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.N21.ATCL.1

STT	Họ và tên	MSSV	Email
1	Lê Viết Tài Mẫn	20521593	20521593@gm.uit.edu.vn
2	Vũ Hoàng Thạch Thiết	20521957	20521957@gm.uit.edu.vn
3	Hoàng Thanh Lâm	20521513	20521513@gm.uit.edu.vn

2. NỘI DUNG THỰC HIỆN:[1]

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

1. Kịch bản 04

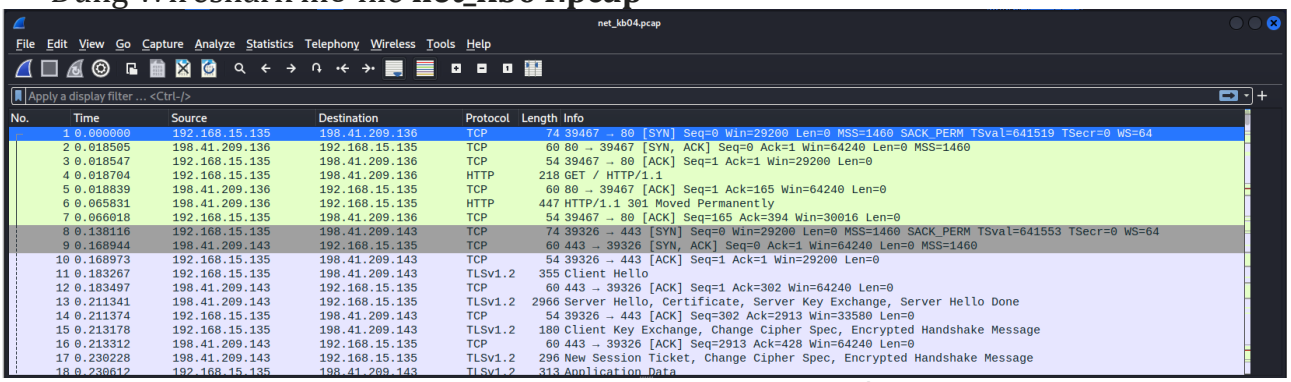
Kịch bản 04. Điều tra trên dữ liệu lưu lượng mạng thu được.

- 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{...}

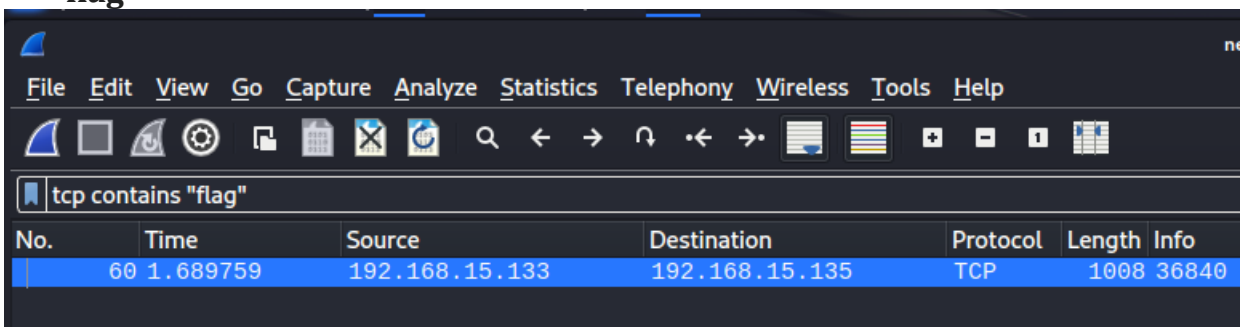
Đáp án:

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

- Dùng Wireshark mở file **net_kb04.pcap**



- Ta sẽ tìm flag trong các gói tin có giao thức TCP trước bằng filter **tcp contains "flag"**



- Đọc nội dung gói tin này:

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

FLAG = 'flag{xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx}'

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

def rot13(s):
        _rot13 = string.maketrans(
        "ABCDEFGHIJKLMabcdefghijklmNOPQRSTUVWXYZnopqrstuvwxyz",
        "NOPQRSTUVWXYZnopqrstuvwxyzABCDEFGHIJKLMabcdefghijklm")
        return string.translate(s, _rot13)

def b64e(s):
        return b64encode(s)

def caesar(plaintext, shift=3):
    alphabet = string.ascii_lowercase
    shifted_alphabet = alphabet[shift:] + alphabet[:shift]
    table = string.maketrans(alphabet, shifted_alphabet)
    return plaintext.translate(table)

def encode(pt, cnt=50):
        tmp = '2{}'.format(b64encode(pt))
        for cnt in xrange(cnt):
                c = random.choice(enc_ciphers)
                i = enc_ciphers.index(c) + 1
                _tmp = globals()[c](tmp)
                tmp = '{}{}'.format(i, _tmp)

        return tmp

if __name__ == '__main__':
        print encode(FLAG,
cnt=?)2Mk16Sk5iakYxVFZoS1RsWnZXbFZaYjFaa1prWmFkMDVWVGs1U2IyODFXa1ZuTUZadU1YVldiVkphVFVaS1dGWXlkbUZXTVdkMVprWnJWMlZHY
```

- File này là một chương trình python có chức năng mã hóa một chuỗi **plaintext**. Và chuỗi sau khi mã hóa là "2Mk…"

```
cnt=?)2Mk16Sk5iakYxVFZoS1RsWnZXbFZaYjFaa1prWmFkMDVWVGs1U2IyODFXa1ZuTUZadU1YVldiVkphVFVaS1dGWXlkbUZXTVdkMVprWnJWMlZHY
zFsWGJscHVVekpOWVZaeFZsUmxWMnR5VkZabU5HaFdaM1pYY0hkdVRXOWFSMVJXYTA5V1YwcElhRVpTVm1WSGExUldWbHBrWm05dk5sSnZVbXhTVm5OW
VZtNW1NV1l4V1dGVWJscFVaWEJoVjFsdVdtUm5iMUpYVjNGS2IxWlViMWhXVnpFd1YwWktkbVpGWVZkbFIxRXdWa1JHVDJZeFRuWlhjRzlUWlZkclkxW
lhZVk5TYmpGSFZsaHJaRkpVYjFOWmJsVXhWakZTVjFkd09WaFNNRlkyVmxjMVIxWldXa1pUY0d0WFpVWnpZbHBYWVhwVFYwWkhWM0JyVG1Wd2EydFdjS
E5MVFVkSllsWnVaMWRsYm5OWldWUk9VMlV4VWxkWGJuZFVWbTl6V0Zad2QyNWtWbHAxVGxabldsWldV0ZXYmxwTFZqRk9kV1pHYTJ0a01YTlpWbGN4W
TJoR1NsZGxNM05yVW00MVdGbFVUa05OVmxwWVprVk9iV1ZXUmpWV2NIZHlaRzlLV0doRk9WVldjRkpVVm5CaFZtaEdaM1ZuUm1kVFpHTldXRlp3TVhwV
k1XZDNVMjl2Vm1ReVVsWldiMmRUVlVaV2RGSndkMjFsU0VKSFZHOWFibFV4V25oUmJqRlhWak5yV0ZSdVdsTldNVko0Vm05T2EwMXdhMVpXY0dGdVpUR
lJZVmR4VWs5V1ZUVllWWEIzWkZkR2JucFdjSGRYWlZWYVlsWXlkalZXY0VwSFYzRmFWV1ZHY3pOWk1tRlhabkJLU0dSRk5WZE5WWE5JVm05U1IxWXlUb
lZOVm1kVVpXNWFWVmxVUm1SVU1WbDZWbkZhVGxKd1VsbFpNRlp1VlhCS1JrNVZiMXBOUjJ0TVdWY3hTMmRIVmtaYVJtZHJaREJ6Y2xaVVJtUldjRlpJV
TI1YVdHUmpWbFZWYjFwNlVtOWFWMWR2WjJ4bFZrWTFWWEExUzFkSFJXSmtSazVYWlZock0xbFZXbVJXTWtaSlZHOVdVMlF6UW1SWFYzZFhaekZaZWsxV
ldsaGxSa3BYVkZablUxTkdXa2hvUm1kWFRWWktNVlp3WVhKa1ZrcFlaMk5DVjFaRmNucGFSRUV4VTBaYWRtUkdVbXhsVjJ0WVYxY3hOOmxXWjFkV2NVc
FhaVlJXVUZad1lYcFNNVnAyWkVkM1YyUmpSa2xXVjNkeVZuQkdkVkpVUWxWV00ydFFWbkF4UjFKV1ZuZGtSMnRPWlZaRllsWXhVa2RsTVZsaVZXOXJWM
lZ2V2xoWlZFSjZabFp2VlZOd09VOVNiMWt5VlZjMVMyUXdNVlpPVlc5YVRVZFNTRlpVUVdGU01WcFpaRVphYkZkRlNrMVdSbHBrVkRGYWRsZHhSbFZsU
mxwUFZtMUdTMU5XWjNaV2IxcHZVbTV6WTFVeGEzWlZNa1Y2WmtaQ1ZtVkhVVEJWWTBaYWFGZE9SbFJ2Vmxka1kxWktWakozWkdVeFdrZFhjVXB0VWxSV
1pGbHVXbVJuVm5ORlVuRmFiMUp2U2pCVmNHRnVWVEF3WVZOdU5WZFdSVzVoV1cxQllWTkdTbmhXYjBwWVVqTnJXRlpHV21SWlZrNVhWVzluVjJWdU5Wa
FVWM2Q2VFZadWVsVnVaMWhTYjNOWVdYRnpTMWxXU25abVNFcFdUWEZPTkZVd1dsTm1NWE5IVkhCclRtVndhMUZXY0RCaFRrZFJZVlZ1WjJ0Tk1sSnlWV
```

- Dựa vào file mã hóa ta sẽ tạo file giải mã để giải mã đoạn trên:

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

FLAG = open("chuoi_ma_hoa.txt").read()
dec_ciphers = ['rot13', 'b64d', 'caesard']

def rot13(s):
        _rot13 = string.maketrans(
        "ABCDEFGHIJKLMabcdefghijklmNOPQRSTUVWXYZnopqrstuvwxyz",
        "NOPQRSTUVWXYZnopqrstuvwxyzABCDEFGHIJKLMabcdefghijklm")
        return string.translate(s, _rot13)

def b64d(s):
        return b64decode(s)

def caesar(ciphertext, shift):
    alphabet = string.ascii_lowercase
    shifted_alphabet = alphabet[shift:] + alphabet[:shift]
    table = str.maketrans(alphabet, shifted_alphabet)
    return str(ciphertext).translate(table)

def caesard(ciphertext, shift=3):
        return caesar(ciphertext, shift=-shift)

def decode(ct):
    while True:
        try:
            i = int(ct[0]) - 1
            i = i % 3
        except:
            print(ct)
            exit(0)
        ct = ct[1:]
        c = dec_ciphers[i]
        _ct = globals()[c](ct)
        ct = _ct
```

- Thực thi file với input là file chuoi_ma_hoa chứa đoạn cần giải mã trên, sau khi thực thi thì ta có được flag là **flag{li0ns_and_tig3rs_4nd_b34rs_0h_mi}**

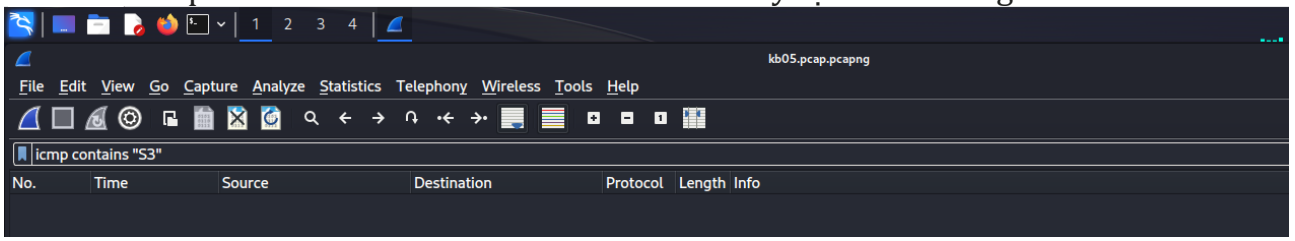
2. Kịch bản 05

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.
- Công cụ: Wireshark, tshark,...

Gợi ý: https://github.com/ctfs/write-ups-2015/tree/master/nuit-du-hack-ctf-quals-2015/forensic/private

- Dùng Wireshark mở file kb05.gz, theo đề có nói thì chú ý gói ICMP nên ta sẽ thử filter các packet ICMP có chứa từ khóa "S3" là 2 ký tự đầu của flag



- Kết quả không có gói tin nào trả về cả nên ta sẽ thử dùng tshark để khai thác các gói tin ICMP xem sao

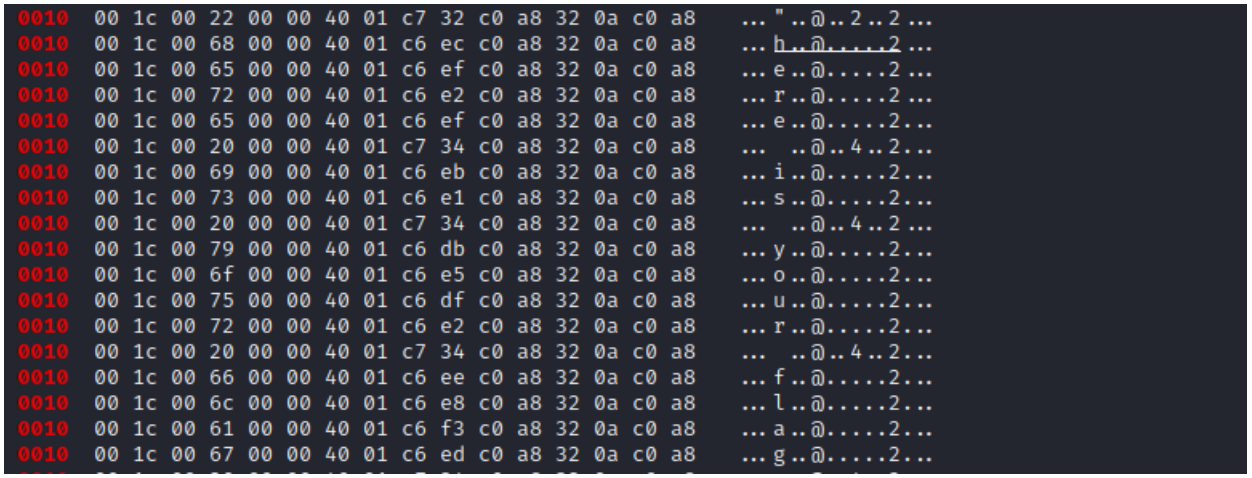
```
┌──(kali㉿kali)-[~/Downloads]
└─$ tshark -r kb05.pcap.pcapng -x 'icmp and ip.src==192.168.50.10'
0000  08 00 27 71 45 e4 c8 00 12 89 00 01 08 00 45 00   ..'qE.........E.
0010  00 38 00 0e 00 00 ff 01 d6 5a c0 a8 32 01 c0 a8   .8.......Z..2...
0020  32 0a 03 01 1e 74 00 00 00 00 45 00 00 41 ed 64   2....t....E..A.d
0030  40 00 3f 11 99 86 c0 a8 32 0a ac 10 15 fe ac 33   @.?.....2......3
0040  00 35 00 2d 31 f5                                 .5.-1.

0000  08 00 27 71 45 e4 c8 00 12 89 00 01 08 00 45 00   ..'qE.........E.
0010  00 38 00 0f 00 00 ff 01 d6 59 c0 a8 32 01 c0 a8   .8.......Y..2...
0020  32 0a 03 01 1e 74 00 00 00 00 45 00 00 41 ed 65   2....t....E..A.e
0030  40 00 3f 11 99 85 c0 a8 32 0a ac 10 15 fe ac 33   @.?.....2......3
0040  00 35 00 2d 31 f5                                 .5.-1.

0000  08 00 27 71 45 e4 c8 00 12 89 00 01 08 00 45 00   ..'qE.........E.
0010  00 38 00 10 00 00 ff 01 d6 58 c0 a8 32 01 c0 a8   .8.......X..2...
0020  32 0a 03 01 61 61 00 00 00 00 45 00 00 32 f7 2a   2...aa....E..2.*
0030  40 00 3f 11 8f cf c0 a8 32 0a ac 10 15 fe b3 5a   @.?.....2......Z
0040  00 35 00 1e e7 ef                                 .5....
```

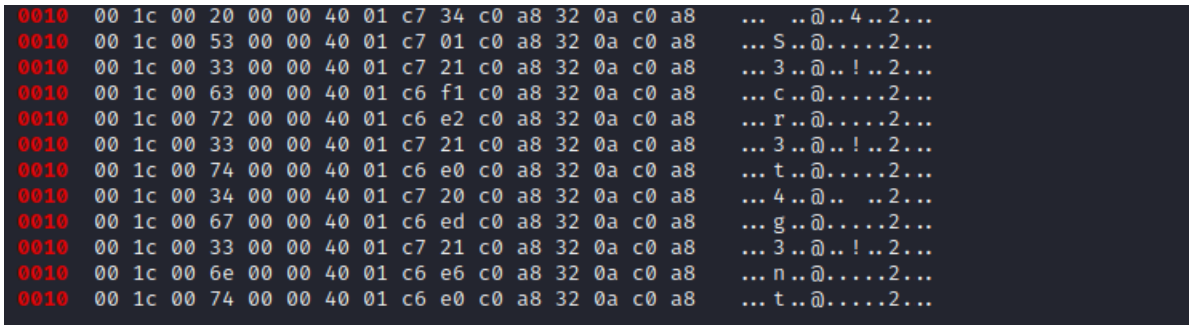
- Lướt xem kết quả trên thì ta để ý có một đoạn xuất hiện chữ flag theo hàng dọc nếu xét các dòng có offset 0010

```
0000  c8 00 12 89 00 01 08 00 27 71 45 e4 08 00 45 00   ........'qE...E.
0010  00 1c 00 66 00 00 40 01 c6 ee c0 a8 32 0a c0 a8   ...f..@.....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 6c 00 00 40 01 c6 e8 c0 a8 32 0a c0 a8   ...l..@.....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 61 00 00 40 01 c6 f3 c0 a8 32 0a c0 a8   ...a..@.....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 67 00 00 40 01 c6 ed c0 a8 32 0a c0 a8   ...g..@.....2...
0020  00 32 08 00 f7 ff 00 00 00 00                     .2........
```

- Vậy ta sẽ thêm grep “0010” vào lệnh trước để chỉ xem các giá trị của dòng có offset 0010 và xem chúng kết hợp với nhau có xuất hiện flag hay không?

```
┌──(kali㉿kali)-[~/Downloads]
└─$ 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...
```





- Thông điệp ta nhận được là here is your flag **S3cr3t4g3nt** -> flag là **S3cr3t4g3nt**

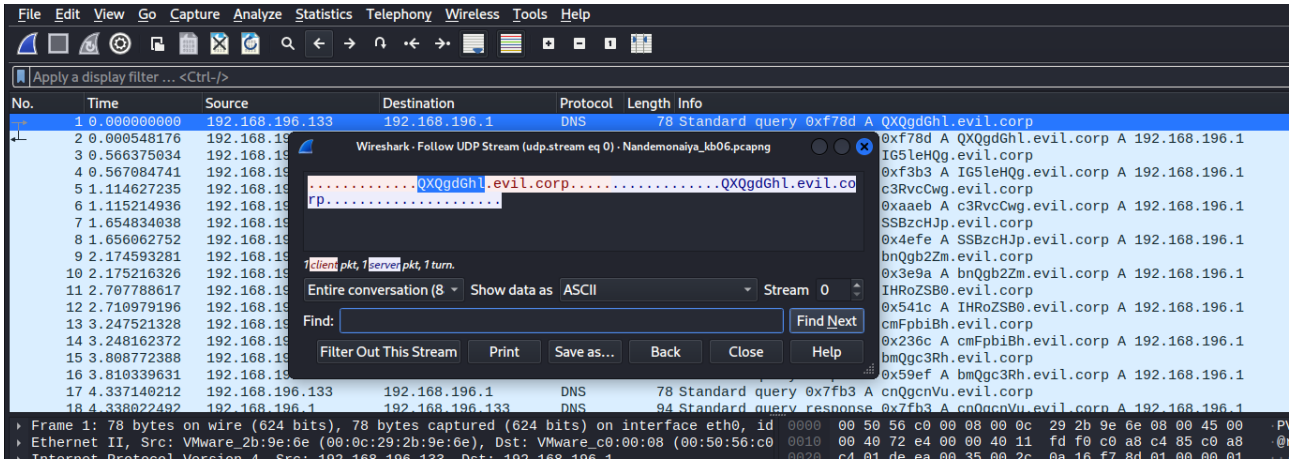
3. Kịch bản 06

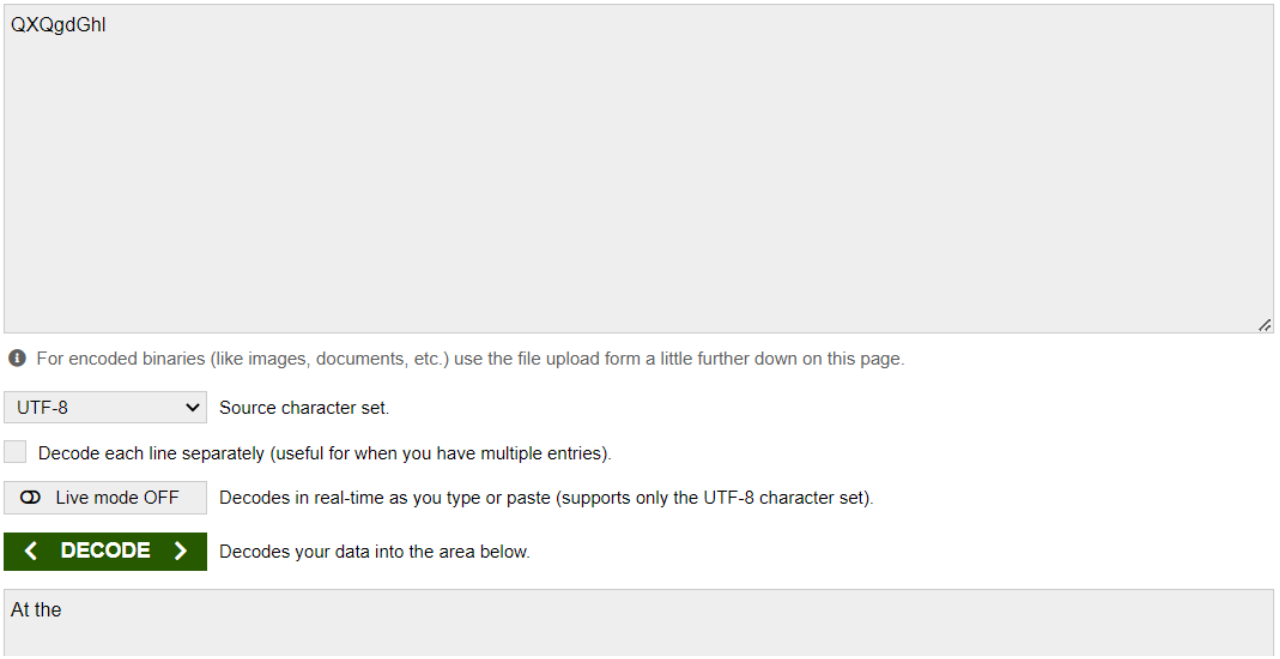
Kịch bản 06. Điều tra trên dữ liệu lưu lượng mạng thu được.

- 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

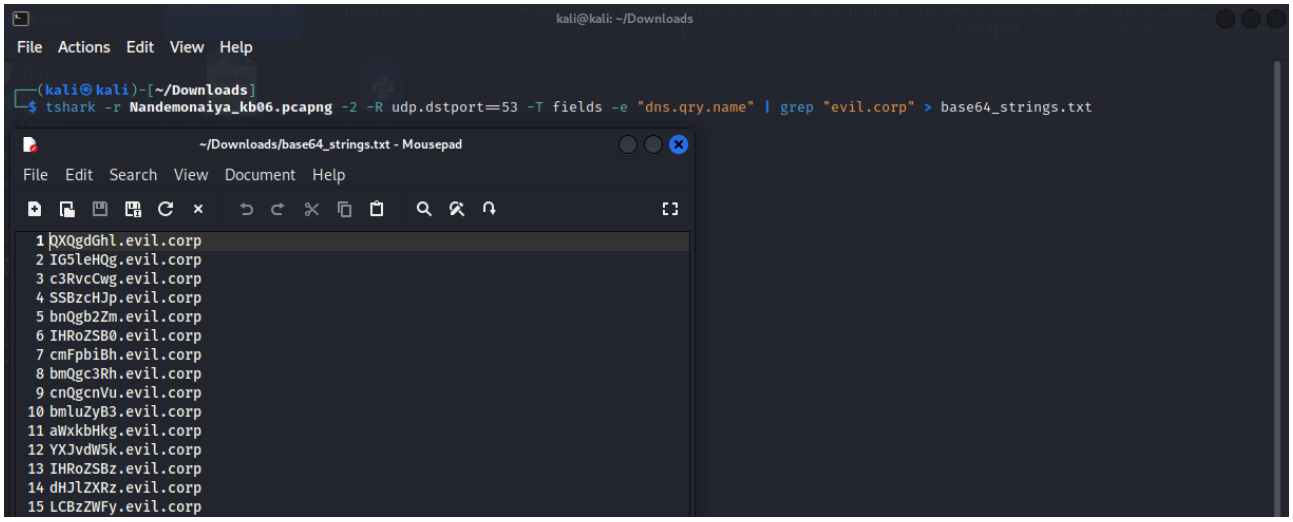
Đáp án:

- Dùng Wireshark mở file Nandemonaiya_kb06.pcap ta để ý thấy trong trường Info, phía trước "**.evil.corp**" luôn có những chuỗi kỳ lạ có vẻ như là Base64, ta sẽ thử decode một đoạn xem có đúng là chuỗi Base64 hay không

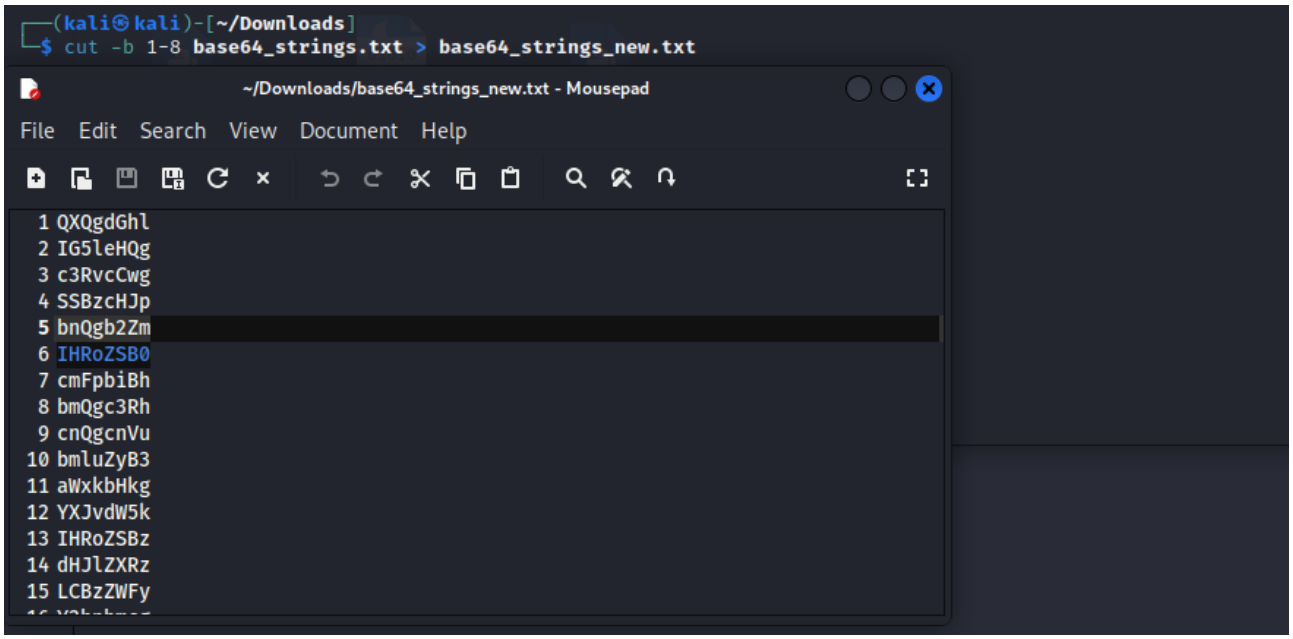




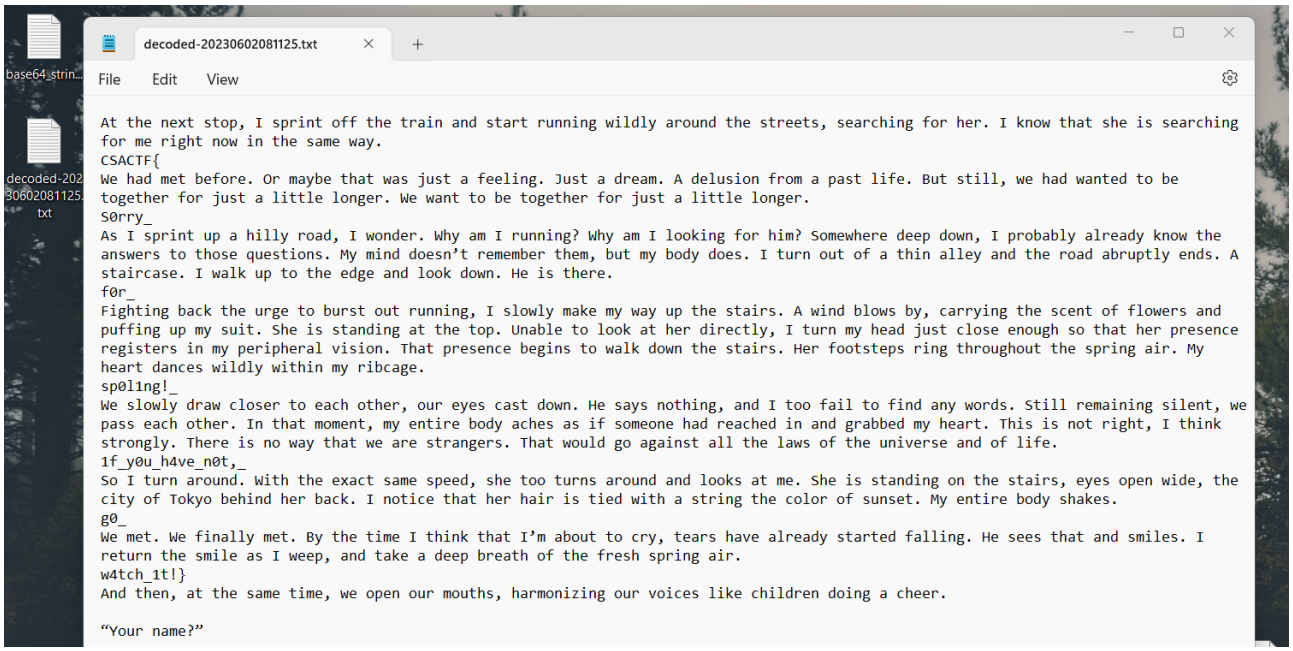
- Chuỗi QXQgdGhI decode Base 64 ta được “At the” -> Có nghĩa -> Ta có ý tưởng tổng hợp các chuỗi tương tự vào 1 file rồi decode 1 lần luôn
- Sử dụng lệnh **tshark -r Nandemonaiya_kb06.pcapng -2 -R udp.dstport==53 -T fields -e "dns.qry.name" | grep "evil.corp" > base64_strings.txt** để lấy các chuỗi có chứa “.evil.corp” về và lưu vạo file base64_strings.txt
  Trong đó udp.dstport==53 tức là ta chỉ lấy các gói tin UDP gởi tới port 53, nếu lấy hết thì sẽ bị lặp chuỗi mất và -e “dns.qry.name” tức là ta chỉ lấy tên truy vấn DNS



- Tiếp theo ta sẽ lấy các chuỗi Base64 và bỏ vào file mới, do mỗi chuỗi Base64 chỉ có 8 ký tự nên ta sẽ dùng lệnh **cut -b 1-8** để lấy các ký tự từ 1-8 ở mỗi dòng



- Tiếp tục sử dụng tool [Base64 Decode and Encode - Online] để decode file base64_strings_new.txt



- Kết hợp các chuỗi được chèn ở giữa các đoạn ta có được flag là **CSACTF{S0rry_f0r_sp0l1ng!_ g0_w4tch_1t!}**

Sinh viên đọc kỹ yêu cầu trình bày bên dưới trang này

YÊU CẦU CHUNG

- Sinh viên tìm hiểu và thực hiện bài tập theo yêu cầu, hướng dẫn.
- Nộp báo cáo kết quả chi tiết những việc (**Report**) bạn đã thực hiện, quan sát thấy và kèm ảnh chụp màn hình kết quả (nếu có); giải thích cho quan sát (nếu có).
- Sinh viên báo cáo kết quả thực hiện và nộp bài.

Báo cáo:

- File .DOCX và .PDF. Tập trung vào nội dung, không mô tả lý thuyết.
- Nội dung trình bày bằng Font chữ Times New Romans/ hoặc font chữ của mẫu báo cáo này (UTM Neo Sans Intel/UTM Viet Sach)– cỡ chữ 13. Canh đều (Justify) cho văn bản. Canh giữa (Center) cho ảnh chụp.
- Đặt tên theo định dạng: [Mã lớp]-ExeX_GroupY. (trong đó X là Thứ tự Bài tập, Y là mã số thứ tự nhóm trong danh sách mà GV phụ trách công bố).

  *Ví dụ: [*NT101.K11.ANTT*]-Exe01_Group03.*
- Nếu báo cáo có nhiều file, nén tất cả file vào file .ZIP với cùng tên file báo cáo.
- Không đặt tên đúng định dạng – yêu cầu, sẽ **KHÔNG** chấm điểm bài nộp.
- Nộp file báo cáo trên theo thời gian đã thống nhất tại courses.uit.edu.vn.

Đánh giá:

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

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

HẾT