Sri Lanka Institute of Information Technology

Final Project Report

ISP Project Report

Information Security Project 2021

Project ID: **ISP-21-REG-19**

Submitted by:

IT Number	Name
Eranda H.P.D	IT19029146
S.M.Rathnayaka	IT19040936

14/11/2021

Date of submission

Abstract

The final report of the CTF Box project is created in order to comprehensively explain you about the pathway that we have used in carrying out this particular CTF Box project. For the ease of understanding, we have divided this report for several sub-sections.

Under the Introduction section, we are going to explain you about the problem that we have selected and the storyline. After that we are going to explain you about the scope of our CTF Box project and the target audience of it. This also includes the benefits, objectives and the goals of our CTF Box project. Furthermore mentioned, this also consists with the corporate goals & business strategies as well. Since we have selected a topic which is more related to the healthcare industry, we are always focusing on protecting the patient records & its database system.

After the Introduction section, we are going to explain you about the methodologies that we have used when implementing our CTF Box. Under this section we are hoping to discuss about the Requirement Analysis Stage, Design Stage, Implementation Stage & the Testing Stage.

After the Methodologies section, we are going to explain you on the practical evaluation of the project topic. This particular section consists with the Assessment of the Project Results, Lessons we've learnt and the Future Researching Areas.

Then as the final section, we are going to provide the summarized conclusion of our CTF Box Project & the references we have used throughout this report. Furthermore mentioned, as some additional information, we are going to use the Appendix A: section to demonstrate our all the test results in order of a complete walkthrough.

Acknowledgement

As the 3rd year 2nd semester Cyber Security undergraduates at SLIIT, under the module **"Information Security Project" (IE3092),** we were asked to create a CTF Box for a given specific field. After done creating the CTF Box, we are supposed to host it in an online gamified real-world lab platform called as **"TryHackMe".**

So, the topic that we have chosen for our CTF box is **"A cyber-attack towards the Patients' Records Management System in a hospital"**. Since our CTF Box is mainly focused on the healthcare sector, we have selected our target audience as all the medical staff that are working in the hospital who are responsible in the field of cyber security.

First of all, I would like to be very grateful to our lecturer in charge Dr. Lakmal Rupasinghe for giving us this wonderful opportunity to enhance our knowledge and experience on CTF challenges. His immense kind & consistent support helped us a lot specially in choosing a better topic for this assignment. Not only that, he also checked our CTF Box assignment progress regularly by providing us with different submissions. This has provided us with a valuable opportunity to correct our mistakes weekly in our assignment. Finally, I would like to express my and my teammate's greatest gratitude for the associated lecturers & instructors for guiding us throughout this entire research project.

Declaration

We declare that this project report or part of it was not a copy of a document done by any organization, university any other institute or a previous student project group at SLIIT and was not copied from the Internet or other sources.

Project Details

Project Title	**TheMoneyHeist - A cyber-attack towards the Patient Records Management System in a Hospital**
Project ID	**ISP-21-REG-19**

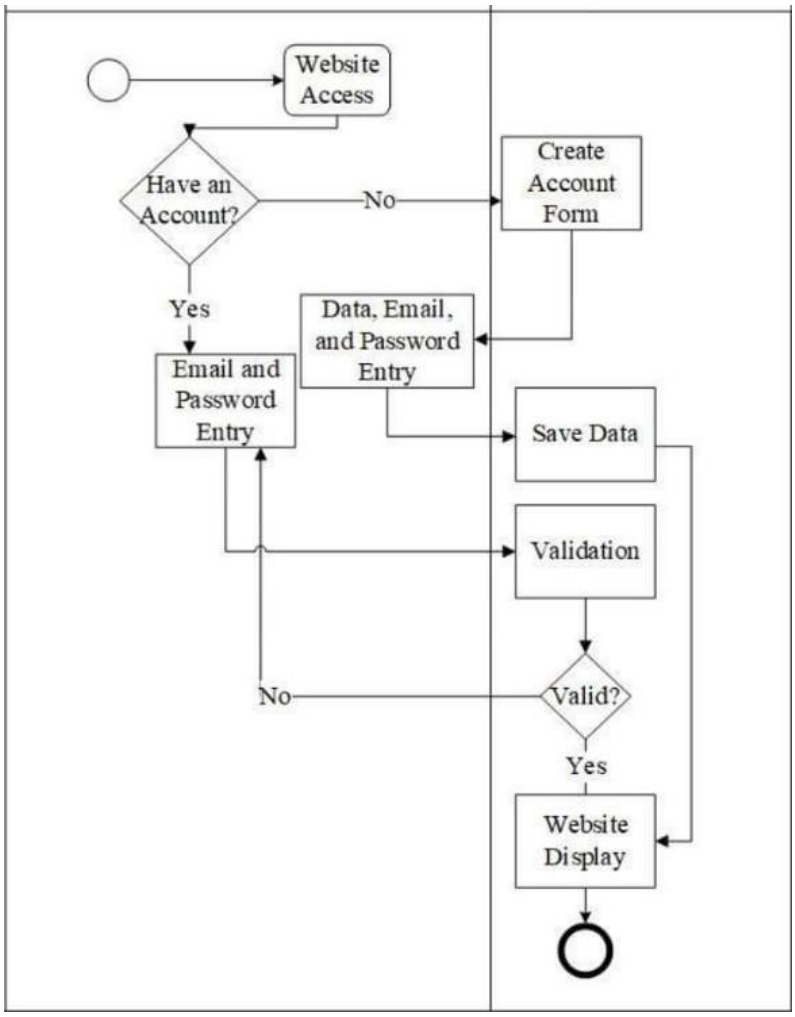
Group Members

Reg. No	Name	Signature
IT19029146	Eranda H.P.D	
IT19040936	S.M.Rathnayaka	

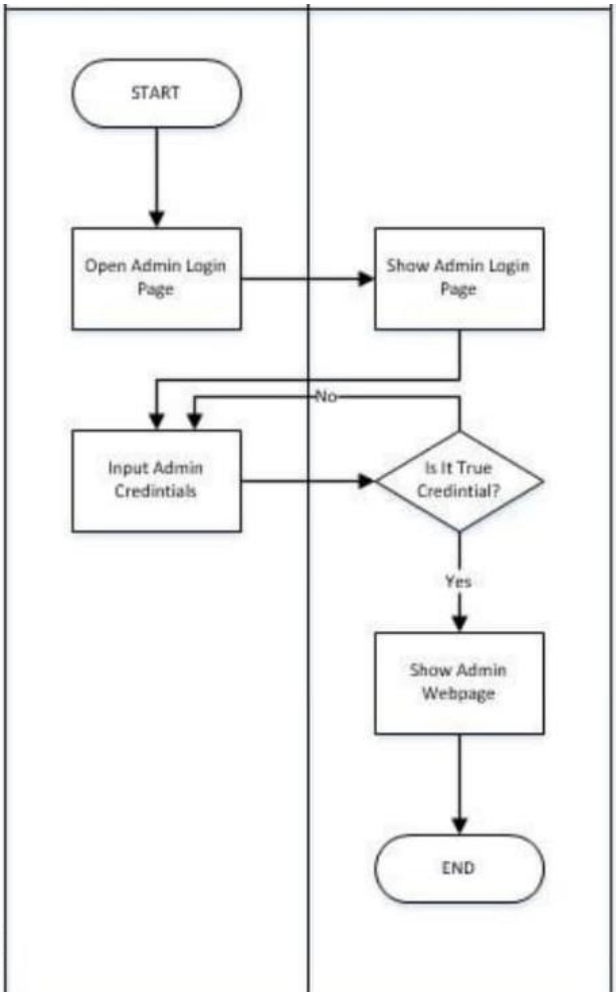
Table of Contents

List of Figures

User Login Activity Diagram **:-**



Admin Login Activity Diagram :-



1. Introduction

1.1 Problem Statement

In this CTF Box, our main purpose is to make the industry professionals aware about the current cyber security threats while providing them with unique entertainment. This particular CTF Box is mainly focused on the Healthcare Industry. So, as our vulnerable target system we are going to use an Electronic Medical Record (EMR) System called as “OpenEMR”. After building our CTF Box completely, we are planning on hosting our CTF Box in TryHackMe online gamified real-world labs platform. So that, any person in the world is allowed to play this CTF box. So, it opens up new pathways to gain knowledge on cyber-attacks & countermeasures while providing a unique entertaining experience.

Through this CTF Box we are mainly covering the current cyber security threats in the Healthcare Industry. So, we are planning on selling this CTF Box for various hospitals in Sri Lanka for a reasonable price. Our main target audience is all the medical staff that is responsible for the information & systems security of the hospital. This may include Database Security Teams, Network Security Teams etc. By doing so, we are hoping to raise awareness for the IT security engineers in all the healthcare networks about how to secure their systems against malicious cyber-attacks that may occur in the future. Our CTF Box consists with several categories of functions. In order to unlock the next level, a particular user has to solve the previous level successfully. If a person takes a particular challenge and finds the flag, he needs to submit it to the scoring system and get the points then move on to the next challenge. It is the only method of proceeding forward throughout this CTF Box. For an average user, it takes about 6 - 8 hours to complete all the levels.

1.2 Product Scope

In this particular CTF Box, we are going to use a group of programs that work together to support the execution of the application. This includes Operating Systems, Protocols, Runtime Environments, Databases, Function Calls, Architectural Layers, Installable Files, Patches, Coding Frameworks, Web Servers, Client Interface Tools etc. So, as our main target database, we have chosen a free & open-source Electronic Medical Record (EMR) System called as "OpenEMR". This particular system consists with many different functionalities including electronic billing, patient scheduling, e-prescribing, patient portal, clinical decision supporting system etc.[8] Furthermore mentioned, we have implemented this particular pre-built EMR system in a separate Linux virtual machine environment using Docker. So, in our CTF Box, what we are expecting from the CTF Players is to discover certain vulnerabilities in the OpenEMR System and exploit them in a certain way.[9] At the end of each & every successful exploitation for a particular level, they will be able to find a level-specific flag. If the players are able to identify all the flags successfully, then they get the opportunity to submit those flags in our custom-made web application. It is the only method that a player can proceed forward through this CTF Box. Additionally, when developing this particular CTF Box, we are planning on using various cyber security technologies & software as shown below.[5]

i. XAMPP

- *A Web application development environment including MySQL database, PHP, Apache HTTP server, PHPMyAdmin and SQLite Manager*

ii. Cryptool

- *Open-Source e-learning tool illustrating cryptographic and cryptanalytic concepts.*

iii. String to Hex Converter Online

- *An easy-to-use tool to convert Plain String data to Hexadecimal.*

iv. Steganography Online tool

- *A free online tool to encode a hidden message into an image.*

v. **Base64 Encoder & Decoder**

- *A free online tool to encode and decode data to & from Base64 format.*

vi. **QR Code Generator**

- *A free online tool to make our own QR Codes to redirect to a particular site.*

vii. **SAML Tracer**

- *A simple add-on for web browsers to collecting SAML Traces.*

viii. **Wireshark**

- *A network packet analyzer that is used to present the captured packet data in as much detail as possible.*

ix. **ROT13 Encoder & Decorder**

- *A free online tool to replace a letter with the letter 13 letters after it in the alphabet.* ***(A shift cipher mechanism)***

x. **Steghide**

- *A Linux based steganography tool which is used to hide various kinds of image & audio files using a passcode.*

xi. **Online Image Encryption tool** ***(To use CBC Method in Cryptography)***

- *A free online tool to make your images unrecognizable using the secret key.*

Furthermore, when building this CTF box, we have used the Jeopardy-style board with challenges worth different numbers of points. [3] We have decided to create & design this CTF Box up to maximum of 18 levels. Additionally, after the implementation of all the levels, we have decided to host our CTF Box in TryHackMe gamified online platform. In order to gain access for the CTF challenges, a person has to complete a challenge & submit the flag correctly.[4] Depending on the demand, we are hoping to sell our particular product for a

reasonable price. Additionally, after they are done playing the CTF, we are hoping to gather their feedback as well. So that it would come in handy when we improve our CTF Box further.

1.3 Project Report Structure

From this section onwards, for the ease of understanding we have divided our report into several sub-sections. After the Introduction, the first sub-section that we are going to explain you about is the "Methodology" section. In here, basically we are going to explain you about various development stages in our CTF Box project. Under the Methodology section, the 1st development stage that we are going to explain you is the "Requirement Analysis" phase. In this stage, mainly we gather the necessary & specific requirements for our CTF Box project. For the ease of explanation, in here, we are going to add certain diagrams, pictures and tables. After the "Requirement Analysis" stage, we are going to explain you about the "Design" stage. In this stage also, we are going to add certain diagrams and images that are relevant for our CTF Box project. After we are done with that stage, we are going to explain you about the "Implementation" stage. In here, we are going to include several source codes, algorithms and related programming & implementation languages. As the final stage we are going to explain you about the "Testing" stage. In here basically, we are going to describe you about the test results and the relevant proofs that ensures the success of our CTF Box project.

After we are finished with the "Methodology" section, we are going to explain you about the "Evaluation" stage. In here, we are going to explain you about the techniques that we have used in analyzing the data. Furthermore mentioned, we are also going to highlight about the failures of the final product and further improvements of it. Then we are going to explain you about the lessons that we have learnt from this project. At the end, we are going to use all our gathered experiences in order to provide with various suggestions for future development purposes.

Then as the last section, we are going to provide you with a brief conclusion about our entire CTF Box project. This section is going to be consist with the strong / positive factors, weak / negative factors and potential further improvements. At the end of this report, referencing will be provided according to the IEEE standard. As additional information, at the end of this report, we are going to demonstrate all our CTF Box challenges as a complete walkthrough.

2. Methodology

2.1 Requirements and Analysis

Functional Requirements

The main functional requirement that is associated with this feature is the secure submission of the flags appropriately. This requirement can be easily identified by using the tag “F1R1”. So, in order to achieve this functional requirement, we used a strong verification & validation process. So that none of the CTF players would be able to bypass this particular security mechanism and proceed to the next level without submitting the correct flag.

The second functional requirement that is associated with this feature is the input validation. This particular requirement can be easily identified by using the tag “F1R2”. So, in order to achieve this functional requirement, we implemented regular expressions in order to validate user inputs. So, if a particular user tries on performing SQL Injection attack, Buffer Overflow attack or XSS attack it can be easily mitigated by using those input validation mechanisms. Furthermore, we used to define a minimum and maximum value range check for numerical parameters and dates. So, if an user tries on performing any of those above mentioned attempts, he/ she will be locked out from his/ her account and will be blacklisted for a certain amount of time.

Non-Functional Requirements

Performance Requirements :-

Speed of the website, Response time of pages, CTF box and other operations are including Navigation and data operations are well organized. This system is always available. Users can access the system at any time using a web browser. System is easy to use, easily accessible, and always has other functional requirements. In the event of a failure, it may take less than an hour to reset the system and deliver it to the user. The user interface for the software is compatible to any browser. (Google chrome, Internet Explorer, Mozilla) .

- **Response time -** The system responds within 1 second.
- **Capacity -** system should support more than 500 users at a time
- **User Interface -** The user interface screen responds within 5 seconds.
- **Compatibility -** The system must comply with Microsoft Accessibility.

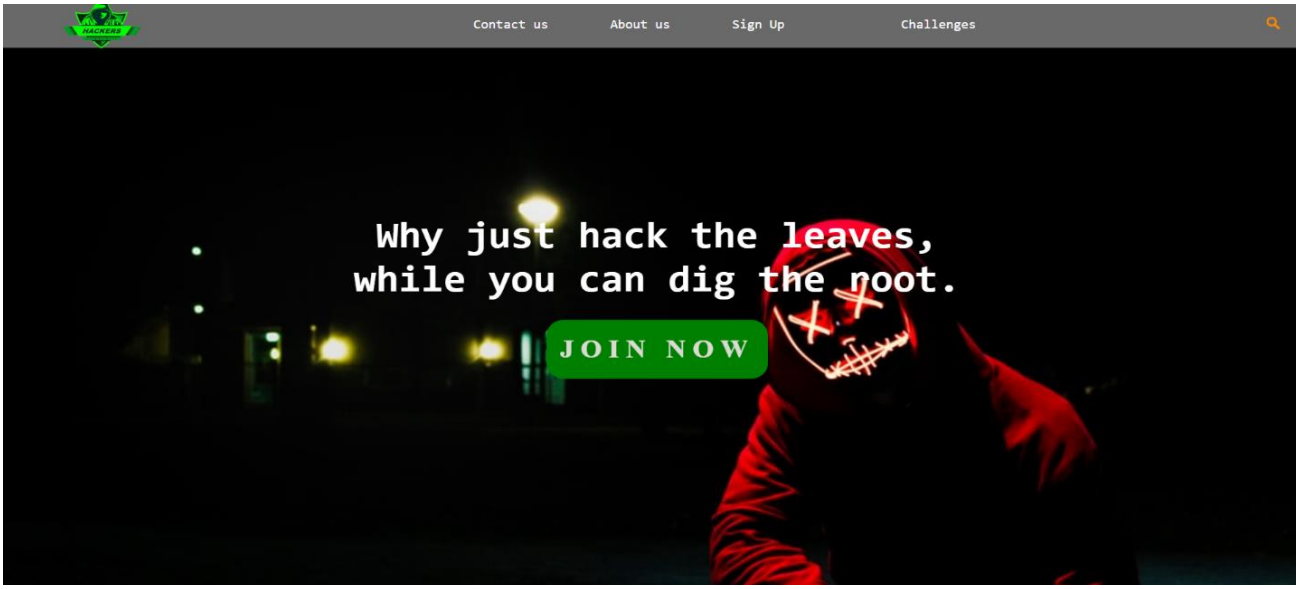
Safety Requirements :-

If a major failure, such as a disk crash, causes significant damage to a large portion of the database, the recovery system restores an earlier copy of the database that was backed up for archive storage and restores the status by reapplying or redoing committed transaction operations from the backed up log, up to the time of failure . Security Requirements Encryption mechanisms are used to protect the user's sensitive information, such as passwords and other user records. The system understands who is accessing the system because all administrative and data entry operators have unique access. Only system administrators can access the system, and no one else can gain unauthorized access. All communications with users across the system are well protected. Sensitive information such as user login credentials and CTF flags are encrypted and securely transferred.

External Interface Requirements

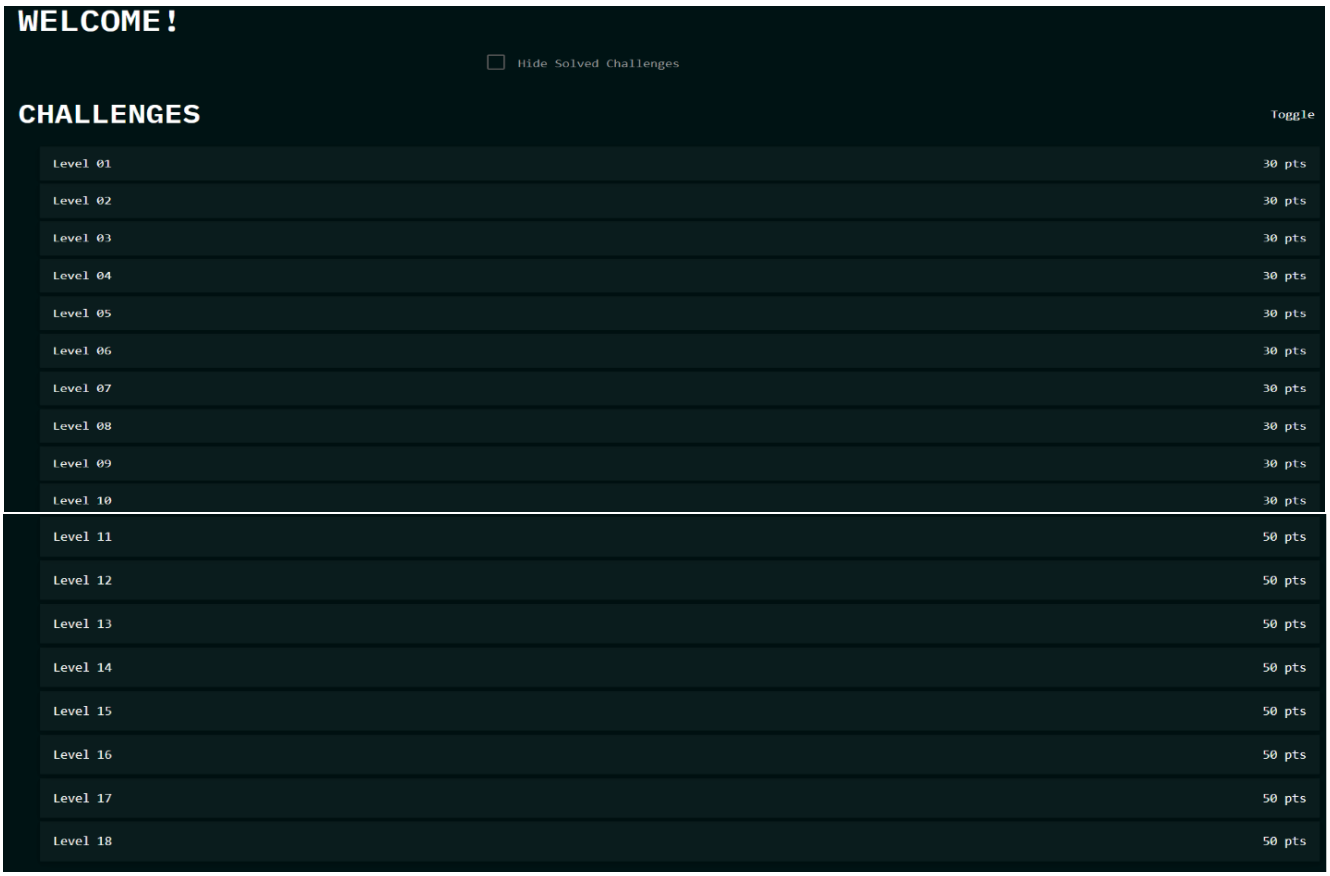
User Interfaces :-

The GUI of the web site is created using HTML and the backend is made up of PHP and MySQL which can be accessed by different users. The interface of this website is easy to use, and the interface has been developed in a user-friendly manner. The navigations and images of the website create an attractive environment for the user. Users can contact us with messages, emails or if their needs are not met. English will be the language of the content. Buttons and functions are used to help users access and proceed with the Website. Keyboard shortcuts are provided, which give users the ability to work successfully and quickly with challenges and provide various error messages to guide them towards any kind of error. All fonts used are user friendly and keep pages light, so it doesn't take long to load a tab. The instructions are created to give users an accurate idea of the CTF challenge.



Home Interface

Sign In Interface



Challenges Interface

Hardware Interfaces *:-*

Processor:- Intel

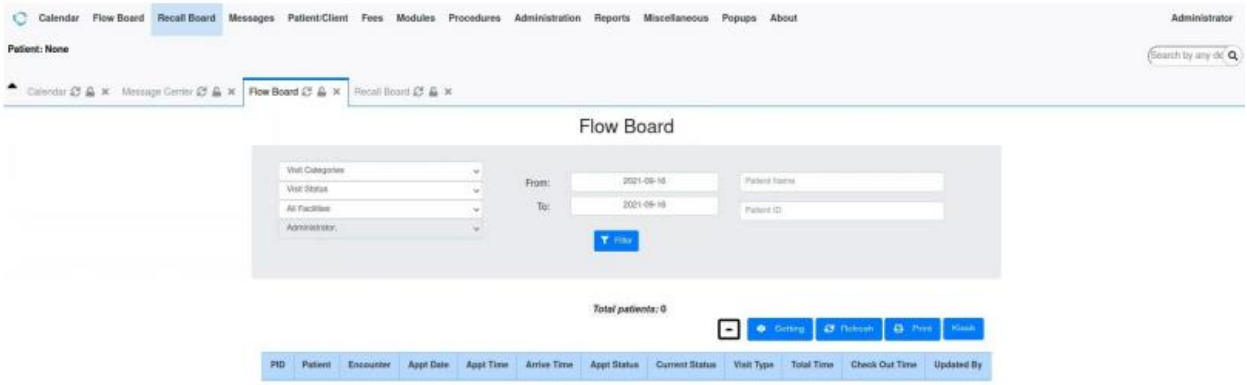
RAM:- 4GB or Higher

These requirements will help to load the system and the challenges effectively and the user can play the CTF Box without any system error or a delay. Since the program must be running over the Internet, all the hardware needed to connect to the Internet acts as the system hardware interface. For example, modem, WAN-LAN and Ethernet cross cables.

Software Interfaces *:-*

We have selected the Windows operating system for its excellent support and user friendliness and Linux Ubuntu operating system to implement CTF box. To store user records, we chosen XAMPP database. Uses HTML, PHP, Java Script, Python, Bootstrap CSS & jQuery for Front-end and backend.

- The system communicates with the configuration to identify all the components that make up the configuration.
- The system communicates with the database to store and validate data.
- The system communicates with the web browser to download files via the internet.
- The system communicates with all types of operating systems to download tools and to
- complete the CTF challenges.



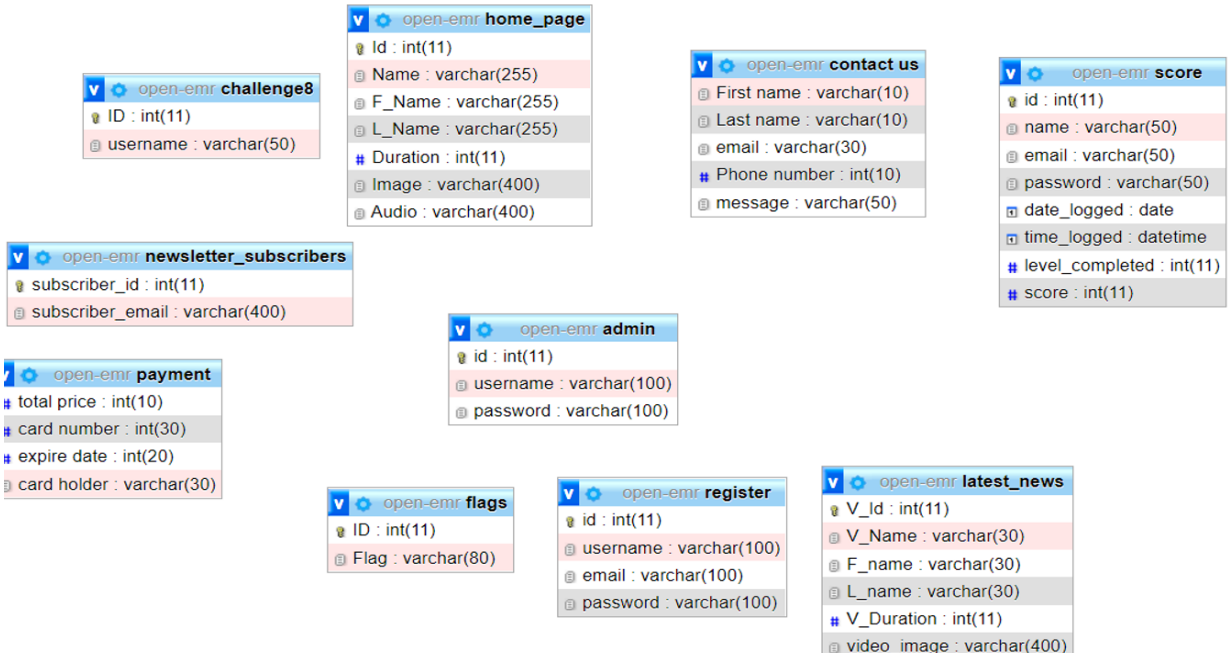
OpenEMR software Interface

Communications Interfaces **:-**

The system uses the HTTP protocol for communication over the Internet, and intranet communication over the Internet via the TCP / IP protocol suite. CTF box supports all types of web browsers.

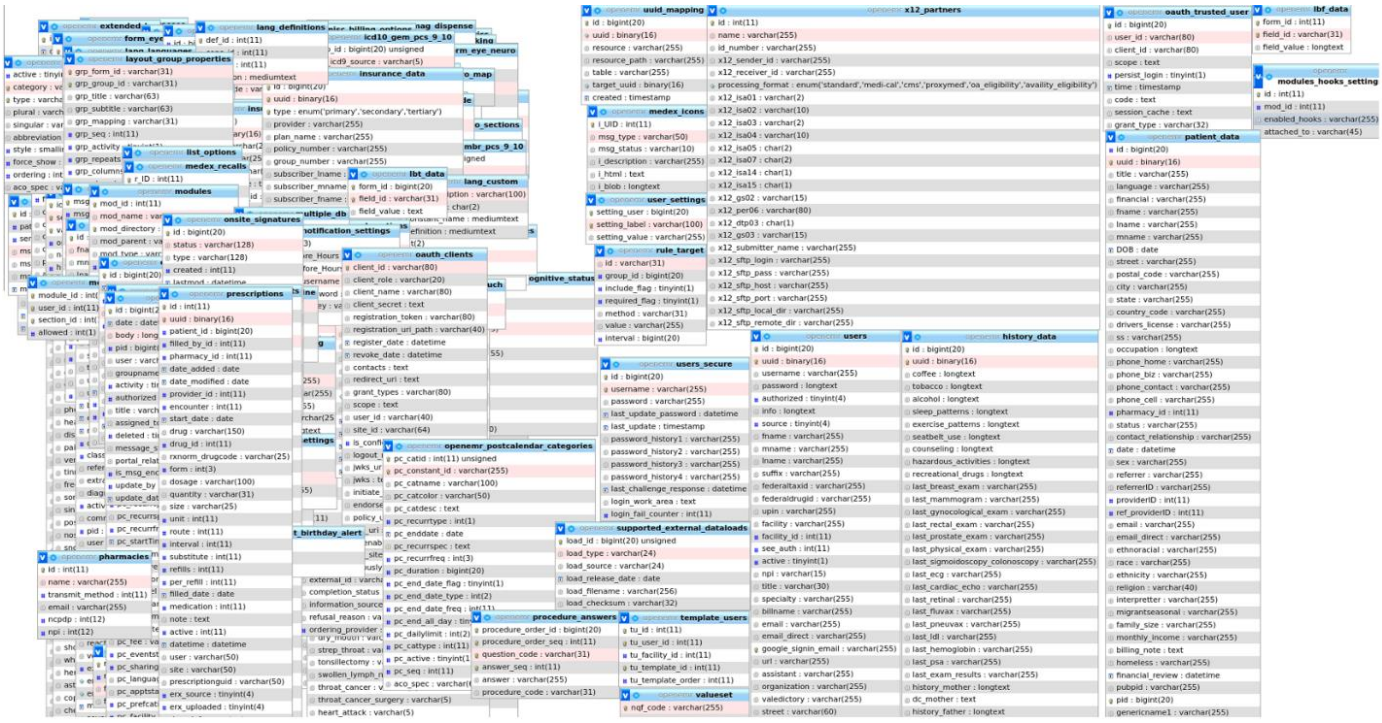
2.2 Design

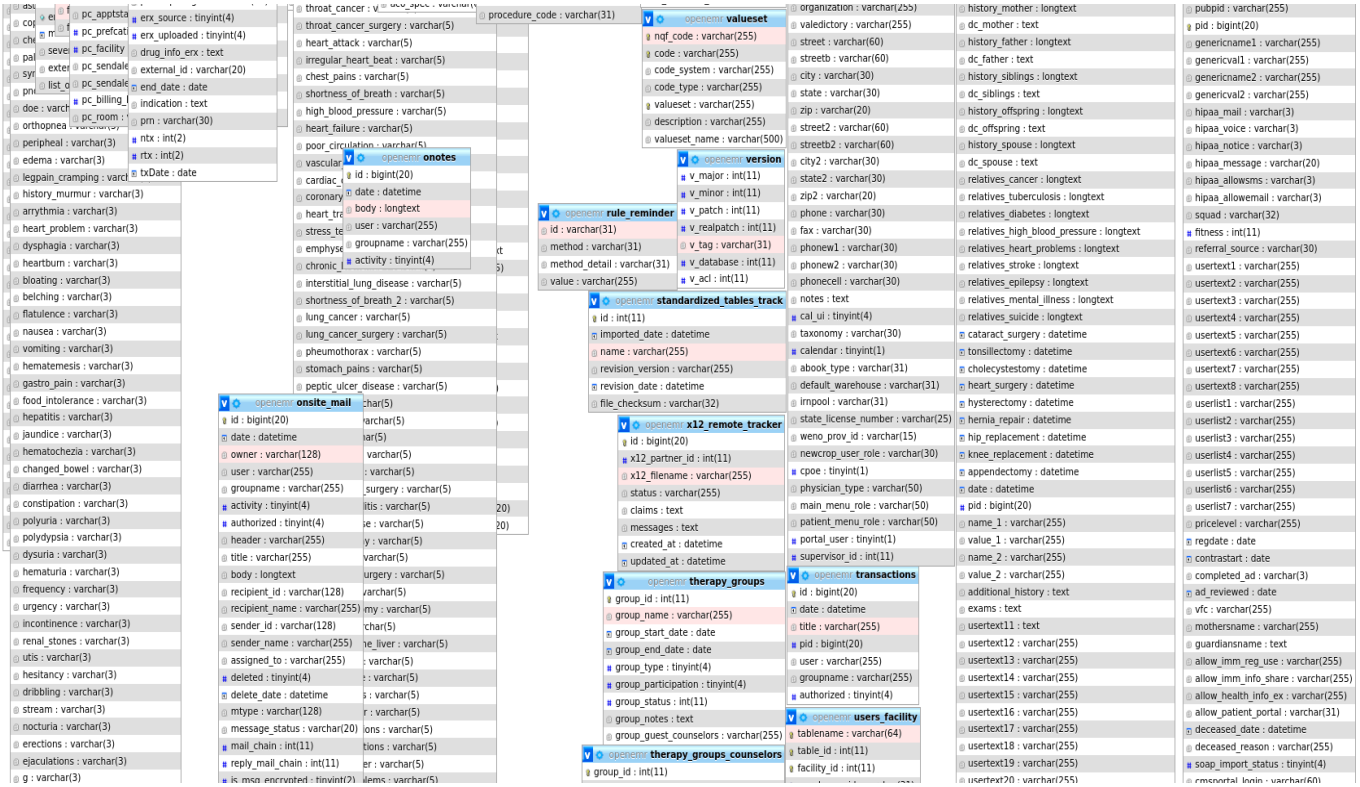
The website is designed with proper security features. Vulnerable web pages are designed to solve challenges and help users learn web risks. In our CTF Box web application, we are going to use normal HTTP Protocol since it's just a simple web application for submitting the flags. However, when it comes to the user authentication process, we followed the industry standard security protocols & necessary implementations. Below are the database tables of the website.

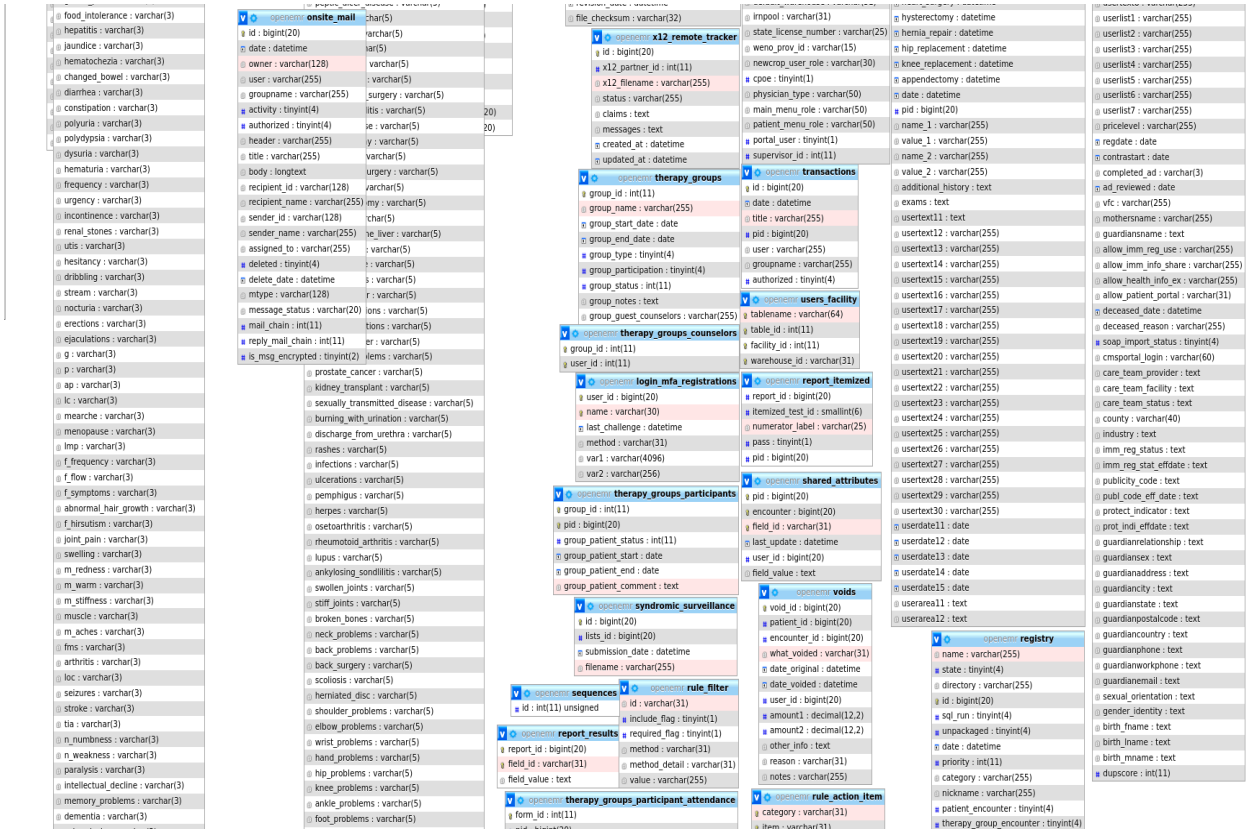


Website Database

When it comes to the system requirements *of* OpenEMR , *it* does not require that much of computing resources because of its flexibility due to the docker container. Below are the database tables of the OpenEMR.







2.3 Implementation

The database backend is implemented in PHP and the front end using HTML, CSS and JavaScript. Validation of users and CTF Flags are done by PHP. The scoreboard is primarily made using PHP, and the level and score are increased and displayed each time the user submits the flag. The Apache MySQL database is used to store data. Apache phpMyAdmin is a free PHP written software tool intended to manage the administration of a MySQL or MariaDB database server. phpMyAdmin can handle many administrative tasks, including creating a database, executing queries, and adding user accounts.

Sublime text is used for code editing. Sublime Text is a sophisticated text editor for code, markup and prose and also it is known for being lightweight, fast and responsive. Below are the PHP codes used for user and CTF flag validation.

```php
<?php

session_start();

$DATABASE_HOST = 'localhost';
$DATABASE_USER = 'root';
$DATABASE_PASS = '';
$DATABASE_NAME = 'open-emr';

$con = mysqli_connect($DATABASE_HOST, $DATABASE_USER, $DATABASE_PASS, $DATABASE_NAME);
$flag = 'crypto{f!N4llY_y0U_H4v3_gR4Nt3d_fUll_c0NtR0l_0v3R_Th3_5y5T3m}';

  if ($_POST['Flag1'] === $flag) {

    $query = ("UPDATE score SET level_completed = '1', score = '30'  WHERE score.name = '".$_SESSION["name"]."';");

   if($con->query($query) === true){

 echo '<script>document.location.href="score.php"</script>';

}

else {
  echo 'Not updated';
}
  }
  else {
    echo "<script>alert('ERROR:Incorect Flag')</script>";
  }
```

```php
<?php

require_once("configurations/dbconfig.php");

?>
<?php
session_start();

$DATABASE_HOST = 'localhost';
$DATABASE_USER = 'root';
$DATABASE_PASS = '';
$DATABASE_NAME = 'open-emr';

$con = mysqli_connect($DATABASE_HOST, $DATABASE_USER, $DATABASE_PASS, $DATABASE_NAME);

if ($stmt = $con->prepare('SELECT id, password FROM score WHERE name = ?')) {

  $stmt->bind_param('s', $_POST['username']);
  $stmt->execute();

  $stmt->store_result();
}
$stmt->store_result();
if ($stmt->num_rows > 0) {
  $stmt->bind_result($id, $password);
  $stmt->fetch();

  if ($_POST['password'] === $password) {
    session_regenerate_id();
    $_SESSION['loggedin'] = TRUE;
    $_SESSION['name'] = $_POST['username'];
    $_SESSION['id'] = $id;
    header('Location:index.php');
  }
  else {
    echo "<script>alert('ERROR:Incorect password')</script>";
  }
}
$stmt->close();

?>
```

2.4 Testing

The program has tested performance results through some test steps. The test was performed as a final step in the development of the site. The following tests were performed to verify the performance of the website.

- **Website functionality testing.**
- **Forms testing for all pages**
- **The input data validity**
- **Allowed values for the data field**
- **Invalid input values for the data field**
- **Cookies testing**
- **HTML/CSS validation**

***Content Testing checklist* :-**

- **No grammar or spelling errors**
- **The pictures are kept in the right sizes**
- **Website color scheme and font size**
- **Contents are informative, understandable, structured and logically relevant**
- **The instructions are clear and contain accurate information**

3. Evaluation

3.1 Assessment of the Project results

Qualitative Analysis **:-**

Here we did the analysis mainly by answering questions like 'why', 'what' or 'how'. Each of these questions is addressed through a questionnaire. This data analysis also helped us to make decisions and find optimal solutions to the problems we face.

During the assessment of the project results we found,

- **What went right**
- **What went wrong**
- **What needs to be improved**

3.2 Lessons Learned

We have identified weak areas and challenges in implementing the website which will help us in our future improvements. We also learned to manage time with our project work (time management) and it is easy to find more faults by inspecting the parts before the whole. We learnt that during the evaluation process.

3.3 Future Work

We hoping to get the feedback from the players in order to improve our system by adding more security features and also hope to develop all levels to give the user a better challenge. We will find optimal solutions to the problems we have identified so that we can improve the quality of the website. It will help us to sell our CTF box to the healthcare sector to educate health workers about security vulnerabilities.

4. Conclusion

This CTF box is primarily designed to focus on the health sector, but any user can play this CTF box to improve their knowledge of web and shell related vulnerabilities and to learn about the importance of cyber security. This CTF box contains 18 challenges covering many popular webs and shell-based vulnerabilities. The main goal is to find the admin username and password for OpenEMR. Once the user finds the credentials, the patient's data is revealed, which means that the data has been breached.

We have encountered various problems in implementing the challenges and solved the problems with optimal solutions. While developing the website, we learned various new techniques that we can apply to develop and maintain the website. We also learned about security gaps and the mitigation that can apply to them.

5. References

[1] "/app.assembla.com," assembla, 2010. [Online]. Available: https://app.assembla.com/wiki/show/csci4200-group3/External_interface_requirements

[2] R. Bandakkanavar, "krazytech.com," 2018. [Online]. Available: https://krazytech.com/projects/sample-software-requirements-specificationsrs-report-airline-database

[3] "CTF Design Guidelines - Google Docs." Google Docs, https://docs.google.com/document/d/1QBhColOjT8vVeyQxM1qNE-pczqeNSJiWOEiZQF2SSh8/preview Accessed 08 September 2021.

[4] Researcher, Adam Schaal, Principal Application Security. "Tips and Tactics for Creating Your Own Capture-the-Flag Event." Contrast Security | Secure Software Faster, https://www.contrastsecurity.com/security-influencers/tips-tactics-ctf-event Accessed 08 September 2021.

[5] "Tools and Resources to Prepare for a Hacker CTF Competition or Challenge - Infosec Resources." Infosec Resources, https://resources.infosecinstitute.com/topic/tools-of-trade-and-resources-to-prepare-in-a-hacker-ctf-competition-or-challenge/ Accessed 08 September 2021.

[6] The Hackers Meetup. "Beginner's Guide to Capture the Flag (CTF) | by The Hackers Meetup | Medium." Medium, Medium, 23 Sept. 2020, https://thehackersmeetup.medium.com/beginners-guide-to-capture-the-flag-ctf-71a1cbd9d27c

[7] csivitu. “GitHub - Csivitu/Ctf-Challenges: An Aggregation of CTF Challenges and Write-Ups for Csictf 2020!” GitHub, https://github.com/csivitu/ctf-challenges Accessed 08 September 2021.

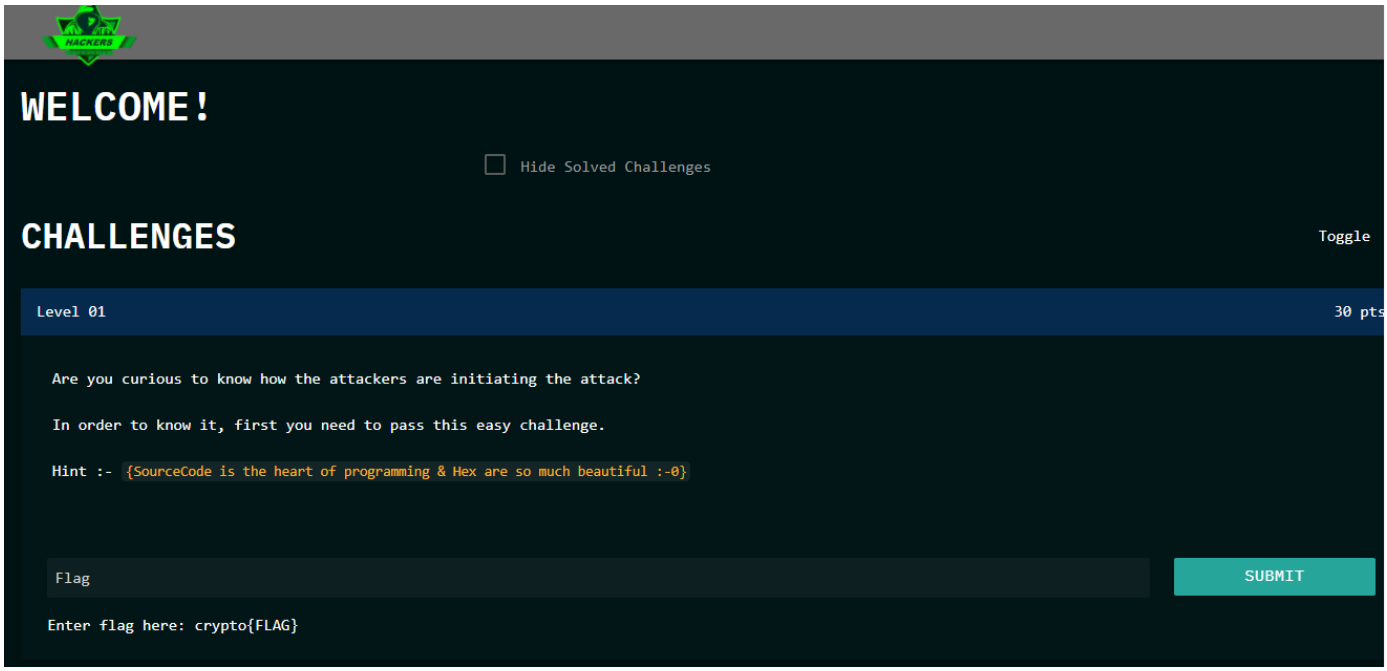
[8] “OpenEMR | Electronic Medical Records (EHR) Software | 2021 Reviews,” *Software Connect*. https://softwareconnect.com/ehr/openemr/ (accessed Sep. 10, 2021).

[9] “Open-emr Openemr : List of security vulnerabilities,” www.cvedetails.com. https://www.cvedetails.com/vulnerability-list/vendor_id-12269/product_id-23156/Open-emr-Openemr.html (accessed Sep. 10, 2021).

[10] “OpenEMR Installation Guides - OpenEMR Project Wiki,” www.open-emr.org. https://www.open-emr.org/wiki/index.php/OpenEMR_Installation_Guides (accessed Sep. 10, 2021).

Appendix A: Test Results

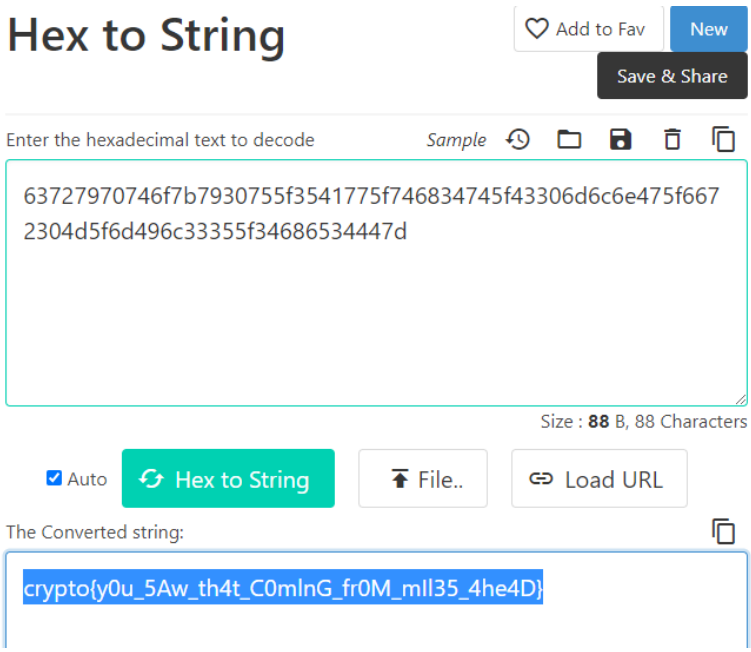
1)Level 1 :-



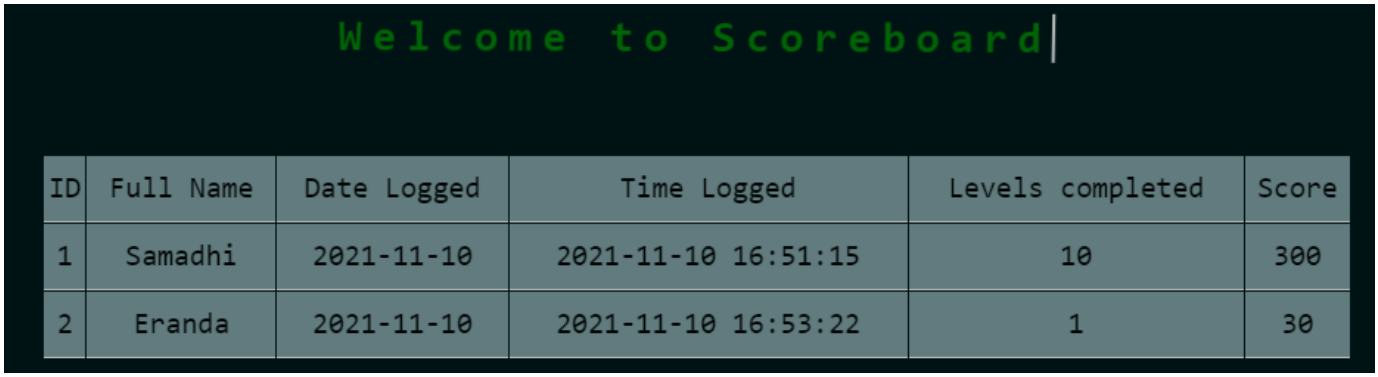
- As you can see in the above image, in order to solve the 1st challenge, first you need to carefully analyze the source code. So, let's go ahead and perform it.

```
<li class="challenge" data-category="58" data-stage="challenges">
    <div id="header-fflags" data-challenge="fflags" class="collapsible-header">

        <i id="check-fflags" class="fas fa-star gold-text tooltipped" style="display: none;" data-tooltip="You have solved this challenge" data-tooltip-id="e5cec386-a110-67a2-e890-661636f5b919"></i>
        <i id="uncheck-fflags" class="far fa-star grey-text tooltipped" data-tooltip="You have not yet solved this challenge" data-tooltip-id="53d55652-4e7e-5e27-c6a3-34ce76d961e1"></i>

        <div class="challenge-text truncate">Level 01</div>
        <span class="right">
            30 pts
            <span class="mobileSolves">



            </span>
    </div>
    <div class="collapsible-body" style="display: none;">
        <p>
            Are you curious to know how the attackers are initiating the attack?<br><br>In order to know it, first you need to pass this easy challenge.<br><br>Hint :- <code>{SourceCode is the heart of program

        </p>


        <!-- flaG iS 63727970746f7b7930755f3541775f746834745f43306d6c6e475f6672304d5f6d496c33355f34686534447d -->
        <br>

         <form class="flag-form" action="flag.php" data-challenge="nc-intro-2" data-points="5" method="POST" id="flagform-nc-intro-2">
            <div class="row no-bot">
                <div class="col s12 m10 flag-submit-line">
                    <div class="input-field">
                        <input type="text" name="Flag" placeholder="Flag" required>

                            Enter flag here: crypto{FLAG}

                    </div>
                </div>
                <div class="col s12 m2">
                    <button class="btn waves-effect waves-light flag-submit" type="submit" name = "submit"><div class="loader">Loading</div>Submit</button>
                </div>
                <input name="_csrf_token" type="hidden" value="tt16xziavuua9nu3inbgyj6r6i318rybgeaeh1gwz0c5691b9rhyoaqzlbjaf1xg">
            </div>
        </form>
        <br>
```

- As you can see in the above image, a flag value for this level is hidden inside the source code as a comment. However, it is provided in hexadecimal format. So before submitting the correct flag, first you need to convert the flag value to a string value.

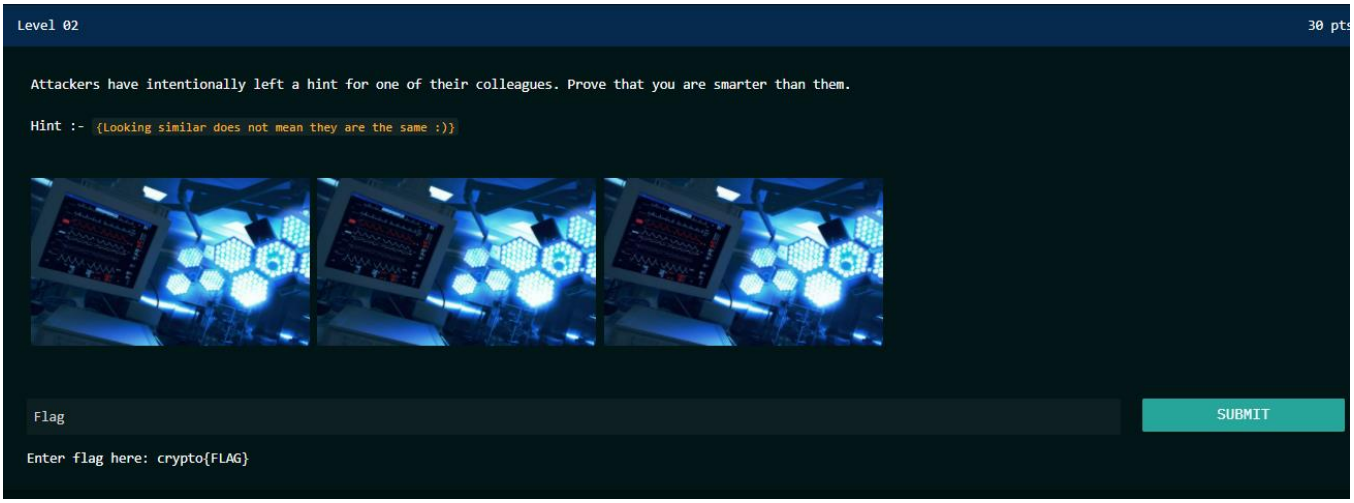


- So, when you submit the correct flag which is in the “string” format, the system accepts your answer is correct and your marks will be automatically incremented in the scoreboard.



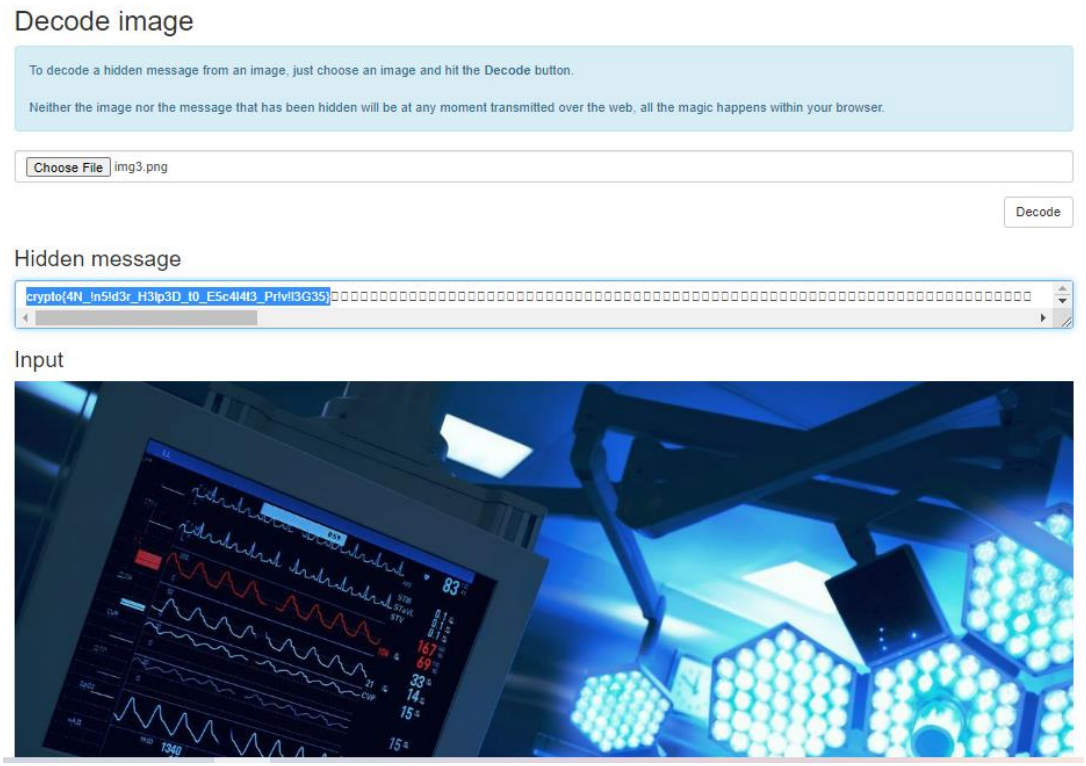
ID	Full Name	Date Logged	Time Logged	Levels completed	Score
1	Samadhi	2021-11-10	2021-11-10 16:51:15	10	300
2	Eranda	2021-11-10	2021-11-10 16:53:22	1	30

2)Level 2 :-

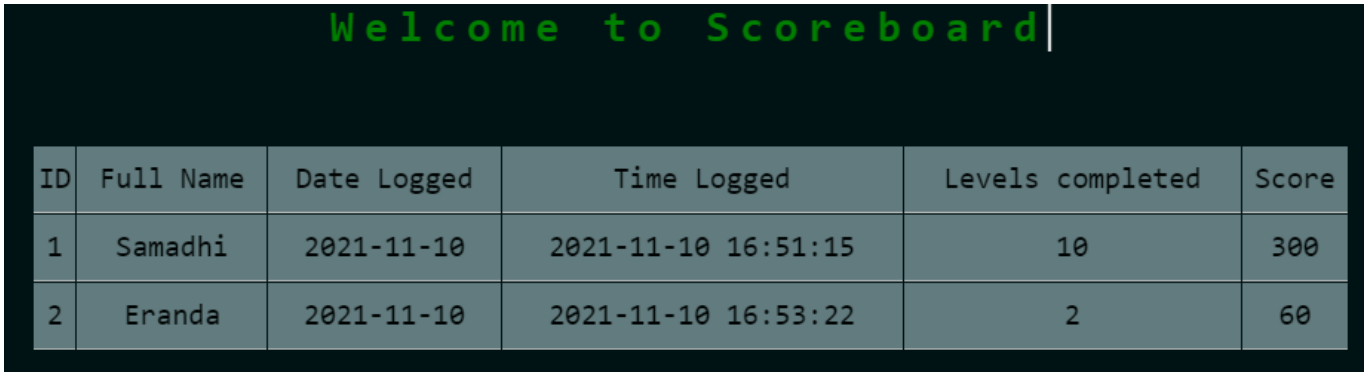


- As you can see in the above image, this level is completely based on steganography. In order to solve this challenge, first you need to download & save those 3 images in your computer. After that you need to upload those 3 images one after another to an online steganography tool, in order to find out the hidden secret message.

- However, when you upload the first 2 images and check them, you won't be able to find out any useful information as a secret message. But when you upload the 3rd image, in here you are able to successfully identify the hidden secret message inside the image. So, that would be the correct flag for the challenge 2.

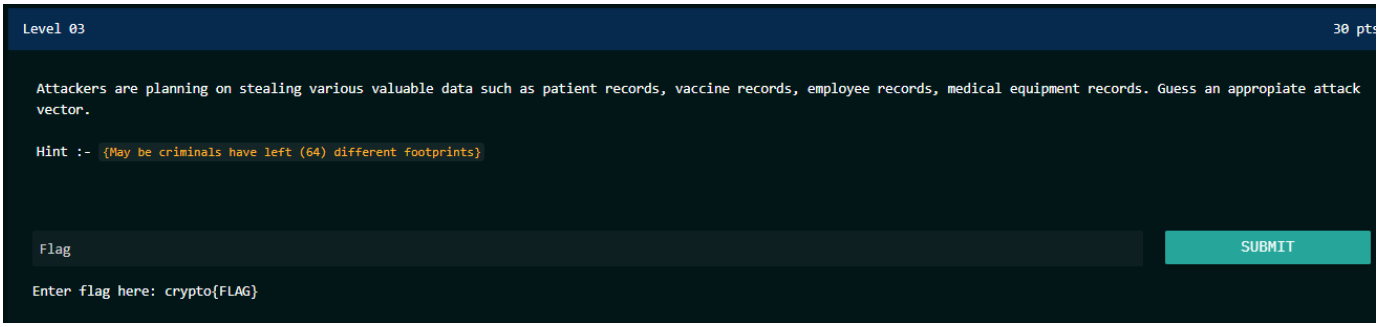


- Since, we have successfully found out the correct flag for the challenge 2, let's go ahead and submit it in the system.



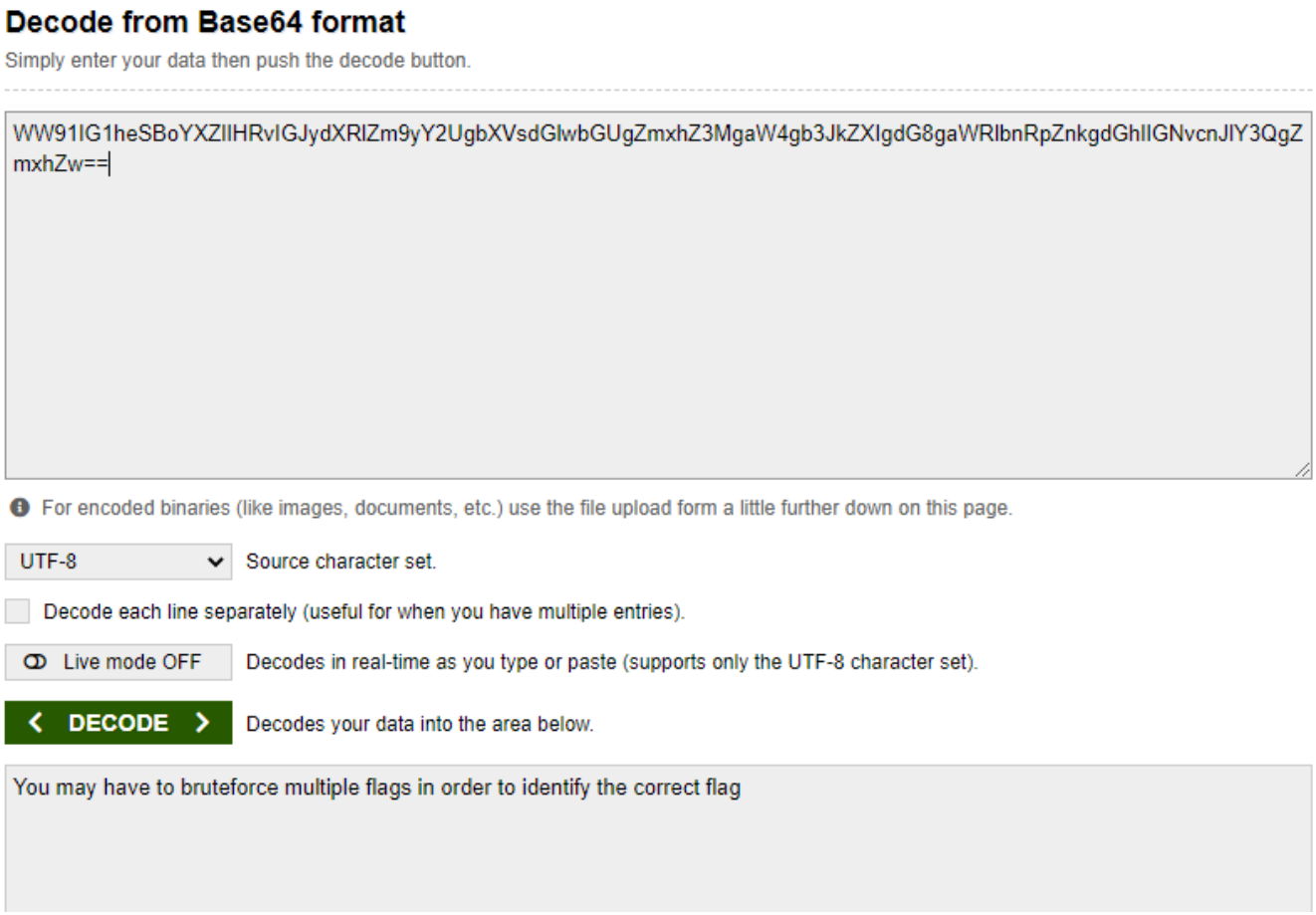
ID	Full Name	Date Logged	Time Logged	Levels completed	Score
1	Samadhi	2021-11-10	2021-11-10 16:51:15	10	300
2	Eranda	2021-11-10	2021-11-10 16:53:22	2	60

3)Level 3 :-



- When it comes to the level 3, this is actually somewhat challenging. So, in order to solve this level, as the very 1st step, you need to carefully analyze the source code.

- So, when you give your attention towards the comments section, you are able to find something called as “Hint”. But as you can see, all those lines are human unreadable. The reason behind it is that, all those lines are encoded with Base64 encoder. So, in order to understand those lines, 1st you need to successfully decode those lines.



- So, now we have successfully decoded the 1st line of the comment which is the “Hint”. As you can see in the above image, it says that you may get multiple flags after decoding all the lines. So, let’s go ahead and decode all of those lines first.

Decode from Base64 format

Simply enter your data then push the decode button.

JEw0ZzEgPSBXMzRLXzByXzV0MGwzTl9DcjNkM050ITRsNQ==

For encoded binaries (like images, documents, etc.) use the file upload form a little further down on this page.

UTF-8 Source character set.

Decode each line separately (useful for when you have multiple entries).

Live mode OFF Decodes in real-time as you type or paste (supports only the UTF-8 character set).

DECODE Decodes your data into the area below.

$L4g1 = W34K_0r_5t0l3N_Cr3d3Ntl4l5

Decode from Base64 format

Simply enter your data then push the decode button.

JGw0ZzIgPSByNE41MG0zdzRyMw==

For encoded binaries (like images, documents, etc.) use the file upload form a little further down on this page.

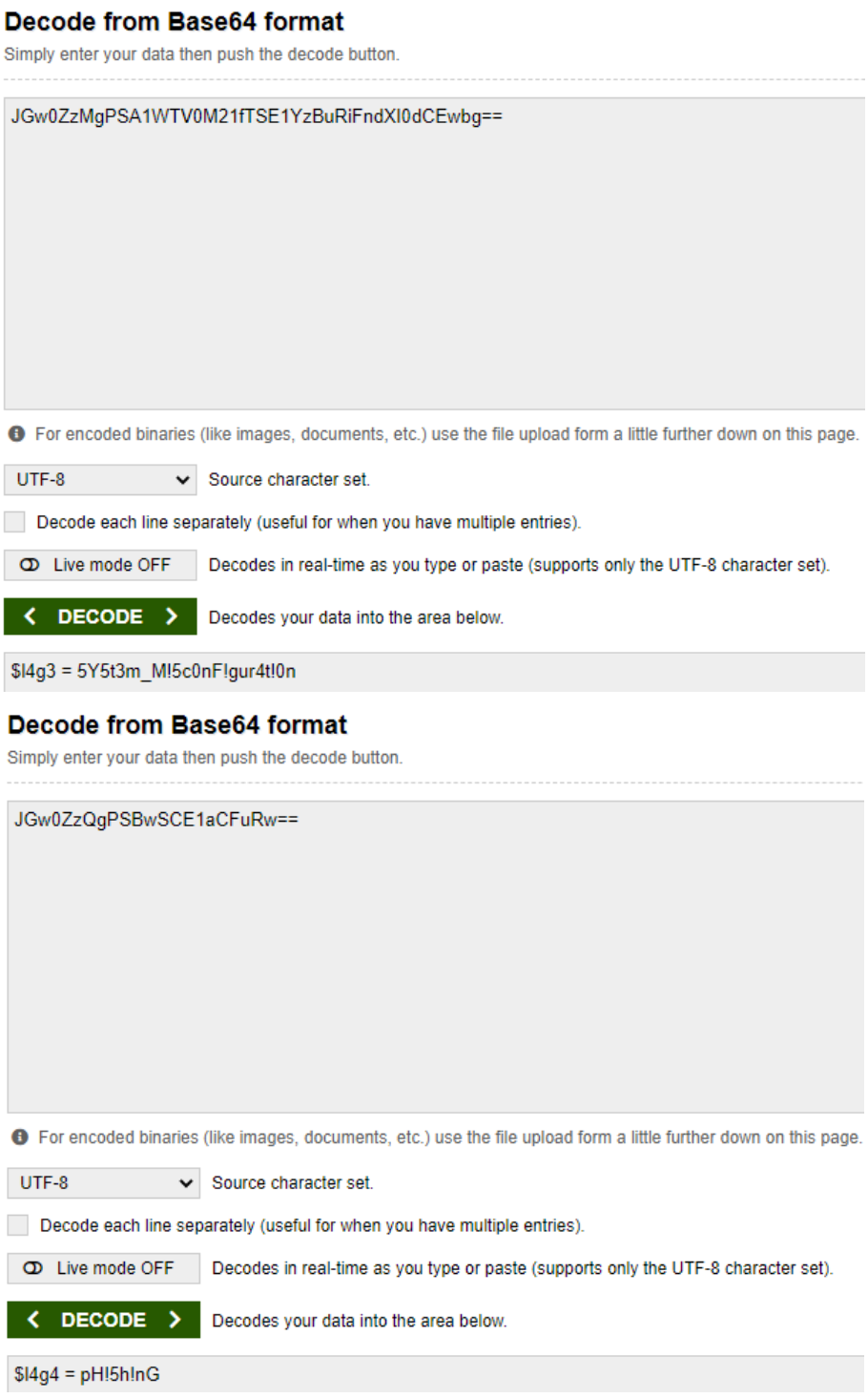
UTF-8 Source character set.

Decode each line separately (useful for when you have multiple entries).

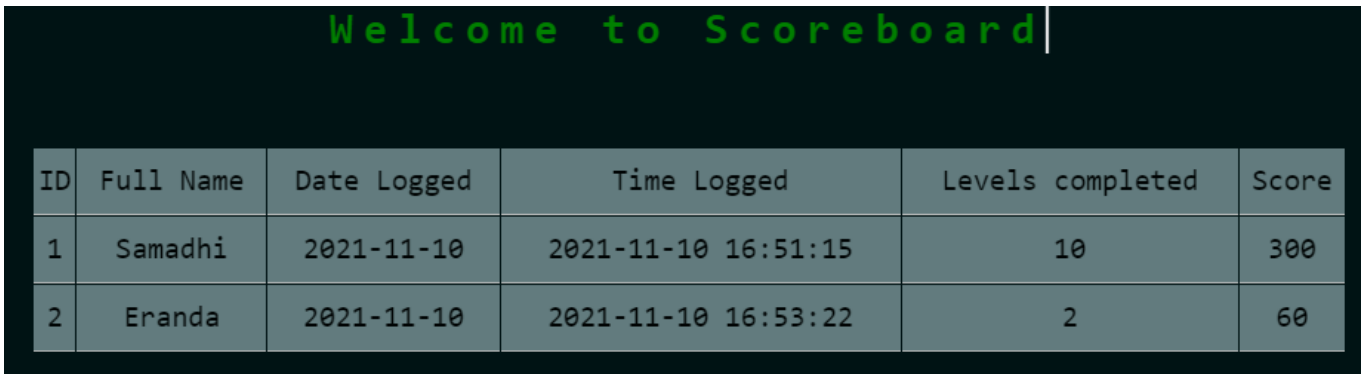
Live mode OFF Decodes in real-time as you type or paste (supports only the UTF-8 character set).

DECODE Decodes your data into the area below.

$l4g2 = r4N50m3w4r3



- Since now we have got 4 different flag values, in order to find out the exactly correct flag, you may have to perform some brute force attempts as well. So, when you try out all the different 4 flag values, you are able to identify the exactly correct flag as shown in the following image.



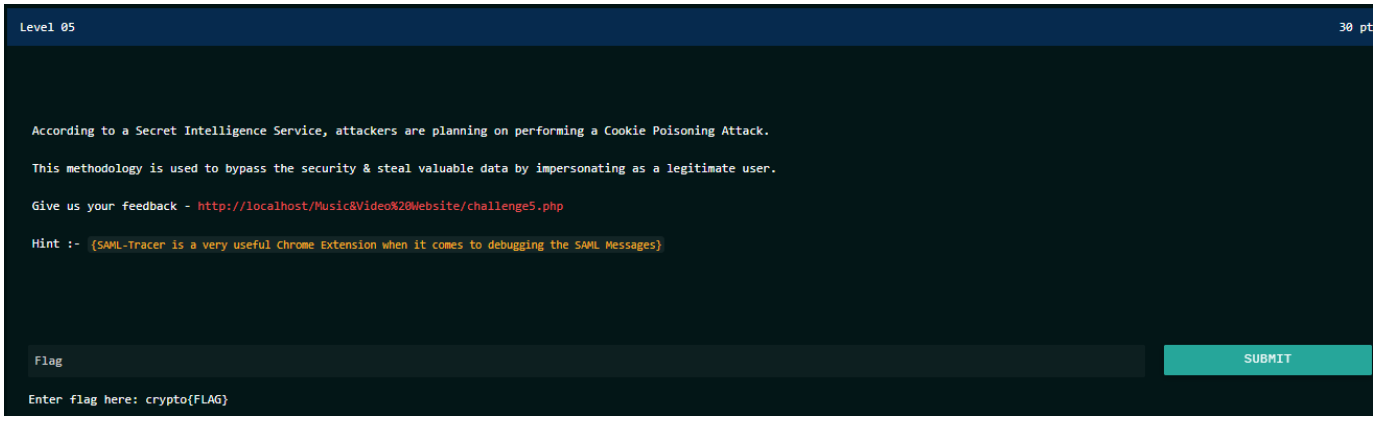
ID	Full Name	Date Logged	Time Logged	Levels completed	Score
1	Samadhi	2021-11-10	2021-11-10 16:51:15	10	300
2	Eranda	2021-11-10	2021-11-10 16:53:22	2	60

4)Level 4 :-

There is a QR Code to be scanned. After he scanned it you are able to find the flag.

FLAG:- crypto{y0ur_DaTa_baSe_is==_OpEn_EmR}

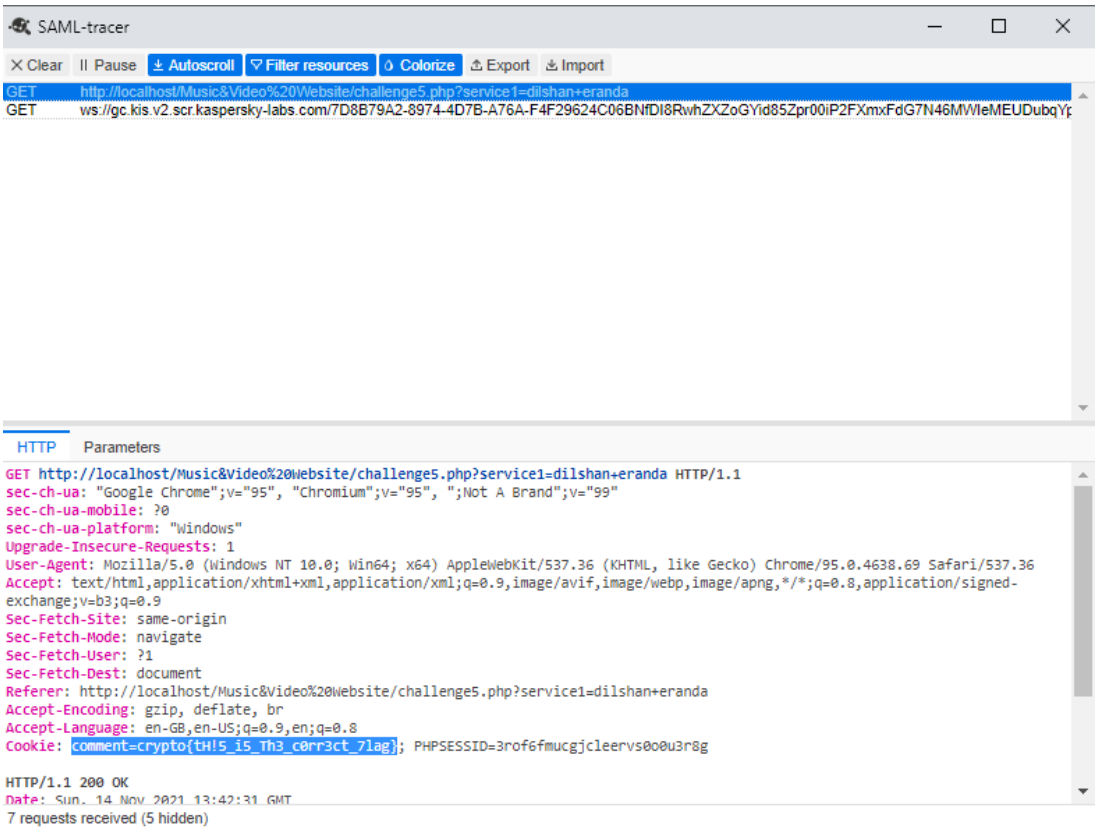
5)Level 5 :-



- Before starting to attempt the challenge 5, 1st you need to install the **"SAML Tracer"** Plugin to your web browser. In here, we are going to use the SAML Tracer in order to capture the site cookie that might become useful in finding the correct flag.

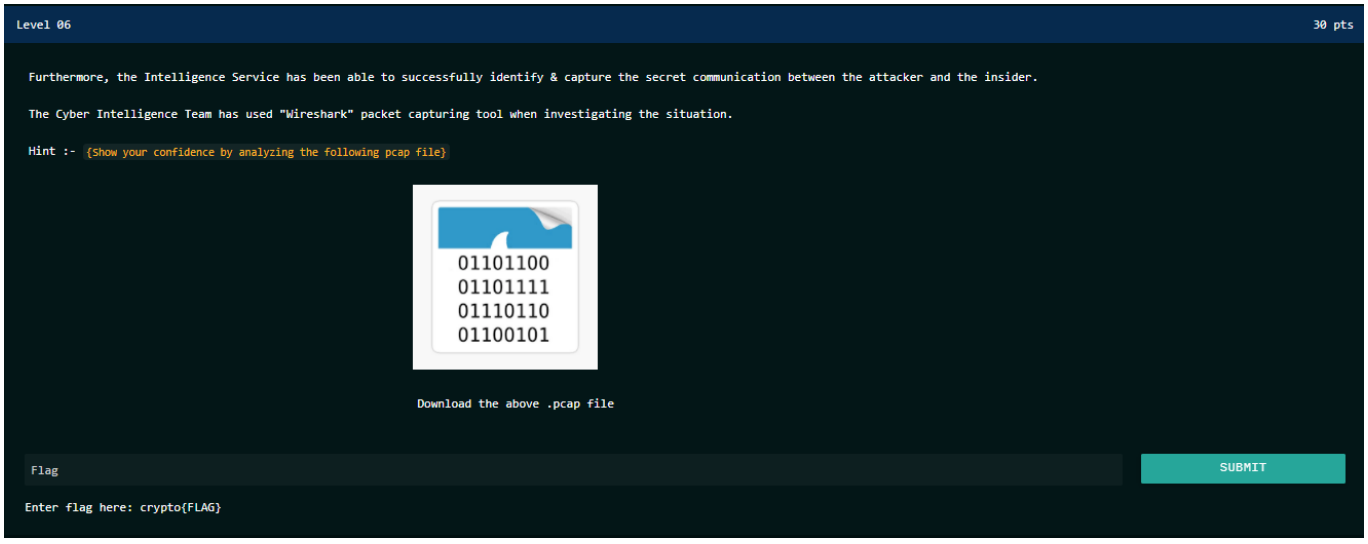
- After you have successfully added that plugin to your web browser, then you need to click on the "red-color link". Then it will popup a text box for you to write anything in it.

- After typing something inside the text box, you need to click on the "submit" button. Then soon after you need to open the SAML Tracer plugin as well.



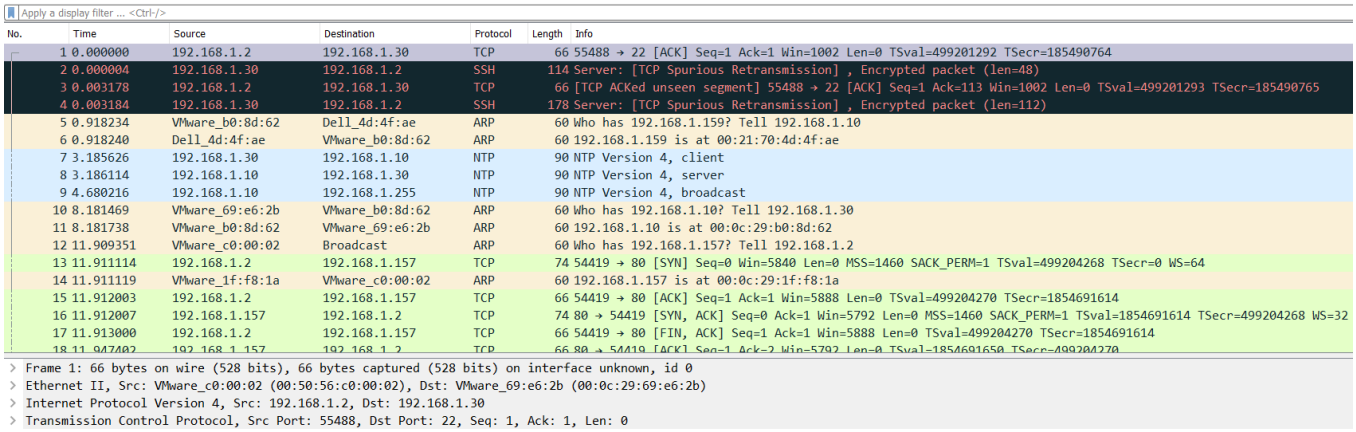
- So, as you can see in the above image, the correct flag for the level 5 is hidden inside the cookie value.

6)Level 6 :-

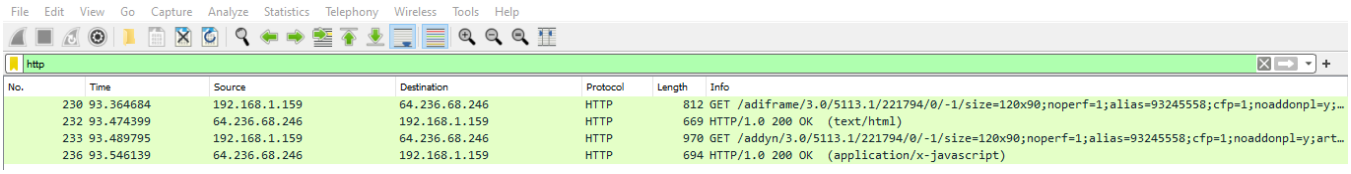


- In this level, it has provided us with a .pcap file to be downloaded. So, you need to download this .pcap file and then open it using the Wireshark packet capturing tool.

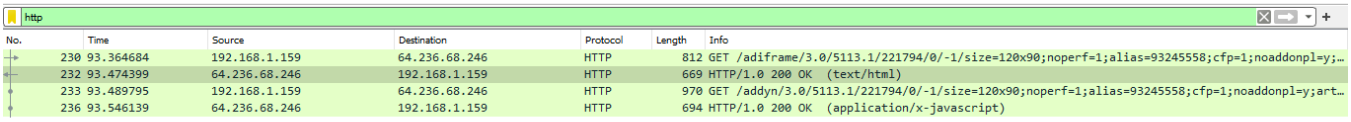
- So, when you analyze this .pcap file using the Wireshark, you will be able to see lot of captured packets.

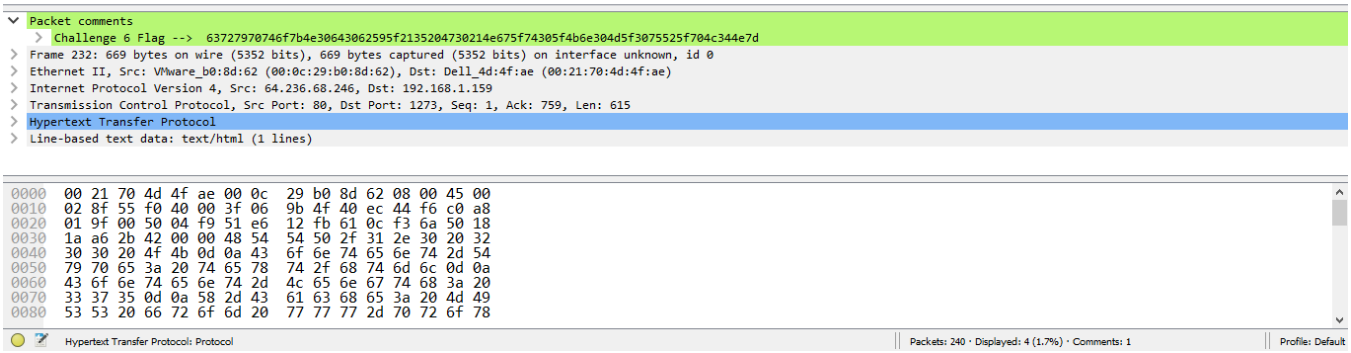


- However, in most of the times, human-readable and meaningful data are mostly in http packets. So, you need to use Wireshark to filter out HTTP packets.

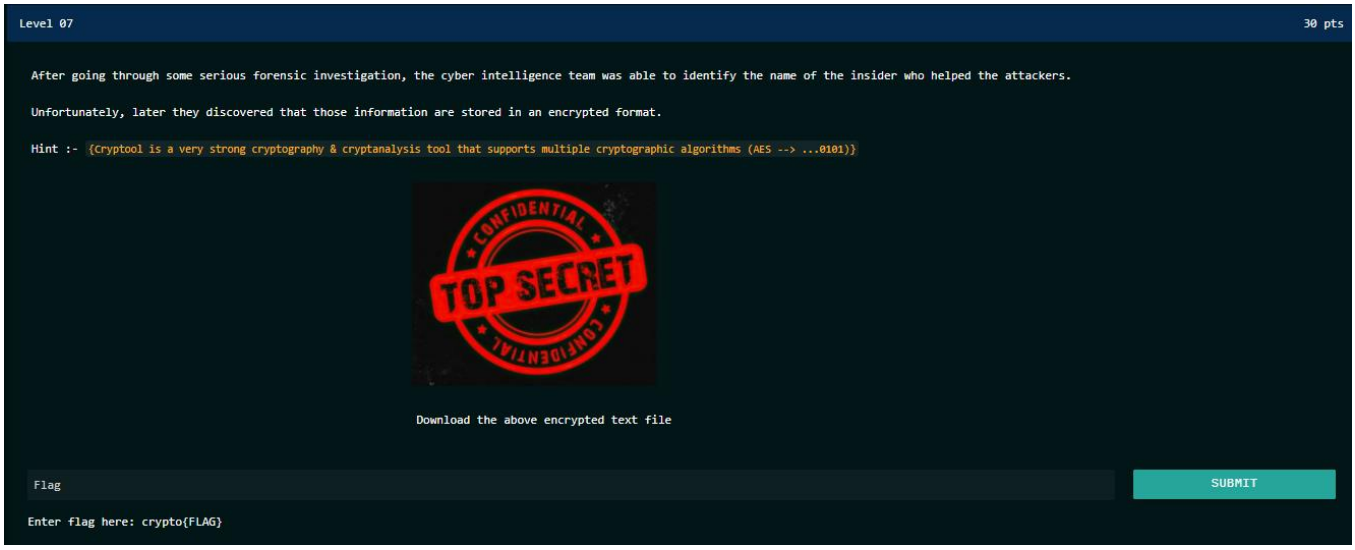


In here, when you click on the 2nd packet, it highlights some useful information in green color. So, when you click on it, you will be able to discover the correct flag for the level 6.





7)Level 7 :-



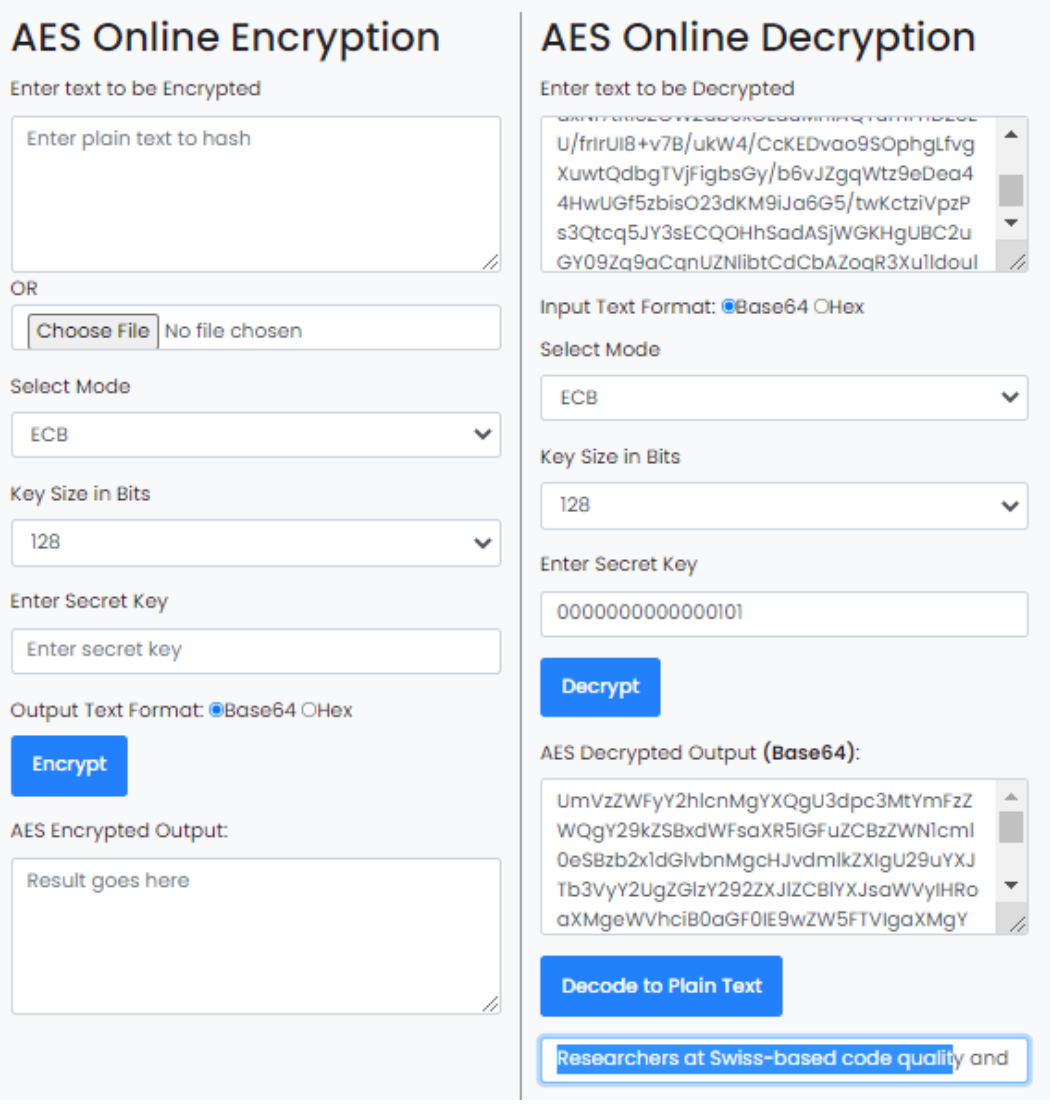
- In this level, it tells us to download the encrypted text file that is provided for us. So, let's go ahead and download that text file.

142iBOtv4OC8QHQKCfzQpgxU8H/w9wuJPThgkQewEe1DKXGWkY9XhY1pwziUaO2RqSbcj7nUkR5kYVuZYkeCVZMZtipEYX4qRWIEosvYOy68ObFg4fmsDqch+jWzQryPnAaalx4S/tQ5wcQVIvM6DMOtkdoxh86Z3m3qFOlc3RDEyg0BGxnr02SGxfZUK9yoXmLuviem4vfZu0V6HzLIVV0eEsA4q7OIycqf5LP+ciFu
vpD+Gl20PyXjRpWgG/hDXpyG5V/x269Ay4RX0W6C4LUzcUFQS+pkTVBlZ4IO2ZHGRpzCmZOLsrD0dHulnxkSQnrL18Ts1Znze40ieMIQE8CF4mFMNRwuNxmo0Mc9LqNRS8gIUh0x5PRKVSlA8qt/31FLWEkd11Ei7FbY7x0V+TwJbIMwQtnsN63u+o05v0bqz9g12p8BZzPrX2qDthcsNmg9Li8umwG4CrWG3gxiYB4e
teHKvKGrDBa1Msd3U9zEPKlz8tdT5/icW0Iu6Tcv4tV2+1YkpBtbO/rd+Mlt7C+LWTgl2TNmWFgheuneNRNvq14zYgEVcezhnBOT7PYxJDHeKFJc3W+6xGKLBnjpOUtBtLpbmromfJtc+pr6m/CjGzI1gQoRC8U8RZ1d70pjjxNNS95KlsCwbG6S5BLAFpYmD/TKv4PwXhgkCk4/u/4KsbfQDV30BttiyZE5GQtJXqLi
e7XKdxruWgLDbSQgZsj8oZPS+lZJNIIdChBa5gAikwegc2an7AIGfLPG3haoredAq4+N3EP4FeN1N3UjdBcLQ8YXDSNFaQToQT8Qhmb2kVafkXtL/w7t/O+37Hc1ve9eDcvAQO4r3HW5qQo6Dvlf2aVbSFef5Ufu0vIWhGYMEA7GiW3/EApCI65teZXqARm6Z/i1xVfU0EpTpeHu3vZLgFGdbM0bKKMquLbO9vjKy/i3
gO0v1mLbTOMuyOyhuvSDlZcPDsPvcUKAV0W9B88x23gL6wYQ70pZ2eaaAOe5NSpZoug4uXFFpurw4MhH40gaeKBRC5X4nijJdHGr3YuSCRXy8L4/xq60C+GwT8Le+H0NvGlz99+s+9nB2/XVm2iLk0a4584wYgl7apfQb1p4t3s8wS+6akvBDNoPNwn5lnIkzAKgqHBEc3g4/Tky2Qga1FH2vfjUgkEUrCXkht9nJYcd
rbgpYhRozVvAiH2wvgtDc5CHpWtYGOmdmOoqpynBubTOH+2WcDBgAh93rQW1DYBz7JzDlpk/Cnx4cNKm3FHAHFjB7d84HpbruX5Q5u10OrCs5lj0Gdd8uqoOMDsf4s0wuIzvt1G0OHcmuw1K7OHf6JCC+wGZXRC352BMcEzMDgNNKiaGQ5Xio/g9iK2Ze7HvJsFTYlOkS0dqcWOlTbMfsMRawkjg4OYTHu7v1jR5W5mB
oFIF7OCWwVj7nIlMuQpout3xdkMspW8Qlz431XXWubGVCpOrTA2l62HwAVTVAecSKHOe4wRaZ3f4oZIdYBX2xD3glclv4DqgntUMo9rlhPT16s5ZfDXsfbznMyMjXE9xLkncR8nF1h51mMVLuqB7L+5SSM/5nuTDOac9Mv+GQQhfPN7incebpg6K0c0wyFfpzmGrrwbC0TxXknEkEEV6Zz1wBlm8X87YHfBmesmeJlP
RBeh1OwtIXLOkfiMQJ3F/hLJcKR6/GVvp3CDNaWVGGrP87jXKQrMHOSF06CBHzzcmNu/cT5ruyYkFqcnYZmp118j3X8xy7H90SLj1BXhF7W/S9HjWTFkkDr09ZHyqFuwsDVd3speLO8O2aZHm5nssz16Se8W4qTzJmHx4g6XH2/XoKCqVwJikJ0SsOUmLmKmMs/F5QOt6ez3NP14ogMESGGzLQXd4B1eknd1S4w2J06m
EJsMuzyuYWORNnKN6G7/KjbtjeI8pRfRFs6SMGQQTh4K3OyEMfCpb7V1QAbqCetwgP6eOwm5FOGRAPMhjWCK2RqbcsJ0rZarBSo/zsuHDqOkeI6ZlhgY46qO3x+WHOGE1a0QYwU0BKkA5S8C+QI0+Cv4NGKo7moPqhbHhR3eqYfxTgb2CVQ7hmkwr/jxSQRKj0I5ysZklEAxl7HUGiUvxEOGQjWL0evnTQqVSj60sBCh
u1HbXKMcE8v+Gl7euot36CfYTntp3D/cr8K8/V1C/MSR9WM2NawbKjDbypCXd05+OaX+kYp8AUUsUGyt8h0YUbo3JvTgZplUTcl91ITvQDYSoygFIGT64qwF2f7LojjSxFPtZJjsWmPRHOG62kj4s/WGZug2ShJJH0swnF8+l7kvVy7eQzPA1x8OpAl823FPUiX+tyGvPZbY2BA3mAOHIrj+vKxRLZZ9AfleYXu1jePH
TM3MQ6OmYdv7NYJkzehXW7Fs7sndU1qio/yAYi6xq1YiNoC/rbR8WLu3reu2GrHQdXa+GFeXoZDpGzwtqu0FNimejQnhkTle7N/MSJh0lELZ2djVYDzM99YwMTDuNJdNoZnAoiRpvNhg3TdxEnQp+BsIUZL0OeE6WrqoFlp59/0G9xCJxEz97ZheZRzXRQVJjgxonkQ3D41JjUSNjrxHgM/GulGxLNxdaXSTFF/aaoP
3Out/fb2TP5lqazbhGjydkmlk2T+EovF8usT5t2xOADBnAhf3LVwpwu4rcojNOz0Cgqwizyt8h2f4Tj/zljZECK/XJ83DB0lPW6fToALkKovWc7zZqok7s2vjEAQ00ZWusLz+Bc6Xb4KpPhZRsOlHIbt87dEyv+IrcvL2DFDwO+/riQ7+NMmofEkh6YorqWqWF4a3ZfUsXgYSUVzfHlOJbBq6tGDSAvBwD2zPKaXmMz
jyUumzomRNa7VX92ckpZNkg+4ZvQ58B3dtmXmQJn/UHES4A2UnZaxNf7tMIeZOW2ub0xCLauMhiAQTumFnDZeEU/frlrUI8+v7B/ukW4/CcKEDvao9SOphgLfvgXuwtQdbgTVjFigbsGy/b6vJZgqWtz9eDea44HwUGf5zbisO23dKM9iJa6G5/twKctziVpzPs3Qtcq5JY3sECQOHhSadA5jWGKHgU8C2uGY09Zq9aC
qnUZNlibtCdCbAZoqR3Xu1ldouleej3I+At/Rm6twi4/dLZ2VhnNHCwZHM7PhgZxNtrs

- So, as you can see in the above image, the text file that is provided for us is completely human unreadable. So, in order to understand this, 1st we need to decrypt this text using Cryptool or any other online decryption tool.

- But when you try to decrypt the text using the tool, it asks us to "Enter a Secret Key". So, in order to find the Secret Key, you need to look at the hint that is provided in the question itself.

- But in the hint, it has only provided the last 4 digits of the Secret Key. But as we all know, inside a Secret Key there has to be 16 digits. That means all the first 12 digits of the Secret Key has to be 0's.



- So, as you can see in the above image, after the decryption, now the text has turned into a human readable format. But 1st let's copy that decrypted text to a separate text file.

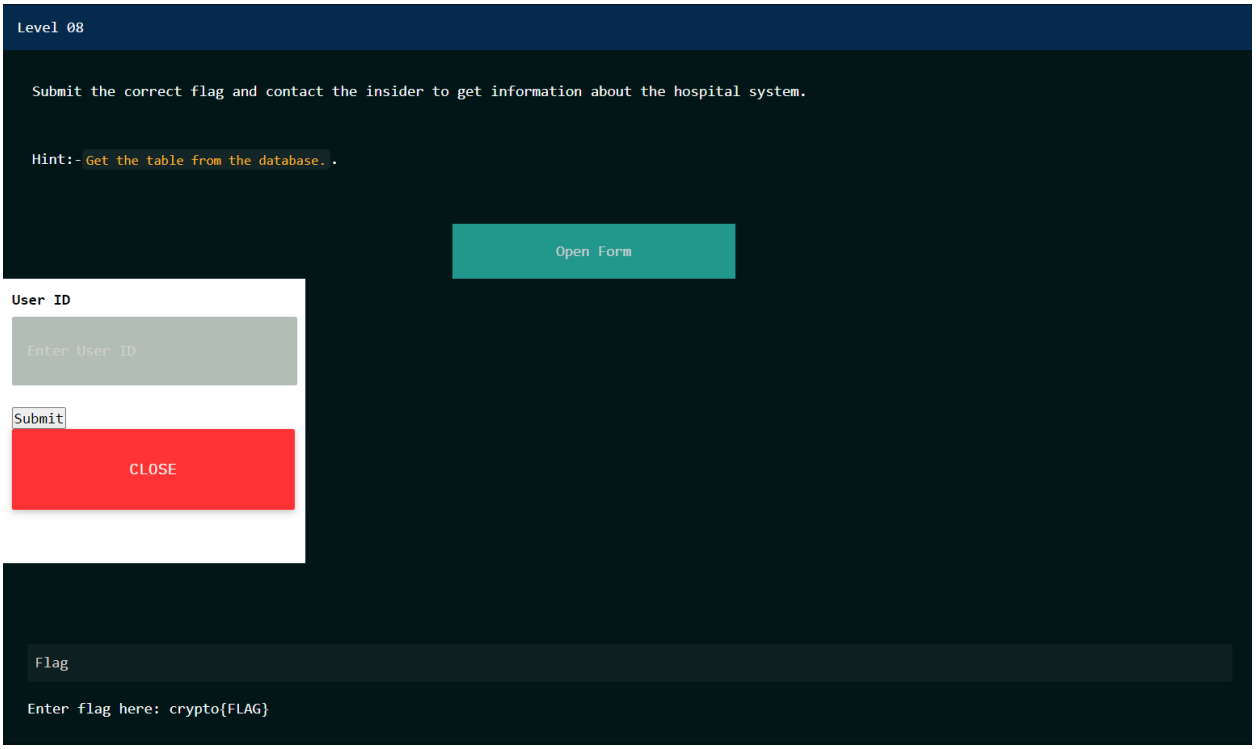
Researchers at Swiss-based code quality and security solutions provider SonarSource discovered earlier this year that OpenEMR is affected by four types of vulnerabilities that impact servers using the Patient Portal component. The list of vulnerabilities includes command injection, persistent cross-site scripting (XSS), insecure API permissions, and SQL injection. The Patient Portal enables healthcare organizations to allow their patients to perform various tasks online, such as communicating with doctors, filling out new patient registration forms, making appointments, making payments, and requesting prescription refills. However, SonarSource researchers determined that if the Patient Portal is enabled and accessible from the internet, an attacker could take complete control of the OpenEMR server by chaining the vulnerabilities theyâ□□ve found. According to SonarSource, the Patient Portal has its own API interface, which can be used to control all portal actions. Using this API requires authentication, but the researchers found a way to bypass it, allowing them to access and make changes to patient data, or to change information associated with backend users, such as administrators. An attacker who is able to change administrator account data can exploit the persistent XSS vulnerability to inject malicious code that would get executed when the targeted admin logs in to their account. The Flag --> crypto{!F_tH!5_4tT4cK_5uCc33D3d_y0u_G3t_5o%_sH4r3_0f_M0n3Y} The JavaScript code triggered through the XSS vulnerability can then exploit the command injection vulnerability found by the researchers. The ability to execute arbitrary OS commands enables the attacker to take complete control of the OpenEMR server. Alternatively, if the attacker targets a user with lower privileges rather than an administrator, they can exploit the SQL injection vulnerability to gain access to the patient database and steal potentially valuable data. Exploitation of the XSS and command injection flaws requires admin privileges, but the SQL injection bug can be exploited with regular user privileges. SonarSource discovered the vulnerabilities in OpenEMR 5.0.2.1 and they were patched with the release of version 5.0.2.2 in August. Details of the flaws were only made public now to give users enough time to install the update.

- If you carefully analyze the above decrypted text, you will be able to discover the correct flag for the level 7.

account data can exploit the persistent XSS vulnerability to inject malicious code that would get executed when the targeted admin logs in to their account. The Flag --> crypto{!F_tH!5_4tT4cK_5uCc33D3d_y0u_G3t_5o%_sH4r3_0f_M0n3Y} The JavaScript code triggered through the XSS vulnerability can then exploit the command injection vulnerability found by the researchers. The ability to execute arbitrary OS commands

8)Level 8 :-

There is a specific login form. This login form is connected to a small database of 5 records. So, a player needs to perform SQL Injection attacks in order to login and retrieve all the records. So, the flag is available inside one of those records.



The user should submit the following query to get the table data

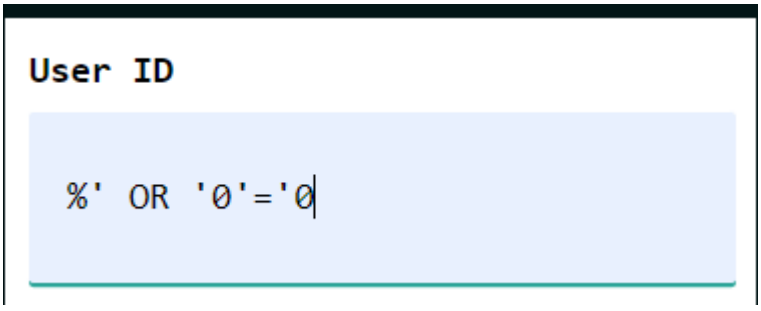
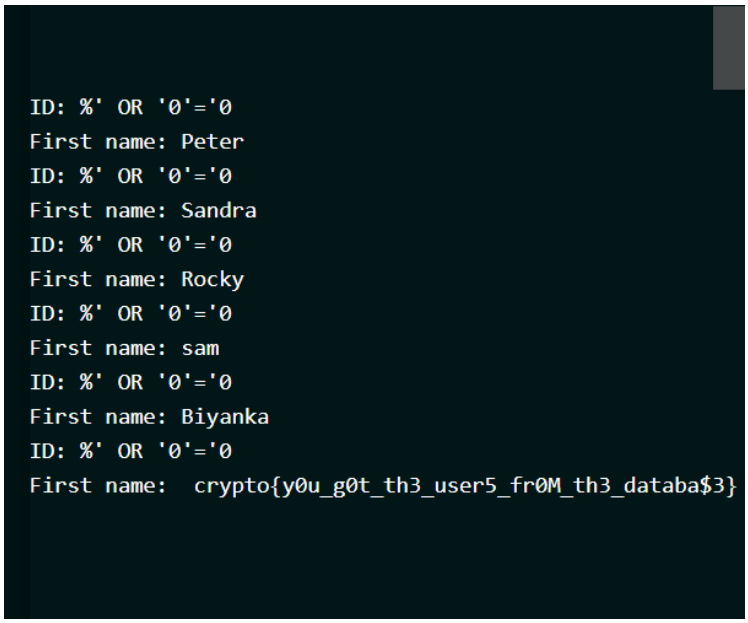


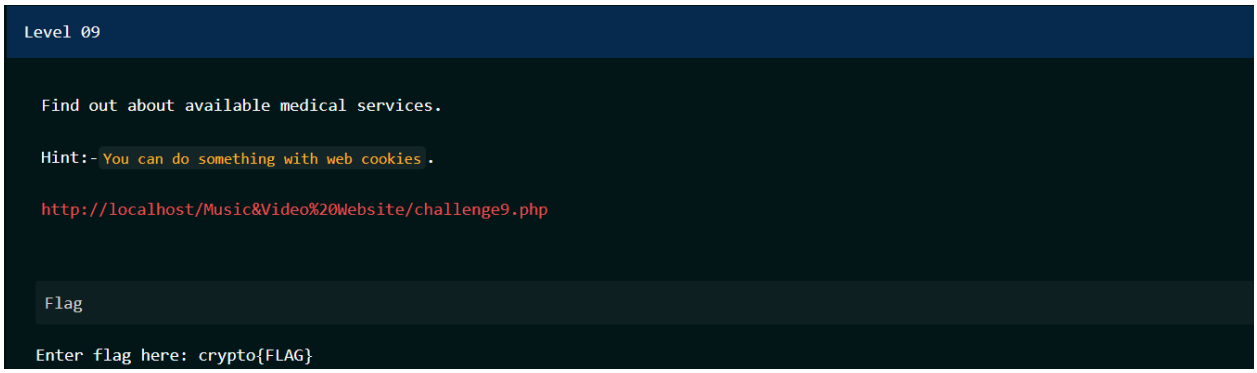
Table data will then be displayed.



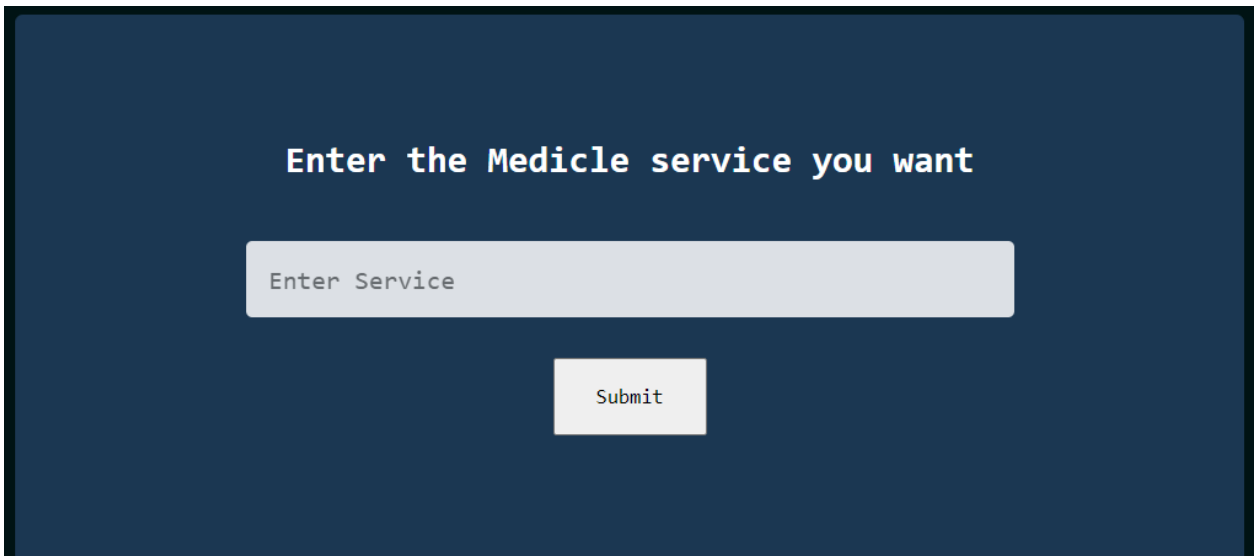
FLAG:- crypto{y0u_g0t_th3_user5_fr0M_th3_databa$3}

9)Level 9 :-

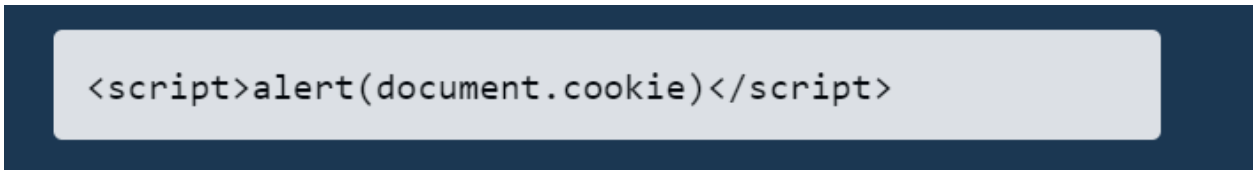
There will be a link that is provided for the players which will redirect to a new page. In this page, the player can input the “type of medical service they want”. Ex:- Mental Health, Dental Care, Preventive Care, Physical Therapy etc. So, when a player inputs a service, it will just display “You have successfully registered for <Example_Service>. The appointment details will be informed shortly. Thank you for registering for our service & have a nice day!” But if the player performs an XSS attack, he will be able to retrieve the value of currently used web-cookie. So, that will be the flag for this challenge.



The following page will appear when the user clicks the URL.



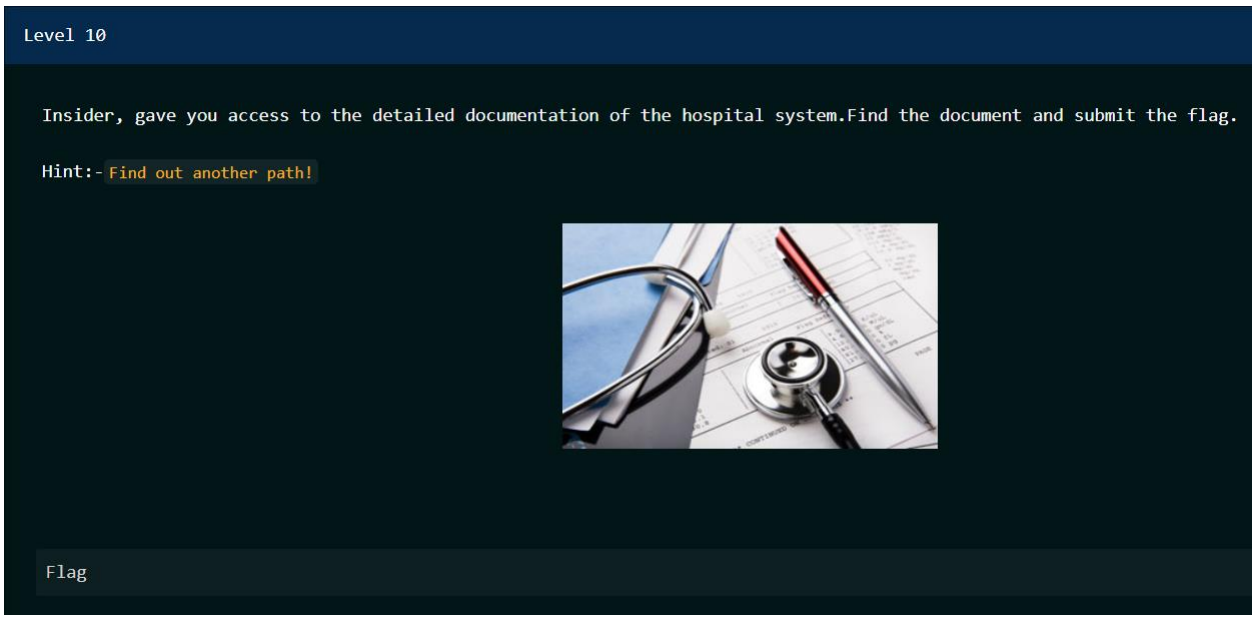
Perform the XSS attack.

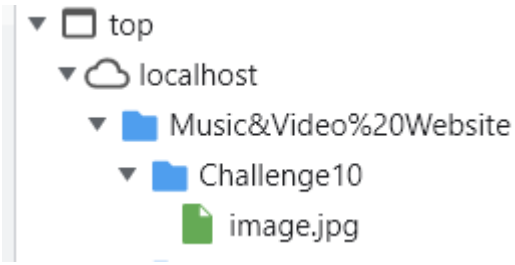


FLAG:- crypto{tH!5_1s_th3_53rvi53_y0u_5ub5cr1b3d}

10) Level 10 :-

In this level, in the hint it says to find out another path. The player can check the source code to see if there are any paths specified in the code. Inside the source code there is a folder called Level 10. Then the player needs to navigate to that particular folder and check its content. So, inside the Level 10 folder, there is another folder called “hidden”. Inside that “hidden” folder, you are able to see the flag inside a text file.

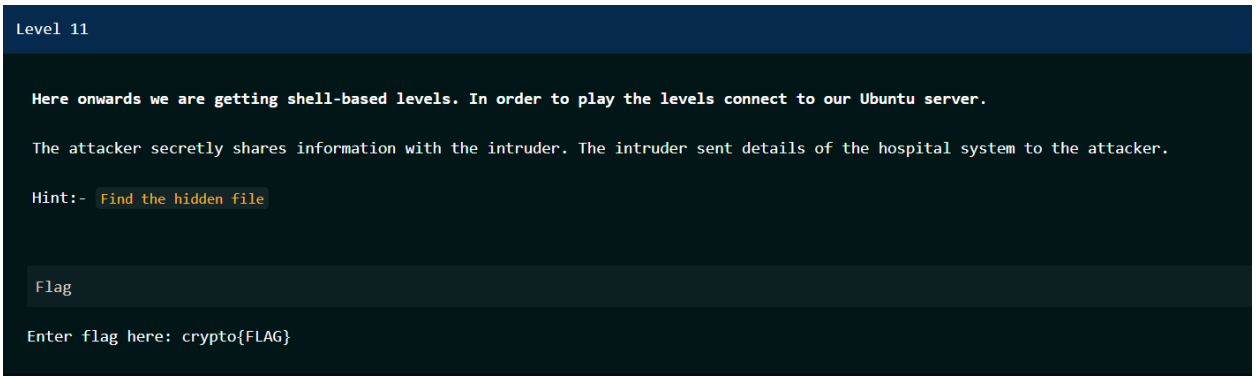




FLAG:- crypto{ThIs_IsA_ToP_SeCrET_DoCuMeNt}

That is all the web-based levels and here onwards we are getting shell-based levels. In order to play the levels connect to our Ubuntu server.

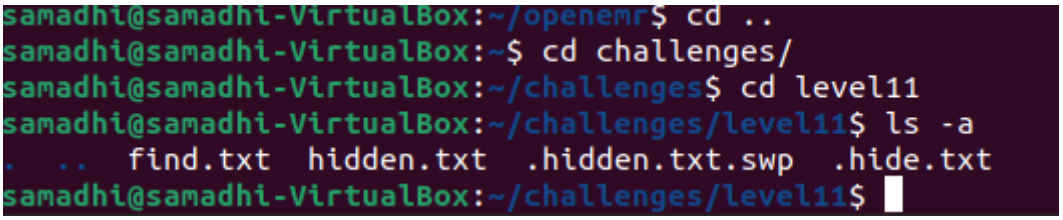
11) Level 11 :-



In here, as the very 1st step, you need to **view all the hidden files** that are inside of the **"level11"** directory.

cd level11

ls -a

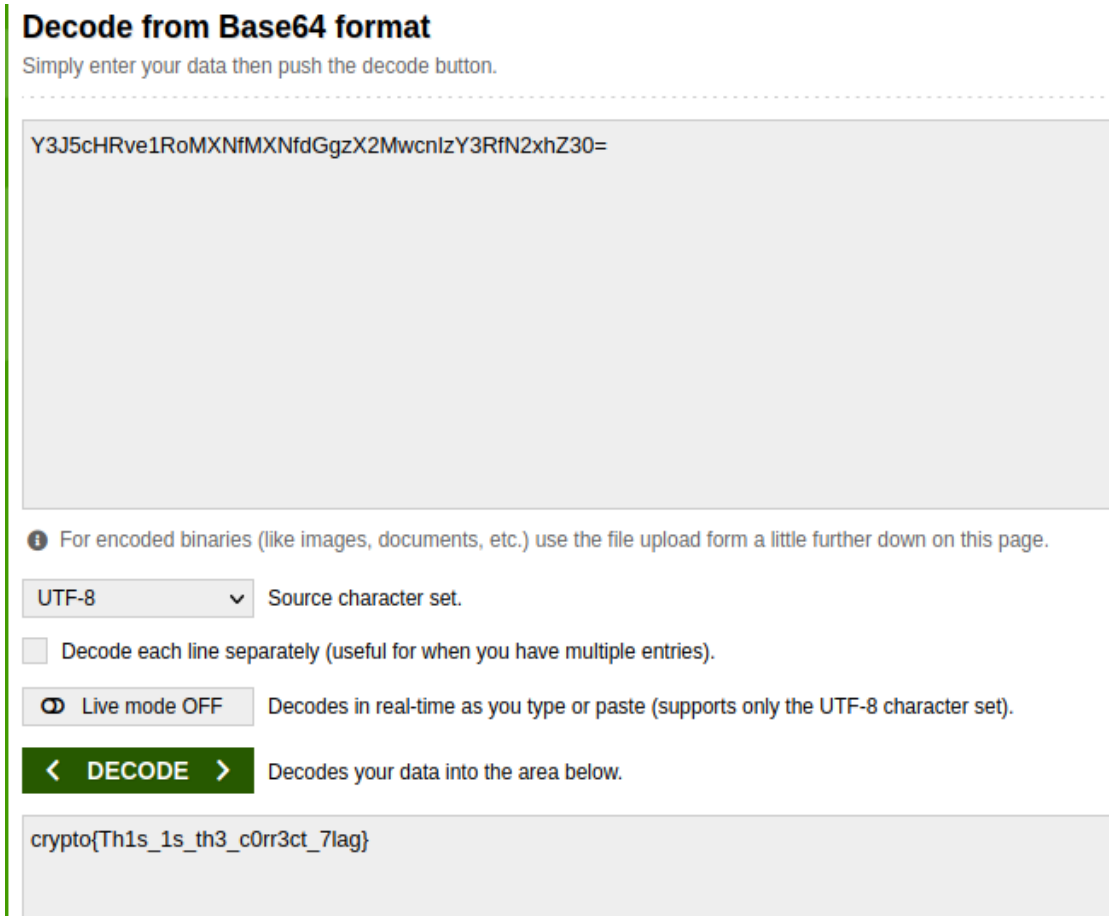


As you can see the name of the hidden file is **".hide.txt".** So, let's go ahead and open it.

cat .hide.txt

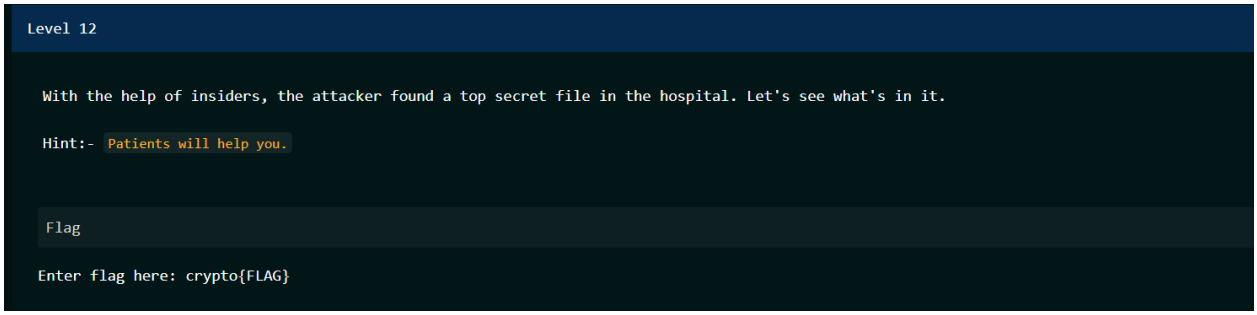
```
samadhi@samadhi-VirtualBox:~/challenges/level11$ cat .hide.txt
64 - Y3J5cHRve1RoMXNfMXNfdGgzX2MwcnIzY3RfN2xhZ30=
samadhi@samadhi-VirtualBox:~/challenges/level11$
```

However, as you can see in here, the flag is given in **base64 encoded format**. So, you need to decode it first before submitting it directly.



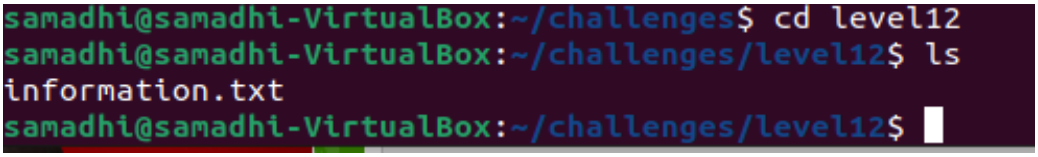
FLAG:- crypto{Th1s_1s_th3_c0rr3ct_7lag}

12) Level 12 :-



When you navigate to the directory of “level12”, you are able to find a text document named as “information.txt”. Inside this particular text file there are thousands of various string values. So, the correct flag for this level is stored among those string values.

cd level12

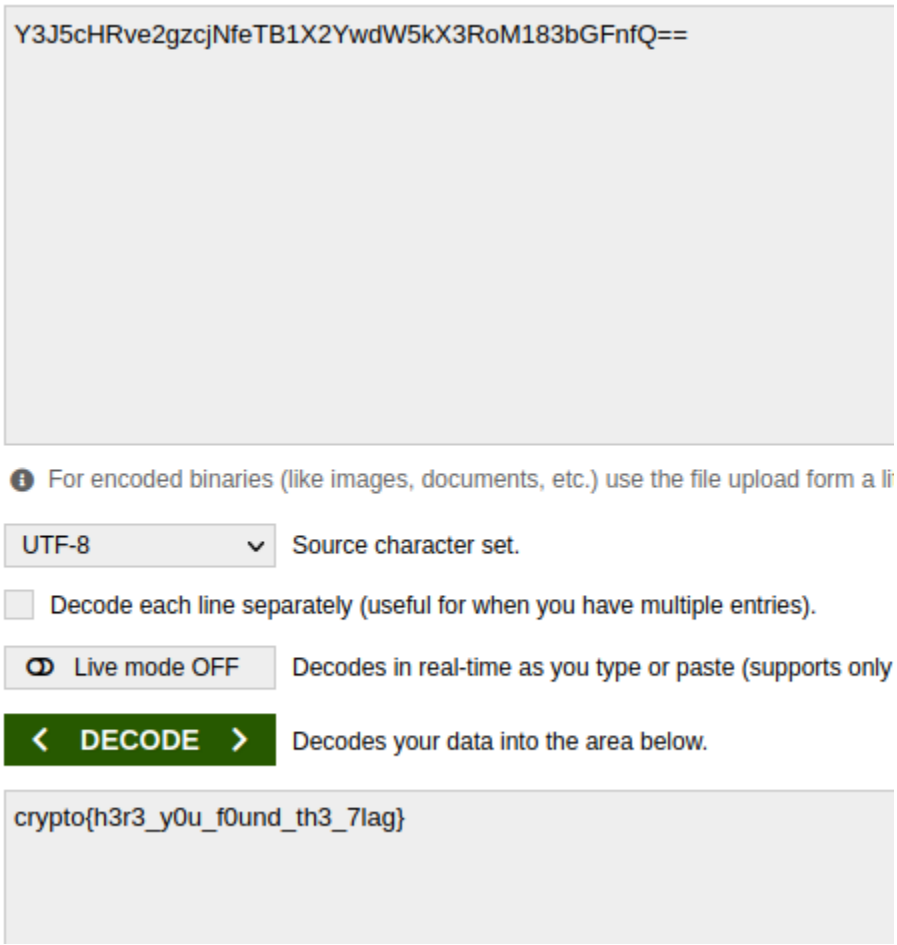


But when you go to the hint, it says that “Patients will help you”. That means, the correct flag is located next to the “Patients” string.

cat information.txt | grep “Patients”

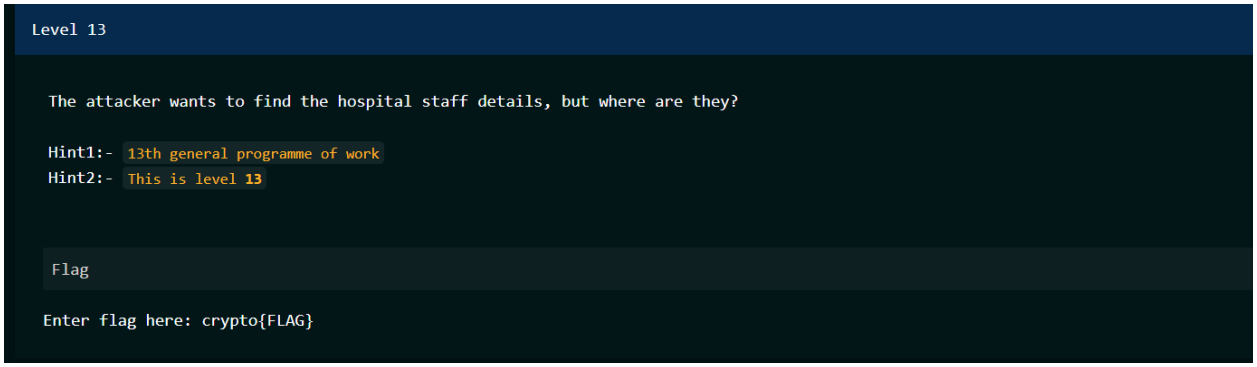
```
samadhi@samadhi-VirtualBox:~/challenges/level12$ cat information.txt | grep "Patients"
Patients    Y3J5cHRve2gzcjNfeTB1X2YwdW5kX3RoM183bGFnfQ==
samadhi@samadhi-VirtualBox:~/challenges/level12$
```

Then you need to decode the above string to get the flag.



FLAG:- crypto{h3r3_y0u_f0und_th3_7lag}

13) Level 13 :-



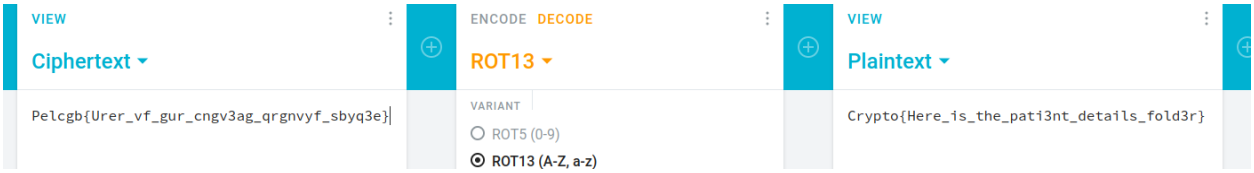
When you navigate to the directory of **"level13"**, you are able to find multiple other directories. **Inside each & every directory, there is a text file named as "flag.txt".**

But in order to pass this level, you need to **find the correct "flag.txt" file.** As usual, when you look at the hint, it says that **"13th general programme of work".** So, what you need to do is that, you need to **google this phrase as it is.** Then you need to navigate to the 1st web site within the search results. In this web page, you are able to find out a keyword named as **"triple billion targets".**

So, inside our virtual environment also, there is a directory called as **"The_triple_billions_targtes".** Now you need to open the **"flag.txt"** file inside that directory. So, when you open that you are able to find the correct flag for this level.

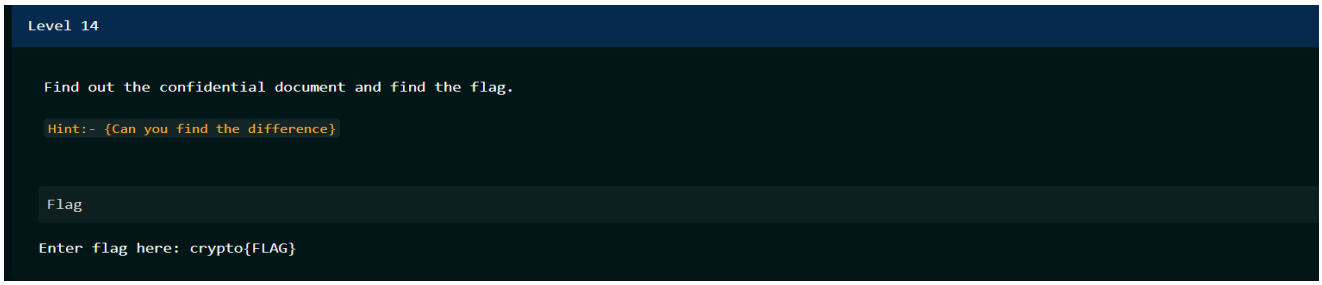
```
samadhi@samadhi-VirtualBox:~/challenges/level13/confidential$ cd ..
samadhi@samadhi-VirtualBox:~/challenges/level13$ cd The_triple_billions_targets/
samadhi@samadhi-VirtualBox:~/challenges/level13/The_triple_billions_targets$ ls
flag.txt
samadhi@samadhi-VirtualBox:~/challenges/level13/The_triple_billions_targets$ cat flag.txt


Confidentiality is the principle and practice of keeping sensitive information private unless the owner or custodian of the data gives explicit consent for it to be shared with
another party. Confidentiality may also refer to the request to honor the principle and practice.

To ensure confidentiality, owners and custodians of sensitive data implement policies governing the kinds of information that warrant protection. Based on that, they define a nu
mber of processes for the settings, devices, and persons involved in the handling and storage of data. These include educating and training employees and those they serve; inves
ting in and maintaining the facilities, hardware, and software where data resides and travels; keeping records of sensitive data's movements; and data loss prevention (DLP) plan
ning and mitigation.

The Health Insurance Portability and Accountability Act of 1996 (HIPAA)

Pelcgb{Urer_vf_gur_cngv3ag_qrgnvyf_sbyq3e}

Confidentiality is the principle and practice of keeping sensitive information private unless the owner or custodian of the data gives explicit consent for it to be shared with
another party. Confidentiality may also refer to the request to honor the principle and practice.

To ensure confidentiality, owners and custodians of sensitive data implement policies governing the kinds of information that warrant protection. Based on that, they define a nu
mber of processes for the settings, devices, and persons involved in the handling and storage of data. These include educating and training employees and those they serve; inves
ting in and maintaining the facilities, hardware, and software where data resides and travels; keeping records of sensitive data's movements; and data loss prevention (DLP) plan
ning and mitigation.

The Health Insurance Portability and Accountability Act of 1996 (HIPAA) is one
samadhi@samadhi-VirtualBox:~/challenges/level13/The_triple_billions_targets$
```

However, the flag is encoded with the **"ROT13 Encoder"** *(A Shift Cipher method)*. The reason we used "13" is that, it says like that in the 2nd hint.



FLAG:- Crypto{Here_is_the_pati3nt_details_fold3r}

14) Level 14 :-



When you navigate to the directory of **"level14"**, you are able to find 5 text files. All those 5 text files consist with **similar looking set of string values.** However, there is **1 text file, that is different from all the other text files.** So, you need **compare all the files with each other** in order to find that **particular difference**.

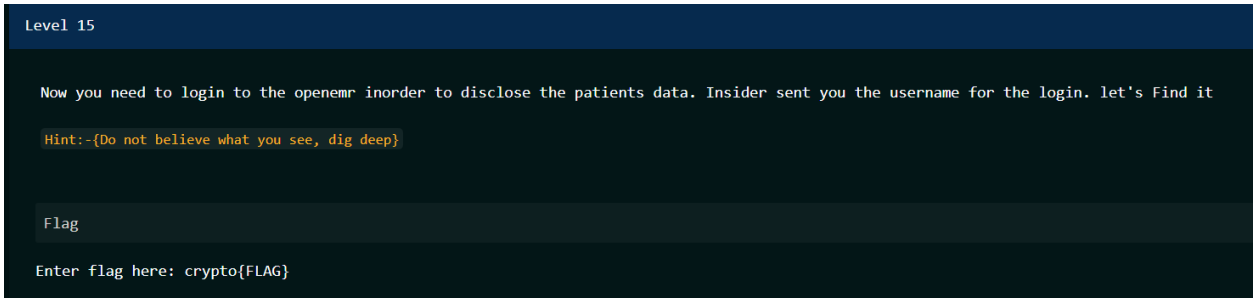
cd level14

diff PHI.txt security_details.txt

```
samadhi@samadhi-VirtualBox:~/challenges/level13$ cd ..
samadhi@samadhi-VirtualBox:~/challenges$ cd level14
samadhi@samadhi-VirtualBox:~/challenges/level14$ ls
clinical_details.txt  Employee_details.txt  patients_details.txt  PHI.txt  security_details.txt
samadhi@samadhi-VirtualBox:~/challenges/level14$ diff PHI.txt clinical_details.txt
3933d3932
< crypto{y0u_g0t_th3_pr1v1l3g3s_t0_th3_syst3m}
samadhi@samadhi-VirtualBox:~/challenges/level14$
```

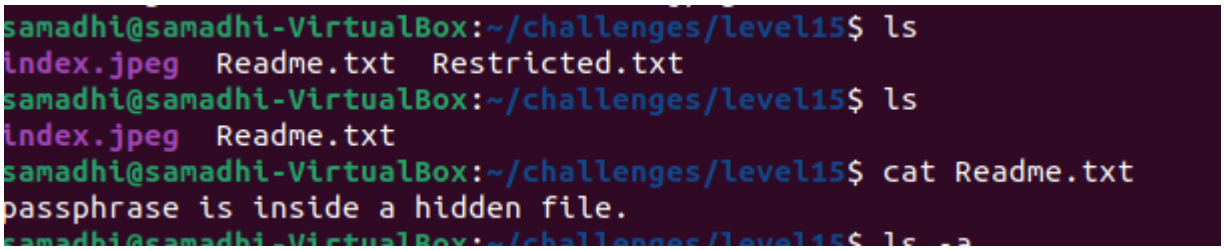
FLAG:- crypto{y0u_g0t_th3_pr1v1l3g3s_t0_th3_syst3m}

15) Level 15 :-



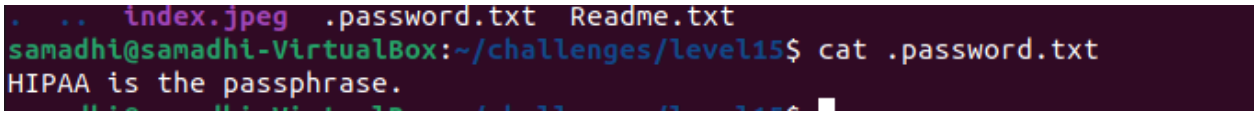
When you navigate to the **"Readme.txt"** file, it says that the **"passphrase is inside a hidden file."**

So, this particular passphrase is required when decrypting the given image.



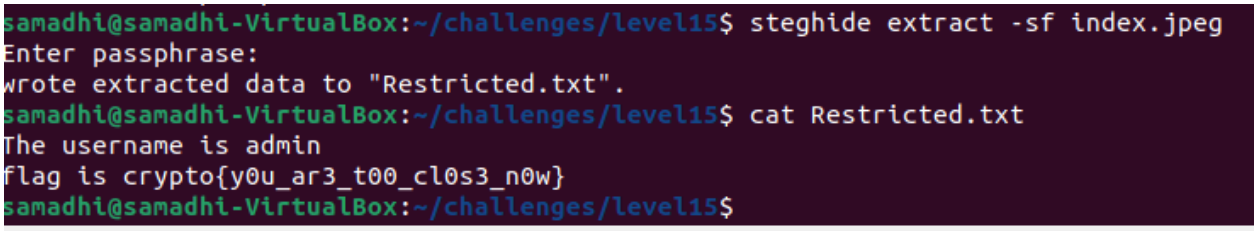
In order to find the passphrase, you need to navigate to the hidden files in that directory.

ls -a . Inside the **".password.txt"** file, the passphrase is given as **"HIPAA".**



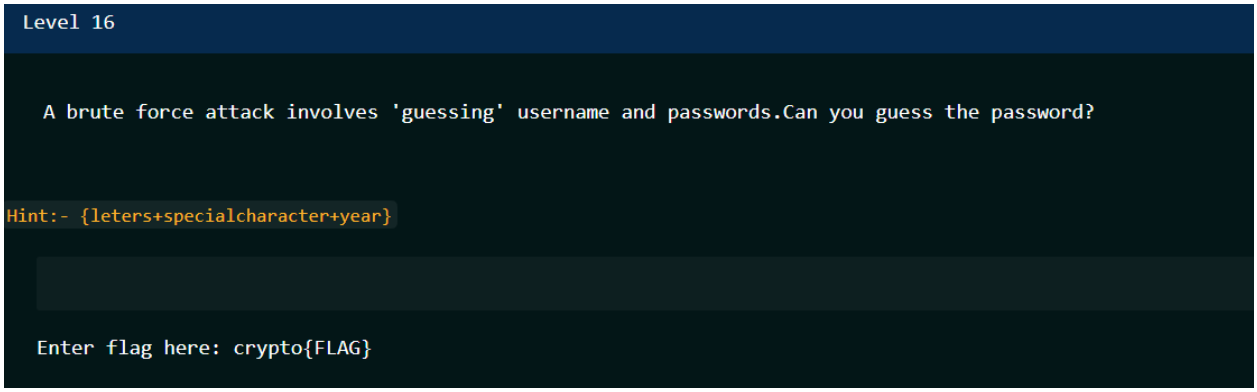
steghide extract -sf index.jpeg

After that, another text file will be automatically created called as **"Restricted.txt".** Inside this text file you are able to find the correct flag for this level.

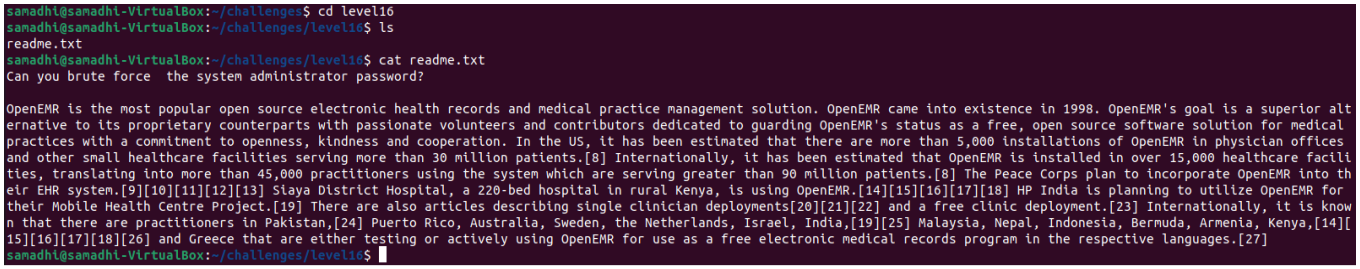


FLAG:- crypto{y0u_ar3_t00_cl0s3_n0w}

16) Level 16 :-

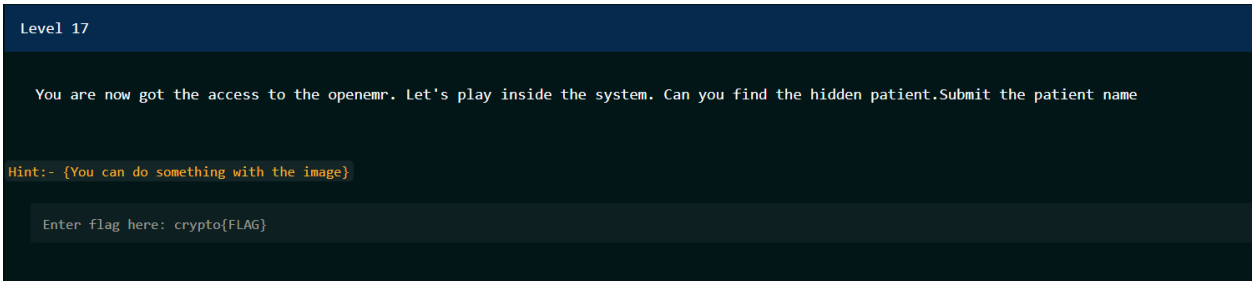


When you navigate to the **"readme.txt"** file in the **level 16** directory, you are able to see a paragraph. So, the player needs to read this paragraph carefully and then need to find the correct keyword combination in order to pass this level. Hints says the format of the password as ***{letters+specialcharacter+year}.***

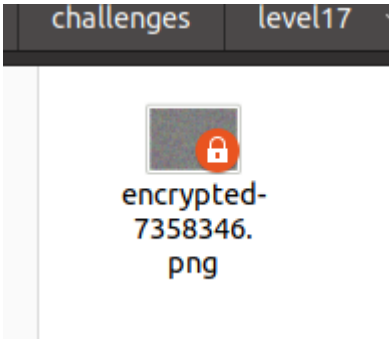


Correct Password:- OpenEMR1998

17) Level 17 :-

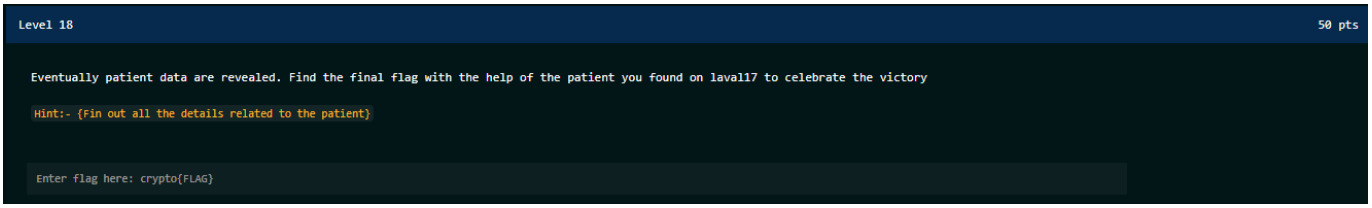


In this level there is a encrypted image file inside the folder . You need to decrypt the image in order to get the Patient name.



Patient Name:- Eranda

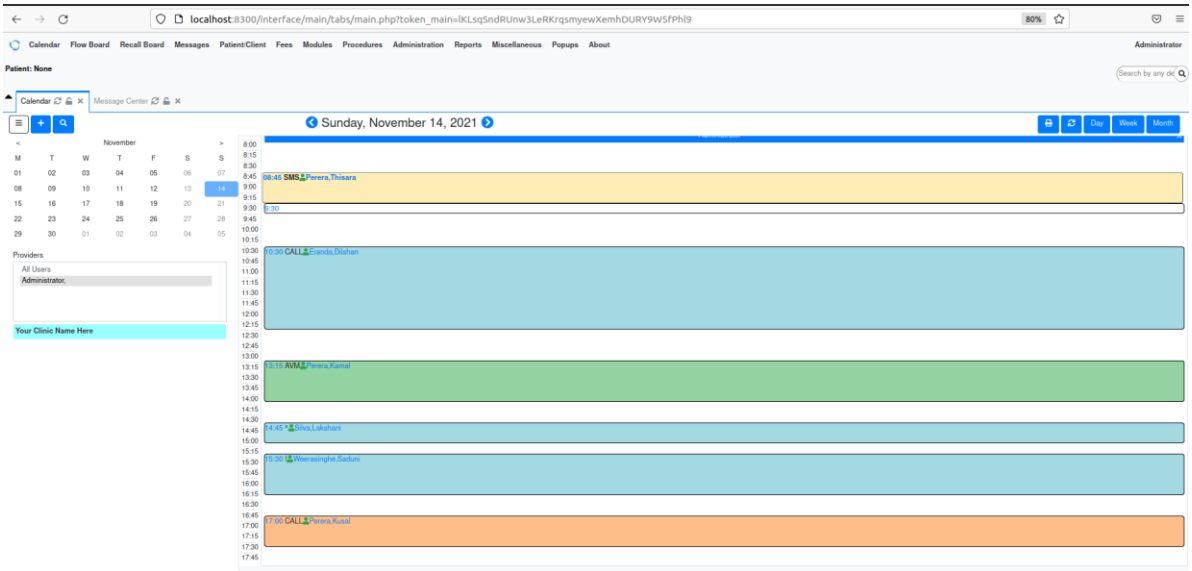
18) Level 18 :-

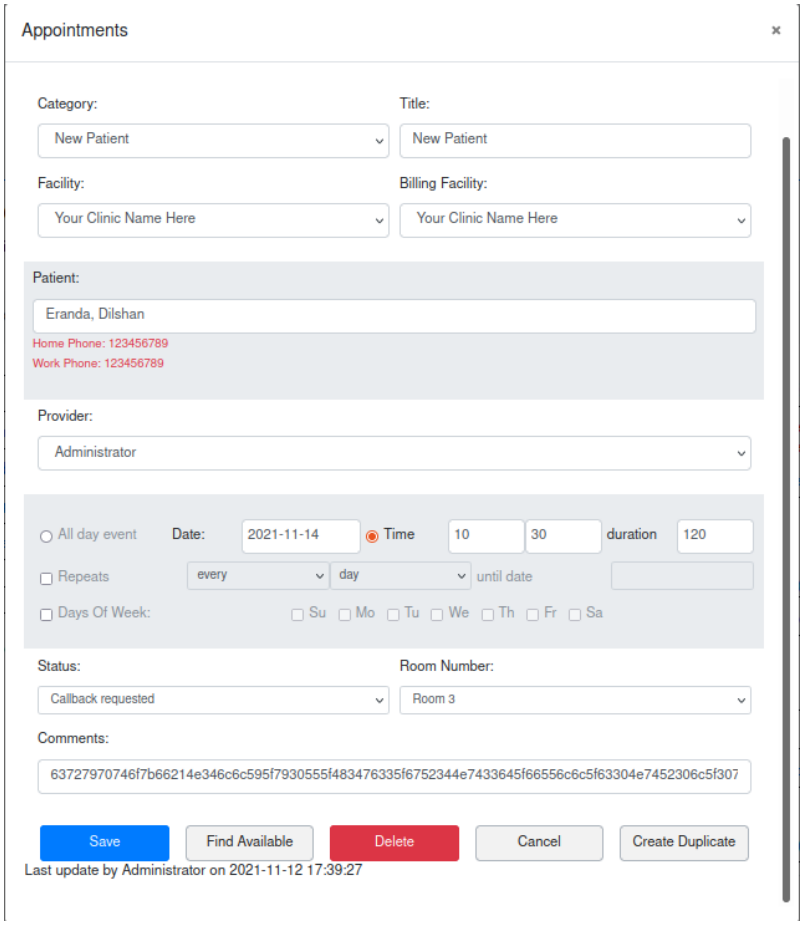


- In challenge 18, it tells us to find out the flag using the OpenEMR Patients' Record Database. Since this is the final level, they haven't provided us with any hint either.

- However, if you critically think about this situation, in the challenge 17, we discovered some secret text hidden inside an unrecognizable image. So, we need to take that secret text as the clue in order to proceed through this level.

- When you navigate to the OpenEMR Patients' Record Database, you are able to find out lot of information. However, when you navigate to the appointments section on the date of 14th November, you will be able to find a name called as "Dilshan Eranda". So, let's go ahead and open up that appointment.





- As you can see in the above image, the comment section of that particular patient's appointment seems to be bit suspicious. However, that comment also provided in a human unreadable format. So, let's go ahead and try to decode it.

So, as you can see in the above image, that would be our last flag for our final challenge in the CTF Box that we have created.