

페이스북 정찰과 소프트웨어 변조 공격 분석

 genians.co.kr/blog/threat_intelligence/pretexting

Genians

April 12, 2026



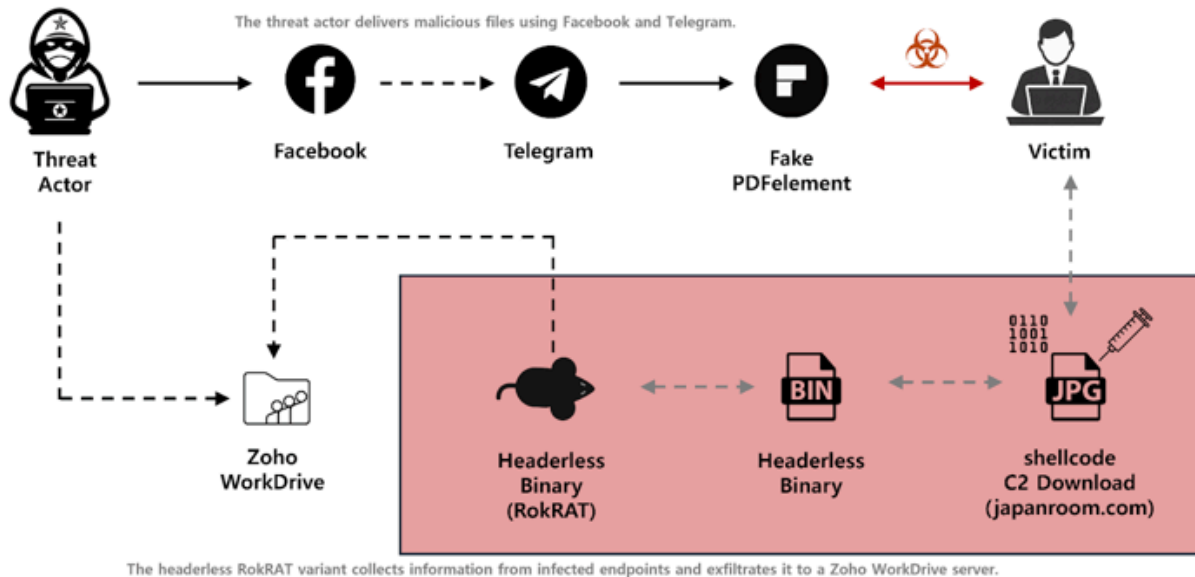
◆ 주요 결과 (Key Findings)

- 북한 평양·평성 출신 표기 특정 페이스북 계정 2개를 활용한 정찰 수행
- 페이스북 친구 추가로 신뢰 구축 후 메신저 전환, 특정 주제로 표적 유인
- 암호화된 PDF 군사 무기 문서 제공을 빙자하여 전용 PDF Viewer 설치 유도
- 정교하게 변조된 Wondershare PDFelement 설치본을 통한 Shellcode 실행 및 초기 침투
- 일본 부동산 정보 서울지점 사이트를 경유한 JPG 위장 페이로드를 통한 후속 명령 수행
- 식별된 침해지표(IoC) 기반 탐지와 APT 회피 기법 대응을 위한 행위 기반 EDR 체계 필요

1. 개요 (Overview)

지니언스 시큐리티 센터(Genians Security Center)는 APT37 위협 행위자가 소셜 네트워크 플랫폼을 활용하여 수행한 표적 침투 캠페인에 대한 심층 분석을 수행하였습니다.

분석 결과, 해당 위협 행위자는 북한 평양·평성 지역으로 등록된 페이스북 계정 2개를 운용하며 공격 표적을 정찰·선별하고, 친구 요청을 통해 신뢰 관계를 구축한 뒤 메신저 채널로 전환하여 특정 주제로 표적을 유인하는 소셜 엔지니어링 기반의 1차 접근을 수행한 것으로 확인되었습니다.



[그림 1-1] 전체 공격 흐름도

페이스북 메신저나 텔레그램 등으로 암호화된 PDF 형식의 군사 무기 기술 문서를 제공하는 것처럼 위장하여, 이를 열람하기 위한 전용 PDF Viewer 프로그램의 설치를 유도하는 프리텍스팅 (Pretexting) 전술이 사용된 것으로 확인되었습니다.

프리텍스팅은 위협 행위자가 허위 시나리오나 명분을 사전에 만들어 놓고, 이를 통해 피해자가 자연스럽게 특정 행동을 하도록 유도하는 소셜 엔지니어링 기법입니다.

이번 사례에서는 "암호화된 군사 문서를 열람하려면 전용 뷰어가 필요하다"는 그럴듯한 구실을 만들어, 피해자가 스스로 악성 프로그램을 설치하도록 유도한 것이 이에 해당합니다.

PDF Viewer는 정교하게 변조된 Wondershare PDFelement 설치본으로, 실행 시 내장된 shellcode가 동작하여 초기 침투를 달성하는 정황이 식별되었습니다.

특히 공격자는 일본 부동산 정보 서비스의 서울지점 웹사이트를 C2 인프라로 악용하며, JPG 이미지 파일로 위장된 악성 페이로드를 통해 후속 명령을 전달·실행하는 다단계 공격 체계를 구성한 것으로 분석됩니다.

이는 정상 소프트웨어 변조, 합법 웹사이트 악용, 파일 확장자 위장이 결합된 고도의 탐지 회피 전략으로 평가됩니다.

본 위협 인텔리전스 분석 보고서는 페이스북 기반 표적 정찰 기법, 변조된 설치 패키지를 통한 shellcode 실행 구조, C2 통신 채널 은닉 방식, 그리고 이미지 위장 페이로드 전달 메커니즘 등 주요 기술적 아티팩트를 중심으로 공격 전술·기법·절차(TTP)를 체계적으로 분석합니다.

또한 식별된 침해지표(IoC) 기반의 시그니처 탐지와 함께, 정상 프로세스 위장 및 합법 인프라 악용 행위를 효과적으로 식별하기 위한 엔드포인트 행위 기반 탐지(EDR) 중심 대응 체계의 필요성을 제시합니다.

2. 배경 (Background)

국가 배후 위협 그룹 APT37은 소셜 엔지니어링 기법과 탐지 회피 전략을 지속적으로 고도화하며 표적 공격 활동을 전개하고 있습니다.

2025년 5월 보고한 '[작전명 토이박스 스토리\(Operation ToyBox Story\)](#)'에서는 한국 국가안보전략 싱크탱크를 위장한 스피어피싱과 클라우드 기반 C2 채널을 활용한 탐지 우회 전술이 확인된 바 있습니다.

이어 8월 보고한 '[RokRAT 셸코드 및 스테가노그래피 기반 위협 분석](#)'에서는 이미지 파일 내부에 악성 페이로드를 은닉한 뒤 셸코드로 실행하는 고도의 회피 기법이 식별되었습니다. 같은 해 12월 보고한 '[작전명 아르테미스\(Operation Artemis\)](#)'에서는 HWP 문서를 매개로 한 DLL 사이드로딩 기법을 통해, 정상 문서 열람 과정에서 악성 DLL이 로드되어 초기 침투가 이루어지는 정황이 확인되었습니다.

이번 캠페인은 이러한 전술적 진화의 연장선에 있으며, 초기 접근 벡터를 페이스북 소셜 네트워크로 전환한 것이 핵심적 특징입니다.

위협 행위자는 페이스북 및 텔레그램 등을 통한 신뢰 구축과 프리텍스팅 기반의 악성 소프트웨어 설치 유도, 변조된 설치 패키지 내 shellcode 실행, 합법 웹사이트를 경유한 C2 통신 은닉을 결합하여 다단계 공격 체계를 구성하였습니다.

이는 정상 소프트웨어 변조와 합법 인프라 악용, JPG 파일 확장자 위장이 결합된 고도의 회피 전략으로, 기존 보안 탐지 체계의 우회를 의도한 것으로 평가됩니다.

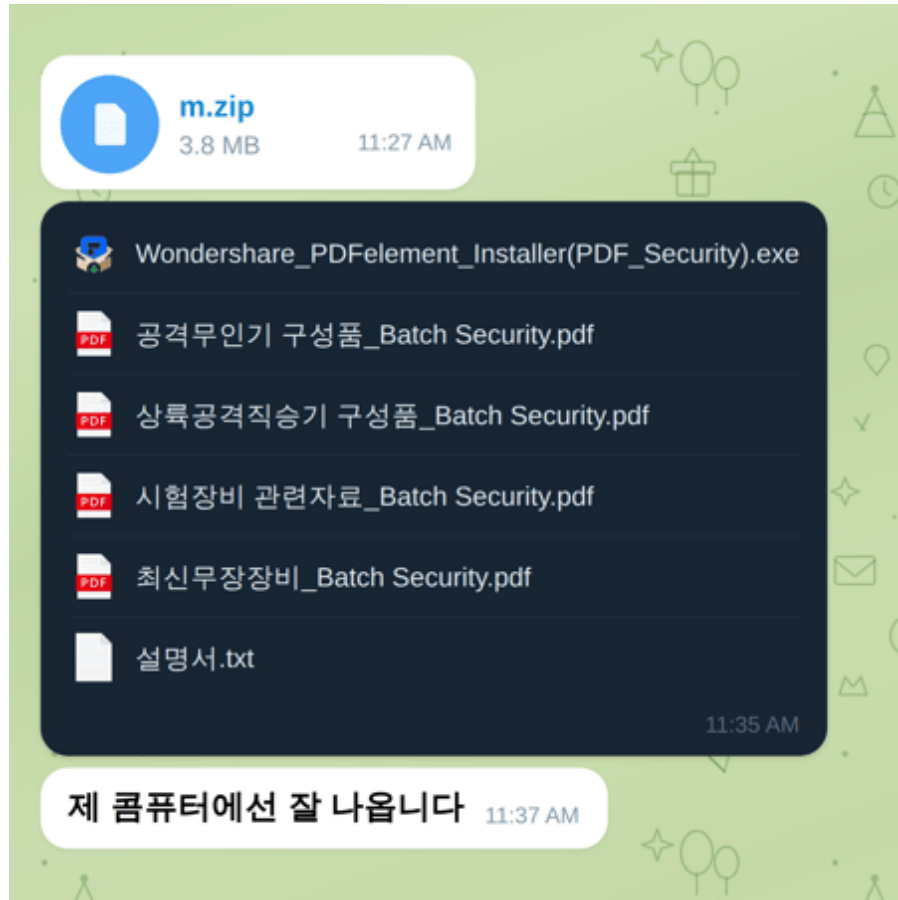
이러한 배경 하에 본 보고서는 APT37의 소셜 네트워크 기반 표적 정찰 및 침투 전술의 진화 양상을 조망하고, 공격 전술·기법·절차(TTP)의 기술적 특징을 체계적으로 분석합니다.

3. 공격 시나리오 (Attack Scenario)

초기 침투 단계에서는 소셜 네트워크 기반의 신뢰 구축과 프리텍스팅을 결합한 다단계 소셜 엔지니어링 전술이 주요 공격 벡터로 활용된 것으로 확인되었습니다.

위협 행위자는 먼저 페이스북을 통해 공격 표적과 친구 관계를 형성한 뒤, 메신저의 1:1 대화 채널에서 인사 및 자기소개를 교환하며 초기 신뢰 관계를 구축하였습니다.

이후 군사 무기 관련 기밀 자료를 제공할겠다는 거래 조건을 제시하여 표적의 관심을 유도한 다음, 텔레그램 등 보안성이 높은 별도의 메신저 채널로 대화를 전환하여 악성 파일을 전달하는 방식이 관찰되었습니다.



[그림 3-1] 텔레그램으로 파일을 전달하는 화면

위협 행위자는 텔레그램을 통해 "m.zip"이라는 압축 파일을 전송하였습니다. 해당 압축 파일 내부에는 PDF 뷰어로 위장한 악성 실행 파일과 함께 군사 관련 문서명을 가진 다수의 미끼 PDF 파일, 프로그램 사용 설명서로 위장된 TXT 파일이 포함되어 있었습니다.

특히 텔레그램 대화 내용 중 '컴퓨터'의 북한식 외래어 표기인 '컴퓨터'가 사용된 점, '설명서.txt' 파일 내부에 '프로그램', '화일' 등은 해당 위협 행위자의 출신 또는 배후를 추정할 수 있는 언어적 지표 (Linguistic Indicator)로 주목됩니다.

- 1.Wondershare_PDFelement_Installer(PDF_Security)프로그램을 설치합니다.
- 2.Wondershare프로그램을 열고 Batch Security항목을 선택합니다.
- 3.Batch Security에 들어간다음 암호화된 화일들을 선택하여 놓고 복호화 단추를 선택합니다.

[그림 3-2] '설명서.txt' 내용 중 일부 화면

텔레그램으로 전달된 파일은 암호화 압축 형태로 제공되었으며, 별도로 공유된 비밀번호를 통해 압축을 해제하면 내부에 암호화된 다수의 PDF 문서와 함께 EXE 설치 파일이 포함되어 있는 것으로 분석되었습니다.

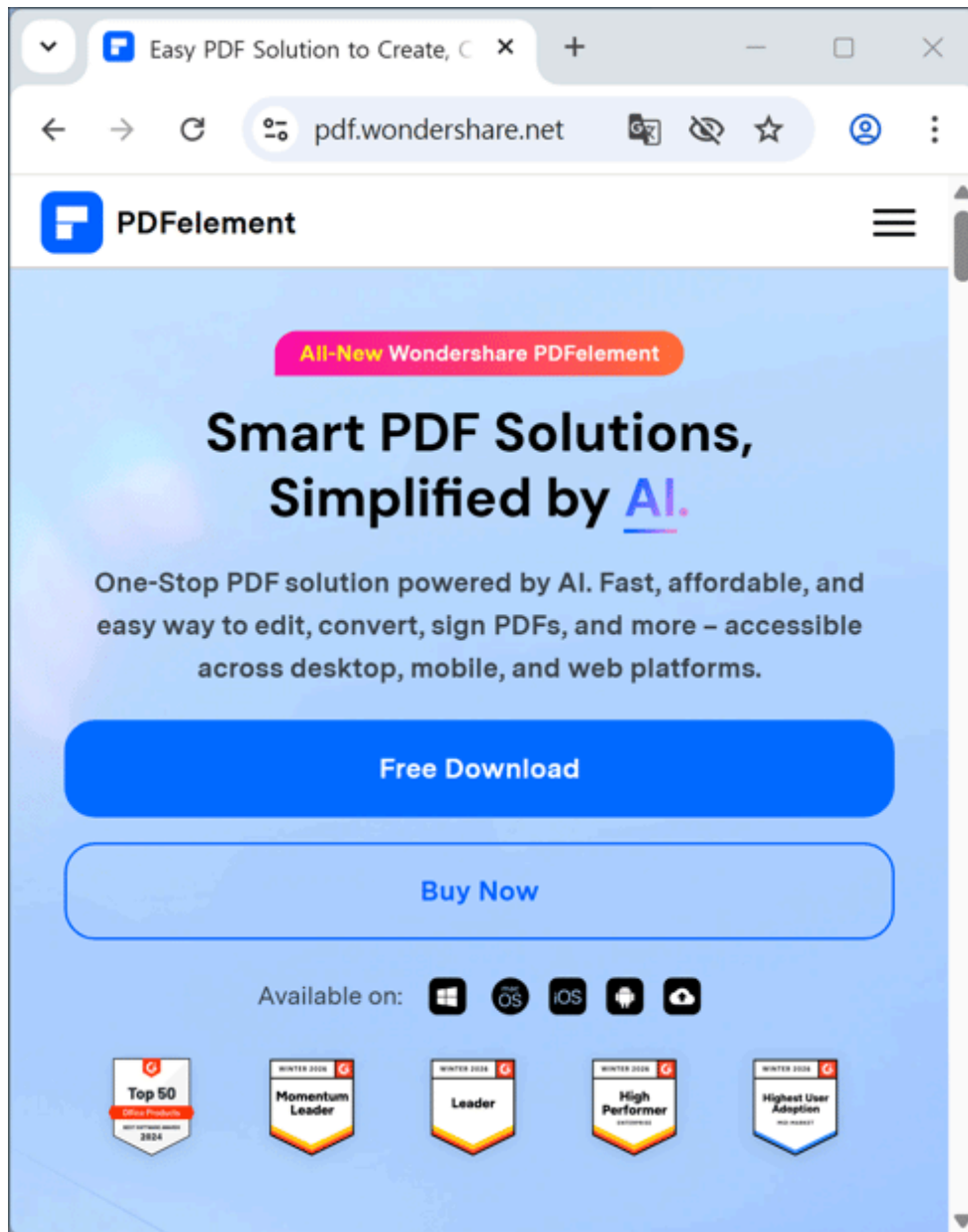
해당 설치 파일은 'Wondershare'사가 공식 배포하는 PDFelement 설치본과 외형상 유사하나, 실제로는 정상 파일을 정교하게 변조한 악성 파일로 확인되었습니다.

사용자가 이를 실행할 경우 내부에 삽입된 shellcode가 동작하여 악성 명령이 수행되며, 이를 통해 초기 침투가 달성되는 구조로 분석되었습니다.

4. 상세 분석 (Detailed Analysis)

4-1. 악성 파일 흐름

텔레그램을 통해 유포된 악성 파일은 정상 소프트웨어를 정교하게 모방한 것으로 확인되었습니다.



[그림 4-1] PDFelement 공식 사이트 화면

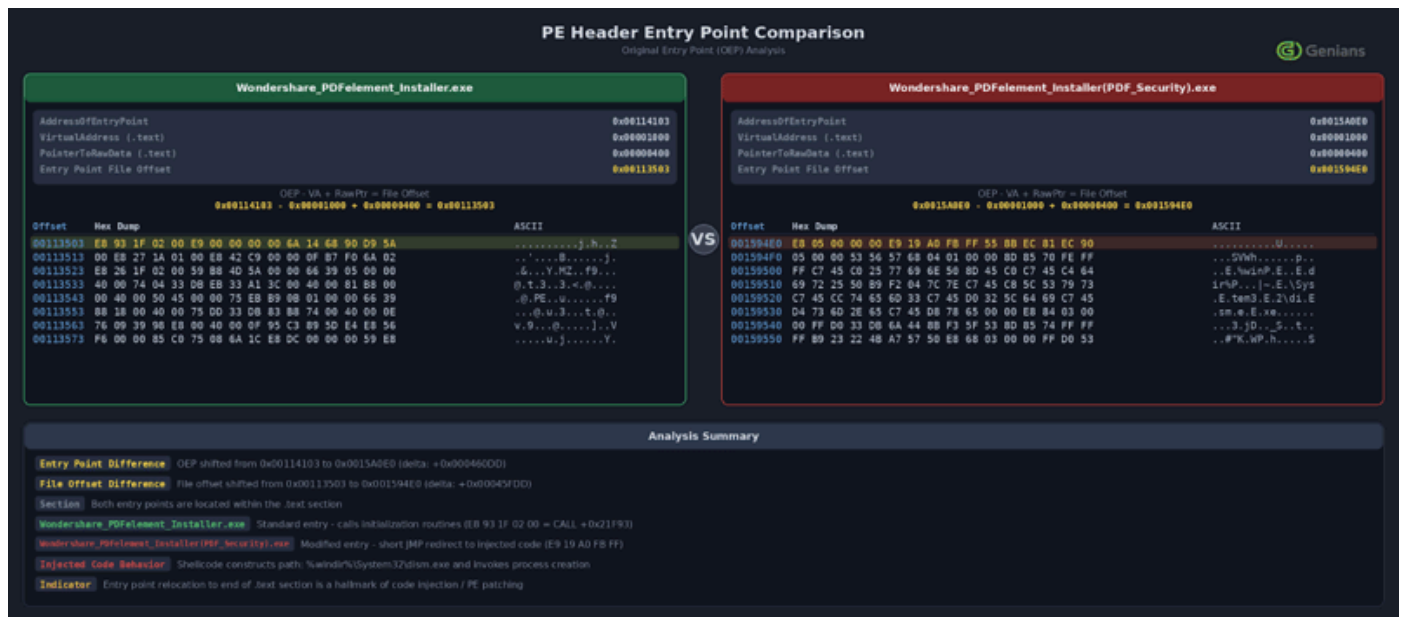
Wondershare 공식 사이트에서 제공되는 정상 설치본의 파일명은 'Wondershare_PDFelement_Installer.exe'인 반면, 위협 행위자가 텔레그램으로 유포한 악성 파일에는 'Wondershare_PDFelement_Installer(PDF_Security).exe'라는 파일명이 사용되었습니다.

이는 원본 파일명에 '(PDF_Security)'라는 문구를 의도적으로 삽입하여 보안 관련 정상 업데이트 또는 보안 강화 버전으로 인식하도록 유도한 것으로, 표적이 별다른 의심 없이 실행하도록 설계된 전형적인 파일명 위장(Filename Masquerading) 기법에 해당합니다.

또한, 공식 사이트에서 배포되는 정상 설치 파일에는 Wondershare의 유효한 디지털 서명(Digital Signature)이 탑재되어 파일의 무결성과 출처를 검증할 수 있는 반면, 텔레그램을 통해 유포된 조작된 설치 파일에는 디지털 서명이 포함되어 있지 않았습니다.

이는 위협 행위자가 정상 바이너리를 변조하거나 독자적으로 재구성하는 과정에서 원본 코드 서명이 무효화된 것으로 판단되며, 디지털 서명의 부재는 해당 파일이 공식 배포 채널을 통해 제공된 정상 소프트웨어가 아님을 식별할 수 있는 핵심 지표 중 하나입니다.

위협 행위자는 변조된 설치 프로그램의 메인 함수는 그대로 유지한 채, Entry Point만을 임의로 조작하는 방식을 사용하였습니다.



[그림 4-2] Entry Point 비교 화면

정상 설치 파일의 시작 주소(AddressOfEntryPoint)는 0x00114103으로, 프로그램 코드가 저장된 .text 섹션의 앞부분에 위치하며 표준적인 초기화 과정을 수행하는 정상적인 구조를 갖추고 있습니다. 이를 파일 내 실제 위치로 환산하면 0x00113503에 해당하며, 이 지점에서 프로그램의 정상적인 실행이 시작됩니다.

- 0x00113503
 - OEP - VirtualAddress + PointerToRawData
 - 0x00114103 - 0x00001000 + 0x00000400

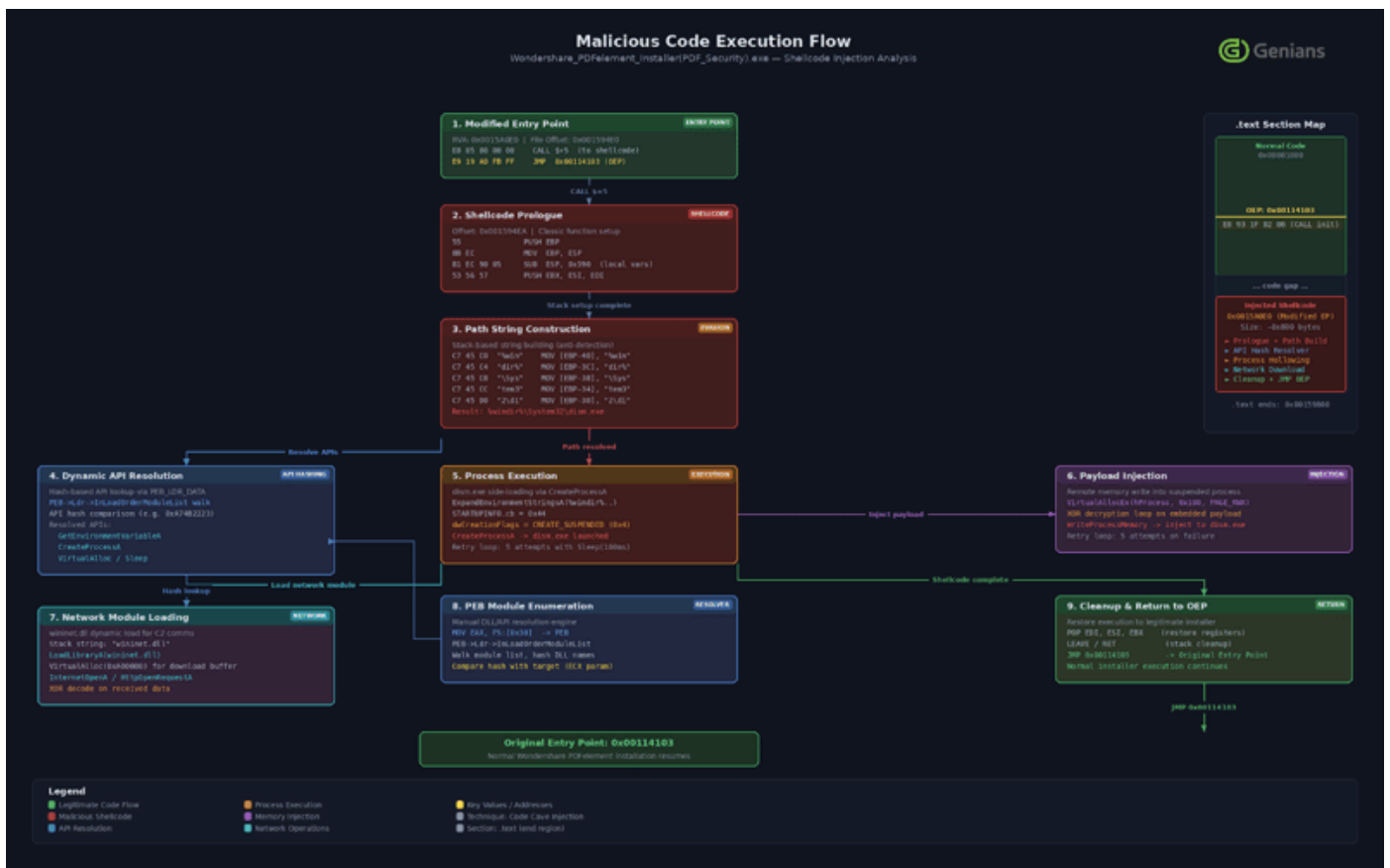
반면, 변조된 설치 파일에서는 시작 주소가 0x0015A0E0으로 변경되어 있습니다.

이 주소는 .text 섹션의 끝부분에 해당하는 파일 위치 0x001594E0에 자리하고 있으며, 정상 파일 대비 약 0x460DD 뒤쪽으로 이동된 것입니다.

위협 행위자는 .text 섹션 말미에 존재하는 사용되지 않는 빈 공간(Code Cave)에 약 2KB 규모의 악성 셸코드를 삽입한 뒤, 프로그램이 실행될 때 이 코드가 가장 먼저 동작하도록 시작 지점을 재설정 하였습니다.

이처럼 시작 지점을 코드 영역의 빈 공간으로 옮기고, 악성코드 실행 후 원래 위치로 되돌아가는 기법은 'PE 패칭(PE Patching)' 또는 '코드 케이브 인젝션(Code Cave Injection)'으로 불리는 대표적인 실행 파일 변조 수법입니다.

변조된 시작 지점(0x0015A0E0)에서의 실행 흐름은 다음과 같이 전개됩니다.



[그림 4-3] 악성 코드 명령 실행 흐름

프로그램이 시작되면, CALL 명령(E8 05 00 00 00)을 통해 바로 다음에 위치한 셸코드 본체 (0x001594EA)로 실행이 넘어갑니다. 동시에 역방향 점프 명령(E9 19 A0 FB FF)이 준비되어 있어, 악성코드 실행이 끝난 후 원래의 정상 시작 지점(0x00114103)으로 돌아갈 수 있는 복귀 경로를 미리 확보해 둡니다.

셸코드 본체는 일반적인 함수와 동일한 형태로 시작됩니다. 먼저 현재 스택 상태를 저장(PUSH EBP / MOV EBP, ESP)한 후, SUB ESP, 0x590 명령으로 1,424바이트의 임시 작업 공간을 스택에 마련합니다. 이어서 주요 레지스터 값을 보존(PUSH EBX / PUSH ESI / PUSH EDI)하여 이후 악성 행위를 수행하기 위한 실행 환경을 완성합니다.

다음 단계에서는 정적 분석 도구의 탐지를 우회하기 위한 문자열 은닉 기법이 사용됩니다. 대상 파일 경로를 바이너리에 직접 기록하는 대신, MOV 명령을 통해 4바이트씩 나누어 스택에 순차적으로 적재하는 방식으로 %windir%\System32\dism.exe 경로를 실행 시점에 동적으로 조립합니다.

이 방식을 사용하면 해당 경로 문자열이 파일 내에 평문 형태로 노출되지 않기 때문에, 특정 문자열 패턴을 기반으로 악성 여부를 판별하는 탐지 기법을 효과적으로 우회할 수 있습니다.

참고로 DISM(Deployment Image Servicing and Management)은 Microsoft Windows에서 시스템 이미지 파일(WIM, VHD, FFU 등)을 생성·수정·관리하기 위한 명령줄 기반 유틸리티입니다. 시스템 관리자와 고급 사용자가 Windows 설치 이미지와 오프라인 운영체제를 유지 관리할 때 주로 사용되며, Windows 7 이후 버전에서 기본으로 포함되어 있습니다.

셸코드는 프로그램의 함수 목록표(Import Address Table)를 직접 참조하지 않고, Windows 운영체제가 내부적으로 관리하는 프로세스 정보 구조체(PEB, Process Environment Block)를 경유하여 필요한 시스템 함수들의 주소를 직접 찾아냅니다.

이 해시 기반 함수 탐색 엔진은 셸코드 전반에 걸쳐 반복적으로 호출되며, 환경변수 확장(GetEnvironmentVariableA), 프로세스 생성(CreateProcessA), 메모리 할당(VirtualAlloc, VirtualAllocEx), 메모리 쓰기(WriteProcessMemory), 대기(Sleep) 등 핵심 시스템 함수의 주소를 프로그램 실행 중에 동적으로 확보합니다.

함수 주소 확보가 완료되면, ExpandEnvironmentStringsA를 호출하여 앞서 조립한 %windir%\System32\dism.exe 경로의 환경 변수 부분(%windir%)을 실제 시스템 경로(예: C:\Windows)로 변환합니다.

그 다음 프로세스 생성에 필요한 초기 설정 구조체(STARTUPINFO, 68바이트)를 준비하고, 생성 옵션에 CREATE_SUSPENDED(0x4)를 지정하여 CreateProcessA를 호출합니다.

이를 통해 Windows의 정상 시스템 도구인 dism.exe가 일시 중단(suspended) 상태로 생성됩니다.

일시 중단 상태란 프로세스가 메모리에 로드되었으나 아직 실행이 시작되지 않은 상태를 의미하며, 생성에 실패할 경우 0.1초 간격으로 최대 5회까지 재시도하는 안정성 확보 로직이 포함되어 있습니다.

프로세스를 일시 중단 상태로 생성하는 이유는, 해당 프로세스가 실행을 시작하기 전에 메모리 공간을 조작하여 악성 코드를 삽입하기 위한 사전 준비 과정이기 때문입니다.

dism.exe 프로세스가 성공적으로 생성되면, VirtualAllocEx를 통해 해당 프로세스의 메모리 공간에 읽기, 쓰기, 실행이 모두 가능한(PAGE_EXECUTE_READWRITE) 영역을 할당합니다.

이어서 셸코드 내부에 숨겨져 있던 암호화된 악성 코드를 XOR 연산 루프를 통해 복호화하고, WriteProcessMemory를 사용하여 복호화된 실행 코드를 dism.exe의 할당된 메모리 영역에 기록합니다.

이 과정에서도 실패 시 최대 5회 재시도하는 동일한 안정성 로직이 적용됩니다. 이와 동시에, 셸코드는 C2와 통신하기 위한 네트워크 기능을 별도로 준비합니다.

인터넷 통신에 필요한 라이브러리(wininet.dll) 이름을 스택에 동적으로 구성한 뒤 LoadLibraryA를 호출하여 해당 라이브러리를 메모리에 로드하고, VirtualAlloc으로 약 10MB 규모의 데이터 수신 버퍼를 확보합니다.

서버로부터 수신된 데이터는 XOR 연산을 통해 복호화된 후 후속 공격 단계에 활용됩니다. 모든 악성 행위가 완료되면, 셸코드 함수는 RET 명령으로 반환됩니다.

그리고 사전에 준비해 둔 JMP 0x00114103 명령을 통해 원래의 정상 시작 지점으로 실행 흐름을 복귀시키며, 이 시점부터 Wondershare PDFelement의 정상적인 설치 과정이 진행됩니다.

사용자의 화면에서는 프로그램이 정상적으로 설치되는 것처럼 보이지만, 그 이면에서는 정상 프로세스에 악성코드를 삽입하는 프로세스 인젝션, 암호화된 페이로드의 복호화, 외부 공격자 서버와의 통신 채널 수립 등 일련의 악성 행위가 이미 완료된 상태입니다.

이처럼 정상 설치 과정과 악성 행위를 교묘하게 결합하여 사용자가 이상을 인지하기 극히 어렵도록 설계되어 있습니다.

4-2. C2 복호화 로직 심층 분석

셸코드의 전체 실행 흐름은 12개 스테이지로 구분되며, 각 스테이지는 고유한 오프셋 범위 내에서 독립적인 기능을 수행합니다.

Stage 1에서 변조된 Entry Point로 진입한 셸코드는 Stage 2~3에서 실행 환경을 구성하고 대상 경로를 동적으로 생성합니다. Stage 4~7에서 프로세스 생성, 메모리 할당, 페이로드 복호화 및 원격 주입, 스레드 실행에 이르는 핵심 공격 체인을 완성합니다.

Stage 8에서 원래의 Entry Point로 복귀하여 정상 설치 프로세스를 진행함으로써 사용자의 인지를 회피하며, Stage 9~10은 원격 스레드 내에서 C2 서버와의 HTTP 통신 및 2차 페이로드 다운로드를 수행합니다.

Stage 11~12는 셸코드 전반에 걸쳐 반복 호출되는 지원 엔진으로, PEB 기반 API 해시 해석과 위치 독립적 주소 산출 기능을 제공합니다. 아래 표는 각 스테이지의 오프셋 범위, 기능명, 핵심 동작을 요약한 것입니다.

Stage	Offset	Function	Key Operations
-------	--------	----------	----------------

1	0x0000-0x000A	Entry Point Redirect	CALL \$+5로 셸코드 진입, JMP 0x00114103으로 OEP 복귀 경로 확보
2	0x000A-0x0016	Shellcode Prologue	스택 프레임 생성, 1,424바이트(0x590) 로컬 변수 공간 확보
3	0x0016-0x005C	Path String Construction	4바이트 단위 MOV DWORD PTR 명령 7회로 %windir%\System32\dism.exe 경로를 스택에 동적 구성
4	0x005C-0x00E3	API Resolution & CreateProcessA	PEB 해시 기반 API 해석, CREATE_SUSPENDED(0x04) 플래그로 dism.exe 프로세스 생성
5	0x00E3-0x011E	VirtualAllocEx	원격 프로세스 메모리 공간에 PAGE_EXECUTE_READWRITE(0x40) 권한으로 실행 가능 영역 할당
6	0x011E-0x01E1	XOR Decrypt & WriteProcessMemory	get-EIP 스텝으로 블록 주소 획득, 키 0x6D로 XOR 복호화, WriteProcessMemory로 원격 프로세스에 페이로드 및 C2 URL 기록
7	0x01E1-0x0224	CreateRemoteThread	dism.exe 프로세스 내 원격 스레드 생성, 삽입된 악성 코드 실행 개시
8	0x0224-0x0228	Epilogue & Return to OEP	레지스터 복원, RET → JMP 0x00114103으로 정상 설치 프로세스 복귀
9	0x0229-0x02D1	wininet.dll Loader	wininet.dll 동적 로드, 10MB(0xA00000) 다운로드 버퍼 할당, 2차 페이로드 수신 후 0x55 0x8B 시그니처 검증 및 메모리 직접 실행
10	0x02D2-0x03E4	HTTP C2 Download	InternetOpenA → InternetOpenUrlA → HTTP 200 검증 → InternetReadFile 루프로 데이터 수신

11	0x03E5- 0x054F	PEB Walker / API Hash Resolver	FS:[0x30] → PEB → Ldr → InLoadOrderModuleList 순회, ROR-13 해싱으로 DLL+함수 결합 해시 비교, API 절대 주소 반환
12	0x0550- 0x055E	get-EIP Stub	CALL/MOV EAX,[ESP]/ADD EAX,0xA 패턴으로 현재 실행 위치 기반 암호화 데이터 Blob(0x055F) 주소 동적 산출 - Blob (Binary Large Object)

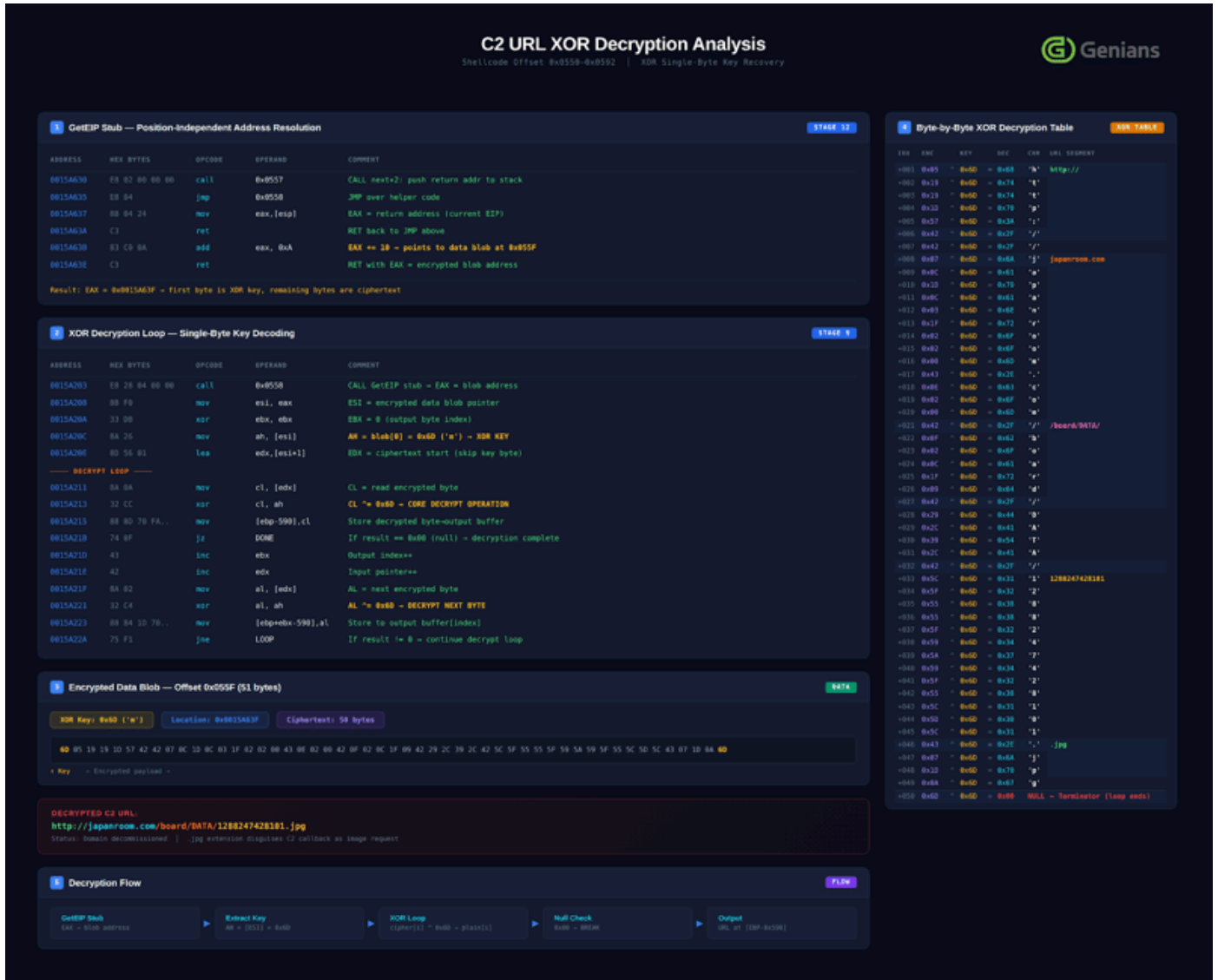
[표 4-1] 셸코드 전체 실행 흐름 요약표

셸코드 내부에 내장된 C2 URL은 단일 바이트 XOR 암호화로 보호되어 있으며, 런타임에 복호화되어 사용됩니다.

아래 인포그래픽은 이 복호화 과정을 5개 섹션으로 구분하여 설명합니다.

첫 번째 섹션에서는 get-EIP Stub(오프셋 0x0550)의 어셈블리 구조를 보여줍니다.

CALL/MOV EAX,[ESP]/ADD EAX,0xA 명령 패턴을 통해 현재 실행 위치를 동적으로 산출하고, Stub으로부터 정확히 10바이트 뒤에 위치한 암호화 데이터 Blob(0x055F)의 절대 주소를 EAX 레지스터에 반환합니다.



[그림 4-4] C2 URL XOR 복호화 과정 분석

두 번째 섹션에서는 XOR 복호화 루프의 어셈블리 코드를 상세히 기술합니다.

Blob의 첫 번째 바이트(0x6D, ASCII 'm')를 XOR 키로 추출한 뒤, 두 번째 바이트부터 순차적으로 각 바이트를 0x6D와 XOR 연산하여 평문으로 변환합니다. 연산 결과가 0x00(널 바이트)이 되면 루프가 종료됩니다.

세 번째 섹션에서는 암호화 데이터 Blob의 원본 구조를 제시합니다.

오프셋 0x055F에 위치한 총 51바이트 중 첫 1바이트가 XOR 키, 이후 49바이트가 암호화된 URL 페이로드이며, 마지막 1바이트(0x6D)는 키와 동일한 값으로 XOR 연산 시에 결과값이 0x00이 되어 별도의 길이 정보 없이 복호화 루프의 자연스러운 종료를 보장합니다.

네 번째 섹션에서는 50바이트 전체의 바이트 단위 XOR 복호화 테이블을 제공합니다.

암호화 값, XOR 키, 복호화 결과, ASCII 문자, URL 세그먼트 구분이 행 단위로 나열되어 있으며, 프로토콜(http://), 도메인(japanroom[.]com), 경로(/board/DATA/), 파일명(1288247428101), 확장자(.jpg)가 순차적으로 복원되는 과정을 확인할 수 있습니다.

다섯 번째 섹션에서는 최종 복호화된 C2 URL 주소를 보여줍니다.

이 URL은 .jpg 확장자를 사용하여 정상적인 이미지 요청으로 위장함으로써 네트워크 모니터링 및 URL 필터링을 우회하도록 설계되어 있습니다.

`http://japanroom[.]com/board/DATA/1288247428101.jpg`

4-3. 2차 페이로드 분석

C2 서버로부터 1288247428101.jpg 파일의 다운로드가 정상적으로 완료되면, 셸코드는 수신된 데이터를 이미지가 아닌 암호화된 2차 페이로드로 처리합니다.

셸코드는 수신 데이터에 대해 즉시 2차 XOR 복호화를 수행합니다.

수신 데이터의 첫 번째 바이트를 XOR 키로 사용하여 두 번째 바이트부터 마지막 바이트까지 순차적으로 XOR 연산을 적용하는 방식이며, 이는 C2 URL 복호화에 사용된 것과 동일한 단일 바이트 XOR 기법입니다.

실제 파일의 첫 번째 바이트는 0x6F이며, 이 값이 나머지를 복호화하는 키로 사용됩니다. 복호화가 완료되면, 셸코드는 복호화된 데이터가 유효한 실행 코드인지 검증합니다.

구체적으로 복호화된 데이터의 첫 번째 바이트가 0x55(PUSH EBP 명령)인지, 두 번째 바이트가 0x8B(MOV 명령의 첫 바이트)인지를 확인합니다. 이 두 바이트는 x86 함수의 표준 프롤로그인 55 8B EC(PUSH EBP / MOV EBP, ESP)의 시작부에 해당합니다.

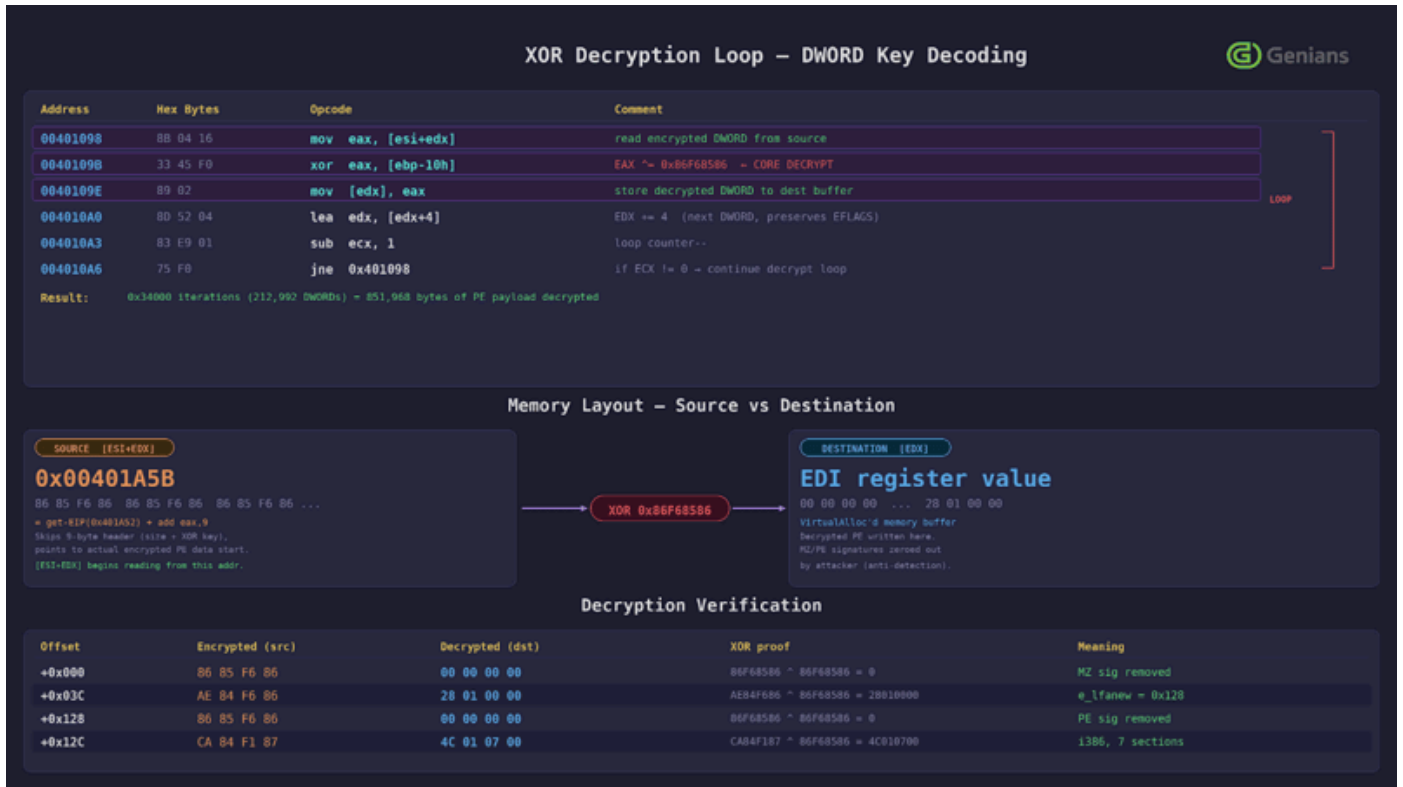
시그니처 검증에 실패하면 셸코드는 Sleep으로 5초간 대기한 뒤 C2 서버에 대한 다운로드를 처음부터 재시도합니다.

이 재시도 루프는 무한 반복으로 설계되어 있어, C2 서버의 일시적 장애나 전송 오류 시에도 유효한 페이로드를 수신할 때까지 지속적으로 접속을 시도합니다.

시그니처 검증에 성공하면, LEA EAX, [ESI+1] 명령으로 XOR 키 바이트를 건너뛴 실제 셸코드의 시작 주소를 산출한 뒤, CALL EAX 명령으로 복호화된 2차 셸코드를 메모리 상에서 직접 실행합니다.

이 전체 과정에서 2차 페이로드는 디스크에 어떠한 파일로도 저장되지 않으며, 메모리에서 복호화된 후 즉시 실행되는 파일리스(Fileless) 방식으로 동작합니다.

또한 .jpg 확장자를 가진 URL을 통해 정상적인 이미지 요청으로 위장하고, XOR 암호화된 전송 데이터와 메모리 전용 실행 방식의 조합을 통해 네트워크 탐지, 파일 기반 탐지, 정적 분석을 동시에 우회하도록 설계되어 있습니다.



[그림 4-5] 2차 페이로드 XOR 복호화 루프 분석

0x6F XOR 키로 복호화된 셸코드는 먼저 자신이 메모리 어디에 위치하는지 파악하는 과정을 수행합니다. 이를 위해 get-EIP 가젯(0x401A47)이라는 기법을 사용합니다. 일반적인 프로그램은 자신이 메모리 어느 주소에 로드되었는지 쉽게 알 수 있지만, 셸코드는 매번 다른 위치에 로드될 수 있기 때문에 자신의 현재 주소를 스스로 알아내야 합니다.

get-EIP 가젯은 x86 프로세서의 call 명령이 실행될 때 다음 명령의 주소를 자동으로 스택에 저장하는 특성을 이용합니다. 셸코드는 의도적으로 바로 옆 주소를 call로 호출한 뒤, 스택에 쌓인 복귀 주소를 꺼내오는 방식으로 현재 위치(0x401A52)를 알아냅니다.

이 위치에는 9바이트짜리 작은 정보 블록이 존재합니다. 여기에는 암호화된 데이터의 전체 0xD0000(851,968 바이트) 크기와 암호를 풀기 위한 키 값(0x86F68586)이 기록되어 있습니다. 셸코드는 이 두 값을 읽어낸 뒤 9바이트를 건너뛰어, 실제 암호화된 데이터가 시작되는 주소 0x401A5B를 확정합니다.

다음으로 셸코드는 Windows 운영체제에 메모리 공간 할당을 요청합니다. VirtualAlloc 함수를 호출하여 0xD0400 크기의 빈 메모리 영역을 확보하며, 이 공간이 암호를 풀어낸 결과물이 저장될 장소가 됩니다. 이 주소는 EDI 레지스터에 보관됩니다.

이제 실제 암호 해제 작업이 시작됩니다. 셸코드는 데이터를 4바이트(DWORD) 단위로 처리하며, 전체 바이트를 복호화합니다. 원본 위치(0x401A5B)에서 암호화된 4바이트를 꺼내고, 이 값을 키(0x86F68586)와 XOR 연산하여 원래 데이터로 되돌린 뒤, 새로 할당한 메모리(EDI)에 저장합니다.

복호화가 완료되면, 새 메모리에는 완전한 실행 파일(PE)이 복원됩니다. 다만 위협 행위자는 탐지를 피하기 위해 실행 파일의 대표적인 식별 표시인 MZ 서명과 PE 서명을 의도적으로 지워둔 상태입니다. 그러나 파일 내부 구조 정보(섹션 테이블, импорт 테이블 등)는 정상적으로 복원되므로, 셸코드에 내장된 자체 로더가 이를 읽어 메모리에 배치하고 최종 악성코드를 실행합니다.

컴파일 타임스탬프가 0으로 제거되어 있어 제작 시점의 추적을 의도적으로 방해하고 있습니다. 이 페이로드의 가장 주목할 특징은 합법적인 클라우드 서비스를 명령제어(C2) 채널로 악용한다는 점입니다. Zoho WorkDrive의 OAuth2 API를 통해 공격자의 명령을 수신하고 탈취한 데이터를 유출하며, 이를 위한 OAuth 인증 토큰 2세트가 바이너리 내부에 하드코딩되어 있습니다.

이러한 합법적 클라우드 서비스 기반 통신은 일반적인 업무용 트래픽과 구분하기 어려워 네트워크 보안 장비의 탐지를 효과적으로 우회합니다. 페이로드가 수행하는 주요 악성 기능은 다음과 같습니다.

- 스크린샷 캡처 기능으로 GetDC와 GetSystemMetrics를 통해 바탕화면의 해상도를 파악하고, CreateCompatibleBitmap과 BitBlt로 화면 전체를 복사한 후 GDI+ 라이브러리의 JPEG 인코더로 변환하여 C2 서버로 전송합니다.
- cmd.exe /c "%s" 형식의 원격 명령 실행 기능을 갖추고 있어 C2에서 전달되는 임의의 시스템 명령을 실행할 수 있습니다.
- GetComputerNameW와 GetUserNameW로 호스트 및 사용자 정보를 수집하고, GetLogicalDrives와 GetDiskFreeSpaceExW로 디스크 구성을 열거합니다.
- 외부 서비스 ipinfo[.]io/json을 통해 피해 시스템의 공인 IP 및 지리적 위치를 확인하는 시스템 정찰 기능을 수행합니다.
- CryptoPP 라이브러리를 활용하여 AES-CBC 대칭 암호화와 RSA 공개키 암호화를 지원합니다.

보안 제품 회피와 관련하여, CreateToolhelp32Snapshot과 Process32NextW를 통해 실행 중인 프로세스 목록을 열거하고 360Tray.exe(Qihoo 360 보안 제품)의 존재 여부를 확인합니다.

그리고 명령에 따라 OfficeUpdate.exe라는 정상 소프트웨어 업데이트 프로그램으로 위장한 파일을 AppData 경로에 배치합니다.

또한 21개의 다양한 User-Agent 문자열을 내장하여 네트워크 트래픽을 정상적인 웹 브라우저 활동으로 위장합니다.

Client_id	1000.PZCKAWFHJPS9A6WU5G*****
	1000.LS8YMLB2Z0BSHSL3VW*****
Refresh_Token	1000.c637b3e7d74c2d678663454d16311b15.12b060c2ad23ecafc959*****
	1000.085128b4e96633c82beb2101f5c525e4.6ee7ce2c1952f14794d0*****

Client_Secret a9e22f09e5e54c7ee7b3e9ac4340d7*****

4cb82c2055249110fc86893de90a2d*****

[표 4-2] Zoho WorkDrive 인증 정보

감염된 단말에서 컴퓨터 이름, 사용자 이름, Windows 버전, IP 주소 및 지리적 위치(국가, 도시), SMBIOS 하드웨어 식별 정보, 실행 중인 프로세스 목록, 화면 스크린샷(JPEG), 그리고 특정 확장자(.DOC, .XLS, .PPT, .PDF, .HWP, .TXT, .M4A, .AMR)의 문서 및 스마트폰 녹음 파일을 수집하여 Zoho WorkDrive 클라우드 저장소로 유출합니다.

수집된 데이터를 전송하기 전에 AES-256-CBC 암호화가 적용됩니다.

더불어 바이너리 내부에 하드코딩된 **#FBI#TOOLKIT#GIDRA@TEAM**, **@-IV-FBI-SERVER2** 문자열은 HTTP GET 요청이 성공하면, 응답 데이터에 대해 AES 복호화에 사용됩니다.

0AC840	72 00 6D 00 61 00 6C 00 00 00 00 00 2E 00 58 00	r.m.a.l.....X.
0AC850	4C 00 53 00 00 00 00 00 2E 00 44 00 4F 00 43 00	L.S.....D.O.C.
0AC860	00 00 00 00 2E 00 50 00 50 00 54 00 00 00 00 00	P.P.T.....
0AC870	2E 00 54 00 58 00 54 00 00 00 00 00 2E 00 4D 00	..T.X.T.....M.
0AC880	34 00 41 00 00 00 00 00 2E 00 41 00 4D 00 52 00	4.A.....A.M.R.
0AC890	00 00 00 00 2E 00 50 00 44 00 46 00 00 00 00 00	P.D.F.....
0AC8A0	2E 00 48 00 57 00 50 00 00 00 00 00 41 00 6C 00	..H.W.P.....A.l.
0AC8B0	6C 00 00 00 47 00 45 00 54 00 00 00 68 00 74 00	l...G.E.T...h.t.
0AC8C0	74 00 70 00 3A 00 2F 00 2F 00 31 00 32 00 37 00	t.p.:././1.2.7.
0AC8D0	2E 00 30 00 2E 00 30 00 2E 00 31 00 00 00 00 00	..0...0...1.....
0AC8E0	2F 00 00 00 4F 00 66 00 66 00 69 00 63 00 65 00	/...0.f.f.i.c.e.
0AC8F0	55 00 70 00 64 00 61 00 74 00 65 00 2E 00 65 00	U.p.d.a.t.e...e.
0AC900	78 00 65 00 00 00 00 00 2F 00 6A 00 73 00 6F 00	x.e...../j.s.o.
0AC910	6E 00 00 00 69 00 70 00 69 00 6E 00 66 00 6F 00	n...i.p.i.n.f.o.
0AC920	2E 00 69 00 6F 00 00 00 63 6F 75 6E 74 72 79 00	..i.o...country.
0AC930	63 69 74 79 00 00 00 00 69 70 00 00 00 00 00 00	city....ip.....
0AC940	00 00 00 00 5F 04 26 0E 06 07 08 82 68 83 A5 AE	&.....h...
0AC950	3A 4B 8A 6A 23 46 42 49 23 54 4F 4F 4C 4B 49 54	:K.j#FBI#TOOLKIT
0AC960	23 47 49 44 52 41 40 54 45 41 4D 00 40 2D 49 56	#GIDRA@TEAM.@-IV
0AC970	2D 46 42 49 2D 53 45 52 56 45 52 32 11 00 00 00	-FBI-SERVER2....
0AC980	00 00 00 00 00 00 00 80 FF FF FF FF FF FF 7F	

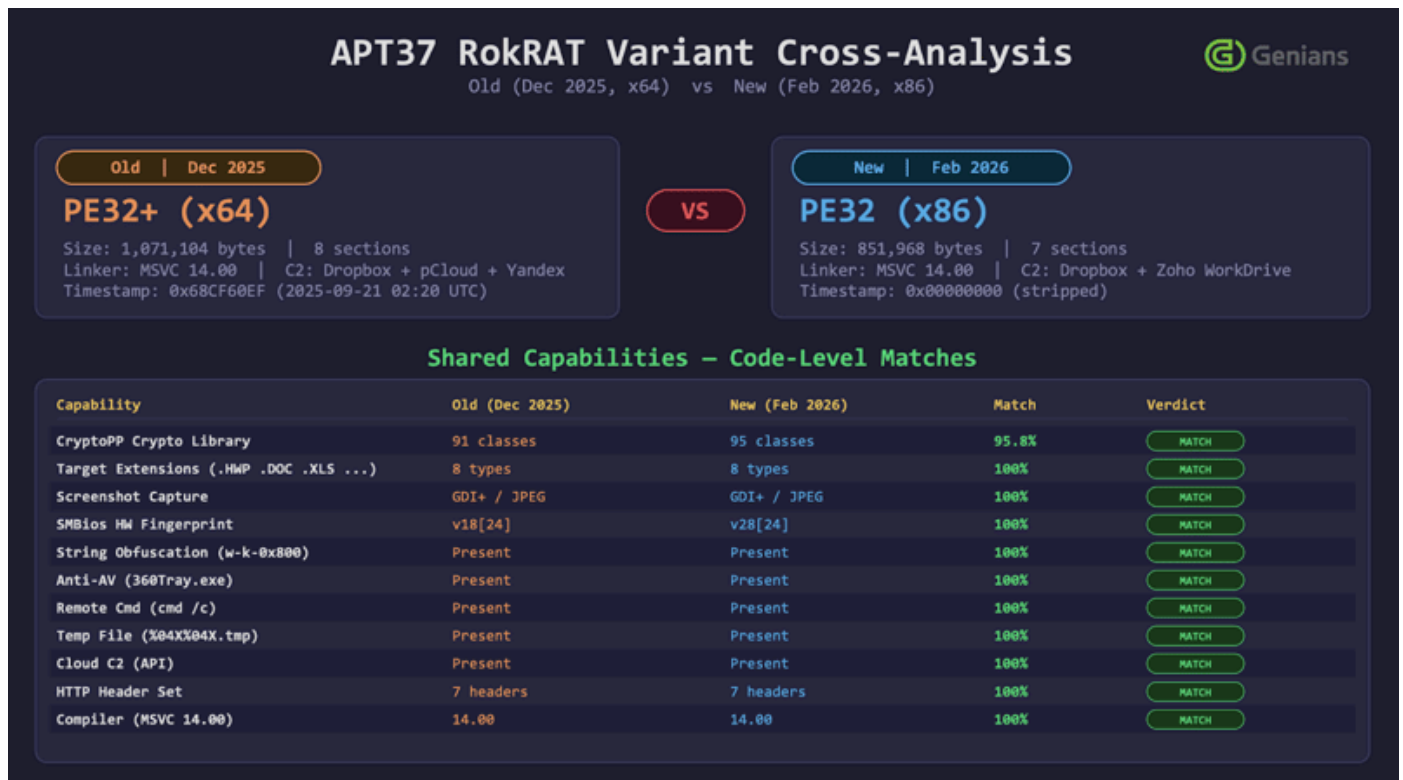
[그림 4-6] 바이너리 내부에 포함된 문자열

5. 위협 귀속 (Threat Attribution)

5-1. APT37 RokRAT 비교

지니언스 시큐리티 센터(GSC)는 다양한 사이버 위협 인텔리전스(CTI) 분석 보고서를 통해 APT37 캠페인의 활동을 지속적으로 추적하며, 해당 위협 행위자가 수행해 온 주요 사이버 작전의 특징과 사례를 축적해 왔습니다.

이러한 분석 결과는 단일 공격 사례를 넘어 장기간에 걸쳐 관측된 캠페인 단위의 활동을 이해하는데 중요한 참고 자료로 활용되고 있습니다.



[그림 5-1] 2025년 12월 RokRAT 변종과 비교 화면

APT37의 RoKRAT은 시스템 정보 수집, 명령 실행, 스크린샷 탈취 등 정찰과 정보수집에 초점을 둔 백도어로, 본체 기능은 비교적 안정적으로 유지하면서 장기간 여러 작전에서 반복 재사용되어 왔습니다. 지난해 12월 발견된 변종과 비교해 본 결과, 코드 레벨 유사도가 매우 높다는 것을 알 수 있습니다.

초기 침투 단계에서는 LNK 바로가기 파일과 HWP 악성 문서를 주요 경로로 활용하며, 스피어 피싱을 통해 대학교수·방송작가·싱크탱크 행사 등 공신력 있는 명의를 사칭하여 표적의 신뢰를 확보한 뒤 최종 페이로드를 파일리스 형태로 메모리에서만 실행하는 방식을 일관되게 유지하고 있습니다.

그러나 이번에는 특정 문서 뷰어 설치용 프로그램을 정교하게 조작하는 방식을 사용했고, Zoho WorkDrive 클라우드를 적극적으로 사용하였습니다.

특히 배포 및 탐지 회피 기법은 시간이 지남에 따라 뚜렷하게 고도화되고 있습니다. 초기 PowerShell 기반 셸코드 로딩에서 Sysinternals 정상 유틸리티를 악용한 DLL 사이드 로딩, JPEG 스테가노그래피를 통한 페이로드 은닉, 그리고 최대 3중 레이어에 이르는 다단계 XOR 복호화 체인으로 진화하면서, 정상 문서와 프로세스 뒤에 악성 행위를 은밀히 숨기는 운영 방식이 더욱 정교해지고 있습니다.

이는 RoKRAT이 기능 변화보다 전달·실행·회피 체인의 진화에 초점을 맞추고 있음을 보여줍니다.

C2 인프라 역시 Dropbox, pCloud, Yandex Cloud, Zoho WorkDrive 등 합법적 클라우드 서비스의 API를 악용하여 정상 트래픽으로 위장하는 전략을 지속하고 있으며, 동일 계정 ID가 서로 다른 클라우드에 같은 날짜에 가입된 정황이 확인되어 공격자가 다중 클라우드 인프라를 체계적으로 통합 운영하고 있음을 시사합니다.

따라서 본 위협 활동은 충분한 기술적 연계 근거를 바탕으로 APT37 위협 그룹의 소행으로 판단됩니다.

5-2. Facebook 및 Zoho 활용 정보

위협 행위자가 Facebook에서 사용한 계정 **richardmichael0828** 및 **johnsonsophia0414**는 모두 2025년 11월 10일에 동일하게 생성된 것으로 확인되었습니다. 이러한 계정 생성 시점의 일치는 동일 행위자 또는 동일 조직에 의해 운영되었을 가능성을 시사합니다.

네트워크 인프라 측면에서는, 해당 행위자가 주로 Astrill VPN을 활용하고 있으며, 공통적으로 미국 소재 IP 주소인 **38.32.68[.]195**가 반복적으로 관찰되고 있습니다.

해당 IP는 VPN 서비스 대역에 속한다는 점에서 중요 식별 정보로 활용하기에는 제한이 있으나, Recorded Future 및 Silent Push의 분석 보고서(["PurpleBravo's Targeting of the IT Software Supply Chain"](#), ["Silent Push Pivots into New Lazarus Group Infrastructure, Acquires Sensitive Intel Related to \\$1.4B ByBit Hack and Past Attacks"](#))를 통해 동일 IP와 연관된 과거 공격 인프라 이력이 확인된 바 있습니다. 이는 해당 인프라가 특정 위협 그룹과 연계되어 재사용되었을 가능성을 뒷받침합니다.

초기 접근 단계에서는, 위협 행위자가 **북한 평성시 및 평양시 출생**으로 기록된 Facebook 계정을 활용하여 공격 대상자에게 접근하는 양상이 확인되었습니다. 이후 신뢰 관계를 형성한 뒤, Telegram, 이메일 등 외부 채널로 유도하여 악성 파일을 전달하는 전형적인 소셜 엔지니어링 기반 공격 흐름을 보이고 있습니다.

한편, APT37은 과거 2017년 **scott.snyder@zoho.com** 계정을 활용한 스피어 피싱 공격이 관찰된 바 있으며, 이후에도 Zoho 서비스가 지속적으로 악용된 사례가 다수 확인되고 있습니다.

특히 2025년 하반기에는 **leon91729@zoho.com** 계정이 추가로 확인되었습니다. 같은 시기, Zoho WorkDrive를 C2 인프라로 활용한 악성코드와 북한 글씨체인 '**천리마체**'로 작성된 미끼 문서가 함께 발견되었으며, 이 과정에서 **kingtiger1970** 계정이 가입에 사용된 정황도 확인되었습니다.

더불어 C2 통신 과정에서 "**JinHyok**"이라는 디버깅 문자열이 식별되었으며, 이는 북한 연계 위협 그룹에서 관찰된 식별 요소로서 공격 주체의 귀속(Attribution)에 있어 중요한 단서로 작용할 수 있습니다.

6. 결론 및 대응 (Conclusion)

본 위협은 Facebook 기반 표적 정찰을 통해 공격 대상자와의 신뢰 관계를 형성한 이후, Telegram 등 외부 메신저로 전환하여 악성 파일 실행을 유도하는 다단계 소셜 엔지니어링 공격으로 확인되었습니다.

위협 행위자는 암호화된 군사 문서를 열람하기 위한 전용 프로그램 설치를 요구하는 방식으로 사용자의 경계심을 낮춘 뒤, 정상 소프트웨어로 위장한 변조된 Wondershare PDFelement 설치 파일을 실행하도록 유도하였습니다. 해당 실행 파일은 내부에 삽입된 shellcode를 통해 초기 침투를 수행하며, 이후 추가 페이로드를 단계적으로 실행하는 구조를 갖습니다.

특히 일본 부동산 정보 사이트의 서울지점 웹사이트를 C2 인프라로 악용하고, JPG 이미지 파일로 위장된 페이로드를 통해 후속 명령을 전달하는 방식은 정상 트래픽과 구분을 어렵게 만드는 고도의 은닉 기법으로 확인되었습니다.

또한 본 캠페인은 소셜 네트워크 기반 초기 접근, 프리텍스팅 기반 악성코드 실행 유도, shellcode 기반 초기 침투, 이미지 위장 페이로드를 활용한 후속 명령 수행으로 이어지는 다단계 공격 구조를 보이며, 이는 APT37의 기존 공격 전술과 높은 유사성을 갖습니다. 특히 북한식 언어적/지역적 특징과 공격 흐름, 인프라 운용 방식 등을 종합할 때, 동일 계열 또는 연계된 위협 행위자에 의해 수행되었을 가능성이 높은 것으로 판단됩니다.

이러한 특성을 고려할 때, 본 위협은 확보된 IoC 기반 대응과 함께, ▲SNS 기반 접근 및 관계 형성 행위 ▲정상 소프트웨어로 위장된 실행 파일 동작 ▲shellcode 실행 및 후속 페이로드 적재 ▲정상 웹사이트를 경유한 C2 통신 ▲이미지 파일 기반 은닉 페이로드 실행 등 공격 전 과정에 대한 행위 기반 분석이 요구됩니다.

따라서 조직은 초기 침투 단계부터 실행, 지속성 확보, C2 통신, 후속 명령 수행까지 위협 행위자의 전술·기법·절차(TTP)를 통합적으로 탐지할 수 있는 EDR 중심의 대응 체계를 구축해야 합니다. 특히 정상 프로세스 위장 및 합법 인프라 악용과 같은 고도화된 회피 기법에 대응하기 위해, 단일 이벤트가 아닌 행위 간 상관관계를 기반으로 한 탐지 및 위협 헌팅 체계의 병행이 필수적입니다.

6-1. 소셜 네트워크 기반 초기 접근 대응

본 위협은 Facebook 등 SNS를 활용한 관계 형성 이후 공격이 수행되는 구조를 보이므로, 기존 이메일 중심 보안 정책만으로는 초기 침투를 효과적으로 차단하기 어렵습니다.

특히 북한, 군사, 기술 협력, 고수익 거래 등 특정 주제를 기반으로 접근하여 신뢰를 형성한 뒤 악성 파일 실행을 유도하는 소셜 엔지니어링 기법이 활용된 점에서, 사용자 행동이 보안 수준에 직접적인 영향을 미치는 것으로 분석됩니다.

이에 따라 조직은 SNS 및 메신저를 통한 비업무적 접촉과 파일 전달 행위에 대한 보안 가이드를 마련하고, 외부 파일 수신 시 출처 검증 절차를 명확히 하는 한편, 업무와 무관한 파일 실행을 제한하는 정책을 수립해야 합니다.

또한 암호화된 압축파일과 별도 비밀번호 전달 방식, 정상 소프트웨어 설치를 요구하는 외부 요청 등 공격에 활용되는 주요 기법에 대해 사용자 교육을 강화할 필요가 있습니다.

결과적으로 본 위협은 기술적 취약점보다는 사용자 신뢰를 악용하는 공격이라는 점에서, 단순 악성코드 탐지를 넘어 사용자 인식 기반의 대응 체계가 필수적입니다.

특히 정상으로 보이는 요소를 악용하는 공격이 증가하는 환경에서는, 사용자 판단 기반의 보안 인식이 조직 전체 보안 수준을 좌우하는 핵심 방어 요소로 작용합니다.

6-2. 변조된 정상 소프트웨어 실행 탐지 대응

본 공격에서는 정상 소프트웨어로 위장한 변조된 Wondershare PDFelement 설치 파일이 활용되었습니다.

이는 단순 파일 해시 기반 탐지나 평판 기반 차단을 우회할 수 있는 대표적인 기법입니다.

이에 따라 EDR에 기록된 이벤트 로그를 통해 다음과 같은 위협 행위를 중점적으로 헌팅해야 합니다.

- 정상 소프트웨어 실행 직후 비정상 프로세스 생성
- 설치 프로그램 실행 이후 PowerShell, CMD 등 스크립트 인터프리터 호출
- 사용자 실행 컨텍스트와 무관한 외부 네트워크 통신 발생
- 정상 설치 흐름과 다른 프로세스 트리 구조

특히 "정상 프로그램 → 비정상 행위"로 이어지는 행위 체인을 기반으로 탐지 로그를 분석함으로써, 변조된 설치 파일 기반 공격을 효과적으로 식별할 수 있습니다.

6-3. 위협 헌팅 및 지속적 모니터링 체계

APT37과 같은 지능형 위협 그룹은 장기 잠복과 은닉을 전제로 공격을 수행하는 특징을 가지므로, 단발성 탐지 중심의 대응으로는 한계가 존재합니다.

이에 따라 조직은 위협 헌팅 기반의 능동적인 탐지 체계를 운영하고, 지속적인 모니터링을 통해 잠재적 침해 징후를 식별할 수 있어야 합니다.

특히 SNS 기반 접근 이후 발생한 비정상 실행 이력, 정상 소프트웨어 실행 이후의 행위 변화, 일정 기간 내 반복되는 유사 프로세스 패턴, 외부 통신 이력 등을 종합적으로 분석함으로써 숨겨진 위협을 식별할 필요가 있습니다.

또한 과거 로그 데이터를 기반으로 한 사후 분석과 함께, 장기적인 행위 추적을 통해 초기 침투 시점을 역추적할 수 있는 체계도 병행되어야 합니다.

이러한 위협 헌팅 및 지속적 모니터링 체계는 고도화된 APT 공격에 대한 탐지 공백을 최소화하는데 중요한 역할을 합니다.

6-4. C2 통신 및 이미지 위장 페이로드 대응

위협 행위자는 정상 웹사이트를 경유한 C2 통신과 JPG 파일로 위장된 페이로드를 활용하여 탐지 회피를 시도하였습니다. 이는 네트워크 기반 탐지를 어렵게 만드는 대표적인 기법입니다.

이에 따라 다음과 같은 탐지 전략이 요구됩니다.

- 정상 웹사이트 접속 이후 비정상 데이터 송수신 패턴 식별
- 이미지 파일 요청 대비 비정상적인 응답 크기 또는 구조 분석
- 사용자 행위와 무관한 주기적 외부 통신 탐지
- 네트워크 트래픽 내 비정상 인코딩 또는 데이터 은닉 패턴 식별

특히 단순 도메인 차단이 아닌, "정상 서비스 내 비정상 행위"를 식별할 수 있는 네트워크 행위 기반 분석이 중요하고, Zoho WorkDrive 접속 시도 여부를 파악해 보는 것도 중요합니다.

6-5. 'Genian Insights E' 기반 통합 대응 전략

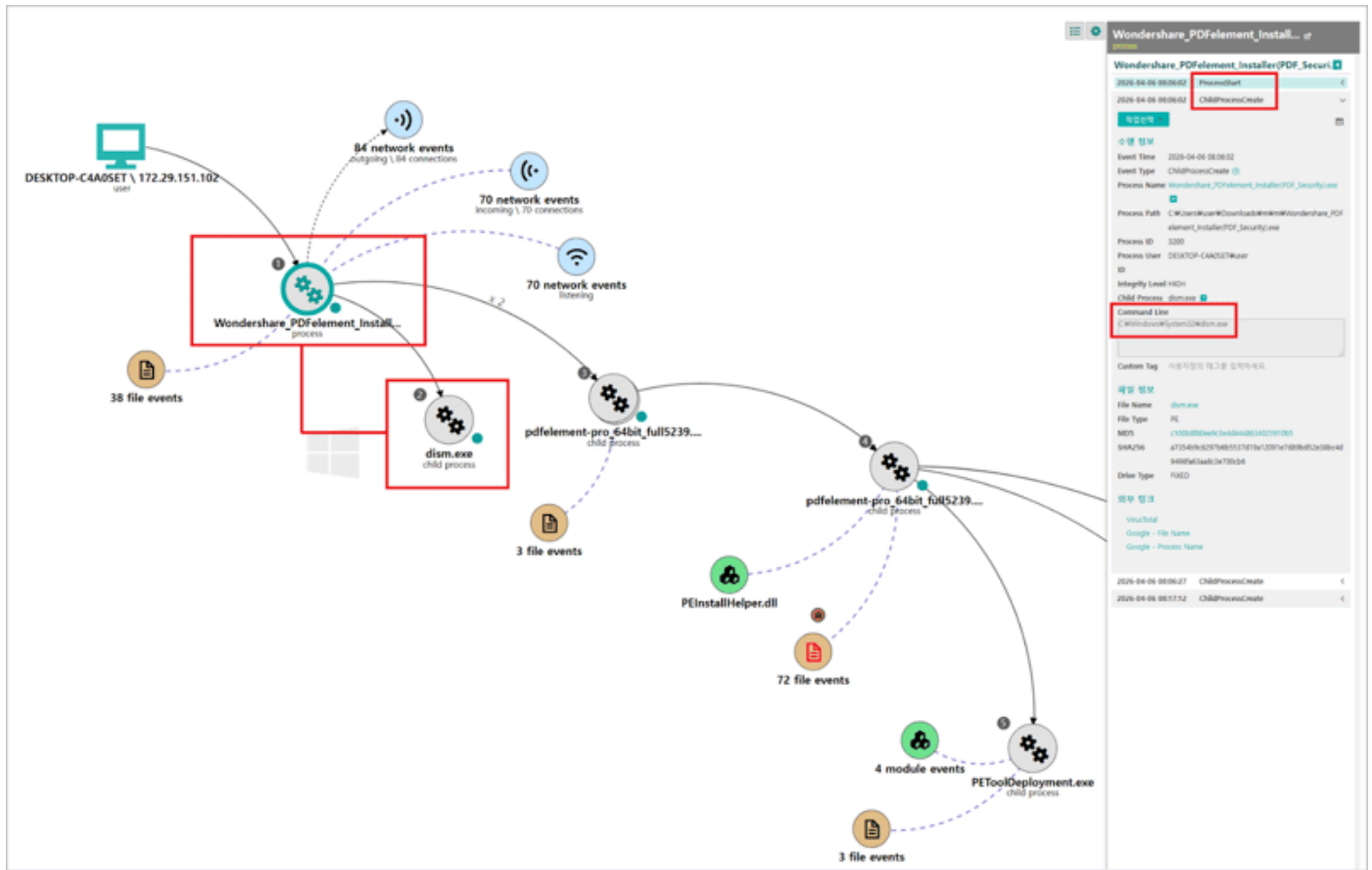
본 위협은 소셜 엔지니어링, 정상 소프트웨어 위장, 메모리 기반 실행, 합법 인프라 악용이 결합된 복합 공격으로 확인되었으며, 단일 보안 솔루션만으로는 효과적인 대응이 어렵습니다. 따라서 EDR을 중심으로 초기 실행부터 후속 행위까지 전체 공격 흐름을 가시화하고, 프로세스 트리 기반의 이상 행위 탐지 및 이벤트 간 상관 분석을 수행할 수 있는 통합 대응 체계가 요구됩니다.

'[Genian Insights E](#)'는 다양한 위협 대응을 위해 단일 에이전트 기반으로 구성된 통합 엔드포인트 보안 플랫폼입니다.

실시간 행위 기반 탐지 및 위협 대응을 수행하는 **EDR**(Endpoint Detection and Response), 시그니처 기반으로 악성코드를 탐지·치료하는 **Anti-Virus**, 파일 암호화 행위의 실시간 차단과 중요 문서의 자동 백업·복원을 지원하는 **Anti-Ransom** 기능을 제공합니다. 또한 USB·외장하드 등 저장매체 사용을 통제하여 내부 정보 유출을 방지하는 **Device Control** 기능도 함께 제공합니다.

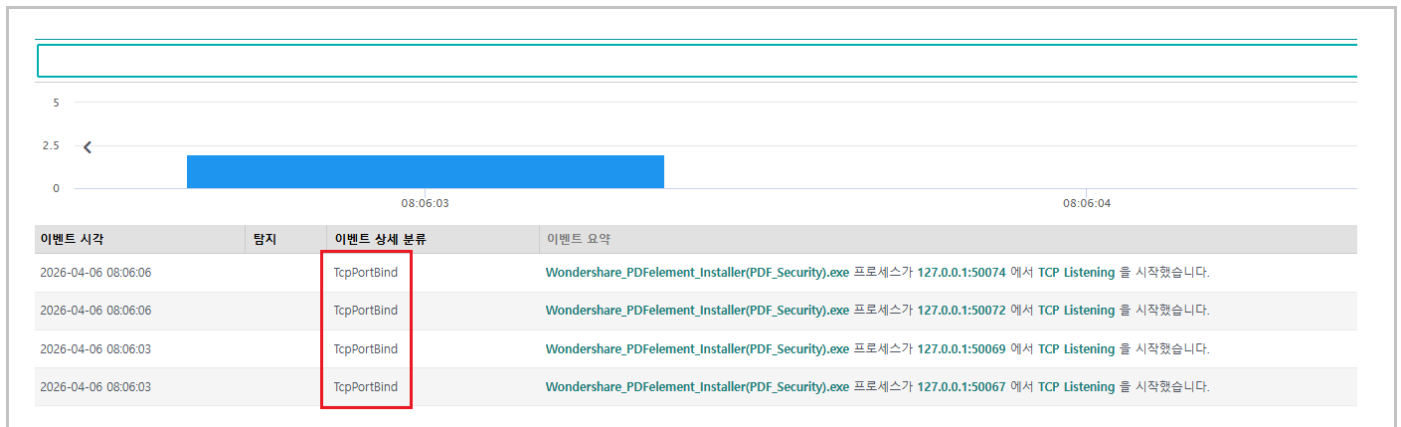
특히 파일, 프로세스, 네트워크 이벤트를 개별적으로 분석하는 수준을 넘어, 공격 흐름 전체를 하나의 맥락으로 연결하여 분석하는 것이 중요합니다.

이를 통해 의심 행위가 식별된 단말에 대해 신속한 격리 및 추가 확산 차단 조치를 수행할 수 있으며, IoC 기반 탐지와 행위 기반 탐지를 병행함으로써 탐지 정확도를 높일 수 있습니다. 결과적으로 EDR은 본 공격과 같이 다단계로 전개되는 APT 위협에 대해 조기 탐지 및 선제 대응을 가능하게 하는 핵심 보안 체계로 활용될 수 있습니다.



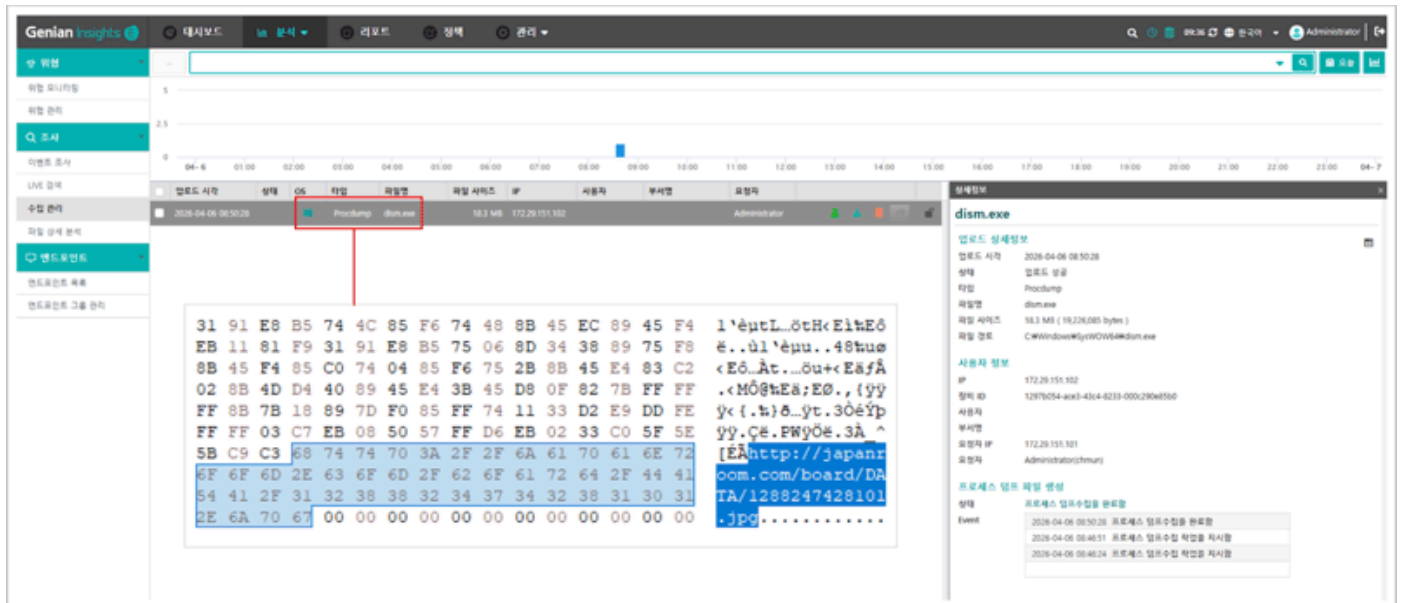
[그림 6-1] EDR 기반 초기 유입 흐름 파악

Genian Insights E의 Attack Storyline 기능을 활용하면, Wondershare_PDFelement_Installer(PDF_Security).exe 프로세스 시작시 자식 프로세스로 dism.exe 파일이 실행된 것을 관찰할 수 있습니다.



[그림 6-2] C2 통신 시도 화면

shellcode가 실행되면 네트워크 통신을 시도하지만, 분석 시점에는 C2 서버가 이미 차단된 상태여서 정상적인 접속은 이루어지지 않았습니다. 또한 악성코드는 TcpPortBind를 통해 지속적으로 TCP Listening을 수행하는 것을 볼 수 있습니다.



[그림 6-3] 프로세스 덤프를 통한 C2 추출

EDR 관리자는 dism.exe 프로세스를 덤프하여, Hex Editor로 내부에 은닉된 C2 주소를 추출할 수 있습니다.

7. IoC (Indicator of Compromise)

MD5

c681fe3f42e82e9240afe97c23971cbc

d44a22d2c969988a65c7d927e22364c8

28d0143718153bf04c1919a26bb70c2d

36be2cbb59cd1c3f745d5f80f9aee21c

C2

japanroom[.]com

38.32.68[.]195

222.122.49[.]15