

[DFC 2022 - 204] Write-Up

| | |
|------------|--|
| 작성자 | 윤지원 |
| 분석 일자 | 2024.05.23 |
| 작성 일자 | 2024.05.23 |
| 분석 대상 | trudy_pc.ad1 |
| 문서 버전 | 1.0 |
| 작성자 E-mail | yoonjw0827@gmail.com |

0. 목차

1. 문제3

2. 분석 도구3

3. 환경3


4. Write-Up.....4

5. Flag..... 11

6. 별도 첨부 12

7. Reference 13

1. 문제

| | |
|----------|---|
| URL | |
| 문제 내용 | <p>Description Investigators raided the office after receiving an anonymous tip that a spy was targeting someone. Investigators collected some data while the spy was destroying evidence. Analyze the collected image to find the orders and missions the spy received.</p> <p>Questions</p> <ol style="list-style-type: none"> 1) When is the spy's mission date? 2) Where is the spy's mission location? 3) Who is the spy targeting? |
| 문제 파일 |  trudy_pc.ad1 |
| 문제 유형 | Disk forensics |
| 난이도 | 3 / 3 |

2. 분석 도구

| 도구명 | 다운로드 링크 | Version |
|--------------------|---|---------|
| FTK Imager | https://www.exterro.com/digital-forensics-software/ftk-imager | 4.7.1.2 |
| DB Browser(SQLite) | https://sqlitebrowser.org/dl/ | 3.12.2 |
| Wireshark | Wireshark · Download | 3.4.7 |

3. 환경

| OS |
|-------------------|
| Windows 11 64-bit |

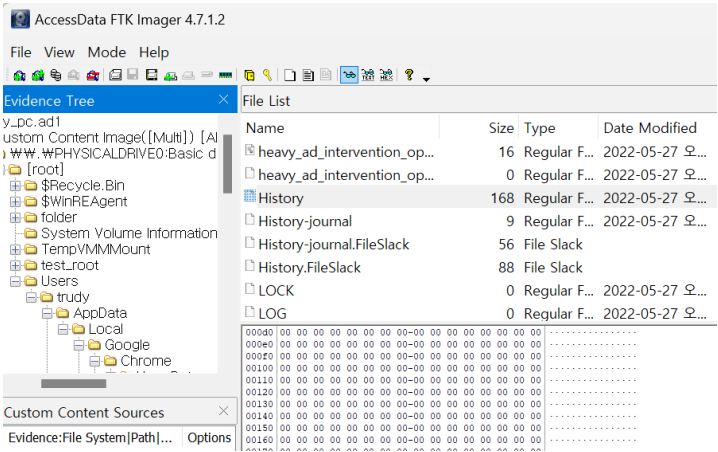
4. Write-Up

| | |
|-----------|--|
| 파일명 | trudy_pc.ad1 |
| 용량 | 67MB |
| SHA256 | b6718d717d16ca2fe049ac81001b94cde4fc433998d4b2bda4a578b71f595675 |
| Timestamp | 2022-05-30 16:15:47 |

문제를 요약하자면 스파이가 누군가를 목표로 하고 있으며, 수집된 이미지를 분석하여 스파이가 받은 명령과 임무를 찾는 문제이다.

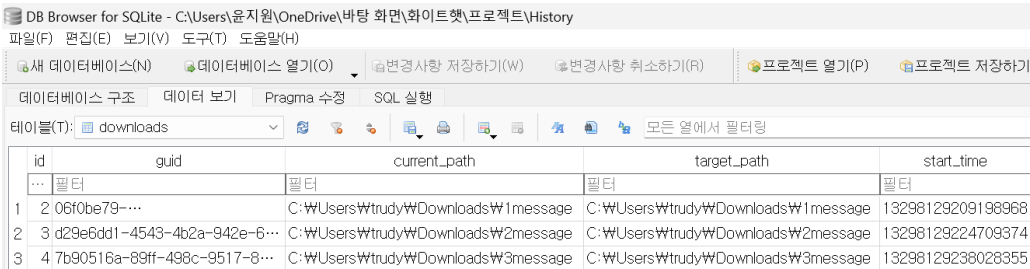
1. 스파이의 임무 수행일은 언제인가?

우선 이 스파이가 언제 임무를 수행했는지를 보기 위해서는 과거 기록을 살펴보는 것이 좋을 것이라고 생각하여 FTK Imager 를 이용하여 파일을 열고, History 라는 파일이 있는지 살펴보았다. 그 결과, [root]\Users\trudy\AppData\Local\Google\Chrome\User Data\Default\History 에서 History 파일을 발견할 수 있었다.



[사진 1] History 파일을 발견한 모습

이 파일을 Export한 다음, DB Browser for SQLite 도구를 이용하여 열어보았다. 데이터베이스 구조를 살펴보니 downloads와 urls라는 테이블을 볼 수 있었다. 스파이가 다운받은 파일과 방문한 url 주소를 알아보면 임무를 수행한 방법과 시간을 알 수 있을 것 같아서 데이터 보기를 이용하여 각각의 테이블을 열어보았다.



| id | guid | current_path | target_path | start_time |
|----|--------------------------------|------------------------------------|------------------------------------|-------------------|
| 1 | 2 06f0be79-... | C:\Users\trudy\Downloads\W1message | C:\Users\trudy\Downloads\W1message | 13298129209198968 |
| 2 | 3 d29e6dd1-4543-4b2a-942e-6... | C:\Users\trudy\Downloads\W2message | C:\Users\trudy\Downloads\W2message | 13298129224709374 |
| 3 | 4 7b90516a-89f-498c-9517-8... | C:\Users\trudy\Downloads\W3message | C:\Users\trudy\Downloads\W3message | 13298129238028355 |

[사진 2] downloads 테이블



DB Browser for SQLite - C:\Users\윤지환\OneDrive\배달의민서\사이트\프로젝트\History

파일(F) 편집(E) 보기(V) 도구(T) 도움말(H)

새 데이터베이스(N) 데이터베이스 열기(O) 변경사항 저장하기(W) 변경사항 취소하기(R) 프로젝트 열기(P) 프로젝트 저장하기(V) 데이터베이스 연결(A) 데이터베이스 닫기(C)

데이터베이스 구조 데이터 보기 Pragma 수정 SQL 실행

데이터베이스: urls

| | id | url | title | visit_count | typed_count |
|----|-----|---|---|-------------|-------------|
| | ... | ... | ... | ... | ... |
| 1 | 1 | https://gmail.com/ | Gmail | 1 | 1 |
| 2 | 2 | https://www.google.com/gmail/ | Gmail | 1 | 0 |
| 3 | 3 | https://mail.google.com/mail/ | Gmail | 1 | 0 |
| 4 | 4 | https://mail.google.com/mail/u/0/ | 받은편지함 (15) - dcf.tudy@gmail.com - Gmail | 2 | 0 |
| 5 | 5 | https://accounts.google.com/ServiceLogin?service=mail&passive=120960&osid=1&continue=https://mail.google.com/mail/u/0/&flow=https://mail.google.com/mail/u/0/&emr=1 | Gmail | 2 | 0 |
| 6 | 6 | https://accounts.google.com/signin/v2/identifier?... | Gmail | 1 | 0 |
| 7 | 7 | https://accounts.google.com/signin/v2/challenge/pwd?... | Gmail | 1 | 0 |
| 8 | 8 | https://accounts.google.com/CheckCookie?... | 계정 복구 옵션 | 1 | 0 |
| 9 | 9 | https://accounts.google.com/ServiceLogin?continue=https://mail.google.com/mail/u/0/&flow=https://mail.google.com/mail/u/0/&emr=1 | 계정 복구 옵션 | 1 | 0 |
| 10 | 10 | https://myaccount.google.com/accounts/SetOSID?authuser=0&continue=https://mail.google.com/mail/u/0/&flow=https://mail.google.com/mail/u/0/&emr=1 | 계정 복구 옵션 | 1 | 0 |
| 11 | 11 | https://myaccount.google.com/security/signoptions/recovery-options-collection?... | 계정 복구 옵션 | 1 | 0 |
| 12 | 12 | https://myaccount.google.com/signoptions/recovery-options-collection?... | 계정 복구 옵션 | 1 | 0 |
| 13 | 13 | https://myaccount.google.com/signoptions/recovery-options-collection?... | 계정 복구 옵션 | 2 | 0 |
| 14 | 14 | https://accounts.google.com/ServiceLogin?... | 받은편지함 (15) - dcf.tudy@gmail.com - Gmail | 1 | 0 |
| 15 | 15 | https://mail.google.com/accounts/SetOSID?... | 받은편지함 (15) - dcf.tudy@gmail.com - Gmail | 1 | 0 |
| 16 | 16 | https://accounts.google.com/accounts/SetOSID?ssdc=1&sid=ALWU2csQkAgU7R4b4m1O1DaeWntfy2Ee0/0?2EEmUJZM70sZS6ZJqpZga&gafNXSEvLA7eq1Bk5Zu/... | 받은편지함 (15) - dcf.tudy@gmail.com - Gmail | 1 | 0 |
| 17 | 17 | https://accounts.google.com/accounts/SetOSID?... | 받은편지함 (15) - dcf.tudy@gmail.com - Gmail | 1 | 0 |
| 18 | 18 | https://mail.google.com/mail/u/0/?pli=1 | 받은편지함 (15) - dcf.tudy@gmail.com - Gmail | 1 | 0 |
| 19 | 19 | https://mail.google.com/mail/u/0/#inbox | 받은편지함 (15) - dcf.tudy@gmail.com - Gmail | 4 | 0 |
| 20 | 20 | https://mail.google.com/mail/u/0/#inbox/PfIczGpGBFGGpcwVwmrM2TB8bqPfs | Ah... Ah... - dcf.tudy@gmail.com - Gmail | 1 | 0 |
| 21 | 21 | https://mail.google.com/mail/u/0/#inbox/PfIczGpGBFGGTPd4rcwaFxbWt2dv | Trudy, Good luck... - dcf.tudy@gmail.com - Gmail | 4 | 0 |
| 22 | 22 | https://mail.google.com/mail/u/0/#inbox/PfIczGpGBFGGTPd4rcwaFxbWt2dv?project=1&messagePartId=0.1 | Trudy, Good luck... - dcf.tudy@gmail.com - Gmail | 3 | 0 |
| 23 | 23 | https://mail.google.com/mail/u/0/#inbox/PfIczGpGBFGGTPd4rcwaFxbWt2dv?project=1&messagePartId=0.1 | Trudy, FYI - dcf.tudy@gmail.com - Gmail | 5 | 0 |
| 24 | 24 | https://mail.google.com/mail/u/0/#inbox/PfIczGpGBFGGTPd4rcwaFxbWt2dv?project=1&messagePartId=0.1 | Trudy, FYI - dcf.tudy@gmail.com - Gmail | 4 | 0 |
| 25 | 25 | https://mail.google.com/mail/u/0/#inbox/PfIczGpGBFGGTPd4rcwaFxbWt2dv?project=1&messagePartId=0.1 | Trudy, God Bless You - dcf.tudy@gmail.com - Gmail | 2 | 0 |
| 26 | 26 | https://mail.google.com/mail/u/0/#inbox/PfIczGpGBFGGTPd4rcwaFxbWt2dv?project=1&messagePartId=0.1 | Trudy, God Bless You - dcf.tudy@gmail.com - Gmail | 2 | 0 |
| 27 | 27 | https://chrome.google.com/webstore?hl=ko | Chrome 앱 스토어 - 확장 프로그램 | 1 | 0 |
| 28 | 28 | https://chrome.google.com/webstore/category/extensions?hl=ko | Chrome 앱 스토어 - 확장 프로그램 | 1 | 0 |

[사진 3]을 살펴보면 스파이가 gmail로 이메일을 받은 것을 볼 수 있는데, 메일 내용은 크게 3개로, 차례대로 **'Trudy, good luck', 'Trudy, FYI', 'Trudy, God Bless You'**이다. 그리고 [사진 2]의 다운로드 경로들을 보면 **1message, 2message, 3message**가 있는 것으로 보아 이 메시지들을 차례로 다운받았을 것이라고 추측할 수 있다. 이를 확실히 하기 위해 downloads 테이블의 뒷부분까지 확인한 결과, [사진 4]와 같이 다운로드 경로가 [사진 3]의 메일 주소와 일치하는 것을 확인할 수 있다.

AccessData FTK Imager 4.7.1.2

File View Mode Help

Evidence Tree

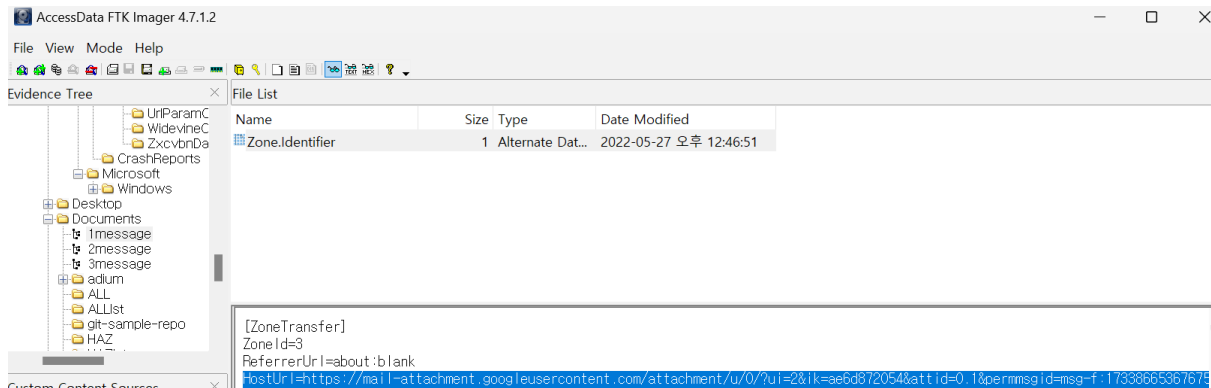
- UriParamC
- WidevineC
- ZxcvbnDa
- CrashReports
- Microsoft
 - Windows
- Desktop
- Documents
 - 1message
 - 2message
 - 3message

File List

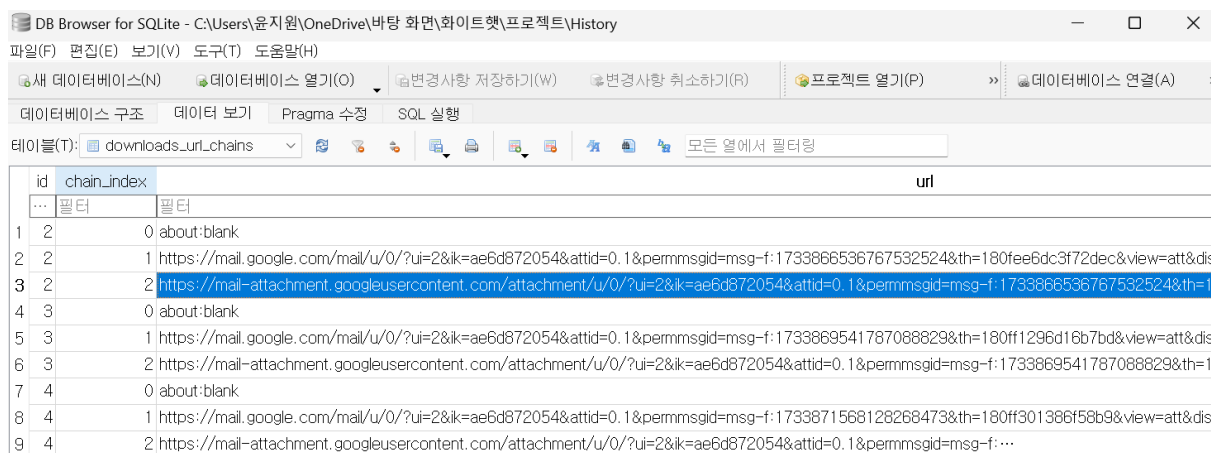
| Name | Size | Type | Date Modified |
|---------------------------|--------|-----------------|------------------------|
| ASP.NET Monsters #93 R... | 3,107 | Regular File | 2021-10-19 오전 11:51:02 |
| asbuitinventory.lst | 1 | Regular File | 2021-10-18 오후 4:19:45 |
| ADReport.htm | 2 | Regular File | 2021-10-19 오전 3:48:41 |
| 3message | 10,176 | Regular File | 2022-05-27 오후 12:47:19 |
| 2message | 1,009 | Regular File | 2022-05-27 오후 12:47:05 |
| 1message | 1,735 | Regular File | 2022-05-27 오후 12:46:51 |
| !\$30 | 16 | NTFS Index A... | 2022-05-27 오후 12:47:42 |

5

[WHS-2] .iso



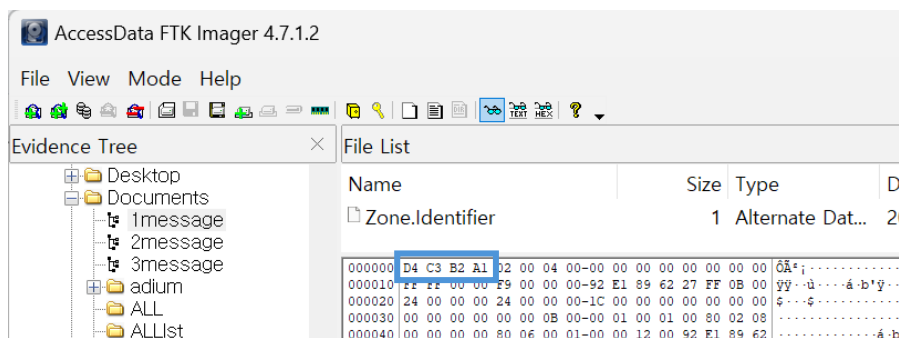
[사진 6] 1message의 Zone.Identifier의 HostUrl



[사진 7] downloads_rul_chains 테이블의 HostUrl

혹시나 다른 파일일 수도 있어서 메시지 파일들을 클릭해보니, 각각 [사진 6]과 같이 Zone.Identifier라는 파일이 들어있었다. 이곳의 HostUrl과 [사진 7]에서 확인할 수 있는 HostUrl이 동일하다는 것을 확인할 수 있기 때문에 [사진 4]와 [사진 5]의 message 파일들은 같은 파일이라고 할 수 있다.

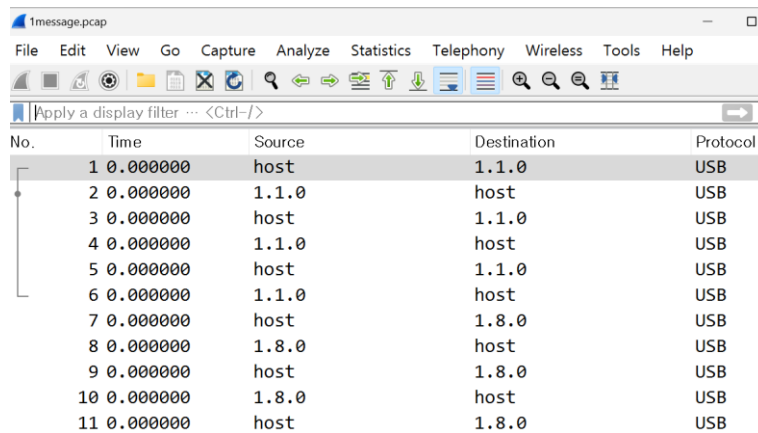
이제 이 message 파일들을 분석하기 위해 살펴보니, [사진 8]과 같이 3개의 파일 모두 'D4 C3 B2 A1'로 시작한다는 것을 알 수 있었다. 이것은 pcap 파일의 시그니처라는 것을 알아냈고, Wireshark를 이용한 분석이 필요할 것이라고 생각하였다.



[사진 8] 1message 파일의 헤더

[WHS-2] .iso

3개의 파일은 모두 Export 한 다음, 확장자를 pcap로 수정하고 Wireshark를 이용하여 열어보았다. 이를 살펴보니, usb를 이용한 통신이라는 것을 확인할 수 있었다.



| No. | Time | Source | Destination | Protocol |
|-----|----------|--------|-------------|----------|
| 1 | 0.000000 | host | 1.1.0 | USB |
| 2 | 0.000000 | 1.1.0 | host | USB |
| 3 | 0.000000 | host | 1.1.0 | USB |
| 4 | 0.000000 | 1.1.0 | host | USB |
| 5 | 0.000000 | host | 1.1.0 | USB |
| 6 | 0.000000 | 1.1.0 | host | USB |
| 7 | 0.000000 | host | 1.8.0 | USB |
| 8 | 0.000000 | 1.8.0 | host | USB |
| 9 | 0.000000 | host | 1.8.0 | USB |
| 10 | 0.000000 | 1.8.0 | host | USB |
| 11 | 0.000000 | host | 1.8.0 | USB |

[사진 9] 1message.pcap 파일

쭉 살펴보니 주소가 1.1.0에서 1.9.0까지 존재했고, 각각 사용한 장치가 다른 것 같았다. 이는 [사진 10]을 통해 확인할 수 있다.

| | | | | | | |
|------------|-------|-------|--|-------|-------|-----|
| 4 0.000000 | 1.1.0 | host | 14 0.000000 | 1.3.0 | host | USB |
| 5 0.000000 | host | 1.1.0 | 15 0.000000 | host | 1.3.0 | USB |
| 6 0.000000 | 1.1.0 | host | | | | |
| 7 0.000000 | host | 1.8.0 | | | | |
| | | | SB USB EVICE DESCRIPTOR bLength: 18 bDescriptorType: 0x01 (DEVICE) bcdUSB: 0x0200 bDeviceClass: Device (0x00) bDeviceSubClass: 0 bDeviceProtocol: 0 (Use class code info from Interface Descrip bMaxPacketSize0: 8 idVendor: Lenovo (0x17ef) idProduct: ThinkPad Compact Keyboard with TrackPoint (0x6047) | | | |

[사진 10] 1.1.0은 키보드, 1.3.0은 ThinkPad Compact Keyboard with TrackPoint 사용

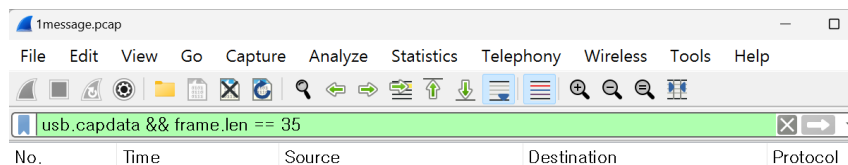
이렇게 쭉 보던 중, 1.5.1을 발견하였는데, USB function 부분이 다른 것들은 USB function이라고 적혀 있는 데에 비해, 여기는 [사진 11]을 통해 interrupt라는 단어가 있는 것을 발견했다. 따라서 차별점이 있다고 생각하여 더 알아보니, 여기서 사용한 장치는 HID 장치임을 알 수 있었다.

| | | | |
|---|-------|-------|----|
| 65 0.277463 | 1.5.1 | host | US |
| 66 0.277491 | host | 1.5.1 | US |
| 67 0.285411 | 1.5.1 | host | US |
| IRP USBD_STATUS: USBD_STATUS_SUCCESS (0x00000000) USB Function: USB_FUNCTION_BULK_OR_INTERRUPT_TRANSFER (0x0009) > IRP information: 0x01, Direction: PDO -> FDO USB bus id: 1 Device address: 5 > Endpoint: 0x81, Direction: IN USB transfer type: USB_INTERRUPT (0x01) Packet Data Length: 8 [Request in: 62] [Time from request: 0.015954000 seconds] [bInterfaceClass: HID (0x03)] HID Data: 0000fc000000fcff | | | |

[사진 11] 1.5.1의 장치는 HID 장치

[WHS-2] .iso

따라서 HID 데이터를 살펴보기 위해 위에 필터에 'usb.capdata && frame.len == 35'를 입력해보았다. 이것은 usb 패킷이면서 프레임 길이가 35인 패킷을 검색하는 것인데, 길이를 35로 정한 이유는 일반적으로 USB HID 장치에서 전송할 때 사용되는 고정된 크기이기 때문이다. 그러나 검색 결과는 아무것도 나오지 않았다. 그러나 [사진 13]과 같이 앞의 capdata 조건을 제외하고 검색하면 프레임 길이가 35인 패킷들이 전부 나타났다. 그래서 우선 cmd를 이용하여 'frame.len == 35' 조건에 해당하는 패킷들의 capdata를 **tshark를 통해 1m_151.txt**라는 텍스트 파일로 추출했다.



[사진 12] capdata와 프레임 길이 조건으로 검색한 결과

| No. | Time | Source | Destination | Protocol |
|-----|----------|--------|-------------|----------|
| 31 | 0.141412 | 1.5.1 | host | USB |
| 33 | 0.149407 | 1.5.1 | host | USB |
| 35 | 0.157438 | 1.5.1 | host | USB |
| 37 | 0.165469 | 1.5.1 | host | USB |
| 39 | 0.173413 | 1.5.1 | host | USB |
| 41 | 0.181498 | 1.5.1 | host | USB |
| 43 | 0.189424 | 1.5.1 | host | USB |

[사진 13] 프레임 길이 조건으로만 검색한 결과

```
C:\Users\윤지원\OneDrive\바탕 화면\화이트햇프로젝트>tshark -r "C:\Users\윤지원\OneDrive\바탕 화면\화이트햇프로젝트\1message.pcap" -Y "frame.len == 35" -T fields -e usb.capdata > 1m_151.txt
```

[사진 14] 1m_151.txt 생성 명령어

이후로 막혀서 write-up을 참고한 결과, 1m_151.txt에 출력된 8바이트 데이터가 마우스 패킷 데이터와 유사하여 이를 **파싱하면 마우스가 움직인 경로를 시각화**할 수 있다는 정보를 얻었다. 따라서 파싱을 진행하기 위해 UsbMiceDataHacker 코드를 사용한다고 했는데, 처음에는 이 코드가 이미지 파일 안에 존재하는 줄 알고 열심히 찾아봤는데, 결국 찾지 못했다. 그래서 풀 기미가 안 보이던 중에, 혹시나 해서 깃허브에 검색해봤더니 해당 코드를 발견할 수 있었다. 따라서 <https://github.com/laziok/UsbMiceDataHacker2021> 에 있는 코드를 알맞게 수정하여 파싱하면 될 것 같다고 생각했다.

UsbMiceDataHacker.py 코드는 Wireshark에서 캡처된 USB 마우스 데이터를 분석하고 시각화하는 기능을 가지고 있었다. 따라서 이를 [사진 15]와 같이 수정하고 **UsbMiceDataHacker_1.py**라는 이름으로 저장하였다.


```
# get data of pcap
.....
command = "tshark -r %s -T fields -e usb.capdata > %s" % (
    pcapFilePath, DataFileName)
print(command)
os.system(command)

# read data
with open(DataFileName, "r") as f:
    for line in f:
        data.append(line[0:-1])

# handle move
.....
with open(pcapFilePath, "r") as f:
    for line in f:
        data.append(line[0:-1])

for i in data:
    Bytes = i.split(":")
    if len(Bytes) == 8:
        horizontal = 1 # -
        vertical = 2 # |
    elif len(Bytes) == 4:
        horizontal = 1 # -
        vertical = 2 # |
    else:
        continue
```

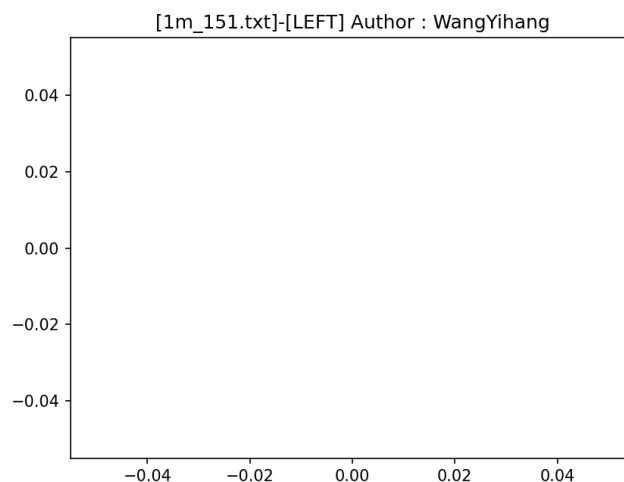
[사진 15] UsbMiceDataHacker_1.py의 일부분

기존의 코드와 달라진 점은 저 주석 처리된 초록색 부분이 tshark를 이용하여 pcapdata를 추출하는 명령인데, 이미 cmd를 통해 1m_151.txt로 생성해 놓았기 때문에 주석 처리를 해 놓았다. 또한 아래 for문에서 가로 세로 길이를 보기 편하도록 조금씩 조정해주었다.

이렇게 수정한 코드를 [사진 16]의 명령어를 이용하여 실행해주었고, [사진 17]과 같은 결과가 나왔다.

```
C:\Users\윤지원\OneDrive\바탕 화면\화이트햇\프로젝트>python UsbMiceDataHacker_1.py 1m_151.txt LEFT
'rm' 은(는) 내부 또는 외부 명령, 실행할 수 있는 프로그램, 또는
배치 파일이 아닙니다.
```

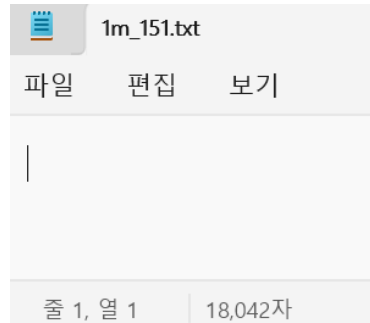
[사진 16] UsbMiceDataHacker_1.py 실행



[사진 17] UsbMiceDataHacker_1.py 실행 결과

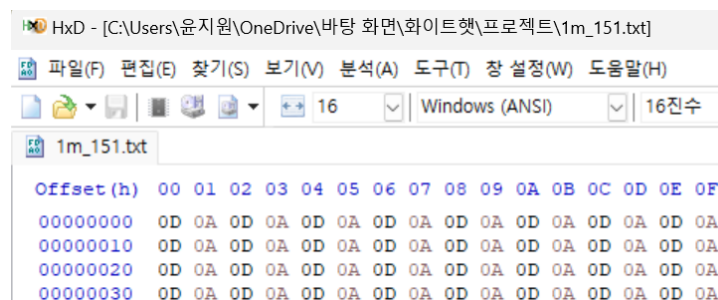
[WHS-2] .iso

시각화 창이 나온 것을 보면 명령어나 코드가 틀린 것은 아닌 것 같아서 무엇이 문제인지 알아보니, 1m_151.txt 파일이 문제였다. 원래 파일 내용에 capdata가 출력되어야 하는데, 보이는 데이터가 없는데 글자 수는 많이 존재하는 것을 확인할 수 있었다.



[사진 18] 1m_151.txt 내용

따라서 텍스트 파일에 문제가 생긴 줄 알고 HxD에도 넣어보았는데, [사진 19]와 같이 0D와 0A만 계속해서 나타나는 것을 볼 수 있었다. 이것은 tshark 명령어가 제대로 실행되지 않았거나, 필터 조건에 맞는 데이터가 없는 경우 발생한 것이라고 한다.



[사진 19] 1m_151.txt를 HxD에 넣은 모습

따라서 tshark 명령어가 문제인지 알아보기 위해 cmd에서 'tshark -r "C:\Users\윤지원\OneDrive\바탕 화면\화이트햇\프로젝트\1message.pcap" -Y "frame.len == 35" -T fields -e usb.capdata' 명령어를 입력해보았다. 그 결과, 출력 되는 패킷도 있고 출력이 안되는 패킷도 있었다. 따라서 그냥 **패킷 파일에 usb.capdata 필드가 없는 것**이라고 추측하였다.

그러나 이렇게 되면 마우스 경로 시각화를 할 수 없게 되어 결국 해결에 실패하였다.

2, 3번 문제도 이와 유사한 방식으로 해결하는 문제이기 때문에 해결하지 못하였다.

5. Flag

6. 별도 첨부

7. Reference

- [URL]