Sri Lanka Institute of Information Technology

Final Project Report

ISP Project Report

Information Security Project 2021

Project ID: [redacted]

Submitted by:

IT Number	Name
[redacted]	[redacted]
[redacted]	Ekanayake E. M. I. R

Date of submission

14.11.2021

Abstract

SR86 Hijack is a capture the flag (CTF) program developed as for the requirements specified in the information security project module. This CTF program is developed in order to increase the practical knowledge of the players who are interested in information security-based tools and technologies.

In the CTF, the main scenario is based on an investigation scenario of an aircraft hijacking situation. The players will have to go through various types of practical information security forensic based levels in order to retrieve the flags required in the CTF levels.

In the initial parts of this final report, an introduction to the CTF development, a description about the scenario, and an introduction about the tools and technologies that were used to develop the CTF is included. In addition, details about the database development are included. The methodology that was used is thoroughly described in the methodology section. It includes brief descriptions about the levels and the methodology to play the various levels included in the CTF.

Additionally, how the **TryHackMe** implementation of the CTF was done and the development of the website is also described in the methodology section of the document. In the final sections of the document, an evaluation about the methodologies used, brief description about the problems occurred and the future possibility for the development of the SR86 Hijack CTF is discussed.

Acknowledgement

We are grateful to Dr.Lakmal Rupasinghe, Mrs. Chathu udagedara, Ms. Menaka moonamaldeniya, lecturer and instructors for the Information security project module for granting this opportunity to us to gain knowledge about how to create a CTF box, how to implement the levels, understand what are the tools and techniques that can be used for the implementation, how to host a CTF in a public platform such as TryHackMe properly and how to document a proper report for created CTF challenge in an academically accepted manner.

In addition, we are grateful for the continuous guidance, support and feedback received the implement this CTF.

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	SR86 Hijack: a scenario of a cargo aircraft with COVID 19 vaccines hijack situation
Project ID	[redacted]

Group Members

Reg. No	Name	Signature
[redacted]	[redacted]	[redacted]
[redacted]	Ekanayake E. M. I. R	Ekanayake.

Table of Contents

List of Figures

List of Tables

1. Introduction

1.1 Problem Statement

This product is a CTF (*Capture the Flag*) challenge which is required for the information security project module conducted in 3rd year July/December semester. This CTF is mainly based on a scenario of an airplane hijacking situation which is related to the aviation industry. The purpose of designing a CTF with such a scenario is to attract players who are interested in aviation related incidents and military intervention related incidents. In addition, training and awareness for the aviation related staff and military staff is one of the main goals of the CTF design.

1.2 Product Scope

The CTF is focused on aviation industry which considered as an industry that uses highly critical systems. In this particular scenario the mainly considered scopes are the aircraft control center systems and communication systems that are used for routing of aircrafts. Specifically, some functionalities that are used in **ACARS** systems (Aircraft communications addressing and reporting system) that is used to communicate with the aircrafts is one of the systems that are considered. **ACARS** is used to communicate short messages between aircraft and ground stations such as, Weather messages, air traffic related messages or voice messages. A typical system architecture of an ACARS system is represented by the image below.

Future Air Nav system (**FANS**) and **ACARS** is used to communicate using short text messages. Technologies used in FANS and ACARS are,

- VHF datalink (X.25 routing protocol) – TCP/IP in OSI model
- GSM communication
- SATCOM

Most standard ACARS transmit these messages in human readable format although some aircraft do encrypt the messages. In the scenario of SR86 hijack, the short messages between Air traffic control (ATC) tower and the aircraft gets intercepted by a false message. The selected features and technologies used in the architecture of this ACARS system components will be researched further for the implementation of this CTF box.

The main objectives and goals of the CTF are,

- This CTF box can be used by an Aircraft control center to train their staff to handle a similar situation.
- Train aircraft handling related staff on the proactive measures that must be taken in a hijacking situation.
- Secure the possible loopholes in an aircraft controlling system.
- Train military personnel to handle an aircraft related hostage situation.

1.3 Project Report Structure

The methodology includes the design constraints of the CTF development. The development of the TryHackMe implementation and the development of the web application with the related back-end database development is included with the steps followed.

After the design phase, a detailed description about the levels and the methods to play the levels is described. In the latter part of the document, an evaluation of the methodologies followed is described with the problems faced. In addition, the scope for the future development possibilities of the SR86 Hijack CTF is discussed.

2. Methodology

2.1 Requirements and Analysis

The SR86 Hijack CTF is a new self- contained CTF challenge that is planned to host in a web interface and using a **TryHackMe** room. The players can use the **TryHackMe** room or the website to begin playing the CTF and to submit the flags through the flag submission panels. Through the web interface and provided hints, players should access the FTP servers hosted in an UBUNTU virtual machine which will have the flags and hints related to other levels of CTF.

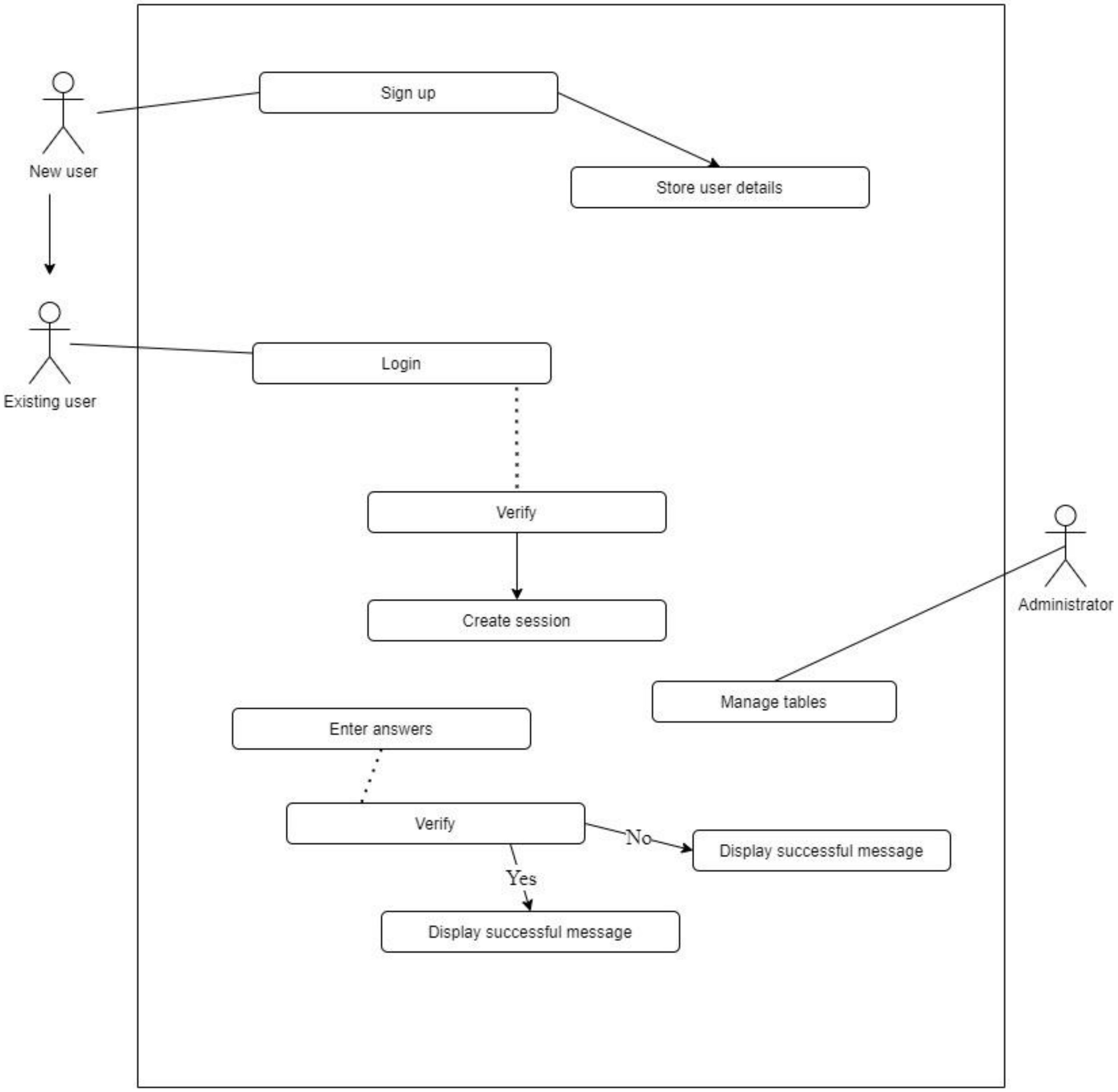
The CTF initially contains a web interface which give the users access to the CTF related materials, a login page and flag submission panels for various levels. In addition, a **TryHackMe** room implementation is done for better access and to increase the number of players. Through the hidden hints and flags inside the website, players can access to the server based on Ubuntu virtual machine and engage in playing other levels and submit the retrieved flags related to each level. Overall number of levels of the CTF is 8 levels that are mainly focused on forensic investigation-based technologies such as Cryptographic algorithms, steganography and password cracking tools such as **THC Hydra** and **Hashcat**.

Tools and technologies that are used for the design of the CTF is listed below.

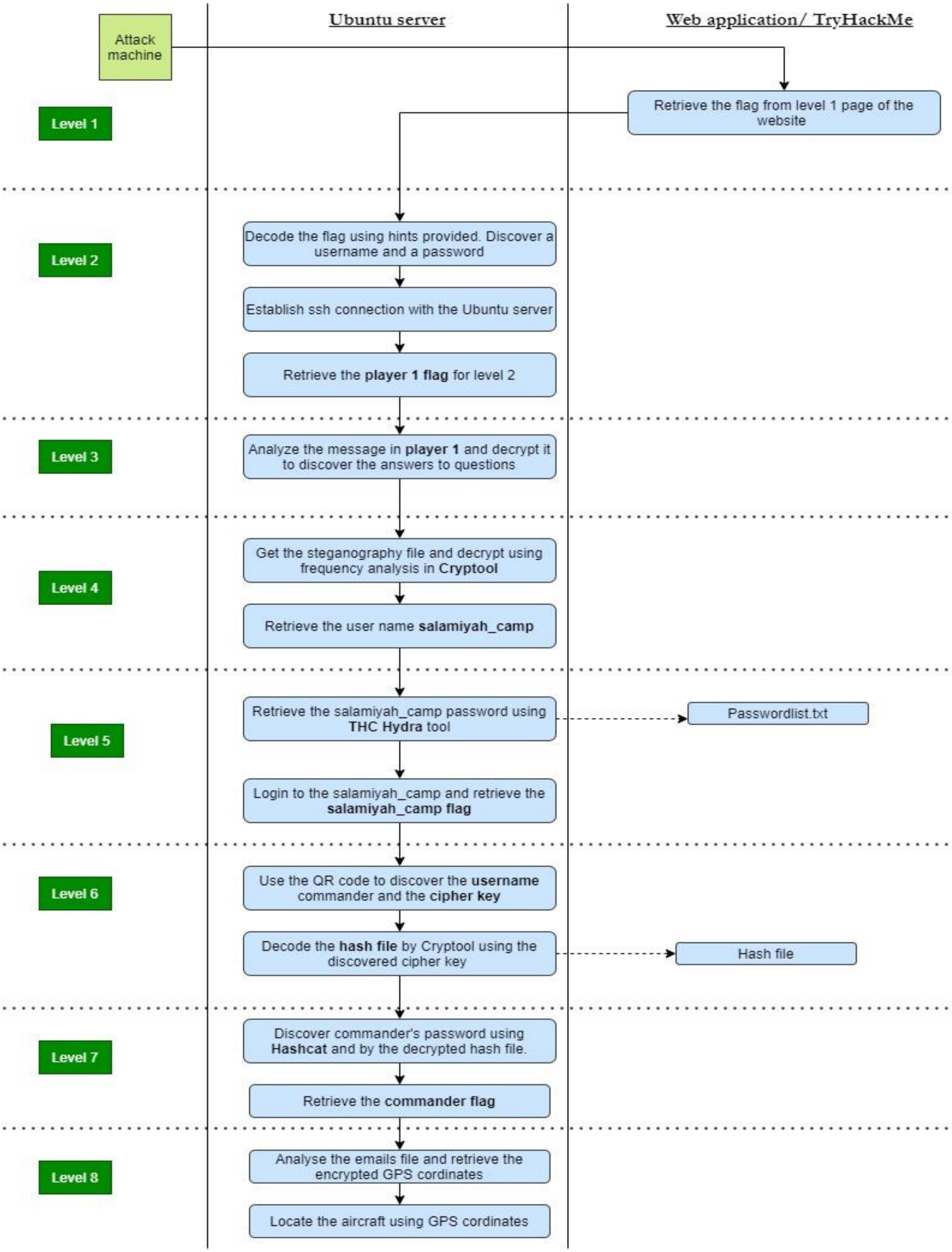
- **Web application** – HTML, CSS, JavaScript, PHP
- **Database** – MySQL, PhpMyAdmin
- **CTF VM –** Ubuntu Server 18
- **Attack machines** – THC Hydra, Hashcat, Cryptool, Steganography, Base64 Encoder/Decoder, QR

2.2 Design

SR86 Hijack ER diagram - Database functions



Level explanation activity diagram

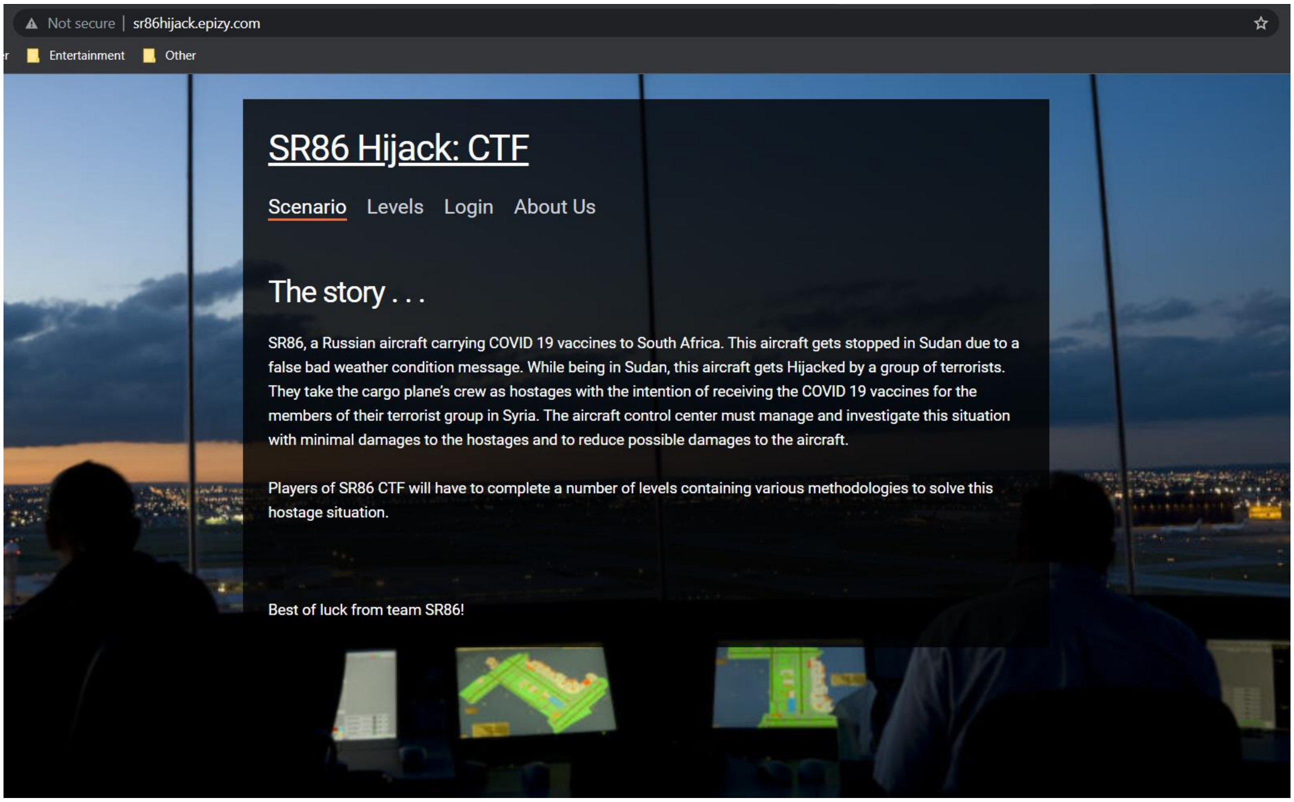

Attack machine
Ubuntu server
Web application/ TryHackMe
Level 1
Retrieve the flag from level 1 page of the website
Level 2
Decode the flag using hints provided. Discover a username and a password
Establish ssh connection with the Ubuntu server
Retrieve the player 1 flag for level 2
Level 3
Analyze the message in player 1 and decrypt it to discover the answers to questions
Level 4
Get the steganography file and decrypt using frequency analysis in Cryptool
Retrieve the user name salamiyah_camp
Level 5
Retrieve the salamiyah_camp password using THC Hydra tool
Passwordlist.txt
Login to the salamiyah_camp and retrieve the salamiyah_camp flag
Level 6
Use the QR code to discover the username commander and the cipher key
Decode the hash file by Cryptool using the discovered cipher key
Hash file
Level 7
Discover commander's password using Hashcat and by the decrypted hash file.
Retrieve the commander flag
Level 8
Analyse the emails file and retrieve the encrypted GPS cordinates
Locate the aircraft using GPS cordinates

2.3 Implementation

Web application

Web application contains all the details about the levels of the CTF and also contains a login, signup pages for players to login to the CTF playing environment as an additional method to the TryHackMe implementation.

Link: sr86hijack.epizy.com



SR86 Hijack: CTF

The story . . .

SR86, a Russian aircraft carrying COVID 19 vaccines to South Africa. This aircraft gets stopped in Sudan due to a false bad weather condition message. While being in Sudan, this aircraft gets Hijacked by a group of terrorists. They take the cargo plane's crew as hostages with the intention of receiving the COVID 19 vaccines for the members of their terrorist group in Syria. The aircraft control center must manage and investigate this situation with minimal damages to the hostages and to reduce possible damages to the aircraft.

Players of SR86 CTF will have to complete a number of levels containing various methodologies to solve this hostage situation.

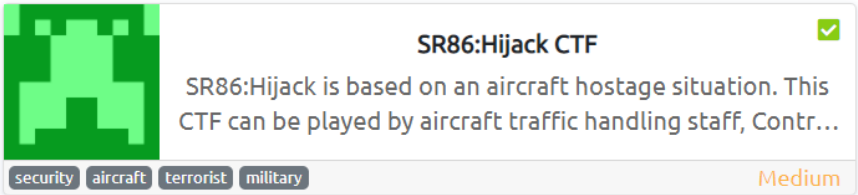
Best of luck from team SR86!

TryHackMe implementation

According to the requirements of the CTF development in Information security project, a **TryHackMe** room was created to host the SR86 Hijack CTF.

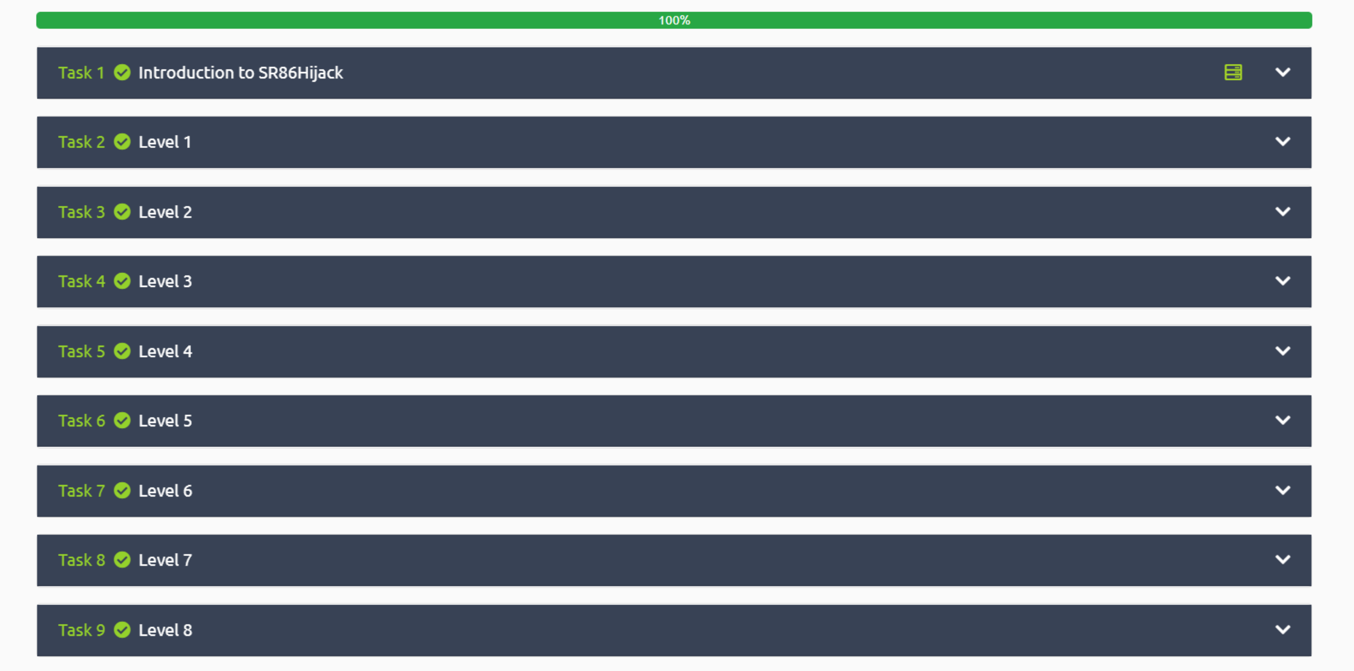
This room enables a built-in scoreboard and other necessary analysis tools to analyze users' progress with the completion of CTF challenges.

Link: tryhackme.com/jr/sr86hijackctf



SR86:Hijack CTF

SR86:Hijack is based on an aircraft hostage situation. This CTF can be played by aircraft traffic handling staff, Contr…



Task 1 Introduction to SR86Hijack

Task 2 Level 1

Task 3 Level 2

Task 4 Level 3

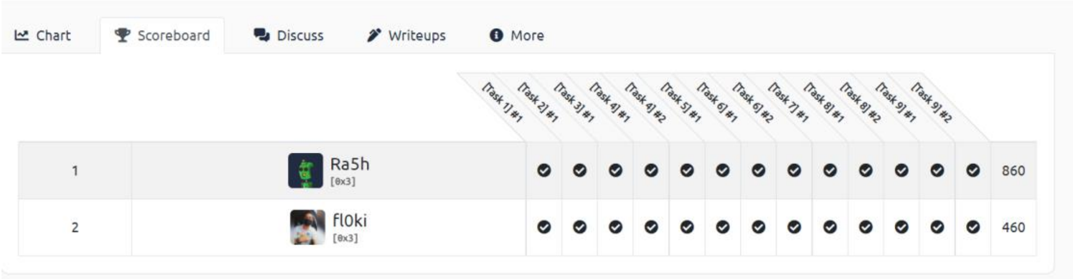
Task 5 Level 4

Task 6 Level 5

Task 7 Level 6

Task 8 Level 7

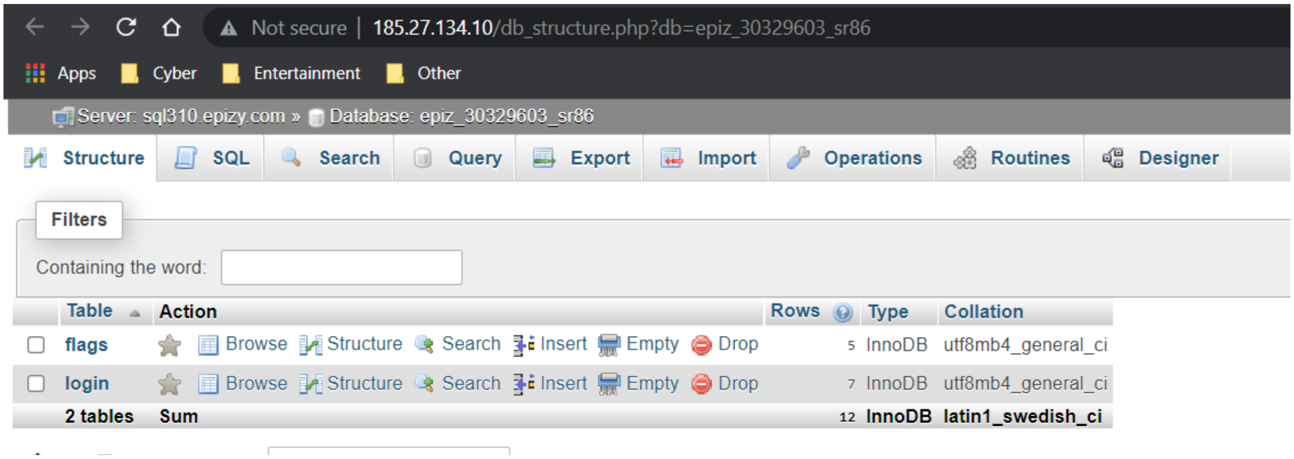
Task 9 Level 8



		[Task 1] #1	[Task 2] #1	[Task 3] #1	[Task 4] #1	[Task 4] #2	[Task 5] #1	[Task 6] #1	[Task 6] #2	[Task 7] #1	[Task 8] #1	[Task 8] #2	[Task 9] #1	[Task 9] #2	
1	Ra5h [0x3]	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	860
2	fl0ki [0x3]	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	460

Backend

phpMyAdmin panel

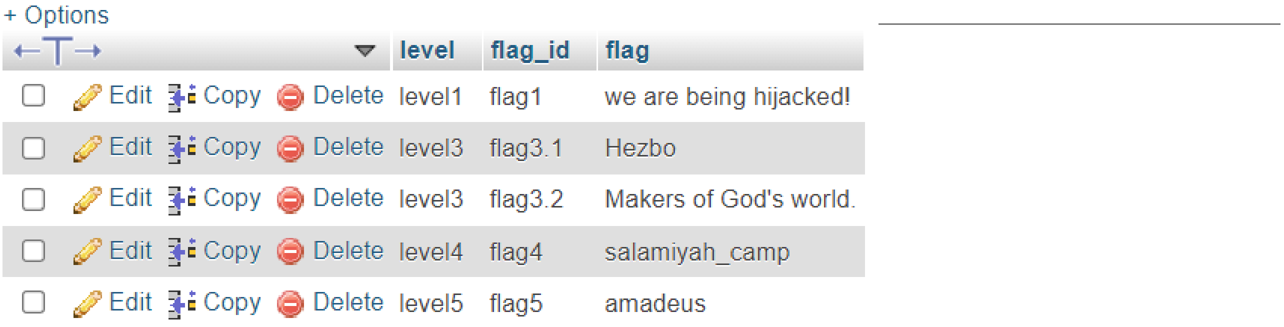


Not secure | 185.27.134.10/db_structure.php?db=epiz_30329603_sr86

Server: sql310.epizy.com » Database: epiz_30329603_sr86

Containing the word:

	Table	Action	Rows	Type	Collation
□	flags	☆ Browse Structure Search Insert Empty Drop	5	InnoDB	utf8mb4_general_ci
□	login	☆ Browse Structure Search Insert Empty Drop	7	InnoDB	utf8mb4_general_ci
	2 tables	Sum	12	InnoDB	latin1_swedish_ci

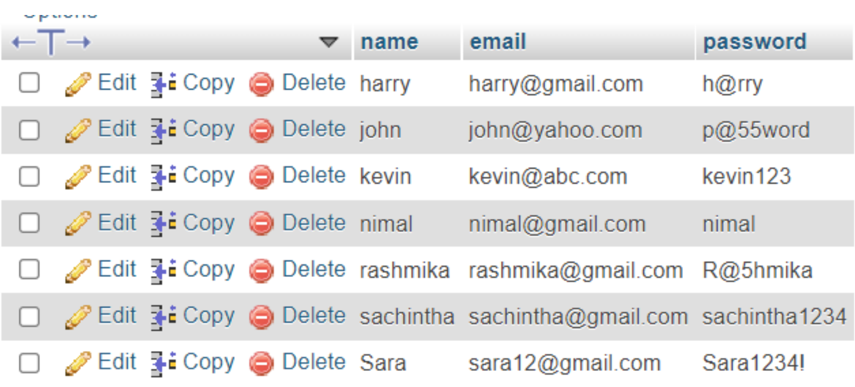


+ Options

		level	flag_id	flag
□	Edit Copy Delete	level1	flag1	we are being hijacked!
□	Edit Copy Delete	level3	flag3.1	Hezbo
□	Edit Copy Delete	level3	flag3.2	Makers of God's world.
□	Edit Copy Delete	level4	flag4	salamiyah_camp
□	Edit Copy Delete	level5	flag5	amadeus

Table 1.0

User login table



		name	email	password
□	Edit Copy Delete	harry	harry@gmail.com	h@rry
□	Edit Copy Delete	john	john@yahoo.com	p@55word
□	Edit Copy Delete	kevin	kevin@abc.com	kevin123
□	Edit Copy Delete	nimal	nimal@gmail.com	nimal
□	Edit Copy Delete	rashmika	rashmika@gmail.com	R@5hmika
□	Edit Copy Delete	sachintha	sachintha@gmail.com	sachintha1234
□	Edit Copy Delete	Sara	sara12@gmail.com	Sara1234!

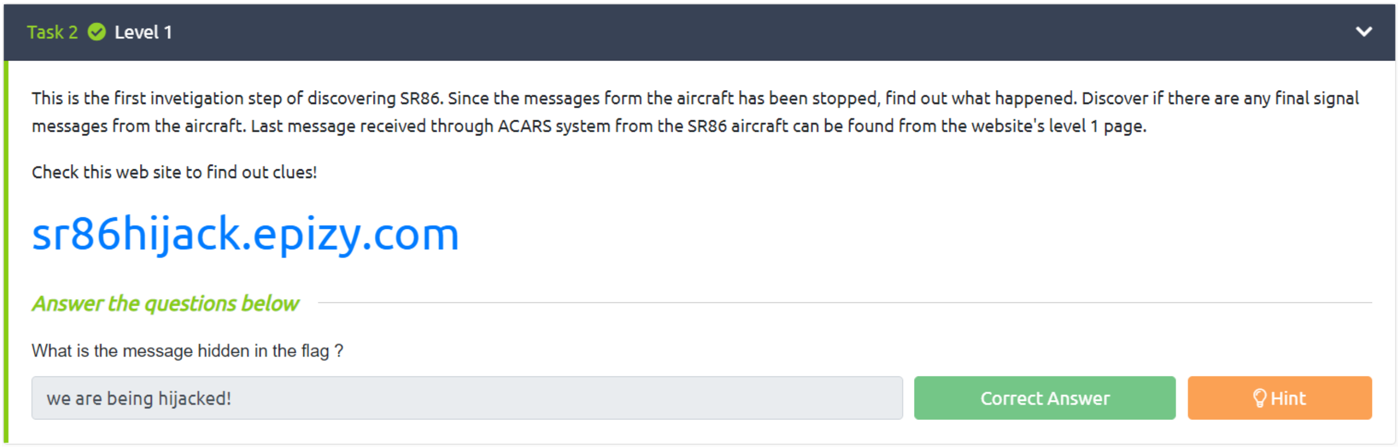
Table 2.0

These tables manage the user logins and the answer submissions from the web application end of the CTF development.

Level 1

The objective in this level is to find the hidden flag in the level 1 web page source code. The flag is base 64 encoded.

As a Hint base 64 decoder links are provided



Task 2 ✓ Level 1

This is the first invetigation step of discovering SR86. Since the messages form the aircraft has been stopped, find out what happened. Discover if there are any final signal messages from the aircraft. Last message received through ACARS system from the SR86 aircraft can be found from the website's level 1 page.

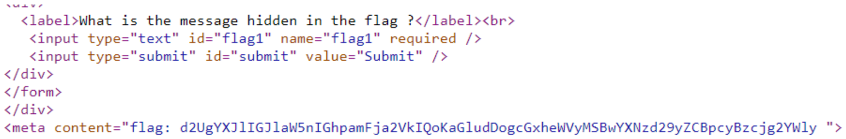
Check this web site to find out clues!

sr86hijack.epizy.com

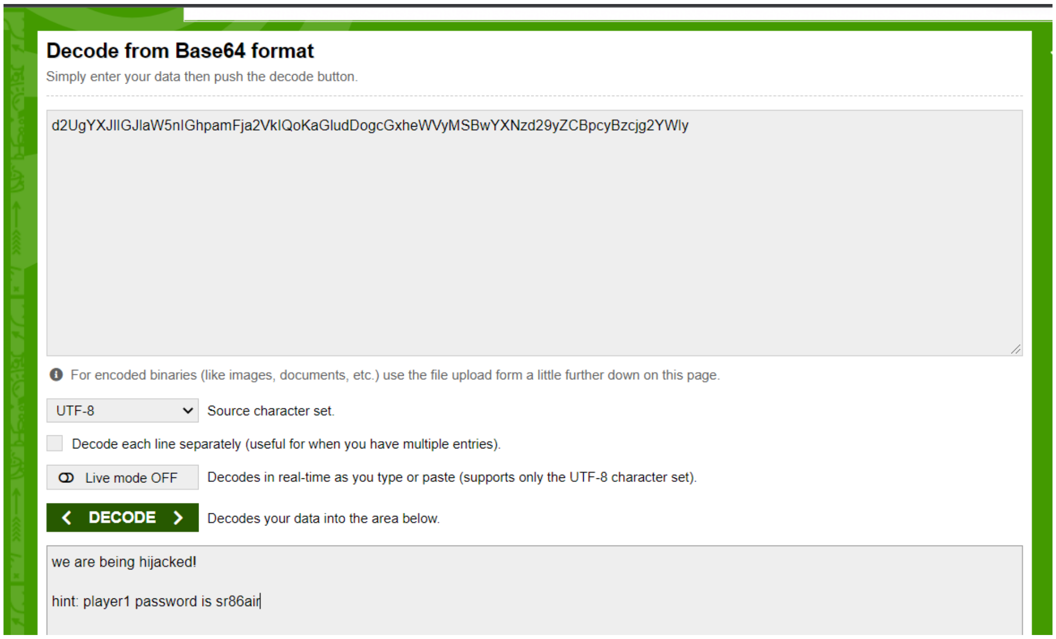
Answer the questions below

What is the message hidden in the flag ?

we are being hijacked!



```
  <label>What is the message hidden in the flag ?</label><br>
   <input type="text" id="flag1" name="flag1" required />
   <input type="submit" id="submit" value="Submit" />
</div>
</form>
</div>
<meta content="flag: d2UgYXJlIGJlaW5nIGhpamFja2VkIQoKaGludDogcGxheWVyMSBwYXNzd29yZCBpcyBzcjg2YWly ">
```



Decode from Base64 format

Simply enter your data then push the decode button.

```
d2UgYXJlIGJlaW5nIGhpamFja2VkIQoKaGludDogcGxheWVyMSBwYXNzd29yZCBpcyBzcjg2YWly
```

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.

```
we are being hijacked!

hint: player1 password is sr86air
```

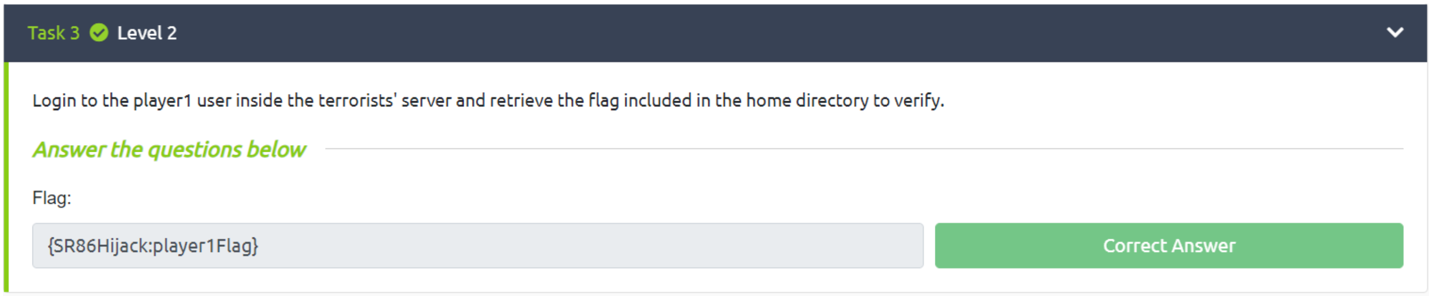
Questions:

What is the message hidden in the flag?
we are being hijacked!

Level 2

Player needs to connect to the player 1 user account in the ubuntu server using the discovered credentials in the previous level.

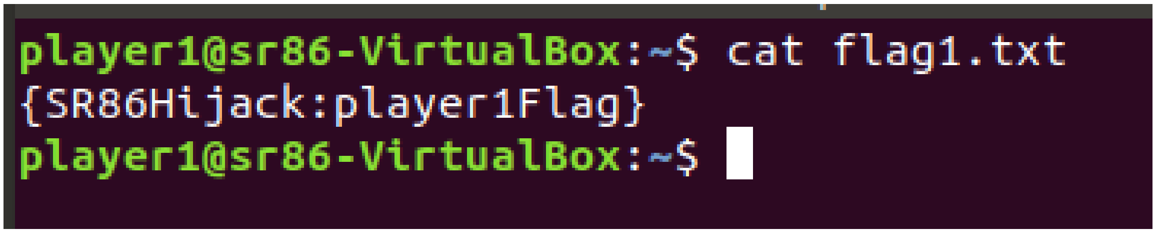
After established a SSH connection player need to submit the flag for the Player 1.



Login to the player1 user inside the terrorists' server and retrieve the flag included in the home directory to verify.

Answer the questions below

Flag: {SR86Hijack:player1Flag}



```
player1@sr86-VirtualBox:~$ cat flag1.txt
{SR86Hijack:player1Flag}
player1@sr86-VirtualBox:~$
```

Flag:

{SR86Hijack:player1Flag}

Level 3

In the level 3, player is already logged in to terrorist group system using the player1 user account. In this level, player needs to find the terrorist group's name as well as their motto by steganography tool that is provided as a hint.

Staganography tool: https://stylesuxx.github.io/steganography/

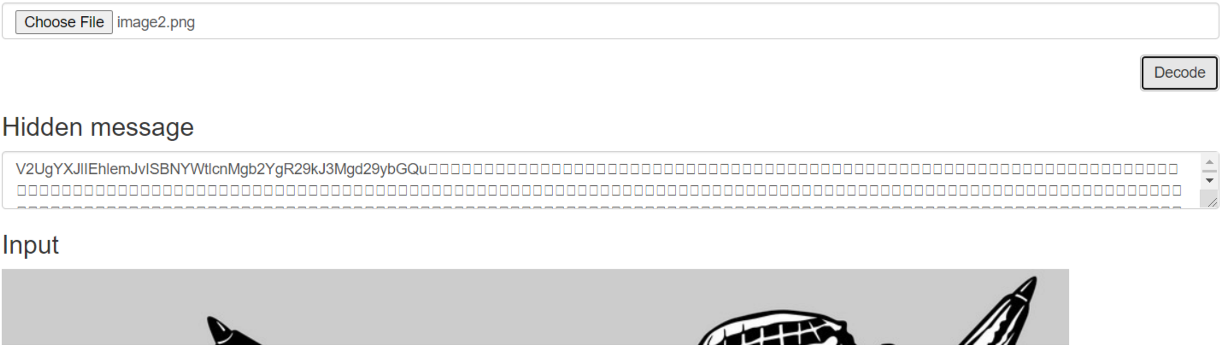


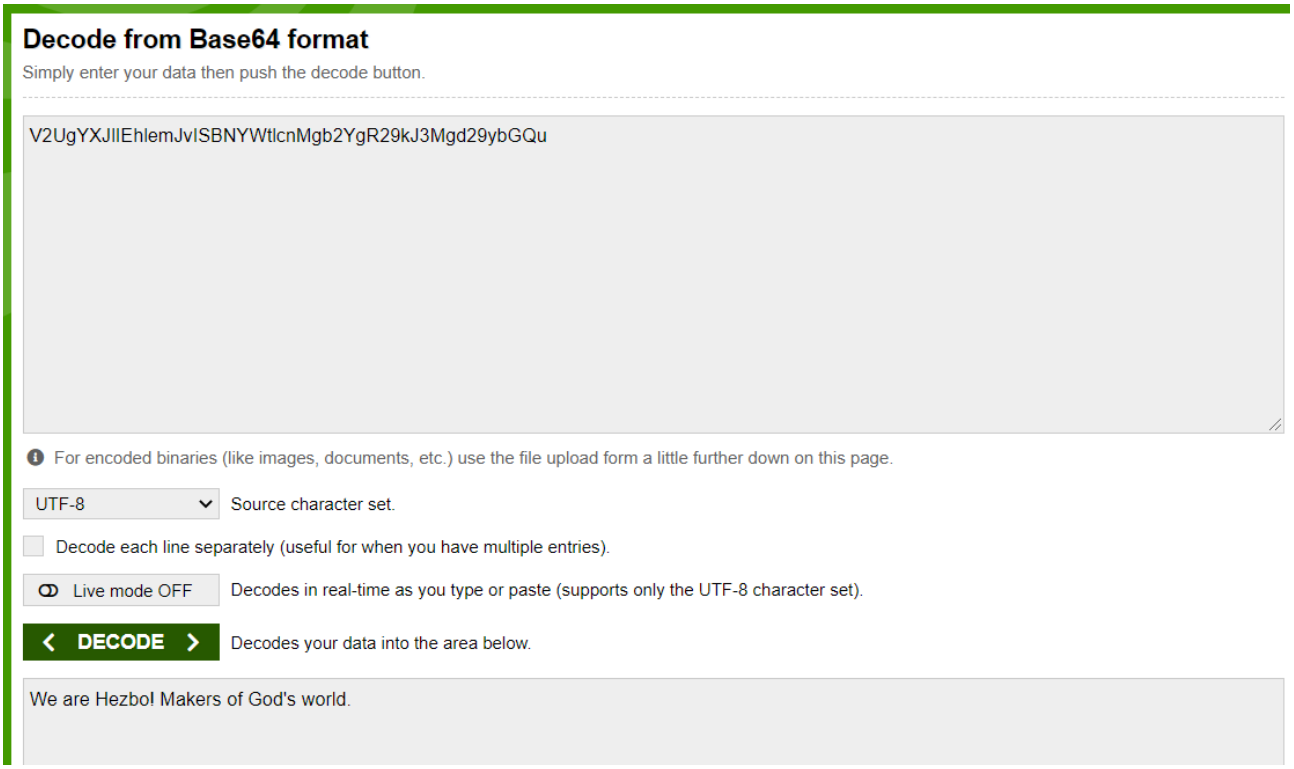
image2.png

Hidden message

```
V2UgYXJlIEhlemJvISBNYWtlcnMgb2YgR29kJ3Mgd29ybGQu□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
```

Input

Then the player will have to decrypt the base64 encoded message retrieved from the stagonography task.



Decode from Base64 format

Simply enter your data then push the decode button.

```
V2UgYXJlIEhlemJvISBNYWtlcnMgb2YgR29kJ3Mgd29ybGQu
```

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.

```
We are Hezbo! Makers of God's world.
```

Questions:

What is the Terrorist group's name?
Hezbo

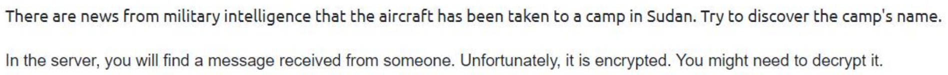
What's their motto?
Makers of God's world

Level 4

In the level 4, player needs to find a hidden message in the level 4 directory which was received. This message is encoded using Caesar cipher. An image of Caesar is provided as a hint for players to guess which cryptographic algorithm to be used when decoding the message.

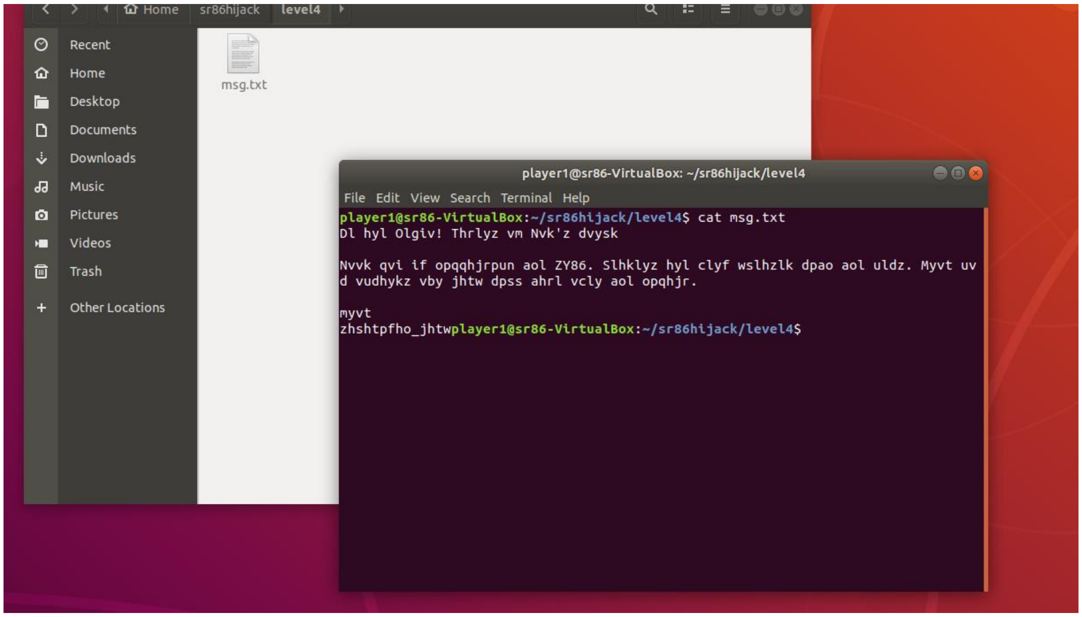
In addition, a hint is provided to use Cryptool as the decoding tool.

Task 5 ✓ Level 4



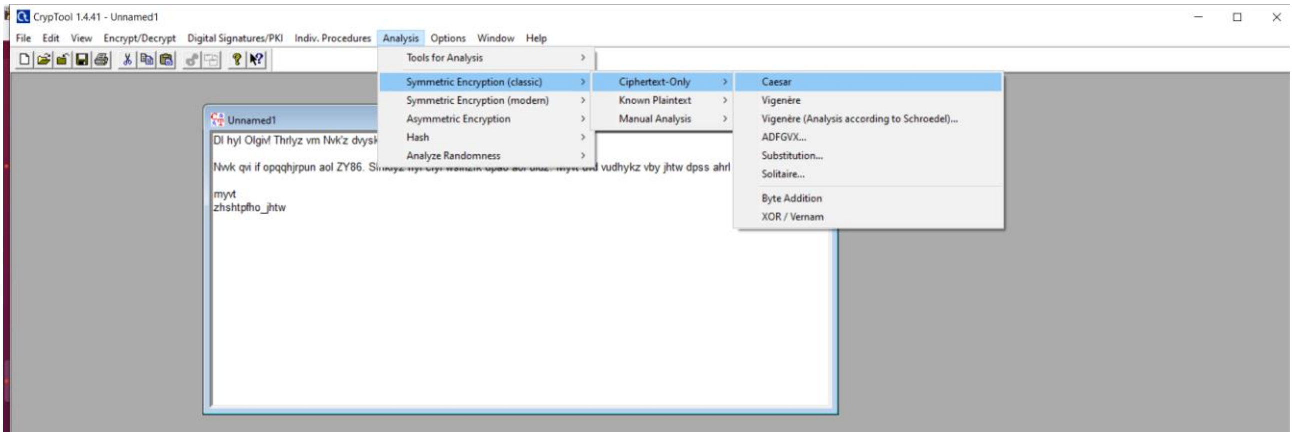
There are news from military intelligence that the aircraft has been taken to a camp in Sudan. Try to discover the camp's name.

In the server, you will find a message received from someone. Unfortunately, it is encrypted. You might need to decrypt it.



```
player1@sr86-VirtualBox:~/sr86hijack/level4$ cat msg.txt
Dl hyl Olgiv! Thrlyz vm Nvk'z dvysk

Nvvk qvi if opqqhjrpun aol ZY86. Slhklyz hyl clyf wslhzlk dpao aol uldz. Myvt uv
d vudhykz vby jhtw dpss ahrl vcly aol opqhjr.

myvt
zhshtpfho_jhtwplayer1@sr86-VirtualBox:~/sr86hijack/level4$
```

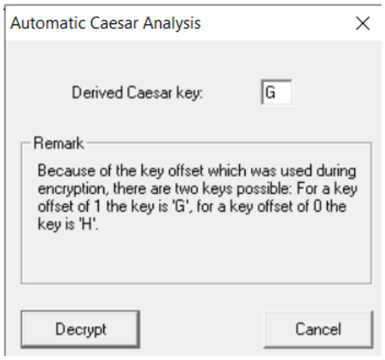
Players must use the frequency analysis to discover the cipher key that is used to encrypt the message


CrypTool 1.4.41 - Unnamed1
File Edit View Encrypt/Decrypt Digital Signatures/PKI Indiv. Procedures Analysis Options Window Help
Tools for Analysis
Symmetric Encryption (classic)
Symmetric Encryption (modern)
Asymmetric Encryption
Hash
Analyze Randomness
Ciphertext-Only
Known Plaintext
Manual Analysis
Caesar
Vigenère
Vigenère (Analysis according to Schroedel)...
ADFGVX...
Substitution...
Solitaire...
Byte Addition
XOR / Vernam
Unnamed1

Dl hyl Olgiv! Thrlyz vm Nvk'z dvysk

Nwk qvi if opqqhjrpun aol ZY86. Si vudhykz vby jhtw dpss ahrl

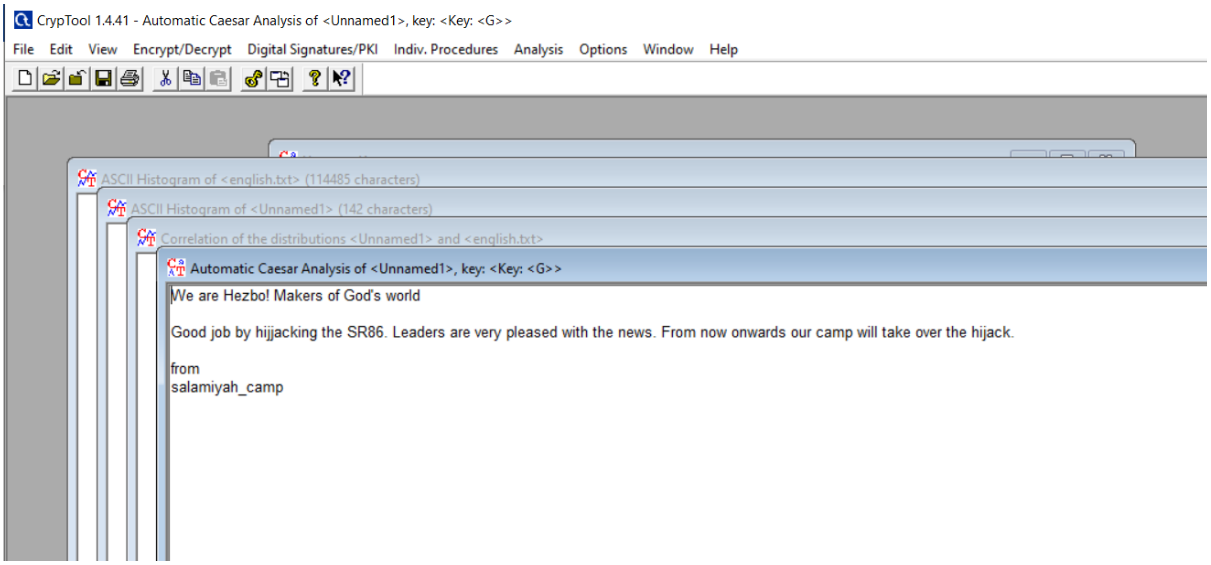
myvt
zhshtpfho_jhtw



Derived Caesar key: G

Remark

Because of the key offset which was used during encryption, there are two keys possible: For a key offset of 1 the key is 'G', for a key offset of 0 the key is 'H'.


CrypTool 1.4.41 - Automatic Caesar Analysis of <Unnamed1>, key: <Key: <G>>
File Edit View Encrypt/Decrypt Digital Signatures/PKI Indiv. Procedures Analysis Options Window Help
ASCII Histogram of <english.txt> (114485 characters)
ASCII Histogram of <Unnamed1> (142 characters)
Correlation of the distributions <Unnamed1> and <english.txt>
Automatic Caesar Analysis of <Unnamed1>, key: <Key: <G>>

We are Hezbo! Makers of God's world

Good job by hijjacking the SR86. Leaders are very pleased with the news. From now onwards our camp will take over the hijack.

from
salamiyah_camp

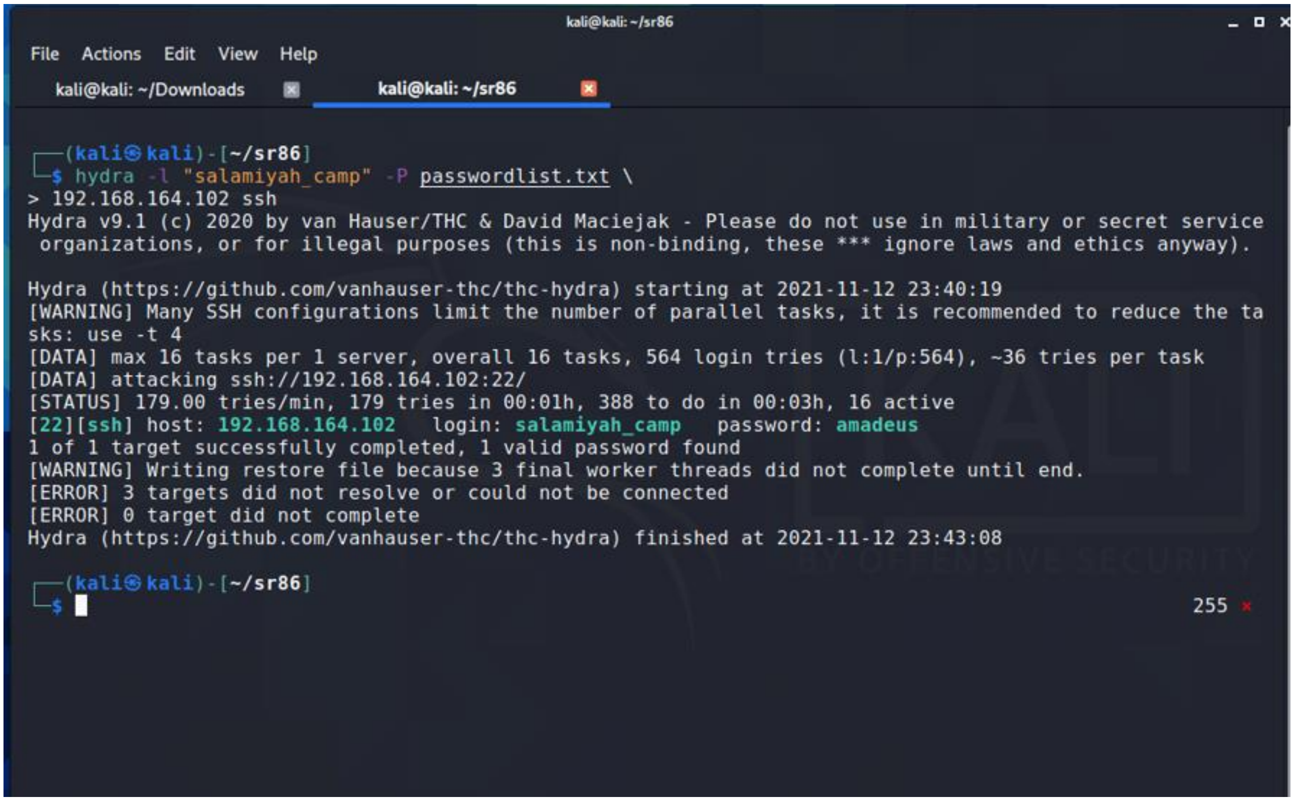
Questions:

What is the username discovered?

salamiyah_camp

Level 5

In the level 5, need to find password for login to another account known as salamiyah_camp that was discovered in the previous level. Player needs to use tool **Hydra** for the password brute forcing attack in order to discover the password for the salamiyah_camp.



```
┌──(kali㉿kali)-[~/sr86]
└─$ hydra -l "salamiyah_camp" -P passwordlist.txt \
> 192.168.164.102 ssh
Hydra v9.1 (c) 2020 by van Hauser/THC & David Maciejak - Please do not use in military or secret service
 organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2021-11-12 23:40:19
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the ta
sks: use -t 4
[DATA] max 16 tasks per 1 server, overall 16 tasks, 564 login tries (l:1/p:564), ~36 tries per task
[DATA] attacking ssh://192.168.164.102:22/
[STATUS] 179.00 tries/min, 179 tries in 00:01h, 388 to do in 00:03h, 16 active
[22][ssh] host: 192.168.164.102   login: salamiyah_camp   password: amadeus
1 of 1 target successfully completed, 1 valid password found
[WARNING] Writing restore file because 3 final worker threads did not complete until end.
[ERROR] 3 targets did not resolve or could not be connected
[ERROR] 0 target did not complete
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2021-11-12 23:43:08

┌──(kali㉿kali)-[~/sr86]
└─$ █                                                                                              255 ×
```

Password list that is needed for the Hydra is provided as a hint in the level description.
Players need to retieve the flag that is created for the salamiyah_camp and provide it as an answer.

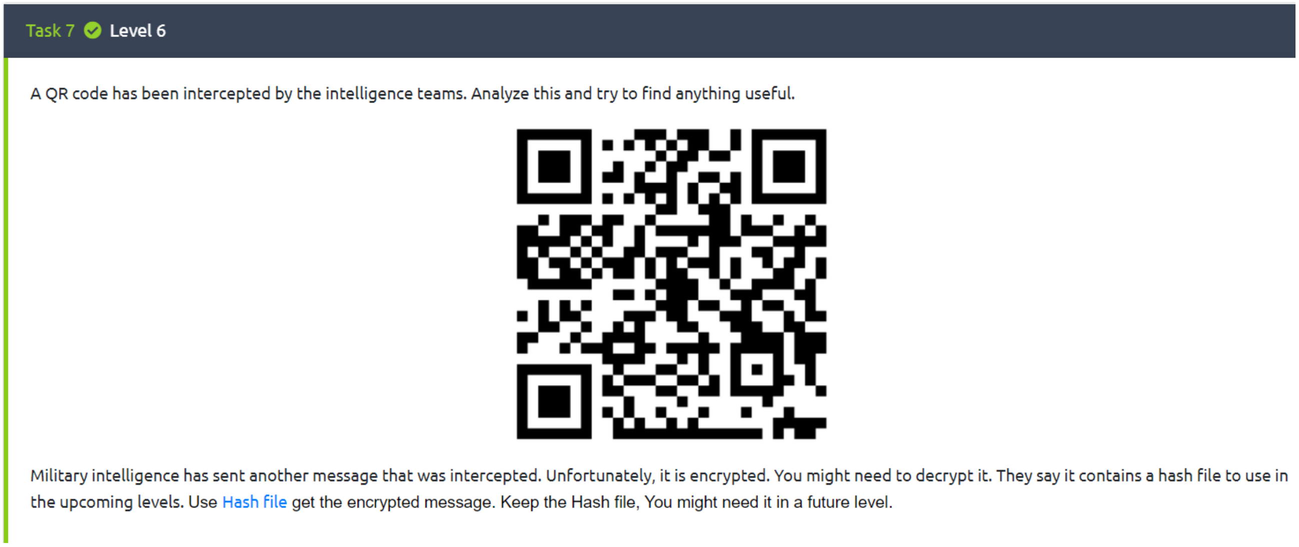
Questions:

What is the discovered password for the user ?
amdeus

Flag:
{SR86Hijack:salamiyah_campFlag}

Level 6

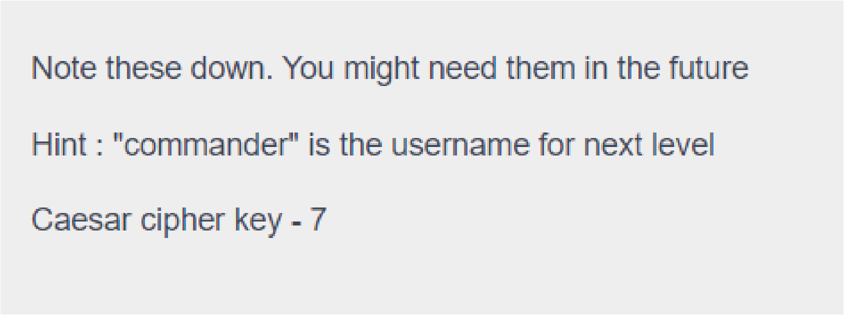
In the level 6, player must discover a username for login to the next account named as commander, which is a user account belongs to a higher-ranking terrorist. Player should scan the provided QR codes and retrieve the hidden message and discover the username.



A QR code has been intercepted by the intelligence teams. Analyze this and try to find anything useful.

Military intelligence has sent another message that was intercepted. Unfortunately, it is encrypted. You might need to decrypt it. They say it contains a hash file to use in the upcoming levels. Use Hash file get the encrypted message. Keep the Hash file, You might need it in a future level.

https://me-qr.com/text/105042/show

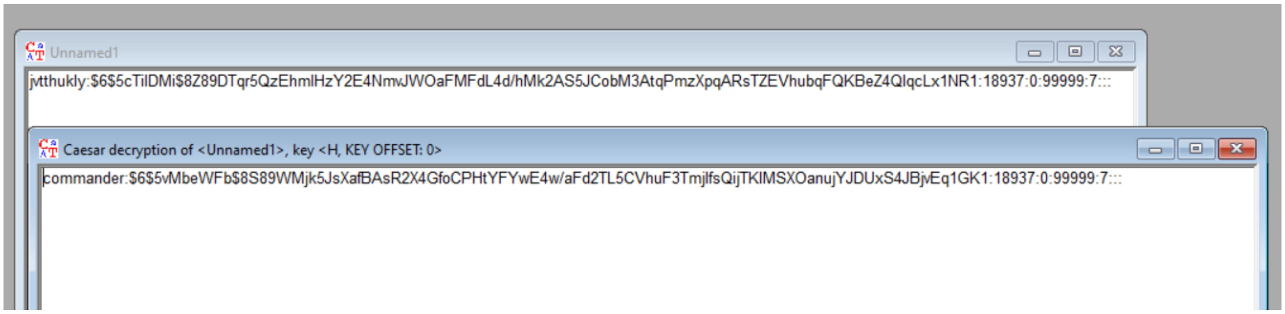


Note these down. You might need them in the future

Hint : "commander" is the username for next level

Caesar cipher key - 7

Along with the username, a Caesar cipher key is provided which is needed to decrypt a Hash file using Cryptool. Player should decrypt the provided Hash file using this key and retrieve the decrypted hash file to use in the next level.



```
jvtthukly:$6$5cTiIDMi$8Z89DTqr5QzEhmlHzY2E4NmwJWOaFMFdL4d/hMk2AS5JCobM3AtqPmzXpqARsTZEVhubqFQKBeZ4QIqcLx1NR1:18937:0:99999:7:::
```

```
commander:$6$5vMbeWFb$8S89WMjk5JsXafBAsR2X4GfoCPHtYFYwE4w/aFd2TL5CVhuF3TmjlfsQijTKIMSXOanujYJDUxS4JBjvEq1GK1:18937:0:99999:7:::
```

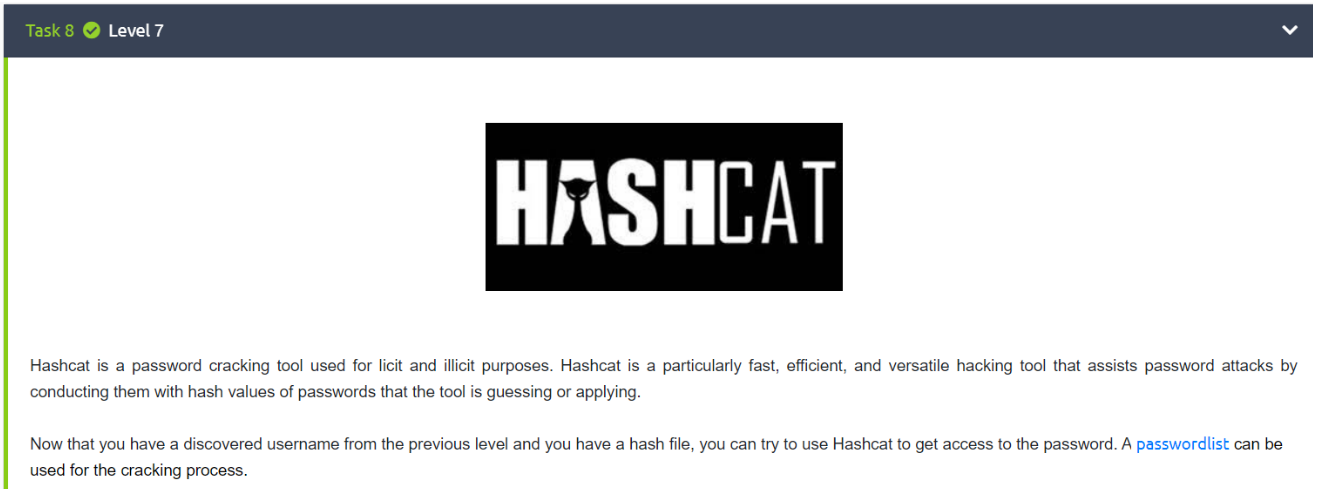
Player will have to keep the decrypted hash file to use in the next level.

Question:

What is the username discovered?
Commander

Level 7

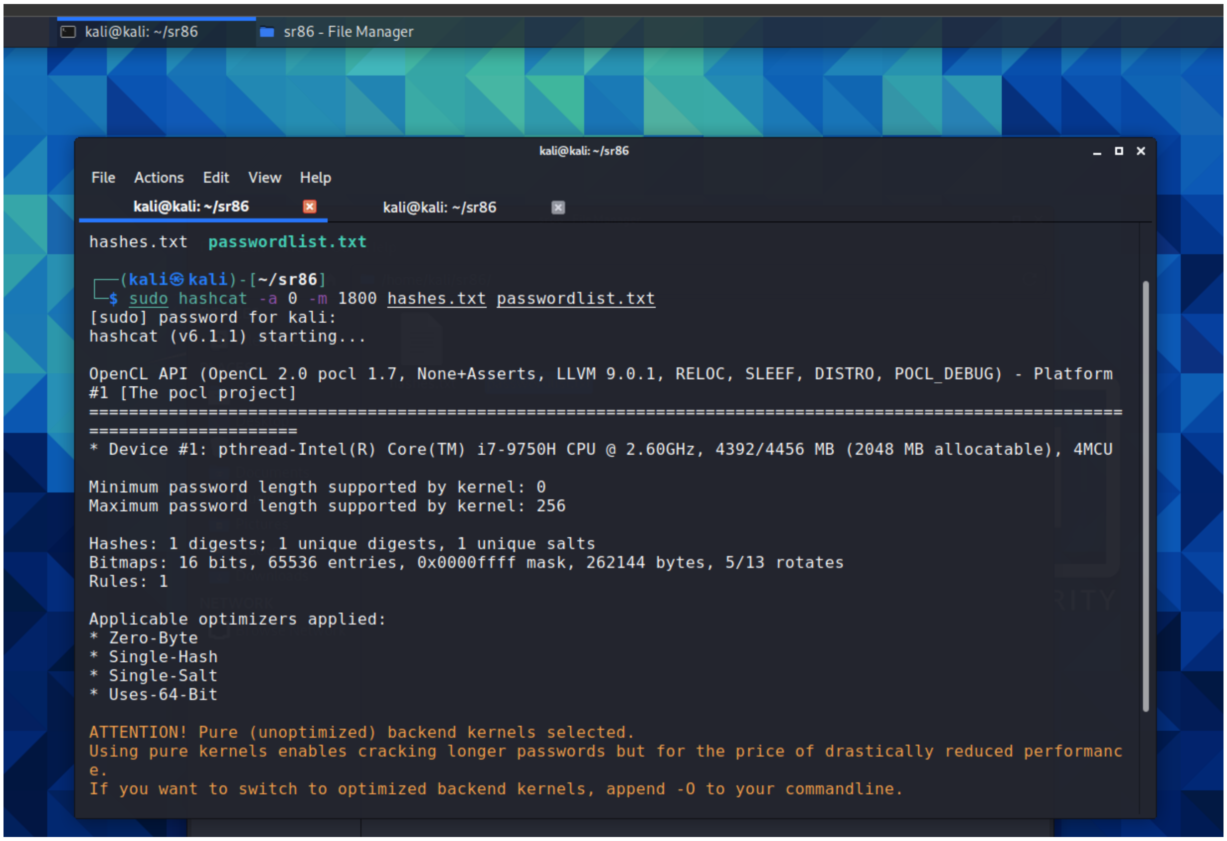
In the level 7, players need to find the password to login to the next account which is known as commander which was discovered in the level 6. Player needs to use the **Hashcat** tool for to discover the password of the commander account.



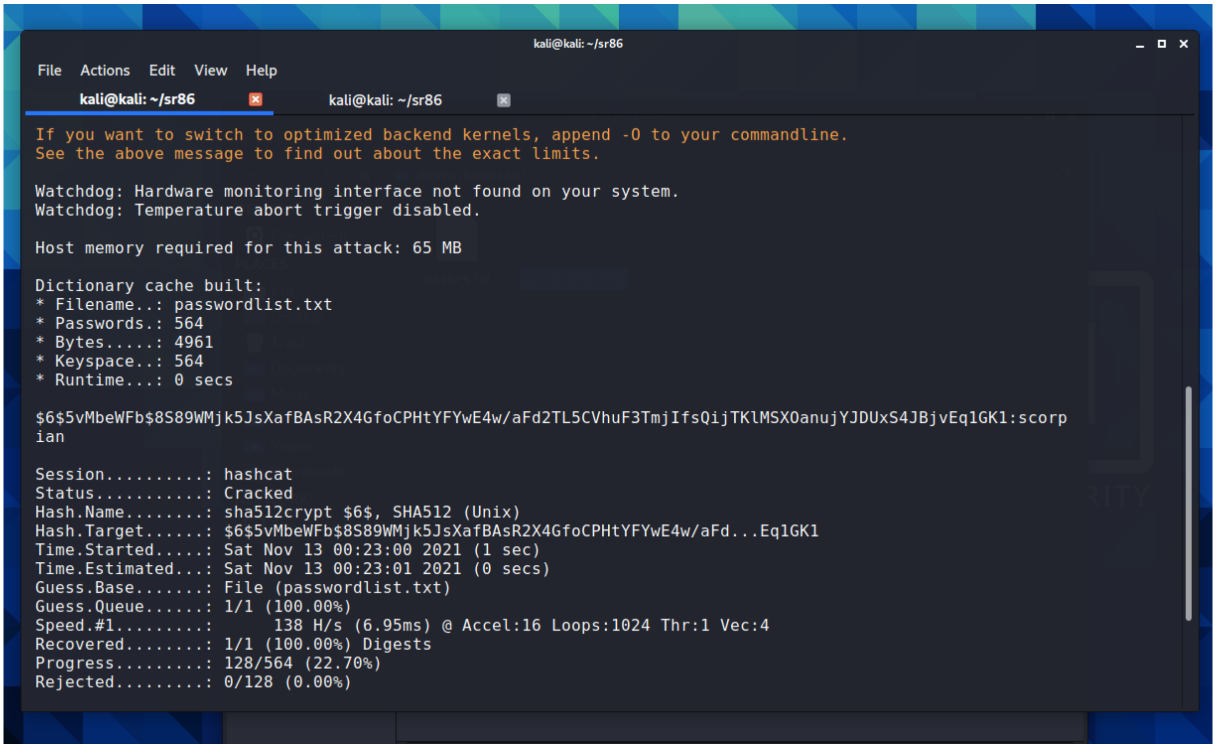
Hashcat is a password cracking tool used for licit and illicit purposes. Hashcat is a particularly fast, efficient, and versatile hacking tool that assists password attacks by conducting them with hash values of passwords that the tool is guessing or applying.

Now that you have a discovered username from the previous level and you have a hash file, you can try to use Hashcat to get access to the password. A passwordlist can be used for the cracking process.

Password list that is needed for the Hashcat is provided as a hint in the level description.



```
hashes.txt  passwordlist.txt

┌──(kali㉿kali)-[~/sr86]
└─$ sudo hashcat -a 0 -m 1800 hashes.txt passwordlist.txt
[sudo] password for kali:
hashcat (v6.1.1) starting...

OpenCL API (OpenCL 2.0 pocl 1.7, None+Asserts, LLVM 9.0.1, RELOC, SLEEF, DISTRO, POCL_DEBUG) - Platform
#1 [The pocl project]
==========================================================================================================
=====================
* Device #1: pthread-Intel(R) Core(TM) i7-9750H CPU @ 2.60GHz, 4392/4456 MB (2048 MB allocatable), 4MCU

Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 256

Hashes: 1 digests; 1 unique digests, 1 unique salts
Bitmaps: 16 bits, 65536 entries, 0x0000ffff mask, 262144 bytes, 5/13 rotates
Rules: 1

Applicable optimizers applied:
* Zero-Byte
* Single-Hash
* Single-Salt
* Uses-64-Bit

ATTENTION! Pure (unoptimized) backend kernels selected.
Using pure kernels enables cracking longer passwords but for the price of drastically reduced performanc
e.
If you want to switch to optimized backend kernels, append -O to your commandline.
```



```
If you want to switch to optimized backend kernels, append -O to your commandline.
See the above message to find out about the exact limits.

Watchdog: Hardware monitoring interface not found on your system.
Watchdog: Temperature abort trigger disabled.

Host memory required for this attack: 65 MB

Dictionary cache built:
* Filename..: passwordlist.txt
* Passwords.: 564
* Bytes.....: 4961
* Keyspace..: 564
* Runtime...: 0 secs

$6$5vMbeWFb$8S89WMjk5JsXafBAsR2X4GfoCPHtYFYwE4w/aFd2TL5CVhuF3TmjIfsQijTKlMSXOanujYJDUxS4JBjvEq1GK1:scorp
ian

Session..........: hashcat
Status...........: Cracked
Hash.Name........: sha512crypt $6$, SHA512 (Unix)
Hash.Target......: $6$5vMbeWFb$8S89WMjk5JsXafBAsR2X4GfoCPHtYFYwE4w/aFd...Eq1GK1
Time.Started.....: Sat Nov 13 00:23:00 2021 (1 sec)
Time.Estimated...: Sat Nov 13 00:23:01 2021 (0 secs)
Guess.Base.......: File (passwordlist.txt)
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........:      138 H/s (6.95ms) @ Accel:16 Loops:1024 Thr:1 Vec:4
Recovered........: 1/1 (100.00%) Digests
Progress.........: 128/564 (22.70%)
Rejected.........: 0/128 (0.00%)
```

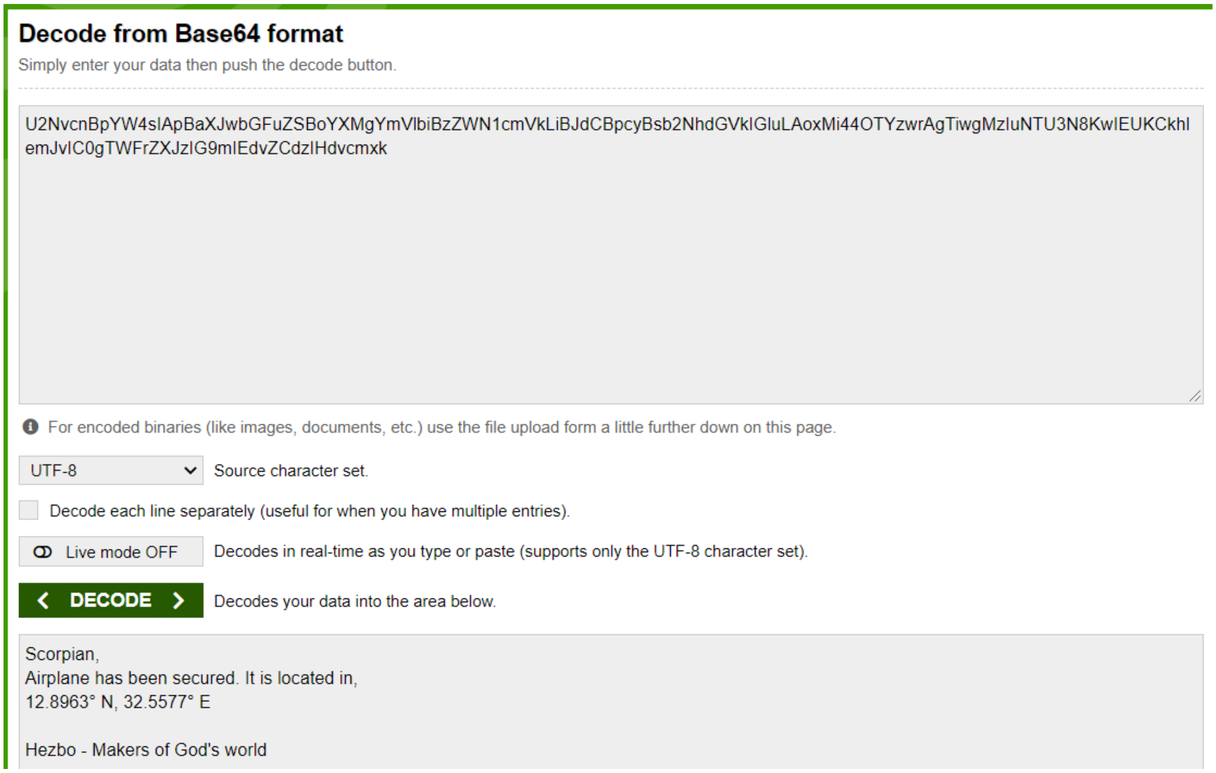
Questions:

What is the password discovered?
Scorpian

Flag:
{SR86Hijack:commanderFlag}

Level 8

In the level 8, player will have to decrypt the base64 encoded message and retrieve the GPS codes in the hidden message which stored in commander account.



Decode from Base64 format

Simply enter your data then push the decode button.

U2NvcnBpYW4sIApBaXJwbGFuZSBoYXMgYmVlbiBzZWN1cmVkLiBJdCBpcyBsb2NhdGVkIGluLAoxMi44OTYzwrAgTiwgMzIuNTU3N8KwIEUKCkhl
emJvIC0gTWFrZXJzIG9mIEdvZCdzIHdvcmxk

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

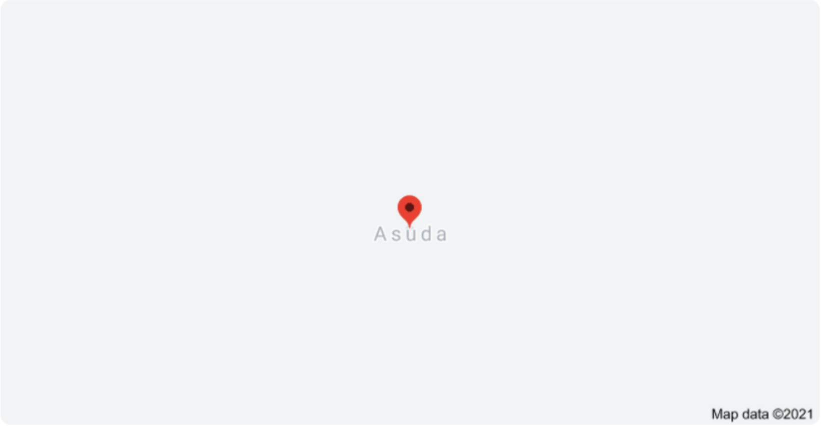
UTF-8 Source character set.

□ 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.

```
Scorpian,
Airplane has been secured. It is located in,
12.8963° N, 32.5577° E

Hezbo - Makers of God's world
```



Map for 12.8963° N, 32.5577° E

Questions:

What are the GPS coordinates discovered?
12.8963° N, 32.5577° E

What is the name of the location that the aircraft is hidden?
Asuda

After discovering the location of the aircraft. The assigned tasks for the SR86 Hijack CTF challenge will be completed.

2.4 Testing

Different variations of tests were conducted in order to test the functionalities of the developments in the CTF. Evidence that were collected are displayed in the Appendix A : test results section.

I. **TryHackMe** Virtual machine connection
II. Answer submissions
III. Web application's connection with the back end

3. Evaluation

3.1 Assessment of the Project results

Scoreboards were used to analyze the project results. In the TryHackMe implementation, a score board is used to store the player progresses and their relevant scores. Using that dashboard, an analysis could be performed based on player progresses.

3.2 Lessons Learned

Many lessons were learned while developing the SR86 Hijack CTF challenges. Initially the development of the CTF was began with minimum knowledge on CTF challenges and the tools and technologies used when creating CTF challenges. Overcoming the skill gap and fulfil the requirements of the CTF development increased the group members knowledge and as well as the confidence.

3.3 Future Work

In the future, this development can be modified towards the level of a real-world industry based aviation security investigation training material. Usage of industrial systems would improve the functionality of the CTF challenges, and it will maximize the output that is received from such challenges.

4. Conclusion

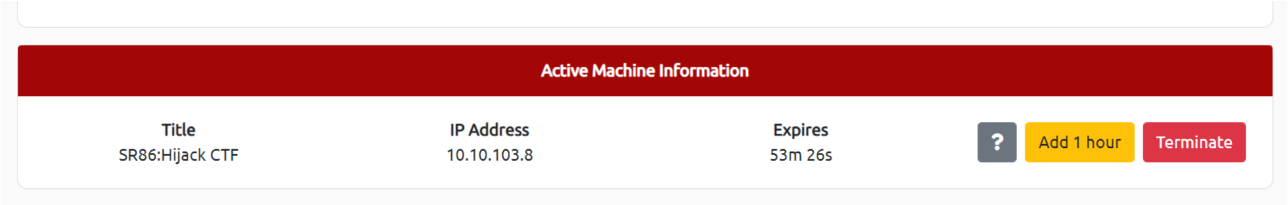
This CTF development contained a CTF challenge that was designed based on a hypothetical scenario regarding the aviation industry which is considered as an industry that uses highly critical systems. The purpose of designing a CTF with such a scenario was to attract players who are interested in aviation related incidents and military intervention related incidents. In addition, training and awareness for the aviation related staff and military staff was one of the main goals of the CTF design. Through the development it is concluded the objectives are achieved towards a certain level.

Although the intended objectives were achieved, there are visible limitations to the system developed. Lack of industry level aviation-based systems, secure servers are considered to be some of the limitations identified.

In future research on a CTF that is based on aviation ACARS systems, these identified shortcomings can be eliminated or minimized. Through such a development, the reach of the CTF challenge players will be able to increase. In addition, it will be able to be used in the industry level investigation trainings. In conclusion, as an overall beginner level CTF challenge, it is concluded that the SR86 Hijack achieved its intended objectives.

Appendix A: Test Results

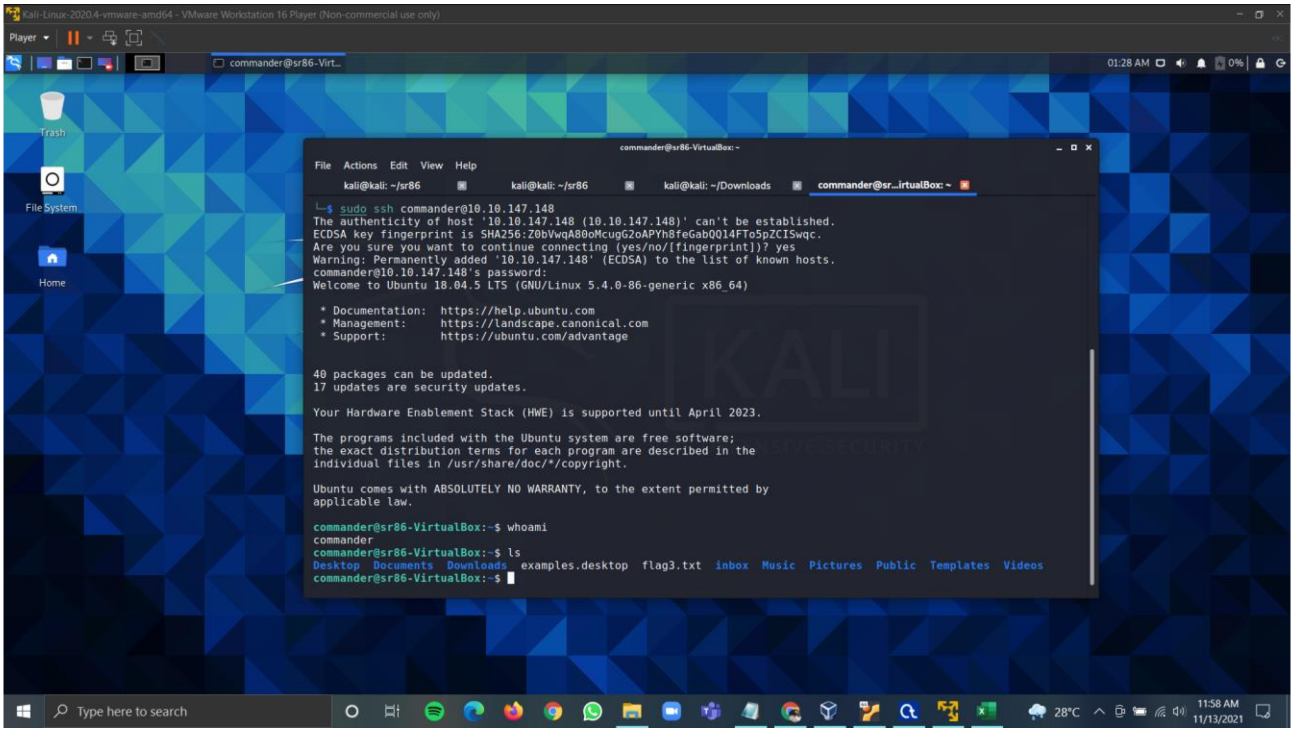
I. TryHackMe VM connection results



Active Machine Information

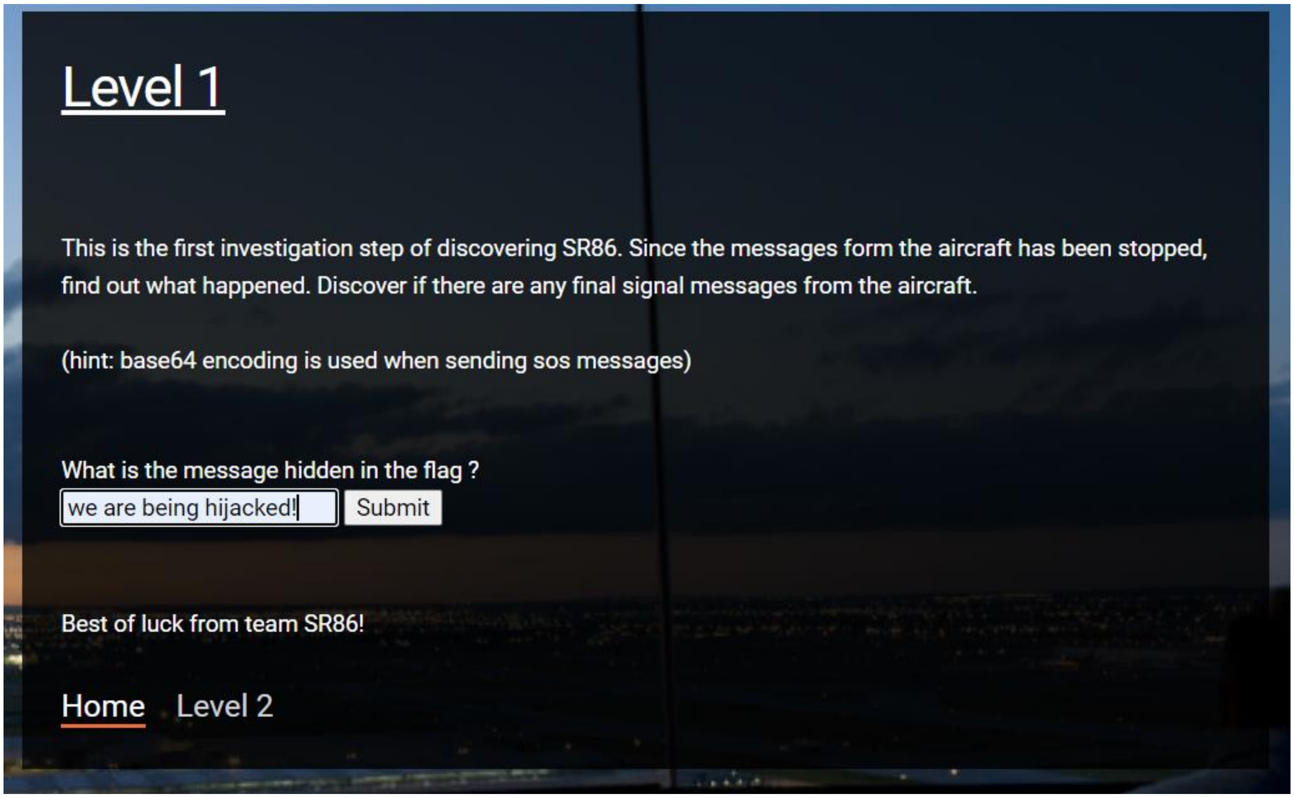
Title	IP Address	Expires
SR86:Hijack CTF	10.10.103.8	53m 26s

ssh connection established between the attack machine and the TryHackMe VM



```
└─$ sudo ssh commander@10.10.147.148
The authenticity of host '10.10.147.148 (10.10.147.148)' can't be established.
ECDSA key fingerprint is SHA256:Z0bVwqA80oMcugG2oAPYh8feGabQQ14FTo5pZCISwqc.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.147.148' (ECDSA) to the list of known hosts.
commander@10.10.147.148's password:
Welcome to Ubuntu 18.04.5 LTS (GNU/Linux 5.4.0-86-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage


40 packages can be updated.
17 updates are security updates.

Your Hardware Enablement Stack (HWE) is supported until April 2023.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

commander@sr86-VirtualBox:~$ whoami
commander
commander@sr86-VirtualBox:~$ ls
Desktop  Documents  Downloads  examples.desktop  flag3.txt  inbox  Music  Pictures  Public  Templates  Videos
commander@sr86-VirtualBox:~$ 
```

II. Answer submissions



Level 1

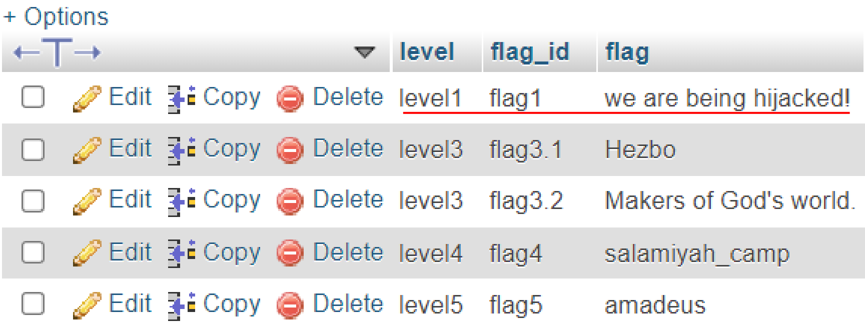
This is the first investigation step of discovering SR86. Since the messages form the aircraft has been stopped, find out what happened. Discover if there are any final signal messages from the aircraft.

(hint: base64 encoding is used when sending sos messages)

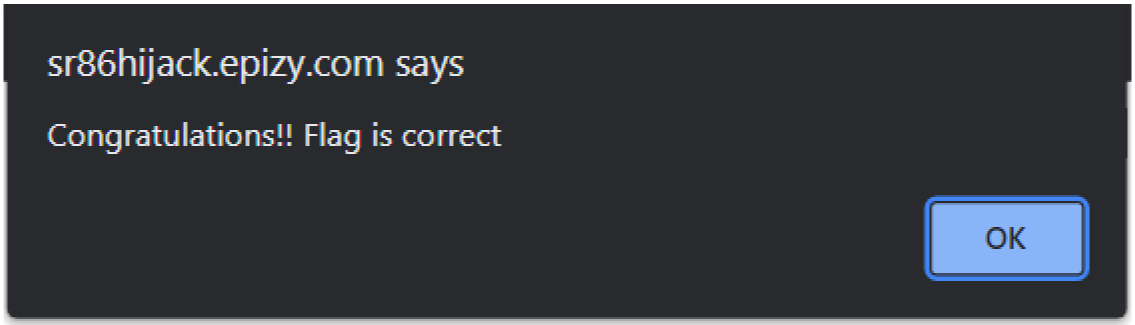
What is the message hidden in the flag ?

we are being hijacked!

Best of luck from team SR86!



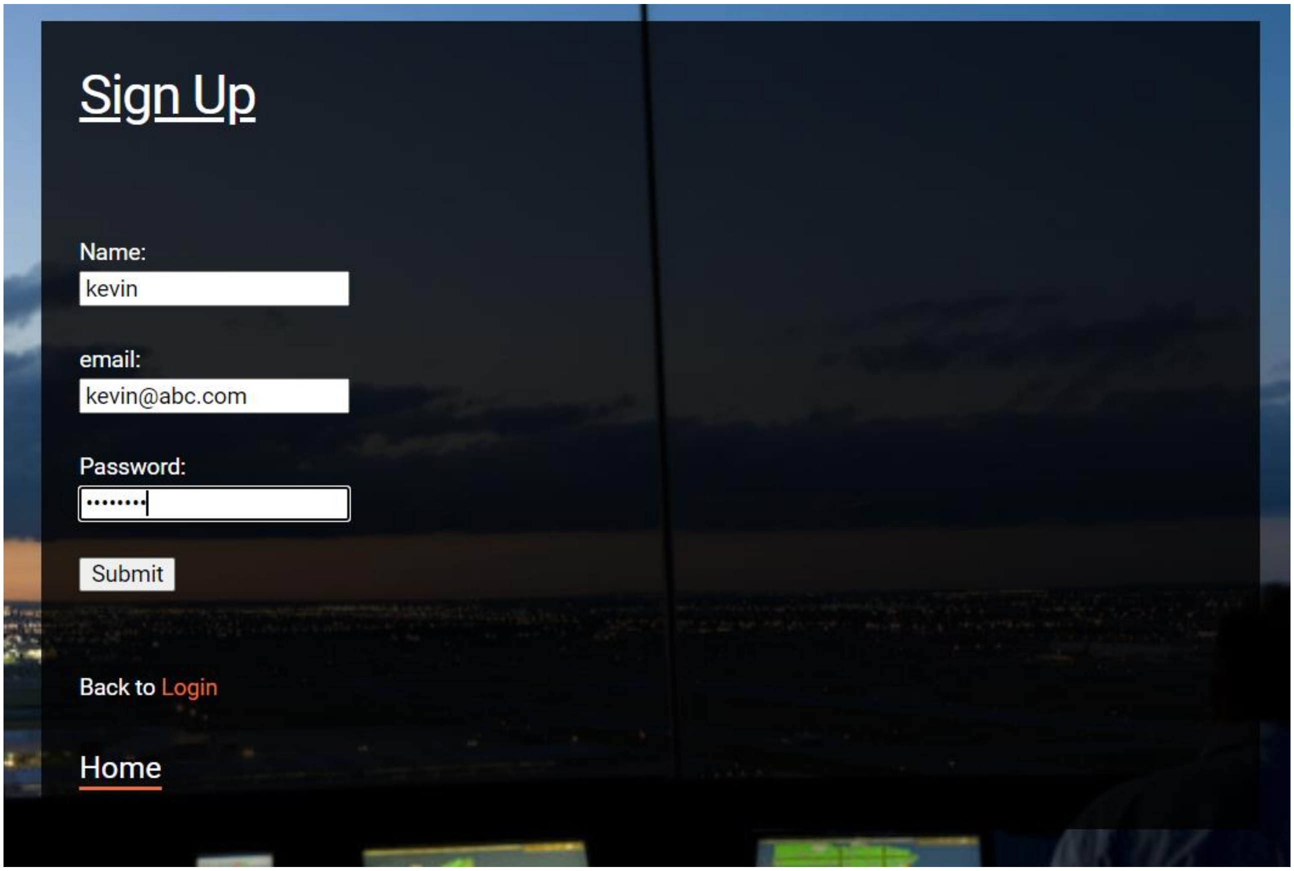
←T→				level	flag_id	flag
□	Edit	Copy	Delete	level1	flag1	we are being hijacked!
□	Edit	Copy	Delete	level3	flag3.1	Hezbo
□	Edit	Copy	Delete	level3	flag3.2	Makers of God's world.
□	Edit	Copy	Delete	level4	flag4	salamiyah_camp
□	Edit	Copy	Delete	level5	flag5	amadeus



sr86hijack.epizy.com says

Congratulations!! Flag is correct

III. Web application’s connection with the database backend



Sign Up

Name:
kevin

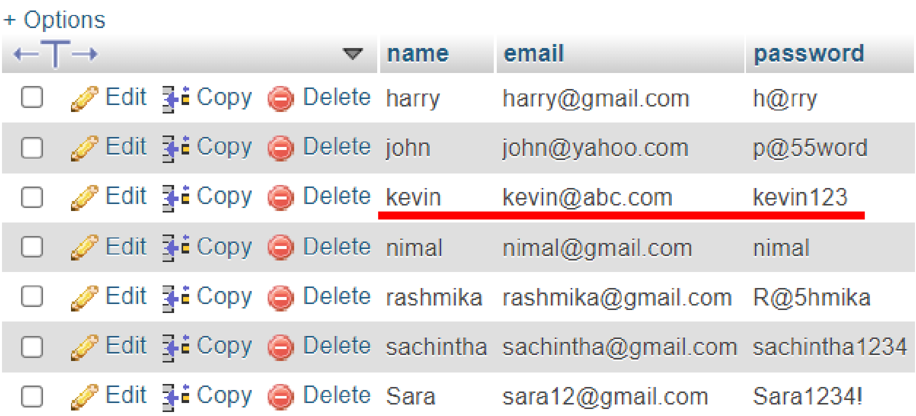
email:
kevin@abc.com

Password:
••••••••

Submit

Back to Login

Home



+ Options

←T→	▼	name	email	password
☐	Edit Copy Delete	harry	harry@gmail.com	h@rry
☐	Edit Copy Delete	john	john@yahoo.com	p@55word
☐	Edit Copy Delete	kevin	kevin@abc.com	kevin123
☐	Edit Copy Delete	nimal	nimal@gmail.com	nimal
☐	Edit Copy Delete	rashmika	rashmika@gmail.com	R@5hmika
☐	Edit Copy Delete	sachintha	sachintha@gmail.com	sachintha1234
☐	Edit Copy Delete	Sara	sara12@gmail.com	Sara1234!