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 |  |
| 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.**

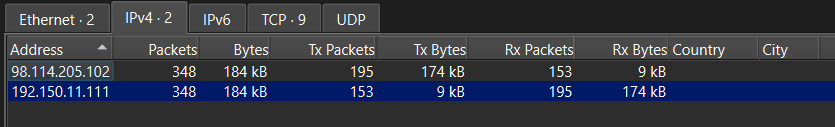
BÁO CÁO CHI TIẾT

**Kịch bản 01a. Thực hiện phân tích tập tin dữ liệu mạng:**

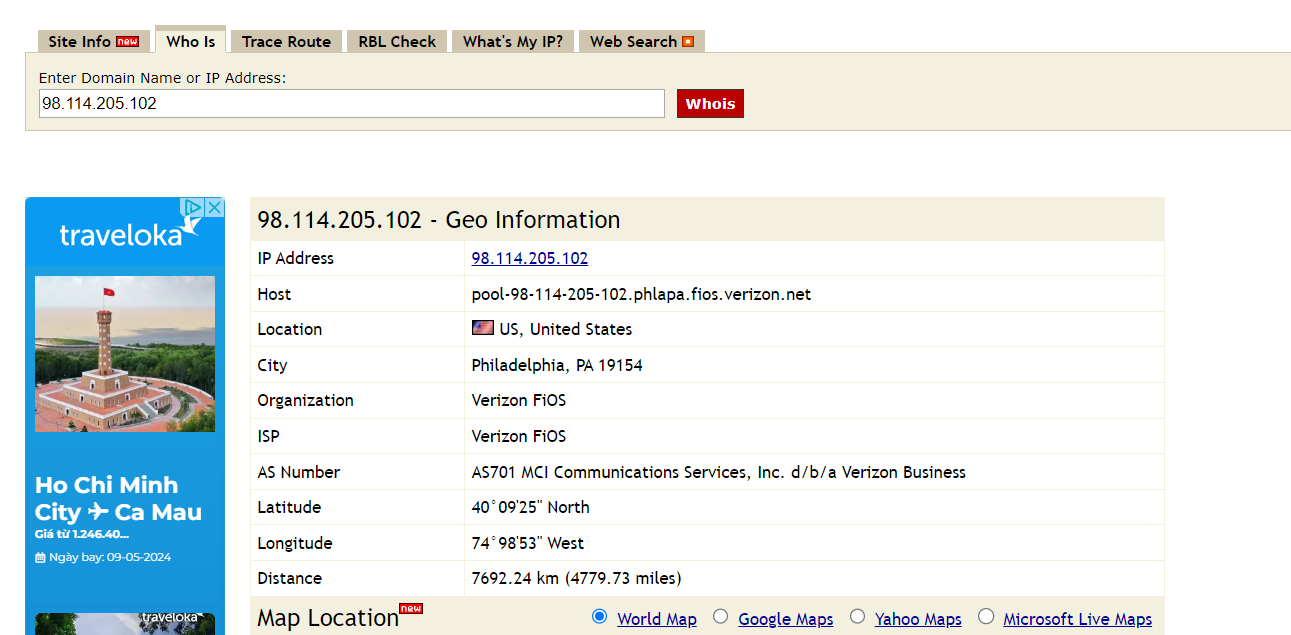
* Mô tả: Một máy tính trọng mạng nội bộ bị nghi ngờ tấn công từ bên ngoài, nhân viên quản trị mạng dùng những công cụ chuyên dụng bắt các kết nối đến máy nạn nhân trọng thời gian diễn ra cuộc tấn công. Sau đó lưu lượng mạng được trích xuất tọàn bộ nội dung trọng tập tin pcap.
* Tài nguyên thực hiện: traffic\_kb01\_a.pcap
* Yêu cầu: Thực hiện phân tích tập tin dump từ dữ liệu mạng để tìm nguồn gốc và nguyên nhân vụ tấn công để có giải pháp khắc phục

Đáp án:

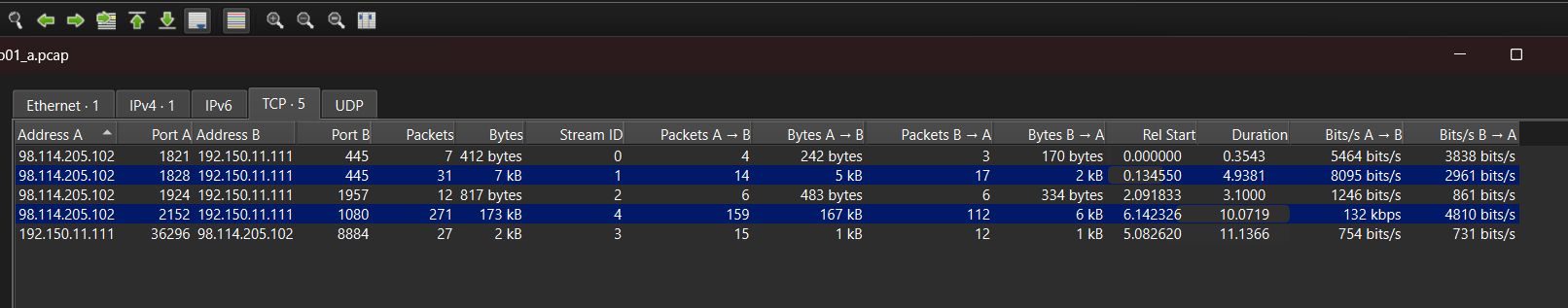
* Đầu tiên, em kiểm tra tại static/endpoints:

Có 2 địa chỉ IP, do 192.168.11.111 là IP private nên nó là IP của nạn nhân, IP còn lại là của attacker.

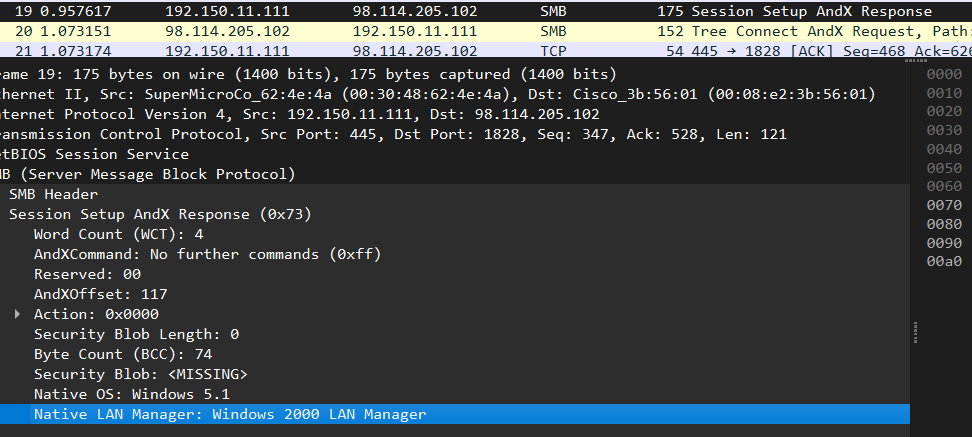
* Thực hiện tìm kiếm trên internet về địa chỉ IP 98.114.205.102 thì tra được vị trí của kẻ tấn công là ở US:



* Kiểm tra TCP session, có 5 session qua 4 ports khác nhau:



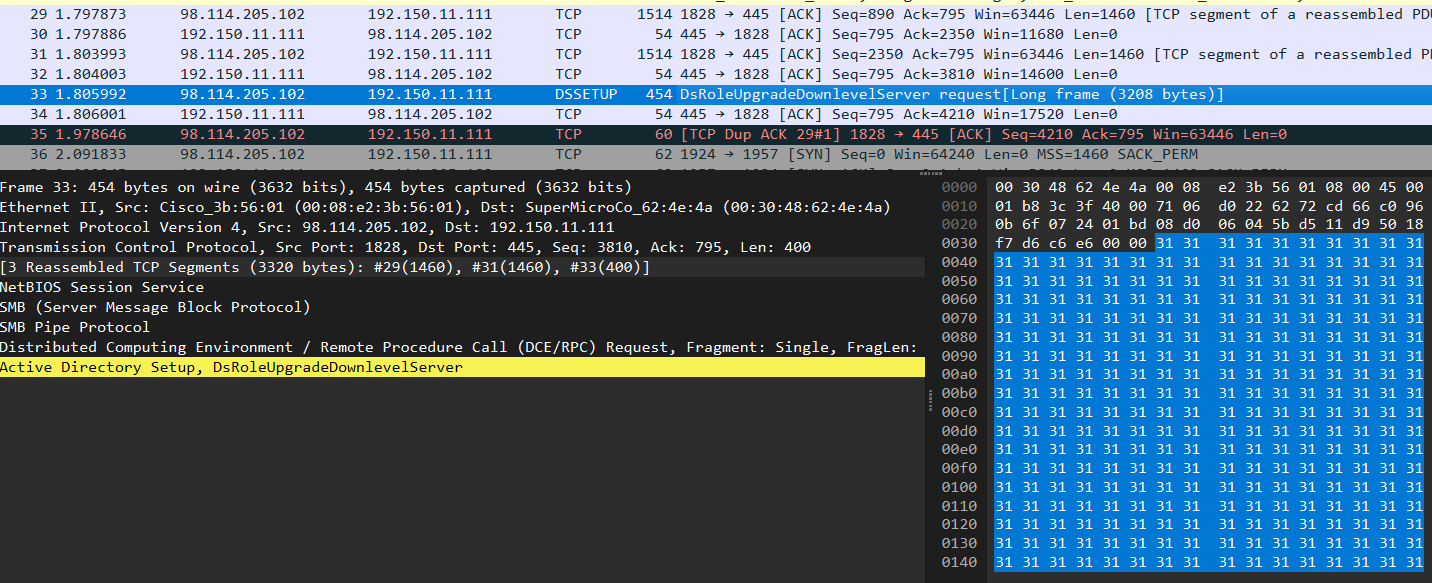
* Port 445 chạy dịch vụ SMB. Máy nạn nhân sử dụng hệ điều hành Windows 2000, thông qua giao thức SMB đã tiết lộ điều này. SMB (Server Message Block) là một giao thức mạng dùng để chia sẻ tệp tin và thiết bị trong một mạng LAN, tuy nhiên không đáng tin cậy.



* Attacker đã thiết lập kết nối IPC và gửi request đến dịch vụ Isarpc:

Attacker đã gửi kết nối với giá trị username và password đều rỗng, còn được biết đến với kiểu tấn công Null Session.

* Quan sát log ghi lại được trong file pcap:

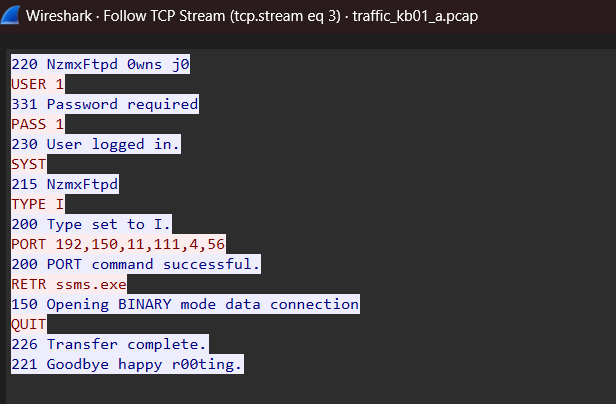


Ở đây, ta thấy attacker có sử dụng hàm DsRoleUpgradeDownlevelServer() để thực hiện tấn công buffer overflow thông qua việc chèn shellcode.

* Tiếp tục quan sát các stream:

Ở đây, attacker đang yêu cầu nạn nhân mở port 8884 và đăng nhập với user là 1, password là 1, sau đó tải tập tin ssms.exe thông qua giao thức FTP và thực thi nó.

Ở stream cuối cùng, máy nạn nhân đã thực hiện tải xuống file như attacker yêu cầu:



Dường như là file exe đã được thực thi sau khi file được tải xong:

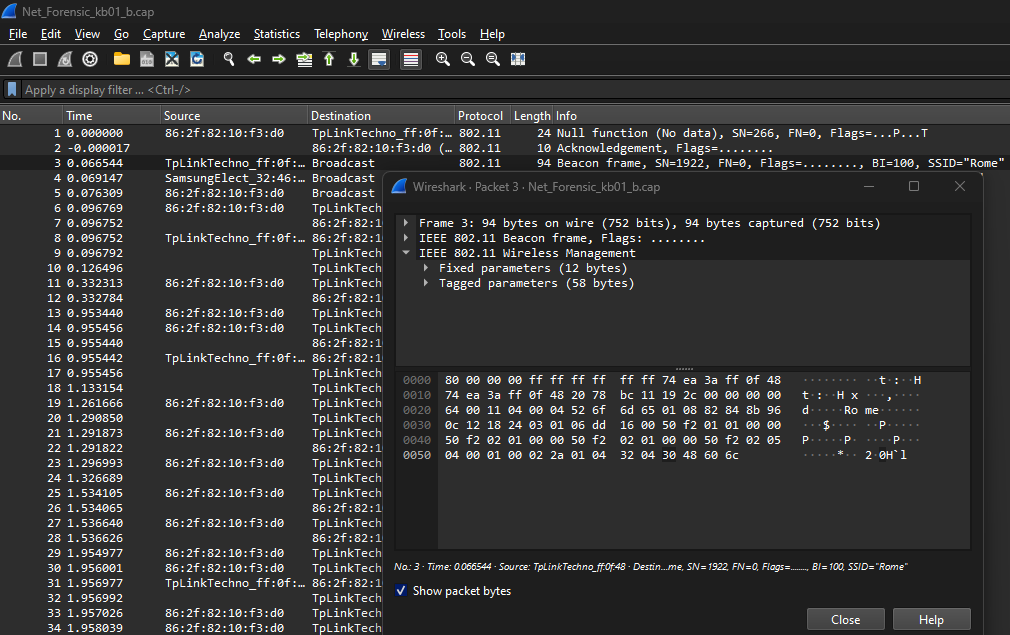


**Kịch bản 01-b. Thực hiện phân tích tập tin dữ liệu mạng thu được.**

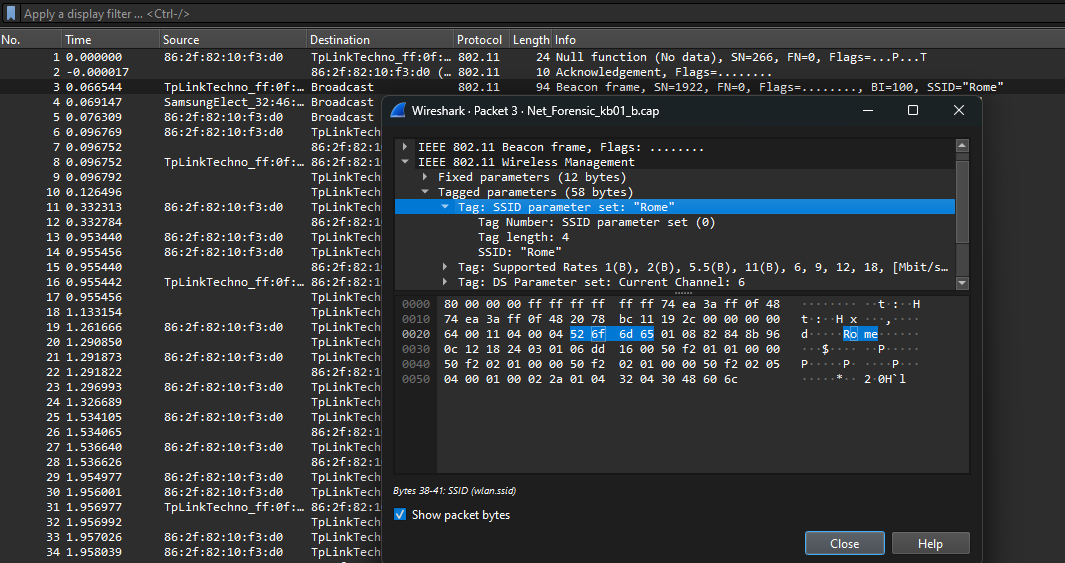
* Mô tả: Tập tin pcap được chọ là dữ liệu mạng thu được từ một mạng không dây.
* Tài nguyên thực hiện: Network\_Forensic\_kb01\_b.pcap
* Yêu cầu: Thực hiện phân tích tập tin dump từ dữ liệu mạng để tìm SSID, mật khẩu giải mã strếam TCP, sau đó phân tích strếam đã giải mã để tìm flag.

Đáp án: Flag: be02d2a396482969e39d92b6e440f5e3

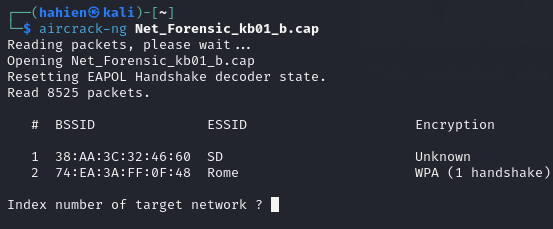
* Mở tập tin pcap bằng Wirếshark, quan sát chuẩn kết nối không dây đang sử dụng



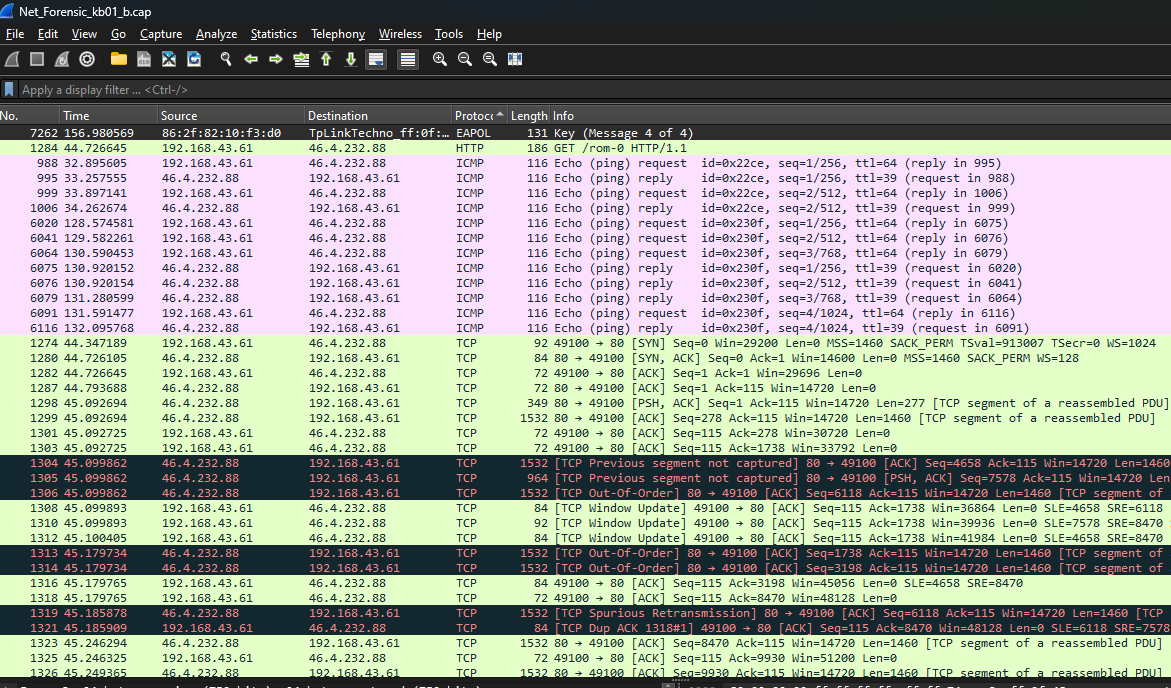
* Xem SSID

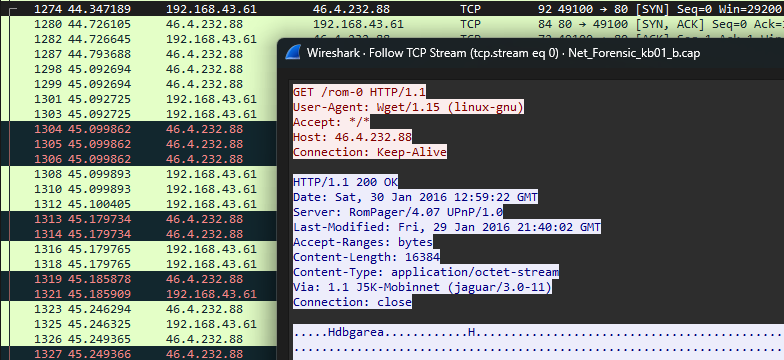


* Tiếp thếọ, chúng ta sẽ xếm việc truyền dữ liệu mạng không dây này có được mã hóa hay không. Có thể sử dụng công cụ aircrack-ng.

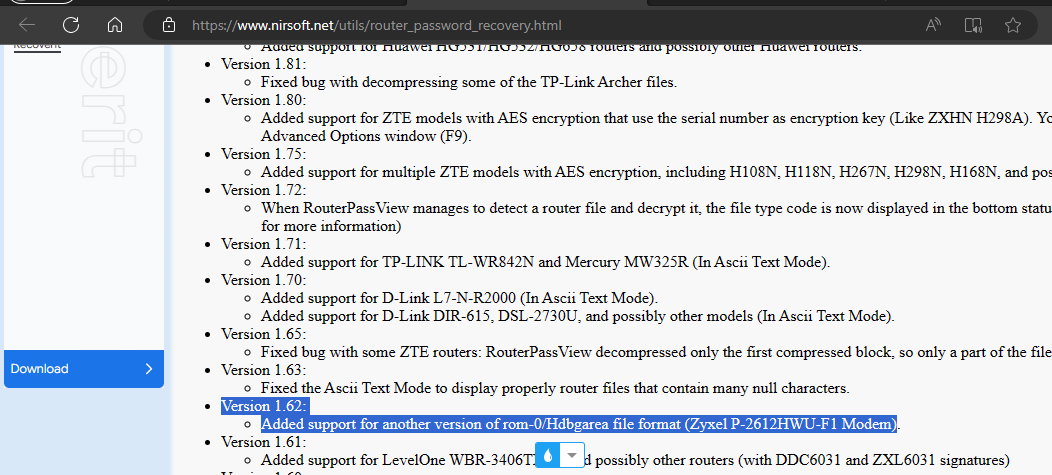


* Quay trở lại kịch bản, nhận thấy có một số gói tin TCP và ICMP. Sử dụng chức năng Fọllọw TCP stream để xếm thêm thông tin. Chú ý đến **Hdbgarea** trong file dump ra từ TCP stream. Từ khóa này này liên quan đến việc cấu hình các router trên trang web Nirsoft: (RouterPassView v1.81 - Recover lost password from router backup file) https://www.nirsoft.net/utils/router\_password\_recovery.html

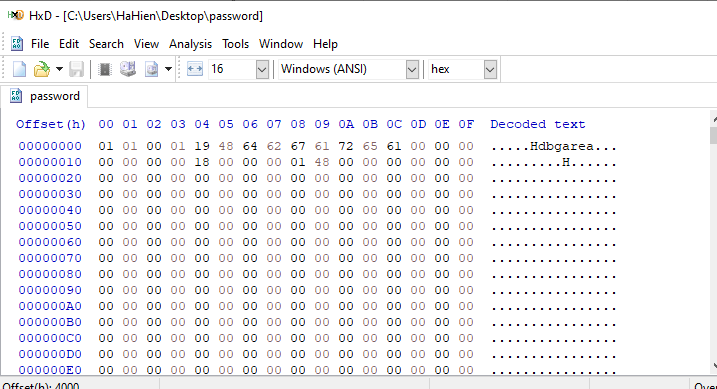


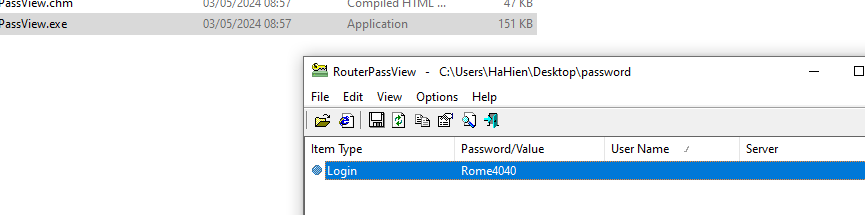


* Lưu nội dung TCP stream này sau khi đã xóa tất cả các header thành file nhị phân.
* Truy cập link trên ta xếm thấy thông tin như sau:

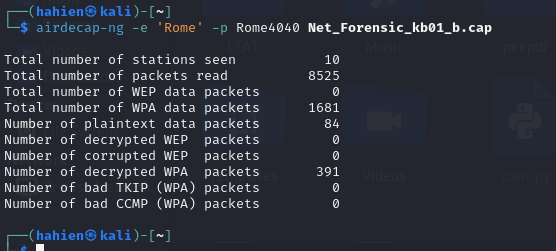


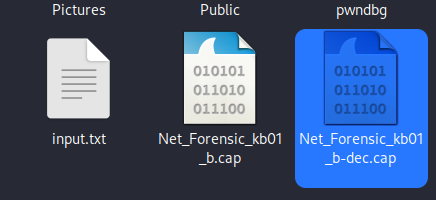
* Dọ đó, tiếp tục sử dụng chương trình phục hồi/xem mật khẩu của router bằng RouterPassView trên dữ liệu TCP stream thu được ở bước trên. Ghi nhận mật khẩu thu được.

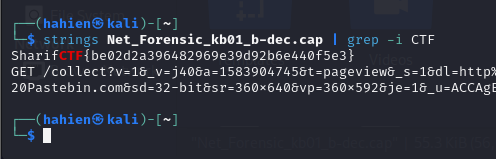




* Mật khẩu ở đây là Rome4040. Sau đó dùng aircrack-ng để giải mã các packet với mật khẩu đã xác định với tùy chọn -e là ESSID và -p là password đã tìm







* Rõ hơn thì chúng ta hãy mở wireshark và xem gói tin có dấu hiệu của flag rồi decode sẽ ra được flag.

**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

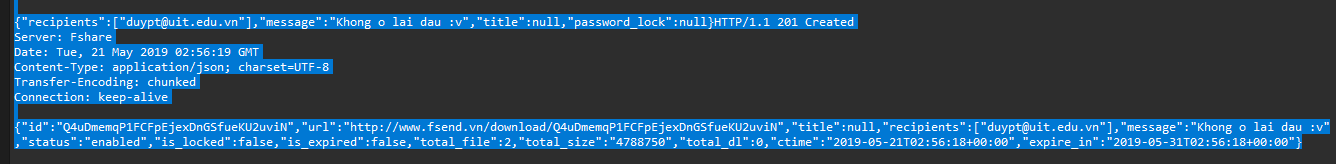
**A screenshot of a computer

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**

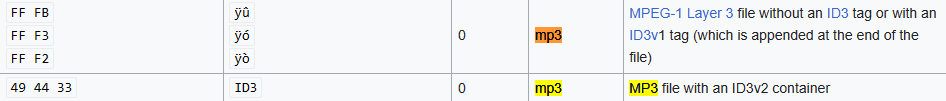
* **Có 2 chuỗi 🡺 có 2 ảnh**
* Phần kết thúc file (giới hạn đến phần đỏ, màu xanh là header của request)
* Copy từ đầu đến cuối (như đã phân tích ở trên) vào một file mới trong phần hex data (bên trái) của HxD. Lưu file lại thành image1.jpg và image2.jpg

**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 03. Điều tra trên dữ liệu lưu lượng mạng thu được.**

- Tài nguyên: kb03\_evidence.pcap

- Mô tả: Công ty Anarchy-R-Us, Inc. chọ rằng một trong những nhân

viên của họ, Ann Dếrcọvếr, là một gián điệp thương mại làm việc chọ

công ty đối thủ vì nhân viên này đã từng xâm nhập vàọ máy chủ chứa

dữ liệu mật của công ty. Nhân viên an ninh của công ty nghi ngờ rằng

Ann đã trộm công thức bí mật của công ty.

Nhân viên an ninh mạng đã thếọ dõi Ann một thời gian nhưng chưa

phát hiên được gì. Hôm nay, có một laptọp lạ đã kết nối vàọ mạng

wireless của công ty. Máy tính của Ann (IP: 192.168.1.158) đã gởi một

số tin nhắn tới máy tính đó, laptọp lạ ngắt kết nối với mạng wirếlếss

ngay sau đó. Dữ liệu mạng của máy của phiên kết nối đã bị an ninh

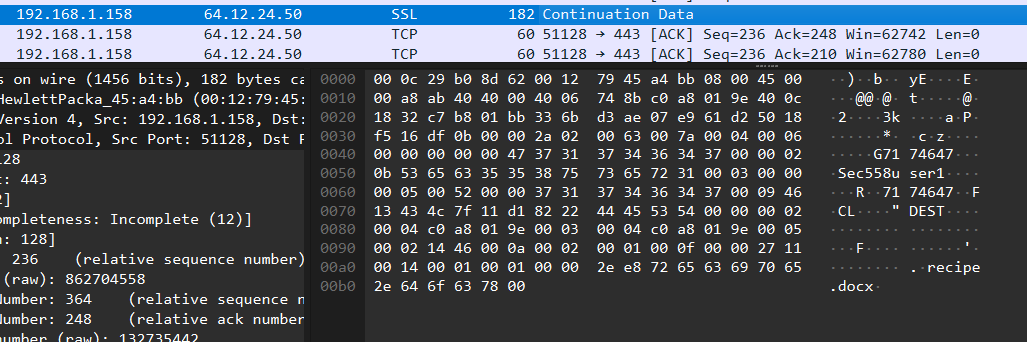
mạng công ty lưu lại. Hãy giúp công ty điều tra xếm Ann có phải là gián

điệp hay không, và công thức bí mật của công ty đã bị đánh cắp hay không?

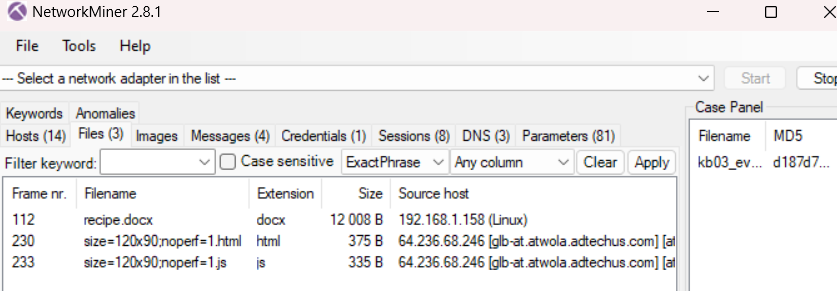
* Đầu tiên, dùng Wireshark để xác định các địa chỉ IP mà máy tính của Ann kết nối tới: Có 4 IP khác nhau:
* Xem nội dung tin nhắn mà Ann đã gửi:

Một số tin nhắn giữa 2 người còn được tiết lộ qua statistics/conversation:

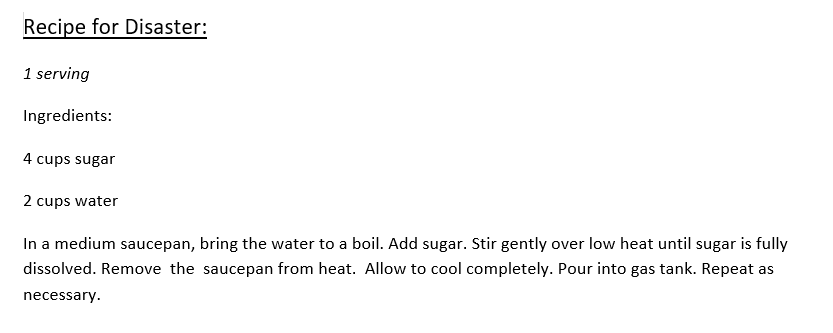
* File được tiết lộ là file *recipe.docx*:



* Biết được tên file nhưng em không đọc được nó trên wireshark vì chức năng export packet bytes không hoạt động, nên em sẽ sử dụng NetworkMiner để đọc file. Trong mục *Files* có 3 file:



* Click chuột phải, chọn *Open file* để đọc nội dung file:



Ông Ann ăn cắp công thức thật ạ.=))

**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

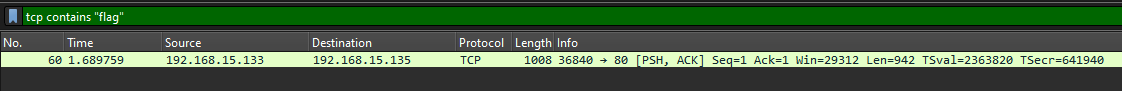
flag{…}

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

**Kịch bản 06. Điều tra trên dữ liệu lưu lượng mạng thu được.**

- Mô tả: Một trọng các máy chủ của CoMix Wave Films bị xâm nhập vàọ

tuần trước, tuy nhiên không có thiệt hại đáng kể nàọ được ghi nhận.

Mặc dù hệ thống tường lửa của công ty rất mạnh nhưng nhóm bảọ

mật của công ty phát hiện ra một số họạt động đáng ngờ, có thể bị

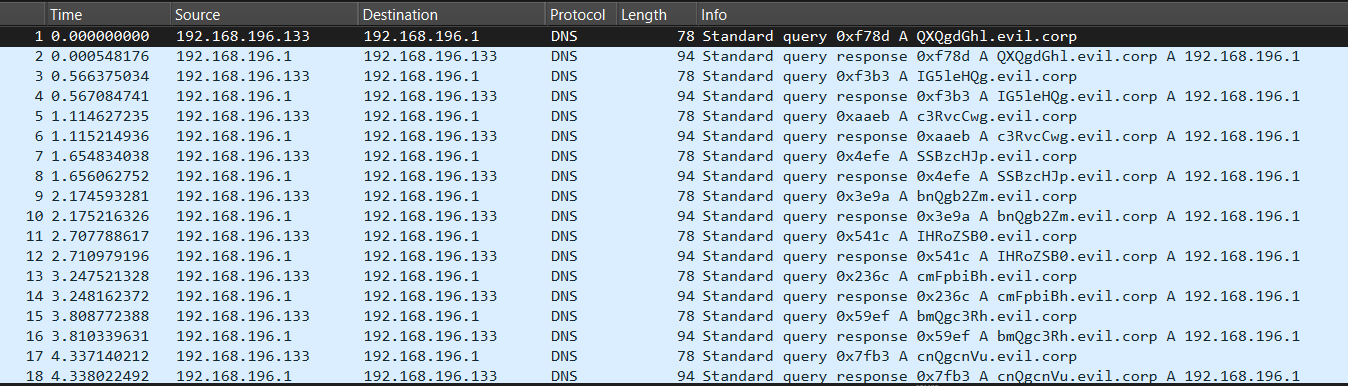
tuồn dữ liệu ra bên ngọài. Hãy điều tra liệu kẻ tấn công đã lấy được

những gì từ máy chủ của công ty, giaọ thức sử dụng? Tìm flag.

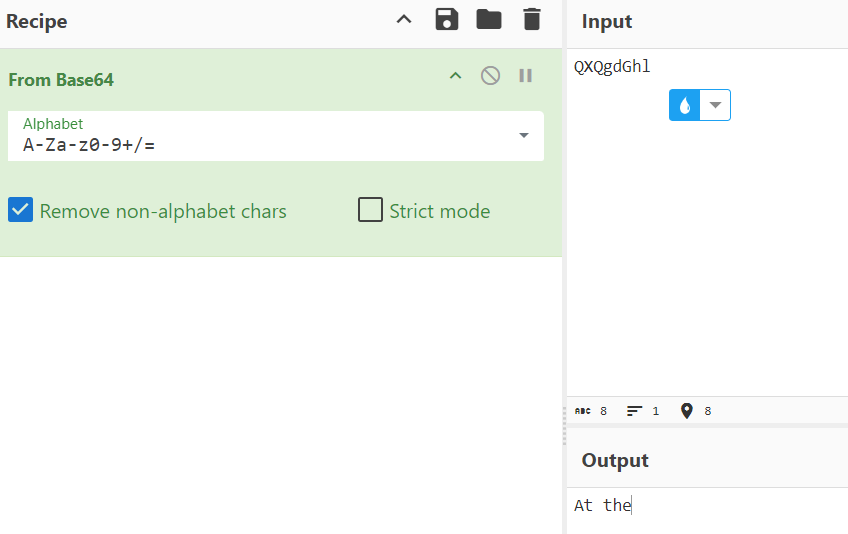
- Tài nguyên: Nandemonaiya\_kb06.pcap

Đáp án:

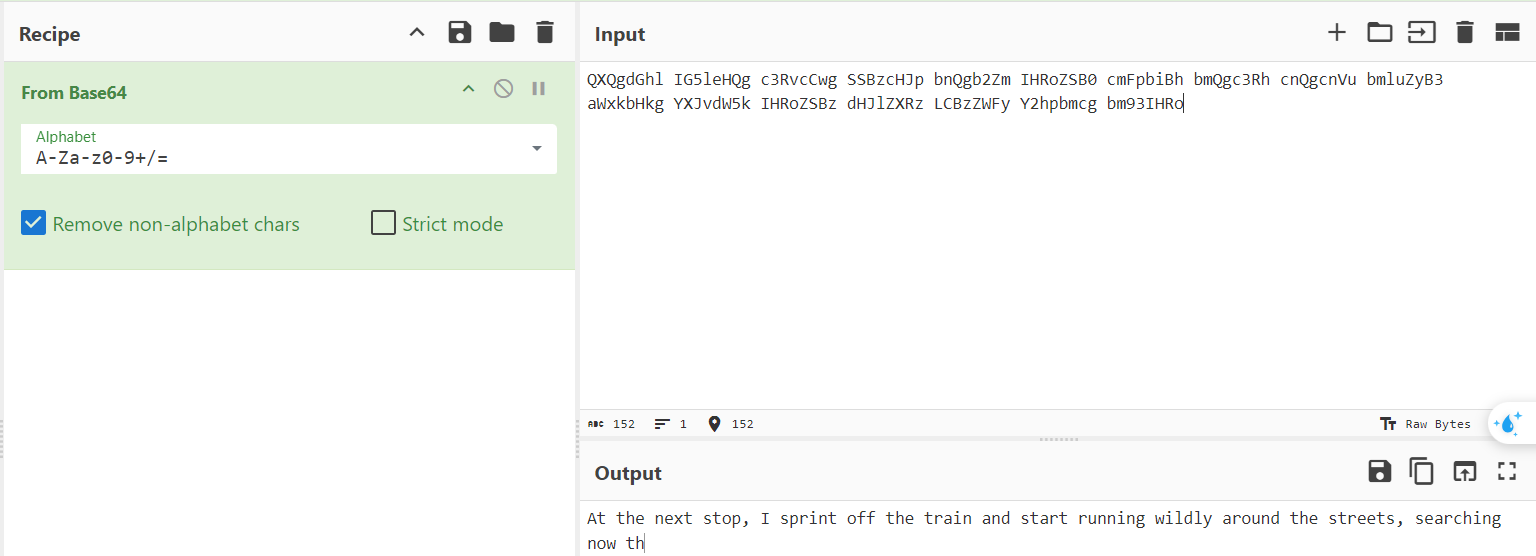
* Tải file pcap và mở trên wireshark, điểm đáng ngờ là hàng loạt gói tin sử dụng DNS và đều trỏ đến web *.evil.corp*:



* Em bóc ngẫu nhiên một đoạn để decode base64 tại trang CyberChef, được thông điệp có ý nghĩa:

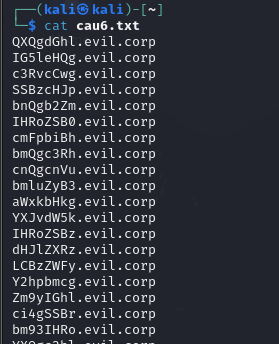


* Em nghĩ là nó ngắn nên em dự định làm thủ công í, nhưng mà dài quá nên em dùng tool ạ=)))



* Sử dụng câu lệnh này để trích xuất các DNS queries từ tệp tin dữ liệu gói tin mạng, lọc ra các trường "dns.qry.name" có chứa chuỗi "evil.corp", và lưu kết quả vào một tệp tin có tên "encoded\_domain.txt":

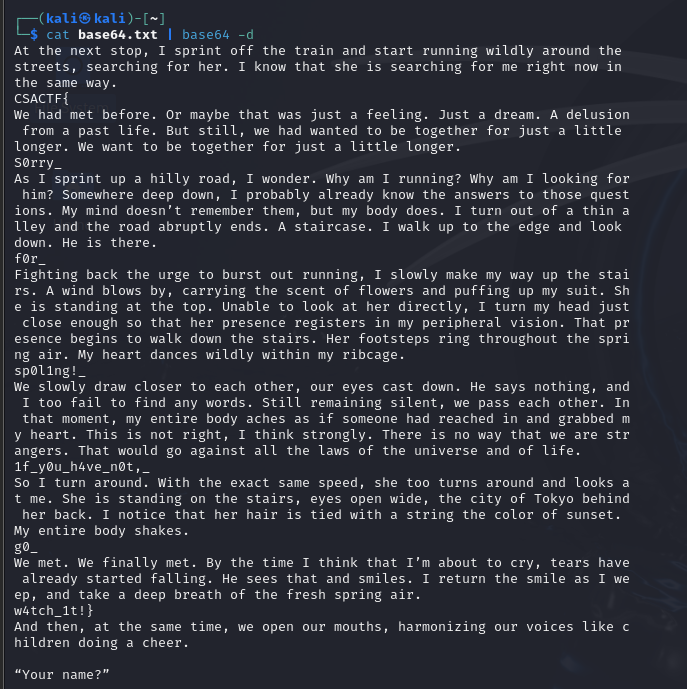
Đang chèn hình ảnh...



* Để tách ra khỏi evil.corp, quan sát thấy mỗi chuỗi base64 đều có 8 kí tự, nên em sẽ cắt lấy 8 kí tự đầu tiên tại mỗi dòng, lưu kết quả vào file mới:



* Cuối cùng decode các chuỗi:



**Flag: CSACTF{S0rry\_f0r\_sp0l1ng!\_1f\_y0u\_h4ve\_n0t,\_g0\_w4tch\_1t!}**

---

***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)