Đại học Quốc gia Thành phố Hồ Chí Minh
Trường Đại học Công nghệ Thông tin (UIT)

BÁO CÁO BÀI TẬP

Môn học: Bảo mật Web và Ứng dụng
Kỳ báo cáo: Buổi 03 (Session 03)
Tên chủ đề: Steganography & Steganalysis
GVHD: Đoàn Minh Trung

1. THÔNG TIN CHUNG:

(Liệt kê tất cả các thành viên trong nhóm)
Lớp: NT334.N21.ATCL

STT	Họ và tên	MSSV	Email
1	Lê Viết Tài Mẫn	20521593	20521593@gm.uit.edu.vn
2	Hoàng Thanh Lâm	20521513	20521513@gm.uit.edu.vn
3	Vũ Hoàng Thạch Thiết	20521957	20521957@gm.uit.edu.vn

2. NỘI DUNG THỰC HIỆN:[1]

STT	Công việc	Kết quả tự đánh giá	Người đóng góp
1	Kịch bản 01		
2	Kịch bản 02		
3	Kịch bản 03		
4	Kịch bản 04		
5	Kịch bản 05		
6	Kịch bản 06		
7	Kịch bản 07		
8	Kịch bản 08		

Phần bên dưới của báo cáo này là tài liệu báo cáo chi tiết của nhóm thực hiện.

[1] Ghi nội dung công việc, các kịch bản trong bài Thực hành

BÁO CÁO CHI TIẾT

1. Kịch bản 06

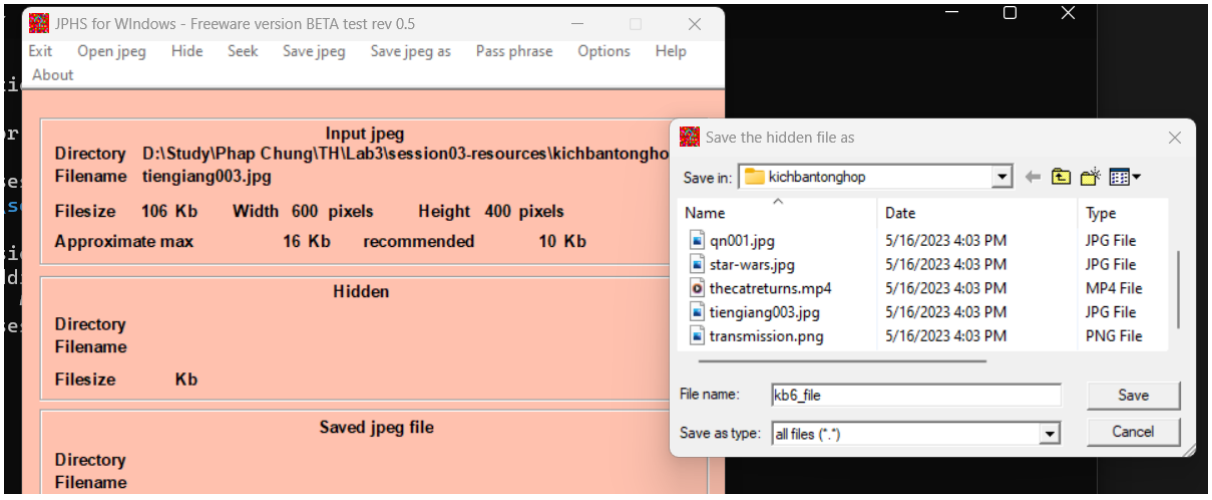
Kịch bản 06. Thực hiện phân tích:

- Tài nguyên: tiengiang003.jpg
- Yêu cầu – Gợi ý: Tìm thông điệp (flag) được ẩn giấu. Thuật toán dùng tìm ra flag liên quan đến việc thay thế các kí tự trong chuỗi ban đầu thành chuỗi chỉ gồm 2 kí tự a và b.

Đáp án:

```
PS D:\Study\Phap Chung\TH\Lab3\session03-resources\stegdetect04_session03> .\stegbreak.exe -r .\rules.ini -f .\rockyou.t
xt "D:\Study\Phap Chung\TH\Lab3\session03-resources\kichbantonghop\tiengiang003.jpg"
Loaded 1 files...
D:\Study\Phap Chung\TH\Lab3\session03-resources\kichbantonghop\tiengiang003.jpg : jphide[v5]()
Processed 1 files, found 1 embeddings.
Time: 1 seconds: Cracks: 4751,   4751.0 c/s
PS D:\Study\Phap Chung\TH\Lab3\session03-resources\stegdetect04_session03>
```

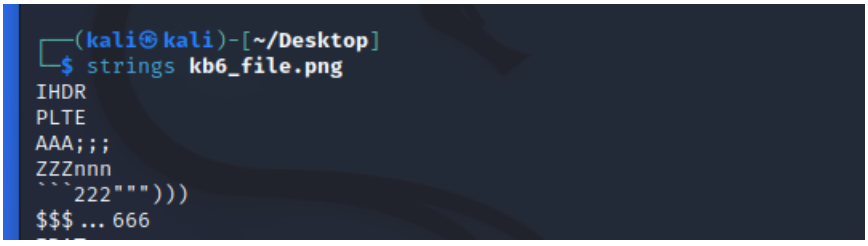
- Sử dụng stegbreak với bộ wordlist rockyou để kiểm tra
- Ta thấy có 1 file ẩn

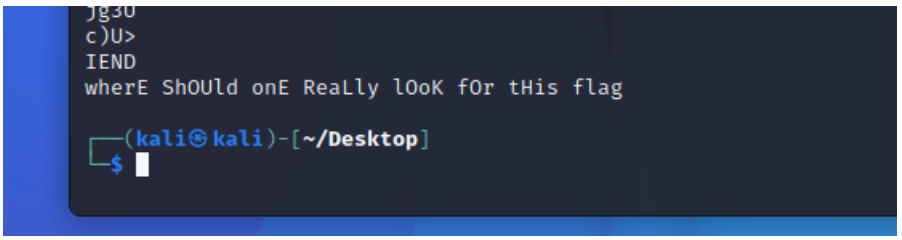


- Sử dụng JPHS để trích xuất và đặt tên là kb6_file

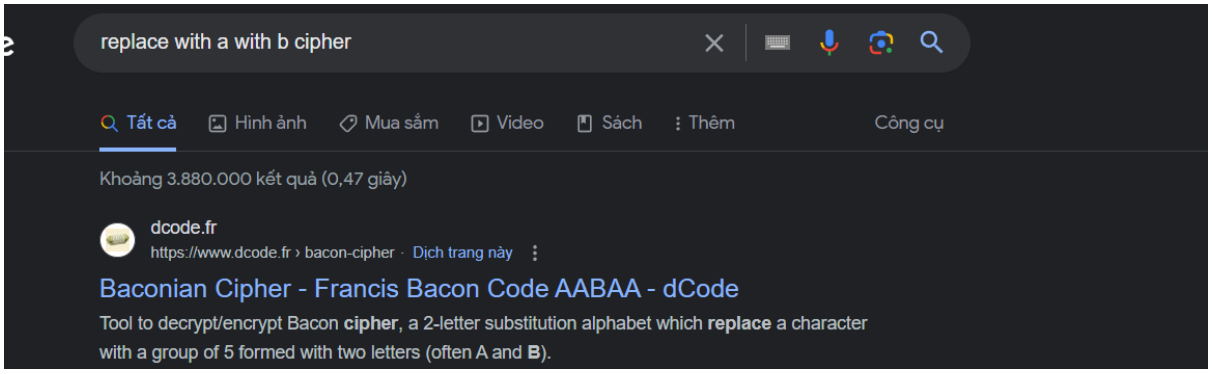
- Ta xem định dạng file là file png

- Ta đổi tên file thành kb6_file.png và mở lên là hình ảnh chứa dòng chữ "Harry Potter"





- Sử dụng lệnh strings để lấy các chuỗi đọc được thì có 1 chuỗi là "wherE ShOUld onE ReaLly lOoK fOr tHis flag"
- Đây có thể là flag nhưng có thể nó đã bị mã hóa dựa theo gợi ý của đề bài
- Giờ ta cần tìm thuật toán để có thể giải mã



- Tìm kiếm thì biết đây là loại cipher baconian
- được sử dụng để mã hóa các thông điệp bằng cách thay thế mỗi chữ cái trong văn bản bằng một chuỗi gồm các ký tự "A" và "B". Mỗi chữ cái trong bảng chữ cái được ánh xạ tương ứng với một chuỗi gồm 5 ký tự "A" hoặc "B".
- Tham khảo link : [Steganography: Because Who Doesn't Love Bacon? | Ball in your Court (craigball.net)]
- Thì ta có thể biết được 1 số cách để áp dụng baconian cipher vào steganography của ta
- Theo chuỗi của ta thì có thể nhận thấy là ký tự thường và ký tự hoa cũng giống như dạng binary
- Ta giả định rằng chữ chường là "A" và chữ hoa là "B" thì chuỗi của ta sẽ thành
  ➔ AAAAB BABBAA AAB BAABAA ABAB ABA ABAA AAAA

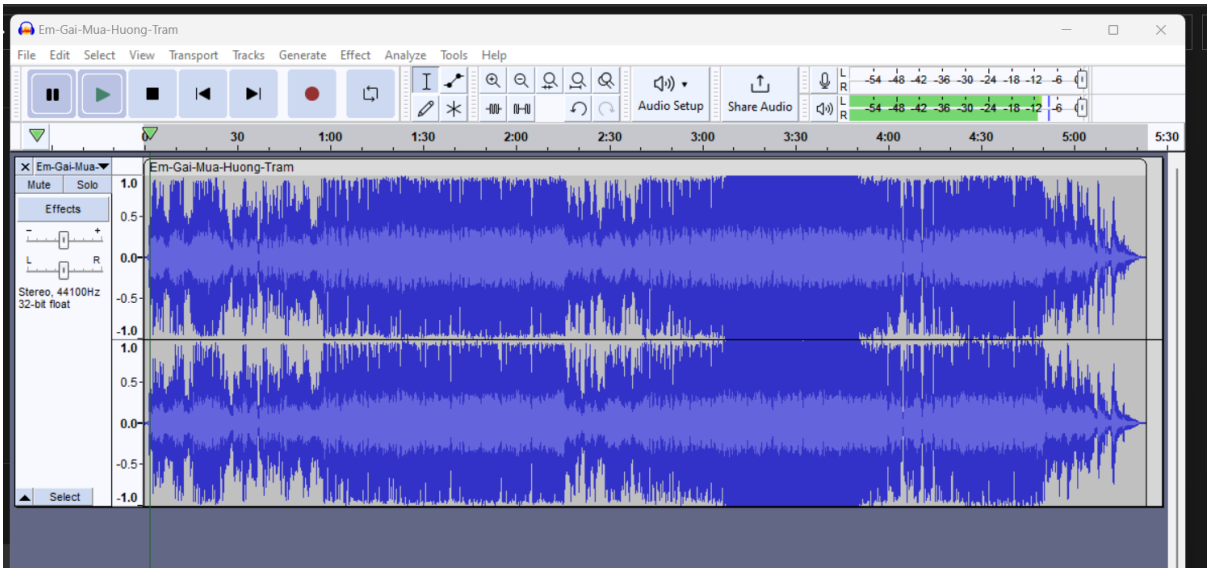
- Đưa vào để giải mã
  ➔ Flag: BYDELTA

2. Kịch bản 07

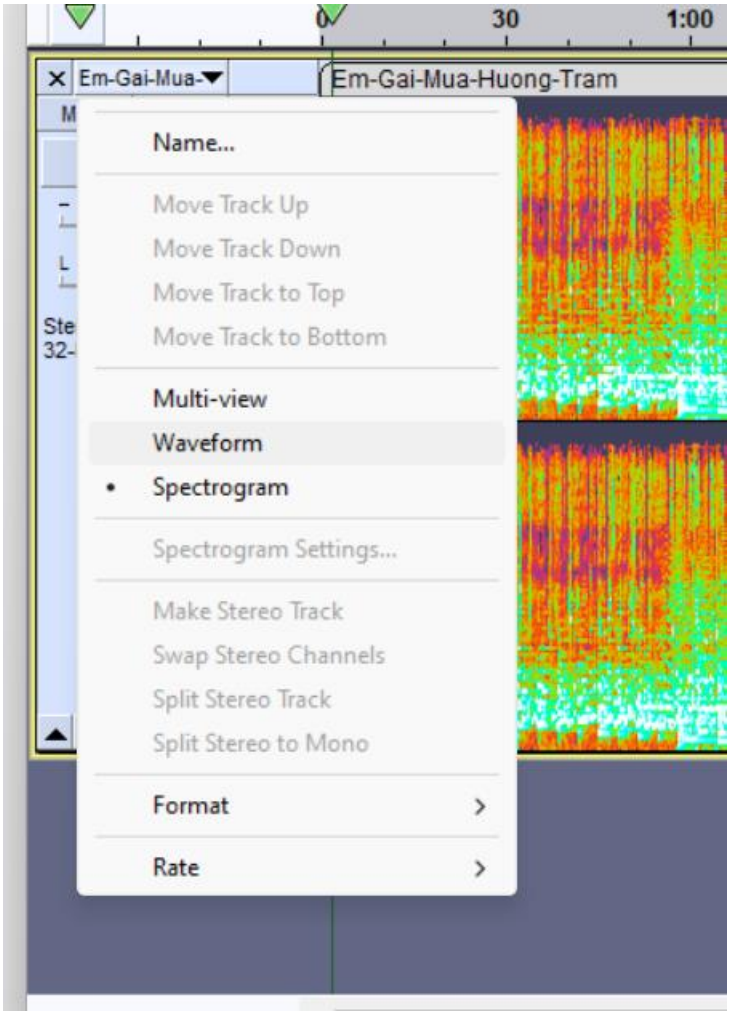
Kịch bản 07. Thực hiện phân tích, tìm thông tin ẩn giấu:

- Tài nguyên: kb07-res (Tìm thông tin ẩn giấu trong Em-Gai-Mua-Huong-Tram.mp3, capture-the-flag.png)

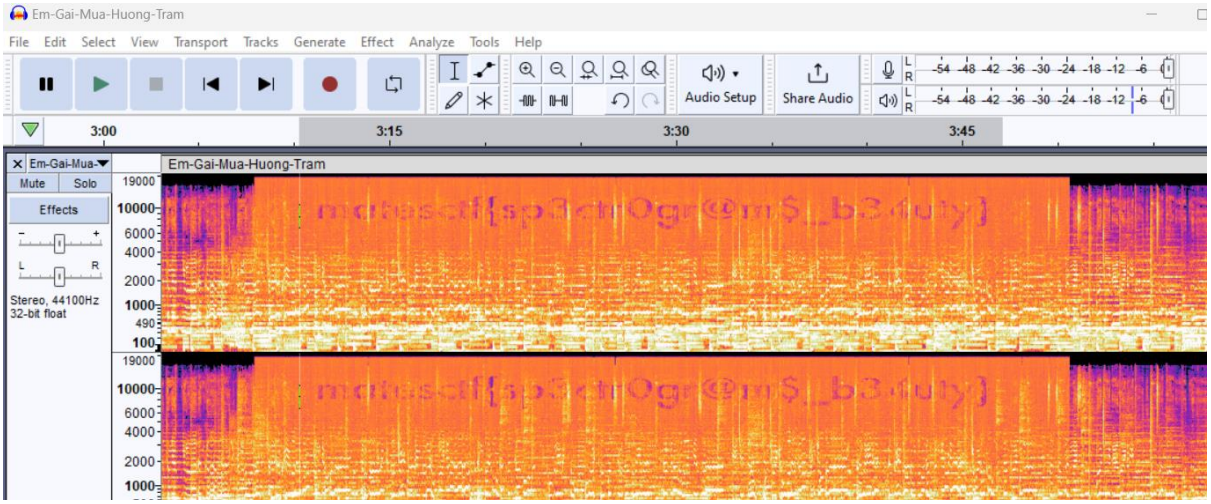
Đáp án:



- Tham khảo link: [Steganography - CTF Playbook (gitbook.io)]
- Ta sử dụng phần mềm adacity để có thể xem file dưới dạng spectrogram
- Mặc định thì audacity sẽ hiển thị dưới dạng waveform



- Ta chỉnh nó sang dạng spectrogram



- Khi zoom lên thì ta thấy có 1 dòng thông điệp được ẩn dấu trong file âm thanh này

➔ Flag: matesctf{sp3ctr0gr@m$_b34uty}

3. Kịch bản 08:

Kịch bản 08. Thực hiện phân tích, tìm thông tin ẩn giấu:

- Tài nguyên: LoveLetter.txt
- Yêu cầu – Gợi ý: Có gì đó đáng ngờ trong bức thư tình mà bạn đang đọc. Nhân viên điều tra cũng nghĩ rằng bức thư tình này chứa một thông điệp bí mật nào đó. Hãy tìm thông điệp được ẩn giấu (flag). Flag có dạng "FLAG-*"
- Link CTF: https://ringzer0ctf.com/challenges/215

- Đầu tiên ta sử dụng HxD để xem raw của file này:

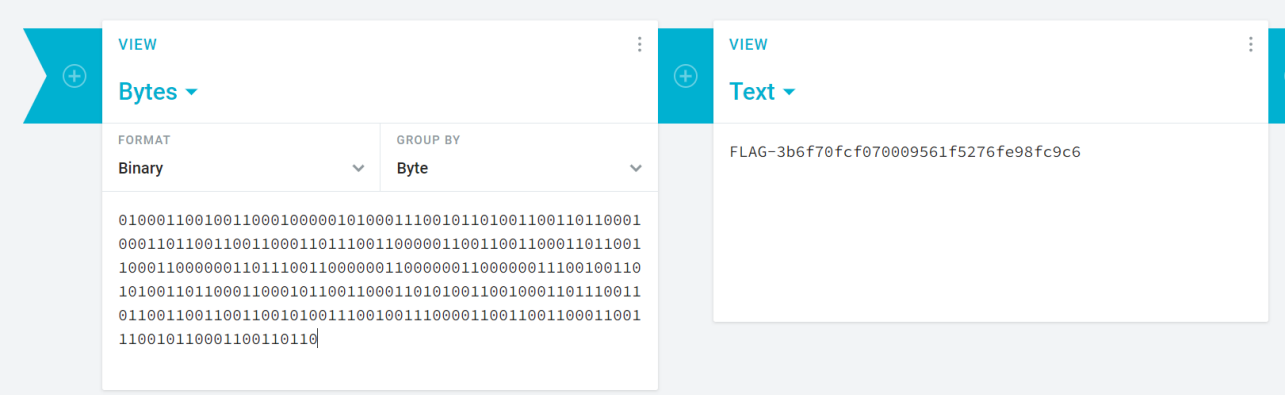
LoveLetter.txt

Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Decoded text
00000000	49	20	77	65	6E	74	A0	74	6F	20	74	68	65	20	70	61	I went to the pa
00000010	72	6B	20	74	6F	64	61	79	2C	A0	73	61	77	A0	61	20	rk today, saw a
00000020	6C	6F	74	20	6F	66	A0	66	69	73	68	2E	20	46	69	73	lot of fish. Fis
00000030	68	20	61	72	65	A0	63	6F	6F	6C	2C	A0	62	75	74	20	h are cool, but
00000040	74	68	65	79	20	61	72	65	6E	27	74	20	6D	79	A0	66	they aren't my f
00000050	61	76	6F	72	69	74	65	20	61	6E	69	6D	61	6C	21	21	avorite animal!!

- Ta để ý là khoảng trắng có tới tận 2 dạng hex là 20 và A0
- Sau một hồi tìm tòi và tham khảo thì ta biết được cách giải là tạo một chuỗi binary từ các ký tự khoảng trắng này và sau đó giải mã theo hex (mã hóa 8 bit tương ứng 1 ký tự).
- Ta chia làm 2 trường hợp:
  - TH1: Set khoảng trắng 20 là 0, khoảng trắng A0 (160) là 1
  - TH2: Set khoảng trắng 20 là 1, khoảng trắng A0 (160) là 0
- Đối với TH1 ta có được chuỗi binary là:

01000110010011000100000101000111001011010011001101100010001101100110011000110111001100000110011001100011011001100011000000110111001100000011000000110000001110010011010100110110001100010110011000110101001100100011011100110110001100110011001010011100100111000011001100110001100111001011000110011001100

Sau khi giải theo hex thì ta có kết quả sau chính là flag:



- Vậy flag là **FLAG-3b6f70fcf070009561f5276fe98fc9c6** -> Không cần quan tâm TH2

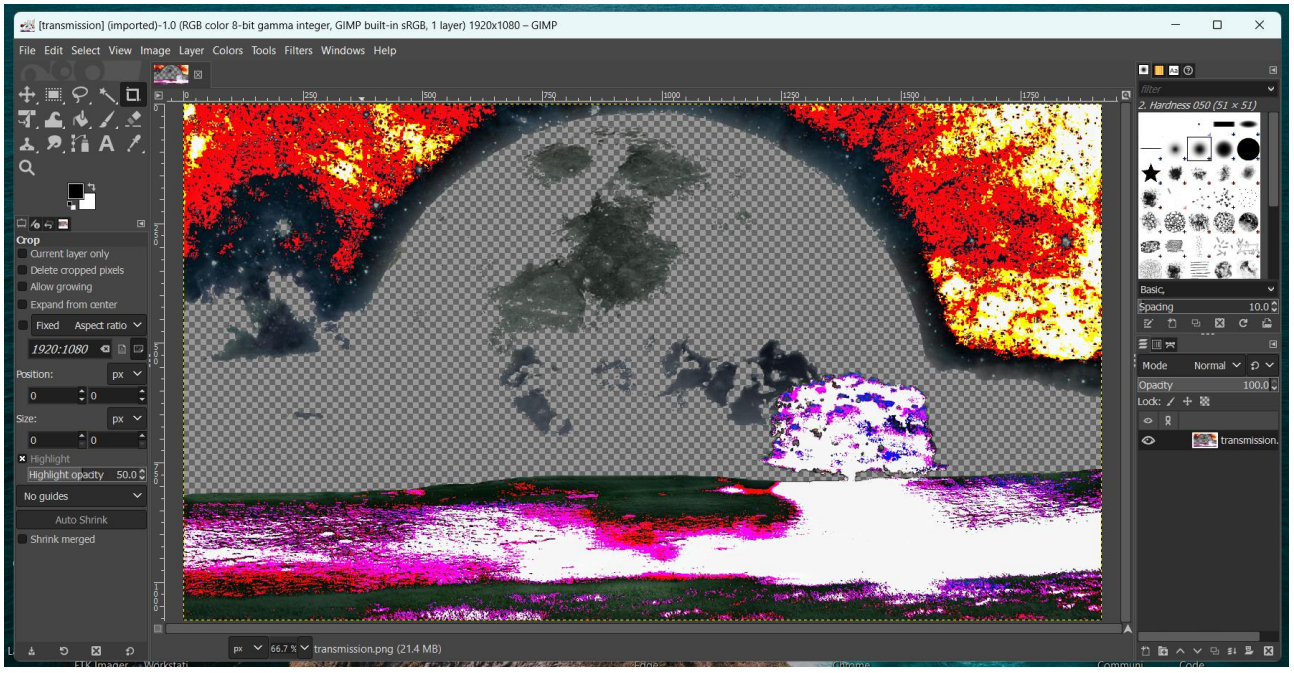
4. Kịch bản 09:

Kịch bản 09. Thực hiện phân tích, tìm thông tin ẩn giấu:

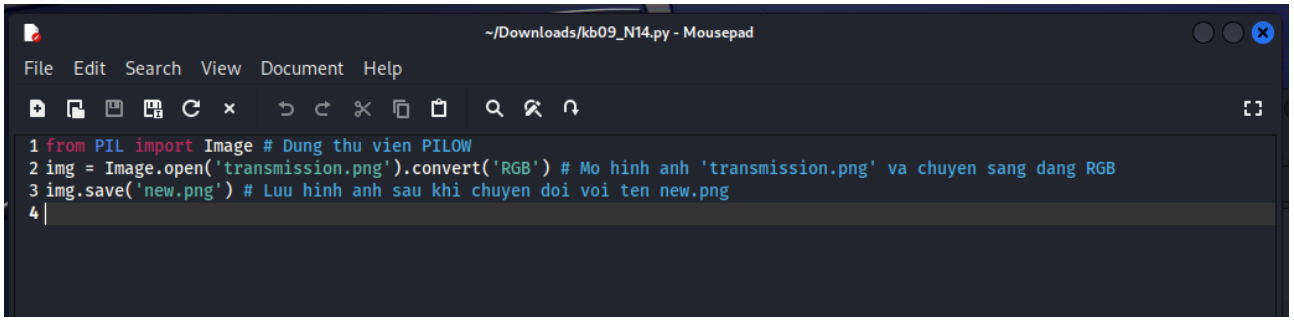
- Tài nguyên: transmission.png
- Yêu cầu – Gợi ý: Tìm thông điệp được ẩn giấu bằng các công cụ đã học trong buổi này.

Đáp án:

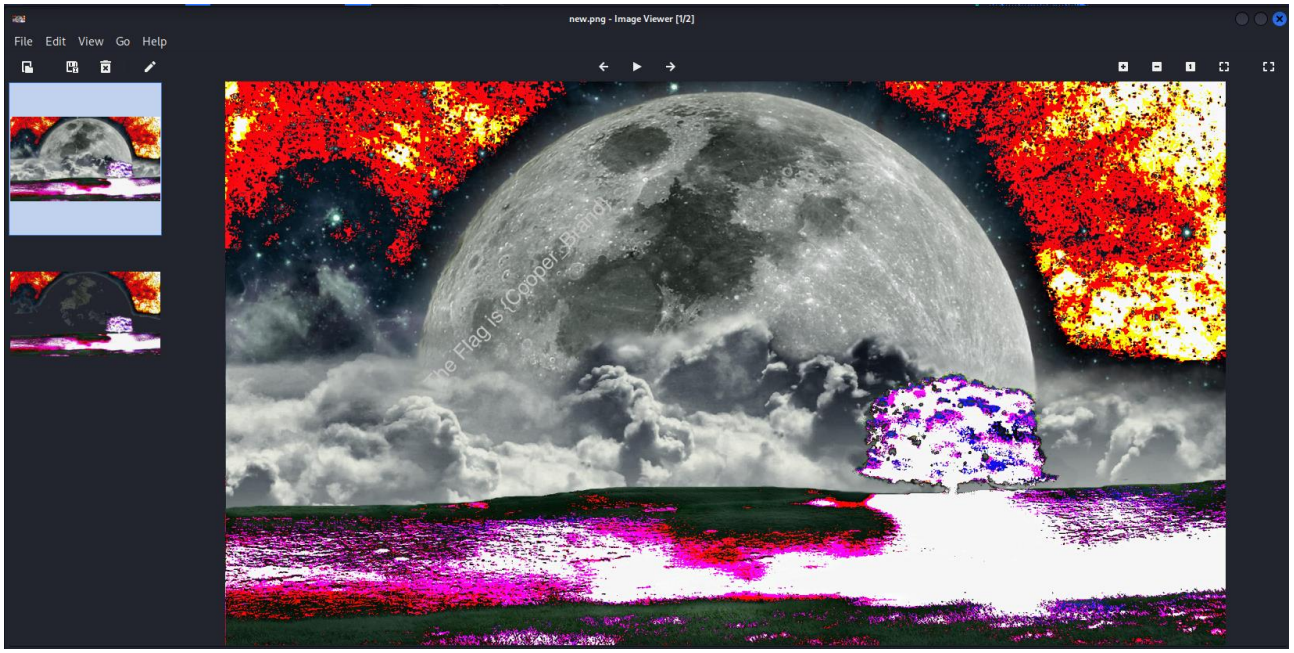
- Khi ta mở ảnh bằng GIMP thì kết quả như sau:



- Có một vùng bị làm trong suốt (transparent). Ta sẽ thử chuyển đổi nó sang dạng RGB để xem thực chất hình ảnh này có chứa gì không.
- Ta tạo một file python có chức năng chuyển đổ như trên



- Kết quả sau khi chuyển đổi ta có được flag là **{Cooper_Brand}**



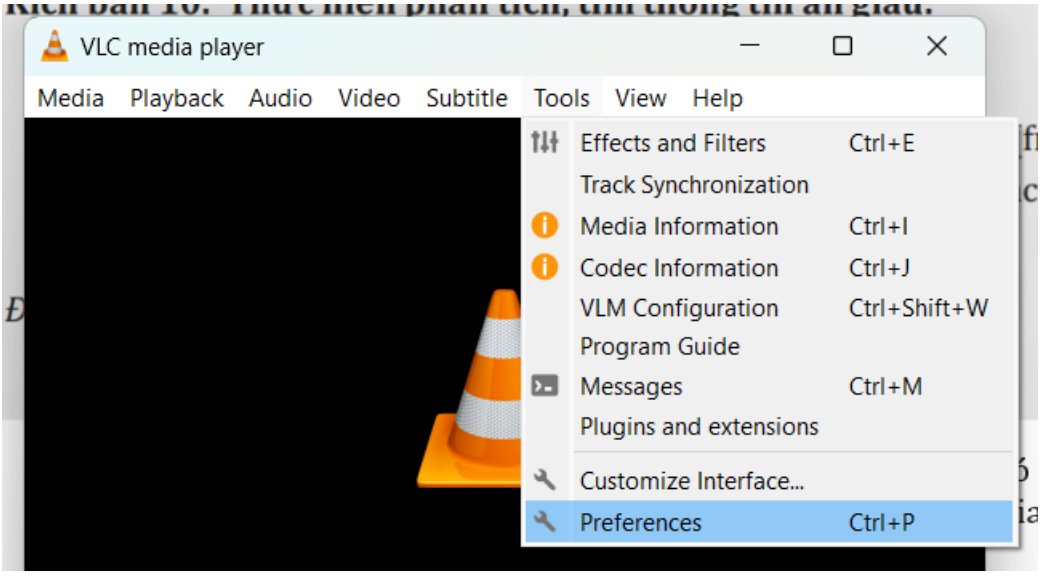
5. Kịch bản 10:

Kịch bản 10. Thực hiện phân tích, tìm thông tin ẩn giấu:

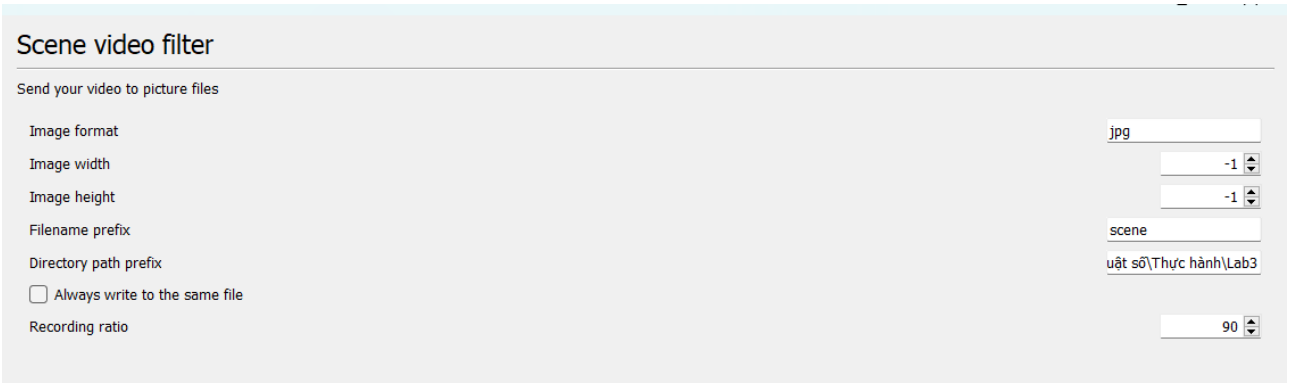
- Tài nguyên: thecatreturns.mp4
- Yêu cầu – Gợi ý: Tìm sự khác biệt giữa các khung hình (frame) trong đoạn phim đã cho. Chuyển nội dung đoạn phim thành các khung hình để phân tích. Công cụ ffmpeg, ImageJ.

Đáp án:

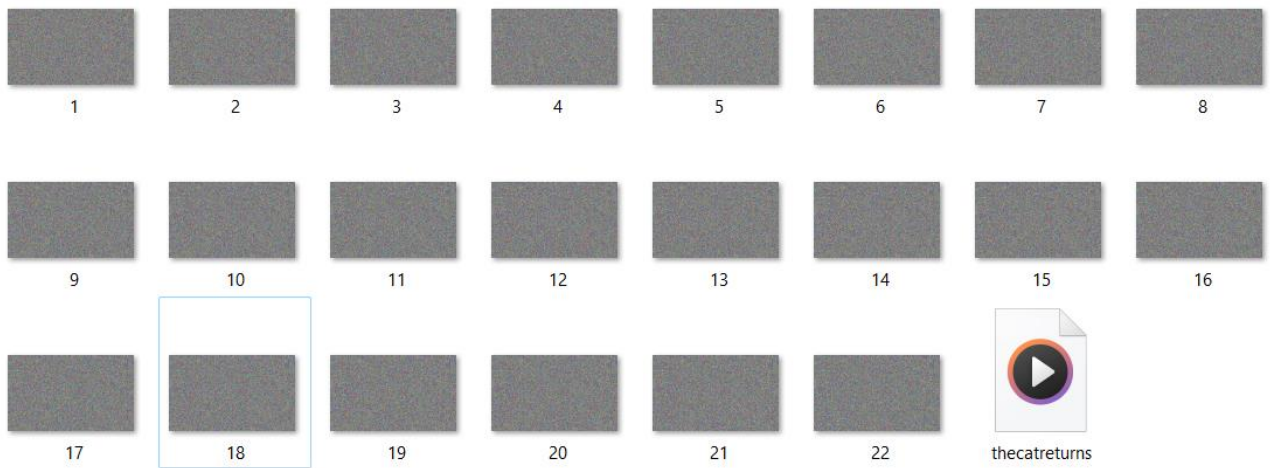
- Ta có thể dễ dàng nhận ra rằng video này bị làm nhiễu và cũng có thể nhận thấy một vài chuyển động nào đó trong video nên ý tưởng ở đây là chia video thành các frame dưới dạng hình ảnh.
- Công cụ VLC có thể giúp ta thực hiện công việc trên. Đầu tiên ta mở Preference lên như trong hình



- Tiếp theo chọn All ở Show Setting -> Video -> Filters -> Scene Filter
- Tại tab Scene video filter ta điền các thông tin như hình, ở chỗ Recording ratio ta để là 90 -> cứ 3s ta lưu 1 ảnh (do tốc độ khung hình của video là 30fps)

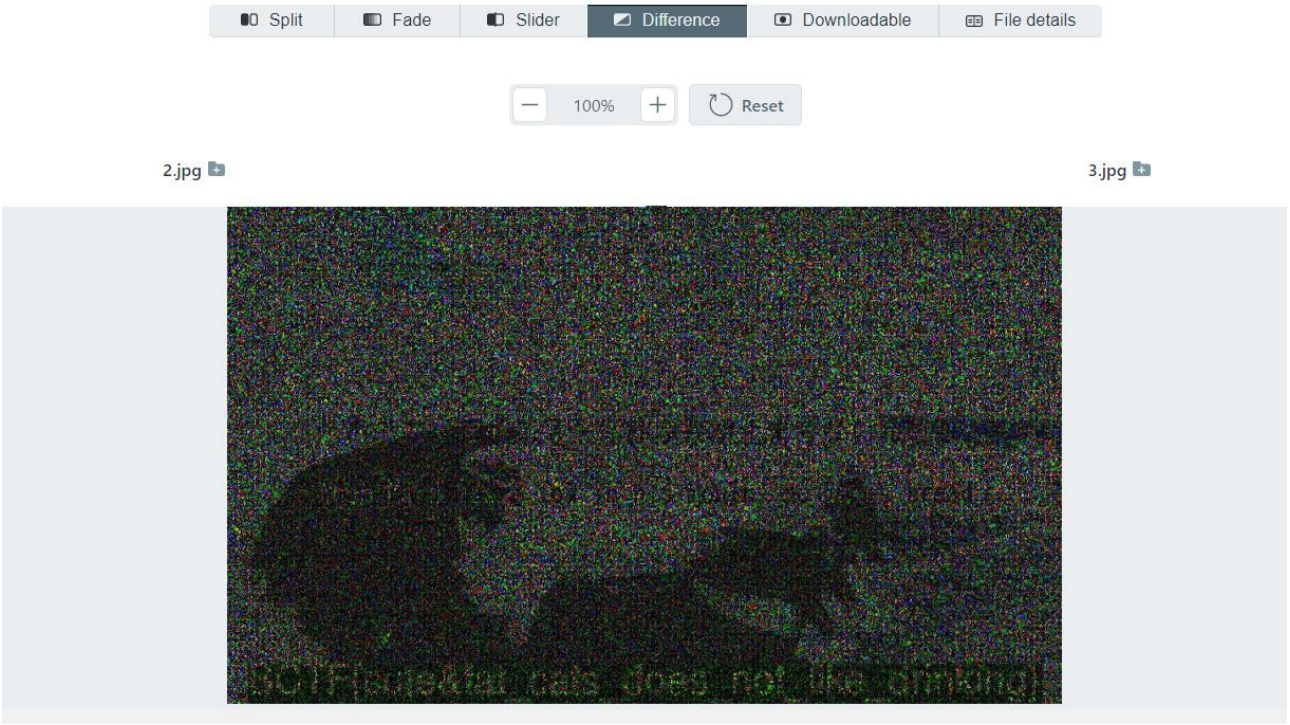


- Sau khi chia video ra hình ảnh theo frame ta đượck kết quả:



- Ta tiến hành so sánh 1 và 2 bằng công cụ diffchecker:

- Không có thông tin liên quan tới flag, ta tiếp tục so sánh 2 và 3:



- Tìm được flag là SCTF{cute&fat_cát_does_not_like_onnking}

Sinh viên đọc kỹ yêu cầu trình bày bên dưới trang này

YÊU CẦU CHUNG

- Sinh viên tìm hiểu và thực hành theo hướng dẫn.
- Nộp báo cáo kết quả chi tiết những việc (**Report**) bạn đã thực hiện, quan sát thấy và kèm ảnh chụp màn hình kết quả (nếu có); giải thích cho quan sát (nếu có).
- Sinh viên báo cáo kết quả thực hiện và nộp bài.

Báo cáo:

- File .PDF. Tập trung vào nội dung, không mô tả lý thuyết.
- Nội dung trình bày bằng Font chữ Times New Romans/ hoặc font chữ của mẫu báo cáo này (UTM Neo Sans Intel/UTM Viet Sach)– cỡ chữ 13. Canh đều (Justify) cho văn bản. Canh giữa (Center) cho ảnh chụp.
- Đặt tên theo định dạng: [Mã lớp]-SessionX_GroupY. (trong đó X là Thứ tự buổi Thực hành, Y là số thứ tự Nhóm Thực hành/Tên Cá nhân đã đăng ký với GV).

  *Ví dụ: [*NT101.K11.ANTT*]-Session1_Group3.*
- Nếu báo cáo có nhiều file, nén tất cả file vào file .ZIP với cùng tên file báo cáo.
- Không đặt tên đúng định dạng – yêu cầu, sẽ **KHÔNG** chấm điểm.
- Nộp file báo cáo trên theo thời gian đã thống nhất tại courses.uit.edu.vn.

Đánh giá: Sinh viên hiểu và tự thực hiện. Khuyến khích:

- Chuẩn bị tốt.
- Có nội dung mở rộng, ứng dụng trong kịch bản/câu hỏi phức tạp hơn, có đóng góp xây dựng.

Bài sao chép, trễ, … sẽ được xử lý tùy mức độ vi phạm.

HẾT