ltd dba SWISS	Switzerland	LX	724	724

After decoding the whole thing, get FLAGISBOL33TFLY6ING2DUTCHMAN

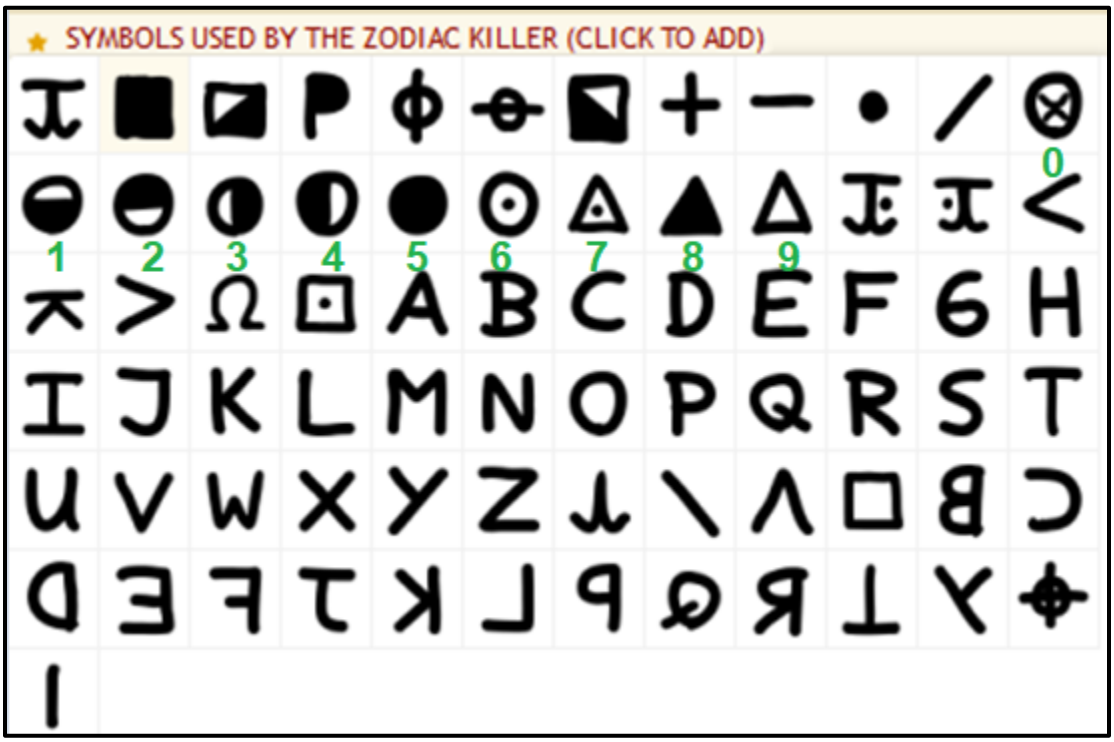
FLAG: BOL33TFLY6ING2DUTCHMAN

SHENG XIAO

Zodiac Cipher with a twist.

The encrypted text cannot be decrypted in the standard way. While the encrypted text is indeed in zodiac symbols, a bit of out-of-the-box thinking is needed for this one. This is hinted by the text in the picture, as the original zodiac message is not so easily readable.

To solve this, need to "watch" the keyboard on dcode.fr, zodiac killer cipher. Using "10337" as the benchmark, assign the rest of the numeric digits onto the symbols.



FLAG: BO1337{F4EC90216D2D7D5EDB7C201919FCE008E8}

DARCHROW

This is the second of the two challenges that I didn't solve in the CTF but included in this write-up. The title and description of the challenge hinted at Enigma cipher.

```
ACZQQOKLHUALHPTXGXKEPWTLDJUCEORHRKTQRTVKXOWRBYACB
UKRUPCEAQRAKMZRJHCEHZJWJKSGTLTMOXTEJHLPPEHXJBQQW
KXYQNKFTOKBJUVFRUFCSQMTXHTTDJSOFWHYNSOHVEZGQVURJF
JALJEXQWWYWIZBJLMOYDMGNXOMRMLQRSYWZHJGBLCSHNMYFXE
SJNFBBITRXHKZYQGYIPEUYTNFXSSPCXIZJMRCTLHUUHBFEXIVBM
EURYMPAZATXUVNVQRSLPFVWWBBUHOEMXYRPMLTYZXLHSAPMMM
QOEHXKOCDWBSWDMTFMSMFBNCQGMQHHJPQYKJPZNMYYDKZYZX
UHOOHAIAFGMDBMYAEQPRSUVQKGEZSA
```

However, there is no information on the cleartext, key used, rotor ring setting, reflector and plugboard. Which means that this cipher is not intended to be solved in the standard way.

To solve this, just throw the ciphertext into dcode.fr, Enigma Encoder and Decoder. Keep all the default setting, press encrypt/decrypt. The output will be:

```
ISTOPENQUIRESTOPFORYOUSTOPTORWADANDDECRPYTTHISSTOPFORALANTURINGSTOPFORHI
SSUCCESSSTOPOFDECRYPTINGSTOPALLTHEGERMANSTOPMESSAGESTOPTHEFLAGISBOBRACESE
NIGMAFORALANTURINGBRACESONLYTHOSEWHOAREASLEEPDONTMAKEMISTAKESGETNOCRITI
QUESEEMSEVERYBODYSWORRIEDBOUTTHINGSTHATWEARETHINKINGANDWHENTHEREMEDYS
THEENEMYYOUHIDESELFDEPRECATIONUPYOURSLEEVEANDSELFSERVINGFRIENDSWHOLEAVE
WHENYOUARESINKING
```

In the middle of the text there is
"FLAG IS BO BRACES ENIGMA FOR ALAN TURING BRACES"

FLAG: BO1337{ENIGMAFORALANTURING}

RANTAIANBLOK

Given an URL to a smart contract:

https://testnet.bscscan.com/address/0xc669100117c2e8b0492bd2f03a9a64b459776e62

An additional hint is given which is also needed to solve this challenge, the ABI of the smart contract.

```
"abi": [
  {
    "inputs": [],
    "stateMutability": "nonpayable",
    "type": "constructor"
  },
  {
    "inputs": [],
    "name": "flag",
    "outputs": [
      {
        "internalType": "string",
        "name": "",
        "type": "string"
      }
    ],
    "stateMutability": "view",
    "type": "function"
  }
],
```

With the address of a smart contract and the ABI, just code out a client to interact with the contract. I did mine is JavaScript.

```
<!DOCTYPE html>
<html>
<head>
</head>
<body>
<script src="https://cdn.jsdelivr.net/npm/web3@latest/dist/web3.min.js"></script>
<script>
const web3 = new Web3(new Web3.providers.HttpProvider('https://data-seed-prebsc-1-
s1.binance.org:8545'))

web3.eth.getBlockNumber(function (error, result) {
  console.log(result)
})


const abi = [
```

```
  {
    "inputs": [],
    "stateMutability": "nonpayable",
    "type": "constructor"
  },
  {
    "inputs": [],
    "name": "flag",
    "outputs": [
      {
        "internalType": "string",
        "name": "",
        "type": "string"
      }
    ],
    "stateMutability": "view",
    "type": "function"
  }
]

const contractaddr = "0xC669100117c2e8b0492bD2f03a9a64B459776e62"

const contract = new web3.eth.Contract(abi, contractaddr)
contract.methods.flag().call(function (err, res) {
 if (err) {
  console.log("An error occured", err)
  return
 }
 console.log("Flag: ", res)
})

</script>
</body>
</html>
```

Run the JavaScript and get flag.

```
21217048
Flag:  a82cbce07689283cfc897f4310b634d3e3f8e751
»
```

FLAG: BO1337{a82cbce07689283cfc897f4310b634d3e3f8e75}

REDPOINT

Bigbrain challenge.

Given:

FLAG: BO1337{screwdriver}

WEB

BREAK THE STORAGE

First web challenge, check source finds the JavaScript file, check the JavaScript file:

```
let database="data";var attempt=10;function validate(){var
t=document.getElementById("username").value,a=document.getElementById("password").value;$.
get(database,function(e){return
jsons=JSON.parse(e),normalUser=jsons.user.normal,t==normalUser.username&&a==normalUser.p
assword?(window.localStorage.setItem("userID",normalUser.userID),alert("Login
successfully"),!(window.location="profile.html")):(attempt--,alert("You have left "+attempt+"
attempt;"),0==attempt?(document.getElementById("username").disabled=!0,document.getElement
ById("password").disabled=!0,!(document.getElementById("submit").disabled=!0)):void 0)})}
```

From above, just submit the form once and there will be new data received by the browser containing JSON with our creds.


```
{"user":{"normal":{"userID":"1","username":"BattleOf1337","password":"BattleOf1337"},"super":{"
userID":"356a192b7913b04c54574d18c28d46e6395428ab","username":"root","password":""}}}
```

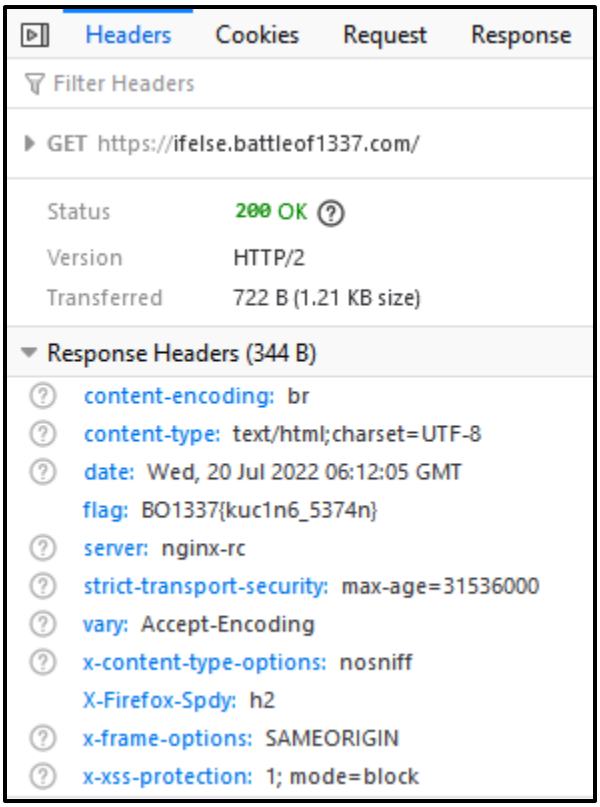

After login with the correct creds, a blank page is shown. Check source again for JavaScript file:

```
let
database="data";$.get(database,function(a){jsons=JSON.parse(a),getFlag=jsons.user.super,window.
localStorage.userID==getFlag.userID&&alert("BO1337{a2c13e70ff50376e259ddb5bd5e54a69b16e
569f}"),0==window.localStorage.length&&(window.location="login.html")});
```

FLAG: BO1337{a2c13e70ff50376e259ddb5bd5e54a69b16e569f}

CAT-DALMANTION

Access the website, click one of the buttons, check response header.



FLAG: BO1337{kuc1n6_5374n}

AND EKCELI

Website asking for password, check page source

```
<script>unlockPage=()=>{for(j=function(){for(h=document.getElementById("codePass").value,""=
<div id="unlockCode">
  <input placeholder="Enter Webpage Password" id="codePass">
  <input type="button" value="Unlock Page" onclick="unlockPage()" />
</div>
<!-- THE PASSWORD IS DOWN HERE -->

<!--
                    .=#%@@@@@%%%%%%%%%%%%%%%%%%%%@@@@@@@@@@@%%%##**+==-:.
                  :#@#=.            .:::::::::::::::::::------===+*#@@@@@%#+=:
                :#@*.          .----.   ..:::::::::..          ...        .:-=*@@#=.
               +@%.         :--: .---:::.           .:::::::::::::::::::::::--  .=#@%=
              #@+        ::: .---::-:::::::::::::+.             ..::::::::::          -#@+
             %@=       ::  --:.--:                +            ...          :--         =@#
           .%@=          := :=:                   -            -:            .+         @@
          :%@=               .  .-+*#%%%%%%%%+-                -                        *@=
        .*@%:                 =%@@@@@@@@@@.:-+#@#=                .-+*#####*+:            *@*:
     :#@%##+=-:.      =+-   #@%*#@@@@@@@@*+-.  =@@=          :=%@@@@@@%%@@@@@- --::::::: :*@%-
   .#@*+-.    :====--.  .: .===-:..:+-:-=+#@@#*@@-      =@@@@@@%#*+==--:.            -=..#@*
  =@%:+.   :#@%*++=+#@@#=-.      .-*@#.       :+*-          :@@=                 ::::..-=::+.#@:
 =@# -.   +@#:     *:  :=*#@@%%%@@#+.                        #@:        --. .+%@#**#@@+ +   +=@=
 *@  =   -@*      *@@*:                                     -@%=.      -#@@@@%. :    .. =   +=@=
 #%  =   *@. .-+%@@=*%@#+-                .=:-#%@%-            =*@%=            =@:      +   ++@-
 *@: :: -@*.=+++*@@.  .-*%@%*=:    .--:::.  @%.  ::::.           %@@@+.          %@@-  .=   +=@@
 -@%. +  *@:     +@%:      :#@@@@%#+=:       %@+ +@#***+        :%@*==.::--.    *@@@@  :----@@:
  -@@- =-::      :#@@@*=.   -@+  :-+*%@@@#*=-:==            -%++%@*.          .-*@@@#@@-     :@@:
   .*@#=..:        +@@##@@@@@@*.       .:+@@@@@%*+=-::.    :-=-       .-=*%@%@@-*@@@+     @@:
     :#@%-          :@@- :+@@@@@@#+=:    =@#  .:-=+*%@@@@@@@%%%%@@@@%@@@=. +@= @@@+    :@*
       -%@=           *@*  :@@+#@@@@@@%*=@@:          @@.    ..#@*.      #@=   *@#=@@@*    -@+
         #@*           -%@==@*   :=*%@@@@@@%#+=--:::%@=:::::*@#:::--=#@%+#@@@@@@@+    -@+
          *@*            =@@@.         :@@*#%@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@=    -@+
           *@#.            -#@*-      :@@    :-+*%@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@.    -@+
            -@@=             :+@@*-  .@@:          #@=:-==++%@@%%%%@@@@#*#@@=#@#*@*     -@+
              +@@=              :*@@#@@=           #@.       +@+     %@: .@@: #@*@*      -@+
                =%@+:.:--  ::-:   .=*%@%#+=-:.    %@.       @@.    #@+   #@#=*#@#-       :@*
                  -#@%+: :--: .---:     .:=+*#%@@@@@%#****%@@**#@@@@@@%#*+=:          :@#
                     -*@@*=. :---. :---:          ....:::::::::....           .:        .@#
                        :+#@%*-. .::::::..:--::.            .                -=-    :    @@
                            -+#@@#+:   ::::::==-:::::::::::::::::::::::::-:     :==    %@
                                .-*%@%+-.      .::::::::::::::....           .---.       %@.
                                     -+%@%*=.                  ...........           #@*
                                         :+#@@#+=-:.                              -@@=
                                             .-+*#%@@@@*+=:.                    .-+@@+.
                                                     .:-+*#%@@%%##***+++++++**#@@#+:
-->
```

Insert neuron activation meme. Password is “DOWN HERE”. Input password and unlock page, a new JavaScript will be loaded.

The secret textbox has a onchange listener and will spook you if you type anything into it. To prevent heart attack to the current reader, I won't show the jump scare here.

Investigate the new JavaScript, there is a huge chunk of obfuscated JavaScript in the middle:

```
function _0x5673(_0x31110e,_0x1ff010){var _0x572bf1=_0xd79b();return
_0x5673=function(_0x378094,_0x36c51f){_0x378094=_0x378094-0x16b;var
_0x385838=_0x572bf1[_0x378094];return _0x385838;},_0x5673(_0x31110e,_0x1ff010);}var
_0x13ea90=_0x5673;(function(_0x49595a,_0x44fd92){var
_0x298734=_0x5673,_0x859d6=_0x49595a();while(!![]){try{var _0x4f4415=-
parseInt(_0x298734(0x176))/0x1+-parseInt(_0x298734(0x16c))/0x2*(-
parseInt(_0x298734(0x16f))/0x3)+-parseInt(_0x298734(0x170))/0x4+-
parseInt(_0x298734(0x175))/0x5+-parseInt(_0x298734(0x181))/0x6*(-
parseInt(_0x298734(0x172))/0x7)+-parseInt(_0x298734(0x180))/0x8+-
parseInt(_0x298734(0x173))/0x9*(-
parseInt(_0x298734(0x17f))/0xa);if(_0x4f4415===_0x44fd92)break;else
_0x859d6['push'](_0x859d6['shift']());}catch(_0x2048f1){_0x859d6['push'](_0x859d6['shift']());}}}
(_0xd79b,0xa0988),eval(function(_0x5f41de,_0x312d2d,_0x35d16b,_0x1e1fb3,_0x3aed22,_0x3791
32){var _0x3584c4=_0x5673,_0x30157b=(function(){var _0x3a3eaf=!![];return
function(_0x17a20a,_0x30e01a){var _0x2e8fe6=_0x3a3eaf?function(){var
_0xc66494=_0x5673;if(_0x30e01a){var
_0x21069b=_0x30e01a[_0xc66494(0x171)](_0x17a20a,arguments);return
_0x30e01a=null,_0x21069b;}}:function(){};return
_0x3a3eaf=![],_0x2e8fe6;};}()),_0x177552=_0x30157b(this,function(){var
_0x279b56=_0x5673;return
_0x177552[_0x279b56(0x17c)]()['search'](_0x279b56(0x186))[_0x279b56(0x17c)]()[_0x279b56(
0x16b)](_0x177552)[_0x279b56(0x185)]('(((.+)+)+)+$');});_0x177552();var
_0x4f1531=(function(){var _0x497357=!![];return function(_0x138d50,_0x58d359){var
_0x2731ce=_0x497357?function(){var _0x1a4fd9=_0x5673;if(_0x58d359){var
```

```
_0x2bec4f=_0x58d359[_0x1a4fd9(0x171)](_0x138d50,arguments);return
_0x58d359=null,_0x2bec4f;}}:function(){};return
_0x497357=![],_0x2731ce;};}()),_0x40b1b0=_0x4f1531(this,function(){var
_0x505b1b=_0x5673,_0x2cec88=function(){var
_0x1b1852=_0x5673,_0x147740;try{_0x147740=Function(_0x1b1852(0x16d)+_0x1b1852(0x17e)
+');')();}catch(_0x55c1ba){_0x147740=window;}return
_0x147740;},_0x1cde78=_0x2cec88(),_0x1ecbcb=_0x1cde78['console']=_0x1cde78[_0x505b1b(0x1
74)]||{},_0x2735e7=['log',_0x505b1b(0x188),_0x505b1b(0x183),_0x505b1b(0x187),'exception',_0x
505b1b(0x177),'trace'];for(var
_0x49ea25=0x0;_0x49ea25<_0x2735e7[_0x505b1b(0x17d)];_0x49ea25++){var
_0x370105=_0x4f1531[_0x505b1b(0x16b)]['prototype'][_0x505b1b(0x16e)](_0x4f1531),_0x36578
4=_0x2735e7[_0x49ea25],_0x5e6883=_0x1ecbcb[_0x365784]||_0x370105;_0x370105[_0x505b1b(
0x17b)]=_0x4f1531[_0x505b1b(0x16e)](_0x4f1531),_0x370105[_0x505b1b(0x17c)]=_0x5e6883[_
0x505b1b(0x17c)][_0x505b1b(0x16e)](_0x5e6883),_0x1ecbcb[_0x365784]=_0x370105;}});_0x40b
1b0();if(_0x3aed22=function(_0x580fba){var
_0x175c6a=_0x5673;return(_0x580fba<0x3e?'':_0x3aed22(parseInt(_0x580fba/0x3e)))+((_0x580f
ba%=0x3e)>0x23?String[_0x175c6a(0x17a)](_0x580fba+0x1d):_0x580fba[_0x175c6a(0x17c)](0x2
4));},!''[_0x3584c4(0x184)](/^/,String)){for(;_0x35d16b--
;)_0x379132[_0x3aed22(_0x35d16b)]=_0x1e1fb3[_0x35d16b]||_0x3aed22(_0x35d16b);_0x1e1fb3=
[function(_0x2d2680){return _0x379132[_0x2d2680];}],_0x3aed22=function(){var
_0x2b1632=_0x3584c4;return _0x2b1632(0x182);},_0x35d16b=0x1;}for(;_0x35d16b--
;)_0x1e1fb3[_0x35d16b]&&(_0x5f41de=_0x5f41de[_0x3584c4(0x184)](new
RegExp('\x5cb'+_0x3aed22(_0x35d16b)+'\x5cb','g'),_0x1e1fb3[_0x35d16b]));return
_0x5f41de;}(_0x13ea90(0x178),0x0,0x45,_0x13ea90(0x179)[_0x13ea90(0x189)]('|'),0x0,{})));func
tion _0xd79b(){var
_0x34c987=['2025pEzLqD','console','3165485SGgtaf','653966ouPOVC','table','o\x208=2.3(\x22p\x
22);9\x20q(){r\x20b=2.d.s.e,c=2.d.t.e,a=\x22u\x20v\x20w\x20x\x20y\x20z\x20A\x20B\x20C.\x
22;D\x20E==b||\x22\x22==b?(f(\x22I\x20g\x20a\x20F,\x20I\x20G\x20H\x20a\x20J.\x22),!1):c
==a[12]+a[10]+a[16]+a[K]+a[L]+a[7]+a[h]+a[M]+a[6]+a[i]+a[14]+a[h]+a[j]+a[N]+a[16]+a[k]+a[j]+
a[14]+a[k]+a[i]+a[10]+a[O]?(f(\x22P\x20Q..\x20R\x20I\x20g\x20S..\x22),2.3(\x22l\x22).4.5=\x
22m\x22,2.3(\x22T\x22).4.U=\x22V(W.X)\x22,!1):Y\x200}9\x20n(){2.3(\x22l\x22).4.5=\x22m\x
22,2.3(\x22Z\x22).4.5=\x2211\x22}8.13(\x2215\x22,n);','||document|getElementById|style|visibi
lity|||_0x872f99|function||||_0xaf458f|value|alert|am|36|26|29|41|_0x387654|hidden|_0x069420|c
onst|_0x1a990e|_0x001337|var|_0x143209|_0x4f8765|The|quick|brown|fox|jumps|over|the|lazy|
dog|return|null|person|should|have||name|21|24|31|28|37|Thank|you|now|free|_0xa4557d|backg
roundImage|url|74eee63dc70adf02e115e178e360f8c875471ded66f9e64d91ad41212d50433b|jpe
g|void|_0x4f7a88||visible||addEventListener||input|','fromCharCode','__proto__','toString','length','{
}.constructor(\x22return\x20this\x22)(\x20)','126590ZGJDdU','5524432XauVwI','362892MvjUJn',
'\x5cw+','info','replace','search','(((.+)+)+)+$','error','warn','split','constructor','82PIeVLm','return\
x20(function()\x20','bind','13830ZjTFcf','4752536atkjJT','apply','91nskkQw'];_0xd79b=function(){
return _0x34c987;};return _0xd79b();}
```

Throw the whole thing into de4js and let it deobfuscate.

```
const _0x872f99 = document.getElementById("_0x1a990e");

function _0x001337() {
    var b = document._0xaf458f._0x143209.value,
        c = document._0xaf458f._0x4f8765.value,
        a = "The quick brown fox jumps over the lazy dog.";
    return null == b || "" == b ? (alert("I am a person, I should have a name."), !1) : c == a[12] + a[10] + a[16] + a[21] +
a[24] + a[7] + a[36] + a[31] + a[6] + a[26] + a[14] + a[36] + a[29] + a[28] + a[16] + a[41] + a[29] + a[14] + a[41] + a[26] +
a[10] + a[37] ? (alert("Thank you.. now I am free.."), document.getElementById("_0x387654").style.visibility = "hidden", docu
ment.getElementById("_0xa4557d").style.backgroundImage = "url(74eee63dc70adf02e115e178e360f8c875471ded66f9e64d91ad41212d50433
b.jpeg)", !1) : void 0
}

function _0x069420() {
    document.getElementById("_0x387654").style.visibility = "hidden", document.getElementById("_0x4f7a88").style.visibility =
"visible"
}
_0x872f99.addEventListener("input", _0x069420);
```

From the deobfuscated JavaScript above, notice “c == a[12] + a[10] + a[16] + a[21] + a[24] + a[7] + a[36] + a[31] + a[6] + a[26] + a[14] + a[36] + a[29] + a[28] + a[16] + a[41] + a[29] + a[14] + a[41] + a[26] + a[10] + a[37]” and a = "The quick brown fox jumps over the lazy dog.".

Solve the c, get the flag.

```
main.js                                                    Run        Output

var a = "The quick brown fox jumps over the lazy dog."                node /tmp/tWqK24vie5.js
                                                                      obfuscationarefornoobz
var c = a[12] + a[10] + a[16] + a[21] + a[24] + a[7] + a[36] + a[31] + a[6] + a[26] + a[14] + a[36] +
    a[29] + a[28] + a[16] + a[41] + a[29] + a[14] + a[41] + a[26] + a[10] + a[37]

console.log(c)
```

FLAG: BO1337{obfuscationarefornoobz}

ALICEINWONDERLAND

The rabbit holes. Multiple hints are given by the organizer, in the end pointing to directory bruteforcing, and final URL will be a long one.

Checking robots.txt

```
User-agent: flag-grabber
Disallow: /flag-generator.php
Allow: /

User-agent: *
Allow: /
Disallow: /00
Disallow: /01
Disallow: /02
Disallow: /03
Disallow: /04
Disallow: /05
Disallow: /06
Disallow: /07
Disallow: /08
Disallow: /09
Disallow: /0a
Disallow: /0b
Disallow: /0c
Disallow: /0d
Disallow: /0e
Disallow: /0f
Disallow: /10
Disallow: /11
Disallow: /12
Disallow: /13
```

flag-generator.php is just there to mess with people. The 2 characters directories shown above goes from 00 to FF, representing hexadecimals.

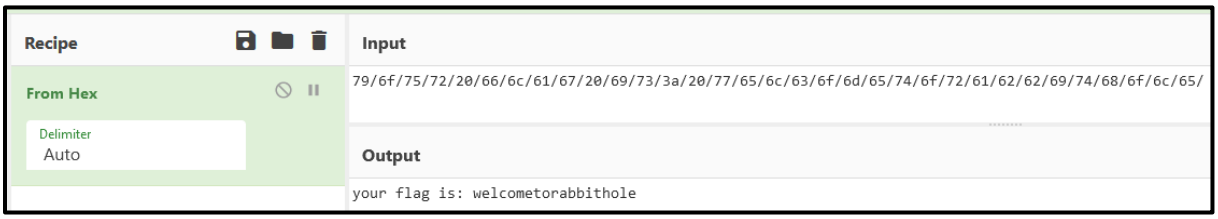
Run wfuzz on all those directories will give us 301 and pepe. But when run wfuzz to fuzz twice on directories, interesting output can be seen.

```
root@kali:/media/sf_VM_Shared/bo1337# wfuzz -c -w word -w word --hc 404 http://rabbit.battleof1337.com:1337/FUZZ/FUZ2Z
 /usr/lib/python3/dist-packages/wfuzz/__init__.py:34: UserWarning:Pycurl is not compiled against Openssl. Wfuzz might n
********************************************************
* Wfuzz 3.1.0 - The Web Fuzzer                         *
********************************************************

Target: http://rabbit.battleof1337.com:1337/FUZZ/FUZ2Z
Total requests: 65536

=====================================================================
ID           Response   Lines    Word       Chars       Payload
=====================================================================

000030579:   301        9 L      28 W       341 Ch      "77 - 72"
000031088:   301        9 L      28 W       341 Ch      "79 - 6f"

Total time: 130.5611
Processed Requests: 65536
Filtered Requests: 65534
Requests/sec.: 501.9563
```

Following the lead, bruteforce the rest of the URL until reaching the end.

```
root@kali:/media/sf_VM_Shared/bo1337# wfuzz -c -w word --hc 404 http://rabbit.battleof1337.com:1337/79/6f/75/72/20/66/6c/61/67/20/69/73/3a/20/77/65/6
 /usr/lib/python3/dist-packages/wfuzz/__init__.py:34: UserWarning:Pycurl is not compiled against Openssl. Wfuzz might not work correctly when fuzzing
********************************************************
* Wfuzz 3.1.0 - The Web Fuzzer                         *
********************************************************

Target: http://rabbit.battleof1337.com:1337/79/6f/75/72/20/66/6c/61/67/20/69/73/3a/20/77/65/6c/63/6f/6d/65/74/6f/72/61/62/62/69/74/68/6f/6c/65/FUZZ
Total requests: 256

=====================================================================
ID           Response   Lines    Word       Chars       Payload
=====================================================================


Total time: 0
Processed Requests: 256
Filtered Requests: 256
Requests/sec.: 0

root@kali:/media/sf_VM_Shared/bo1337#
```

Convert the hexadecimal part of the URL into plaintext and get flag.



FLAG: BO1337{welcometorabbithole}