KARA
Korea Anti Ransomware Alliance

2025.2Q

KARA 랜섬웨어 동향 보고서

KARA 랜섬웨어 동향 보고서

EQST Lab 팀 이호석, 정민수, 조효제, 이현아, 이승호

랜섬웨어 트렌드

1. 2 분기 TREND

TREND

- **RansomEXX :** Microsoft 의 권한 상승 취약점(CVE-2025-29824)을 악용
- **Qilin :** SAP NetWeaver 취약점(CVE-2025-31324, CVE-2025-42999)을 악용

THREAT

- **2 분기 Top5 랜섬웨어 :** Qilin, Akira, Play, SafePay, Lynx
- **Qilin 랜섬웨어 :** RansomHub 그룹 해체 후 활동량 증가

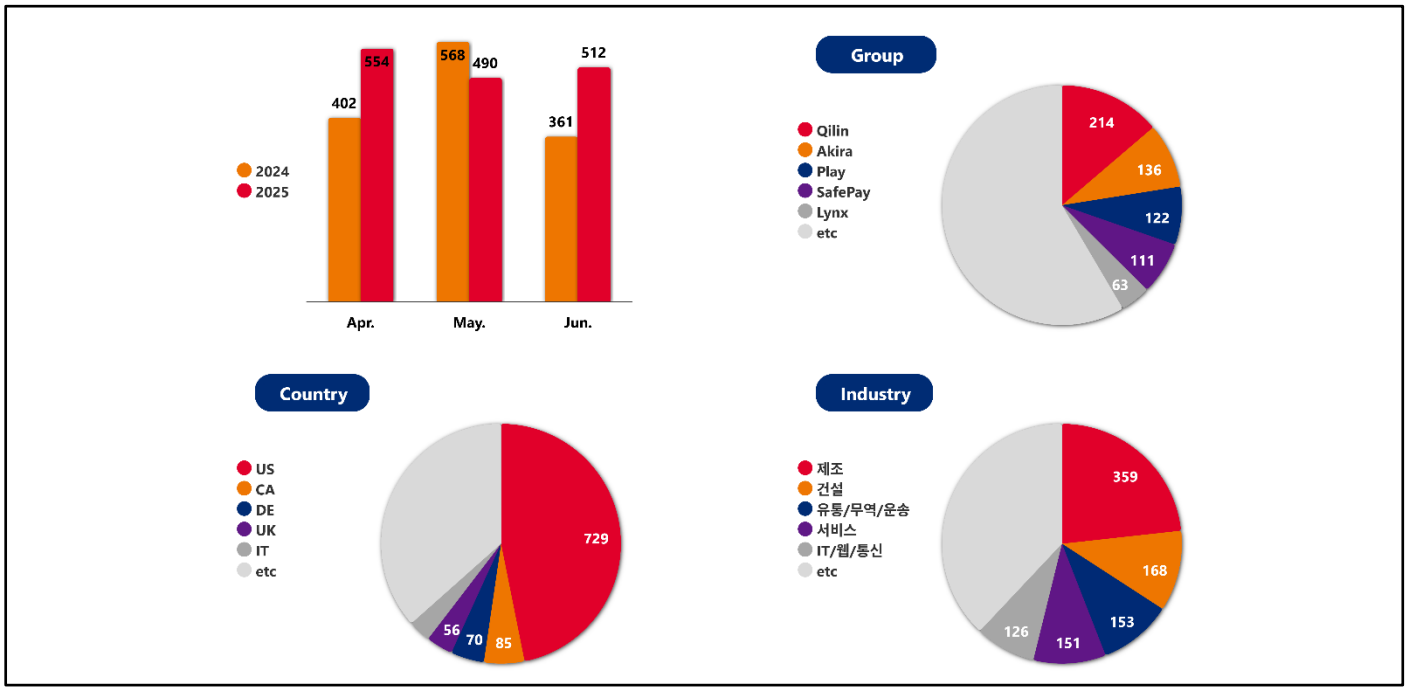
EXPLOIT

- **0-day :** CVE-2025-29824, CVE-2025-31324, CVE-2025-42999
- **1-day :** CVE-2024-57727, CVE-2024-21762, CVE-2024-55591

TARGET

- SAP NetWeaver 취약점을 통한 초기 침투
- 전체 공격 중 제조 23%, 미국 47%

2. 2 분기 랜섬웨어 활동 통계



[그림 1] 랜섬웨어 그룹 활동

2025 년 2 분기 랜섬웨어 피해 사례는 총 1,556 건으로, 2024 년 동기 대비 약 17% 증가했으나, 2025 년 1 분기 대비 약 40% 감소한 수치를 기록했다. 이러한 감소는 지난 분기 활발히 활동하며 많은 피해자를 발생시킨 주요 그룹들이 2 분기 들어 활동을 중단한 데 따른 결과로 보인다. 특히 크리티컬한 제로데이 취약점이 발생하면 이를 공격에 활발히 활용하는 Clop 랜섬웨어 그룹의 활동량 감소와, 최근 몇 년간 활발히 활동해온 RansomHub 그룹이 4 월 1 일부터 활동을 중단하면서 전체적인 공격 건수가 크게 감소한 것으로 보인다.

RansomHub 그룹이 2025 년 4 월 1 일을 기점으로 활동을 중단한 이후, Qilin 그룹의 활동이 눈에 띄게 증가한 것으로 나타났다. 일부 분석에 따르면, RansomHub 의 공격자 일부가 Qilin 그룹에 합류한 것으로 추정되고 있으며, 이를 뒷받침하듯 Qilin 의 월평균 피해자 수는 RansomHub 가 활동을 중단한 4 월 이후 기존 35 건에서 70 건으로 두 배 가까이 증가한 것으로 집계되었다.

전 분기 동안 총 222 건의 공격을 기록했던 Akira 랜섬웨어 그룹은, 이번 2 분기에는 136 건의 공격을 수행하며 활동 규모가 다소 감소했으나, Qilin 에 이어 두 번째로 많은 공격 건수를 기록한 그룹으로 확인되었다. 이 그룹은 Cisco·SonicWall VPN[1] 장비의 MFA[2] 미적용 환경이나 취약한 RDP[3]를 악용하고, 미국·캐나다·유럽 기업을 주로 표적으로 삼았다.

Play 랜섬웨어 그룹은 2 분기 동안 총 122 건의 피해자를 다크 웹에 게시하며 높은 수준의 활동을 보였다. 이들은 주로 Cisco ASA(Adaptive Security Appliance) 방화벽의 취약점을 활용해 초기 침투를 수행한 뒤, 윈도우 커먼 로그 파일

[1] VPN(Virtual Private Network): 공용 네트워크를 통해 전송되는 데이터를 암호화하여 안전하게 통신할 수 있도록 하는 가상 사설망 기술.

[2] MFA(Multi-Factor Authentication): 사용자가 계정에 접근할 때 두 가지 이상의 서로 다른 인증 요소를 요구해 보안을 강화하는 인증 방식.

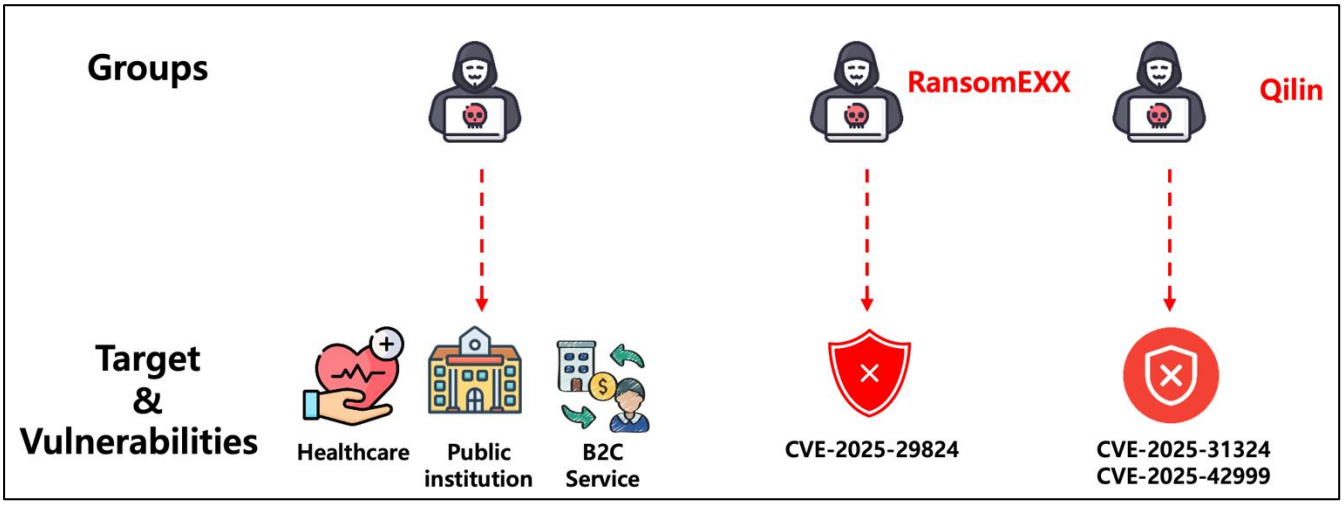
[3] RDP(Remote Desktop Protocol): Microsoft 가 개발한 원격 제어 프로토콜

시스템(CLFS)의 권한 상승 취약점인 CVE-2025-29824 를 악용해 SYSTEM 권한을 획득한 것으로 나타났다. 해당 취약점은 공식 패치가 배포되기 전부터 공격에 활용된 제로데이였으며, 이를 확인한 Microsoft 는 4 월 8 일 긴급 보안 패치를 발표했다.

이외에도 SafePay, 그리고 INC 랜섬웨어의 후속 버전으로 추정되는 Lynx 랜섬웨어 역시 최근 활발한 활동을 이어가고 있다.

가장 많은 랜섬웨어 공격을 받은 국가는 전 분기와 동일하게 미국과 캐나다로 나타났다. 독일은 이번 분기 70 건의 피해가 집계되며 전 분기 5 위에서 3 위로 상승했고, 그 뒤를 이어 영국과 이탈리아 역시 다수의 공격을 받아, 상위 피해국 순위에 포함되었다. 산업별 피해 현황을 보면, 제조업이 여전히 가장 많은 피해를 입은 산업 군으로 나타났으며, 그 뒤를 이어 건설, 유통·무역·운송, 서비스, IT/웹/통신 등의 순으로 피해가 집중되었다. 특히 건설 분야는 전 분기 대비 두 단계 상승해 이번 분기 두 번째로 많은 피해를 입은 산업 군으로 기록되었다.

3. 랜섬웨어 트렌드



[그림 2] 2025 년 2 분기 랜섬웨어 트렌드

✓ 직접적인 사용자 피해를 초래하는 B2C[4] 대상 랜섬웨어 공격

2025년 2분기에는 단순한 정보 탈취나 파일 암호화 수준을 넘어, 소비자가 직접적인 피해를 입는 B2C 서비스 대상의 랜섬웨어 공격이 다수 발생했다.

2025년 5월, 공학 및 과학 계산 소프트웨어인 MATLAB과 Simulink를 개발한 MathWorks가 랜섬웨어 공격을 받아 Cloud Center, License Center, File Exchange, 온라인 스토어 등 주요 플랫폼이 마비되었다. 5월 18일 서비스가 갑작스럽게 중단되면서 전 세계 사용자들은 라이선스 인증, 소프트웨어 다운로드, 로그인 등에 심각한 장애를 겪었으며, 특히 프로젝트를 진행 중이던 연구자, 엔지니어, 교수, 학생들은 연구 지연과 실무 차질을 겪으며 큰 피해를 입었다. MathWorks는 공격 직후 연방기관에 신고하고 외부 보안 전문가들과 함께 조사를 진행하며, 5월 21일까지 MFA 및 SSO[5] 인증 시스템을 복구하여 일부 기능에 대해 단계적인 접근을 허용했다. 그러나 전체 복구에는 시간이 더 걸려, 일부 서비스는 2주 이상 지연되거나 성능 저하가 발생했다. MathWorks는 사건 발생 며칠 후 랜섬웨어에 의한 피해임을 공개했으나, 공격자에 대한 구체적인 정보는 밝히지 않았다. 현재까지 공격의 배후를 자처한 조직은 없으며, 고객 데이터 유출 정황도 발견되지 않았다. 포렌식 분석 결과, 지속적인 내부 침입 흔적은 없었던 것으로 확인되었고, 회사는 본 공격을 금전적 목적의 상업용 랜섬웨어 공격으로 판단하며, 보안 시스템 강화를 통해 향후 위협에 대비 중이라고 밝혔다.

이어 2025년 6월 9일, 예스24 역시 랜섬웨어 공격을 받아 시스템이 마비되었다. 이 공격으로 예스24의 서버 정보들이 암호화되었고, 도서 검색·주문, 전자책 열람, 공연·팬미팅 티켓 예매 등 핵심 서비스가 전면 중단되었다. 홈페이지와 앱은 나흘 이상 접속이 불가능한 상태가 이어졌으며, 특히 전자책의 경우 기존에 구매한 콘텐츠에 접근할 수 없어 사용자들의 불편이 발생했다. 또한 공연 및 팬미팅 일정이 연기되거나 취소되면서 소비자의 불만이 가중되었다. 예스24는 공격 직후 관리자 계정을 회복했으며, 개인정보 유출 여부는 현재 조사 중이라고 밝혔다. 이후 6월 16일 랜섬웨어 공격 사실을 공식적으로 인정하고 사과문을 발표했으며, 도서·티켓 구매 등 소비자의 직접적인 피해가 발생한 핵심 기능부터 우선 복구한 뒤, 리뷰 등 부가 서비스도 순차적으로 정상화했다. 하지만 사건이 발생하고 두 달이 지난 8월 11일 경 다시 한번 랜섬웨어의 공격을 받아 시스템이

[4] B2C(Business to Consumer): 기업이 최종 소비자를 대상으로 상품이나 서비스를 직접 판매·제공하는 거래 형태.

[5] SSO(Single Sign-On): 한 번의 로그인으로 여러 시스템이나 서비스에 재인증 없이 접근할 수 있도록 해주는 인증 방식.

마비되었으며, 이로 인해 시스템을 복구하던 시간 동안 다시 한번 소비자가 서비스 이용에 불편을 겪어야 했다.

이처럼 연달아 발생한 두 사건으로 기존의 단순한 정보 유출이나 기업 간 서비스 장애와 달리, 일반 소비자가 실질적인 불편과 피해를 받는 사례가 증가하고 있다.

✓ 헬스케어 분야의 지속적인 피해

2025년 2분기 동안 다수의 랜섬웨어 그룹들이 전 세계 헬스케어 분야를 지속적으로 표적으로 삼으며 공격을 이어갔다. 특히 Interlock 그룹은 미국의 대표적인 투석 치료 전문 업체 다비타(DaVita)를 공격해 네트워크 일부를 암호화하고, 약 20TB의 데이터를 탈취했다고 주장했으며, 협상 결렬 후 약 1.5TB에 달하는 환자 정보를 다크 웹에 공개했다.

이어서 5월에는 미국 오하이오주의 케터링 헬스(Kettering Health)가 Interlock의 공격으로 병원 전산망이 마비되어 진료 시스템 전반에 큰 차질이 발생했다. Qilin 그룹 또한 5월 26일 미국 뉴잉글랜드 지역의 의료 네트워크 코버넌트 헬스(Covenant Health)를 공격해 산하 병원 세 곳의 진료 시스템에 침투하고, 최소 7,864명의 환자 정보를 유출했으며, 이후 해당 기관을 다크 웹 유출 목록에 등록했다가 삭제한 정황이 확인되었다.

한편 독일에서는 4월 말 SafePay 랜섬웨어가 AWO Stadtkreis Gießen e.V.라는 의료복지 기관을 공격했으며, 대만의 ChangShen 병원은 NightSpire 조직의 공격을 받아 약 800GB에 달하는 환자 데이터를 탈취당한 것으로 보고되었다. 2025년 6월 초, 중동 지역에서도 랜섬웨어 피해가 발생했다. 아랍에미리트의 아메리칸 병원 두바이(American Hospital Dubai)는 Gunra 랜섬웨어의 공격을 받아 핵심 시스템이 마비되었으며, 공격자는 약 4억 5천만 건의 의료 기록을 암호화했다고 주장했다.

과거에 발생한 피해가 최근에야 밝혀진 사례도 존재한다. 미국 미시간주의 맥라렌 헬스 케어(McLaren Health Care)는 2024년 7~8월 사이에 발생한 랜섬웨어 공격으로 74만 명 이상의 환자 정보가 유출되었음을 2025년 6월에 공식적으로 공개했다. 이는 헬스케어 분야가 지난해부터 꾸준히 피해를 받아오고 있는 분야임을 의미한다.

이러한 공격들이 연이어 발생함에 따라, 미국의 FBI, CISA, 보건복지부(HHS)는 Interlock 그룹이 의료기관을 위협하고 있다며 7월 공동 주의보를 발표했고, 의료산업 정보공유기관인 H-ISAC 역시 6월 보고서를 통해 최근 의료기관을 겨냥한 랜섬웨어 공격과 VPN 취약점 악용이 급증하고 있음을 지적하며 주의를 당부했다. 과거에는 수사기관의 주목을 피하기 위해 의료기관이나 주요 기반 시설에 대한 공격을 자제하던 랜섬웨어 그룹들이, 최근에는 민감한 정보를 다루는 조직이 금전 요구에 더 쉽게 응할 것이라 판단해 이들을 집중적으로 노리는 경향을 보이고 있다.

✓ 공공 서비스를 마비시키는 랜섬웨어 공격 확산

2025년 2분기에는 지방정부, 카운티 보안관실, 법원 등 공공기관을 표적으로 한 랜섬웨어 공격이 전 세계적으로 잇따라 발생했다.

4월 중순에는 미국 테네시주의 해밀턴 카운티 보안관실은 랜섬웨어 공격으로 시스템이 마비 되었고. 같은 시기 위스콘신주 아이오와 카운티에서 의심스러운 네트워크 활동이 감지되어 전체 시스템을 오프라인으로 전환했다. 이후 5월 9일 이들은 랜섬웨어 감염으로 인한 서비스 지연임을 주민에게 안내했다. 4월 29일에는 일리노이주 듀페이지 카운티에서는 보안관실, 법원, 서기국 등이 공격을 받아 시스템이 차단되었다.

지자체 전체가 마비된 사례도 있었다. 4월 18일 텍사스주 애빌린 시는 Qilin 랜섬웨어 조직의 공격을 받아 서버가 중단되고, 약 477GB의 데이터가 암호화되었다. 공격자는 5월 27일까지 몸값을 요구했으나, 시는 협상을 거부하고 전체 인프라 교체 작업에 착수했다. 현재까지 유출된 데이터의 악용 정황은 발견되지 않았으나, 만일의 상황에 대비해 계정 모니터링 및 비밀번호 변경을

권고했다.

오클라호마주 더런트 시는 6월 1일 랜섬웨어 공격으로 웹사이트 및 디지털 결제 시스템이 중단되었으며, 경찰 통신센터도 네트워크 장애로 911 응답 지연이 우려되었다. 같은 시기 오하이오주 로레인 카운티에서는 수십 개의 시스템이 오프라인 되고, 법원 업무가 일시 중단되었다가 복구되었다. 푸에르토리코 법무부 역시 범죄정보국 시스템이 공격을 받아, 증명서 발급 서비스를 일시적으로 중단하는 조치를 취했다.

이처럼 2025년 4~6월 사이 발생한 공공기관 대상 공격은 주로 정부 및 기관, 법원 등과 같은 행정 및 사법기관을 노렸으며, 시민의 일상생활에 직결되는 서비스 중단으로 이어졌다. 주요 행정기관에 대한 공격은 사용자 편의성을 저해할 뿐만 아니라, 탈취된 민감 정보가 악용되어 2차 피해로 이어질 가능성도 있어 보안에 각별한 주의가 요구된다.

✓ 취약점을 이용한 랜섬웨어 공격

2025년 4월 8일, Microsoft의 보안 조직 MSTIC는 윈도우 커먼 로그 파일 시스템(Common Log File System)에서 표준 사용자 권한을 SYSTEM 권한으로 상승시킬 수 있는 제로데이 취약점(CVE-2025-29824)이 실제 랜섬웨어 공격에 악용되고 있음을 확인하고, 긴급 패치를 배포했다. 이 취약점은 RansomEXX(Storm-2460) 그룹이 랜섬웨어 캠페인에서 활용한 것으로 분석되었으며, 공격자는 CLFS 드라이버를 통해 권한을 상승시킨 뒤 PipeMagic [6] 악성코드를 사용해 winlogon.exe에 페이로드를 주입하고, LSASS[7] 메모리 덤프 및 파일 암호화를 수행한 것으로 나타났다.

미국 CISA는 KillSec(또는 Kill Security)으로 알려진 랜섬웨어 그룹이 CrushFTP 서버의 인증 우회 취약점인 CVE-2025-31161을 악용해 공격을 수행했다고 밝혔다. 공격자는 해당 취약점을 통해 인증 없이 파일 전송 서버에 로그인한 뒤, 민감한 정보를 대량 탈취한 것으로 확인되었으며, 이후 탈취한 정보를 공개하지 않는 대가로 피해자에게 협상을 시도하는 형태로 공격을 수행했다.

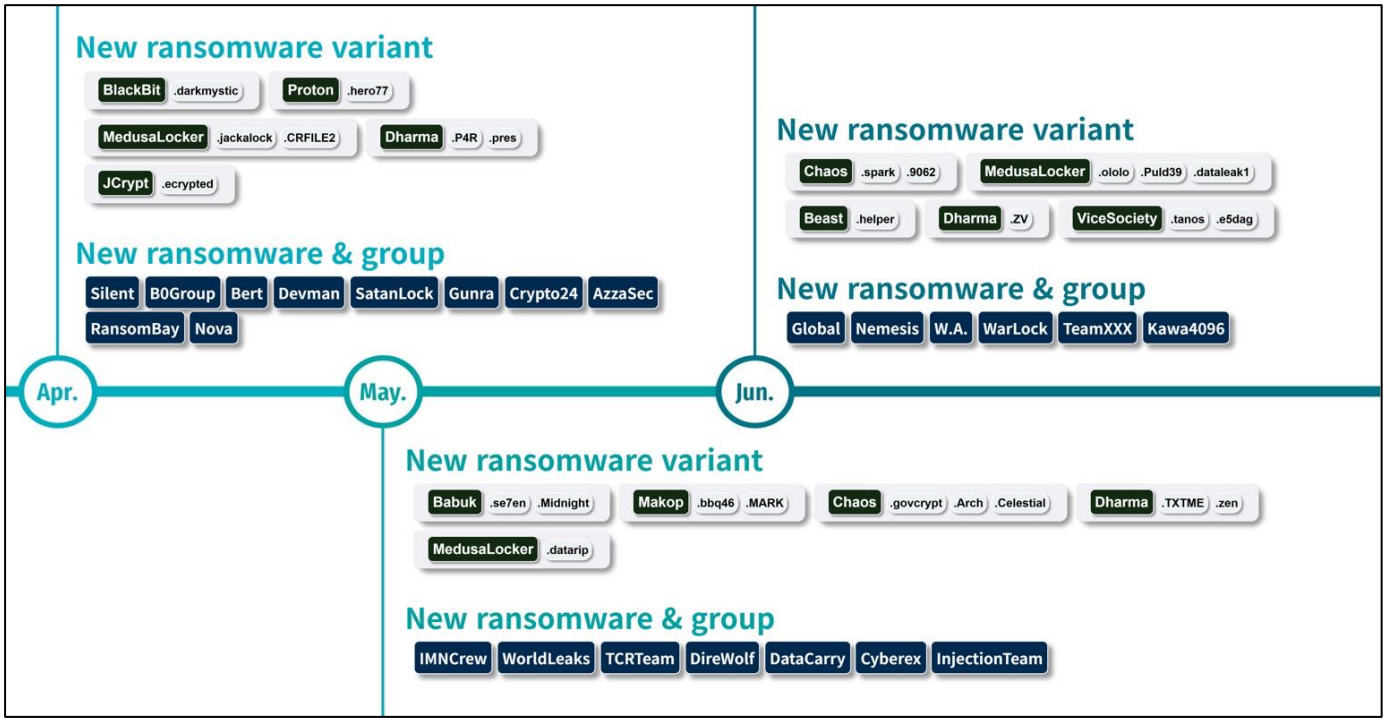
이어 4월 28일에는 SAP NetWeaver Visual Composer의 metadatauploader 엔드포인트에서 인증 우회가 가능한 취약점(CVE-2025-31324)이 실제 공격에 악용되고 있음이 공개되었다. 이 취약점은 인증 없이 특수한 POST 요청만으로 웹셸(helper.jsp, cache.jsp)을 서버에 업로드할 수 있어, 공격자는 이를 통해 지속적인 접근 권한을 확보하고 내부망 침투를 시도했다.

6월에는 SimpleHelp 원격 관리 도구에서 경로 탐색 취약점(CVE-2024-57727)이 활발히 악용되고 있는 정황이 포착되었으며, 미국 사이버안보·인프라보안국(CISA)은 이에 대해 긴급 보안 권고문을 발표했다. 해당 취약점은 공격자가 특정 파일 시스템의 임의 경로에 접근해 악성 파일을 업로드할 수 있도록 허용하며, Play와 DragonForce 등 랜섬웨어 그룹이 이를 악용한 사례가 보고됐다.

[6] PipeMagic: 2022 년 발견된 플러그인 기반의 백도어

[7] LSASS(Local Security Authority Subsystem Service): Windows 운영체제에서 보안 정책 적용, 사용자 인증, 액세스 토큰 생성 등을 담당하는 핵심 프로세스.

4. 신규 랜섬웨어 및 그룹 활동



[그림 3] 신규/변종 랜섬웨어

2025년 2분기에는 지난 분기에 이어 신규 랜섬웨어 그룹의 활발한 등장과 기존 그룹의 리브랜딩이 두드러졌다. Gunra, Devman, Kawa4096 등 신생 그룹들은 Windows, Linux, ESXi[8] 등 다양한 플랫폼을 대상으로 공격을 전개했으며, 일부는 파일 암호화 이전에 데이터를 탈취하는 이중 강탈 방식을 적극 활용했다. 특히 일부 그룹은 RaaS[9] 플랫폼을 운영하며 파트너 모집과 수익 공유 모델을 통해 세력을 빠르게 확장했다. 아래는 이번 분기에 새롭게 등장하거나 활동이 확인된 주요 랜섬웨어 그룹에 대한 설명이다.

- Gunra

  Gunra는 2025년 4월 등장한 신생 랜섬웨어 그룹으로 Conti 기반의 변종 랜섬웨어를 공격에 활용한다. 이들의 초기 침투 방식에 대해서는 현재까지 밝혀지지 않았으며, 침투 이후 내부를 탐색해 윈도우, 리눅스 버전의 랜섬웨어를 유포한다. 또한 파일 암호화 이전 데이터를 탈취하는 이중 강탈 방식을 활용해 공격하고 있으며, 협상이 되지 않을 경우 자신들의 자체 DLS[10] 사이트에 탈취한 정보를 업로드하며 피해자를 압박한다.

- Devman

[8] ESXi: VMware에서 개발한 하이퍼바이저로, 서버 하드웨어에 직접 설치되어 가상 머신을 생성·관리할 수 있도록 하는 가상화 플랫폼

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

[10] DLS (Dedicated Leaks Sites) : 공격자들이 운영하는 랜섬웨어 PR 및 탈취 데이터 공유 사이트

25년 4월부터 활동을 시작한 Devman 그룹은 현재까지 활발히 활동중인 랜섬웨어 그룹이다. 이들은 공격 초기 자체 랜섬웨어가 아닌 다른 그룹의 랜섬웨어를 활용해 공격을 수행했지만, 활동 시작 후 한달이 지난 5월부터 Mamona 랜섬웨어 기반의 자체 랜섬웨어를 개발해 공격을 수행했다. 또한 이들은 "Devman's Place"라는 자체 DLS를 운영하고 있으며, X(트위터)를 활용해 피해 기업에 대해 게시하거나, 랜섬웨어 개발 화면 등을 공유하며 자신들의 기술력을 과시하고 있다.

- Nova

RALord 조직의 리브랜딩 그룹인 Nova는 4월 28일 DLS를 개설하며 활동을 시작했으며 DLS에 게시했던 피해자들 중 일부가 RALord 시절의 피해자로 확인되었다. 이들은 RaaS 플랫폼을 운영하며 파트너로부터 일부의 수수료를 받아 운영 중이며, Windows, Linux, VMware ESXi용 랜섬웨어 등을 제공하며 빠르게 확장 중에 있다.

- Global

2025년 6월 2일 활동을 시작한 'Global'은 BlackLock과 Mamona 랜섬웨어를 운영했던 해커가 새롭게 개설한 RaaS 플랫폼으로 확인된다. 이 그룹은 Nova와 마찬가지로 파트너와 수익을 공유하는 형태로 운영하며 파트너를 모집하고 있다. 랜섬웨어 배포 외에도 VPS[11] 기반 DLS 호스팅, AI 챗봇을 이용한 다국어 협상 기능 등을 제공하며, 공격자가 더욱 쉽게 공격을 수행할 수 있도록 다양한 서비스를 지원하고 있다.

- Kawa4096

Kawa4096은 6월에 발견된 랜섬웨어 그룹이다. 해당 그룹은 Akira 그룹과 같은 디자인으로 DLS를 개설했으며, Qilin의 것과 유사한 내용으로 랜섬 노트를 구성했다는 특징이 있다. 등장 이후 6월 한달간 8개의 피해 조직에 대한 유출 데이터를 게시했고, 주로 미국과 일본에서 피해가 발생했다.

- WorldLeaks

2024년 11월 17일에 활동을 종료한 Hunters International 그룹의 리브랜딩으로 알려진 WorldLeaks는 랜섬웨어 공격 대신 데이터 탈취를 통한 금전 갈취를 목적으로 하는 전략을 채택했다. 데이터 탈취에 사용되는 악성코드는 자체 개발로 구현했으며, 탐지가 불가능하다고 주장했다. DLS에는 유출 데이터 목록과, 공지사항 카테고리가 존재하며, 기자 인증 시 유출 데이터 미리보기를 제공하는 카테고리 등으로 구성되어 있다. 5월에 DLS를 개설한 WorldLeaks는 2분기 동안 31개의 조직에 대한 공격을 주장했으며, 대부분의 피해는 미국과 제조업, 의료계에서 발생한 것으로 확인되었다.

- DireWolf

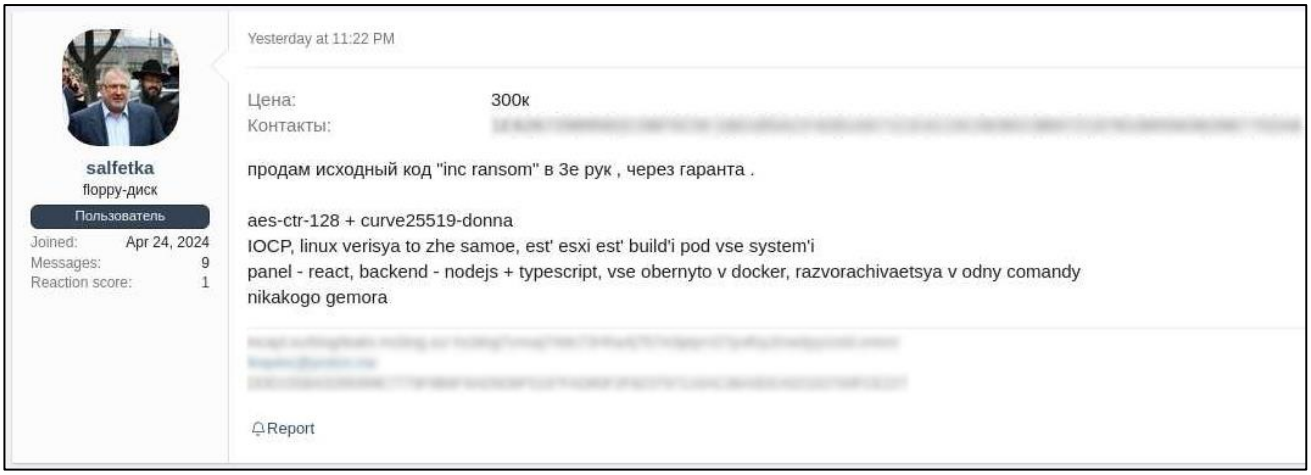
5월에 활동을 시작한 DireWolf 랜섬웨어 그룹은 2분기 동안 16개의 조직에 대한 공격을 주장했으며, 싱가포르와 제조업이 가장 많은 피해를 입은 것으로 확인됐다. DireWolf는 각 피해자마다 어떤 파일을 탈취했는지, 언제 업로드 했는지 등을 기록한 뒤, 협상에 실패하거나 일정 기간이 지나면 모든 사용자가 접근할 수 있도록 다크 웹에 게시하고 있다. DireWolf

[11] VPS(Virtual Private Server): 물리적 서버를 가상화 기술로 분할하여 각 사용자가 독립된 서버 환경처럼 사용할 수 있도록 제공하는 호스팅 서비스.

랜섬웨어는 최초 실행 종료 시 C:\runfinish.exe 경로에 빈 파일 하나를 생성해 해당 시스템이 이미 감염됐는지 여부를 확인하며, 사고 조사나 분석을 방해하기 위해 이벤트 로그 비활성화 및 삭제를 진행하고 복구 및 백업 파일을 삭제한다는 특징을 가지고 있다.

INC Ransomware 그룹 상세 분석

1. 개요



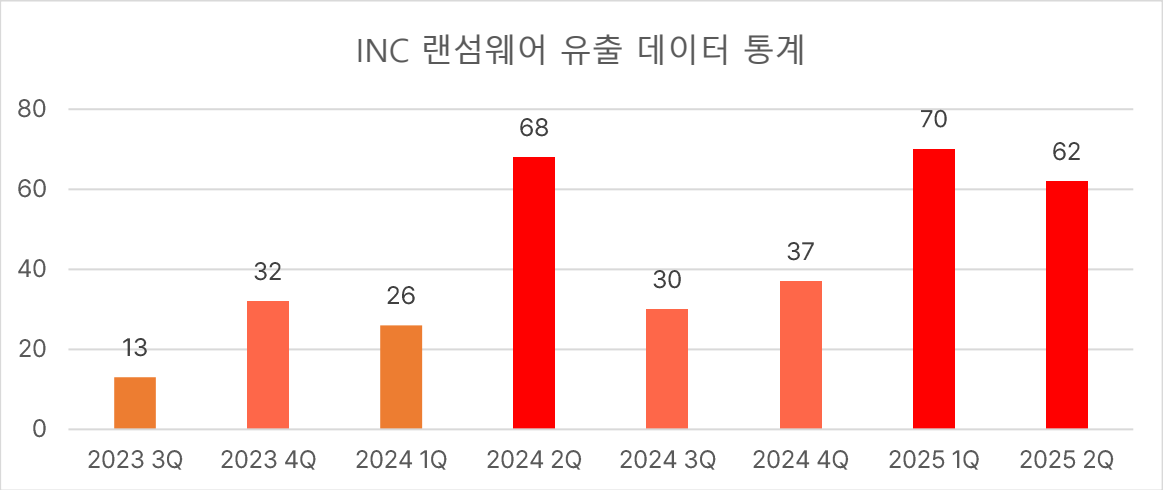
[그림 4] INC 랜섬웨어 소스 코드 판매 의혹

INC 랜섬웨어 그룹은 2023 년 7 월에 등장한 RaaS(Ransomware-as-a-Service) 그룹으로, 등장 이후 현재까지 386 개에 달하는 조직의 공격을 주장했으며 대부분의 피해는 미국에서 발생한 것으로 확인됐다. 특히 2024 년 7 월 말 새롭게 등장한 Lynx 랜섬웨어 그룹과 유사성이 짙은 모습을 확인할 수 있으며, 이는 INC 그룹이 랜섬웨어 소스 코드를 다크 웹 포럼에서 판매한 이후에 포착된 정황과 맞물려 두 그룹 사이의 연관성을 강하게 시사하고 있다.

랜섬웨어 소스 코드를 판매한 사용자 "salfetka"는 두 개의 다크 웹 포럼에 글을 올려, Windows 및 Linux/ESXi 버전의 랜섬웨어 소스 코드를 모두 판매한다고 밝히며, 가격은 30 만 달러(한화 약 4 억 2 천만 원)로 책정하고 구매자는 최대 3 명으로 제한하겠다고 언급했다. 해당 게시글에서 확인할 수 있는 세부 정보에 따르면, 이 랜섬웨어는 AES-CTR-128 과 Curve25519-Donna 알고리즘을 이용한 하이브리드 암호화 방식을 적용하고 있으며, 작성자인 salfetka 는 2024 년 3 월부터 해당 포럼에서 활동해 온 것으로 보인다. 또한 그는 최대 7,000 달러에 네트워크 접근 권한을 구매하려 한 정황이 있으며, 랜섬웨어 공격 수익금의 일부를 초기 침투 브로커에게 제공하겠다고 제안한 바 있어, 그가 INC 랜섬웨어 소스 코드의 실제 판매자일 가능성이 매우 높은 것으로 보인다.

결정적인 단서로는, "salfetka"가 자신의 포럼 서명에 INC 랜섬웨어 관련 페이지 URL 을 명시해 두었다는 점, 그리고 해당 시기와 맞물려 INC 와 기능적으로 유사한 Lynx 랜섬웨어가 새롭게 등장했다는 점이 있다. 일부에서는 이를 리브랜딩의 일환으로 해석하고 있지만, 단순히 소스 코드를 구매해 새로운 그룹으로 활동을 시작한 별개의 주체일 가능성도 배제할 수 없다.

2. INC Ransomware 통계



INC 랜섬웨어 그룹은 의료, 제조, 공공기관 등 운영의 연속성이 중단될 경우 치명적인 타격을 입을 수 있는 산업군을 주된 공격 대상으로 삼고 있으며, 실제 피해 조직 대부분도 이러한 분야에 속한다. 이러한 타깃 선정은 특히 민감한 데이터를 다루는 의료 기관이나 주요 협회 등을 공격할 경우, 해당 데이터를 악용한 협박이나 2 차 피해 발생 가능성이 높아, 피해자가 금전을 지불할 가능성이 더 크다고 판단했기 때문으로 풀이된다. 또한, 지금까지 발생한 피해의 대다수가 미국과 영국 등 영미권에 집중되어 있으며, 다크 웹 포럼에서 소스 코드를 판매한 것으로 추정되는 사용자가 러시아어를 사용하는 점을 고려할 때, INC 그룹이 러시아어권 기반의 조직일 가능성도 존재하는 것으로 보인다.

3. INC 랜섬웨어 상세 분석

INC 랜섬웨어는 2024 년에 개발된 C++ 기반 버전과 최근에 개발된 Rust 기반 버전이 존재하며, 각 버전마다 암호화 방식에 일부 차이가 있다. 본 보고서에서는 2024 년 C++ 버전을 기준으로 분석을 진행하고, 보고서 말미에서 Rust 버전의 변경된 점을 짧게 비교한다.

랜섬웨어가 실행되면 암호화를 위한 준비단계가 시작된다. INC 랜섬웨어는 총 공개된 10 개의 인자와, 숨겨진 –safe-mode 인자 1 개로 구성되어 있으며, 각 인자에 따라 플래그 값을 설정하여 암호화를 수행한다.

```
int __thiscall f_Help_4073D0(void *this)
{
  logging_406110("USAGE:\n");
  sub_404800(L"\t%s [ARGUMENTS]\n\n", this);
  logging_406110("ARGUMENTS:\n");
  logging_406110("\t--file <FILE>\t\tEncrypt only selected file\n");
  logging_406110("\t--dir <DIRECTORY>\tEncrypt only selected directory\n");
  logging_406110("\t--mode <MODE>\t\tChoose mode for file encryption (fast, medium, slow)\n");
  logging_406110("\t--ens\t\t\tEncrypt network shares\n");
  logging_406110("\t--lhd\t\t\tLoad hidden drives\n");
  logging_406110("\t--sup\t\t\tStop using process\n");
  logging_406110("\t--hide\t\t\tHide console window\n");
  logging_406110("\t--kill\t\t\tKill processes/services by mask\n");
  logging_406110("\t--debug\t\t\tEnable debug mode\n");
  return logging_406110("\t--help\t\t\tDisplay this message\n");
```

[그림 5] INC 랜섬웨어 실행 인자

실행 옵션	옵션 설명
--file	특정 파일을 대상으로 암호화를 수행
--dir	특정 경로를 대상으로 암호화를 수행
--mode	fast, medium, slow 설정으로 암호화 블록 크기를 설정
--ens	네트워크로 공유된 폴더를 암호화
--lhd	연결되지 않은 공유 폴더를 연결해 암호화
--sup	암호화를 위해 백업 및 문서 프로그램 등을 종료
--hide	실행중인 랜섬웨어 콘솔 화면을 숨김
--kill	실행중인 프로세스 및 서비스를 종료
--debug	암호화 과정에서의 디버그 메시지를 출력
--help	실행 옵션을 출력
--safe-mode	자신을 서비스로 등록하고 안전 모드로 재부팅

인자에 따른 실행 옵션 설정이 완료되면 암호화 준비 과정을 시작한다. 우선 kill 옵션이 활성화된 경우 모든 파일을 문제없이 암호화 하기 위해 실행중인 프로세스와 서비스를 확인해 특정 문자열이 포함된 서비스 및 프로세스를 종료한다.

```
pe.dwSize = 556;
hSnapshot = CreateToolhelp32Snapshot(0xFu, 0);
Process32FirstW(hSnapshot, &pe);
do
{
  pTargetProcess = TargetProcess;
  do
  {
    if ( wcsstr(pe.szExeFile, *pTargetProcess) )// "sql"
                                                // "veeam"
                                                // "backup"
                                                // "exchange"
                                                // "java"
    {
      v1 = OpenProcess(1u, 0, pe.th32ProcessID);
      v2 = v1;
      if ( v1 )
      {
        TerminateProcess(v1, 9u);
        CloseHandle(v2);
      }
    }
    ++pTargetProcess;
  }
  while ( pTargetProcess < &NumbOfProcess );
}
while ( Process32NextW(hSnapshot, &pe) );
return CloseHandle(hSnapshot);
```

[그림 6] 실행중인 프로세스 탐색

타깃 프로세스 문자열		
sql	veeam	backup
exchange	java	
타깃 서비스 문자열		
sql	veeam	backup
exchange		

프로그램은 파일 암호화 작업 전, 프로세서 개수를 확인하고, “프로세서 수 × 4” 만큼의 워커 스레드를 미리 생성하여 스레드 풀 크기를 결정한다. 이후 CreateIoCompletionPort 함수를 통해 I/O 완료 포트(IOCP)를 생성하고, 생성된 워커 스레드들은 GetQueuedCompletionStatus 호출을 통해 암호화 작업 패킷을 대기 및 처리하도록 구성한다.

```
HANDLE Create_IO_Port_4056A0()
{
  signed int v0; // edi
  signed int i; // esi
  HANDLE result; // eax

  GetSystemInfo(&SystemInfo);
  v0 = 4 * SystemInfo.dwNumberOfProcessors;
  CompletionPort = CreateIoCompletionPort(0xFFFFFFFF, 0, 0, 0);
  lpHandles = malloc(v0 >> 30 != 0 ? -1 : 4 * v0);
  for ( i = 0; i < v0; ++i )
    lpHandles[i] = CreateThread(0, 0, t_ReadAndWriteFile_405DC0, CompletionPort, 0, 0);
  result = CompletionPort;
  ExistingCompletionPort = CompletionPort;
  return result;
```

[그림 7] 스레드 풀 생성

암호화된 폴더에 생성할 랜섬 노트를 디코딩 한다. text 와 html 형태의 랜섬 노트는 파일 내 Base64 로 인코딩 되어 저장되어 있으며, 이를 디코딩 한 뒤 파일에 저장된 감염자의 ID 를 합친 후 전역 변수에 복호화 된 데이터의 주소를 저장하며, 이후 암호화 대상 폴더에 파일로 생성한다.

```
pcbBinary = 0;
v0 = lstrlenA(b64_note);
CryptStringToBinaryA(b64_note, v0, 1u, 0, &pcbBinary, 0, 0);
v1 = malloc(pcbBinary);
CryptStringToBinaryA(b64_note, v0, 1u, v1, &pcbBinary, 0, 0);
lpString2 = v1;
v1[pcbBinary] = 0;
lpString2 = ChangeUserID_4062C0(v1);
v2 = lstrlenA(b64_note_html);
CryptStringToBinaryA(b64_note_html, v2, 1u, 0, &pcbBinary, 0, 0);
v3 = malloc(pcbBinary);
CryptStringToBinaryA(b64_note_html, v2, 1u, v3, &pcbBinary, 0, 0);
RansomNote = v3;
v3[pcbBinary] = 0;
result = ChangeUserID_4062C0(v3);
RansomNote = result;
return result;
```

[그림 8] 랜섬 노트 복호화 및 감염자 ID 추가

lhd 옵션이 활성화 된 경우 시스템의 숨겨진 네트워크 드라이브를 모두 암호화 하기 위해 A ~ Z 드라이브를 탐색하고, 드라이브가 있다면 마운트를 시도한다.

```
do
{
  if ( !v0 )
    break;
  if ( GetVolumePathNamesForVolumeNameW(v4, szVolumePathNames, 0x78u, &cchReturnLength)
    && lstrlenW(szVolumePathNames) == 3 )
  {
    szVolumePathNames[0] = 0;
  }
  else
  {
    v6 = lpszVolumeMountPoint[v0--];
    if ( SetVolumeMountPointW(v6, v4) )
    {
      if ( flag_dbg_4287B0 )
        logging_404800(L"   [+] Mounted %s\n", v6);
    }
    else if ( flag_dbg_4287B0 )
    {
      LastError = GetLastError();
      logging_404800(L"   [-] Failed to mount %s Error: %d\n", v6, LastError);
    }
    v5 = FirstVolumeW;
  }
}
while ( FindNextVolumeW(v5, v4, 0x8000u) );
```

[그림 9] 숨겨진 드라이브 마운트

암호화를 위한 스레드 풀 생성까지 완료되면 암호화 로직이 실행된다. 특정 파일 또는 경로가 지정되어 실행하는 경우가 아니라면 파일 암호화를 위해 각 드라이브를 A 에서 Z 드라이브까지 탐색하며, 탐색하는 드라이브가 존재할 경우 드라이브의 복구를 막기 위해 볼륨 쉐도우 카피를 삭제한다. 일반적인 랜섬웨어는 볼륨 쉐도우 카피 삭제를 위해 파워쉘과 WMI 를 이용해 저장된 볼륨 쉐도우를 삭제하지만, INC 랜섬웨어는 DeviceIoControl API 에 IOCTL_VOLSNAP_SET_MAX_DIFF_AREA_SIZE 에 해당하는 0x53c028 값을 전달해 볼륨 쉐도우 크기를 제한하는 방식으로 볼륨 쉐도우 카피를 삭제한다.

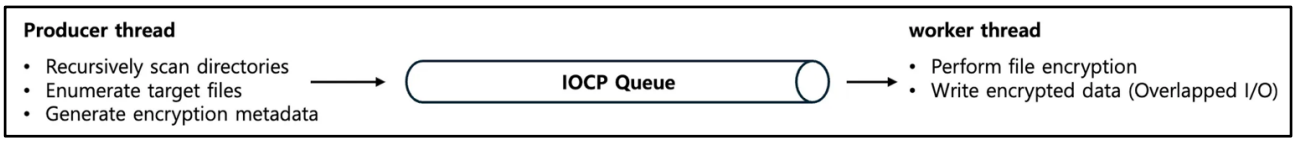
```
else
{
  if ( DeviceIoControl(result, 0x53C028u, InBuffer, 0x18u, 0, 0, &BytesReturned, 0) )
  {
    if ( dword_4287B0 )
      print_log(L"[+] Successfully delete shadow copies from %c:/ \n", a1);
  }
```

[그림 10] DeviceIoControl 을 이용한 볼륨 쉐도우 카피 삭제

삭제가 완료되면 각 드라이브의 경로를 인자로 스레드를 생성해 예외 경로나 예외 확장자 파일을 제외한 파일들을 암호화 대상으로 지정한다.

예외 파일 확장자		
exe	msi	dll
inc	INC	
예외 경로		
windows	program files	program files(x86)
$RECYCLE.BIN	appdata	

INC 랜섬웨어는 Windows 의 I/O Completion Port(IOCP) 기반 비동기 작업 처리 모델을 활용해 파일 암호화를 수행한다. 먼저, IOCP 큐를 생성하고, 암호화 대상 드라이브별로 파일을 탐색하는 전용 스레드를 생성 후 암호화 대상 파일 목록을 수집한다. 이후, 암호화에 필요한 정보(파일 경로, 핸들, 암호화 키 등)를 담은 작업 구조체를 IOCP 큐에 전달한다. 워커 스레드는 GetQueuedCompletionStatus 를 통해 큐에 등록된 작업 구조체를 가져와 암호화 작업을 수행한다.



[그림 11] IOCP 기반 비동기 파일 암호화

파일 암호화를 위한 키 생성 및 암호화 작업에 앞서, 악성코드는 대상 파일에 대해 약 0x23 바이트 크기의 쓰기 테스트를 수행하여 해당 파일에 대한 접근 및 쓰기 권한을 확인한다. 파일 쓰기에 실패하면 다음 단계로 파일의 소유권을 탈취하고 ACL(Access Control List)을 수정하여 쓰기 권한을 강제로 획득한다. 추가적으로 sup 옵션이 활성화된 경우에는 암호화 대상 파일을 점유 중인 프로세스를 강제로 종료한 뒤 다시 쓰기 가능 여부를 확인한 뒤 암호화 준비 과정을 시작한다.

```
FileAttributesW = GetFileAttributesW(pObjectName);
SetFileAttributesW(pObjectName, FileAttributesW & 0xFFFFFFFE);
if ( ChekeFileWrite_405800(pObjectName)
  || f_sup_42927C && Kill_Related_Process_4053E0(pObjectName) == 1 && ChekeFileWrite_405800(pObjectName)
  || (FileW = SetACL_407280(pObjectName), FileW == HANDLE_FLAG_INHERIT)
  && (LOBYTE(FileW) = ChekeFileWrite_405800(pObjectName), FileW) )
```

[그림 12] IOCP 기반 비동기 파일 암호화

먼저, 악성코드 내부에 Base64 로 인코딩 되어 저장된 공격자의 Curve25519 공개키를 디코딩한다. 이후, CryptGenRandom API 를 이용해 랜덤한 32 바이트 길이의 개인키를 생성한다. 이 개인키와 공격자의 공개키를 기반으로 Curve25519 ECDH(Elliptic Curve Diffie-Hellman) 연산을 수행하여 공유 비밀(shared secret)을 생성한다. 공유 비밀은 두 개의 키 페어 중 하나의 공개키와 상대방의 비밀키만으로도 동일하게 연산할 수 있기 때문에, 공격자의 비밀키를 노출하지 않고도 파일 암호화가 가능하며, 피해자의 공개키만 안다면 공격자의 비밀키로 동일한 공유키를 생성해 복호화가 가능하다.

생성된 공유 비밀은 SHA-512 해시 함수를 통해 해시값으로 변환되며, 이 중 첫 16 바이트가 AES 암호화에 사용된다. 해당 AES 키는 Key Expansion 과정을 거쳐 총 176 바이트 크기의 키 스케줄 테이블로 확장되며, 이 키 테이블은 IOCP 큐를 통해 워커 스레드에 전달되어 파일 암호화 작업에 사용된다.

키 생성이 끝나면 암호화할 대상 파일의 정보들과, 암호화 대상 파일 끝에 추가될 Footer 가 포함된 구조체를 구성한 뒤 큐를 통해 암호화 워커로 전달한다. 워커에서는 전달받은 구조체를 파싱 한 뒤 특정 필드의 플래그 값을 기준으로 1 -> 2 -> 0 -> 3 순으로 분기문을 수행한다.

플래그 값	플래그 설명
0	암호화 대상 파일의 크기에 따라 암호화 대상 블록을 읽어온다.
1	AES-128 을 통해 버퍼에 읽어온 파일 데이터를 암호화 한다.
2	파일 끝에 생성한 공개키를 포함한 Footer 를 추가한다.
3	파일 암호화 후, MoveFile API 를 통해 기존 파일의 확장자를 .INC 로 변경하여 덮어쓴다

```
switch ( Overlapped->flag )
{
  case 0:                                   // case 0: get block
    field_encMode_138 = Overlapped->file_pubkey_F8.enc_process_init_flag_40;
    if ( !field_encMode_138 )
    {
      filesize_high_24 = Overlapped->filesize_high_24;
      szTargetFile = Overlapped->szTargetFile;
      giga_2_flag = Overlapped->file_pubkey_F8.szBlock_44 >> 31;// check_2gigabyte
      szBlock = Overlapped->file_pubkey_F8.szBlock_44;
      v39 = filesize_high_24;
      if ( szBlock + *&Overlapped->next_block_offset_low_28 >= __SPAIR64__(filesize_high_24, szTarget
```

```
case 1:                                     // case 1: encrypt
  szWriteBlock_4 = Overlapped->szWriteBlock_4;
  expand_key_38 = Overlapped->sha_first16_expand_key_38;
  enc_data_buf = Overlapped->bufEncFile;
  ++Overlapped->file_pubkey_F8.encBlockCount_4c;
  v4->flag = 2;
  encrypt_401300(expand_key_38, enc_data_buf, szWriteBlock_4);
  WriteFile(v4->hEncTarget_14, v4->bufEncFile, szWriteBlock_4, 0, v4);
  hIoPort = lpThreadParameter;
  ReadFile = ::ReadFile;
  goto LABEL_2;
```

```
case 2:                                     // case 2: write footer
  v38 = Overlapped;
  Overlapped->cur_block_offset_low_8 = Overlapped->szTargetFile;
  v4->cur_block_offset_high_C = v4->filesize_high_24;
  hFile = v4->hEncTarget_14;
  v4->flag = 0;
  WriteFile(hFile, &v4->file_pubkey_F8, 0x50u, 0, v38);
  hIoPort = lpThreadParameter;
```

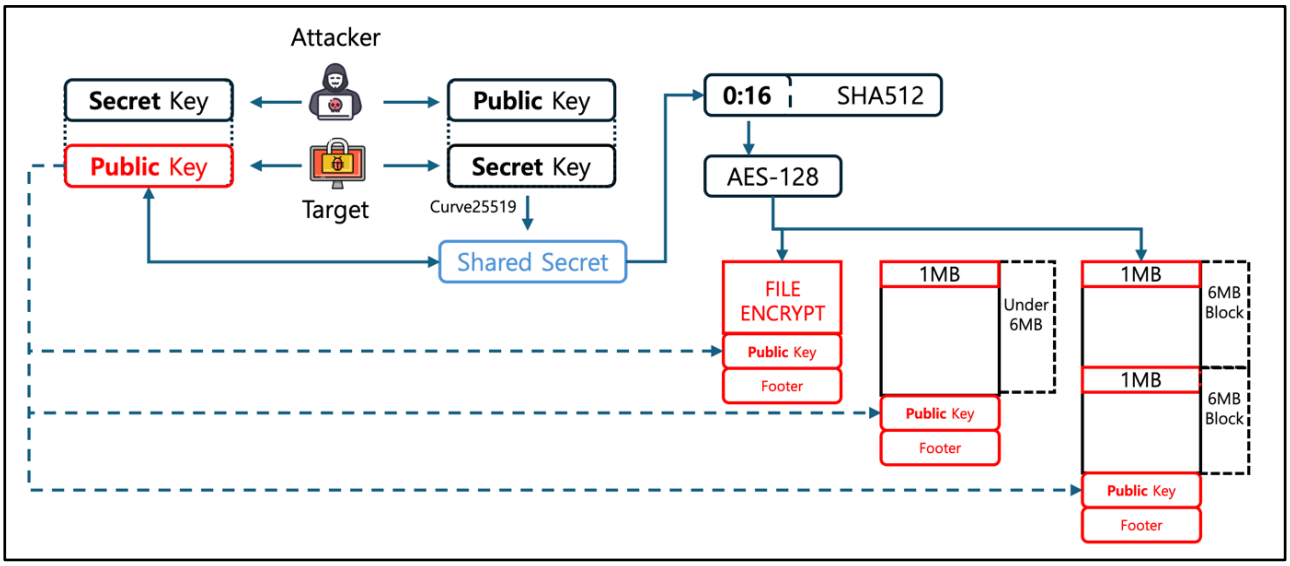
```
case 3:                                     // case 3: End
  CloseHandle(Overlapped->hEncTarget_14);
  InterlockedDecrement(&Addend);
  MoveFileExW(v4->buf_origin_target_filename_18, v4->p_EncryptedFileName_1c, 9u);
  j___free_base(v4->buf_origin_target_filename_18);
  j___free_base(v4->p_EncryptedFileName_1c);
  j___free_base(v4->bufEncFile);
  j___free_base(v4);
  hIoPort = lpThreadParameter;
```

[그림 13] 플래그 별 처리 로직

INC 랜섬웨어는 파일을 암호화할 때 기본적으로 1MB 단위의 블록을 사용하며 파일 크기에 따라 부분 암호화를 수행한다.

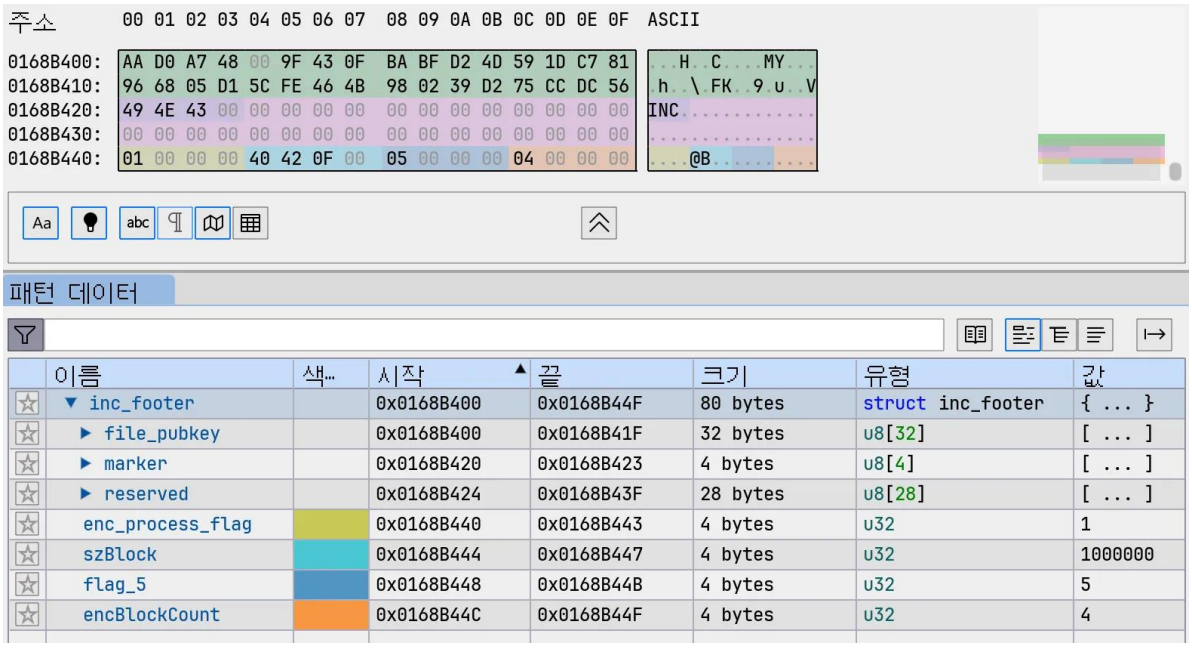
- 파일 크기가 1MB 이하일 경우, 전체 파일을 암호화
- 1MB 초과 6MB 이하인 경우, 파일의 상위 1MB 만 암호화

- 6MB를 초과하는 경우, 파일을 6MB 단위로 나누고, 각 블록의 상위 1MB 씩을 반복적으로 암호화



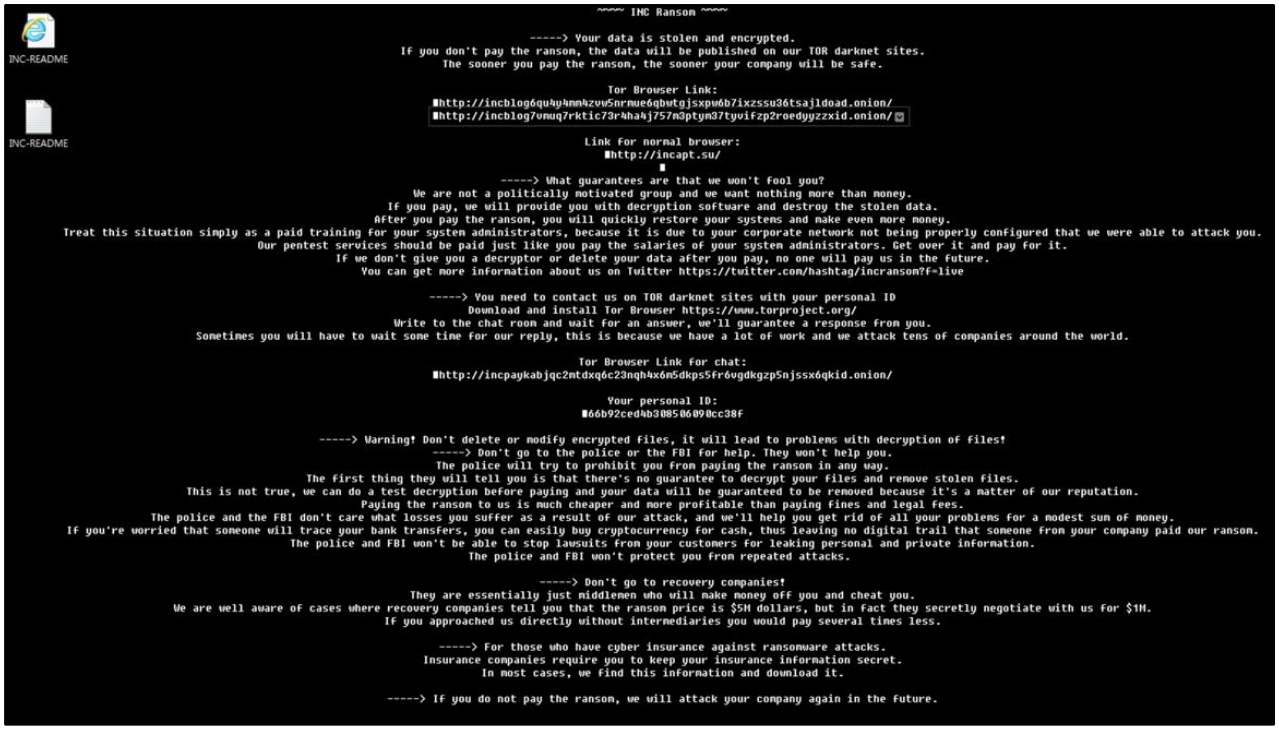
[그림 14] Shared Secret를 이용한 파일 암호화

파일 암호화가 완료되면, 암호화에 사용된 비밀키와 쌍을 이루는 공개키, 마커 문자열 "INC", 암호화 옵션, 블록 크기, 암호화된 블록 개수 등의 정보를 포함하는 0x50 바이트 크기의 Footer 구조체가 파일 끝에 저장된다.

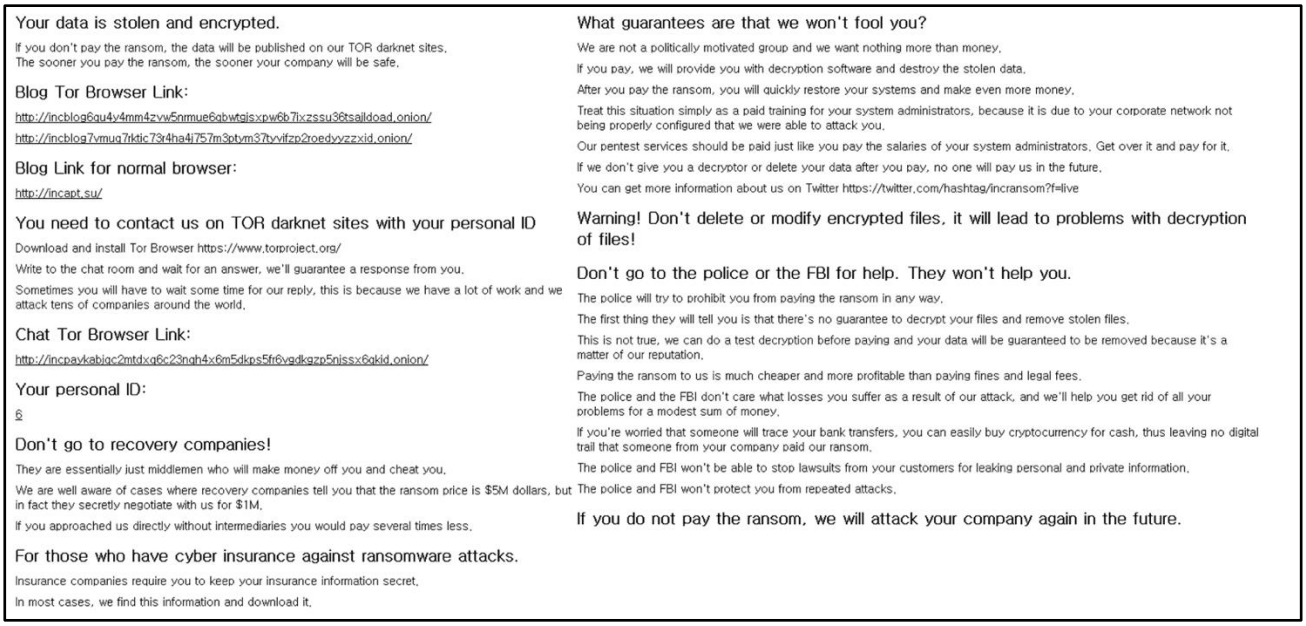


[그림 15] INC 랜섬웨어 Footer 정보

파일 암호화가 완료 후 배경화면을 랜섬 노트의 내용으로 변경한 뒤 종료한다.



[그림 16] INC 랜섬웨어에 감염된 시스템 바탕화면



[그림 17] INC 랜섬웨어 HTML 랜섬 노트

4. Rust 기반 INC 랜섬웨어

2025 년 5 월, 기존 INC 랜섬웨어의 다음 버전으로 추정되는 Rust 기반의 INC 랜섬웨어가 새롭게 등장했다. Rust 버전은 기존 C++ 버전과 마찬가지로 IOCP 기반의 구조, 암호화 대상 정보를 구조체에 담아 워커 스레드에 전달하는 방식, 플래그 값을

기반으로 한 파일 암호화 및 저장, 블록 단위 암호화 등 핵심 동작 방식은 동일하게 유지하고 있다. 다만, 프로그램 실행 인자와 암호화 방식 등 일부 세부 동작은 변경되었다.

실행 옵션	기능 설명	기존	Rust 기반
--file	특정 파일을 대상으로 암호화	O	O
--dir	특정 경로를 대상으로 암호화	O	O
--mode	fast, medium, slow 설정으로 암호화 블록 크기를 설정	O	O
--ens	네트워크로 공유된 폴더를 암호화	O	X
--lhd	연결되지 않은 공유 폴더를 연결해 암호화	O	X
--sup	암호화를 위해 백업 및 문서 프로그램 등을 종료	O	O
--hide	실행중인 랜섬웨어 콘솔 화면을 숨김	O	O
--kill	실행중인 프로세스 및 서비스를 종료	O	X
--debug	암호화 과정에서의 디버그 메시지를 출력	O	X
--def	파일 암호화 없이 랜섬 노트만 출력	X	O
--help	실행 옵션을 출력	X	O
--version	실행 파일명 출력	X	O
--proc	인자로 받은 프로세스 종료 (기존 –kill 옵션 분리)	X	O
--serv	특정 서비스 종료 마스크(기존 –kill 옵션 분리)	X	O

```
USAGE:
        noaslr_new_inc.exe [FLAGS] [OPTIONS]

FLAGS:
        --def         Don't encrypt files, just left notes everywhere
    -h, --help        Prints help information
        --hide        Hide console
        --sup         Stop using process
    -V, --version     Prints version information

OPTIONS:
        --dir <directory(ies)>     Encryption directory(ies) (e.g. C:₩,D:₩,E:₩)
        --file <file(s)>           Encryption of file(s) (e.g. 1.txt,2.txt,3.txt)

        --mode <mode>              Encryption mode (fast, medium, slow)
        --proc <proc>              Kill processes by mask (e.g. veeam,backup,sql)

        --serv <serv>              Kill services by mask (e.g. veeam,backup,sql)
```

[그림 18] Rust 버전 실행 옵션

암호화 대상 디렉토리 내에서 특정 폴더 및 확장자를 제외한 모든 파일을 수집한 뒤 암호화를 수행한다. 이때 암호화 대상 경로 및 확장자 일부가 변경되었으며, 이를 반영하여 제외된 경로 및 확장자를 탐색 대상에서 필터링한 후, 남은 파일들을 대상으로 암호화를 진행한다. 아래 표에서 붉은색으로 표시된 확장자 및 경로는 Rust 버전에서 새롭게 추가된 대상이며, 취소선으로 표시된 항목은 제외된 문자열이다.

예외 파일 확장자		
exe	~~msi~~	dll
~~inc~~	INC	log

예외 경로		
windows	program files	program files(x86)
$RECYCLE.BIN	appdata	programdata
all users	sophos	

파일 암호화 방식은 C++ 버전과 마찬가지로 Curve25519 를 이용한 공유 비밀 생성 방식을 유지하지만, 이후 단계의 해시 생성 및 암호화 알고리즘이 변경되었다. 기존 버전에서는 공유 비밀을 SHA-512 로 해시한 뒤, 이 중 상위 16 바이트를 AES-128 암호화 키로 사용했으나, Rust 버전에서는 SHA-256 을 사용하고, 파일 암호화는 Salsa20 알고리즘을 통해 수행되도록 변경되었다.

5. 결론

INC 랜섬웨어 그룹은 RaaS 모델을 기반으로 운영되는 조직으로, 공격자가 직접 랜섬웨어를 개발하지 않아도 손쉽게 공격을 실행할 수 있도록 지원하고 있다. 이러한 구조는 사이버 범죄의 진입 장벽을 대폭 낮추어 다양한 계열사들을 유입시키는 데 기여하며, 범죄의 서비스화와 산업화를 가속시키고 있다. 특히 INC 는 소스 코드를 판매한 정황이 포착된 만큼 다른 공격자가 이를 기반으로 유사 랜섬웨어를 제작하거나 새로운 변종을 확산시킬 가능성이 존재한다. 이러한 2 차 위협은 단일 그룹의 활동에만 국한되지 않고, 다양한 위협 주체의 출현으로 이어질 수 있어 이에 따른 적절한 보안 대책 마련에 대한 중요성이 강조되고 있다.

6. IoCs

INC Ransomware SHA256
508a644d552f237615d1504aa1628566fe0e752a5bc0c882fa72b3155c322cef
7f104a3dfda3a7fbdd9b910d00b0169328c5d2facc10dc17b4378612ffa82d51
463075274e328bd47d8092f4901e67f7fff6c5d972b5ffcf821d3c988797e8e3
502332b1b5a04ec85ef72e75535469f9e1ae61fe6b1aa6b55274626f320415c2
02472036db9ec498ae565b344f099263f3218ecb785282150e8565d5cac92461
b97417afbd789a21c3f4fe33bf7501bf2cf3f5c735cb4a1258ffd8a32b443e6a
33c1f433dcd7dcccbd8bdd6d418f63285b42484a3022c9ccfb88cc24be18cd5d
e17c601551dfded76ab99a233957c5c4acf0229b46cd7fc2175ead7fe1e3d261
b62c5c88f31f9c06ac7302fab1a5d05aa7a46a302c32a63ffa238f3b5b6aa3fb
5a8883ad96a944593103f2f7f3a692ea3cde1ede71cf3de6750eb7a044a61486
864c1c803ef3b4ca6ab50482820d4da1c5cb23378ca52c689b32e205ae4c1ccf
a5925db043e3142e31f21bc18549eb7df289d7c938d56dffe3f5905af11ab97a
909033ac13a6114191e0821fa49aee1bf5517d7849251b4d1c135f4cd7ffeecf
4da86516bf85633f3db22c328825e5fe1e30ef4581179a3b886a1a2ed8ecb401
26cca9c6d1c591761dd3c1cdbadc589f28bd87b7ddfe1961491396de09577fc8
1df4a74fbe8a9875a4386960f1006d29de7907af830b4c8a30a643e752299030
ee1d8ac9fef147f0751000c38ca5d72feceeaae803049a2cd49dcce15223b720
d147b202e98ce73802d7501366a036ea8993c4c06cdfc6921899efdd22d159c6
INC Ransomware(Rust Ver) SHA256
b1815ef993b2649be791f0cf4249e502e7c3763fe69451b8b32508089e15d103

61f70b9a0bde499d764807fe24517e64ea0130a3f6e493ead360058e59854776
e86487e21cde16fb15341fa7f9cee283944873c82bb06a4c25e32c0b16d08a85
fd0dbc6d941ff76e5204df4c644ba0d3241d05995f30e6b837618cd9dcc8b99c
17317ee3c9bd706ef2942a38f55c05176e4abdf377a5b72250d89ebf2a795ca0
71a2cdf9d39c4ba7b7bc3091c38b0be1624b3171fb3a649ad2a6430f9ae30f82
5bcb8a717e160a88b7cf95f0ad778ad7446041565356beb332676e33ff4f1829
c3d8c86844d7bb86dea31bcb0510542fb59bc3bc67a4e3df59f72da5c8509f54
56da934a117689fe0f26d0cf6cf7650f94e2f9d625d9ac066ae0096a3fc7eab4
be9e1fd4dcf8a644aba70c8e92fa07a54d0ce96fb74217b48991700d281083bd

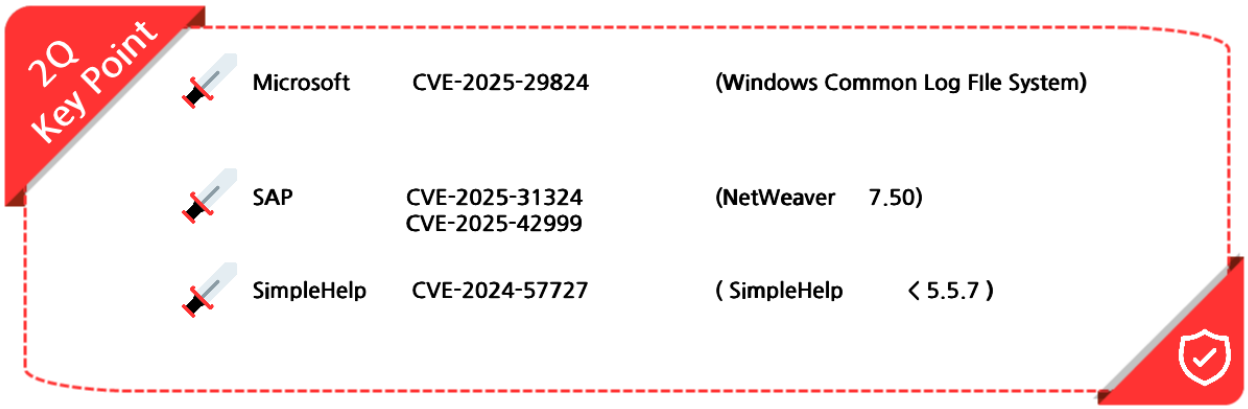
INC Ransomware 구조체

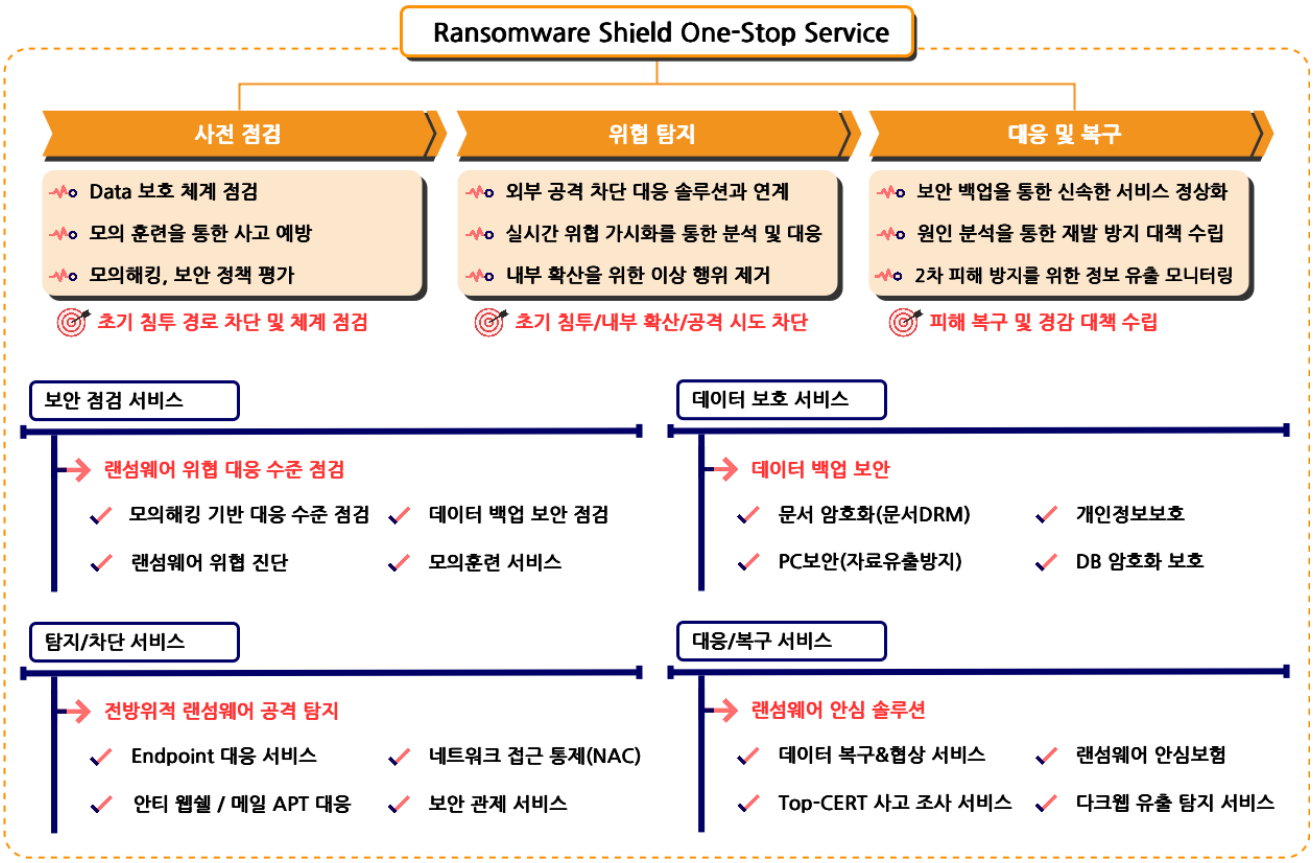
```
struct struct_enc // size = 0x148
{
    int overlapped_status;              // IOCP 상태 플래그 또는 결과 코드
    int write_block_size;               // 현재 쓰기 블록 크기
    int cur_block_offset_low;           // 현재 블록 오프셋 (하위 32비트)
    int cur_block_offset_high;          // 현재 블록 오프셋 (상위 32비트)
    int reserved;                       // 예약 필드
    int h_target_file;                  // 대상 파일 핸들
    int p_original_filename_buf;        // 원본 파일명 버퍼 포인터
    int p_encrypted_filename;           // 암호화된 파일명 포인터
    int target_file_size_low;           // 대상 파일 크기 (하위 32비트)
    int target_file_size_high;          // 대상 파일 크기 (상위 32비트)
    int next_block_offset_low;          // 다음 블록 오프셋 (하위 32비트)
    int next_block_offset_high;         // 다음 블록 오프셋 (상위 32비트)
    int p_encryption_buffer;            // 암호화된 데이터 버퍼 포인터
    int enc_flags;                      // 플래그 (전체 암호화 여부 등)

    char sha_first16_expand_key[176]; // SHA512 결과의 앞 16바이트 기반 AES 키 확장 결과

    int sha_second16;                   // SHA512 결과의 두 번째 16바이트 값
    char reserved_EC[12];               // 예약 필드
    inc_footer footer;                  // 암호화된 파일 끝에 추가되는 INC footer
}

struct inc_footer // size = 0x50
{
    char file_public_key[32];       // 암호화에 사용된 파일 공개키
    char footer_marker[4];          // 고정 마커 "INC\0"
    char reserved[28];              // 예약 공간
    int init_flag;                  // 암호화 처리 여부를 나타내는 초기화 플래그 (1 = 암호화됨)
    int encryption_block_size;      // 블록 단위 암호화 크기 (e.g., 0x100000 = 1MB)
    int reserved_flag;              // 예약 플래그
    int encrypted_block_count; // 암호화된 블록 수
};
```

랜섬웨어 Mitigations

1. 랜섬웨어 대응방안 안내

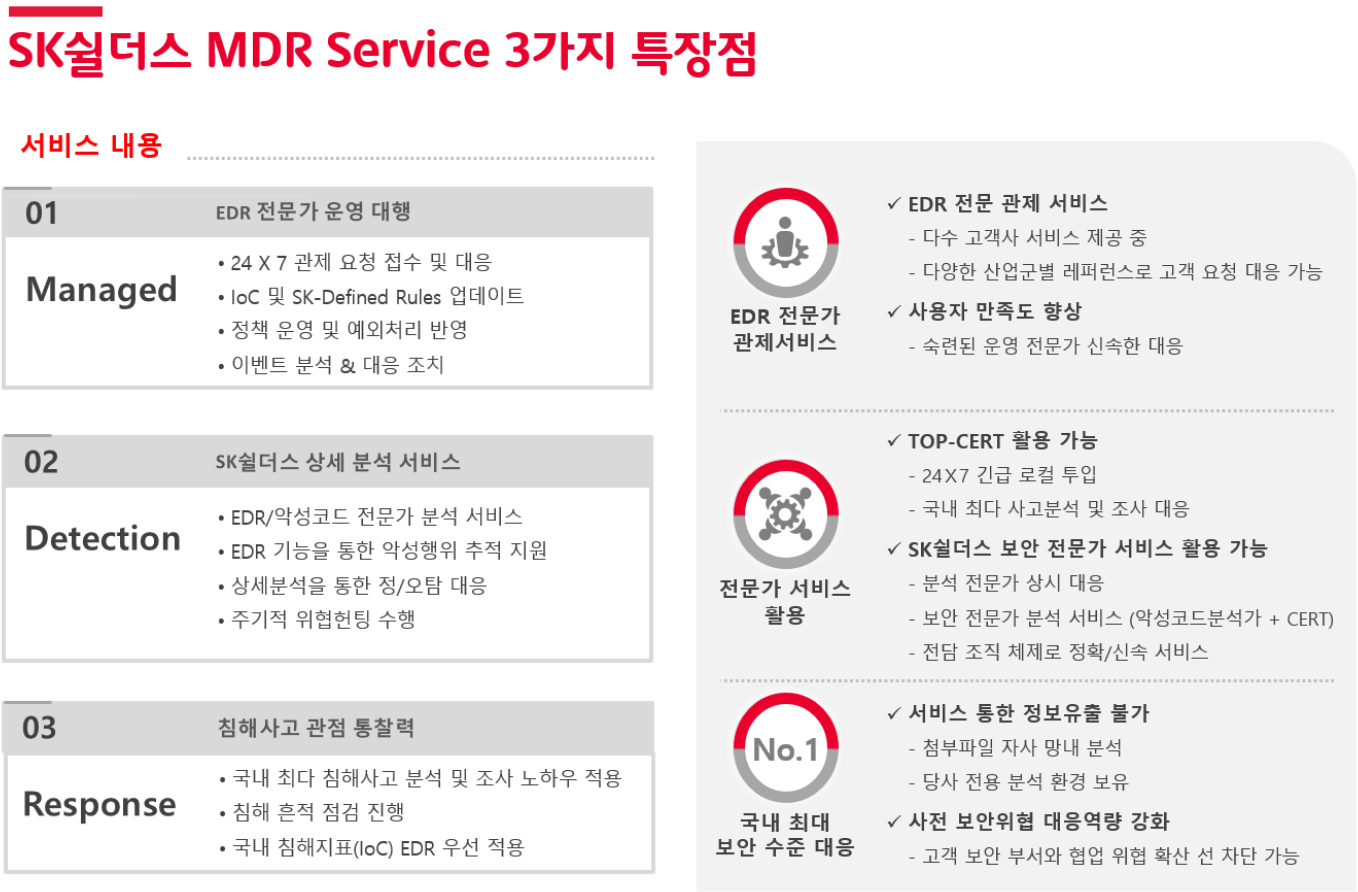
이번 분기에는 Play 랜섬웨어가 Cisco ASA 방화벽의 취약점과 CLFS 권한 상승 제로데이(CVE-2025-29824)를 연계해 침투와 권한 상승을 수행했으며, RansomHub의 인력을 흡수한 것으로 추정되는 Qilin은 공격 건수가 두배 가까이 증가하며 높은 활동량을 보였다. 주요 랜섬웨어 그룹들의 공격들은 네트워크 인프라의 취약점과 인증 미흡 지점을 노린 초기 침투 후, 시스템 권한 획득 및 데이터 탈취, 협상 실패 시 유출까지 이어지는 이중 갈취 전략을 따른다. 이에 대응하기 위해서는 방화벽과 VPN 장비 등 외부 노출 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.



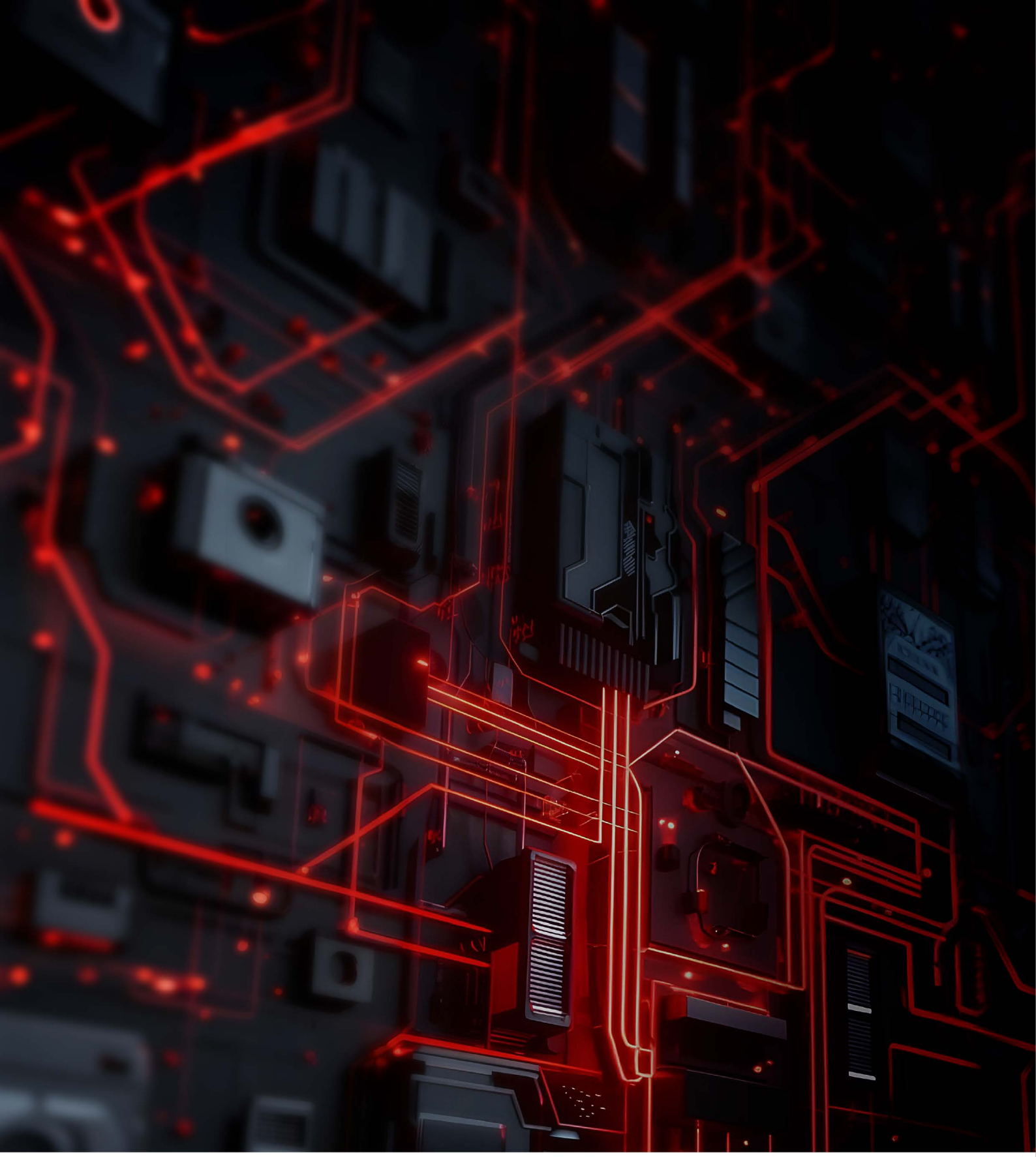


2. SK 쉴더스 MDR 서비스

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