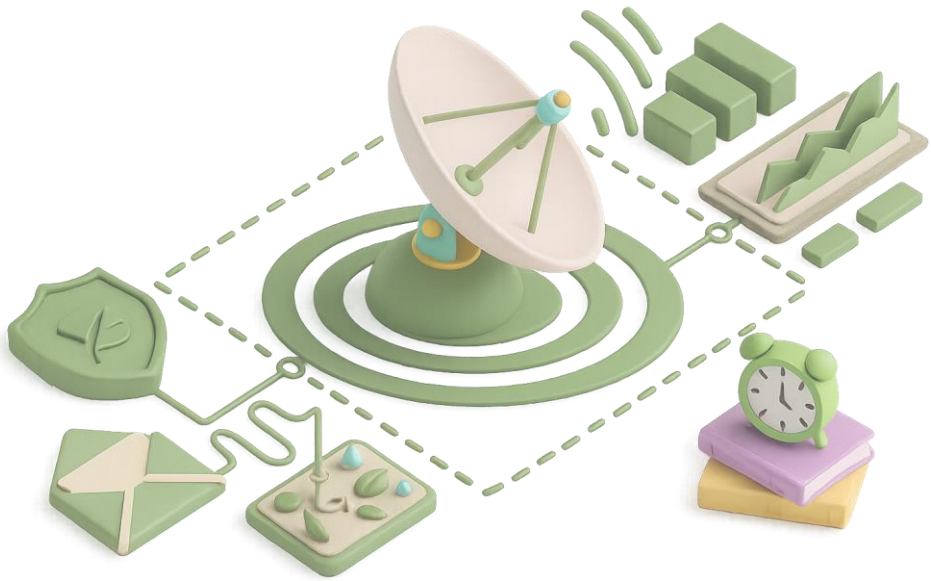
SGA 보안레이더

보안위협 리포트

Vol. 17
2026.10

10월의 악성코드 분석

DeadLock 랜섬웨어 악성코드



sga 에스지에이솔루션즈(주)

SGA보안레이더 발행본 다운로드



앞서 발행된 SGA보안레이더를 다운받으실 수 있습니다.

	주요이슈	다운로드
2025. 7.	SwaetRAT	Download
2025. 8.	링크파일 활용한 백도어 악성코드	Download
2025. 9.	Gunra Ransomware	Download
2025. 10.	LockBit 4.0 랜섬웨어	Download
2025. 11.	RokRAT	Download
2025. 12	HybridPetaya	Download
2026. 1.	DarkCloud Straler	Download
2026. 2.	문서위장 악성파일	Download
2026. 3.	와이퍼(Wiper) 악성코드	Download
2026. 4.	LiteLLM 악성코드	Download
2026. 5.	LockBit 5.0 랜섬웨어	Download
2026. 6.	카카오페이지 위장 악성코드	Download
2026. 7.	젠틀맨(Gentle men) 랜섬웨어	Download
2026. 8.	매크로를 이용한 MS워드 악성코드	Download
2026. 9.	GigaWiper 백도어 악성코드	Download

CONTENTS

발행일자: 2026년 10월

1. 26. 9월 보안 동향
2. 악성코드 통계 및 분석
3. 피싱메일 분석
4. 악성코드 분석
5. 주요 보안 뉴스



1. 2026년 9월 보안 동향

2026년 9월 보안동향을 조사한 결과 사이버 공격이 지속적으로 늘어나고 있는 것으로 나타났다.

해킹 조직 UNC6240, 인사 및 재무 업무에 쓰이는 피플소프트의 제로데이 취약점 악용

공격자들이 인사 및 재무 업무에 널리 쓰이는 오라클 피플소프트(PeopleSoft)를 노려 웹애플리케이션방화벽(WAF)을 우회하고 서버에 웹셸을 심고 있는 것으로 나타났다.

맨디언트는 샤이니헌터스와 연결된 공격 조직인 UNC6240이 취약한 환경관리허브(PSEMHUB) 경로를 막아 둔 방화벽 규칙을 우회하도록 공격 코드를 고쳤다고 밝혔다.

공격자는 경로 내 한 글자를 인코딩 문자로 변형하여 방화벽과 리버스 프록시의 문자열 대조 규칙을 우회했으며, 서버가 이를 디코딩하여 취약한 자바 서블릿으로 전달하도록 유도한 것으로 나타났다.

공격자는 웹셸 두 개를 업로드하고 정상 설치 파일로 위장한 사이드아이(SideEye) 백도어를 실행하여 외부 서버와 통신하며 서버에 저장된 자격 증명을 탈취하고 역방향 프록시를 구축한 것으로 확인된다.

북한과 연계된 사이버 공격 조직, 국내 자동차 및 미디어 기업을 겨냥한 리눅스 백도어 '테드(ted)' 유포

해외 보안 기업인 래피드7은 북한 연계 조직이 국내 자동차 및 미디어 기업을 장기적으로 감시할 목적으로 '테드(ted)'라는 백도어를 설계했다고 밝혔다.

공격자들은 그룹웨어 포털 취약점을 이용해 최초 침투한 뒤, SSH 키로거를 통해 관리자 접속 정보를 수집하여 내부 시스템으로 확장한 것으로 확인된다.이들은 악성코드 배포 전 서버 내 크론드(crond)나 HA Proxy의 설치 여부를 점검하는 스테이저를 활용해 CurlRAT를 유포한 것으로 나타났다.

테드(ted) 백도어는 부하 분산 프로그램인 HA Proxy의 맞춤형 플러그인 형태로 설치되어 정상 기능을 수행하면서 내부 API 및 프로세스 구조를 빌려 통신을 가로채고 탐지를 피한 것으로 나타났다.

이외에도 pstatic.net 등 국내 정상 CDN 도메인을 위장하거나 주요 리눅스 기본 바이너리 변조 등을 복합적으로 적용하여 공격 활동을 은폐한 것으로 확인된다.

2. 악성코드 통계 및 분석

2026년 9월 한 달 동안 사용자 PC에서 **탐지된 악성코드는 총 58,905건으로 확인**되었다.

악성코드 탐지 통계와 관련하여 가장 많이 탐지된 악성코드 유형은 Trojan으로 확인되었고, 그 뒤를 Exploit, Virus 형태의 악성코드 유형이 차지하였다. 전월과 비교해 보면 세부적으로 Exploit, Virus에 대한 탐지 비율이 증가하였다.

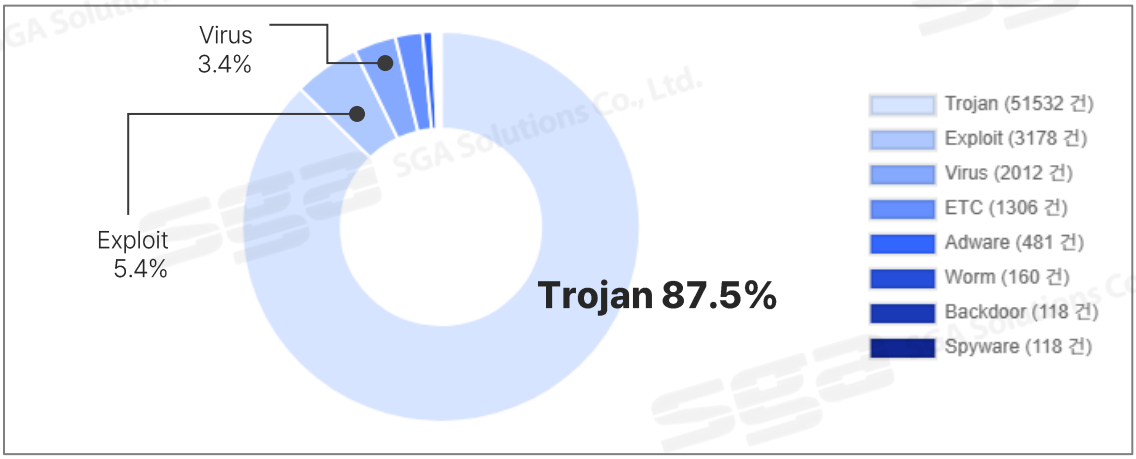
수집된 피싱 메일은 531건이었으며, 사용자를 속이기 위해 가장 많이 사용된 메일 본문 유형은 '업무 및 계약 문서 확인'이었다.

■ 유형별 탐지 통계

2026년 9월 한 달 동안 사용자 PC에서 탐지된 악성코드의 유형을 확인한 결과 **Trojan 형태의 악성코드가 51,532건(87.5%)으로 1위를 차지**했다. Ransomware의 탐지 비율은 지속적으로 높았고, 2026년 8월 대비 BitcoinMiner의 탐지 비율이 증가하였다

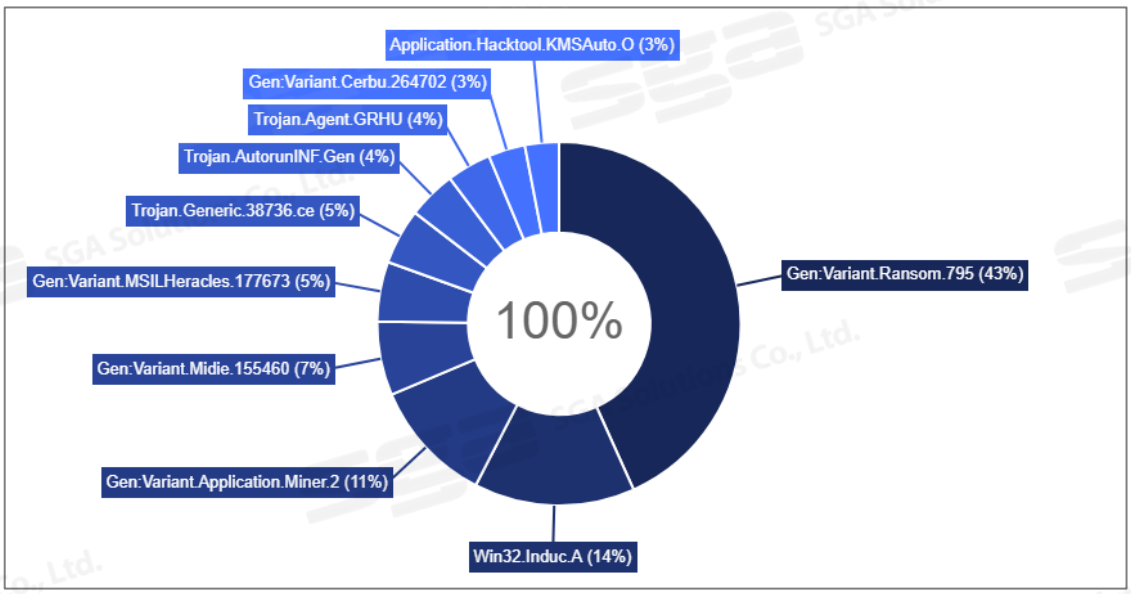
컴퓨터의 소프트웨어나 하드웨어 관련 제품의 버그, 보안 취약점 등 **설계상 결함을 이용해 공격하는 Exploit 형태의 악성코드가 3,178건(5.4%)으로 2위를 차지**했다. Exploit 형태의 악성코드 중 Hacktool의 탐지 비율이 높았고, 2026년 08월보다 KMSActivator 악성코드의 탐지 비율이 증가하였다.

마지막으로 자기 스스로는 행동할 수 없고, 정상 프로그램에 기생하여 실행되는 **Virus 형태의 악성코드가 2,012건(3.4%)으로 3위를 차지**했다. Virus 형태의 악성코드인 Induc.A의 탐지 비율은 여전히 높으며, 8월에 탐지되지 않았던 Autorun 악성코드가 탐지되었다.



[2026년 9월 유형별 탐지 통계]

■ 악성코드 TOP 10 탐지 통계



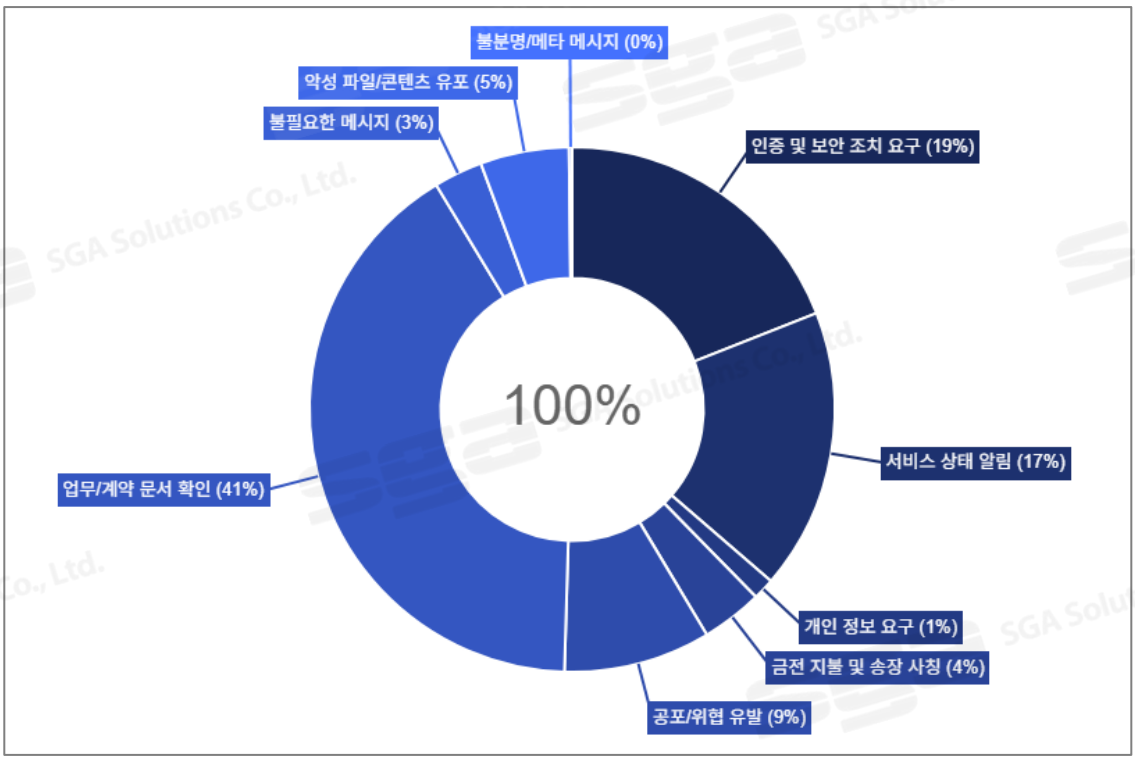
[2026년 9월 악성코드 TOP 10 탐지 통계]

2026년 9월 한 달 동안 사용자 PC에서 많이 탐지된 악성코드 중 사용자 PC의 파일을 암호화하여 사용자가 사용할 수 없게 만들며 암호화를 풀어주는 조건으로 **금전을 요구하는 악성 소프트웨어의 진단명인 Ransom.795가 1위를 차지**했다.

델파이의 특정 라이브러리가 감염된 후 컴파일 과정에서 생성되는 EXE 및 DLL 등에 **바이러스 코드가 삽입되어 악성 행위를 하는 악성코드 진단명인 Win32.Induc.A가 탐지**되었다.

3위는 사용자 PC의 자원을 무단으로 도용해 **BitCoin을 채굴하고 공격자의 지갑으로 전송하는 악성 소프트웨어인 Gen:Variant.Application.Miner.2가 탐지**되었다.

■ 피싱 메일 본문 및 공격 유형 통계



[2026년 9월 악성코드 피싱 메일 본문 유형 통계]

2026년 9월 한 달 동안 자사에 수집된 피싱 메일은 531건이며, **"업무 및 계약 문서 확인" 내용이 담긴 피싱 메일이 217건(40.9%)으로 가장 많이 수집**되었다. 사용자에게 업무 내용이 작성된 제안서를 보내 악성 파일을 실행하도록 유도하는 메일이 가장 많았으며, 그 외에도 견적서를 위장한 메일이 많았다.

그 다음으로 **"서비스 상태 알림" 내용을 담은 피싱 메일이 98건(18.5%)으로 수집**되었다.
이메일 서비스 이용 제한으로 인해 계정 로그인을 유도하거나, 비밀번호 만료 안내를 통해 본문 내 URL 클릭을 유도하는 형태의 메일이 많았다.

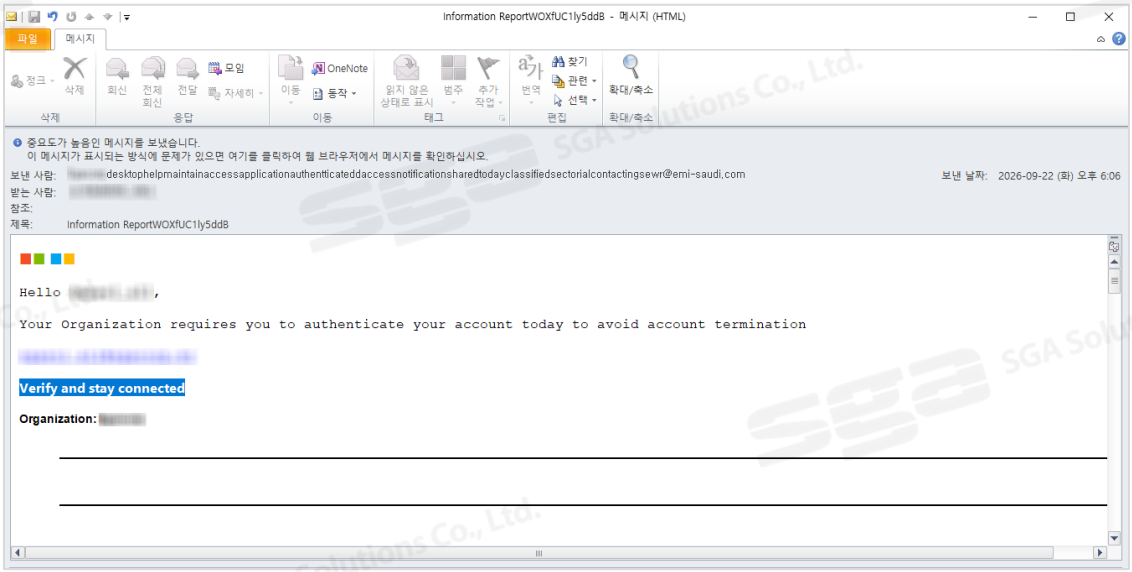
첨부 파일이 포함된 피싱 메일은 89건이 수집되었으며, 첨부 파일의 종류는 Script 파일 74건, PDF 파일 9건 그 외 6건으로 수집되었다.수집된 첨부 파일 중 대부분이 Script와 PDF 파일을 이용해 사용자를 속이는 형태의 악성코드였으며, 최종적으로는 특정 영역을 클릭하도록 유도하는 형태로 확인된다. 이러한 악성코드에는 로그인을 유도하기 위한 악성 도메인 또는 추가 악성 행위를 위해 다운로드하는 URL이 담겨 있었다.

이처럼 사용자로 하여금 클릭을 할 수밖에 없는 내용을 담은 주제로 작성되어 있어 사용자의 주의가 필요하다.

3. 피싱 메일 분석

수집된 메일 중 계약 문서 메일을 위장한 피싱 메일에 대해 분석을 진행하였으며 분석 내용은 다음과 같다.

- 메일 본문 확인



[계정 인증을 유도하는 피싱 메일 본문]

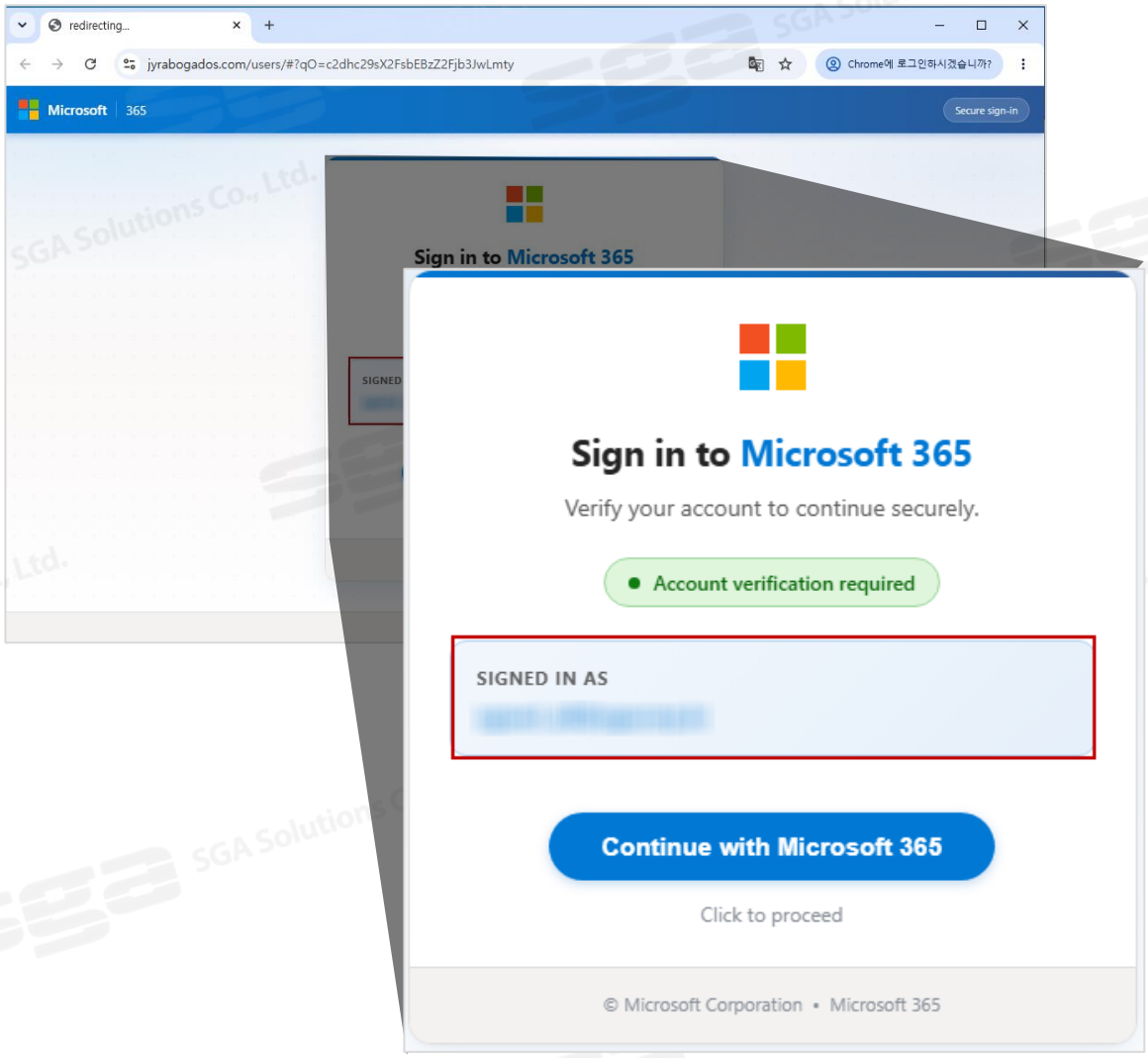
피싱 메일의 제목은 'Information ReportWOXfUC1ly5ddB'로, 정보 보고서로 위장하고 있다.

메일 본문에는 계정 해지를 방지하려면 금일 중으로 계정 인증을 완료해야 한다는 내용이 담겨 있으며, 이를 통해 사용자의 인증을 유도한다.

본문 하단의 'Verify and stay connected' 문구에는 하이퍼링크가 삽입되어 있어 링크 클릭 시 피싱 URL로 연결된다.

해당 URL은 Microsoft 365 계정 로그인 페이지로 위장한 피싱 페이지로 확인되었으며, 로그인 정보를 입력하면 해당 정보가 공격자의 C2 서버로 전송된다.

- 피싱 URL 접속 확인



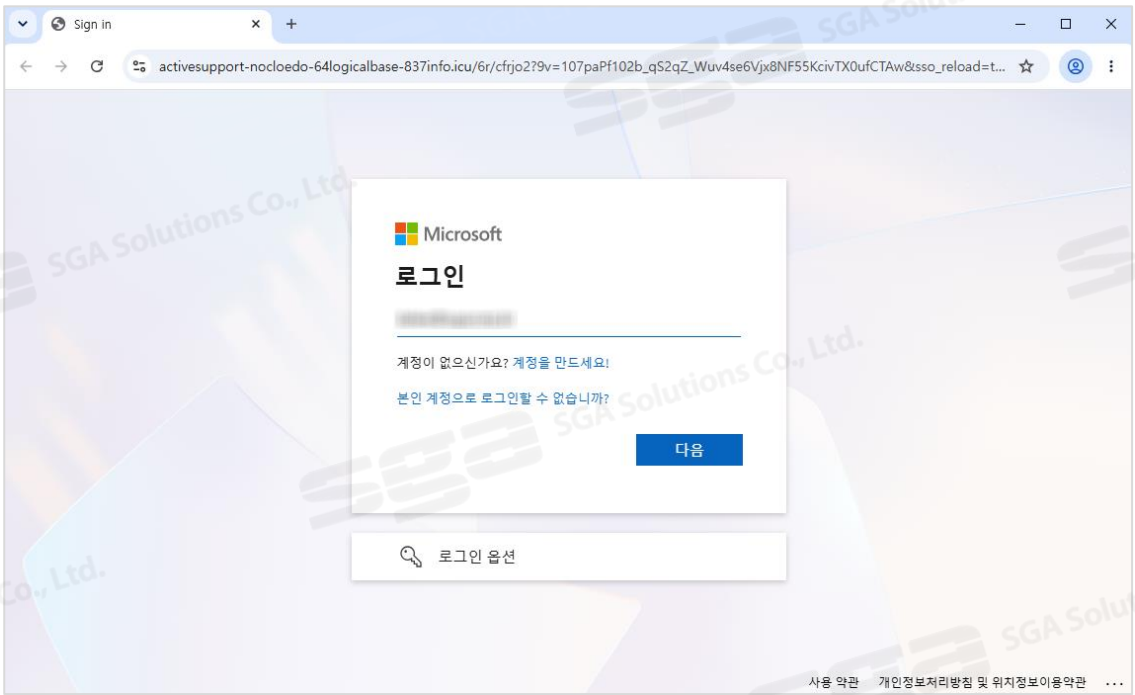
[위장된 Microsoft 365 계정 인증 페이지]

메일 본문에 삽입된 하이퍼링크에 접속하면 Microsoft 365 계정 인증을 요구하는 피싱 사이트로 연결된다.

해당 사이트는 사용자를 속이기 위해 실제 Microsoft 로고와 Microsoft 365 문구를 삽입하여 정상 로그인 페이지로 위장하고 있다.

'SIGNED IN AS' 문구 아래에는 메일 수신자의 이메일 주소가 '사용자@회사도메인' 형식으로 자동 입력되어 있으며, 하단의 'Continue with Microsoft 365' 버튼을 클릭하면 계정 정보 입력을 위한 로그인 페이지로 이동한다.

- 계정 존재 여부 검증 확인



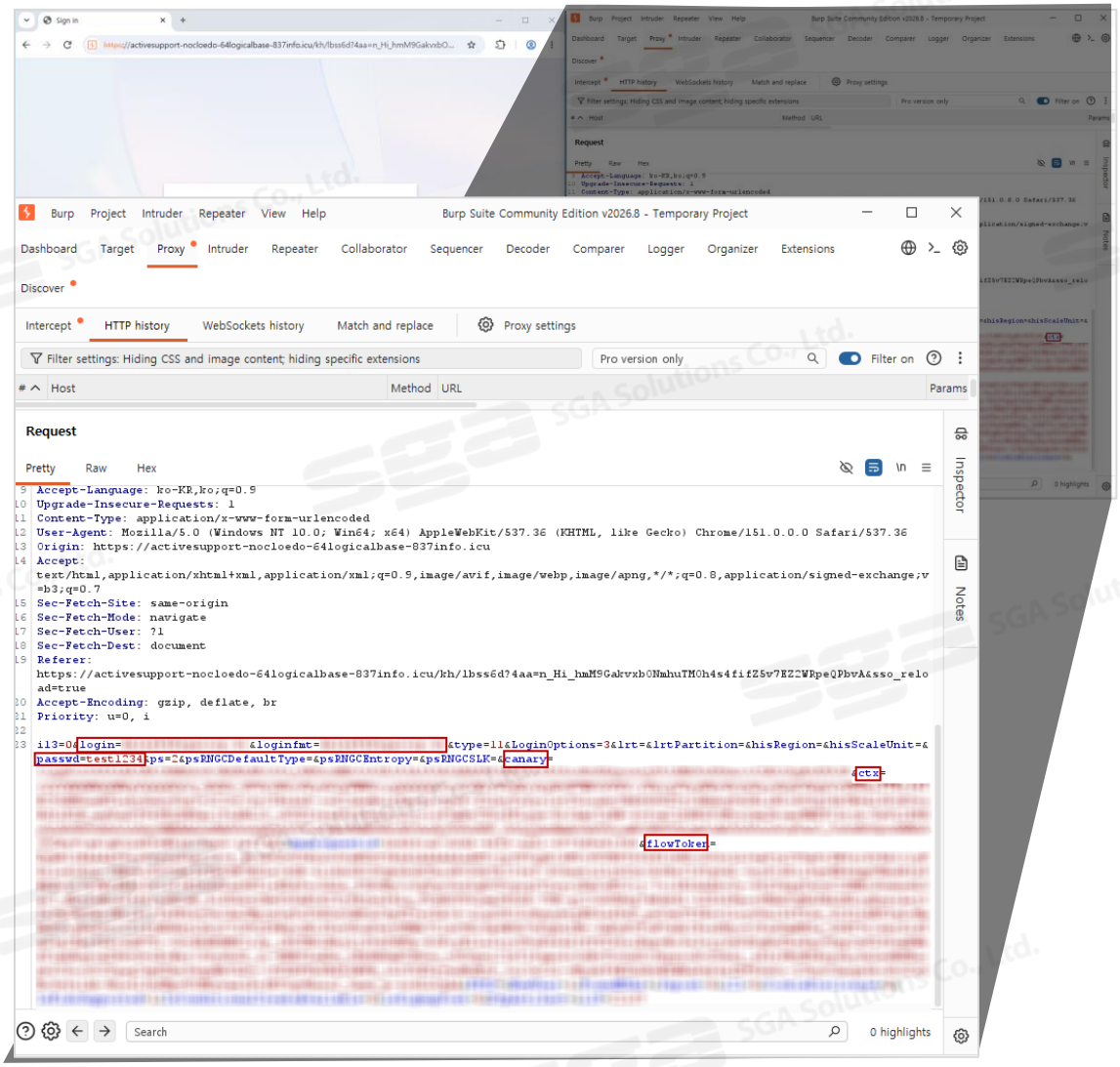
[위장된 Microsoft 365 로그인 페이지]

해당 페이지는 중간자 공격(AiTM) 방식으로 동작하는 피싱 사이트로, 앞선 페이지와 마찬가지로 수신자의 계정 주소가 자동 입력되어 있다.

입력된 계정 주소는 공격자 서버를 거쳐 실제 Microsoft 365 로그인 페이지로 전달되어 계정의 존재 여부가 검증된다.

계정이 존재하면 비밀번호 입력 단계로 넘어가고, 존재하지 않으면 '사용자 이름이 올바르지 않다'는 오류 메시지가 출력되어 더 이상 진행되지 않는다.

- 계정 비밀번호 입력 후 전송되는 데이터 확인



[위장된 Microsoft 365 비밀번호 입력 페이지 및 전송 데이터]

계정 존재 여부 검증을 통과하면 다음 페이지에서 Microsoft 365 계정의 비밀번호를 입력받는다. 입력된 계정 정보는 공격자 서버로 전송되며, 로그인 과정에 필요한 ctx, flowToken, canary 등의 값과 함께 실제 Microsoft 서버로 실시간 중계된다.

이후 사용자가 인증을 완료하면 Microsoft가 발급한 세션 쿠키까지 공격자에게 탈취되고, 공격자는 이 쿠키를 이용해 다중 인증(MFA)을 우회하고 사용자 계정에 접근하여 2차 피해를 일으킬 수 있다.

“ 배너를 클릭하고 바이러스체이서 10 AI 를 다운로드하세요!
바이러스체이서 10AI 는 무료로 사용하실 수 있습니다. ”



4. 악성코드 분석

9월에는 DeadLock 랜섬웨어 악성코드가 수집되었으며, 분석 내용은 다음과 같다.

■ 개요

DeadLock 랜섬웨어는 2025년 7월경 처음 발견된 금전적 목적의 랜섬웨어로, Rust 언어로 개발된 고도의 암호화 기법을 사용한다.

2026년 9월 기준, 이들 그룹의 데이터 유출 사이트에 등재된 피해 기관만 전 세계 40여 개국, 100곳 이상으로 집계되었다. 주요 표적은 유럽 및 북미 지역에 집중되어 있으며, 산업별로는 전문 서비스(25%), 제조(22%), IT 기술(10%) 분야의 피해 비중이 가장 높게 나타난다.

또한, 위협 동향에서 확인되는 이들의 가장 두드러진 특징은 탈중앙화된 작전 인프라(Decentralized Infrastructure)를 구축하여 운영 복원력을 극대화했다는 점이다.

랜섬노트 HTML 내부에서 기능 구현을 위한 컨트랙트와 RPC 목록이 확인되었으며, 서버가 차단되더라도 컨트랙트 값만 갱신하면 협상 채널이 복원되므로 기존의 C2 서버 차단(Takedown)만으로는 작전을 중단시키기 어렵다.

다른 랜섬웨어 그룹과 차별화되는 특징으로 엔드포인트 보안 솔루션을 우회하기 위해 정상 Baidu 백신 드라이버의 취약점(CVE-2024-51324)을 악용하는 BYOVD(Bring Your Own Vulnerable Driver) 기법을 적극적으로 활용하는 것이 확인된다.

수집된 DeadLock 랜섬웨어는 인자 값(파일 경로) 유무와 관리자 권한 여부에 따른 동작이 크게 3가지로 확인되며, 본 분석서는 인자 값 없이 관리자 권한을 부여하여 실행시킨 경우를 분석한다.

■ 주요 기능

- 파일 암호화
- 서비스 강제 종료
- 프로세스 강제 종료
- 보안 제품 무력화
- 이벤트 로그 삭제
- 권한 상승
- 배경화면 변경
- 금전 요구

■ 상세 분석

- 문자열 XOR 복호화



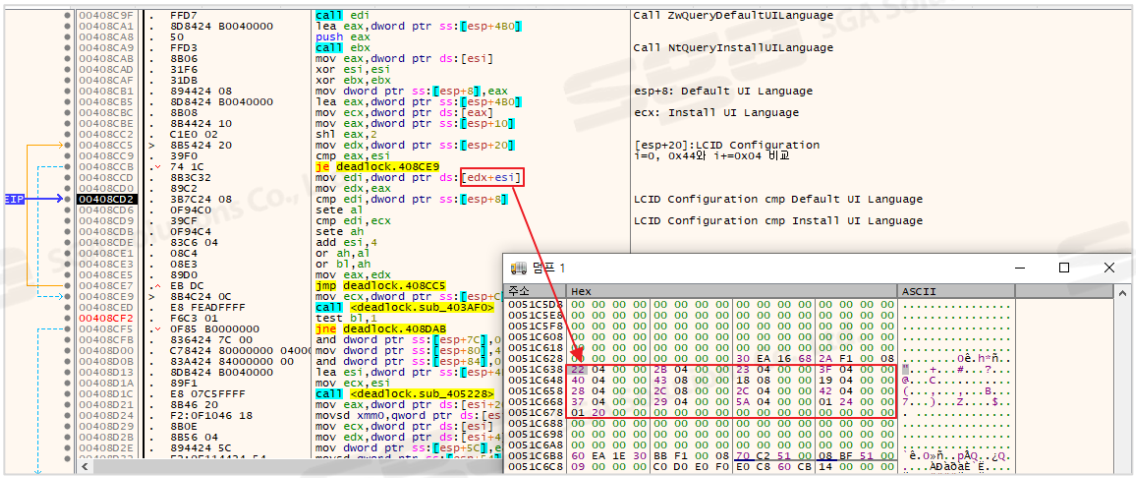
[문자열 복호화 확인]

DeadLock 랜섬웨어 실행 시 가장 먼저 8Byte 크기의 하드코딩된 키(MECMWYIS)를 이용하여 약 58KByte 분량의 데이터를 XOR 연산으로 복호화한다.

이를 통해 악성 행위에 필요한 핵심 문자열들을 메모리상에 재정의하며, 복호화 된 데이터에는 암호화 제외 대상, 랜섬노트 등이 포함된다.

또한 악성 행위에 사용될 API에 대해서 개별적으로 할당된 고유한 키 값을 이용한 실시간 XOR 복호화를 수행함으로써, 분석가의 정적 분석을 어렵게 한다.

- 대상 국가 확인



[킬스위치 기능 확인]

악성 행위를 진행 전, 랜섬웨어는 킬스위치(Kill-Switch) 조건을 확인하기 위해 NtQueryDefaultUILanguage와 NtQueryInstallUILanguage API를 사용하여 운영체제의 시스템 언어와 사용자 인터페이스 언어(LCID)를 수집한다.

이후 총 17개의 언어와 대조하며, 해당 목록은 구소련권11개, 중동권 4개, 조지아와 이란으로 구성된다.

이와 같은 언어 목록과 피해 PC의 언어를 비교하여 포함되어 있다면 즉시 종료하고 자가 삭제를 진행하며, 해당 CIS 관련 회피는 러시아어권 기반 공격자 그룹의 전형적 지표에 해당한다.

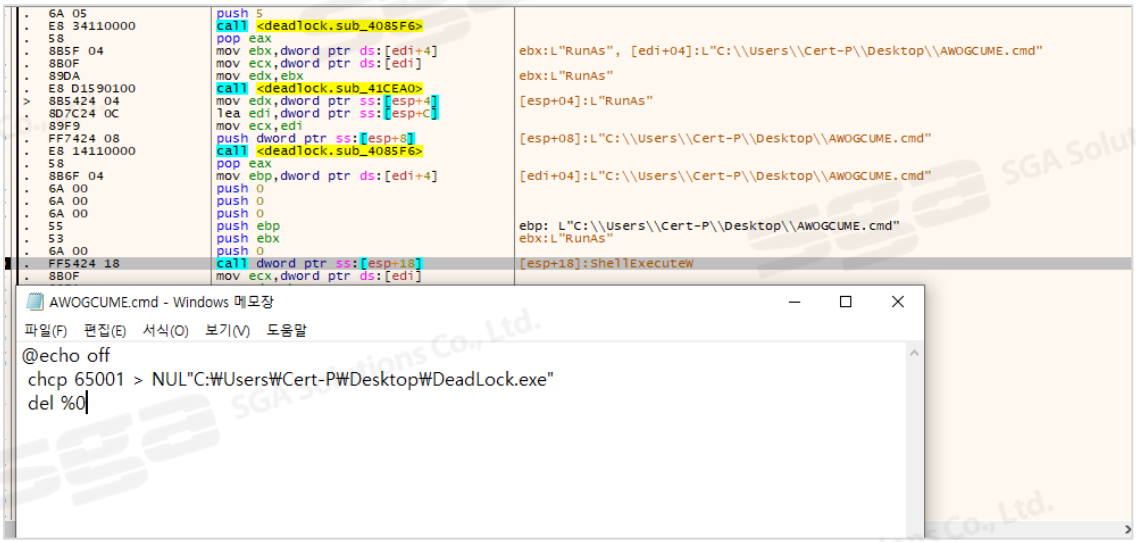
피해 PC의 언어와 대조하는 목록은 다음과 같다.

#	HEX	10진	로케일	언어
1	0x0422	1058	uk-UA	우크라이나어
2	0x042B	1067	hy-AM	아르메니아어
3	0x0423	1059	be-BY	벨라루스어
4	0x043F	1087	kk-KZ	카자흐어
5	0x0440	1088	ky-KG	키르기스어
6	0x0843	2115	uz-Cyrl-UZ	우즈베크어 (키릴)
7	0x0818	2072	ro-MD	루마니아어 (몰도바)
8	0x0419	1049	ru-RU	러시아어
9	0x0428	1064	tg-Cyrl-TJ	타지크어 (키릴)
10	0x082C	2092	az-Cyrl-AZ	아제르바이잔어 (키릴)

▼ 다음 장에 계속

#	HEX	10진	로케일	언어
11	0x042C	1068	az-Latn-AZ	아제르바이잔어 (라틴)
12	0x0442	1090	tk-TM	투르크멘어
13	0x0437	1079	ka-GE	조지아어
14	0x0429	1065	fa-IR	페르시아어 (이란)
15	0x045A	1114	syr-SY	시리아어
16	0x2401	9217	ar-YE	아랍어 (예멘)
17	0x2001	8193	ar-OM	아랍어 (오만)

- 권한 상승 - 토큰 확인 및 RunAs 기반 재실행 스크립트



[재실행 코드 확인]

랜섬웨어는 현재 실행 중인 자신의 프로세스가 관리자 권한으로 승인(UAC Elevation)되었는지 여부를 확인하기 위해, GetCurrentProcess와 OpenProcessToken, GetTokenInformation API를 사용하여 토큰의 TokenInformationClass값이 20(0x14, TokenElevation)인지 검사한다.

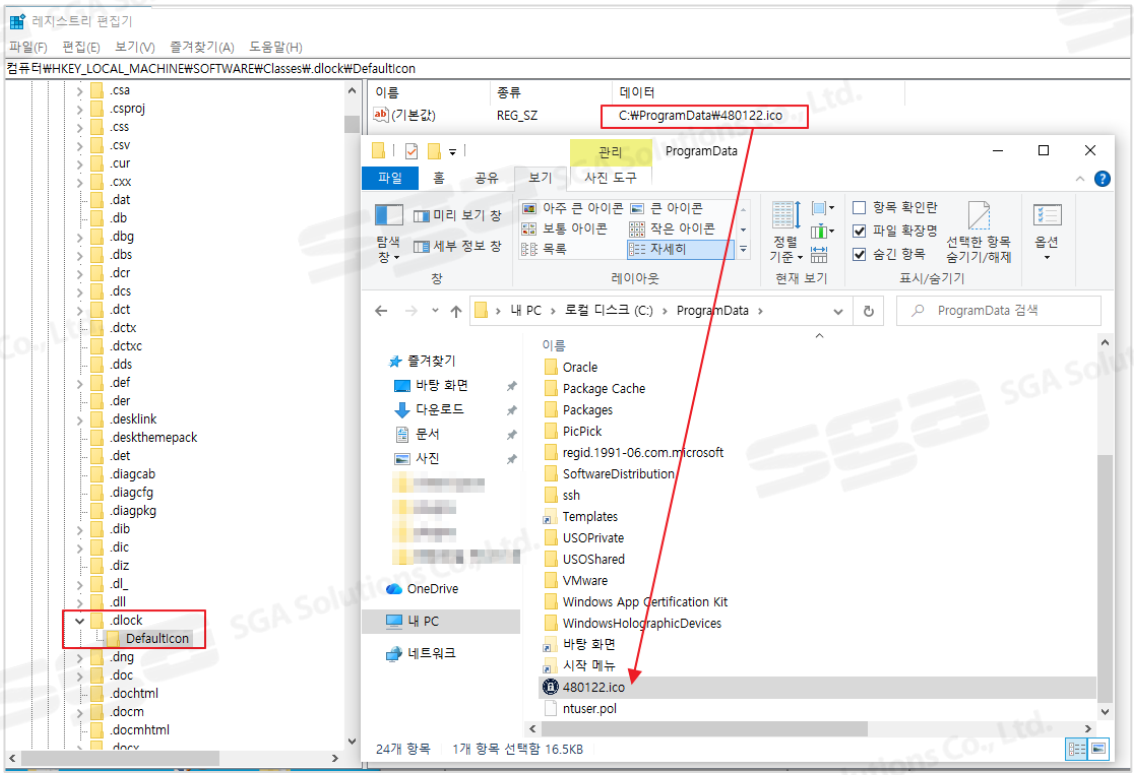
검사 결과 관리자가 아니라면, 권한 상승을 위한 임시 윈도우즈 배치 파일(예: AWOGCUME.cmd)을 생성하며, 이후 랜섬웨어는 ShellExecuteW API에 RunAs를 인자로 전달하여 해당 배치 파일을 실행한다. 이 과정은 Appinfo 서비스에 로그온 시 부여받은 사용자의 토큰 중 관리자 토큰을 배치 파일에 부여하여 실행하게 된다.

최종적으로 실행된 AWOGCUME.cmd은 DeadLock 랜섬웨어를 재실행하고 자가 삭제로 흔적을 삭제한다.

▼ 다음 장에 계속

```
@echo off   :: 명령어 실행 과정을 화면에 표시하지 않도록 숨김
chcp 65001 > NUL   ::명령 프롬프트의 인코딩을 UTF-8(65001)로 변경하고, 변경 메시지를 화면에 띄우지 않음
"<현재 존재하는 경로>\DeadLock.exe"   ::실제 존재하는 경로에 맞춰 DeadLock.exe을 실행
del %0:   ::배치 파일 자기 자신을 컴퓨터에서 삭제 (%0은 현재 실행 중인 파일 경로를 의미)
```

- 휴지통 비우기 & 아이콘 등록



[랜섬웨어 dlock 아이콘 등록]

본격적인 암호화 작업에 앞서, 랜섬웨어의 실행 효율성을 높이고 피해자의 데이터 복구를 차단하기 위하여 휴지통 비우기를 수행한다.

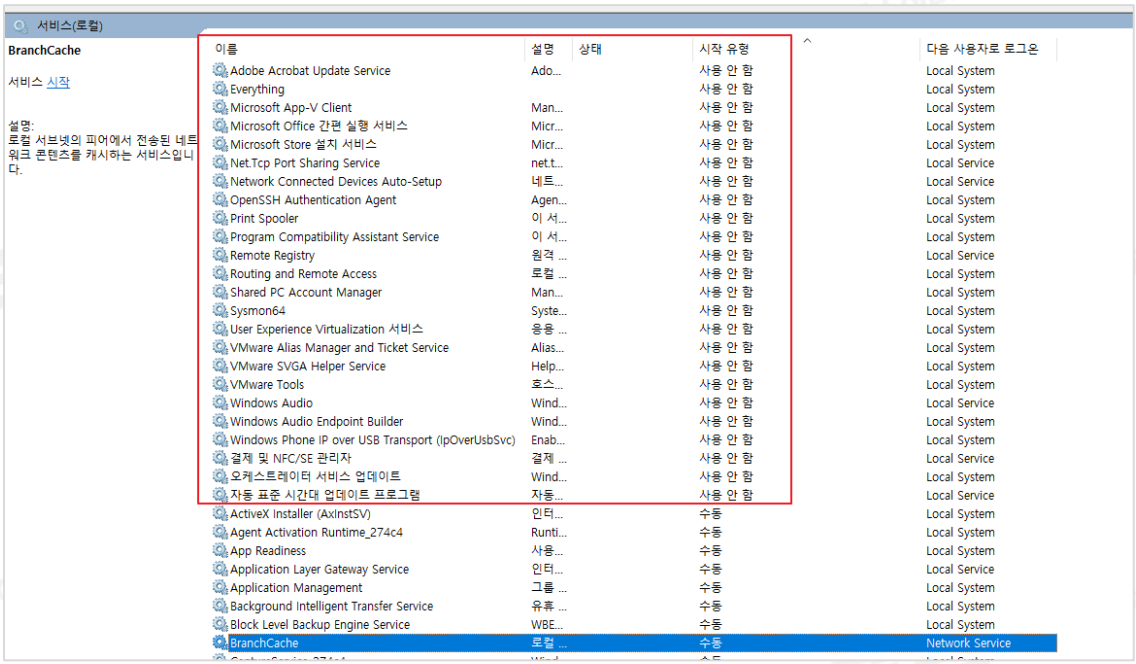
이를 위해 SHEmptyRecycleBinW(NULL, NULL, 7) API를 사용하여 시스템에 존재하는 모든 드라이브의 휴지통을 사용자의 동의 없이 일괄적으로 비운다.

이와 함께 암호화된 파일의 시각적 효과를 극대화하기 위해, 악성코드 바이너리 내부에 하드코딩되어 있는 아이콘(ico) 데이터를 바탕으로 C:\ProgramData\<UID>.ico(예: 480122.ico) 경로에 아이콘 파일을 생성한다.

이후 HKEY_LOCAL_MACHINE\SOFTWARE\Classes\.dlock\DefaultIcon에 .dlock 확장자에 대한 기본 아이콘을 앞서 생성한 아이콘 파일로 적용한다.

▼ 다음 장에 계속

- 화이트리스트 기반 서비스 및 프로세스 종료



[서비스 "사용 안 함" 변경 확인]

랜섬웨어의 파일 암호화에 앞서 원활한 악성 행위 수행을 위해 화이트리스트 방식을 적용하여 종료할 시스템 내 서비스와 프로세스를 선별한다. 여기서 적용된 화이트리스트 방식은 공격자가 사전에 준비한 목록을 제외한 나머지 모든 항목을 공격 대상으로 삼는 기법을 말한다.

먼저 서비스를 대상으로 OpenSCManagerA 및 EnumServicesStatusExW API로 구동 중인 전체 서비스를 열거하고, lpServiceName API로 이름을 추출한 서비스 이름을 소문자로 변환하여 216개의 목록과 대조한다. 화이트리스트 방식 기반으로 일치하면 동작은 유지하지만, 불일치할 경우 OpenServiceW API로 대상 서비스의 핸들 값을 얻은 후 ChangeServiceConfigW, ControlService, CloseServiceHandle API를 사용하여 시작 유형을 '사용 안 함'으로 변경한다. 실제 <서비스 "사용 안 함" 변경 확인>과 같이 VM TOOL이나 Everything 서비스는 목록에 존재하지 않아 '사용 안 함'으로 변경된 것을 확인할 수 있다.

프로세스 역시 서비스와 동일한 방식으로 제어되며, CreateToolhelp32Snapshot, Process32First, Process32Next API를 통해 현재 실행 중인 프로세스의 파일명을 수집하고 이를 화이트리스트 목록과 대조한다. 대조 결과 화이트리스트에 존재하지 않는 프로세스가 발견되면 OpenProcess API를 통해 핸들을 얻고 TerminateProcess API를 호출하여 즉각 강제 종료를 수행한다.

서비스 화이트리스트 216개는 다음과 같다.

▼ 다음 장에 계속

분류	서비스
사용자별 서비스 관련	UdkUserSvc*, DevicesFlowUserSvc*, AarSvc*, WpnUserService*, CDPUserSvc*, cbdhsvc*, UserDataSvc*, PimIndexMaintenanceSvc*, UnistoreSvc*, OneSyncSvc*
Hyper-V, 가상화 관련	vmicguestinterface, vmicheartbeat, vmickvpexchange, vmicrdv, vmicshutdown, vmictimesync, vmicvss, vmicvmsession, Vmcompute, Vmms, nvspwmi, vhdsvc, Hvhost
검색, 인덱싱, 저장소 관련	MSSEARCH, Wsearch, PimIndexMaintenanceSvc, Fontcache*, Storsvc, Vss, Swprv, Vds
Windows 업데이트, 설치, 배포 관련	WsusService, WaaSMedicSvc, Wuauserv, Trustedinstaller, msiserver, DeviceInstall, Appxsvc, ClipSVC, Dosvc, DusmSvc, Wudfsvc
네트워크, 연결 관련	ALG, PeerDistSvc, Eaphost, WlanSvc, WwanSvc, dot3svc, server, workstation, Lanmanserver, Lanmanworkstation, Netman, Netprofm, Netsetupsvc, WZCSVC, Nlasvc, Nla, SharedAccess, Nsi, lltdsvc, wcncsvc, Webclient, Winhttpautoproxysvc
보안, 인증, 정책 관련	WinDefend, Bfe, Certpropsvc, Certsvc, Cryptsvc, Efs, NtLmSsp, Ikeext, Keyiso, Kpssvc, Kdc, Kdssvc, Mpssvc, Policyagent, Samss, Seclogon, Securityhealthservic, SgrmBroker, Vaultsvc
디렉터리, 도메인, 파일 서비스 관련	Adws, adfssrv, Dfs, Dfsr, NfsClnt, NfsService, Dhcp, Dhcpserver, Dns, Dnscache, Dnsproxy, Netlogon, Ntds, Ntfrs, SmbWitness, Clussvc, WINS, Lmhosts
원격 접속, 터미널 서비스 관련	Remoteaccess, Remoteregistry, Termservice, Tssdis, TScPubRPC, TermServLicensing, Tsgateway, Umrdpservice, Sessionenv, Rasauto, Rasman, Sstpsvc, RDMS
시스템 핵심, 이벤트, RPC 관련	Aelookupsvc, Apphostsvc, Appinfo, Appmgmt, Brokerinfrastructure, Coremessagingregistrar, Dcomlaunch, Eventlog, Eventsystem, Lsm, LSM, Rpcss, Rpceptmapper, Rpchttplbs, Schedule, Sens, Systemeventsbroker, Power, Profsvc, Plugplay, UPlugPlay
장치, 사용자 환경, UI 관련	Deviceassociationservice, TabletInputService, DispBrokerDesktopSvc, Hidserv, Lfsvc, Licensemanager, LicenseService, NgcSvc, NgcCtnrSvc, Usermanager, Uxsms, Themes, Tiledatamodelsvc, Shellhwdetection, Stisvc, Wpdbusenum, Wpnservice
데이터 수집, 진단, 관리 관련	Diagtrack, Dps, Dsmsvc, Dssvc, dmwappushservice, dmserver, Ramgmtsvc, SrmReports, SrmSvc, WSRM, Wdiservicehost, Winmgmt, Winrm, Wmiapsrv, W32time, Timebroker, Timebrokersvc
응용 프로그램, 서드파티, 기타 관련	Anydesk, dbxsvc, dbupdate, dbupdatem, FileSyncHelper, LxpSvc, OneDrive Updater Service, Browser, Bthavctpsvc, camsvc, Cdpsvc, CscService, ddpvssvc, MMCSS, Msmq, Ncbservice, ProtectedStorage, rpcapd, sexesvc, silsvc, smphost, Staterepository, Sysmain, Tapisrv, Tokenbroker, Trkwks, Ualsvc, uhssvc, Upnphost, Was, Wcmsvc, wscsvc, Wlidsvc

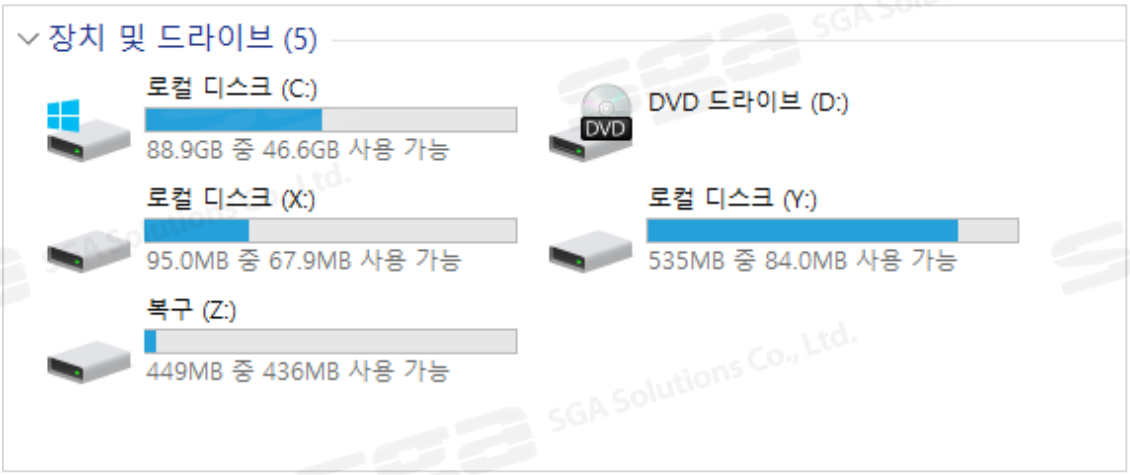
▼ 다음 장에 계속

프로세스 화이트리스트 기반 목록은 총 80개로 다음과 같다.

항목	비고
시스템 핵심, 종료 불가	conhost, csrss, dllhost, idle, lsaiso, lsass, lsm, registry, rundll32, services, smss, svchost, svhost, system, wininit, winlogon, wmiprvse, wudfhost
셸, UI	applicationframehost, ctfmon, dwm, explorer, fontdrvhost, logonui, runtimebroker, shellexperiencehost, shellhost, sihost, taskhostw, wwahost
검색 인덱싱	searchapp, searchfilterhost, searchhost, searchindexer, searchprotocolhost, searchui
공격자 도구, 원격 접속	anydesk, cmd, mstsc, netscan, openconsole, powershell, putty, rclone, rustdesk
클라우드 동기화 (백업 오염 경로)	dropbox, filecoauth, googledrivefs, onedrive, onedrive.sync.service, owncloud
MS 보안	msmpeng, nissrv, securityhealthservice, smartscreen
백업, 스냅샷	vssvc
도메인, 서버 역할	certsrv, clussvc, dfsrs, dfssvc, dns, ismserv, microsoft.activedirectory.webservices, msdtc, rhs
Hyper-V, 가상화	vds, vmcompute, vmms, vmwp
기타, Windows 내장, 응용	aggregatorhost, dashost, filesynchelper, kihost, outlook, rdpclip, spoolsv, taskmgr, trustedinstaller, windowsterminal, wlanext

▼ 다음 장에 계속

- 은닉 볼륨 마운트 시도



[은닉 볼륨 마운트 결과]

DeadLock 랜섬웨어는 암호화 피해 범위를 극대화하기 위해 시스템 내에 존재하는 은닉 볼륨을 탐색하여 강제 마운트한다.

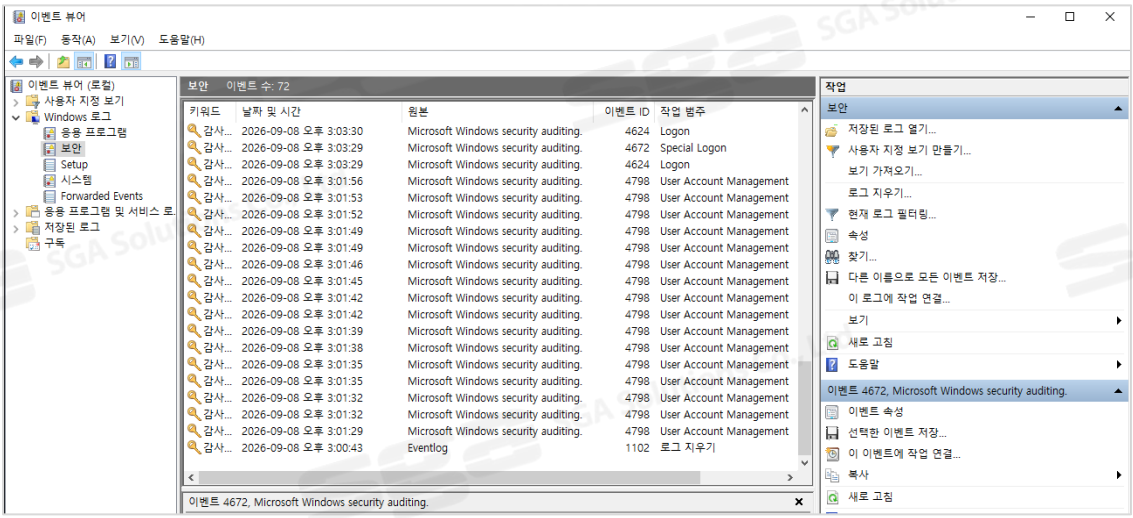
우선 FindFirstVolumeW와 FindNextVolumeW API를 사용하여 볼륨을 찾은 후, GetVolumePathNamesForVolumeNameW API로 해당 볼륨의 마운트 경로의 존재 여부를 확인한다.

마운트되지 않은 볼륨이 확인되면 SetVolumeMountPointW API를 사용하여 Z부터 A까지 역순으로 비어 있는 드라이브 문자를 할당하여 강제 마운트를 수행한다.

이러한 과정에서 볼륨 관련 별도의 예외 처리가 적용되지 않아, 숨겨진 복구 파티션이나 운영체제 부팅에 필수적인 시스템 볼륨 영역까지 마운트되어 암호화되므로 운영체제의 부팅 기능이 파괴되는 치명적인 결과를 초래한다.

▼ 다음 장에 계속

- 윈도우 이벤트 로그 삭제, 접근 권한 변경



[이벤트 로그 삭제 결과]

DeadLock 랜섬웨어의 안티 포렌식 기능은 구형 API, 최신 API, 레지스트리 권한 변조를 통해 시스템의 모든 로깅을 파괴하고 접근을 영구 차단하도록 설계된 것으로 알려져 있다.

반면에 현재 분석 샘플에서는 다소 다르게 동작하는 것이 확인되었다.

먼저 구형 이벤트 로깅 API(advapi32의 OpenEventLogW와 ClearEventLogW)로 내장 9개 채널(Application, Security, Setup, Servicing, Eventlog, Forwarded Events, Windows PowerShell, System, HardwareEvents)과 열거된 전체 채널의 로그를 삭제한다.

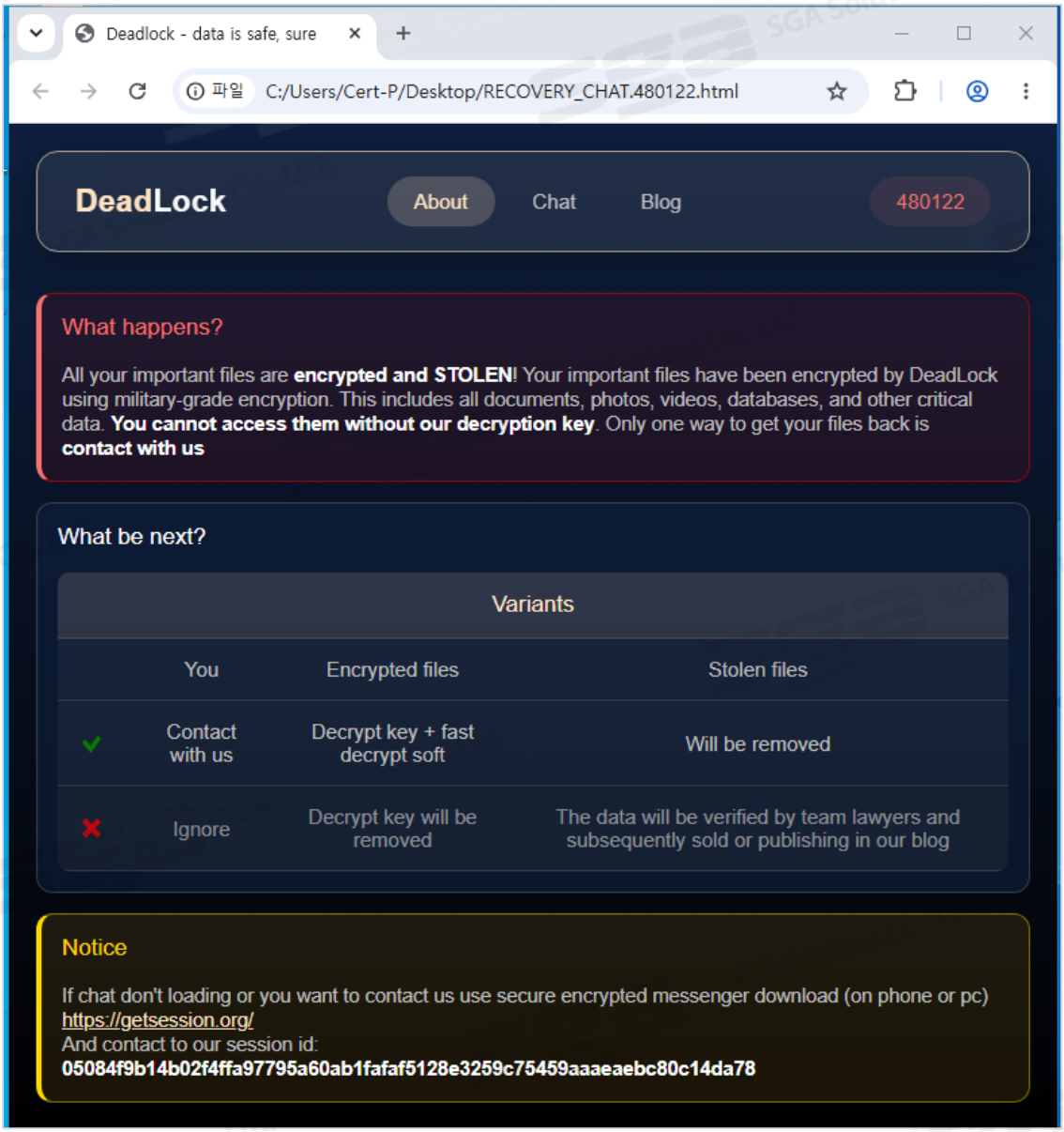
다만 이 API는 구형 로그만 인식하고 wevtapi는 채널 이름 열거에만 사용되므로, 최신 운영 채널은 삭제되지 않았을 가능성이 존재한다.

또한, 레지스트리를 통한 로깅 비활성화(Enabled)는 데이터 포인터가 NULL로 전달되어 실패하는 것으로 확인되었다.

접근 권한 변경(ChannelAccess)은 기록 권한이 제거된 SDDL(Security Descriptor Definition Language)이 정상적으로 전달되는 시도지만, 영향에 대해서는 추가적인 확인이 필요하다.

▼ 다음 장에 계속

- 랜섬노트 생성 - html



[브라우저에서 확인한 HTML 랜섬노트]

파일 암호화 작업이 완료되면 악성코드는 모든 드라이브의 최상위(Root) 디렉터리와 모든 사용자 바탕화면에 RECOVERY_CHAT.<UID>.html 파일을 일괄 생성한다.

해당 파일은 단순한 협박문이 아니라, Polygon 블록체인의 스마트 컨트랙트에서 협상 프록시 주소를 조회해 Session 메신저와 연결하는 자체 완결형 협상 클라이언트이다.

▼ 다음 장에 계속

- 암호화 대상 탐색

```
attr_gate_pass = check_file_attributes((int *)&work_str,1);
if (attr_gate_pass != 0) {
  if ((file_attr & 0x10) == 0) {
                /* 경로를 소문자로 변환 (대소문자 변경으로 회피 불가) */
    to_lowercase(&work_str,(int)path_ptr,path_len);
                /* [1차] 확장자·파일명 제외 목록 72개 대조 -> 해당 시 제외
                   */
    is_excluded = (**(code **)(ml_ext_exclude[1] + 0x14))(*ml_ext_exclude,&work_str);
    if (is_excluded == '\0') {
      to_lowercase(&work_str,(int)path_ptr,path_len);
                /* [2차] 경로 제외 목록 141개 대조, 해당 시 제외(파일은
                   1·2차 모두 통과해야 암호화) */
      is_excluded = (**(code **)(m0_path_exclude[1] + 0x14))(*m0_path_exclude,&work_str);
      if (is_excluded == '\0') {
        task_counter = (int *)*ctx_saved;
        LOCK();
        iVar4 = *task_counter;
        *task_counter = *task_counter + 1;
        UNLOCK();
        if (*task_counter == 0 || SCARRY4(iVar4,1) != *task_counter < 0) {
                /* WARNING: Does not return */
          pcVar1 = (code *)invalidInstructionException();
          (*pcVar1)();
        }
        local_460 = pathcopy_len;
        work_str = pathcopy_cap;
        uStack_464 = SUB42(pathcopy_ptr,0);
        uStack_462 = (undefined2)((uint)pathcopy_ptr >> 0x10);
        local_45c = SUB41(task_counter,0);
        uStack_45b = (undefined1)((uint)task_counter >> 8);
        uStack_45a = (undefined2)((uint)task_counter >> 0x10);
                /* 암호화 큐 투입 */
        (**(code **)(m2_encrypt_queue[1] + 0x14))(*m2_encrypt_queue,&work_str,task_counter);
        goto LAB_004081fa;
      }
    }
  }
  else {
    to_lowercase(&work_str,(int)path_ptr,path_len);
                /* 디렉터리 경로 제외 목록만 대조, 통과 시 하위로 재귀 */
    is_excluded = (**(code **)(m0_path_exclude[1] + 0x14))(*m0_path_exclude,&work_str);
    if (is_excluded == '\0') {
```

[암호화 대상 판정 로직]

암호화 대상도 화이트리스트 방식으로 파일의 제외 목록을 이용하며, 경로와 파일명에 특정 문자열이 포함되었는지 검사한다.

이러한 방식은 미리 시스템 유지에 필요한 핵심 영역을 제외 목록으로 설정하여 시스템의 치명적 손상을 방지한다.

또한 모든 문자열 판정 로직은 검사 직전에 완전 유니코드 기반의 소문자 변환 처리를 거치며 대소문자 가리지 않고 대상을 확정한다.

먼저 경로의 경우 총 47개 항목이 존재한다. 하지만 여기서 실질적으로 사용되는 항목은 다음과 같은 중복을 제거한 11개로 확인된다.

▼ 다음 장에 계속

제외 경로

AESDecrypt(cipherHex, keyHex) string
$windows.~bt, $windows.~ws, Users\All Users\application data, config.msi, system volume information, users\all users\microsoft, windows\, users*\appdata, program files (x86)\, program files\, programdata\

다음 단계로 파일명에 대한 특정 문자열들로 총 72개이며, 각각 확장자 관련 51개와 시스템 파일명 15개, 실시간 자기 산출물 6개로 구성된다.

이 중 자기 산출물은 고유 ID(UID)를 포함하는 항목들로 암호화에 랜섬노트, 바탕화면 이미지, 암호화 확장자, 암호화 관련 소통을 위한 HTML 파일로 이뤄져 있다.

다만, 항목 중 중복되는 값들이 확인되며, 중복 값을 제외하면 총합은 72개가 아닌 70개가 된다.

실시간 자기산출물 4개(중복 값 제거)

•UID(480122).dlock, UID(480122).bmp, UID(480122).txt, UID(480122).html

파일 명 15개

\GDIPFONTCACHEV1.DAT, \bootnxt, \ntldr, \autorun.inf, \bootsect.bak, \ntuser.dat, \boot.ini, \d3d9caps.dat, \ntuser.dat.log, \bootfont.bin, \desktop.ini, \ntuser.ini, \bootmgr, \iconcache.db, \thumbs.유

확장자 51개

.386, .hlp, .msu, .adv, .hta, .nls, .ani, .icl, .nomedia, .bat, .icns, .ocx, .bin, .ico, .pdb, .cab, .ics, .prf, .cmd, .idx, .ps1, .com, .key, .rom, .cpl, .ldf, .rtp, .cur, .lnk, .scr, .deskthemepack, .lock, .search-
ms, .diagcab, .mod, .shs, .diagcfg, .mpa, .spl, .diagpkg, .msc, .sys, .dll, .msi, .theme, .drv, .msp, .themepack, .exe, .msstyles, .wpx

▼ 다음 장에 계속

- 파일 크기별 암호화 배치

```
00411974 f2 0f 58 c8     ADDSD      XMM1,XMM0
00411978 f2 0f 5e        DIVSD      XMM1,qword ptr [LAB_00425260]          / 100.0   백분율 환산
         0d 60 52
         42 00
00411980 f2 0f 59 cc     MULSD      XMM1,XMM4                              x r   r = 선택된 비율 (100/50/25/10)
00411984 83 ec 08        SUB        ESP,0x8                                [기호] N = 원본 파일 크기 (byte)
00411987 f2 0f 11        MOVSD      qword ptr [ESP]=>local_e88,XMM1        ceil 인자 전달 = N x r / 100
         0c 24
0041198c e8 e0 1a        CALL       MSVCRT.DLL::ceil                       [1] E = ceil(N x r / 100)   암호화할 총 바이트
         01 00
00411991 58              POP        EAX
00411992 59              POP        param_1
00411993 dd 9c 24        FSTP       double ptr [ESP + local_a48]           E 저장
         38 04 00 00
0041199a f2 0f 10        MOVSD      XMM0,qword ptr [ESP + local_a48]
         84 24 38
         04 00 00
004119a3 66 0f 28        MOVAPD     XMM1,xmmword ptr [ESP + local_e60[0]]  N 로드 (원본 파일 크기)
         4c 24 20
004119a9 f2 0f 5c c8     SUBSD      XMM1,XMM0                              N - E   평문으로 남길 총 바이트
004119ad f2 0f 59        MULSD      XMM0,qword ptr [LAB_00425350]          E / 512   암호화 단위는 512 byte 고정
         05 50 53
         42 00
004119b5 66 0f 29        MOVAPD     xmmword ptr [ESP + local_e60[0]],XMM1
         4c 24 20
004119bb 83 ec 08        SUB        ESP,0x8                                [기호] C = 512 byte 청크 개수
004119be f2 0f 11        MOVSD      qword ptr [ESP]=>local_e88,XMM0
         04 24
004119c3 e8 a9 1a        CALL       MSVCRT.DLL::ceil                       [2] C = ceil(E / 512)   암호화 블록 개수
         01 00
004119c8 58              POP        EAX
004119c9 59              POP        param_1
004119ca dd 9c 24        FSTP       double ptr [ESP + local_a40]           C 저장
         40 04 00 00
004119d1 66 0f 28        MOVAPD     XMM0,xmmword ptr [ESP + local_e60[0]]  (N - E) 로드
         44 24 20
004119d7 f2 0f 5e        DIVSD      XMM0,qword ptr [ESP + local_a40]       (N - E) / C   평문량을 청크수로 분배
         84 24 40
         04 00 00
004119e0 83 ec 08        SUB        ESP,0x8                                [기호] S = 청크 사이 건너뛸 간격 (byte)
004119e3 f2 0f 11        MOVSD      qword ptr [ESP]=>local_e88,XMM0
         04 24
004119e8 e8 8a 1a        CALL       MSVCRT.DLL::floor                      [3] S = floor((N - E) / C)   건너뛸 간격 확정
         01 00
004119ed 58              POP        EAX
004119ee 59              POP        param_1
004119ef dd 9c 24        FSTP       double ptr [ESP + local_a38]           S 저장 -> 이후 파일 seek 에 사용
         48 04 00 00
```

[파일 크기별 암호화 구간 산출 코드]

랜섬웨어의 가장 중요한 기능인 암호화 효율을 높이기 위해 파일 크기별로 암호화 구간을 다르게 설정한다.

기본 암호화 단위는 512 Byte로 고정되나, 대상 파일의 크기별로 암호화되는 비율과 청크 사이 얼마나 건너뛰는지가 다른 것으로 확인된다.

해당 샘플은 다섯 구간으로 나뉘며, 약 50MB 미만은 암호화 비율 100%와 약 118MB 미만은 비율 50%, 약 500MB 미만은 비율 25%, 약 1GB 미만은 10%, 약 1GB 이상은 청크 모드로 분기하여 비율 계산을 수행하지 않고, 선두 약 49.5MB만 순차 암호화한다.

부분 암호화 비율이 적용되는 각 구간에서는 암호화 총량 E=ceil(크기x비율/100), 청크 개수 C=ceil(E/512), 건너뛰기 간격 S=floor((크기-E)/C), 청크 위치는 반개구간 [i×(512+S), i×(512+S)+512)에 할당된다.

▼ 다음 장에 계속

- 파일 암호화 알고리즘

```
                    a += b   (b는 여기서만 state[idx_b] 직접 참조)
0040deac 01 fe          ADD         a,EDI                                   a += b
0040deae 31 f0          XOR         d,a                                     d ^= a
                    [1/4] d <<<= 16
0040deb0 c1 c0 10       ROL         d,16                                    [1/4] d <<<= 16    ChaCha20 회전 (Salsa20 이면 7)
0040deb3 01 c1          ADD         c,d                                     c += d
0040deb5 31 cf          XOR         b_in,c                                  b ^= c
                    [2/4] b <<<= 12   -- 이후 b_in 은 b_rot(회전된 b) 값을 유지
0040deb7 c1 c7 0c       ROL         b_rot,12                                [2/4] b <<<= 12    ChaCha20 회전 (Salsa20 이면 9)
0040deba 01 fe          ADD         a,b_rot                                 a += b
0040debc 89 34 9a       MOV         dword ptr [idx_b + EBX*0x4],a
0040debf 31 c6          XOR         a,d                                     d ^= a
0040dec1 8b 44 24 1c    MOV         d,dword ptr [ESP + idx_d]
                    [3/4] d <<<= 8
0040dec5 c1 c6 08       ROL         a,0x8                                   [3/4] d <<<= 8     ChaCha20 회전 (Salsa20 이면 13)
0040dec8 89 34 82       MOV         dword ptr [idx_b + d*0x4],a
0040decb 8b 04 24       MOV         d,dword ptr [ESP]=>local_14
                    c += d   -- 변수 d 재사용: 이 줄부터 d 는 c 역할
0040dece 01 ce          ADD         a,c                                     c += d
0040ded0 89 34 aa       MOV         dword ptr [idx_b + EBP*0x4],a
                    b ^= c   -- 변수 d 재사용: 이 줄부터 d 는 b 역할
0040ded3 31 fe          XOR         a,b_rot                                 b ^= c
                    [4/4] b <<<= 7
0040ded5 c1 c6 07       ROL         a,0x7                                   [4/4] b <<<= 7     ChaCha20 회전 (Salsa20 이면 18)
0040ded8 89 34 82       MOV         dword ptr [idx_b + d*0x4],a
0040dedb 83 c4 04       ADD         ESP,0x4
0040dede 5e             POP         a
0040dedf 5f             POP         b_rot
0040dee0 5b             POP         EBX
0040dee1 5d             POP         EBP
0040dee2 c3             RET
```

[ChaCha20 쿼터라운드 확인]

임시 개인키는 CryptGenRandom API로 32 Byte 난수를 파일마다 새로 생성한다. 이후 이를 설정 블록에 내장된 공격자 공개키와 결합하여 secp256k1 곡선 기반 ECDH 연산을 거쳐 공통 키를 산출한다. 이렇게 생성된 공통 키는 파일 암호화에 사용된 키를 암호화하는 데 사용된다.

실제 파일 암호화 과정에서는 ChaCha 계열 알고리즘 특유의 쿼터라운드(16, 12, 8, 7)와 더블 라운드 10회 반복(총 20라운드)이 확인되며, 이를 통해 ChaCha20 알고리즘이 사용된 것을 알 수 있다.

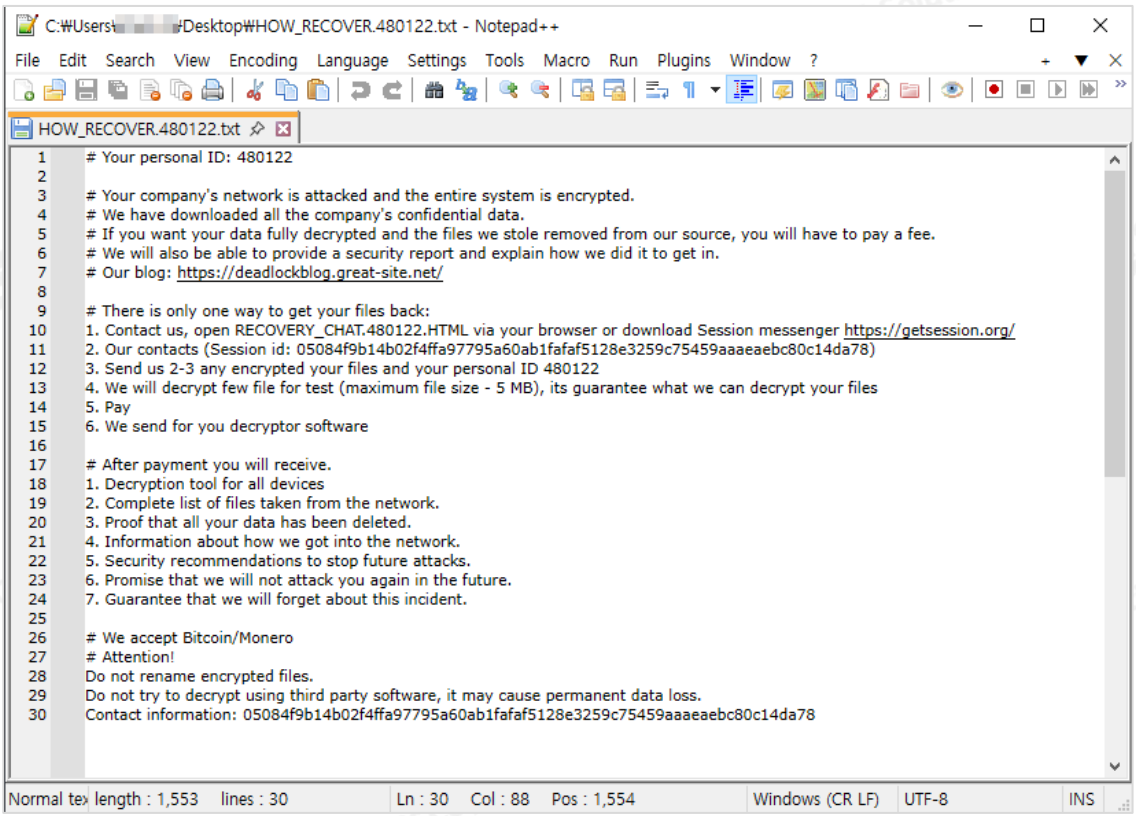
파일 암호화에 사용되는 키 또한 CryptGenRandom API로 파일마다 새로 생성되며, 키 길이는 시그마 상수 “expand 32-byte k”이 사용된 점을 근거로 256비트로 확인된다.

이렇게 생성된 키는 파일 암호화에 사용된 후 앞서 구한 공통 키를 이용하여 RFC 8439 ChaCha20-Poly1305 알고리즘으로 다시 암호화된다.

암호화된 키 데이터는 암호화 대상 파일의 데이터 마지막에 추가 저장되고 결과적으로 최종 암호화된 파일은 암호화된 데이터와 암호화된 암호 키, 논스 12 Byte, 임시 공개키 33 Byte로 구성된다.

▼ 다음 장에 계속

- 랜섬노트 생성 - txt



[랜섬노트 내용 확인]

랜섬노트는 파일의 암호화 작업이 모두 완료된 후, 피해자에게 감염 사실을 통보하는 내용을 담아 생성된다.

▼ 다음 장에 계속

- 바탕화면 변경 및 셀프 삭제



[변경된 바탕화면]

DeadLock 랜섬웨어는 파일 암호화 작업을 완료 후, 피해자에게 감염 사실을 인지시키는 단계와 자기 흔적을 지우는 단계를 차례로 수행한다.

이러한 마지막 단계는 명령행 인자 없이 실행된 경우만 수행하며, 특정 인자를 주고 실행된 경우에는 암호화만 수행하고 배경화면 변경과 자가 삭제를 모두 실행하지 않는 것으로 확인된다

“ viruschaser.com 에서
최신의 MS Windows 패치 업데이트를 확인하세요. ”



5. 주요 보안 뉴스

오라클 피플소프트 공격 다시 고개 들어…
대문자 하나만 퍼센트 기호로 바꿔 웹 방화벽 뚫고 웹셸 심어

공격자들이 인사·재무 업무에 널리 쓰이는 오라클 피플소프트(PeopleSoft)를 노려 웹애플리케이션방화벽(WAF)을 우회하고 서버에 웹셸을 심고 있다.

맨디언트는 샤이니헌터스와 연결된 공격 조직 UNC6240이 취약한 환경관리허브(PSEMHUB) 경로를 막아 둔 방화벽 규칙을 우회하도록 공격 코드를 고쳤다고 밝혔다.

- 출처: https://www.boannews.com/news/articleView.html?idxno=146061

북한 연계 해커, 새 리눅스 첩보 도구로 한국 자동차 기업·미디어 노려

북한과 연계된 사이버 공격 조직이 한국의 자동차·미디어를 겨냥해 리눅스 서버에 첩보 도구들을 심은 정황이 드러났다.

보안 기업 래피드7(Rapid7)은 7일(현지시각)이 북한 연계 조직이 한국의 자동차 기업과 미디어 조직을 장기적으로 감시하기 위한 도구를 설계했다는 분석을 내놨다.

- 출처: https://www.boannews.com/news/articleView.html?idxno=145770

SGA솔루션즈 엔드포인트 보안 솔루션

AI 기반 차세대 안티바이러스 솔루션
VirusChaser 10™ AI

패치 관리 솔루션
PatchChaser

PC 보안 수준 진단 솔루션
VirusChaser 내PC지키미

보안 레이더에서 전하는 보안 권고 사항

9월에 조사된 보안 동향 및 악성코드 분석을 통해 확인된 주요 공격 사례별 보안 권고 사항은 다음과 같다.

구분	위협 내용	권고 사항
사외 동향 UNC6240 / CVE-2026-35273 WAF 우회 및 SideEye 백도어	UNC6240이 PeopleSoft의 PSEMHUB 경로 내 한 글자를 인코딩 문자로 변형하여 WAF와 리버스 프록시의 문자열 규칙을 우회하고 CVE-2026-35273 자바 역직렬화 취약점을 이용해 웹셸 2개를 업로드한 뒤 정상 설치 파일로 위장한 SideEye 백도어를 실행하여 서버에 저장된 자격 증명 탈취	• CVE-2026-35273 보안 패치 적용 • WAF 우회 탐지 룰 강화(경로 인코딩 변형 패턴 포함) • 서버 내 비정상 파일 업로드 및 웹셸 탐지 모니터링
사외 동향 북한 연계 조직 / HA Proxy 플러그인 위장 ted 백도어 설치	북한 연계 조직이 그룹웨어 포털 취약점으로 최초 침투한 뒤 SSH 키로거로 관리자 접속 정보를 수집하여 내부 시스템으로 확장하였으며, HA Proxy의 맞춤형 플러그인 형태로 ted 백도어를 설치하여 내부 통신을 가로채고, pstatic.net 등 국내 정상 CDN 도메인 위장 및 리눅스 기본 바이너리 변조로 공격 활동 은폐	• 그룹웨어 포털 취약점 패치 적용 • HA Proxy 플러그인 설치 현황 점검 및 무결성 검증 • 리눅스 기본 바이너리 무결성 검증 • SSH 접근 로그 이상 여부 모니터링
피싱 메일 분석서 Microsoft 365 사칭 / 계정 정보 및 세션 쿠키 탈취	Microsoft 365 로그인 페이지로 위장한 AiTM(중간자 공격) 피싱 사이트가 수신자 이메일을 자동 입력하여 신뢰를 유도하고, 입력된 계정 정보를 ctx, flowToken 등의 값과 함께 실제 Microsoft 서버로 실시간 중계하여 세션 쿠키를 탈취함으로써 MFA를 우회하고 사용자 계정에 접근	• Microsoft 365 계정 관련 메일 수신 시 공식 사이트에 직접 접속 • FIDO2 기반 피싱 저항성 MFA 적용 • 비정상 로그인 시도 및 세션 쿠키 탈취 탐지 모니터링
분석서 DeadLock 랜섬웨어 / CVE-2024-51324 BYOVD 기반 보안 제품 무력화	DeadLock 랜섬웨어가 Baidu 백신 드라이버의 취약점(CVE-2024-51324)을 악용하는 BYOVD 기법으로 보안 솔루션을 우회하고 커널 수준의 악성 행위를 수행	• CVE-2024-51324 보안 패치 적용 • 취약한 서명 드라이버 차단 정책 적용 • 비정상 드라이버 로드 모니터링
분석서 DeadLock 랜섬웨어 / 화이트리스트 방식의 서비스, 프로세스 강제 종료	화이트리스트 방식으로 서비스 목록(216개)에 없는 모든 서비스를 '사용 안 함'으로 변경하고 프로세스 목록(80개)에 없는 프로세스를 강제 종료하여 보안 제품 및 백업 관련 프로세스를 포함한 시스템 전반의 서비스를 무력화	• 랜섬웨어 행위 기반 탐지 룰 적용 • 보안 솔루션 및 백업 관련 서비스 상태 주기적 점검
분석서 DeadLock 랜섬웨어 / 은닉된 볼륨 강제 마운트 및 파일 암호화	마운트되지 않은 은닉 볼륨을 Z부터 역순으로 강제 마운트하여 복구 파티션과 시스템 볼륨까지 암호화 대상에 포함하고 파일마다 ChaCha20 키와 secp256k1 ECDH로 산출한 공통 키를 결합하는 이중 암호화 구조로 복호화를 차단	• 중요 데이터 오프라인 백업 • 비정상 볼륨 마운트 행위 모니터링

SGA
보안레이더



www.sgasol.kr

경기도 의왕시 광진말로 54, 의왕 스마트시티퀀텀 B동 5층 525호

Copyright©2026 SGA Solutions co. Ltd,. All Rights Reserved.