

302 – Do not blink

Team Information

Team Name: kimbabasaksaksak

Team Member: Jaeheon Kim, Donghyun Kim, Soyoung Yoo, Minhee Lee

Email Address: uaaooong@gmail.com

Instructions

Description Analyze the video and prevent a terrorist attack!

Target	Hash (MD5)
Seoul.mp4	4c4ee9010efd0b056a8143ba1e168dce

Questions

- 1) When is the attack scheduled? (50 points)
- 2) What is the cryptographic key is needed to identify the location of the attack? (125 points)
- 3) Where is the attack scheduled? (125 points)

Teams must:

- Develop and document the step-by-step approach used to solve this problem to allow another examiner to replicate team actions and results.
- Specify all tools used in deriving the conclusion(s).

Tools used:

Name:	ffmpeg	Publisher:	FFmpeg developers
Version:	4.4.2		
URL:	https://ffmpeg.org/		

Name:	DCode	Publisher:	Digital Detective
Version:	5.5		
URL:	https://www.digital-detective.net/dcode/		

Name:	CyberChef	Publisher:	
Version:			
URL:	https://gchq.github.io/CyberChef/		

Name:	StegSolve	Publisher:	Caesum
Version:	1.3		
URL:	http://www.caesum.com/handbook/Stegsolve.jar		

Step-by-step methodology:

Q1. When is the attack scheduled? (50 points)

MP4 컨테이너를 구성하는 5개의 박스 중, movi는 보통 AVI 컨테이너에 존재하는 박스이다.

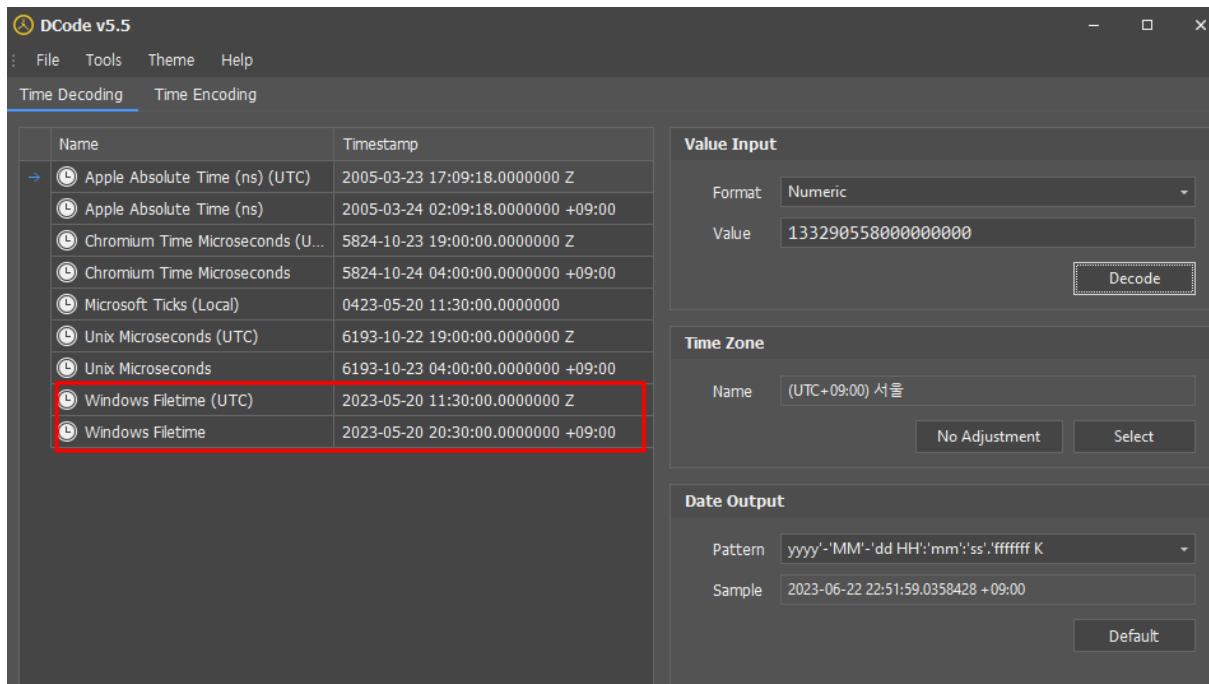
Template Results - MP4.bt				
	Name	Value	Start	Size
> Box[0]		ftyp	0h	18h
> Box[1]		free	18h	C4D85h
> Box[2]		skip	C4D9Dh	2808h
> Box[3]		mdat	C75A5h	D6C7B25Dh
> Box[4]		moov	D6D42802h	C513Fh
> Box[5]		movi	D6E07941h	C220A0h

또한, movi 내의 데이터 값도 비정상적인 것을 보아 의도적으로 숨겨진 데이터라는 것을 알 수 있다. movi 데이터 중 '133290558000000000'는 인코딩 된 시각 값이다.

D6E0:78E0h	74 38 05 78 05 0E 73 3E 47 39 78 08 05 72 09	Pixel><Spheri
D6E0:78F0h	63 61 6C 3A 43 72 6F 70 65 64 41 72 65 61 54	cal:CroppedAreaT
D6E0:7900h	6F 70 50 69 78 65 6C 73 3E 30 3C 2F 47 53 70 68	opPixels>0</GSph
D6E0:7910h	65 72 69 63 61 6C 3A 43 72 6F 70 65 64 41 72	erical:CroppedAr
D6E0:7920h	65 61 54 6F 70 50 69 78 65 6C 73 3E 3C 2F 72 64	eaTopPixels></rd
D6E0:7930h	66 3A 53 70 68 65 72 69 63 61 6C 56 69 64 65 6F	fsSphericalVideo
D6E0:7940h	3E 00 C2 20 A0 6D 6F 76 69 00 00 00 31 33 33 32	>.A movi...1332
D6E0:7950h	39 30 35 35 38 30 30 30 30 30 30 30 00 00 00	90558000000000
D6E0:7960h	00 33 64 35 66 65 61 30 38 31 30 65 39 35 64 38	.305Teau81ue9008
D6E0:7970h	35 32 37 64 37 31 34 38 35 66 63 38 61 65 30 38	527d71485fc8ae08
D6E0:7980h	63 65 62 61 62 32 31 37 39 66 62 63 34 38 30 64	cebab2179fbc480d
D6E0:7990h	35 37 34 31 64 31 61 39 61 62 62 63 66 37 32 30	5741d1a9abbcf720
D6E0:79A0h	63 38 31 39 37 31 30 62 66 66 63 65 34 65 37 32	c819710bffe4e72
D6E0:79B0h	66 62 37 39 61 36 64 34 38 35 37 35 65 62 33 30	fb79a6d48575eb30
D6E0:79C0h	34 37 66 31 33 66 62 61 33 31 64 64 34 36 64 33	47f13fba31dd46d3
D6E0:79D0h	32 31 62 62 34 62 66 37 33 35 61 61 63 33 38 35	21bb4bf735aac385
D6E0:79E0h	63 38 62 62 38 31 38 62 32 63 36 36 66 37 66 65	c8bb818b2c66f7fe
D6E0:79F0h	34 38 38 66 37 64 33 30 62 31 63 62 66 66 37 34	488f7d30b1cbff74
D6E0:7A00h	65 65 38 38 62 62 66 66 38 63 37 37 30 33 31 39	ee88bbff8c770319
D6E0:7A10h	34 34 34 33 38 37 39 61 66 61 34 35 62 62 31 30	4443879afa45bb10
D6E0:7A20h	30 37 30 31 62 36 34 33 35 35 34 36 38 61 36 61	0701b64355468a6a
D6E0:7A30h	62 38 64 32 37 66 62 61 38 31 37 38 30 64 66 63	b8d27fba81780dfc
D6E0:7A40h	32 62 66 32 39 32 39 36 61 33 66 64 34 31 66 33	2bf29296a3fd41f3
D6E0:7A50h	37 66 39 66 36 66 36 34 37 32 35 32 30 66 62 32	7f9f6f6472520fb2
D6E0:7A60h	63 33 37 38 33 33 33 37 61 66 36 30 38 63 37 32	c3783337af608c72
D6E0:7A70h	66 32 63 36 39 61 38 65 64 31 34 31 35 30 32 62	f2c69a8ed141502b

Template Results - MP4.bt						
	Name	Value	Start	Size	Color	Comm
> Box[0]		ftyp	0h	18h	Fg: Bg:	File Type Box
> Box[1]		free	18h	C4D85h	Fg: Bg:	Free Space Box
> Box[2]		skip	C4D9Dh	2808h	Fg: Bg:	Unknown box type
> Box[3]		mdat	C75A5h	D6C7B25Dh	Fg: Bg:	Media Data Box
> Box[4]		moov	D6D42802h	C513Fh	Fg: Bg:	Movie Box
✓ Box[5]		movi	D6E07941h	C220A0h	Fg: Bg:	Unknown box type
✓ struct boxheader hdr		movi [size=12722328]	D6E07941h	8h	Fg: Bg:	
uint32 size		12722336	D6E07941h	4h	Fg: Bg:	
✓ struct fourcc type		movi	D6E07945h	4h	Fg: Bg:	
byte value[4]		movi	D6E07945h	4h	Fg: Bg:	

DCode 도구를 사용하여 시각 값을 디코딩하면, 예정된 공격 시각은 '2023-05-20 20:30:00 UTC+9'라는 것을 알 수 있다.



Q2. What is the cryptographic key is needed to identify the location of the attack? (125 points)

ffprobe로 Seoul.mp4의 상세 정보를 확인하여 동영상의 프로젝션 타입이 equirectangular임을 알 수 있다.

```

kva8S0OH-NAGQ9BLARU::fsw.$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 --enable-gpl --disable-stripping --enable-gnutls --enable-ladspa --enable-libaom --enable-libass --enable-libbluray --enable-libsbs2b --enable-libcaca --enable-libcdio --enable-libdav1d --enable-libtinfo --enable-libfontconfig --enable-libfreetype --enable-libgsm --enable-libgssm --enable-libgstreamer --enable-liblame --enable-libopenjpeg --enable-libopus --enable-libpangocairo --enable-libpango --enable-libpostproc --enable-libpulse --enable-librav1e --enable-librsvg --enable-librubberband --enable-libshaderc --enable-libspeex --enable-libssh --enable-libsvt-av1 --enable-libtheora --enable-libtwolame --enable-libvidstab --enable-libvorbis --enable-libvpx --enable-libx264 --enable-libx265 --enable-libxml2 --enable-libyuv --enable-libz --enable-libzimg --enable-libzvbi --enable-lv2 --enable-openssl --enable-opengl --enable-png --enable-sdl2 --enable-pocketsphinx --enable-libsrt --enable-libsvt --enable-libtesseract --enable-libwebp --enable-libxcb --enable-libxkbcommon --enable-libxvid --enable-libzmq --enable-libzstd --enable-mbedtls --enable-nasm --enable-nvenc --enable-opencl --enable-openmp --enable-openssl --enable-postproc --enable-pthreads --enable-riscv --enable-runtime-cpudetect --enable-sdl2 --enable-speex --enable-tesseract --enable-theora --enable-thrust --enable-vulkan --enable-waf --enable-x11 --enable-xcb --enable-xlib --enable-zimg --enable-zlib-ng --enable-zstd
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
libbswscale   5.  9.100 / 5.  9.100
libswresample  3.  9.100 / 3.  9.100
libpostproc   55. 10.100 / 55. 10.100
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:(eng): Video: h264 (Main) (avc1 / 0x31367F61), 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     : 7Mainconcept Video Media Handler
vendor_id        : [0][0][0]
encoder          : AVC Coding
Side data:
stereo3d: 2D
spherical: equirectangular (0.000000/0.000000/0.000000)

```

Equirectangular의 경우, 구를 직사각형 이미지로 표현하는 VR 영상 프로젝션 타입이다. 동영상의 프레임을 추출하면 아래와 같이 모든 방향이 왜곡된 형태로 나타난다.



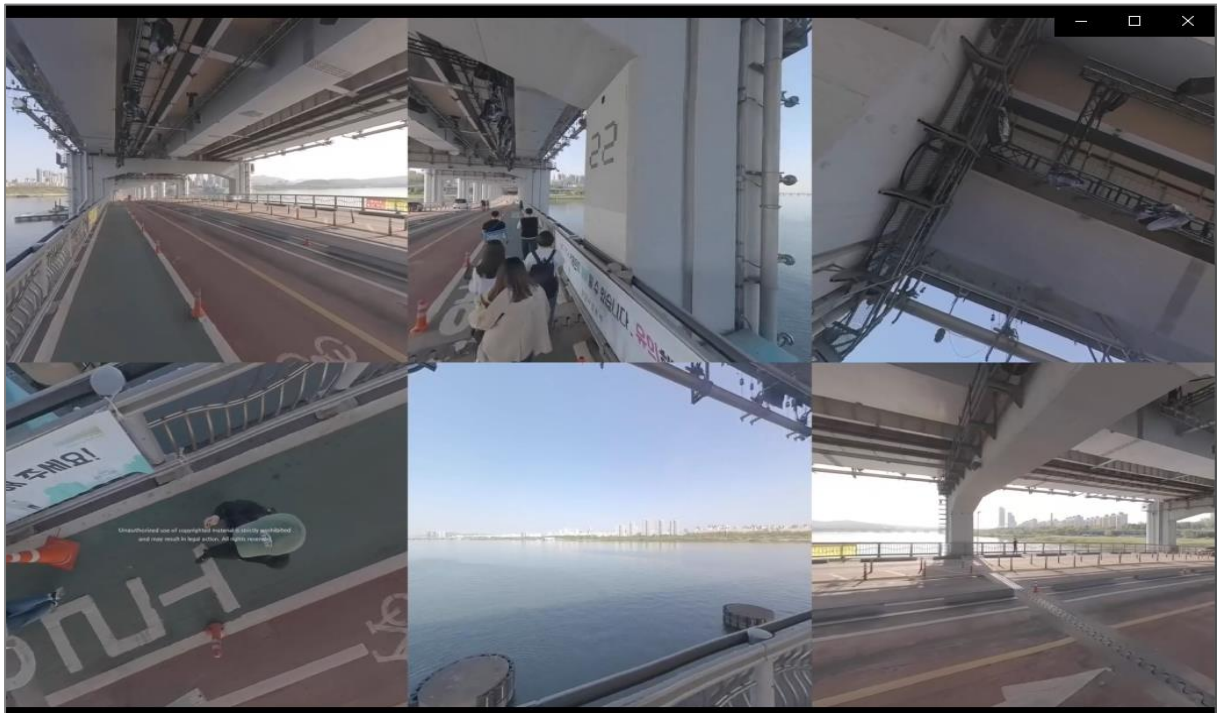
왜곡 없는 프레임을 얻기 위해 ffmpeg를 이용해 동영상의 프로젝션 타입을 cubemap으로 변환한다. cubemap의 경우, 360도 환경을 정육면체로 보고 각 방향을 6개의 정사각형으로 표현해 왜곡 없이 모든 면을 얻을 수 있다.

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

이후 6개의 면을 별도의 동영상으로 분할한다.

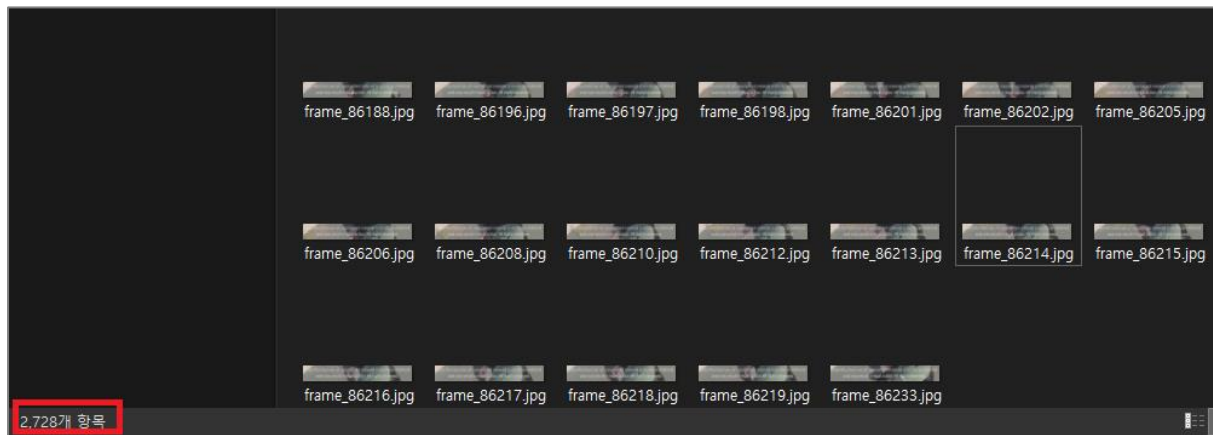
```
ffmpeg -i output.mp4 -filter_complex "[0:v]split=6[c0][c1][c2][c3][c4][c5];  
[c0]crop=iw/3:ih/2:0:0[c0];  
[c1]crop=iw/3:ih/2:iw/3:0[c1];  
[c2]crop=iw/3:ih/2:2*iw/3:0[c2];  
[c3]crop=iw/3:ih/2:0:ih/2[c3];  
[c4]crop=iw/3:ih/2:iw/3:ih/2[c4];  
[c5]crop=iw/3:ih/2:2*iw/3:ih/2[c5]  
" -map "[c0]" c0.mp4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

분할된 영상을 확인하면 다음과 같이 모든 각도를 한 번에 확인할 수 있다.



해당 영상에서 암호화 키를 나타낼 수 있는 부분 중 하나인 자막에 초점을 두고 자막 영역의 RGB 값을 비교하여 자막 값이 달라지는 경우 해당 프레임을 저장하는 *코드를 만들어 실행한다.

*첨부파일 내 suspect-words.py 파일 참조

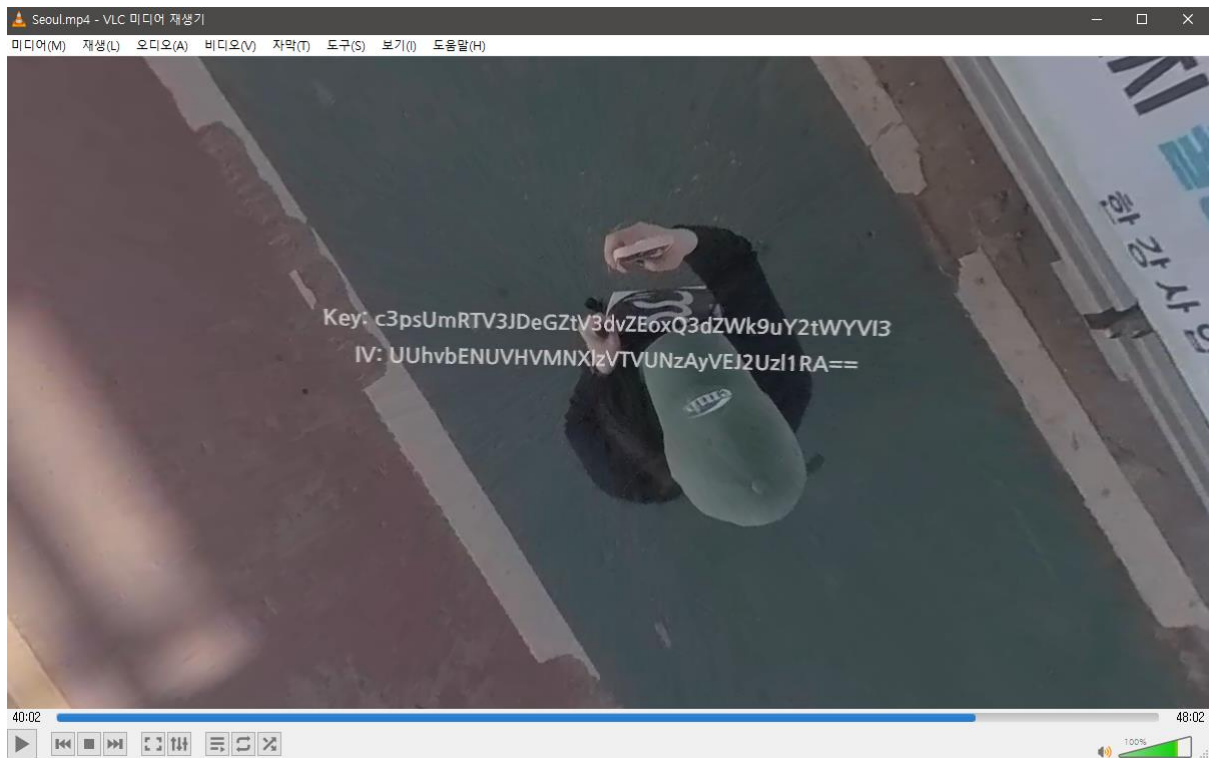


영상 전체에서 특정 RGB 값을 벗어나는 프레임들은 2,728개이며 해당 자막 부분들을 Tesseract OCR을 사용하여 CSV에 기록하는 *코드를 만들어 실행한다.

*첨부파일 내 ocr.py 파일 참조

061	2079	frame_70233.jpg	thorized use of copyrighted material is strictly prohibited and may result in legal action. All rights reserved.	
062	2080	frame_70234.jpg	: Use of copyrighted material is strictly prohibited and may result in legal action. All rights reserved.	
063	2081	frame_70236.jpg	aAd #Nah use of copyrighted material is strictly prohibited and may result in legal action. All rights reserved.	
065	2083	frame_70267.jpg	Unauthoranduse of copyrighted material is strictly prohibited and may result in legal action. All rights reserved.	
066	2084	frame_70268.jpg	질4Unauthorized use of copyrighted material is strictly prohibited and may result in legal action. (All rights reserved)	
067	2085	frame_70269.jpg	Unauthorandise of copyrighted rial is strictly prohibited and may result in legal action. All rights reserved.	
088	2106	frame_7120.jpg		
090	2108	frame_71478.jpg		
091	2109	frame_71531.jpg	Unauthorized use of copyrighted material is strictly prohibited and may result in legal action. All rights reserved.	
096	2114	frame_718.jpg	aByriohthedin legal actioy > ,	
111	2129	frame_71955.jpg	Unauthorized use of copyrighted material is strictly prohibited and may result in legal action. All rights reserved.	
115	2133	frame_72000.jpg	y 4K 1sUmRTV3lDeGZtyStvZE6xO3dZWk9uY2tWYVl3.IV: UUhbvENUVHVVMWN: INzAyVEl2Uz1 RA=:	
119	2137	frame_721.jpg		
129	2147	frame_72615.jpg	rized use of copyrighted material is strictly prohibited and may result in legal action. All rights reserved.	
140	2158	frame_75903.jpg	d use bat ro ic ited legal action. All rights reserved.	
142	2160	frame_75949.jpg	Usrized use righted mi?ay rest f actiTh	
143	2161	frame_75958.jpg	li.. is strictly legal action. All rights reserved.	

72,000번 프레임(영상시간 40:02)에서 기존 자막 포맷을 벗어난 문자열이 존재했으며 해당 프레임 확인 결과 Key, Iv를 확인할 수 있다.

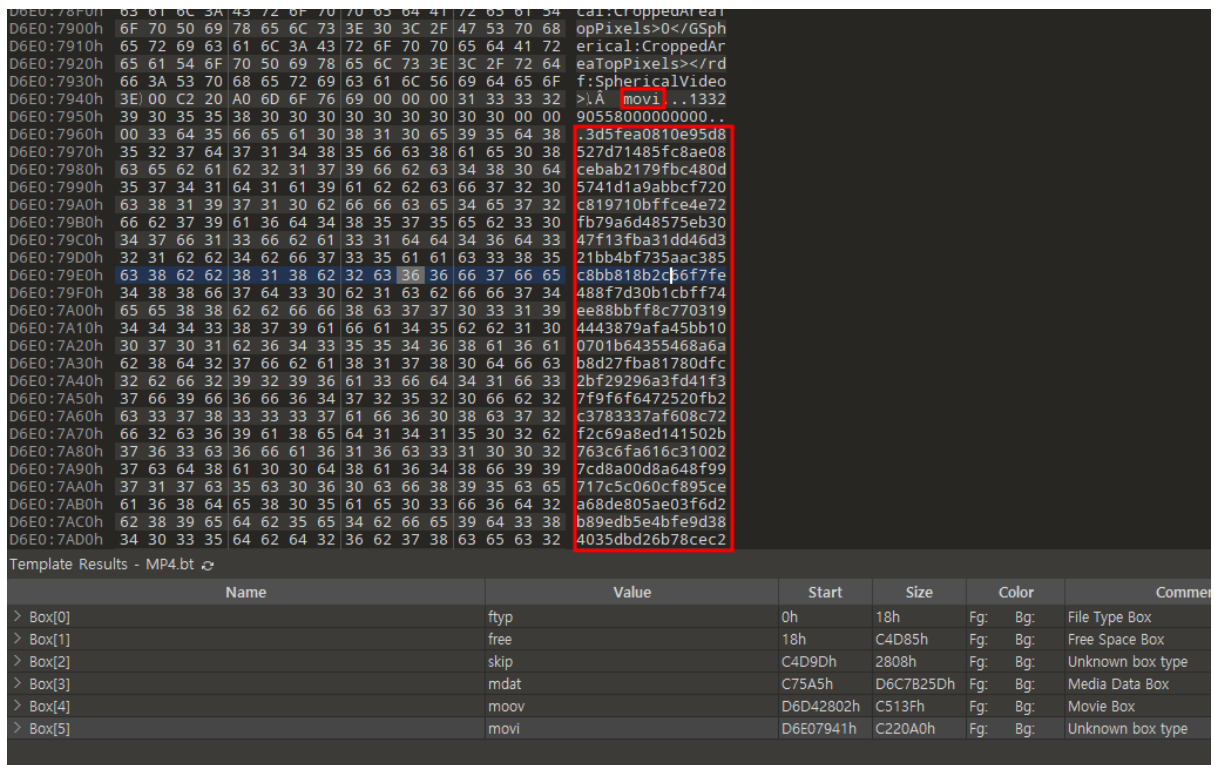


Key : c3psUmRTV3JDeGZtV3dvZEoxQ3dZWk9uY2tWYVI3

IV : UUhbENUVHVMNXIzVTVUNzAyVEJ2UzI1RA==

Q3. Where is the attack scheduled? (125 points)

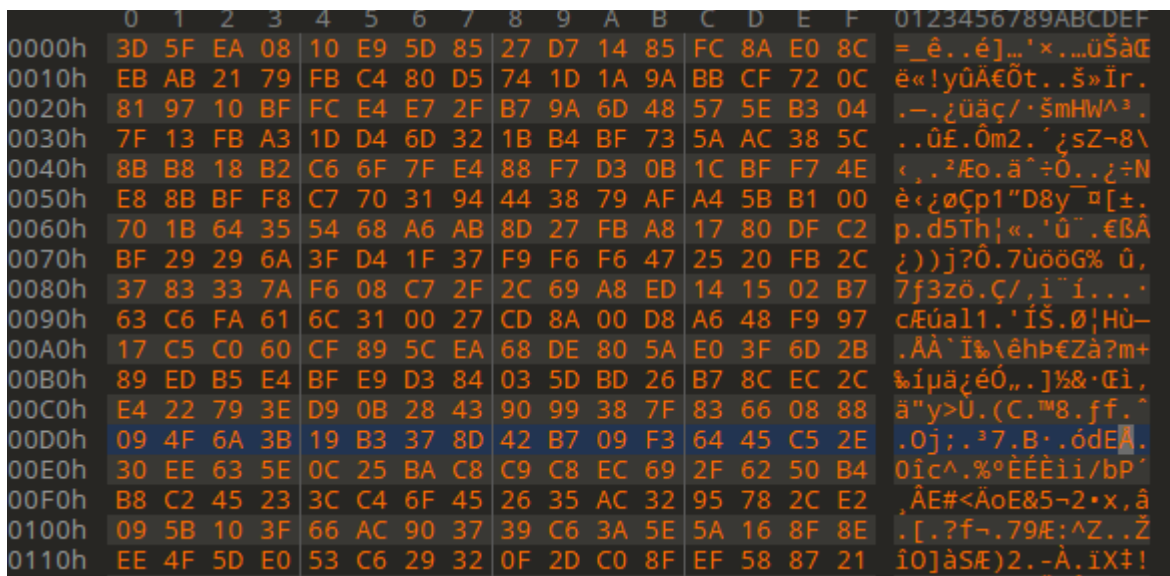
movi 박스에는 시각 값 이후에 약 12MB 정도의 HEX 문자열이 저장되어 있다.



Template Results - MP4.bt

Name	Value	Start	Size	Color	Comment
> Box[0]	ftyp	0h	18h	Fg: Bg:	File Type Box
> Box[1]	free	18h	C4D85h	Fg: Bg:	Free Space Box
> Box[2]	skip	C4D9Dh	2808h	Fg: Bg:	Unknown box type
> Box[3]	mdat	C75A5h	D6C7825Dh	Fg: Bg:	Media Data Box
> Box[4]	moov	D6D42802h	C513Fh	Fg: Bg:	Movie Box
> Box[5]	movi	D6E07941h	C220A0h	Fg: Bg:	Unknown box type

해당 문자열을 HEX 데이터로 변환하여 확인하면, 엔트로피가 높은 것을 보아 암호화된 데이터라고 추측할 수 있다.



파일 복호화에 사용되는 Key와 IV는 Base64로 디코딩 가능한 문자열이며, Base64로 디코딩한 결과는 다음과 같다.

- Key : szlRdSWrCxfmWwodJ1CwYZOnckVaR7

- IV : QHolCTTuL5ysU5T702TBvS9uD

Cyberchef로 Key와 IV를 사용하여 다양한 방법으로 복호화를 시도해보던 중, 다음과 같은 옵션을 적용했을 때 PNG 이미지 파일로 복호화가 되었다.

The screenshot shows the CyberChef web interface. A recipe titled 'DES Decrypt' is active. The 'Key' field contains the Base64 string 'szlRdSWrCxfmWwodJ1CwYZOnckVaR7' and the 'IV' field contains 'QHolCTTuL5ysU5T702TBvS9uD'. Both fields are set to 'HEX' encoding. The 'Mode' is 'CBC/NoPadding', 'Input' is 'Raw', and 'Output' is 'Raw'. The 'Input' pane shows a long Base64-encoded string. The 'Output' pane shows the decoded result, which is a PNG image. A red box highlights the 'PNG' icon in the output pane.

복호화된 이미지 파일은 다음과 같다.



이미지 파일에는 스테가노그래피 기법이 적용되어 있어, StegSolve를 사용하여 숨겨진 문자열을 복구했다.



'///lure.expose.sleeping'는 3개의 단어를 이용해 고유 코드를 부여한 지리 코드인 what3words이다. what3words에서 코드에 해당하는 위치를 확인하면 국립고궁박물관인 것을 알 수 있다. 즉, 공격이 예정된 장소는 '국립고궁박물관 입구'이다.

