Sri Lanka Institute of Information Technology

Final Project Report

ISP Project Report

Information Security Project 2022

Project ID: 19

Submitted by:

IT Number	Name
IT19154640	Weerasiri H.A.K.D.
IT19990200	K A P A K P Kathriarachchi

Date of submission
06th June 2022

Abstract

The world around us is rapidly digitalizing, as well as the internet is nearly ubiquitous, making cyber security an unavoidable aspect of our daily life. Capture-the-flag games have had a good influence on Individuals' levels of motivation and involvement Capture-the-flag games were found to lead to vastly higher learning outcomes and a greater awareness of cybersecurity in several research.
Additional benefits included improved practical understanding in cyber security, higher grades, and more assurance in cyber security abilities.

Organizing such games was discovered to be a difficult task, and as a result, knowledge is required from both organizers and participants in capture-the-flag games. Capture-the-flag game settings are complicated, and support personnel are required to organize such games. It was discovered that designing the tasks to be properly hard was a difficult process, and a related challenge was difficulty avoidance. This document describes the Introduction, Methodology, and Evaluation of our CTF box "TheBlackList."

Acknowledgement

The authors are grateful to Dr. Lakmal Rupasinghe, our lecturer in charge of the Information Security Project module, who has offered advice and inspiration since the start of this study. I would also want to thank Ms. Laneesha Ruggahakotuwa, Ms. Chathu Udagedara, and Ms. Menaka Moonamaldeniya for their assistance in understanding the topic and ensuring the success of this event. Finally, I'd want to express my heartfelt appreciation to everyone who contributed knowledge and opportunities to assist us finish this job on schedule.

Declaration

We certify that even this project document, or any section of it, is not really a copy of a based on the discovery by any institution, college, or other organization, or a prior student project group at SLIIT, and that it was not taken from the Internet or other sources.

Project Details

Project Title	The Black List
Project ID	ISP-22-REG-19

Group Members

Reg. No	Name	Signature
IT19154640	Weerasiri H.A.K.D.	
IT19990200	K A P A K P Kathriarachchi	

Table of Contents

1. Introduction

1.1 Problem Statement

Capture-the-flag (CTF) games are challenges in which the aim is to uncover concealed flags in a specific computer space. The ecosystem might be as little as a single website or as large as a whole network of computers. Jeopardy and attack-defense are the two most prevalent forms of CTF games. A Jeopardy-style tournament often includes numerous categories with various tasks to complete. In an attack-defense competition, each team is assigned a network or a host computer to protect while attempting to exploit the other teams.

The primary goal of this research is to investigate how CTF games help education while also investigating potential downsides and obstacles associated with the format. The scope of this study is not limited to CTF games. Other kinds of gamified and offensive cyber security education being investigated as needed. As a result, the main study question is "what are the benefits and drawbacks of adopting capture-the-flag games in cyber security education?" A supplementary research question is "is there any quantitative proof of discovered benefits and drawbacks?" CTF activities are a fun and gratifying method to learn different areas of cyber security, and the world around us is rapidly digitalizing, with the internet nearly everywhere, making cyber security an unavoidable, albeit often seemingly inconsequential, part of our lives. The advantages of employing CTF games in cyber security education include increased student motivation and improved practical understanding because of the method. The format's challenges included significant educational requirements both from organizers and the performers, as well as complicated technological needs.

1.2 Product Scope

Challenge type is **WEB**

This sort of challenge focuses on locating and exploiting software vulnerabilities. This might include assessing the participants' understanding of SQL Injection, XSS (Cross-Site Scripting), CSRF, and other topics.

Audience

- This project directly affects Linux users, and it may be able to impact their existing Linux OS settings or configuration.

- This would be a great aid in fixing flaws and vulnerabilities that have already happened in the Linux environment in order to carry out work for Linux Developers / Backend Developers.
- It is recommended that research students study all sections of this manual to gain a general understanding of the workflow and technical aspects of the program.
- This is beneficial to testers and may be used as documentation to learn about the interfaces.
- Threat Intelligence would be able to obtain information in a variety of methods after exhausting all possibilities, which is also a great approach to inspire them when they are involved in such a scenario.

Scenario

The world's most wanted fugitive was taken to the custody by the FBI. They found a laptop that was used by the fugitive to monitor his tasks. This laptop contains a list of deadly criminals which is called “The Blacklist” and the FBI needs access to that list. Attacker Used to put info about other ranked hackers’ details into his created databased, which stored on web page, with Tight secure involved it. For that FBI chooses professionals with security skills. To get access to The Blacklist they must use security and hacking techniques.

Specific Target System – web applications and databases

Technologies use – HTTP,PHP,SQL,CSS,Javascript,html,apache,Kali linux O/S

Main expected outcomes of the project:

Our CTF challenge is perfect for experts in threat intelligence and security operations centers. They may efficiently hone their security expertise and Sell it to an organization, Host in TryHackMe.

1.3 Project Report Structure

The following chapters will be covered in this project report. Requirements and analysis, design, implementation, and testing are all part of the technique. The review and conclusion contain an evaluation of the results, lessons were learned, and next steps.

2. Methodology

2.1 Requirements and Analysis

The following chapters will be covered in this project report. Requirements and analysis, design, implementation, and test are all part of the technique. The review and conclusion contain an evaluation of the results, lessons were learned, and next steps.

Various difficulties, A variety of challenge categories, Increasing the level of difficulty, Adequate instruction for novices, such as offering reading material Information on the comparison of real-world attack vectors and CTF challenges.

2.2 Design

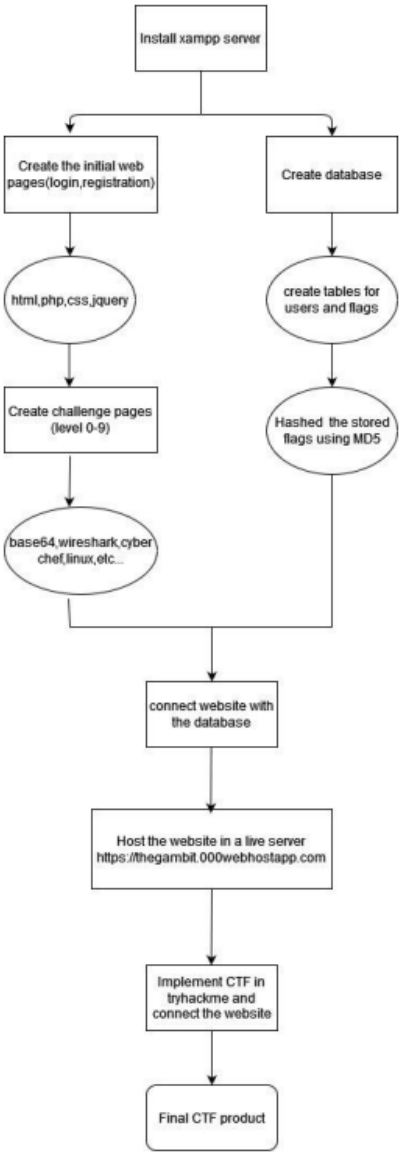
Based on findings, a prototype CTF was created to introduce people to critical cyber security principles. We used a prioritizing strategy to prioritize the needs pertaining to content because there are several CTF challenge types. Competitions were chosen based on the shortest possible design time.

This analysis resulted in the design of:

Web difficulties using OWASP web top ten examples Cryptography problems with samples of various encryption and encoding systems Forensics difficulties, including file forensics and steganography instances, Various tasks that put broad abilities to the test, such as Linux command line expertise, programming knowledge, and the ability to utilize a number of tools, OSINT obstacles that foster creativity and renaissance.

Although Pwn and Reverse Engineering tasks were also possible, they would take a long time to construct due to the substantial knowledge and proficiency required in low level programming languages and assembly code. Although Pwn and Reverse Engineering tasks were also possible, they would take a long time to construct due to the substantial knowledge and expertise required in low level programming languages and assembly code.

Here is the overview of the design of CTF box.



2.3 Implementation

Tools and techniques used to create web interface

- Xampp - XAMPP is a free and open-source cross-platform web server solution stack package developed by Apache Friends, consisting mainly of the Apache HTTP Server, MariaDB database, and interpreters for scripts written in the PHP and Perl programming languages.
- Html
- Css
- Jquery

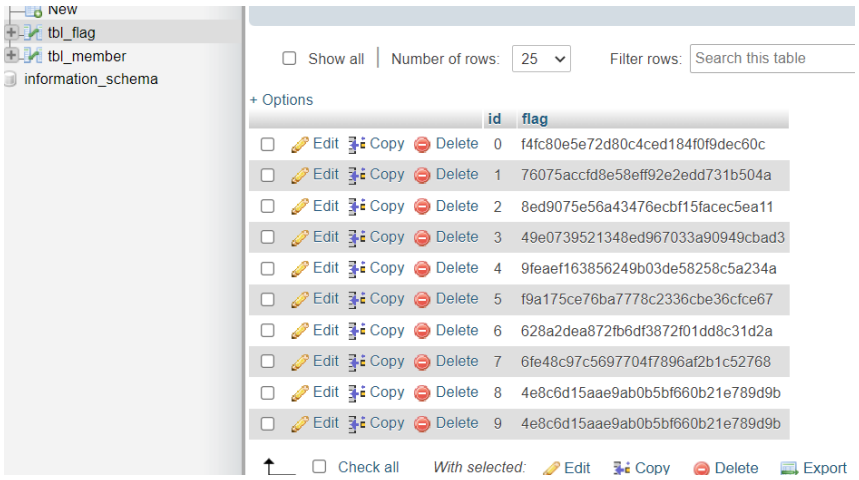
Tools and techniques used to create CTF challenges

- Level 0 - Base 64 decoder
- Level 1 – Fernet decoder,cyberchef,malborge interpreter
- Level 2 – dirbuster & gobuster
- Level 3 - Exif tool and steghide(in linux environment)
- Level 4 - Wireshark packet analysis
- Level 5 - This flag based on cryptography with cease cipher
- Level 6 - Sonic visualizer
- Level 7 - zip tools and Linux commands (chmod +x,ltrace,strings,strcmp)

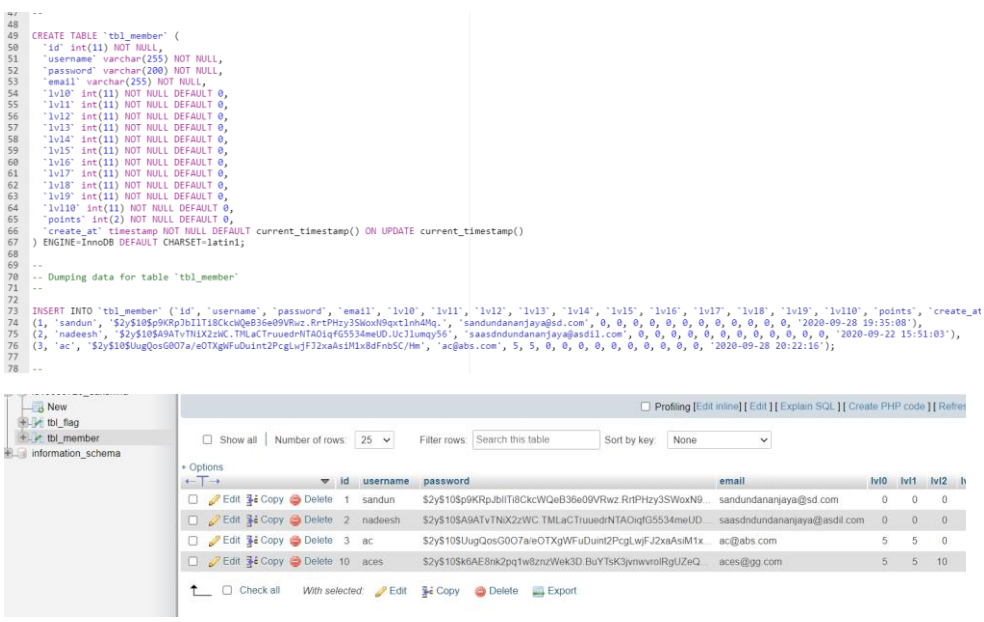
Implementation of DBMS

- ‘tbl_flag’ creating

```sql
--
-- Table structure for table `tbl_flag`
--

CREATE TABLE `tbl_flag` (
  `id` int(2) NOT NULL,
  `flag` varchar(255) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--
-- Dumping data for table `tbl_flag`
--

INSERT INTO `tbl_flag` (`id`, `flag`) VALUES
(0, 'f4fc80e5e72d80c4ced184f0f9dec60c'),
(1, 'c98b8b5385c34b66da50d038de45eb46');
```



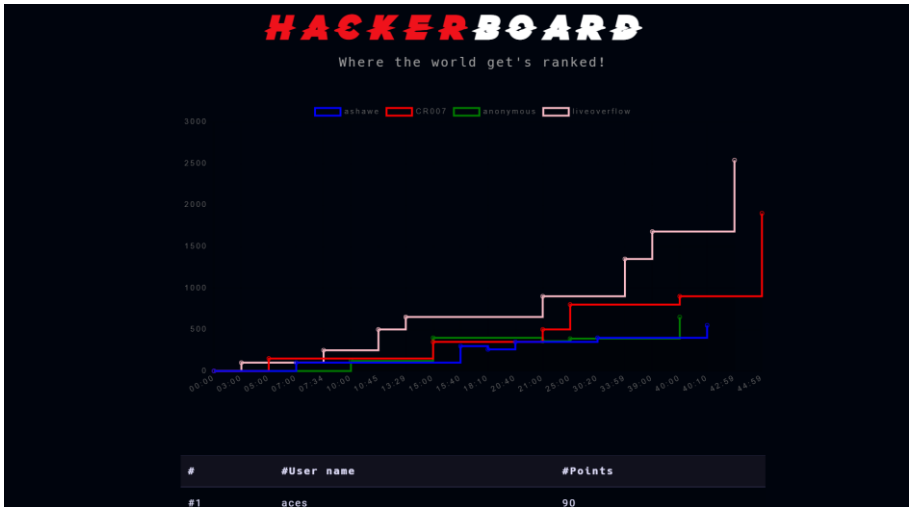
- "tbl_member" creating



```sql
--

CREATE TABLE `tbl_member` (
  `id` int(11) NOT NULL,
  `username` varchar(255) NOT NULL,
  `password` varchar(200) NOT NULL,
  `email` varchar(255) NOT NULL,
  `lvl0` int(11) NOT NULL DEFAULT 0,
  `lvl1` int(11) NOT NULL DEFAULT 0,
  `lvl2` int(11) NOT NULL DEFAULT 0,
  `lvl3` int(11) NOT NULL DEFAULT 0,
  `lvl4` int(11) NOT NULL DEFAULT 0,
  `lvl5` int(11) NOT NULL DEFAULT 0,
  `lvl6` int(11) NOT NULL DEFAULT 0,
  `lvl7` int(11) NOT NULL DEFAULT 0,
  `lvl8` int(11) NOT NULL DEFAULT 0,
  `lvl9` int(11) NOT NULL DEFAULT 0,
  `lvl10` int(11) NOT NULL DEFAULT 0,
  `points` int(2) NOT NULL DEFAULT 0,
  `create_at` timestamp NOT NULL DEFAULT current_timestamp() ON UPDATE current_timestamp()
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Dumping data for table `tbl_member`
--

INSERT INTO `tbl_member` (`id`, `username`, `password`, `email`, `lvl0`, `lvl1`, `lvl2`, `lvl3`, `lvl4`, `lvl5`, `lvl6`, `lvl7`, `lvl8`, `lvl9`, `lvl10`, `points`, `create_at`
(1, 'sandun', '$2y$10$p9KRpJbIlTi8CkcWQeB36e09VRwz.RrtPHzy3SWoxN9qxtlnh4Mq.', 'sandundananjaya@sd.com', 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, '2020-09-28 19:35:08'),
(2, 'nadeesh', '$2y$10$A9ATvTNiX2zWC.TMLaCTruuedrNTAOiqfG5534meUD.UcJlumqy56', 'saasdndundananjaya@asdil.com', 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, '2020-09-22 15:51:03'),
(3, 'ac', '$2y$10$UugQosG0O7a/eOTXgWFuDuint2PcgLwjFJ2xaAsiM1x8dFnb5C/Hm', 'ac@abs.com', 5, 5, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, '2020-09-28 20:22:16');

--
```

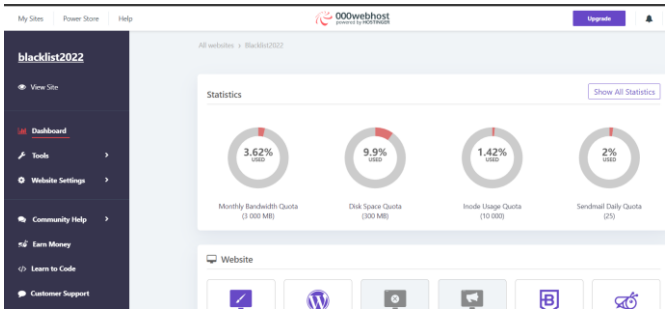
Scoreboard Implementation

```php
<?php
const HOST = 'localhost';

const USERNAME = 'id19050723_dakshina_uname';

const PASSWORD = 'W<PvGR4P=nw9^I{g';

const DATABASENAME = 'id19050723_dakshina';

// Create connection
$conn = new mysqli(HOST, USERNAME, PASSWORD, DATABASENAME);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
?>

<!DOCTYPE html>
<html lang="en">

<head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
    <title>TheGambit</title>
```



Host website in a live server

We import web pages and the databases to a free domain hosting server



Tryhackme Implementation

tryhackme.com/jr/blacklist2022

2.4 Testing

Testing is the most important component of the project. The Try hack me implementation required to be tested unit by unit. We performed a test to see if the jobs were compatible with the website and could be properly executed once each one was done while they were being produced stepwise. We had ten tasks in our project, and we examined each one to determine if it could be played by a user by looking at the features that the developers had included. Whenever it comes to website creation, we created pages to assist with the process. Following completion, the developers were required to test each page individually. If the current page has a link to a different page.

During the testing implementation, a few small flaws were detected and swiftly resolved. Following that, developers must inspect the application's backend to ensure that the data entered by the users is valid. Specific users of the web application were brought in for integrated testing.

The following stage is to ensure that all jobs contribute to the different online tools and platforms. During the testing implementation, a few small difficulties were detected and swiftly resolved. Following that, developers must inspect the application's backend to ensure that the data entered by the users is valid. A system check is conducted after unit and integrated testing. Following that,

there should be a discussion with all of the project's developers and others, and then acceptance testing should commence. The project was completed after those test runs. Both the "Try Hack Me" implementation and the website were verified to be bug-free following the tests.

3. Evaluation

3.1 Assessment of the Project results

Walkthroughs were conducted with three people, each lasting roughly 30 minutes, to assess the tool's usability. The participants were final-year computing students, two of whom had never participated in a CTF before.

The assessment comments were mostly favorable, showing that the tool has the ability to pique students' interest in cybersecurity and lead them while developing their practical abilities. Furthermore, the study found that the CTF experience (use) discouraged certain students from pursuing a profession in cyber security or ethical hacking in particular.

3.2 Lessons Learned

We learned following things by creating this CTF.

- Basic enumeration of HTTP.
- Steganography using steghide.
- Basics of SSH.
- Abuse sudo to get root privilege escalation.

3.3 Future Work

Customized CTF challenges focusing on the learning perspective and providing considerable educational context will be produced in the future. In this approach, certain components may be updated or developed to provide enhanced gamification features, quizzes, and evaluation methods.

One crucial element would be to integrate storytelling components to discover and evaluate the potential of using CTF systems and customizable CTF challenges for educational purpose, not only in information security but also in related topics like user privacy and privacy-aware data governance, and to capitalize on the outcomes of based tasks.

4. Conclusion

The main purpose of this study was to learn to think creatively on how a system may be attacked and to look at how CTF games could improve education. We conducted a comparison study for various aspects of cyber security, highlighting the specific operating systems, technologies, and tools for each level of our CTF box, and we were able to draw conclusions about the advantages and disadvantages of chosen innovations. Considering each technology, we chose the most appropriate technologies and tools for our CTF based on the purpose and audience that we chose. Additional elements that may enhance the platforms were also suggested.

5. References

2022, C. M. (n.d.). *Learning Resources*. Retrieved from picoCTF: https://picoctf.org/resources

aurelius. (2018, 04 22). *Tools and resources to prepare for a hacker CTF competition or challenge*. Retrieved from INFOSEC: https://resources.infosecinstitute.com/topic/tools-of-trade-and-resources-to-prepare-in-a-hacker-ctf-competition-or-challenge/

Capture The Flag 101. (n.d.). Retrieved from ctf101.org: https://ctf101.org/

CTFs. (n.d.). *Introduction*. Retrieved from CTF Resources: https://ctfs.github.io/resources/index.html

Lightfoot, J. (n.d.). *So, You Want to CTF? (A Beginner's Guide to CTFing)*. Retrieved from J Lightfoot: https://jaimelightfoot.com/blog/so-you-want-to-ctf-a-beginners-guide/

Appendix A: Test Results

Walkthrough for setup

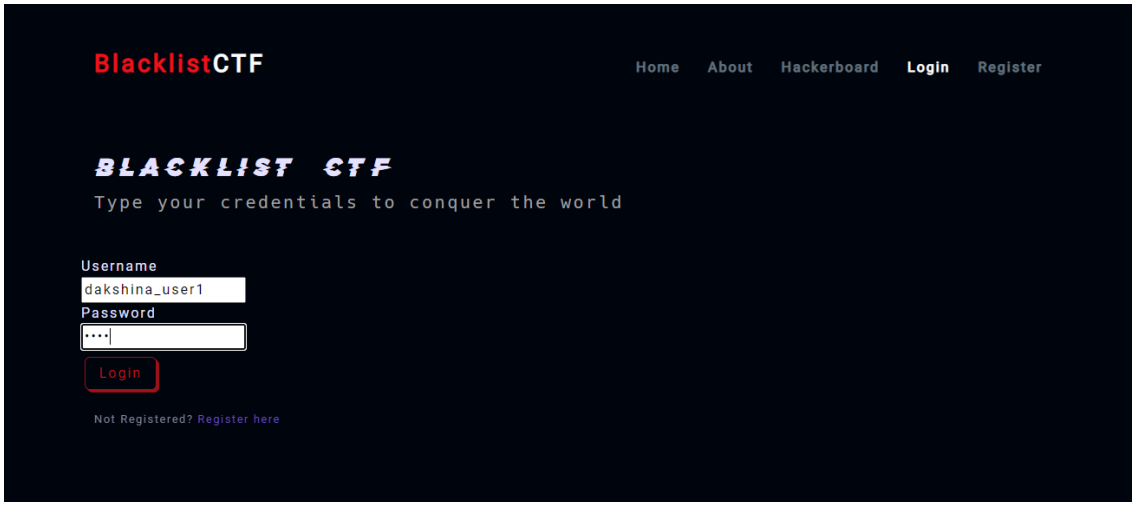
1. Link to the main website(CTF):

   When the user is new to the CTF, that user must be registered to the site,

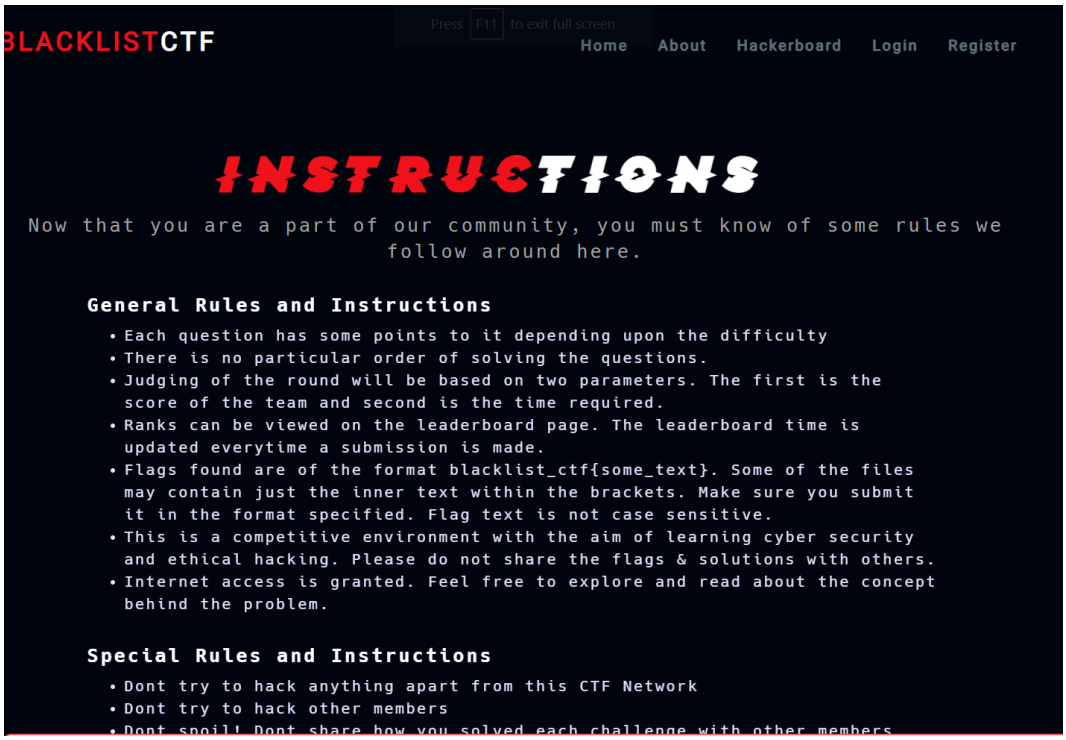
   After the registration, user can grab the way to CTF and all of instruction are given in there.

Interface

Register as new user

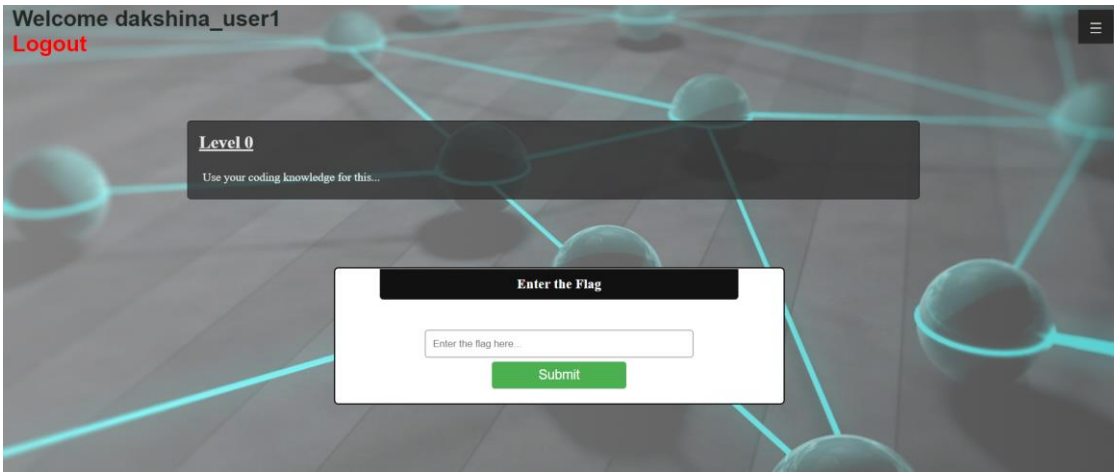


Register user must logging from here.



Instruction will be given to further go.

Walkthrough of Level

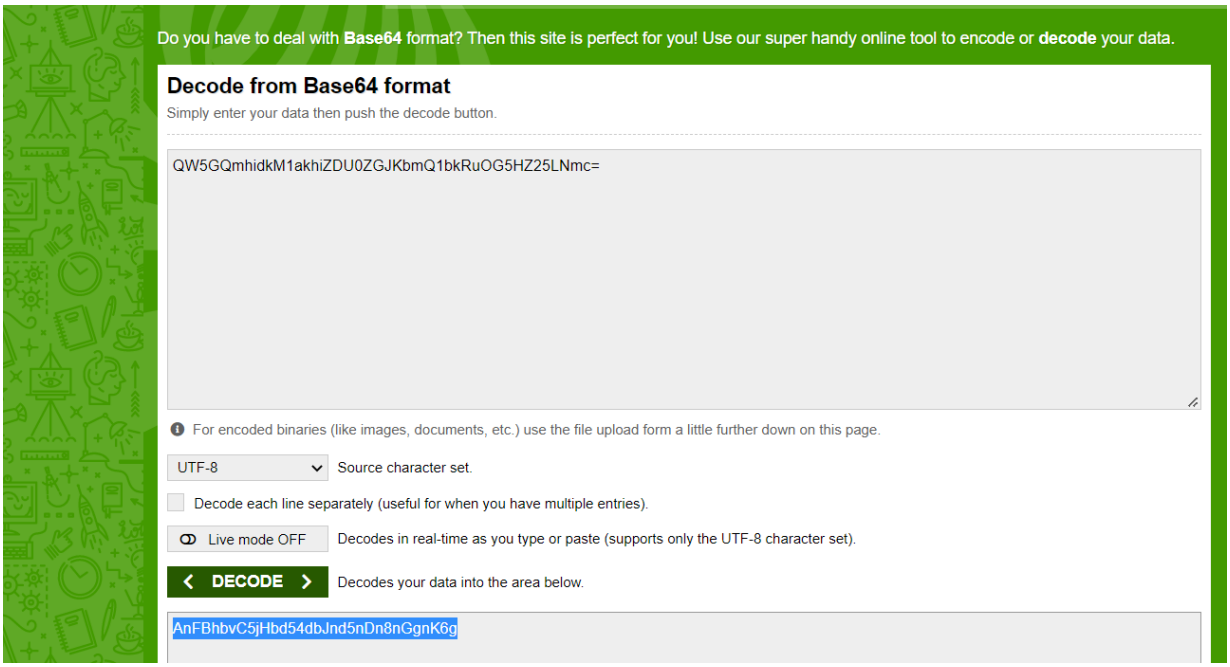


Level 0

Can be use small code knowledge, user has to view the source code of the site. After going to scroll down , there is a hint to flag,

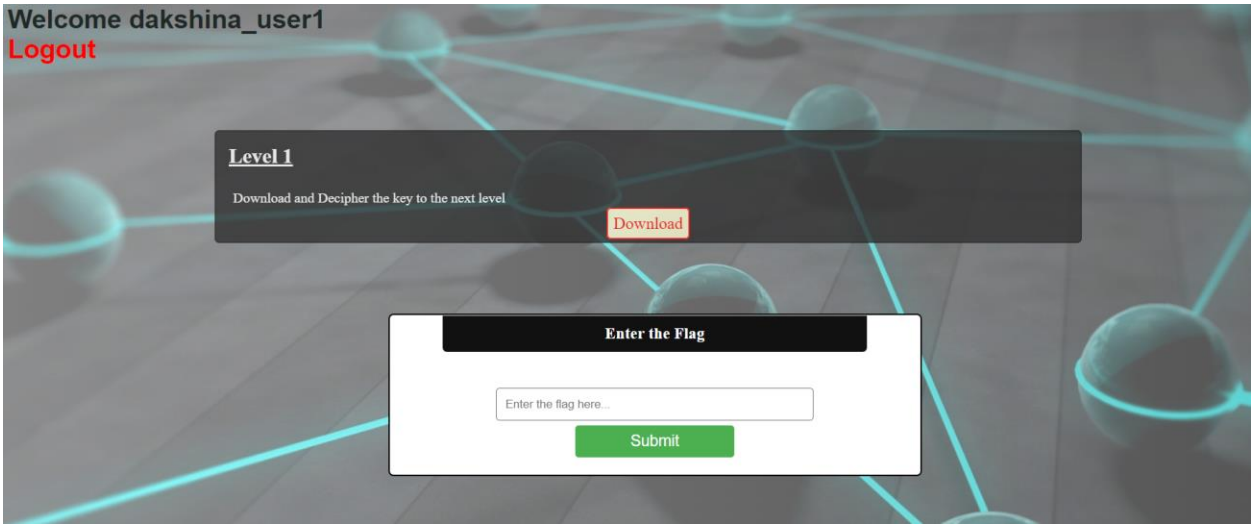
```
            $('.error-field').first().focus();
            valid = false;
        }
        return valid;
    }
</script>

<script>
/* Set the width of the sidebar to 250px and the left margin of the page content to 250px */
function openNav() {
  document.getElementById("mySidebar").style.width = "200px";
  /* QW5GQmhidkM1akhiZDU0ZGJKbmQ1bkRuOG5HZ25LNmc= -> b64 */
  document.getElementById("main").style.marginLeft = "200px";
}

/* Set the width of the sidebar to 0 and the left margin of the page content to 0 */
function closeNav() {
  document.getElementById("mySidebar").style.width = "0";
  document.getElementById("main").style.marginLeft = "0";
}
</script>


<style>.footer,.generic-footer{margin-bottom:98px}@media (min-width:374px){.footer,.generic-footer{margin-bott
</HTML>
```

Small hint to use base 64, user must be decode it with base 64 decoder,



Flag is : **AnFBhbvC5jHbd54dbJnd5nDn8nGgnK6g**

Level 01



First download the files,

Essential links that need to go ,

https://asecuritysite.com/encryption/ferdecode

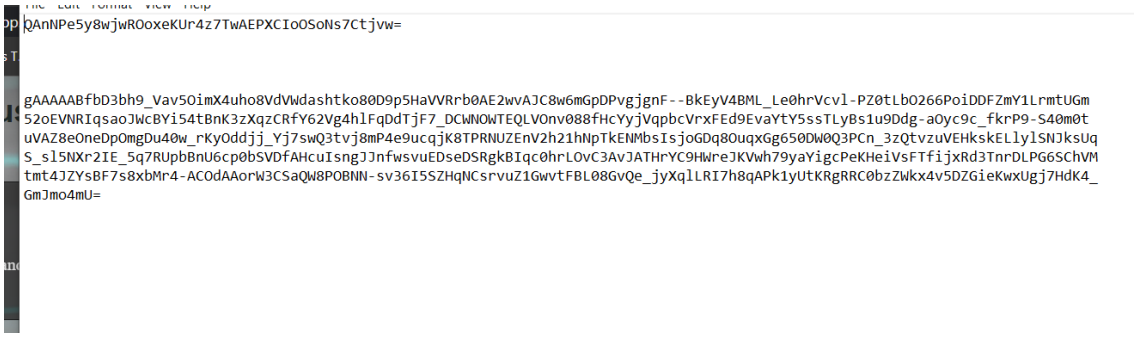
- for fernet

https://gchq.github.io/CyberChef/

-for decrypt

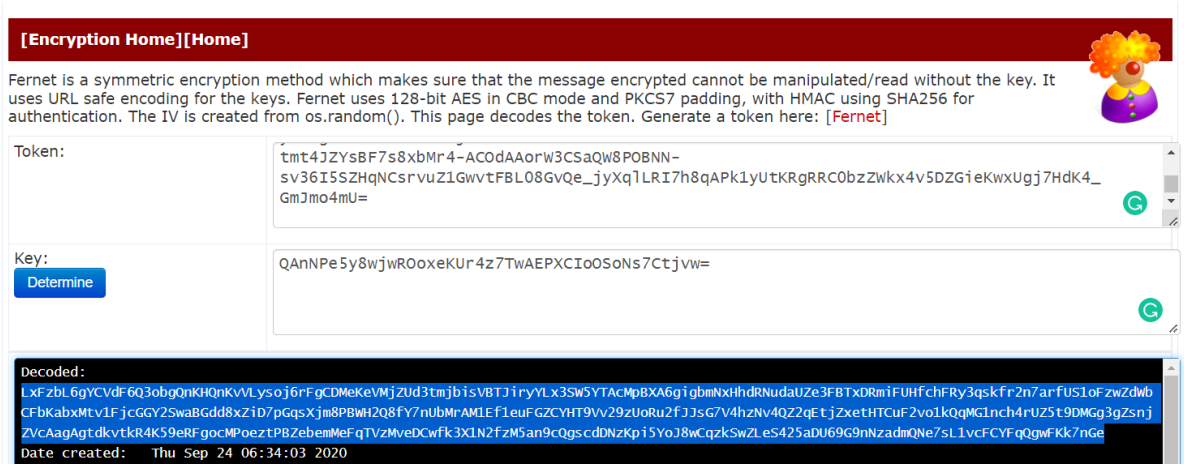
https://malbolge.doleczek.pl

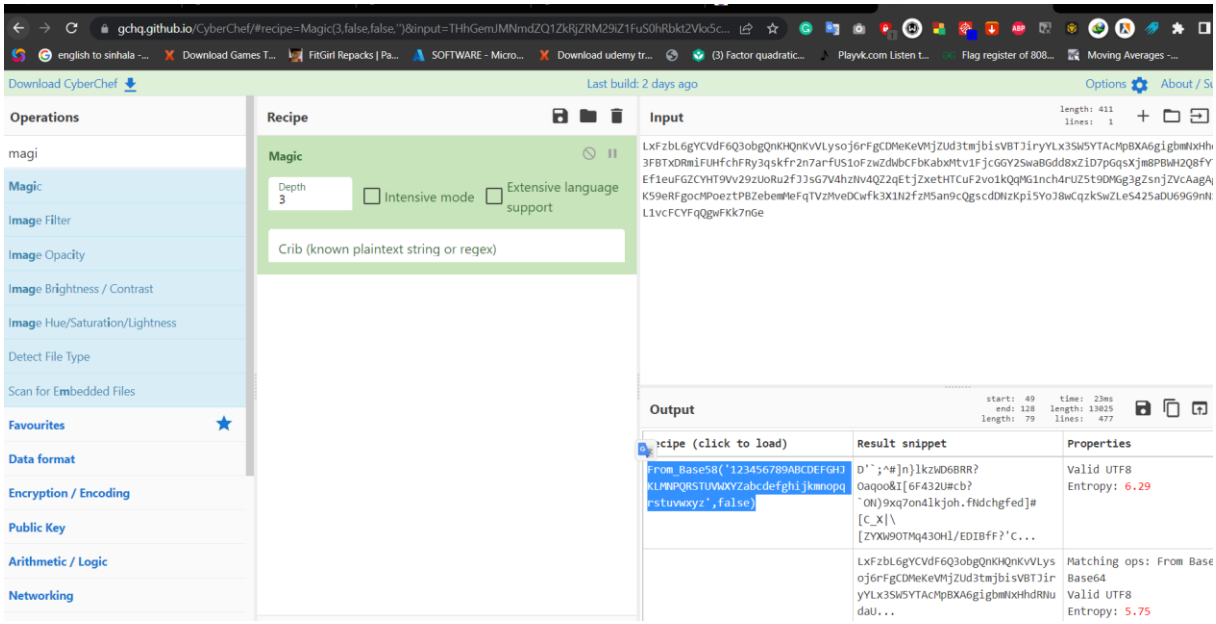
- for find the flag

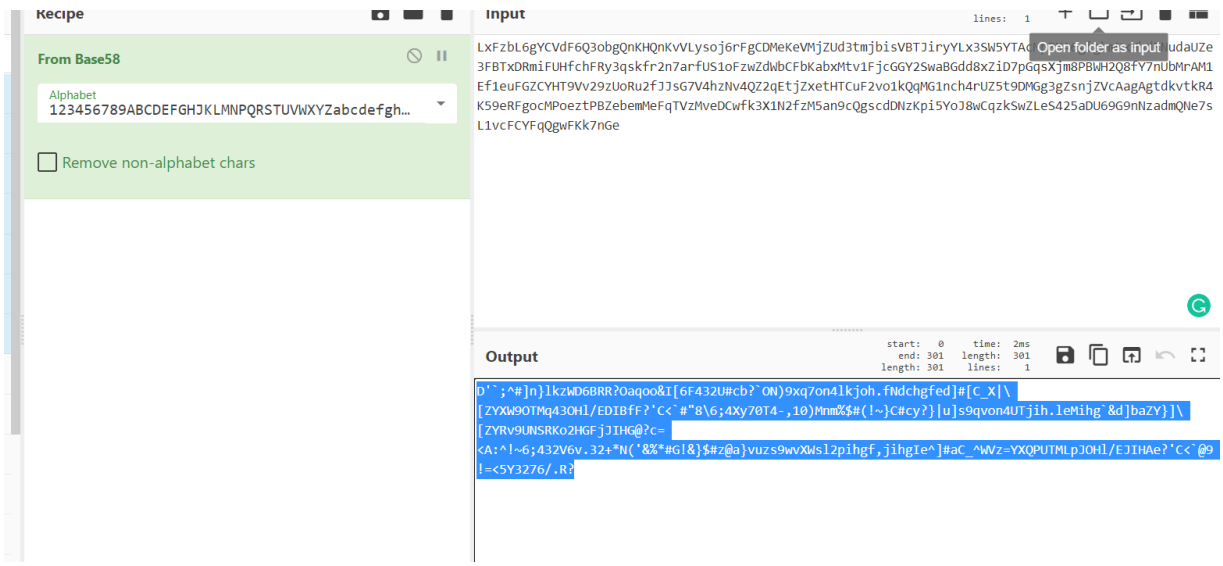

QAnNPe5y8wjwROoxeKUr4z7TwAEPXCIoOSoNs7Ctjvw=

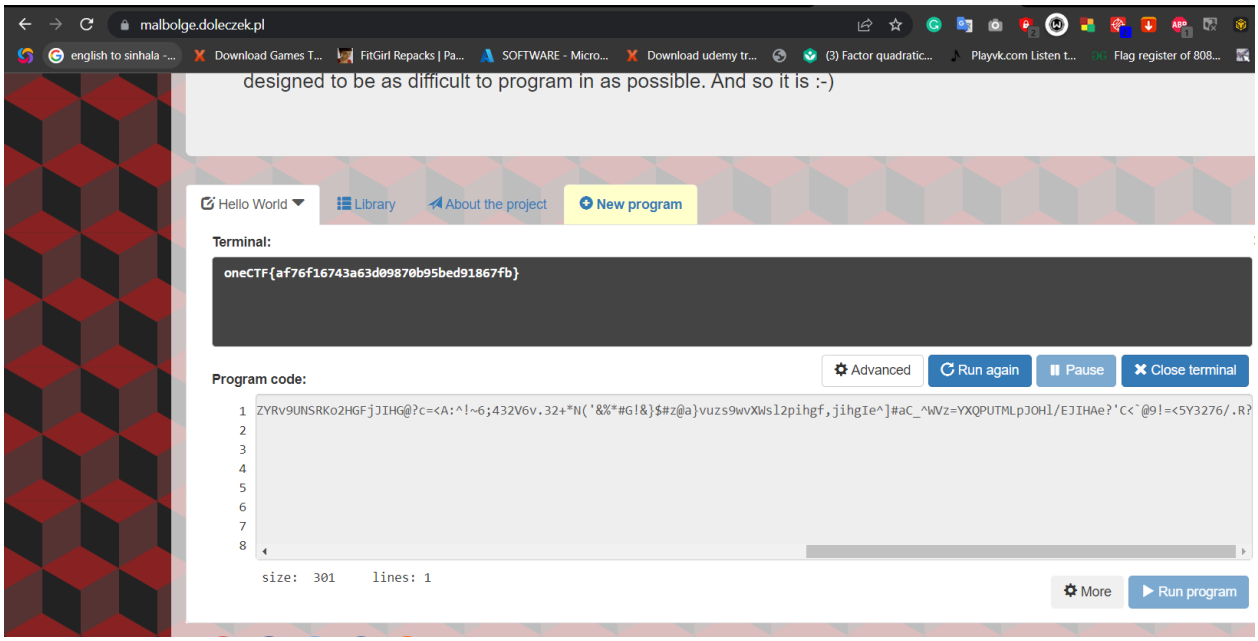
gAAAAABfbD3bh9_Vav5OimX4uho8VdVWdashtko80D9p5HaVVRrb0AE2wvAJC8w6mGpDPvgjgnF--BkEyV4BML_Le0hrVcvl-PZ0tLbO266PoiDDFZmY1LrmtUGm
52oEVNRIqsaoJWcBYi54tBnK3zXqzCRfY62Vg4hlFqDdTjF7_DCWNOWTEQLVOnv088fHcYyjVqpbcVrxFEd9EvaYtY5ssTLyBs1u9Ddg-aOyc9c_fkrP9-S40m0t
uVAZ8eOneDpOmgDu40w_rKyOddjj_Yj7swQ3tvj8mP4e9ucqjK8TPRNUZEnV2h21hNpTkENMbsIsjoGDq8OuqxGg650DW0Q3PCn_3zQtvzuVEHkskELlylSNJksUq
S_sl5NXr2IE_5q7RUpbBnU6cp0bSVDfAHcuIsngJJnfwsvuEDseDSRgkBIqc0hrLOvC3AvJATHrYC9HWreJKVwh79yaYigcPeKHeiVsFTfijxRd3TnrDLPG6SChVM
tmt4JZYsBF7s8xbMr4-ACOdAAorW3CSaQW8POBNN-sv36I5SZHqNCsrvuZ1GwvtFBL08GvQe_jyXqlLRI7h8qAPk1yUtKRgRRC0bzZWkx4v5DZGieKwxUgj7HdK4_
GmJmo4mU=

In this level gives us a value.Its looks like a base 64 value,but its not a base 64 ,it's a encryption key for the AES encryption call fernet.Its a URL safe encryption.there are so many online tools for decryption.we had to decrypt the and message as well.



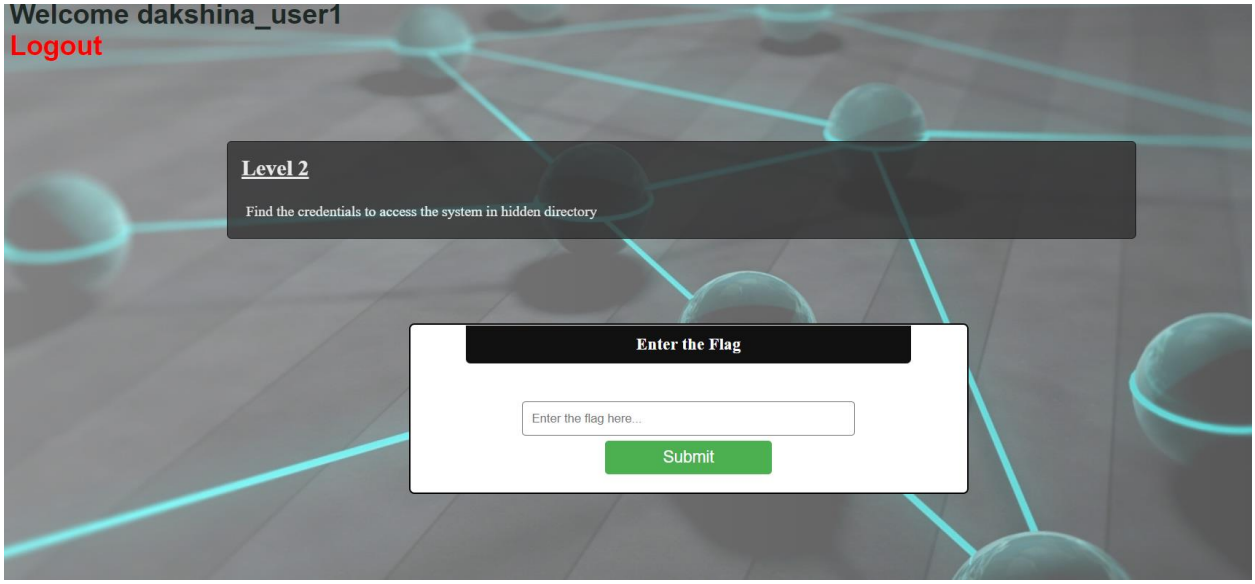

gchq.github.io/CyberChef/#recipe=Magic(3,false,false,'')&input=THhGemJMNmdZQ1ZkRjZRM29iZ1FuS0hRbkt2Vko5c...
Download CyberChef
Last build: 2 days ago
Options
About / S
Operations
magi
Magic
Image Filter
Image Opacity
Image Brightness / Contrast
Image Hue/Saturation/Lightness
Detect File Type
Scan for Embedded Files
Favourites
Data format
Encryption / Encoding
Public Key
Arithmetic / Logic
Networking
Recipe
Magic
Depth
3
Intensive mode
Extensive language support
Crib (known plaintext string or regex)
Input
length: 411
lines: 1
Output
start: 49
end: 128
length: 79
time: 23ms
length: 13025
lines: 477
Recipe (click to load)
Result snippet
Properties
From_Base58('123456789ABCDEFGHJKLMNPQRSTUVWXYZabcdefghijkmnopqrstuvwxyz',false)
D'`;^#]n}lkzWD6BRR?Oaqoo&I[6F432U#cb?`ON)9xq7on4lkjoh.fNdchgfed]#[C_X|\[ZYXW9OTMq43OHl/EDIBfF?'C...
Valid UTF8
Entropy: 6.29
LxFzbL6gYCVdF6Q3obgQnKHQnKvVLysoj6rFgCDMeKeVMjZUd3tmjbisVBTJiryYLx3SW5YTAcMpBXA6gigbmNxHhdRNudaU...
Matching ops: From Base64
Base64
Valid UTF8
Entropy: 5.75


Recipe
From Base58
Alphabet
123456789ABCDEFGHJKLMNPQRSTUVWXYZabcdefgh...
Remove non-alphabet chars
Input
lines: 1
Open folder as input
Output
start: 0
end: 301
length: 301
time: 2ms
length: 301
lines: 1
D'`;^#]n}lkzWD6BRR?Oaqoo&I[6F432U#cb?`ON)9xq7on4lkjoh.fNdchgfed]#[C_X|\[ZYXW9OTMq43OHl/EDIBfF?'C<`#"8\6;4Xy70T4-,10)Mnm%%$#(!~}C#cy?}|u]s9qvon4UTjih.leMihg`&d]baZY}]\[ZYRv9UNSRKo2HGFjJIHG@?c=<A:^!~6;432V6v.32+*N('&%*#G!&}$#z@a}vuzs9wvXWsl2pihgf,jihgIe^]#aC_^WVz=YXQPUTMLpJOHl/EJIHAe?'C<`@9!=<5Y3276/.R?



Flag is : **oneCTF{af76f16743a63d09870b95bed91867fb}**

Level 02

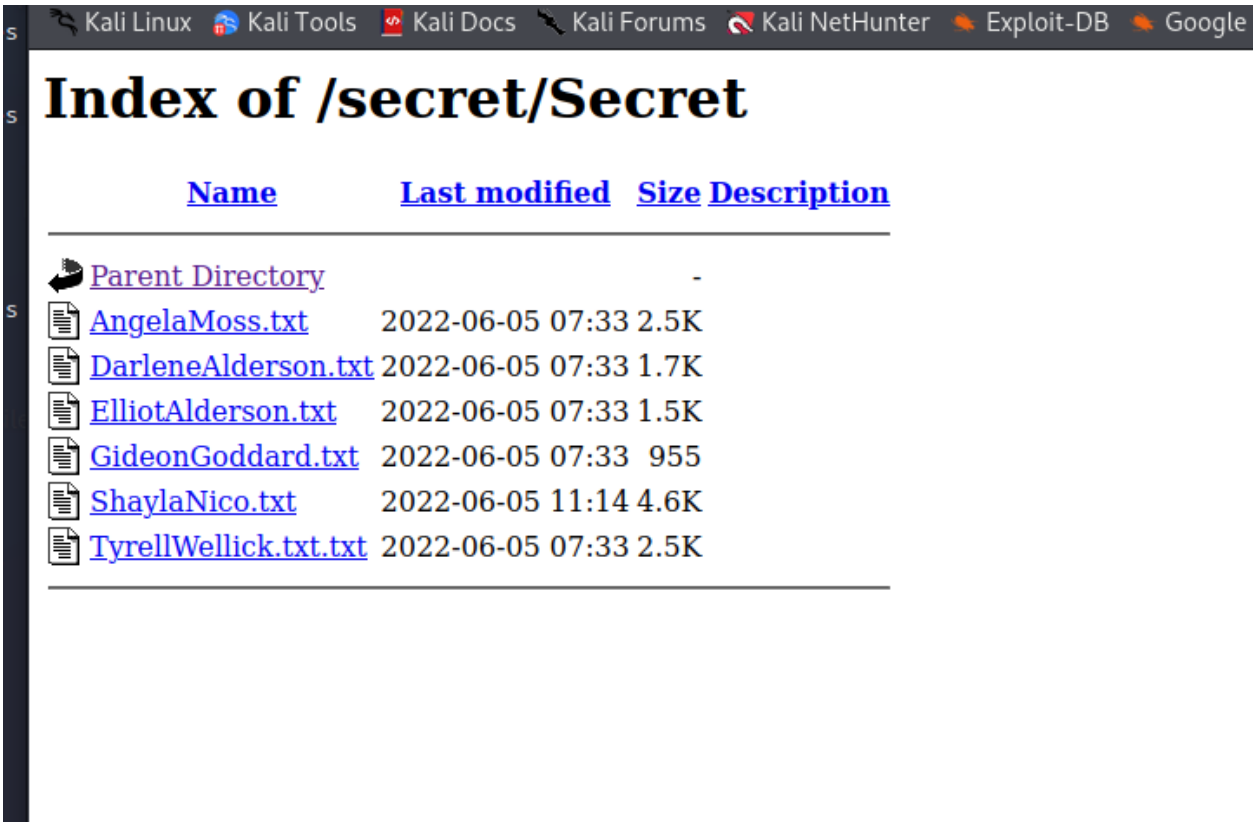


In this level user need to find the hidden directory to get the flag.User can use directory bruteforce tools like gobuster,dirbuster,ffuf.For this demonstration we gonna use ffuf tool.

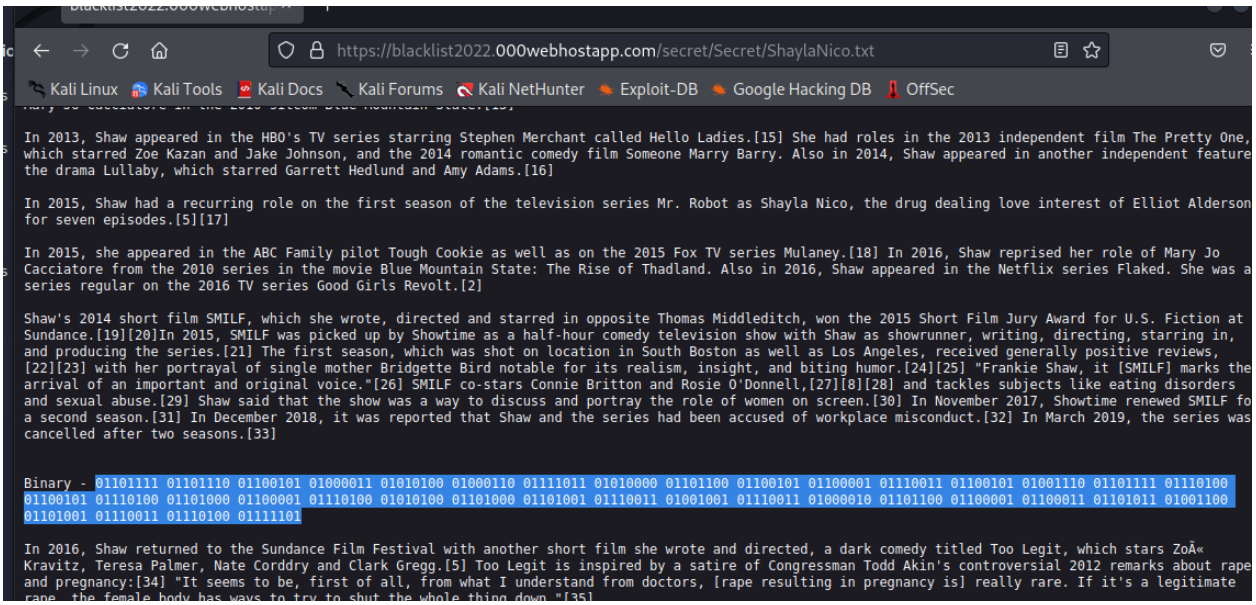
```
┌──(root㉿kali)-[~]
└─# ffuf -w /usr/share/wordlists/dirbuster/directory-list-lowercase-2.3-medium.txt:FUZZ -u https://theblacklistctf.000webhostapp.com//FUZZ

        /'___\  /'___\           /'___\
       /\ \__/ /\ \__/  __  __  /\ \__/
       \ \ ,__\\ \ ,__\/\ \/\ \ \ \ ,__\
        \ \ \_/ \ \ \_/\ \ \_\ \ \ \ \_/
         \ \_\   \ \_\  \ \____/  \ \_\
          \/_/    \/_/   \/___/    \/_/

       v1.3.1 Kali Exclusive <3
_____________________________________________________________________________

 :: Method           : GET
 :: URL              : https://theblacklistctf.000webhostapp.com//FUZZ
 :: Wordlist         : FUZZ: /usr/share/wordlists/dirbuster/directory-list-lowercase-2.3-medium.txt
 :: Follow redirects : false
 :: Calibration      : false
 :: Timeout          : 10
 :: Threads          : 40
 :: Matcher          : Response status: 200,204,301,302,307,401,403,405
_____________________________________________________________________________

                        [Status: 403, Size: 20474, Words: 5432, Lines: 661]
serial                  [Status: 403, Size: 20474, Words: 5432, Lines: 661]
logo                    [Status: 403, Size: 20474, Words: 5432, Lines: 661]
privacy                 [Status: 403, Size: 20474, Words: 5432, Lines: 661]
about                   [Status: 403, Size: 20474, Words: 5432, Lines: 661]
# directory-list-lowercase-2.3-medium.txt [Status: 403, Size: 20474, Words: 5432, Lines: 661]
blog                    [Status: 403, Size: 20474, Words: 5432, Lines: 661]
news                    [Status: 403, Size: 20474, Words: 5432, Lines: 661]
```

```
# directory-list-lowercase-2.3-medium.txt [Status: 200, Size: 14670, Words: 1674, Lines:
                        [Status: 200, Size: 14670, Words: 1674, Lines: 102]
#                       [Status: 200, Size: 14670, Words: 1674, Lines: 102]
# Copyright 2007 James Fisher [Status: 200, Size: 14670, Words: 1674, Lines: 102]
assets                  [Status: 301, Size: 5645, Words: 280, Lines: 8]
css                     [Status: 301, Size: 5642, Words: 280, Lines: 8]
test                    [Status: 301, Size: 5643, Words: 280, Lines: 8]
lib                     [Status: 301, Size: 5642, Words: 280, Lines: 8]
js                      [Status: 301, Size: 5641, Words: 280, Lines: 8]
com                     [Status: 301, Size: 5642, Words: 280, Lines: 8]
vendor                  [Status: 301, Size: 5645, Words: 280, Lines: 8]
sql                     [Status: 301, Size: 5642, Words: 280, Lines: 8]
fonts                   [Status: 301, Size: 5644, Words: 280, Lines: 8]
secret                  [Status: 301, Size: 5645, Words: 280, Lines: 8]
:: Progress: [5450/207643] :: Job [1/1] :: 129 req/sec :: Duration: [0:00:46] :: Errors:
zsh: suspended  ffuf -w  -u https://thegambit.000webhostapp.com//FUZZ
```

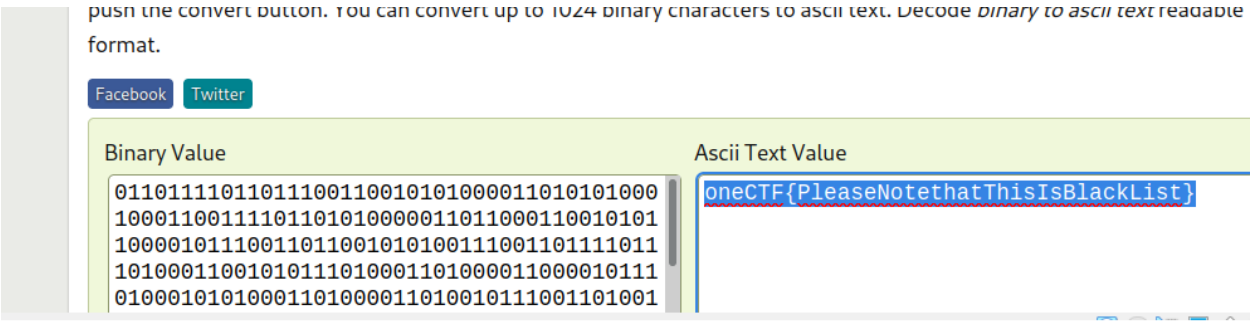
Find the directory called /Secret/. Inside the directory there is a few text files available.



Go with each file, and there is a one file that keeping binary code,

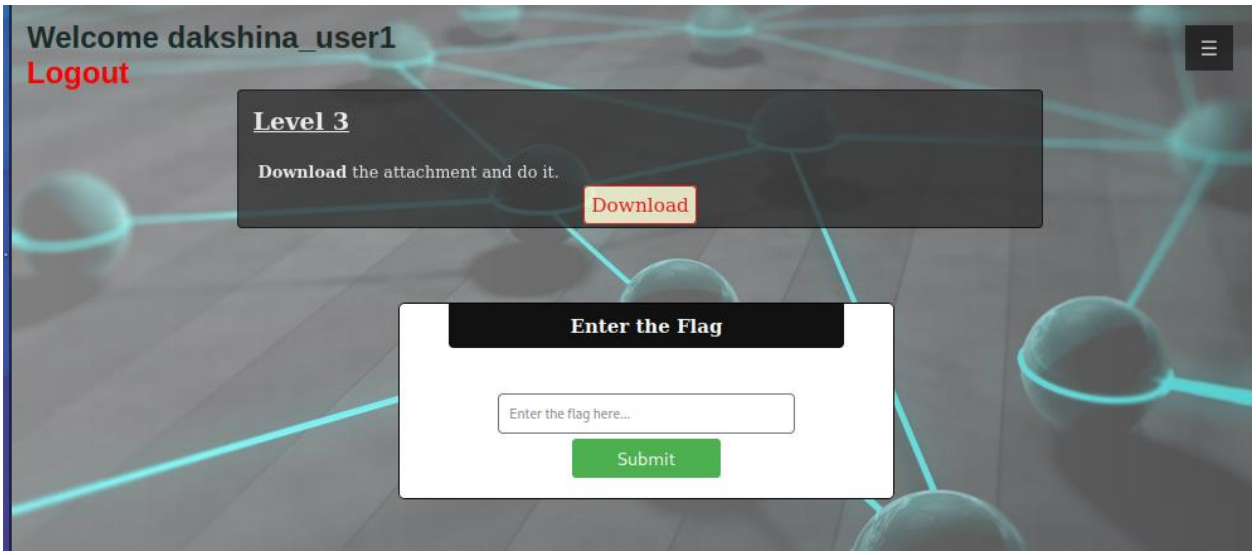


Inside the ShaylaNico.txt have a binary code. Use any binary decoder to decode the flag and later on submit in the submission form

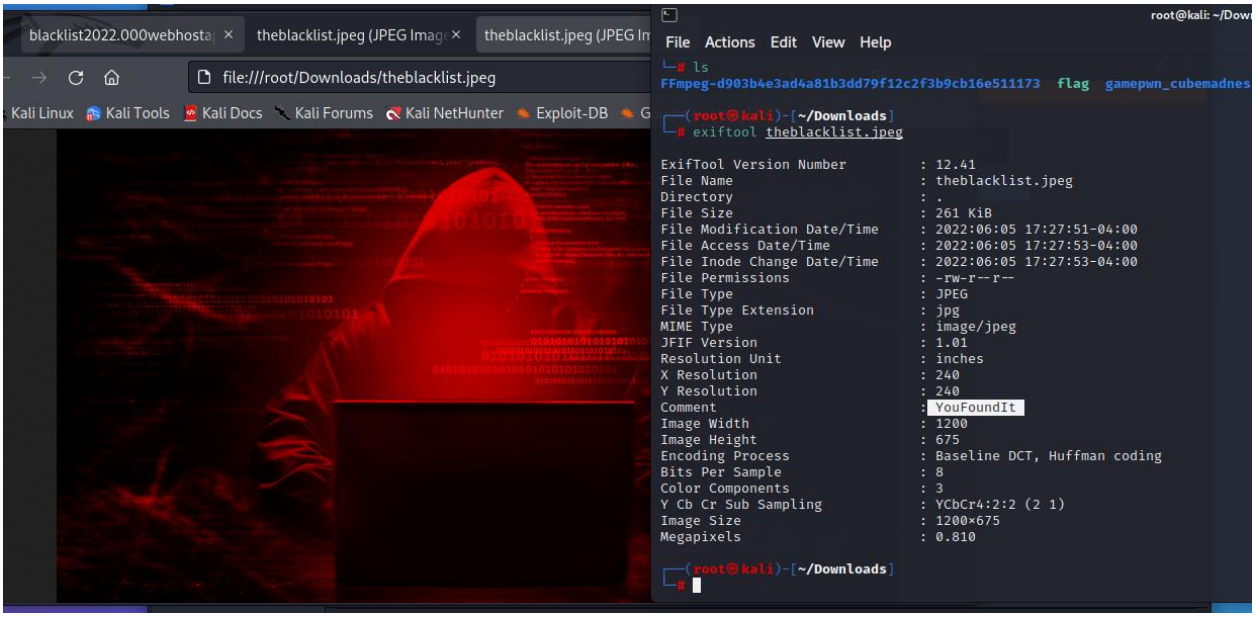


Flag is : **oneCTF{PleaseNotethatThisIsBlackList}**

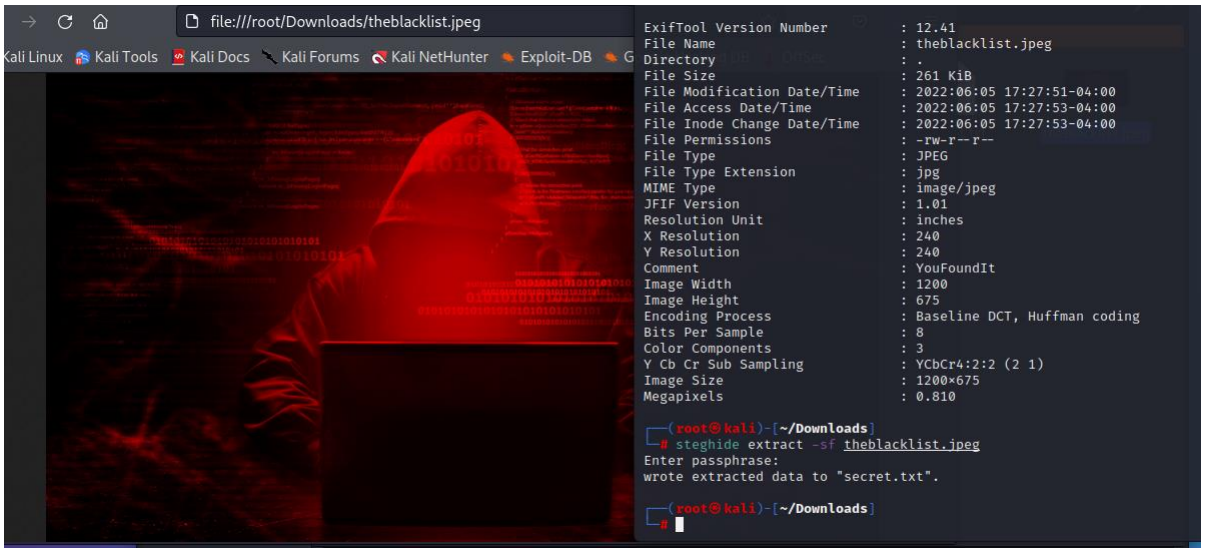
Level 03

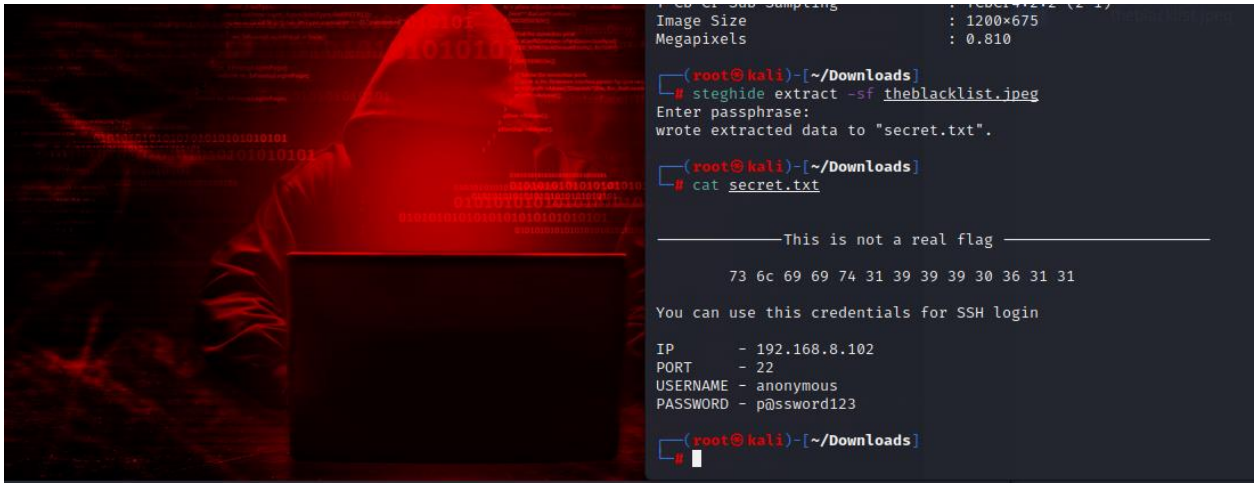


This is a steganography challenge. After downloading the Image file to a Linux environment. Scan the image for file type. The hints suggest of the METADATA, because of that we need a tool to see METADATA of the image. After enough research and the hint suggests Exiftool. Download and install the tool with the command: “sudo apt-get install exiftool”. After installing check, the image with the tool: “exiftool thegambit.jpg”. It shows a Comment with a passphrase. Next the hint points us of a tool to extract data hidden in the image

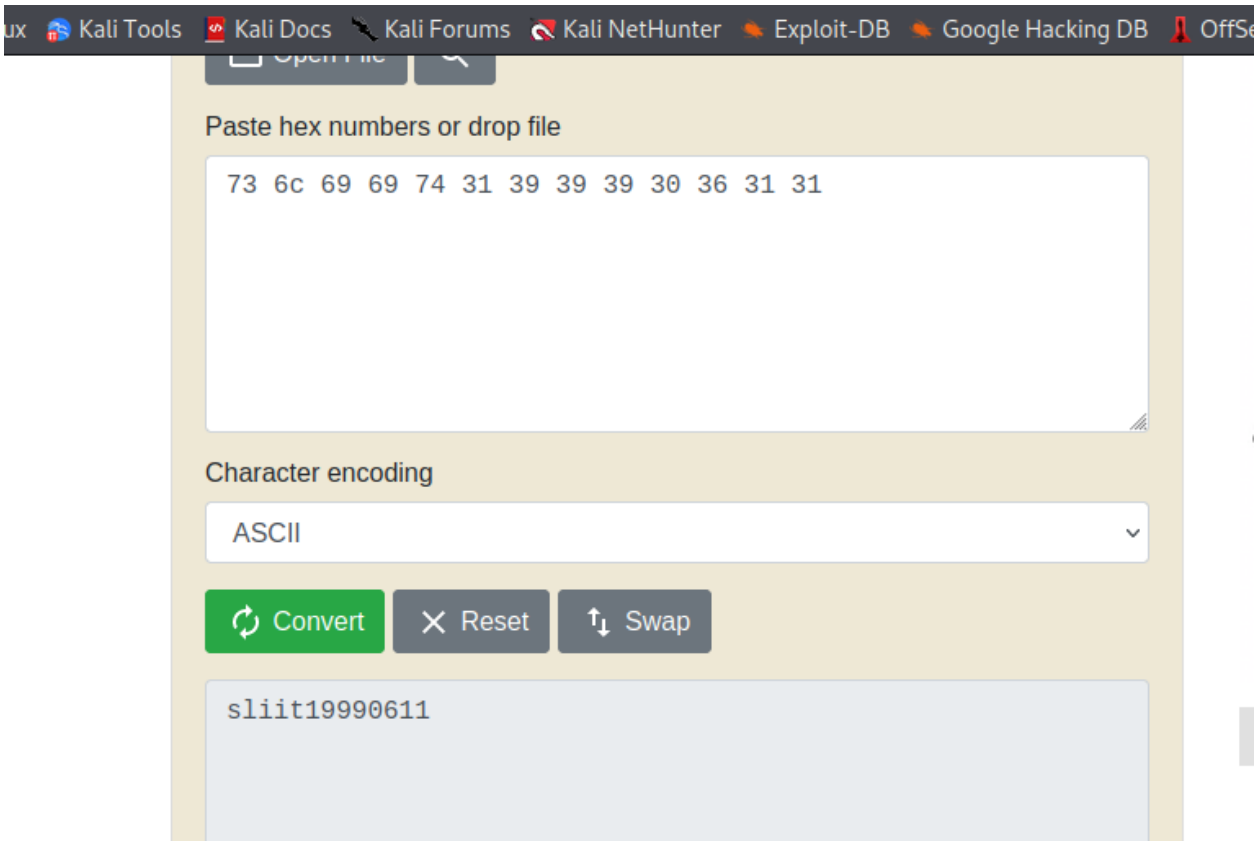


Install: “sudo apt-get install steghide”. Run the command: “steghide extract -sf thegambit.jpg”. Next the passphrase will be required, enter it. New file “secret” without an extension is extracted out of the image. Open it to find the FLAG:



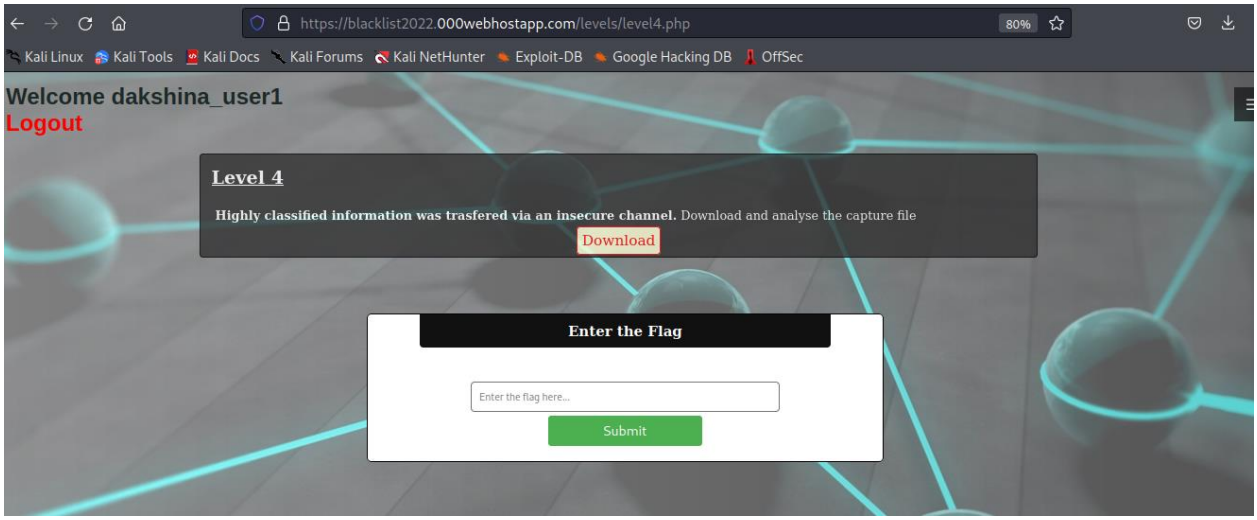


According to the above image the flag is encoded. Use any hex decoder to decode the flag and later on submit in the submission form



Flag is : **sliit19990611**

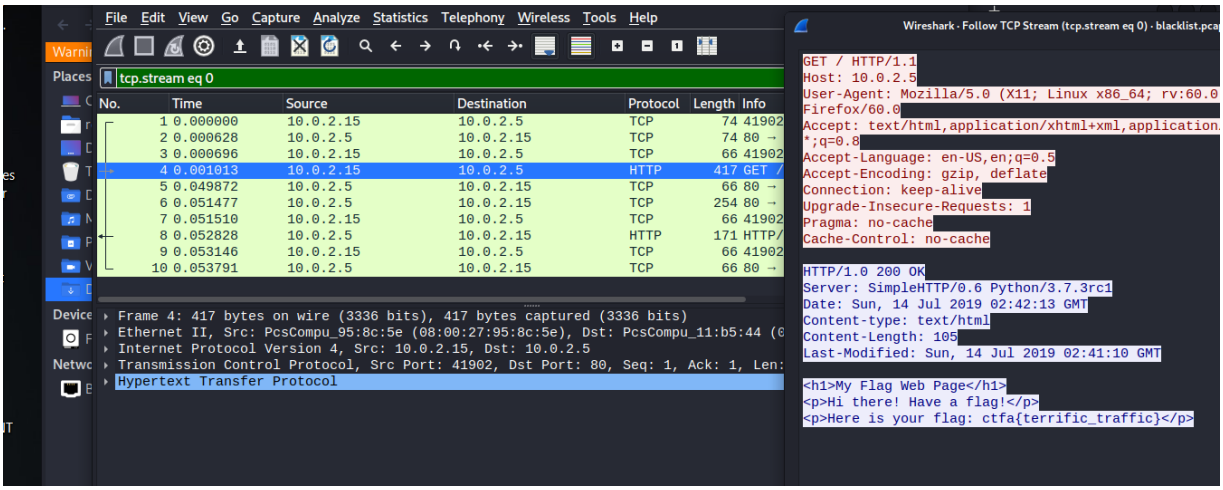
Level 04



This is a forensics challenge.User need to download the pcap file open it from wireshark.

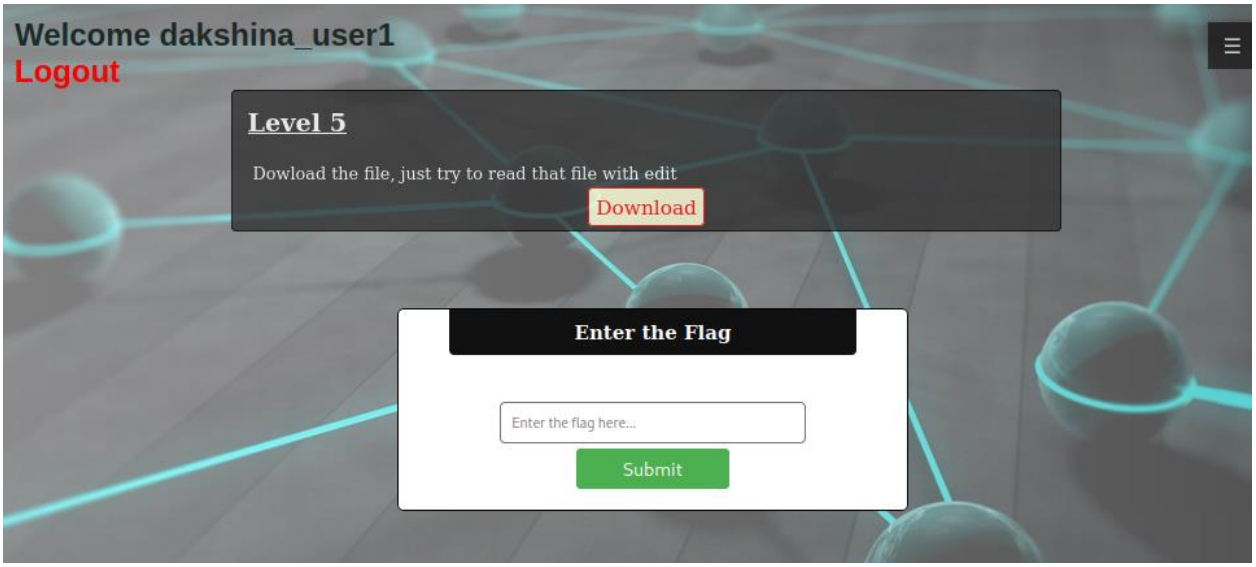
Then click on a packet and right click->follow tcp stream

Then click on a packet and right click->follow tcp stream

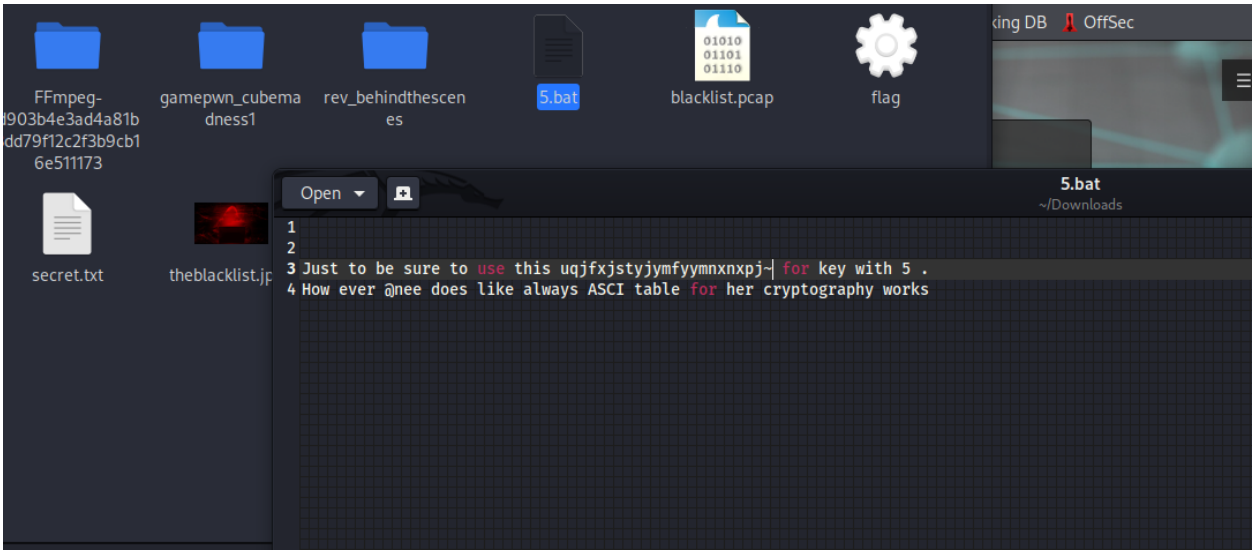


Flag is : **ctfa{terrific_traffic}**

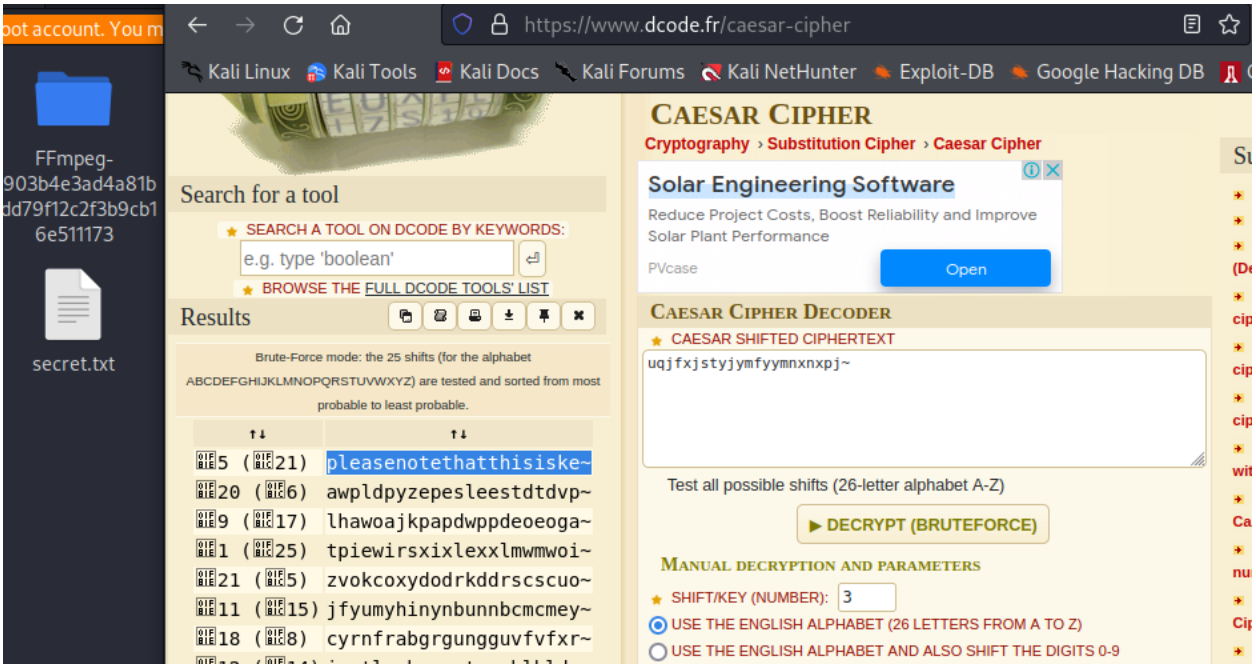
Level 05



This is the implementation of cease cipher,

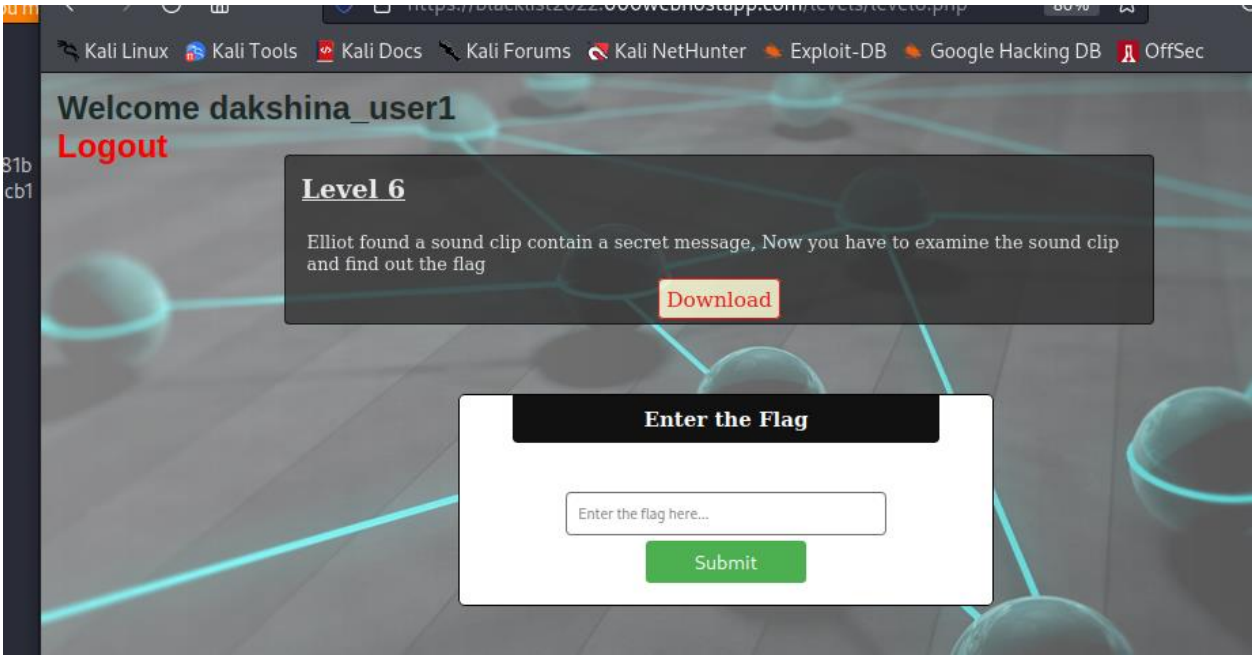


With the given scenario, encrypted message must decrypt with 5 keys in cease cipher with ASCII Table,



Flag is: **pleasenotethatthisiskey**

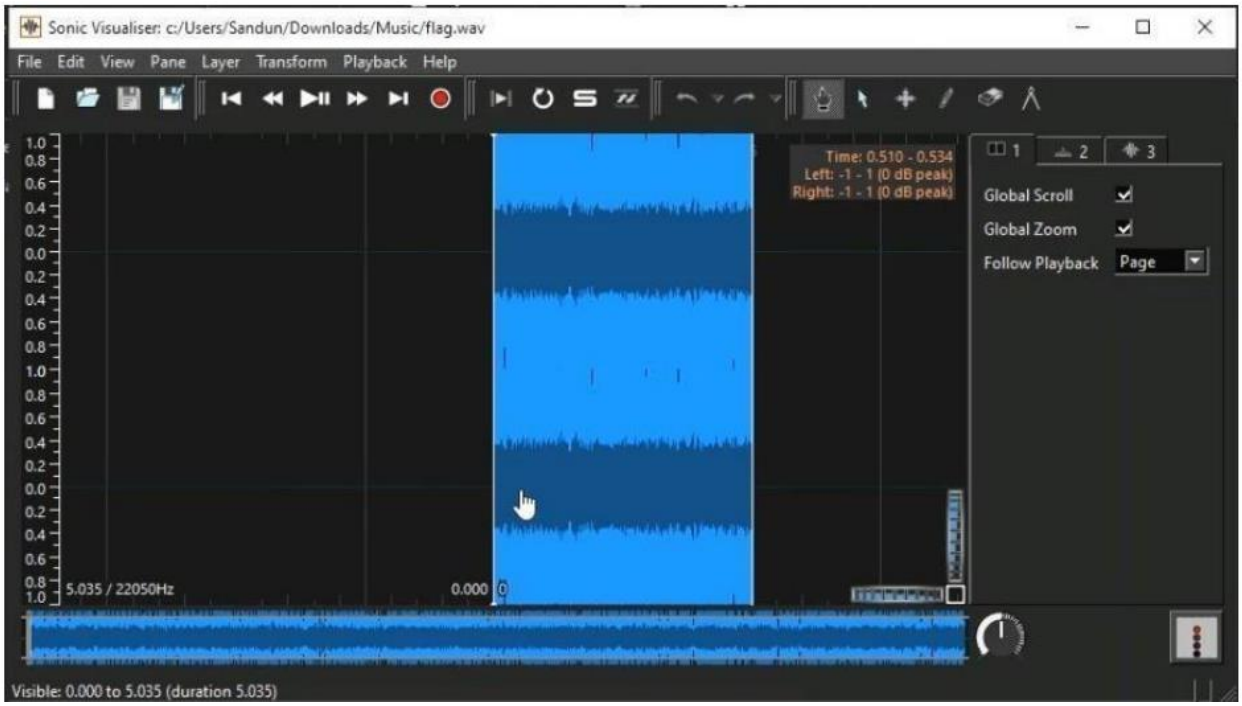
Level 06



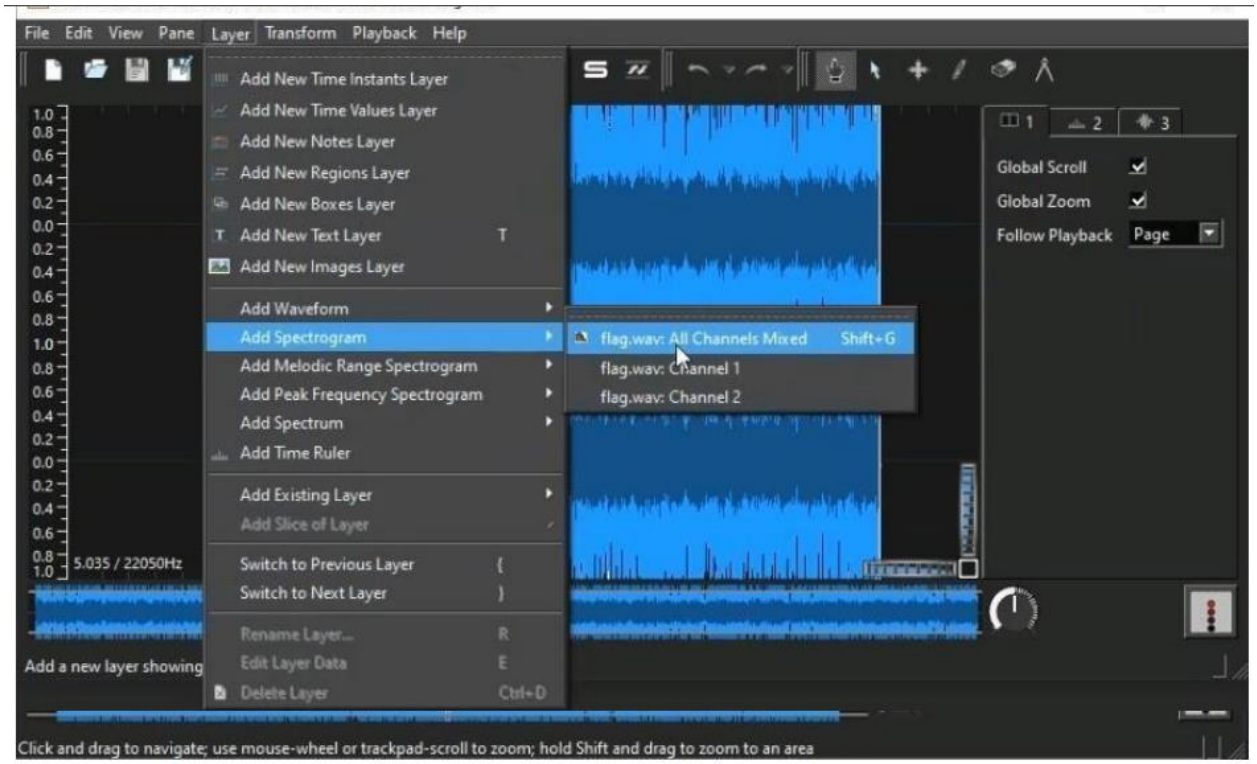
First download the sound clip.

Then we want a Sonic Visualiser software.

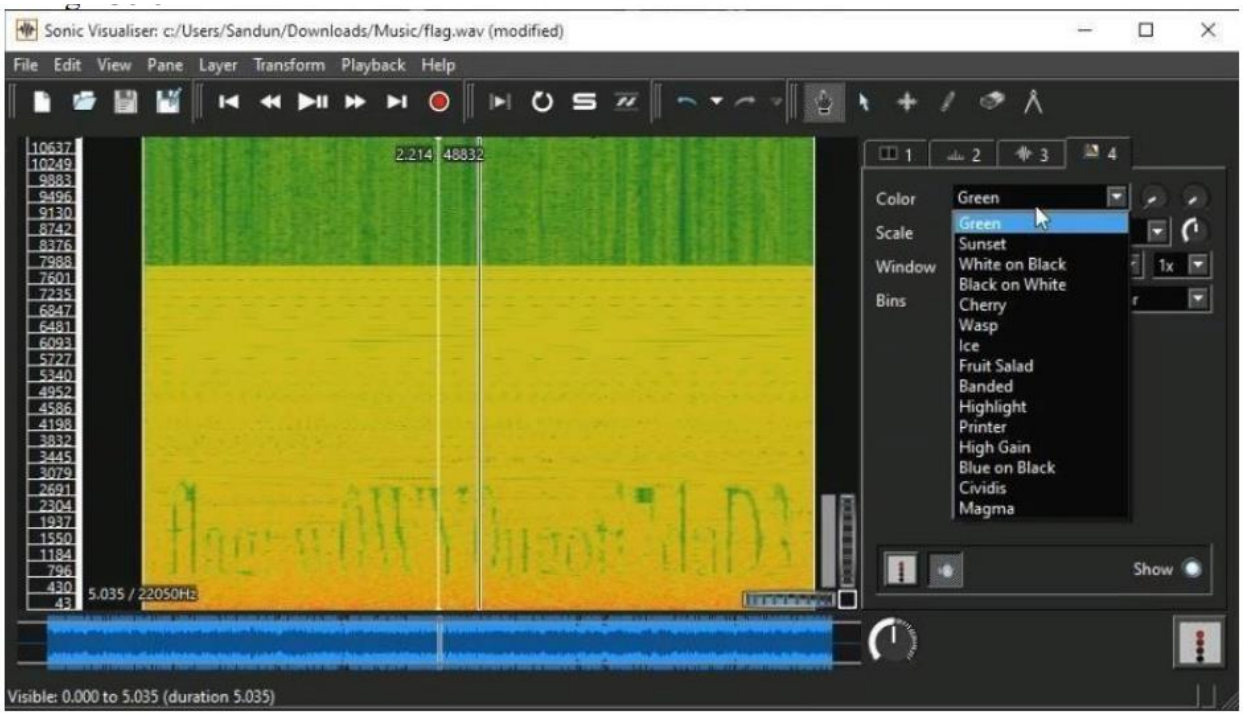
Then open the sound clip on Sonic Visualiser.



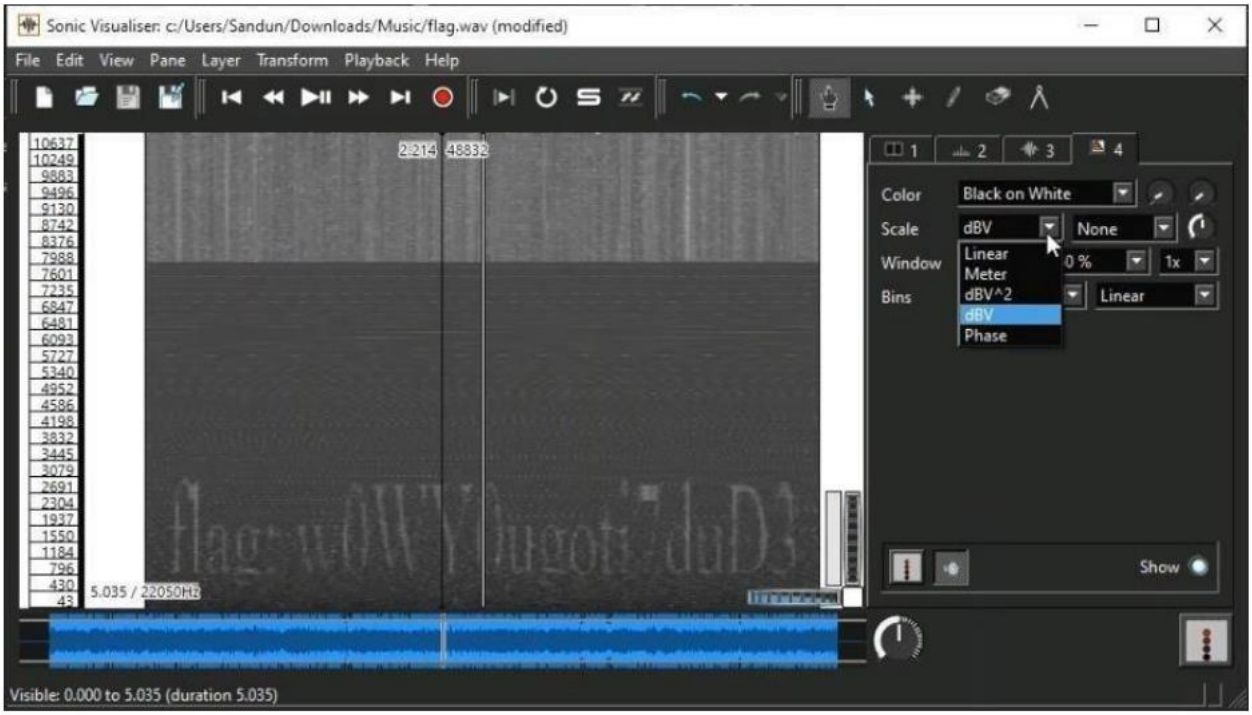
Then go to Layer > Add Spectrogram > flag.wav: All Channels Mixed and add that. Then we want to change Color, Scale, Window and Bins to get clear Image in Spectrogram layer. Then we can saw the flag in that.



Change Color to black and white

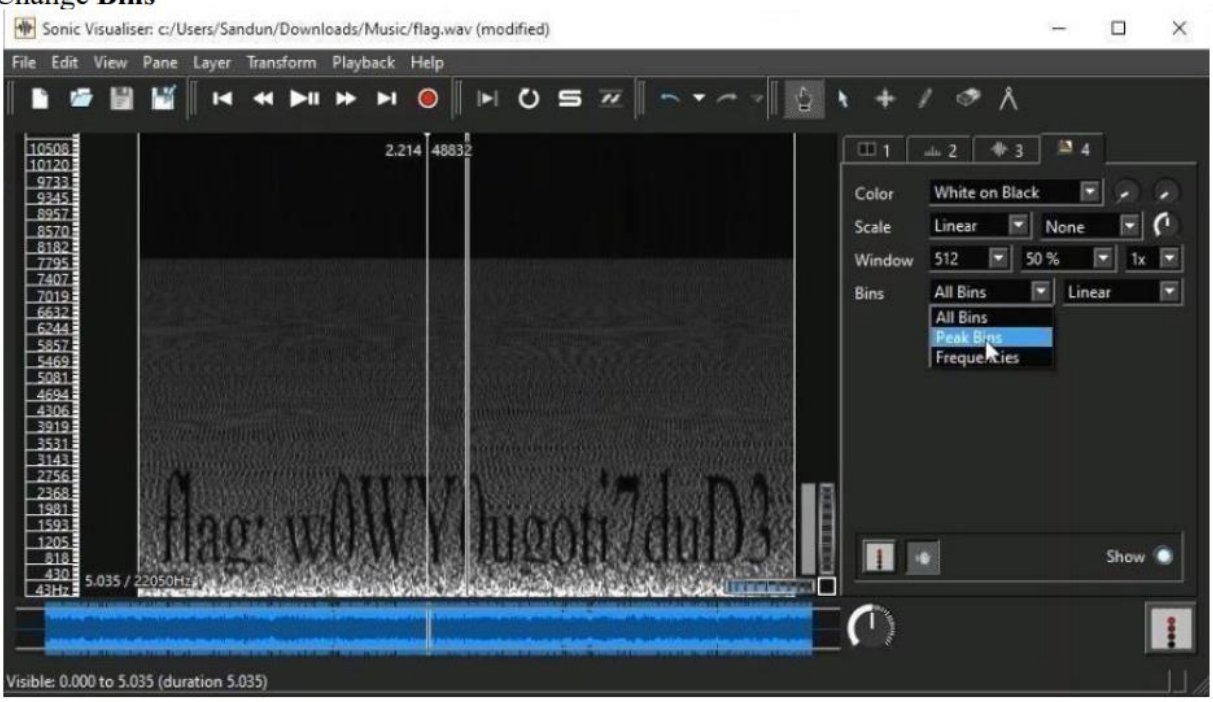


Change Scale to linear



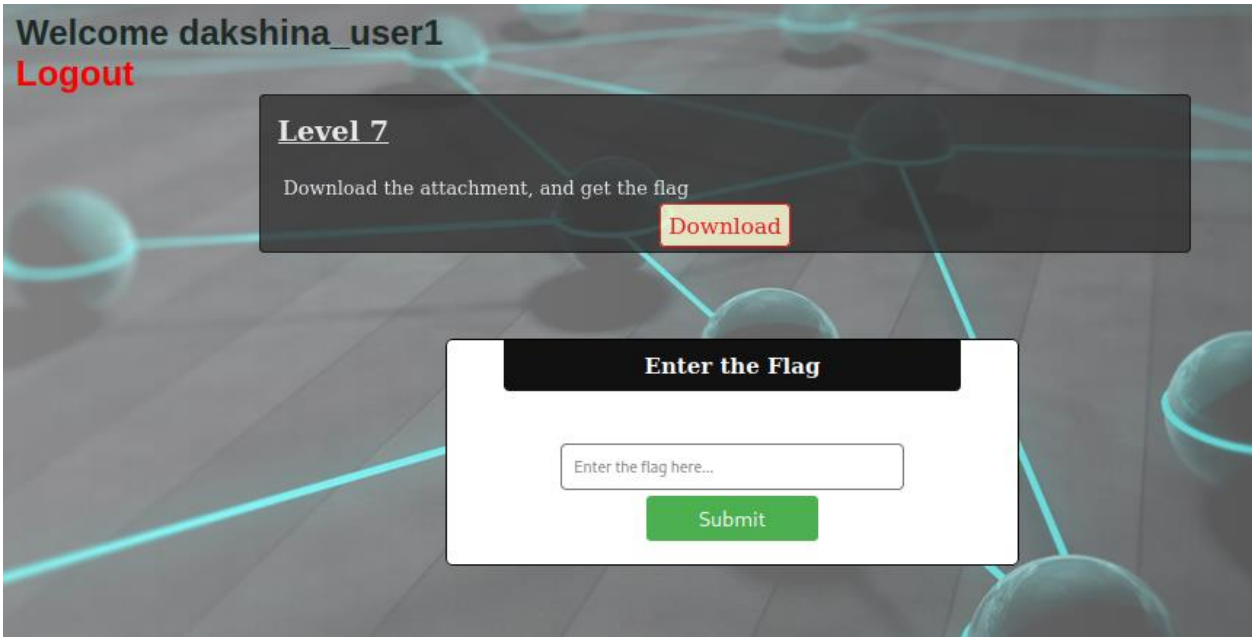
Change Window to 512

Change **Bins**



Flag can be seen as above **: woWYOugoti7duD3**

Level 07



First, I download the attachment on my Linux machine -> Using wget ‘path’ command. and then I unzip the .zip file. -> Using unzip flag.zip Then I get the details about flag file (to know about what kind of file). Using -> file flagfile flag

That flag file is executable file. but file can't execute because of their wasn't permission. So I use command as -> chmod +x flag Then I execute the flag file using -> ./flag command.

```
└─# ls
5.bat            FFmpeg-d903b4e3ad4a81b3dd79f12c2f3b9cb16e511173  gamepwn_cubemadness1  secret.txt
blacklist.pcap   flag                                             rev_behindthescenes   theblacklist.jpeg

┌──(root㉿kali)-[~/Downloads]
└─# unzip flag
Archive:  flag
replace flag? [y]es, [n]o, [A]ll, [N]one, [r]ename: y
  inflating: flag

┌──(root㉿kali)-[~/Downloads]
└─# ls
5.bat            FFmpeg-d903b4e3ad4a81b3dd79f12c2f3b9cb16e511173  gamepwn_cubemadness1  secret.txt
blacklist.pcap   flag                                             rev_behindthescenes   theblacklist.jpeg

┌──(root㉿kali)-[~/Downloads]
└─# chmod +x flag

┌──(root㉿kali)-[~/Downloads]
└─# file flag
flag: ELF 64-bit LSB pie executable, x86-64, version 1 (SYSV), dynamically linked, interpreter /lib64/ld-linux-x86-64.so.2, for GNU/Linux 3.2.
0, BuildID[sha1]=2fb84c55912cd8657e1d45140dfcbab1a96cc00f, not stripped

┌──(root㉿kali)-[~/Downloads]
└─# ./flag

Mr. Robot CTF!

Enter the passphrase:
```

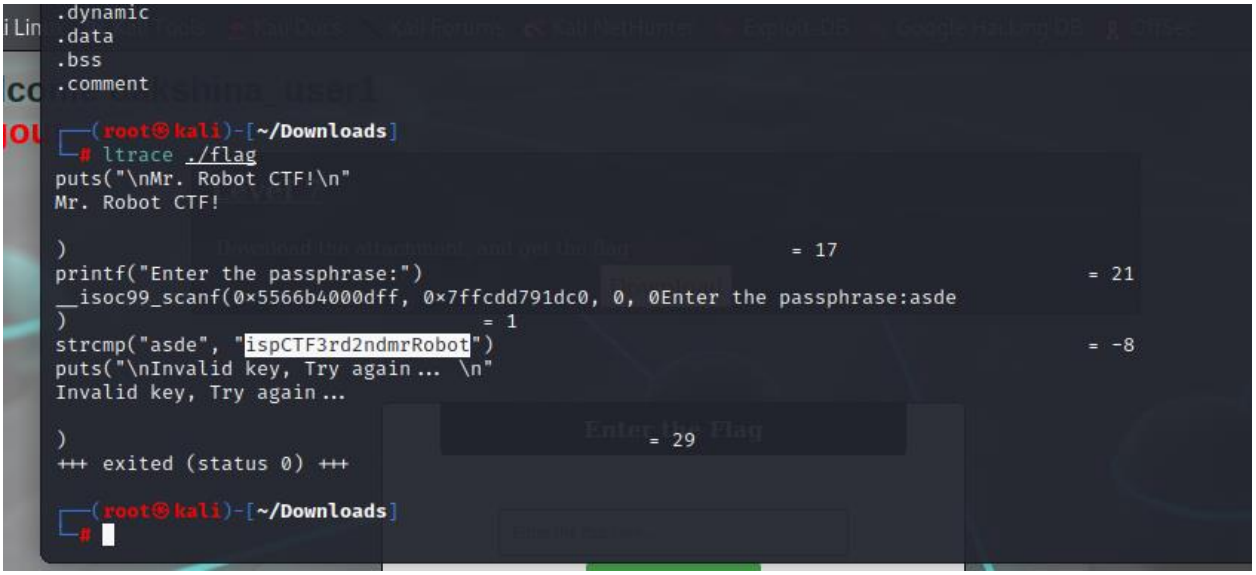
That ask Passphrase, But I do not know the passphrase. I tried random password as flag their was a message Invalid key

Then I use strings ./flag command to determine the contents of and to extract text from binary files.

```
root@kali: ~/Downloads
File  Actions  Edit  View  Help
┌──(root㉿kali)-[~/Downloads]
└─# strings ./flag
/lib64/ld-linux-x86-64.so.2
libc.so.6
__isoc99_scanf
puts
__stack_chk_fail
printf
__cxa_finalize
strcmp
__libc_start_main
GLIBC_2.7
GLIBC_2.4
GLIBC_2.2.5
_ITM_deregisterTMCloneTable
__gmon_start__
_ITM_registerTMCloneTable
AWAVI
AUATL
[]A\A]A^A_
Mr. Robot CTF!
Enter the passphrase:
Invalid key, Try again...
||||||||||||||||||||||||||||||||||||
||      N0oneC4nD0th!sp4n          ||
||    B3N1cet0Othersdr3nkjsq       ||
||       N00n3cand0itaBl3          ||
||     d0B3strLe79N24J@p4n         ||
||       H0p3yoU9N24J@p4n          ||
```

There are so many strings I cannot all of those as passphrase key. Then I use ltrace ./flag command to find passphrase. After using ltrace command that ask Passphrase I do not know the Passphrase.

Then I Enter asde for passphrase, but you can use any value as passphrase. That compare Entered value and matched passphrase under strcmp. ltrace is a program that simply runs the specified command until it exits. It intercepts and records the dynamic library calls which are called by the executed process and the signals which 43 are received by that process. It can also intercept and print the system calls executed by the program. The strcmp() function compares the two strings s1 and s2. It returns an integer less than, equal to, or greater than zero if s1 is found, respectively, to be less than, to match, or be greater than s2.



That selected value is the Passphrase of the flag file. Then I enter that value for passphrase in flag file.