



# BÁO CÁO THỰC HÀNH

Môn học: Pháp chứng kỹ thuật số  
Lab 3: Steganography & Steganalysis

GVHD: Đoàn Minh Trung

1. **THÔNG TIN CHUNG:**  
(Liệt kê tất cả các thành viên trong nhóm)  
Lớp: NT334.P11.ATCL.1

STT	Họ và tên	MSSV	Email
1	Nguyễn Đại Nghĩa	21521182	21521182@gm.uit.edu.vn
2	Phạm Hoàng Phúc	21521295	21521295@gm.uit.edu.vn
3	Lê Xuân Sơn	21521386	21521386@gm.uit.edu.vn

2. **NỘI DUNG THỰC HIỆN:**<sup>1</sup>

STT	Công việc	Kết quả tự đánh giá
1	Bài tập 1	100%
2	Bài tập 2	0%
3	Bài tập 3	100%
4	Bài tập 4	100%
5	Bài tập 5	100%
6	Bài tập 6	100%
7	Bài tập 7	100%
8	Bài tập 8	100%
9	Bài tập 9	100%
10	Bài tập 10	100%

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

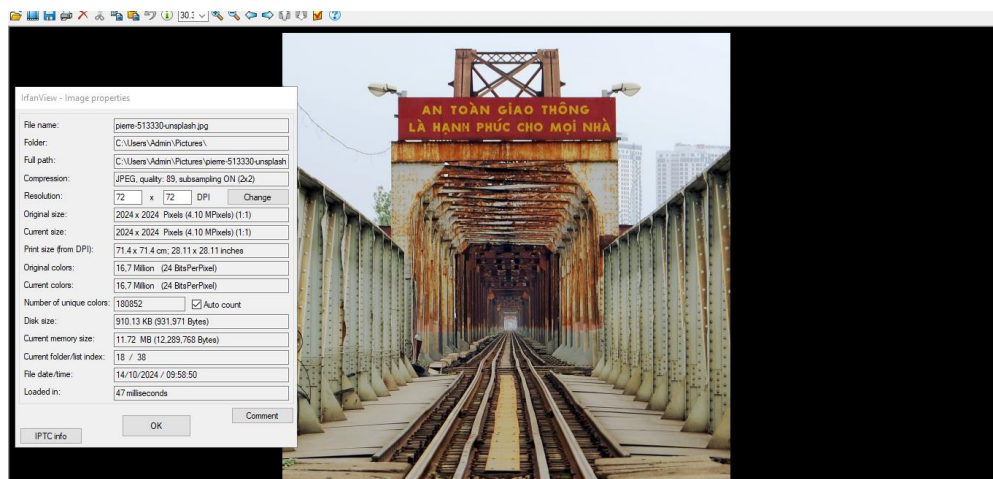
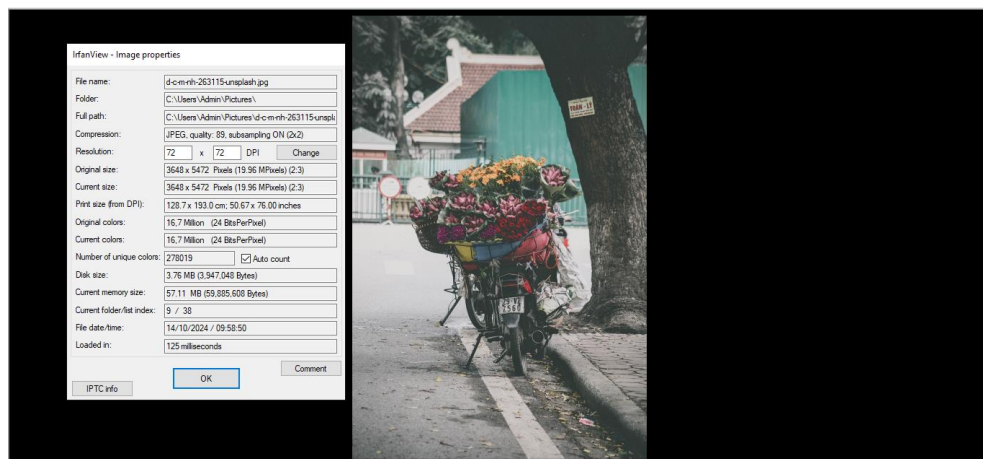
<sup>1</sup> Ghi nội dung công việc, các kịch bản trong bài Thực hành

## BÁO CÁO CHI TIẾT

### Kịch Bản 1-a:

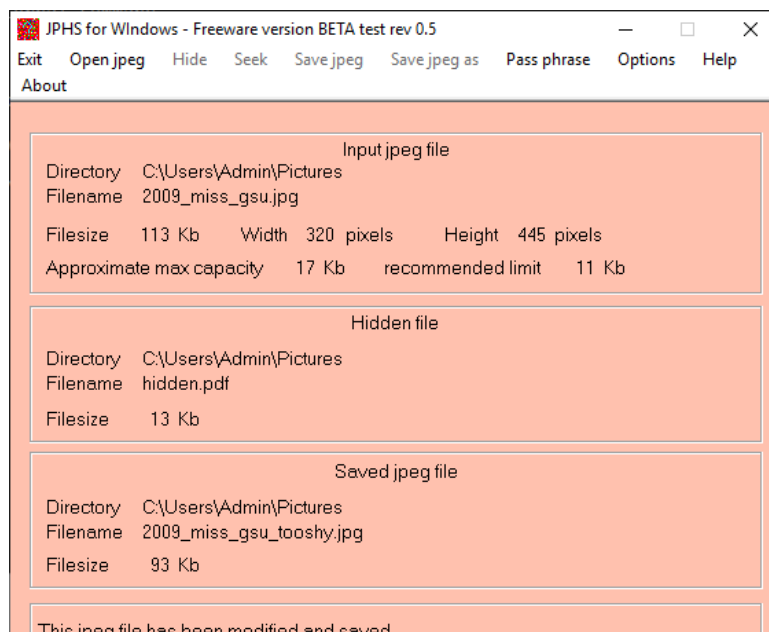
Thực hiện vào image -> infor để coi thông tin của 2 ảnh.

Cả 2 đều không có exif info

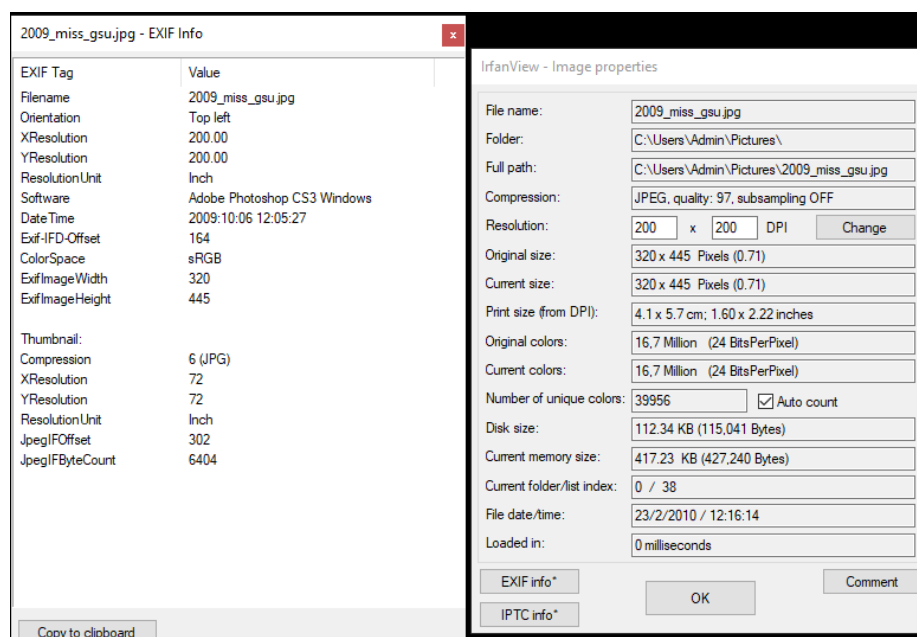


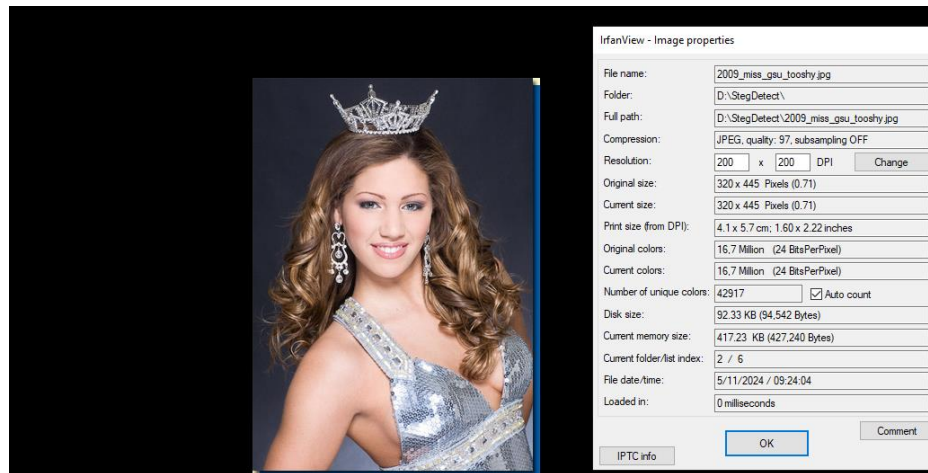
### Kịch Bản 1-b:

Sử dụng JPHS, thực hiện giấu file pdf vào trong ảnh.



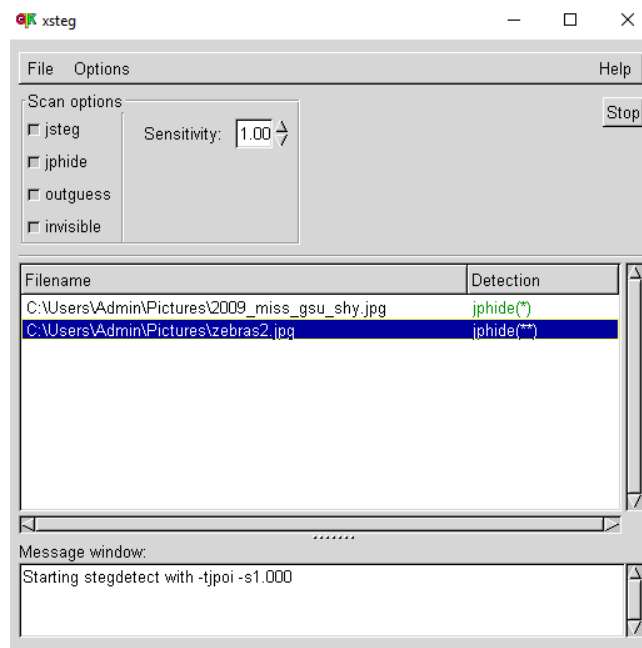
So sánh giữa thông tin 2 ảnh gốc và ảnh đã được sửa đổi





### Kịch Bản 1-c:

Sử dụng xsteg, ta phát hiện được là hình ảnh đã được thay đổi bằng jphide.



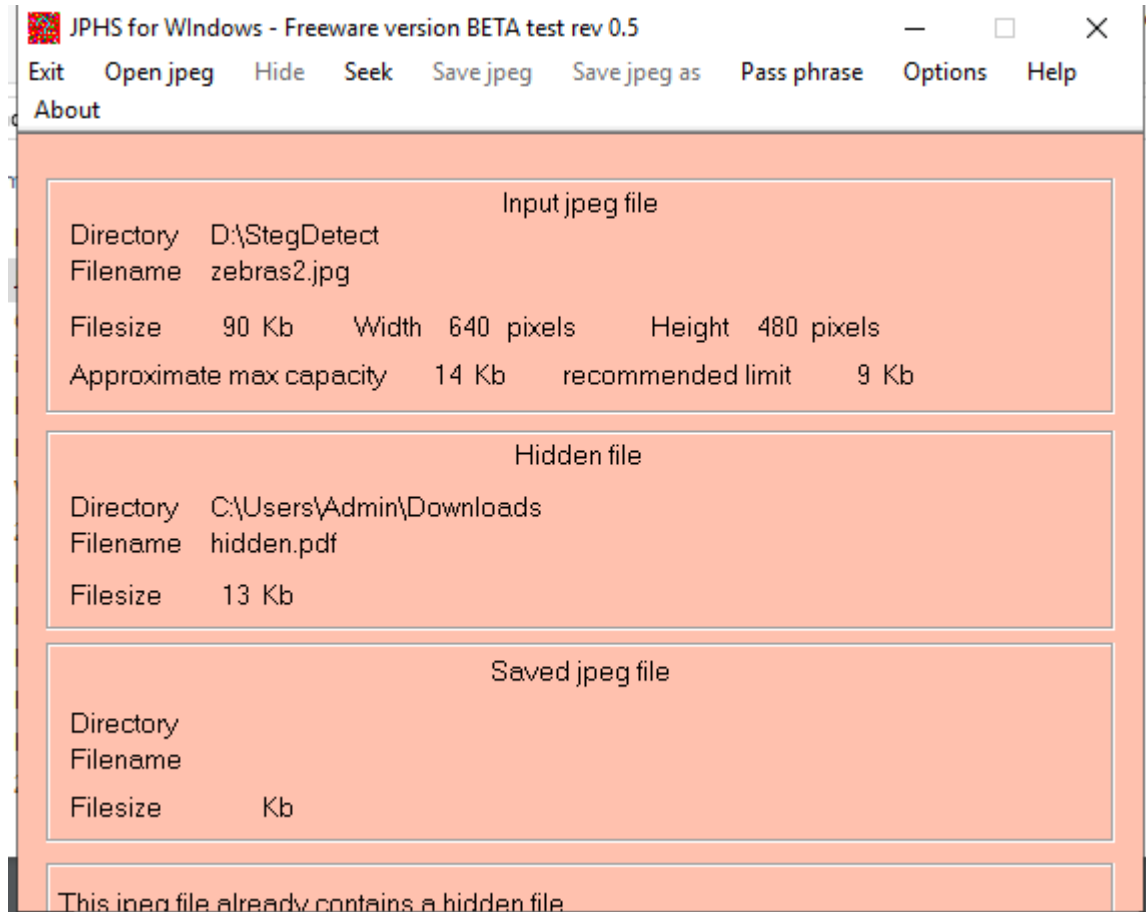
Sử dụng stegdetect, ta bruteforce mật khẩu đã dùng để giấu thông tin bằng jphide, được pass là together

```
D:\StegDetect>.\stegbreak.exe -r rules.ini -f MedDict.DIC zebras2.jpg
Loaded 1 files...
zebras2.jpg : jphide[v5](together)
Processed 1 files, found 1 embeddings.
Time: 5 seconds: Cracks: 68607, 13721.4 c/s
```

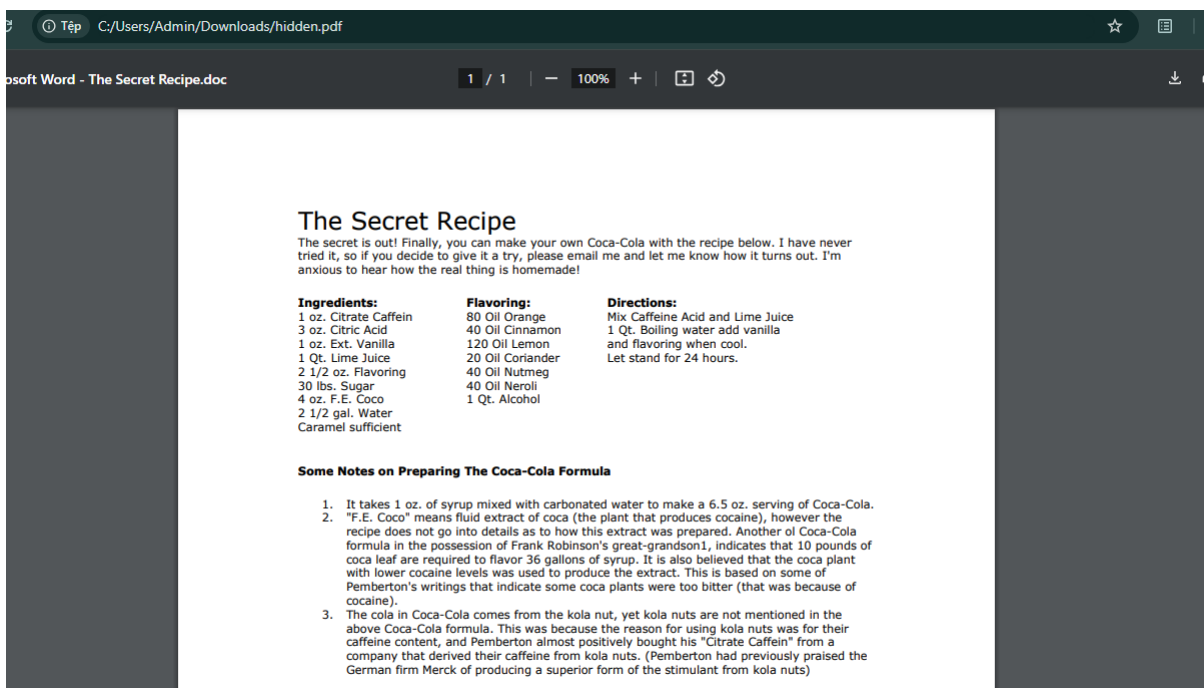
Dưới đây là pass của ảnh đã thay đổi ở bài trước

```
D:\StegDetect>.\stegbreak.exe -r rules.ini -f MedDict.DIC 2009_miss_gsu_tooshy.jpg
Corrupt JPEG data: 31 extraneous bytes before marker 0xd9
Loaded 1 files...
2009_miss_gsu_tooshy.jpg : jphide[v5](help)
Processed 1 files, found 1 embeddings.
Time: 2 seconds: Cracks: 31784, 15892.0 c/s
```

Thực hiện seek ảnh bằng jphide để tìm ra file được ẩn giấu trong ảnh bằng mật khẩu đã bruteforce được. Chuyển nó về định dạng pdf

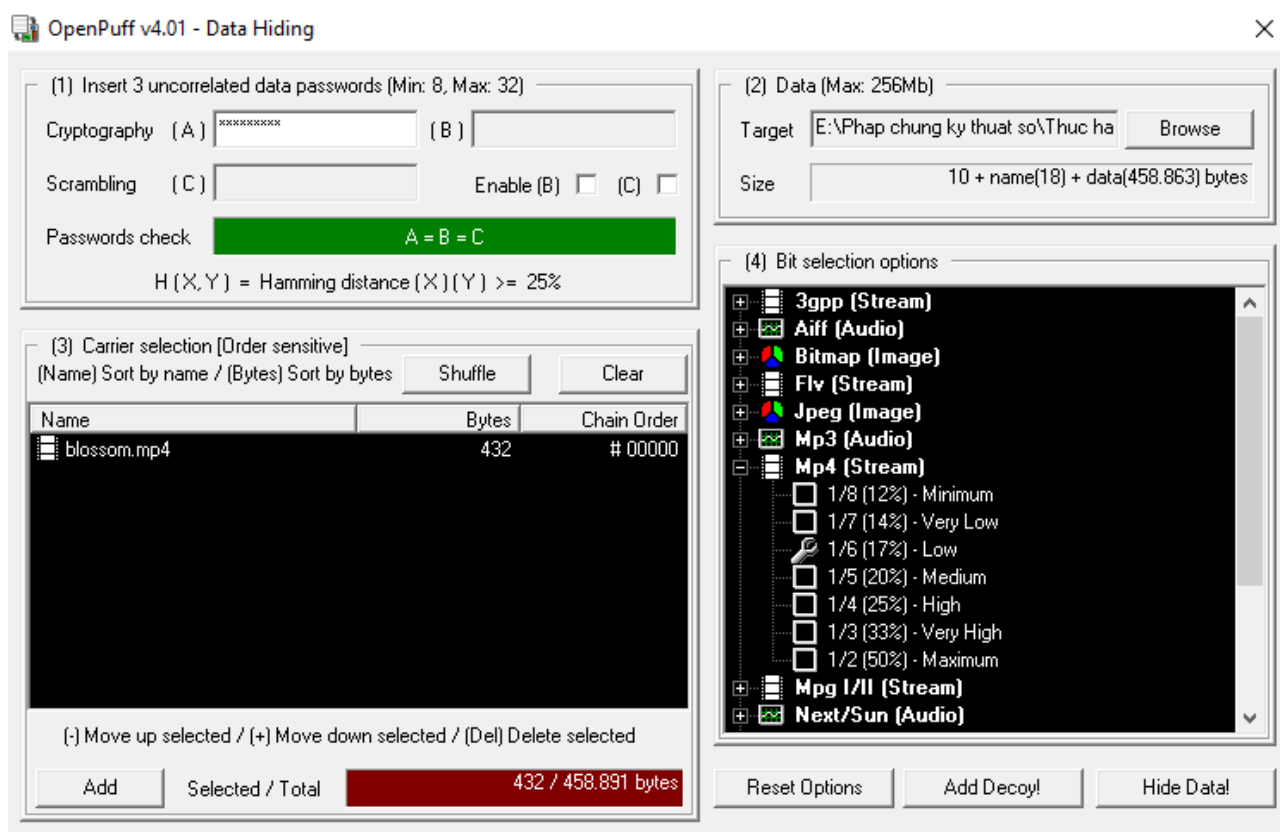


Ta được file đã được ẩn giấu trong tấm ảnh



## Kịch Bản 2:

Bởi vì không tìm tải được Our Secret nên em sẽ sử dụng phần mềm khác là OpenPuff, nhưng kết quả không mấy khả quan do carrier không đủ để chứa file ảnh:



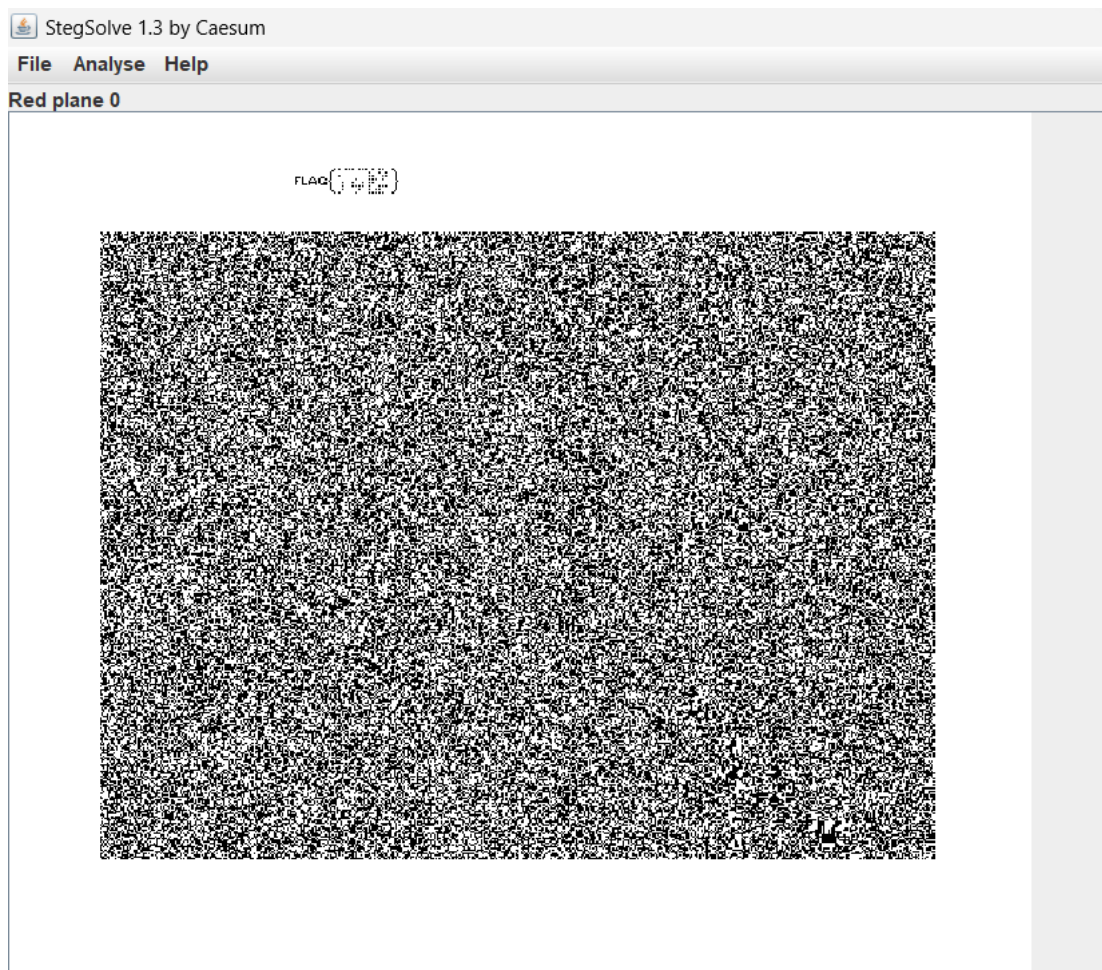
Ngoài OpenPuff ra thì em cũng không tìm ra được bất kì phần mềm miễn phí khác có khả năng hide ảnh vào một file mp4.

Thế nên nhóm em đành chịu với kịch bản 2 này.

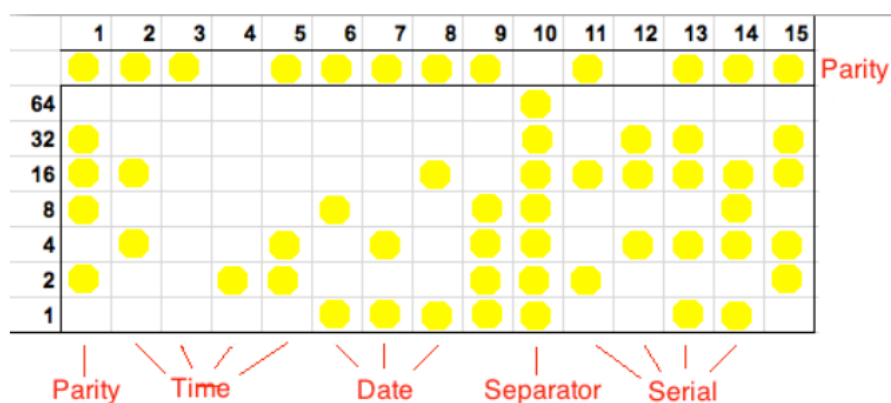
## Kịch bản 03:

Sử dụng công cụ **StegSolve** điều chỉnh ảnh đến **Red plane 0** để tìm ra FLAG được viết dưới dạng bảng chữ nổi cho người khiếm thị.





Theo thông tin mà đề bài cung cấp, số seri sẽ nằm từ cột 11-13 hoặc từ cột 11-14



Giải mã Flag

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	x		x	x	x		x	x	x		x		x	x	
64										x			x		x
32	x									x	x			x	x
16						x				x	x	x		x	
8		x			x			x		x	x				
4		x			x	x	x	x		x			x	x	x
2		x					x			x		x	x		
1	x					x				x	x	x	x	x	
											57	19	71	53	



Số seri là: **711957** hoặc **53711957**

### Kịch bản 4:

Bước đầu em xem qua thử các thông tin cơ bản của file jpg này:

```
File Actions Edit View Help
(nghianguyen@kali)-[~/phap chung/lab 3]
$ file star-wars.jpg
star-wars.jpg: JPEG image data, JFIF standard 1.01, resolution (DPI), density 96x96, segment length 16, baseline, precision 8, 655x368, components 3
(nghianguyen@kali)-[~/phap chung/lab 3]
$
```

Tiếp đến là binwalk:

```
(nghianguyen@kali)-[~/phap chung/lab 3]
$ binwalk star-wars.jpg

DECIMAL      HEXADECIMAL  DESCRIPTION
-----
0            0x0          JPEG image data, JFIF standard 1.01

(nghianguyen@kali)-[~/phap chung/lab 3]
$
```

Vẫn chưa thấy gì đáng ngờ cả.

Em chạy tiếp với option -W để xem nếu có output khác:

```
(nghianguyen@kali)-[~/phap chung/lab 3]
$ binwalk -w star-wars.jpg

OFFSET      star-wars.jpg
0x00000000  FF D8 FF E0 00 10 4A 46 49 46 00 01 01 01 00 60 | .....JFIF.....`|
0x00000010  00 60 00 00 FF DB 00 43 00 20 16 18 1C 18 14 20 | .`.....C.....|
0x00000020  1C 1A 1C 24 22 20 26 30 50 34 30 2C 2C 30 62 46 | ...$.60P40,,0bF|
0x00000030  4A 3A 50 74 66 7A 78 72 66 70 6E 80 90 B8 9C 80 | J:Ptfzxrfpn....|
0x00000040  88 AE 8A 6E 70 A0 DA A2 AE BE C4 CE D0 CE 7C 9A | ...np.....|.|
0x00000050  E2 F2 E0 C8 F0 B8 CA CE C6 FF DB 00 43 01 22 24 | .....C."$|
0x00000060  24 30 2A 30 5E 34 34 5E C6 84 70 84 C6 C6 C6 C6 | $0*0^44^ ..p....|
0x00000070  C6 C6 C6 C6 C6 C6 C6 C6 C6 C6 C6 C6 C6 C6 C6 | .....|
0x00000080  C6 C6 C6 C6 C6 C6 C6 C6 C6 C6 C6 C6 C6 C6 C6 | .....|
0x00000090  C6 C6 C6 C6 C6 C6 C6 C6 C6 C6 C6 C6 C6 FF C0 | .....|
0x000000A0  00 11 08 01 70 02 8F 03 01 22 00 02 11 01 03 11 | ....p...."......|
0x000000B0  01 FF C4 00 1F 00 00 01 05 01 01 01 01 01 01 00 | .....|
0x000000C0  00 00 00 00 00 00 00 01 02 03 04 05 06 07 08 09 | .....|
0x000000D0  0A 0B FF C4 00 85 10 00 02 01 03 03 02 04 03 05 | .....|
0x000000E0  05 04 04 00 00 01 7D 01 02 03 00 04 11 05 12 21 | .....}.....!|
0x000000F0  31 41 06 13 51 61 07 22 71 14 32 81 91 A1 08 23 | 1A..Qa."q.2....#|
0x00000100  42 B1 C1 15 52 D1 F0 24 33 62 72 82 09 0A 16 17 | B ... R.. $3br....|
0x00000110  18 19 1A 25 26 27 28 29 2A 34 35 36 37 38 39 3A | ...%6'()*456789:|
0x00000120  43 44 45 46 47 48 49 4A 53 54 55 56 57 58 59 5A | CDEFGHIJSTUVWXYZ|
0x00000130  63 64 65 66 67 68 69 6A 73 74 75 76 77 78 79 7A | cdefghijstuvwxyz|
0x00000140  83 84 85 86 87 88 89 8A 92 93 94 95 96 97 98 99 | .....|
```

Kết quả là ở cuối em thấy được một đoạn binary sau:

```
0x00004E90  8A 4C D1 4C 91 68 A2 8A 06 14 51 45 00 14 51 45 | .L.L.h....QE..QE|
0x00004EA0  00 14 51 45 00 14 51 45 00 14 51 45 00 14 51 45 | ..QE..QE..QE..QE|
0x00004EB0  00 14 51 45 30 0A 5C D2 51 40 1F FF D9 31 30 30 | ..QE0.\.Q@ ...100|
0x00004EC0  31 31 30 31 30 31 30 31 30 31 30 31 30 31 30 31 | 1101010101010101|
0x00004ED0  31 31 30 31 30 31 30 31 30 31 30 31 30 31 30 31 | 1101010011010101|
0x00004EE0  30 31 30 31 31 31 30 31 30 31 30 31 30 31 31 31 | 0101110101010011|
0x00004EF0  31 31 30 0A XX XX XX XX XX XX XX XX XX XX XX XX | 110.....|
```

Em ban đầu chia nó ra thành 8 bits mỗi đoạn, nhưng không phân tích được gì.

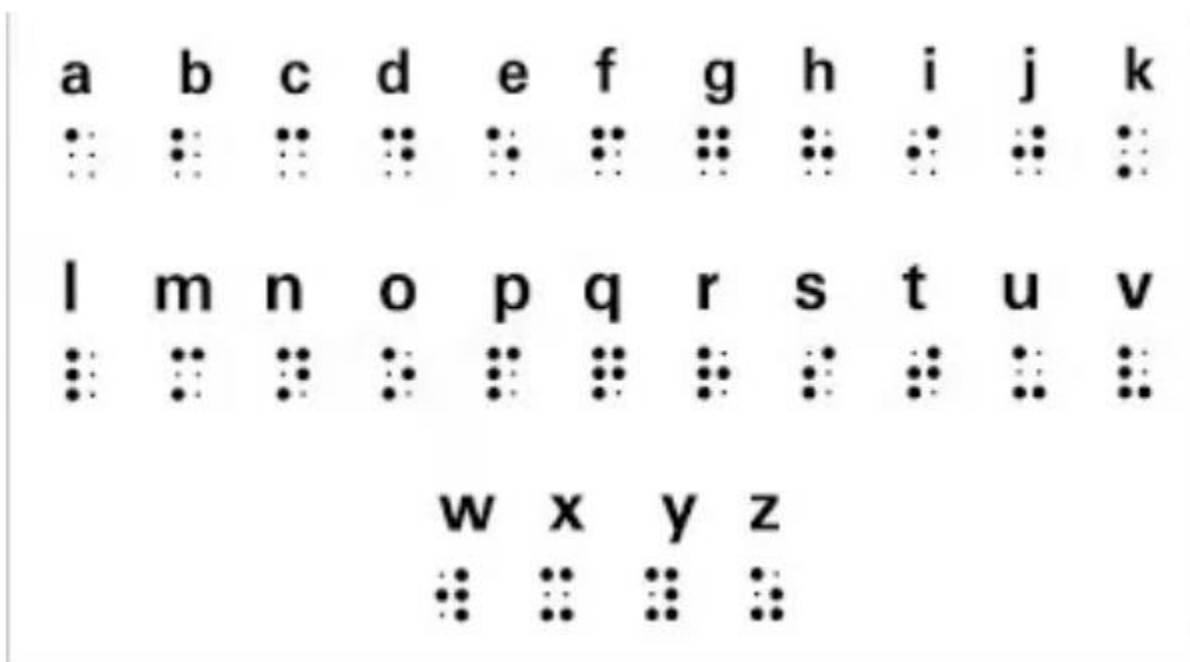
Sau đó em thấy được braille binary với 6 bits, nên em đã chia theo 6 bits và dựa vào braille binary phân tích được như dưới đây:



\*Untitled - Notepad

File Edit Format View Help

```
100110 101010 101010 111010 100110 101010 101110 101010 011110
d      o      o      r      d      o      n      o      t
```



Tiếp đến là em sử dụng steghide để lấy thứ đã được nhúng vào file pdf này, với pass là chuỗi kí tự đã tìm được ở trên:


```
(nghianguyen@kali)-[~/phap chung/lab 3]
$ steghide extract -sf star-wars.jpg
Enter passphrase:
wrote extracted data to "flag.txt".

(nghianguyen@kali)-[~/phap chung/lab 3]
$ cat flag.txt
YmVjb21lYWplZGltYXN0ZXJ5b3V3aWxs

(nghianguyen@kali)-[~/phap chung/lab 3]
$
```

Kết quả là em lấy được là 1 file flag.txt với nội dung như trong hình.


Giải mã đoạn kí tự đó với base64, em tìm được chuỗi cần tìm là “becomeajedimasteryouwill” như dưới:

 <https://www.base64decode.org>

## Decode from Base64 format

Simply enter your data then push the decode button.

YmVjb21lYWplZGltYXN0ZXJ5b3V3aWxs


 For encoded binaries (like images, documents, etc.) use the file

UTF-8

▼

Source character set.

☐ Decode each line separately (useful for when you have multiple

 Live mode OFF

Decodes in real-time as you type or paste

< **DECODE** >

Decodes your data into the area below.

becomeajedimasteryouwill

BỘ MÔN  
AN TOÀN THÔNG TIN

Báo cáo môn học  
HỌC KỲ I – NĂM HỌC 2024-2025

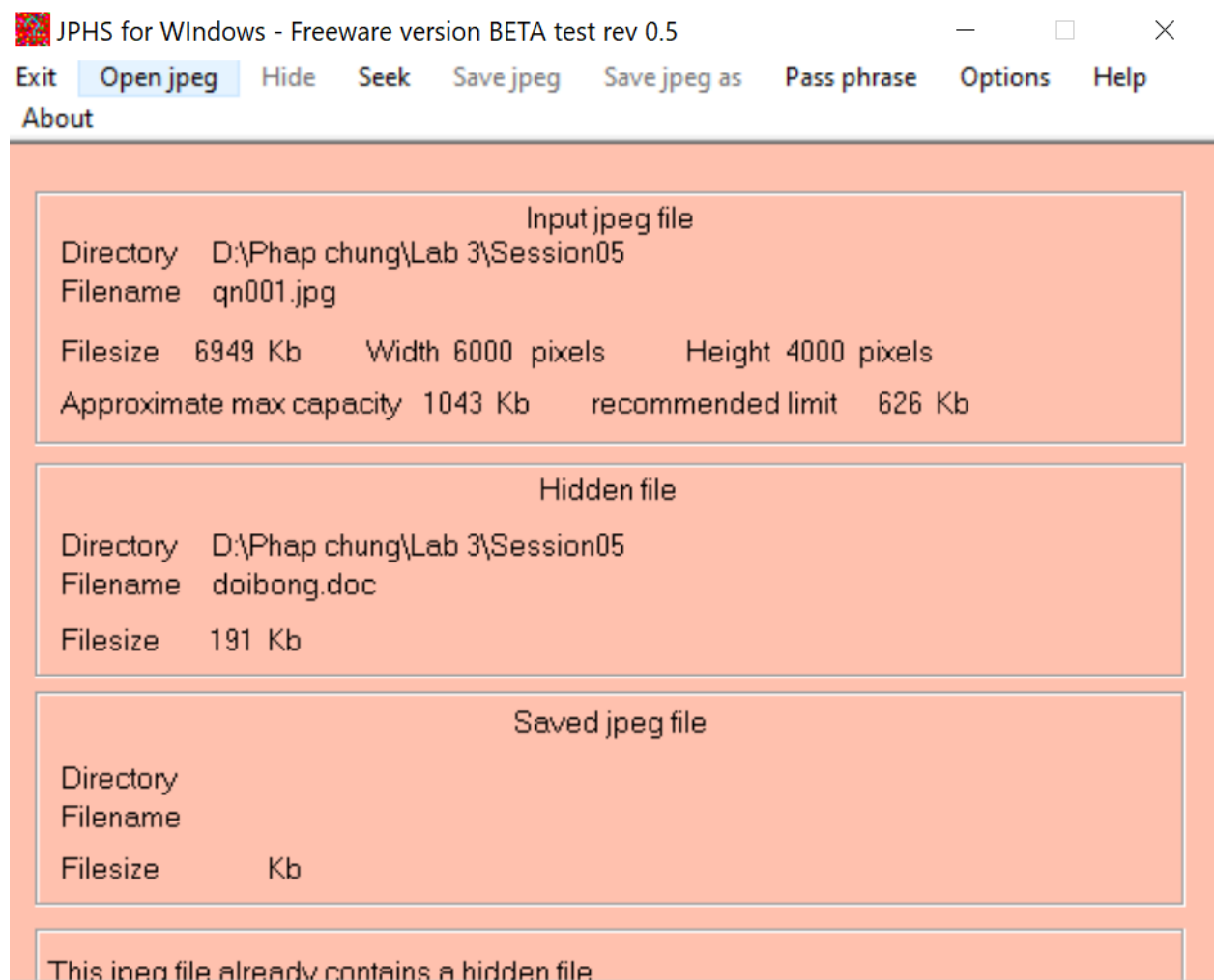
**Kịch bản 5:**

Cùng với tool stegbreak, em sử dụng rockyou.txt làm dictionary, vì đây là một dictionary rất nổi tiếng cho việc break pass steganography:

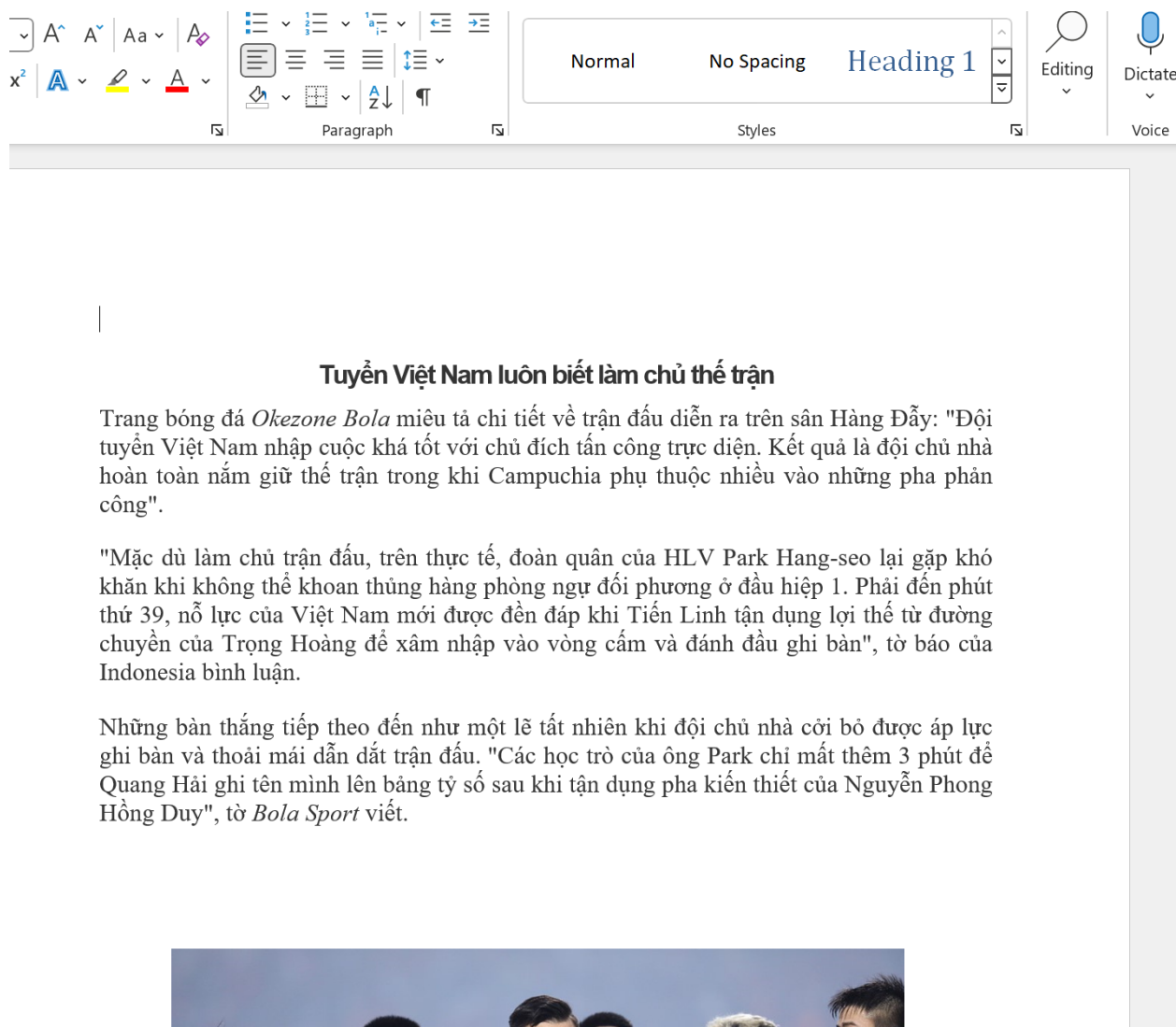
```
D:\Phap chung\Lab 3>stegdetect04_session03\stegbreak.exe -r stegdetect04_session03\rules.ini -f .\rockyou.txt Session05\qn001.jpg
Corrupt JPEG data: bad Huffman code
Loaded 1 files...
Session05\qn001.jpg : jphide[v5]()
Processed 1 files, found 1 embeddings.
Time: 1 seconds: Cracks: 4751, 4751.0 c/s
```

Kết quả cho em thấy được có 1 file nhúng vào trong file jpg này, và passparse thì không có.

Tiếp theo em sử dụng JPHS để lấy file được nhúng vào file jpg:



File này phải đặt thành .doc thì mới đọc được, đây là 1 file doc viết về đội tuyển Việt Nam:

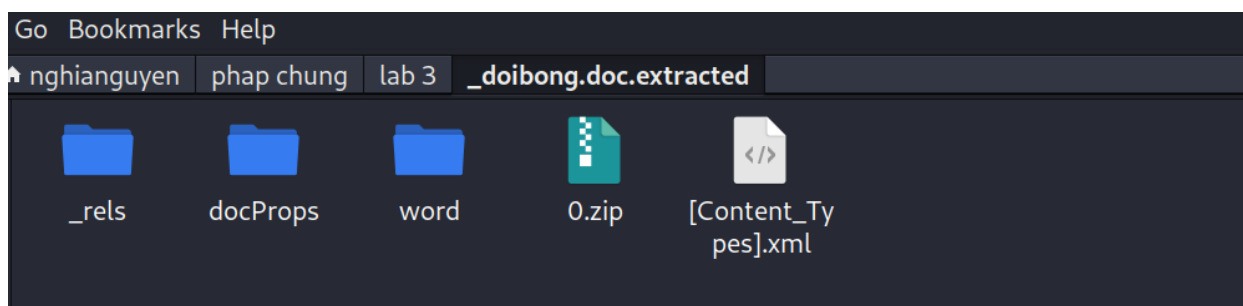


Em tiếp tục phân tích tệp file doc này với binwalk -e để trích xuất các dữ liệu được tìm thấy trong file doc này:

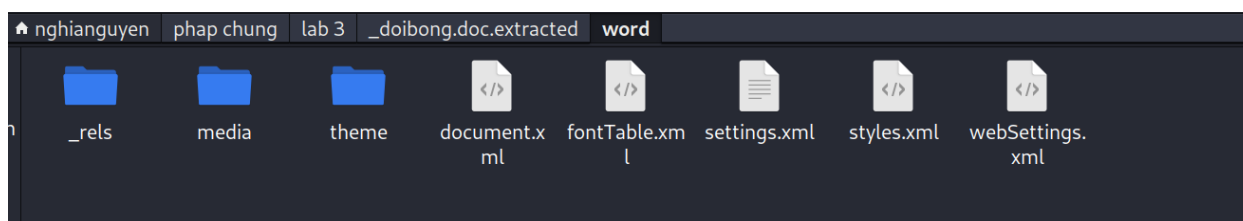
```
(ngianguyen@kali) ~/phap chung/lab 3
$ binwalk -e doibong.doc
```

DECIMAL	HEXADECIMAL	DESCRIPTION
0	0x0	Zip archive data, at least v2.0 to extract, compressed size: 359, uncompressed size: 1363, name: [Content_Types].xml
928	0x3A0	Zip archive data, at least v2.0 to extract, compressed size: 239, uncompressed size: 590, name: _rels/.rels
1728	0x6C0	Zip archive data, at least v2.0 to extract, compressed size: 3915, uncompressed size: 19670, name: word/document.xml
5600	0x163A	Zip archive data, at least v2.0 to extract, compressed size: 264, uncompressed size: 949, name: word/_rels/document.xml.rels
6276	0x1884	Zip archive data, at least v1.0 to extract, compressed size: 178935, uncompressed size: 178935, name: word/media/image1.jpg
185262	0x2D3AE	Zip archive data, at least v2.0 to extract, compressed size: 1538, uncompressed size: 7076, name: word/theme/theme1.xml
186851	0x2D9E3	Zip archive data, at least v2.0 to extract, compressed size: 1118, uncompressed size: 3160, name: word/settings.xml
188016	0x2DE70	Zip archive data, at least v2.0 to extract, compressed size: 3267, uncompressed size: 31584, name: word/styles.xml
191328	0x2EB60	Zip archive data, at least v2.0 to extract, compressed size: 471, uncompressed size: 2670, name: word/webSettings.xml
191849	0x2ED69	Zip archive data, at least v2.0 to extract, compressed size: 576, uncompressed size: 1968, name: word/fontTable.xml
192473	0x2FDD9	Zip archive data, at least v2.0 to extract, compressed size: 386, uncompressed size: 747, name: docProps/core.xml
193170	0x2F292	Zip archive data, at least v2.0 to extract, compressed size: 479, uncompressed size: 992, name: docProps/app.xml
194731	0x2F8AB	End of Zip archive, footer length: 22

Đây là các file đã trích xuất được:



Trong đó có 1 file tên là document.xml:



Nội dung của file này như sau:




```

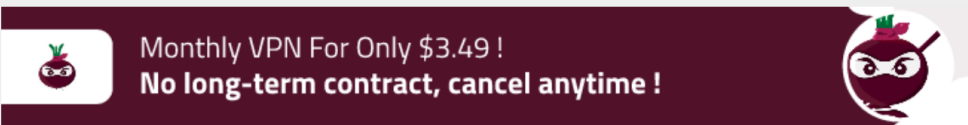
file:///home/ngianguyen/phap chung/lab 3/_doibong.doc.extracted/word/document.xml
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec
This XML file does not appear to have any style information associated with it. The document tree is shown below.
<?xml version="1.0" encoding="UTF-16" standalone="yes" type="text/xml">
<w:document mc:Ignorable="w14 w15 w16se w16cid wp14">
  <w:body>
    <w:p w14:paraId="03FFD0D9" w14:textId="77777777" w:rsidR="00B74DD0" w:rsidRPr="00415551" w:rsidRDefault="00B74DD0" w:rsidP="00B74DD0">
      <w:pPr>
        <w:pStyle w:val="NormalWeb"/>
        <w:spacing w:before="0" w:after="0" w:lineRule="auto"/>
        <w:rFonts w:ascii="Arial" w:hAnsi="Arial" w:cs="Arial"/>
      </w:pPr>
    </w:p>
    <w:p w14:paraId="24E82010" w14:textId="77777777" w:rsidR="00415551" w:rsidRPr="00415551" w:rsidRDefault="00415551" w:rsidP="00415551">
      <w:pPr>
        <w:shd w:val="clear" w:color="auto" w:fill="FFFFFF"/>
        <w:spacing w:before="150" w:after="150" w:line="240" w:lineRule="auto"/>
        <w:jc w:val="center"/>
        <w:textAlignment w:val="baseline"/>
        <w:outlineLvl w:val="2"/>
      </w:pPr>
      <w:r>
        <w:rFonts w:ascii="Arial" w:eastAsia="Times New Roman" w:hAnsi="Arial" w:cs="Arial"/>
        <w:b/>
        <w:bCs/>
        <w:color w:val="333333"/>
        <w:spacing w:val="15"/>
        <w:sz w:val="27"/>
        <w:szCs w:val="27"/>
        <w:lang w:eastAsia="zh-CN"/>
      </w:r>
    </w:p>
    <w:r w:rsidRPr="00415551">
      <w:rPr>
        <w:rFonts w:ascii="Arial" w:eastAsia="Times New Roman" w:hAnsi="Arial" w:cs="Arial"/>
        <w:b/>
        <w:bCs/>
        <w:color w:val="333333"/>
        <w:spacing w:val="15"/>
        <w:sz w:val="27"/>
        <w:szCs w:val="27"/>
        <w:lang w:eastAsia="zh-CN"/>
      </w:rPr>
      <w:t>Tuyển Việt Nam luôn biết làm chủ thế trận</w:t>
    </w:r>
  </w:p>
  <w:p w14:paraId="7822C2E0" w14:textId="77777777" w:rsidR="00415551" w:rsidRPr="00415551" w:rsidRDefault="00415551" w:rsidP="00415551">
    <w:pPr>
      <w:shd w:val="clear" w:color="auto" w:fill="FFFFFF"/>
      <w:spacing w:after="0" w:line="240" w:lineRule="auto"/>
      <w:jc w:val="both"/>
    </w:pPr>
  </w:p>

```

Theo em tìm hiểu thử qua thì trong file này có chứa 1 số kí tự là ngôn ngữ brainfuck.  
Thế nên em sẽ sử dụng công cụ Brainfuck Translator để giải:



## Brainfuck Translator



```

<w:pgSz w:w="12240" w:h="15840"/>
<w:pgMar w:top="1440" w:right="1440"
w:bottom="1440" w:left="1440" w:header="720"
w:footer="720" w:gutter="0"/>
<w:cols w:space="720"/>
<w:docGrid w:linePitch="360"/>
</w:sectPr>
</w:body>
</w:document>

```

```

Forensics05@UIT{Vietnam-win-Cambodia}

```

Argument(s)

Encode

Decode

Kết quả nhận được chính là “Forensics05@UIT{Vietnam-win-Cambodia}”.

## Kịch bản 6:

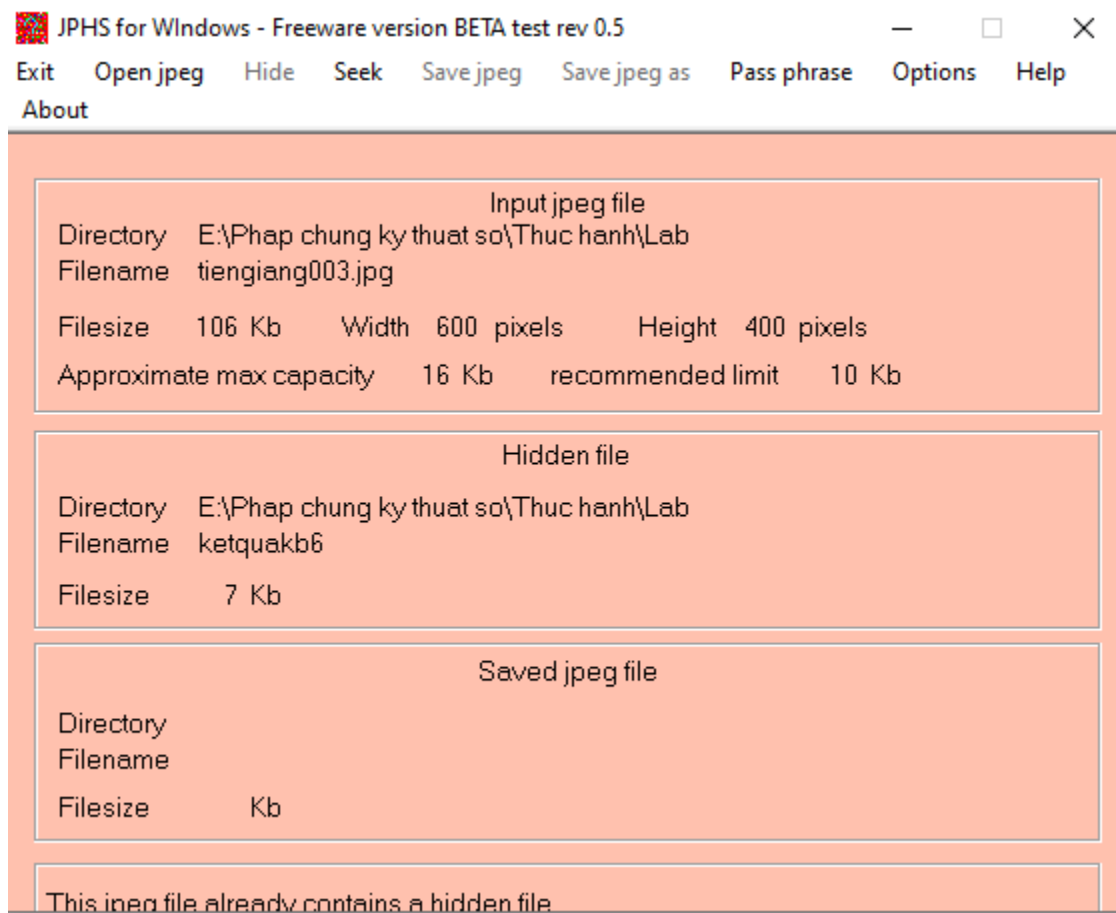
Bước đầu em sẽ sử dụng cách như kịch bản 5 đã làm để tìm ra passphrase cho file ảnh này:

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19045.5011]
(c) Microsoft Corporation. All rights reserved.

E:\Phap chung ky thuat so\Thuc hanh\Lab 3\session03-resources>stegdetect04_session03\stegbreak.exe -r stegdetect04_session03\rules.ini -f .\rockyou.txt kichbantonghop\tiengiang003.jpg
Loaded 1 files...
kichbantonghop\tiengiang003.jpg : jphide[v5]()
Processed 1 files, found 1 embeddings.
Time: 0 seconds: Cracks: 4751,      Inf c/s

E:\Phap chung ky thuat so\Thuc hanh\Lab 3\session03-resources>
```

Kết quả là không có passphrase, tiếp theo em sẽ sử dụng JPHS để lấy file được nhúng ra:



Em tiến hành đưa file đã lấy ra được vào trong kali để phân tích sâu hơn:

```
(nghianguyen@kali)-[~/PhapChung/ThucHanh3]
$ binwalk -W ketquakb6
```

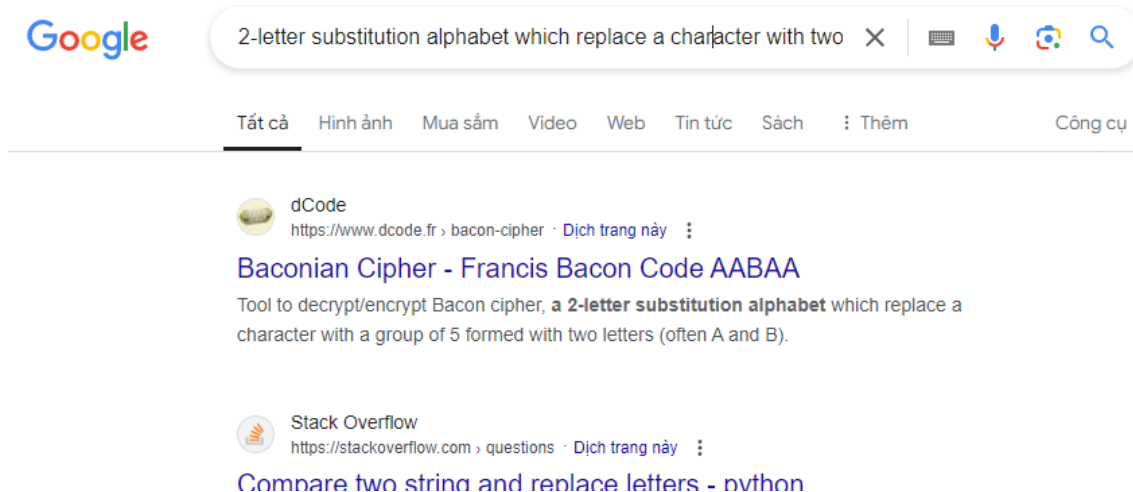
OFFSET	ketquakb6
0x00000000	89 50 4E 47 0D 0A 1A 0A 00 00 0D 49 48 44 52   .PNG.....IHDR
0x00000010	00 00 01 81 00 00 00 83 08 03 00 00 00 5C 38 57   ..... \8W
0x00000020	A9 00 00 00 84 50 4C 54 45 FF FF FF 03 03 03 00   .....PLTE.....
0x00000030	00 00 EC EC EC E1 E1 E1 E5 E5 E5 DD DD DD B7 B7   .....
0x00000040	B7 F7 F7 F7 EF EF EF FC FC FC E9 E9 E9 81 81 81   .....
0x00000050	C6 C6 C6 BB BB BB D0 D0 D0 41 41 41 3B 3B 3B 8A   .....AAA;;;.
0x00000060	8A 8A 93 93 93 67 67 67 A9 A9 A9 53 53 53 D7 D7   .....ggg ... SSS ..
0x00000070	D7 5A 5A 5A 6E 6E 6E CD CD CD 7A 7A 7A 9E 9E 9E   .ZZZnnn ... zzz ...
0x00000080	4B 4B 4B A5 A5 A5 60 60 60 32 32 32 22 22 22 29   KKK ... ``222""")
0x00000090	29 29 91 91 91 86 86 86 45 45 45 18 18 18 0F 0F   )).....EEE.....
0x000000A0	0F 24 24 24 2E 2E 2E 36 36 36 0E 0E 0E 37 BB 29   .\$\$\$ ... 666 ... 7.)
0x000000B0	A5 00 00 17 1A 49 44 41 54 78 9C ED 5D E7 7A E2   .....IDATx.. ].z.
0x000000C0	BC 12 06 51 6C 7A 87 00 A6 27 D9 90 DC FF FD 1D   ... Qlz ... '.....
0x000000D0	AB CF 8C 24 5B 26 90 EC F9 96 F9 B1 4F D6 C6 B2   ... \${6.....0 ...
0x000000E0	AC 57 D3 47 52 AD F6 A4 27 7D 87 D2 76 B3 D9 EB   .W.GR ... ' } .. v ...
0x000000F0	A4 BF DD 8D 7F 98 06 4C D3 E9 7A FC 3C 5E 73 62   .....L .. z.<^sb
0x00000100	E3 DF EE D4 3F 45 3D E6 D2 CB 6F 77 EA DF 22 0F   ....?E= ... ow .. ".
0x00000110	02 EB DF EE D3 BF 45 7F 5C 04 A6 FA 5E 2B DB 3E   .....E.\ ... ^+.>

Em sử dụng lệnh binwalk -W để xem file này, ở cuối em thấy 1 chuỗi như sau:

```
0x00001780 CF 35 7A D2 CF D0 E6 57 42 A3 4F B2 D4 F8 C5 C0 | .5z....WB.O.....|
0x00001790 DC 93 04 B1 6F D6 4F 3E E9 BB 34 FE 2B 52 45 FF | .....o.O> .. 4.+RE.|
0x000017A0 32 6D EF 50 A9 F0 A4 EF 50 F6 0C 0B FD 32 FD 50 | 2m.P....P....2.P|
0x000017B0 99 CA 93 82 D4 FB F8 ED 1E FC EB D4 AD 54 6D FD | .....Tm.|
0x000017C0 A4 07 D0 D3 21 FB 6D 7A 9A 42 4F BA 91 FE 07 4A | .....!mz.BO....J|
0x000017D0 4A 1C 61 CC 1C 73 6B 00 00 00 00 49 45 4E 44 AE | J,a..sk....IEND.|
0x000017E0 42 60 82 77 68 65 72 45 20 53 68 4F 55 6C 64 20 | B`.wherE.SHOULD.|
0x000017F0 6F 6E 45 20 52 65 61 4C 6C 79 20 6C 4F 6F 4B 20 | onE.ReaLly.lOoK.|
0x00001800 66 4F 72 20 74 48 69 73 20 66 6C 61 67 XX XX XX | fOr.tHis.flag ...|
```

Với việc gợi ý trong bài như sau: “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.”

Em có thử search trên google thì tìm ra được đây chính là Baconian Cipher:



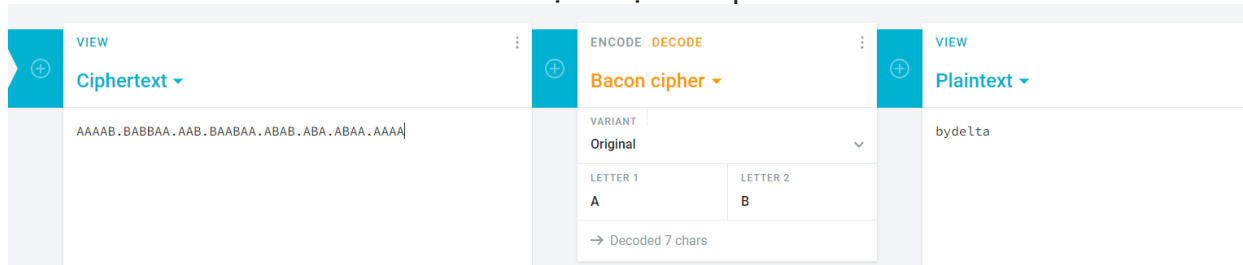
Nếu nghĩ theo hướng Baconian Cipher cho chuỗi này, thì có thể cách ẩn giấu thông điệp ở đây là sử dụng chữ in hoa và chữ thường để đại diện cho "A" và "B" trong Baconian Cipher. Theo cách chữ thường sẽ đại diện cho "A" và chữ hoa sẽ đại diện cho "B".

Với chuỗi vừa này mà em tìm được là: “.whErE.ShOUld.onE.ReALly.lOoK.fOr.tHis.flag”.

Khi chuyển chuỗi sang kí tự như ý tưởng trên thì nó sẽ như sau:

“AAAAB.BABBAA.AAB.BAABAA.ABAB.ABA.ABAA.AAAA”.

Tiến hành decode chuỗi trên thì em nhận được kết quả như sau:

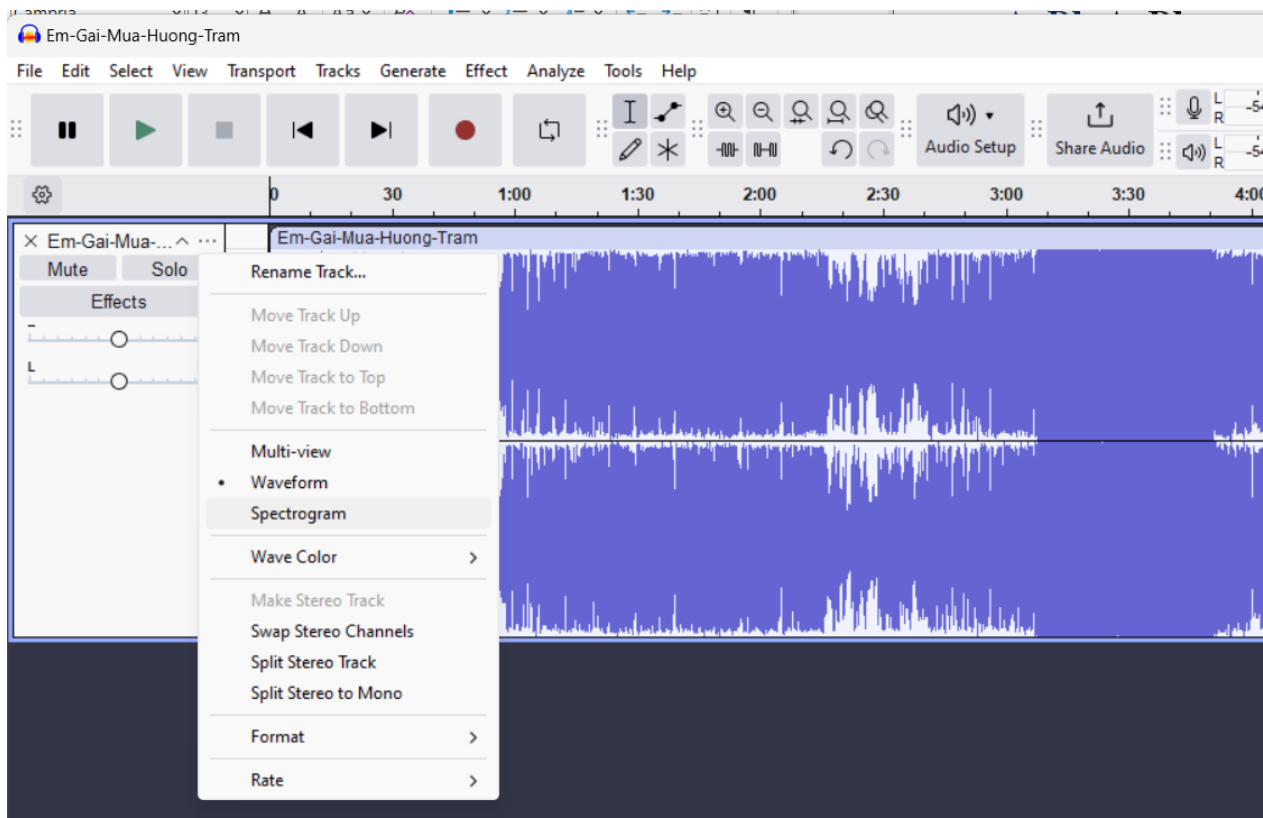


Vậy chuỗi cần tìm ở đây là “bydelta”.

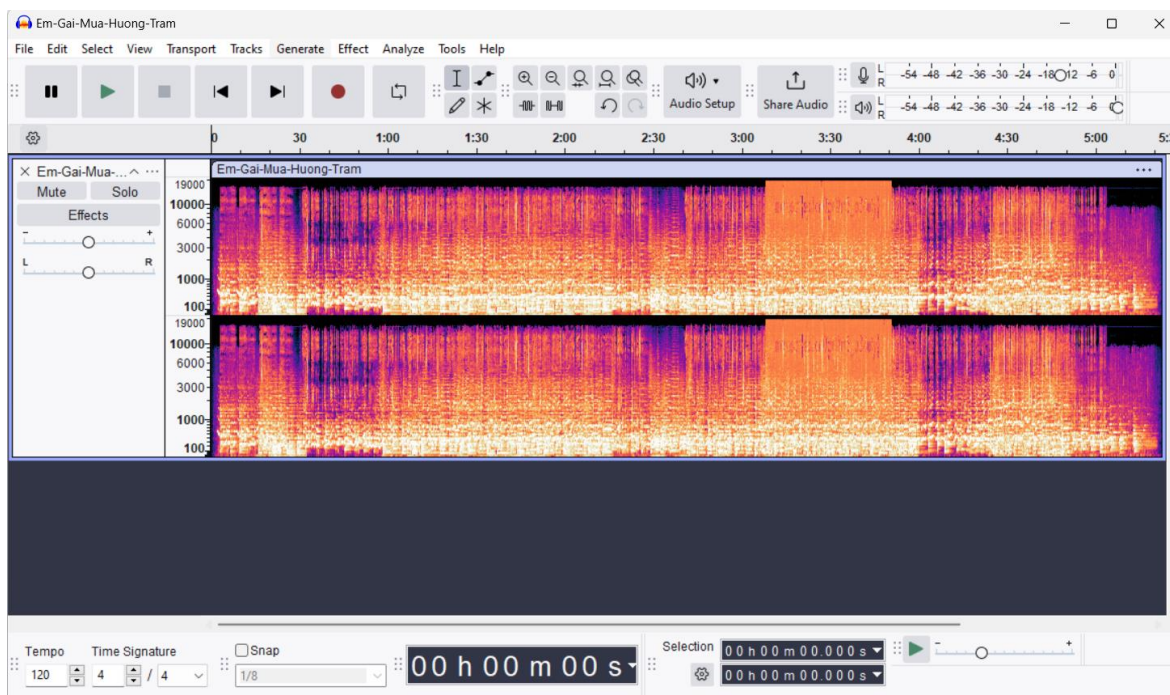
### Kịch bản 07:

Sử dụng phần mềm chỉnh sửa âm thanh **Audacity**

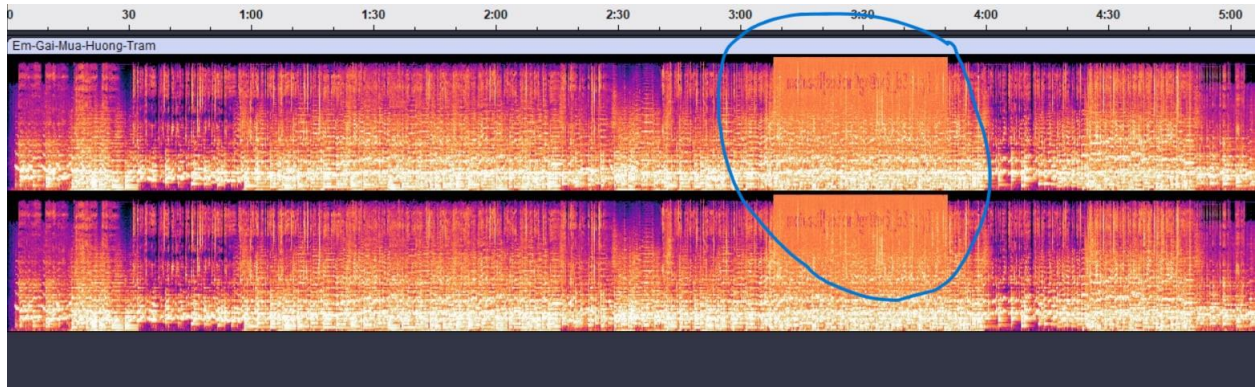
Mở file và chọn định dạng **Spectrogram**



Sau khi đổi định dạng, ta tìm được một đoạn âm thanh khả nghi

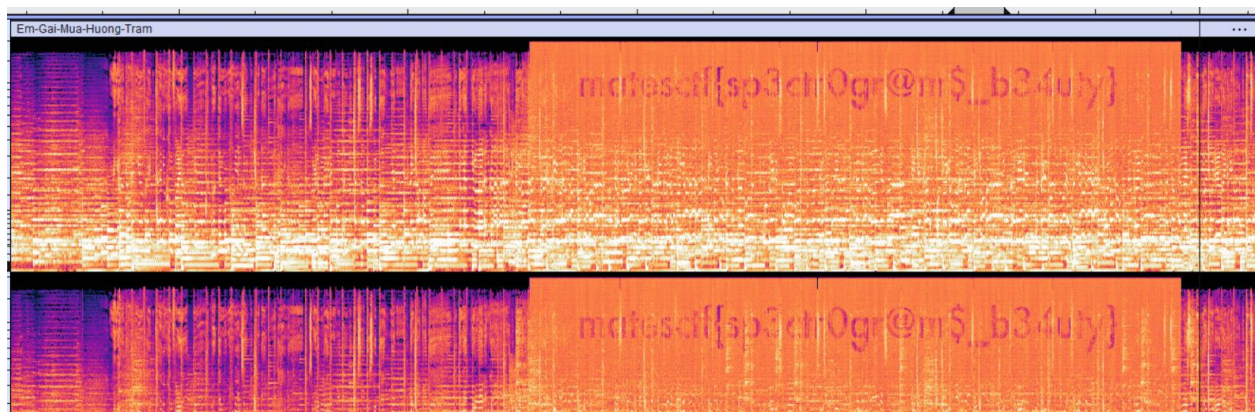






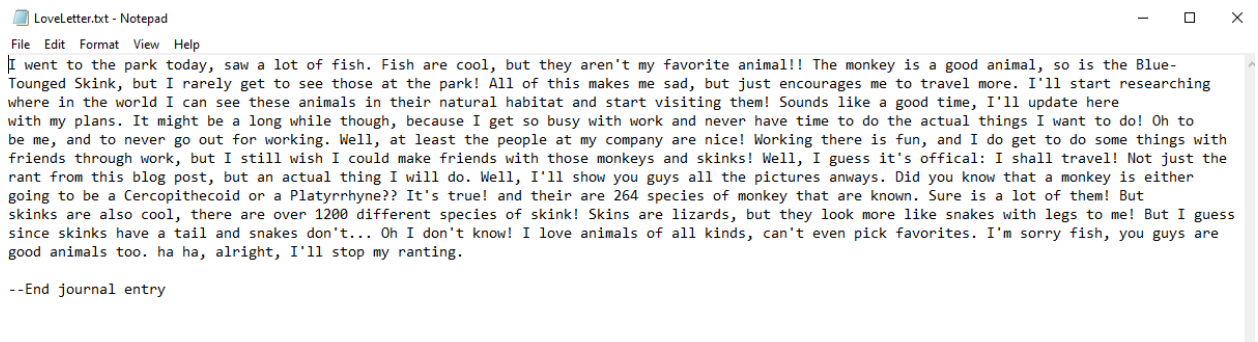
Phóng to đoạn âm thanh này, ta tìm được flag

`matesctf{sp3ctr0gr@m$_b34uty}`

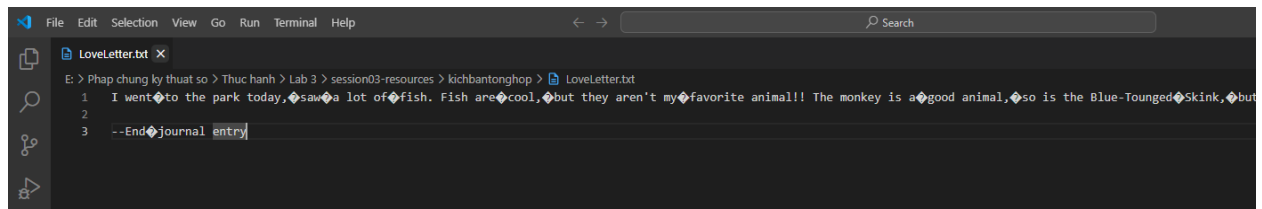


## Kịch bản 8:

Khi mở file txt này bằng notepad thì em thấy nó khá bình thường, không có gì đặc biệt cả:



Nên em sẽ sử dụng vscode, vì nó có hỗ trợ hiển thị các ký tự đặc biệt để xem xem sao:



Em thấy được có xuất hiện các ký tự với dấu ?, trùng hợp là nó xuất hiện ngay chỗ khoảng trắng cách ký tự các chữ.

Không hẳn khoảng trắng nào cũng có, sẽ có khoảng trắng có và khoảng trắng không có ký tự dấu ? này.

Khả năng em nghĩ rằng ở đây khoảng trắng được dùng để chèn vào thông điệp ẩn, với việc bit có 2 số là 0 và 1, ở khoảng trắng trong file txt này cũng có chỗ có và chỗ không có ký tự đặc biệt.

Thế nên sẽ có 2 trường hợp xảy ra, trường hợp 1 là khoảng trắng bình thường sẽ đại diện cho 0 và khoảng trắng đặc biệt sẽ đại diện cho 1. Trường hợp 2 thì sẽ ngược lại với trường hợp 1.

Em sẽ tiến hành thử trường hợp 1 trước:



### Đoạn Mã Python Để Giải Mã

Dưới đây là đoạn mã Python dựa trên giải pháp:

```
python Sao chép mã  
  
import binascii  
  
# Mở và đọc nội dung của tệp LoveLetter.txt  
with open("LoveLetter.txt", "r", encoding="utf-8") as file:  
    data = file.read()  
  
# Khởi tạo chuỗi kết quả nhị phân  
binary_string = ""  
  
# Duyệt qua từng ký tự trong nội dung  
for char in data:  
    if ord(char) == 32:      # Nếu ký tự là khoảng trắng thông thường  
        binary_string += "0"  
    elif ord(char) == 160:   # Nếu ký tự là khoảng trắng không ngắt dòng (non-breaking space)  
        binary_string += "1"  
  
# Chuyển đổi chuỗi nhị phân thành hex và sau đó giải mã thành ASCII  
flag = binascii.unhexlify('%x' % int(binary_string, 2)).decode('utf-8')  
print("Flag:", flag)
```

Em sử dụng GPT để viết và chạy code theo ý tưởng trường hợp 1.

Kết quả nhận được như sau:



LoveLetter.txt

Tài liệu



đây là file cần giải, hãy chạy trực tiếp file này với code bạn vừa đưa cho tôi



Lỗi này xảy ra do ký tự không hợp lệ trong tệp khi sử dụng mã hóa UTF-8. Để khắc phục, tôi sẽ thử mở tệp bằng cách sử dụng mã hóa ISO-8859-1 (Latin-1), hỗ trợ các ký tự mở rộng như ký tự không ngắt dòng ( `0xA0` ). Tôi sẽ thực hiện lại quá trình giải mã. [\[-\]](#)

Flag đã được giải mã thành công:



Sao chép mã

FLAG-3b6f70fcf070009561f5276fe98fc9c6

Bạn có thể sử dụng flag này để hoàn thành thử thách CTF của mình. [\[-\]](#)



Với việc em sử dụng flag trên để submit thử coi trường hợp 1 của em đúng không thì em nhận được kết quả đã thành công:

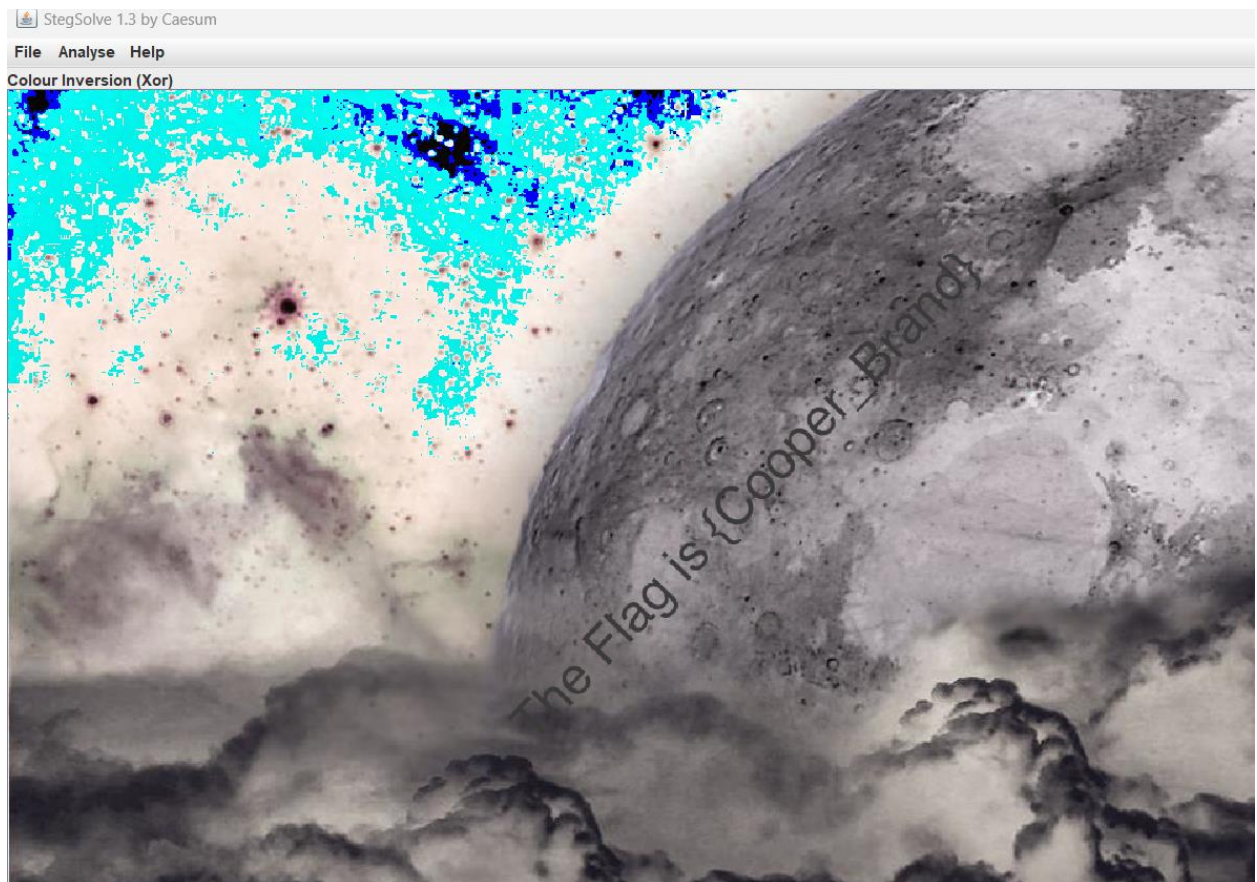
#### List of all challenges you have completed

Validation date	Challenge	Number of points
2024-11-13 07:40:15	Love Letter	3

Vậy kết quả cần tìm ở đây chính là “FLAG-3b6f70fcf070009561f5276fe98fc9c6”.

#### Kịch bản 09:

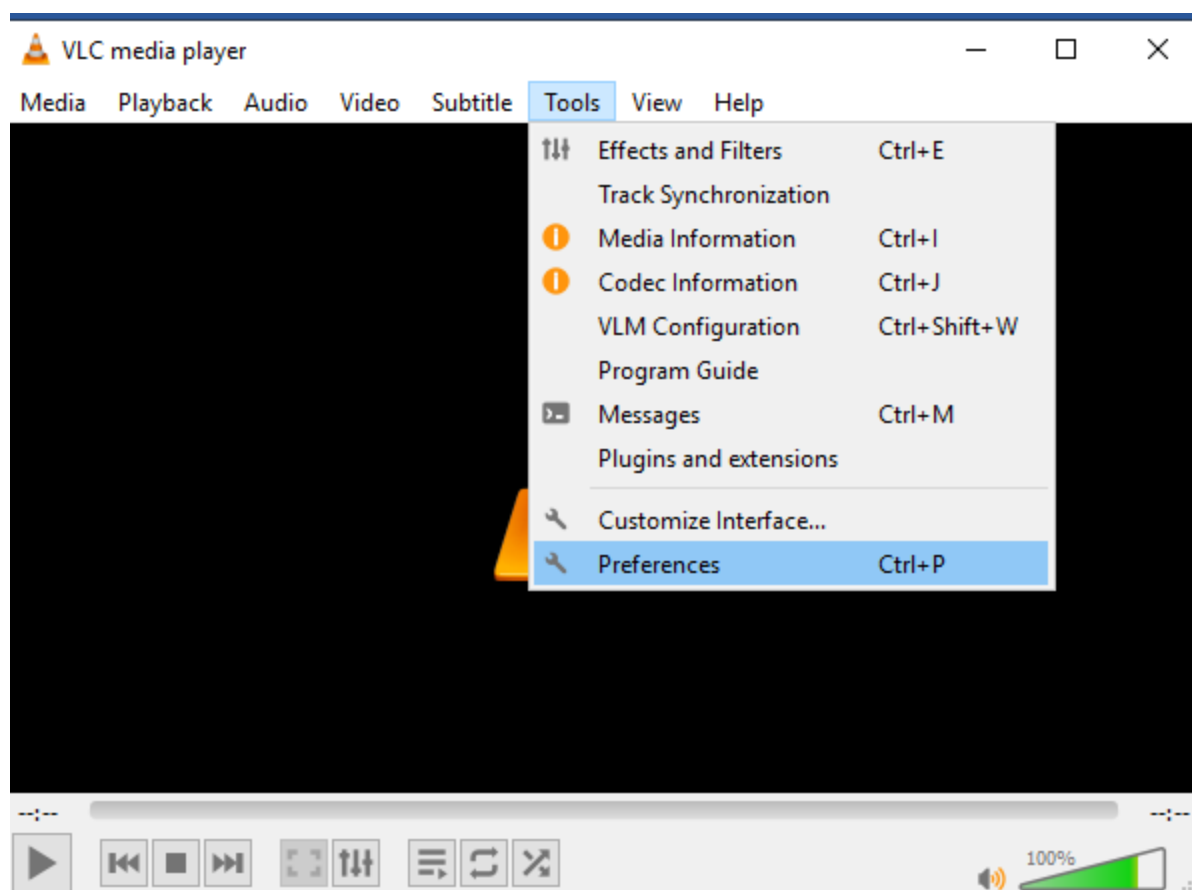
Sử dụng công cụ **StegSolve**, điều chỉnh ảnh đến **Colour Inversion (Xor)**, ta tìm được flag **{Cooper\_Brand}**



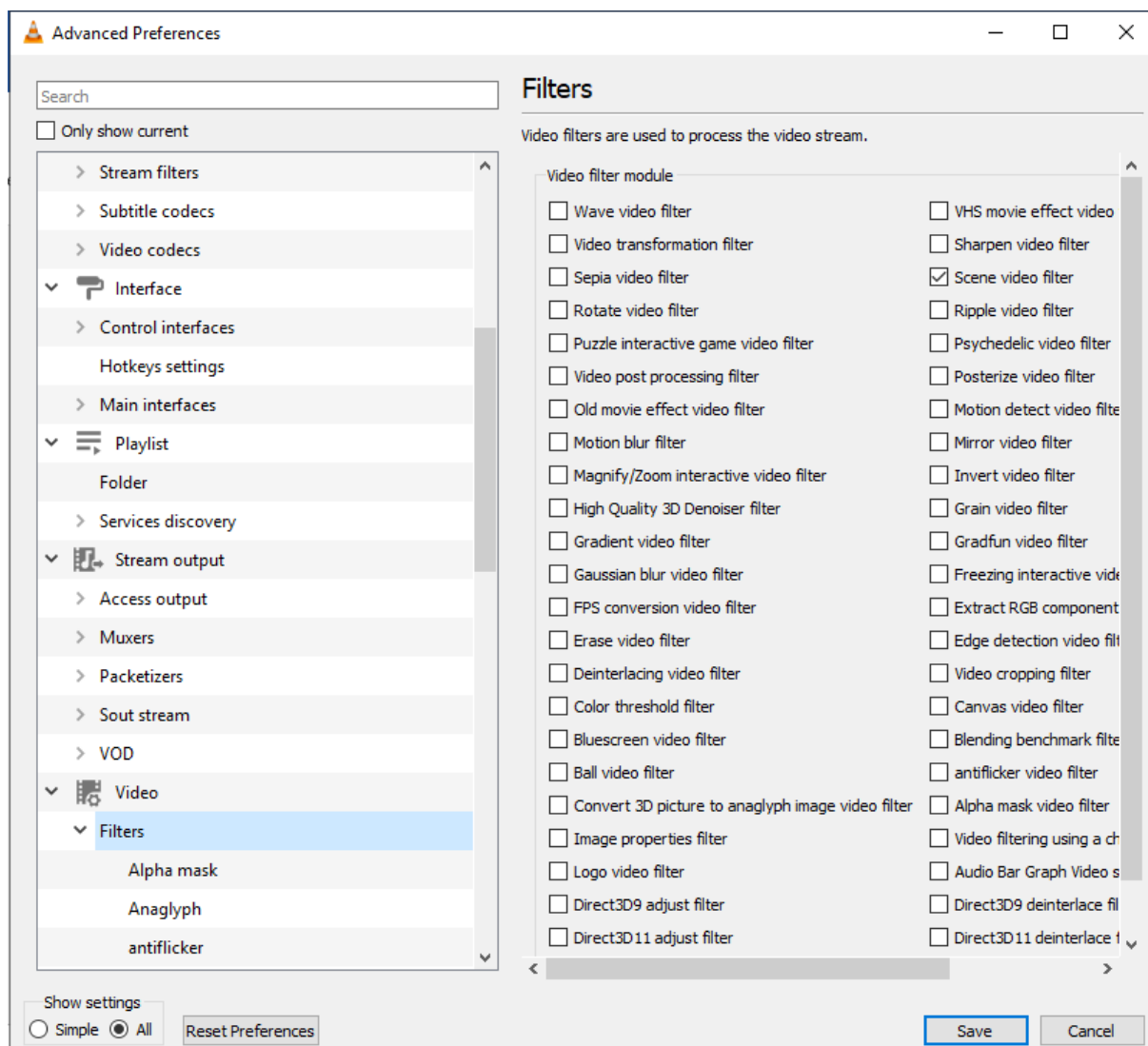
#### Kịch Bản 10:

Với chall ctf này, ta sẽ sử dụng tính năng auto chụp các frame của vlc media.

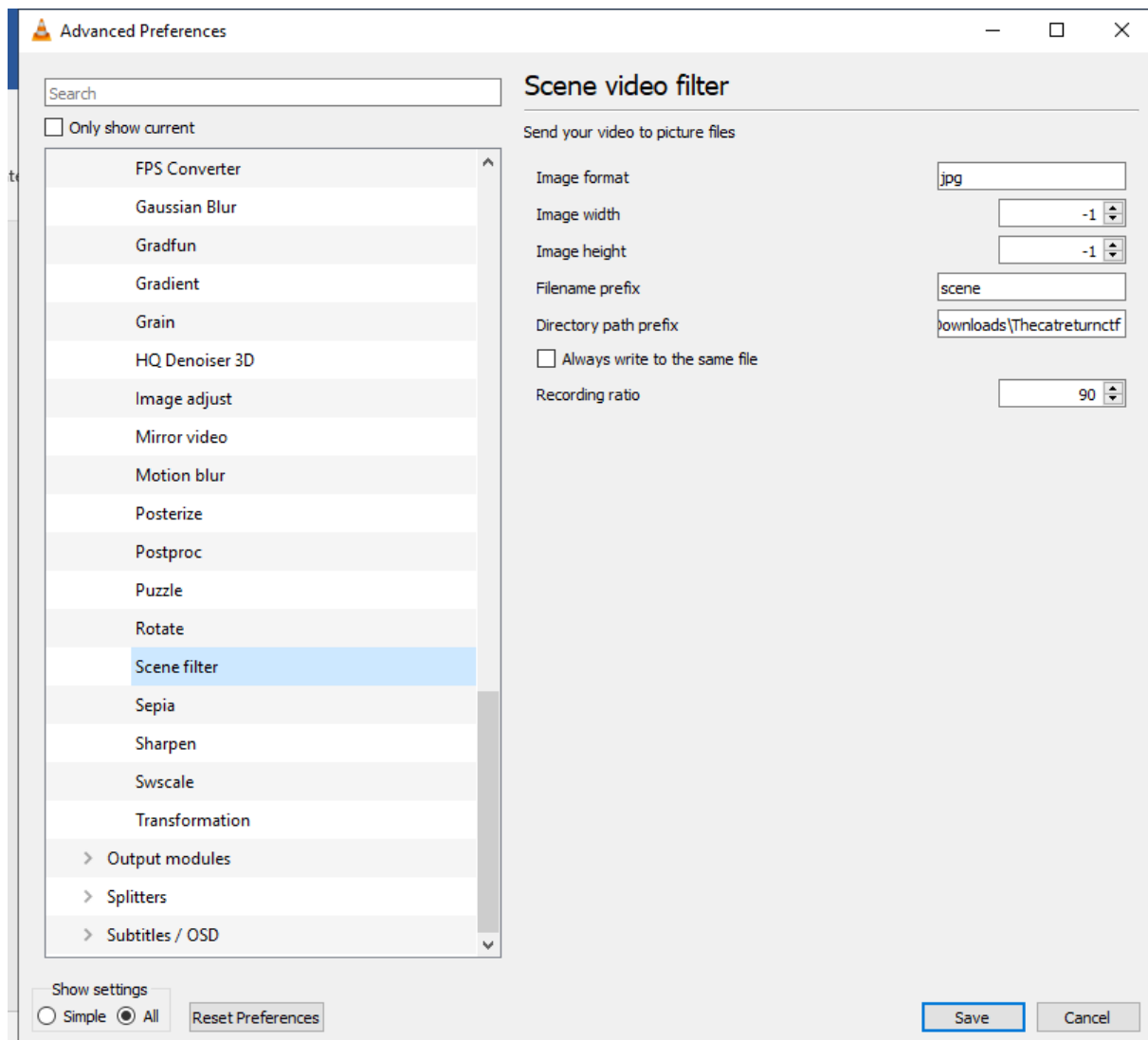
Thực hiện vào và preference trong mục tools và chọn all



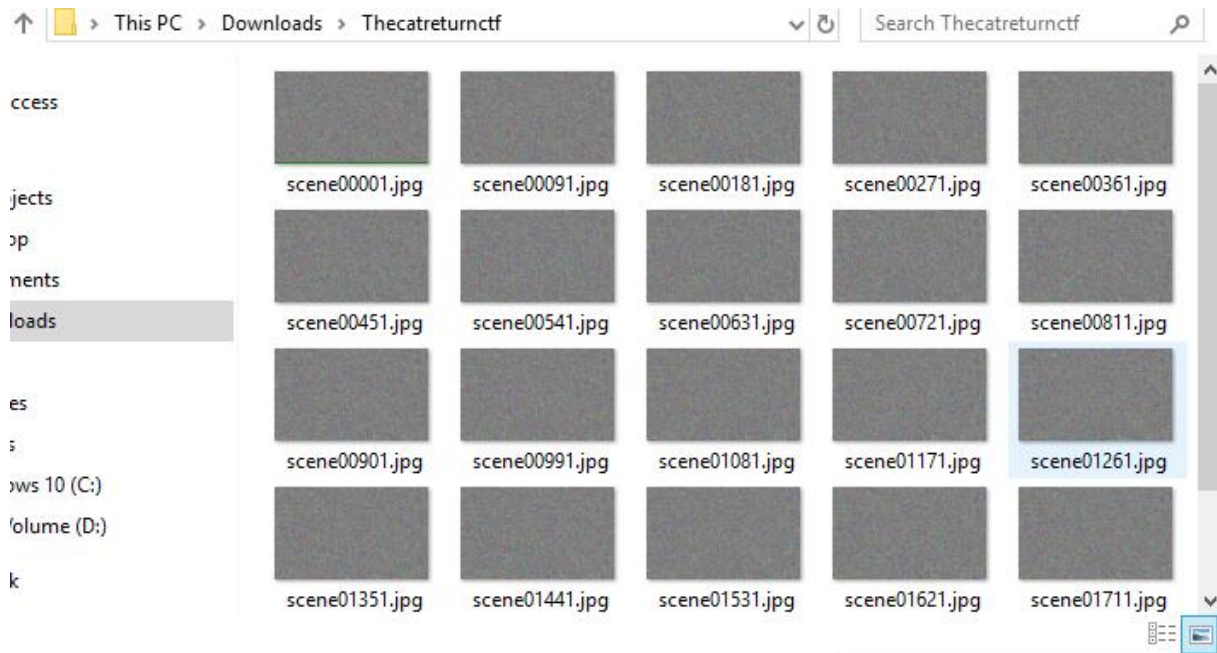
Ta chọn scene video filter trong mục filters



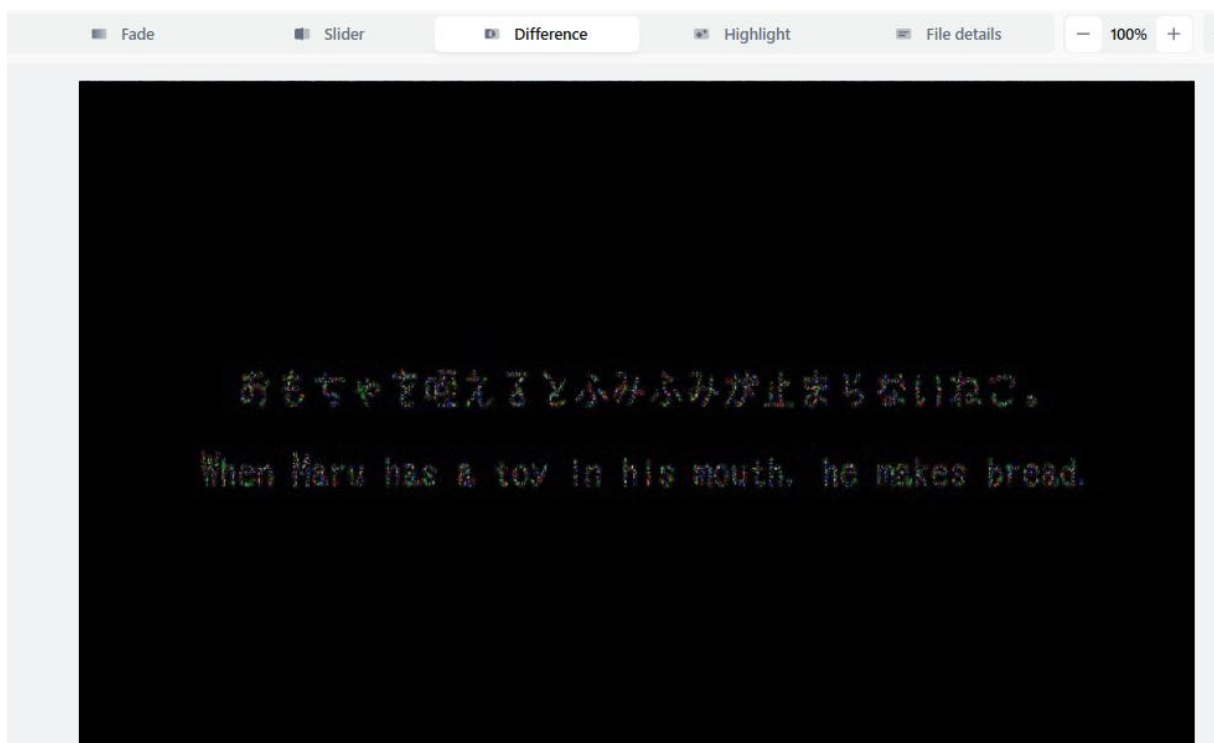
Trong mục con sence filter của mục filters đặt định dạng ảnh lưu là jpg và recording ratio là 90. Directory path prefix sẽ là folder lưu các frame đã được chụp lại.



Sau khi Save, tắt VLC và bật lại dưới quyền Administrator để vlc được quyền lưu file vào folder. Thực hiện chạy video và được kết quả dưới đây.



Sử dụng trang web diffchecker, ta so sánh các frame với nhau. Với frame 001 và 091, ta được kết quả sau.



Rất may mắn là flag của chall nằm ở diff của frame 001 và frame 191





Ta có được flag:

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

---

*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 thống nhất tại [courses.uit.edu.vn](https://courses.uit.edu.vn).

### Đánh giá:

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

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

**HẾT**