FINANCIAL SECURITY INSTITUTE

Digital Forensics Challenge 2023

kimbabasaksaksak

153 - Partition Finder

FINANCIAL SECURITY INSTITUTE

Description

Analyze the RAW disk image file to answer the question. (The filesystem for Windows was formatted in Windows 10.)

Questions 1

Find all deleted partitions, and analyze the following information for each partition. (100 points)

- Partition type, volume serial number, formatted date/time, capacity, 1st sector(physical)

Questions 2

Which volumes are connected to more than one system? Analyze the following information for those volumes. (50 points)

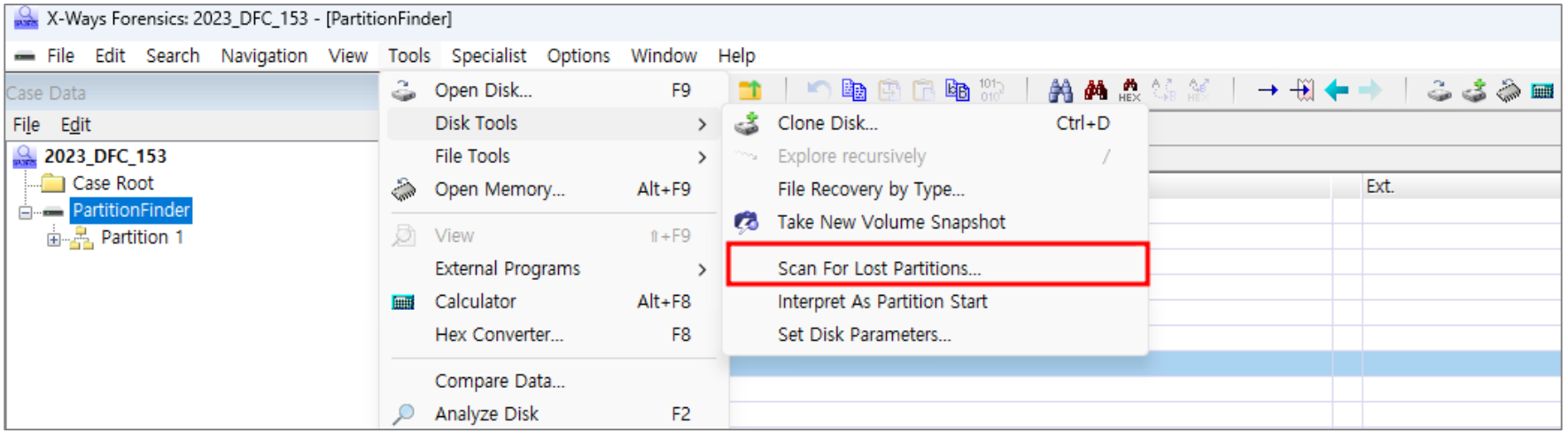
- Partition type, volume serial number, formatted date/time, capacity, 1st sector(physical)

153 - Partition Finder: Question 1

FINANCIAL SECURITY INSTITUTE

Deleted Partition Recovery

- Lost partitions can be recovered using X-Ways Forensics tool
- X-Ways forensics tool supports recovery of partitions formatted with FAT12, FAT16, FAT32, exFAT, NTFS, HFS+, HFSJ, HFSX, Ext2, Ext3, Ext4 file systems.



153 - Partition Finder: Question 1

FINANCIAL SECURITY INSTITUTE

Deleted Partition Recovery

- 'partition, not referenced in partition table' in the PartitionFinder tab are recovered partitions.
- A total of 11 partitions were recovered.

Name	Existent	Description	Ext.	Type	1st sector ▲	F
Start sectors	❊	virtual (for examination purposes)			0	
Partition 1	✓	partition, existing		FAT16	2,048	
Partition 02	X	partition, not referenced in partition table		FAT16	4,196,352	
Unpartitioned space	❊	virtual (for examination purposes)			6,293,504	
Partition 03	X	partition, not referenced in partition table		NTFS	12,582,912	
Partition 04	X	partition, not referenced in partition table		NTFS	41,943,040	
Partition 05	X	partition, not referenced in partition table		Ext4	96,468,992	
Partition 07	X	partition, not referenced in partition table		FAT16	115,343,360	
Partition 08	X	partition, not referenced in partition table		NTFS	123,729,920	
Partition 06	X	partition, not referenced in partition table		exFAT	159,383,552	
Partition 09	X	partition, not referenced in partition table		exFAT	159,383,564	
Partition 10	X	partition, not referenced in partition table		FAT16	184,549,376	
Partition 11	X	partition, not referenced in partition table		Ext2	188,743,680	
Partition 12	X	partition, not referenced in partition table		NTFS	213,909,504	

153 - Partition Finder: Question 1

FINANCIAL SECURITY INSTITUTE

Analysis of recovered partitions

- In the list of recovered partitions, Partition 6 and Partition 9 are exFAT file systems and are the same file system. Therefore, there are a total of 10 partitions recovered.
- The volume boot record for an exFAT file system is 12 sectors in size. The backup volume boot record exists immediately after the volume boot record. The first sector of the volume boot record is the VBR, so the VBR backup copy is located 12 sectors afterthe VBR. So the X-Ways Forensics tool recognized two VBRs and output two partitions.

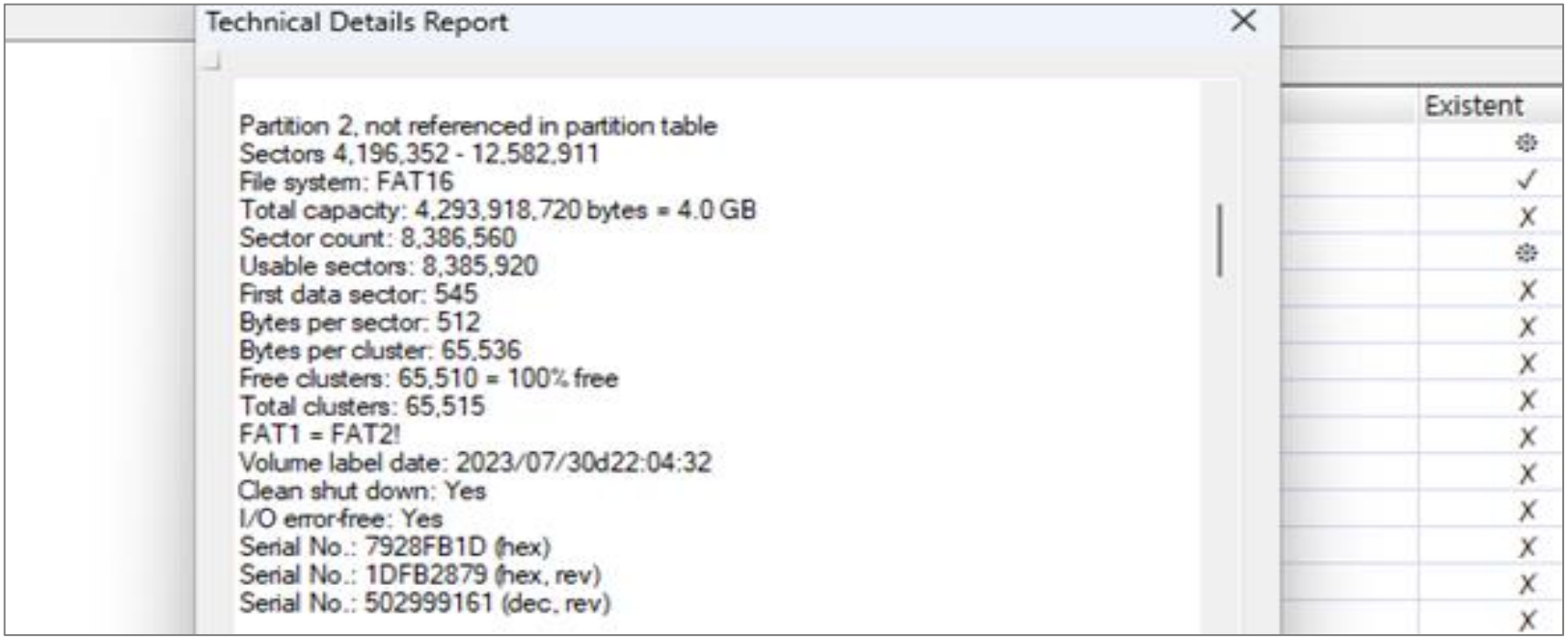
Name	Existent	Description	Ext.	Type	1st sector
Start sectors	❊	virtual (for examination purposes)			0
Partition 1	✓	partition, existing		FAT16	2,048
Partition 02	X	partition, not referenced in partition table		FAT16	4,196,352
Unpartitioned space	❊	virtual (for examination purposes)			6,293,504
Partition 03	X	partition, not referenced in partition table		NTFS	12,582,912
Partition 04	X	partition, not referenced in partition table		NTFS	41,943,040
Partition 05	X	partition, not referenced in partition table		Ext4	96,468,992
Partition 07	X	partition, not referenced in partition table		FAT16	115,343,360
Partition 08	X	partition, not referenced in partition table		NTFS	123,729,920
Partition 06	X	partition, not referenced in partition table		exFAT	159,383,552
Partition 09	X	partition, not referenced in partition table		exFAT	159,383,564
Partition 10	X	partition, not referenced in partition table		FAT16	184,549,376
Partition 11	X	partition, not referenced in partition table		Ext2	188,743,680
Partition 12	X	partition, not referenced in partition table		NTFS	213,909,504

153 - Partition Finder: Question 1

FINANCIAL SECURITY INSTITUTE

Check partition type, volume serial number, capacity, and 1st sector values

- Using the Technical Details Report function of the X-Ways Forensics tool, you can easily check the Partition type, volume serial number, capacity, and 1st sector information required by the problem. However, additional analysis is required to find the formatted data/time.



153 - Partition Finder: Question 1

FINANCIAL SECURITY INSTITUTE

Check formatted date/time

Note

- In the question, it was given that the Windows file system was formatted in Windows 10. When formatting in Windows 10, a System Volume Information folder is created, and the creation date/time of the folder is the format date/time.

01. FAT16 File System

- When the FAT16 file system is formatted by specifying a volume label, a date/time value is recorded in the volume label entry. The date/time value usually matches the format date/time unless there are special cases.
- Recovered FAT16 file systems exist both with and without volume labels.
- In the case of FAT16 with a volume label, the date/time value is different from the creation date/time of System Volume Information. The difference is about 16 minutes. The exact cause of this phenomenon is unknown, but it is presumed to have occurred due to various reasons, such as the formatting tool used or the operating system settings. In the question, a hint was given that the Windows file system was formatted in Windows 10, so our team submitted the System Volume Information creation date/time value as the format date/time.
- In the case of FAT16 without a volume label, our team submitted the System Volume Information creation date/time value as the format date/time.

153 - Partition Finder: Question 1

FINANCIAL SECURITY INSTITUTE

Check formatted date/time

02. exFAT File System

- Due to the structure of the exFAT file system, it is basically difficult to know the format date/time.
- In the question, a hint was given that the Windows file system was formatted in Windows 10, so our team submitted the System Volume Information folder creation date/time as the format date/time.

03. NTFS File System

- The date/time that the NTFS file system was formatted can be determined by the creation date/time of file system metadata files such as $MFT and $Logfile files.
- The restored NTFS file systems were almost identical except for slight differences in the file system metadata creation date/time and the System Volume Information folder creation date/time. There were some that were 9 hours different from each other, but this is presumed to be due to different time zone processing for each format tool. Therefore, our team submitted the NTFS file system format date/time based on the file system metadata creation date/time.

153 - Partition Finder: Question 1

FINANCIAL SECURITY INSTITUTE

Check formatted date/time

04. Ext2 File System

- For Ext2, the format date/time is not recorded, and there is no file system metadata file with a timestamp, so the format date/time cannot be determined by default.
- Since Ext2 is not a Windows file system, it is difficult to estimate the format date/time using the System Volume Information folder. Therefore, our team submitted that the formatted time was unknown.

05. Ext4 File System

- The date/time that the Ext4 file system was formatted can be determined by the creation date/time of file system metadata files such as .journal files.
- Since Ext4 is not a Windows file system, our team submitted the creation date/time of the .journal file as the formatted time.

153 - Partition Finder: Question 1

FINANCIAL SECURITY INSTITUTE

Question 1 Answer

Note

- Date/Time values in FAT12/16/32 file systems are recorded based on the local time zone. This is because the FAT file system was designed without UTC time in mind during its early stages of development. Therefore, date/time values recorded in a FAT file system are stored in the local time zone of the system on which the file was created, modified, or accessed.

- Answer 1

Partition Type	FAT16
Volume Serial Number	79 28 FB 1D (hex)
Formatted date/time	2023/07/30 22:20:32 LT
Capacity	4,293,918,720 bytes = 4.0 GB
1st sector(physical)	4,196,352

- Answer 2

Partition Type	NTFS
Volume Serial Number	C0 0E EB 1D 34 C3 D9 01 (hex)
Formatted date/time	2023/07/30 22:20:59 (UTC+0)
Capacity	42,949,672,960 bytes = 40.0 GB
1st sector(physical)	12,582,912

153 - Partition Finder: Question 1

FINANCIAL SECURITY INSTITUTE

Question 1 Answer

- Answer 3

Partition Type	NTFS
Volume Serial Number	90 E6 54 94 ED C2 D9 01 (hex)
Formatted date/time	2010/05/04 03:24:57 (UTC+0)
Capacity	37,580,960,256 bytes = 35.0 GB
1st sector(physical)	41,943,040

- Answer 4

Partition Type	Ext4
Volume Serial Number	-
Formatted date/time	2023/07/30 22:25:41 (UTC+0)
Capacity	32,212,254,720 bytes = 30.0 GB
1st sector(physical)	96,468,992

- Answer 5

Partition Type	FAT16
Volume Serial Number	B0 ED 13 C7 (hex)
Formatted date/time	2023/07/30 22:57:28 LT
Capacity	4,293,918,720 bytes = 4.0 GB
1st sector(physical)	115,343,360

- Answer 6

Partition Type	NTFS
Volume Serial Number	10 E2 4B 05 41 C3 D9 01 (hex)
Formatted date/time	2023/07/30 23:53:21 (UTC+0)
Capacity	64,692,944,896 bytes = 60.3 GB
1st sector(physical)	123,729,920

153 - Partition Finder: Question 1

FINANCIAL SECURITY INSTITUTE

Question 1 Answer

- Answer 7

Partition Type	exFAT
Volume Serial Number	4D 20 01 1E (hex)
Formatted date/time	2023/07/30 13:26:23 (UTC+0)
Capacity	46,439,333,888 bytes = 43.3 GB
1st sector(physical)	159,383,552

- Answer 8

Partition Type	FAT16
Volume Serial Number	A5 3A 01 1E (hex)
Formatted date/time	2023/07/30 22:26:51 (UTC+0)
Capacity	2,147,483,648 bytes = 2.0 GB
1st sector(physical)	184,549,376

- Answer 9

Partition Type	Ext2
Volume Serial Number	-
Formatted date/time	-
Capacity	12,884,901,888 bytes = 12.0 GB
1st sector(physical)	188,743,680

- Answer 10

Partition Type	NTFS
Volume Serial Number	80 06 24 3B 35 C3 D9 01 (hex)
Formatted date/time	2023/07/30 22:28:57 (UTC+0)
Capacity	18,520,997,888 bytes = 17.2 GB
1st sector(physical)	213,909,504

153 - Partition Finder: Question 2

FINANCIAL SECURITY INSTITUTE

SID folder in $RECYCLE.BIN folder

- Among the recovered partitions, there are partitions in the $RECYCLE.BIN folder that have multiple SID folders with different identifiers of the local computer.

PartitionFinder | PartitionFinder, P4 | PartitionFinder, P8

₩$RECYCLE.BIN

Name	Path	Full path	Existent	Description
.. = (Root directory)			✓	existing
. = $RECYCLE.BIN (3)	₩	₩$RECYCLE.BIN	✓	existing
S-1-5-21-2800687128-566568502-1112790091-1000 (1)	₩$RECYCLE.BIN	₩$RECYCLE.BIN₩S-1-5-21-280...	✓	existing
S-1-5-21-4250255928-1639986778-3672263943-1001 (1)	₩$RECYCLE.BIN	₩$RECYCLE.BIN₩S-1-5-21-425...	✓	existing
S-1-5-21-532960987-376975441-931564778-1001 (1)	₩$RECYCLE.BIN	₩$RECYCLE.BIN₩S-1-5-21-532...	✓	existing

PartitionFinder | PartitionFinder, P4 | PartitionFinder, P8

₩$RECYCLE.BIN

Name	Path	Full path	Existent	Description
.. = (Root directory)			✓	existing
. = $RECYCLE.BIN (2)	₩	₩$RECYCLE.BIN	✓	existing
S-1-5-21-4250255928-1639986778-3672263943-1001 (1)	₩$RECYCLE.BIN	₩$RECYCLE.BIN₩S-1-5-21-425...	✓	existing
S-1-5-21-532960987-376975441-931564778-1001 (1)	₩$RECYCLE.BIN	₩$RECYCLE.BIN₩S-1-5-21-532...	✓	existing

153 - Partition Finder: Question 2

FINANCIAL SECURITY INSTITUTE

SID folder in $RECYCLE.BIN folder

Note

- A SID is a unique value used in Windows to uniquely identify a security principal (e.g. user, group, service, etc.). The structure of SID can be divided into several parts. For example, in S-1-5-21-4250255928-1639986778-3672263943-1001:
  - S: Indicates the start of the SID string.
  - 1: Version number.5: Revision number.
  - 21: Identifier of the institution.
  - 4250255928-1639986778-3672263943: This is the local computer's identifier that uniquely identifies the system.
  - 1001: Indicates a unique account number within the system.

- The aforementioned partitions can be said to be partitions connected to two or more systems because there are two or more SID folders with different local computer identifiers (e.g. 4250255928-1639986778-3672263943) that uniquely identify the system.

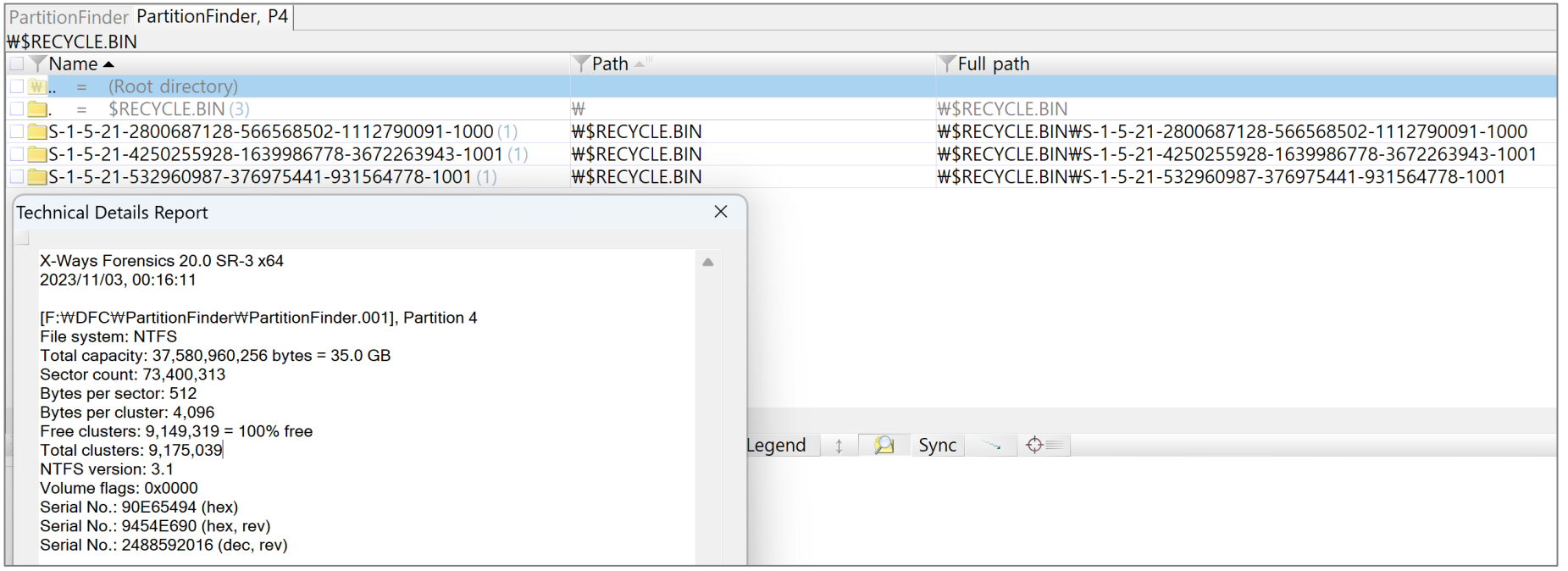
153 - Partition Finder: Question 2

FINANCIAL SECURITY INSTITUTE

Question 2 Answer

- Answer 1

Partition Type	NTFS
1st sector(physical)	41,943,040



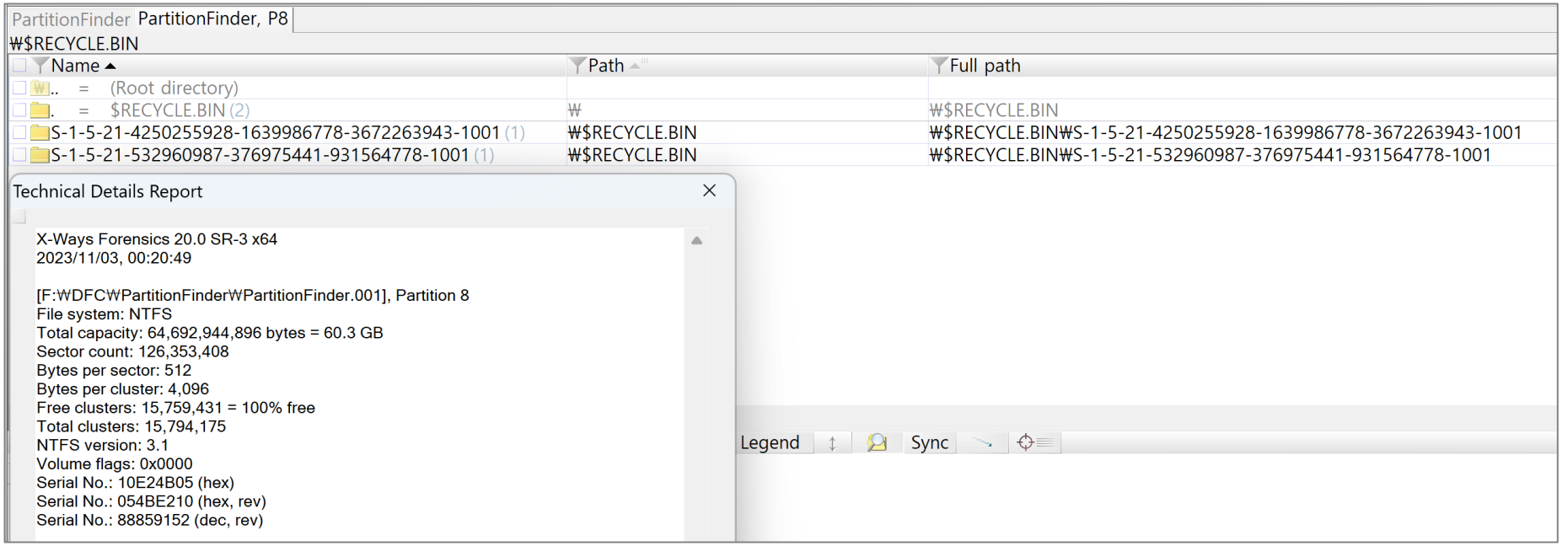
153 - Partition Finder: Question 2

FINANCIAL SECURITY INSTITUTE

Question 2 Answer

- Answer 2

Partition Type	NTFS
1st sector(physical)	123,729,920



302 - Do not blink

FINANCIAL SECURITY INSTITUTE

Description

Analyze the video and prevent a terrorist attack!

Questions 1

When is the attack scheduled? (50 points)

Questions 2

What is the cryptographic key is needed to identify the location the attack? (125 points)

Questions 3

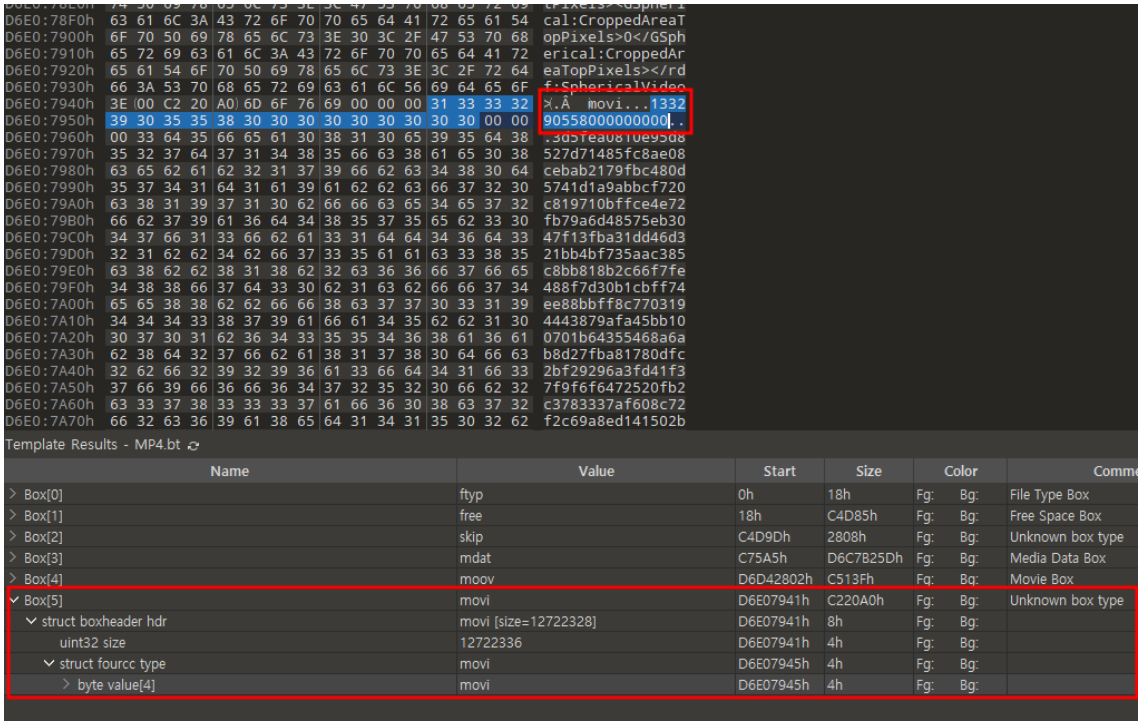
Where is the attack scheduled? (125 points)

302 - Do not blink : Question 1

FINANCIAL SECURITY INSTITUTE

When is the attack scheduled?

- movi box, a component of avi file was found in the given mp4 file
- movi box had abnormal values and there was an encoded time value in it.

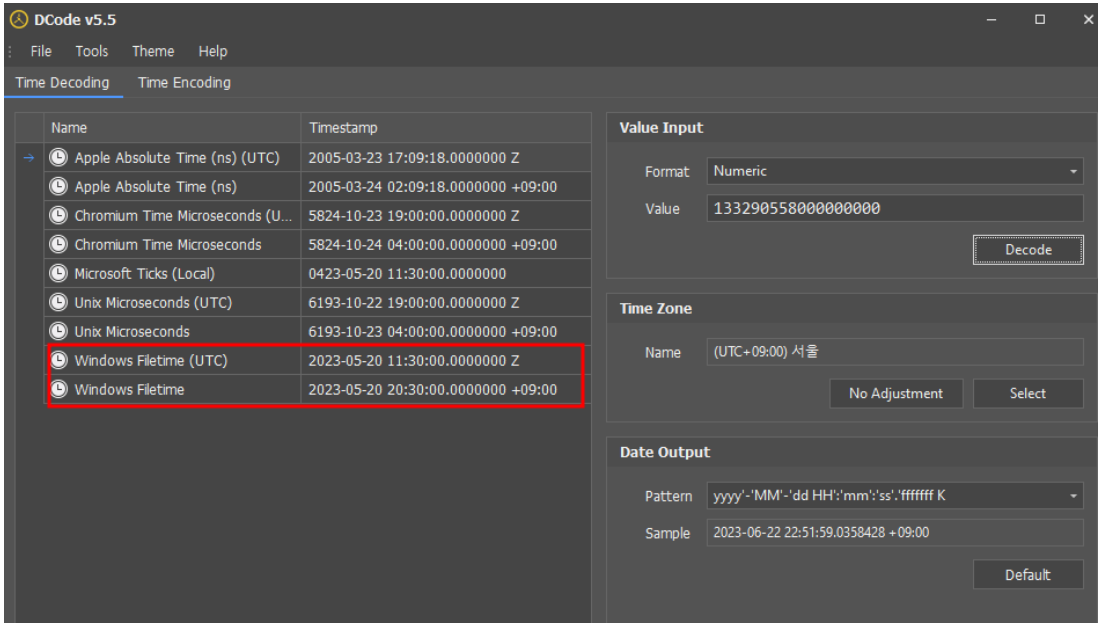


302 - Do not blink : Question 1

FINANCIAL SECURITY INSTITUTE

Question 1 Answer

2023-05-20 20:30:00 UTC+9



302 - Do not blink : Question 2

FINANCIAL SECURITY INSTITUTE

Analysis of the mp4 file

- Check detailed information of Seoul.mp4 with ffprobe
- The projection type of the video is equirectangular

```
wka@BOOK-NAGQM8L4RU:~/Secu$ ffprobe Seoul.mp4
ffprobe version 4.4.2-0ubuntu0.22.04.1 Copyright (c) 2007-2021 the FFmpeg developers
  built with gcc 11 (Ubuntu 11.2.0-19ubuntu1)
  configuration: --prefix=/usr --extra-version=0ubuntu0.22.04.1 --toolchain=hardened --libdir=/usr/lib/x86_64-linux-gnu --incdir=/usr/include/x86_64-linux-gnu --arch=amd64 --e
nable-gpl --disable-stripping --enable-gnutls --enable-ladspa --enable-libaom --enable-libass --enable-libbluray --enable-libbs2b --enable-libcaca --enable-libcdio --enable-li
bcodec2 --enable-libdav1d --enable-libflite --enable-libfontconfig --enable-libfreetype --enable-libfribidi --enable-libgme --enable-libgsm --enable-libjack --enable-libmp3lam
e --enable-libmysofa --enable-libopenjpeg --enable-libopenmpt --enable-libopus --enable-libpulse --enable-librabbitmq --enable-librubberband --enable-libshine --enable-libsnap
py --enable-libsoxr --enable-libspeex --enable-libsrt --enable-libssh --enable-libtheora --enable-libtwolame --enable-libvidstab --enable-libvorbis --enable-libvpx --enable-li
bwebp --enable-libx265 --enable-libxml2 --enable-libxvid --enable-libzimg --enable-libzmq --enable-libzvbi --enable-lv2 --enable-omx --enable-openal --enable-opencl --enable-o
pengl --enable-sdl2 --enable-pocketsphinx --enable-librsvg --enable-libmfx --enable-libdc1394 --enable-libdrm --enable-libiec61883 --enable-chromaprint --enable-frei0r --enabl
e-libx264 --enable-shared
  libavutil      56. 70.100 / 56. 70.100
  libavcodec     58.134.100 / 58.134.100
  libavformat    58. 76.100 / 58. 76.100
  libavdevice    58. 13.100 / 58. 13.100
  libavfilter     7.110.100 /  7.110.100
  libswscale      5.  9.100 /  5.  9.100
  libswresample   3.  9.100 /  3.  9.100
  libpostproc    55.  9.100 / 55.  9.100
Input #0, mov,mp4,m4a,3gp,3g2,mj2, from 'Seoul.mp4':
  Metadata:
    major_brand     : mp42
    minor_version   : 0
    compatible_brands: mp42mp41
    creation_time   : 2023-04-25T14:17:34.000000Z
  Duration: 00:48:02.88, start: 0.000000, bitrate: 10039 kb/s
  Stream #0:0(eng): Video: h264 (Main) (avc1 / 0x31637661), yuv420p(tv, bt709), 3840x1920, 9999 kb/s, 29.97 fps, 29.97 tbr, 30k tbn, 59.94 tbc (default)
    Metadata:
      creation_time   : 2023-04-25T14:17:34.000000Z
      handler_name    : ?Mainconcept Video Media Handler
      vendor_id       : [0][0][0][0]
      encoder         : AVC Coding
    Side data:
      stereo3d: 2D
      spherical: equirectangular (0.000000/0.000000/0.000000)
```

302 - Do not blink : Question 2

FINANCIAL SECURITY INSTITUTE

Analysis of the mp4 file

Equirectangular projection

- Equirectangular is a type of VR image projection used to represent a 3D spherical environment onto a 2D rectangular image.
- Extracting frames from a equirectangular video, all directions appear in a distorted form.

302 - Do not blink : Question 2

FINANCIAL SECURITY INSTITUTE

Analysis of the mp4 file

Cubemap projection

- Cubemap uses the six faces of a cube as the map shape.
- Each face represent a 360-degree view from six different directions: up, down, left, right, front, and back.
- Extracting frames from an equirectangular video appears undistorted

```
ffmpeg -i Seoul.mp4 -vf v360=equirect:c3x2 output.mp4
```

Convert the video into a cubemap format

```
ffmpeg -i output.mp4 -filter_complex \
"[0:v]split=6[c0][c1][c2][c3][c4][c5]; \
[c0]crop=iw/3:ih/2:0:0[c0]; \
[c1]crop=iw/3:ih/2:iw/3:0[c1]; \
[c2]crop=iw/3:ih/2:2*iw/3:0[c2]; \
[c3]crop=iw/3:ih/2:0:ih/2[c3]; \
[c4]crop=iw/3:ih/2:iw/3:ih/2[c4]; \
[c5]crop=iw/3:ih/2:2*iw/3:ih/2[c5]" \
-map "[c0]" c0.mp4 -map "[c1]" c1.mp4 -map "[c2]" c2.mp4 -map "[c3]" c3.mp4 -map "[c4]" c4.mp4 -map "[c5]" c5.mp4
```

Split the six faces into separate videos

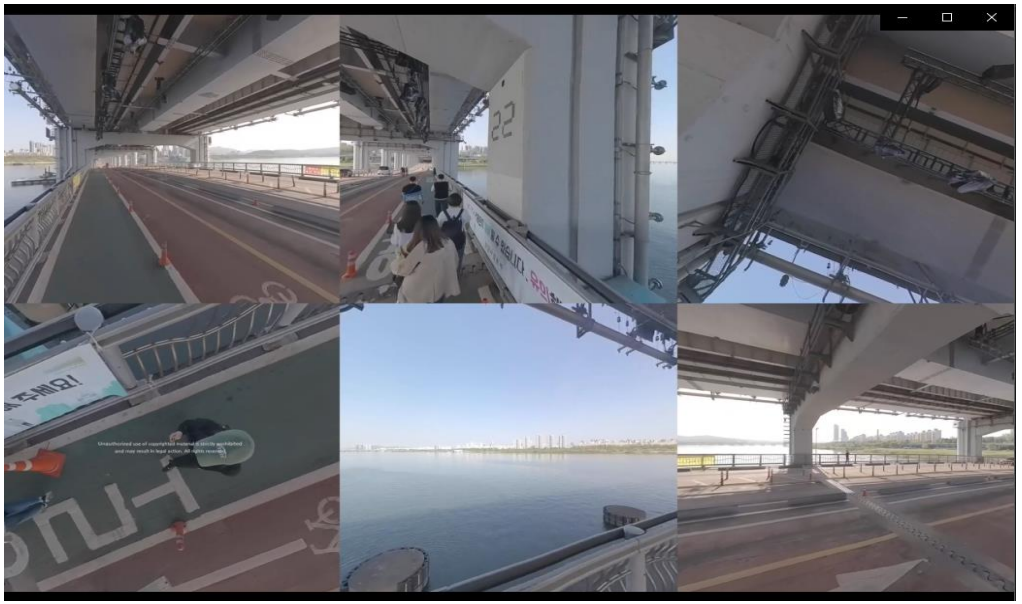
302 - Do not blink : Question 2

FINANCIAL SECURITY INSTITUTE

Analysis of the mp4 file

Cubemap projection

- Cubemap uses the six faces of a cube as the map shape.
- Each face represent a 360-degree view from six different directions: up, down, left, right, front, and back.
- Extracting frames from an equirectangular video appears undistorted

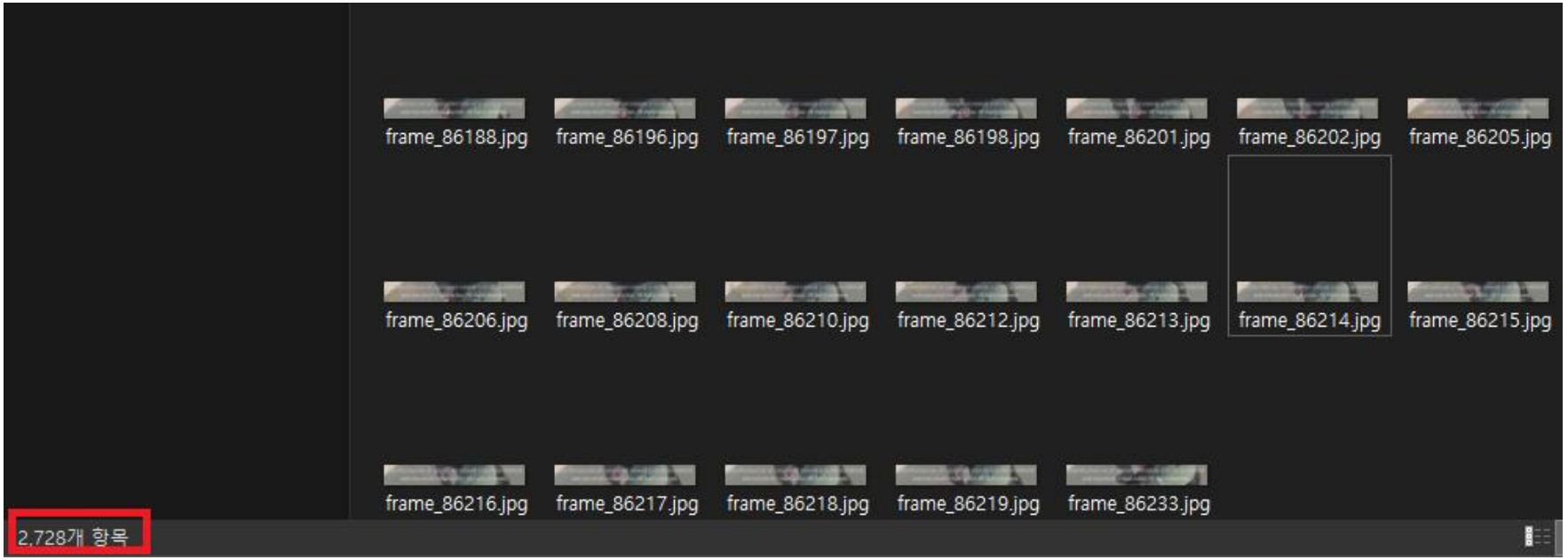


302 - Do not blink : Question 2

FINANCIAL SECURITY INSTITUTE

Find the cryptographic key

- Compare RGB values in the watermark area and save frames when the watermark changes
- Found 2,728 frames that deviated from a specific RGB value



302 - Do not blink : Question 2

FINANCIAL SECURITY INSTITUTE

Find the cryptographic key

- Extract watermark values from 2,728 frames using Tesseract OCR
- Found the cryptographic key in the 72,000th frame.

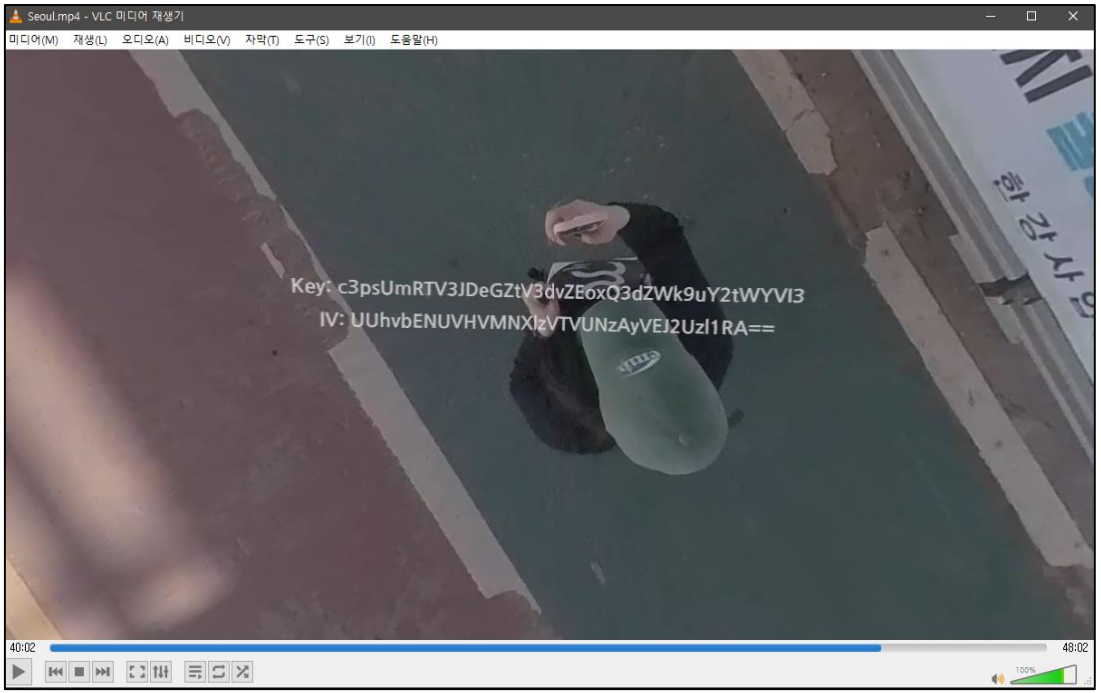
061	2079	frame_70233.jpg	thorized use of copyrightednaterial 짠 strictly prohibitedand may result inJegall@ction. All right re .
062	2080	frame_70234.jpg	: Use of copyrighted material is strictly prohibitedand may result in | ion. All righ!tin legalacti
063	2081	frame_70236.jpg	aAd #Nah use of copyrighte material is strictly prohibitedid may result in legal action. All rights re
065	2083	frame_70267.jpg	Unauthoranduse of copyrighted material is strictly prresult in legaletion. ?쁀ll rights reser
066	2084	frame_70268.jpg	질4Unauthorized use of copyrighted material is strictly prohibitedand may result in legal Action. {All rights reserve
067	2085	frame_70269.jpg	Unauthorandise of copyrighted rial is strictly prohiitedresult in legal agtion. All rights reserves
088	2106	frame_7120.jpg	
090	2108	frame_71478.jpg	
091	2109	frame_71531.jpg	Unauthorized use of copyrighted material is svi conand may result in legal action, All rights reservet
096	2114	frame_718.jpg	aByrioht챕din legal actioy > , |
111	2129	frame_71955.jpg	Unauthorized use of copyrighted m챕ferighisitricty prohibitedand may result in legal aclion ?쁀llgiiiizeserved.
115	2133	frame_72000.jpg	y 4K 1sUmRTV3IDeGZtyStvZE6xO3dZWk9uY2tWYVI3.IV: UUhvbENUVHVMWN: INzAyVEI2UzI1 RA=:
119	2137	frame_721.jpg	
129	2147	frame_72615.jpg	rized use of copyrighted material is stri paneind may resul it in legal action. All rights rese1
140	2158	frame_75903.jpg	d use bat ro ic itedlegal action. All rights a
142	2160	frame_75949.jpg	Usrized use righted mi?쁐ay rest f acti1h
143	2161	frame_75958.jpg	i... is strictlylegal action. All riAll rights

302 - Do not blink : Question 2

FINANCIAL SECURITY INSTITUTE

Question 2 Answer

Key	c3psUmRTV3JDeGZtV3dvZEoxQ3dZWk9uY2tWYVI3
IV	UUhvbENUVHVMNXlzVTVUNzAyVEJ2Uzl1RA==



302 - Do not blink : Question 3

FINANCIAL SECURITY INSTITUTE

Find the encrypted data

- In the 'movi' box, 12MB of hexadecimal string with high entropy is stored after the timestamp.
- Strings with a high entropy are suspected to be encrypted data.

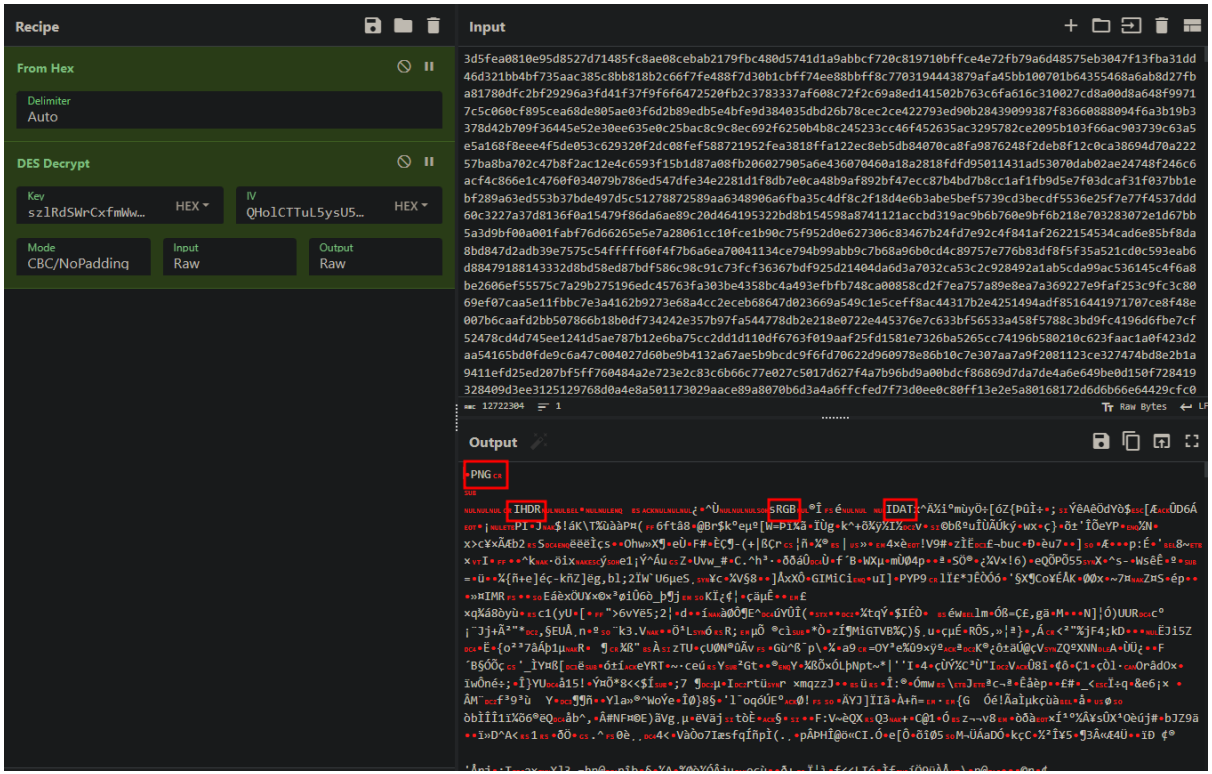
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0123456789ABCDEF
0000h	3D	5F	EA	08	10	E9	5D	85	27	D7	14	85	FC	8A	E0	8C	=_ê..é]…'×.…üŠàŒ
0010h	EB	AB	21	79	FB	C4	80	D5	74	1D	1A	9A	BB	CF	72	0C	ë«!yûÄ€Õt..š»Ïr.
0020h	81	97	10	BF	FC	E4	E7	2F	B7	9A	6D	48	57	5E	B3	04	.—.¿üäç/·šmHW^³.
0030h	7F	13	FB	A3	1D	D4	6D	32	1B	B4	BF	73	5A	AC	38	5C	..û£.Ôm2.´¿sZ¬8\
0040h	8B	B8	18	B2	C6	6F	7F	E4	88	F7	D3	0B	1C	BF	F7	4E	‹¸.²Æo.äˆ÷Ó..¿÷N
0050h	E8	8B	BF	F8	C7	70	31	94	44	38	79	AF	A4	5B	B1	00	è‹¿øÇp1”D8y¯¤[±.
0060h	70	1B	64	35	54	68	A6	AB	8D	27	FB	A8	17	80	DF	C2	p.d5Th¦«.'û¨.€ßÂ
0070h	BF	29	29	6A	3F	D4	1F	37	F9	F6	F6	47	25	20	FB	2C	¿))j?Ô.7ùööG% û,
0080h	37	83	33	7A	F6	08	C7	2F	2C	69	A8	ED	14	15	02	B7	7ƒ3zö.Ç/,i¨í...·
0090h	63	C6	FA	61	6C	31	00	27	CD	8A	00	D8	A6	48	F9	97	cÆúal1.'ÍŠ.Ø¦Hù—
00A0h	17	C5	C0	60	CF	89	5C	EA	68	DE	80	5A	E0	3F	6D	2B	.ÅÀ`Ï‰\êhÞ€Zà?m+
00B0h	89	ED	B5	E4	BF	E9	D3	84	03	5D	BD	26	B7	8C	EC	2C	‰íµä¿éÓ„.]½&·Œì,
00C0h	E4	22	79	3E	D9	0B	28	43	90	99	38	7F	83	66	08	88	ä"y>Ù.(C.™8.ƒf.ˆ
00D0h	09	4F	6A	3B	19	B3	37	8D	42	B7	09	F3	64	45	C5	2E	.Oj;.³7.B·.ódEÅ.
00E0h	30	EE	63	5E	0C	25	BA	C8	C9	C8	EC	69	2F	62	50	B4	0îc^.%ºÈÉÈìi/bP´
00F0h	B8	C2	45	23	3C	C4	6F	45	26	35	AC	32	95	78	2C	E2	¸ÂE#<ÄoE&5¬2•x,â
0100h	09	5B	10	3F	66	AC	90	37	39	C6	3A	5E	5A	16	8F	8E	.[.?f¬.79Æ:^Z..Ž
0110h	EE	4F	5D	E0	53	C6	29	32	0F	2D	C0	8F	EF	58	87	21	îO]àSÆ)2.-À.ïX‡!

302 - Do not blink : Question 3

FINANCIAL SECURITY INSTITUTE

Where is the attack scheduled?

- Decrypting the data with the given Key and IV produces a PNG file.



302 - Do not blink : Question 3

FINANCIAL SECURITY INSTITUTE

Where is the attack scheduled?

- Decrypting the data with the given Key and IV produces a PNG file.

302 - Do not blink : Question 3

FINANCIAL SECURITY INSTITUTE

Where is the attack scheduled?

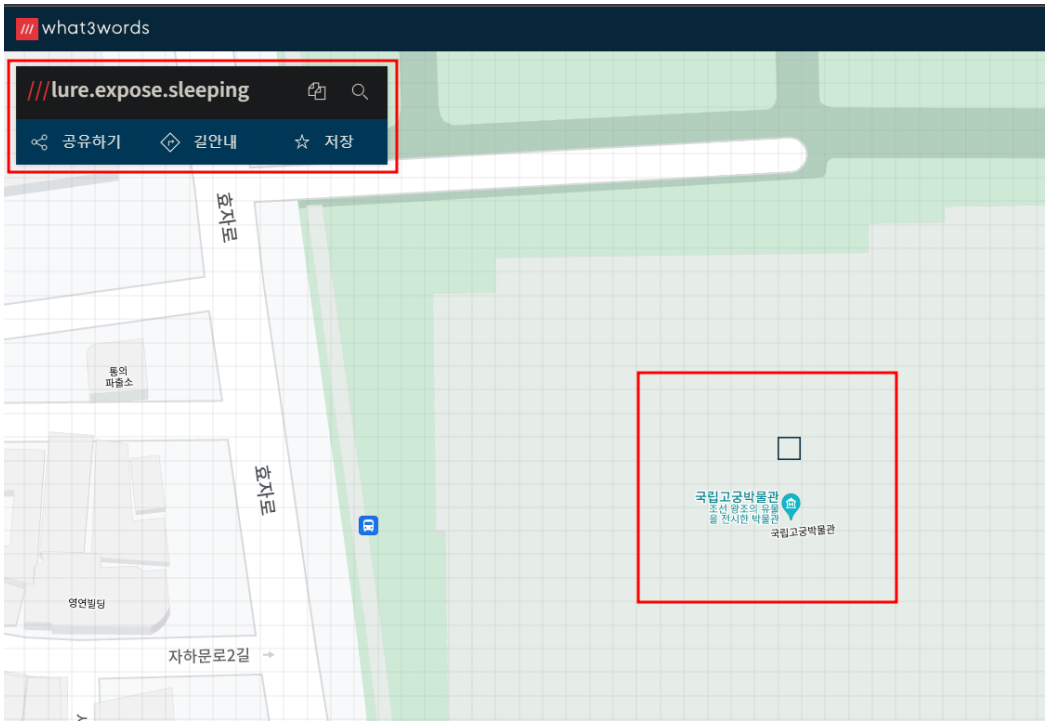
- Recovered the hidden string in the PNG file using StegSolve.

302 - Do not blink : Question 3

FINANCIAL SECURITY INSTITUTE

Question 3 Answer

National Palace Museum of Korea (국립고궁박물관) Entrance



306 - Coin Chaser

FINANCIAL SECURITY INSTITUTE

Description

An investigator was examining a computer in the home of a suspected money launderer and found a cryptocurrency wallet program running. The investigator collected memory dumps and data files from the computer.

Questions 1

When was the crypto wallet program installed? (UTC+0) (20 points)

Questions 2

What time did the suspect encrypt the wallet? (UTC+0) (100 points)

Questions 3

What was the password for the encrypted wallet? (70 points)

Questions 4

Who deposited funds to the suspect's wallet address and who transferred funds from the wallet address? (70 points)

Questions 5

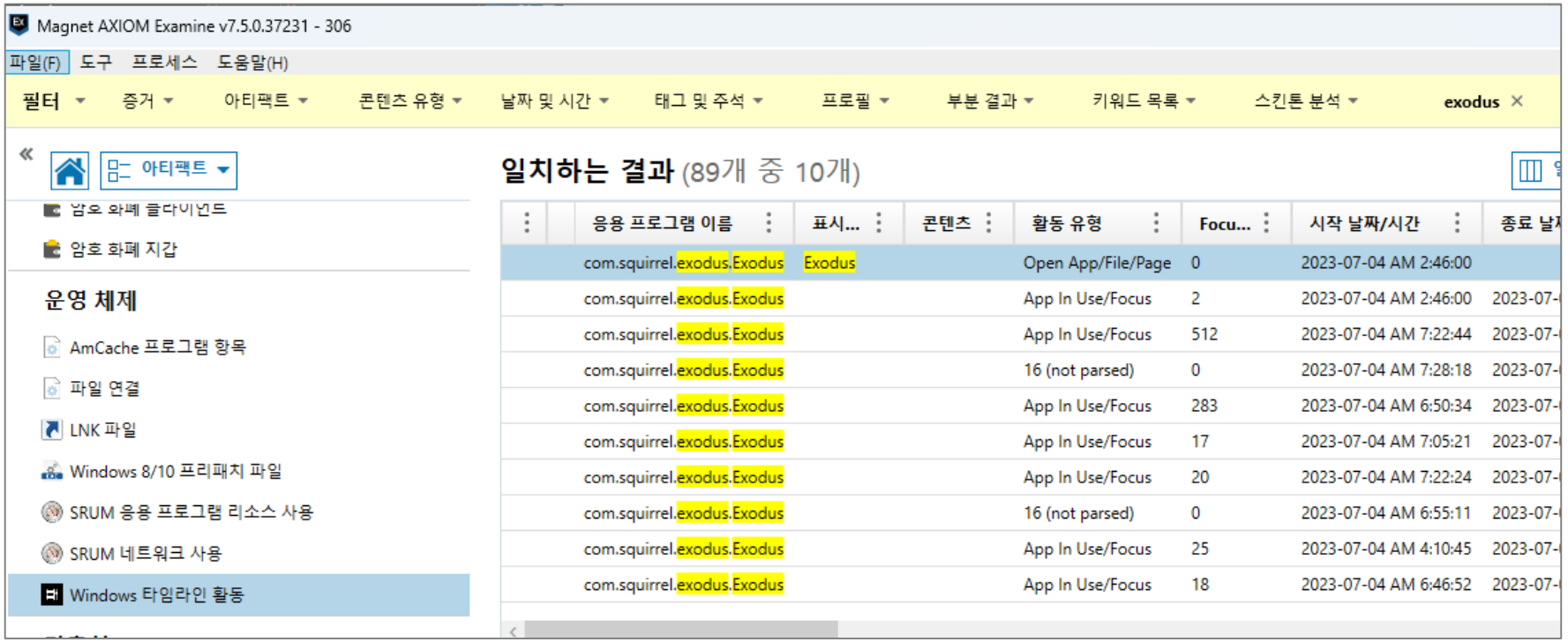
What is the final destination address of the funds transferred from the suspect's wallet address? (40 points)

306 - Coin Chaser : Question 1

FINANCIAL SECURITY INSTITUTE

Find the suspect's crypto wallet program

- Checking the Chrome download history, the suspect downloaded seven wallet program installation files.
- However, among them, the suspect only used the Exodus wallet program.

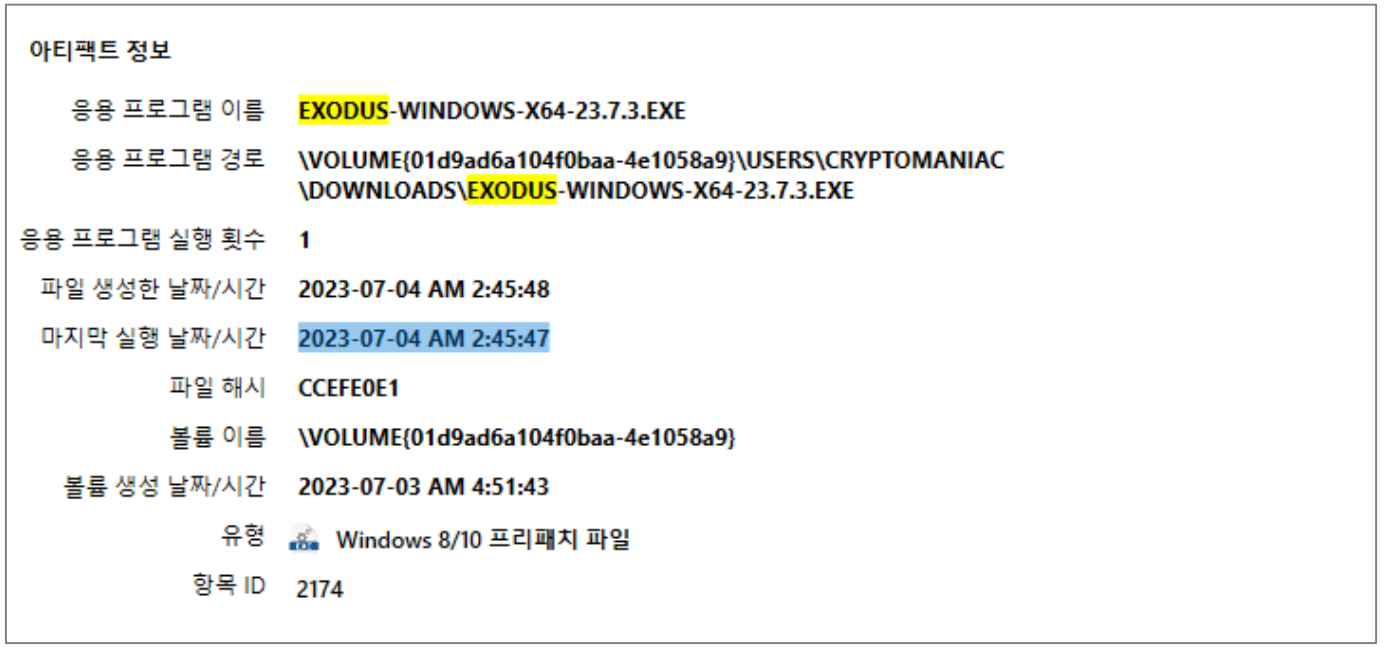


306 - Coin Chaser : Question 1

FINANCIAL SECURITY INSTITUTE

Question 1 Answer

- The Exodus installation program was downloaded from https://downloads.exodus.com/releases/exodus-windows-x64-23.7.3.exe and saved to the path ₩Users₩CryptoManiac₩Downloads₩exodus-windows-x64-23.7.3.exe.
- When checking the Prefetch records, it indicates that the installation program was executed on July 4, 2023, at 02:45:47 (UTC+0). This is the time when the Exodus wallet program was installed.



아티팩트 정보

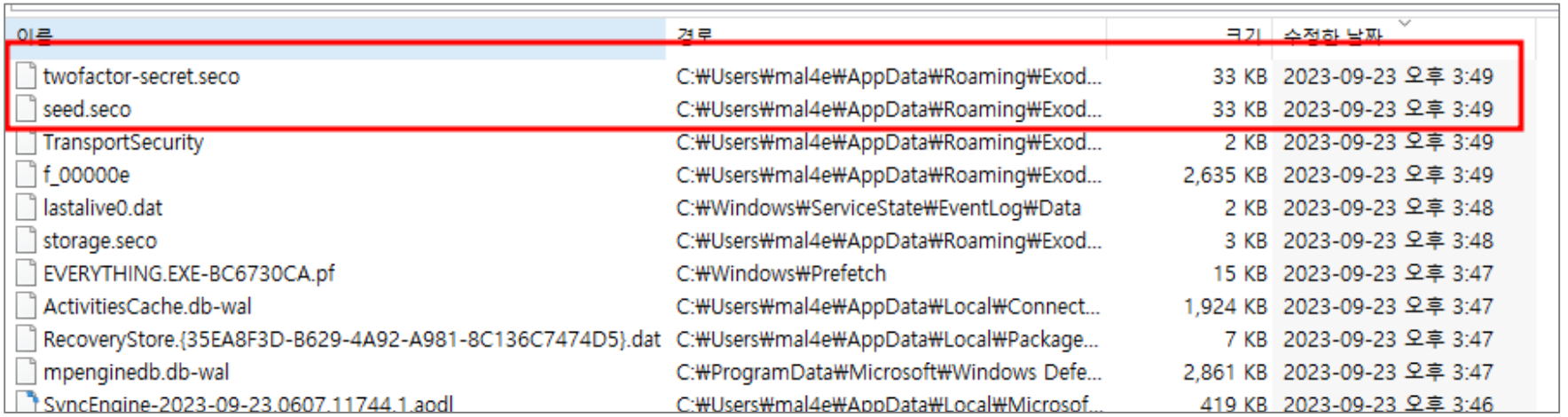
응용 프로그램 이름	EXODUS-WINDOWS-X64-23.7.3.EXE
응용 프로그램 경로	\VOLUME{01d9ad6a104f0baa-4e1058a9}\USERS\CRYPTOMANIAC\DOWNLOADS\EXODUS-WINDOWS-X64-23.7.3.EXE
응용 프로그램 실행 횟수	1
파일 생성한 날짜/시간	2023-07-04 AM 2:45:48
마지막 실행 날짜/시간	2023-07-04 AM 2:45:47
파일 해시	CCEFE0E1
볼륨 이름	\VOLUME{01d9ad6a104f0baa-4e1058a9}
볼륨 생성 날짜/시간	2023-07-03 AM 4:51:43
유형	Windows 8/10 프리패치 파일
항목 ID	2174

306 - Coin Chaser : Question 2

FINANCIAL SECURITY INSTITUTE

What happens when the wallet is encrypted

- The Exodus wallet program version 23.7.3 installed by the suspect allows for wallet encryption using the 'Settings > Backup > Create your password' feature.
- When this function is used to encrypt the wallet, both the ₩AppData₩Roaming₩Exodus₩exodus.wallet₩seed.seco file and the ₩AppData₩Roaming₩Exodus₩exodus.wallet₩twofactor-secret.seco file are modified simultaneously.



이름	경로	크기	수정한 날짜
twofactor-secret.seco	C:₩Users₩mal4e₩AppData₩Roaming₩Exod...	33 KB	2023-09-23 오후 3:49
seed.seco	C:₩Users₩mal4e₩AppData₩Roaming₩Exod...	33 KB	2023-09-23 오후 3:49
TransportSecurity	C:₩Users₩mal4e₩AppData₩Roaming₩Exod...	2 KB	2023-09-23 오후 3:49
f_00000e	C:₩Users₩mal4e₩AppData₩Roaming₩Exod...	2,635 KB	2023-09-23 오후 3:49
lastalive0.dat	C:₩Windows₩ServiceState₩EventLog₩Data	2 KB	2023-09-23 오후 3:48
storage.seco	C:₩Users₩mal4e₩AppData₩Roaming₩Exod...	3 KB	2023-09-23 오후 3:48
EVERYTHING.EXE-BC6730CA.pf	C:₩Windows₩Prefetch	15 KB	2023-09-23 오후 3:47
ActivitiesCache.db-wal	C:₩Users₩mal4e₩AppData₩Local₩Connect...	1,924 KB	2023-09-23 오후 3:47
RecoveryStore.{35EA8F3D-B629-4A92-A981-8C136C7474D5}.dat	C:₩Users₩mal4e₩AppData₩Local₩Package...	7 KB	2023-09-23 오후 3:47
mpenginedb.db-wal	C:₩ProgramData₩Microsoft₩Windows Defe...	2,861 KB	2023-09-23 오후 3:47
SyncEngine-2023-09-23.0607.11744.1.aodl	C:₩Users₩mal4e₩AppData₩Local₩Microsof...	419 KB	2023-09-23 오후 3:46

306 - Coin Chaser : Question 2

FINANCIAL SECURITY INSTITUTE

Question 2 Answer

- Checking the timestamp of the two files in the image, it is evident that they were modified at 06:54:26 (UTC+0) on July 4, 2023. This indicates the time when the suspect encrypted the wallet.

Name	seed.seco
Path	₩Users₩CryptoManiac₩AppData₩Roaming₩Exodus₩exodus.wallet
Full path	₩Users₩CryptoManiac₩AppData₩Roaming₩Exodus₩exodus.wallet₩seed.seco
Existent	√
Description	existing
Ext.	seco
Type	seco
Type status	not verified
Type descr.	seco
Category	Other/unknown type
Evidence object	C
Parent name	exodus.wallet
Size	32.5 KB (33,284)
Created	2023/07/04d04:10:45 +0
Modified	2023/07/04d06:54:26 +0
Accessed	2023/07/04d06:54:26 +0
Attr.	A
ID	5684

Name	twofactor-secret.seco
Path	₩Users₩CryptoManiac₩AppData₩Roaming₩Exodus₩exodus.wallet
Full path	₩Users₩CryptoManiac₩AppData₩Roaming₩Exodus₩exodus.wallet₩twofactor-secret.seco
Existent	√
Description	existing
Ext.	seco
Type	seco
Type status	not verified
Type descr.	seco
Category	Other/unknown type
Evidence object	C
Parent name	exodus.wallet
Size	32.5 KB (33,284)
Created	2023/07/04d04:10:46 +0
Modified	2023/07/04d06:54:26 +0
Accessed	2023/07/04d06:54:26 +0
Attr.	A
ID	5686

306 - Coin Chaser : Question 3

FINANCIAL SECURITY INSTITUTE

Find the encryption key

- When you encrypt the wallet and restart the wallet program, you will need to enter the password again.
- When you enter the password and dump the memory of the Exodus program, it can be observed that the password remains in plaintext.

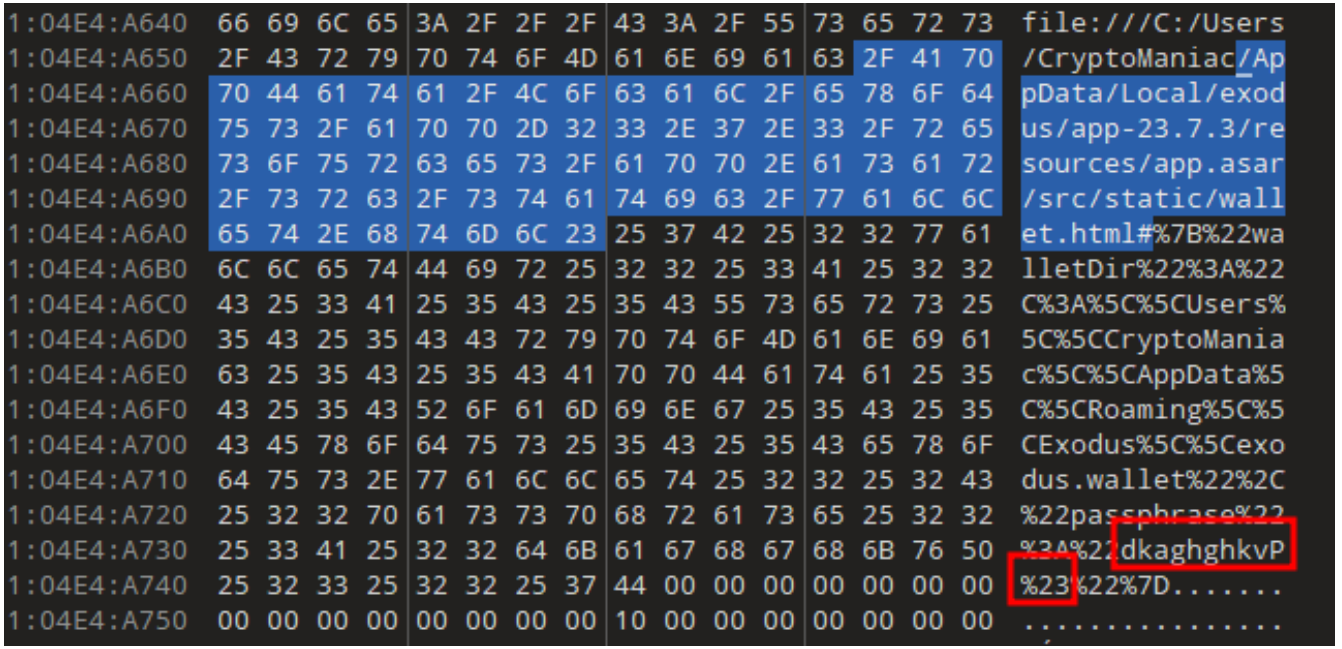
```
0050:2A80  00 00 00 F9 00 00 00 66 69 6C 65 3A 2F 2F 2F 43  ...ù...file:///C
0050:2A90  3A 2F 55 73 65 72 73 2F 6D 61 6C 34 65 2F 41 70  :/Users/mal4e/Ap
0050:2AA0  70 44 61 74 61 2F 4C 6F 63 61 6C 2F 65 78 6F 64  pData/Local/exod
0050:2AB0  75 73 2F 61 70 70 2D 32 33 2E 37 2E 33 2F 72 65  us/app-23.7.3/re
0050:2AC0  73 6F 75 72 63 65 73 2F 61 70 70 2E 61 73 61 72  sources/app.asar
0050:2AD0  2F 73 72 63 2F 73 74 61 74 69 63 2F 77 61 6C 6C  /src/static/wall
0050:2AE0  65 74 2E 68 74 6D 6C 23 25 37 42 25 32 32 77 61  et.html#%7B%22wa
0050:2AF0  6C 6C 65 74 44 69 72 25 32 32 25 33 41 25 32 32  lletDir%22%3A%22
0050:2B00  43 25 33 41 25 35 43 25 35 43 55 73 65 72 73 25  C%3A%5C%5CUsers%
0050:2B10  35 43 25 35 43 6D 61 6C 34 65 25 35 43 25 35 43  5C%5Cmal4e%5C%5C
0050:2B20  41 70 70 44 61 74 61 25 35 43 25 35 43 52 6F 61  AppData%5C%5CRoa
0050:2B30  6D 69 6E 67 25 35 43 25 35 43 45 78 6F 64 75 73  ming%5C%5CExodus
0050:2B40  25 35 43 25 35 43 65 78 6F 64 75 73 2E 77 61 6C  %5C%5Cexodus.wal
0050:2B50  6C 65 74 25 32 32 25 32 43 25 32 32 70 61 73 73  let%22%2C%22pass
0050:2B60  70 68 72 61 73 65 25 32 32 25 33 41 25 32 32 64  phrase%22%3A%22d
0050:2B70  66 63 31 71 32 77 33 65 34 72 25 32 32 25 37 44  fc1q2w3e4r%22%7D
0050:2B80  00 00 00 05 11 00 00 06 00 00 00 55 6B 8D 00 E5  ...........Uk..å
```

306 - Coin Chaser : Question 3

FINANCIAL SECURITY INSTITUTE

Find the encryption key

- To search for the wallet password in the suspect's PC memory file, an attempt was made to perform a string search in the memory file using the string '/AppData/Local/exodus/app-23.7.3/resources/app.asar/src/static/wallet.html#' located near the password.
- As a result, a string 'dkaghghkvP#' was discovered, which is suspected to be the password.



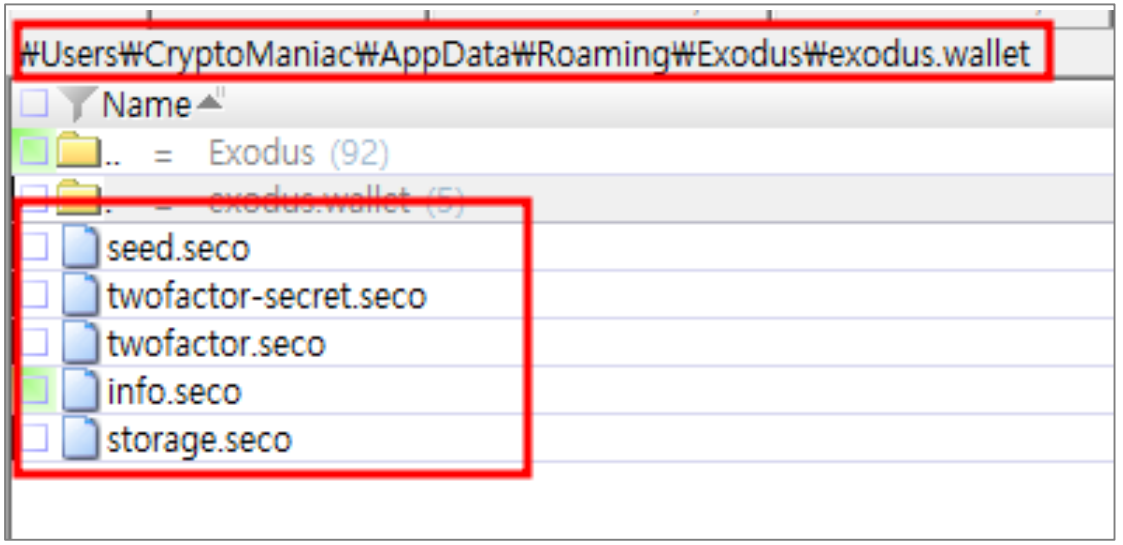
```
1:04E4:A640  66 69 6C 65 3A 2F 2F 2F 43 3A 2F 55 73 65 72 73  file:///C:/Users
1:04E4:A650  2F 43 72 79 70 74 6F 4D 61 6E 69 61 63 2F 41 70  /CryptoManiac/Ap
1:04E4:A660  70 44 61 74 61 2F 4C 6F 63 61 6C 2F 65 78 6F 64  pData/Local/exod
1:04E4:A670  75 73 2F 61 70 70 2D 32 33 2E 37 2E 33 2F 72 65  us/app-23.7.3/re
1:04E4:A680  73 6F 75 72 63 65 73 2F 61 70 70 2E 61 73 61 72  sources/app.asar
1:04E4:A690  2F 73 72 63 2F 73 74 61 74 69 63 2F 77 61 6C 6C  /src/static/wall
1:04E4:A6A0  65 74 2E 68 74 6D 6C 23 25 37 42 25 32 32 77 61  et.html#%7B%22wa
1:04E4:A6B0  6C 6C 65 74 44 69 72 25 32 32 25 33 41 25 32 32  lletDir%22%3A%22
1:04E4:A6C0  43 25 33 41 25 35 43 25 35 43 55 73 65 72 73 25  C%3A%5C%5CUsers%
1:04E4:A6D0  35 43 25 35 43 43 72 79 70 74 6F 4D 61 6E 69 61  5C%5CCryptoMania
1:04E4:A6E0  63 25 35 43 25 35 43 41 70 70 44 61 74 61 25 35  c%5C%5CAppData%5
1:04E4:A6F0  43 25 35 43 52 6F 61 6D 69 6E 67 25 35 43 25 35  C%5CRoaming%5C%5
1:04E4:A700  43 45 78 6F 64 75 73 25 35 43 25 35 43 65 78 6F  CExodus%5C%5Cexo
1:04E4:A710  64 75 73 2E 77 61 6C 6C 65 74 25 32 32 25 32 43  dus.wallet%22%2C
1:04E4:A720  25 32 32 70 61 73 73 70 68 72 61 73 65 25 32 32  %22passphrase%22
1:04E4:A730  25 33 41 25 32 32 64 6B 61 67 68 67 68 6B 76 50  %3A%22dkaghghkvP
1:04E4:A740  25 32 33 25 32 32 25 37 44 00 00 00 00 00 00 00  %23%22%7D.......
1:04E4:A750  00 00 00 00 00 00 00 00 10 00 00 00 00 00 00 00  ................
```

306 - Coin Chaser : Question 3

FINANCIAL SECURITY INSTITUTE

Question 3 Answer

- If the five files located in the ₩Users₩CryptoManiac₩AppData₩Roaming₩Exodus₩exodus.wallet folder on the suspect's PC, which contain the wallet data, are copied to the same path on a test environment PC where Exodus is installed, it is possibleto run the wallet program with the suspect's wallet data.
- When the wallet program is executed, and the password 'dkaghghkvP#' is entered, the wallet is successfully decrypted, allowing access to the suspect's wallet data.

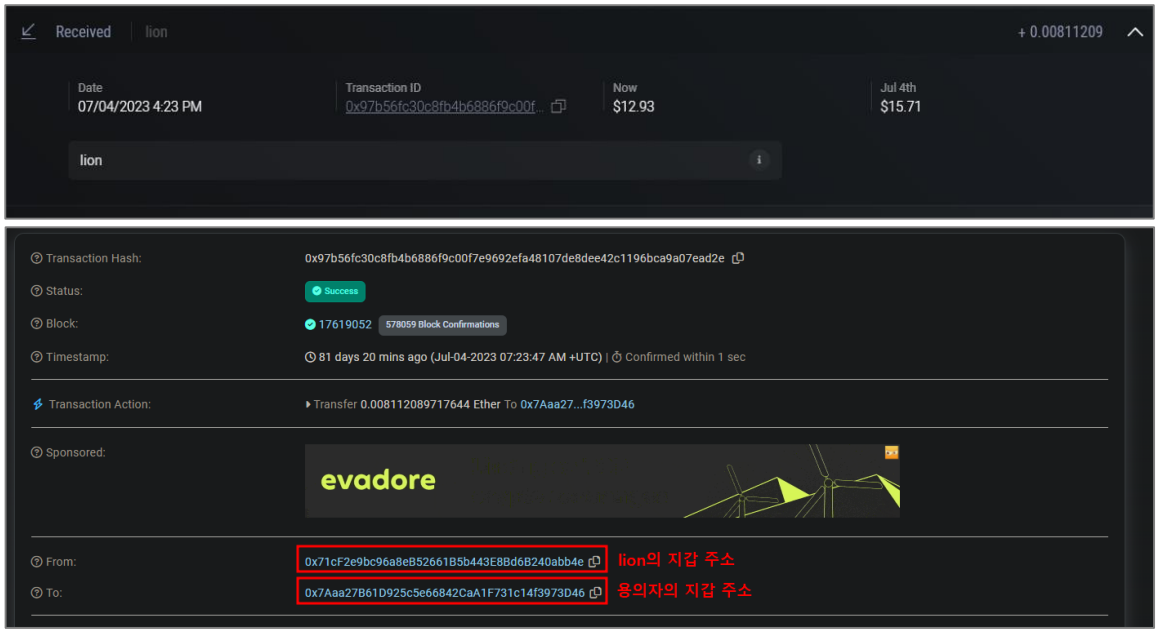


306 - Coin Chaser : Question 4

FINANCIAL SECURITY INSTITUTE

Question 4 Answer

- When checking the suspect's wallet, only Ethereum Activity is found.
- The person who deposited funds into the suspect's wallet is 'lion,' and the Transaction ID for this transaction is 0x97b56fc30c8fb4b6886f9c00f7e9692efa48107de8dee42c1196bca9a07ead2e.
- Using Etherscan to verify this transaction record, it can be determined that lion's wallet address is 0x71cF2e9bc96a8eB52661B5b443E8Bd6B240abb4e.

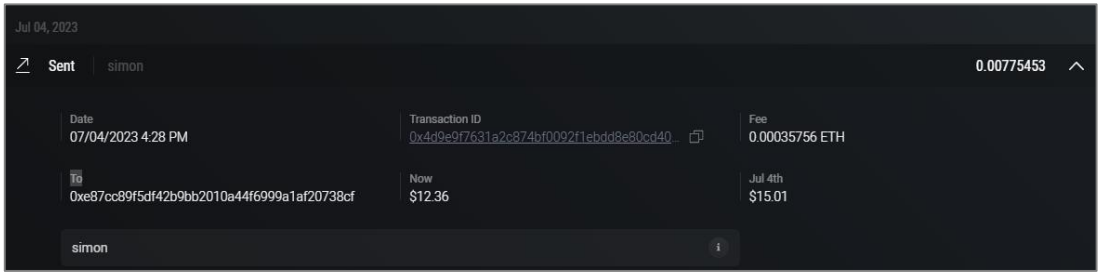


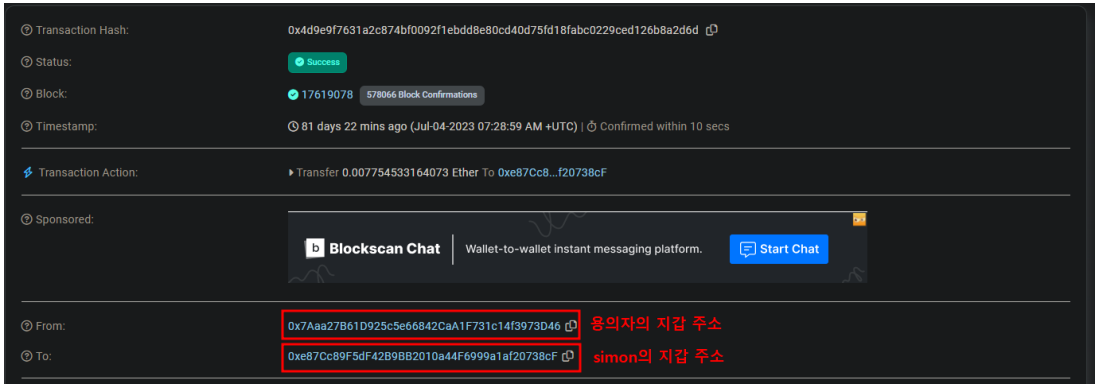
306 - Coin Chaser : Question 4

FINANCIAL SECURITY INSTITUTE

Question 4 Answer

- On the other hand, the recipient to whom the suspect transferred funds is 'simon,' and the Transaction ID for this transaction is 0x4d9e9f7631a2c874bf0092f1ebdd8e80cd40d75fd18fabc0229ced126b8a2d6d.
- Using Etherscan to verify this transaction record, it can be determined that simon's wallet address is 0xe87Cc89F5dF42B9BB2010a44F6999a1af20738cF.



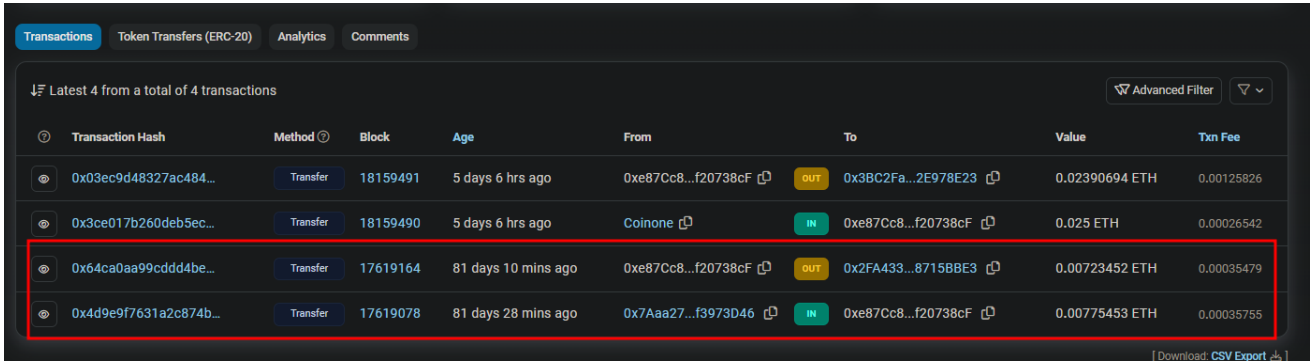


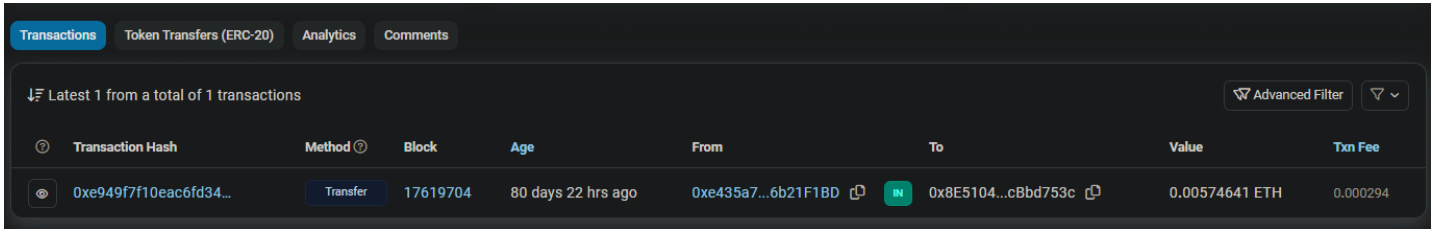
306 - Coin Chaser : Question 5

FINANCIAL SECURITY INSTITUTE

Question 5 Answer

- When checking Simon's transaction records, it is evident that he transferred the funds received from the suspect to the address 0x2FA433c7A349CE4737c32578Ac5d5be28715BBE3.
- Following the transaction history of the wallet, it can be determined that the received funds are being held and not furthertransmitted to another wallet.
- In other words, 0x8E510474bA2F602f2D27BDdd68C02A85cBbd753c is the final destination address.





401 - Find an illegally filmed image

FINANCIAL SECURITY INSTITUTE

Description

The police obtained a Pixel 3, Android 10 version smartphone with illegally filmed images of the suspect in the illegal filming incident. Just before confiscation, the suspect locked the screen of the smartphone he was using, and the suspect does not give out the password to unlock the smartphone. For the smartphone, kernel memory dump and encrypted user data partition image were obtained through a mobile forensic tool. Decrypt illegally shot images to collect evidence.

Questions 1

Obtain all FBE Master Keys of the acquired smartphone. (60 points)

Questions 2

Find all the file names of image files in the DCIM folder. (60 points)

Questions 3

Find all of the following 4 types of Extend Attributes related to FBE for each image file obtained in Problem 2. (80 points)

Questions 4

Obtain all the Derivation Encryption Keys for each image file obtained in Problem 2. (100 points)

Questions 5

Decrypt the data of the image file with largest file size among the image files in the DCIM folder, and submit a list of all tweak values used in the decryption process and the SHA256 hash value of the decrypted image file. (100 points)

401 - Find an illegally filmed image: Question 1

FINANCIAL SECURITY INSTITUTE

Use Open Source tool (fbekeyfind)

- Obtain FBE keys from memory dumps is already a open source tool made python.
- To use this tool, must have aeskeyfind, The Sleuth Kit(ext4-fbe) pre-installed.

Get the Code

- The Sleuth Kit: https://faui1-gitlab.cs.fau.de/tobias.gross/sleuthkit-ext4-fbe
- pytsk: https://faui1-gitlab.cs.fau.de/tobias.gross/pytsk3
- dfVFS: https://faui1-gitlab.cs.fau.de/tobias.gross/dfvfs
- Plaso: https://faui1-gitlab.cs.fau.de/tobias.gross/plaso
- fbekeyfind: https://faui1-files.cs.fau.de/public/one_key_to_rule/fbekeyfind.tar.gz

Dr.-Ing. Tobias Groß

Department of Computer Science
Chair of Computer Science 1 (IT Security Infrastructures)

tobias.gross@cs.fau.de
https://www1.informatik.uni-erlangen.de/staff/tobiasgross

- https://www.cs1.tf.fau.de/research/system-security-group/one-key-to-rule/

401 - Find an illegally filmed image: Question 1

FINANCIAL SECURITY INSTITUTE

Use Open Source tool (fbekeyfind)

- Derive the top two key candidates distributed in the memory dump.
- Usage : python findMasterKeys.py --part [userdata.img] [kernel_ram.dmp]

```
root@ddddh:~/windows/dfc/401/401 - Find an illegally filmed image/fbekeyfind# python3 findMasterKeys.py --part ../dump_images/userdata.img ../dump_images/ramdump
Extracting Encryption Attributes |################################| 2369/2369
Found 1843 Files with Encryption Attributes in Image ../dump_images/userdata.img
Using aeskeyfind
Found 1877 AES-256-Bit Keys in Memory Dump ../dump_images/ramdump
0 of 1843 have unexpected encryption mode
Cracking |################################| 3457468/3457468
Result for Master Key Descriptor: ec34dd2d7a20bf2e
Found Keys: a45d06abee12df0b54cbf332b936d2fe937dd7cae36074e8f42a828d509144f4, 6b65d32093a85bb36e70dfd8540f9de5c6e0a7553f8c50e64a893f05796904db
With Hits: 553, 48, 1, 1, 1, 1, 1, 1, 1, 1
Result for Master Key Descriptor: e8aece005a118a72
Found Keys: 15584d919ff3a922b18f699dc845944a18ba1d8263666aa69269682915652d49, b9bcd156330e8b27adce3b067619a959eae170a59e262e705f8200498dfb3e86
With Hits: 289, 218, 1, 1, 1, 1, 1, 1, 1, 1
Result for Master Key Descriptor: aff2ea76f5190ee6
Found Keys: f8a84e33e0c4efba5908da3cd1dd2349eae901aa5f9bc99b9f218434f074faef, ae96a98ef97afa52e8545f12475640361358f38706153e225408f09e8ee57483
With Hits: 491, 27, 1, 1, 1, 1, 1, 1, 1, 1
```

401 - Find an illegally filmed image: Question 1

FINANCIAL SECURITY INSTITUTE

Question 1 Answer

Descriptor	Master Key 1	Master Key 2
ec34dd2d7a20bf2e	• a45d06abee12df0b54cbf332b936d2fe937dd7cae36074e8f42a828d509144f4	• 6b65d32093a85bb36e70dfd8540f9de5c6e0a7553f8c50e64a893f05796904db
e8aece005a118a72	• 15584d919ff3a922b18f699dc845944a18ba1d8263666aa69269682915652d49	• b9bcd156330e8b27adce3b067619a959eae170a59e262e705f8200498dfb3e86
aff2ea76f5190ee6	• f8a84e33e0c4efba5908da3cd1dd2349eae901aa5f9bc99b9f218434f074faef	• ae96a98ef97afa52e8545f12475640361358f38706153e225408f09e8ee57483

ec34dd2d7a20bf2e:a45d06abee12df0b54cbf332b936d2fe937dd7cae36074e8f42a828d509144f4:6b65d32093a85bb36e70dfd8540f9de5c6e0a7553f8c50e64a893f05796904db
e8aece005a118a72:15584d919ff3a922b18f699dc845944a18ba1d8263666aa69269682915652d49:b9bcd156330e8b27adce3b067619a959eae170a59e262e705f8200498dfb3e86
aff2ea76f5190ee6:f8a84e33e0c4efba5908da3cd1dd2349eae901aa5f9bc99b9f218434f074faef:ae96a98ef97afa52e8545f12475640361358f38706153e225408f09e8ee57483

401 - Find an illegally filmed image: Question 2

FINANCIAL SECURITY INSTITUTE

Question 2 Answer

- Use the TSK(The Sleuth Kit)-fls to figure out the names or inodes of files in the file system
- Usage : fls –K [fbe.keys] –rF [userdata.img] | grep DCIM

```
root@ddddh:~/windows/dfc/401/401 - Find an illegally filmed image/fbekeyfind# ./fstools/fls -K fbe.keys  -rF ../dump_images/userdata.img | grep DCIM
r/r $ 106511:   media/0/DCIM/52e3d54a4855ae14f1dc8460962e33791c3ad6e04e507749742c78d6944cc3_640.jpg
r/r $ 106512:   media/0/DCIM/52e4d5414a5ab10ff3d8992cc12c30771037dbf852547848702a7fd0954b_640.jpg
r/r $ 106513:   media/0/DCIM/52e5d7434253a814f1dc8460962e33791c3ad6e04e507440742a7ad2974bc1_640.jpg
r/r $ 106514:   media/0/DCIM/55e6dd474b5baa14f1dc8460962e33791c3ad6e04e5074417c2d78d19f48c3_640.jpg
r/r $ 106515:   media/0/DCIM/55e8d5424b56a514f1dc8460962e33791c3ad6e04e5074417d2e72d29e4ac3_640.jpg
r/r $ 106516:   media/0/DCIM/55e8dd4a4255b10ff3d8992cc12c30771037dbf85254794e73277bd7954a_640.jpg
r/r $ 106517:   media/0/DCIM/black-coffee-1867753_640.jpg
r/r $ 106518:   media/0/DCIM/g312a1db9dd11930ce7698981e6f1353258fa24ff7faae68e148ffdd4b16d18e958e78de6751cd7df595657572cb372c9_640.jpg
r/r $ 106519:   media/0/DCIM/g91d447ecfe72f9eec67075695beb60be3f06e9f341d675a76e847a2fd150139425d7ca3a1de19130389065a4706df7a5_640.jpg
r/r $ 106520:   media/0/DCIM/g9eea99cb46a0b08d4521d561dd9788383f3163d335eb709f68476e600561afdcaa297e9090a7cf64b66266cc6374d867_640.jpg
r/r $ 106521:   media/0/DCIM/peas-580333_640.jpg
```

media/0/DCIM/52e3d54a4855ae14f1dc8460962e33791c3ad6e04e507749742c78d6944cc3_640.jpg
media/0/DCIM/52e4d5414a5ab10ff3d8992cc12c30771037dbf852547848702a7fd0954b_640.jpg
media/0/DCIM/52e5d7434253a814f1dc8460962e33791c3ad6e04e507440742a7ad2974bc1_640.jpg
media/0/DCIM/55e6dd474b5baa14f1dc8460962e33791c3ad6e04e5074417c2d78d19f48c3_640.jpg
media/0/DCIM/55e8d5424b56a514f1dc8460962e33791c3ad6e04e5074417d2e72d29e4ac3_640.jpg
media/0/DCIM/55e8dd4a4255b10ff3d8992cc12c30771037dbf85254794e73277bd7954a_640.jpg
media/0/DCIM/black-coffee-1867753_640.jpg
media/0/DCIM/g312a1db9dd11930ce7698981e6f1353258fa24ff7faae68e148ffdd4b16d18e958e78de6751cd7df595657572cb372c9_640.jpg
media/0/DCIM/g91d447ecfe72f9eec67075695beb60be3f06e9f341d675a76e847a2fd150139425d7ca3a1de19130389065a4706df7a5_640.jpg
media/0/DCIM/g9eea99cb46a0b08d4521d561dd9788383f3163d335eb709f68476e600561afdcaa297e9090a7cf64b66266cc6374d867_640.jpg
media/0/DCIM/peas-580333_640.jpg

401 - Find an illegally filmed image: Question 3

FINANCIAL SECURITY INSTITUTE

Get file attributes

- Use the TSK(The Sleuth Kit)-istat to get a lot of information about a file, including its encryption mode
- Usage : istat –f [filesystem] [userdata.img] [inode]

```
root@ddddh:~/windows/dfc/401/401 - Find an illegally filmed image/fbekeyfind# ./fstools/istat -f ext4 ../dump_images/userdata.img 106511
inode: 106511
Allocated
Group: 13
Generation Id: 2987584812
uid / gid: 1023 / 1057
mode: rrw-rw-r--
Flags: No A-Time, Compression Error, Extents,
size: 72530
num of links: 1

Extended Attributes  (Block: 426515)
security.selinux=u:object_r:media_rw_data_file:s0

Extended Attributes  (Inode Included)
FBE Content Mode: 1 (AES 256 XTS)
FBE Name Mode: 4 (AES 256 CTS)
FBE Key Descriptor: EC 34 DD 2D 7A 20 BF 2E
FBE Nonce: D6 69 8B 2F 31 8D C3 E5 93 4F 9F C4 14 53 FD 51

Inode Times:
Accessed:       2023-07-07 20:25:38.000000000 (KST)
File Modified:  2023-07-07 20:25:38.000000000 (KST)
Inode Modified: 2023-07-07 20:27:54.098443709 (KST)
File Created:   2023-07-07 20:27:54.098443709 (KST)

Direct Blocks:
426516 426517 426518 426519 426520 426521 426522 426523
426524 426525 426526 426527 426528 426529 426530 426531
426532 426533
```

401 - Find an illegally filmed image: Question 3

FINANCIAL SECURITY INSTITUTE

Developing automation tools

- Written code to get attributes of the target files using python's pwntools library
- First, Make a function to get the inodes of the target files using fls

```
def run_fls(keyfile, imgfile, filter=''):
    with process(['./fstools/fls', '-K', keyfile, '-rF', imgfile]) as p:

        result = []
        while True:
            try:
                line = p.recvline().decode(errors='ignore')
                if filter in line:
                    inode    = line.split(':')[0].split(' ')[-1]
                    filename = line.split(':')[1][:-1]

                    result.append( {'inode':inode,
                                    'filename':filename})
            except EOFError:
                break

    return result
```

401 - Find an illegally filmed image: Question 3

FINANCIAL SECURITY INSTITUTE

Developing automation tools

- Written code to get attributes of the target files using python's pwntools library
- Second, Use istat to get the file attributes

```
def run_istat(imgfile, inode):
    with process(['./fstools/istat', '-f', 'ext4', imgfile, inode]) as p:
        result = {'content_mode_value':'', 'title_mode_value':'', 'descriptor':'', 'nonce':''}
        while True:
            try:
                line = p.recvline().decode(errors='ignore')

                if 'FBE Content Mode:' in line:
                    result['content_mode_value'] = line.split(":")[1][1:2]

                elif 'FBE Name Mode:' in line:

                    result['title_mode_value'] = line.split(":")[1][1:2]

                elif 'FBE Key Descriptor' in line:
                    result['descriptor'] = line.split(":")[1][:-1].replace(" ", "")

                elif 'FBE Nonce' in line:
                    result['nonce'] = line.split(":")[1][:-1].replace(" ", "")

            except EOFError:
                break

    return result
```

401 - Find an illegally filmed image: Question 3

FINANCIAL SECURITY INSTITUTE

Question 3 Answer

Note

- All target files have a content encryption mode of 1(AES-256-XTS) and a name encryption mode of 4(AES-256-CTS)

52e3d54a4855ae14f1dc8460962e33791c3ad6e04e507749742c78d6944cc3_640.jpg:1:4:EC34DD2D7A20BF2E:D6698B2F318DC3E5934F9FC41453FD51
52e4d5414a5ab10ff3d8992cc12c30771037dbf852547848702a7fd0954b_640.jpg:1:4:EC34DD2D7A20BF2E:F74D2D27FB6DE66BEB50431CBEDB2F49
52e5d7434253a814f1dc8460962e33791c3ad6e04e507440742a7ad2974bc1_640.jpg:1:4:EC34DD2D7A20BF2E:CF92299A65DB6027C56EE9DA77B8C3B8
55e6dd474b5baa14f1dc8460962e33791c3ad6e04e5074417c2d78d19f48c3_640.jpg:1:4:EC34DD2D7A20BF2E:DFAD5017231C4B2D61E181D41D0939BB
55e8d5424b56a514f1dc8460962e33791c3ad6e04e5074417d2e72d29e4ac3_640.jpg:1:4:EC34DD2D7A20BF2E:949CEC68E90588FD715096E1B1AA5D3D
55e8dd4a4255b10ff3d8992cc12c30771037dbf85254794e73277bd7954a_640.jpg:1:4:EC34DD2D7A20BF2E:5FCBBF00942D371E1B73ACE806786AFA
black-coffee-1867753_640.jpg:1:4:EC34DD2D7A20BF2E:51C34B6909EA3598DAAD02214F2AF0F6
g312a1db9dd11930ce7698981e6f1353258fa24ff7faae68e148ffdd4b16d18e958e78de6751cd7df595657572cb372c9_640.jpg:1:4:EC34DD2D7A20BF2E:4BB0AB4E5C3AC8F00B14260418402A06
g91d447ecfe72f9eec67075695beb60be3f06e9f341d675a76e847a2fd150139425d7ca3a1de19130389065a4706df7a5_640.jpg:1:4:EC34DD2D7A20BF2E:904D446A5F5A7710BF87A4DCCB59851E
g9eea99cb46a0b08d4521d561dd9788383f3163d335eb709f68476e600561afdcaa297e9090a7cf64b66266cc6374d867_640.jpg:1:4:EC34DD2D7A20BF2E:AC4589492E37397F1E26343E2D8C57D7
peas-580333_640.jpg:1:4:EC34DD2D7A20BF2E:2674FC48B1E313579AFB5C25C80AADC6

401 - Find an illegally filmed image: Question 4

FINANCIAL SECURITY INSTITUTE

Implementation of the key derivation

- Proceed to encrypt the master key in AES-128-ECB mode using the nonce value as the key
- The generated value is used as a key when decrypting the encrypted file contents

```
1  static int derive_key_aes(u8 deriving_key[FS_AES_128_ECB_KEY_SIZE],
2                                          const struct fscrypt_key *source_key,
3                                          u8 derived_raw_key[FS_MAX_KEY_SIZE])
4  {
5          /* ... */
6          struct crypto_skcipher *tfm = crypto_alloc_skcipher("ecb(aes)", 0, 0);
7          /* ... */
8          res = crypto_skcipher_setkey(tfm, deriving_key,
9                                                  FS_AES_128_ECB_KEY_SIZE);
10         /* ... */
11         sg_init_one(&src_sg, source_key->raw, source_key->size);
12         sg_init_one(&dst_sg, derived_raw_key, source_key->size);
13         skcipher_request_set_crypt(req, &src_sg, &dst_sg, source_key->size,
14                                                           NULL);
15         res = crypto_wait_req(crypto_skcipher_encrypt(req), &wait);
16         /* ... */
17         return res;
18 }
```

Listing 1: Implementation of the key derivation function (KDF) in the Android kernel source.

- One Key to Rule Them All: Recovering the Master Key from RAM to break Android's File-Based Encryption – 5p

401 - Find an illegally filmed image: Question 4

FINANCIAL SECURITY INSTITUTE

Developing automation tools

- The automation tool developed in Prob3 can get the nonce value withistat function

```
def derive_key_aes(master_key, nonce):
    cipher = Cipher(algorithms.AES(nonce), modes.ECB(), backend=default_backend())
    encryptor = cipher.encryptor()

    derived_key = encryptor.update(master_key) + encryptor.finalize()
    return derived_key
```

401 - Find an illegally filmed image: Question 4

FINANCIAL SECURITY INSTITUTE

Question 4 Answer

52e3d54a4855ae14f1dc8460962e33791c3ad6e04e507749742c78d6944cc3_640.jpg:d6698b2f318dc3e5934f9fc41453fd51:399a4ca0c2af48e9d7f8448c1dcb6b32ef5a267852252ba38b2f7870a3b20444c81feb383c3e0d06958390e929590e085e7a702b3e0de1ff6e7fefde3b5c5825
52e4d5414a5ab10ff3d8992cc12c30771037dbf852547848702a7fd0954b_640.jpg:f74d2d27fb6de66beb50431cbedb2f49:6c1c651e49eedf560dbdf6254144390241ffdc956eb72646b26dae5eac974c29974aaa1abd4a709dac3f21093b25a1e170538a46b818fb55de9b9760c6f0d983
52e5d7434253a814f1dc8460962e33791c3ad6e04e507440742a7ad2974bc1_640.jpg:cf92299a65db6027c56ee9da77b8c3b8:32a1fcaf50cd0c5fe65c77d4feab8ee499ca56f387660ff871ea1ed26e9a0f840a20e2c2392b49820685085c9318dcab2c51722c2d2c992dae884227956ccdb9
55e6dd474b5baa14f1dc8460962e33791c3ad6e04e5074417c2d78d19f48c3_640.jpg:dfad5017231c4b2d61e181d41d0939bb:4c2633b0025cf260c06f161ffaf63eec0374f266a25b06d945b41791b705bf1fda2062e5e5276c915d6b4c139839487d08014c937323788060326c5fa9089be3
55e8d5424b56a514f1dc8460962e33791c3ad6e04e5074417d2e72d29e4ac3_640.jpg:949cec68e90588fd715096e1b1aa5d3d:f0fcddc1e3acb4b1ab1af541d4080d1fd4e27e661f0a3d4bb808eccad569bed6971193b929ea4a4f07f78ca017f1bc5182e4e72d956bb2215081abd1efc16f50
55e8dd4a4255b10ff3d8992cc12c30771037dbf85254794e73277bd7954a_640.jpg:5fcbbf00942d371e1b73ace806786afa:4289bb1d732afb0c4e13bddc01cbc9e13f993f059b3a4d03ed75553741f63244a79b8dc1b12637452514d254f47c29c745b12cac2c9488f720c4c85aba05852d
black-coffee-1867753_640.jpg:51c34b6909ea3598daad02214f2af0f6:ce65d1067b0e99ea24969f5d68fa37479e2e5c2938d7ecd73e48e165b5255d80bd2ca92e93cd7bdbc6c67371264db5acc7b48ff91225f2252dfb51d3a997cad1
g312a1db9dd11930ce7698981e6f1353258fa24ff7faae68e148ffdd4b16d18e958e78de6751cd7df595657572cb372c9_640.jpg:4bb0ab4e5c3ac8f00b14260418402a06:9723762d0cbaf02619ef0640e7679b679563f4c348fef286998587f37e157961b72b877f3d01bed88b3dfe5ec2ee4d2940d6472fc3c11641f9c8e81150d5e545
g91d447ecfe72f9eec67075695beb60be3f06e9f341d675a76e847a2fd150139425d7ca3a1de19130389065a4706df7a5_640.jpg:904d446a5f5a7710bf87a4dccb59851e:1f1edae1d15e44e583f1408a63495e5fef299da257647f1d7ad9e3aaabf91a63ed7d6400235f5f4789883cefd0987ac1552f30751fdb5fd23eb2e5189986974c
g9eea99cb46a0b08d4521d561dd9788383f3163d335eb709f68476e600561afdcaa297e9090a7cf64b66266cc6374d867_640.jpg:ac4589492e37397f1e26343e2d8c57d7:51b351438327ef7d04da0c5df809b41f21561af2fab87665453a86fe2b5a602c2daa2e71c00e77bbd199f2c8fa7636a736618d27fbb494d3b397db760ba2fba2
peas-580333_640.jpg:2674fc48b1e313579afb5c25c80aadc6:b7c83d30c9db1be792930a0d3082782a4a34ee060c0c5f4cf03533df3beed7f4c096de4d9c559b5130d799e3e04a60894cc9787b87a0942a98936b0426c52519

401 - Find an illegally filmed image: Question 5

FINANCIAL SECURITY INSTITUTE

Developing automation tools

- The tweak value used when decrypting the file contents will use the logical block number
- Logical block number always starts at 0 and changed to a 16-bytes little-endian format when used a tweak value

```
list_of_target = run_fls(keyfile, imgfile, filter='DCIM')
for target in list_of_target:
    filename = target["filename"].split("/")[-1]
    result = run_istat(imgfile, target['inode'])
    result['fn'] = filename

    if result['size'] > largest:
        largest = result['size']

    masterkey = keydata[result['Descriptor']]
    xtskey = derive_key_aes(masterkey, result['Nonce'])
    decrypted_data = b""

    list_of_tweak = []
    for lblk_num in range(len(result['Blocks'])): # logical block number
        tweak = lblk_num.to_bytes(16, byteorder='little')
        list_of_tweak.append(tweak)

        cipher = Cipher(
            algorithms.AES(xtskey),
            modes.XTS(tweak),
            backend=default_backend()
        )
        decryptor = cipher.decryptor()
        encrypted_data = run_blkcat(imgfile, str(result['Blocks'][lblk_num]))
        decrypted_data += decryptor.update(encrypted_data) + decryptor.finalize()

    result['tweaks'] = list_of_tweak
    file_list.update({result['size']: result})

    print('[*] filename : %s' % filename)
    print('[*] size : %s' % result['size'])
    print('[*] sha256 hash : %s' % hashlib.sha256(decrypted_data[:result['size']]).hexdigest())
```

401 - Find an illegally filmed image: Question 5

FINANCIAL SECURITY INSTITUTE

Question 5 Answer

Tweak values

0x00,0x01,0x02,0x03,0x04,0x05,0x06,0x07,0x08,0x09,0x0a, 0x18,

...

0x19,0x1a,0x1b,0x1c,0x1d,0x1e,0x1f,0x20,0x21,0x22

Sha256 hash

33a51636684e932a956d53a686e63bfd2528bf2e87ad55fdede78261e5c68ef3