

# BÁO CÁO THỰC HÀNH

Môn học: Pháp chứng kỹ thuật số

Lab 2: Steganography & Steganalysis

GVHD: Đoàn Minh Trung

## 1. THÔNG TIN CHUNG:

(Liệt kê tất cả các thành viên trong nhóm)

Lớp: NT334.P21.ANTT.1

STT	Họ và tên	MSSV	Email
1	Trần Huỳnh Tiến	22521476	22521476@gm.uit.edu.vn
2	Nguyễn Ngọc Xuân Tùng	22521619	22521619@gm.uit.edu.vn
3	Đào Xuân Vinh	22521666	22521666@gm.uit.edu.vn

## 2. NỘI DUNG THỰC HIỆN:<sup>1</sup>

STT	Công việc	Kết quả tự đánh giá
1	Kịch bản 1b	100%
2	Kịch bản 3	100%
3	Kịch bản 5	100%
4	Kịch bản 6	100%
5	Kịch bản 7	100%
6	Kịch bản 9	100%
7	Kịch bản 10	100%
8	Writeup CTF Challenge	100%

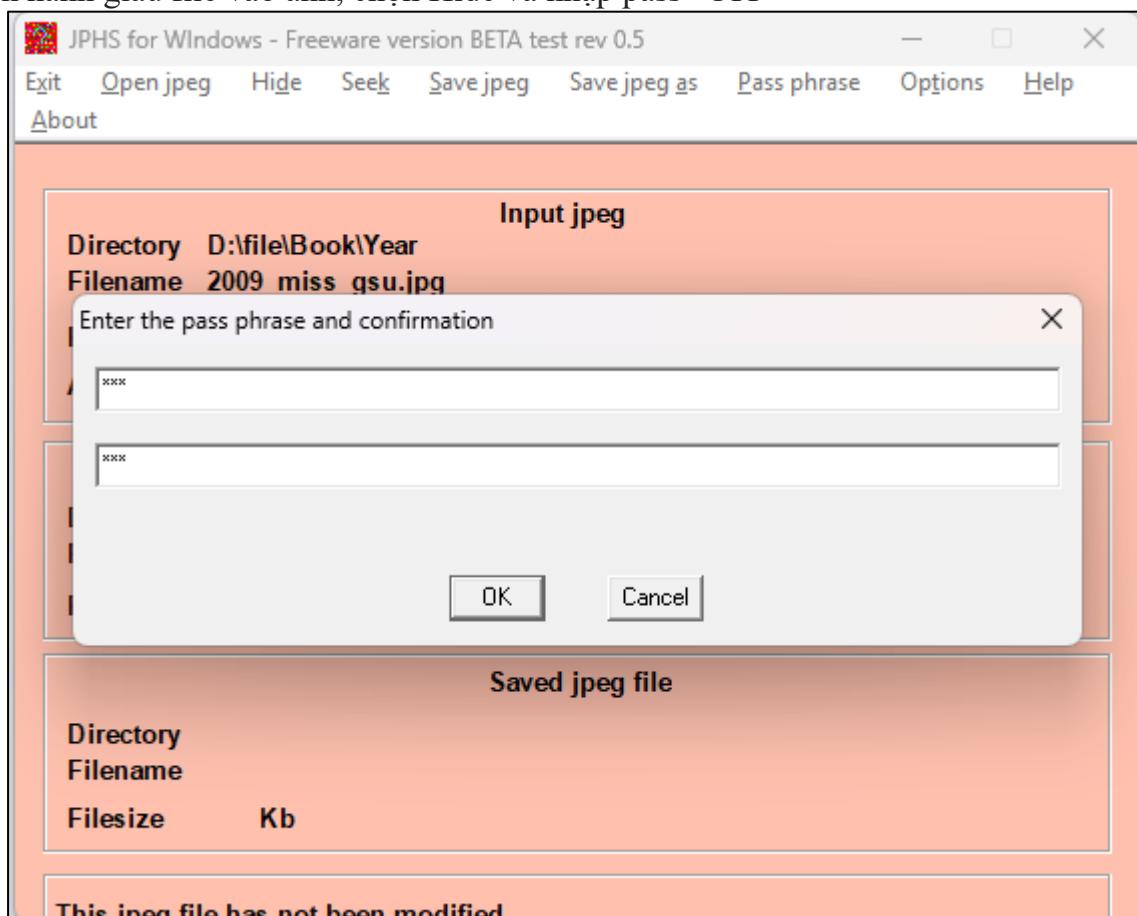
Phần bên dưới của báo cáo này là tài liệu báo cáo chi tiết của nhóm thực hiện.

<sup>1</sup> Ghi nội dung công việc, các kịch bản trong bài Thực hành

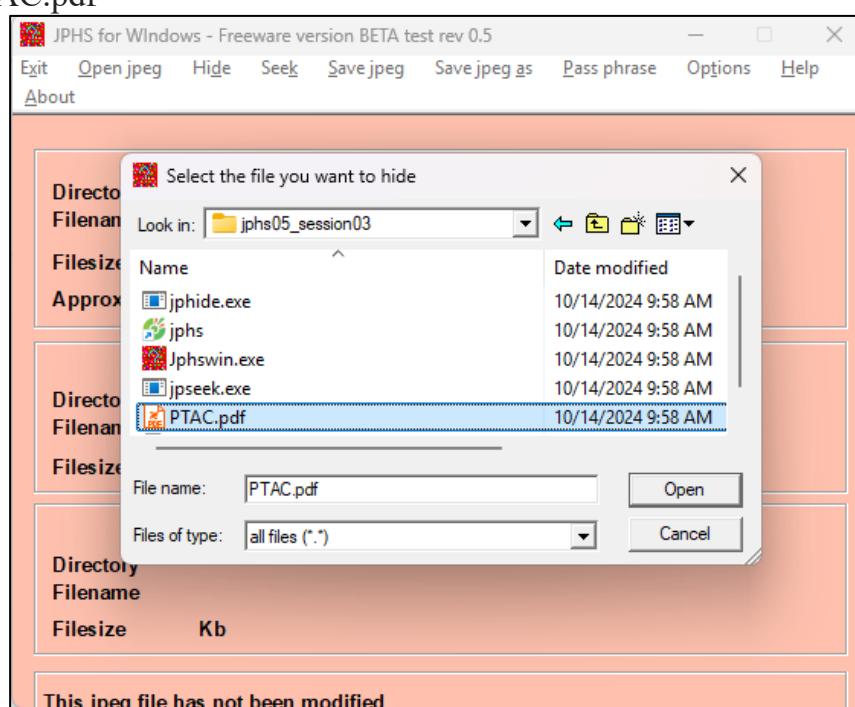
## BÁO CÁO CHI TIẾT

### Kịch bản 01-b: Giấu tin và giải mã thông tin trong ảnh

- Tiến hành giấu file vào ảnh, chọn Hide và nhập pass “UIT”

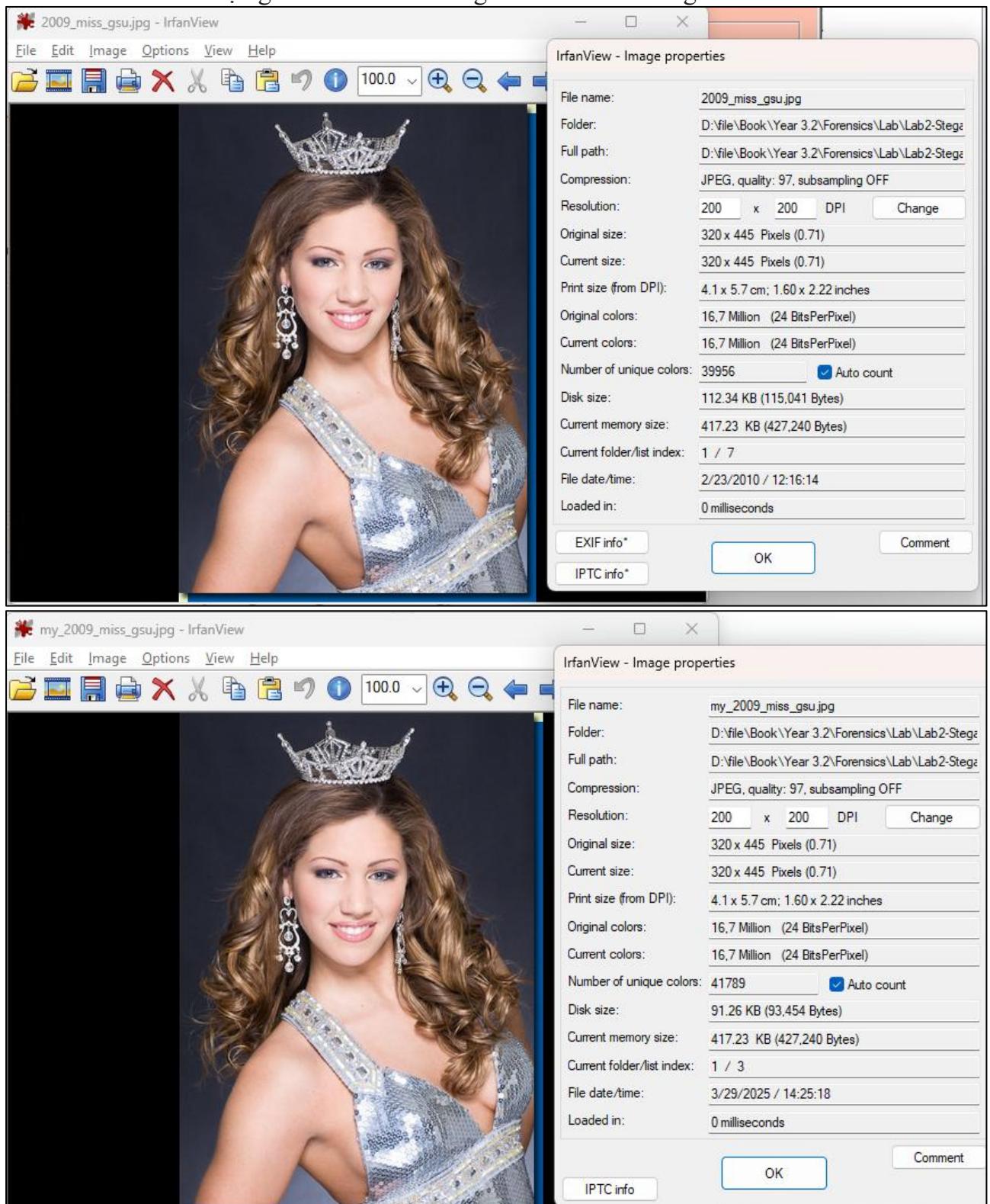


- Chọn file PTAC.pdf



## Lab 2: Steganography & Steganalysis

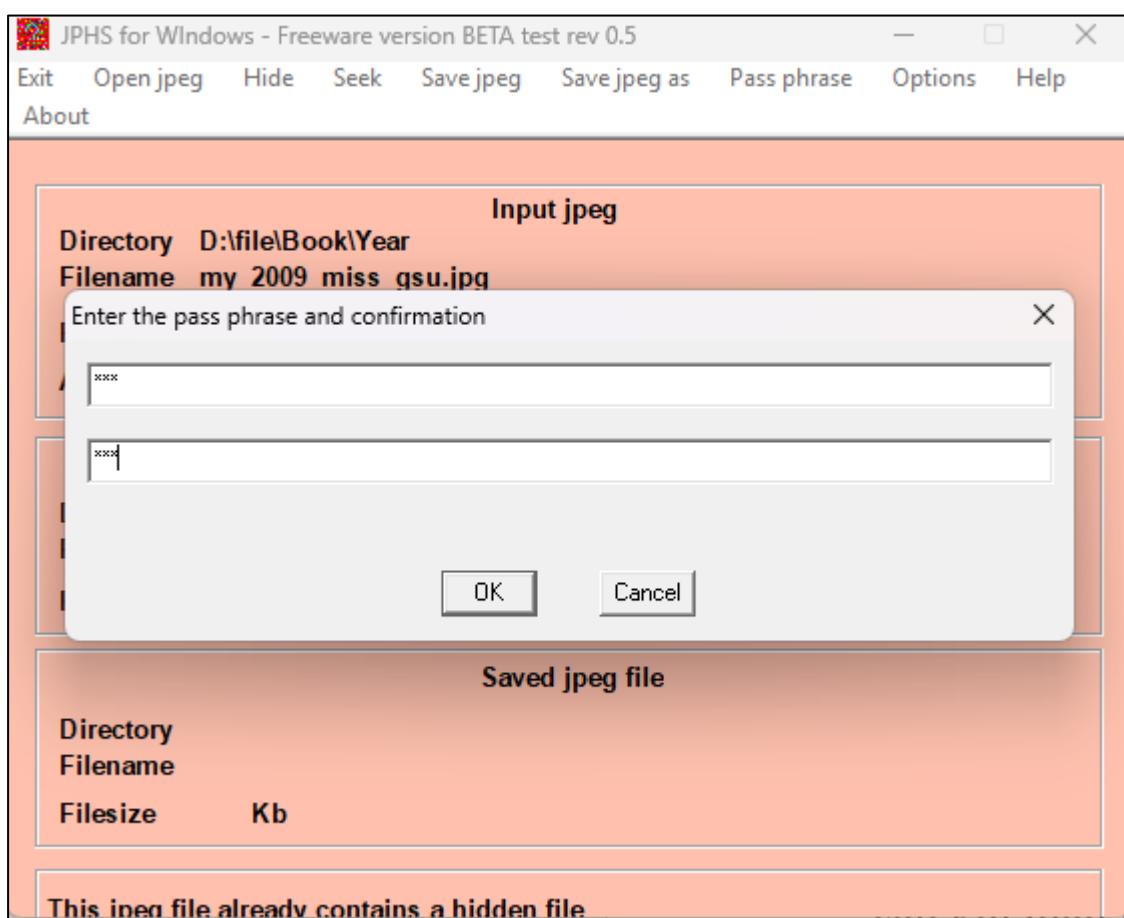
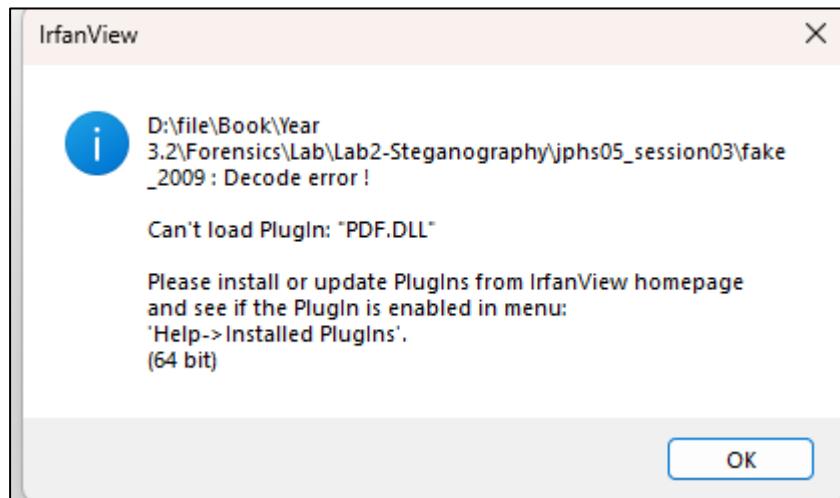
- Mở file ảnh mới được giấu và so sánh thông tin với ảnh cũ bằng IrfanView



- Ta thấy có 2 sự khác biệt nằm ở unique color và Disk size, trong đó unique color của ảnh sau khi giấu nhiều hơn ảnh gốc, điều này chứng tỏ ảnh được giấu có nhiều bit màu hơn vì có thông tin khác nằm trong nó. Còn Disk size của ảnh gốc lại lớn hơn ảnh được giấu, điều này có thể là do nó được lưu lại với mức nén cao hơn để giữ cho độ phân giải và Memory size không thay đổi.

## Lab 2: Steganography & Steganalysis

- Tiến hành lấy thông tin đã giấu bằng cách chọn Seek sau đó nhập pass “UIT” vào, lưu tệp là “fake\_2009” không có đuôi file, sau khi mở bằng IranView thì báo lỗi thiếu plugin PDF nên ta sẽ đổi đuôi file sang pdf



## Lab 2: Steganography & Steganalysis

5

- So sánh file pdf được lấy ra và file pdf gốc, ta thấy không có sự khác biệt nào

The screenshot shows a table of extracted data from the PDF. The columns are: Done, Name, User Name, E-mail, Password, and External Login Info. The data consists of many rows of user information, mostly consisting of 'your current password' and 'pusd11user name'.

Done	Name	User Name	E-mail	Password	External Login Info.
	Ann Rawlings	ARawlings	ARawlings@peoriaud.k12.az.us	your current password	pusd11user name
	Chris Kuczka	CKuczka	CKuczka@peoriaud.k12.az.us	your current password	pusd11user name
	Cindy Callaway	CCallaway	CCallaway@peoriaud.k12.az.us	your current password	pusd11user name
	Dave Carlson	DCarlson	DCarlson@peoriaud.k12.az.us	your current password	pusd11user name
	David Snyder	DSnyder	DSnyder@peoriaud.k12.az.us	your current password	pusd11user name
	Jo Little	JLittle	JITTLE@peoriaud.k12.az.us	your current password	pusd11user name
	Julia Erickson	JErickson	JErickson@peoriaud.k12.az.us	your current password	pusd11user name
	Larry Buchanan	LBuchanan	LBuchanan@peoriaud.k12.az.us	your current password	pusd11user name
	Lissa Cuellar	LCuellar	LCuellar@peoriaud.k12.az.us	your current password	pusd11user name
	Maggie Oney	MOney	MOney@peoriaud.k12.az.us	your current password	pusd11user name
	Nan Hart-DAC	NHart-DAC	NHart-DAC@peoriaud.k12.az.us	your current password	pusd11user name
	Nathan Bowler	NBowler	NBowler@peoriaud.k12.az.us	your current password	pusd11user name
	Patti Beltram	PBeltram	PBeltram@peoriaud.k12.az.us	your current password	pusd11user name
	Phil Valentine	PValentine	PValentine@peoriaud.k12.az.us	your current password	pusd11user name
	Robert Keagle	RKeagle	RKeagle@peoriaud.k12.az.us	your current password	pusd11user name
	Rosemary Martin-Moore	RMMoore	RMMoore@peoriaud.k12.az.us	your current password	pusd11user name
	Samantha Middagh	SMiddagh	SMiddagh@peoriaud.k12.az.us	your current password	pusd11user name
	Sarah Balder	SBalder	SBalder@peoriaud.k12.az.us	your current password	pusd11user name
	Shonna Mandia	SMandia	SMandia@peoriaud.k12.az.us	your current password	pusd11user name
	Steve Savoy	SSavoy	SSavoy@peoriaud.k12.az.us	your current password	pusd11user name
	Teri Nevezar	TNevezar	TNevezar@peoriaud.k12.az.us	your current password	pusd11user name
	Terrie Rust	TRust	TRust@peoriaud.k12.az.us	your current password	pusd11user name
	Valerie Naish	VNaish	VNaish@peoriaud.k12.az.us	your current password	pusd11user name
	Bill Copeland	BCopeland	GSMOM@COX.NET	BCopeland@COX.NET	BCopeland
	Tammyra Edgin	Tammyra.Edgin	tammara@microsoft.com	TeCom23	pusd11textTammyra.Edgin
	Diane Douglas	Diane.Douglas	dmddouglas@cox.net	DdNet21	pusd11textDiane.Douglas
	Mary Crespino	Mary.Crespino	crespy@cox.net		

The screenshot shows the properties dialog box for 'fake\_2009.pdf'. The tabs are General, Digital Signatures, Security, Details, and Previous Versions. The General tab is selected.

**General**

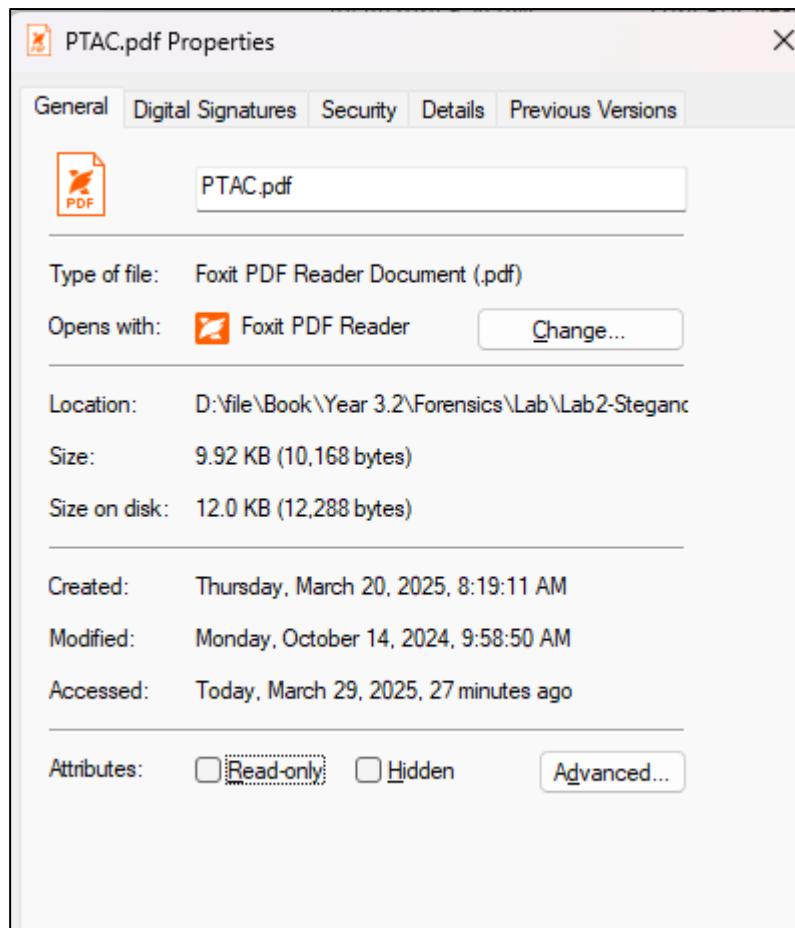
Type of file: Foxit PDF Reader Document (.pdf)  
Opens with: Foxit PDF Reader

Location: D:\file\Book\Year 3.2\Forensics\Lab\Lab2-Steganc  
Size: 9.92 KB (10,168 bytes)  
Size on disk: 12.0 KB (12,288 bytes)

Created: Saturday, March 29, 2025, 2:49:12 PM  
Modified: Saturday, March 29, 2025, 2:49:12 PM  
Accessed: Today, March 29, 2025, 2:50:04 PM

Attributes:  Read-only  Hidden

## - Lab 2: Steganography & Steganalysis



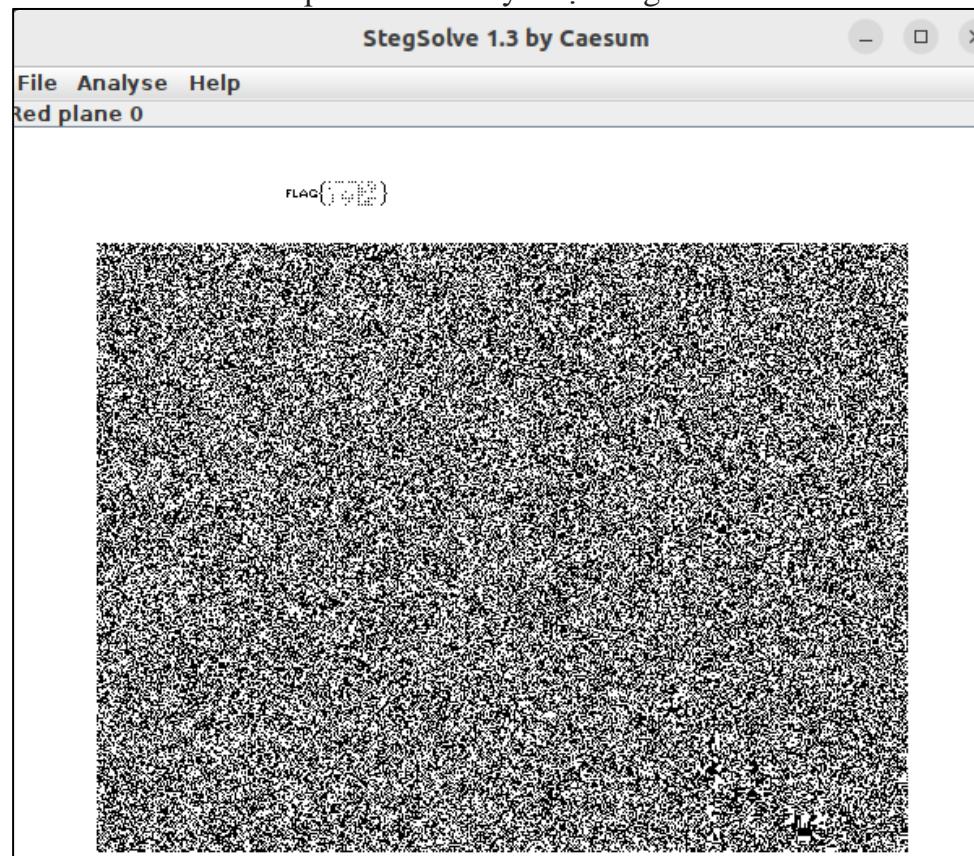
### Kịch bản 03. Điều tra thông tin được ẩn giấu

- Tài nguyên: kb03-suspicion.png
  - Mô tả: Bức ảnh scan này đã được phục hồi từ các tập tin của một cựu nhân viên của Hiệp hội Ngó ngắn Miêu Quốc. Nhân viên điều tra cần phải tìm ra số sê-ri của máy in này để có thể xác định vị trí của thiết bị. Tìm số sê-ri của máy in

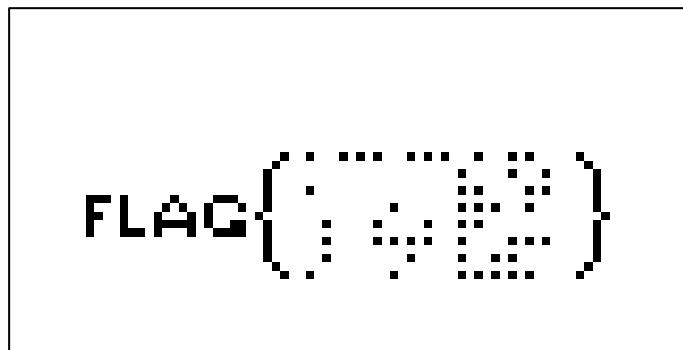
- Dùng strings xem file

## Lab 2: Steganography & Steganalysis

- Mở stegsolve và bấm đến Red plane 0 thì thấy được flag



- Phóng ra cho dễ nhìn

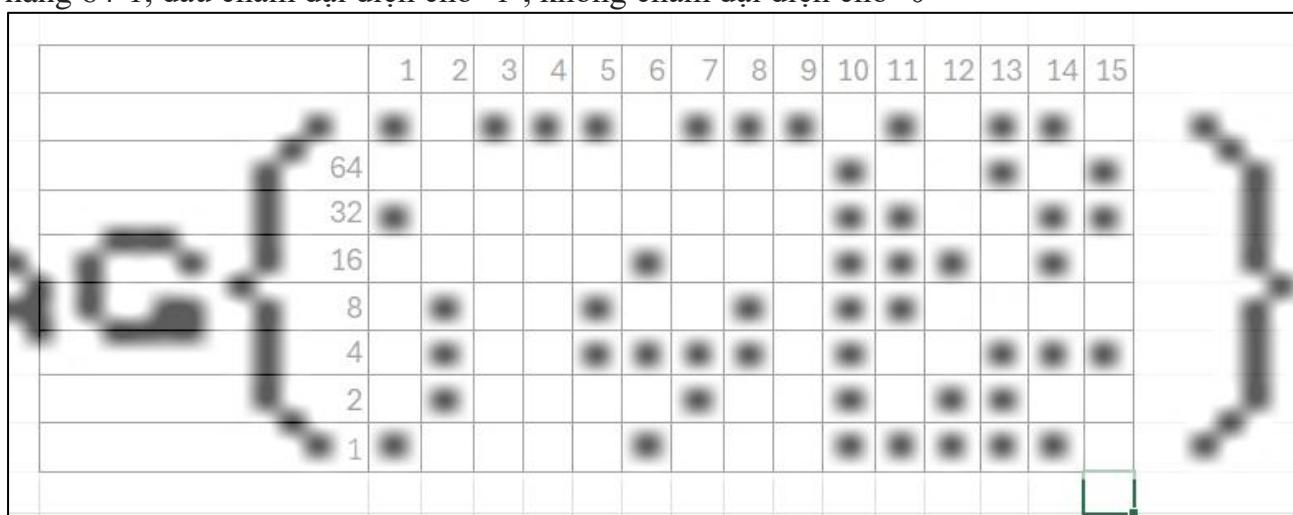


- Dựa vào phân vùng dữ liệu bên dưới ta thấy số serial cần tìm nằm trong vùng: cột 11-14 hàng 64-1

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
64																Unused
32																Minute
16																Hour
8																Day
4																Month
2																Year
1																Serial



- Đã nhập xong, cách đọc số serial sẽ là đọc thông tin từ phải qua trái ở các cột 14-11, từ hàng 64-1, dấu chấm đại diện cho ‘1’, không chấm đại diện cho ‘0’



- Từ đó ta có được mã binary tương ứng từ phải qua như sau:

0110101 1000111 0010011 0111001

53        71        19        57

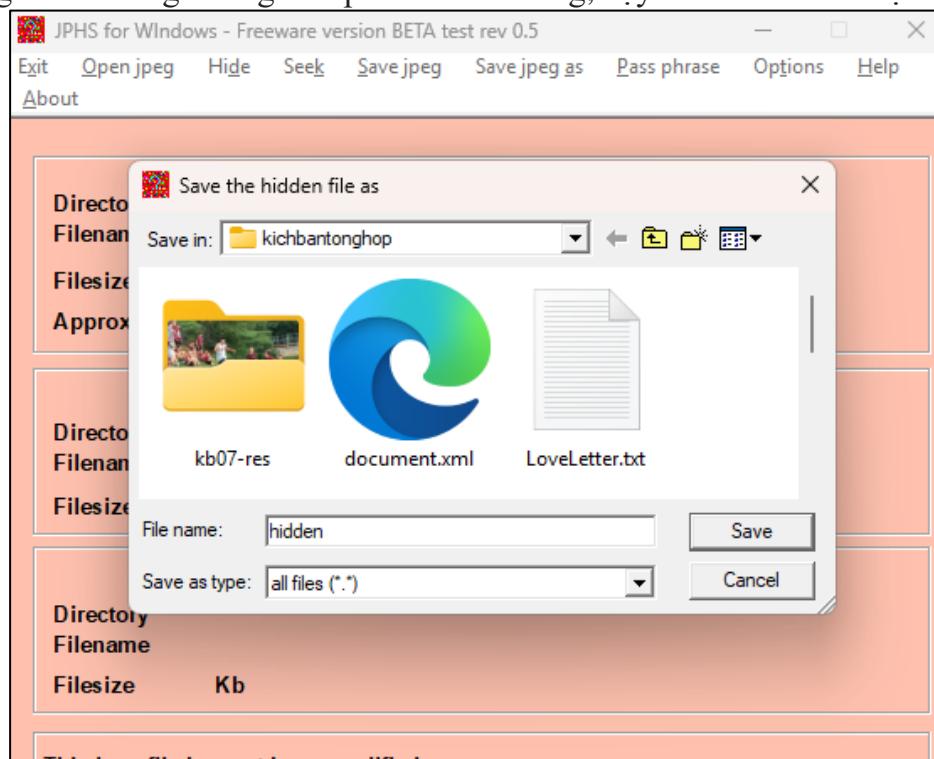
- Vậy số serial là: 197153 hoặc 57197153

#### Kịch bản 05. Thực hiện phân tích, tìm thông tin ẩn giấu trong ảnh

- Tài nguyên thực hiện: qn001.jpg

- Yêu cầu – Gợi ý: Tìm thông điệp (flag) được ẩn giấu. Thông tin flag liên quan đến **Đội tuyển bóng đá nam Việt Nam**.

- Dùng JPHS để kiểm tra file, ta thấy file có kích thước khá lớn, chắc chắn có giấu gì trong đó, thử dùng seek nhưng không biết pass nên để trống, vậy mà nó vẫn ra được file



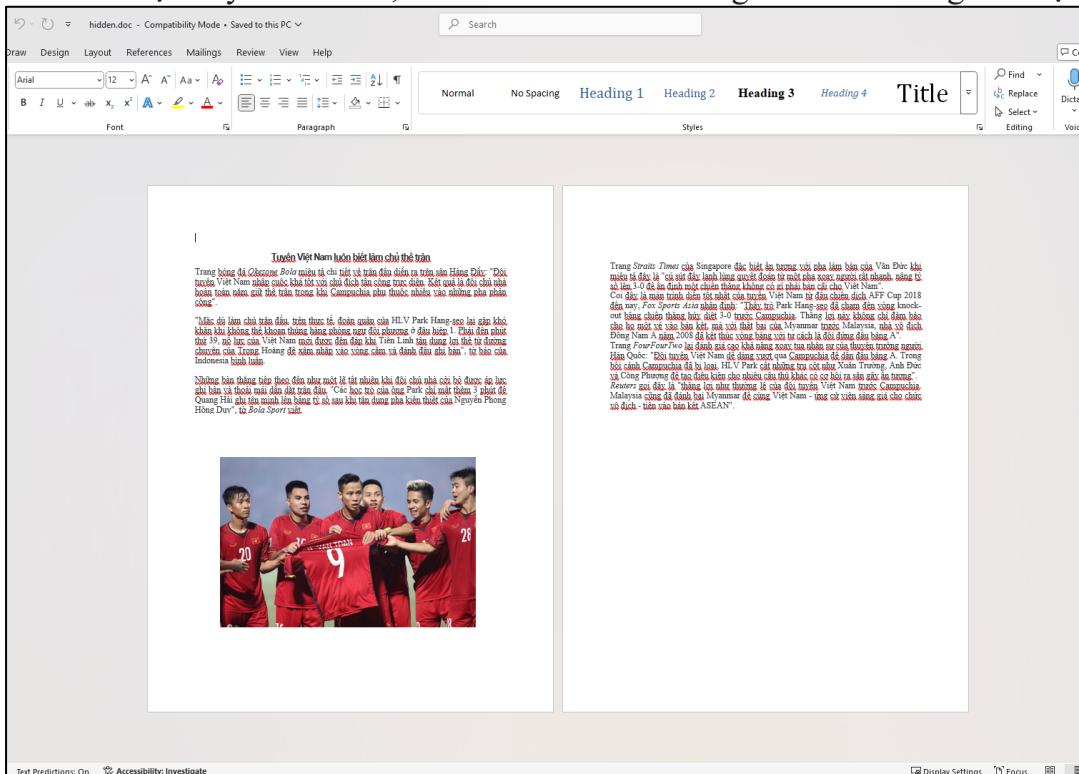
## Lab 2: Steganography & Steganalysis

6

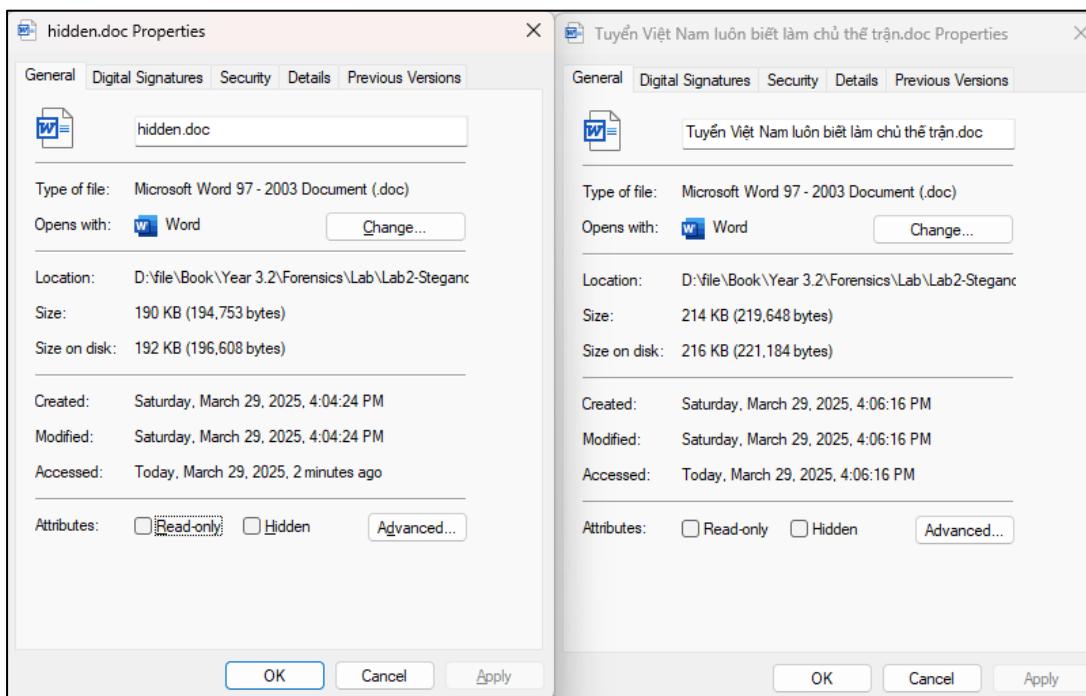
- Lưu file là hidden, ta cop file hidden qua Ubuntu để kiểm tra loại file bằng lệnh `file hidden`

```
emay@emay:~/Downloads$ file hidden
hidden: Microsoft Word 2007+
emay@emay:~/Downloads$ file hidden
```

- Giờ đã biết được đây là file doc, đổi đuôi file và mở bằng word xem có gì thú vị không



- Một bài viết về trận thắng của đội tuyển Việt Nam trước Campuchia năm 2018, không có thông tin gì đáng lưu ý, ta thử cop hết nội dung bỏ vào một file khác để kiểm tra kích thước file

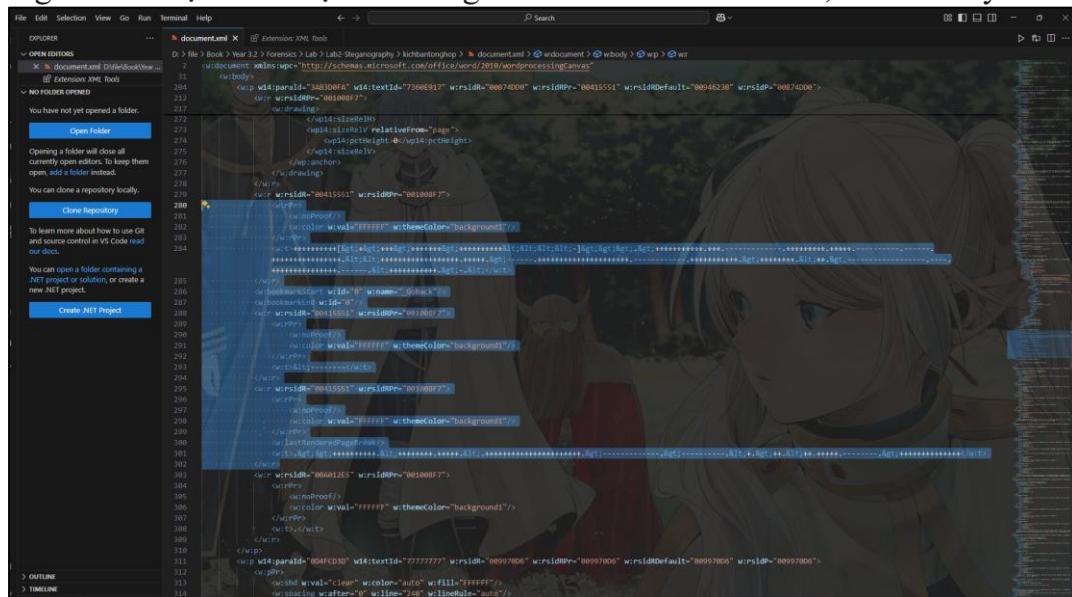


## - Lab 2: Steganography & Steganalysis

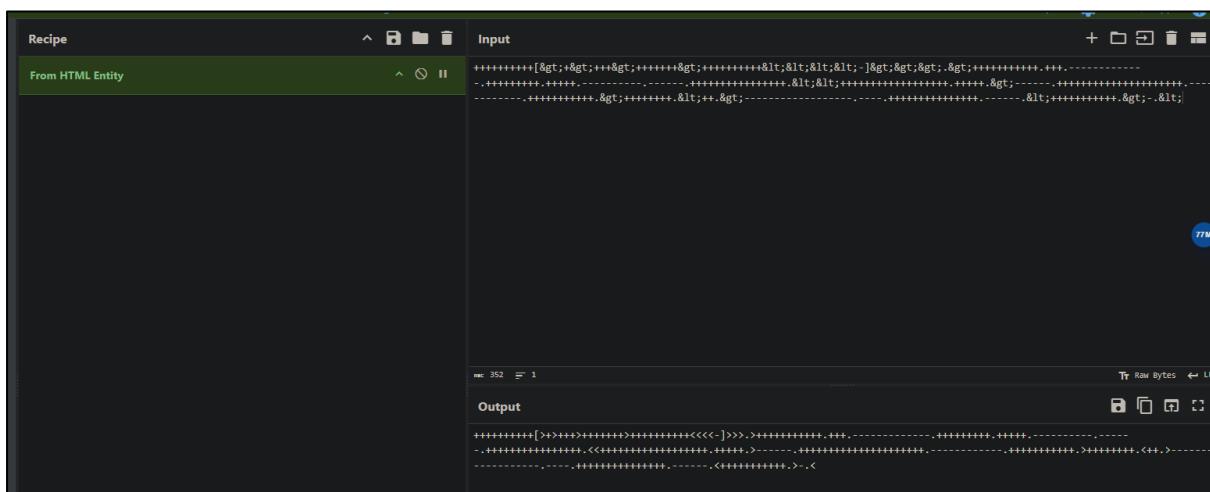
- Tuy cop hết nội dung được hiển thị nhưng file mới có vẻ lớn hơn file gốc, vậy có thể file gốc đã được nén ở mức cao hơn với một nội dung ẩn nào đó
  - Ta biết file .doc thực ra cũng là một file được nén nên ta sẽ giải nén nó ra những file xml để xem.

```
emay@emay:~/Downloads$ unzip hidden.doc
Archive: hidden.doc
  inflating: [Content_Types].xml
  inflating: _rels/.rels
  inflating: word/document.xml
  inflating: word/_rels/document.xml.rels
extracting: word/media/image1.jpg
  inflating: word/theme/theme1.xml
  inflating: word/settings.xml
  inflating: word/styles.xml
  inflating: word/webSettings.xml
  inflating: word/fontTable.xml
  inflating: docProps/core.xml
  inflating: docProps/app.xml
```

- Nội dung chính được hiển thị nằm trong file word/document.xml, mở file này ra để xem



- Giữa những nội dung thực tế được hiển thị khi mở bằng word thì ta bắt gặp một đoạn ký tự la, pass lên cyberchef thì decode được từ HTML Entity



## Lab 2: Steganography & Steganalysis

- Nhìn mật mã có vẻ là search thử vài tool thì tìm được [tool này](#) giúp identify cyber, pass thử một đoạn thì trả về kết quả Brainfuck (?)

- Tiện là trên trang này cũng có decode Brainfuck nên ta bỏ nguyên đoạn ký tự kia vào thì ra được flag

**FLAG: Forensics05@UIT{Vietnam-win-Cambodia}**

### 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 ký tự trong chuỗi ban đầu thành chuỗi chỉ gồm 2 ký tự a và b.

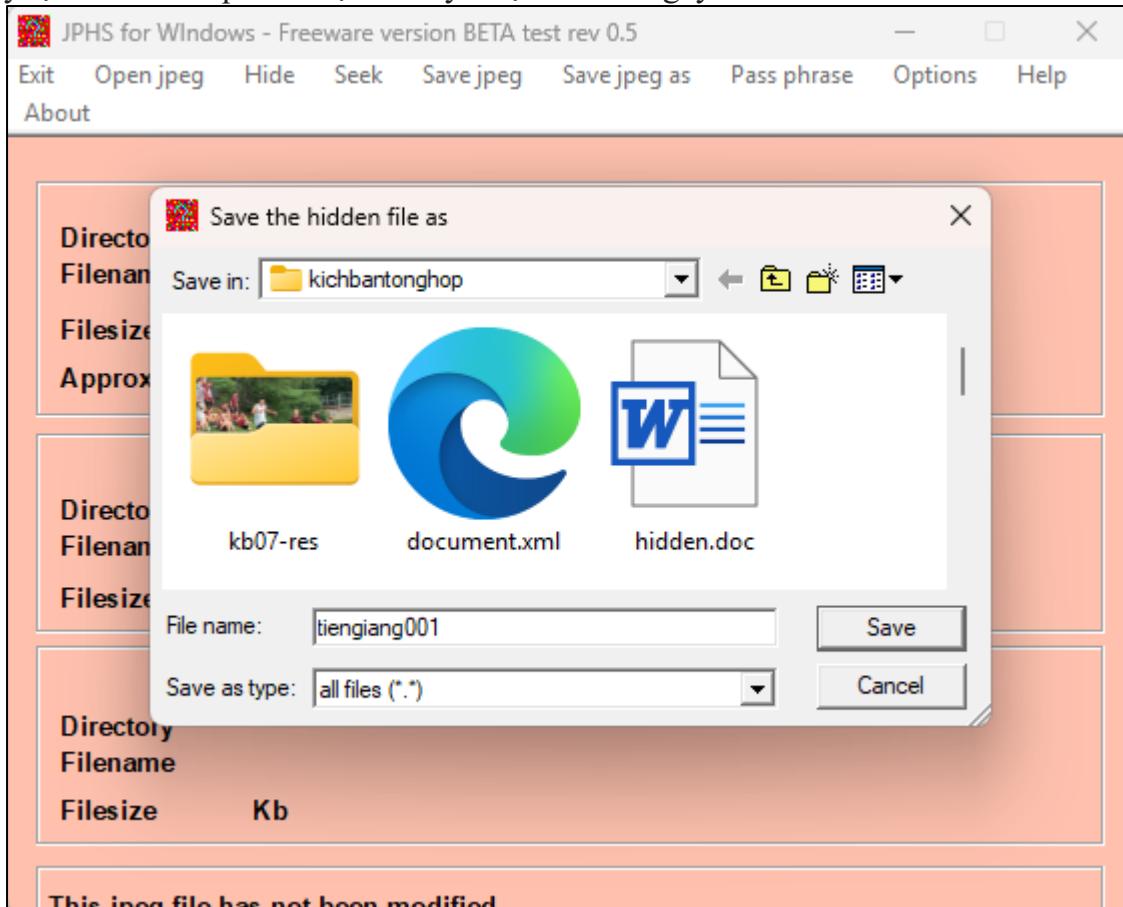
- Thủ dùng vài tool như stegdetect hay stegbreak với wordlist cho sẵn trong tài nguyên (chủ yếu vọc tool) thì không ra gì lầm (P/s: giải mã không cần pass nên pass trong wordlist cho sẵn sai hết, đổi sang wordlist “rockyou.txt” thì vẫn ra)

```
D:\file\Book\Year 3.2\Lab\Lab2-Steganography\stegdetect04_session03>stegdetect.exe -t jopi tiengiang003.jpg
tiengiang003.jpg : negative
```

## Lab 2: Steganography & Steganalysis

```
D:\file\Book\Year 3.2\Forensics\Lab\Lab2-Steganography\stegdetect04_session03>stegbreak.exe -r rules.ini -f usr/share/dict/words.txt tiengiang003.jpg
Loaded 1 files...
tiengiang003.jpg : negative
Processed 1 files, found 0 embeddings.
Time: 0 seconds: Cracks: 392, Inf c/s
```

- Quay lại với JPHS quen thuộc thì lấy được file ẩn ngay



- Xem loại file, biết được đây là file png

```
emay@emay:~/Desktop/lab2/kichbantonghop$ file tiengiang001
tiengiang001: PNG image data, 385 x 131, 8-bit colormap, non-interlaced
```

- Dùng strings xem có gì thú vị không

```
tiengiang001: PNG image data, 385 x 131, 8-bit colormap, non-interlaced
emay@emay:~/Desktop/lab2/kichbantonghop$ strings tiengiang001 > strings_tiangian
g001.txt
emay@emay:~/Desktop/lab2/kichbantonghop$ open strings_tiangiang001.txt
```

- Hiếm hoi lắm mới có bài mà ra strings ra được file bé thế này, luốt xuống thì thấy được gì đó

```
58 uyDY$  
59 Gm#Lm  
60 l)7n  
61 ^D0  
62 ^NG L  
63 P{)*#  
64 2-Jv  
65 OCQ0Z  
66 - gL_  
67 PLp  
68 u) Ic  
69 JgJU  
70 JU-  
71 IEND  
72 where ShOUld onF Really DoK For tHIS fLe
```

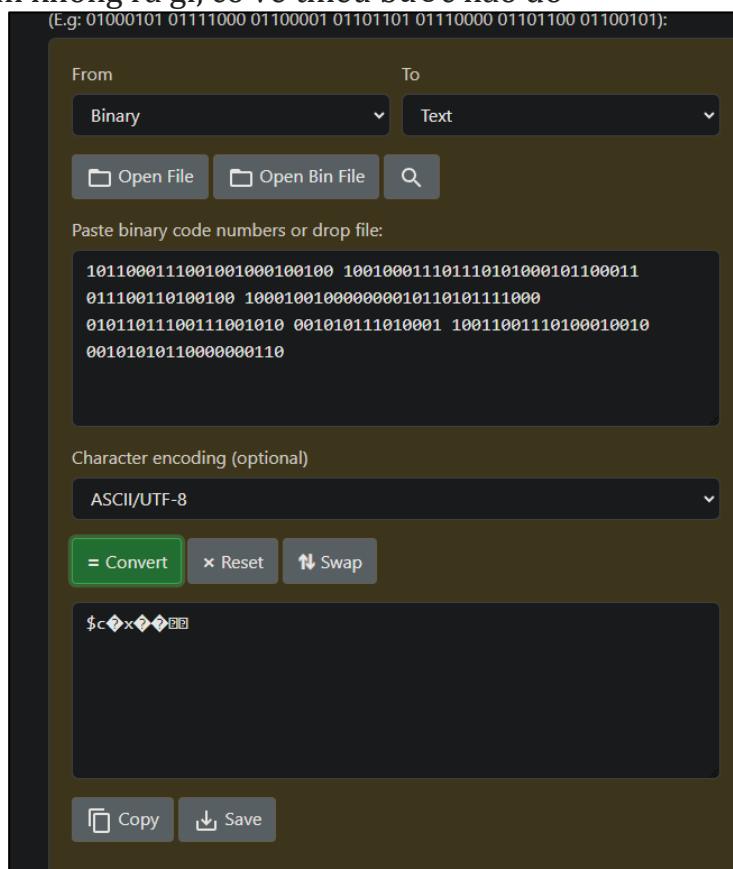
Plain Text ▾ Tab Width: 8 ▾ Ln 72, Col 1 ▾ INS

## - Lab 2: Steganography & Steganalysis

- Theo hint đề bài thì thuật toán mã hóa có liên quan gì đó đến a và b, hỏi anh ruột ChatGPT thì được giới thiệu đến mã hóa bacon, xem trên [geeksforgeeks](#) thì đây là mã hóa chuyển các chữ cái thành các chữ a và b theo từng nhóm 5 ký tự, có sẵn code nên thử xem

```
emay@emay:~/Desktop/lab2/kichbantonghop$ python3 decode_bacon.py  
1011000111001001000100100 100100011101110101000101100011 011100110100100 1000100  
10000000010110101111000 01011011100111001010 001010111010001 1001100111010001001  
0 0010101011000000110  
emay@emay:~/Desktop/lab2/kichbantonghop$ rm -f authora-decode_bacon.py
```

- Convert binary thì không ra gì, có vẻ thiếu bước nào đó



- Sau một lúc đọc kỹ thì hóa ra từ chuỗi mã hóa sẽ có 2 cách viết, tùy theo quy định mà 1 cách sẽ là a còn 1 cách sẽ là b, sau khi chuyển đổi chuỗi mã hóa sang chuỗi chỉ gồm kí tự a và b thì chia thành từng nhóm 5 kí tự và bắt đầu giải mã.

The writer must make use of two different [typefaces](#) for this cipher. After preparing a false message with the same number of letters as all of the *As* and *Bs* in the real, secret message, two typefaces are chosen, one to represent *As* and the other *Bs*. Then each letter of the false message must be presented in the appropriate typeface, according to whether it stands for an *A* or a *B*.<sup>[4]</sup>

- Ta có chuỗi gốc là: “wherE ShOUld onE ReAllY lOoK fOr tHis flag” bao gồm chữ hoa và chữ thường. quy đổi chữ thường = a/chữ hoa = b.

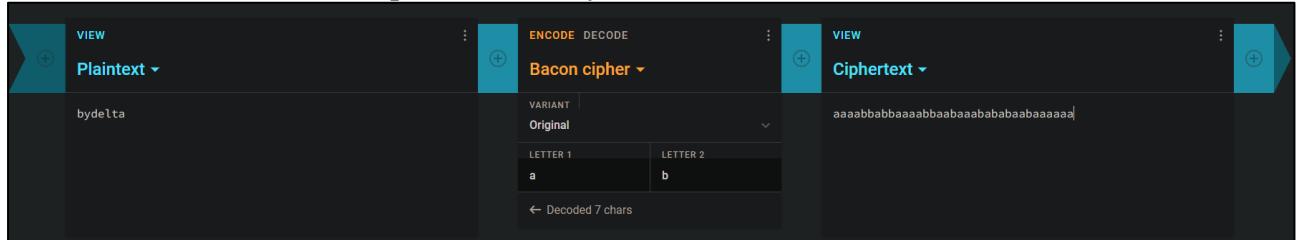
- Viết script nhỏ để đổi sang chuỗi a/b (đau mắt quá không convert bằng cơm được)

```
1 def case_to_ab(text):
2     return ''.join('b' if c.isupper() else 'a' for c in text if c.isalpha())
3
4 input_text = "whErE ShOuLd oNLY ReAlLy lOOk fOr tHiS flAg"
5 ab_output = case_to_ab(input_text)
6 print(f"Input: {input_text}")
7 print(f"a/b: {ab_output}")
```

## Lab 2: Steganography & Steganalysis

```
emay@emay:~/Desktop/lab2/kichbantonghop$ python3 decode_bacon.py
Input: wherE SHOULd oN E ReAlly looK fOr tHiS flag
a/b: aaaabbabaaaabbabaababababaaaaaa
```

- Decode chuỗi a/b ra được plaintext là “bydelta”

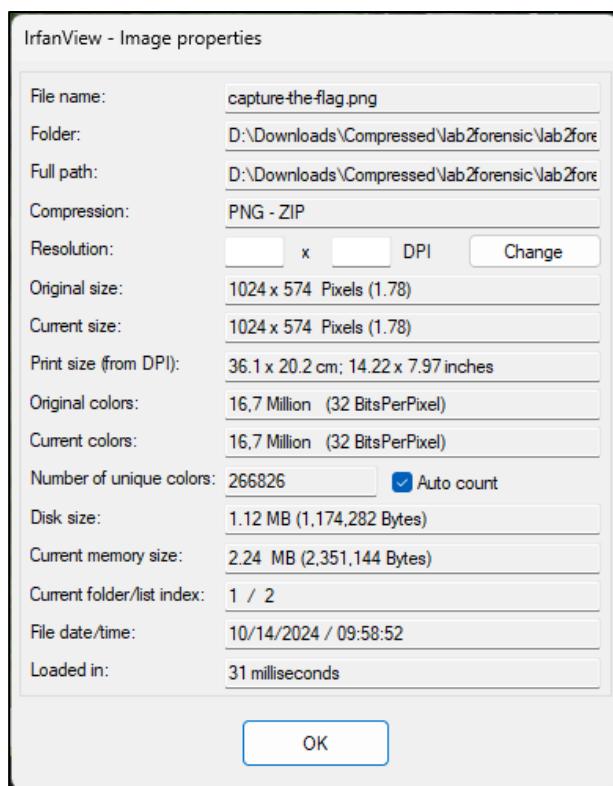


- “bydelta” có nghĩa nên chắc flag là cái này.

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

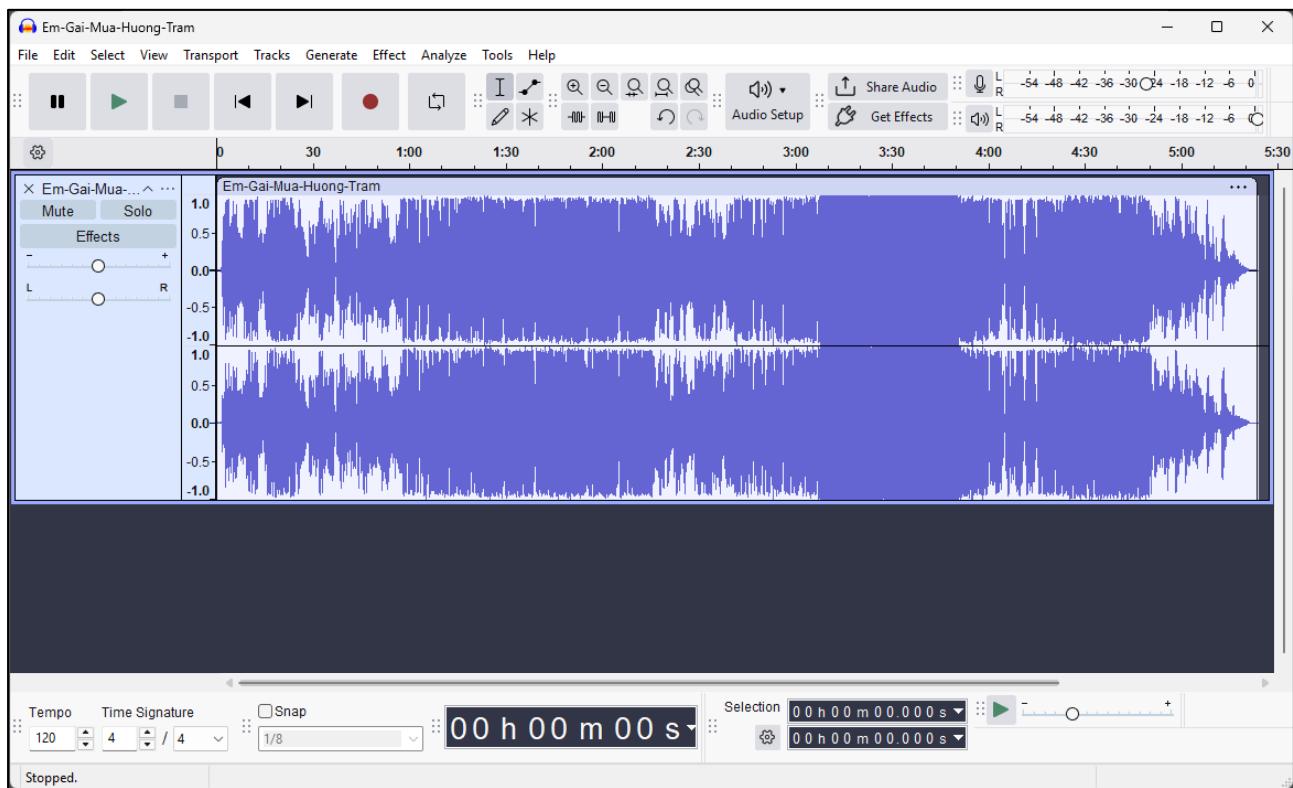
- Tài nguyên: Em-Gai-Mua-Huong-Tram.mp3, capture-the-flag.png

Kiểm tra thông tin của file png thấy không có gì đáng ngờ

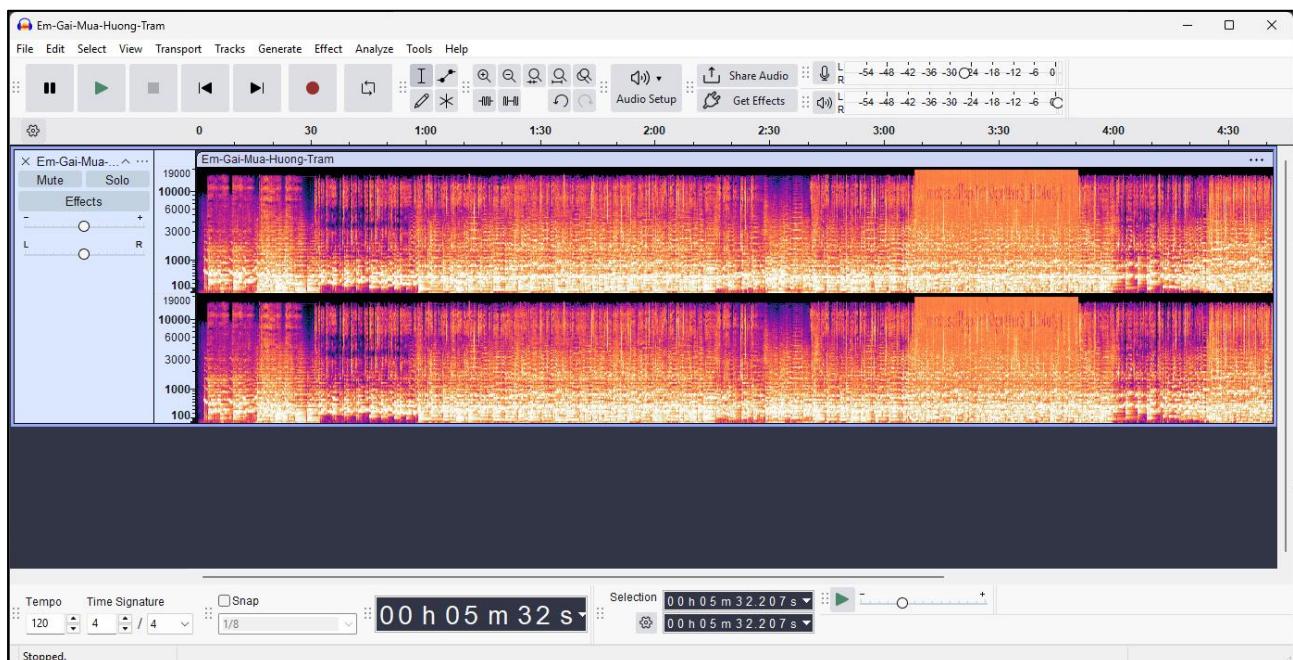


## Lab 2: Steganography & Steganalysis

Mở file mp3 bằng phần mềm audacity

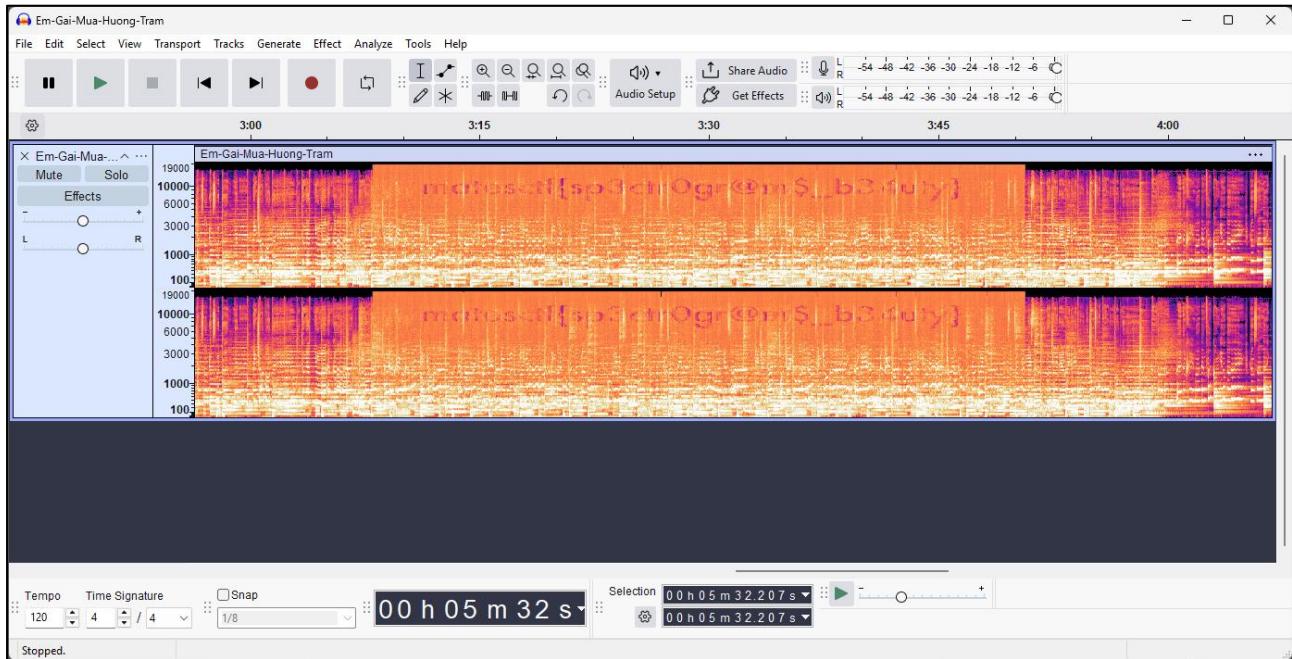


Chuyển sang dạng spectrogram



## Lab 2: Steganography & Steganalysis

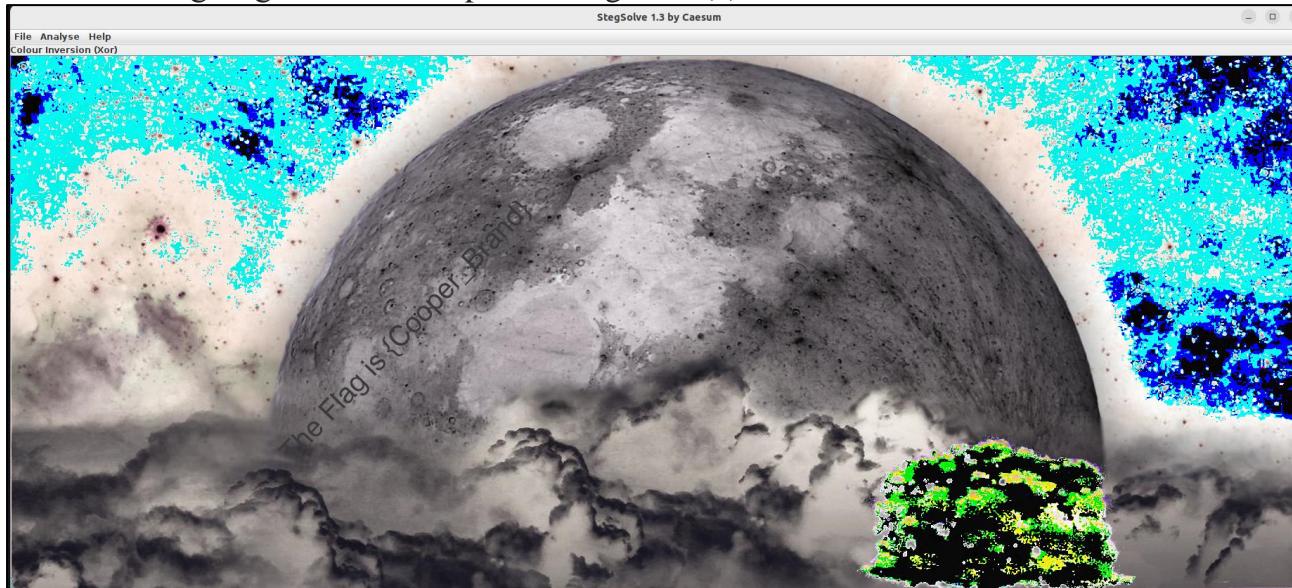
Phân tích phần âm thanh có dấu hiệu bị chỉnh sửa tìm được flag:  
**matesctf{sp3ctr0gr@m\$\_b34uty}**



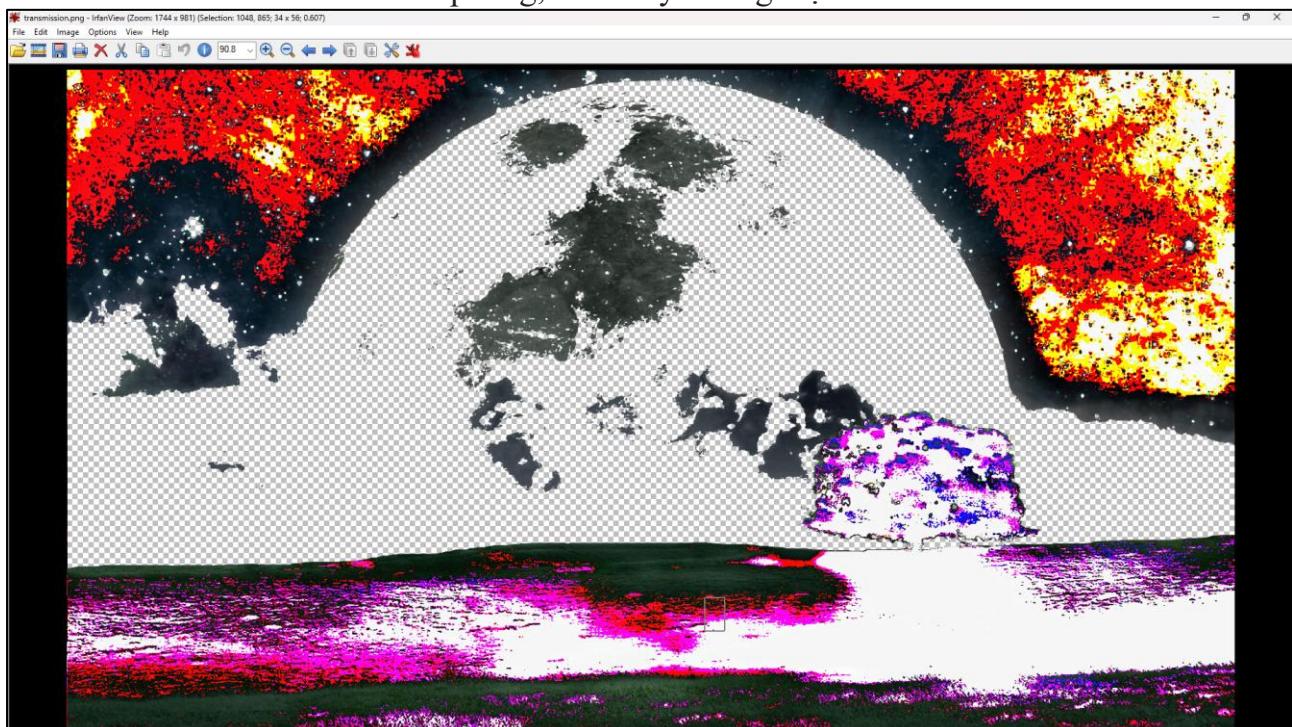
### Kịch bản 09. Thực hiện phân tích, tìm thông tin ẩn giấu:

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

- Mở ảnh bắng stegsolve, click 1 phát ra flag luôn (?)



- Hơi nghi ngờ nên em xem bằng irfanview xem sao, thấy ở khúc gắn flag là một vùng xóa phông, chắc đây là flag thật.

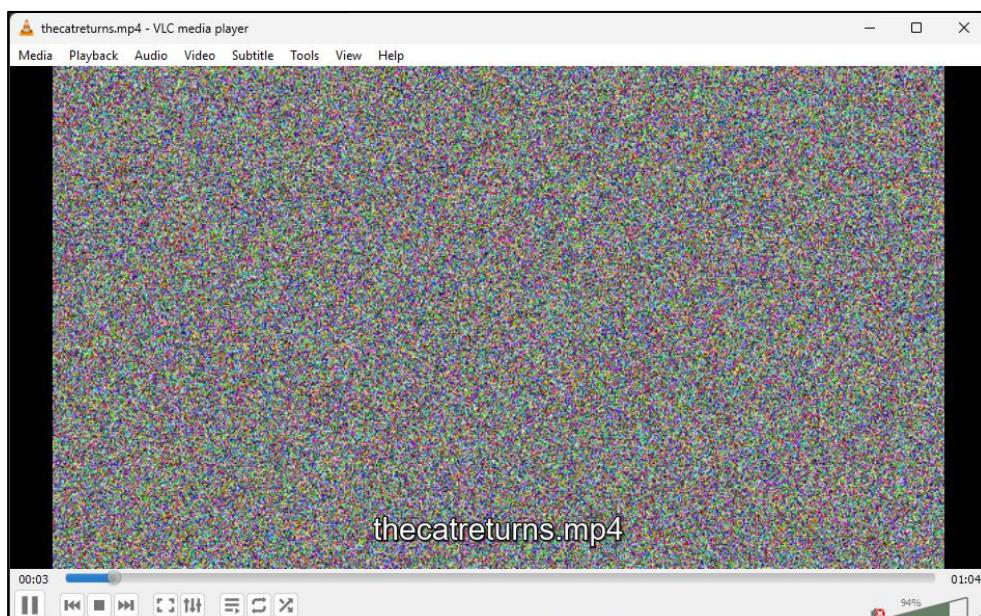


**FLAG: {Cooper\_Brand}**

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

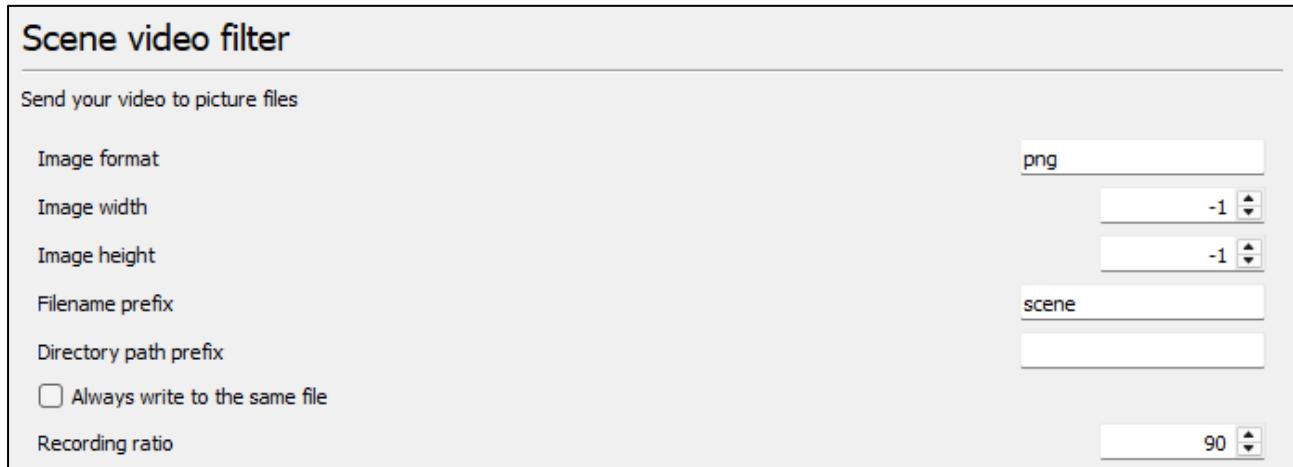
- Tài nguyên: thecatreturns.mp4
- Yêu cầu – Gợi ý: Tìm sự khác biệt giữa các khung hình (frame) trong đoạn phim đã cho. Chuyển nội dung đoạn phim thành các khung hình để phân tích. Công cụ ffmpeg, ImageJ.

Vì không dùng được công cụ để xuất nên em dùng VLC để phân tích frame

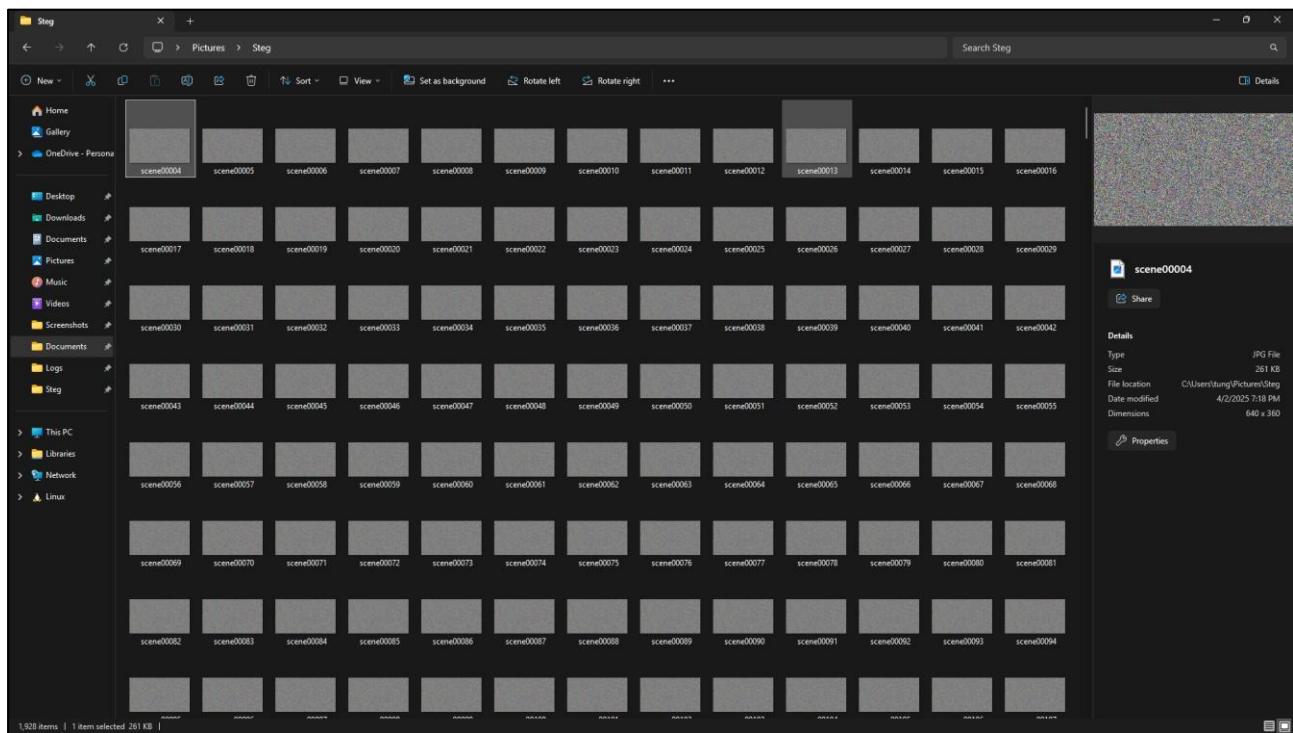


## Lab 2: Steganography & Steganalysis

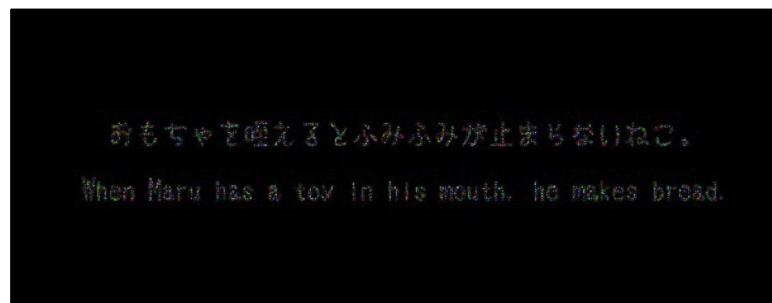
Setting VLC để lấy frame



Tiến hành chụp lại các frame

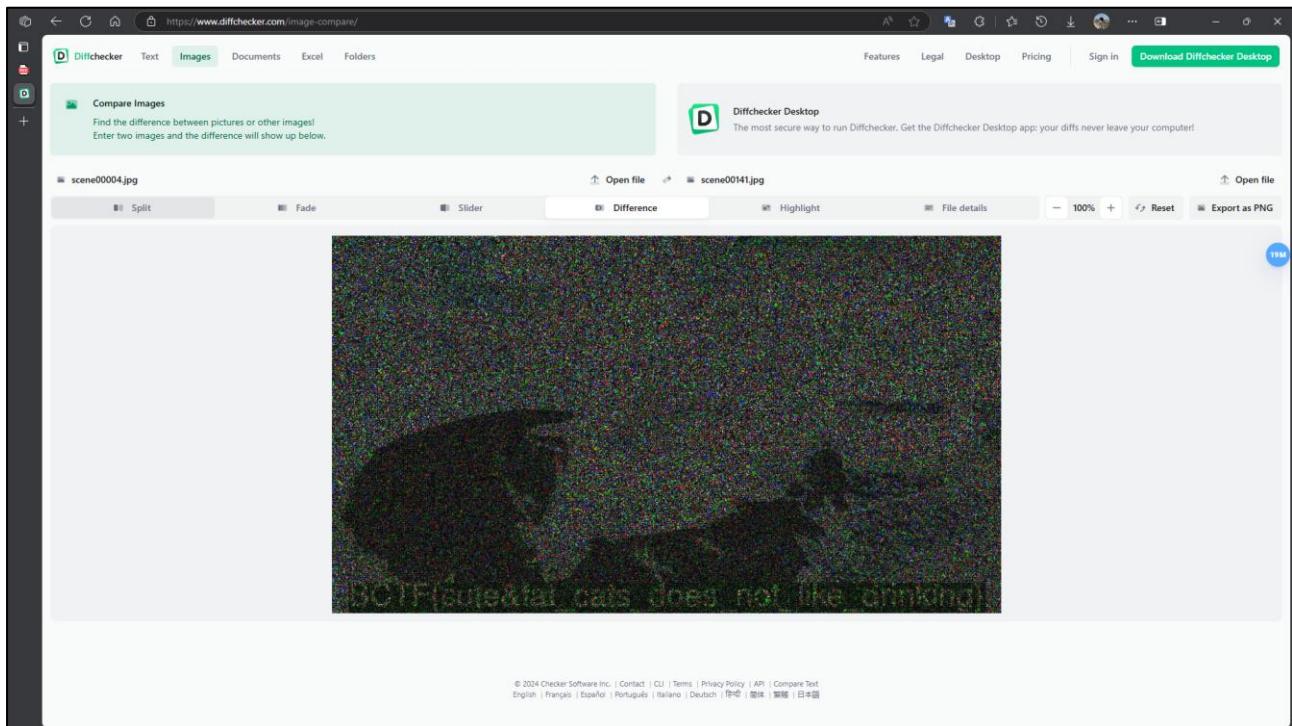


Dùng diffchecker để tìm difference giữa các frame, tìm được 1 thông điệp trong video



Tiếp tục check tìm được flag ở frame 141

BCTF{cute&fat\_cats\_does\_not\_like\_drinking}



### CTF Challenge

#### 1. St3g0

St3g0

Medium Forensics picoCTF 2022 steganography

AUTHOR: LT 'SYREAL' JONES (FT. DJROBIN17)

Description

Download this image and find the flag.

- Download image

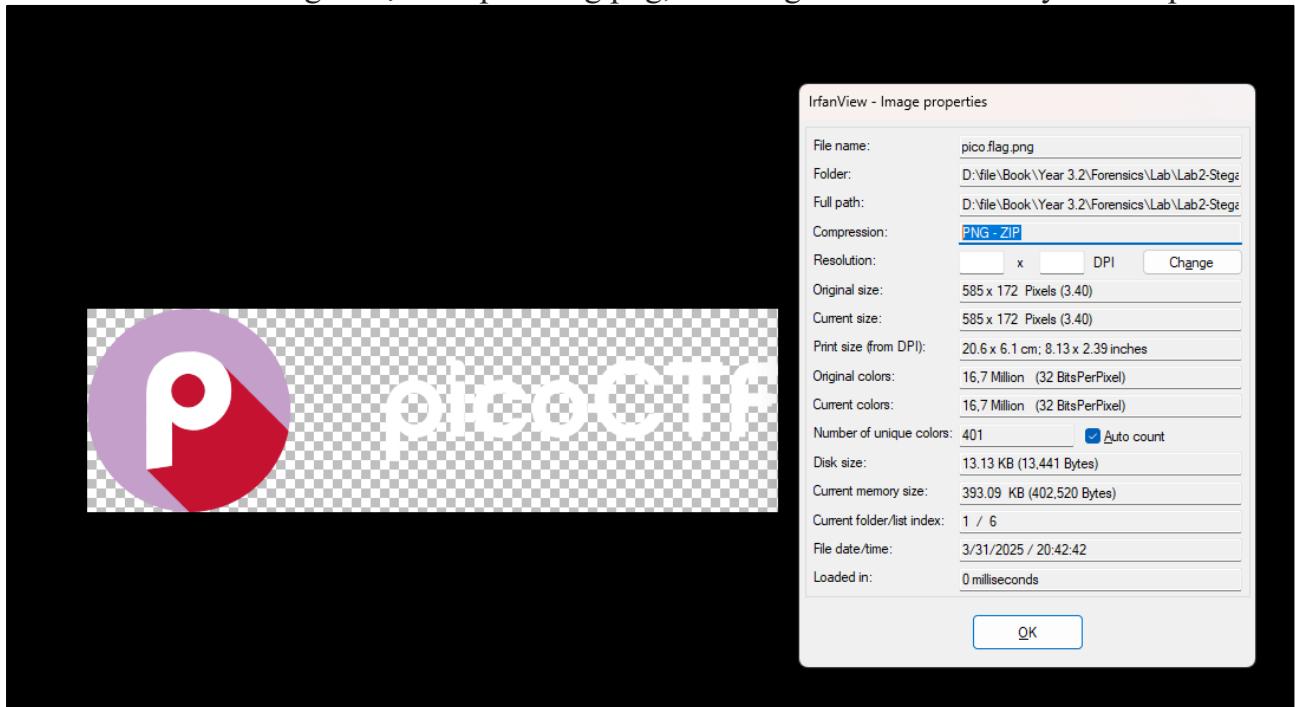
debug info: [u:399501 e: p: c:305 i:294879]

10.718 users solved

Hints ? 1

Submit Flag

- Download image được file pico.flag.png, mở bằng Irfanview thì thấy có compression



- Thử dùng vài tool như unzip hay 7z thì lỗi, dùng thử binwalk thì biết nó là compression zlib

```
emay@emay:~/Downloads$ binwalk pico.flag.png
```

DECIMAL	HEXADECIMAL	DESCRIPTION
0	0x0	PNG image, 585 x 172, 8-bit/color RGBA, non-interlaced
41	0x29	Zlib compressed data, default compression

- Sau một lúc tra cứu thì thấy loại compression này là bình thường đối với file PNG, thử extract nó ra bằng binwalk thì cũng không thấy gì đặc biệt
- Tra thêm một vài tool trên [Stego Tricks](#) thì thấy giới thiệu zsteg

### **zsteg for PNG and BMP Files**

zsteg specializes in uncovering hidden data in PNG and BMP files. Installation is done via `gem install zsteg`, with its [source on GitHub](#).

- Install và dùng thử thì cũng hiệu quả thật

```
emay@emay:~/Downloads$ zsteg pico.flag.png
[picoCTF{7h3r3_15_n0_5p00n_a1062667}]$t3g0"
[E2A5q4EwUSA"]
[APAAQTAAG]
[HUUUUUUU"]

file: Matlab v4 mat-file (little endian) >\004<\305P, numeric, rows 0, columns 0
file: Matlab v4 mat-file (little endian) | <\243, numeric, rows 0, columns 0
file: gfxboot compiled html help file
file: Targa image data (16-273) 65536 x 4097 x 1 +4352 +4369 - 1-bit alpha - right "\021\020\001\001\021\021\001\021\021\001"
file: 0420 Alliant virtual executable not stripped
file: Targa image data - Map 272 x 17 x 16 +257 +272 - 1-bit alpha "\020\001\021\001\021\020\001\020\001\020\001"
file: Targa image data - Map 273 x 272 x 16 +1 +4113 - 1-bit alpha "\020\001\001\001"
file: Novell LANalyzer capture file
file: Applesoft BASIC program data, first line number 8
file: Novell LANalyzer capture file
```

- Giải thích: zsteg phát hiện và trích xuất dữ liệu ẩn trong file PNG bằng kỹ thuật LSB. Khi chạy lệnh trên nó xuất ra nhiều dòng kiểm tra các kênh màu khác nhau với các bit khác nhau. Ta thấy flag ở dòng đầu tiên là do nó xuất theo thứ tự: b1,rgb,lsb,xy

Trong đó:

- + b1: Phân tích bit thứ 1 (bit ít quan trọng nhất).
- + rgb: Xét toàn bộ kênh màu RGB
- + lsb: LSB steganography (thay đổi bit cuối).
- + xy: Đọc ảnh theo thứ tự pixel trên trục tọa độ (x, y).

**FLAG:** picoCTF{7h3r3\_15\_n0\_5p00n\_a1062667}

### 2. hideme

AUTHOR: GEOFFREY NJOGU

Description

Every file gets a flag.

The SOC analyst saw one image been sent back and forth between two people.

They decided to investigate and found out that there was more than what meets the eye [here](#).

debug info: [u:399501 e: p: c350 i:294967]

13.906 users solved

Hints ? (None)

Submit Flag

- Xem thì thấy ảnh vẫn có chứa gì trong đó

File name:	flag.png
Folder:	C:\Users\daoxu\Downloads\
Full path:	C:\Users\daoxu\Downloads\flag.png
Compression:	PNG - ZIP
Resolution:	512 x 504 DPI
Original size:	512 x 504 Pixels (1.01)
Current size:	512 x 504 Pixels (1.01)
Print size (from DPI):	18.1 x 17.8 cm; 7.11 x 7.00 inches
Original colors:	16.7 Million (32 BitsPerPixel)
Current colors:	16.7 Million (32 BitsPerPixel)
Number of unique colors:	1791
Disk size:	42.05 KB (43,058 Bytes)
Current memory size:	1008.04 KB (1,032,232 Bytes)
Current folder/list index:	13 / 21
File date/time:	3/31/2025 / 22:16:02
Loaded in:	0 milliseconds

OK

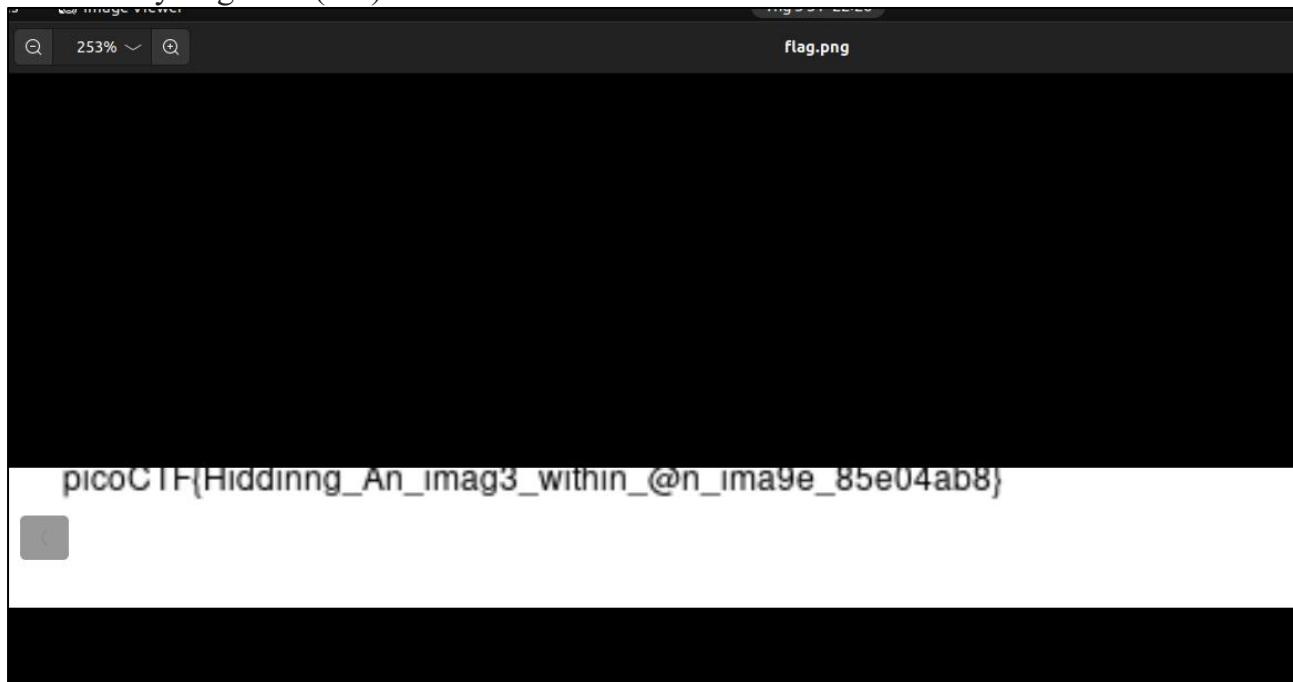
## Lab 2: Steganography & Steganalysis

- Binwalk ra được file secret/flag.png

```
emay@emay:~/Downloads$ binwalk flag.png
DECIMAL      HEXADECIMAL      DESCRIPTION
----          -----          -----
0            0x0              PNG image, 512 x 504, 8-bit/color RGBA, non-interlaced
41           0x29             Zlib compressed data, compressed
39739        0x9B3B          Zip archive data, at least v1.0 to extract, name: secret/
39804        0x9B7C          Zip archive data, at least v2.0 to extract, compressed size: 2997, uncompressed size: 3152, name: secret/flag.png
43036        0xA81C          End of Zip archive, footer length: 22

emay@emay:~/Downloads$ binwalk -e flag.png
DECIMAL      HEXADECIMAL      DESCRIPTION
----          -----          -----
0            0x0              PNG image, 512 x 504, 8-bit/color RGBA, non-interlaced
41           0x29             Zlib compressed data, compressed
39739        0x9B3B          Zip archive data, at least v1.0 to extract, name: secret/
39804        0x9B7C          Zip archive data, at least v2.0 to extract, compressed size: 2997, uncompressed size: 3152, name: secret/flag.png
43036        0xA81C          End of Zip archive, footer length: 22
```

- Thấy flag luôn (????)



- Ngựa ngựa đi check thêm vài info khác để hiểu thêm cách giấu thông tin

```
emay@emay:~/Downloads$ zsteg flag.png
[?] 3319 bytes of extra data after image end (IEND), offset = 0x9b3b
extradata:0 .. file: Zip archive data, at least v1.0 to extract, compression method=store
00000000: 50 4b 03 04 0a 00 00 00 00 00 37 10 70 56 00 00 |PK.....7.pV...
00000010: 00 00 00 00 00 00 00 00 00 07 00 1c 00 73 65 |.....se|
00000020: 63 72 65 74 2f 55 54 09 00 03 89 78 12 64 89 78 |cret/UT....x.d.x|
00000030: 12 64 75 78 0b 00 01 04 00 00 00 00 04 00 00 00 |.dux....|
00000040: 00 50 4b 03 04 14 00 00 00 08 00 37 10 70 56 90 |.PK.....7.pV.|
00000050: ce 3f 0a b5 0b 00 00 50 0c 00 00 0f 00 1c 00 73 |.?....P.....s|
00000060: 65 63 72 65 74 2f 66 6c 61 67 2e 70 6e 67 55 54 |ecret/flag.pngUT|
00000070: 09 00 03 89 78 12 64 89 78 12 64 75 78 0b 00 01 |....x.d.x.dux...|
00000080: 04 00 00 00 00 04 00 00 00 00 cd 57 6b 3c d3 7d |.....Wk<.}|
00000090: 1b ff 3b 64 3a 38 94 66 e4 cc 9d 3b 37 b2 10 1a |...;d:8.f...;7...|
000000a0: 92 35 39 64 68 c8 74 3b e5 30 6d e5 6c 8d 2d a7 |.59dh.t;.0m.l.-.|
000000b0: 6e ee 92 42 69 21 87 95 f2 b0 e4 dc 56 9b 32 72 |n..Bi!.....V.2r|
000000c0: e7 90 63 88 4c 4c c4 34 4c ca c8 0c cf 9e 97 cf |..c.LL.4L.....|
000000d0: 8b e7 fd f3 7d 71 7d af e3 e7 f7 fb 5c 9f eb c5 |.....}q}.....\...|
000000e0: 75 65 7a b8 39 ca ed 39 b4 07 00 00 39 67 27 04 |uez.9..9....9g'|
000000f0: 0a 00 24 d1 62 dd 54 51 2c 00 41 84 fe 79 31 49 |..$.b.TQ,.A..y1I|
```

emay@emay:~/Downloads\$ pngcheck flag.png

flag.png additional data after IEND chunk

ERROR: flag.png

## Lab 2: Steganography & Steganalysis

- Ở đây thì ta có thể thấy dấu hiệu cho thấy các thông tin được nén trong hex của png. PNG là định dạng chunk-based với chunk cuối là IEND để đánh dấu kết thúc file png, thông tin được giấu sau chunk này nên file PNG vẫn mở lên xem bình thường, nhưng khi dùng các lệnh check mã của file thì sẽ báo lỗi.
- Lệnh pngcheck báo lỗi có additional data sau chunk này, lệnh zsteg cho thấy phần hex của chunk này. PK là ký tự bắt đầu chunk, search thêm thì biết file zip thường bắt đầu bằng các byte 50 4b 03 04 xx 00.
- Ok vậy là hiểu kỹ thuật giấu thông tin sau các chunk của png rồi.

**FLAG:** picoCTF{Hiddinng\_An\_imag3\_within\_@n\_imagine\_85e04ab8}

### 3. MSB

- Theo decription của challenge thì có vẻ ảnh này sử dụng msb – kỹ thuật thay đổi bit quan trọng nhất để ẩn dữ liệu
- Mở ảnh thì thấy bị rối màu, thử test bằng cách tool trong bài học thì toàn LSB nên không ra gì



## - Lab 2: Steganography & Steganalysis

- Đi search vài tool MSB thì tìm được [tool này](#), decode được cả lsb và msb.
  - Test ngay

```
emay@emay:~/Downloads$ python3 sigBits.py -t=msb Ninja-and-Prince-Genji-Ukiyoe-Utagawa-Kunisada.flag.png
Done, check the output file!
emay@emay:~/Downloads$ cat sigBits.py | less

[1]+  Stopped                  cat sigBits.py | less
emay@emay:~/Downloads$ ls
bin flag.png Ninja-and-Prince-Genji-Ukiyoe-Utagawa-Kunisada.flag.png ninjahex.txt outputSB.txt pico.flag.png sigBits.py stegsolve.sh
emay@emay:~/Downloads$ cat outputSB.txt | less

[2]+  Stopped                  cat outputSB.txt | less
emay@emay:~/Downloads$
```

- Trông có vẻ đau mắt

- Grep thì thấy flag

```
emay@emay:~/Downloads$ cat outputSB.txt | grep -o "picoCTF"  
picoCTF
```

- Gren ra h&agrave;n flag

```
emay@emay:~/Downloads$ cat outputSB.txt | grep -o -E "picoCTF.{0,50}"  
picoCTF{15_yOur_que57_qu1x071c_0r_h3r01c_06326238}Thou h  
emay@emay:~/Downloads$
```

FLAG: picoCTF{15\_your\_flag57\_qu1x071c\_Or\_h3r01c\_06326238}

## Lab 2: Steganography & Steganalysis

### 4. Glory of the Garden

Glory of the Garden 

Easy Forensics picoCTF 2019

AUTHOR: JEDAVIS/DANNY

Description

This [garden](#) contains more than it seems.

Hints ? 1 What is a hex editor?

76,157 users solved

 89% Liked 

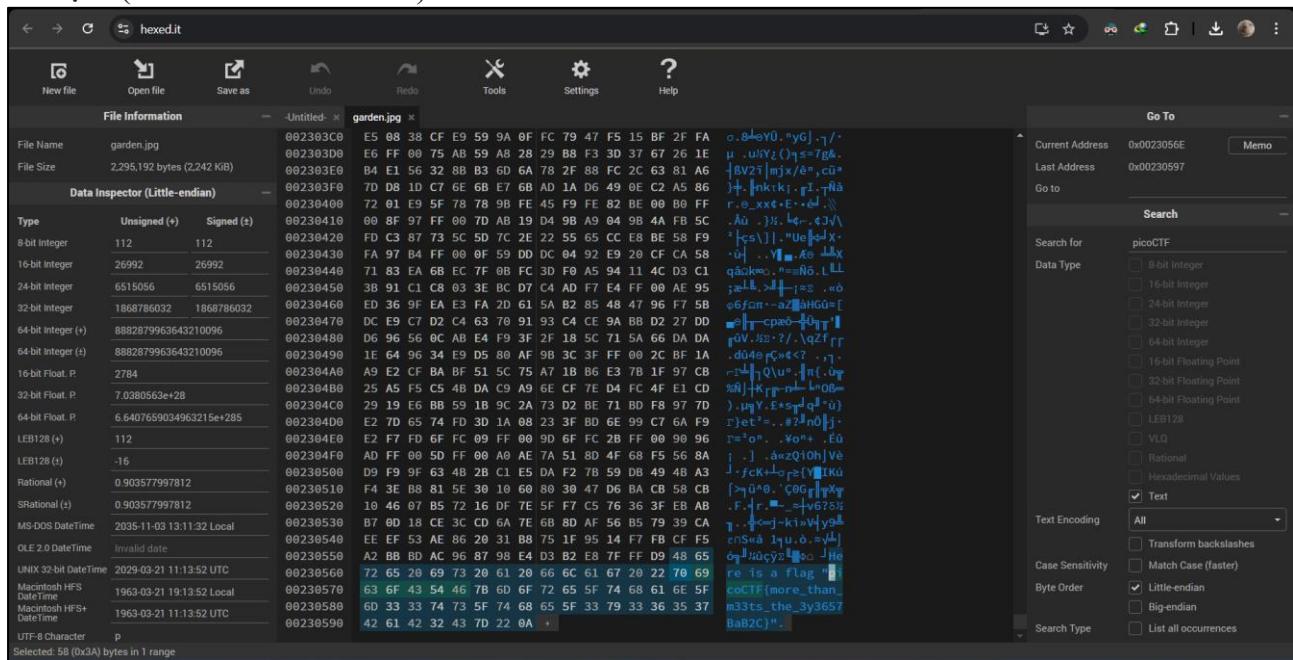
picoCTF{FLAG} Submit Flag

Bài cho thông tin là 1 file ảnh kèm với 1 hint về hex editor.



## Lab 2: Steganography & Steganalysis

Mở ảnh ra thì thấy đây chỉ là 1 bức ảnh bình thường, sử dụng một công cụ [hex editor online](#) mở tạm (do lười mở VMware):



Do công cụ này có sẵn phần search nên chỉ với việc search cụm “picoCTF” thì đã tìm ra được flag.

**Flag: picoCTF{more\_than\_m33ts\_the\_3y3657BaB2C}**

---

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

## YÊU CẦU CHUNG

- Sinh viên tìm hiểu và thực hiện bài tập theo yêu cầu, 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 **.DOCX** và **.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]-ExeX\_GroupY. (trong đó X là Thứ tự Bài tập, Y là mã số thứ tự nhóm trong danh sách mà GV phụ trách công bố).

*Ví dụ: /NT101.K11.ANTT]-Exe01\_Group03.*

- 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 bài nộp.**
- Nộp file báo cáo trên theo thời gian đã thông nhất tại [courses.uit.edu.vn](http://courses.uit.edu.vn).

### Đánh giá:

- Hoàn thành tốt yêu cầu được giao.
- Có nội dung mở rộng, ứng dụng.

*Bài sao chép, trẽ, ... sẽ được xử lý tùy mức độ vi phạm.*

**HẾT**