BÁO CÁO THỰC HÀNH

**Môn học: Pháp chứng kỹ thuật số**

**Lab 4: Network Forensic**

*GVHD: Đoàn Minh Trung*

1. **THÔNG TIN CHUNG:**

*(Liệt kê tất cả các thành viên trong nhóm)*

Lớp: NT334.O21.ATTN

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Họ và tên** | **MSSV** | **Email** |
| 1 | Phạm Ngọc Thơ | 21522641 | 21522641@gm.uit.edu.vn |
| 2 | Hà Thị Thu Hiền | 21522056 | 21522056@gm.uit.edu.vn |

1. **NỘI DUNG THỰC HIỆN:[[1]](#footnote-1)**

|  |  |  |
| --- | --- | --- |
| **STT** | **Công việc** | **Kết quả tự đánh giá** |
| 1 | Kịch bản 1a, 1b (đã báo cáo ở lớp) | 100% |
| 2 | Kịch bản 2 | 100% |
| 3 | Kịch bản 3 (đã báo cáo ở lớp) | 100% |
| 4 | Kịch bản 4 | 100% |
| 5 | Kịch bản 5 | 100% |
| 6 | Kịch bản 6 (đã báo cáo ở lớp) | 100% |

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

BÁO CÁO CHI TIẾT

**Kịch bản 02: Điều- tra trên dữ liệu lưu lượng mạng thu được.**

- Tài nguyên: capture-output\_kb02.7z

- Yêu cầu: Thực hiện phân tích các request DNS, các truy cập HTTP đến

các trang web nào. Người dùng đã gửi một số tập tin thông qua một

trang web. Xác định dịch vụ mà người dùng sử dụng để chuyển tập

tin, thông tin người nhận (email, thông điệp lời nhắn, tên file đã gửi).

Trích xuất nội dung các file đã gửi.

*Đáp án:*

* Để xác định user đang truy cập đến những trang web nào,chúng ta sử dụng lệnh sau để sắp xếp lại cho dễ nhìn.

tshark -r capture-output\_kb02.pcap -Y http.request -T fields -e http.request.full\_uri | sort | uniq -c

**A screenshot of a computer

Description automatically generated**

* **Các domain trên là 2 trang web:**
  + [**http://fsend.vn**](http://fsend.vn)
  + [**https://www.fshare.vn/**](https://www.fshare.vn/)
* 2 trang web này dùng để upload file lên
* Dùng Wireshark để xem thông tin các packet có request method là POST trên các URL này. Dùng filter **http.file\_data** :

**A screenshot of a computer

Description automatically generated**

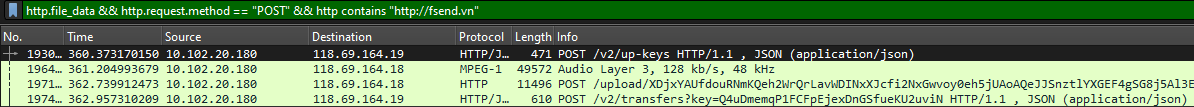
* Chúng ta xem HTTP stream thử có gì

**A screenshot of a computer

Description automatically generated**

* Chúng ta thấy chưa có gì đặc biệt ở POST packet này.
* Thử request POST và URL chứa [**http://fsend.vn**](http://fsend.vn)

http.file\_data && http.request.method == "POST" && http contains “<http://fsend.vn>”

****

* Packet đầu tiên có file được upload là 1 file mp3 và 1 file image

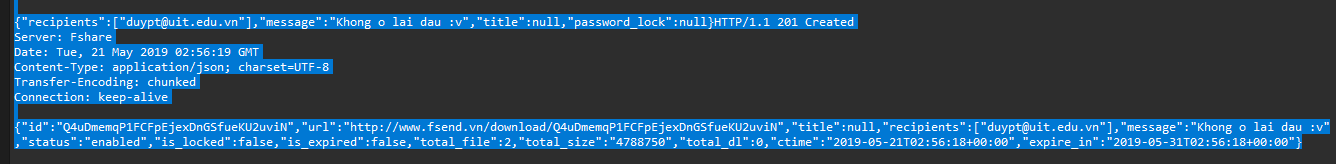
**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

* Người nhận (recipient): **duypt@uit.edu.vn**
* Thông điệp (message): “**Khong o lai dau :v”**

****

* **Trích xuất nội dung file**
* Packet còn lại hiển thị nội dung sau:

**A screenshot of a computer

Description automatically generated**

* Ta đã có nội dung file, chuyển sang view dưới dạng Raw (Show data as) , và tìm chuỗi định dạng file (chữ ký file) hợp lệ [List of file signatures - Wikipedia](https://en.wikipedia.org/wiki/List_of_file_signatures)

**A screenshot of a computer

Description automatically generated**

* Phần bắt đầu là phần bắt đầu từ chuỗi (search với chuỗi chữ ký trên “FFD8FFE000104A46”)

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

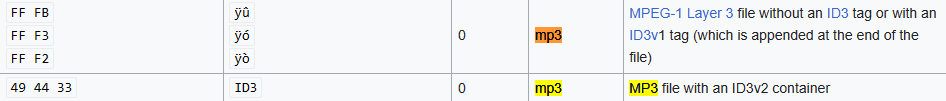
* **Có 2 chuỗi è có 2 ảnh**
* Phần kết thúc file (giới hạn đến phần đỏ, màu xanh là header của request)
* Copy từ đầu đến cuối (như đã phân tích ở trên) vào một file mới trong phần hex data (bên trái) của HxD. Lưu file lại thành image1.jpg và image2.jpg

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

****

**A screenshot of a computer

Description automatically generated**

* Làm tương tự như với ảnh , nhưng điểm bắt đầu nội dung file là 494433 -> Đưa vào HxD -> Save as - > file.mp3

**A screenshot of a video

Description automatically generated**

**Kịch bản 04. Điều tra trên dữ liệu lưu lượng mạng thu được.**

- Tài nguyên: net\_kb04.pcap

- Yêu cầu – Gợi ý: Đây là dữ liệu mạng thu được khi bắt gói tin duyệt

web trong một khọảng thời gian. Tìm flag, biết flag có định dạng

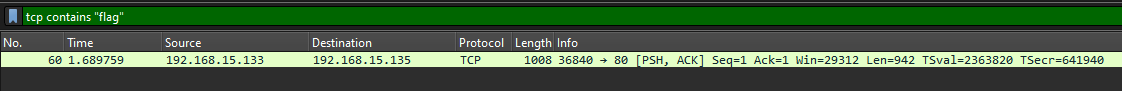
flag{…}

Đáp án:

A screenshot of a computer

Description automatically generated

* Đề bài là kiếm ra được flag{}, nên ta sử dụng filter để lọc xem có gì không



A screenshot of a computer

Description automatically generated

* Copy ra ngoài để dễ phân tích
* import string
* import random
* from base64 import b64encode, b64decode
* FLAG = 'flag{xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx}'
* enc\_ciphers = ['rot13', 'b64e', 'caesar']
* *# dec\_ciphers = ['rot13', 'b64d', 'caesard']*
* def rot13(s):
* \_rot13 = string.maketrans(
* "ABCDEFGHIJKLMabcdefghijklmNOPQRSTUVWXYZnopqrstuvwxyz",
* "NOPQRSTUVWXYZnopqrstuvwxyzABCDEFGHIJKLMabcdefghijklm")
* return string.translate(s, \_rot13)
* def b64e(s):
* return b64encode(s)
* def caesar(plaintext, shift=3):
* alphabet = string.ascii\_lowercase
* shifted\_alphabet = alphabet[shift:] + alphabet[:shift]
* table = string.maketrans(alphabet, shifted\_alphabet)
* return plaintext.translate(table)
* def encode(pt, cnt=50):
* tmp = '2{}'.format(b64encode(pt))
* for cnt in xrange(cnt):
* c = random.choice(enc\_ciphers)
* i = enc\_ciphers.index(c) + 1
* \_tmp = globals()[c](tmp)
* tmp = '{}{}'.format(i, \_tmp)
* return tmp
* if \_\_name\_\_ == '\_\_main\_\_':
* print encode(FLAG, cnt=?)
* Ta thấy nội dung file là một chương trình mã hóa một chuỗi plaintext thành ciphertext. Ciphertext ở phía sau cùng.
* Bây giờ nhiệm vụ của chúng ta là phân tích file encode và tìm cách viết file decode.

A screen shot of a computer code

Description automatically generated

* Ta có thể sử dụng hàm **encode** trong script này để mã hóa một chuỗi bằng cách chọn ngẫu nhiên một trong ba phương pháp mã hóa: ROT13, base64 encoding hoặc Caesar cipher. Mỗi lần ta gọi **encode**, nó sẽ thêm một bước mã hóa mới và sau đó trả về chuỗi đã mã hóa.
* Để giải mã chuỗi được mã hóa bằng hàm encode trong mã nguồn trên, chúng ta cần thực hiện các bước giải mã theo đúng thứ tự ngược lại với quá trình mã hóa.
* Dưới đây là các bước giải mã tương ứng:
  + Giải mã từ base64:
    - Sử dụng hàm b64decode từ module base64 để giải mã chuỗi base64.
  + Giải mã các phép biến đổi:
    - Đọc từng ký tự trong chuỗi giải mã và xác định loại phép biến đổi bằng cách đọc ký tự đầu tiên của từng khối.
    - Sử dụng hàm tương ứng (rot13, caesar) để giải mã từng khối dựa trên loại phép biến đổi đã xác định.
* import string
* import random
* from base64 import b64encode, b64decode
* FLAG = open("ciphertext.txt").read()
* dec\_ciphers = ['rot13', 'b64d', 'caesard']
* def rot13(s):
* rot13 = string.maketrans(
* "ABCDEFGHIJKLMabcdefghijklmNOPQRSTUVWXYZnopqrstuvwxyz",
* "NOPQRSTUVWXYZnopqrstuvwxyzABCDEFGHIJKLMabcdefghijklm")
* return string.translate(s, \_rot13)
* def b64d(s):
* return b64decode(s)
* def caesar(plaintext, shift=3):
* alphabet = string.ascii\_lowercase
* shifted\_alphabet = alphabet[shift:] + alphabet[:shift]
* table = string.maketrans(alphabet, shifted\_alphabet)
* return plaintext.translate(table)
* def caesard(ciphertext, shift=3):
* return caesar(ciphertext, shift=-shift)
* def decode(ct):
* while True:
* try:
* i = int(ct[0]) - 1
* except:
* print(ct)
* exit(0)
* ct = ct[1:]
* c = dec\_ciphers[i]
* \_ct = globals()[c](ct)
* ct = \_ct
* if \_\_name\_\_ == '\_\_main\_\_':
* decode(FLAG)

Flag: **flag{li0ns\_and\_tig3rs\_4nd\_b34rs\_0h\_mi}**

**Kịch bản 05. Điều tra trên dữ liệu lưu lượng mạng thu được.**

- Tài nguyên thực hiện: kb05.gz

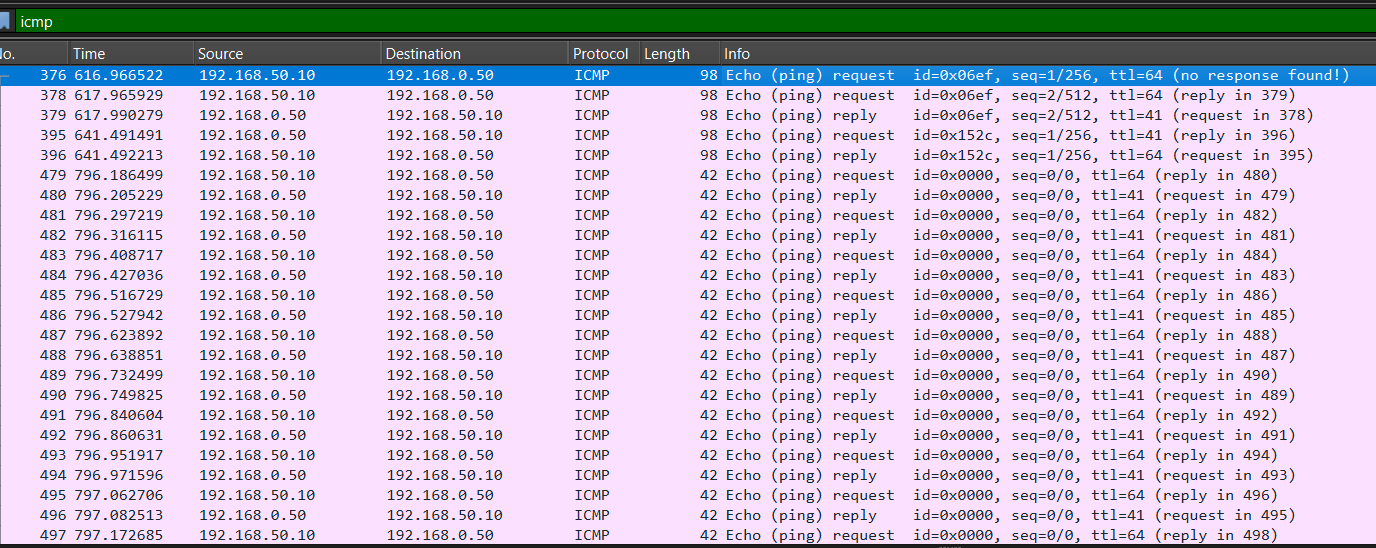
- Yêu cầu – Gợi ý: Xác định các kết nối trọng dữ liệu thu được. Chú ý các

gói ICMP, trường giá trị Identifiers của các gói để tìm flag. Flag có định

dạng bắt đầu bằng chuỗi “S3”, với tổng chiều dài là 11 kí tự

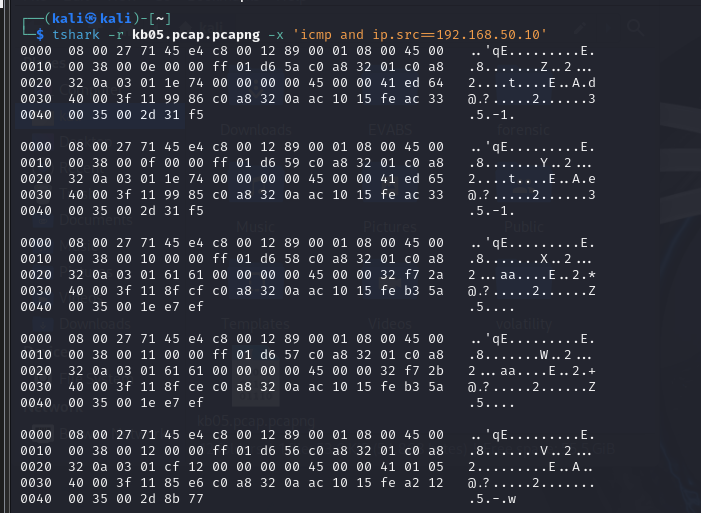
Đáp án:

* Đề có gợi ý tập trung vào ICMP, nên em sẽ filter ICMP trước:

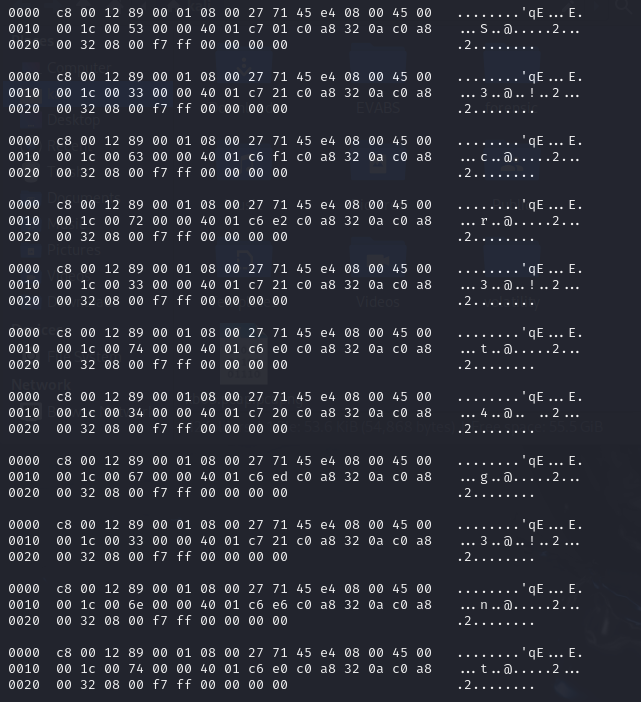


Ở những dòng đầu tiên trong hình trên, quan sát sự thay đổi của id và seq, dường như có một thông điệp đang được trao đổi.

* Dùng tshark để trích xuất thông tin:

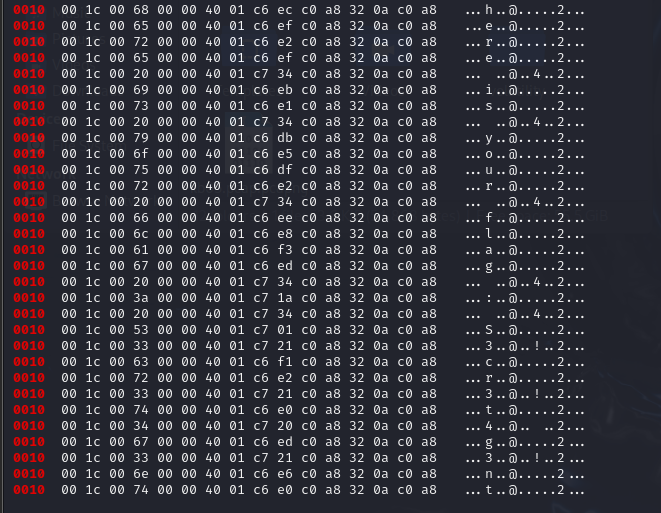


Kết quả hiển thị khá dài nhưng khi quan sát đoạn kết quả cuối, em tìm thấy được những kí tự bắt đầu bằng S3 – hint về flag đề bài cung cấp:



* Để chắc chắn, em sẽ dùng tshark để trích xuất các giá trị tại offset *0010*:

*tshark -r kb05.pcap.pcapng -x 'icmp and ip.src==192.168.50.10' | grep 0010*



Flag: **S3cr3t4g3nt**

---

***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.

**Đánh giá**:

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

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

**HẾT**

1. Ghi nội dung công việc, các kịch bản trong bài Thực hành [↑](#footnote-ref-1)