



## BÁO CÁO THỰC HÀNH LAB 4

Môn học: Pháp chứng kỹ thuật số

**Nhóm: Pha Pha**

**THÀNH VIÊN THỰC HIỆN:**

STT	Họ và tên	MSSV
1	Nguyễn Đoàn Xuân Bình	19521265
2	Trần Hoàng Khang	19521671
3	Nguyễn Mỹ Quỳnh	19520241

# BÁO CÁO CHI TIẾT

**Lưu ý:** Trong bài có sử dụng một số tool mà trên các repo chính thống, hay các trang chủ và các nguồn download chính thức bị lỗi/gặp vấn đề/dead link thì mình có thể dùng Wayback Machine để “lùi lại” và xem những version trước của trang web download đó (theo mình làm thì mình lấy phiên bản cách đây 6-9 năm) thì mình download và dùng được tool như bình thường



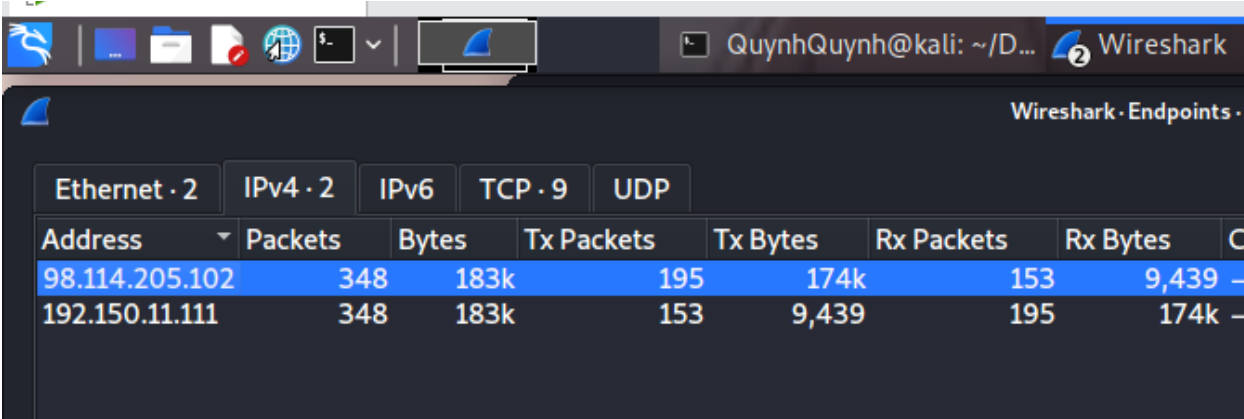
## Kịch bản 01-a. Thực hiện phân tích tập tin dữ liệu mạng.

- 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 trong thời gian diễn ra cuộc tấn công. Sau đó lưu lượng mạng được trích xuất toàn 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

Chọn Menu Statistics/Endpoint List/IP v4 để xem danh sách các IP bắt được.

Ở đây ta thấy chỉ có 2 IP, ta có thể dự đoán:

- 192.150.11.111 là IP private, chính là IP của nạn nhân
- 98.114.205.102 mang địa chỉ IP public, là IP của kẻ tấn công



The screenshot shows the Wireshark interface with the 'Endpoints' tab selected. It displays a table of network endpoints. The table has columns for Address, Packets, Bytes, Tx Packets, Tx Bytes, Rx Packets, and Rx Bytes. Two endpoints are listed: 98.114.205.102 and 192.150.11.111. The first endpoint (98.114.205.102) is highlighted in blue.

Address	Packets	Bytes	Tx Packets	Tx Bytes	Rx Packets	Rx Bytes
98.114.205.102	348	183k	195	174k	153	9,439
192.150.11.111	348	183k	153	9,439	195	174k



Ngoài ra, ở tab ethernet ta có thêm thông tin máy kẻ tấn công có địa chỉ MAC là 00:08:e2:3b:56:01(Cisco).

Ethernet · 2	IPv4 · 2	IPv6	TCP · 9	UDP		
Address	Packets	Bytes	Tx Packets	Tx Bytes	Rx Packets	Rx Bytes
00:08:e2:3b:56:01	348	183k	195	174k	153	
00:30:48:62:4e:4a	348	183k	153	9,439	195	

Tìm thêm thông tin về kẻ tấn công sử dụng trang web:

<http://cqcouter.com/whois/>

Ta được kết quả chi tiết về thông tin như host, location, city, ISP, ... và nhiều thông tin khác nữa.

98.114.205.102 - Geo Information	
IP Address	<a href="#">98.114.205.102</a>
Host	pool-98-114-205-102.phlpa.fios.verizon.net
Location	 US, United States
City	Philadelphia, PA 19154
Organization	Verizon FiOS
ISP	Verizon FiOS
AS Number	AS701 MCI Communications Services, Inc. d/b/a Verizon Business
Latitude	40° 09'25" North
Longitude	74° 98'53" West
Distance	7692.24 km (4779.73 miles)
Map Location <sup>new</sup> <input checked="" type="radio"/> World Map <input type="radio"/> Google Maps <input type="radio"/> Yahoo Maps <input type="radio"/> Microsoft Live Ma	
	

## 98.114.205.102 - Whois Information

```
#
# ARIN WHOIS data and services are subject to the Terms of Use
# available at: https://www.arin.net/resources/registry/whois/tou/
#
# If you see inaccuracies in the results, please report at
# https://www.arin.net/resources/registry/whois/inaccuracy_reporting/
#
# Copyright 1997-2022, American Registry for Internet Numbers, Ltd.
#

NetRange: 98.108.0.0 - 98.119.255.255
CIDR: 98.108.0.0/14, 98.112.0.0/13
NetName: VIS-BLOCK
NetHandle: NET-98-108-0-1
Parent: NET98 (NET-98-0-0-0)
NetType: Direct Allocation
OriginAS:
Organization: Verizon Business (MCICS)
RegDate: 2008-04-02
Updated: 2022-04-29
Ref: https://rdap.arin.net/registry/ip/98.108.0.0

OrgName: Verizon Business
OrgId: MCICS
Address: 22001 Loudoun County Pkwy
City: Ashburn
StateProv: VA
PostalCode: 20147
Country: US
RegDate: 2006-05-30
Updated: 2022-04-29
```

Tiếp theo xem số phiên TCP hiện có dùng Menu Statistics → Conversations, tab TCP.

Kết quả cho thấy có 5 phiên TCP qua các cổng khác nhau:

QuynhQuynh@kali: ~/Desktop

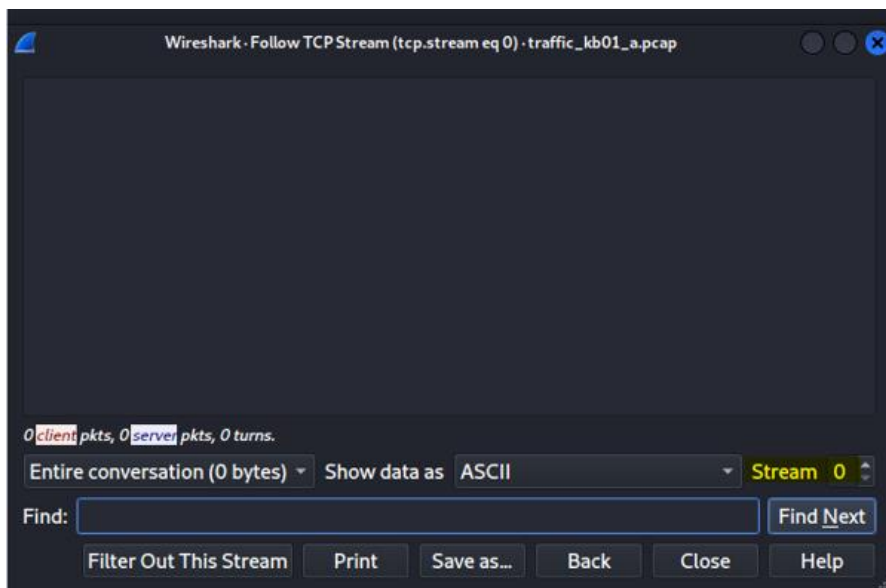
Ethernet · 1	IPv4 · 1	IPv6	TCP · 5	UDP					
Address A	Port A	Address B	Port B	Packets	Bytes	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A
98.114.205.102	1821	192.150.11.111	445	7	412	4	242	3	170
98.114.205.102	1828	192.150.11.111	445	31	6,825	14	4,997	17	1,828
98.114.205.102	2152	192.150.11.111	1080	271	173k	159	167k	112	6,056
192.150.11.111	1957	98.114.205.102	1924	12	817	6	334	6	483
192.150.11.111	36296	98.114.205.102	8884	27	2,069	15	1,051	12	1,018

Đến đây ta tiến hành phân tích từng phiên.

Phiên 1: 98.114.205.102:1821 => 192.150.11.111:445

Address A	Port A	Address B	Port B	Packets	Bytes	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Start	Duration	Bits/s A → B	Bits/s B → A
98.114.205.102	1821	192.150.11.111	445	7	412	4	242	3	170	0.000000	0.3543	5,464	
98.114.205.102	1828	192.150.11.111	445	31	6,825	14	4,997	17	1,828	0.134550	4.9381	8,095	
98.114.205.102	2152	192.150.11.111	1080	271	173k	159	167k	112	6,056	6.142326	10.0719	132k	
192.150.11.111	1957	98.114.205.102	1924	12	817	6	334	6	483	2.091833	3.1000	861	
192.150.11.111	36296	98.114.205.102	8884	27	2,069	15	1,051	12	1,018	5.082620	11.1366	754	

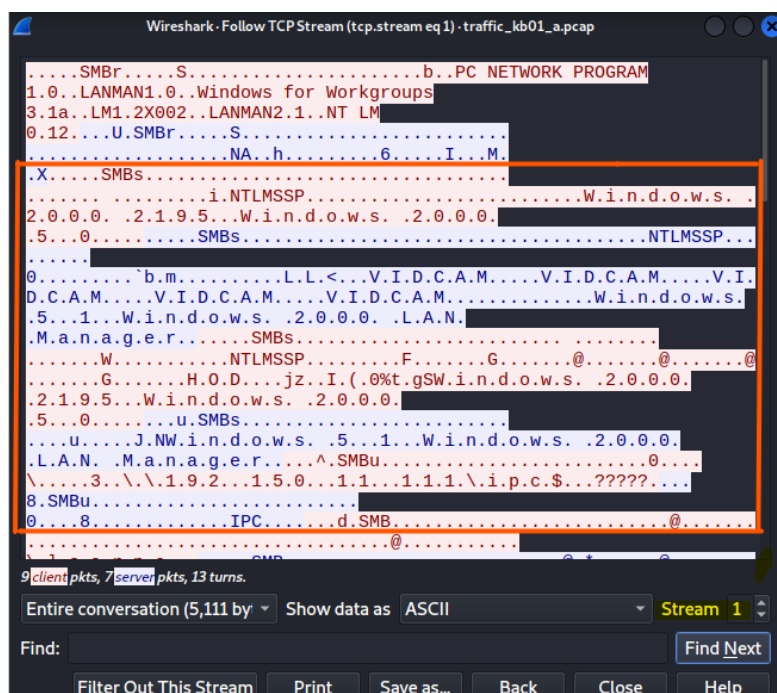
Nội dung TCP Stream không có gì nên ta có thể thấy là ở phiên đầu tiên attacker chỉ tiến hành quét port 445(dịch vụ SMB), cung cấp khả năng chia sẻ file giữa các máy tính hoặc máy in và máy tính.



Phiên 2: 98.114.205.102:1828 => 192.150.11.111:445

Address A	Port A	Address B	Port B	Packets	Bytes	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Start	Duration	Bits/s A → B	Bits/s B
98.114.205.102	1821	192.150.11.111	445	7	412	4	242	3	170	0.000000	0.3543	5,464	
98.114.205.102	1828	192.150.11.111	445	31	6,825	14	4,997	17	1,828	0.134550	4.9381	8,095	
98.114.205.102	2152	192.150.11.111	1080	271	173k	159	167k	112	6,056	6.142326	10.0719	132k	
192.150.11.111	1957	98.114.205.102	1924	12	817	6	334	6	483	2.091833	3.1000	861	
192.150.11.111	36296	98.114.205.102	8884	27	2,069	15	1,051	12	1,018	5.082620	11.1366	754	

Follow TCP stream thì ta nhận thấy thông tin thể hiện máy tính nạn nhân chạy hệ điều hành windows, cụ thể là windows xp hoặc windows 2000



Ngoài ra, chú ý cổng 445 được attacker quét trên máy nạn nhân. Đây là cổng chạy dịch vụ SMB từng được biết đến với việc dính một số lỗ hổng bảo mật.

Tiến hành filter các gói tin thuộc phiên này, ta thấy attacker gửi yêu cầu kết nối tới \$IPC (Path : \\192.150.11.111\\$(ipc)) để có thể gửi lệnh đến nạn nhân

ne	Source	Destination	Protocol	Length	Info
723001	192.150.11.111	98.114.205.102	SMB	311	Session Setup AndX Response, NTLMSSP_CHALLENGE, Error: STATUS_MOR...
840405	98.114.205.102	192.150.11.111	SMB	276	Session Setup AndX Request, NTLMSSP_AUTH, User: \
840419	192.150.11.111	98.114.205.102	TCP	54	445 → 1828 [ACK] Seq=347 Ack=528 Win=8576 Len=0
957617	192.150.11.111	98.114.205.102	SMB	175	Session Setup AndX Response
073151	98.114.205.102	192.150.11.111	SMB	152	Tree Connect AndX Request, Path: \\192.150.11.111\ipc\$
073174	192.150.11.111	98.114.205.102	TCP	54	445 → 1828 [ACK] Seq=468 Ack=626 Win=8576 Len=0
189374	192.150.11.111	98.114.205.102	SMB	114	Tree Connect AndX Response
307145	98.114.205.102	192.150.11.111	SMB	158	NT Create AndX Request, FID: 0x4000, Path: \lsarpc
307168	192.150.11.111	98.114.205.102	TCP	54	445 → 1828 [ACK] Seq=528 Ack=730 Win=8576 Len=0
424860	192.150.11.111	98.114.205.102	SMB	193	NT Create AndX Response, FID: 0x4000
542389	98.114.205.102	192.150.11.111	DCERPC	214	Bind: call_id: 1, Fragment: Single, 1 context items: DSSETUP V0.0...

Tiếp theo gọi hàm DsRoleUpgradeDownlevelServer đồng thời gửi đến một victim một đoạn dữ liệu khá lớn. Tìm hiểu thêm về 'DsRoleUpgradeDownlevelServer' trên mạng thì ta biết được phiên bản remote Windows chứa một lỗ hổng trong chức năng 'DsRolerUpgradeDownlevelServer' của Local Security Authority Server Service (LSASS) cho phép kẻ tấn công thực thi mã tùy ý trên máy chủ từ xa với các đặc quyền hệ thống. Nó là một lỗi về Buffer Overflow của dịch vụ SMB có mã là MS04-011 Microsoft LSASS Service DsRolerUpgradeDownlevelServer Overflow.

31	1.803993	98.114.205.102	192.150.11.111	TCP	1514	1828 → 445 [ACK] Seq=2350 Ack=795 Win=63446 Len=1460 [T...
32	1.804003	192.150.11.111	98.114.205.102	TCP	54	445 → 1828 [ACK] Seq=795 Ack=3810 Win=14600 Len=0
33	1.805992	98.114.205.102	192.150.11.111	DSSETUP	454	DsRoleUpgradeDownlevelServer request[Long frame (3208 by...
34	1.806001	192.150.11.111	98.114.205.102	TCP	54	445 → 1828 [ACK] Seq=795 Ack=4210 Win=17520 Len=0
35	1.978646	98.114.205.102	192.150.11.111	TCP	60	[TCP Dup ACK 29#1] 1828 → 445 [ACK] Seq=4210 Ack=795 Wi...
38	2.134590	192.150.11.111	98.114.205.102	DSSETUP	162	DsRoleUpgradeDownlevelServer response[Long frame (20 byt...
40	2.379299	98.114.205.102	192.150.11.111	TCP	60	1828 → 445 [ACK] Seq=4210 Ack=903 Win=63338 Len=0

Từ đây có thể đoán được đó là một đoạn shellcode được bao quanh bởi hàng loạt các giá trị NOP (x90) mà attacker sử dụng để điều khiển từ xa.

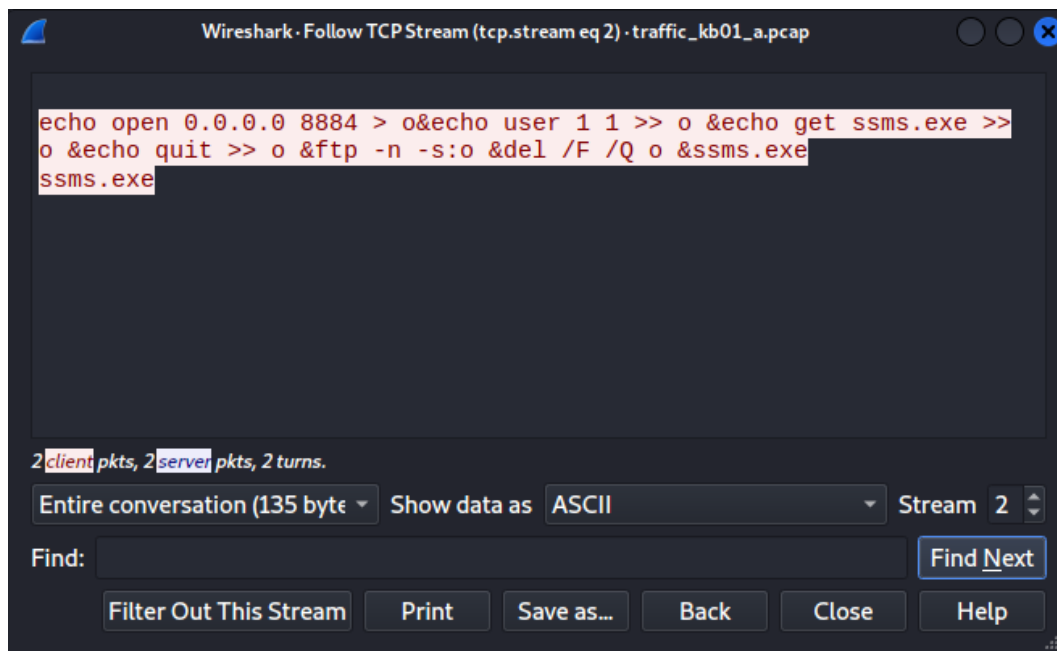
▶ NetBIOS Session Service
▶ SMB (Server Message Block Protocol)
▶ SMB Pipe Protocol
▶ Distributed Computing Environment / Remote Procedure Call (DCE/RPC) Request, Fr
▶ <b>Active Directory Setup, DsRoleUpgradeDownlevelServer</b>
Operation: DsRoleUpgradeDownlevelServer (9)
[Response in frame: 38]
▶ Long frame
00e0 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 ..... ..
00f0 90 90 90 90 90 90 90 90 90 90 90 90 90 90 ..... ..
0100 90 90 90 90 90 90 90 90 90 90 90 90 90 90 ..... ..
0110 90 90 90 90 90 90 90 90 90 90 90 90 90 90 ..... ..
0120 33 c9 66 b9 7d 01 80 34 0a 99 e2 fa eb 05 e8 eb 3.f.}..4 ..... ..
0130 ff ff ff 70 95 98 99 99 c3 fd 38 a9 99 99 99 12 ...p.... ..8.....
0140 d9 95 12 e9 85 34 12 d9 91 12 41 12 ea a5 12 ed ....4.. ..A.....
0150 87 e1 9a 6a 12 e7 b9 9a 62 12 d7 8d aa 74 cf ce ...j.... b....t..
0160 c8 12 a6 9a 62 12 6b f3 97 c0 6a 3f ed 91 c0 c6 ....b.k. ..j?....
Frame (454 bytes)
Reassembled TCP (3320 bytes)



Phiên 3: 192.150.11.111:1957 <= 98.114.205.102:1924

Address A	Port A	Address B	Port B	Packets	Bytes	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Start	Duration	Bits/s A → B	Bits/s B → A
98.114.205.102	1821	192.150.11.111	445	7	412	4	242	3	170	0.000000	0.3543	5,464	
98.114.205.102	1828	192.150.11.111	445	31	6,825	14	4,997	17	1,828	0.134550	4.9381	8,095	
98.114.205.102	2152	192.150.11.111	1080	271	173k	159	167k	112	6,056	6.142326	10.0719	132k	
192.150.11.111	1957	98.114.205.102	1924	12	817	6	334	6	483	2.091833	3.1000	861	
192.150.11.111	36296	98.114.205.102	8884	27	2,069	15	1,051	12	1,018	5.082620	11.1366	754	

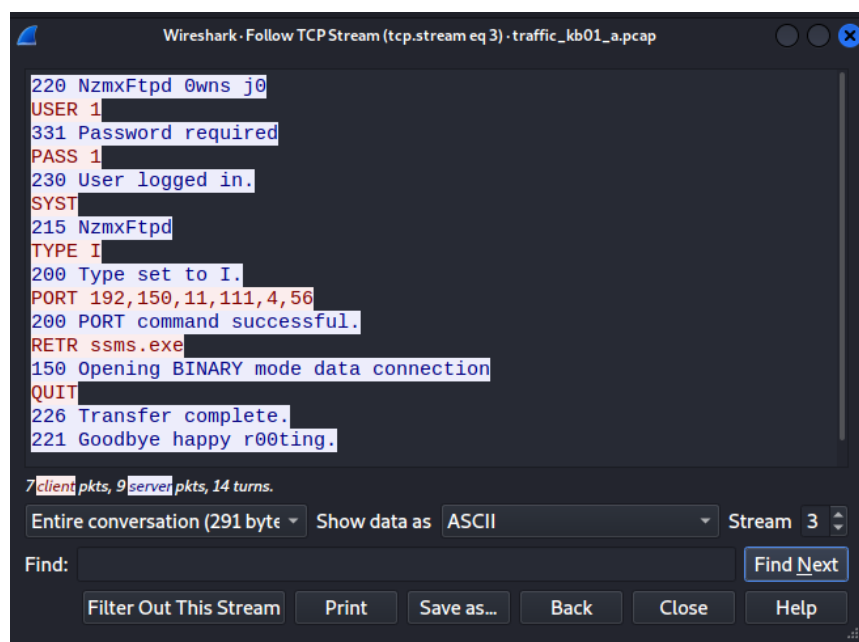
Ở phiên này ta có thể dự đoán attack gửi một chuỗi câu lệnh đến port 1957 vừa mở của victim sử dụng shellcode phía trên. Lệnh cmd yêu cầu tải 1 file có tên là ssms.exe thông qua FTP.



Phiên 4: 192.150.11.111:36296 => 98.114.205.102:8884

Address A	Port A	Address B	Port B	Packets	Bytes	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Start	Duration	Bits/s A → B	Bits/s B → A
98.114.205.102	1821	192.150.11.111	445	7	412	4	242	3	170	0.000000	0.3543	5,464	
98.114.205.102	1828	192.150.11.111	445	31	6,825	14	4,997	17	1,828	0.134550	4.9381	8,095	
98.114.205.102	2152	192.150.11.111	1080	271	173k	159	167k	112	6,056	6.142326	10.0719	132k	
192.150.11.111	1957	98.114.205.102	1924	12	817	6	334	6	483	2.091833	3.1000	861	
192.150.11.111	36296	98.114.205.102	8884	27	2,069	15	1,051	12	1,018	5.082620	11.1366	754	

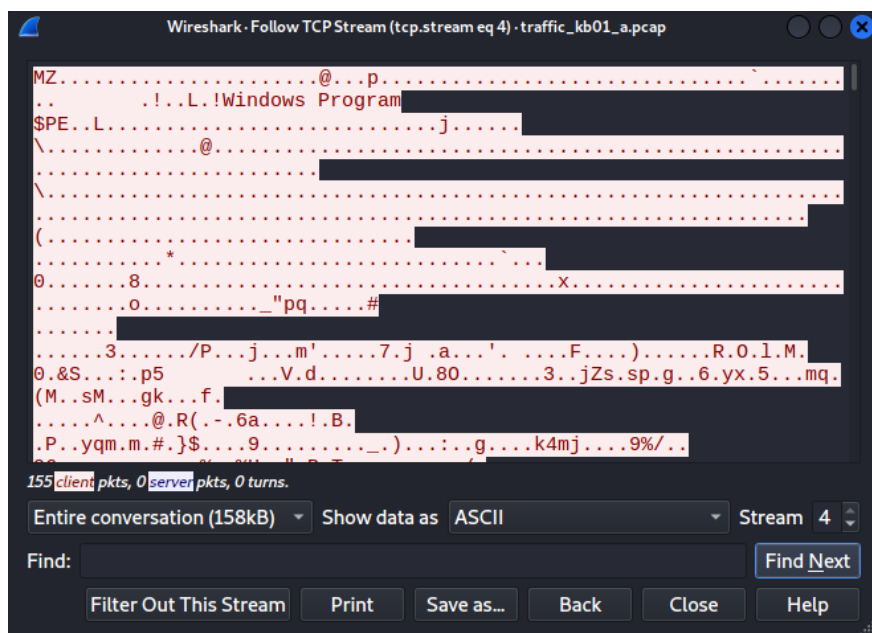
Tại đây, ta thấy nạn nhân thực hiện các câu lệnh ở phiên bên trên, kết nối tới FTP server và tải file ssms.exe về máy.



Phiên 5: 98.114.205.102:2152 => 192.150.11.111:1080

Address A	Port A	Address B	Port B	Packets	Bytes	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Start	Duration	Bits/s A → B	Bits/s B → A
98.114.205.102	1821	192.150.11.111	445	7	412	4	242	3	170	0.000000	0.3543	5,464	
98.114.205.102	1828	192.150.11.111	445	31	6,825	14	4,997	17	1,828	0.134550	4.9381	8,095	
98.114.205.102	2152	192.150.11.111	1080	271	173k	159	167k	112	6,056	6.142326	10.0719	132k	
192.150.11.111	1957	98.114.205.102	1924	12	817	6	334	6	483	2.091833	3.1000	861	
192.150.11.111	36296	98.114.205.102	8884	27	2,069	15	1,051	12	1,018	5.082620	11.1366	754	

File được ssms.exe được tải về:





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

- Mô tả: Tập tin pcap được cho 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 stream TCP, sau đó phân tích stream đã giải mã để tìm flag.
- Đáp án: Flag: be02d2a396482969e39d92b6e440f5e3

Lấy thông tin cơ bản traffic network bằng **aircrack-ng**, trích xuất thông tin **Wifi Encryption (WPA)**.

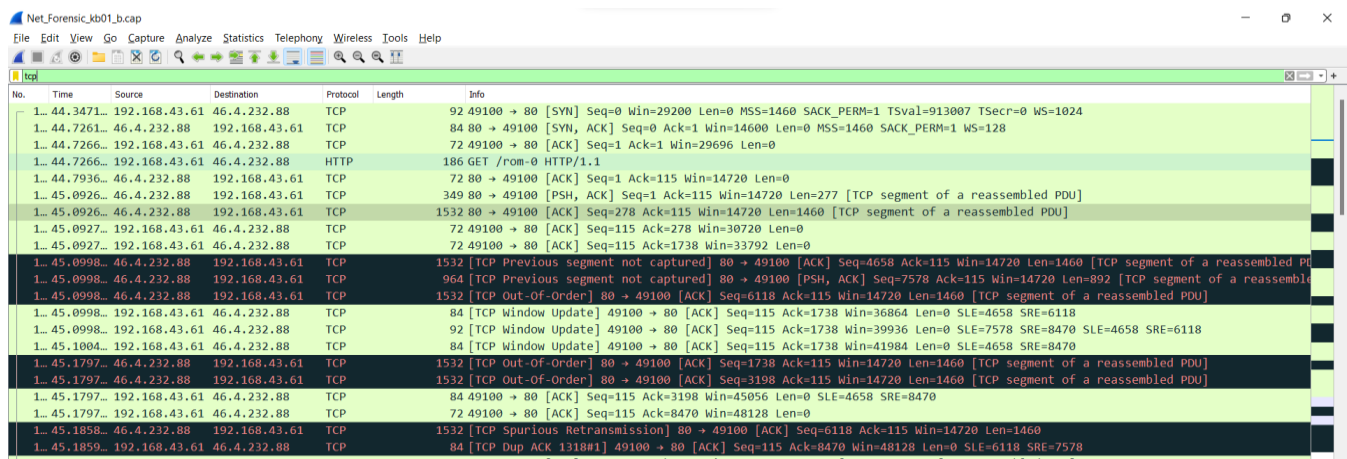
```
(virus@virus)-[~/Desktop]
$ aircrack-ng Net_Forensic_kb01_b.cap
Reading packets, please wait...
Opening Net_Forensic_kb01_b.cap
Read 8525 packets.

# BSSID ESSID Encryption
1 38:AA:3C:32:46:60 SD Unknown
2 74:EA:3A:FF:0F:48 Rome WPA (1 handshake)

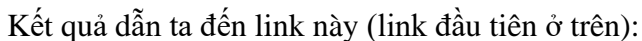
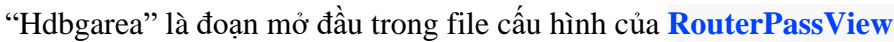
Index number of target network ?
```

*\*Note: Loại encryption này cũng khá cũ rồi, giờ mình thấy người ta hay xài WPA2*

Dùng Wireshark để đào sâu thêm, lọc các gói tin TCP

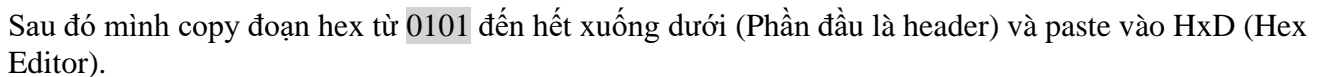


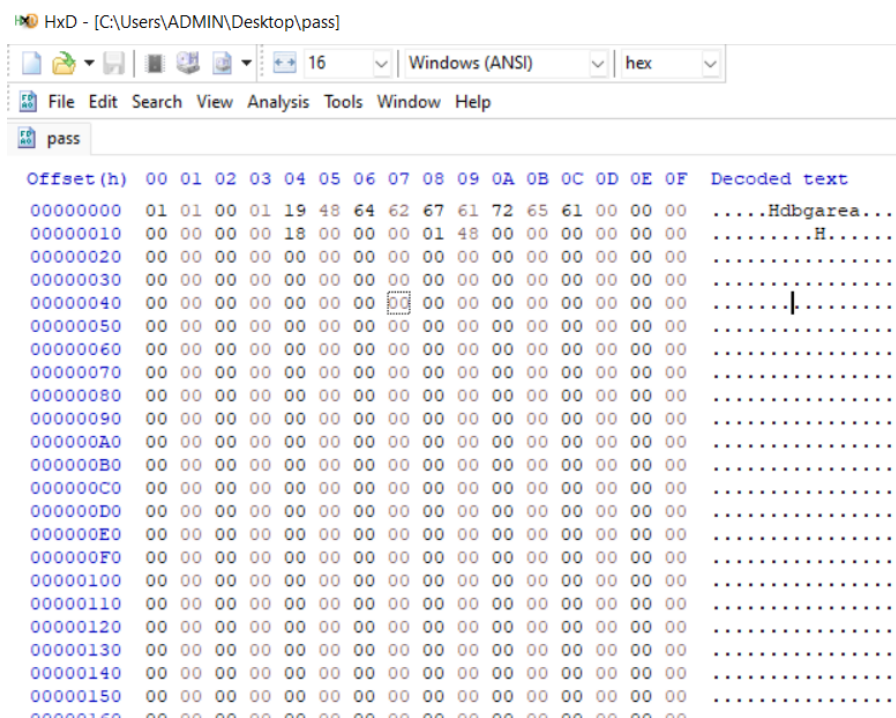
Chuột phải vào gói tin TCP bất kỳ -> TCP Stream hoặc (Ctrl + Alt + Shift + T). Ta sẽ thấy full nội dung được gửi đi.



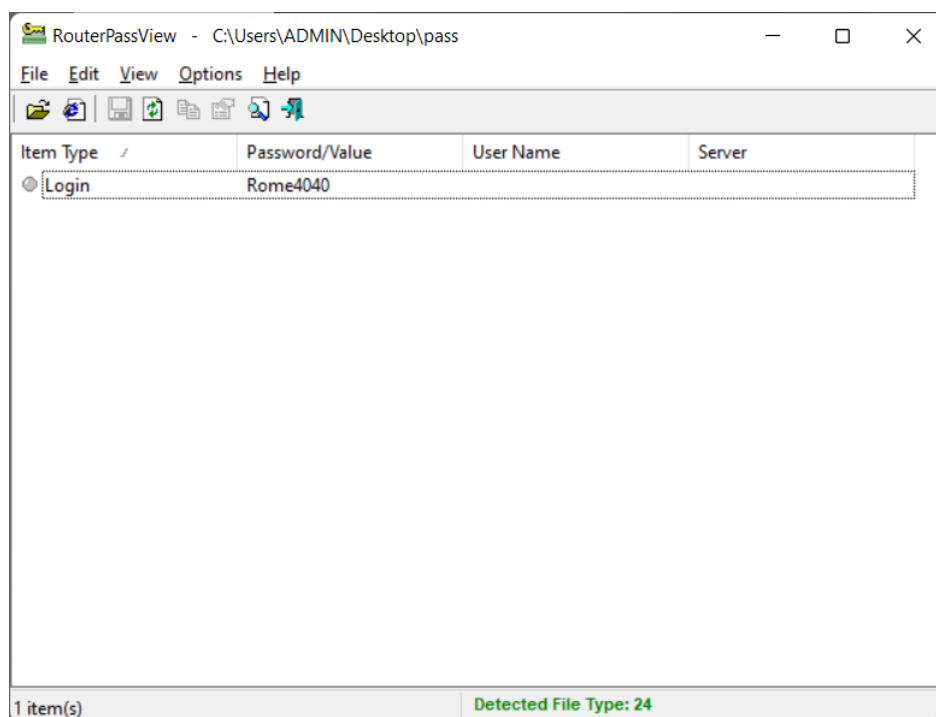
- Version 1.62:
  - Added support for another version of rom-0 **Hdbgarea** file format (Zyxel P-2612HWU-F1 Modem).
- Version 1.61:
  - Added support for LevelOne WBR-3406TX v2 and possibly other routers (with DDC6031 and ZXL6031 signatures)
- Version 1.60:
  - Added support for decompression of rom-0 **Hdbgarea** file format, which is used in multiple routers, including Huawei Echolife HG510a/HG520s/HG520b/HG520c, TP-LINK TD-W8901N, TP-LINK TD-W8916, TP-LINK TD-W8901G, TP-LINK TD-W8951ND, TP-LINK TD-W8817, SmartAX MT880a/MT8804/MT880d/MT882a, Zyxel AMG1302, and possibly others. Be aware that in table mode, only the login password of the router is displayed, but you can find all other data if you switch to Hex Dump mode

**Khoa Mạng máy tính và Truyền thông**





Lưu lại với tên file **config**. Sau đó bỏ vào phần mềm **RouterPassView** ở trên để recover lại password.

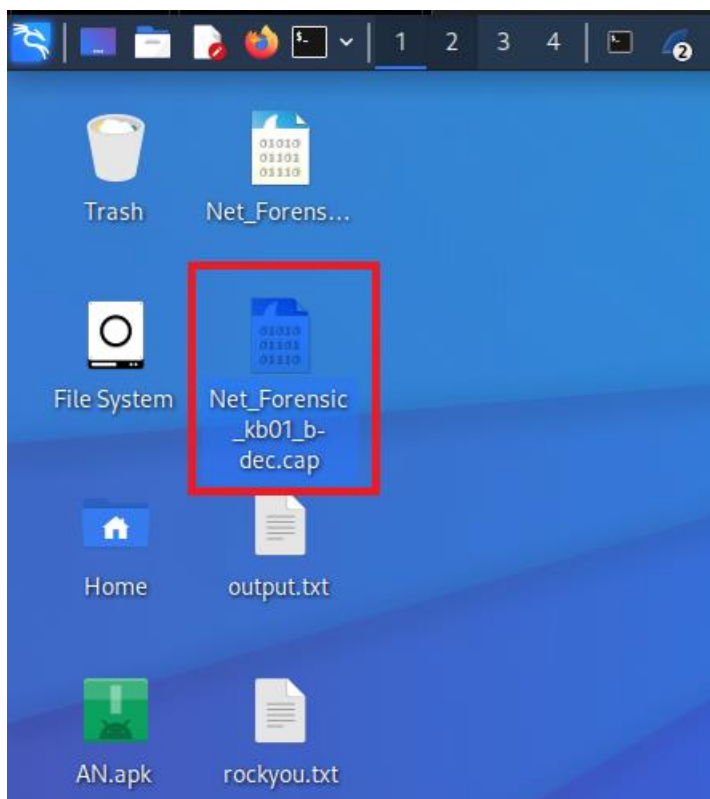


Mật khẩu ở đây là **Rome4040**. Sau đó dùng aircrack-ng để giải mã các packet với mật khẩu đã xác định với tùy chọn **-e** là **ESSID** và **-p** là **password** đã tìm

```
airdecap-ng -e 'Rome' -p Rome4040 Net_Forensic_kb01_b.cap
```



```
(virus@virus)-[~/Desktop]
$ airdecap-ng -e 'Rome' -p Rome4040 Net_Forensic_kb01_b.cap 130 x
Total number of stations seen      10
Total number of packets read      8525
Total number of WEP data packets   0
Total number of WPA data packets  1681
Number of plaintext data packets   84
Number of decrypted WEP packets    0
Number of corrupted WEP packets    0
Number of decrypted WPA packets    391
Number of bad TKIP (WPA) packets   0
Number of bad CCMP (WPA) packets   0
```



Kiểm tra có flag trên toàn bộ gói tin?

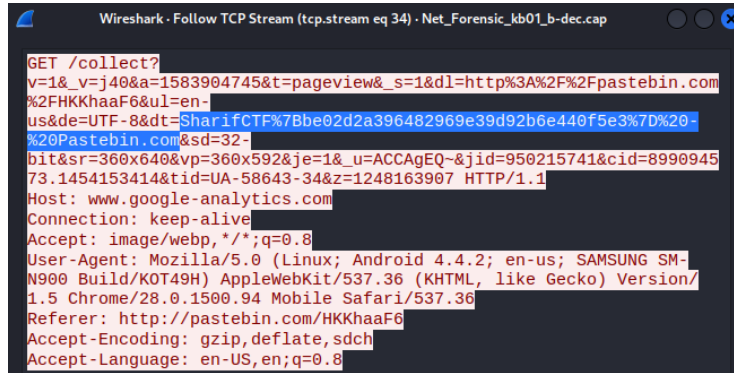
```
(virus@virus)-[~/Desktop]
$ strings Net_Forensic_kb01_b-dec.cap | grep -i ctf 1 x
SharifCTF{be02d2a396482969e39d92b6e440f5e3}
GET /collect?v=1&v=j40&a=1583904745&t=pageview&s=1&dl=http%3A%2F%2F
pastebin.com%2FHKKhaaF6&ul=en-us&de=UTF-8&dt=SharifCTF%7Bbe02d2a39648
2969e39d92b6e440f5e3%7D%20-%20Pastebin.com&sd=32-bit&sr=360x640&vp=36
0x592&je=1&u=ACCAgEQ~&jid=950215741&cid=899094573.1454153414&tid=UA-
58643-34&z=1248163907 HTTP/1.1
```

Như ta thấy thì dùng cách này có thể thấy được flag. Vì đây là một bài ctf nên đến đây có thể xong rồi. Tuy nhiên, mình nên làm challenge với góc độ phân tích packet nên mình sẽ examine lại file đã được decrypt trên.

Ta thấy gói packet có timestamp 343 có định dạng flag:

Time	Source	Destination	Protocol	Length	Info
88.22.869476	192.168.1.4	95.172.94.57	HTTP	706	GET /pixel;r=505560555;a=p-cirF4xglUzNc;fpan=0;fpa=P0-1073353128-1454153794039;ns=0;ce=1;cm=;je=1;sr=360x640x32;enc=n;dst=1;
92.23.191526	192.168.1.4	74.125.136.102	HTTP	968	GET /_utm.gif?utmvt=5.6.7&utms=5&utmn=stackexchange.com&utmcs=UTF-8&utmsr=360x640&utmvp=980x1611&utmcs=32-bit
94.23.230440	192.168.1.4	104.16.118.182	HTTP	848	GET /topbar/get-unread-counts?_145415864343 HTTP/1.1
105.23.762919	192.168.1.4	173.194.112.51	HTTP	725	GET /uds/css/small-logo.png HTTP/1.1
108.23.968616	192.168.1.4	74.125.136.102	HTTP	644	GET /generate_204 HTTP/1.1
164.25.408614	192.168.1.4	173.194.112.51	HTTP	725	GET /uds/css/small-logo.png HTTP/1.1
200.35.164392	192.168.1.4	74.125.136.113	HTTP	392	GET /generate_204 HTTP/1.1
237.41.240174	192.168.1.4	74.125.136.102	HTTP	202	GET /generate_204 HTTP/1.1
247.42.502320	192.168.1.4	74.125.136.138	HTTP	227	GET /generate_204 HTTP/1.1
261.46.072240	192.168.1.4	74.125.136.113	HTTP	202	GET /generate_204 HTTP/1.1
281.55.204837	192.168.1.4	213.186.33.2	HTTP	541	GET /resources/favicon.ico HTTP/1.1
311.65.977001	192.168.1.4	104.20.64.56	HTTP	914	POST /post.php HTTP/1.1
317.66.758824	192.168.1.4	104.20.64.56	HTTP	632	[TCP ACKed unseen segment] GET /HKKhaf6 HTTP/1.1
319.67.547836	104.20.64.56	192.168.1.4	HTTP	1454	[TCP Previous segment not captured] Continuation
325.67.645670	192.168.1.4	104.20.64.56	HTTP	308	[TCP ACKed unseen segment] GET /pastebin.min.css?v=972 HTTP/1.1
339.68.847977	192.168.1.4	204.11.109.68	HTTP	415	GET /real/tags/Pastebincom/Safe/tags.js HTTP/1.1
343.69.887060	192.168.1.4	74.125.136.102	HTTP	735	GET /collect?v=1&v=140&a=1583904745&t=pageview&s=1&dl=http%3A%2F%2Fpastebin.com%2FHKKhaf6&ul=en-us&de=UTF-8&dt=SharifCTF%7Bbe02d2a396482969e39d92b6e440f5e3%7D%20%20Pastebin.com&sd=32-bit&sr=360x640&vp=360x592&je=1&u=ACCAgEQ~&jid=950215741&cid=899094573.1454153414&tid=UA-58643-34&z=1248163907 HTTP/1.1

Follow stream để xem toàn bộ flag:



Đưa vào [URL Decode](#) để xem truy vấn gốc:

[< DECODE >](#) Decodes your data into the area below.

SharifCTF{be02d2a396482969e39d92b6e440f5e3} - Pastebin.com

Flag: **SharifCTF{be02d2a396482969e39d92b6e440f5e3}**

## 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.

Gợi ý: Wireshark/tshark

❖ Thực hiện phân tích các request DNS, các truy cập HTTP đến các trang web nào

Để xác định được user truy cập trang web nào. Ta có combo payload sau:

```
tshark -r capture-output_kb02.pcap -Y http.request -T fields -e http.request.full_uri
```

Với **-r** là file pcap cần phân tích, **-Y** là filter (syntax như Wireshark), **-T** là dạng xuất ra (ở đây là fields) và **-e** là trường thông tin được lấy ra.

Các URL được lấy ra khá nhiều và bị trùng lặp. Để cho đẹp hơn thì mình nên **sort** lại (để các link giống nhau gần nhau) và sau đó **uniq** theo số dòng (tức là số lần xuất hiện)

```
tshark -r capture-output_kb02.pcap -Y http.request -T fields -e http.request.full_uri | sort | uniq -c
```



Toàn bộ output:

```
(virus@virus)-[~/Desktop]
$ tshark -r capture-output_kb02.pcap -Y http.request -T fields -e http.request.full_uri | sort | uniq -c
357 http://10.102.20.169:8080/ping
148 http://10.102.20.169:8080/v2-beta/publish
28 http://239.255.255.250:1900*
1 http://connectivity-check.ubuntu.com/
1 http://fsend.vn/img/slides/slide-2.png
1 http://fsend.vn/img/slides/slide-3.png
1 http://fsend.vn/Roboto-Bold.c0f1e4a4fdb8048c72e.woff2
1 http://fsend.vn/Roboto-Light.3c37aa69cd77e6a53a06.woff2
1 http://fsend.vn/Roboto-Regular.5136cbe62a63604402f2.woff2
1 http://fsend.vn/v2/services
1 http://fsend.vn/v2/transfers?key=Q4uDmemqP1FCFpEjexDnGSfueKU2uviN
1 http://fsend.vn/v2/up-keys
2 http://fsend.vn/v2/up-keys/Q4uDmemqP1FCFpEjexDnGSfueKU2uviN/upload
1 http://linkmaker.itunes.apple.com/assets/shared/badges/vi-vn/appstore-lrg.svg
2 http://ocsp2.globalsign.com/gsalphasha2g2
1 http://ocsp2.globalsign.com/gsorganizationvalsha2g2
18 http://ocsp.comodoca.com/
30 http://ocsp.digicert.com/
3 http://ocsp.godaddy.com/
5 http://ocsp.int-x3.letsencrypt.org/
21 http://ocsp.pki.goog/GTSGIAG3
2 http://ocsp.sca1b.amazontrust.com/
2 http://ocsp.sectigo.com/
2 http://ocsp.trustwave.com/
1 http://status.geotrust.com/
1 http://status.rapidssl.com/
1 http://tuoitre.vn/
2 http://up.fshare.vn/upload/dZFL+bxh+3-P3-GAqMhhaORkNJcYxR6ITPZLZBzywLUWX2twgbTa7ZH0tsPUJ45wPUUYvq
UceOhozr46?flowChunkNumber=1&flowChunkSize=20000000&flowCurrentChunkSize=4698321&flowTotalSize=4698321&flowIdentifier=4698321-Anh-Oi-O-Lai-Chi-Pu-Dat-Gmp3&flowFilename=Anh-Oi-O-Lai-Chi-Pu-Dat-G.mp3&flowRelativePath=Anh-Oi-O-Lai-Chi-Pu-Dat-G.mp3&flowTotalChunks=1
2 http://up.fshare.vn/upload/XDjxYAUfdouRNmKQeh2WrQrLavWDINxXJcfi2NxGwvoy0eh5jUAoAQeJJSnztLYXGEF4gSG8j5A13EOI?flowChunkNumber=1&flowChunkSize=20000000&flowCurrentChunkSize=90429&flowTotalSize=90429&flowIdentifier=90429-image.jpg&flowFilename=image.jpg&flowRelativePath=image.jpg&flowTotalChunks=1
```

- ❖ 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).

Ở đây, như hình trên, ta có thể search google với các domain trên và thấy user sử dụng 2 trang web chính để upload file:

- <http://fsend.vn>
- <https://www.fshare.vn/>

Dùng Wireshark để xem thông tin các packet có request method là POST trên các URL này. Dùng bộ lọc `http.file_data` :

capture-output\_kb02.pcap

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

http.file\_data

No.	Time	Source	Destination	Protocol	Length	Info
82	0.57247...	10.102.20.169	10.102.20.166	HTTP	222	HTTP/1.1 200 OK (text/plain)
83	0.57252...	10.102.20.169	10.102.20.166	HTTP	222	HTTP/1.1 200 OK (text/plain)
1..	0.63631...	10.102.20.167	10.102.20.169	HTTP/JSON	171	POST /v2-beta/publish HTTP/1.1, JavaScript Object Notation (application/json)
1..	0.64000...	10.102.20.166	10.102.20.169	HTTP/JSON	1969	POST /v2-beta/publish HTTP/1.1, JavaScript Object Notation (application/json)
2..	2.57987...	10.102.20.169	10.102.20.166	HTTP	222	HTTP/1.1 200 OK (text/plain)
2..	2.58042...	10.102.20.169	10.102.20.166	HTTP	222	HTTP/1.1 200 OK (text/plain)
7..	4.58829...	10.102.20.169	10.102.20.166	HTTP	222	HTTP/1.1 200 OK (text/plain)
7..	4.58840...	10.102.20.169	10.102.20.166	HTTP	222	HTTP/1.1 200 OK (text/plain)
8..	5.60178...	10.102.20.166	10.102.20.169	HTTP/JSON	589	POST /v2-beta/publish HTTP/1.1, JavaScript Object Notation (application/json)
8..	5.60180...	10.102.20.167	10.102.20.169	HTTP/JSON	589	POST /v2-beta/publish HTTP/1.1, JavaScript Object Notation (application/json)
9..	6.59762...	10.102.20.169	10.102.20.166	HTTP	222	HTTP/1.1 200 OK (text/plain)
9..	6.60002...	10.102.20.169	10.102.20.166	HTTP	222	HTTP/1.1 200 OK (text/plain)
1..	8.64443...	10.102.20.180	172.217.161.163	OCSP	449	Request
1..	8.68517...	172.217.161.1...	10.102.20.180	OCSP	767	Response
1..	8.96916...	10.102.20.169	10.102.20.166	HTTP	222	HTTP/1.1 200 OK (text/plain)
1..	8.96924...	10.102.20.169	10.102.20.166	HTTP	222	HTTP/1.1 200 OK (text/plain)
1..	10.1217...	10.102.20.180	172.217.161.163	OCSP	449	Request

Thực hiện xem một số POST packet:

Wireshark · Follow HTTP Stream (tcp.stream eq 14) · capture-output\_kb02.pcap

POST /v2-beta/publish HTTP/1.1  
 Host: 10.102.20.169:8080  
 User-Agent: Go-http-client/1.1  
 Content-Length: 15858  
 Authorization: Basic N0ZDMzcwNTQwNzkwOTIARTRBCRDg6NVJSazQ1MWF1MmVOWjRkN0oxRHlxQWJmc2hqWTRHdVRqV3p0TlhuWg==  
 Content-Type: application/json  
 Accept-Encoding: gzip

```
{
  "id": "6ca76caa-0afa-4d57-75cd-fe02034f5e70",
  "links": null,
  "actions": null,
  "data": {
    "options": {
      "instances": true
    },
    "resources": [
      {
        "kind": "physicalHost",
        "name": "docker-nodes-b",
        "systemContainer": "",
        "type": "physicalHost",
        "uuid": "b0935bcf-6d57-4988-4373-a815b789ab42",
        {
          "hostname": "docker-nodes-b",
          "kind": "docker",
          "labels": {
            "io.rancher.host.agent_image": "rancher/agent:v1.2.11",
            "io.rancher.host.docker_version": "18.09",
            "io.rancher.host.linux_kernel_version": "4.4",
            "io.rancher.host.os": "linux",
            "localStorageMb": 30225510,
            "physicalHostUuid": "b0935bcf-6d57-4988-4373-a815b789ab42",
            "systemContainer": "",
            "type": "host",
            "uuid": "ce0f4ff7-5d78-40ad-57d2-27077ccfbc4e",
            {
              "hostUuid": "ce0f4ff7-5d78-40ad-57d2-27077ccfbc4e",
              "kind": "docker",
              "name": "docker-nodes-b Storage Pool",
              "systemContainer": "",
              "type": "storagePool",
              "uuid": "ce0f4ff7-5d78-40ad-57d2-27077ccfbc4e-pool",
              {
                "addresses": "10.102.20.166",
                "hostUuid": "ce0f4ff7-5d78-40ad-57d2-27077ccfbc4e",
                "systemContainer": "",
                "type": "ipAddress",
                "uuid": "10.102.20.166",
                {
                  "systemContainer": "",
                  "type": "hostUuid",
                  "uuid": "ce0f4ff7-5d78-40ad-57d2-27077ccfbc4e",
                  {
                    "created": 1553653830,
                    "dockerId": "50e7b3a0c0b99250392fe88bda8f739835a4e8831792e8e6f3d1df65b6965472",
                    "image": "ghost",
                    "labels": {
                      "io.rancher.cni.network": "ipsec",
                      "io.rancher.cni.wait": "true",
                      "io.rancher.container.ip": "10.42.90.233/16",
                      "io.rancher.container.mac_address": "02:27:44:5c:83:91",
                      "io.rancher.container.name": "ghost-ghost-1",
                      "io.rancher.container.uuid": "13dbc6b5-4d0f-4e93-9c15-db3bbe8acd71",
                      "io.rancher.environment.uuid": "adminProject",
                      "io.rancher.project.name": "ghost",
                      "io.rancher.project_service.name": "ghost/ghost",
                      "io.rancher.service.deployment.unit": "5d9d534e-6f78-430a-9efb-3a5270397121",
                      "io.rancher.service.hash": "a8aeca2e57f74ca15a8c8c6e7c2995a0586102ce",
                      "io.rancher.service.launch.config": "io.rancher.service.primary.launch.config",
                      "io.rancher.stack.name": "ghost",
                      "io.rancher.stack.uuid": "5d9d534e-6f78-430a-9efb-3a5270397121",
                      "io.rancher.stack.uuid": "5d9d534e-6f78-430a-9efb-3a5270397121"
                    }
                  }
                }
              }
            }
          }
        }
      ]
    }
  }
}
```

75 client pkts, 75 server pkts, 149 turns.

Entire conversation (446 kB)

Show data as ASCII

Find:  Find Next

Filter Out This Stream Print Save as... Back Close Help

Ta thấy không có gì đặc biệt lắm. Thử tìm kiếm chặt chẽ hơn bằng cách thêm bộ lọc với các **request POST** và URL chứa **"http://fsend.vn"**. Filter Wireshark:

**http.file\_data && http.request.method == "POST" && http contains "http://fsend.vn"**

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

No.	Time	Source	Destination	Protocol	Length	Info
193096	360.373170150	10.102.20.180	118.69.164.19	HTTP/JSON	471	POST /v2/up-keys HTTP/1.1, JavaScript Object Notation (application/json)
196460	361.204993679	10.102.20.180	118.69.164.18	HTTP	49572	POST /upload/dZFL+bxh+3-P3-GAqMhhaORkNjcyXR6ITPZLZBzywLUNX2twgbTa7ZHotsPUJ45wPUUYvqUc
197158	362.7399912473	10.102.20.180	118.69.164.18	HTTP	11496	POST /upload/XDjxYAUfduoRnmKQeh2WqRqLavMDINXJcfi2Nkgwvov0eh5jUAoAQeJJSnztlyXGEF4gSG8
197492	362.957310209	10.102.20.180	118.69.164.19	HTTP/JSON	610	POST /v2/transfers?key=Q4u0MemqP1FCFpEjexDnGSfueKU2uviN HTTP/1.1, JavaScript Object

May mắn thay, **follow** packet đầu tiên ta thấy có chứa mọi thông tin ta cần. File được upload gồm 1 file mp3 và 1 file ảnh:

**Anh-Oi-O-Lai-Chi-Pu-Dat-G.mp3**

```
{ "file_name": "Anh-Oi-O-Lai-Chi-Pu-Dat-G.mp3", "file_size": 4698321 } HTTP/1.1 200 OK
Server: Fshare
Date: Tue, 21 May 2019 02:56:15 GMT
Content-Type: application/json; charset=UTF-8
Transfer-Encoding: chunked
Connection: keep-alive
Vary: Accept-Encoding
Vary: Accept-Encoding
Content-Encoding: gzip
```

## image.jpg

```
{ "file_name": "image.jpg", "file_size": 90429 } HTTP/1.1 200 OK
Server: Fshare
Date: Tue, 21 May 2019 02:56:17 GMT
Content-Type: application/json; charset=UTF-8
Transfer-Encoding: chunked
Connection: keep-alive
Vary: Accept-Encoding
Vary: Accept-Encoding
Content-Encoding: gzip
```

## Thông tin gửi:

```
{ "recipients": [ "duypt@uit.edu.vn" ], "message": "Khong o lai dau :v", "title": null, "password_lock": null } HTTP/1.1 201 Created
Server: Fshare
Date: Tue, 21 May 2019 02:56:19 GMT
Content-Type: application/json; charset=UTF-8
Transfer-Encoding: chunked
Connection: keep-alive
```

```
174
{ "id": "Q4uDmemqP1FCFpEjexDnGSfueKU2uviN", "url": "http://www.fsend.vn/download/
Q4uDmemqP1FCFpEjexDnGSfueKU2uviN", "title": null, "recipients": [ "duypt@uit.edu.vn" ], "message": "Khong o lai dau
:v", "status": "enabled", "is_locked": false, "is_expired": false, "total_file": 2, "total_size": "4788750", "total_dl":
0, "ctime": "2019-05-21T02:56:18+00:00", "expire_in": "2019-05-31T02:56:18+00:00" }
0
```

- Người nhận (recipient): [duypt@uit.edu.vn](mailto:duypt@uit.edu.vn)
- Thông điệp (message): “Khong o lai dau :v”
- Tiêu đề: bỏ trống (null)

## Trích xuất file:

- Xem packet đã gửi có chứa file:
- + Packet thứ 2 trong 4 packet đã fileter ở trên -> *Follow Stream*

```
POST /upload/dZFL+bxh+3-P3-GAqMhhaORkNJcYxR6ITPZLZBzywLUWX2twgbTa7ZH0tsPUJ45wPUUYvqUceOhozn46?
flowChunkNumber=1&flowChunkSize=20000000&flowCurrentChunkSize=4698321&flowTotalSize=4698321&flowIdentifier=4698321-Anh-Oi-O-Lai-
Chi-Pu-Dat-Gmp3&flowFilename=Anh-Oi-O-Lai-Chi-Pu-Dat-G.mp3&flowRelativePath=Anh-Oi-O-Lai-Chi-Pu-Dat-G.mp3&flowTotalChunks=1
HTTP/1.1
Host: up.fshare.vn
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:66.0) Gecko/20100101 Firefox/66.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://fsend.vn/
Content-Range: bytes 0-4698320/4698321
Content-Type: audio/mpeg
Content-Length: 4698321
Origin: http://fsend.vn
Connection: keep-alive

ID3.....GAPIC..9Z...image/jpeg.....JFIF.....Lavc58.14.100....C.....
....

.....!...1.A"Qa.q.2....R.#B..3rb.
$.C....S....s..T.d.....!..QA1.q".a..2.R...B.#.r..b3.....
$.s.DCS....".....?..*.j...j...j...j...j...j...+.[V.....uYj-E.*.....*....Ee.r...*.R.r...kR.
(..)...T...%I...~4W..k
```

Phần bắt đầu, (search với chuỗi chữ ký trên ***“FFD8FFE000104A46”***)

FF D8 FF DB	ÿøÛ	0	jpg jpeg	JPEG raw or in the JFIF or Exif file format <sup>[11]</sup>
FF D8 FF E0 00 10 4A 46 49 46 00 01	ÿøäNULDLEJFIFNULSOH			
FF D8 FF EE	ÿøÿi			
FF D8 FF E1 ?? ?? 45 78 69 66 00 00	ÿøä??ExifNULNUL			

[illegible]

Follow TCP Stream (tcp.stream eq 3738) · capture-output\_kb02.pcap

368 client pkts, 3 server pkts, 5 turns.

Entire conversation (4792 KB)

Show data as Raw

Stream 3738

Find Next

Filter Out This Stream

Print

Save as...

Back

Close

Help

```

83d3992af6be5984fdc73654ad7e25fa471141055e2aa7c4a598d814cd31765d62ea6f1fe83c3da3d90aa6e1cb5f48f3020c25d01eec7e528ad050a719869140
8eff06f7a170282f529935c8245cb688d1e2aa7c54da728ef6aad78800231ebf085f07781199b3e40497cb2bc7c5807b99f58b37c0439f8d7b8e0dc8665f4e7c
5a2726c95cda4508d64cca34cc5a9c0ea3a635182154bd65e653d406a42e40f9265d395db1e5cd7b730001683f540fb12c2b1035bb4fc620a7e11b8058d1094
29224a08a822580043096aeb3031442367e33d7c49be66f831407e253556b8848dd07e0ff70cdb1714a168f7b80c1778954ca9f92738db68ff003dc13184fd30
ac0d97c0c43594619b945ddad580bbb32082dcbf7362ebd43154a0f18fe6515ac3e413a2ef844aaa6e1f922d8eb02ecefbe8559dd8a74054f9f7c0c09b45dde
1931d46bca0d71b1096732d65678f24d56bb332d0ce15860d91f06522088d7508b3a5758e5f9052bedab4bd7537c895e8d4ba1d807a12aa43f80582159667633
9b5ab13ba47cf537cba0e1079abf872c377147ec0f71181409af4bfa8890a35f67f7101af70e6054a583c2387e6381f837348948bec8bf0885c1b2d6cd47cd1a
e98e0dfa2f60a8e2622b55a50415da45300d432f001452258278999c7ea7375290d12f05d4bcdcb2fa06178e2f4dc3572c0af23ecd32d482fb9c5f67d180884e
b81a145ed1ea886bfaafa201bbd623ce49624e9fc3dcf297c0b9459a783d406994529a733243473e245955051f3676618ad2d9626f7c12fa33747107c263be5c
0cd1337191062a1935c2882c5a7219a2d00efb7eb62a9676475837d65f8c9c8a1c25a27e76a15da34bc4c2b3e496643e62d6c656ed805052308d44a5d03f4
133436fe0029a420d2d0c13db0e0bba6e19168df05c128605256c43f04a41383dd31a3ea09a88e4b4794ddaf788ca0f0b15d05b87cd3e25808abd54ab82fca7
266ab4c5877563e3be08d564ebd414bac17a7f009372ae9f511abc51b644a031e6225d3632c33f57dc0133a086b86c3f242caa92b5e0616eafaf31c30f1f71d
8965c7d8b828a2403109878f10700a3f323d95e1e2cbd252ace25c6b8a7f0cb06932c7cfea4065eb5055d6b1103634f4c352df915fadd52f32facc630bc2
36d2ddefef2d2ca4a0323603e94641a545c301f8b328d92c655b21aac8115723ca16d45ac82f164a61a5a1d26c814ce6003c06b3e614c00d7f3c13a69544
3a2a7957c8313b08943765b143b0c92994ef4422f3ad236555c23df64319fccaf3b9f13ffffd9
485454507f312c3120323030204f4b0d0a5365727665723a206673686172652d6e67696e780d0a446174653a205475652c203231204d61792032303139203032
3a35363a31370474d540d0a436f6e7465674652d547970653a20746578742f706c61696e0d0a5472616e736665722d456e636f6364696e673a206368756e6b6564
0d0a436f6e6e656374696f6e3a206b6565702d616c6976650d0a50726167d6d13a206e6f2d63616368650d0a582d436f6e74655e742d547970652d4f7074696f
6e733a206e6f736e696660d0a4163636573732d436f6e74726f6c2d416c6c6f772d4f726967696e3a202a0d0a4163636573732d436f6e74726f6c2d416c6c6f
772d4d6574686f67733a2047455452c20504f53542c2050555452c204f505494f4e530d0a4163636573732d436f6e74726f6c2d416c6c6f772d48656164657273
3a20436f6e74656e742d52616e67652c20436f6e74656e742d446973706f736974696f6e2c20436f6e74656e742d547970650d0a4163636573732d436f6e7472
6f6c2d416c6c6f772d43726564656e7469616c733a20747275650d0a66732d7365727269643a203637383631313036310d0a0d0a33370d0a7b22736563
757265223a302c226e616d655223a22696d6167652e6a7067222c2268657363223a2222c2273697a65223a39303432397d0a0d0a300d0a0d0a

```



FF FB	ÿù	0	mp3	MPEG-1 Layer 3 file without an ID3 tag or with an ID3v1 tag (which is appended at the end of the file)
FF F3	ÿó			
FF F2	ÿð			
49 44 33	ID3	0	mp3	MP3 file with an ID3v2 container

[illegible]

A screenshot of a media player window. The video frame shows a woman with dark hair, wearing a red cardigan over a white top, holding a large, light-brown earthenware jar. She is looking down at the jar with a contemplative expression. The background is dark with some blurred lights. The video title 'Anh Ơi Ở Lại' is visible in the top left corner of the video frame. The media player interface includes a progress bar at the bottom, showing a duration of 0:01:56 out of 0:03:02. Standard playback controls like play/pause, stop, previous, next, and volume are visible at the bottom.

**Khoa Mạng máy tính và  
Truyền thông**





```

Wireshark · Follow TCP Stream (tcp.stream eq 2) · kb03_evidence.pcap

*...a.....E4628778....Sec558user1.....Here's the secret recipe... I just downloaded it from the file
server. Just copy to a thumb drive and you're good to go &gt;:-)....*..b.".....F.....Sec558user1..*.V.....
*.A.....E.....P.....p..p.....P.....p..p.&.'.....
.....U4.....|.....h.....p..@.&.'.....
|.....h.....p..@.&.'*.V.....E4628778....Sec558user1*.c.z.....G7174647....Sec558user1.....R..7174647.
F.CL..."DEST.....F.
.....recipe.docx.*.V.....
*.c.....G.....P.....p..p.._w.....P.....p..p.&a .....
.....U.....|.....h.....p..@.&a .....
|.....h.....p..@.&a .....*.V.....G7174647....Sec558user1*.V..{.....*..
7174647....Sec558user1.....J.H.....+..1n.....+..0.....J.....7174647.
F.CL..."DEST.....*.V..".....*.1.....Sec558user1..*.V.....*.y..N.....w.....Sec558user1.....J.H.....+..
1n.....+..0.....J.....a.....X....<HTML><BODY><FONT FACE="Arial" SIZE=2 COLOR=#000000>thanks dude</FONT></BODY></HTML>.
.....+..1n.....+..0.....*.V..".....*.Sec558user1..*.V.....+
Q.....L.....Sec558user1.....J.H.....+..1n.....+..0.....J.....S.....j....<HTML><BODY><FONT FACE="Arial"
SIZE=2 COLOR=#000000>can't wait to sell it on ebay</FONT></BODY></HTML>.
.....+..1n.....+..0.....*.V..".....+
.....Sec558user1..*.V..".....
+.....Sec558user1..*.d..".....H.....Sec558user1..*.e.J.....I5088496....Sec558user1...".....see
you in hawaii!....*.f.".....J.....Sec558user1..*.V.....
...+ @.....I.....P.....p..p..a.....P.....p..p.&.....
.....V~.....|.....h.....p..@.&.....
|.....h.....p..@.&.*.V.....I5088496....Sec558user1

```

Ta thấy mặc dù nội dung đã bị mã hóa nhưng vẫn tiết lộ được một phần nào các thông điệp đáng nghi như đoạn thông điệp:

“Here's the secret recipe... I just downloaded it from the file server. Just copy to a thumb drive and you're good to go”

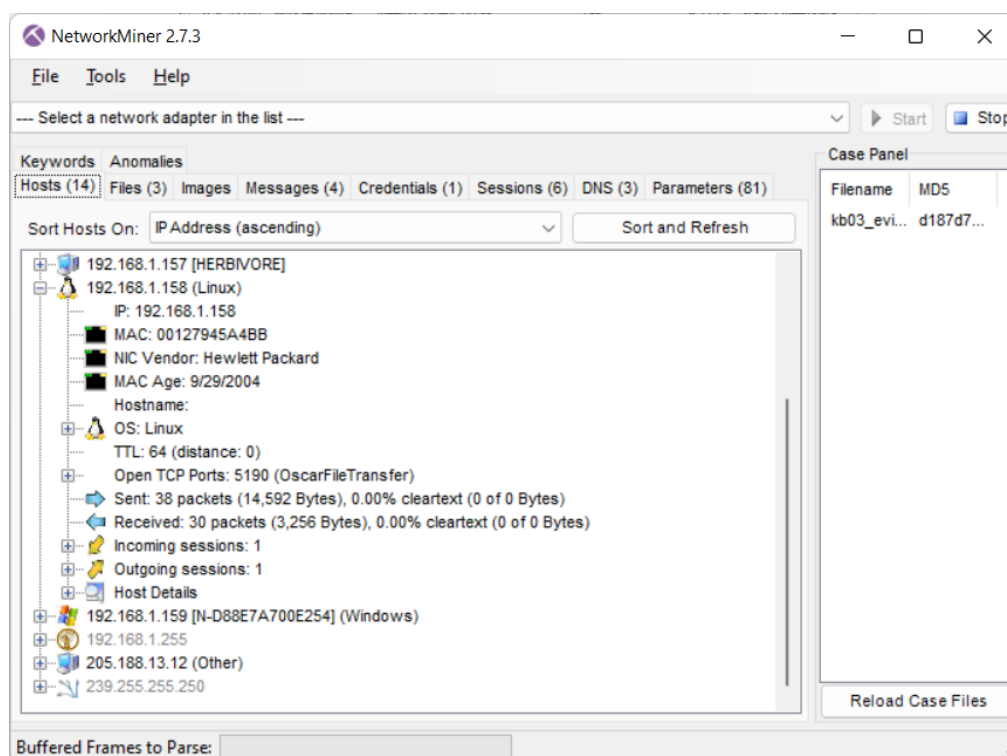
Hay tên file :

recipe.docx

Đến đây kết luận thì cũng hơi vội 😊. Chưa chắc người ta đã làm tình làm tội gì vì chưa biết được file ở trên có thực sự chứa thông tin nhạy cảm của công ty hay không.

Ở đây, mình sẽ sử dụng Tool **NetworkMiner**. Tool này khá mạnh mẽ trong việc hỗ trợ “phân tích sẵn có”, nhận đầu vào là tập tin *.pcap*, *.dump*, *.dmp*, ..... và phân tích các trường thông tin tổng quan nhưng quan trọng như *hosts*, *files*, *images*, ... xuất hiện trong file đã bắt được.

Import file đầu vào, xem thông tin với IP **192.168.1.158**.



Chuyển sang tab *Files* -> Thấy trích xuất được 3 files

Keywords		Anomalies					
Hosts (14)		Files (3)					
Filter keyword:		<input type="checkbox"/> Case sensitive         ExactPhrase         Any column         Clear         Apply					
Frame nr.	Filename	Extension	Size	Source host			
112	recipe.docx	docx	12 008 B	192.168.1.158 (Linux)			
230	size=120x90;noperf=1.html	html	375 B	64.236.68.246 [glb-at.atwola.adtechus.com]			
233	size=120x90;noperf=1.js	js	335 B	64.236.68.246 [glb-at.atwola.adtechus.com]			

Trong đó ta cần quan đến file **recipe.docx** -> Chuột phải vào file -> Chọn *Open File*. Ta thấy được nội dung file như sau:

### Recipe for Disaster:

1 serving

Ingredients:

4 cups sugar

2 cups water

In a medium saucepan, bring the water to a boil. Add sugar. Stir gently over low heat until sugar is fully dissolved. Remove the saucepan from heat. Allow to cool completely. Pour into gas tank. Repeat as necessary.

Vậy là đã đủ chứng minh thanh niên này bịp rồi 😡. Giờ thì kick ông ra khỏi công ty thôi

<https://github.com/ctfs/write-ups-2015/tree/master/csaw-ctf-2015/forensics/transfer-100>

## Statistics -> Protocol Hierachy

Ta thấy nội dung file là một chương trình mã hóa một chuỗi plaintext thành ciphertext. Và cipher ta có là chuỗi dài lộn ngược ở phía sau. Bây giờ nhiệm vụ của chúng ta là phân tích file encode và tìm cách viết file decode.

Phân tích:

Hàm chính *encode()*

```
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
```

Hàm này thực hiện đưa số 2 vào trước chuỗi plaintext, sau đó đưa vào vòng for và trong mỗi lần lặp thì chọn random một thuật toán trong mảng enc\_ciphers ['rot13', 'b64e', 'caesar']. Thực hiện đến 50 lần (vì tham số mặc định là 50).

Các hàm như **rot13**, **b64e**, **caesar** có chức năng mã hóa nhất định, tuân theo quy luật của 3 loại mã hóa cơ bản như đã biết: *ROT13*, *base64*, *Caesar*

Hàm **rot13** không cần viết lại

Hàm **base64** dùng sẵn thư viện để decode

Hàm **Caesar** decode sẽ tương đương với việc truyền tham số trái dấu so với hàm encode

Build hàm decode, việc decode thực hiện tương tự nhưng nghịch đảo lại.

**\*Note:** Vì số random trong thuật toán encode lộn ngược nên bản chất ciphertext vẫn chứa số random này

Lưu ciphertext vào một file khác *ciphertext.txt*. Chương trình decode hoàn chỉnh:

```
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(
        "ABCDEFGHIJKLMNOPQRSTUVWXYZnopqrstuvwxyz",
        "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 ký tự.
- Công cụ: Wireshark, tshark,...

**Gợi ý:** <https://github.com/ctfs/write-ups-2015/tree/master/nuit-du-hack-ctfquals-2015/forensic/private>

Thực hiện khảo sát sơ lược có các protocols nào: *Statistics -> Protocol Hierarchy*

Protocol	Percent Packets	Packets	Percent Bytes	Bytes	Bits/s	End Packets	End Bytes	End Bits/s
▼ Frame	100.0	553	100.0	36749	360	0	0	0
▼ Ethernet	100.0	553	21.1	7742	75	0	0	0
▼ Logical-Link Control	76.5	423	55.1	20246	198	0	0	0
Spanning Tree Protocol	74.0	409	39.0	14315	140	409	14315	140
Cisco Discovery Protocol	2.5	14	12.5	4592	45	14	4592	45
▼ Internet Protocol Version 6	1.1	6	0.7	240	2	0	0	0
Internet Control Message Protocol v6	1.1	6	0.3	128	1	6	128	1
▼ Internet Protocol Version 4	19.9	110	6.0	2200	21	0	0	0
▼ User Datagram Protocol	3.6	20	0.4	160	1	0	0	0
Domain Name System	3.6	20	1.6	590	5	20	590	5
Transmission Control Protocol	0.2	1	0.1	33	0	0	0	0
Internet Control Message Protocol	16.1	89	4.2	1552	15	89	1552	15
Data	0.5	3	0.4	139	1	3	139	1
Address Resolution Protocol	2.2	12	1.2	444	4	12	444	4

Ở đây ta thấy có khá nhiều giao thức: LLC, STP, Ipv6, ARP, ... Để có nói chủ đề đến các gói **ICMP**, và flag có định dạng bắt đầu bằng chuỗi "**S3**". Vậy ta có filter sau:

```
icmp matches "^S3"
```

Kết quả không khả thi, không trả về gói tin nào. Tính mở bằng Network Miner mà thấy đuôi là pcapng (chỉ có bản Pro mới "chơi" được) nên thôi. 😞

Thử loay hoay tìm các giao thức khác (DNS, HTTP) thì thấy thẳng TCP có duy nhất 1 gói tin

tcp						
No.	Time	Source	Destination	Protocol	Length	Info
388	629.148459	192.168.0.50	192.168.50.10	TCP		67 4242 → 9000 [SYN] Seq=0 Win=8192 Len=13

Ở đây ta thấy IP Source là **192.168.0.50** gửi gói tin đến **192.168.50.10**.

**\*Note:** Ở đây nếu ta xem nội dung gói tin TCP trên ta sẽ thấy có một dòng chữ

**CDAISIWILLWIN** . Search thông điệp này ra ta sẽ có mấy bài write-up :v

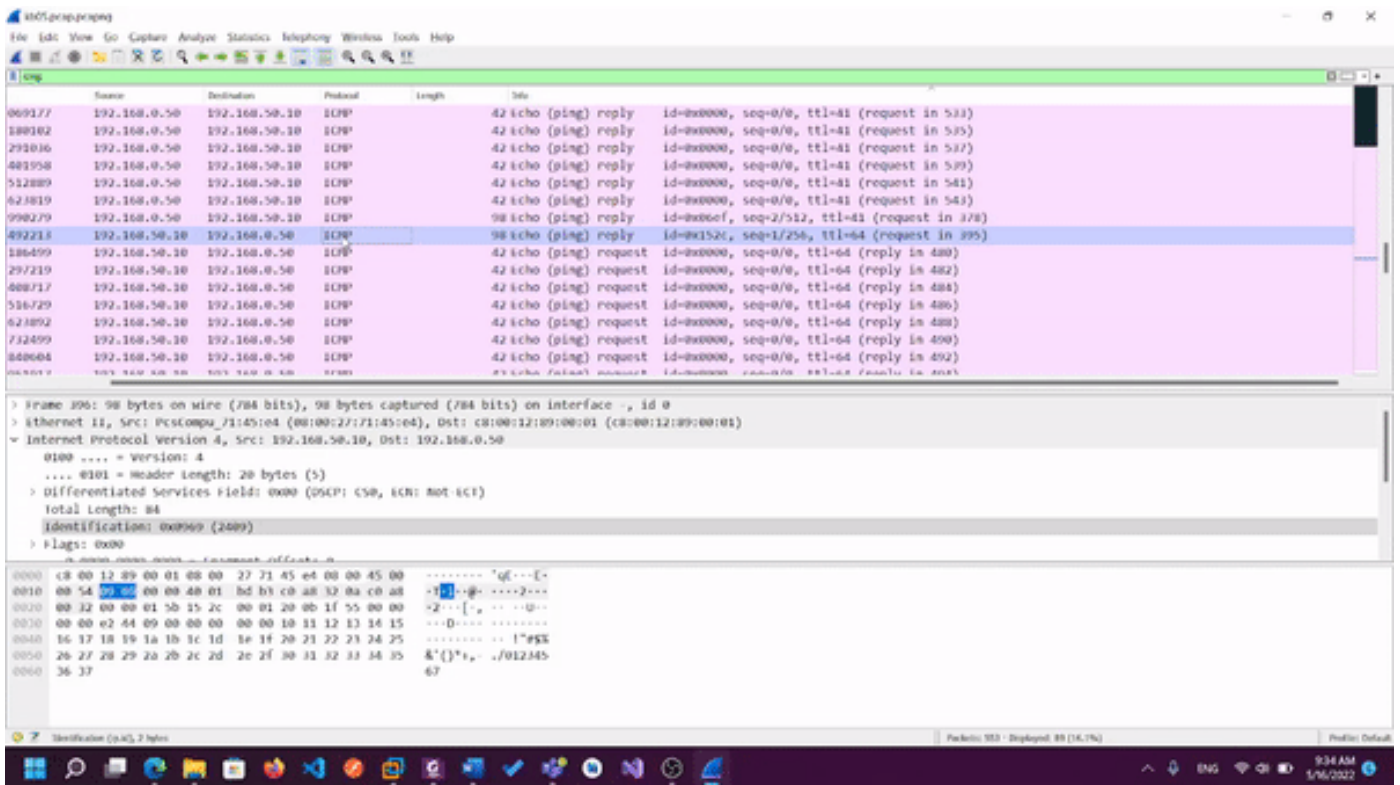
Không có manh mối gì khác, ta thử tập trung vào các gói **ICMP**

```
icmp
```

icmp						
No.	Time	Source	Destination	Protocol	Length	Info
376	616.966522	192.168.50.10	192.168.0.50	ICMP	98	Echo (ping) request id=0x06ef, seq=1/256, ttl=64 (no response found!)
378	617.965929	192.168.50.10	192.168.0.50	ICMP	98	Echo (ping) request id=0x06ef, seq=2/512, ttl=64 (reply in 379)
396	641.492213	192.168.50.10	192.168.0.50	ICMP	98	Echo (ping) reply id=0x152c, seq=1/256, ttl=64 (request in 395)
479	796.186499	192.168.50.10	192.168.0.50	ICMP	42	Echo (ping) request id=0x0000, seq=0/0, ttl=64 (reply in 480)
481	796.297219	192.168.50.10	192.168.0.50	ICMP	42	Echo (ping) request id=0x0000, seq=0/0, ttl=64 (reply in 482)
483	796.408717	192.168.50.10	192.168.0.50	ICMP	42	Echo (ping) request id=0x0000, seq=0/0, ttl=64 (reply in 484)
485	796.516729	192.168.50.10	192.168.0.50	ICMP	42	Echo (ping) request id=0x0000, seq=0/0, ttl=64 (reply in 486)
487	796.623892	192.168.50.10	192.168.0.50	ICMP	42	Echo (ping) request id=0x0000, seq=0/0, ttl=64 (reply in 488)
489	796.732499	192.168.50.10	192.168.0.50	ICMP	42	Echo (ping) request id=0x0000, seq=0/0, ttl=64 (reply in 490)
491	796.840604	192.168.50.10	192.168.0.50	ICMP	42	Echo (ping) request id=0x0000, seq=0/0, ttl=64 (reply in 492)
493	796.951917	192.168.50.10	192.168.0.50	ICMP	42	Echo (ping) request id=0x0000, seq=0/0, ttl=64 (reply in 494)
495	797.062706	192.168.50.10	192.168.0.50	ICMP	42	Echo (ping) request id=0x0000, seq=0/0, ttl=64 (reply in 496)
497	797.172685	192.168.50.10	192.168.0.50	ICMP	42	Echo (ping) request id=0x0000, seq=0/0, ttl=64 (reply in 498)
499	797.283743	192.168.50.10	192.168.0.50	ICMP	42	Echo (ping) request id=0x0000, seq=0/0, ttl=64 (reply in 500)
501	797.389133	192.168.50.10	192.168.0.50	ICMP	42	Echo (ping) request id=0x0000, seq=0/0, ttl=64 (reply in 503)
504	797.500470	192.168.50.10	192.168.0.50	ICMP	42	Echo (ping) request id=0x0000, seq=0/0, ttl=64 (reply in 505)

Ta để ý trường **Id** có gì đó lạ, giống như đang ẩn giấu thông tin gì đó - Ta lần lượt lướt các gói tin từ trên xuống dưới và để ý giá trị ở trường **Identifications** thay đổi liên tục và tạo thành message nào đó.





Dùng tshark để trích xuất thông tin & nhìn rõ hơn:

```
tshark -r kb05.pcap.pcapng -x 'icmp and ip.src==192.168.50.10'
```

```
0010 00 1c 00 22 00 00 40 01 c7 32 c0 a8 32 0a c0 a8 ... ..@..2..2...
0020 00 32 08 00 f7 ff 00 00 00 00 ... .2.....

0000 c8 00 12 89 00 01 08 00 27 71 45 e4 08 00 45 00 ..... 'qE... E.
0010 00 1c 00 68 00 00 40 01 c6 ec c0 a8 32 0a c0 a8 ... h..@.....2...
0020 00 32 08 00 f7 ff 00 00 00 00 ... .2.....

0000 c8 00 12 89 00 01 08 00 27 71 45 e4 08 00 45 00 ..... 'qE... E.
0010 00 1c 00 65 00 00 40 01 c6 ef c0 a8 32 0a c0 a8 ... e..@.....2...
0020 00 32 08 00 f7 ff 00 00 00 00 ... .2.....

0000 c8 00 12 89 00 01 08 00 27 71 45 e4 08 00 45 00 ..... 'qE... E.
0010 00 1c 00 72 00 00 40 01 c6 e2 c0 a8 32 0a c0 a8 ... r..@.....2...
0020 00 32 08 00 f7 ff 00 00 00 00 ... .2.....

0000 c8 00 12 89 00 01 08 00 27 71 45 e4 08 00 45 00 ..... 'qE... E.
0010 00 1c 00 65 00 00 40 01 c6 ef c0 a8 32 0a c0 a8 ... e..@.....2...
0020 00 32 08 00 f7 ff 00 00 00 00 ... .2.....

0000 c8 00 12 89 00 01 08 00 27 71 45 e4 08 00 45 00 ..... 'qE... E.
0010 00 1c 00 20 00 00 40 01 c7 34 c0 a8 32 0a c0 a8 ... ..@..4..2...
0020 00 32 08 00 f7 ff 00 00 00 00 ... .2.....

0000 c8 00 12 89 00 01 08 00 27 71 45 e4 08 00 45 00 ..... 'qE... E.
0010 00 1c 00 69 00 00 40 01 c6 eb c0 a8 32 0a c0 a8 ... i..@.....2...
0020 00 32 08 00 f7 ff 00 00 00 00 ... .2.....

0000 c8 00 12 89 00 01 08 00 27 71 45 e4 08 00 45 00 ..... 'qE... E.
```

Ta dễ ý thấy các chữ cái “h”, “e”, “r”, “e”, “i”, ... tạo thành một thông điệp và ta muốn lấy ở dòng có offset “0010” nên mình sẽ **grep** chuỗi này

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

- “0010” là giá trị version của IP (IPv4) tại Tầng IP (Layer 3) trong gói tin

```
> Frame 481: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface -, id 0
> Ethernet II, Src: PcsCompu_71:45:e4 (08:00:27:71:45:e4), Dst: c8:00:12:89:00:01 (c8:00:12:89:00:01)
√ Internet Protocol Version 4, Src: 192.168.50.10, Dst: 192.168.0.50
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
> Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
  Total Length: 28
  Identification: 0x0068 (104)
> Flags: 0x00
  ...0 0000 0000 0000 = Fragment Offset: 0
  Time to Live: 64
  Protocol: ICMP (1)
  Header Checksum: 0xc6ec [validation disabled]
```

- Filter theo IP Source là **192.168.50.10** vì các gói tin này mang lại sự thay đổi và có thể tạo thành thông điệp như phân tích trên.

```
(virus@virus)-[~/Desktop]
$ tshark -r kb05.pcap.pcapng -x 'icmp and ip.src==192.168.50.10' | grep 0010
0010 00 38 00 0e 00 00 ff 01 d6 5a c0 a8 32 01 c0 a8 .8.....Z..2...
0010 00 38 00 0f 00 00 ff 01 d6 59 c0 a8 32 01 c0 a8 .8.....Y..2...
0010 00 38 00 10 00 00 ff 01 d6 58 c0 a8 32 01 c0 a8 .8.....X..2...
0010 00 38 00 11 00 00 ff 01 d6 57 c0 a8 32 01 c0 a8 .8.....W..2...
0010 00 38 00 12 00 00 ff 01 d6 56 c0 a8 32 01 c0 a8 .8.....V..2...
0010 00 38 00 13 00 00 ff 01 d6 55 c0 a8 32 01 c0 a8 .8.....U..2...
0010 00 38 00 14 00 00 ff 01 d6 54 c0 a8 32 01 c0 a8 .8.....T..2...
0010 00 38 00 15 00 00 ff 01 d6 53 c0 a8 32 01 c0 a8 .8.....S..2...
0010 00 38 00 16 00 00 ff 01 d6 52 c0 a8 32 01 c0 a8 .8.....R..2...
0010 00 38 00 17 00 00 ff 01 d6 51 c0 a8 32 01 c0 a8 .8.....Q..2...
0010 00 38 00 18 00 00 ff 01 d6 50 c0 a8 32 01 c0 a8 .8.....P..2...
0010 00 38 00 19 00 00 ff 01 d6 4f c0 a8 32 01 c0 a8 .8.....O..2...
0010 00 38 00 1a 00 00 ff 01 d6 4e c0 a8 32 01 c0 a8 .8.....N..2...
0010 00 38 00 1b 00 00 ff 01 d6 4d c0 a8 32 01 c0 a8 .8.....M..2...
0010 00 38 00 1c 00 00 ff 01 d6 4c c0 a8 32 01 c0 a8 .8.....L..2...
0010 00 38 00 1d 00 00 ff 01 d6 4b c0 a8 32 01 c0 a8 .8.....K..2...
0010 00 38 00 1e 00 00 ff 01 d6 4a c0 a8 32 01 c0 a8 .8.....J..2...
0010 00 38 00 1f 00 00 ff 01 d6 49 c0 a8 32 01 c0 a8 .8.....I..2...
0010 00 38 00 20 00 00 ff 01 d6 48 c0 a8 32 01 c0 a8 .8.....H..2...
0010 00 38 00 21 00 00 ff 01 d6 47 c0 a8 32 01 c0 a8 .8.....G..2...
0010 00 54 00 00 40 00 40 01 87 1c c0 a8 32 0a c0 a8 .T..@..2...
0010 00 54 00 00 40 00 40 01 87 1c c0 a8 32 0a c0 a8 .T..@..2...
0010 00 54 09 69 00 00 40 01 bd b3 c0 a8 32 0a c0 a8 .T.i..@..2...
0010 00 1c 00 22 00 00 40 01 c7 32 c0 a8 32 0a c0 a8 ... " ..@..2...
0010 00 1c 00 68 00 00 40 01 c6 ec c0 a8 32 0a c0 a8 ... h..@..2...
0010 00 1c 00 65 00 00 40 01 c6 ef c0 a8 32 0a c0 a8 ... e..@..2...
0010 00 1c 00 72 00 00 40 01 c6 e2 c0 a8 32 0a c0 a8 ... r..@..2...
0010 00 1c 00 65 00 00 40 01 c6 ef c0 a8 32 0a c0 a8 ... e..@..2...
0010 00 1c 00 20 00 00 40 01 c7 34 c0 a8 32 0a c0 a8 ... ..@..4..2...
```

Flag: S3cr3t4g3nt

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

- Mô tả: Một trong các máy chủ của CoMix Wave Films bị xâm nhập vào tuần trước, tuy nhiên không có thiệt hại đáng kể nào được ghi nhận. Mặc dù hệ thống tường lửa của công ty rất mạnh nhưng nhóm bảo mật của công ty phát hiện ra một số hoạt động đáng ngờ, có thể bị tuồn dữ liệu ra bên ngoài. Hãy điều tra liệu kẻ tấn công đã lấy được những gì từ máy chủ của công ty, giao thức sử dụng? Tìm flag.

- Tài nguyên: Nandemonaiya\_kb06.pcap

**Yêu cầu – Gợi ý:** <https://bitbucket.org/kscrivs/netsec-0x325->

Mở file pcap bằng Wireshark, ta thấy trước mắt là rất nhiều gói tin DNS

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	192.168.196.133	192.168.196.1	DNS	78	Standard query 0xf78d A QXQgdGhl.evil.corp
2	0.000548176	192.168.196.1	192.168.196.133	DNS	94	Standard query response 0xf78d A QXQgdGhl.evil.corp A 192.168.196.1
3	0.566375034	192.168.196.133	192.168.196.1	DNS	78	Standard query 0xf3b3 A IG5leHQg.evil.corp
4	0.567084741	192.168.196.1	192.168.196.133	DNS	94	Standard query response 0xf3b3 A IG5leHQg.evil.corp A 192.168.196.1
5	1.114627235	192.168.196.133	192.168.196.1	DNS	78	Standard query 0xaaeb A c3RvcCwg.evil.corp
6	1.115214936	192.168.196.1	192.168.196.133	DNS	94	Standard query response 0xaaeb A c3RvcCwg.evil.corp A 192.168.196.1
7	1.654834038	192.168.196.133	192.168.196.1	DNS	78	Standard query 0x4efe A SSBzcHJp.evil.corp
8	1.656062752	192.168.196.1	192.168.196.133	DNS	94	Standard query response 0x4efe A SSBzcHJp.evil.corp A 192.168.196.1
9	2.174593281	192.168.196.133	192.168.196.1	DNS	78	Standard query 0x3e9a A bnQgb2Zm.evil.corp
10	2.175216326	192.168.196.1	192.168.196.133	DNS	94	Standard query response 0x3e9a A bnQgb2Zm.evil.corp A 192.168.196.1
11	2.707788617	192.168.196.133	192.168.196.1	DNS	78	Standard query 0x541c A IHROZSB0.evil.corp
12	2.710979196	192.168.196.1	192.168.196.133	DNS	94	Standard query response 0x541c A IHROZSB0.evil.corp A 192.168.196.1
13	3.247521328	192.168.196.133	192.168.196.1	DNS	78	Standard query 0x236c A cmFpbiBh.evil.corp
14	3.248162372	192.168.196.1	192.168.196.133	DNS	94	Standard query response 0x236c A cmFpbiBh.evil.corp A 192.168.196.1
15	3.808772388	192.168.196.133	192.168.196.1	DNS	78	Standard query 0x59ef A bmQgc3Rh.evil.corp

Điều đáng ngờ ở đây là các query DNS trả về tên miền **.evil.corp** (nghe như trong phim *Mr. Robot* @.@). Ta thấy có chuỗi trông giống base64 encode trên tên miền. Thử một chuỗi đầu xem sao:

```
(virus@virus)-[~]
$ echo "QXQgdGhl" | base64 -d
At the
```

Vậy là một chuỗi có ý nghĩa, dùng tshark trích xuất ra toàn bộ rồi decode vài cái thử :

- Port 53 – DNS
- -2 filter qua 2 giai đoạn vào buffer
- dns.qry.name: lấy tên miền dns được query ra

```
tshark -r Nandemonaiya_kb06.pcapng -2 -R udp.dstport==53 -T fields -e "dns.qry.name" | grep "evil.corp"
```

```
(virus@virus)-[~/Desktop]
$ tshark -r Nandemonaiya_kb06.pcapng -2 -R udp.dstport==53 -T fields -e "dns.qry.name" | grep "evil.corp"
QXQgdGhl.evil.corp
IG5leHQg.evil.corp
c3RvcCwg.evil.corp
SSBzcHJp.evil.corp
bnQgb2Zm.evil.corp
IHROZSB0.evil.corp
cmFpbiBh.evil.corp
bmQgc3Rh.evil.corp
cnQgc3Vu.evil.corp
bmluZyB3.evil.corp
aWxkbHkg.evil.corp
YXJvdW5k.evil.corp
IHROZSBz.evil.corp
dHJlZXRz.evil.corp
LCBzZWYy.evil.corp
Y2hpbmcg.evil.corp
Zm9yIGhl.evil.corp
ci4gSSBr.evil.corp
bm93IHRO.evil.corp
```

Thử decode base64 tiếp

```
(virus@virus)-[~]
$ echo "IG5leHQg" | base64 -d
next
```

OK, vậy giờ mình sẽ viết một payload command để decode hết những chuỗi trên một lúc. Đưa output tên domain vào 1 file riêng biệt (**encoded\_domain.txt**)

```
tshark -r Nandemonaiya_kb06.pcapng -2 -R udp.dstport==53 -T fields -e "dns.qry.name" | grep "evil.corp" > encoded_domain.txt
```

Do output như mình thấy bên trên thì chuỗi base64 đều có length giống nhau trong các domain, vậy nên cũng tiện. Ta dùng command **cut**:

```
(virus@virus)-[~/Desktop]
$ cut -b 1-8 encoded_domain.txt
QXQgdGhl
IG5leHQg
c3RvcCwg
SSBzcHJp
bnQgb2Zm
IHROZSB0
cmFpbjBh
bmQgc3Rh
cnQgcjVu
bmluZyB3
aWxkbHkg
YXJvdW5k
IHROZSBz
dHJlZXRz
LCBzZWYy
Y2hpbmcg
Zm9yIGhl
ci4gSSBr
bm93IHRO
YXQgc2hl
```

Ở đây mình sẽ đưa các giá trị này vào file *base64\_strings.txt*:

```
cut -b 1-8 encoded_domain.txt > base64_strings.txt
```

Sau đó decode từng dòng trên file này:

```
cat base64_strings.txt | base64 -d
```

```
(virus@virus)-[~/Desktop]
$ cat base64_strings.txt | base64 -d
At the next stop, I sprint off the train and start running wildly around the streets, searching for her. I know that
she is searching for me right now in the same way.
CSACTF{
We had met before. Or maybe that was just a feeling. Just a dream. A delusion from a past life. But still, we had wa
nted to be together for just a little longer. We want to be together for just a little longer.
Sorry_
As I sprint up a hilly road, I wonder. Why am I running? Why am I looking for him? Somewhere deep down, I probably a
lready know the answers to those questions. My mind doesn't remember them, but my body does. I turn out of a thin al
ley and the road abruptly ends. A staircase. I walk up to the edge and look down. He is there.
f0r_
Fighting back the urge to burst out running, I slowly make my way up the stairs. A wind blows by, carrying the scent
of flowers and puffing up my suit. She is standing at the top. Unable to look at her directly, I turn my head just
close enough so that her presence registers in my peripheral vision. That presence begins to walk down the stairs. H
er footsteps ring throughout the spring air. My heart dances wildly within my ribcage.
sp0lling!_
We slowly draw closer to each other, our eyes cast down. He says nothing, and I too fail to find any words. Still re
maining silent, we pass each other. In that moment, my entire body aches as if someone had reached in and grabbed my
heart. This is not right, I think strongly. There is no way that we are strangers. That would go against all the la
ws of the universe and of life.
If_y0u_h4ve_n0t,_
So I turn around. With the exact same speed, she too turns around and looks at me. She is standing on the stairs, ey
es open wide, the city of Tokyo behind her back. I notice that her hair is tied with a string the color of sunset. M
y entire body shakes.
g0_
We met. We finally met. By the time I think that I'm about to cry, tears have already started falling. He sees that
and smiles. I return the smile as I weep, and take a deep breath of the fresh spring air.
w4tch_1t!}
And then, at the same time, we open our mouths, harmonizing our voices like children doing a cheer.

"Your name?"
```

Flag là các segment rời rạc được chèn vào và đứng thành một dòng riêng lẻ, đó cũng là một lời xin lỗi muộn màng tương tự. Nhưng mà chúng ta đảm bảo rằng attacker sẽ bị vô tù 💥

Flag: **CSACTF{S0rry\_f0r\_sp0llng!\_1f\_y0u\_h4ve\_n0t,\_g0\_w4tch\_1t!}**

**HẾT**