



작성자	류나연
분석 일자	2024.05.28-6.09
작성 일자	2024.06.09
분석 대상	Windows11.dd.zip
문서 버전	1.0
작성자 E-mail	01star01ek@gmail.com

0. 목차

1. 문제3

2. 분석 도구3

3. 환경3


4. Write-Up.....4

5. Flag..... 11

6. 별도 첨부 12

7. Reference 14

1. 문제

URL	-
문제 내용	(*별도 첨부)
문제 파일	 Windows11.dd.zip
문제 유형	Disk forensic
난이도	3 / 3

2. 분석 도구

도구명	다운로드 링크	Version
NTFS Log Tracker	https://sites.google.com/site/forensicnote/ntfs-log-tracker	1.71
FTK Imager	https://www.exterro.com/ftk-product-downloads/ftk-imager-version-4-7-1	4.7.1.2
Systools Outlook system viwer	https://systools-ost-viewer.software.informer.com/5.0/#google_vignette	5.0
DB browser for sqllite3	https://sqlitebrowser.org/	3.12.2
HXD	https://mh-nexus.de/en/hxd/	2.5
Kernel for Outlook Express	https://kernel-for-outlook-express.software.informer.com/	Evaluation

3. 환경

OS
Windows 11 Home



4. Write-Up

Q1. What is the SHA1 hash value of a document file that the researcher received from the broker? (20 points)

[illegible]

해당 파일의 압축을 해제하고 이를 ftk imager로 열어 해당 이미징 이미징 파일의 구조를 확인하였다. 특이점이 있는지 찾던중 basic data partition(3)의 root 폴더 내에서 \$LOGFILE, \$MFT, \$I30이 있는 것을 보고 일단 db로 만들어 확인해보야겠다고 느꼈다.

NITS Log Tracker v1.71

Target Path : C:\Users\W01staf\OneDrive\WPB\BWP\Logfile [Clear]

\$binImb\$ File Path : C:\Users\W01staf\OneDrive\WPB\BWP\BIM30 [Clear] Paste

Source Files Folder Path (for UsnJnl Record Carving) : UsnJnl's Carving Alignment : [B ->] [Clear]

Option :

\$MFT File Path : C:\Users\W01staf\OneDrive\WPB\BWP\MFT [Clear]

Open SQLite DB File :

SQLite DB File Path : [Suspectious Behavior Detection] [Clear] Open

Search [CSV Export]

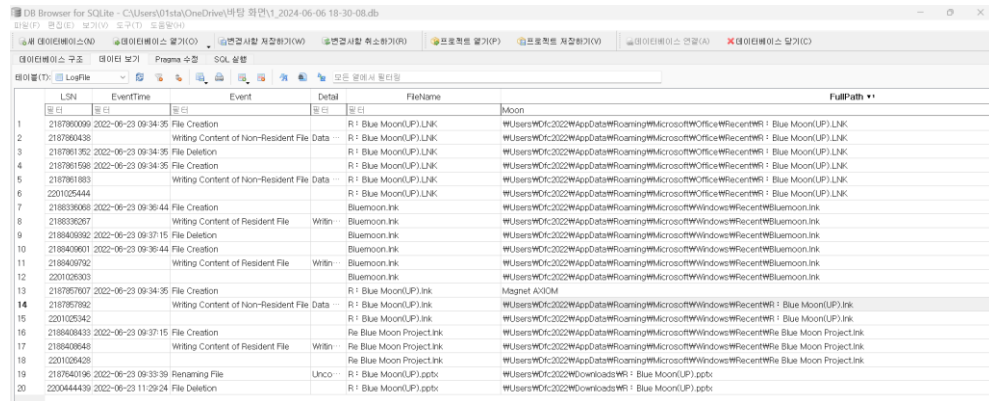
\$LogFile \$binImb\$ \$LogFile/Search Result \$binImb\$/Search Result Suspectious Behavior Detection

< > Page :

4

[WHS-2] .iso

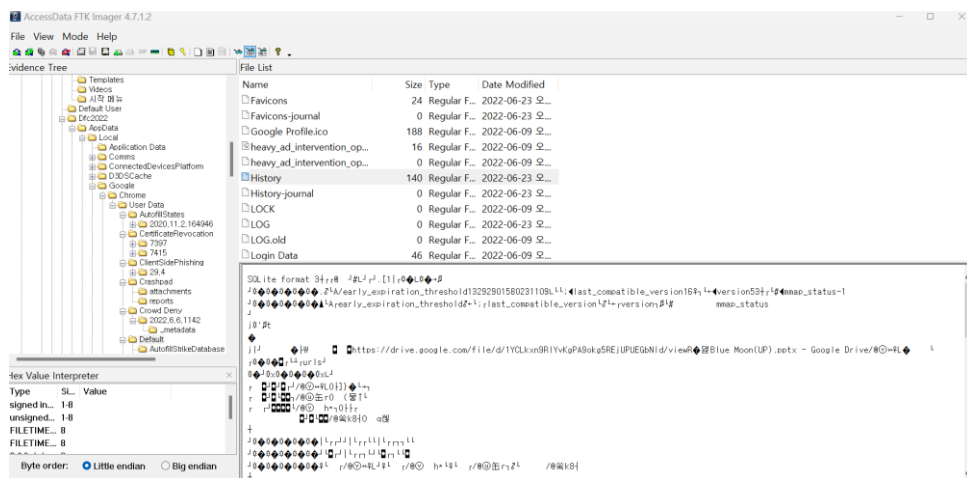
이에 따라 해당 파일들을 추출하여 NTFS log tracker로 db형식의 파일로 만들었다.



LSN	EventTime	Event	Detail	FileName	FullPath
1	218780009	2022-06-23 09:34:35	File Creation	R: Blue Moon(UP).lnk	\\Users\\WDR\\2022\\AppData\\Roaming\\Microsoft\\Windows\\Recent\\Blue Moon(UP).lnk
2	218780048		Writing Content of Non-Resident File Data	R: Blue Moon(UP).lnk	\\Users\\WDR\\2022\\AppData\\Roaming\\Microsoft\\Windows\\Recent\\Blue Moon(UP).lnk
3	2187801352	2022-06-23 09:34:35	File Deletion	R: Blue Moon(UP).lnk	\\Users\\WDR\\2022\\AppData\\Roaming\\Microsoft\\Windows\\Recent\\Blue Moon(UP).lnk
4	2187801598	2022-06-23 09:34:35	File Creation	R: Blue Moon(UP).lnk	\\Users\\WDR\\2022\\AppData\\Roaming\\Microsoft\\Windows\\Recent\\Blue Moon(UP).lnk
5	2187801883		Writing Content of Non-Resident File Data	R: Blue Moon(UP).lnk	\\Users\\WDR\\2022\\AppData\\Roaming\\Microsoft\\Windows\\Recent\\Blue Moon(UP).lnk
6	2201025444			R: Blue Moon(UP).lnk	\\Users\\WDR\\2022\\AppData\\Roaming\\Microsoft\\Windows\\Recent\\Blue Moon(UP).lnk
7	2188190068	2022-06-23 09:36:44	File Creation	BlueMoon.lnk	\\Users\\WDR\\2022\\AppData\\Roaming\\Microsoft\\Windows\\Recent\\BlueMoon.lnk
8	2188030207		Writing Content of Resident File	BlueMoon.lnk	\\Users\\WDR\\2022\\AppData\\Roaming\\Microsoft\\Windows\\Recent\\BlueMoon.lnk
9	2188409392	2022-06-23 09:37:15	File Deletion	BlueMoon.lnk	\\Users\\WDR\\2022\\AppData\\Roaming\\Microsoft\\Windows\\Recent\\BlueMoon.lnk
10	2188409601	2022-06-23 09:36:44	File Creation	BlueMoon.lnk	\\Users\\WDR\\2022\\AppData\\Roaming\\Microsoft\\Windows\\Recent\\BlueMoon.lnk
11	2188409702		Writing Content of Resident File	BlueMoon.lnk	\\Users\\WDR\\2022\\AppData\\Roaming\\Microsoft\\Windows\\Recent\\BlueMoon.lnk
12	2201025033			BlueMoon.lnk	\\Users\\WDR\\2022\\AppData\\Roaming\\Microsoft\\Windows\\Recent\\BlueMoon.lnk
13	2187867007	2022-06-23 09:34:35	File Creation	R: Blue Moon(UP).lnk	Magnet AIOOM
14	2187867092		Writing Content of Non-Resident File Data	R: Blue Moon(UP).lnk	\\Users\\WDR\\2022\\AppData\\Roaming\\Microsoft\\Windows\\Recent\\Blue Moon(UP).lnk
15	2201025242			R: Blue Moon(UP).lnk	\\Users\\WDR\\2022\\AppData\\Roaming\\Microsoft\\Windows\\Recent\\Blue Moon(UP).lnk
16	2188408432	2022-06-23 09:37:15	File Creation	R: Blue Moon Project.lnk	\\Users\\WDR\\2022\\AppData\\Roaming\\Microsoft\\Windows\\Recent\\Blue Moon Project.lnk
17	2188408648		Writing Content of Resident File	R: Blue Moon Project.lnk	\\Users\\WDR\\2022\\AppData\\Roaming\\Microsoft\\Windows\\Recent\\Blue Moon Project.lnk
18	2201024928			R: Blue Moon Project.lnk	\\Users\\WDR\\2022\\AppData\\Roaming\\Microsoft\\Windows\\Recent\\Blue Moon Project.lnk
19	2187861092	2022-06-23 09:33:39	Renaming File	Unco-	\\Users\\WDR\\2022\\Downloads\\Blue Moon(UP).pptx
20	2200444339	2022-06-23 11:29:24	File Deletion	R: Blue Moon(UP).pptx	\\Users\\WDR\\2022\\Downloads\\Blue Moon(UP).pptx

[사진 3] db 파일에서 발견한 Blue Moon(UP).lnk 기록

해당 데이터베이스 기록에서 BLUE MOON과 관련된 기록을 발견할 수 있었다. 이에 따라 해당 위치들을 방문해보였으나 파일은 찾을 수 없었다. 또한 .LNK 형식의 기록들은 결국 파일이 없기에 접속해봤자 파일을 볼 수 없었다. 그래도 이를 통해 해당 디스크내 다른 기록 또는 파일이 존재할 것임을 추측하게 되었고, DFC 2022 폴더를 집중적으로 탐색하게 되었다.

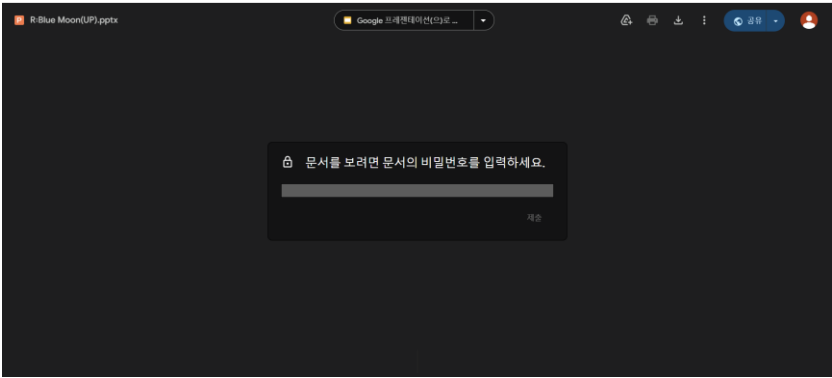


Name	Size	Type	Date Modified
Favicons	24	Regular F...	2022-06-23 0...
Favicons-journal	0	Regular F...	2022-06-23 0...
Google Profile	188	Regular F...	2022-06-09 0...
heavy_ad_intervention.op...	16	Regular F...	2022-06-09 0...
heavy_ad_intervention.op...	0	Regular F...	2022-06-09 0...
History	140	Regular F...	2022-06-23 0...
History-journal	0	Regular F...	2022-06-23 0...
LOCK	0	Regular F...	2022-06-09 0...
LOG	0	Regular F...	2022-06-23 0...
LOG.old	0	Regular F...	2022-06-09 0...
Login Data	46	Regular F...	2022-06-09 0...

[사진 4] chrome history 에서 발견한 다운로드 url

해당 과정에서 다운로드로 받은 파일이기에 특정 웹사이트에서 다운로드 받았을 수도 있겠다는 생각이 들었다. 따라서 chrome의 history 내용을 확인하였다. 그리고 해당 기록들에서 파일과 관련된 링크를 찾을 수 있었다. 따라서 해당 드라이브 사이트에 접속해보았다.

[WHS-2] .iso



[사진 5] 드라이브에 올려져 있던 Blue Moon(UP).pptx

파일에 암호가 걸려있는 Blue Moon(UP).pptx를 다운로드 받았다.

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

새로운 기능 및 개선 사항에 대 한 최신 PowerShell을 설치 하세요! https://aka.ms/PSWindows

PS C:\Users\01sta> certutil -hashfile "C:\Users\01sta\Downloads\R-Blue Moon(UP).pptx" sha256
SHA256의 C:\Users\01sta\Downloads\R-Blue Moon(UP).pptx 해시:
35ea004c5a1a44c7084b4ca7cf0e897865c07b3811401ae4a7a707350b779d54
CertUtil: -hashfile 명령이 성공적으로 완료되었습니다.
PS C:\Users\01sta> |
```

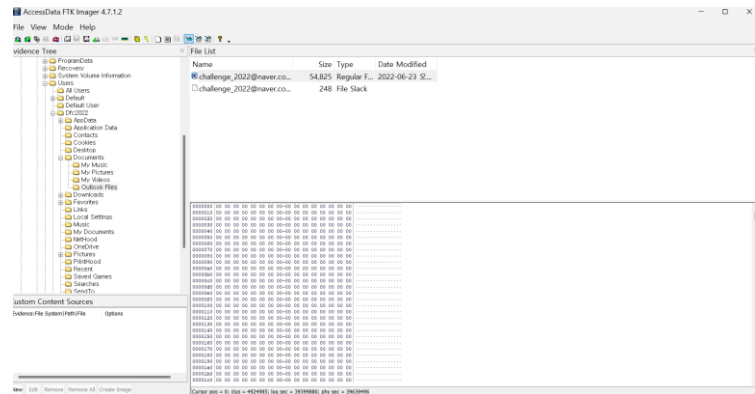
[사진 6] sha256값 구하기

그 후 windows power shell을 사용하여 해당 파일의 sha256의 값을 구했다.

답은 35ea004c5a1a44c7084b4ca7cf0e897865c07b3811401ae4a7a707350b779d54 이다.

- Q2. What is the password of the file that the researcher received from the broker? (150 points)
- Q3. When did the researcher read the e-mail containing the password of the document file? (UTC+9) (80 points)
- Q4. What is the GPS information of the place where the researcher is supposed to meet the broker? (50 points)

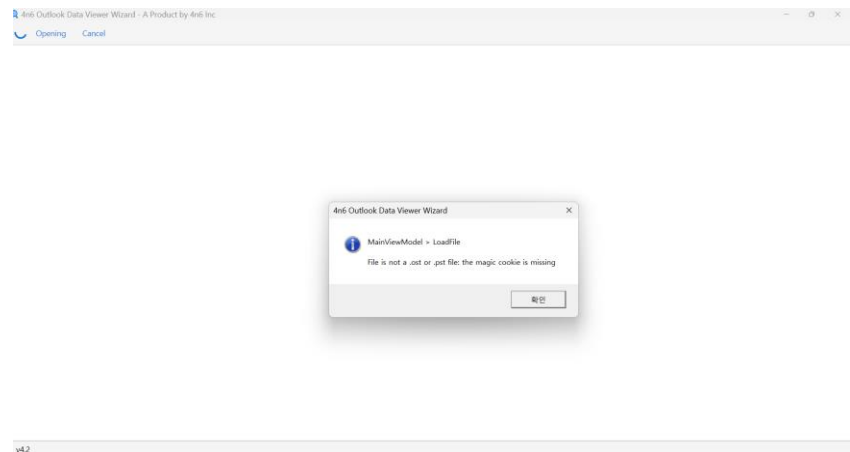
2-1. 구조 파악



[사진 7] .pst 파일을 발견한 모습

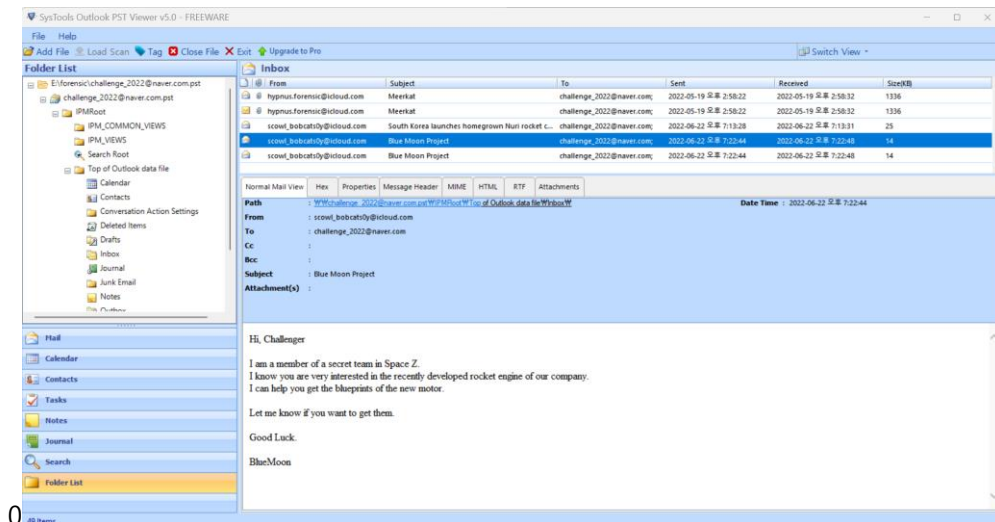
2번 문제를 잘 살펴보면 받았다 라는 키워드가 중요하다. 이를 통해 이메일 또는 ssh 연결을 의심하였고 관련 증거를 찾던 중 outlook의 파일인 .pst 파일을 발견하였다. 따라서 해당 파일 내에 분명히 관련된 기록이 있을 것이라고 생각하여 이를 열어보고자 했다.

2-2. 분석



[사진 8] 열리지 않는 .pst 파일

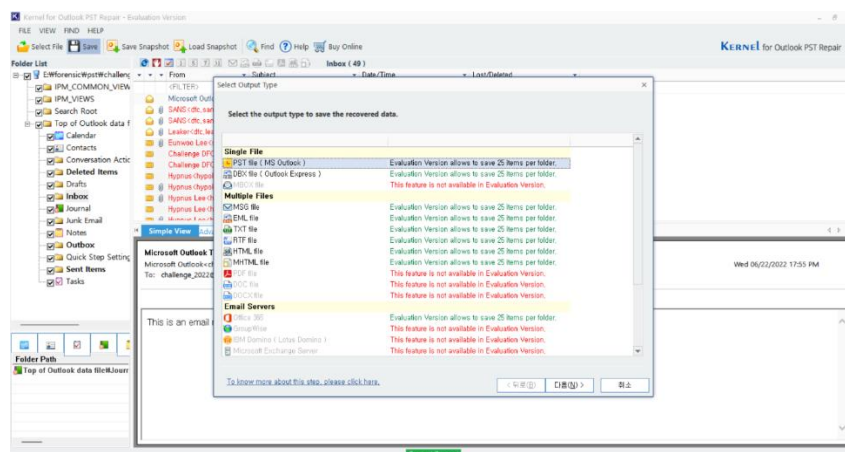
따라서 4n6 Outlookdata viwer Wizard를 통해 열어보려고 했으나 열리지 않았다. 이에 툴의 문제인가 싶어 가능한 경우의 툴들은 거의 다 적용해보았으나 열리지 않았고 이에 따라 pst 파일을 복원 또는 편집해야 하나 싶었다.



[사진 9] blue moon project에 관해 이야기를 나눈 메일 기록

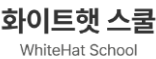
(outlook-pst-viewer, EncaseUS Data, Recovery 등 다 확인이 불가능 했는데 이 툴에서는 확인 가능했다.)

그러나 신기하게도 Systools Outlook PST viewer에서는 이를 확인할 수 있었다. 그럼에도 불구하고 해당 기록에서는 Blue Moon project의 관해 인사만 나눌 뿐 세부적인 정보는 이야기하지 않고 있었다. 이에 따라 해당 파일을 복구해보고자 했다.



[사진 10] pst 파일로 재 저장

일단 해당 파일이 .pst 파일로 인식되지 않으므로 인식을 위해 .pst 파일로 추출하여 저장하였다. HXD 로 열어보니 재저장 한 파일은 pst 의 헤더 메타데이터를 가지고있었다. 그럼에도 불구하고 이를 실행했을 때 변화가 없었다. 따라서 세부적으로 분석을 해보아겠다고 생각했다.

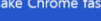


화이트햇 스쿨
WhiteHat School

[사진 11] HXD를 통해 열어본 pst 파일

GoldFynch PST 분석기

PST/OST 파일을 검증하고 분석하는 브라우저 내 도구입니다.



복호화 코드를 찾아보던 중 pst 파일을 쉽게 분석할 수 있는 사이트를 찾아 이를 소개하고자 설명한다. <https://goldfynch.com/pst-analyzer/> 로 해당 사이트에 업로드시 쉽게 분석하여 준다. 이를 통해 순열 암호화가 된 것이 확실함을 한번더 확신할 수 있었다.

<https://gist.github.com/knez/870a2c4c70287e0b075d0a2f925d40cb>

```

decode.py > ...
19 | return bytes(mpbCryptFrom512[i] for i in payload)
20 |
21 |
22 | if len(sys.argv) != 2:
23 |     print('Provide folder with encoded .txt attachments')
24 |     exit(1)
25 |
26 | for root, dirs, files in os.walk(sys.argv[1]):
27 |     for file in files:
28 |         if file.endswith('.txt'):
29 |             full_path = os.path.join(root, file)
30 |             with open(full_path, 'rb') as f:
31 |                 payload = f.read()
32 |
33 |             with open(full_path + '_decoded', 'wb') as d:
34 |                 d.write(decode(payload))
35 |                 print('Decoded file', file)

```

문제 출력 디버그 콘솔 터미널 포트

PS C:\Users\01sta\OneDrive\바탕 화면\컴공과제\보안\화이트햇\forensic> python decode.py ./pst

[사진 13] 디코딩을 하여 파일을 복원하는 모습

코드를 실행해보았더니 정상적으로 .txt_decoded 파일이 추출되었으며 이를 통해 복호화되어 있던 메시지를 확인하였다.

Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Decoded text
03255550	6C	65	2E	63	6F	6D	2F	70	72	65	73	65	6E	74	61	74	le.com/presentat
03255560	69	6F	6E	2F	64	2F	31	59	43	4C	6B	78	6E	39	52	6C	ion/d/1YCLkxn9Rl
03255570	59	76	4B	67	50	41	39	6F	6B	67	35	52	45	6A	55	50	YvKgPA9okg5REjUP
03255580	55	45	47	62	4E	6C	64	2F	65	64	69	74	3F	75	73	70	UEGbNld/edit?usp
03255590	3D	73	68	61	72	69	6E	67	26	61	6D	70	3B	6F	75	69	=sharing&oi
032555A0	64	3D	31	30	32	33	36	39	32	39	37	37	36	33	36	31	d=10236929776361
032555B0	36	30	31	30	34	39	34	26	61	6D	70	3B	72	74	70	6F	6010494&rtpo
032555C0	66	3D	74	72	75	65	26	61	6D	70	3B	73	64	3D	74	72	f=true&sd=tr
032555D0	75	65	3C	2F	61	3E	3C	62	72	3E	3C	2F	64	69	76	3E	ue </div>
032555E0	3C	64	69	76	3E	3C	62	72	3E	3C	2F	64	69	76	3E	3C	<div> </div><
032555F0	64	69	76	3E	59	6F	75	20	63	61	6E	20	6F	70	65	6E	div>You can open
03255600	20	74	68	65	20	66	69	6C	65	20	77	69	74	68	20	6B	the file with k
03255610	2D	43	68	61	6C	6C	65	6E	67	65	5F	22	64	66	63	22	-Challenge " <u>dfd</u> "
03255620	5F	32	30	32	32	2A	20	2E	3C	62	72	3E	3C	2F	64	69	_2022* . </di
03255630	76	3E	3C	64	69	76	3E	3C	62	72	3E	3C	2F	64	69	76	v><div> </div>
03255640	3E	3C	64	69	76	3E	54	68	65	20	61	70	70	6F	69	6E	><div>The appoin
03255650	74	6D	65	6E	74	20	74	69	6D	65	20	61	6E	64	20	70	tment time and p
03255660	6C	61	63	65	20	61	72	65	20	69	6E	20	74	68	65	20	lace are in the
03255670	66	69	6C	65	2E	3C	62	72	3E	3C	2F	64	69	76	3E	3C	file. </div><
03255680	64	69	76	3E	3C	62	72	3E	3C	2F	64	69	76	3E	3C	64	div> </div><d
03255690	69	76	3E	47	6F	6F	64	20	4C	75	63	6B	2E	3C	62	72	iv>Good Luck.
032556A0	3E	3C	2F	64	69	76	3E	3C	64	69	76	3E	3C	62	72	3E	></div><div>
032556B0	3C	2F	64	69	76	3E	3C	64	69	76	3E	42	6C	75	65	4D	</div><div>BlueM
032556C0	6F	6F	6E	3C	62	72	3E	3C	2F	64	69	76	3E	3C	64	69	oon </div><di
032556D0	76	3E	3C	64	69	76	3E	3C	62	72	3E	3C	2F	64	69	76	v><div> </div>

[사진 14] 복호화 되어 가려져 있던 내용

이를 통해 해당 2번의 답인 비밀번호는 **dfd** 임을 알 수 있다.

또한 해당 메일을 본 시간과 따로 장소에 대해 보낸 메일도 이를 통해 알 수 있다.

5. Flag

1. 35ea004c5a1a44c7084b4ca7cf0e897865c07b3811401ae4a7a707350b779d54
2. dfd
3. 2022-06-23
4. 위도(Latitude) : 35.4541137 / 경도(Longitude) : 128.3681031

6. 별도 첨부

- 원본 영문 문제

Description Police arrested a broker for leaking blueprints of Space Z's new engine. The broker stated that he passed on information about a document file (**R:Blue Moon(UP).pptx**) that includes the appointment time and place through e-mail. The officer confiscated the researcher's computer for digital forensic analysis.

Target	Hash (MD5)
Windows11.dd.zip	68b05a9c173c9d8d8ea679cbcca3df67

Questions

Please solve all problems based on the time zone of the system.

Data in any language other than English is not relevant to problem-solving.

- 1)What is the SHA1 hash value of a document file that the researcher received from the broker? (20 points)
- 2)What is the password of the file that the researcher received from the broker? (150 points)
- 3)When did the researcher read the e-mail containing the password of the document file? (UTC+9) (80 points)
- 4)What is the GPS information of the place where the researcher is supposed to meet the broker? (50 points)

[번역본]

경찰은 Z 우주의 새로운 엔진의 청사진을 유출한 혐의로 브로커를 체포했습니다. 그 브로커는 이메일을 통해 약속 시간과 장소가 포함된 문서 파일(R:Blue Moon(UP.pptx)에 대한 정보를 전달했다고 진술했습니다. 그 경찰관은 디지털 포렌식 분석을 위해 연구원의 컴퓨터를 압수했습니다.

시스템의 시간대를 기준으로 모든 문제를 해결해 주시기 바랍니다.

영어 이외의 다른 언어로 된 데이터는 문제 해결과 관련이 없습니다.

- 1) 연구자가 브로커로부터 받은 문서파일의 SHA1 해시값은? (20점)
- 2) 연구원이 브로커에게 받은 파일의 비밀번호는? (150점)
- 3) 연구자는 문서파일의 비밀번호가 포함된 이메일을 언제 읽었습니까? (UTC+9) (80점)
- 4) 연구원이 중개인을 만나기로 한 장소의 GPS 정보는? (50점)

- 디코딩 코드

```
- import os
import sys

def decode(payload):
    mpbbCryptFrom512 = [71, 241, 180, 230, 11, 106, 114, 72, 133, 78,
158, 235, 226, 248, 148, 83, 224, 187, 160, 2,
232, 90, 9, 171, 219, 227, 186, 198, 124, 195,
16, 221, 57, 5, 150, 48, 245, 55, 96, 130, 140,
201, 19, 74, 107, 29, 243, 251, 143, 38, 151,
202, 145, 23, 1, 196,
50, 45, 110, 49, 149, 255, 217, 35, 209, 0, 94,
121, 220, 68, 59, 26,
40, 197, 97, 87, 32, 144, 61, 131,
185, 67, 190, 103, 210, 70, 66, 118,
192, 109, 91, 126, 178, 15, 22, 41,
60, 169, 3, 84, 13, 218, 93, 223,
246, 183, 199, 98, 205, 141, 6, 211,
105, 92, 134, 214, 20, 247, 165, 102,
117, 172, 177, 233, 69, 33, 112, 12,
135, 159, 116, 164, 34, 76, 111, 191,
31, 86, 170, 46, 179, 120, 51, 80,
176, 163, 146, 188, 207, 25, 28, 167,
99, 203, 30, 77, 62, 75, 27, 155,
79, 231, 240, 238, 173, 58, 181, 89,
4, 234, 64, 85, 37, 81, 229, 122,
137, 56, 104, 82, 123, 252, 39, 174,
215, 189, 250, 7, 244, 204, 142, 95,
239, 53, 156, 132, 43, 21, 213, 119,
52, 73, 182, 18, 10, 127, 113, 136,
253, 157, 24, 65, 125, 147, 216, 88,
44, 206, 254, 36, 175, 222, 184, 54,
200, 161, 128, 166, 153, 152, 168, 47,
14, 129, 101, 115, 228, 194, 162, 138,
212, 225, 17, 208, 8, 139, 42, 242,
237, 154, 100, 63, 193, 108, 249, 236]

    return bytes(mpbbCryptFrom512[i] for i in payload)

if len(sys.argv) != 2:
    print('Provide folder with encoded .txt attachments')
    exit(1)

for root, dirs, files in os.walk(sys.argv[1]):
    for file in files:
        if file.endswith('.txt'):
            full_path = os.path.join(root, file)
            with open(full_path, 'rb') as f:
                payload = f.read()

            with open(full_path + '_decoded', 'wb') as d:
                d.write(decode(payload))
            print('Decoded file', file)
```

7. Reference

- Pst 메타데이터
https://learn.microsoft.com/en-us/openspecs/office_file_formats/ms-pst/c9876f5a-664b-46a3-9887-ba63f113abf5
https://developer.skao.int/projects/ska-pst/en/latest/api/metadata_mapping.html
- PST 분석 사이트
<https://goldfynch.com/pst-analyzer/>
- 파티션 복구
<https://lemonpoo22.tistory.com/208>
- 디코딩 코드
<https://gist.github.com/knez/870a2c4c70287e0b075d0a2f925d40cb>