# 악성코드 분석 보고서

**파일명** : CEaN.exe

**SHA-256** : 4a4d0da0f8c4cd9a46178150f755a1348100b2c5b471874f8a898258c39a26a4

**작성자** : 김윤성

**보고서 작성일** : 26.06.19

# 목차

# 1. 개요

## 1.1 파일(분석 샘플) 정보

**표 1. 분석 샘플 정보**

| **파일명** | **CEaN.exe** |
|---|---|
| 별칭 | JVKPZv.exe / RFQ_Purchase_Order_Quotation_Forms.exe / x1nhss.exe 등등 |
| 크기 | 1023.50 KB (1,048,064 bytes) |
| 악성코드 유형 | 트로이목마[1], 인포스틸러[2] |
| 패밀리 | AgentTesla |
| 파일 타입 | Win32 EXE / .NET[3](MSIL), VB.NET 컴파일, .NET v4.0.30319, 32bit, GUI, 미서명 |
| MD5 | `82c32ffb327a0a2d20050d6db05408f0` |
| SHA-1 | `76a487a95dd5aa6910fdb37aeafa824aa4edab26` |
| SHA-256 | `4a4d0da0f8c4cd9a46178150f755a1348100b2c5b471874f8a898258c39a26a4` |
| 타임스탬프 | 2026-05-13 02:40:40 UTC |
| 출처 | MalwareBazaar(User threatcat_ch) |
| 진단명(V3) | Trojan/Win.Phonzy.C5882775 |

## 1.2 분석 요약

본 보고서는 트로이목마·인포스틸러 유형의 악성코드 CEaN.exe(AgentTesla 계열, **VirusTotal 54/69**)를 VMware 기반 격리 환경에서 기초·정적·동적·네트워크 분석한 보고서이다.

CEaN.exe는 .NET 계열의 악성코드로 실행 시 자신을 `%APPDATA%\Roaming\JVKPZv.exe`로 **복사하고** `S·H` 속성으로 **은폐한다**. 이어 `HKCU\...\Run` 키에 PowerShell 실행 항목을 등록하고 .ps1 런처를 드롭해, 로그인 시마다 본체가 재실행되는 **지속성을 확보한다**(정상 도구를 악용하는 **LotL 기법**).

CEaN.exe는 브라우저(Chrome·Edge 등), VPN·FTP·메일 등의 **자격증명을 광범위하게 탐색**하고, **저장된 비밀번호와 Windows 자격증명까지 수집**한다.

유출에 앞서 ip-api[.]com으로 **분석 환경 여부를 정찰**한 뒤, 수집 정보를 onionmail[.]org로 **평문 SMTP를 통해 유출**한다. 메일 전송 완료 응답을 확인함으로써 **실제 데이터가 유출**되는 것을 확인했다.

분석 결과를 바탕으로 파일·레지스트리·네트워크 침해지표(IOC)[4]를 정리하고 Snort 탐지 룰을 작성·검증하였다. 코드 및 메모리 단위 분석은 본 분석 범위에서 제외하였다.

---

[1] 정상 프로그램인 척 위장해 사용자가 직접 실행하도록 속이는 악성코드.

[2] PC에 저장된 계정·비밀번호 등 정보를 몰래 훔쳐 빼내는 악성코드.

[3] .NET은 마이크로소프트의 개발·실행 환경, MSIL은 그 중간언어, VB.NET은 .NET 기반 언어다. .NET 악성코드는 코드가 숨겨져 분석이 까다롭다.

[4] 침해지표. 감염 여부를 판단하는 단서가 되는 파일 해시·IP·경로 등의 흔적.

# 2. 분석환경

## 2.1 분석 VM 환경

**표 2. 분석 VM 환경**

| **가상머신 소프트웨어** | **VMware** |
|---|---|
| 가상머신 버전 | Pro 25H2u1 (V25.0.1.25219725) |
| 가상머신 네트워크 설정 | NAT |
| 사용 OS | Windows 11 Pro |

## 2.2 악성코드 분석 도구

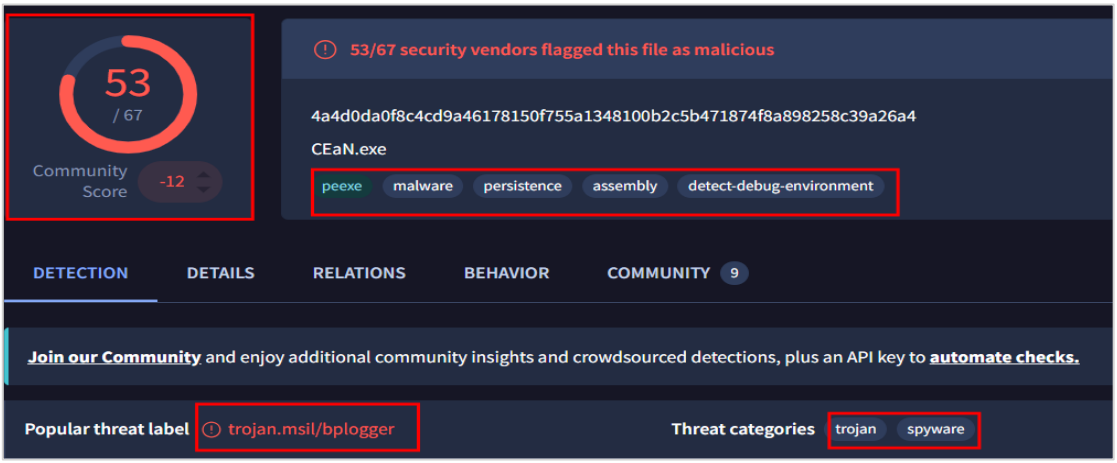
**표 3. 악성코드 분석 도구 목록**

| **분류** | **도구명** | **주요 용도** |
|---|---|---|
| 자동화 분석 | VirusTotal | 자동화 분석 실제 분석 시 내용 참고 |
| 기초분석 | ExeinfoPE | 파일 기본 정보 및 패킹 유무 확인 |
| | DIE | 파일 기본 정보 및 패킹 유무 확인 |
| 언패킹 | mal_unpack, de4dot | 패킹되어 있는 파일을 풀어 문자열을 확보 |
| 정적분석 | PEview | PE 구조 상세 확인, Import 확인 |
| | BinText | 악성코드에서 사용되는 문자열 확인 |
| 동적분석 | Process Explorer (ProcExp) | 프로세스(부모-자식) 트리구조 확인 |
| | Autoruns | 지속성 확보 확인 |
| | System Explorer | 스냅샷을 통한 파일 변경 이력 확인 |
| | Process Monitor (ProcMon) | 악성코드 행위 자세한 로그 확인 |
| | CurrPorts | 악성코드와 연결된 IP, 포트 확인 |
| | SmartSniff | 실제 네트워크 패킷 요약 확인 |
| | Wireshark | 네트워크 패킷 상세 확인 |

# 3. 기초분석

## 3.1 VirusTotal(자동화 분석)

VirusTotal(이하 VT)를 통해 자동화 분석 결과 해당 CEaN.exe는 전체 69개의 엔진에서 54개의 탐지 결과를 보였다. 트로이목마·스파이웨어 유형이며 MSIL, BPLogger, AgentTesla 패밀리군으로 확인되었다. 가장 많이 확인된 라벨은 [trojan.msil/bplogger]이다. 행위라벨을 보면 분석환경 탐지를 시도하고 지속성 확보와 자기복사의 행위를 한다는 것을 유추할 수 있다.




**[그림 1. VirusTotal — 탐지 결과 및 주요 태그]**

샘플의 오리지널 네임은 "CEaN.exe"이지만 관측된 파일명 목록에서 "RFQ_Purchase_Order_Quotation_Forms.exe"와 같은 주문양식으로 위장하는 흔적이 발견되었다. 파일 서명이 없는 것 역시 위장 파일의 증거 중 하나로 트로이목마의 특성을 뒷받침해 준다.


**[그림 2. VirusTotal — 위장 파일명 및 미서명 확인]**

VT를 통해 해당 파일이 어떤 네트워크 통신을 하고 어떤 파일을 드롭하는지 빠르게 참고할 수 있다.

**표 4. VirusTotal — 네트워크 통신 및 드롭 파일**

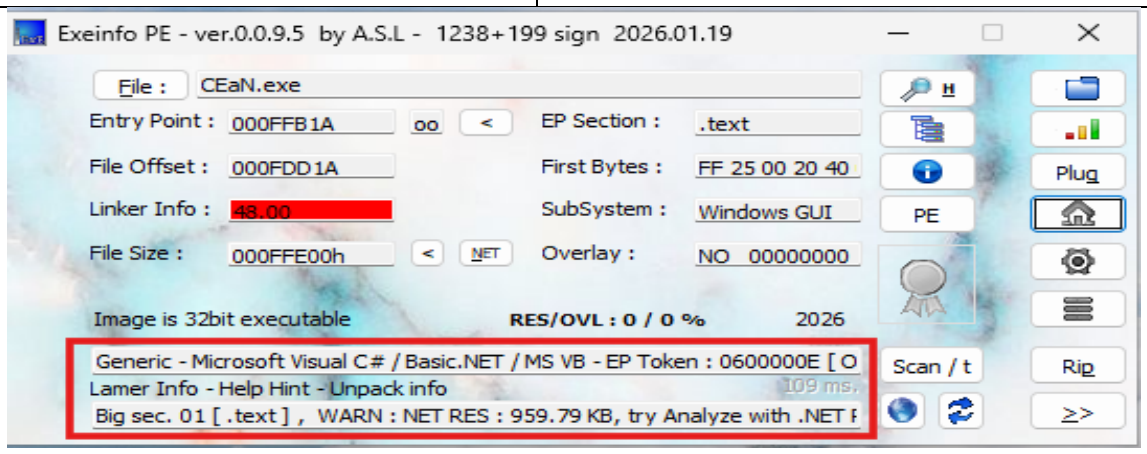
| 구분 | 내용 | 비고 |
|---|---|---|
| URL | `http://ip-api[.]com/line/?fields=hosting` | |
| 도메인 | ip-api[.]com | 합법적인 IP 확인 서비스 |
| | mail[.]onionmail[.]org | 합법적인 익명 메일 서비스 |
| IP | 162[.]159[.]36[.]2 | 실제 분석에서 확인되지 않음 |
| | 208[.]95[.]112[.]1 | US |
| Dropped Files | __PSScriptPolicyTest_vlzwlq13.1cy.psm1 | |

## 3.2 파일 식별 및 패킹 확인

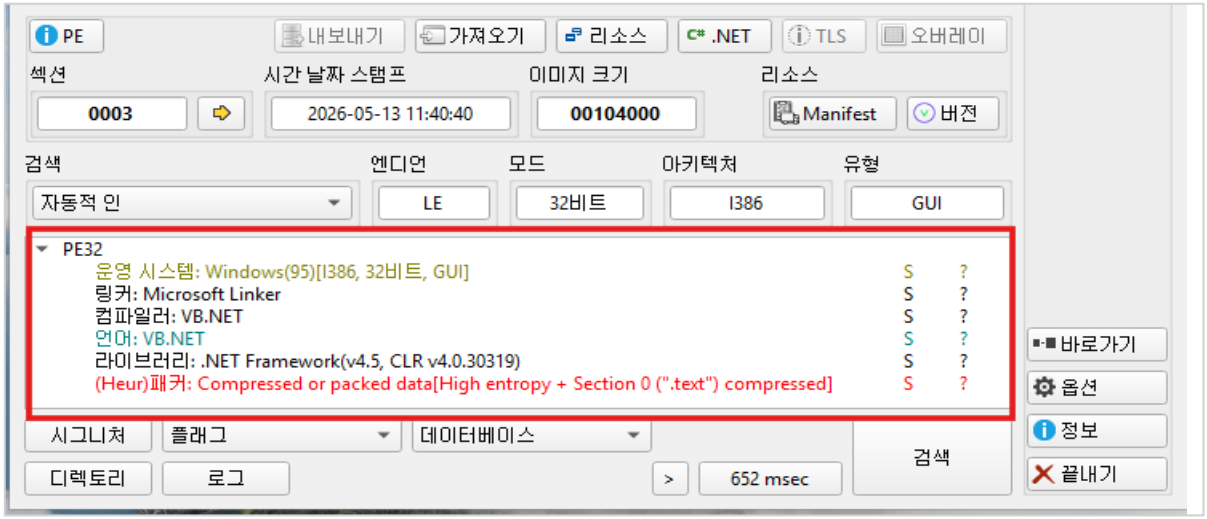
ExeinfoPE와 DIE를 사용하여 교차 검증한 결과 CEaN.exe는 VB.NET이 사용된 32비트 환경의 GUI 프로그램으로 서명정보가 없는 것이 확인되었다. DIE 확인결과 패킹됨[5](98%)이 확인되었고 특히 .text는 약 7.85의 엔트로피[6] 값으로 압축/암호화가 의심된다. UPX와 같은 외부 표준 패커는 아닌 것으로 확인된다.

**표 5. 파일 기본 정보 및 패킹 판정**

| 항목 | 값 |
|---|---|
| 타입 | VB.NET / .NET v4.0.30319 |
| 아키텍처 | 32비트 GUI |
| 서명 | 미서명 |
| Entry Point[7] | 000FFB1A (.text) |
| 패킹 판정 | 패킹됨 98% (DIE), 표준 패커 아님 |




**[그림 3. ExeinfoPE — 파일 정보 확인 및 패킹 확인]**



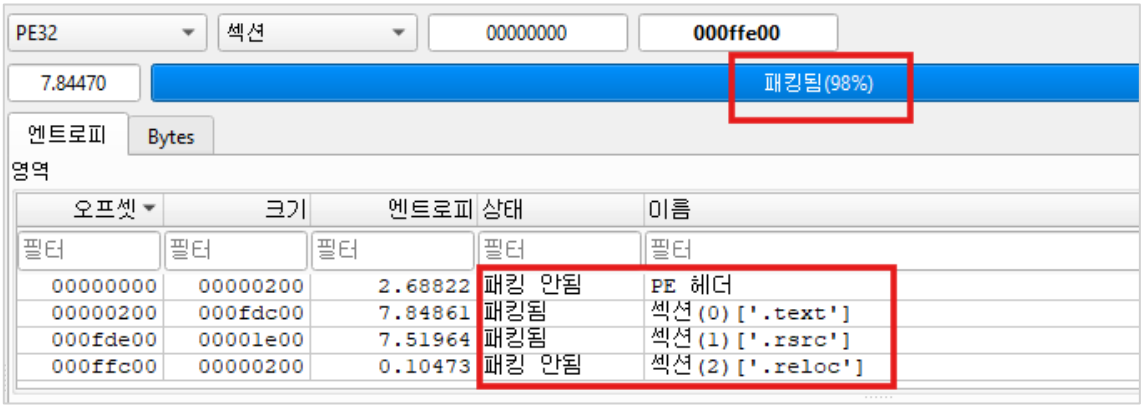

**[그림 4. DIE — 파일정보 확인 및 패킹 확인]**

---

[5] 실행 파일을 압축·암호화해 내부 코드를 숨기는 기법. 분석을 어렵게 한다.

[6] 데이터의 무작위성 정도를 0~8 사이 값으로 나타낸 지표. 보통 7 이상이면 압축·암호화 가능성이 높다고 본다.

[7] 프로그램 실행 시 코드가 가장 먼저 시작되는 진입점 주소.



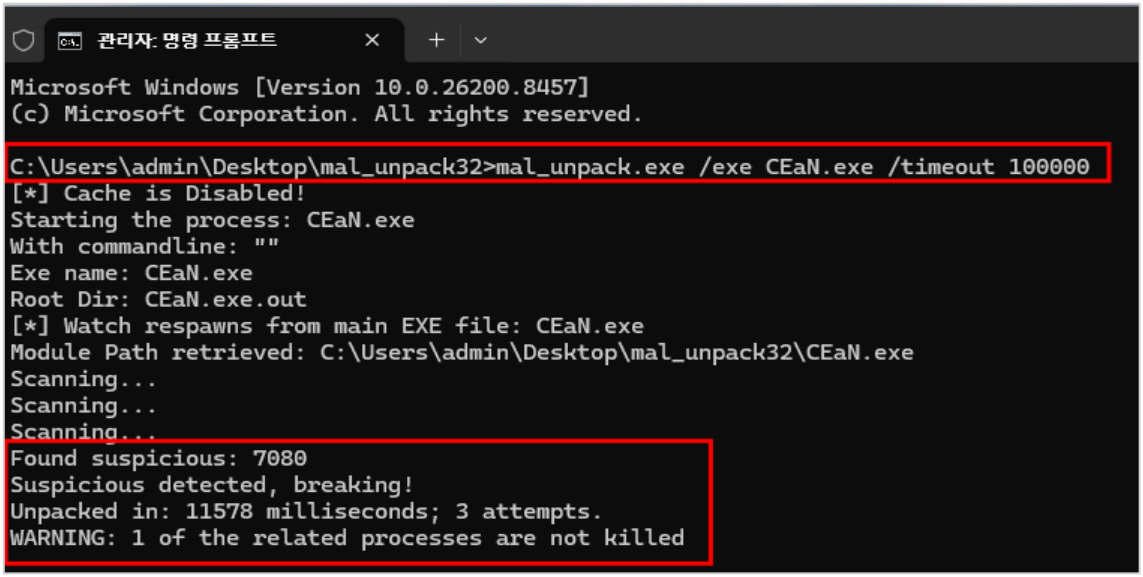

**[그림 5. DIE — 엔트로피 정보확인]**

# 4. 정적분석

## 4.1 언패킹

.NET 파일의 난독화[8] 해제 도구인 de4dot으로 언패킹을 시도하였지만 Detected Unknown Obfuscator(난독화 도구 식별 실패)가 출력되며 난독화 도구를 식별하지 못했다. 결과적으로 -cleaned.exe가 붙은 파일은 생성되었으나 실질적인 언패킹은 이루어지지 않았다.

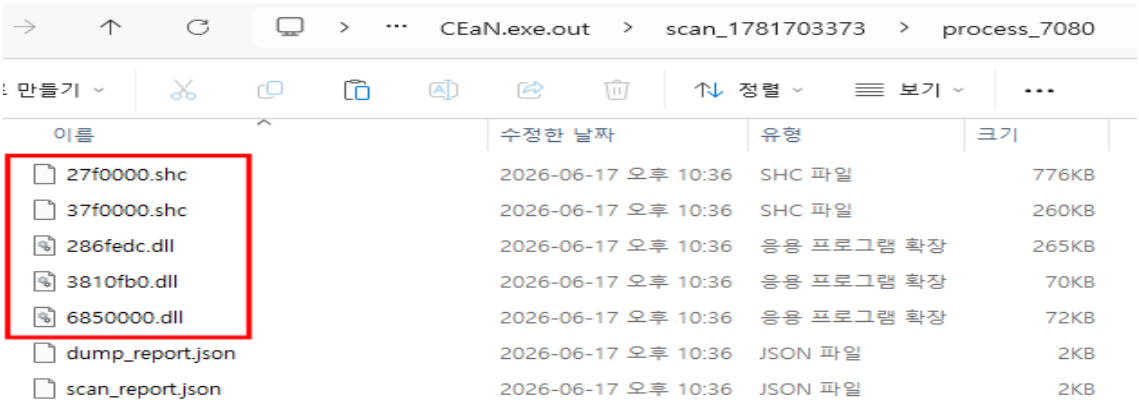
추가로 메모리 덤프[9] 방식을 사용해 언패킹하는 mal_unpack으로 언패킹을 재시도하였다. 그 결과 덤프된 5개의 모듈을 얻을 수 있었다. 하지만 완벽하게 언패킹된 것은 아니고 일부 문자열을 확인할 수 있었다.




**[그림 6. mal_unpack — 메모리 덤프 언패킹 실행]**

[8] 코드나 글자를 일부러 알아보기 어렵게 바꿔 분석을 방해하는 기법.

[9] 실행 중인 프로그램의 메모리 상태를 그대로 파일로 떠내는 기법. 압축이 풀린 코드를 메모리에서 확보할 때 쓴다.




**[그림 7. 탐색기 — 덤프된 5개 모듈 파일]**

## 4.2 PE 구조[10]

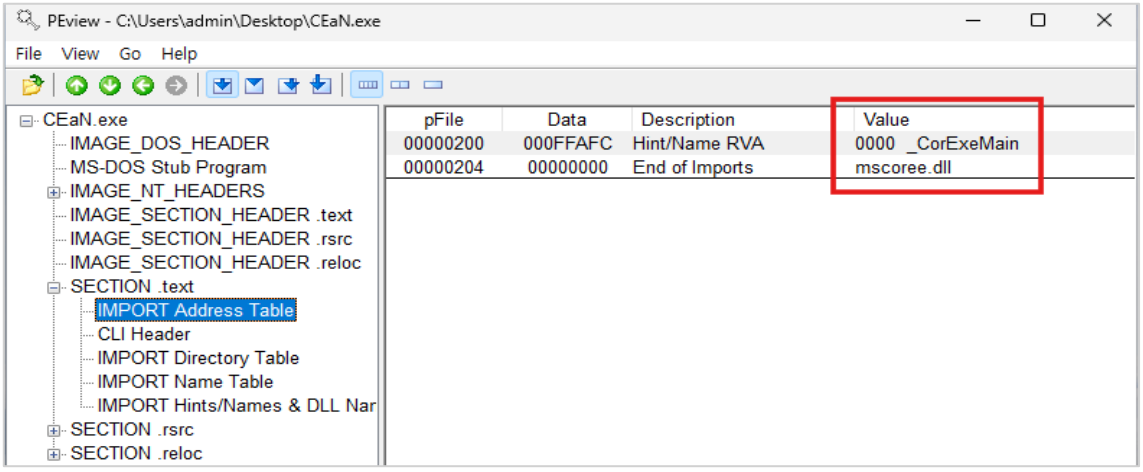
PEview 확인 결과 .text, .rsrc, .reloc 3개의 섹션으로 구성되어 있다. .text와 .rsrc는 Raw Size와 Virtual Size가 거의 일치한다. .reloc는 다소 차이가 있지만 파일 정렬과 같은 이유로 정상적인 차이이다.

**표 6. PE 섹션 구조**

| 섹션명 | Raw Size | Virtual Size | 비고 |
|---|---|---|---|
| .text | 000FDC00 | 000FDB50 | |
| .rsrc | 00001E00 | 00001D78 | |
| .reloc | 00000200 | 0000000C | |

## 4.3 Import 분석

PEview로 Import를 확인한 결과, Import DLL은 `mscoree.dll` 하나, 함수는 `CorExeMain` 하나뿐이었다. 이는 .NET 실행파일의 100% 전형적인 형태로, 실제 악성 기능은 PE Import 테이블이 아니라 .NET 런타임(CLR)[11] 내부에서 처리되기 때문이다. 따라서 Import를 통한 정적분석은 불가능하다.




**[그림 8. PEview — Import 테이블 구조]**

[10] 윈도우 실행 파일(.exe 등)이 어떻게 구성되는지를 정한 표준 파일 형식.

[11] .NET 프로그램을 실행·관리하는 런타임 엔진. .NET 악성코드의 실제 동작은 이 내부에서 처리된다.

## 4.4 문자열 분석

**표 7. 주요 문자열 분석**

| 문자열 | 비고 |
|---|---|
| HotelManager,BillingForm<br>Select Checked-In Reservation:등등 | 호텔 예약·관리 프로그램인 척 위장하려고 넣은 단어들로 추정 |
| AI-driven autonomous system optimization and self-healing platform | 호텔 예약 및 관리 프로그램으로 위장하기 위해 삽입된 문자열로 추정 |
| AesManaged, AesCryptoServiceProvider, SymmetricAlgorithm | AES(데이터를 자물쇠로 잠그는 표준 암호화 방식). 악성코드가 자기 설정이나 훔친 정보를 숨기는 데 사용 |
| DpapiDataProtector | DPAPI[12](윈도우가 비밀번호를 안전하게 보관할 때 쓰는 기능). 원래는 보호용인데, 악성코드가 이걸 거꾸로 써서 브라우저에 저장된 비밀번호를 다시 풀어냄 → 자격증명(계정·비밀번호) 탈취의 핵심 단서 |
| Microsoft Enhanced RSA and AES Cryptographic Provider | 윈도우가 기본 제공하는 암호화 기능 이름 |
| ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/= | Base64(데이터를 글자로 바꿔 포장하는 방식). 훔친 정보를 밖으로 빼낼 때 포장용으로 추정 |
| CEaN.resources, Culture=ko-KR, BillingForm.ko-KR.resources, BillingForm.ko.resources | 한국어(ko-KR) 표시 데이터 포함 → 한국 사용자를 노린 정황 |
| zh-Hans, zh-Hant, zh-CHS, zh-CHT, zh-TW | 중국어(간체·번체) 표시 데이터 → 한·중 등 동아시아 사용자 대상 정황 |

# 5. 동적분석

## 5.1 프로세스

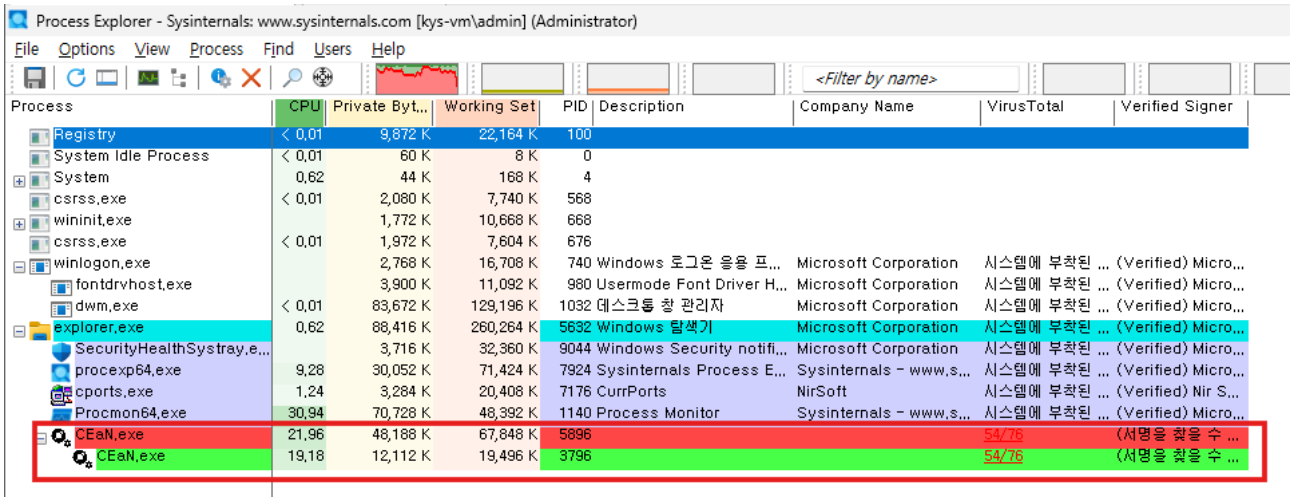
### 프로세스 생성

Process Explorer로 확인한 결과 CEaN.exe를 실행하면 별도의 외부 프로세스를 생성하지 않고, 자기 자신을 다른 PID의 자식 프로세스로 재실행(복제)한다. 부모 프로세스는 종료되고 자식 프로세스가 메인으로 동작한다. 이는 .NET 계열 파일이 실행 시 메모리에서 페이로드[13]를 복호화·실행하는 구조이며, 자기 재실행 과정에서 언패킹된 코드가 메모리에서 동작하는 것으로 보인다. 이 과정에서 악성코드는 디스크 내 파일을 검사하는 온디스크 백신 스캔[14]을 회피하는 효과를 가진다.

[12] Data Protection API. 윈도우가 비밀번호·인증서를 안전하게 암호화해 보관하는 자체 기능. 악성코드가 이를 거꾸로 이용해 저장된 비밀번호를 복호화한다.

[13] 악성코드가 실제로 실행하려는 핵심 악성 코드·기능 부분.

[14] 백신이 하드디스크에 저장된 파일을 검사하는 기본 방식. 메모리에서만 동작하면 이 검사를 피하게 된다.



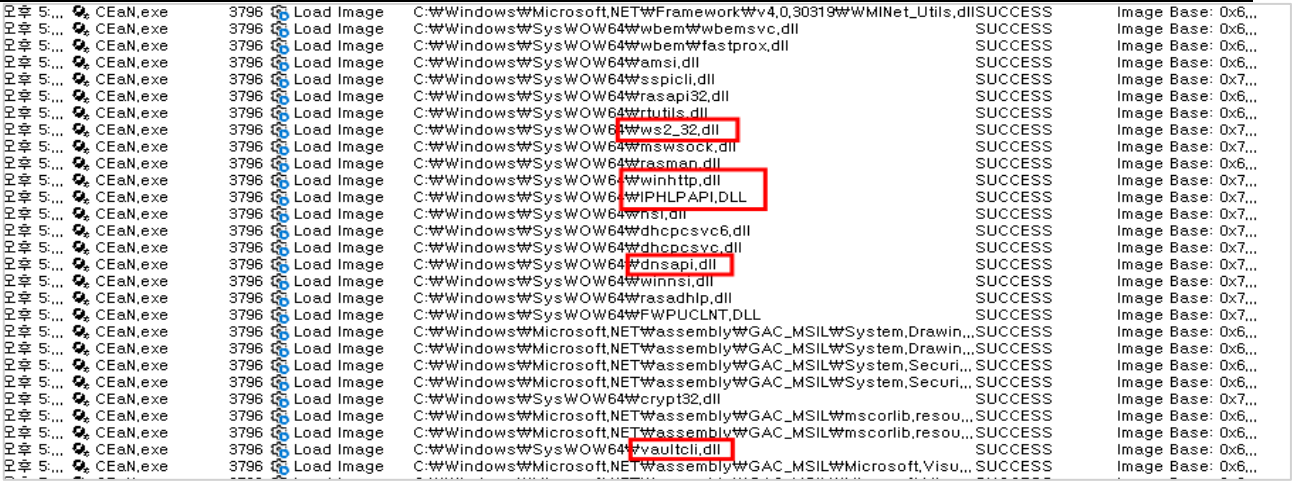

**[그림 9. Process Explorer — 프로세스 재실행 구조 (PID 가변)]**

## 프로세스 로드 DLL

앞서 정적분석에서 `mscoree.dll` 하나만 확인되었지만 ProcMon을 통해 CEaN.exe 프로세스에 실제 로드되는 DLL을 확인한 결과 네트워크·암호화·자격증명 등 실제 악성 기능에 필요한 DLL이 동적으로 로드되었다. 이를 통해 악성코드가 정보를 탈취해서 네트워크 통신으로 유출한다는 가능성이 높아졌다.

**표 8. 동적 로드 DLL 목록**

| DLL 종류 | 추정 용도 | 기능 분류 |
|---|---|---|
| ws2_32.dll, winhttp.dll, dnsapi.dll | 소켓 통신, HTTP 요청[15], DNS 해석[16] | 네트워크 |
| bcrypt.dll, rsaenh.dll | 복호화·암호화 연산 | 암호화 |
| vaultcli.dll | Windows 자격증명 보관소 접근 | 자격증명 |




**[그림 10. ProcMon — 동적 로드 DLL 목록]**

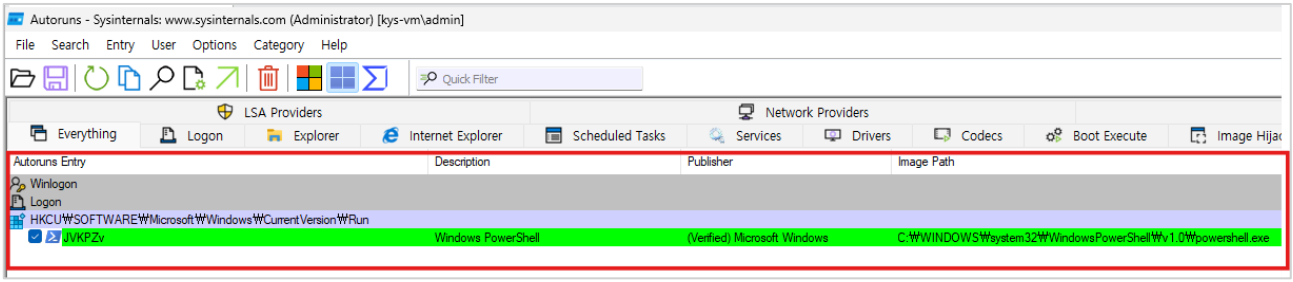
[15] 웹 통신에 쓰는 규칙. 암호화 없이 평문으로 오간다(기본 80번 포트).

[16] 도메인 이름을 실제 IP 주소로 바꿔 주는, 인터넷 전화번호부 같은 체계.

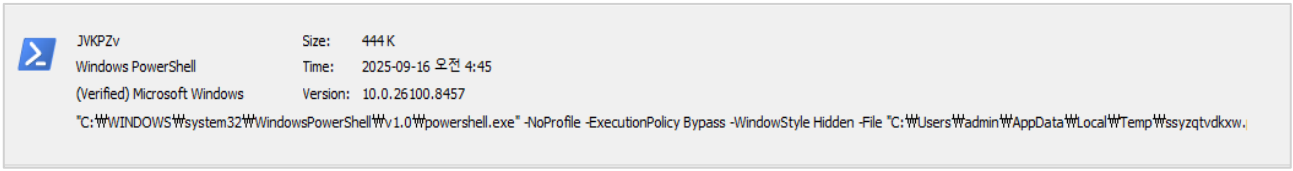
## 5.2 레지스트리

Autoruns를 사용하여 CEaN.exe를 실행하기 전과 후를 비교해본 결과 사용자 로그인 시 실행되는 `HKCU\...\Run` 키 경로에 새로운 값이 등록되어 있었다[17]. 등록된 값 이름은 JVKPZv이며 윈도우 기본 프로그램인 PowerShell을 사용하여 "`%LOCALAPPDATA%\Temp`"(고정) 경로에 있는 페이로드인 `bxxjheyu31h.ps1`(랜덤 파일명)을 실행하게 되어 있다. 또한 `-NoProfile` , `-ExecutionPolicy Bypass`, `-WindowStyle Hidden`의 옵션이 포함되어 있다. 각각 흔적 최소화, 차단 정책 우회, 창 숨기기를 의미한다. 이를 종합하면 전형적인 LotL(Living-off-the-Land) 기법[18]이 된다. 즉 사용자가 컴퓨터에 재로그인 하는 시점부터 사용자 몰래 PowerShell이 실행되고 악성 페이로드를 실행하는 자동 실행기반의 메커니즘이 확인된다. 실제 지속성이 확보되는지는 추가 검증을 한다.

*반복된 분석을 통해 Run 키 값이 JVKPZv로 동일했지만 다른 분석 환경에서도 값이 같다는 것은 확인되지 않았다.




**[그림 11. Autoruns — 자동 실행 등록]**



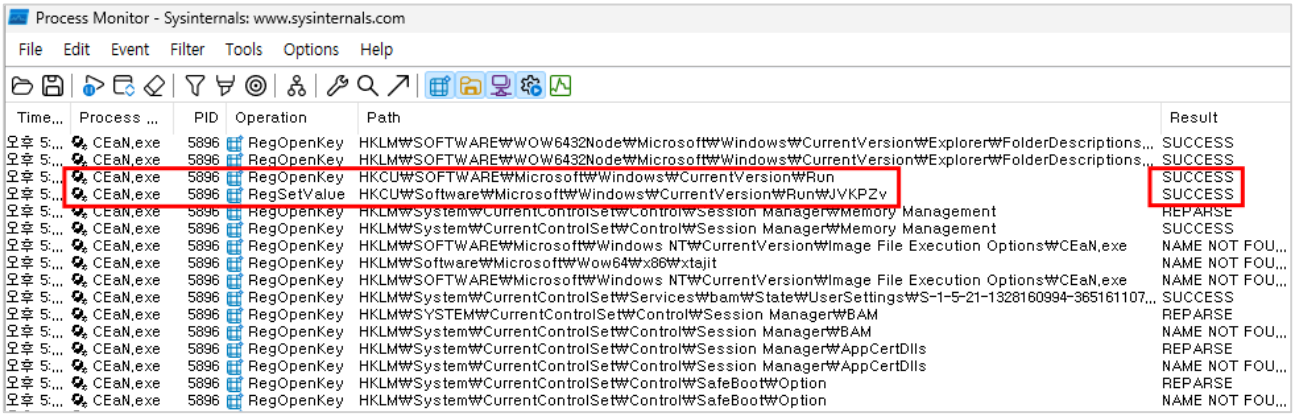

**[그림 12. Autoruns — Run 키에 등록된 PowerShell 실행 명령]**

ProcMon으로 추가 확인한 결과 CEaN.exe가 직접 RegSetValue를 수행해 `HKCU\...\Run`에 JVKPZv 값을 추가하는 로그를 확인했다.

---

[17] 컴퓨터에 로그인할 때마다 지정한 프로그램이 자동 실행되도록 등록해 두는 윈도우 레지스트리 경로.

[18] Living-off-the-Land. 외부 악성 도구를 내려받지 않고, 윈도우에 기본 설치된 정상 프로그램(예: 파워셸)을 무기처럼 악용하는 공격 기법.



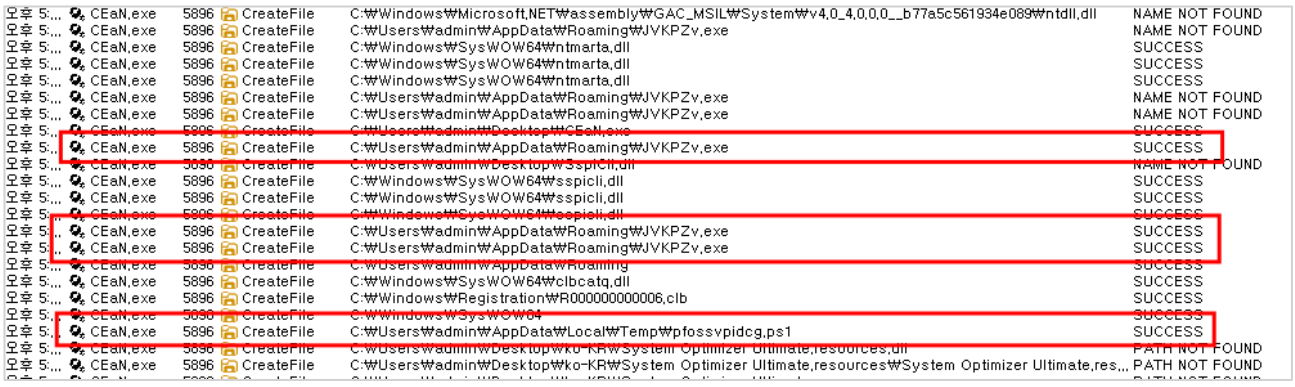

**[그림 13. ProcMon — HKCU Run 키 레지스트리 등록 성공]**

## 5.3 파일

### 파일 생성

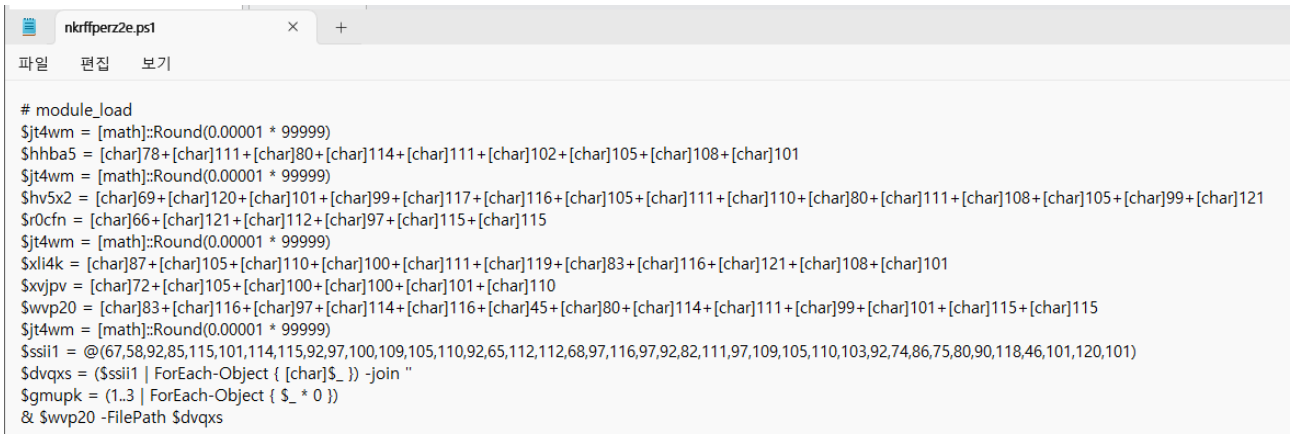
System Explorer로 CEaN.exe 실행 전후 변경된 파일을 확인한 결과, `%LOCALAPPDATA%\Temp` 경로에 Run 키가 실행하는 페이로드인 [가변].ps1이, `%APPDATA%\Roaming` 경로에는 JVKPZv.exe 파일이 생성된 것을 확인했다. ProcMon으로 상세 로그를 확인한 결과 두 파일을 생성한 주체는 CEaN.exe로 판별되었다.


**[그림 14. System Explorer — 드롭된 페이로드 및 자기 복사본]**



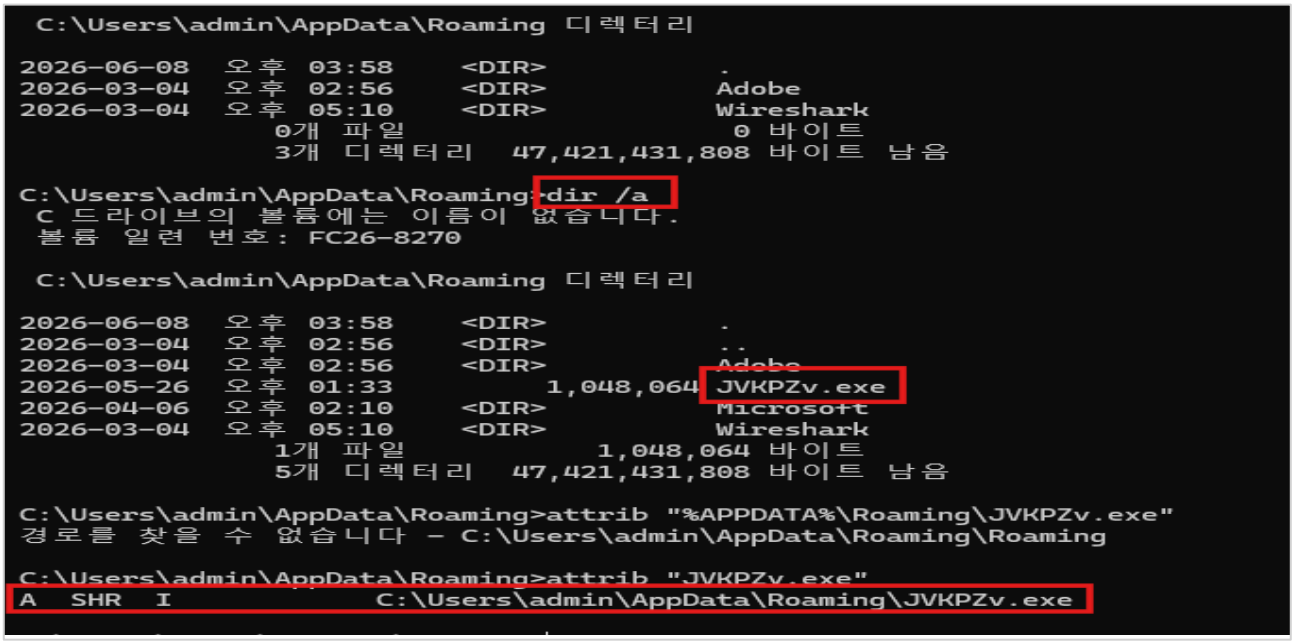

**[그림 15. ProcMon — 페이로드 및 복사본 생성 로그]**

.ps1 파일을 확인한 결과, 평문이 아닌 문자 코드 합산 방식으로 난독화되어 있었다. 이를 클로드 AI를 활용하여 디코딩한 결과 `Start-Process -FilePath "%APPDATA%\Roaming\JVKPZv.exe"` 스크립트를 확보했다. 즉 Run 키에 등록된 내용은 `%APPDATA%\Roaming\JVKPZv.exe`를 실행하라는 것이다.




**[그림 16. 메모장 — 난독화된 지속성 런처(.ps1)]**

JVKPZv.exe를 확인하려 폴더를 열었으나 파일이 보이지 않았다. 숨김 처리로 판단해 GUI에서 숨김 항목 표시를 켰지만 여전히 보이지 않았고, CLI(`dir /a`)로 재확인한 결과 파일에 `S`(System)+`H`(Hidden)+`I`(Not content indexed) 속성[19]이 부여되어 일반 사용자가 찾을 수 없게 설정돼 있었다.



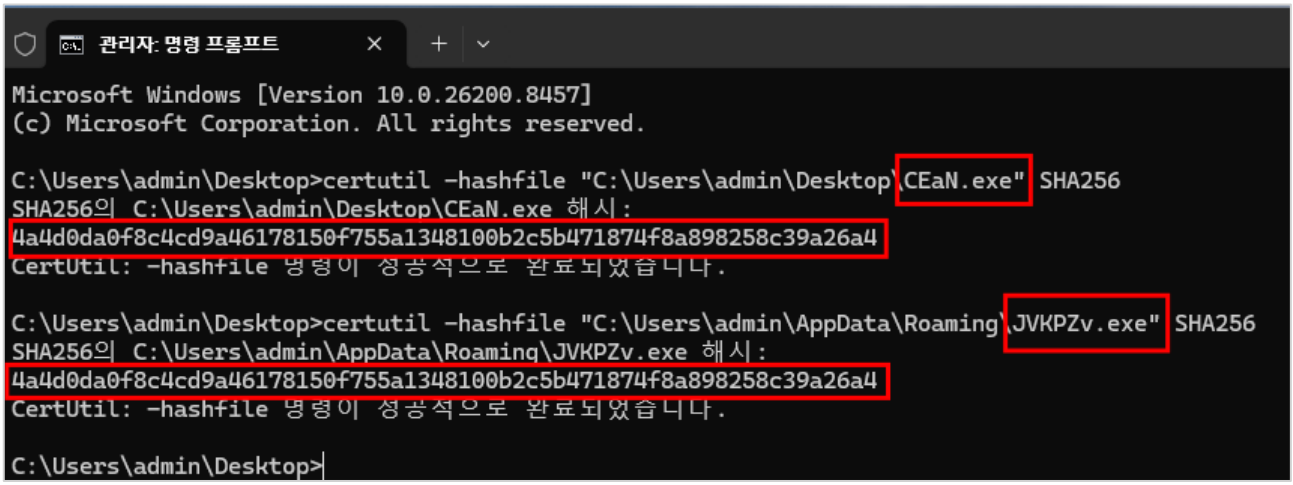

**[그림 17. CMD — 복사본 파일 S·H 속성 은폐 확인]**

JVKPZv.exe 파일의 정체를 확인하기 위해 터미널을 통해 SHA-256 해시값[20]을 추출하였고 원본 파일의 해시값과 비교한 결과 JVKPZv.exe의 정체는 원본 파일의 복사본으로 판정되었다.

---

[19] System·Hidden. 파일에 시스템·숨김 속성을 부여해 일반 탐색기에서 보이지 않게 하는 윈도우 파일 속성.

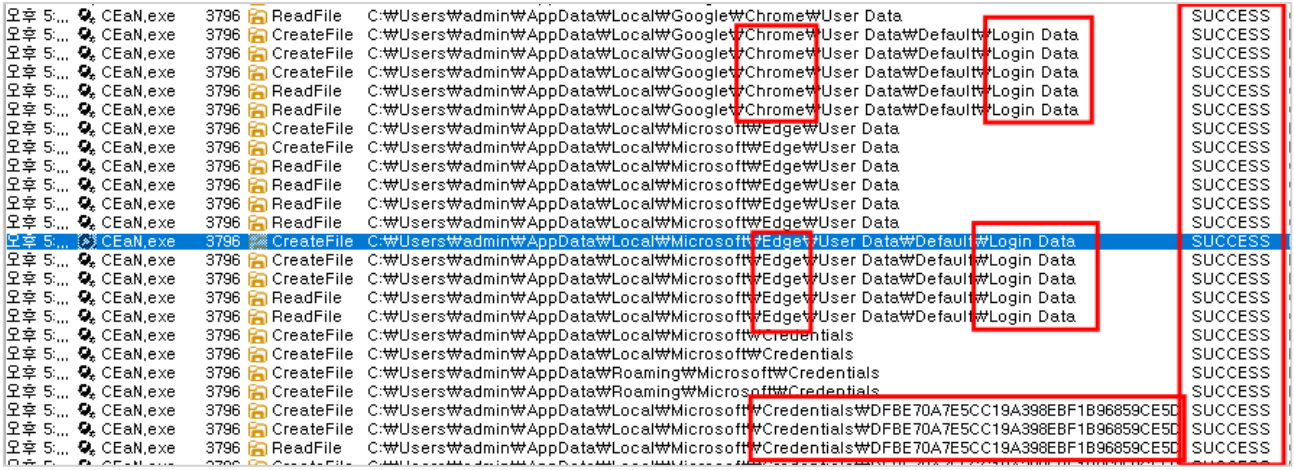
[20] 파일 내용을 고정된 길이의 문자열로 바꾼 고유 지문값. 같은 파일인지 확인할 때 쓴다.




**[그림 18. CMD — 원본 및 복사본 SHA-256 해시 일치]**

## 정보수집 행위

ProcMon의 CreateFile·ReadFile 로그를 확인한 결과, CEaN.exe가 정상적으로는 접근할 이유가 없는 다수 애플리케이션의 폴더에 접근하는 것을 확인했다. 설치된 브라우저인 Chrome과 Edge의 경우, `User Data\Default\Login Data`(브라우저가 저장한 로그인 정보 DB)를 CreateFile로 연 뒤 ReadFile로 내용을 실제로 읽는 데 성공(SUCCESS)했다. 나아가 `...\Microsoft\Credentials`(윈도우 자격증명 저장소)와 `...\Microsoft\Protect`(DPAPI 마스터키 — 브라우저가 암호화해 저장한 비밀번호를 복호화하는 데 필요한 열쇠)까지 읽어냈다.




**[그림 19. ProcMon — 로그인 데이터 및 자격증명 접근]**

반면 이 PC에 설치되지 않은 다수 애플리케이션에 대해서도 동일한 방식으로 접근을 시도했으나, 파일이 존재하지 않아 실패(PATH/NAME NOT FOUND)했다. 대상은 브라우저(Opera, Brave, Vivaldi 등), 메일 클라이언트(Thunderbird, Foxmail 등), FTP 클라이언트(FileZilla, SmartFTP 등), VPN(NordVPN), 메신저(Discord 등) 등 광범위하다.


**[그림 20. ProcMon — 전수 탐색 시도]**

즉 CEaN.exe는 알려진 자격증명 저장 위치를 광범위하게 전수 탐색하고, 존재하는 대상은 실제로 읽어내는 인포스틸러 행위를 수행한다.

## 5.4 네트워크

### 더미데이터 생성

앞서 ProcMon에서 CEaN.exe가 사용자 자격증명에 접근·수집하는 것을 확인했다. 네트워크 분석에서 유출 정황을 명확히 관찰하기 위해, 브라우저(Chrome, Edge)에 더미 로그인 데이터를 사전 입력했다.


**[그림 21. Chrome — 유출 테스트용 더미 자격증명]**

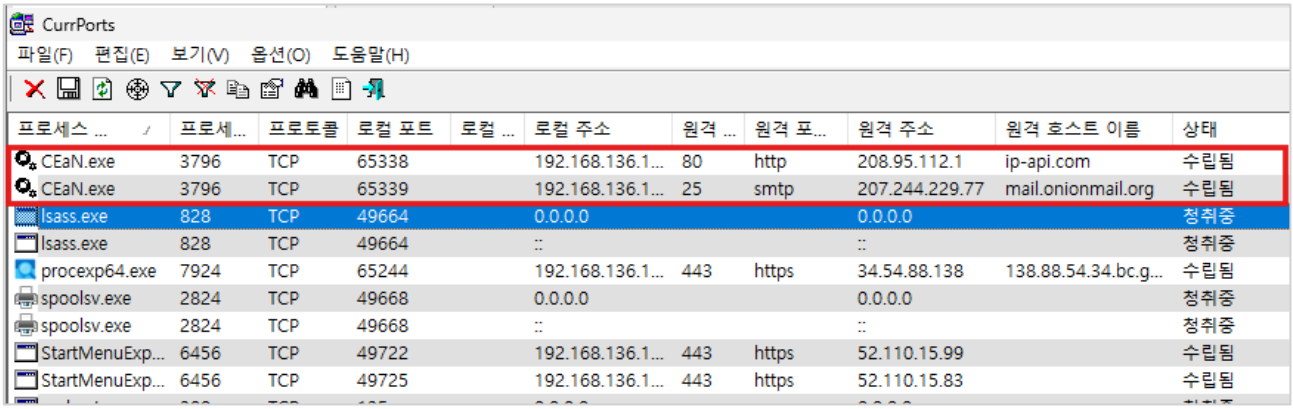
### 네트워크 연결

CurrPorts로 확인한 결과, CEaN.exe가 두 개의 IP(208[.]95[.]112[.]1:80, 207[.]244[.]229[.]77:25)와 TCP[21]·ESTABLISHED 상태로 연결을 수립한 것이 확인됐다. SmartSniff로 통신 내용을 확인한 결과, 208[.]95[.]112[.]1(ip-api[.]com)과는 HTTP 평문 통신이, 207[.]244[.]229[.]77 (mail[.]onionmail[.]org)와는 SMTP 평문 통신이 오가는 것을 확인했다[22].

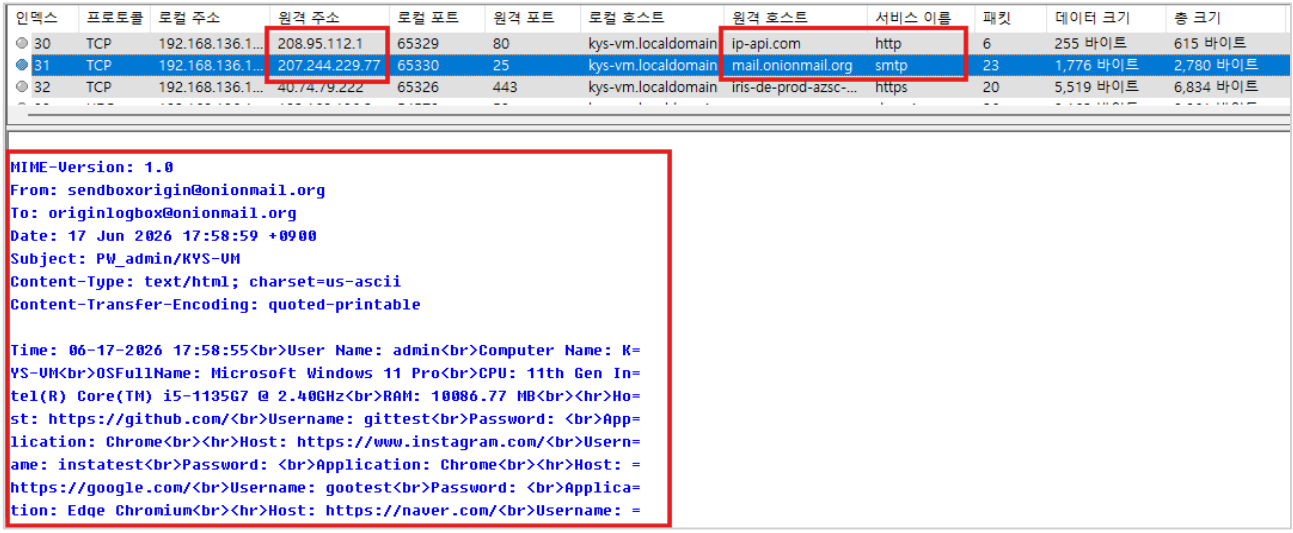
---

[21] 데이터를 빠짐없이 주고받도록 연결을 맺는 통신 방식.

[22] 메일을 보낼 때 쓰는 통신 규칙(기본 25번 포트).



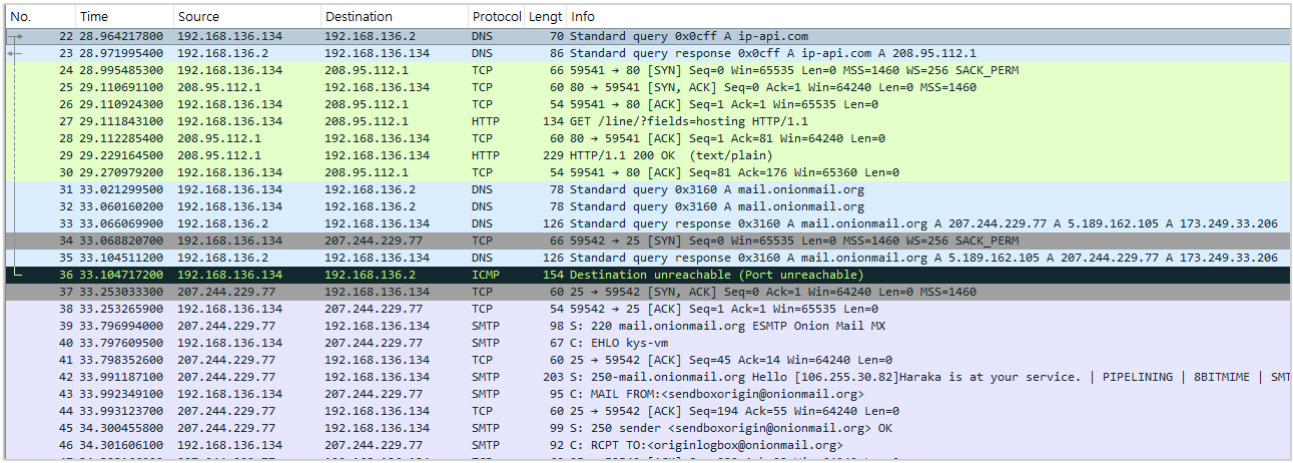

**[그림 22. CurrPorts — 외부 통신 IP 및 포트 연결]**




**[그림 23. SmartSniff — HTTP 및 SMTP 평문 통신]**

## 패킷 분석

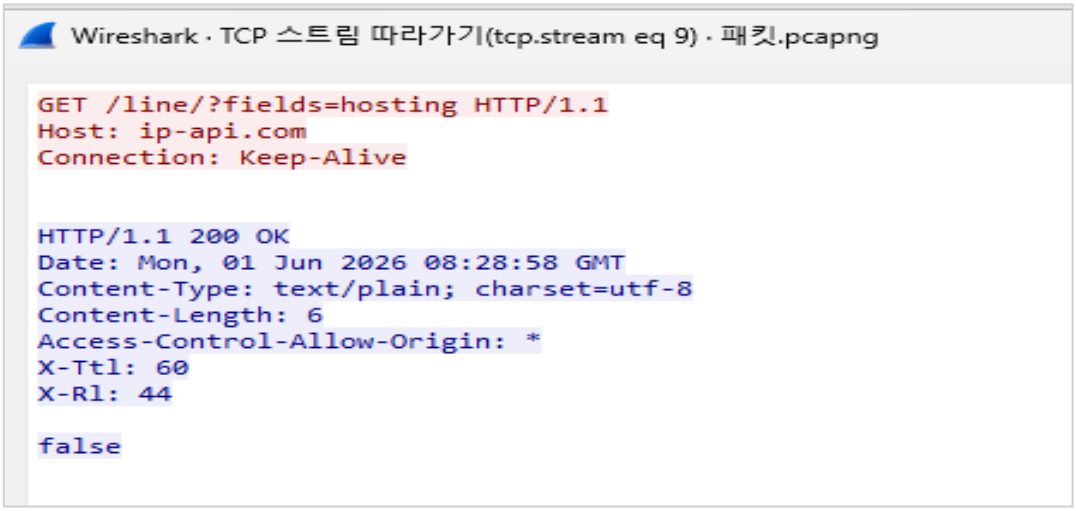
Wireshark로 실제 패킷을 캡처해, DNS 질의로 IP를 확인한 뒤 HTTP·SMTP 통신이 이뤄지는 흐름을 확인했다.




**[그림 24. Wireshark — 통신 패킷 전체 흐름]**

악성코드는 먼저 ip-api[.]com에 대한 DNS 질의로 208[.]95[.]112[.]1을 수신했다. 이어 `GET /line/?fields=hosting` 요청으로 자신의 IP가 호스팅(데이터센터) 환경인지 조회했고, 응답 false(일

반 사용자 환경)를 받았다. 이는 CEaN.exe가 악성코드 분석 환경(샌드박스[23])을 회피하기 위한 사전 정찰 행위로 추정된다.




**[그림 25. Wireshark — 호스팅 환경(데이터센터) 정찰 패킷]**

이어서 메일서버의 도메인인 mail[.]onionmail[.]org에 대한 DNS 질의를 전송했고 3개의 IP를 수신하였다.

**표 9. DNS 질의 및 응답**

| | |
|---|---|
| DNS 질의 | Standard query A mail[.]onionmail[.]org |
| DNS 응답 | A 5[.]189[.]162[.]105, A 173[.]249[.]33[.]206, A 207[.]244[.]229[.]77 |

*SMTP 통신에 연결되는 IP는 3개 중 매번 달라진다.

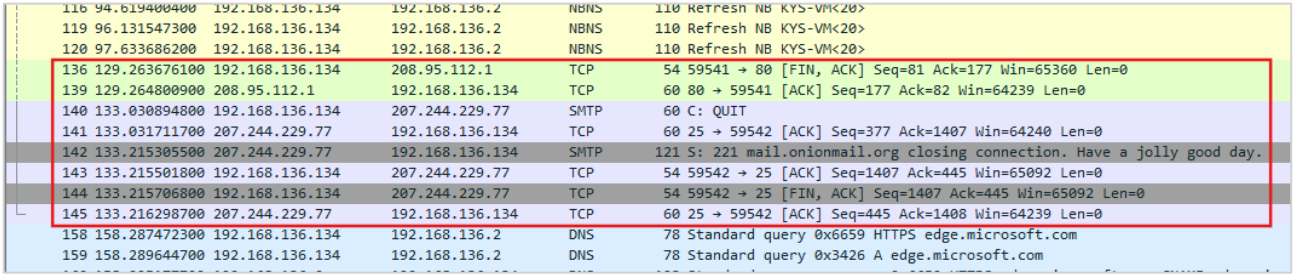
CEaN.exe는 3개의 IP 중 207[.]244[.]229[.]77을 선택해 SMTP 통신 연결을 수립하였고 데이터 탈취에 해당하는 통신을 이어나갔다. [그림 26]을 보면 패킷의 내용이 이메일 형식인 것을 확인할 수 있다. 본문에 평문으로 담긴 정보는 다음과 같다.

- 시스템정보: User Name(admin), Computer Name(KYS-VM), OS(Windows 11 Pro), CPU(i5-1135G7), RAM
- 브라우저 자격증명(더미): instagram·google·naver 등의 Username, Password는 전 항목 공란으로 비워져있음
- Windows Credential 항목: Microsoft 계정 MS 앱의 인증 정보, 공유 폴더 접근 계정 정보 등이 담긴 Windows 자격증명 저장소

[23] 악성코드를 안전하게 실행·관찰하기 위한 격리된 자동 분석 환경.


**[그림 26. Wireshark — 평문 SMTP 자격증명 유출 본문 일부]**

메일 전송이 완료된 후에는 두 개의 IP 모두 정상적으로 연결이 종료된 것을 확인할 수 있다.




**[그림 27. Wireshark — 유출 전송 완료 및 통신 종료]**

## 5.5 지속성 확보 검증

앞선 분석에서 `HKCU\...\Run` 경로에 JVKPZv 값이 등록된 것을 확인했다. 해당 값은 사용자에게 보이지 않게 PowerShell로 `%LOCALAPPDATA%\Temp`의 .ps1 파일을 실행하는 내용이었고, 이 .ps1은 `%APPDATA%\Roaming\JVKPZv.exe`를 실행하는 것이었다.

실제 지속성 동작을 검증하기 위해 분석 VM을 재부팅했다. 그 결과 로그인 시점에 ProcMon에서 Run 키 → PowerShell → JVKPZv.exe 순으로 프로세스가 연결되어 실행되는 것을 관찰했으며, 재실행된 JVKPZv.exe가 앞선 분석과 동일한 자격증명 접근·수집 행위를 재현하는 것을 확인했다.

# 6. 흐름정리

**표 10. 악성코드 실행 흐름 정리**

| # | 단계 | 행위 |
|---|---|---|
| 1 | 실행 | 더블클릭으로 실행 → .NET 런타임이 메모리에서 페이로드 복호화·전개 |
| 2 | 자기복사(드롭) | 자신을 `%APPDATA%\Roaming\JVKPZv.exe`로 복사 |
| 3 | 은폐 | 복사본에 `S·H`(시스템·숨김) 속성 부여 → 탐색기 `·dir`에서 안 보임 |
| 4 | 드롭 | `%LOCALAPPDATA%\Temp\<랜덤>.ps1` (지속성 런처) 생성 |
| 5 | 지속성 등록 | `HKCU\...\Run` 키에 PowerShell 런처 등록 → 다음 로그인부터 자동 실행 |
| 6 | 자격증명 수집 | 브라우저(Chrome·Edge) Login Data 접근 + DPAPI 마스터키 접근 |
| 7 | 유출 | 평문 SMTP로 onionmail 메일 서버에 유출, 250 Message processed 완료 |

# 7. ATT&CK

**표 11. MITRE ATT&CK[24] 매핑**

| 전술 (Tactic) | 기법 (Technique) | 관찰한 행위 |
|---|---|---|
| Execution (실행) | T1059.001 — Command and Scripting Interpreter: PowerShell (명령·스크립트 실행: 파워셸) | 지속성 런처(.ps1)가 PowerShell로 본체 재기동 |
| Persistence (지속성) | T1547.001 — Registry Run Keys / Startup Folder (레지스트리 실행 키 / 시작 폴더) | CEaN.exe가 직접 `HKCU\...\Run` 키 등록, 다음 로그인 시 자동 실행 |
| Defense Evasion (방어 회피) | T1564.001 — Hidden Files and Directories (숨김 파일 및 디렉터리) | 자기복사본에 `S·H` 속성 부여, 탐색기 은폐 |
| Credential Access (자격증명 접근) | T1555.003 — Credentials from Web Browsers (웹 브라우저 자격증명 탈취) | Chrome/Edge Login Data 실접근 + DPAPI 마스터키 접근 |
| Exfiltration (유출) | T1048.003 — Exfiltration Over Unencrypted Non-C2 Protocol (비암호화 비-C2 프로토콜[25]을 통한 유출) | 평문 SMTP로 onionmail 유출, 250 Message processed 완료 확인 |

[24] 전 세계 사이버 공격의 전술·기법을 체계적으로 분류해 둔 표준 지식 체계(프레임워크).

[25] C2(Command and Control). 공격자가 감염 PC를 원격 조종·명령하기 위한 서버·통신 채널.

# 8. 결론

악성코드 분석 결과, **CEaN.exe는 사용자의 자격증명을 탈취하는 인포스틸러로, 스파이웨어 및 트로이목마 유형의 악성코드로 판별되었다.** 분석으로 확인된 주요 악성 행위는 다음과 같다.

1. **자기복제**
2. **복사본 은폐(S·H 속성)**
3. **지속성 확보(HKCU Run)**
4. **정상 도구 악용(LotL)**
5. **브라우저·시스템 자격증명 수집**
6. **분석 환경 탐지**
7. **수집 정보 유출(평문 SMTP)**

**VT 탐지율은 54/69**이며, MSIL·BPLogger·**AgentTesla** 패밀리로 분류된다. VT가 제시한 유형(트로이목마·스파이웨어) 역시 실제 분석 결과와 일치했다.

CEaN.exe는 .NET 기반 파일로, 표준 패커가 아닌 .NET 내부 압축·난독화로 실제 페이로드를 숨긴 형태였다. 완전한 언패킹은 이루어지지 않았으나 mal_unpack으로 압축을 일부 해제했고, **문자열 분석에서 호텔 예약 서비스 또는 AI 서비스로 추정되는 위장 문자열이 확인**되었다. 또한 한국어·중국어 리소스가 포함되어, **한국 등 동아시아 사용자를 대상으로 한 정황도 관찰**되었다.

실행 시 CEaN.exe는 **자기복제 실행(부모→자식 프로세스 재실행)으로 페이로드를 전개**하고, 자기복사본 JVKPZv.exe를 `%APPDATA%\Roaming`에 생성한 뒤 `S·H` 속성(System·Hidden, 일반 탐색기에서 보이지 않게 하는 파일 속성)을 부여해 **은폐**한다. 이어 로그인 시 자동 실행되도록 .ps1 페이로드를 드롭하고, `HKCU Run` 키에 등록해 **지속성을 확보**한다.

정보 수집 단계에서는 **브라우저(Chrome·Edge)에 저장된 로그인 정보와 Windows 자격증명을 실제로 읽어냈으며, VPN·FTP·메일 등 다양한 자격증명 저장소를 광범위하게 전수 탐색했다**. 수집한 정보는 정상 서비스인 ip-api[.]com(IP 정보 조회 서비스)과 onionmail[.]org(익명 메일 서비스)를 경유해 유출한다. 유출에 앞서 ip-api[.]com으로 **자신의 IP가 분석 환경(데이터센터)인지 확인하는 정찰을 거친다**. 자격증명 유출은 **단순 시도가 아니라 SMTP 서버의 메일 전송 완료 응답(250 Message processed)으로 실제 완료가 확인**되며, 유출 직후 네트워크 통신은 신속히 종료된다.

종합하면 CEaN.exe는 실행과 동시에 자격증명을 수집·유출하고 빠르게 동작을 종료하는 한편, 지속성 메커니즘을 통해 재로그인 시마다 동일한 유출 행위를 반복하는 인포스틸러다.

본 분석은 정적·동적·네트워크 분석을 통해 악성코드를 분석하는 데 중점을 두었으며 코드 단위의 분석이나 메모리 분석은 범위에서 제외했다.

# 9. 예방 및 대응

## 9.1 감염 확인

- **레지스트리 확인:** `HKCU\Software\Microsoft\Windows\CurrentVersion\Run` 경로에 JVKPZv라는 이름으로 PowerShell 실행 명령어(숨김 창 실행 및 바이패스 옵션 포함)가 등록되어 있는지 확인한다.
- **파일 확인:** `%APPDATA%\Roaming\` 경로에 JVKPZv.exe 파일이 존재하는지 확인한다. 이 파일은 시스템(S) 및 숨김(H) 속성이 부여되어 일반 탐색기에서는 보이지 않으므로, 명령 프롬프트에서 `dir /a` 명령어를 통해 확인해야 한다.
- **페이로드 확인:** `%LOCALAPPDATA%\Temp\` 경로에 난독화된 텍스트로 작성된 랜덤한 이름의 .ps1 파일이 존재하는지 점검한다.
- **네트워크 확인:** 방화벽이나 DNS 로그에서 mail[.]onionmail[.]org (포트 25)로 향하는 평문 SMTP 통신 이력이나, ip-api[.]com으로 호스팅 환경을 묻는 HTTP GET 요청(`GET /line/?fields=hosting`)이 있었는지 확인한다.

## 9.2 대응

1. **네트워크 차단 (최우선 조치)**
   - 유출 서버인 mail[.]onionmail[.]org에 대한 연결을 방화벽에서 즉시 차단하여, 수집된 정보들이 공격자에게 전송되는 것을 막는다. IP가 유동적이기에 고정으로 확인된 도메인 기반의 차단을 실행한다.
2. **프로세스 종료 및 자동 실행 해제**
   - 작업 관리자 또는 Process Explorer를 통해 실행 중인 CEaN.exe 또는 JVKPZv.exe 프로세스를 종료시킨다.
   - Autoruns 또는 레지스트리 편집기를 실행하여 `HKCU\...\Run`에 등록된 JVKPZv 키 값을 삭제하여 지속성을 끊는다.
3. **악성 파일 삭제**
   - 명령 프롬프트(CLI) 등을 활용해 `%APPDATA%\Roaming\JVKPZv.exe` 복사본 파일과 `%LOCALAPPDATA%\Temp` 폴더 내의 악성 .ps1 런처 파일을 완전히 삭제한다.
4. **모든 자격증명(비밀번호) 변경**
   악성코드가 Chrome, Edge 등 브라우저 로그인 데이터와 Windows 자격증명(DPAPI 우회)을 성공적으로 읽어낸 것이 확인되었으므로, **감염된 PC에서 사용 및 저장했던 모든 웹사이트, 메일, FTP, VPN 등의 비밀번호를 즉시 변경**해야 한다.

## 9.3 예방

- **출처가 불분명한 파일(첨부파일) 실행 금지:** 해당 악성코드는 견적서, 호텔 서비스, AI 서비스로 위장하려는 특징을 보였다. 신뢰할 수 없는 파일(첨부파일)은 주의해야 한다.

- **백신 프로그램 최신화:** V3, Windows Defender와 같은 백신에서 악성코드 탐지를 할 수 있게 실시간 검색을 활성화하고 백신 내 탐지 패턴이 항상 최신화될 수 있도록 백신을 최신 상태로 유지해야 한다.
- **브라우저 비밀번호 저장 기능 최소화:** 본 분석에서 해당 악성코드는 키로깅 행위보다는 브라우저 내부 저장소(Login Data) 및 시스템에 저장된 자격증명(DPAPI 등)을 직접 타깃으로 삼아 탈취하는 것이 확인되었다. 다소 번거롭더라도 중요한 비밀번호는 브라우저 자동 저장 기능 대신, 별도의 안전한 비밀번호 관리 도구를 사용하거나 로그인 시 매번 직접 입력하는 것을 권장한다.
- **2차 인증 활성화:** 만약의 경우 계정 자격증명이 공격자에게 유출되더라도, 이를 즉시 악용할 수 없도록 주요 계정(웹사이트, 메일, VPN 등)에 대해 OTP와 같은 2차 인증 수단을 적극적으로 도입하여 활성화해야 한다.

# 부록 1. IOC 정리

## A. 파일 해시

**부록 표 1. 파일 해시**

| 종류 | 값 | 확인 경로 |
|---|---|---|
| MD5 | 82c32ffb327a0a2d20050d6db05408f0 | VirusTotal |
| SHA-1 | 76a487a95dd5aa6910fdb37aeafa824aa4edab26 | VirusTotal |
| SHA-256 | 4a4d0da0f8c4cd9a46178150f755a1348100b2c5b471874f8a898258c39a26a4 | VirusTotal |
| Imphash[26] | f34d5f2d4577ed6d9ceec516c1f5a744 | VirusTotal |

자기복사본(JVKPZv.exe)은 원본과 SHA-256 동일 = 무변형 복사. 위 해시로 복사본도 함께 탐지된다.

## B. 호스트 IOC — 파일/경로

**부록 표 2. 호스트 IOC — 파일/경로**

| 구분 | 값 | 비고 | 확인 경로 |
|---|---|---|---|
| 자기복사본 | `%APPDATA%\Roaming\JVKPZv.exe` | 관측 환경 고정 / `S·H` 속성 은폐(탐색기 비표시) | `dir /a`, `attrib`, ProcMon |
| 드롭 페이로드 | `%LOCALAPPDATA%\Temp\<랜덤>.ps1` | 파일명 가변 / 지속성 런처(본체 재실행용) | ProcMon, .ps1 디코딩 |

## C. 호스트 IOC — 레지스트리

**부록 표 3. 호스트 IOC — 레지스트리**

| 구분 | 값 | 확인 경로 |
|---|---|---|
| 지속성 (HKCU Run) | HKCU₩Software₩Microsoft₩Windows₩CurrentVersion₩Run\JVKPZv = `powershell.exe` -NoProfile -ExecutionPolicy Bypass -WindowStyle Hidden `-File "%LOCALAPPDATA%₩Temp\<랜덤>.ps1"` | Autoruns, ProcMon |

값 이름 JVKPZv는 복사본명과 동일(관측 환경 고정). 다음 로그인 시 powershell→.ps1→본체 순으로 자동 실행.

## D. 네트워크 IOC

**부록 표 4. 네트워크 IOC**

| 구분 | 값 | 비고 | 확인 경로 |
|---|---|---|---|
| 유출 서버(도메인) | mail[.]onionmail[.]org:25 | 차단 기준. 평문 SMTP | Wireshark, SmartSniff, CurrPorts |
| onionmail 관측 IP | 5[.]189[.]162[.]105 / 207[.]244[.]229[.]77 / 173[.]249[.]33[.]206 | 가변(DNS A레코드 3개 순환). 단독 차단 부적합 | Wireshark(DNS) |
| 정찰 요청 | `ip-api[.]com:80` GET /line/?fields=hosting | | Wireshark, SmartSniff |

[26] 실행파일이 불러오는 함수(Import) 목록을 해시로 만든 값. 유사 악성코드 식별·분류에 사용된다.

| 구분 | 값 | 비고 | 확인 경로 |
|---|---|---|---|
| 공격자 발신 메일 | sendboxorigin@onionmail[.]org | | Wireshark |
| 공격자 수신 메일 | originlogbox@onionmail[.]org | | Wireshark |
| 메일 제목 패턴 | PW_<username>/<computername> | | Wireshark |
| 유출 본문 시그니처 | User Name: / Computer Name: / OSFullName: / Username: / Application: (quoted-printable[27], text/html) | 행위 기반 지표 | Wireshark |

## E. 기타 분석 지표

**부록 표 5. 기타 분석 지표**

| 구분 | 값 | 확인 경로 |
|---|---|---|
| 위장 메타데이터 | HotelManager / OptiMax Quantum Suite / System Optimizer Ultimate (삼중 위장) | BinText, ProcMon |
| 타깃 정황 | BillingForm.ko-KR.resources (한국 타깃) | BinText |
| 자격증명 탈취 정황 | DPAPI(Microsoft₩Protect) + Windows Credentials + 브라우저 Login Data 접근 | ProcMon |
| 관련 패밀리 | AgentTesla (BPLogger) | VirusTotal |

[27] 8비트 데이터를 ASCII 문자로 변환해 메일 본문에 싣는 인코딩 방식.

# 부록 2. Snort 탐지룰(Snort2)

### 1. onionmail 유출 서버 IP 통신 (5.189.162.105, 207.244.229.77, 173.249.33.206)

**탐지룰**: `alert tcp $HOME_NET any -> [5.189.162.105,207.244.229.77,173.249.33.206] any (msg:"[AgentTesla] onionmail exfiltration server IP"; sid:1000001; rev:1;)`

**작성근거**: Wireshark로 관찰한 DNS 응답에서 확인한 onionmail 유출 서버 IP. 이 IP로 향하는 모든 TCP 통신을 탐지.

**오탐 가능성**: 낮음. onionmail은 정상 메일 서비스이긴 하나 익명 메일이라는 특성상 업무 통신에 쓰일 가능성이 낮다. 단, IP는 가변이므로 공격자가 서버를 바꾸면 미탐(도메인·행위 룰로 보완).

### 2. 외부 SMTP(25) 통신

**탐지룰**: `alert tcp $HOME_NET any -> $EXTERNAL_NET 25 (msg:"[AgentTesla] outbound SMTP(25) traffic"; sid:1000004; rev:1;)`

**작성근거**: AgentTesla가 SMTP(25번 포트, 메일 전송 프로토콜)로 탈취한 자격증명을 유출하는 행위를 포착. 목적지·내용 조건 없이 외부로 나가는 25번 통신 전체를 탐지.

**오탐 가능성**: 높음. 정상 메일 서버를 사용하는 환경이면 합법적 SMTP 통신도 모두 탐지된다. onionmail 외 모든 25번 통신을 잡는 광범위 룰이므로 단독 운영보다 상관분석용으로 적합.

### 3. onionmail DNS 질의

**탐지룰**: `alert udp $HOME_NET any -> $EXTERNAL_NET 53 (msg:"[AgentTesla] onionmail DNS query"; content:"onionmail"; nocase; sid:1000005; rev:1;)`

**작성근거**: 유출 서버 도메인을 DNS(53번 포트)로 질의하는 시점을 탐지. DNS 패킷 안에서 도메인은 mail, onionmail, org처럼 점(.) 단위로 잘려 저장되고 각 조각 앞에 길이 숫자가 끼어든다. 그래서 mail.onionmail.org 전체로는 매칭되지 않아, 조각 단위로 온전히 남는 onionmail을 content로 사용.

**오탐 가능성**: 낮음. 단, DNS over HTTPS(DNS 질의를 HTTPS로 암호화) 사용 시 평문 53번 질의가 없어 미탐.

### 4. ip-api DNS 질의

**탐지룰**: `alert udp $HOME_NET any -> $EXTERNAL_NET 53 (msg:"[AgentTesla] ip-api DNS query"; content:"ip-api"; nocase; sid:1000006; rev:1;)`

**작성근거**: 감염 직후 외부 IP·지리정보 정찰을 위한 ip-api 도메인 DNS 질의를 탐지.

**오탐 가능성**: 있음. ip-api는 합법 서비스라 정상 앱의 동일 질의도 탐지될 수 있음. DNS over HTTPS 사용 시 미탐.

### 5. ip-api 정찰 요청

**탐지룰**: `alert tcp $HOME_NET any -> $EXTERNAL_NET 80 (msg:"[AgentTesla] ip-api reconnaissance request"; flow:established,to_server; content:"/line/?fields=hosting"; http_uri; nocase; sid:1000007; rev:1;)`

**작성근거**: ip-api.com에 hosting 필드(지금 실행된 환경이 분석용 클라우드/호스팅 서버인지 묻는 항목)를 질의하는 정찰 URI를 탐지. 도메인 이름만 보는 게 아니라 이 검체만 쓰는 특정 요청 경로(/line/?fields=hosting)를 조건으로 잡아 정상 통신과 잘 구분되게 함.

**오탐 가능성**: 있음. ip-api는 합법 서비스라 동일 요청 형식을 쓰는 정상 앱이면 탐지 가능. HTTPS(443) 사용 시 URI가 암호화되어 미탐.

### 6. SMTP 발신주소 탐지

**탐지룰**: `alert tcp $HOME_NET any -> $EXTERNAL_NET 25 (msg:"[AgentTesla] SMTP sender address"; flow:established,to_server; content:"MAIL FROM"; nocase; content:"sendboxorigin@onionmail.org"; nocase; sid:1000008; rev:1;)`

**작성근거**: 네트워크 분석에서 확정한 유출 발신주소 sendboxorigin@onionmail.org를 탐지. MAIL FROM(발신자를 지정하는 SMTP 명령) 양식과 해당 주소가 한 패킷에 동시 출현할 때만 매칭되도록 두 content를 묶어 특이도를 높임.

**오탐 가능성**: 낮음. 단, 공격자가 발신주소를 변경하면 미탐.

### 7. SMTP 수신주소 탐지

**탐지룰**: `alert tcp $HOME_NET any -> $EXTERNAL_NET 25 (msg:"[AgentTesla] SMTP recipient address"; flow:established,to_server; content:"RCPT TO"; nocase; content:"originlogbox@onionmail.org"; nocase; sid:1000009; rev:1;)`

**작성근거**: 네트워크 분석에서 확정한 유출 수신주소 originlogbox@onionmail.org를 탐지. RCPT TO(수신자를 지정하는 SMTP 명령) 양식과 해당 주소를 묶어 매칭.

**오탐 가능성**: 낮음. 단, 공격자가 수신주소를 변경하면 미탐.

### 8. SMTP 메일 제목 패턴 탐지

**탐지룰**: `alert tcp $HOME_NET any -> $EXTERNAL_NET 25 (msg:"[AgentTesla] SMTP mail subject pattern"; flow:established,to_server; content:"Subject: PW_"; nocase; sid:1000010; rev:1;)`

**작성근거**: 자격증명 유출 메일의 고정 제목 포맷 PW_를 탐지. AgentTesla가 탈취 데이터를 보낼 때 쓰는 제목 패턴.

**오탐 가능성**: 있음. PW_로 시작하는 정상 메일 제목과 겹칠 수 있음. 공격자가 제목 포맷 변경 시 미탐.

### 9. DATA 본문 유출 탐지

**탐지룰**: `alert tcp $HOME_NET any -> $EXTERNAL_NET 25 (msg:"[AgentTesla] DATA body exfiltration"; flow:established,to_server; content:"Username:"; nocase; content:"Computer Name:"; nocase; content:"OSFullName:"; nocase; sid:1000011; rev:1;)`

**작성근거:** 유출 메일 본문(SMTP에서 실제 메일 내용이 실리는 DATA 구간)에 시스템 정보 3종(Username:, Computer Name:, OSFullName:)이 한꺼번에 나타나는 것을 탐지. 문자열 하나가 아니라 세 항목이 동시에 있어야 매칭되도록 해, 본 룰셋에서 정상 통신과 가장 확실하게 구분되는 탐지 규칙.

**오탐 가능성:** 낮음. 이 3개 필드가 한 패킷에 동시에 들어 있는 정상 통신은 드물다. 단, 공격자가 필드명을 바꾸거나 본문을 암호화하면 미탐.