

**FIAP - FACULDADE DE INFORMÁTICA E ADMINISTRAÇÃO PAULISTA**  
**DEFESA CIBERNETICA**



**PROVAS GS 2024**  
**CTF WARGAMES**

**RAMYREZ GUIMARÃES SANTANA**  
**553022**

**SÃO PAULO/SP**  
**2024**

## LISTA DE ILUSTRAÇÕES

- Figura 1 - E-mail.pcapng.....  
Figura 2 - E-mail.pcapng.....  
Figura 3- E-mail.pcapng.....  
Figura 4 - E-mail.pcapng.....  
Figura 5 - E-mail.pcapng.....  
Figura 6 - Pass.pcap.....  
Figura 7 - Pass.pcap.....  
Figura 8 – Pass.pcap.....  
Figura 9 – Pass.pcap.....  
Figura 10 - Apache\_2.log.....  
Figura 11 – Apache\_2.log.....  
Figura 12 - Figura 12 – Arte.jpeg.....  
Figura 13 – New\_insider.pcap.....  
Figura 14 - New\_insider.pcap.....  
Figura 15 – New\_insider.pcap.....  
Figura 16 - Index.html.. .....  
Figura 17 – Index.html.. .....  
Figura 18 – Whack.cap.....  
Figura 19 – Whack.cap.....  
Figura 20 – Whack.cap.....  
Figura 21 – Whack.cap.....  
Figura 22 – Texto.txt.....  
Figura 23 – Texto.txt.....  
Figura 24 – Texto.txt.....  
Figura 25 – Texto.txt.....  
Figura 26 – Texto.txt.....  
Figura 27 – Fiap.png.....  
Figura 28 – Apache.log .....
- Figura 29 – Apache.log .....
- Figura 30 – Key.txt.....  
Figura 31 – Key.txt.....

Figura 32 – Key.txt.....

Figura 33 – Key.txt.....

Figura 34 – Hi04.pcap. ....

Figura 35 – Hi04.pcap. ....

Figura 36 – Hi04.pcap. ....

Figura 37 – Hi04.pcap. ....

## SUMÁRIO

1.	INTRODUÇÃO.....	4
2.	BUSCANDO POR E-MAILS.....	5
3.	SENHA FORTE.....	7
4.	URL SUSPEITA .....	12
5.	PIN .....	14
6.	NEW INSIDER.....	15
7.	DEEP.....	18
8.	WI-FI HACKING .....	20
9.	BRUTE FORCE.....	24
10.	ESTRANHA IMAGEM.....	28
11.	CIFRAGEM FRACA.....	29
12.	DESENCODA .....	31

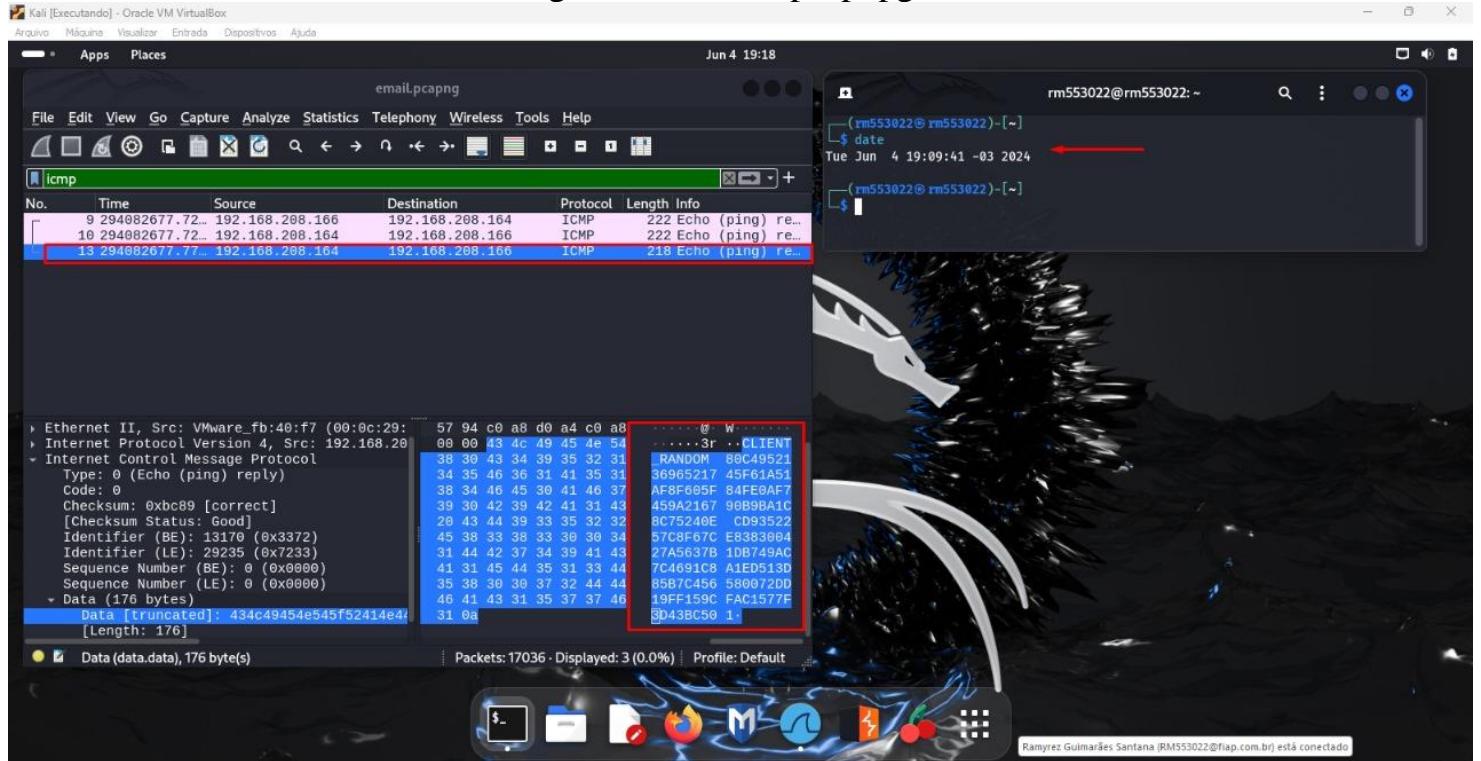
## **1. INTRODUÇÃO**

Este relatório tem como objetivo apresentar, com a ajuda de imagens de descrição detalhada, como foram solucionados os CTFs da prova GS 2024.

## 2. BUSCANDO POR E-MAILS

Neste CTF, tinha como o objetivo de encontrar o remetente (“FROM”), do e-mail presente no tráfego.

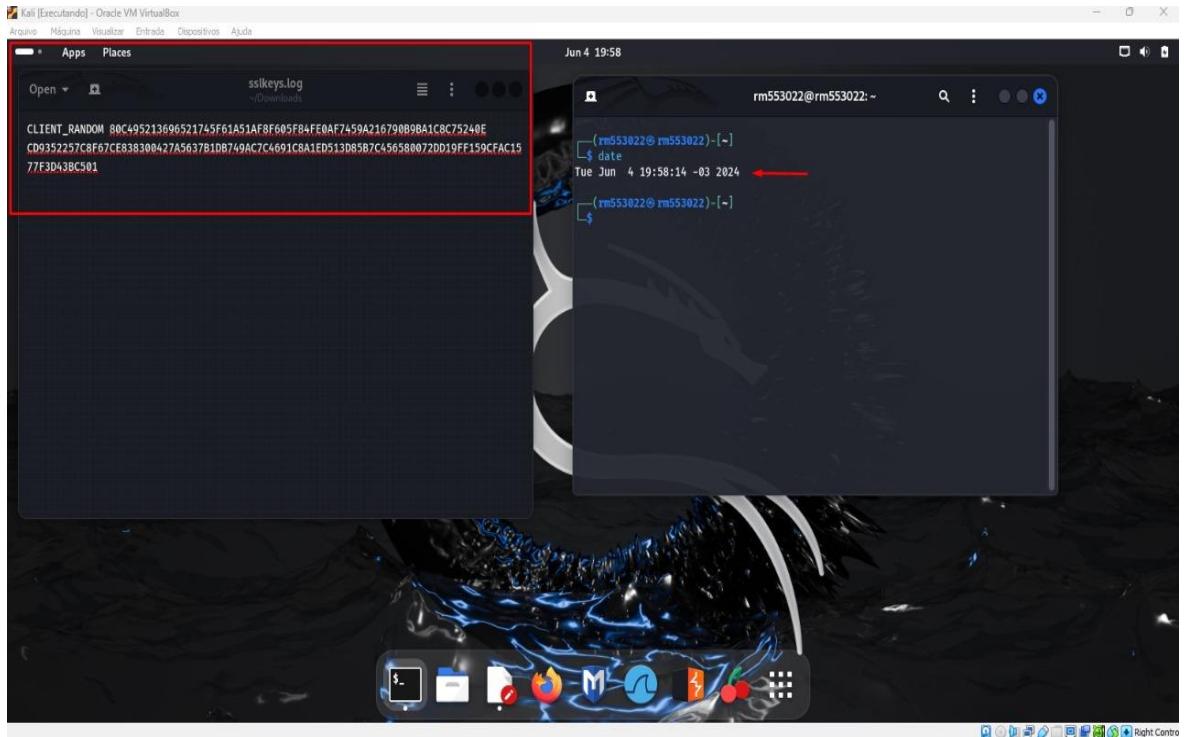
Figura 1 – E-mail.pcapng.



Fonte – Wireshark.

Ao localizar na linha 13 a chave “CLIENT\_RANDOM”, não é parte do protocolo TLS em si, mas pode ser encontrada em logs de sessão TLS, geralmente em registros de handshake.

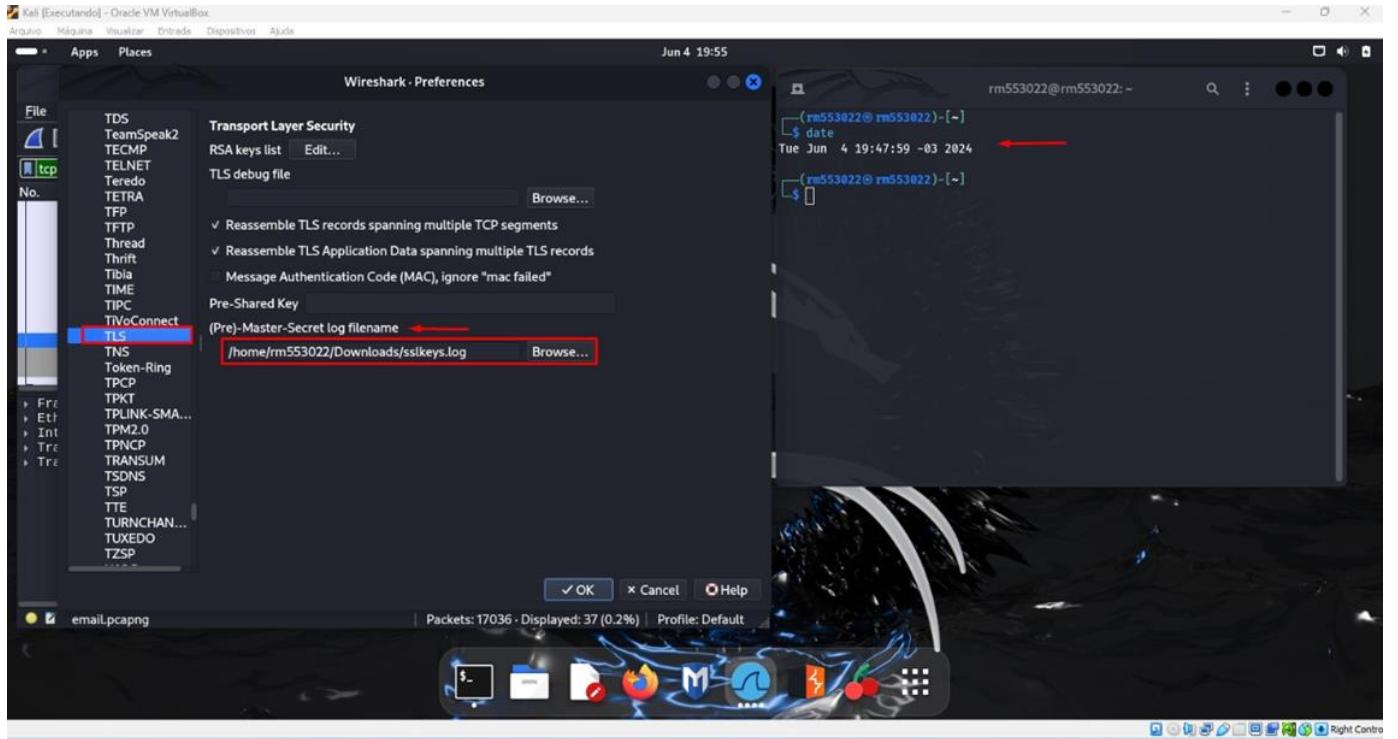
Figura 2 – E-mail.pcapng.



Fonte – Wireshark.

Após criar o arquivo sslkeys.log, com a chave, é necessário aplicar o arquivo nas configurações do wireshark.

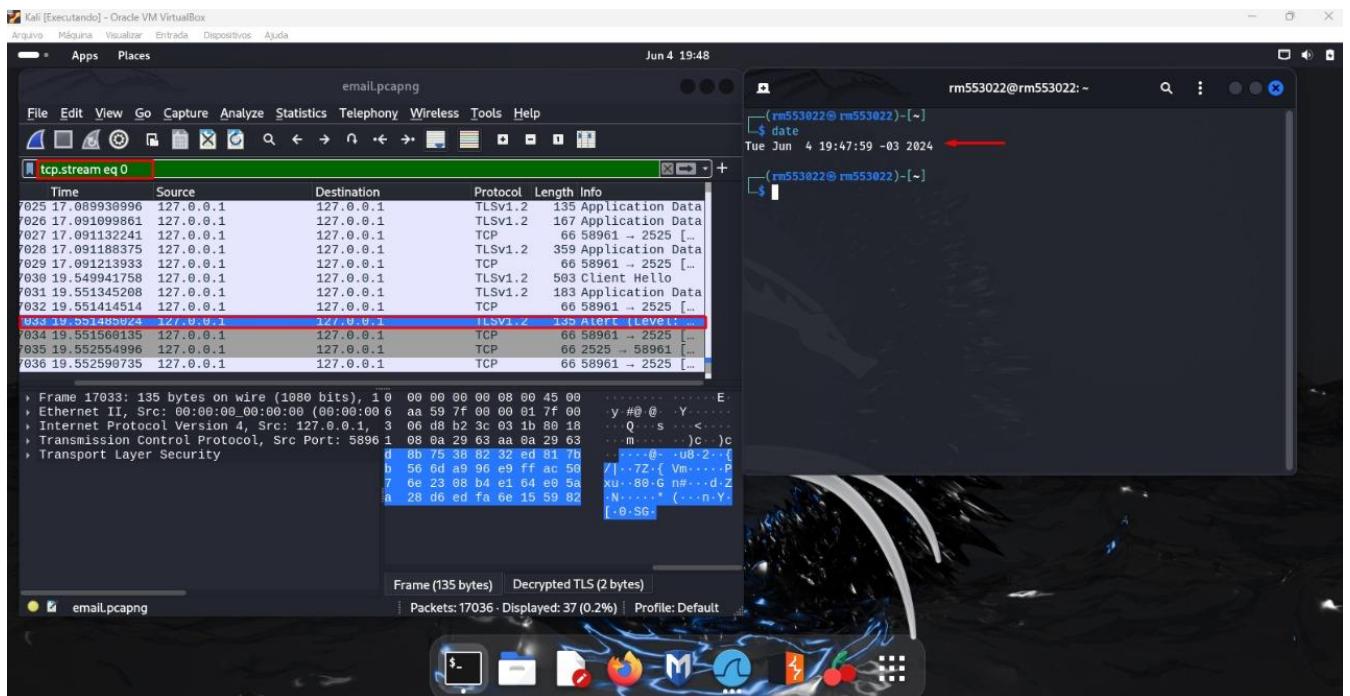
Figura 3 – E-mail.pcapng



Fonte – Wireshark.

Agora, o wireshark será capaz de descriptografar o tráfego TLS utilizando a chave privada fornecida. Você pode visualizar o tráfego descriptografado clicando nos pacotes TLS na janela principal do wireshark e examinando o conteúdo na aba "TLS" (ou "SSL", dependendo da versão) na seção "Secure Sockets Layer". Na figura 4 contém os detalhes dessa demonstração.

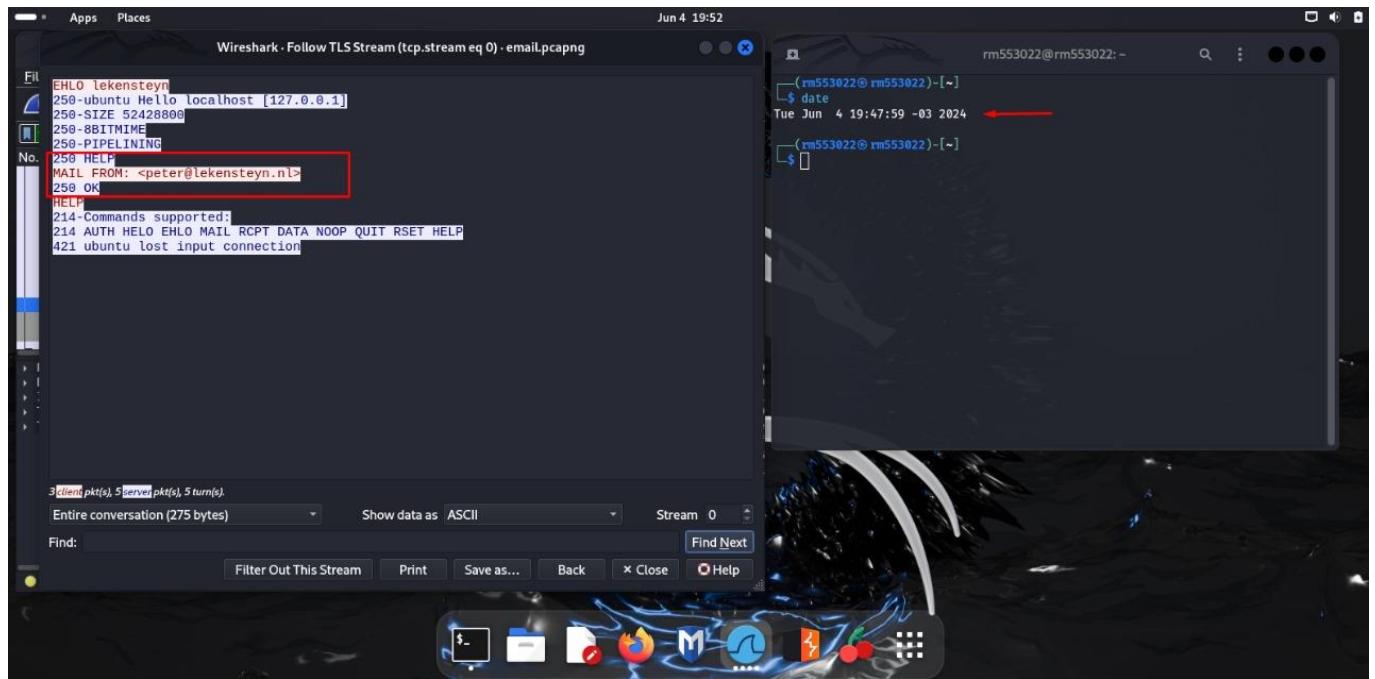
Figura 4 – E-mail.pcapng.



Fonte – Wireshark.

Agora é possível, em “Follow TLS Stream”, identificar o “FROM”.

Figura 5 - E-mail.pcapng.



Fonte - Wireshark.

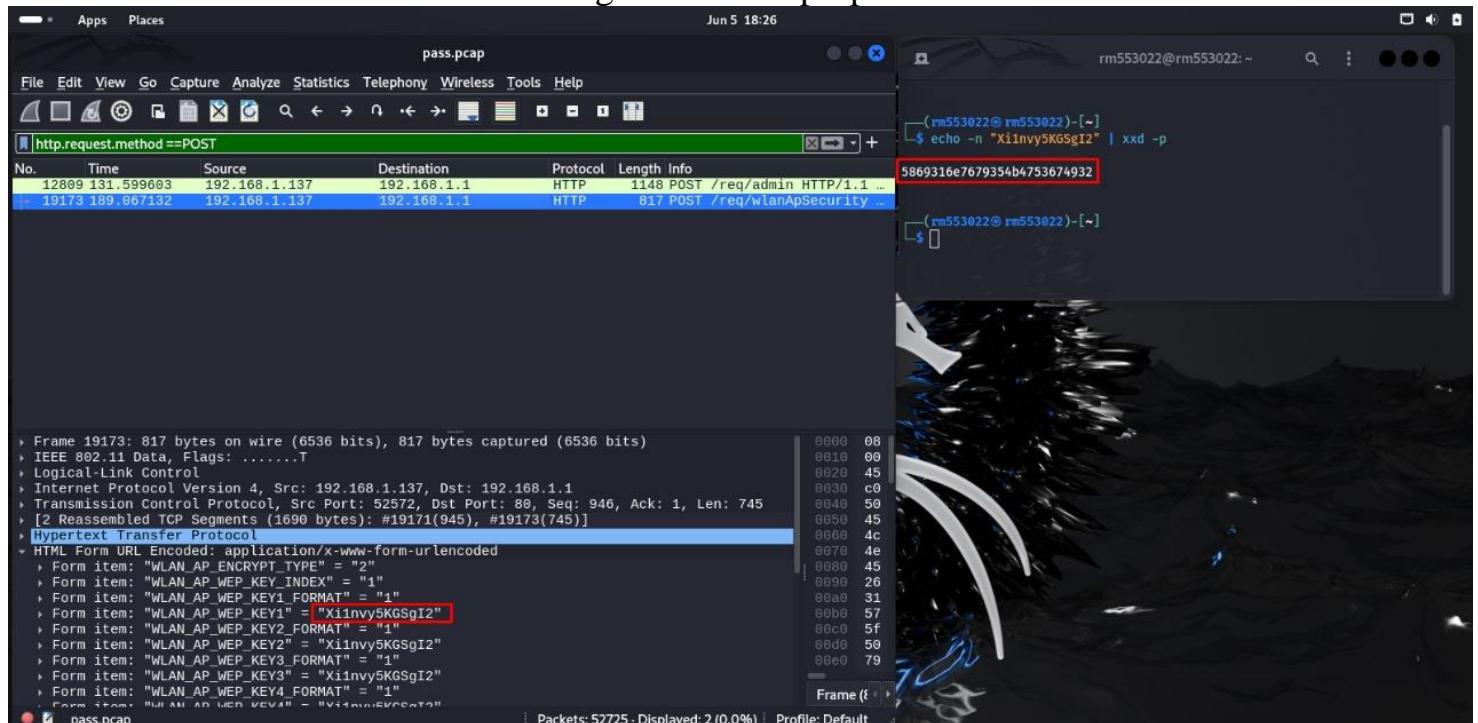
Dessa maneira a resposta então foi obtida de forma rápida e a Flag é:

FIAP{peter@lekensteyn.nl}

### 3. SENHA FORTE

Neste CTF, tinha como o objetivo analisar o arquivo PCAP, e determinar qual será a nova senha utilizada pelo administrador.

Figura 6 – Pass.pcap.

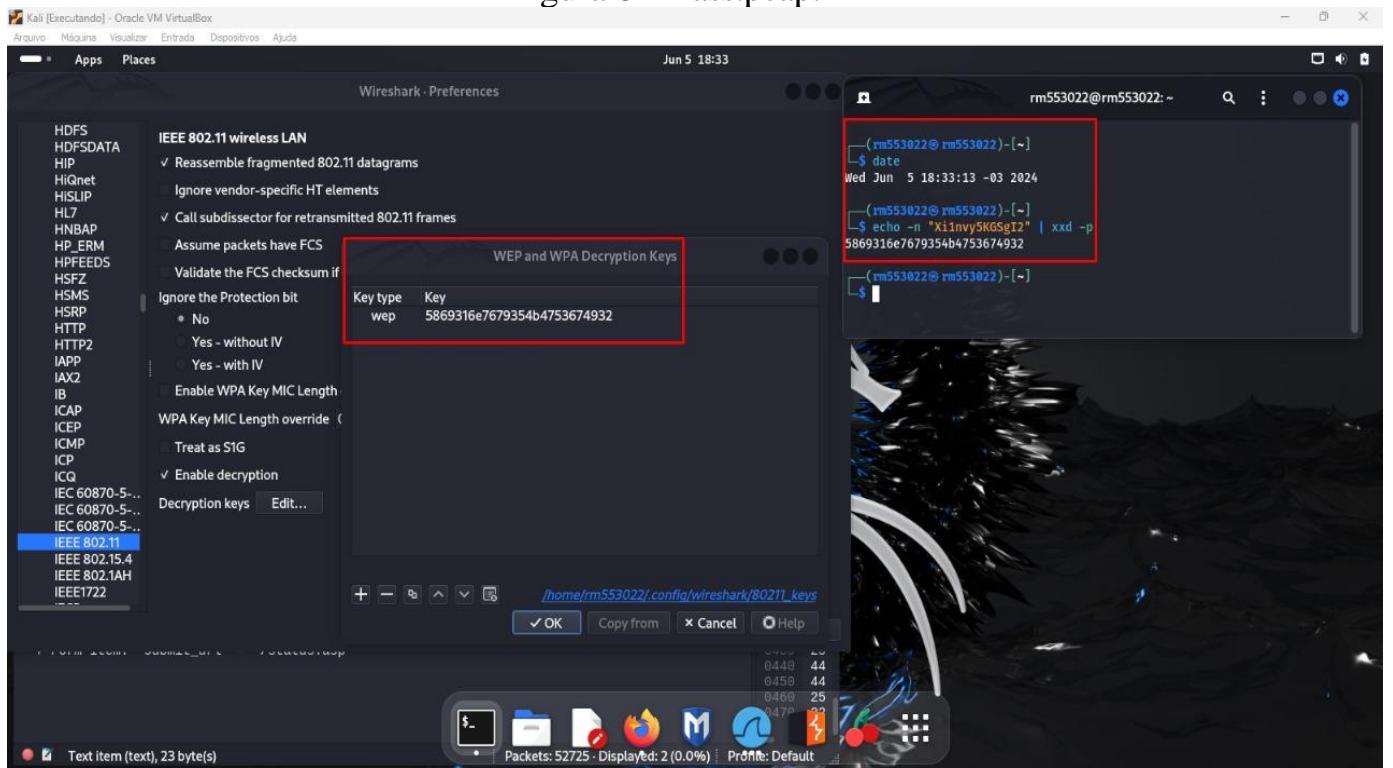


Fonte - Wireshark.

Após filtrar o http, analisando então o “request.method ==POST”.

Localizando o "WLAN\_AP\_WEP\_KEY4" = "Xi1nvy5KGsgI2", conforme figura 5.

Figura 6 – Pass.pcap.

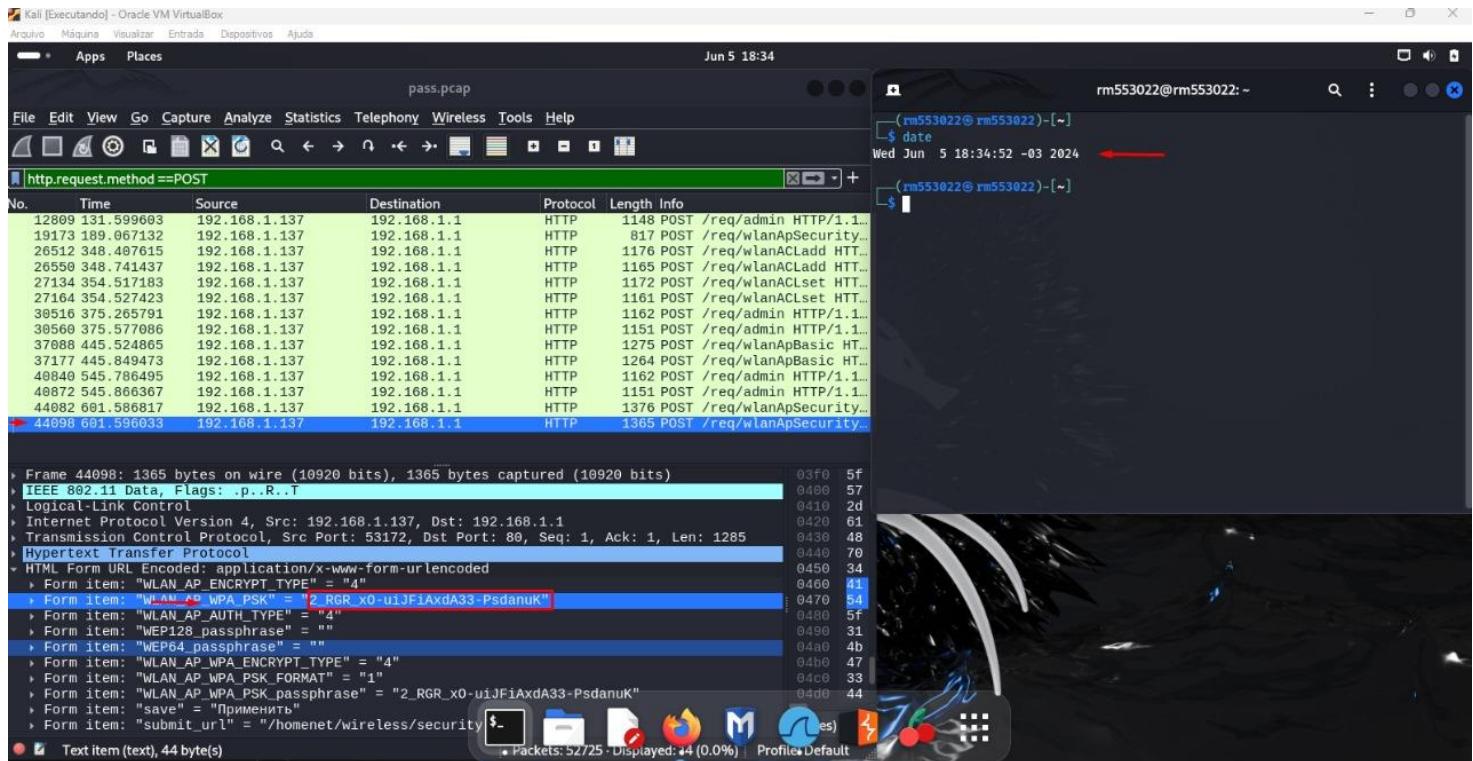


Fonte – Wireshark.

Após configuração, é necessário analisar os logs novamente e com isso, identifico outro elemento que será importante utilizar para descriptografar também o WPA.

Lembrando que o “IEEE 802.11” é um conjunto de padrões para redes locais sem fio (WLANS). Ele define os protocolos de comunicação e as especificações para redes sem fio, incluindo aspectos como frequências de operação, modulação, controle de acesso ao meio e segurança.

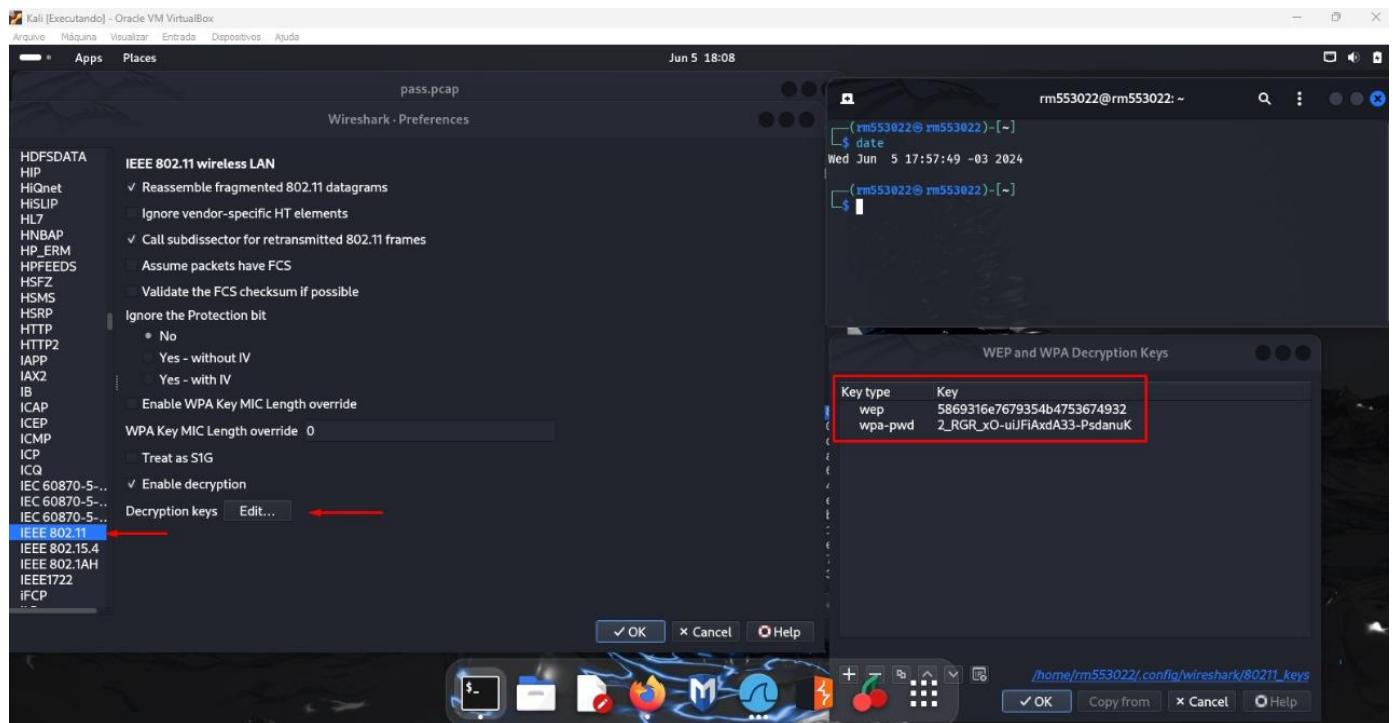
Figura 7 – Pass.pcap.



Fonte – Wireshark.

A seguir foi identificado o valor "WLAN\_AP\_WPA\_PSK\_passphrase" = "2\_RGR\_xO- uiJFiAxdA33-PsdanuK", na figura 8 contém os detalhes da configuração realizada.

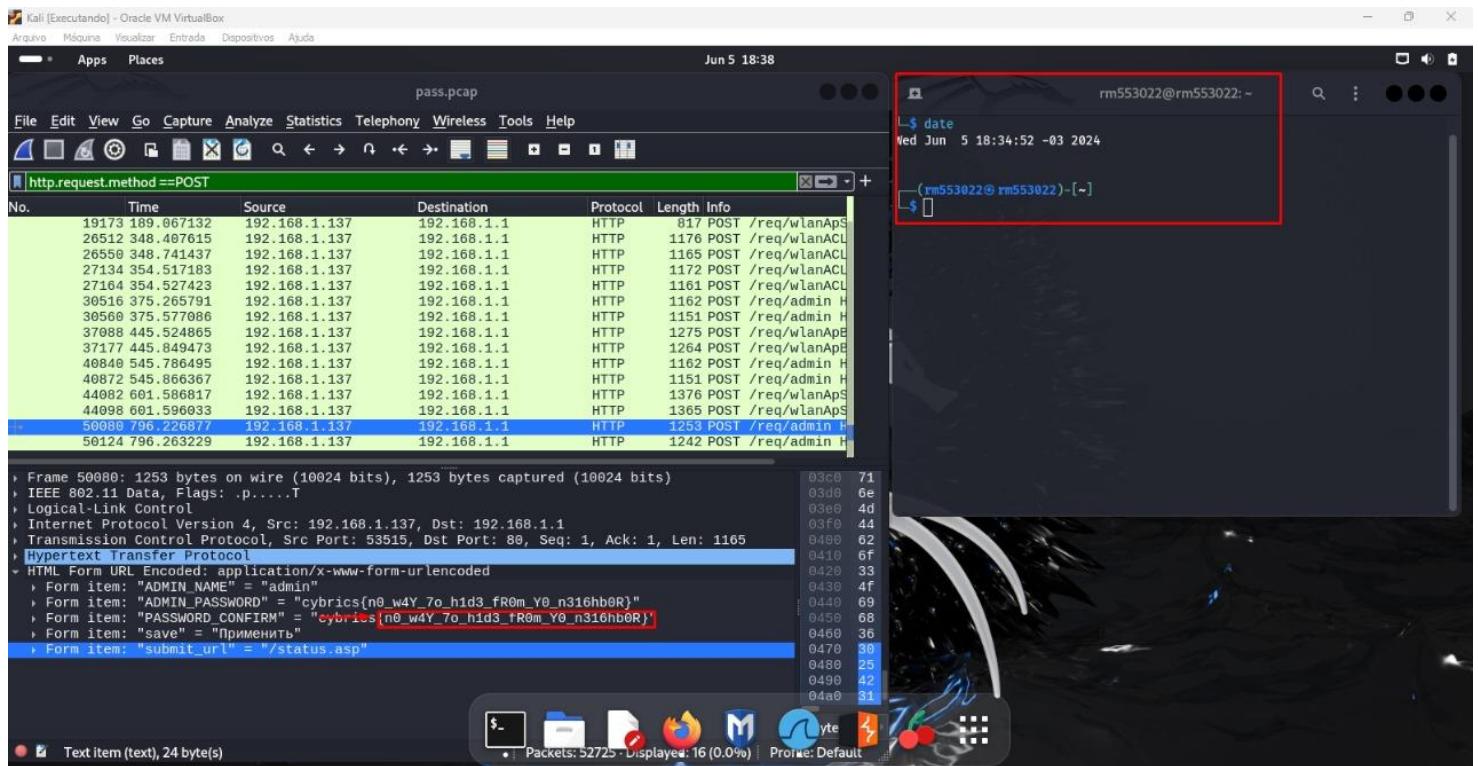
Figura 8 – Pass.pcap.



Fonte – Wireshark.

Após realizar as configurações no WEP e WPA-PWD.

Figura 9 – Pass.pcap.



Fonte – Wireshark.

Após as configurações, foi possível localizar a senha:

Form item:

"ADMIN\_PASSWORD" = "cybrics{n0\_w4Y\_7o\_h1d3\_fR0m\_Y0\_n316hb0R}"

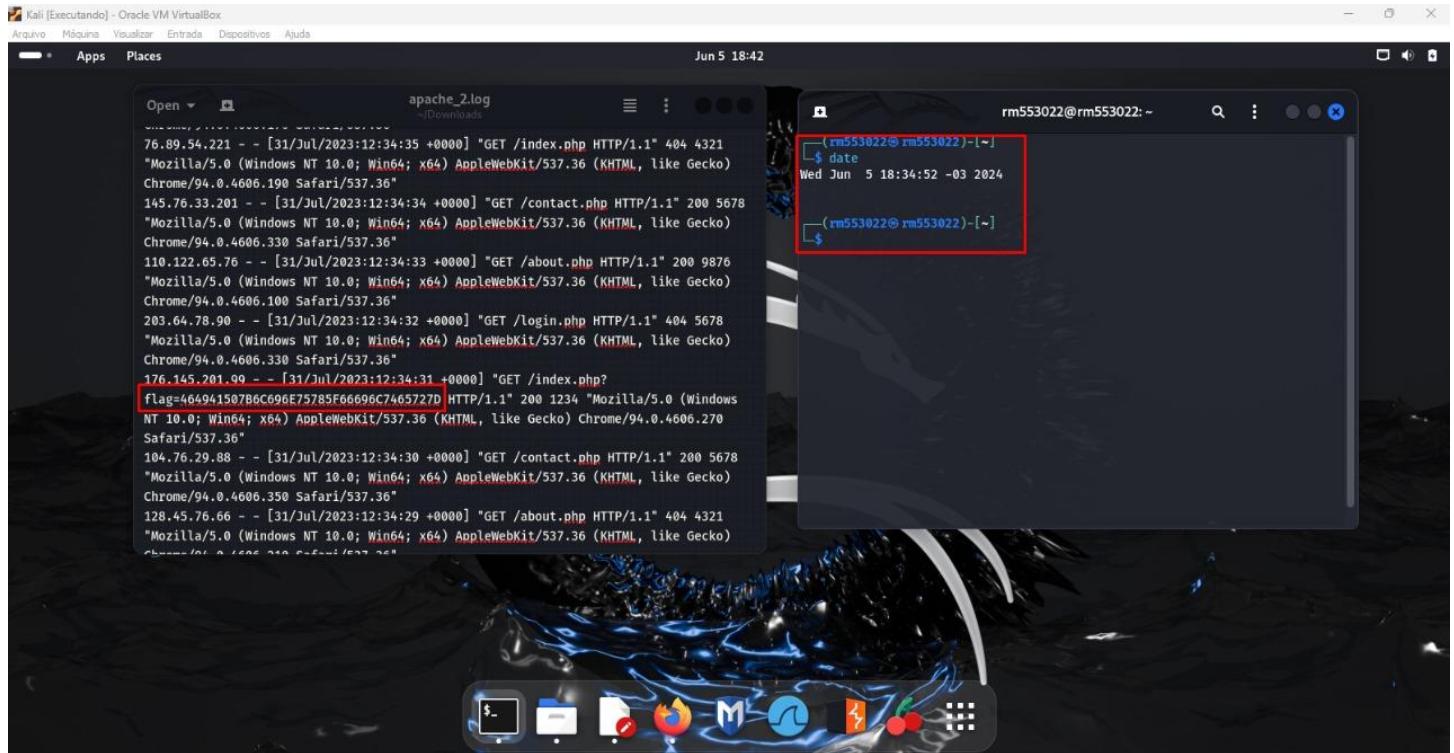
Dessa maneira a resposta então foi obtida de forma rápida e a Flag é:

FIAP{n0\_w4Y\_7o\_h1d3\_fR0m\_Y0\_n316hb0R}

#### 4. URL SUSPEITA

Neste CTF o objetivo é encontrar uma URL especial escondida no log de servidor.

Figura 10 – Apache\_2.log.



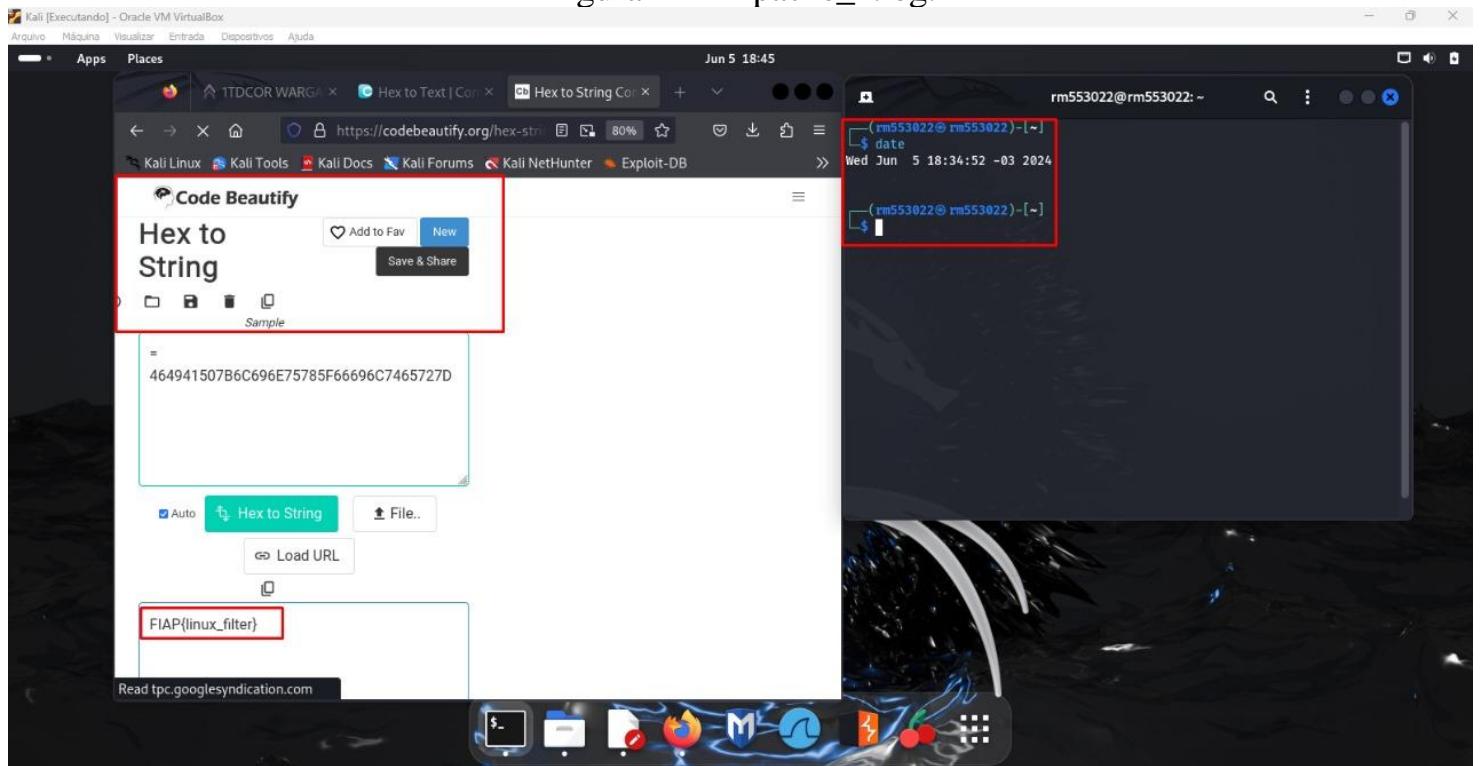
A screenshot of a Kali Linux desktop environment. In the foreground, a terminal window titled "apache\_2.log" displays a log file with numerous entries. One entry stands out with a red box around it: "flag:464941507B6C696E75785F66696C746572D". To the right of this terminal is another terminal window showing the command "date" and its output "Wed Jun 5 18:34:52 -03 2024". The desktop background features a dark, abstract image of water or liquid. A dock at the bottom contains icons for various applications like a terminal, file manager, browser, and file editor.

```
76.89.54.221 - - [31/Jul/2023:12:34:35 +0000] "GET /index.php HTTP/1.1" 404 4321
Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/94.0.4606.190 Safari/537.36"
145.76.33.201 - - [31/Jul/2023:12:34:34 +0000] "GET /contact.php HTTP/1.1" 200 5678
Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/94.0.4606.330 Safari/537.36"
110.122.65.76 - - [31/Jul/2023:12:34:33 +0000] "GET /about.php HTTP/1.1" 200 9876
Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/94.0.4606.100 Safari/537.36"
203.64.78.90 - - [31/Jul/2023:12:34:32 +0000] "GET /login.php HTTP/1.1" 404 5678
Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/94.0.4606.330 Safari/537.36"
176.145.201.99 - - [31/Jul/2023:12:34:31 +0000] "GET /index.php?
flag:464941507B6C696E75785F66696C746572D" HTTP/1.1" 200 1234
Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/94.0.4606.270 Safari/537.36"
104.76.29.88 - - [31/Jul/2023:12:34:30 +0000] "GET /contact.php HTTP/1.1" 200 5678
Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/94.0.4606.350 Safari/537.36"
128.45.76.66 - - [31/Jul/2023:12:34:29 +0000] "GET /about.php HTTP/1.1" 404 4321
Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/94.0.4606.350 Safari/537.36"
```

Fonte – Terminal Kali Linux.

Após analisar os logs, foi aplicado um filtro, e localizado um hex, seguindo a dica “flag=”.

Figura 11 – Apache\_2.log.



Fonte – Code Beautify.

Dessa maneira a resposta então foi obtida de forma rápida e a Flag é:  
FIAP{linux\_filter}.

## 5. PIN

Neste CTF o PIN em questão é muito fraco, o desafio é encontrar.

Figura 12 – Arte.jpeg.

The image shows two terminal windows side-by-side. Both windows are running on a Kali Linux desktop environment, indicated by the window title 'Kali [Executando] - Oracle VM VirtualBox' and the desktop interface.

**Left Terminal:**

- Shows the command `ls -l` being run in the directory `~/Downloads`.
- Highlights the file `arte.jpeg` with a red arrow.
- Shows the command `stegseek arte.jpeg rockyou.txt` being run.
- Shows the output of StegSeek 0.6, which includes:
  - [i] Found passphrase: "0066" B
  - [i] Original filename: "secreto.txt".
  - [i] Extracting to "arte.jpeg.out".

**Right Terminal:**

- Shows the command `ls -l` being run in the directory `~/Downloads`.
- Highlights the file `arte.jpeg` with a red arrow.
- Shows the command `cat arte.jpeg.out` being run.
- Shows the output: `FIAP{crunch_password_dic_creator}`, which is highlighted with a red box.

Fonte – Terminal Kali Linux.

Dessa maneira a resposta então foi obtida de forma rápida como podemos ver nas imagens a seguir:

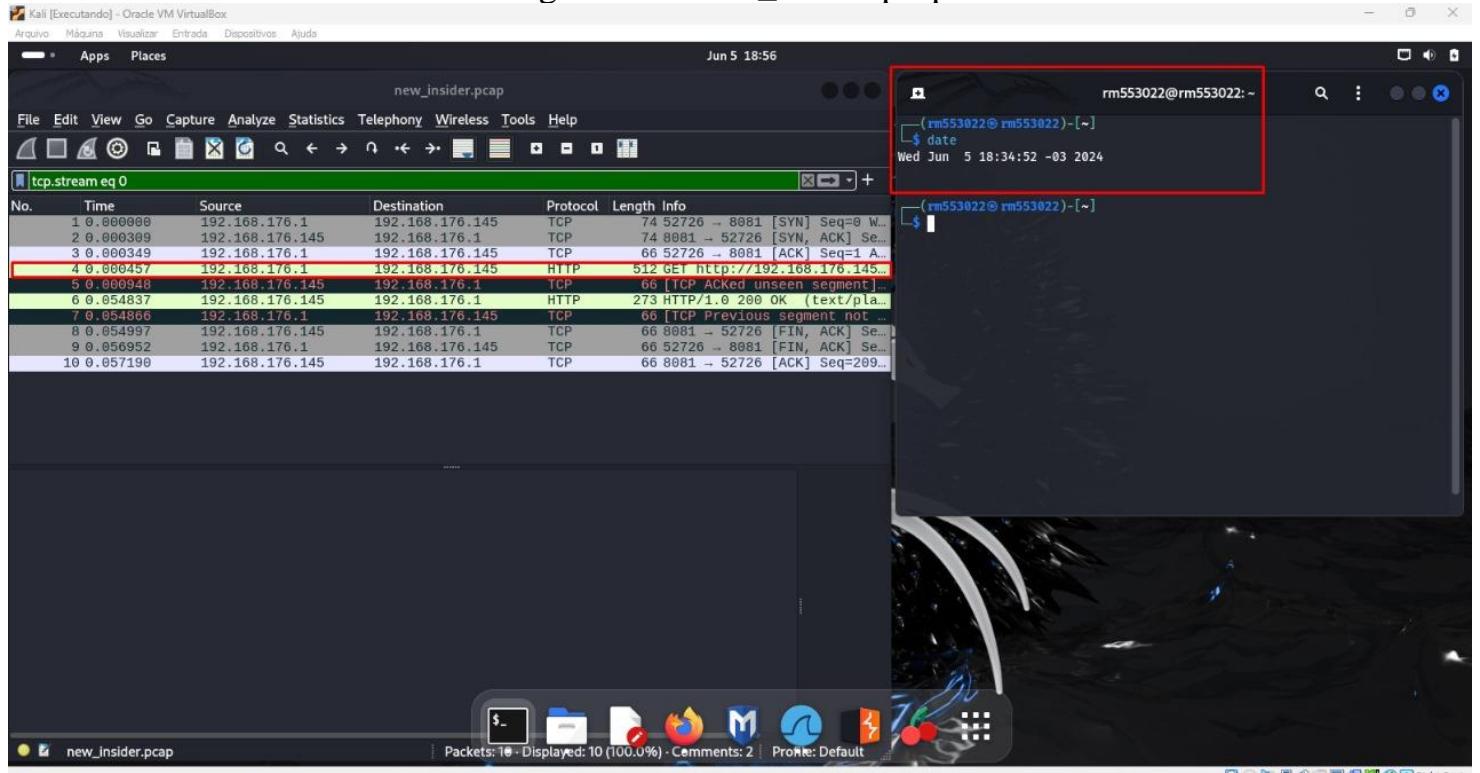
FLAG: FIAP{crunch\_password\_dic\_creator}.

## 6. NEW INSIDER

Neste CTF o desafio é analisar dois artefatos encontrados no sistema.

Um arquivo de captura de pacotes denominado "new\_insider.pcap" e um arquivo zip criptografado chamado "file.zip".

Figura 13 – New\_insider.pcap.



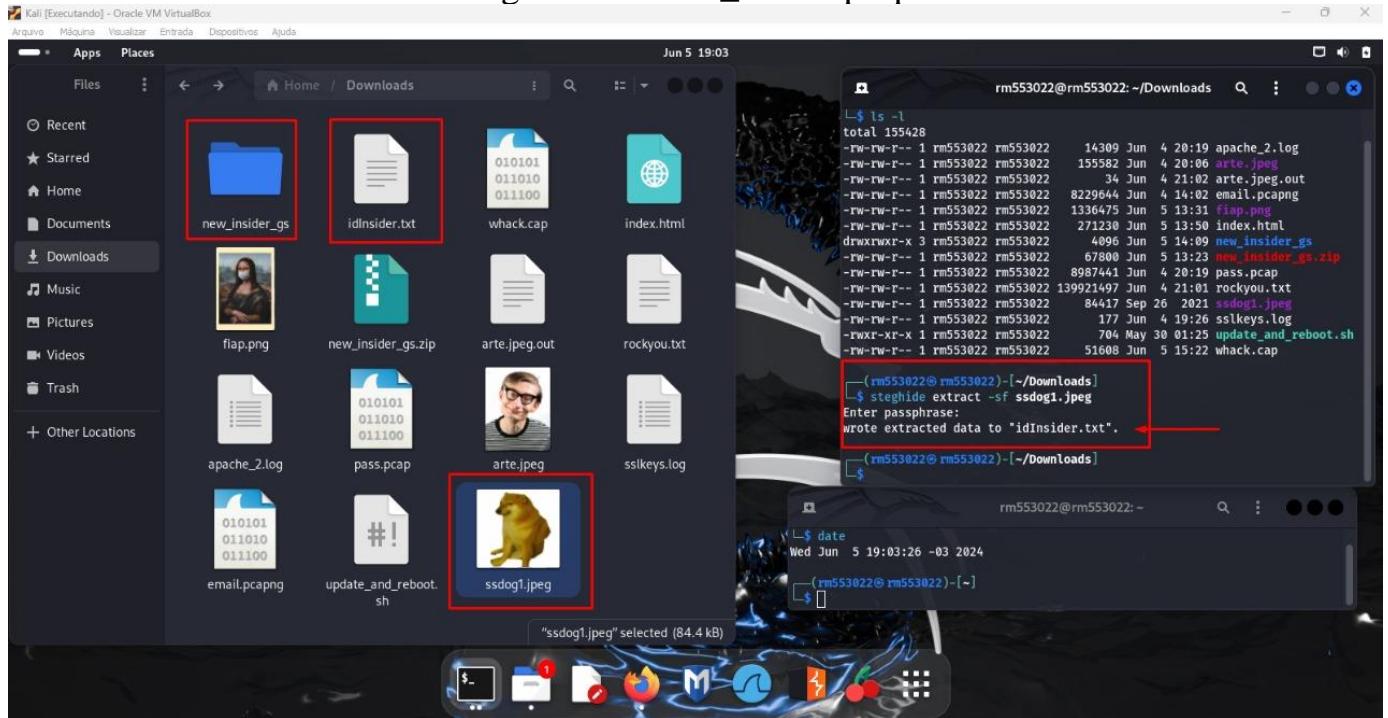
Fonte – Wireshark.

Após extração do arquivo zipado, foi possível localizar o Hash.

Base64:YWRtaW46Ymx1ZXRIYW1vcGVyYXRpb24K, ao decodar, a seguinte resposta é obtida, “admin:blueteamoperation”.

É a senha utilizada para extrair o arquivo “insider.zip”.

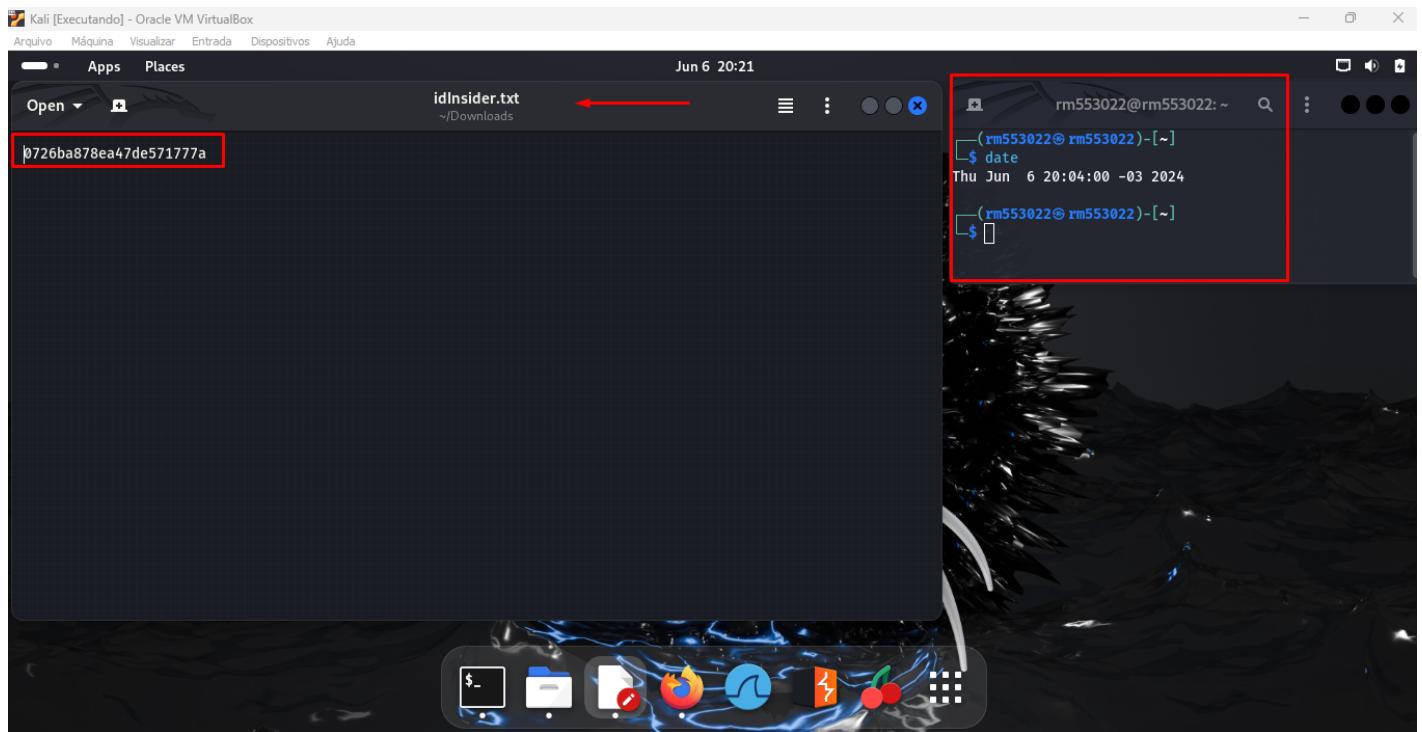
Figura 14 – New\_insider.pcap.



Fonte – Kali Linux.

Após utilizar o steghide, para extrair o arquivo “idInsider.txt”, da imagem “ssdog1.jpeg”.

Figura 15 – New\_insider.pcap.



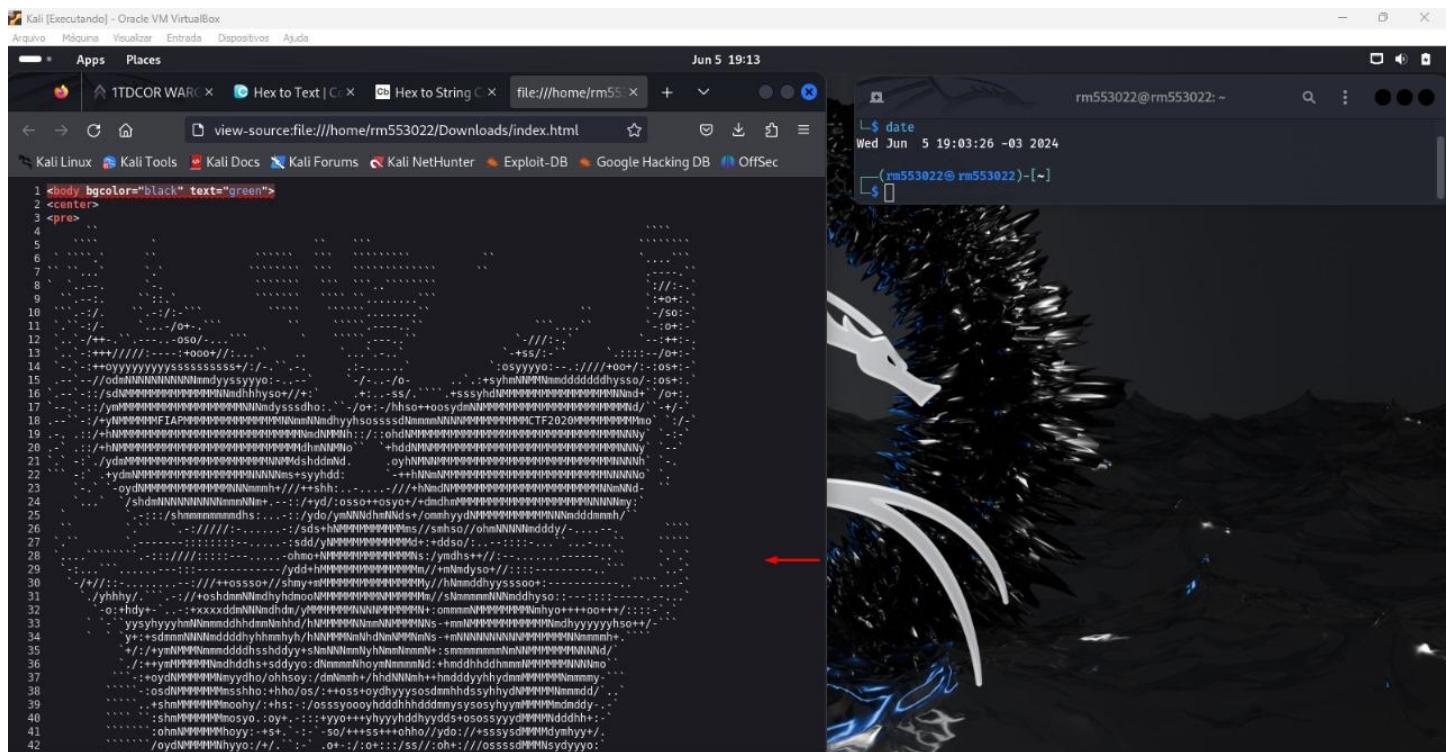
Fonte – Kali Linux.

Dessa maneira a resposta então foi obtida de forma rápida e a Flag é:  
FIAP{crunch\_password\_dic\_creator}

## 7. DEEP

Neste CTF tem uma página que estava contida na Deep Web vazou, o objetivo é analisar o conteúdo e obter a Flag.

Figura 16 – Index.html.



Fonte – Firefox.

Em CTRL+U é possível analisar o código por de trás do “index.html”.

Figura 17 – Index.html.

The screenshot shows a Kali Linux desktop environment. On the left, a terminal window titled 'rm553022@rm553022:~' displays the command 'date' followed by the output 'Wed Jun 5 19:03:26 -03 2024'. A red arrow points from the terminal window towards the browser window. On the right, a web browser window is open to 'view-source:file:///home/rm553022/Downloads/index.html'. The page source code is visible, with line numbers 119 through 128 highlighted by a red box. These lines contain a Base64 encoded string: '1VBORw0KGgoAAAANSUhEUgAAAFqCAYAAAAp0o2cAAAK1wLDQ1BJ08MglHJvZmlsZ0AA5ImVlwdUk8kwx+f70kNCSSACUkLvSceANBDlw6iEpk0hBJD...'. The rest of the page source code is numbered from 90 to 147.

```
90  
91  
92  
93  
94  
95  
96  
97  
98  
99  
100  
101  
102  
103  
104  
105  
106  
107  
108  
109  
110  
111  
112  
113  
114  
115  
116  
117  
118  
119 <table border=0 width=300>  
120   <tr><td>  
121  
122 </td>-  
123 1VBORw0KGgoAAAANSUhEUgAAAFqCAYAAAAp0o2cAAAK1wLDQ1BJ08MglHJvZmlsZ0AA5ImVlwdUk8kwx+f70kNCSSACUkLvSceANBDlw6iEpk0hBJD...  
124  
125 .->  
126  
127 </td></td>  
128 </table>  
129  
130  
131  
132
```

Fonte – Index.html.

Após localizar o Base64, dessa maneira a resposta então foi obtida de forma rápida e a Flag é:  
FIAP{#For3nsic\$!}.

## 8. WI-FI HACKING

Neste CTF, analise o PCAP e informe o nome do cliente contido no frame 210.

Figura 18 – Whack.cap.

The screenshot shows a terminal window on a Kali Linux desktop. The terminal output is as follows:

```
rm553022@rm553022: ~/Downloads
$ ls -l
total 155432
drwxrwxr-- 1 rm553022 rm553022 14309 Jun 4 20:19 apache_2.log
-rw-rw-r-- 1 rm553022 rm553022 15582 Jun 4 20:06 arte.jpeg
-rw-rw-r-- 1 rm553022 rm553022 34 Jun 4 21:02 arte.jpeg.out
-rw-rw-r-- 1 rm553022 rm553022 8229644 Jun 4 14:02 email.pcapng
-rw-rw-r-- 1 rm553022 rm553022 1336475 Jun 5 13:31 fiap.png
-rw-rw-r-- 1 rm553022 rm553022 23 Jun 5 19:03 idInsider.txt
-rw-rw-r-- 1 rm553022 rm553022 271230 Jun 5 13:51 index.html
drwxrwxr-- 3 rm553022 rm553022 4096 Jun 5 19:08 new_insider_gs
-rw-rw-r-- 1 rm553022 rm553022 67800 Jun 5 13:23 new_insider_gs.zip
-rw-rw-r-- 1 rm553022 rm553022 8987441 Jun 4 20:19 pass.pcap
-rw-rw-r-- 1 rm553022 rm553022 139921497 Jun 4 21:01 rockyou.txt
-rw-rw-r-- 1 rm553022 rm553022 84417 Sep 26 2021 ssdgel.jpeg
-rw-rw-r-- 1 rm553022 rm553022 177 Jun 4 19:26 sslkeys.log
-rw-r-xr-x 1 rm553022 rm553022 704 May 30 01:25 update_and_reboot.sh
-rw-rw-r-- 1 rm553022 rm553022 51608 Jun 5 15:22 whack.cap

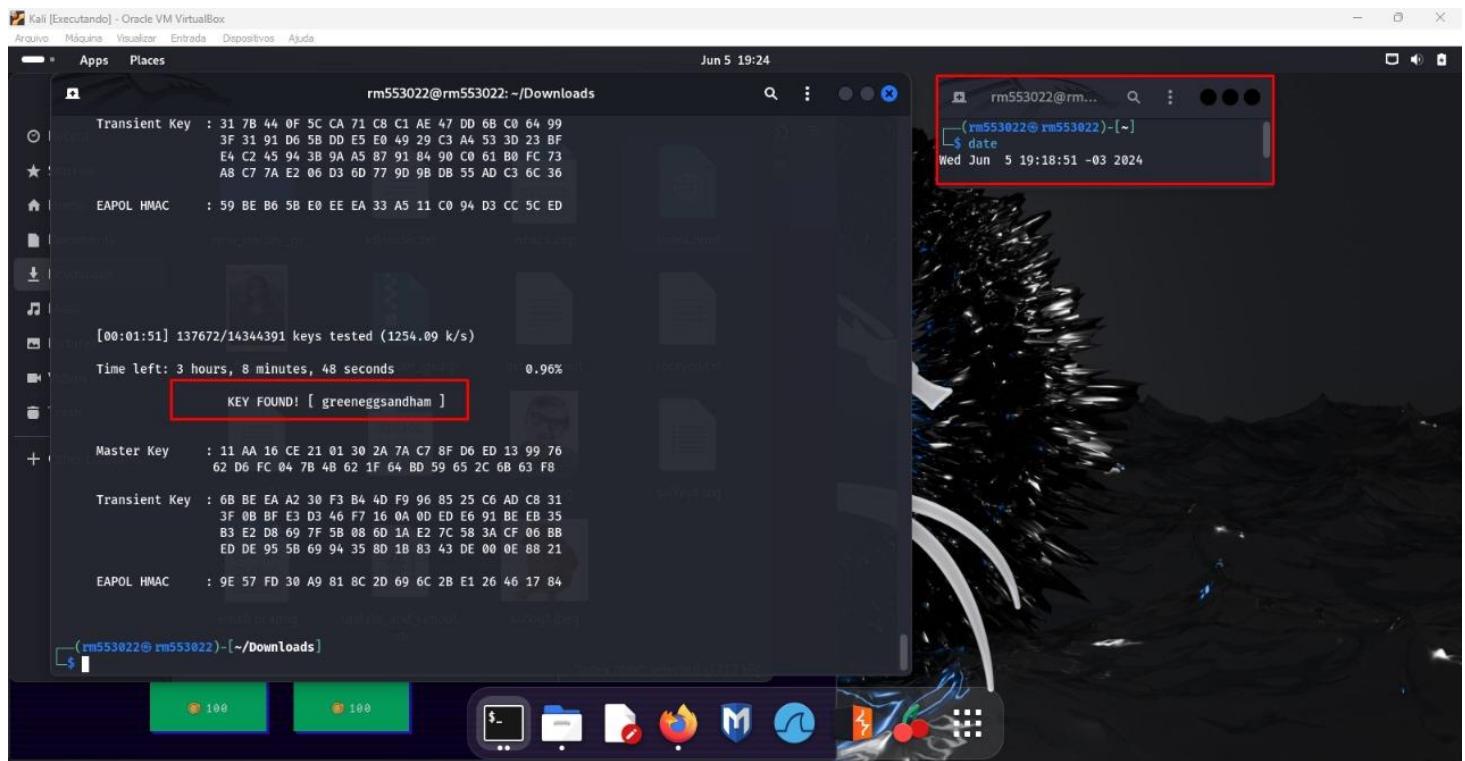
( rm553022@rm553022: ~/Downloads )
$ aircrack-ng whack.cap -w rockyou.txt
```

A red arrow points to the command `aircrack-ng whack.cap -w rockyou.txt`. The terminal window has a red border around its title bar and the command line area.

Fonte – Kali Linux.

Após analisar dos logs, foi identificado a necessidade de rodar o aircrack, no arquivo “Whack.cap”.

Figura 19 – Whack.cap.



```
rm553022@rm553022:~/Downloads
Transient Key : 31 7B 44 0F 5C CA 71 C8 C1 AE 47 DD 6B C0 64 99
                3F 31 91 D6 5B DD E5 E0 49 29 C3 A4 53 3D 23 BF
                E4 C2 45 94 3B 9A A5 87 91 84 90 C0 61 B0 FC 73
                A8 C7 7A E2 06 03 6D 77 9D 9B DB 55 AD C3 6C 36

EAPOL HMAC : 59 BE B6 5B E0 EE EA 33 A5 11 C0 94 D3 CC 5C ED

[00:01:51] 137672/14344391 keys tested (1254.09 k/s)
Time left: 3 hours, 8 minutes, 48 seconds      0.96%
KEY FOUND! [ greeneggsandham ]

Master Key : 11 AA 16 CE 21 01 30 2A 7A C7 8F D6 ED 13 99 76
              62 D6 FC 04 7B 4B 62 1F 64 BD 59 65 2C 6B 63 F8

Transient Key : 6B BE EA A2 30 F3 B4 4D F9 96 85 25 C6 AD C8 31
                 3F 0B BF E3 D3 46 F7 16 0A 0D ED E6 91 BE EB 35
                 B3 E2 D8 69 7F 5B 08 60 1A E2 7C 58 3A CF 06 BB
                 ED DE 95 5B 69 94 35 8D 1B 83 43 DE 00 0E 88 21

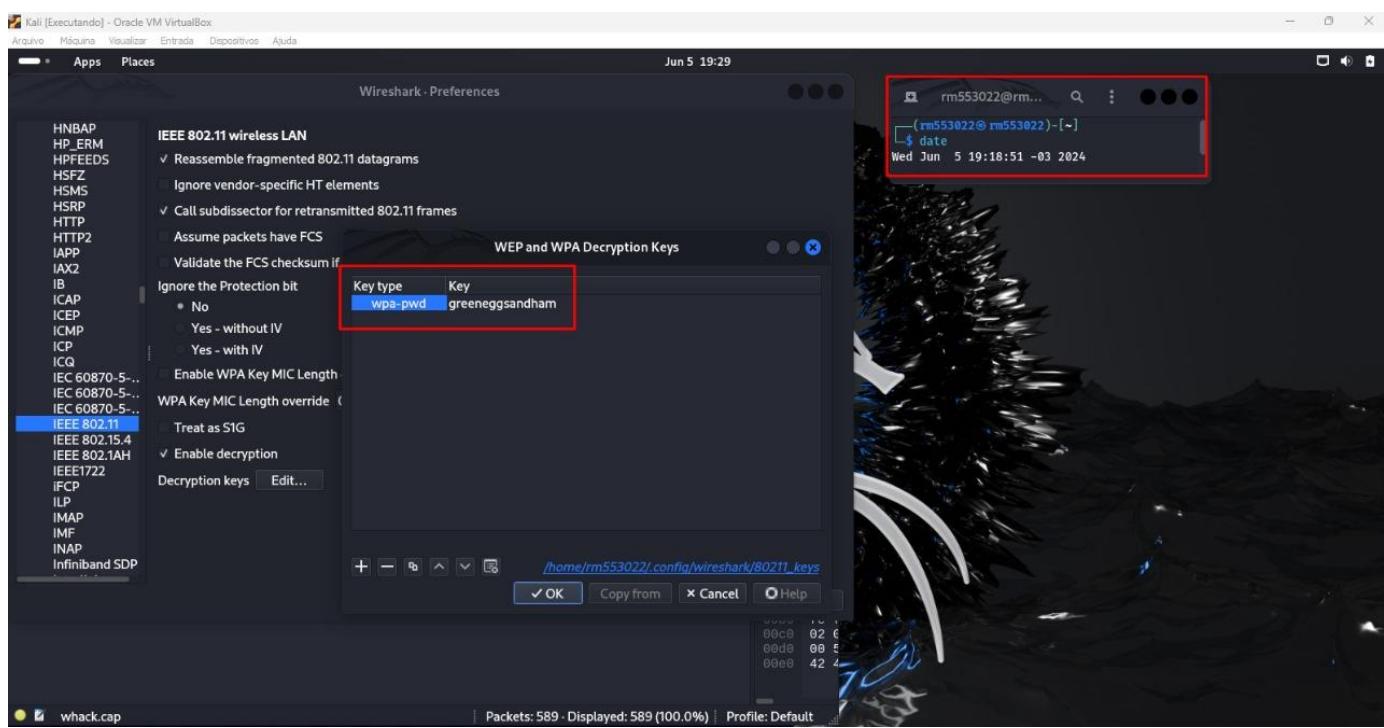
EAPOL HMAC : 9E 57 FD 30 A9 81 8C 2D 69 6C 2B E1 26 46 17 84

(rm553022@rm553022) [~/Downloads]
$
```

Fonte – Kali Linux

Após concluir os comandos do aircrack, foi obtido a chave Key Found :{greeneggsandham} e utilizada na configuração.

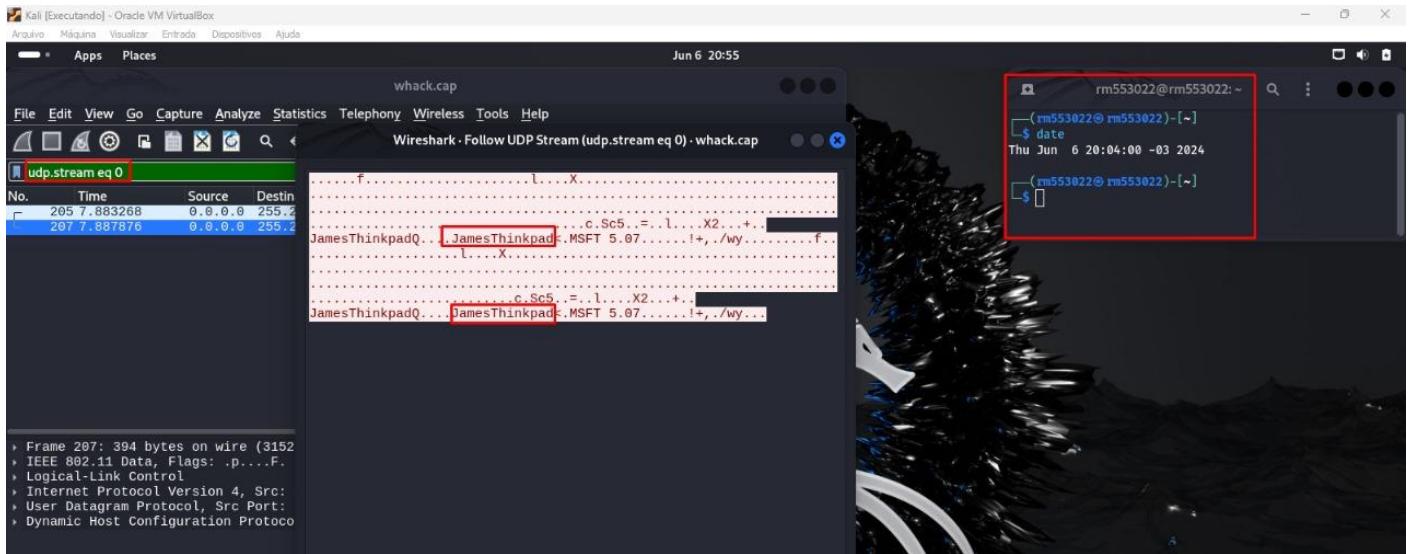
Figura 20 – Whack.cap.



Fonte – Wireshark.

Após a configuração do WPA-PWD.

Figura 21 – Whack.cap.



Fonte – Wireshark.

Filtrando em udp.stream eq 0 em Follow UDP Stream. Dessa maneira a resposta então foi obtida de forma rápida e a Flag é:  
FIAP{JamesThinkpad}.

## 9. BRUTE FORCE

Neste CTF a tarefa é ordenar e identificar os itens únicos no arquivo. Em seguida, localize o valor na linha 3333 e acrescente 4 dígitos numéricos ao final do resultado.

Desta maneira, você terá um dicionário de senhas para desbloquear o PDF e obter a flag.

Figura 22 – Texto.txt.

The screenshot shows a Kali Linux desktop environment with a terminal window open. The terminal history is as follows:

```
rm553022@rm553022:~/Downloads
ls -l
total 76
-rw-rw-r-- 1 rm553022 rm553022 37788 Jun  6 23:27 file.pdf
-rw-rw-r-- 1 rm553022 rm553022 150 Jun  6 23:29 organizador.py
-rw-rw-r-- 1 rm553022 rm553022 31903 Jun  6 23:27 texto.txt

rm553022@rm553022:~/Downloads
date
Thu Jun  6 23:30:38 -03 2024

rm553022@rm553022:~/Downloads
sort texto.txt > wordlist.txt ← Red arrow points here

rm553022@rm553022:~/Downloads
ls -l
total 108
-rw-rw-r-- 1 rm553022 rm553022 37788 Jun  6 23:27 file.pdf
-rw-rw-r-- 1 rm553022 rm553022 150 Jun  6 23:29 organizador.py
-rw-rw-r-- 1 rm553022 rm553022 31903 Jun  6 23:27 texto.txt
-rw-rw-r-- 1 rm553022 rm553022 31903 Jun  6 23:33 wordlist.txt ← Red arrow points here

rm553022@rm553022:~/Downloads
```

To the right of the terminal, there is a code editor window titled "organizador.py" containing the following Python script:

```
prefix = "scooter1"
wordlist = [f"{prefix}{i:04}" for i in range(10000)]

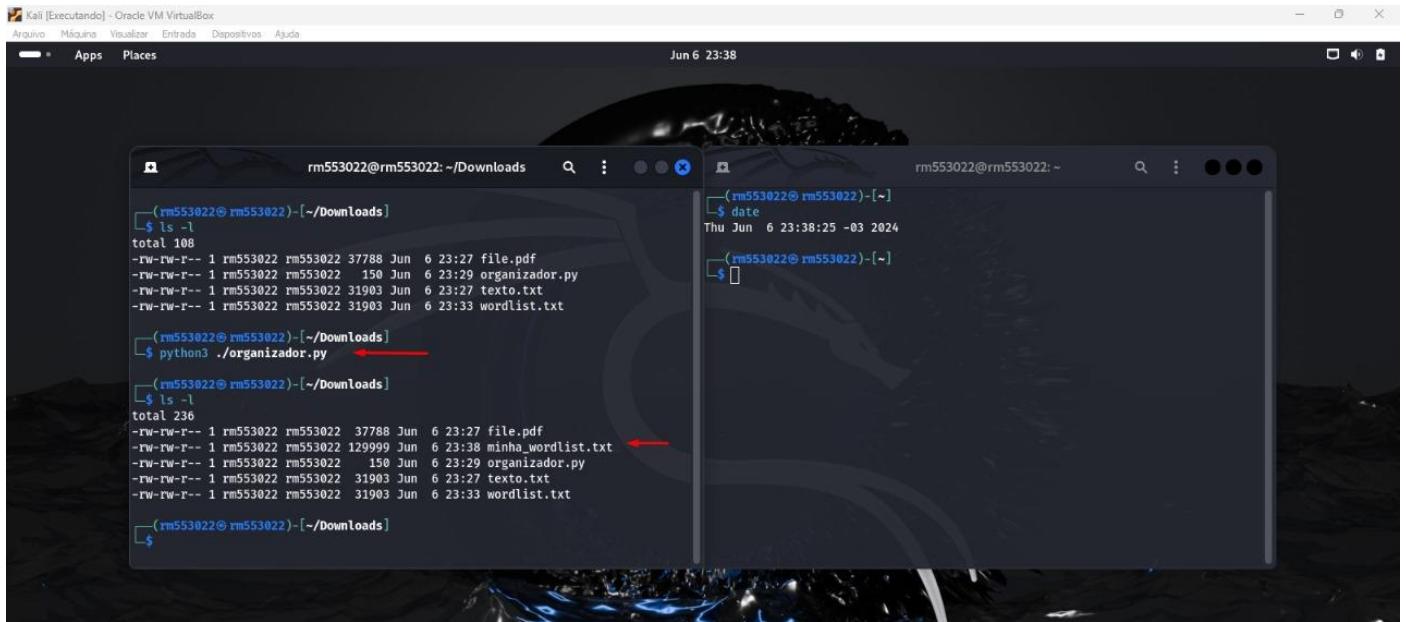
with open('minha_wordlist.txt', 'w') as f:
    f.write("\n".join(wordlist))
```

Fonte – Kali Linux.

Conforme a orientação da introdução do CTF, ao elaborar uma “wordlist customizada”, atribuindo o valor = “scooter1”, após aplicar o comando “sort texto.txt”, o scooter é localizado na linha 3333.

Com isso é criado uma wordlist, do 10000 > 19999, “em organizador.py”.

Figura 23 – Texto.txt.



The screenshot shows a Kali Linux desktop environment with two terminal windows open. The top window is titled 'rm553022@rm553022: ~/Downloads' and displays the command '\$ ls -l' followed by a list of files including 'file.pdf', 'organizador.py', 'texto.txt', and 'wordlist.txt'. A red arrow points to the command '\$ python3 ./organizador.py'. The bottom window is titled 'rm553022@rm553022: ~' and shows the command '\$ date' followed by the output 'Thu Jun 6 23:38:25 -03 2024'. Another red arrow points to the command '\$ ls -l' in this window, which lists files including 'minha\_wordlist.txt'. Both windows show a dark background with a dragon logo.

Fonte – Kali Linux.

Ao gerar o arquivo “minha\_wordlist.txt”, é necessário agora, aplicar os comandos do john, pois é a ferramenta utilizada para o Brute Force.

Figura 24 – Texto.txt.

The screenshot shows two terminal windows side-by-side. The left window is in the Downloads directory and lists files: file.pdf, minha\_wordlist.txt, organizador.py, texto.txt, and wordlist.txt. The command \$ ls -l is shown above the list. The right window shows the output of the date command: Thu Jun 6 23:38:25 -03 2024. Red arrows point from the command \$ pdf2john file.pdf > hash in the left window to the file.hash file in the right window, and from the command \$ John --wordlist=minha\_wordlist.txt hash in the left window to the John process in the right window.

```
rm553022@rm553022:~/Downloads
$ ls -l
total 236
-rw-rw-r-- 1 rm553022 rm553022 37788 Jun 6 23:27 file.pdf
-rw-rw-r-- 1 rm553022 rm553022 129999 Jun 6 23:38 minha_wordlist.txt
-rw-rw-r-- 1 rm553022 rm553022 150 Jun 6 23:29 organizador.py
-rw-rw-r-- 1 rm553022 rm553022 31903 Jun 6 23:27 texto.txt
-rw-rw-r-- 1 rm553022 rm553022 31903 Jun 6 23:33 wordlist.txt

rm553022@rm553022:~/Downloads
$ ls -l
total 240
-rw-rw-r-- 1 rm553022 rm553022 37788 Jun 6 23:27 file.pdf
-rw-rw-r-- 1 rm553022 rm553022 202 Jun 6 23:40 hash
-rw-rw-r-- 1 rm553022 rm553022 129999 Jun 6 23:38 minha_wordlist.txt
-rw-rw-r-- 1 rm553022 rm553022 150 Jun 6 23:29 organizador.py
-rw-rw-r-- 1 rm553022 rm553022 31903 Jun 6 23:27 texto.txt
-rw-rw-r-- 1 rm553022 rm553022 31903 Jun 6 23:33 wordlist.txt

rm553022@rm553022:~/Downloads
$ John --wordlist=minha_wordlist.txt hash
```

Fonte – Kali Linux.

O comando “pdf2john fila.pdf”, gera o hash, importante verificar o arquivo hash e excluir “file.pdf”.

Figura 25 – Texto.txt.

The screenshot shows two terminal windows. The left window shows the command \$ john --wordlist=minha\_wordlist.txt hash being run. The right window shows the date command being run, with a red arrow pointing to its output: Thu Jun 6 23:38:25 -03 2024. Red arrows also point from the command line in the left window to the john process in the right window, and from the hash command in the left window to the hash file in the right window.

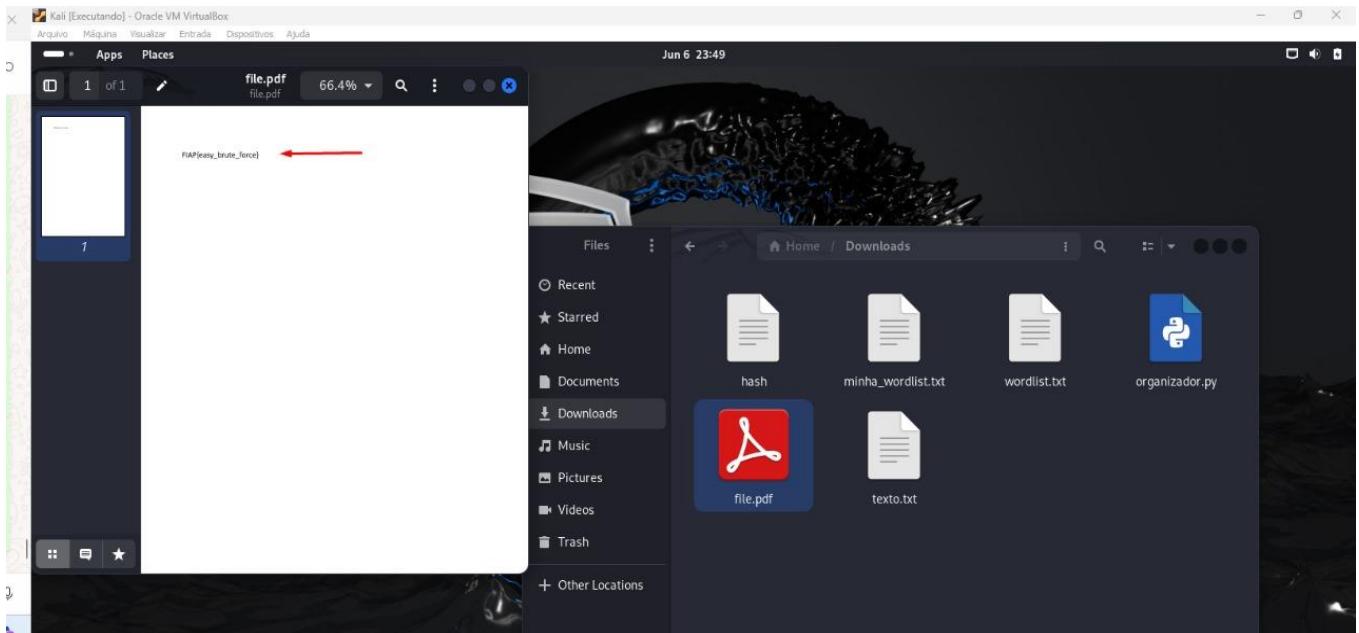
```
rm553022@rm553022:~/Downloads
$ ls -l
total 236
-rw-rw-r-- 1 rm553022 rm553022 37788 Jun 6 23:27 file.pdf
-rw-rw-r-- 1 rm553022 rm553022 193 Jun 6 23:43 hash
-rw-rw-r-- 1 rm553022 rm553022 129999 Jun 6 23:38 minha_wordlist.txt
-rw-rw-r-- 1 rm553022 rm553022 150 Jun 6 23:29 organizador.py
-rw-rw-r-- 1 rm553022 rm553022 31903 Jun 6 23:27 texto.txt
-rw-rw-r-- 1 rm553022 rm553022 31903 Jun 6 23:33 wordlist.txt

rm553022@rm553022:~/Downloads
$ ls -l
total 240
-rw-rw-r-- 1 rm553022 rm553022 37788 Jun 6 23:27 file.pdf
-rw-rw-r-- 1 rm553022 rm553022 202 Jun 6 23:40 hash
-rw-rw-r-- 1 rm553022 rm553022 129999 Jun 6 23:38 minha_wordlist.txt
-rw-rw-r-- 1 rm553022 rm553022 150 Jun 6 23:29 organizador.py
-rw-rw-r-- 1 rm553022 rm553022 31903 Jun 6 23:27 texto.txt
-rw-rw-r-- 1 rm553022 rm553022 31903 Jun 6 23:33 wordlist.txt

rm553022@rm553022:~/Downloads
$ john --wordlist=minha_wordlist.txt hash
```

Fonte – Kali Linux.

Figura 26 – Texto.txt.



Fonte – Kali Linux.

Após a conclusão dos comandos do john, tem o seguinte valor em (?), “scooter1983”, ao utilizar ela no “file.pdf”.

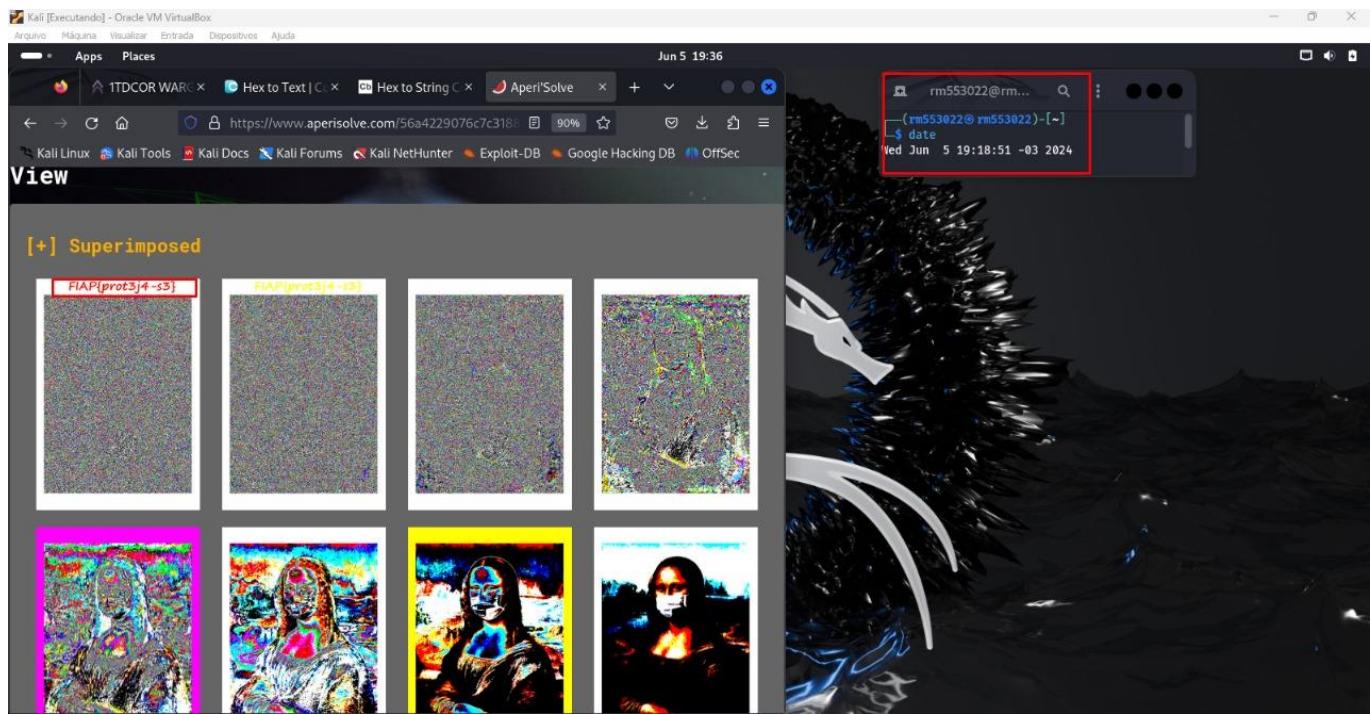
Dessa maneira a resposta então foi obtida de forma rápida a Flag:

FIAP{easy\_brute\_force}.

## 10. ESTRANHA IMAGEM

Nesse CTF, é necessário analisar a imagem, para obter a Flag.

Figura 27 – Fiap.png.



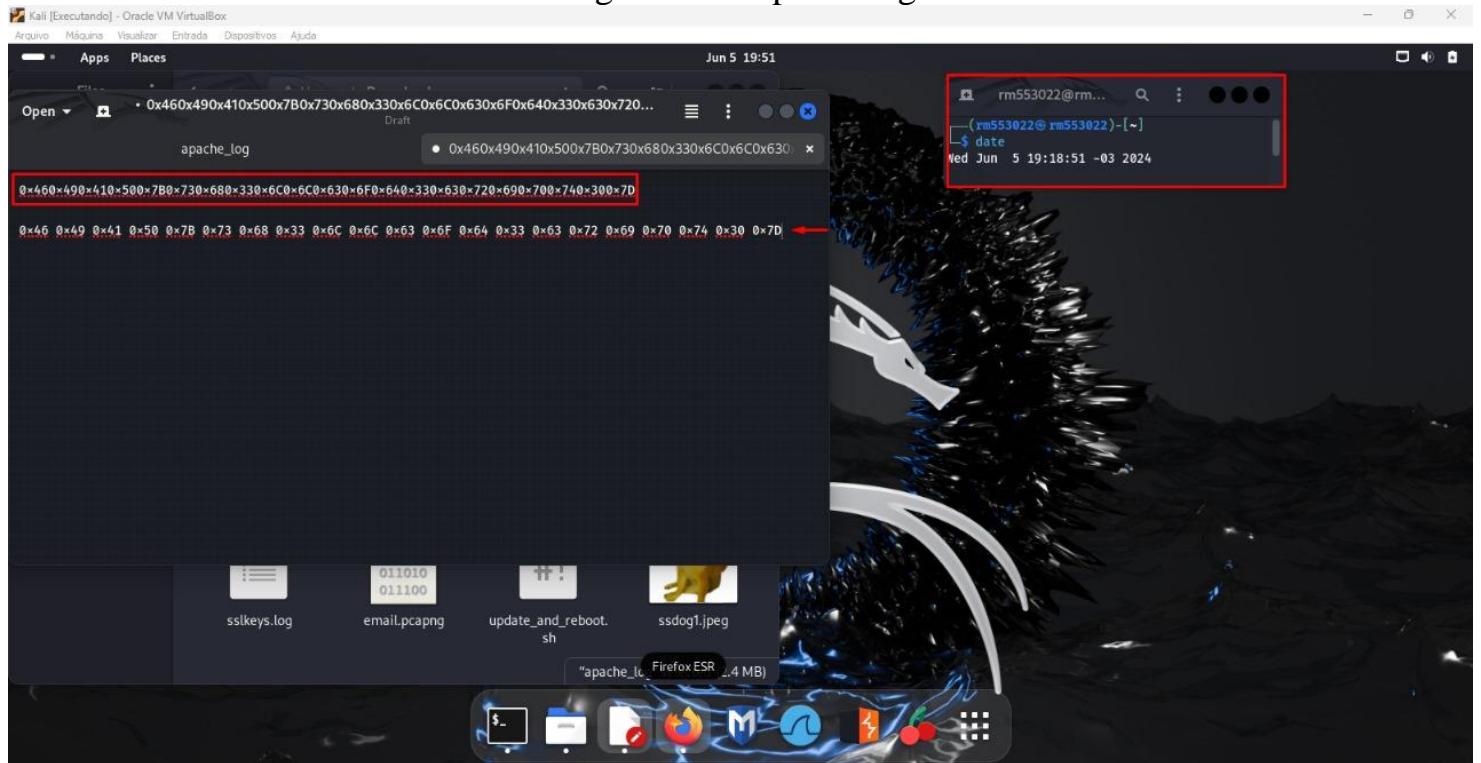
Fonte – Aperi'Solve

Dessa maneira a resposta então foi obtida de forma rápida a Flag: FIAP{prot3j4-s3}

## 11. CIFRAGEM FRACA

Nesse CTF, é necessário analisar o arquivo de log, para obter a Flag.

Figura 28 – Apache.log

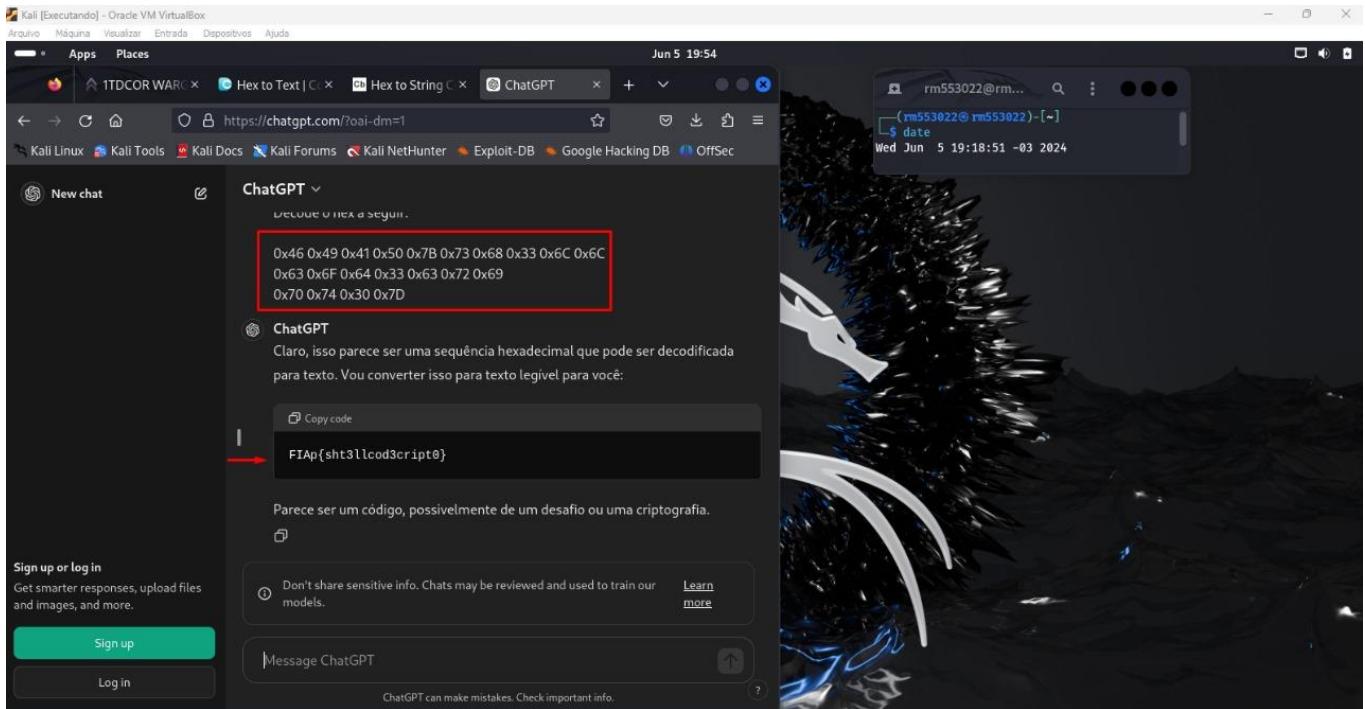


Fonte – Kali Linux.

Após análise dos logs, foi localizado um hexadecimal:

0x460x490x410x500x7B0x730x680x330x6C0x6C0x630x6F0x640x330x630x720x690x700x740x300x7D

Figura 29– Apache.log



Fonte – Chatgpt.

Após separar ele:

0x46 0x49 0x41 0x50 0x7B 0x73 0x68 0x33 0x6C  
0x6C 0x63 0x6F 0x64 0x33 0x63 0x72 0x69 0x70 0x74 0x30 0x7D.

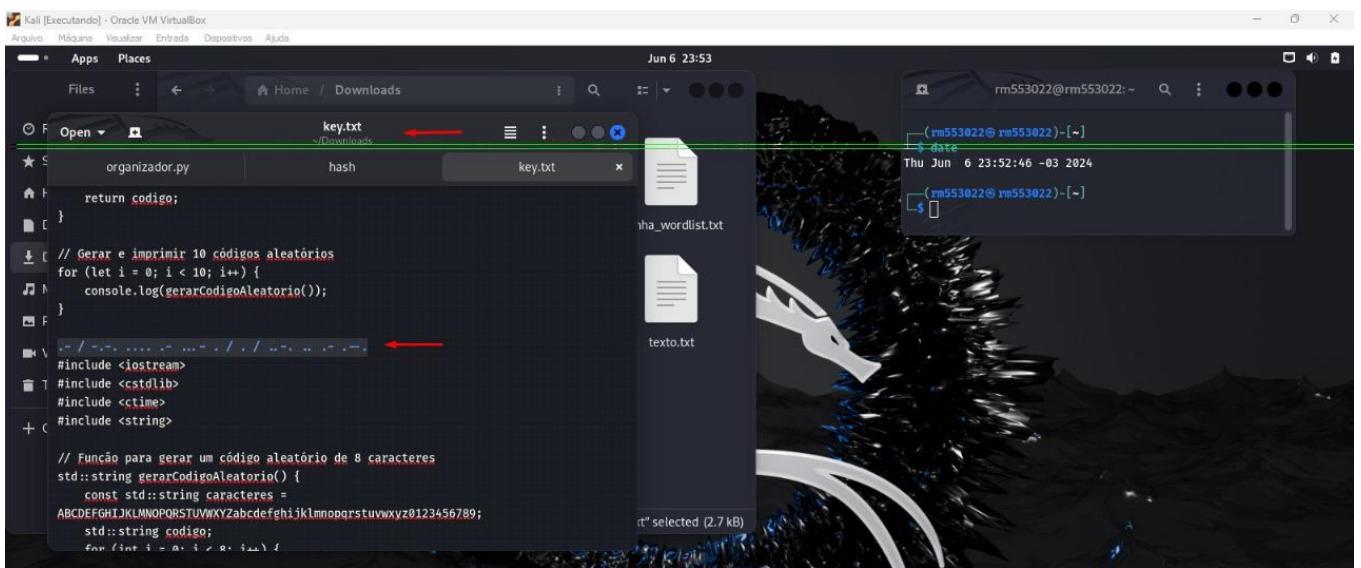
Foi solicitado o decode pelo prompt do chatgpt.

Dessa maneira a resposta então foi obtida de forma rápida a Flag:  
FIAP{sh3llcod3cript0}.

## 12. DESENCODA

Neste CTF, o objetivo é analisar a criptografia e obter a informação sigilosa por de trás dela, são dois arquivos, uma key e um que parece estar codificado.

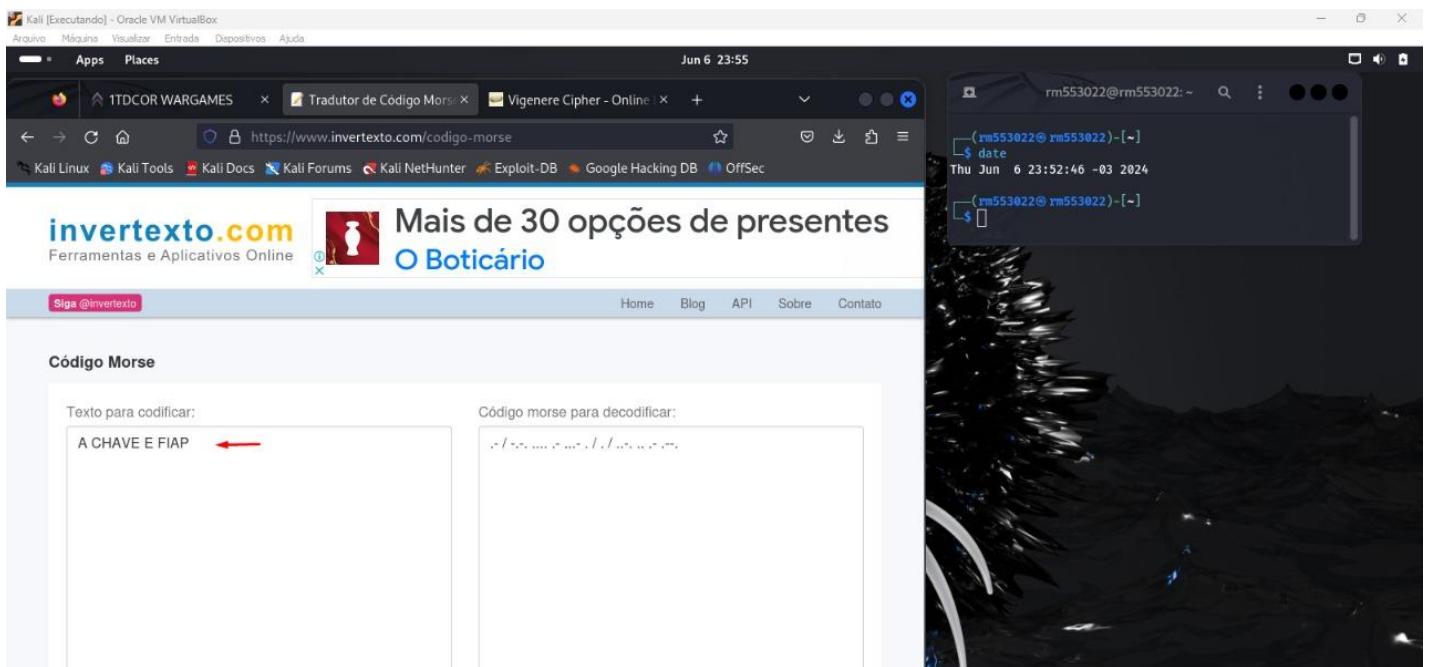
Figura 30 – Key.txt.



Fonte – Kali Linux

Ao decodar o código morse, temos a seguinte expressão:  
“A CHAVE E FIAP”.

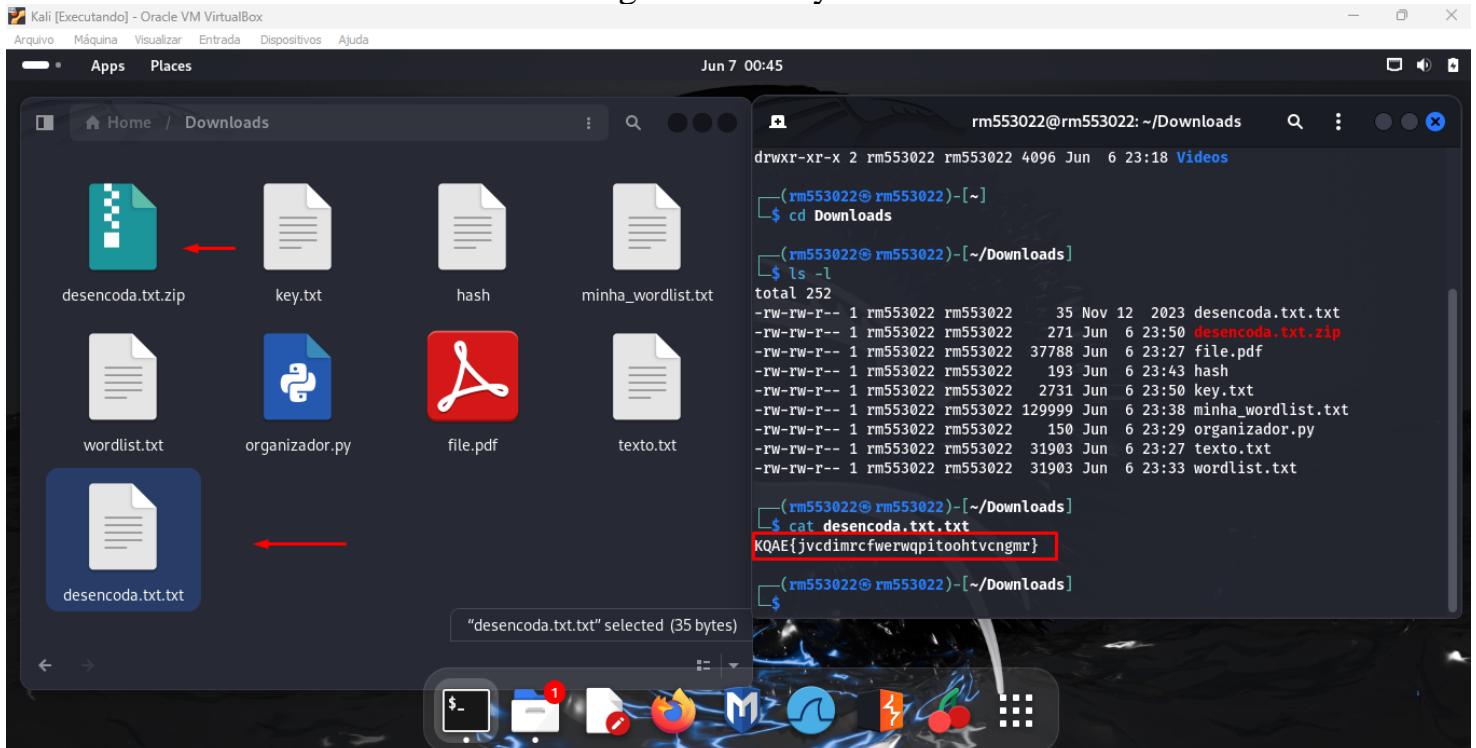
Figura 31 – Key.txt.



Fonte – Kali Linux

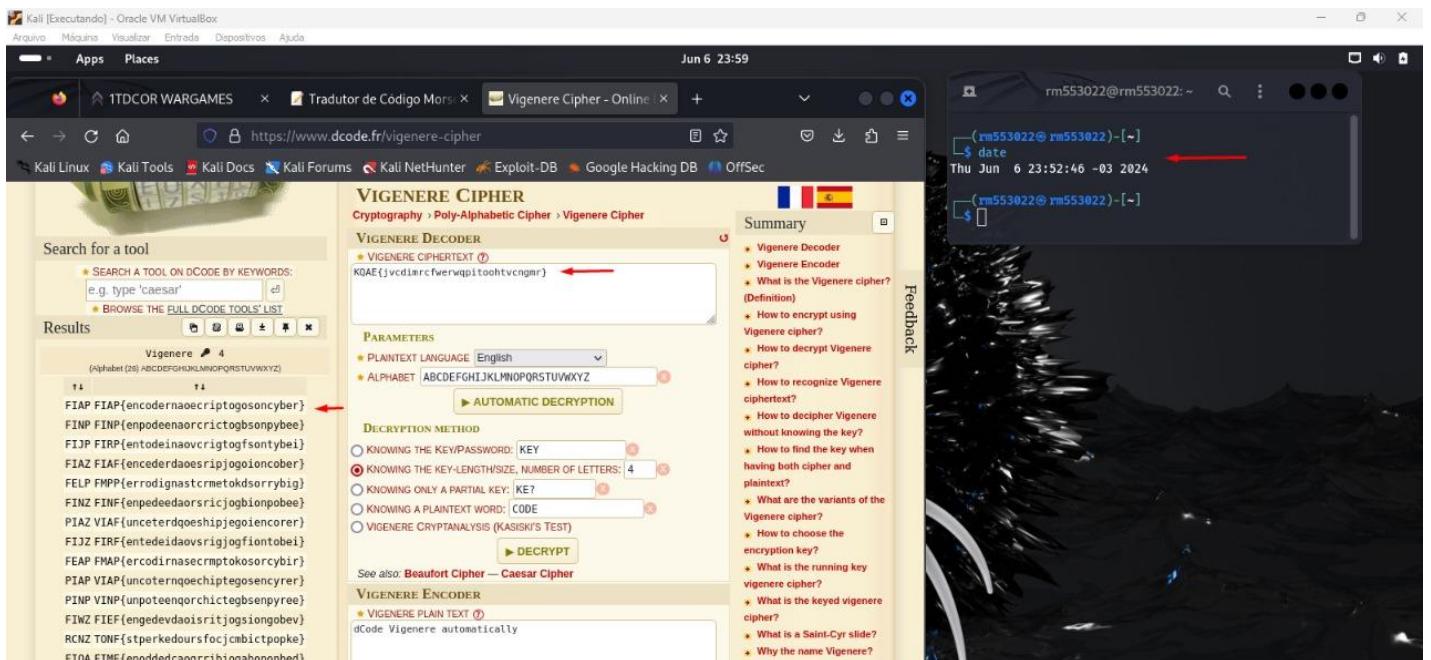
A senha é “fiap”, ela é utilizada para extrair o arquivo “desencoda.txt”.

Figura 32 – Key.txt.



Fonte – Kali Linux

Figura 33 – Key.txt.



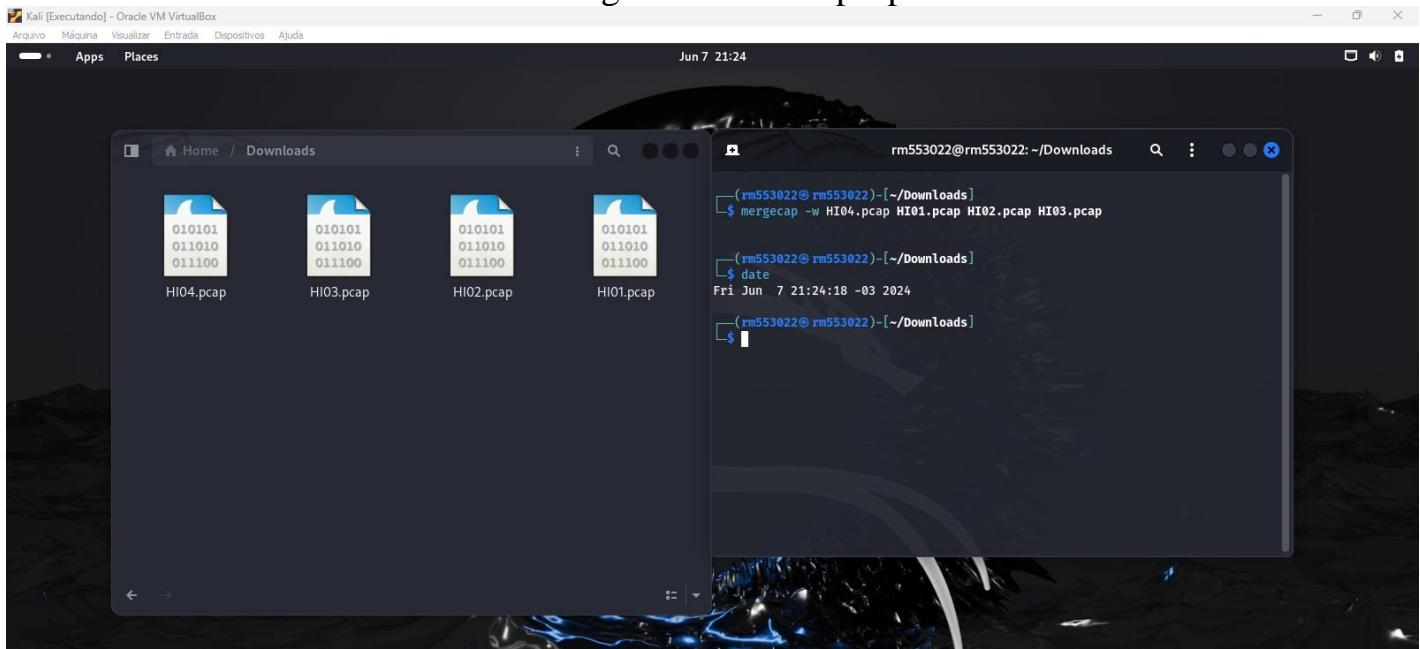
Fonte – dcode.fr.

Dessa maneira a resposta então foi obtida de forma rápida a Flag:  
FIAP{encodernaoecryptogosoncyber}.

## 13. HIDDEN

Para resolver esse CTF temos 3 arquivos HI01.pcap, HI02.pcap e HI03.pcap para analisarmos. Ao abrir os arquivos identifiquei uma comunicação entre dois hosts no protocolo ICMP. Após analisar bem, suspeitei que tenham contido cargas de dados. Para podermos prosseguir com o CTF mergemos os 3 arquivos e colocamos a saída no arquivo output.pcap com o comando mergecap -w output.pcap HI01.pcap HI02.pcap HI03.pcap.

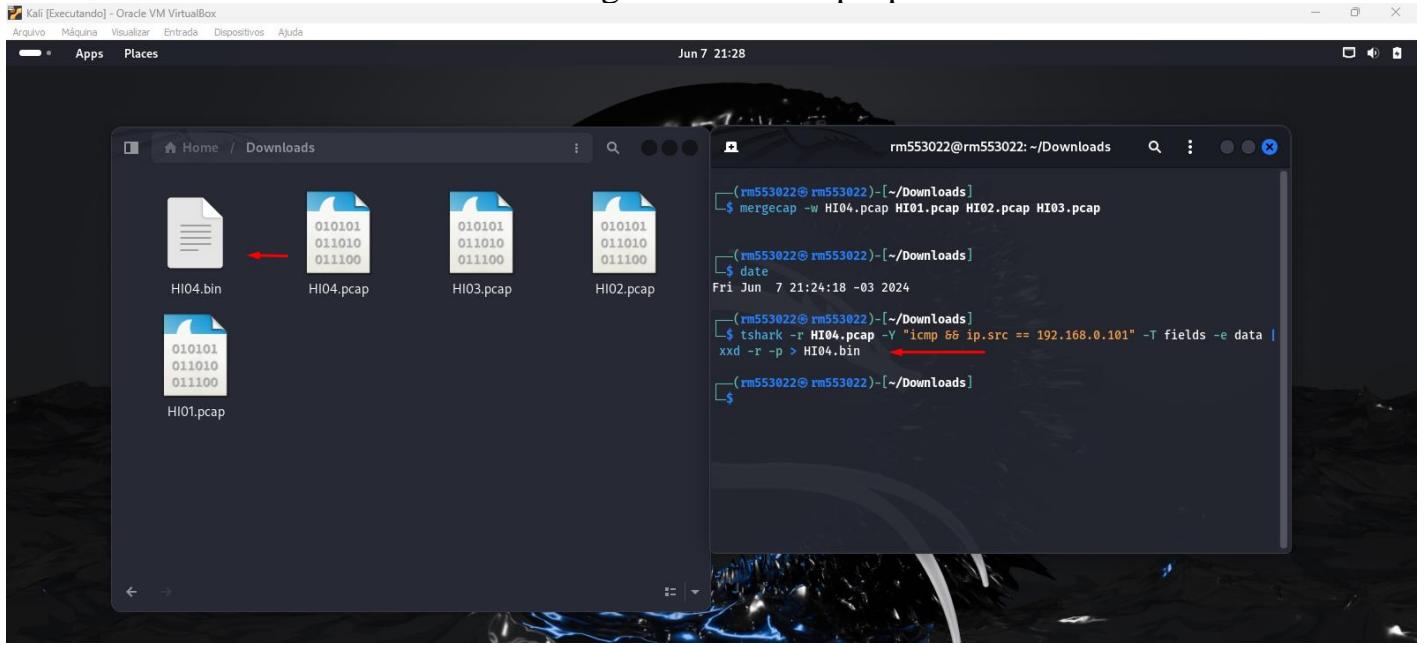
Figura 34 – Hi04.pcap



Fonte – Kali Linux

Assim que mergemos os 3 arquivos precisamos rodar o comando tshark -r output.pcap -Y "icmp && ip.src == 192.168.0.101" -T fields -e data | xxd -r -p > output.bin.

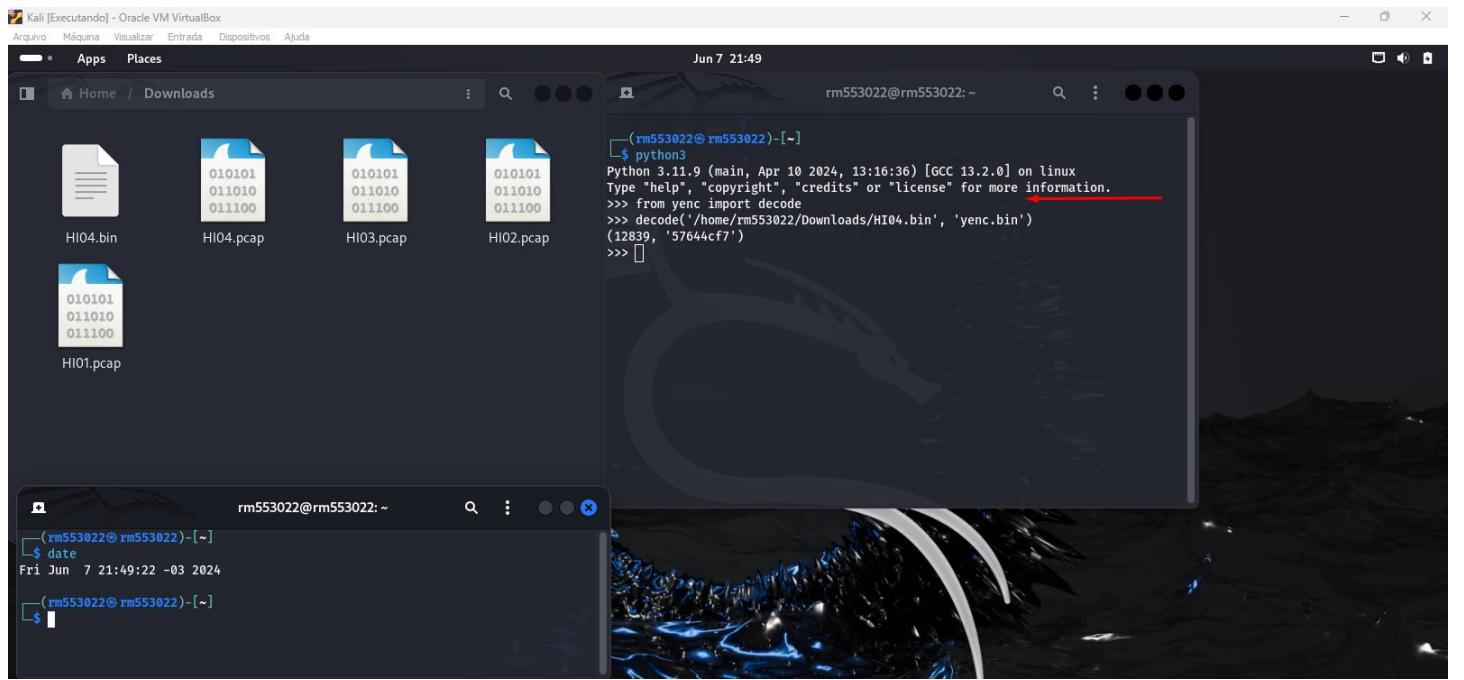
Figura 35 – Hi04.pcap



Fonte – Kali Linux

Agora com o arquivo output.bin usaremos o yEnc que é uma biblioteca Python para decodificar arquivos, para isso, usaremos o comando `from yenc import decode`  
`decode('/home/gabrielrm552788/Desktop/output.bin', 'yenc.bin` salvando o conteúdo decodificado no arquivo `yenc.bin`.

Figura 36 – Hi04.bin

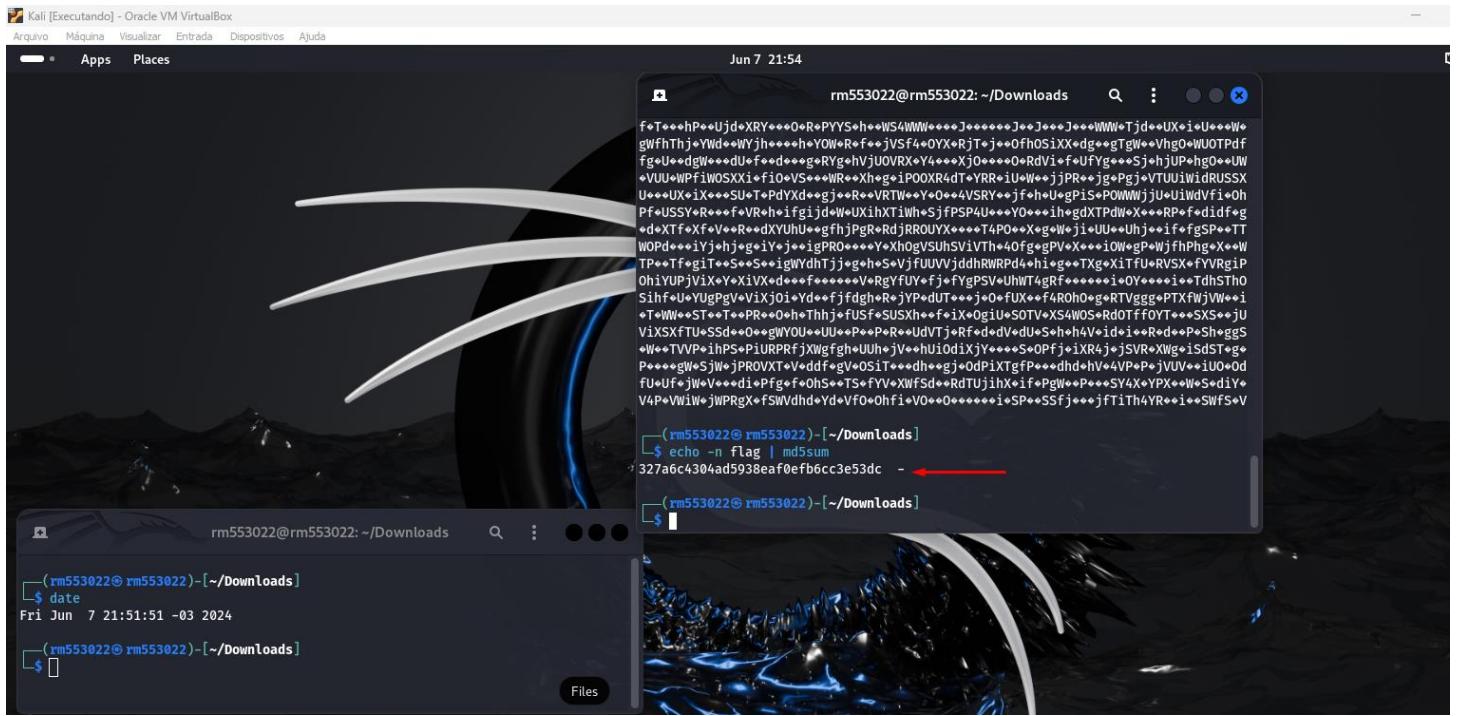


Fonte – Kali Linux

Ao inspecionar o conteúdo decodificado no arquivo yenc.bin encontramos o seguinte código base64 ZWNobyAiZiIKZNobyAibCIKZNobyAiYSIKZNob  
yAiZyIKZNobyAieyIKZNobyAibWQ1c3VtIgplY2hvICIoJ2ZsYWcnKSIK  
ZWNobyAifSIK.

Ao traduzi-lo encontramos a seguinte mensagem. echo "f" echo "l" echo "a" echo "g" echo "{"  
echo "md5sum" echo "('flag')" echo "}"

Figura 37 – Hi04.bin



Fonte – Kali Linux

A mensagem nos traz a informação de um hash MD5sum, para resolver essa parte do desafio, foi necessário rodar esse comando echo -n flag | md5sum que nos traz o seguinte resultado com a flag{327a6c4304ad5938eaf0efb6cc3e53dc}.

Dessa maneira a resposta então foi obtida de forma rápida a Flag: FIAP{327a6c4304ad5938eaf0efb6cc3e53dc}.

## **14. CONCLUSÃO**

Este relatório detalhou de maneira minuciosa a resolução de diversos desafios no âmbito do Capture The Flag (CTF) propostos durante a fase atual da Global Solution no 2º Semestre do curso. Cada desafio, pertencente aos cenários propostos, foi abordado com análises passo-a-passo, enriquecidas com imagens e textos, proporcionando uma compreensão aprofundada do processo de resolução. A abordagem adotada incluiu a exploração de conceitos fundamentais em protocolos de rede, como a análise de arquivos e diretórios e interpretação de códigos base64. Os procedimentos foram conduzidos de maneira a contextualizar não apenas os passos práticos, mas também a lógica e a estratégia por trás de cada decisão. Além dos desafios de conceitos fundamentais de conhecimento de protocolos, enfrentamos cenários relacionados a redes, utilizando a ferramenta Wireshark para análise de pacotes. A resolução desses desafios exigiu a aplicação de filtros específicos e a análise cuidadosa do tráfego de rede para identificar informações relevantes. Ainda, abordamos desafios relacionados a quebra de senhas e decodificação em hash MD5, e também conhecimento na linguagem Python. Esses cenários proporcionaram uma oportunidade de aplicar técnicas de quebra de senhas e análise de formatos de codificação. Em cada desafio, a obtenção das flags foi bem-sucedida, demonstrando a aplicação prática dos conhecimentos adquiridos durante o curso. Cada procedimento foi documentado de acordo com as normas ABNT, proporcionando uma estrutura clara e organizada para a apresentação dos resultados. Este relatório busca não apenas apresentar os resultados, mas também fornecer uma narrativa abrangente que destaca o pensamento crítico, a tomada de decisões estratégicas e a habilidade de enfrentar desafios diversos no contexto de cibersegurança.