



악성 한글 문서

분석 가이드라인 제시

Hy4팀

2020년 08월 19일

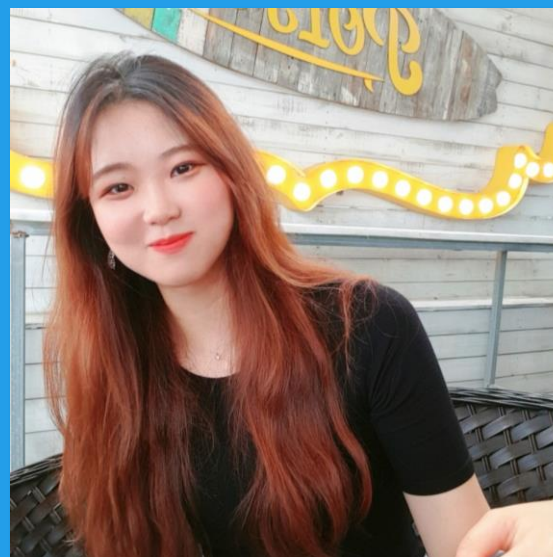
팀장

송태현



팀원

김현진



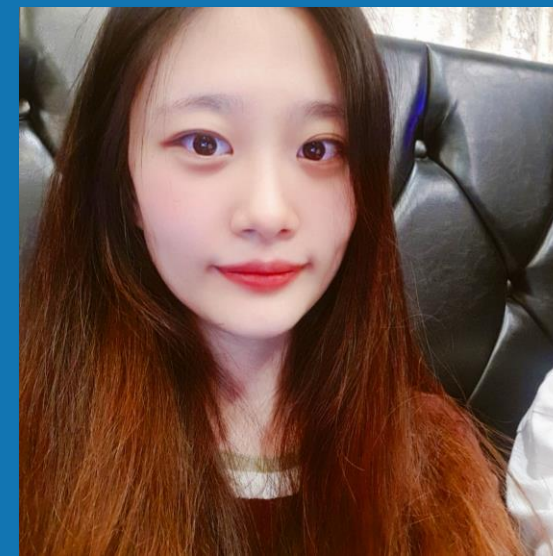
팀원

이다영



팀원

허송이



목차

- 001 프로젝트 개요
- 002 프로젝트 진행
- 003 목표 산출물 및 기대효과



Part 1,

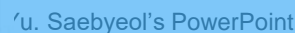
프로젝트 개요

1. 프로젝트 배경
2. 프로젝트 소개
3. 프로젝트 일정

Q

시간	장소	확인자 설명	확인자 소속 · 직위	확인자 이메일
-				
-				
-				
-				
-				

로 받아 보신 적이 있나요?



Q

시간	장소	확인자 설명	확인자 소속·직위	확인자 이메일
-				
-				
-				
-				
-				

정하에 열어 보실 건가요?



1. 프로젝트 배경 – 지속적으로 유포되는 악성 한글 문서

가상자산 거래소 문서 위장 악성코드 발견

김 안랩 "악성 코드 담긴 '드론' 관련 한글 파일 열지마세요"

인 [긴급] 실존기업 사칭 '발주서' 악성메일 줄줄이 유포

좋아요 25개 | 입력: 2020-05-15 18:12



#발주서

#악성메일

실존하는 기업과 직원 사칭...자연스러운 한글 사용으로 속기 쉬워

[보안뉴스 원병철 기자] 14일과 15일 실존하는 기업의 이름을 사칭한 '발주서' 형태의 악성메일이 연이어 발견돼 기업 담당자들의 주의가 요구된다. 특히, 이번 악성메일은 기업에 실제 존재하는 직원의 이름까지 그대로 사용해 자칫 잘못하면 속아 넘어갈 수 있다.

CHOI <slim@bengiamein.tk> undisclosed-recipients:

T 05-14 (CHOI) 발주서 첨부하여 드립니다

중요도가 높음인 메시지를 보냈습니다.

T 05-14 (CHOI)...
1 MB

안녕하세요, T 최입니다.

흥미로운 제목으로
악성코드 감염을 유도시키는
악성 한글 문서가 지속적으로 유포 !!

1. 프로젝트 배경 – 새로운 가이드라인의 필요성

모든 유형의 보고서를 통합한 문서 無

&

분석 시 봐야 할 항목들을 제시해주는 문서 無

단일 샘플

말간 Security Report

No.03 | 2015년 7월

2. 동작 분석

2.1. HWP 문서 (4.2심포기원안.hwp, 소위자료(2014.2.25)_수석전문위원소관.hwp)

두 HWP 작성코드 샘플은 같은 취약점을 이용하여, 본문 내용은 다릅니다.
해당 취약점은 HWP 문서에 있는 과정에서 부적절한 참조로 인해 취약점이며, 최종적으로 HWP_TAG_PARA_LINE_SEG 레코드를 통해 레코드를 구조체로 작성하고 사용하여 이를 이용해 참조하는 객체 주소를 참조하고 fake vtable로 실행 흐름을 바꿀 수 있게 됩니다.

정자는 취약점에 대한 과정이 이미 존재하며, 올바른 참조와 동적 하일 체크로 해결되었습니다.
해당 취약점에서 참조 객체 주소를 변경할 수 있는 부분은 아래와 같습니다.

```
00000000 42 00 60 01 18 00 00 04 00 00 00 00 00 00 00 00
00000010 01 00 00 00 03 00 00 00 00 00 43 04 00 03 02 00
00000020 44 43 68 73 00 00 00 00 00 00 00 00 02 00 02 00
00000030 44 4C 4F 43 00 00 00 00 00 00 00 00 02 00 20 00
00000040 20 00 20 00 20 00 20 00 20 00 20 00 00 00 44 04
00000050 80 00 00 00 00 00 06 00 00 00 45 04 CD 04 00 00
00000060 00 00 16 00 00 00 52 C3 48 73 4F 4F 00 00 41 04
00000070 16 00 0C 03 00 00 00 00 00 18 A6 00 00 00 00 00
00000080 06 00 18 00 00 20 00 00 00 20 00 00 00 18 A6
00000090 00 00 20 00 00 20 00 00 00 00 00 00 18 A6
000000A0 00 00 00 06 00 1C 00 00 20 00 00 00 20 00
000000B0 00 00 20 00 00 20 00 00 20 00 00 00 00 00 00
```

파란색 부분이 객체 주소가 되고, 해당 주소+offset 으로 vtable를 참조하고 함수를 호출합니다.
원래의 HWP_TAG_PARA_LINE_SEG는 새로운 메모리 주소와 같은 구조로 작성해야 하는 구조체이지만 다른 객체로 생각하여 위와 같은 위치에서 또 다른 객체의 주소로 읽게 됩니다.

Heap spray 기법으로 nop like + shellcode로 된 객체 5개를 HWP 파일에 내장하고, 한글에서 파일로 읽으면 메모리 상에 채워져 0x00C000C0에 nop sled가 위치하게 됩니다. 따라서 실행 흐름이 0x00C000C0로 바뀌면 nop like(or al 0d) 코드가 실행된 후 마지막에 메모리가 실행됩니다.

이 작성코드는 DEP(Data Execution Prevention)을 우회하기 위한 ROP 등의 기법이 전혀 쓰이지 않았기 때문에 XP/Vista/7에서 모든 어플리케이션에 대해 DEP만 활성화되어 있다면 작성코드가 실행되는 것을 효과적으로 막을 수 있습니다.

4

SOMANSA

단일 유형

2019. 07

Analysis Report

한글 파일에 숨어든 '고스트'

고스트스크립트 취약점 CVE-2017-8291을 이용한 악성 한글 파일

안랩 시큐리티대응센터(ASEC) 분석팀

AhnLab

안랩(주) | 본사: 서울특별시 강남구 테헤란로 208 (B) 1308 | 대표전화: 02-722-8000 | 팩스: 02-722-8001 | www.ahnlab.com
© AhnLab, Inc. All rights reserved.

2. 프로젝트 소개 – 주제

“

악성 한글 문서 대상

”

유형별 진단 가이드라인 제시

2. 프로젝트 소개 – 주제



2. 프로젝트 소개 - 목표

첫째

팀원 개개인의 악성코드 분석 기술 향상

둘째

진단 기준표를 통해 분석가들의 분석 시간 단축

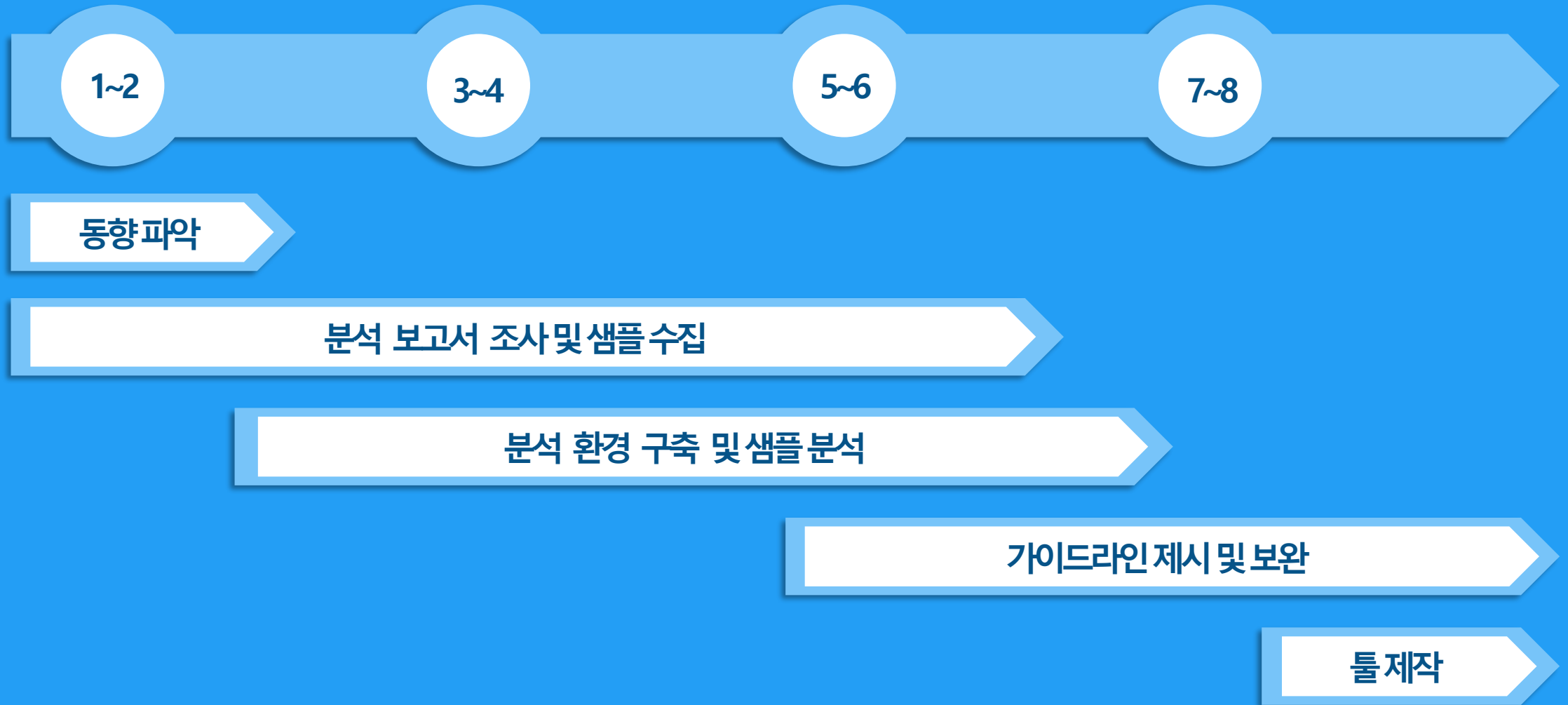
셋째

악성문서 공격유형을 판단하는 툴을 제작하여 신뢰도 향상

2. 프로젝트 일정 - 계획



2. 프로젝트 일정 -WBS



Part 2,

프로젝트 진행

1. 자료조사 및 샘플 조사
2. 분석
3. 가이드라인 작성

1. 자료조사 및 샘플조사 - 동향 파악

Contents

달라진 2020년 한글 파일 공격, 가장 주목할 점은?

1. 2020년에 유포된 악성 한글 파일 동향 03
2. 고스트스크립트 취약점 및 악성 EPS 파일의 변화 분석 06
3. 최근 변경된 포스트스크립트 코드 분석 08
4. 결론 10

ASEC Report Vol.99 2020 Q2

ASEC(Ahnlab Security Emergency response Center, 안랩 시큐리티(대응센터)은 악성코드 및 보안 위협으로부터 고객들을 안전하게 지키기 위하여 보안 전문가로 구성된 글로벌 보안 조직입니다. 아 러포트는 주식회사 안랩의 ASEC에서 작성하며, 주요 보안 위협과 이슈에 대응하는 최신 보안 기술에 대한 요약 정보를 담고 있습니다. 더 많은 정보는 안랩닷컴(www.ahnlab.com)에서 확인하실 수 있습니다.

한글 취약점을 통해 유포되는 악성코드 동향 및 분석

- 여러 문서에 동일한 악성코드를 포함시키며 주로 이메일을 통해 정보 유출 -

□ 개 요

최근 아래 한글(.hwp) 취약점을 통해 악성코드를 유포하고 키보드 입력, 시스템 정보 등을 유출하는 사례가 빈번하게 발견되었습니다. 취약한 한글 파일의 제목은 주요 인사 리스트, 국제 행사 자료 등이며 사용자가 클릭하도록 최근 이슈화된 사건을 다룬다.

< 악성코드를 내포하고 있는 아래 한글 제목 >

구분	발견된 한글 파일 제목
사회적 이슈 관련	수원 토막살해 오원준, 감옥서 의외의 행동, 삼성 이어 LG전자도 정기 세무조사 등
주요 인사 관련	19대 노동계 및 범노동계출신 당선자 프로필(최종), 국방 안보추진단 대사 추천 등
국제 행사 관련	2013년 대구 세계에너지 총회 발표내용 요약, 제57차 IAEA 총회자료 등
대북 이슈 관련	탈북인 인적사항, 북한의 회담전략과 의도 등

**고스트스크립트 취약점을
이용한 악성 한글 파일이
약 2년동안 지속적으로
제작 및 유포**

1. 자료조사 및 샘플조사 - 분석 보고서 및 샘플 조사

안랩, 이글루시큐리티 등
신뢰성 있는 기관의
분석 보고서를 바탕으로
유형 분류 및 유형별 특징 조사

AhnLab_ASEC_한글파일숨어든'고스트'.pdf
estsecurity_report_2004.pdf
estsecurity_report_2006.pdf
아래아한글 Zero-Day 취약점 분석 (이동우).pdf
journal_jkiisc_29-6_2131731120.pdf
Malicious-File-Investigation-Procedures.pdf
MalwareGuideLine_c0nstant.pdf
mcir-guide.pdf
securityse_tistory_com_PMA_lab1-1.pdf
SeQR_중간발표_조재현.pdf
소프트웨어+보안악점+진단가이드.pdf
참고1. 한글 취약점을 통해 유포되는 악성코드 동향 및 분석.pdf
참고2. 한글 文書파일의 靜的分析을 통한 惡性코드 探知에 관한 研究.pdf

포상 추천 후보자 주요 공적
일본관련일일동향
이력서
북한의 회색지대 전략과 대응방안.hwp_
중고폰매각 공고문.hwp_
주간 국제 안보군사 정세.hwp_
저작권 침해 내용증명.hwp_
일일동향보고.hwp_
이때때 단변서 치조분.hwp_

3168&direct=mobile
ews/secuNewsView.do?menu_dist=2&curPage=1

ntSeq=282&page=17
dxno=3146

2209

total : 22]

p_
과 한증관계)....
램.hwp_

현재).hwp*
.hwpxx*

08354-Kimsuky-%EC%A1%B0%EC%A7%81-
B8%80-%EB%AC%B8%EC%84%9C-

d8e13fe5ccc...
4895472d96...
58d6c63140...

_____.hwp

2. 분석 – 분석 환경



VMware



HWP 5.0.3.0.1



Tool

2. 분석 - 분석 방식

A

보고서를 따라 분석

B

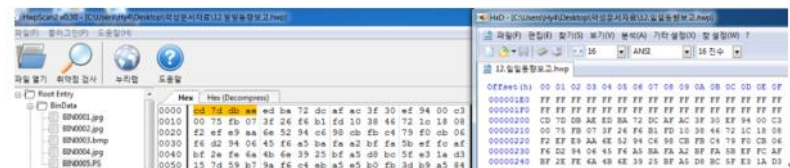
보고서가 없는 샘플 분석

2. 분석

OLE 파일 구조 공부



12. 일일동향 [XOR]



파일 이름 : BIN0005_PS [DocInfo/offset = 0x1600]

Hxd Offset : 0x200

Root entry : 0x02 (시작 offset : 0x600)

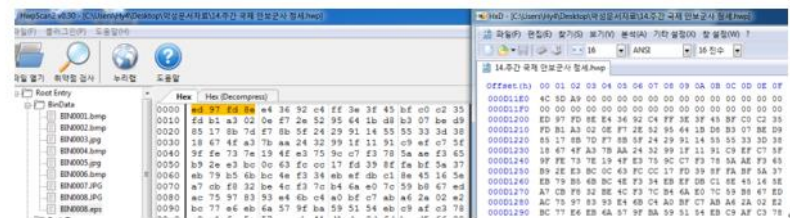
↵

↵

↵

↵

14. 안보군사 [XOR]



파일 이름 : BIN0008_eps

Hxd Offset : 0xD1200

Root entry : 0x01 (시작 offset : 0x400)

2. 분석

```
# -*- coding: utf-8 -*-
import os
import sys
from time import sleep
from io import BytesIO

import zlib # zlib 디컴프레스 위한
import olefile # OLE 파서 이용

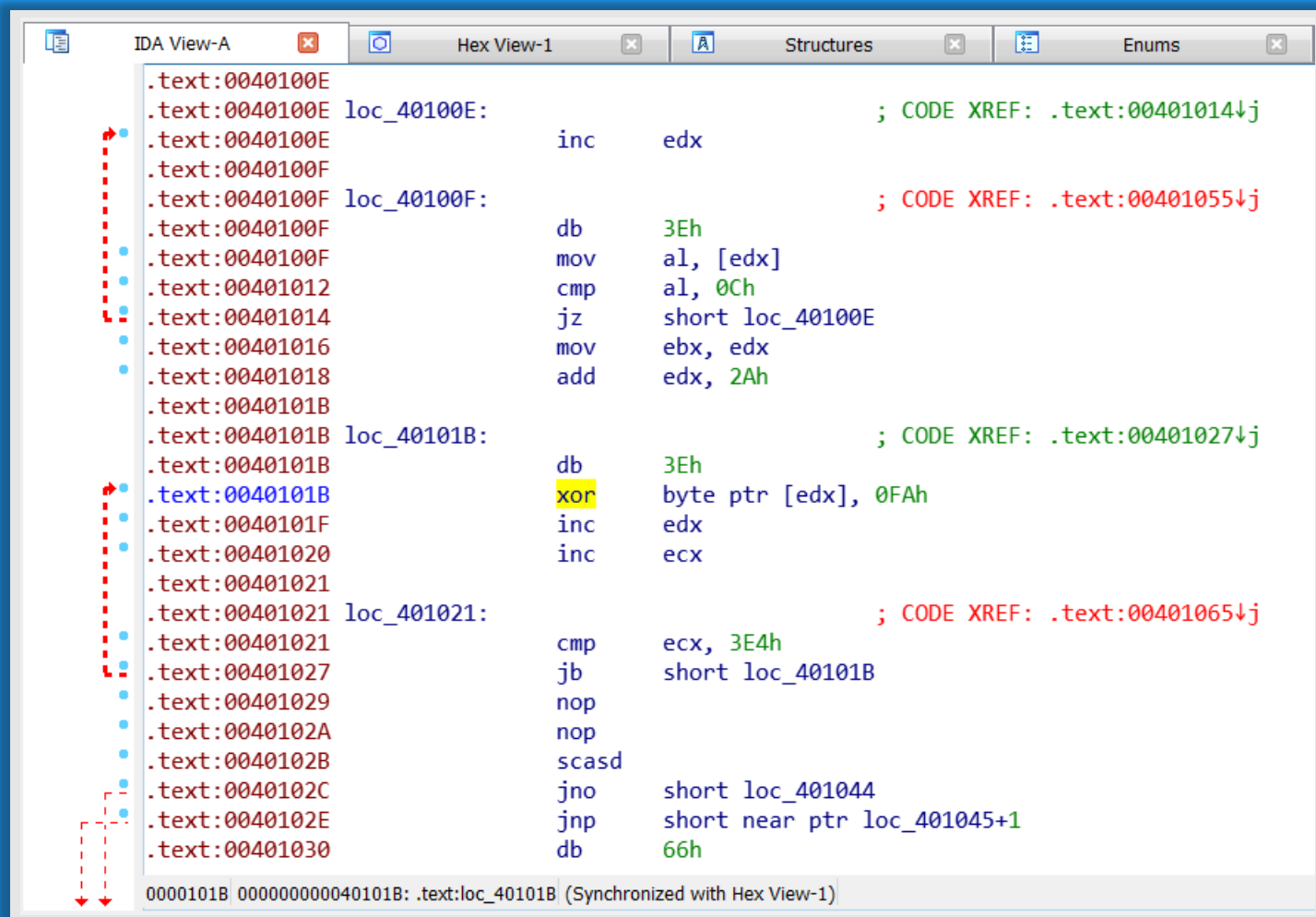
## main

filename = r'C:\Users\windows7\Desktop\real1\1. 한반도통일대토론회.hwpxx'
ole = olefile.OleFileIO(filename)
stream = ole.openstream('BodyText/Section0')
stream = BytesIO(zlib.decompress(stream.read(), -15))
rest = stream.getvalue().hex()
#print(rest)
cnt = 0
for i in range(int(len(rest)/2)):
    cnt += 1
    print(rest[i*2:i*2+2], end = " ")
    if cnt % 16 == 0 :
        print()
print(rest)
```

```
2d 00 36 00 38 00 38 00 39 00 20 00 03 00 6b 6c
68 25 00 00 00 00 00 00 00 00 03 00 73 00 6a 00
68 00 6f 00 6e 00 67 00 40 00 68 00 72 00 69 00
2e 00 63 00 6f 00 2e 00 6b 00 72 00 04 00 6b 6c
68 00 00 00 00 00 00 00 00 00 04 00 0d 00 44 04
80 01 00 00 00 00 05 00 00 00 36 00 00 00 18 00
00 00 46 00 00 00 05 00 00 00 45 04 40 02 00 00
00 00 64 f0 00 00 14 05 00 00 14 05 00 00 51
00 00 0c 03 00 00 00 00 00 00 a0 ab 00 00 00
06 00 47 04 d0 04 6b 6c 68 25 00 88 00 00 00
00 6d 00 61 00 69 00 6c 00 74 00 6f 00 3a 00
00 6a 00 68 00 6f 00 2e 00 6b 00 72 00 3b 00
00 69 00 2e 00 63 00 6f 00 2e 00 6b 00 72 00 3b 00
00 32 00 3b 00 30 00 3b 00 30 00 d6 37 b9 7f 00 00
00 00 00 42 00 60 01 00 0e 00 03 00 00 00 00
09 09 00 a0 0f 00 00 00 74 c7 74 d5 15 c8
c7 f0 c5 6c ad d0 c6
00 2d 00 36 00 32 00
00 20 00 30 00 31 00
00 38 00 2d 00 31 00
00 6b 6c 68 25 00 00
00 6a 00 6c 00 65 00
00 2e 00 63 00 6f 00
6c 68 00 00 00 00 00 00 00 00 04 00 0d 00 44
04 80 01 00 00 00 00 05 00 00 00 37 00 00 00 18
00 00 00 46 00 00 00 05 00 00 00 45 04 40 02 00
00 00 00 84 f8 00 00 14 05 00 00 14 05 00 00 51
04 00 00 0c 03 00 00 00 00 00 00 a0 ab 00 00 00
00 06 00 47 04 b0 04 6b 6c 68 25 00 88 00 00 00
1c 00 6d 00 61 00 69 00 6c 00 74 00 6f 00 3a 00
68 00 6a 00 6c 00 65 00 65 00 40 00 68 00 72 00
69 00 2e 00 63 00 6f 00 2e 00 6b 00 72 00 3b 00
32 00 3b 00 30 00 3b 00 30 00 d6 37 b9 7f 00 00
00 00 Squeezed text (248 lines).
```

**zlib 압축을
해제시키는 코드**

2. 분석



```

.text:0040100E
.text:0040100E loc_40100E:                                ; CODE XREF: .text:00401014↓j
.text:0040100E         inc     edx
.text:0040100F
.text:0040100F loc_40100F:                                ; CODE XREF: .text:00401055↓j
.text:0040100F         db      3Eh
.text:0040100F         mov     al, [edx]
.text:00401012         cmp     al, 0Ch
.text:00401014         jz      short loc_40100E
.text:00401016         mov     ebx, edx
.text:00401018         add     edx, 2Ah
.text:0040101B loc_40101B:                                ; CODE XREF: .text:00401027↓j
.text:0040101B         db      3Eh
.text:0040101B         xor     byte ptr [edx], 0FAh
.text:0040101F         inc     edx
.text:00401020         inc     ecx
.text:00401021 loc_401021:                                ; CODE XREF: .text:00401065↓j
.text:00401021         cmp     ecx, 3E4h
.text:00401027         jb      short loc_40101B
.text:00401029         nop
.text:0040102A         nop
.text:0040102B         scasd
.text:0040102C         jno     short loc_401044
.text:0040102E         jnp     short near ptr loc_401045+1
.text:00401030         db      66h

```

0000101B 000000000040101B: .text:loc_40101B (Synchronized with Hex View-1)

셸코드를
찾은 후 분석 시도

2. 분석 - 공격 유형

**매크로를
이용한 악성코드**

**파일을 생성하는
포스트스크립트**

**셸코드를 포함한
포스트스크립트**

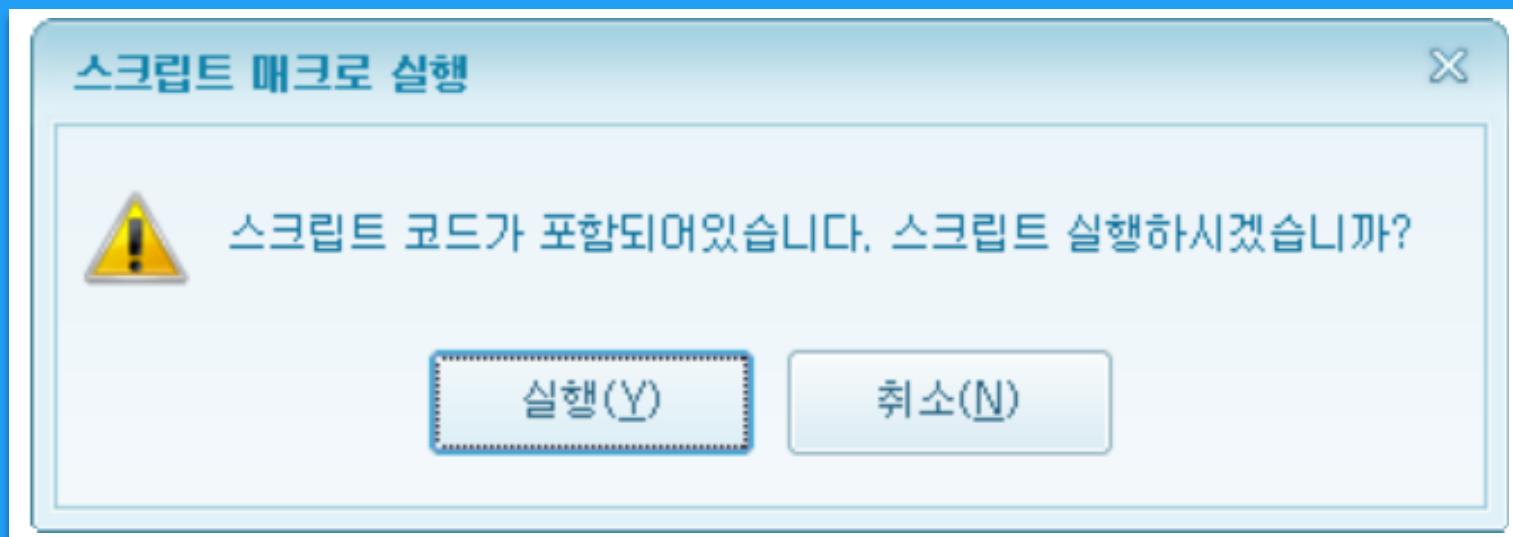
2. 분석 – 매크로를 이용한 악성코드

매크로를 이용한 악성코드

한글 프로그램의 매크로 코드에 악성행위를 할 수 있는

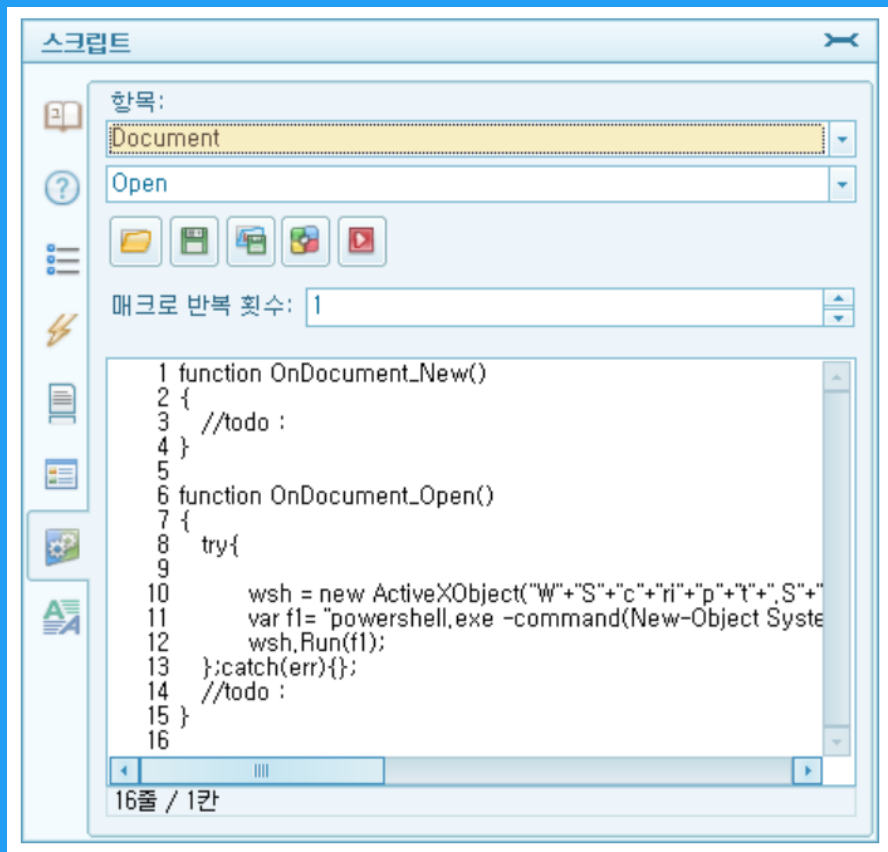
스크립트를 포함하여 문서가 실행되자마자 악성행위를 수행하도록 한다.

2. 분석 - 매크로를 이용한 악성코드



1. 매크로 실행 여부를 확인한다.

2. 분석 - 매크로를 이용한 악성코드



2 powershell을 실행하는 javascript 코드가 포함되어 있다.

2. 분석 – 포스트스크립트(파일 생성)

파일을 생성하는 포스트스크립트

포스트스크립트 언어를 이용하여

파일(악성코드)을 드롭 및 실행한다.

2. 분석 – 포스트스크립트(파일 생성)

<2017년 금융보안 표준화 수요조사 안내>

```
000000F0  72 69 6E 67 73 20 70 31 20 28 5C 5C 2E 2E 5C 5C  rings p1 (\\..\\
00000100  2E 2E 5C 5C 52 6F 61 6D 69 6E 67 5C 5C 4D 69 63  ..\\Roaming\\Mic
00000110  72 6F 73 6F 66 74 5C 5C 57 69 6E 64 6F 77 73 5C  rosoft\\Windows\\
00000120  5C 53 74 61 72 74 20 4D 65 6E 75 5C 5C 50 72 6F  \\Start Menu\\Pro
00000130  67 72 61 6D 73 5C 5C 53 74 61 72 74 75 70 5C 5C  grams\\Startup\\
00000140  48 6E 63 43 68 65 63 6B 2E 6C 6E 6B 29 0D 0A 09  HncCheck.lnk) ...
```

1. 포스트스크립트 파일에 시작프로그램을 등록시키는 경로가 존재

2. 분석 – 포스트스크립트(파일 생성)

<2017년 금융보안 표준화 수요조사 안내>

```
000001E0  66 65 6C 73 65 0D 0A 0D 0A 09 09 09 63 75 72 72  false.....curr
000001F0  65 6E 74 66 69 6C 65 20 64 61 74 61 73 74 72 69  entfile datastri
00000200  6E 67 20 72 65 61 64 68 65 78 73 74 72 69 6E 67  ng readhexstring
00000210  0D 0A 09 09 09 7B 0D 0A 09 09 09 09 6F 75 41 20  .....{.....ouA
00000220  65 78 63 68 20 77 72 69 74 65 73 74 72 69 6E 67  exch writestring
00000230  0D 0A 09 09 09 7D 0D 0A 09 09 09 7B 0D 0A 09 09  .....}{.....
```

2. 포스트스크립트 파일에 'readhexstring'과 'writestring' 문자열이 존재

2. 분석 – 포스트스크립트(셸코드)

셸코드를 포함한 포스트스크립트

한글의 취약점을 이용한 셸코드를 구성하여

공격자가 의도한 행위를 수행한다.

2. 분석 – 포스트스크립트(셸코드)

고스트스크립트 취약점 (CVE-2017-8291)

포스트스크립트 언어를 처리하는 인터프리터인 고스트스크립트가

EPS 파일을 처리하는 과정에서 발생하는 취약점

2. 분석 - 포스트스크립트(셸코드)

<kinu 전문가 자문 요청사항(한미동맹과 한중관계)>

000066d0	43 36 35 32	30 37 31 37	35 36 39 37	34 30 41 30	C6520717569740A0
000066e0	30 30 30 30	30 30 30 30	30 30 30 30	30 30 30 30	0000000000000000
000066f0	30 30 30 3e	20 64 65 66	20 2f 6c 69	6d 69 74 20	000> def /limit
00006700	7b 61 72 20	6c 65 6e 67	74 68 20 2d	31 20 61 64	{ar length -1 ad
00006710	64 7d 64 65	66 20 2f 6c	65 6e 20 7b	61 72 20 6c	d}def /len {ar l
00006720	65 6e 67 74	68 7d 64 65	66 20 2f 73	74 72 20 6c	ength}def /str l
00006730	65 6e 20 73	74 72 69 6e	67 20 64 65	66 20 61 72	en string def ar
00006740	20 30 20 31	20 6c 69 6d	69 74 20 7b	32 20 63 6f	0 1 limit {2 co
00006750	70 79 20 67	65 74 20 30	30 30 20 78	6f 72 20 70	py get 000 xor p
00006760	75 74 20 61	72 7d 66 6f	72 20 70 6f	70 20 73 74	ut ar}for pop st
00006770	72 20 30 20	31 20 6c 69	6d 69 74 20	7b 64 75 70	r 0 1 limit {dup
00006780	20 61 72 20	65 78 63 68	20 67 65 74	20 70 75 74	ar exch get put
00006790	20 73 74 72	7d 66 6f 72	20 0d 0a 2f	49 6d 61 67	str}for ../Imag
000067a0	65 54 79 70	65 20 31 20	64 65 66 20	0d 0a 63 76	eType 1 def ..cv
000067b0	78 20 65 78	65 63 20 65	78 65 63 be	0c 79 0c f5	x exec exec..y..

1. 키 값을 이용하여 셸코드를 xor 연산으로 디코딩하는 루틴이 존재한다.

2. 분석 - 포스트스크립트(셸코드)

<kinu 전문가 자문 요청사항(한미동맹과 한중관계)>

000066d0	43 36 35 32	30 37 31 37	35 36 39 37	34 30 41 30	c6520717569740A0
000066e0	30 30 30 30	30 30 30 30	30 30 30 30	30 30 30 30	0000000000000000
000066f0	30 30 30 3e	20 64 65 66	20 2f 6c 69	6d 69 74 20	000> def /limit
00006700	7b 61 72 20	6c 65 6e 67	74 68 20 2d	31 20 61 64	{ar length -1 ad
00006710	64 7d 64 65	66 20 2f 6c	65 6e 20 7b	61 72 20 6c	d}def /len {ar l
00006720	65 6e 67 74	68 7d 64 65	66 20 2f 73	74 72 20 6c	ength}def /str l
00006730	65 6e 20 73	74 72 69 6e	67 20 64 65	66 20 61 72	en string def ar
00006740	20 30 20 31	20 6c 69 6d	69 74 20 7b	32 20 63 6f	0 1 limit {2 co
00006750	70 79 20 67	65 74 20 30	30 30 20 78	6f 72 20 70	py get 000 xor p
00006760	75 74 20 61	72 7d 66 6f	72 20 70 6f	70 20 73 74	ut ar}for pop st
00006770	72 20 30 20	31 20 6c 69	6d 69 74 20	7b 64 75 70	r 0 1 limit {dup
00006780	20 61 72 20	65 78 63 68	20 67 65 74	20 70 75 74	ar exch get put
00006790	20 73 74 72	7d 66 6f 72	20 0d 0a 2f	49 6d 61 67	str}for ../Imag
000067a0	65 54 79 70	65 20 31 20	64 65 66 20	0d 0a 63 76	eType 1 def . cv
000067b0	78 20 65 78	65 63 20 65	78 65 63 be	0c 79 0c f5	x exec exec..y..

2. 포스트스크립트 파일에 'cvx'와 'exec' 문자열이 존재한다.

2. 분석 – 힙스프레이 공격

힙스프레이 기법

힙에 수많은 NOP 코드와 쉘코드를

스프레이로 뿌리듯이 삽입하는 기법

2. 분석 – 힙스프레이 공격

<차기 정부의 대북 정책 제언을 위한 대토론회>

Offset	Size	Tag	Description
000000AE	0028	HWPTAG_PAGE_DEF	용지 설정
000000DA	001C	HWPTAG_FOOTNOTE_SHAPE	각주/미주 모양
000000FA	001C	HWPTAG_FOOTNOTE_SHAPE	각주/미주 모양
0000011A	000E	HWPTAG_PAGE_BORDER_FILL	쪽 테두리/배경
0000012C	000E	HWPTAG_PAGE_BORDER_FILL	쪽 테두리/배경
0000013E	000E	HWPTAG_PAGE_BORDER_FILL	쪽 테두리/배경
00000150	0010	HWPTAG_CTRL_HEADER	컨트롤 헤더
00000164	0004	HWPTAG_CTRL_HEADER	컨트롤 헤더
0000016C	0010	HWPTAG_LIST_HEADER	문단 리스트 헤더
00000180	0016	HWPTAG_PARA_HEADER	문단 헤더
0000019A	0004	HWPTAG_PARA_TEXT	문단의 텍스트
000001A2	0008	HWPTAG_PARA_CHAR_SHAPE	문단의 글자 모양
000001AE	0016	HWPTAG_PARA_HEADER	문단 헤더
000001C8	0008	HWPTAG_PARA_CHAR_SHAPE	문단의 글자 모양
000001D4	0016	HWPTAG_PARA_HEADER	문단 헤더
000001EE	100035A	HWPTAG_PARA_TEXT	문단의 텍스트
01000550	0008	HWPTAG_PARA_CHAR_SHAPE	문단의 글자 모양
0100055C	0016	HWPTAG_PARA_HEADER	문단 헤더
01000576	0008	HWPTAG_PARA_CHAR_SHAPE	문단의 글자 모양
01000582	0024	HWPTAG_PARA_LINE_SEG	문단의 레이아웃
010005AA	0016	HWPTAG_PARA_HEADER	문단 헤더
010005C4	0020	HWPTAG_PARA_TEXT	문단의 텍스트
010005E8	0008	HWPTAG_PARA_CHAR_SHAPE	문단의 글자 모양
010005F4	0024	HWPTAG_PARA_LINE_SEG	문단의 레이아웃

1. 각 섹션의 Tag 부분에서 데이터의 Size가 비정상적으로 큰 부분이 존재한다.

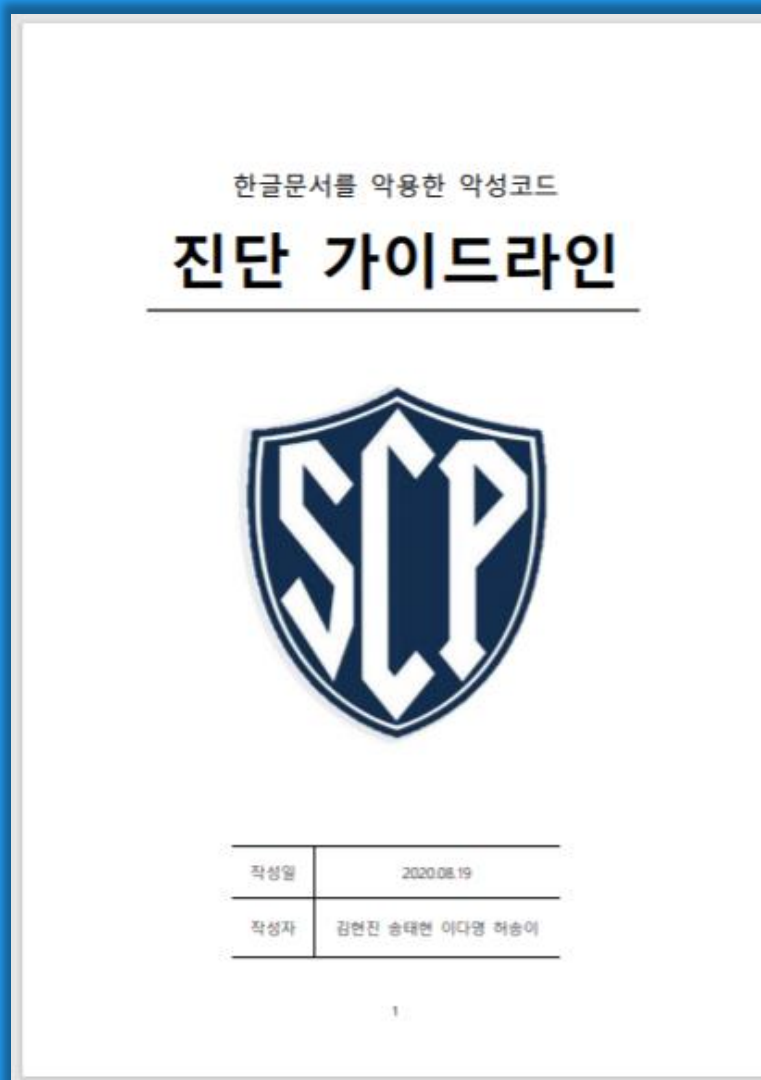
2. 분석 – 힙스프레이 공격

<차기 정부의 대북 정책 제언을 위한 대토론회>

000001E0	14	00	00	00	01	00	00	00	00	00	00	00	00	00	00	43	0CC.
000001F0	F0	FF	5A	03	00	01	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C	8ÿZ.....
00000200	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C
00000210	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C
00000220	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C
00000230	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C
00000240	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C
00000250	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C
00000260	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C
00000270	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C
00000280	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C

2. 섹션 내부에 '0x0C' hex 값이 과다하게 존재한다.

3. 가이드라인 작성



3. 가이드라인 작성 – 개요 및 분석환경

목차

1. 목차	2
2. 개요	3
3. 분석환경 및 한글버전 정보	4
4. 진단 기준표	5
5. 공격유형 비율	6
6. 진단 방식	8
1. 매크로를 악용한 방식	9
2. 포스트 스크립트	11
2-1. 파일이 임베드 된 포스트 스크립트	13
2-2. 웹코드가 삽입된 포스트 스크립트	16
2-2-1. 힙 스프레이	18
7. 진단 항목 검증	20
8. 부록	23

개요

한글 문서(HWP)를 이용한 악성코드는 지속적으로 발견되고 있는 악성코드 공격 방식이다. 한글 문서(HWP)는 주로 우리나라에서 개인뿐만 아니라 정부 기관 그리고 수많은 기업에서 이용하고 있으며, 공격자들은 이러한 점을 악용해 다양한 악성 한글 문서를 유포하고 있다. 그에 따라 악성 한글 문서를 분석한 보고서들은 많이 존재한다. 그러나 단일 샘플 또는 유형에 대한 보고서들이 대부분이고, 유형별로 가이드라인을 제시해주는 통합된 문서는 존재하지 않는다. 따라서 우리는 여러 악성 샘플들을 직접 분석해보는 것과 동시에 기존 분석 보고서들을 참고해 악성 한글 문서 진단표와 진단 가이드라인을 제시하였다.

실습은 기존에 배포된 악성 한글 문서에 관한 다양한 보고서를 참고하여 진행하였으며, 신빙성을 높이기 위해 Anlab, ESTsecurity와 같은 정보 보안 기업 자료를 주로 참고하였다.

분석하는 이들은 진단표를 통해 점검함으로써 분석하는 악성 한글 문서가 어떤 방식으로 악성행위를 수행하는지 빠르게 확인할 수 있다. 또한 실제로 직접 분석할 수 있도록 우리가 확인한 정보들을 진단 내용에 자세하게 나타내었으며, 위에서 언급했듯이 최대한 인증된 자료를 참고하여 제작하였기 때문에 많은 사용자들이 의심하지 않고 다양한 내용을 참고할 수 있기를 바란다.

분석환경 및 한글버전 정보

우리는 잠재적인 악성코드의 위험성을 배제하기 위하여 다음과 같은 환경에서 한글 악성문서샘플을 수집 및 분석하였다. 우리 문서에 나온 샘플을 분석하기를 원하는 사람들을 위해 분석환경 및 한글 버전 정보에 대한 것들을 다음과 같이 기록하였으므로 참고하길 바란다.

① 분석환경



[VMware Workstation_Windows7 x64] 환경에서 실습을 진행하였으며 네트워크 모드는 Host Only 모드를 선택하였고 샘플을 분석할 때 매다 스냅샷을 찍어 감염이 되더라도 다시 복구 할 수 있는 환경을 구성함.

② 한글 오피스 버전

문서 버전:	5.0.3.0.1	프로그램 버전:	8.0.0.455
--------	-----------	----------	-----------

한글 2010 버전 사용. 대부분의 악성 문서의 버전은 5.0.3.0.1로 프로그램 버전만 점진 업그레이드 된 것을 확인할 수 있음.

③ 분석을 위해 사용한 Tool 정보

1. HwpScan2 : Decompress 를 확인하거나 OLE 구조를 확인하기 위해 사용.
2. HxD : 문자열과 hex 값을 알아내고 위치 및 크기를 알아내기 위해 사용.
3. Anyrun : 다양한 한글 악성 문서를 다운받기 위해 사용.
4. WinPrefetchView : 한글 악성문서를 열람했을 때 실행되는 프로그램을 확인하기 위해 사용.

3. 가이드라인 작성 – 진단 기준표

< 진단 기준표 >

공격 유형	진단 기준	링크	Yes	No
공통	1. explore.exe 파일이 실행되었는가?			
	2. powershell.exe 파일이 실행되었는가?			
매크로	3. 문서를 열었을 때, 스크립트 매크로 실행 여부 알림 창이 뜨는가?	#a		
	4. 매크로 코드 편집의 Document - Open 항목에 PowerShell을 사용한 악성 스크립트가 삽입되어 있는가?	#b		
포스트스크립트(공통)	5. 문서를 열었을 때 gbb.exe 파일과 gwin32.exe 파일이 실행되었는가?	#c		
	6. '.EPS' 또는 '.PS'의 확장명을 가진 파일이 HWP 파일 구조의 BinData 스토리지에 zlib 압축이 되어 있는가?	#d		
파일 임베딩	7. 포스트스크립트 파일 내부에 'C:\Users\%UserProfile%\AppData\Roaming\Microsoft\Windows\StartMenu\Programs\Startup\%문자열이 존재하는가?	#e		
	8. 포스트스크립트 파일 내부에 'read' 또는 'write' 문자열이 존재하는가?	#f		
포스트스크립트	9. 포스트스크립트 파일 내부에 특정 변수명(/envstr, /appenv, /file0~ /file10 /path0~ /path10)이 사용되었는가?	#g		
	10. 포스트스크립트 파일 내부에 '4D 5A' hex 값(MZ)이 존재하는가?	#h		
오피스 문서	11. 문서를 열었을 때 'C:\Users\%UserProfile%\AppData\Local\Temp\%Inc\%BinData\다음과 같은 경로에 EPS 파일이 생성되었는가?	#i		
	12. 포스트스크립트 파일 내부에 '\xor(0x78 0x6f 0x72)'이라는 문자열이 존재하는가?	#j		
오피스 문서	13. 포스트스크립트 파일 내부에 '\csc'와 '\exec' 문자열이 존재하는가?	#k		
	14. byte 평균보다 큰 stream이 존재하는가?	#l		
오피스 문서 (CVE-2017-8291)	15. 섹션 내부에 '\x00'을 제외한 특정 hex 값이 일정 횟수 이상 과다하게 반복되는가?	#m		
	16. Nop Sled 이후에 오피스 문서로 의심되는 부분이 존재하는가?	#n		
오피스 문서 (CVE-2017-8291)	17. 복호화된 오피스 문서의 시그니처가 'MZ(4D 5A)'인가?	#o		
합계				

각 항목에 대한 자세한 설명을 링크 걸어두어
손쉽게 참조할 수 있도록 하였다.

각 항목에 해당하는지 체크할 수 있도록
만들어 어떤 유형의 악성코드인지
손쉽게 알 수 있도록 하였다.

3. 가이드라인 작성 - 공격 유형 비율

악성 한글문서 공격유형 비율



악성문서의 공격유형에 관한 공식적인 통계 수치는 존재하지 않아, 이 문서에서는 Anlab ASEC 분석팀의 '악성코드 정보'와 'ASEC REPORT'를 참고하였다. 악성문서 공격유형의 최신 동향에 따라 자료가 업데이트 된다는 점을 고려하여, 2018년부터 2020년 현재까지의 자료에서 나타난 공격 유형의 개수를 비율로 나타내었다.

악성문서 공격 유형 중 셀 코드가 삽입된 포스트스크립트가 가장 많은 비중을 차지하고 있다. 이는 유포된 악성문서의 상당수가 특정 취약점을 이용하고 있기 때문인데, 해당 취약점이 바로 EPS 파일 처리와 관련된 취약점인 CVE-2017-8291이다. 한글과 컴퓨터에서는 이 취약점과 관련된 보안 업데이트를 배포하고 한글 프로그램의 EPS 파일 삽입 및 보기 기능을 제거하였다. 그러나 여전히 EPS 파일을 이용한 악성문서가 유포되고 있다. 한글 문서형 악성코드는 특정 조직을 타겟으로 유포하는데, 한글문서 보안 패치 업데이트가 미흡한 기관이나 기업에서는 아직까지 피해가 발생하고 있다. 또한 최근 동일한 취약점을 이용하면서도 동작 방식을 변화시킨 악성문서가 유포되고 있다. 변화된 점은 기존의 XOR 인코딩이나 난독화 형태를 이용하는 것이 아닌, 형태가 매우 단순하면서도 악성 기능이 실행될 수 있는 포스트스크립트 문법을 사용한다는 것이다. 악성코드의 형태가 단순해짐으로써 정상적인 EPS 파일과 구별하는 것이 어려워지게 되었다. 악성 EPS 파일의 형태는 앞으로도 변경될 가능성이 있기 때문에 피해가 계속될 것으로 예상된다.

3. 가이드라인 작성 – 진단 가이드라인

[1] 매크로를 악용한 방식

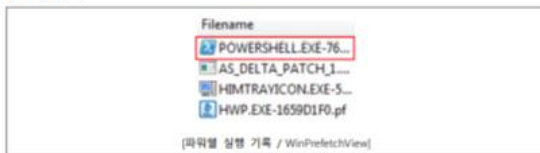
스크립트 매크로는 키보드와 마우스 동작을 포함하여 사용자가 정의하는 동작을 특정 단축키에 기록하여 매크로로 이용할 수 있는 기능이다. 한글문서의 매크로는 자바스크립트 언어를 사용하기 때문에, 매크로 기능을 악용하는 악성 한글문서 또한 자바스크립트 형식으로 구성되어 있다. 이 방식은 2007~2008년에 유행하였다가 이후로는 자주 이용되지 않았지만, 간혹 발견되고 있는 방식이기에 여전히 주의가 필요하다.

▶ 진단 방식

1. 매크로 스크립트가 삽입된 한글 문서를 열람하면, 스크립트 매크로 실행 여부에 관한 창이 뜬다.



2. 파워 셸을 이용하여 악성 스크립트를 실행하므로, 스크립트 매크로를 실행하였을 때 파워 셸이 함께 실행된다.



[2-1] 파일이 임베드된 포스트 스크립트

파일 객체를 생성하고 접근할 수 있는 특성을 이용하여 포스트스크립트 언어를 통해 로컬 시스템에 파일(악성코드)을 드롭 및 실행시키는 방식이다. 최근 공격 방식에서는 단일 파일이 아닌 다수의 파일을 생성하고, 이를 조합해 최종 악성 실행 파일을 생성하는 양상을 보인다.

1. 별다른 후회없이 파일을 드롭 및 실행시키는 방식	
2017년 금융보안 표준화 수요조사 연내	MD5: 12CC43A48D7E538B23688C8E10610A27
2. 보안 솔루션의 탐지를 우회하여 파일을 드롭 및 실행시키는 방식	
[한국정지학회] 준계학술회의 프로그램	MD5: de4b2a970fbfab5209c9aa5c1a541ff76

▶ 진단 방식

1. Decompress한 포스트스크립트 파일에서 다음과 같은 string을 검색해본다.

1-1. %Roaming%\Microsoft\Windows\Start Menu\Programs\Startup*

- 시작프로그램에 exe, lnk, vbs, vbe 등 파일을 생성함으로써 재부팅할 때 감염행동을 시작하도록 한다.

```
000000F0 72 69 6E 67 73 20 70 31 20 26 5C 5C 2E 2E 5C 5C  ringa pl (\\...\
00000100 2E 2E 5C 5C 52 6F 61 6D 69 6E 67 5C 5C 4D 69 63  ..\..\..\..\..\
00000110 72 6F 73 6F 66 74 5C 5C 57 69 6E 66 6F 77 73 5C  E:\..\..\..\..\
00000120 5C 53 74 61 72 74 20 4D 65 6E 75 5C 5C 50 72 6F  \Start Menu\Pro
00000130 6F 72 61 6D 73 5C 5C 53 74 61 72 74 75 70 5C 5C  game\Startup\
00000140 48 6E 63 43 69 63 63 63 2E 6C 6E 6D 29 0D 0A 09  HmCheck.Ink)...

<2017년 금융보안 표준화 수요조사 연내 / HxD64>
```

1-2. readhexstring, writestring

- readhexstring: 16진법으로 표현된 이진 데이터를 읽어 들어는 포스트스크립트 명령어

- writestring: 문자열을 파일로 옮기는 포스트스크립트 명령어

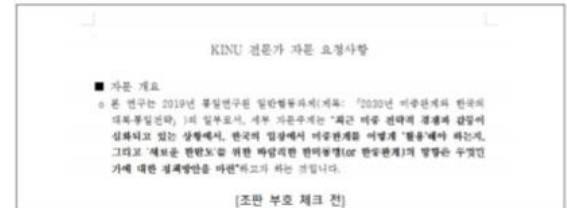
[2-2] 쉘 코드가 삽입된 포스트 스크립트

인터프리터 취약점을 공격하여 셸코드를 실행하는 유형이다. 최근 몇 년간 가장 많이 발견된 취약점은 CVE-2017-8291이다. 이 문서에서는 CVE-2017-8291을 이용하여 셸코드를 삽입한 포스트 스크립트에 대한 진단 방식을 제시한다. 해당 취약점은 포스트스크립트 언어를 처리하는 고스트 스크립트 인터프리터가 EPS 파일을 처리하는 과정에서 발생하는 취약점으로, 일명 '고스트버트 (GhostButt)'이라고 불린다.

분석 샘플	
kinu 전문가 자문 요청사항(한미동맹과 한중관계)	MD5 : 42ae424f27d83fa132b2967b64f6ba21

▶ 진단 방식

1. 보기의 '조판 부호'에 체크를 하면 [그림] 표시를 찾을 수 있으며, 해당 부분이 EPS 파일이다.



3. 가이드라인 작성 - 진단 항목 검증

툴 제작

우리는 진단 기준표를 바탕으로 해당 문서가 어떠한 공격 유형의 악성 문서인지 판단하는 툴을 제작하였다. 툴을 제작한 목적은 사용자가 위 가이드라인의 진단 방식을 신임할 수 있도록 하기 위함이다. 가이드라인 검증에 위해 수행한 과정은 다음과 같다. 우선 툴을 사용하여 해당 문서가 악성 문서가 맞는지 맞다면 어떠한 공격 유형에 속하는지 진단하였다. 이후 Virus Total의 검사 결과와 일치하는지 비교하였다. 또한 사용자가 진단 기준표를 통해 해당 문서의 공격 유형을 판단하고 나서 더욱 확실한 검증이 필요하다면, 툴을 사용하여 정확한 진단 결과를 확인할 수 있다. 툴 실행 화면은 아래와 같다.

```
===== RESTART: C:\Users\truebird\Desktop\Hy4_V3.py =====
-----choose your file-----
1 . 7.kinu.hwp
2 . fileimbedied.hwp
3 . Noname1.hwp
4 . nonamed.hwp
5 . [한국정치학회] 춘계학술회의 프로그램.hwp_
choose number(if 0 == finish) : 4
정상 파일 입니다 !!!
>>>

===== RESTART: C:\Users\truebird\Desktop\Hy4_V3.py =====
-----choose your file-----
1 . 7.kinu.hwp
2 . fileimbedied.hwp
3 . Noname1.hwp
4 . nonamed.hwp
5 . [한국정치학회] 춘계학술회의 프로그램.hwp_
choose number(if 0 == finish) : 1
5.문서를 열었을 때 gbb.exe 파일과 gswin32c.exe 파일이 실행되었는가?
6.'.EPS' 또는 '.PS'의 확장명을 가진 파일이 HWP 파일 구조의 BinData 스토리지에 zlib 압축이 되어있는가?
11.문서를 열었을 때 'C:\Users\[사용자명]\AppData\Local\Temp\Hnc\BinData' 다음과 같은 경로에 EPS 파일이 생성되었는가?
12.포스트스크립트 파일 내부에 'xor(0x78 0x6f 0x72)'이라는 문자열이 존재하는가?
13.포스트스크립트 파일 내부에 'cvx'와 'exec' 문자열이 존재하는가?
14.byte 평균보다 큰 stream이 존재하는가
>>> |
```

툴은 파일을 선택하면 해당되는 진단 기준 항목들이 출력되는 형식으로 구성되어 있다. 첫 번째 파일 선택에서 4번 문서를 선택했을 때, 진단 기준에 모두 해당하지 않으므로 '정상 파일입니다'가 출력되었다. 두 번째 파일 선택에서 1번 문서를 선택했을 때, 포스트 스크립트(공통)와 헬코드가 삽입된 포스트스크립트((CVE-2017-8291)의 진단 기준 항목들

```
===== RESTART: C:\Users\truebird\Desktop\Hy4_V3.py =====
-----choose your file-----
1 . 7.kinu.hwp
2 . fileimbedied.hwp
3 . Noname1.hwp
4 . nonamed.hwp
5 . [한국정치학회] 춘계학술회의 프로그램.hwp_
choose number(if 0 == finish) : 4
정상 파일 입니다 !!!
>>>

===== RESTART: C:\Users\truebird\Desktop\Hy4_V3.py =====
-----choose your file-----
1 . 7.kinu.hwp
2 . fileimbedied.hwp
3 . Noname1.hwp
4 . nonamed.hwp
5 . [한국정치학회] 춘계학술회의 프로그램.hwp_
choose number(if 0 == finish) : 1
5.문서를 열었을 때 gbb.exe 파일과 gswin32c.exe 파일이 실행되었는가?
6.'.EPS' 또는 '.PS'의 확장명을 가진 파일이 HWP 파일 구조의 BinData 스토리지에 zlib 압축이 되어있는가?
11.문서를 열었을 때 'C:\Users\[사용자명]\AppData\Local\Temp\Hnc\BinData' 다음과 같은 경로에 EPS 파일이 생성되었는가?
12.포스트스크립트 파일 내부에 'xor(0x78 0x6f 0x72)'이라는 문자열이 존재하는가?
13.포스트스크립트 파일 내부에 'cvx'와 'exec' 문자열이 존재하는가?
14.byte 평균보다 큰 stream이 존재하는가
>>> |
```

진단 항목의 신뢰성을
심어주기 위해 툴 제작

3. 가이드라인 작성 - 부록

용어 정리

단어	뜻
스크립트 매크로	마우스와 키보드 동작을 그대로 기억했다가 단축키 하나로 실행시켜주는 기능
파워 셸	셸 및 스크립트 언어를 특징으로 하는 명령어 인터프리터이며 거의 모든 작업을 할 수 있고, 명령 프롬프트와 같은 기능을 한다
포스트 스크립트	페이지를 기술하며 디지털 인쇄를 위한 파일 출력하는 일을 하는 프로그래밍 언어
임베드	포함된다.
Eps	디지털 인쇄를 직접 제어하는 부분을 제외하여 만든 포스트 스크립트 규격
Zlib	압축 라이브러리
배치 파일	명령 인터프리터에 의해 실행되게끔 고안된 명령어들이 나열되어 있는 텍스트 파일
셸	운영체제 상에서 다양한 운영체제 기능과 서비스를 구현하는 인터페이스를 제공하는 프로그램
시그니처 (-해치넵)	파일 포맷이 기본이 되는 내용으로, 헤더 시그니처와 푸터 시그니처가 있음
헤더 시그니처	파일의 처음에 존재하는 시그니처
푸터 시그니처	파일의 마지막에 존재하는 시그니처
익스플로잇 취약점 공격	
리터럴	변수에 넣는 변하지 않는 데이터 자체를 의미함.
Nop Sled	실행을 해도 실행할 명령어가 없어 실행흐름을 아래로 흘러보내는 기법
숨은 설정	실제 문서의 내용에는 포함되지 않으면서 파일을 편집하는 사람에게 필요한 예모나 주의 사항등을 기록해 둘 수 있는 기능
MZ	PE 파일의 시그니처로, 다음 예시에서는 시그니처를 파일에서 분리함으로써 보안 솔루션의 탐지를 우회하는데 사용하였다.
envstr	감염 시스템의 %AppData% 경로를 저장한다.이후 추가적으로 생성할 악성코드들에 대한 이미지 경로를 정의한다.
file	취약점이 있는 파일을 file 변수를 사용하여 일부를 만들어 파일 이름은 file 변수를 통해 입력받는 것
path	파라미터로 전달되는 값은 통신을 수행할 때 임의로 지정된다
readhexstring	16진법으로 표현된 이진 데이터를 읽어 들어는 포스트스크립트 명령어
writestring	문자열을 파일로 옮기는 포스트스크립트 명령어

24

MD5

공격 유형별 악성문서 MD5

유형	이름	MD5
파일이 임베드된 포스트 스크립트	2017년 금융보안 표준화 수요조사 안내 [한국정지학회] 순계학술 회의 프로그램	12CC43A48D7E538B23688C8E10610A27 de4b2a970bfab5209c9aa5c1a541ff76
셸 코드가 삽입된 포스트 스크립트	kinu 전문가 자문 요청 사항(한미동맹과 한중관계)	42ae424f27d83fa132b2967b64f6ba21
힙 스프레이(heap Spray) 공격	자기 정부의 대북 정책 제언을 위한 대토론회	2b21faaf82eb90ebd4af235406784db0

분석 샘플 MD5

파일 형식	MD5
HWP	12CC43A48D7E538B23688C8E10610A27
HWP	de4b2a970bfab5209c9aa5c1a541ff76
HWP	42ae424f27d83fa132b2967b64f6ba21
HWP	2b21faaf82eb90ebd4af235406784db0
HWP	bb7ae94a52179f29d179fb51d4906db
HWP	ae5dcb17d5b83fdaec89ea6da8929921
HWP	8b65ea5f29bac6e7f1d976a36c79713
HWP	bbde7c694faf6b450adbfc8ef88a41a
HWP	f392492ef5a1b399b4c0af38810b0d6
HWP	48d9e625ea3efcbcef3963c8714544a7
HWP	7d1d7f9ee0e2f2778d9e941bcafbdb08

25

Reference

- ▶ 금융보안원 / 한글 문서를 이용하는 악성코드 프로파일링
- ▶ INCA Internet Security Analysis & Response Center / 한글 문서에 포함된 매크로 악성코드 분석
- ▶ GLOOsecurity / PostScript를 이용한 한글 악성코드 분석보고서
- ▶ Anlab / 포스트스크립트를 이용한 HWP 한글 문서 악성코드 주의
- ▶ ESTsecurity / Exploit.HWP.Agent
- ▶ Anlab / 달라진 2020년 한글 파일 공격, 가장 주목할 점은?
- ▶ Anlab / 공공기관-기업 위협하는 EPS 파일 공격, 핵심은?

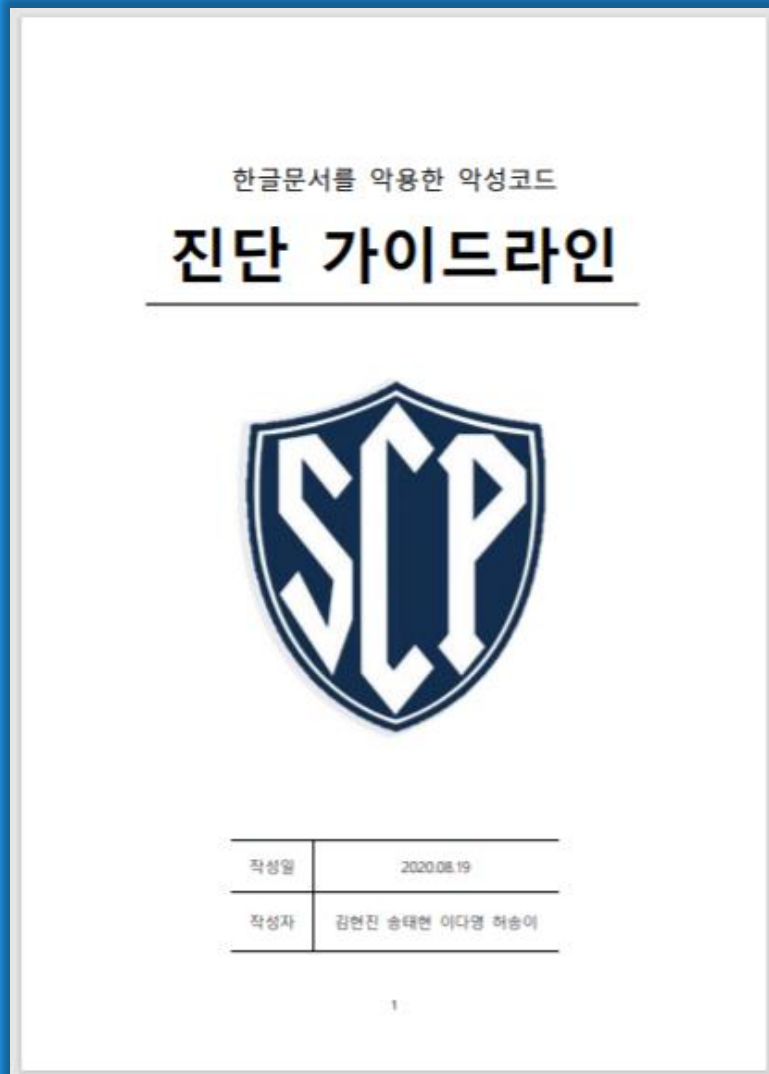
26

Part 3,

산출물 및 향후 계획

1. 가이드라인 초본
2. 가이드라인 보완
3. 논문작성

1. 가이드라인 초본 - 산출물



**악성 한글 문서
진단 가이드라인 초본 제작**

2. 가이드라인 보완 – 향후 계획

**진단항목 추가 및
신뢰도 검증 툴 보완**

3. 논문 작성 – 향후 계획

가이드라인 보완 후
논문 작성을 통해 가이드라인 배포

