SK 쉴더스

Threat Intelligence Report

EQST INSIGHT

EQST(이큐스트)는 'Experts, Qualified Security Team' 이라는 뜻으로
사이버 위협 분석 및 연구 분야에서 검증된 최고 수준의 보안 전문가 그룹입니다.

2025
08

SK 쉴더스

Contents

Headline

Shadow AI : 제조업 기밀 보호를 위한 탐지·통제·거버넌스 전략

사이버반도체사업그룹 반도체보안전략팀 송원준 수석

1. 제조업 환경에서 Shadow AI 가 제기하는 전략적 보안 위협

2023 년 이후, 대규모 언어모델(LLM) 기반 생성형 인공지능(Generative AI)의 상용화는 산업 전반에 걸쳐 업무 자동화, 엔지니어링 최적화, 지식 정제 등 다방면의 혁신을 가속화 시키고 있다. 특히 제조업에서는 제품 설계, 품질 관리, 공정 제어, 생산성 향상 등 다양한 기능 영역에서 AI 의 활용성이 빠르게 부각되고 있다. 그러나 이러한 기술적 진보는 새로운 보안 지형도 형성하고 있다. 그 중심에는 Shadow AI 라는 고위험 사용 패턴이 존재한다.

Shadow AI 는 조직의 공식 승인 없이, 개인 또는 부서 단위에서 AI 서비스를 비공식적으로 사용하는 행위를 의미한다. 특히 제조업과 같은 지식집약적 산업에서는 산업 기밀, 생산 레시피, 라우팅 정보, 설계 도면, 설비 로직 등 핵심 자산이 무단으로 외부에 전송될 경우, 중대한 보안 위협으로 작용할 수 있다. 이는 기존 보안 인프라(DLP, EDR, CASB 등)의 탐지 범위를 우회하고, 조직 내 보안 가시성(Visibility)을 현저히 저하시키는 결과로 이어질 수 있다.

예컨대, R&D 조직의 엔지니어가 LLM 기반 챗봇에 CAD 설계 내용을 설명하며 기술적 피드백을 요청하거나, 제조기술팀이 내부 공정 데이터를 기반으로 레시피 최적화 질문을 입력하는 사례다. 이 때 문제점은 입력값 대부분이 HTTPS 기반의 비정형 API 트래픽으로 전송돼, 조직 내부의 감사 및 접근 통제 범위를 벗어난다는 점이다. 해당 프롬프트가 외부 AI 서비스의 학습 데이터로 활용되거나 장기 저장될 경우, 향후 동일 산업군 AI 학습에 그대로 재사용될 위험이 있다.

또한 Shadow AI 는 정보 유출의 문제를 넘어 규제 불이행, 법적 분쟁, 산업 보호법 위반 등의 2 차적 리스크의 가능성을 초래한다. 특히 산업기술보호법, GDPR, ITAR 등 국내외 보안·기밀 관련 규제 체계에서는 기밀의 관리성 상실 자체를 보호 요건 박탈의 사유로 보기 때문에, 단 1 회의 외부 전송만으로도 특허 자산의 보호 지위를 상실할 수 있다.

따라서 Shadow AI 는 단순한 '사용자 행위 이슈'가 아닌, 산업지식 기반 보안 전략의 구조적 취약지점으로 간주해야 하며, 사전탐지, 행위제어, 프롬프트 단위의 리스크 평가체계 등 통합적인 거버넌스 모델의 수립이 절실하다.

본 Insight 에서는 Shadow AI 의 작동 구조와 제조업 특화 위협 시나리오를 고찰하고, 기술적 탐지 방안과 정책적 대응 전략을 포괄하는 실효성 있는 보안 모델을 제안하고자 한다. 더불어, 글로벌 규제 흐름과 대응 가이드라인을 기반으로, 실무 중심의 운영체계 수립을 위한 참조 프레임워크도 함께 제시한다.

2. Shadow AI 개념과 위협 모델 분석

2.1 Shadow AI 정의 및 행위 특성

Shadow AI 는 조직 내 보안·IT 관리 체계를 거치지 않고 개인 또는 부서 단위에서 비인가된 생성형 AI 도구(LLM, Vision AI, AutoML 등)를 활용하는 사용 행태를 의미한다. 이 과정에서 사용자는 보안이나 데이터 처리 규정을 충분히 인지하지 못한 채, 다음과 같은 행위를 무심코 수행하게 된다.

- 내부 문서, 도면, 공정정보 등을 외부 AI 에 프롬프트 형태로 직접 입력
- 외부 AI 가 생성한 코드·문서를 검증 없이 업무 시스템에 통합
- 기업 외부 서버에 민감 데이터를 자동 저장/캐싱되는 구조를 이해하지 못함

이러한 Shadow AI 의 사용은 표면적으로는 업무 생산성 향상 및 개인 업무 보조를 목적으로 하나, 보안적 관점에서는 비인가 채널을 통한 고위험 데이터 전송 행위로 간주된다.

2.2 Shadow AI 위협 모델 분류 (제조업 중심)

다음은 Shadow AI 의 제조업 환경 내 위협 유형을 행위/위험/영향/예시 중심으로 설명해본다.

① 설계/기술 문서 유출

요소	설명
행위	CAD 도면 설명, 제품 설계 구조 요약 요청 등
위험	설계 노하우, 부품 규격, 포지셔닝 정보가 LLM 에 노출됨
영향	유사 제품 모방, 경쟁사 기술 학습에 악용 가능
예시	“이 구조가 전체적으로 문제가 없나요?”라는 질문에 도면 전체 구조 포함

② 제조 레시피 및 공정 조건 노출

요소	설명
행위	AI 에게 공정 조건 조정, 수율 개선 방식 문의
위험	생산 온도, 속도, 재료 비율 등의 내부 변수 전송
영향	품질 경쟁력 상실, OEM/ODM 경쟁자에 정보 이전
예시	“이 원료 비율에서 결함이 생기는 원인을 분석해줘” 등

③ 품질 데이터 기반 민감 정보 유출

요소	설명
행위	결함 발생 DB, 검사 이미지, 불량 유형 등을 AI 에 입력
위험	제품 결함 데이터와 구조적 약점 정보가 외부에 학습됨
영향	취약 제품 식별 → 악의적 리콜 유도 가능
예시	“이 사진은 왜 B 등급 불량이었는지 설명해줘” 등

④ 자동화 코드/시퀀스 유출 및 오염

요소	설명
행위	PLC 제어 코드, 시퀀스 논리 등을 AI 에 진단 요청
위험	코드 로직 유출 또는 AI 생성 코드에 보안 미흡 로직 존재
영향	설비 정지, 안전사고 유발, OT 공격 확산 가능
예시	AI 가 생성한 코드에 인증 절차 누락 → 외부 명령 삽입 가능

⑤ 사용자 인증정보/시스템 정보 간접 유출

요소	설명
행위	LLM 에 개발 코드나 API 예시 제공
위험	토큰, 계정명, 시스템 포트 구조 노출
영향	공격자에게 내부 API 맵 제공하는 결과 초래
예시	“이 API 로 품질 관리 시스템 연동하는 방법 알려줘” 등

⑥ 학습 재사용에 의한 정보 역추출

요소	설명
행위	LLM 에 반복적으로 내부 정보 포함한 프롬프트 입력
위험	향후 타 사용자 프롬프트에 의해 해당 정보가 생성 응답으로 재노출
영향	정보가 공개된 것과 동일한 효과, 기밀성 상실
예시	“지난번에 입력한 공정 레시피를 다시 보여줘”와 유사한 요청

⑦ 규제 및 컴플라이언스 위반

요소	설명
행위	AI 가 위치한 국외 서버로 기밀 전송 (GDPR, ITAR, 산업기술보호법 등 위반 소지)
위험	규제 미준수, 법적 소송, 인증 취소 위험
영향	기업 이미지 훼손 및 대외 계약 상실
예시	국방 부품 제조사의 설계 도면이 OpenAI 에 전송됨

위 사례 등으로 볼 때 Shadow AI 는 다음과 같은 복합적인 성격을 지닌다.

- Low-intent, High-impact : 사용자 의도는 선의일 수 있으나, 결과는 치명적일 수 있음
- Technical Undetectability : 프롬프트 내 정보는 구조적으로 식별·분류하기 어려움
- Governance 외부성 : 전통적인 정보보호 관리체계 밖에서 행위 발생
- Attack Surface 확장자 : 외부 API/모델 호출이 사실상 새로운 경계면이 됨

2.3 문제점 도출

제조업 조직에서 Shadow AI 는 단순한 직원의 부주의가 아니라, 보안 거버넌스 체계의 구조적 결핍을 드러내는 경고 신호다. 공격자 없이도 내부에서 자산이 탈취되고, 누출된 정보를 되찾을 수 없다는 점에서 이는 비가역적(irreversible) 보안 사고로 간주해야 한다.

따라서 Shadow AI 에 대한 탐지, 예방, 억제 및 사고 대응은 선택이 아닌 디지털 제조 시대의 필수적 보안 전략 항목으로 자리잡아야 한다.

3. 기술적 대응 전략 : 탐지, 통제, 차단 체계 (Detection, Control, Mitigation)

3.1 탐지 전략 (Detection)

Shadow AI 탐지에는 가시성 확보가 핵심이다. HTTPS 기반 AI API 호출, 동적 도메인, 비정형 프롬프트를 효과적으로 식별하기 위해 아래 대응체계를 권장한다.

① 설계/기술 문서 유출

- AI 플랫폼 호출은 SNI(Server Name Indication), User-Agent, DNS(Domain Name System) 요청 패턴으로 식별 가능
- 고급 CASB(Cloud Access Security Broker) 솔루션을 통해 외부 LLM(Large Language Model) API 호출에 대한 실시간 탐지 및 정책 제어 적용
- 단, 기존 CASB 는 Shadow AI 용 탐지가 미비하므로, AI 흐름 포착이 가능한 DSPM(Data Security Posture Management) 기능이 필요

② 프롬프트 컨텐츠 기반 이상 요청 탐지

- "설계", "기밀", "공정", "매출" 등 고위험 키워드 검출을 적용 등 민감한 데이터가 포함 정책 적용
- 자연어 비정형 요청에 대한 AI-aware DLP(Data Loss Prevention)나 프롬프트 인젝션 탐지 규칙 적용

③ Shadow AI 도구 인텔리전스 및 목록화

- 비공식 ChatGPT, Gemini 외에도 Perplexity, DeepSeek 같은 신규 툴 탐지 강화
- DNS, IP, User-Agent 기반 블랙리스트 및 APT(Advanced Persistent Threat) 예방 수준의 탐지 베이스 구축

3.2 통제 전략 (Control)

탐지 이후에는 엄격한 접근 제어 정책과 내부 대체 모델 제공을 통해 Shadow AI 사용을 구조적으로 관리

① RBAC 기반 AI 사용 권한 제한

- RBAC(Role-Based Access Control) 방식으로 부서별 사용 권한 차등 적용
- 설계/R&D 직무는 대외 AI 사용을 차단하고, 마케팅 등은 안전한 요약 기능만 허용
- '최소권한' 원칙에 준하는 정책 수립 및 관리 자동화

② 프록시 차단 및 AI SaaS 블랙리스트

- SaaS(Software as a Service) 형태의 AI 서비스 접근을 HTTPS 프록시에서 차단
- 신규 AI 서비스 자동 탐지 기능으로 가시성 및 통제 범위 확대

③ 사내 프라이빗 LLM 환경 운영

- Azure OpenAI Private Endpoint 등 내부 모델로 전환 유도
- 외부 접속을 사전 통제하여 인프라 경계 내에서 보안 거버넌스 체계 유지

④ AI-aware DLP 기반 민감정보 실시간 필터링

- PII(Personally Identifiable Information), IP(Intellectual Property), mCAD(manufacturing CAD data) 등 민감 정보 탐지
- AI 특화 DLP 제품 등 활용 등

3.3 차단·억제 전략 (Mitigation)

차단 단계는 Zero Trust 기반 데이터 흐름 통제, 사용자 인식 제어 강화, 그리고 사후 대응 체계 수립 포함

① Zero Trust 기반 프롬프트 경로 통제

- ZTNA(Zero Trust Network Access) 기반 인증·허용 방식으로 외부 LLM 요청 경로 제어
- 내부망 → 인터넷 출구 → 외부 AI 서비스까지 '데이터 전송' 단위로 분석 및 제한

② Security Nudging: 경고 UI 및 정책 기반 인식

- KPI(Key Performance Indicator): 부서별 AI 사용 건수, 탐지 횟수, 정책 위반 상승 추이
- 정기 보고로 조직 내 인식 강화 및 책임 공유 체계 확립

③ 탐지 지표 KPI 기반 모니터링/경영 보고

- RBAC(Role-Based Access Control) 방식으로 부서별 사용 권한 차등 적용
- 설계/R&D 직무는 대외 AI 사용을 차단하고, 마케팅 등은 안전한 요약 기능만 허용
- '최소권한' 원칙에 준하는 정책 수립 및 관리 자동화

④ 사고 대응 준비 – 프롬프트 저장, 백업, 분석

- SOC(Security Operation Center) 내 Prompt/Response 로그 분석을 위한 연동
- 사고 발생 시 누출 범위, API 사용 기록, 사용자 ID 추적 등 포함

4. 거버넌스 및 정책 중심의 조직 대응 체계

Shadow AI 의 위협은 기술적 대응만으로는 불완전하다. 사내 구성원의 무지 또는 관행적 사용, 정책 부재 등이 복합적으로 얽혀 있기 때문에, 조직 전반의 보안 의사결정 체계(거버넌스)를 통한 전사적 관리체계 구축이 필요하다.

4.1 Shadow AI 정책 수립 프레임워크

① AI Usage Policy (생성형 AI 사용 정책)

- 모든 구성원이 명확히 이해할 수 있도록 Shadow AI 금지/허용 기준을 문서화
- 필수 포함 항목으로 어떤 AI 도구를 사용할 수 있는가? (허용/제한/금지 목록) / 어떤 데이터가 입력되어서는 안 되는가? (PII, CAD, 설계서, 소스코드 등 예시 제공) / 위반 시의 제재 수준 및 예외 승인 절차 등을 수립

② AI Risk Classification (업무 기반 리스크 등급화)

- 부서/직무/업무 프로세스별로 AI 사용 위험도 등급을 부여하여 관리
- 등급별로 차등적 승인/통제 체계를 수립 (RBAC + AI Usage Scope Matrix)

③ AI 사용 승인 프로세스

- 신규 AI 도구 사용 요청 시, 보안팀 또는 AI 거버넌스 위원회의 사전 심의 필요
- API 통신, 브라우저 확장 기능, 내부망 접속 요청에 대한 기술적 심사 프로세스 운영
- 예외 사용 시 관리자 승인 절차 필수화

4.2 교육 및 조직 인식 제고 전략

Shadow AI 의 80% 이상은 '보안 인식을 하지 못한 채 무심코 사용'한 것으로 분석된다. 따라서 단순한 차단이 아닌 전사 차원의 인식 개선 프로그램이 필수적이다.

① AI 보안 인식 교육 프로그램

- 정기 교육 (반기 1회 이상) 및 전파용 콘텐츠 제작

② 프롬프트 작성 가이드 배포

- '절대 입력 금지 문장 예시' 중심의 현장 실무형 가이드 문서화

③ 보안 책임제 제도 운영

- 각 부서 내 보안 리더 지정 → AI 사용 모니터링, 캠페인 진행, 이슈 리포트
- 보안팀과의 소통 창구 역할 수행
- 보안 이슈 공유 채널 상시 운영

4.3 AI 거버넌스 조직 모델

① AI 사용 통제 전담 TF (AI Risk Control Taskforce)

- 구성 : 보안팀(CISO), 정보팀(CIO), 법무, 내부통제, 각 실무 부서 대표
- 역할 : 사내 AI 도구 화이트리스트 관리 / Shadow AI 탐지 리포트 주간 공유 / 신규 정책 및 위반 대응 협의

② AI Risk Steering Committee

- 경영진 보고 체계로 기능, 리스크 등급 상승 시 빠른 의사결정 구조
- KPI : Shadow AI 탐지율, 위반 건수, 보안 가이드 수강률 등

③ 감사 및 내부통제 연계 안 책임제 제도 운영

- AI 사용 행위에 대한 내부 감사 항목 도입
- 보안 로그, 프롬프트 사용 이력, 외부 접속 이력 등 정기 리포트화

4.4 산업 기준 및 컴플라이언스 대응

제조업 내 보안 거버넌스를 강화하는 동시에, 국내외 법적·산업 기준과의 연계도 필수적이다.

규제 기준	적용 항목	대응 방안
ISO/IEC 42001	생성형 AI 운영 시 거버넌스 체계 수립	AI 위험 등급 분류, 위원회 운영
NIST AI RMF	AI 리스크 관리 프레임워크	Shadow AI 리스크 대응 포함
KISA AI 보안 가이드	국내 산업 기반 AI 보안 권고안	AI 프롬프트 필터링, 민감정보 탐지 포함
GDPR/개인정보보호법	자동화 처리 및 민감정보 유출	AI 입력 사전 탐지 및 마스킹 도입

5. 결론 및 대응 로드맵 제안

5.1 결론

Shadow AI 는 단순한 IT 통제를 넘어, 조직의 기밀정보와 경쟁력을 직접 위협하는 신종 보안 리스크로 부상했다. 특히 제조업 기반 기업의 경우, 설계도면, 공정 노하우, 원가 자료와 같은 산업기밀(Intellectual Property)이 LLM(Large Language Model) 기반 AI 도구를 통해 외부로 유출될 가능성이 높아지는 상황이다.

이 Insight 에서는 이러한 Shadow AI 위협에 대해 기술적, 정책적, 거버넌스 차원의 대응 전략을 통합적으로 제시하였다. 대응의 핵심은 다음과 같다.

- 탐지 : AI-aware DLP, CASB, DSPM 등을 활용한 AI 사용 가시성 확보
- 통제 : AI 사용 정책 수립, 프록시 차단, 역할기반 권한관리(RBAC)
- 차단 : Zero Trust 기반 경로 통제, 경고 UI 및 사후대응 체계
- 거버넌스 : 전사 정책 수립, 부서 리스크 등급화, 교육 및 내부감사 체계화

이러한 대응은 단발성 정책이 아닌, 조직문화와 보안 거버넌스 체계 내재화를 목표로 추진되어야 한다.

5.2 대응 로드맵 제안

다음은 Shadow AI 대응을 위한 3 단계 실행 로드맵이다.

[1 단계 : 가시화 및 인식 재고]

- 목표 : Shadow AI 의 존재와 위협 파악
- 주요 조치 사항
  → Shadow AI 의 존재와 위협 파악
  → 사내 Shadow AI 사용 현황 조사
  → Shadow AI 사고 사례 교육자료 배포
  → 부서별 민감정보 분류 체계 수립

[2 단계 : 정책 및 기술적 통제 수립]

- 목표 : Shadow AI 사용 통제 및 최소화
- 주요 조치 사항
  → AI 사용 정책 수립 및 공지
  → RBAC 기반 AI 사용 권한 설정
  → 민감정보 DLP 정책 적용 및 테스트
  → 프록시 기반 LLM 접속 차단 체계 구축

[3 단계 : 조직 내재화 및 거버넌스]

- 목표 : 대응 체계의 조직 내 정착
- 주요 조치 사항
  → AI 거버넌스 위원회 및 보안 책임제 제도 운영
  → AI 사용 모니터링 및 정기 리포트화
  → AI 보안 인식 정기 교육
  → AI 관련 컴플라이언스 대응체계 정비 (ISO, NIST 등)

5.3 향후 과제 및 제언

- 사내 LLM 도입 검토 : 보안 위험 없이 생성형 AI를 활용할 수 있는 프라이빗 LLM 환경을 구축하고, 이를 통해 외부 Shadow AI 사용을 줄일 수 있음
- AI 특화 보안 솔루션 확대 도입 : 기존 보안 장비만으로는 LLM 의 비정형성 탐지가 어렵기 때문에, AI-aware DLP, Prompt 보안 필터링, 데이터 흐름 탐지 기술 도입이 필요
- 보안팀의 역할 전환 : Shadow AI 대응은 단순 감시가 아닌 'AI 활용 가이드 및 보안 조언자'로서의 역할 확대가 중요함
- 법·규제 대응 체계 고도화 : 생성형 AI 관련 국내외 규제가 빠르게 정비되고 있으므로, 대응 전담조직 및 감사 항목 통합이 요구됨

Shadow AI 는 단순한 기술 도입의 문제가 아닌, 기밀 보호와 조직의 생존을 좌우하는 보안 과제다. 기술, 정책, 문화가 함께 작동하는 다층적 대응이 필요한 시점이다.

Shadow AI로부터 조직의 산업기밀을 지키기 위한 보안 정책 수립이 필요하다면, SK쉴더스가 보유하고 있는 기술 및 정책 노하우를 기반으로 AI 보안 거버넌스를 시작하길 바란다.

■ 참고문헌

[1] Structured, Shadow AI – The Hidden Threat to Governance & Compliance, 25.04

[2] Inteleca, Shadow AI in the Workplace: The Hidden Security and Compliance Risks, 25.03

[3] CIODIVE, AI-generated code leads to security issues for most businesses, 24.01

[4] Nightfall AI, The Nightfall Approach: 5 Ways Our Shadow AI Coverage Differs from Generic DLP, 25.07

[5] NIST AI RISK MANAGEMENT FRAMERK (AI RMF), 23.01

■ 참고 자료

[1] Paloalto, What Is Shadow AI? How It Happens and What to Do About It (Cyberpedia)

[2] ISO/IEC 42001:2023, Information technology — Artificial intelligence — Management system

[3] 행정안전부, 공공기관 AI 보안 가이드라인, 23.10

[4] NIPA, 산업별 생성형 AI 활용과 보안 위협 보고서, 2024

[5] SK쉴더스, EQST Insight 블로그 시리즈 (2023~2024)

Keep up with Ransomware

국내 금융권을 공격한 Gunra 랜섬웨어

■ 개요

2025 년 7 월 랜섬웨어 피해 사례 수는 지난 6 월(512 건)에 비해 하락한 483 건을 기록했다. 피해 건수는 다소 감소했지만, 공격의 정교함과 전략적 다양성은 오히려 강화되는 양상을 보였다. 특히, 취약점을 악용한 침투, AI 기반 협상 자동화 시도, 그리고 법 집행과의 직접적 충돌 사례 등은 7 월의 주요 특징으로 분석된다.

지난 4 월 등장한 Gunra 그룹이 국내 금융기관을 대상으로 공격을 감행한 정황도 확인됐다. 피해 기관은 7 월 14 일 랜섬웨어 공격으로 인해 서비스 제공에 일시적인 차질을 빚었으며, 약 4 일 만에 복구를 완료하고 서비스를 재개했다. 그러나 피해는 서비스 중단에 그치지 않고 데이터 유출로 확산됐다. 공격자는 다크웹 유출 사이트에 해당 기관의 데이터베이스를 탈취했다고 주장하며, 함께 분석할 인원을 모집하는 메시지를 게시했다. 유출된 데이터는 총 13.2TB 분량의 압축 파일로 확인되었으며, 공격자는 순차적 공개를 예고하며 피해 기관에 대한 압박 수위를 높이고 있다.

Akira 그룹은 다중 인증(MFA)[1]이 적용된 환경에서도 패치가 적용된 SonicWall SSL-VPN[2] 장비를 통해 침투한 것으로 보인다. 이로 인해, 제로데이(Zero-Day) 취약점 존재 가능성에 대한 우려가 제기되고 있다. 이는 취약점이 공식적으로 공개되거나 패치되기 전까지 방어가 어려운 고도화된 위협이 여전히 존재함을 시사한다. 또 다른 사례로는 Microsoft SharePoint 의 ToolShell 취약점(CVE-2025-53770)을 악용한 Warlock 랜섬웨어의 확산이 확인되었다. 해당 공격은 미국 내 핵안보청(NNSA), 국립보건원(NIH) 등 주요 정부기관을 포함한 약 400 여 대의 서버에 피해를 입혔다.

공격자들은 기술적인 정교함뿐 아니라 운영 전략 측면에서도 진화를 거듭하고 있다. Global 그룹은 피해 기업과의 협상 과정에 AI 챗봇을 도입하여 인터페이스를 자동화하고, 협상 속도를 높이려는 시도를 보였다. 이러한 움직임은 서비스형 랜섬웨어 RaaS[3] 생태계의 서비스화와 자동화 트렌드를 반영하는 대표적 사례로 분석된다.

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

[2] SSL-VPN(Secure Sockets Layer Virtual Private Network): 인터넷을 통해 암호화된 채널로 내부 네트워크에 원격 접속할 수 있도록 하는 장비

[3] RaaS (Ransomware-as-a-Service): 랜섬웨어를 서비스 형태로 제공해서 누구나 쉽게 랜섬웨어를 만들고 공격할 수 있도록 하는 비즈니스 모델

공격자 활동에 대응하기 위한 법 집행 기관의 움직임도 더욱 강화되는 양상이다. 미국 연방수사국(FBI)은 Chaos 랜섬웨어 그룹 구성원이 보유한 약 240 만 달러 상당의 비트코인을 압수하기 위한 법적 절차에 착수했으며, 7 월 25 일에는 FBI, 유로폴(Europol), 독일 및 네덜란드 경찰 등 국제 수사기관이 협력해 BlackSuit 랜섬웨어 그룹의 다크웹 유출 사이트를 압수하는 데 성공했다. 해당 그룹은 180 건 이상의 피해자를 사이트에 게시했으며, 총 요구 금액은 약 5 억 달러(한화 약 6,948 억 원)에 달하는 것으로 알려졌다.

한편, 법 집행의 제재에도 불구하고 활동을 재개한 집단도 확인됐다. 지난 2 월과 6 월, 프랑스 사이버범죄수사국(BL2C)이 주요 운영자 5 명을 체포하며 폐쇄됐던 해킹 포럼 BreachForums 는 7 월 26 일 복구됐다. 운영자 공지에 따르면, 체포된 인물들은 실질적 관리자 권한이 없었으며, 실제 운영진의 신원을 숨기기 위한 역할로 직함만 부여된 인물이었다고 주장했다. 또한 지난 4 월, 포럼 소프트웨어인 MyBB 의 취약점으로 인해 일시적으로 운영을 중단한 것은 사실이나, 해당 취약점은 이미 해결되었으며 기존 도메인은 법 집행 기관의 요청에 따라 폐쇄했다고 덧붙였다.

반면, 러시아 해킹 포럼 XSS 는 여전히 폐쇄된 상태를 유지하고 있다. 유로폴을 포함한 국제 법 집행 기관은 7 월, 우크라이나에서 해당 포럼의 운영자로 추정되는 인물을 체포했다. 체포된 인물은 약 20 년간 사이버 범죄 생태계에서 활동하며, 광고 및 중개 수수료를 통해 약 700 만 유로(한화 약 80 억 원)를 벌어들인 것으로 확인되었다. 해당 체포 이후, XSS 포럼은 복구되지 않은 상태다.

■ 랜섬웨어 뉴스

Gunra 그룹, 국내 금융기관 대상 랜섬웨어·데이터 유출 공격

- 지난 7월 14일, 국내 금융기관 랜섬웨어 공격으로 서비스 중단 발생
- 서비스 재개에도 불구하고 데이터베이스(13.2TB) 유출 발생
- 탈취 자료 분석 완료 및 공개 예고

Chaos 그룹 대상 암호화폐 자산 압수 및 범죄 수익 회수 절차 추진

- 미국 FBI는 Chaos 랜섬웨어 그룹이 보유한 약 240만 달러 상당의 비트코인 자산에 대해 압수를 위한 법적 절차에 착수
- 해당 암호화폐는 랜섬웨어 피해자 지갑에서 이체된 자금으로, 미국 법무부는 몰수 추진
- 범죄 수익을 직접 겨냥한 이번 조치는 인프라 차단을 넘어선 법 집행의 확장 사례로 주목됨

BlackSuit 그룹에 대한 인프라 차단 및 국제 공조 작전

- FBI, 유로폴, 독일 및 네덜란드 경찰 등 국제 수사기관은 공조 작전을 통해 BlackSuit 그룹의 다크웹 유출 사이트를 압수
- 인프라를 직접 차단한 이번 조치는 랜섬웨어 생태계에 실질적 타격을 가한 사례로 주목받음

BreachForums, 법 집행 압박 속 재등장

- 운영자 체포로 폐쇄됐던 해킹 포럼 BreachForums, 7월 26일 복귀
- 운영자는 체포된 인물들이 실제 관리자 권한을 가진 적이 없으며, 주목을 끌기 위해 직함만 부여했다고 주장.
- 보안 취약점 패치·도메인 교체 완료 후 운영 재개

법 집행으로 폐쇄된 해킹 포럼 XSS

- 유로폴 등 법 집행기관, 7월 XSS 포럼 운영자 우크라이나서 체포
- 운영자는 20년간 활동하며 약 700만 유로(약 80억 원) 수익 올린 것으로 추정
- 체포 작전 이후 XSS 포럼 폐쇄 상태 지속


Akira 그룹 SonicWall SSL VPN 침해
Akira 랜섬웨어가 최신 펌웨어가 적용된 SonicWall SSL VPN 장비를 통해 내부망에 침투한 사례가 발견
MFA 설정이 활성화된 환경에서도 접근이 이루어진 정황이 있어, 알려지지 않은 취약점의 존재 가능성이 제기됨
공식적인 취약점(CVE)은 아직 공개되지 않음
SharePoint 취약점을 악용한 Warlock 랜섬웨어 배포
Storm-2603 으로 추정되는 공격자의 SharePoint 취약점(CVE-2025-53700) 악용
해당 취약점으로 Warlock 랜섬웨어 유포, 약 400대 서버 피해 발생
Global 그룹 협상 자동화를 위한 AI 챗봇 도입 시도
협상 인터페이스에 AI 챗봇을 도입해 피해 기업과의 소통을 자동화하는 시도 진행
협상 과정 단축 및 응답 속도 향상을 통해 금전 갈취 효율성 극대화 시도

그림 1. 랜섬웨어 동향

■ 랜섬웨어 위협

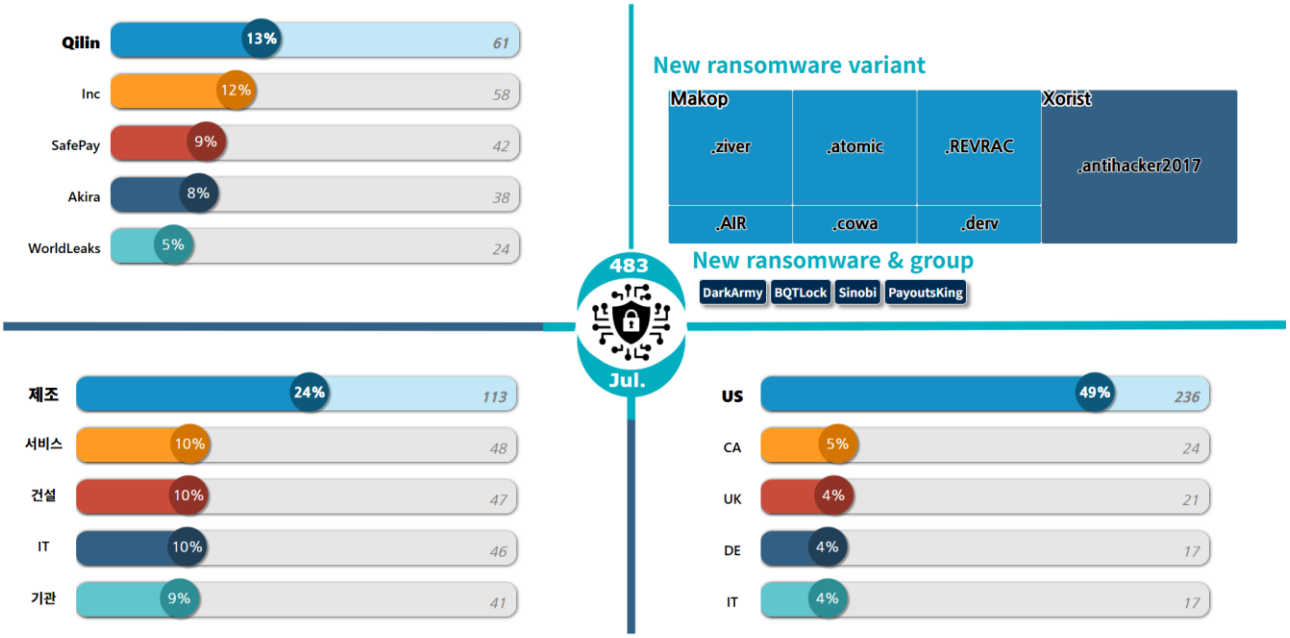


그림 2. 2025 년 7 월 랜섬웨어 위협 현황

새로운 위협

7 월 동안 총 483 건의 피해 사례가 확인됐으며, 이 기간 동안 신규 랜섬웨어 그룹 4 개가 새롭게 등장했다. 각 그룹은 자체 운영 중인 데이터 유출 사이트에 피해 사실을 게시했으며, 확인된 피해 규모는 DarkArmy 11 건, BQTLock 2 건, Sinobi 5 건, Payoutsking 18 건으로 집계됐다.

그림 3. DarkArmy 의 데이터 유출 사이트

DarkArmy 의 다크웹 데이터 유출 사이트 하단 배너에 睡觉的乌鸦(잠자는 까마귀)라는 중국어 표어와 연락 수단으로 QQ·위챗 계정이 병행 표기돼 있다. 종합적으로 추정하면, 개발·운영 주체가 중국어권일 가능성이 높다.

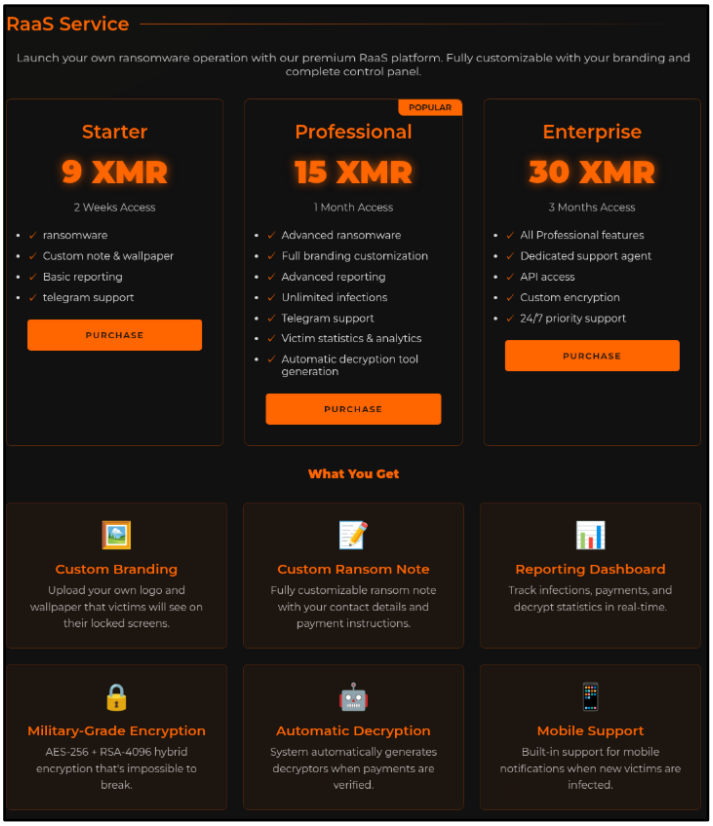


그림 4. BQTLock RaaS 대시보드

BQTLock 는 BQT RaaS 라는 자체 포털을 운영하며, 가입자에게 완전한 커스텀 빌더 [4] 와 통계 대시보드를 제공한다. 포털에는 Starter, Professional, Enterprise 의 3 단계 요금제가 명시되어 있으며 각각 9XMR(한화 약 337 만원), 15XMR(한화 약 562 만원), 30XMR(한화 약 1,124 만원)의 비용을 요구한다. Professional 이상 구독 시 랜섬노트 브랜딩 커스터마이징, 피해자 통계·보고서, 자동 복호화 툴 생성 기능이 활성화된다. 이러한 원스톱 RaaS 플랫폼 구조는 비개발자 공격자까지 빠르게 랜섬웨어 캠페인을 구축·운영하도록 지원해 시장 확산을 가속화할 것으로 보인다.

[4] 빌더(builder): 공격자가 랜섬웨어의 암호화 방식, 몸값, 피해자 메시지, 타깃 경로 등 세부 옵션을 GUI 또는 명령형 인터페이스로 설정하고, 최종 실행 파일을 자동 생성해주는 도구

Top5 랜섬웨어

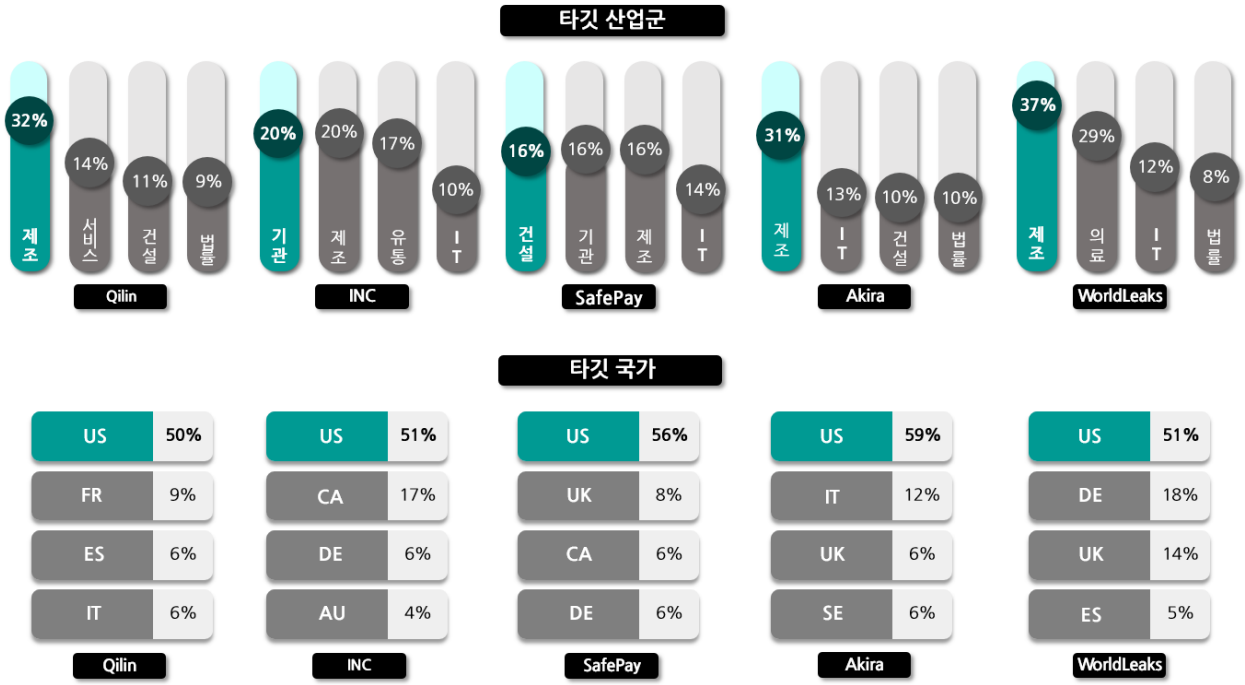


그림 5. 산업/국가별 주요 랜섬웨어 공격 현황

Qilin 그룹은 7 월 29 일 말레이시아의 식품 원료 제조업체 Custom Food Ingredients 를 공격해 핵심 제조 시스템에 침투하고 내부 데이터를 탈취했다고 주장했다. 이들은 다크웹 유출 사이트에 생산·운영 관련 문서 일부의 목록을 게시하며 피해 기업에 대한 압박 수위를 높였다.

Inc 그룹은 7 월 31 일 미국 웨스트 버지니아의 공공 의료기관 West Virginia Primary Care Association 에게 피해를 입혔다고 공개했다. 다크웹 유출 사이트에 피해 사실을 게시하고 연락을 요구했으며, 협상 불발 시 데이터를 공개하겠다고 경고했다. 또한 미국 버지니아주 Albemarle County 행정기관을 공격해 주민 및 직원의 이름, 주소, 사회보장번호(SSN), 운전면허번호 등 수천 건의 개인정보를 유출했다.

SafePay 그룹은 7 월 초 글로벌 IT 유통사 Ingram Micro 를 공격했다고 주장했다. 사건 직후 기업의 웹사이트와 주문 시스템이 일시 중단됐으며, 다크웹 유출 사이트에 피해 기업명을 등록하고 3.5TB 규모의 자료 공개를 예고했다. 이어 7 월 26 일에는 미국 피부과 병원 Southwest Florida Dermatology 를 공격해, 환자의 의료 기록을 포함한 민감한 내부 데이터를 유출했다고 밝혔다.

Akira 그룹은 7 월 25 일 미국 오클라호마시티에 위치한 지식재산 전문 로펌 Dunlap Codding 의 자료를 탈취했다고 주장했다. 다크웹 유출 사이트에는 고객 파일, 재무 자료, 특허 및 법원 관련 문서 등 약 19GB 규모의 자료 공개 예고문이 게시되었다. 또한 스페인의 온라인 뷰티 유통업체 Druni 역시 공격을 받아, 직원 신분증, 재무 기록, 고객 정보 등이 포함된 40GB 분량의 데이터가 유출되었다.

WorldLeaks 그룹은 7 월 21 일 미국의 엔지니어링 서비스 기업 Proactive Engineering Consultants 를 공격했다고 주장했다. 이후 다크웹 유출 사이트를 통해 피해 사실을 공개하고, 5.3TB 에 달하는 설계 및 프로젝트 관련 자료를 유출했다. 이어 미국 건설사 Thomas Bennett & Hunter 를 공격해 사내 프로젝트 및 운영 관련 데이터를 공개했다.

■ 랜섬웨어 집중 포커스

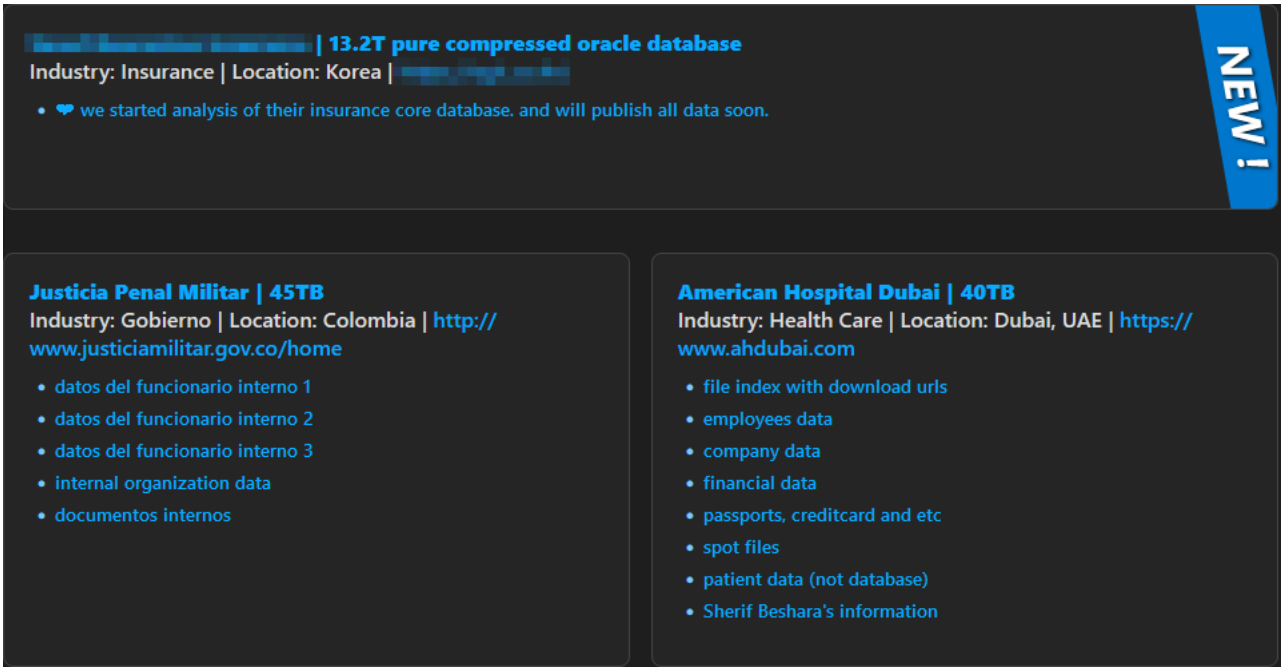


그림 6. Gunra 다크웹 유출 사이트

Gunra 랜섬웨어 그룹은 2025 년 4 월 처음 확인되었으며, 현재까지 총 16 건의 피해자를 전용 데이터 유출 사이트에 게시했다. 전용 사이트는 Tor 네트워크에서 운영되며, 피해자별 기업명·산업·국가와 함께 탈취한 데이터의 종류, 게시 일시, 협상 마감 시한을 명시한다. 이들은 피해자별로 탈취한 데이터의 종류와 게시 일시를 공개하고, 협상이 결렬되거나 정해진 기한이 지나면 해당 자료를 다크웹에 전면 공개하는 방식을 사용한다.

이들은 짧은 협상 기한 설정과 다중 익명 채널 활용이 특징이며, 랜섬노트에는 피해자가 5 일 이내에 연락해야 한다는 기한 강조 문구와 함께 Tox ID 와 메일 주소를 기재해 피해자와 접촉한다. 초기에는 일부 파일의 무료 복호화를 제안한 후, 응답이 없으면 유출 사이트에 피해자 정보를 게시한 뒤 탈취 자료 일부를 공개하며 압박 수위를 높인다. 협상이 장기화하면 추가 데이터를 공개하거나 자료 전체 유출을 경고하는 등 단계적으로 피해자를 위협한다. 게시된 피해자 중에는 2025 년 7 월 발생한 국내 금융권 기업도 포함되어 있으며, 피해 기업의 자료 공개를 예고하며 강도 높은 압박을 가했다.

현재까지 확인된 Gunra 랜섬웨어는 Windows 와 Linux 두 버전이 존재한다. Windows 버전은 암호화 후 디렉터리별로 랜섬 노트를 남기며, Linux 버전은 랜섬 노트를 생성하지 않고, 실행 인자로 지정된 경로와 확장자, 암호화 비율을 바탕으로 파일을 선별해 암호화한다. 두 버전 모두 파일 크기나 유형에 따라 전체 또는 부분 암호화를 적용해 효율성과 속도를 높이는 것이 특징이다. 본 보고서에서는 두 버전을 분석해, Gunra 의 활동 방식과 기술적 특징을 정리함으로써 랜섬웨어 위협에 효과적으로 대비할 수 있도록 하고자 한다.

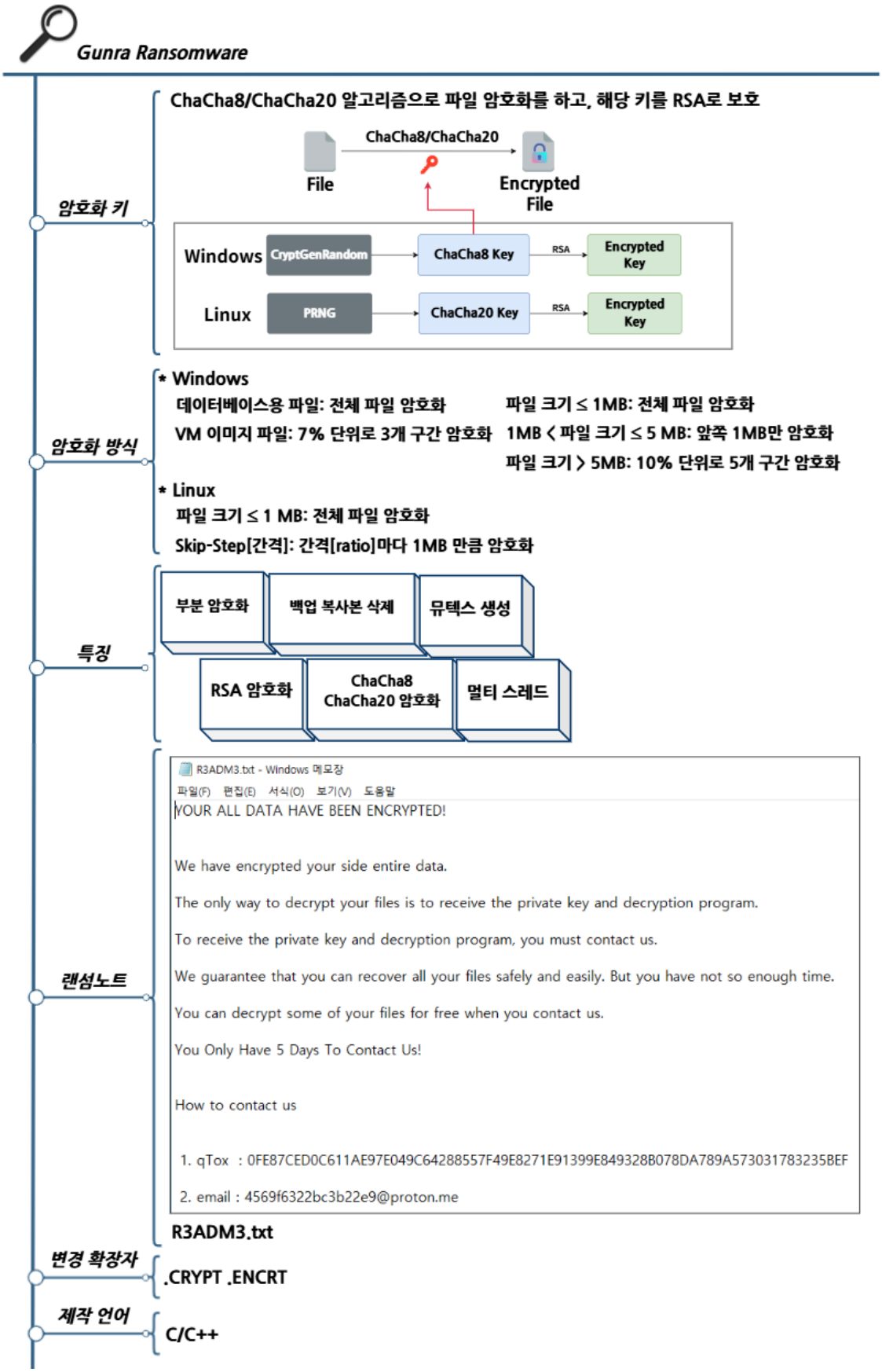


그림 7. Gunra 랜섬웨어 개요

Gunra 랜섬웨어 전략

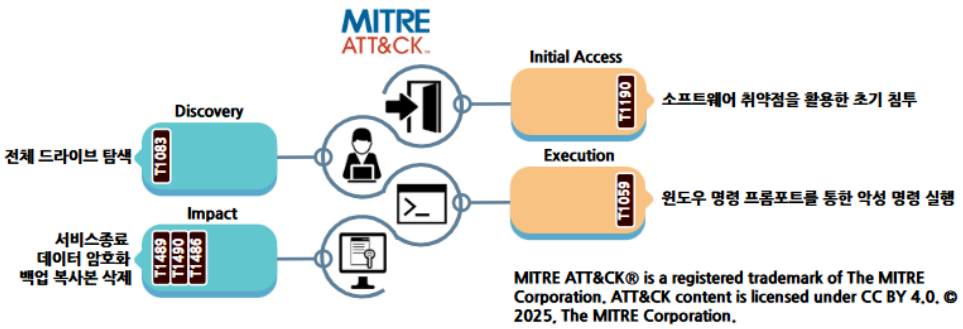


그림 8. Gunra 랜섬웨어 공격 전략

Gunra 랜섬웨어 (Linux 버전)

Gunra 랜섬웨어의 Linux 버전은 다양한 실행 인자를 통해 암호화 동작을 정밀하게 제어할 수 있도록 설계되어 있어, 공격자는 목표 파일의 위치, 확장자, 암호화 강도, 암호화 키 저장 방식 등을 유연하게 설정할 수 있다. Linux 버전의 인자와 기능은 아래 표와 같다.

구분	설명
--threads / -t	파일 암호화 스레드 개수 지정
--path / -p	암호화 대상 지정
--exts / -e	암호화 대상 파일의 확장자 지정(all: 모든 파일, disk: 블록 디바이스)
--ratio / -r	암호화 간격 설정 (MB)
--keyfile / -k	RSA 공개키 파일(.pem) 경로
--store / -s	암호화 키 저장 경로
--limit / -l	최대 암호화 크기 (GB, 0: 전체 파일 암호화)

표 1. Gunra 랜섬웨어 Linux 버전 실행 인자

Linux 버전의 경우 -p 인자를 통해 암호화 대상 파일 또는 경로를 지정한다. 함께 입력한 대상이 단일 파일일 경우 해당 파일에 대해서 암호화를 수행하며, 디렉터리일 경우 해당 디렉터리와 하위 디렉터리를 탐색해 암호화 대상 파일을 암호화한다.

-e 옵션은 암호화 대상 파일의 확장자를 지정하며, 이 옵션이 없거나 'all'로 설정된 경우 파일 내부에 명시된 예외 확장자를 제외한 모든 파일이 암호화 대상이 된다. 또한 'disk'로 설정된 경우 시스템에 존재하는 블록 디바이스 파일만 암호화한다. 암호화 예외 대상 파일 확장자에는 이미 암호화 완료된 파일의 확장자인 .ENCRT 와 랜섬 노트 파일인 R3ADM3.txt 가 포함된다.

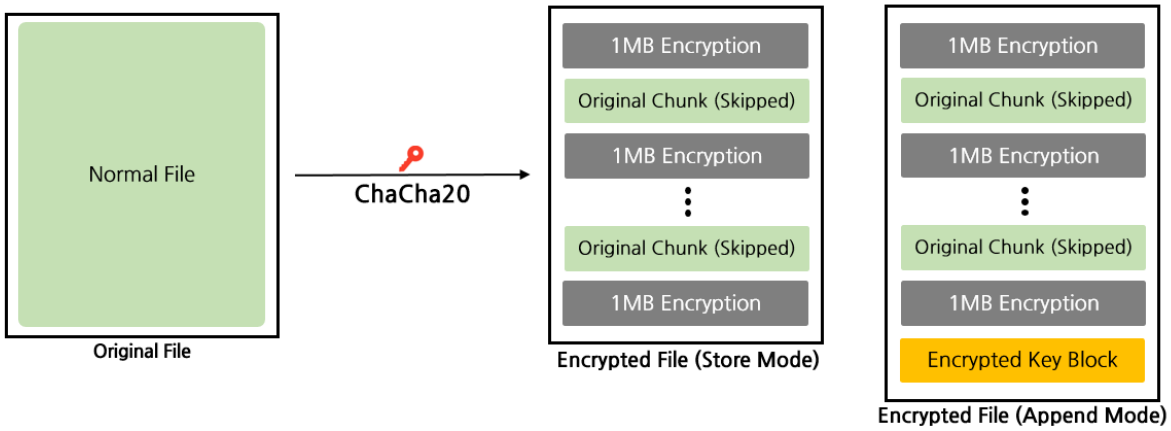


그림 9. Gunra 랜섬웨어 리눅스 버전 암호화

Gunra 랜섬웨어의 리눅스 버전은 파일 암호화에 ChaCha20 알고리즘을 사용한다. 암호화 대상 파일은 –p, -e 인자로 지정한 경로와 확장자에 따라 선정되며, 암호화는 -r 인자값에 기반한 부분 암호화 방식으로 진행된다. Gunra 는 모든 파일을 1MB 단위로 암호화하며, -r 인자를 사용해 1MB 로 암호화 이후 건너뛸 구간의 크기를 설정한다. 예를 들어 -r=5 의 경우, 1MB 를 암호화한 뒤 5MB 를 건너뛰고, 다시 1MB 를 암호화하는 방식으로 반복 수행된다. 이때 건너뛰는 간격만 달라질 뿐, 모든 암호화 구간의 크기는 고정적으로 1MB 이다. 또한 -l 인자를 사용해 GB 단위로 최대 암호화 크기를 지정할 수 있다. 파일 암호화 후, ChaCha20 에 사용된 키, Nonce[5] , -r, -l 인자를 따로 저장하는데, -s 인자의 유무에 따라 저장 방식이 달라진다.

-s 인자를 사용하면 Store Mode 가 적용되어 암호화된 키 블록이 지정한 디렉터리에 별도로 저장된다. 이 과정에서 랜섬웨어는 해당 경로가 존재하는지 확인한 뒤, 원본 파일명을 기반으로 [Filename].keystore 형태의 키 파일을 생성한다. -s 인자를 사용하지 않았다면 Append Mode 가 적용되어 암호화된 키 블록이 암호화된 파일의 끝부분에 추가된다.

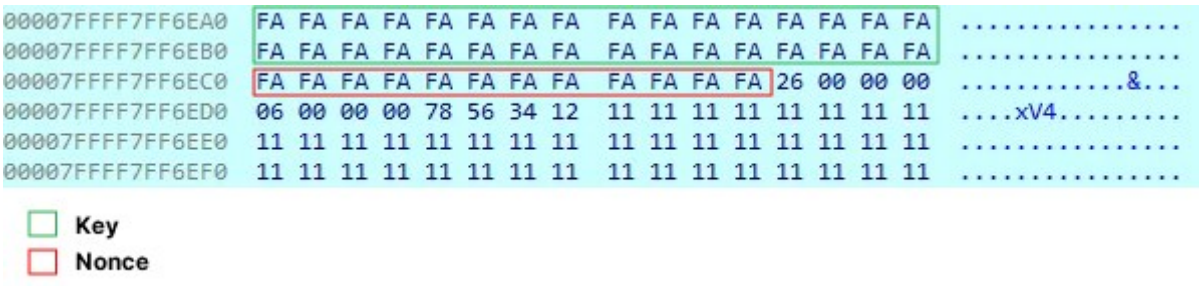


그림 10. Gunra 랜섬웨어 Linux 버전의 취약한 키 생성

[5] Nonce: 암호화에서 보안성과 고유성을 위해 사용되는 임의의 값

추가로, Linux 버전에서는 암호화 키 생성 과정 중 설계상 취약점이 확인됐다. 일반적으로는 암호화 키와 Nonce 를 예측할 수 없도록 충분히 랜덤한 값을 생성하고, 이를 비대칭 키로 암호화해 공격자만 복호화할 수 있게 한다. 그러나 Gunra 의 Linux 버전은 1Byte 단위로 랜덤 값을 생성해 이어 붙이는 비효율적인 방식을 사용한다. 이 과정에서 매 바이트마다 time(0) 함수를 호출해 초 단위의 현재 시간을 시드(seed)로 설정하는데, 32Byte 키와 12Byte Nonce 를 생성하는 데 1 초도 걸리지 않아 동일한 시드로 생성된 값이 반복될 가능성이 높다. 만약 생성 도중 시간이 바뀌더라도 시드 변화 폭이 작아, 공격자가 비교적 쉽게 키와 Nonce 를 예측할 수 있는 상황이 발생한다.

Gunra 랜섬웨어 (Windows 버전)

Gunra 랜섬웨어 Windows 버전은 Linux 버전과 달리 별도의 실행 인자 없이 동작하도록 설계되어 있으며, 실행 중 중복 감염을 방지하기 위한 뮤텍스 이름이나, 암호화에 사용한 키를 보호하기 위한 RSA 공개키 등의 중요한 값들이 내부에 포함되어 있다.

랜섬웨어가 실행되면 중복 실행 방지를 위해 '375345635adfwef39'이라는 이름의 뮤텍스를 생성한다. 이후 시스템 전체 드라이브를 순차적으로 탐색해 랜섬노트를 생성하고, 암호화 대상을 확인한다. 이 때, C 드라이브 내 사용자 폴더와 그 하위 폴더만 탐색하고 나머지 드라이브는 최상위 디렉터리부터 탐색을 진행한다. 암호화 과정에서 특정 폴더명과 확장자 및 파일명은 제외하고 암호화하며, 확인된 예외 대상은 아래 표와 같다.

폴더명	확장자 및 파일명
tmp, winnt, temp, thumb, $Recycle.Bin, $RECYCLE.BIN, System Volume Information, Boot, Windows, Trend Micro	.exe, .dll, .lnk, .sys, msi, R3ADM3.txt, CONTI_LOG.txt

표 2. Gunra 랜섬웨어 Windows 버전 암호화 예외 대상

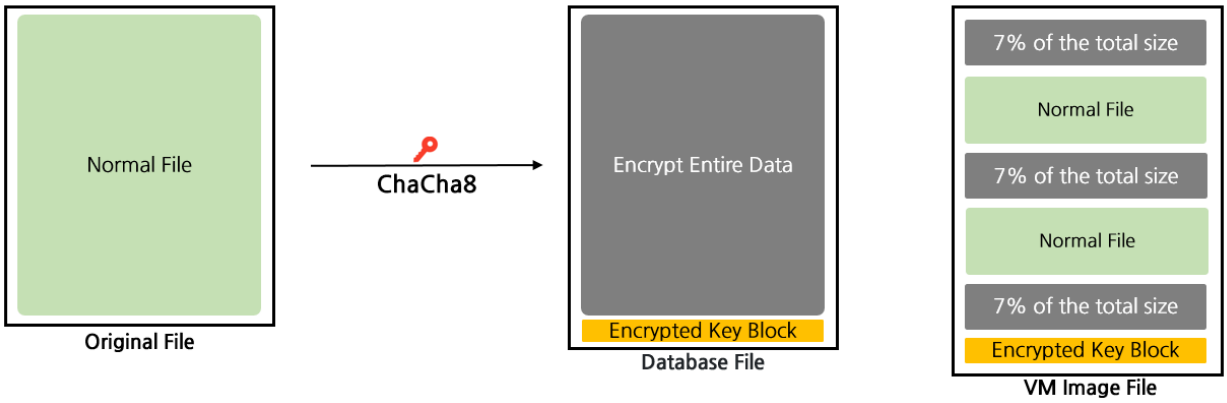


그림 11. Gunra 랜섬웨어 Windows 버전 암호화 방식(파일 확장자 기준)

파일 암호화 방식은 파일의 확장자와 크기에 따라 결정된다. 데이터베이스와 관련된 파일은 크기와 무관하게 전체를 암호화하며, 가상 머신(VM) 이미지와 관련된 파일은 크기와 관계없이 파일의 맨 앞과 뒤, 그리고 중간을 각각 전체 크기의 7% 만큼씩 암호화해 총 21%만 암호화한다. 각각 해당하는 파일 확장자는 아래 표와 같다.

데이터베이스 관련 확장자	VM 관련 확장자
.4dd, .4dl, .accdb, .accdc, .accde, .accdr, .accdt, .accft, .adb, .ade, .adf, .adp, .arc, .ora, .alf, .ask, .btr, .bdf, .cat, .cdb, .ckp, .cma, .cpd, .dacpac, .dad, .dadiagrams, .daschema, .db, .db-shm, .db-wal, .db3, .dbc, .dbf, .dbs, .dbt, .dbv, .dbx, .dcb, .dct, .dcx, .ddl, .dlis, .dp1, .dqy, .dsk, .dsn, .dtsx, .dxl, .eco, .ecx, .edb, .epim, .exb, .fcd, .fdb, .fic, .fmp, .fmp12, .fmpsl, .fol, .fp3, .fp4, .fp5, .fp7, .fpt, .frm, .gdb, .grdb, .gwi, .hdb, .his, .ib, .idb, .ihx, .itdb, .itw, .jet, .jtx, .kdb, .kexi, .kexic, .kexis, .lgc, .lwx, .maf, .maq, .mar, .mas, .mav, .mdb, .mdf, .mpd, .mrg, .mud, .mwb, .myd, .ndf, .nnt, .nrmlib, .ns2, .ns3, .ns4, .nsf, .nv, .nv2, .nwdb, .nyf, .odb, .oqy, .orx, .owc, .p96, .p97, .pan, .pdb, .pdm, .pnz, .qry, .qvd, .rbf, .rctd, .rod, .rodx, .rpd, .rsd, .sas7bdat, .sbf, .scx, .sdb, .sdc, .sdf, .sis, .spq, .sql, .sqlite, .sqlite3, .sqlitedb, .te, .temx, .tmd, .tps, .trc, .trm, .udb, .udl, .usr, .v12, .vis, .vpd, .vvv, .wdb, .wmdb, .wrk, .xdb, .xld, .xmlff, .abcddb, .abs, .abx, .accdw, .adn, .db2, .fm5, .hjt, .icg, .icr, .kdb, .lut, .maw, .mdn, .mdt	.vdi, .vhd, .vmdk, .pvm, .vmem, .vmsn, .vmsd, .nvram, .vmx, .raw, .qcow2, .subvol, .bin, .vsv, .avhd, .vmrs, .vhdx, .avdx, .vmcx, .iso

표 3. 데이터베이스 및 VM 관련 확장자

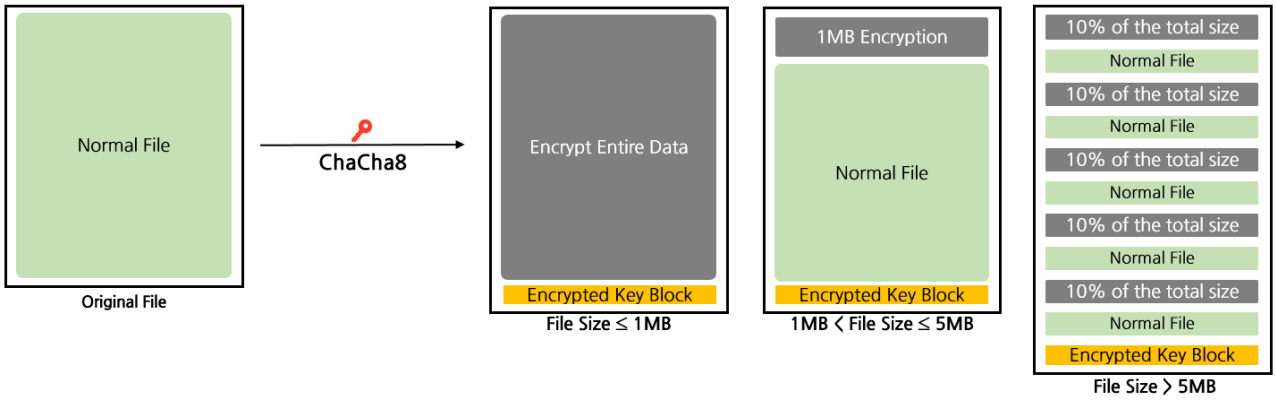


그림 12. Gunra 랜섬웨어 윈도우 버전 암호화 방식

가상머신과 데이터베이스 관련 파일을 제외한 나머지 파일은 크기에 따라 암호화를 수행한다. 파일 크기가 1MB 이하인 경우 전체 파일을 암호화하고, 1MB 를 초과하면서 5MB 이하인 경우에는 파일의 상위 1MB 만 암호화한다. 5MB 를 초과하는 파일은 전체 파일을 10% 크기의 블록으로 나누어 홀수 번째 블록만 암호화한다.

파일 암호화 후 파일의 맨 뒤에 복구에 필요한 데이터를 추가한다. 암호화에 사용한 ChaCha8 키, Nonce, 원본 파일 크기는 물론 어떤 방식으로 암호화를 진행했는지 구분하기 위한 2Bytes 크기의 식별자를 공격자의 RSA 공개키로 암호화한 뒤 추가한다.

Gunra 랜섬웨어는 시스템 복원 기능을 무력화해 사용자의 복구 가능성을 차단한다. 이를 위해 VSS(Volume Shadow Copy Service) 목록을 조회한 뒤 각 항목을 삭제하는 작업을 수행한다. 이 과정에서 랜섬웨어는 WMI[6] 를 통해 SELECT * FROM Win32_ShadowCopy 쿼리를 실행하여 시스템 내 모든 볼륨 섀도 복사본을 검색하고, 조회 결과에서 각 볼륨 섀도 복사본의 고유 ID 값을 추출한다. 이후 해당 ID 값을 이용해 다음 명령어를 생성·실행하여 해당 섀도 복사본을 삭제한다.

```
cmd.exe /c C:\Windows\System32\wbem\WMIC.exe shadowcopy where "ID='%s'" delete
```

표 4. VSC 삭제 명령어

[6] WMI(Windows Management Instrumentation): 윈도우 운영체제의 구성 요소, 상태, 동작 정보를 표준화된 방식으로 조회·관리할 수 있도록 하는 관리 인터페이스

Gunra 랜섬웨어 대응방안

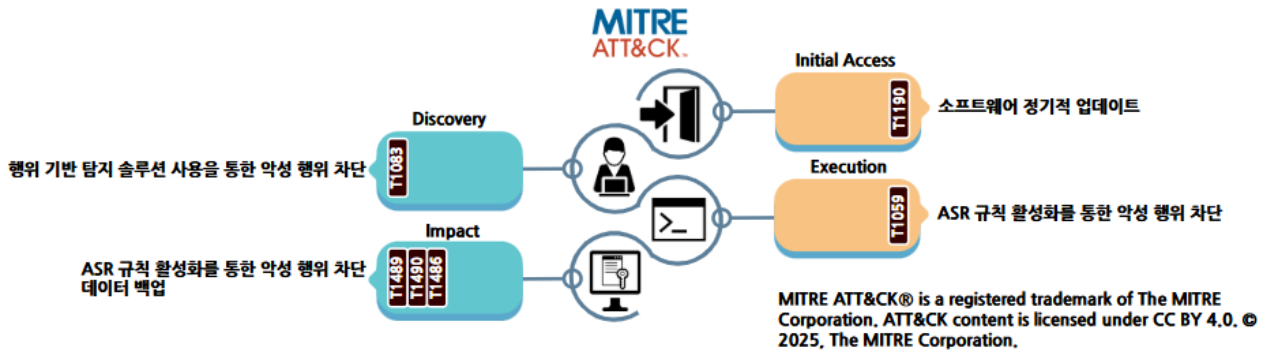


그림 13. Gunra 랜섬웨어 대응방안

Gunra 랜섬웨어의 Windows 버전은 Windows 명령 프롬프트를 활용해 시스템 내 백업 복사본을 삭제한 뒤 파일 암호화를 수행한다. 이에 따라 ASR 규칙을 활성화하면, 백업 삭제 및 암호화와 관련된 비정상적인 프로세스를 사전에 차단하여 악성 행위를 효과적으로 방지 가능하다. 특히, 시스템 복원 지점 삭제와 같은 행위를 탐지·차단할 수 있는 환경 구성이 중요하다. 사전에 정책을 정교하게 설정하고, 불필요한 스크립트 실행 시도를 즉시 차단하는 것도 피해 예방에 큰 도움이 된다.

또한 EDR 솔루션을 도입하고 최신 보안 패치를 적용하여, 알려진 취약점 악용을 통한 침투나 로컬 실행 기반의 비정상적인 동작을 신속히 식별·차단할 수 있도록 해야 한다. 이를 통해 파일 암호화 과정에서 발생하는 행위 기반 패턴을 실시간으로 탐지하고, 악성 프로세스의 실행을 중단할 수 있다. 더불어, EDR·백신·로그 분석 시스템을 연계하여 경고 이벤트를 중앙에서 모니터링하면, 다수의 단말에서 동시다발적으로 발생하는 공격에도 신속한 대응 체계를 갖추게 된다.

아울러, 백업 복사본을 별도의 네트워크 구간이나 외부 저장소, 오프라인 매체에 주기적으로 분산 백업해 두면, 시스템이 암호화되더라도 데이터 복구가 가능하다. 이때 백업 장치 접근 권한을 최소화하고, 정기적으로 복구 테스트를 실시하여 백업 데이터의 무결성을 보장하는 것이 중요하다. 추가로 별도의 네트워크나 저장소의 데이터를 분산해 백업하거나 백업 스케줄과 보관 주기를 다양화해 랜섬웨어의 삭제 시도를 피하는 것도 효과적인 방법이다.

이러한 대응 전략은 Windows 환경뿐 아니라 Linux 버전의 Gunra 랜섬웨어에도 동일하게 적용 가능하다. Linux 환경에서는 서버 등 핵심 인프라를 직접 노리는 경우가 많으므로, OS 특성에 맞는 접근 제어 정책과 서비스 포트 제한, 관리자 계정 관리 강화가 필수적이다. 운영체제와 관계없이 다계층 보안 체계를 적용하면, 랜섬웨어 감염 시 피해를 최소화하고 신속한 복구를 보장할 수 있다.

IoCs

Hash(SHA-256)
91F8FC7A3290611E28A35A403FD815554D9D856006CC2EE91CCDB64057AE53B0
22C47EC98718AB243F2F474170366A1780368E084D1BF6ADCD60450A9289E4BE

■ 참고 사이트

• ArcticWolf (https://arcticwolf.com/resources/blog/arctic-wolf-observes-july-2025-uptick-in-akira-ransomware-activity-targeting-sonicwall-ssl-vpn/)

• Microsoft (https://www.microsoft.com/en-us/security/blog/2025/07/22/disrupting-active-exploitation-of-on-premises-sharepoint-vulnerabilities/)

• Thehackernews (https://thehackernews.com/2025/07/newly-emerged-global-group-raas-expands.html)

• Securit affairs (https://securityaffairs.com/180578/cyber-crime/fbi-seizes-20-btc-from-chaos-ransomware-affiliate.html)

Special Report

제로트러스트 보안전략 : 시스템 (System)

SI/솔루션사업그룹 보안 SI 사업팀 황병권 책임

■ 시스템(System) 필러 개요

제로트러스트 아키텍처에서 시스템 필러는 중요 응용프로그램을 구동하거나 핵심 데이터를 저장·관리하는 서버 전반을 의미한다. 즉, 물리·가상 서버, 하이퍼바이저 위의 VM, 데이터베이스, 파일 서버, DB 서버, 컨테이너·쿠버네티스 노드, 퍼블릭 클라우드 인스턴스까지 모두 범위가 해당된다.

시스템 필러에 제로트러스트 아키텍처를 적용할 때 가장 크게 고려할 점은 다양한 시스템(서버) 운영환경이다. 예전처럼 온프레미스 위주로 구성되어 있던 시스템(서버)이 이제는 퍼블릭·하이브리드 클라우드와 프라이빗 클라우드로 확장되어 운영되고, VM·컨테이너 단위로 빠르게 생성·변경·이관되고 있다. 환경 다각화로 계정, 접근 경로, 설정, 패치, 백업 등과 같은 시스템 영역의 관리 항목이 늘어날 뿐 아니라 통합되지 못한 채 각각 분리된 방식으로 관리되는 경우가 많다. 따라서 시스템 필러의 관리 정책을 표준화하고, 관련 시스템과 연계하여 일관된 통합 관리 체계 구현이 필요하다.

그림 1. 다양한 시스템(서버) 운영환경

제로트러스트 환경에서 시스템 필러는 단순히 서버 개별 보안 강화를 넘어, 서로 다른 환경에서 운영되는 서버들을 공통된 기준으로 묶어 관리하는 데 의미가 있다. 운영체제와 미들웨어가 다양하고, 물리 서버·VM·컨테이너가 혼재된 환경에서는 서버마다 별도의 보안정책을 적용하기 어렵기 때문에, 중앙에서 정의된 정책을 기반으로 통합 관리 체계를 마련하는 것이 핵심이다.

시스템 필러의 접근 방식은 예전처럼 “내부 서버는 신뢰한다”는 전제를 두지 않는다. 접속을 허용한 뒤에도 계정과 세션, 명령·쿼리·변경 작업을 지속적으로 로깅하고 모니터링하여, 필요한 기간과 범위만 권한을 부여하고, 만료 시 자동 회수한다. 서버의 상태(패치 적용, 설정 준수, 취약점·백업 검증 결과 등)도 시스템 계정과 권한을 기반으로, 사용자의 신원과 시스템 상태를 함께 확인해야 한다.

제로트러스트 아키텍처에서 시스템 필러는 단순한 시스템(서버) 역할뿐만 아니라 조직의 가장 중요한 리소스들이 저장되는 공간으로 다양한 환경에 구성되어 있는 시스템들을 식별하고 일괄적으로 관리할 수 있는 체계를 구현해야 한다. 아래의 시스템 필러의 주요 요소와 시스템들이 제로트러스트 환경을 구축하는 기준점이 되어줄 수 있다.

■ 시스템(System) 필러의 주요 요소

시스템 필러는 제로트러스트 아키텍처 내에서 조직의 핵심 자산인 서버, 중요 응용프로그램, 데이터 저장소 등 모든 시스템을 직접적으로 관리·보호하는 중심축으로 기능한다. 온프레미스, 클라우드, 하이브리드 등 다양한 환경에 분산된 시스템은 오늘날 복잡해진 IT 인프라에서 가장 많은 정보와 업무가 직결되는 동시에, 외부·내부 위협의 주요 타깃이 되는 영역이다.

특히, 제로트러스트 환경에서는 단일 시스템에 대한 신뢰도나 위치에 근거한 접근 통제가 불가하며, 모든 시스템 및 내부의 계정, 리소스, 로그, 프로세스에 이르기까지 지속적이고 세밀한 검증과 통합 관리가 필수적으로 요구된다. 시스템 인벤토리, 계정관리, 접근 통제, 보안 정책, 패치 관리, 취약점 관리, 가시성, 시스템 분리 및 정책관리 등 여러 관리적·기술적 요소들이 상호 유기적으로 결합되어야만 조직 전체의 데이터 보호는 물론 업무 연속성, 법적 컴플라이언스까지 아우르는 실질적 보안 수준을 확보할 수 있다.

아래는 시스템 필러의 주요 요소들과 이를 구현하기 위한 구체적인 관리·기술 방안을 제로트러스트 성숙도 관점에서 정리한 내용이다.

1. 시스템 인벤토리

제로트러스트 환경에서 시스템 인벤토리 관리는 조직 내에서 운용되는 모든 서버, 주요 응용 프로그램, 데이터 저장소 등 핵심 시스템을 물리적·논리적으로 식별하고, 항상 최신본으로 유지하는 데서 시작한다. 온프레미스뿐 아니라 클라우드, 하이브리드 환경에 분산된 물리적·가상 서버, 컨테이너, 데이터베이스, 파일서버 등 다양한 시스템을 대상으로 자산관리 체계에 통합적으로 등록하고 목록화해야 한다. 도입 시 단순히 일괄 등록하는 데 그치는 것이 아니라, 시스템의 추가, 변경, 이관, 폐기 등 라이프사이클 전반에 걸쳐 주요 속성(담당자, IP, OS, 역할, 구성상 위치 등)과 상태 정보를 실시간으로 갱신되어야 한다. 이는 곧 정책 자동화의 기반이 된다.

시스템 인벤토리 내 모든 자산은 운영 환경(온프레미스, 클라우드, 하이브리드)과 업무 역할(웹서버, DB 서버, 파일서버 등)에 따라 그룹이 논리적으로 분류되야 한다. 그룹 정보는 수동 문서 관리나 일회성 입력에 머무르지 않고, 시스템의 변화 발생 시 자산관리 시스템과 통합 모니터링 도구가 연동되어 자동으로 분류, 그룹 정책 반영, 최신화가 이루어져야 한다. 이를 통해 각 그룹별 보안 정책, 접근 권한, 모니터링 체계를 일괄적으로 관리하고, 정책 위반이나 이상 징후 발생 시 즉각적으로 식별 및 대응이 가능해진다.

또한, 시스템 구역 정의는 단순 물리적/논리적 분리 이상으로 업무 목적, 데이터 중요도, 네트워크 위치, 보안 요구 수준 등으로 시스템을 다층적으로 세분화해해야 한다.

구역별로 차등화된 접근통제 정책, Micro Segmentation, 세션 단위의 다중 인증(MFA), 실시간 위험 평가 및 정책은 자동화가 적용된다. 또한, 각 구역 간·내의 트래픽 흐름 및 접근 권한도 자산관리 시스템, ICAM, 통합 모니터링 등과의 연계를 통해 자동 조정되고 실시간 가시성이 확보된다.

최적화된 시스템 인벤토리 관리체계에서는 모든 시스템 자산의 상태 변화가 실시간 반영되며, 이 정보를 기반으로 세밀한 권한 통제, 효율적 정책 배포, 신속한 이상 탐지 및 대응, 그리고 체계적인 감사 및 컴플라이언스 관리까지 제로트러스트 실현의 기반이 마련된다.

2. 시스템 계정 관리

제로트러스트 환경에서 시스템 계정 관리는 조직 내에서 운용되는 모든 서버(유닉스, 리눅스, 윈도우 등)와 주요 시스템에 접속할 수 있는 계정 전체를 목록화하고, 용도와 목적에 따라 통합적으로 관리해야 한다. 관리자는 단순히 관리자 권한 계정뿐만 아니라 사용자 권한, 서비스 계정 등 모든 유형의 계정까지 그 속성(사용여부, 권한 레벨, 그룹 소속 등)을 기준으로 분류하고, 변경·삭제 등 라이프사이클 전반을 체계적으로 관리해야 한다.

시스템별로 산재된 계정 정보를 수동 문서 관리에 의존해서는 안된다. 계정 관리 시스템 또는 포털을 통해 모든 계정 정보를 중앙에서 실시간으로 통합 관리할 수 있어야 한다. 각 계정별로 중요한 속성(권한, 소속, 만료, 잠금 등)은 자동으로 반영·갱신되어야 하며, 계정의 생성·변경·삭제 등 상태 변화가 있을 때마다 이상여부를 실시간 확인해 비인가·위험 계정 시 즉각적으로 비활성화 등 조치를 취해야 한다.

계정 관리는 단순 목록화에 그치지 않고 계정별 중요도, 권한, 그룹, 사용 현황 등 다양한 기준에 따라 수동·자동으로 분류·그룹화하고, 분류된 계정 정보가 관련 시스템과 연계해 시스템의 가용성 및 보안성을 함께 개선할 수 있도록 운영해야 한다.

시스템 계정의 보안 설정 역시 중요하다. 비인가 접근 및 계정 오용을 방지하기 위해, 각 계정의 보안 설정(접근 차단, 만료, 최소 권한 등)을 자체 시스템 또는 계정 관리 시스템, ICAM 등과 연동하여 일괄 적용해야 한다. 계정 생성·삭제 시에는 운영체제 종류(리눅스, 윈도우, Mac 등)별로 미리 정의된 보안 정책이 자동 적용되어야 한다. 보안 설정 내역은 통합 모니터링 시스템, 로그 분석 시스템 등과 연계해 실시간 모니터링과 감사를 지원해야 한다.

암호 관리의 경우, 각 계정에는 강력한 암호 정책(최소 길이, 복잡성, 주기적 변경 등)과 중요 계정에 대한 2 차 인증(MFA) 등을 적용해야 한다. ICAM, 인증 시스템, 모니터링 시스템 등과 연계하여 계정 암호 상태와 정책 적용 이력, 변경 내역 등이 중앙에서 통합 관리되어야 하며, 암호 관리 내역은 로그 시스템을 통해 기록하고, SIEM·SOAR 등과 연동하여 위험 계정이나 비정상 변경 행위 발생 시 즉시 식별 및 조치할 수 있어야 한다.

3. 시스템 접근 통제

시스템 접근 통제는 제로트러스트 원칙에 따라 모든 접근 권한을 최소 권한 원칙에 따라 부여해야 하며, 시스템 인벤토리를 참고하여 각 시스템별 목적과 업무 요구에 맞는 세밀한 접근 권한이 설정되어야 한다. 접근 권한 통제는 인증, 허가, 엑세스 제어 등 다양한 구성 요소를 포함해야 한다. 또한, 단일 시스템 내 통제에 그치지 않고 ICAM 등 통합 관리 시스템을 활용하여 네트워크, 애플리케이션과 연계한 전사적 권한 관리도 이루어져야 한다. 각 시스템에는 접근 권한 설정을 통해 특정 IP, Port, 계정 등 다양한 기준으로 접근을 제한하고, 필요에 따라 SSO, IAM 등과 연동하여 IDP 역할을 수행하게 하고 RBAC, ABAC 기반의 세밀한 권한 부여가 이루어져야 한다. 실시간 모니터링 및 분석을 위해서는 ICAM 과 SIEM, SOAR, XDR 등 통합 관제 시스템과 연동하여 접근 권한을 실시간으로 관리하고, 위협 상황에 따라 동적으로 권한이 조정될 수 있어야 한다.

시스템 명령어 통제는 실제 시스템에서 사용되는 명령어 중 위험하거나 취약점이 존재하는 명령어를 식별하고, 통제 대상 명령어와 감사를 위한 정책을 명확히 수립해여야 한다. 단순히 시스템 별 자체 설정이나 특정 쉘의 제한에 그치지 않고, 접근 통제 시스템이나 Secure OS, 통합 관리 시스템 등을 활용해 명령어 통제가 일괄적으로 적용되어야 변경 내역에 대한 추적과 실시간 모니터링이 가능해야 한다. 통제된 명령어의 사용 현황과 이상 징후는 SIEM, SOAR 등과 연동해 실시간으로 분석 및 대응해야 한다. 이로써 정책의 효과성과 적시성을 지속적으로 검증 가능하기 때문이다.

실시간 세션 통제는 시스템 접근의 허용과 차단을 세션 단위로 관리하는 것을 의미한다. 접근 통제 시스템을 통해 각 세션의 생성, 유지, 연장, 만료 등 전 과정을 실시간으로 모니터링하고 관리해야 하며, 이상 행위가 탐지될 경우 즉시 세션을 종료하거나 추가 인증 등 추가 조치를 적용해야 한다. 단순히 시스템별 자체 설정에 의존하지 않고, 계정 관리 시스템 및 접근 통제 시스템과의 연계를 통해 세션 별, 그룹 별, 업무별로 세분화된 정책을 마련해야 한다. 실시간 세션 정보는 통합 관제 및 분석 시스템과 연계하여 전체 보안 수준을 높이는 데 활용한다.

4. 시스템 보안

제로트러스트 환경에서 시스템 보안 관리는 서버 등 주요 시스템에 대한 보안 정책 정의와 실효성 있는 관리 체계 구축을 의미한다. 각 시스템 그룹 또는 자산별로 적용할 보안 정책을 명확히 정의하고, 시스템 자체 보안 기능 설정뿐만 아니라 자산 관리 시스템, 통합 모니터링 시스템 등과 연계하여 정책을 일괄적으로 적용해야 한다. 시스템이 추가되거나 변경될 때는 기존 정책에 따라 보안 설정이 자동으로 반영되고, 실시간 모니터링로 상태 변화에 즉시 대응할 수 있는 체계 마련이 필요하다.

시스템 구성요소와 주요 데이터는 정기적인 백업과 복구 방안을 만들어 운영해야 한다. 하드웨어 고장, 소프트웨어 오류, 침입 공격 등 다양한 위험으로부터 시스템을 보호하기 위해서는 백업 시스템을 활용해 주요 설정 파일, 데이터베이스, 로그 등 핵심 정보를 주기적으로 백업하고, 문제 발생 시 신속하게 복구할 수 있도록 복구 계획을 수립해야 한다. 이중화 구성과 DR 센터 등을 통해 재해 발생 시에도 신속한 복구가 가능하도록 관리하며, 백업 및 복구 정책은 시스템 변경이나 이상 징후 발생 시 자동으로 적용될 수 있도록 해야 한다.

시스템 내부 프로세스에 대해서는 생성, 실행, 모니터링, 종료 등 운영 전반의 과정을 체계적으로 관리해야 한다. 각 시스템별로 중요하거나 위험한 프로세스를 명확히 정의하고, 권한 기반으로 프로세스 생성 및 실행 관리, 그리고 모니터링 시스템과 연동해 사전 지정한 프로세스의 상태를 실시간 감시해야 한다. 비정상 종료나 예상치 못한 행동이 감지될 경우에는 원인 분석과 함께 즉각적인 조치를 취해, 프로세스 이상 징후 알람 등 자동화된 방식으로 관리되어야 한다.

시스템 보안 기능은 운영 및 소프트웨어 업데이트, 중요 파일 무결성, 백신 및 악성 코드 방지, 로그 상태 등 다양한 항목을 주기적으로 점검해야 한다. 수동 체크리스트에 의존하지 않고, 서버 백신, 점검 시스템, 취약점 관리 시스템 등과 연동하여 점검 결과를 실시간으로 가시화하고, 문제가 발견될 경우 즉시 조치할 수 있는 체계를 갖추어야 한다. 점검 항목과 기준은 모니터링 시스템과 머신러닝 등 자동화 도구를 통해 주기적 업데이트·반영되어야 하며, 점검 결과는 리포트 등 다양한 형태로 관리되어야 한다.

5. 시스템 분리

시스템 분리는 시스템 인벤토리를 기반으로 특정 서버 및 중요 시스템을 물리적 또는 논리적으로 분리하여 관리하는 것을 의미한다. 분리 방식에는 실제 물리적 구조 변경이나 망분리, 또는 가상화(VM), 접근제어 정책을 통한 논리적 분리 등이 있으며, 각 시스템의 역할과 중요도, 서비스 특성을 고려해 적용하는 것이 좋다. 시스템 분리를 적용할 때에는 기존 시스템과의 호환성을 유지하면서, 시스템이 추가될 경우에도 유연하게 확장될 수 있도록 설계해야 하며, 분리된 시스템에 대한 별도의 감시, 모니터링, 보안 제어 설정 방안도 함께 마련되어야 한다. 자산 관리 시스템과 연동하여 분리 대상 시스템이 추가되면 각 시스템이 자동으로 라벨링 정책이 상속되고, 논리적·물리적 분리가 체계적으로 되야 한다.

중요도가 높은 특정 시스템에 대해서는 더욱 효과적으로 관리하고 보호할 수 있는 세부적 방안이 필수다. 사고 관리, 변경 관리, 패치 관리, 백업 및 복구 등 다양한 관리 정책과 기술을 적용하여, 시스템 분리 정책에 따라 별도 관리가 필요하며, 서비스별로 웹서비스, DB 인스턴스 등을 각각 물리적 또는 논리적으로 분리하여 체계적으로 운영해야 한다. 중요한 시스템의 상태 변화, 이상 징후 등은 모니터링 시스템을 통해 실시간으로 확인하고, 필요할 경우 자동으로 대응할 수 있는 구조를 구축해야 한다. 주요 시스템이 추가되거나 환경이 변화할 때에도 세부 관리 기술과 정책이 자동 적용되어, 전체 보안 수준이 유지될 수 있도록 별도의 관리 체계를 마련해야 한다.

6. 시스템 정책 관리

시스템 정책 관리는 시스템 환경을 안전하고 효율적으로 운영하기 위해 수립된 관리 정책이 일관되게 적용하는 업무를 의미한다. 시스템 관리 정책에는 시스템 운영, 엑세스 제어, 보안, 문서 관리, 보고 및 분석 등 다양한 기능을 포함해야 하며, 정확성, 완전성, 일관성, 사용자 편의성, 확장성, 보안성 등 비기능의 요구사항도 반드시 충족되어야 한다. 정책 수립 시에는 기존 시스템과의 호환성, 관련 법적·규정 준수 요건을 고려해야 하고, 회사 내규 및 표준 프레임워크를 참고하여 체계적으로 문서화하고 관리해야 한다. 중앙 관리 시스템을 활용해 모든 시스템에 정책이 일관되게 반영·통합 관리되야 한다. 모니터링 시스템 등과 연계해 운영 현황 분석 결과에 따라 정책이 자동 업데이트되고 신규 시스템에도 자동으로 정책이 적용되는 구조를 마련해야 한다.

시스템 정책의 예외 관리도 필수적이다. 예외 정책이 필요한 서버는 고유한 기능을 가진 시스템, 신기술·신기능 테스트 목적의 시스템, 중요 데이터를 다루는 시스템 등이 대상이다. 예외 대상 서버에 대한 목록, 우선순위, 모니터링 및 보고 방안 등이 포함된 예외 정책을 별도로 수립해야 한다. 예외 정책 적용이 필요한 시스템을 체계적으로 목록화하여 수동 또는 중앙 관리 시스템을 통해 관리하며, 모니터링 시스템과 연계해 예외 정책 관련 항목이 실시간 반영되고, 정책 위반이나 이상 징후 발생 시 즉각적으로 대응할 수 있도록 해야 한다.

제로트러스트 기반의 시스템 정책은 단순히 정책을 만들고 시행하는 것으로 끝나지 않는다. 지속적으로 시스템 환경이 변화하기 때문에 정책도 끊임없이 평가·변경·교육·문서 업데이트 등의 과정을 거쳐 지속적으로 관리되어야 하며, 분석 시스템과 연계해 정책의 잠재적 위험, 리스크를 평가하고, 자동화된 정책 생성을 통해 변경된 정책이 시스템에 자동 배포되어 반영될 수 있도록 해야 한다. 정책 관리 체계의 성숙도는 이러한 지속적 관리와 자동화 체계 구축 여부에 따라 결정되며, 전체 보안 수준과 업무 효율성을 동시에 높일 수 있다.

제로트러스트 환경에서 네트워크 분할 전략은 단순한 물리적 경계를 넘어 조직의 다양한 업무 환경과 자산 특성에 맞게 세분화되야 한다. 접근 통제와 자동화된 정책 운영 결합을 통해 네트워크 내 리스크를 최소화하고, 실질적인 내부 확산 방지와 유연한 보안 환경을 동시에 실현시킬 수 있다.

7. 시스템 패치 관리

시스템 패치 관리 정책 시, 운영체제, 애플리케이션, 펌웨어 등 모든 시스템 구성요소에 대한 보안 패치의 관리 기준과 실행 절차를 명확히 정의해야 한다. 패치 관리 정책에는 패치 대상 목록 선정, 패치 배포 및 설치 절차, 패치 실패 시 백업 및 복구 방안, 패치 준수 여부에 대한 실시간 모니터링 등 패치 전 과정의 원칙과 절차가 포함되는 것은 물론, 중앙 관리 시스템이나 패치 관리 시스템(PMS) 등을 통해 일관되고 체계적으로 관리해야 한다. 외부 패치 서버와 연동해 최신 패치 정책을 즉시 반영하고, 모든 시스템이 최신 보안 상태를 유지할 수 있도록 관리 체계를 지속적으로 개선해야 한다.

패치 배포와 수행은 관리 정책에 따라 모든 서버 및 시스템에 정확하고 일관되게 적용되어야 한다. 패치 목록, 우선순위, 배포 및 설치 정보, 사전 기능 테스트 등 패치 배포의 전 단계가 자동화·표준화되어야 하며, 승인된 배포 도구(PMS 등)를 통해 실제 패치 파일이 배포되야 한다. 문제 발생 시에는 백업 및 복원 정책에 따라 신속하게 롤백할 수 있는 구조도 갖추어야 한다. 최신 패치 파일의 경우 샌드박스 등 별도 환경에서 사전 검증 후 실제 시스템에 반영되며, 만약 샌드박스에서 문제가 발생하면 PMS 정책에서 자동으로 예외 처리하도록 해야 한다. 이와 같은 절차를 통해 시스템 운영 안정성과 사용자 편의성을 동시에 확보할 수 있다.

시스템 패치 모니터링은 패치 정책에 따라 패치가 배포되는 전체 과정을 실시간으로 감시하고, 패치 설치 상태와 누락, 실패, 지연 등 다양한 이슈를 즉시 파악할 수 있도록 해야 한다. PMS, 통합 모니터링 시스템 등을 활용해 실시간으로 패치 현황을 가시화하고, 데이터 분석을 통해 패치율 추이, 설치 실패 원인, 재배포 및 자동 롤백 등 후속 조치까지 일괄적으로 관리할 수 있는 체계도 갖추어야 한다. 모니터링 결과는 보고서 등 다양한 형태로 자동 생성·배포하고, 문제 발생 시 즉각적인 알림과 분석을 통해 재배포 프로세스가 자동화될 수 있도록 운영해야 한다.

8. 시스템 로그 관리

시스템 로그 관리는 각 시스템별로 수집 대상 로그를 명확히 정의하고, 로그 수집 정책에 따라 실시간으로 수집·저장하는 체계를 갖추어야 한다. 로그 수집 시에는 기본 제공 도구, 로그 에이전트, 수집 스크립트 등 다양한 방법을 활용해 커스텀 로그, 감사 로그 등 필요한 모든 로그가 빠짐없이 수집될 수 있도록 한다. 여기에, 실시간 수집과 주기적·정기적 수집 대상을 분리하여 효율적 관리가 필요하다. 전사적으로 시스템 로그가 실시간으로 수집·통합될 수 있는 환경을 구축하고, 로그 압축·전송, 안전한 저장·보관 등 데이터 무결성과 보안도 반드시 고려해야 한다.

효율적인 로그 관리를 위해 로그 인덱싱 체계도 반드시 구축해야 한다. 로그 데이터의 실시간 인덱싱, 검색, 필터링, 페이징, 하이라이트, 시각화 등 다양한 기능을 지원할 수 있어야 하며, Splunk, Elasticsearch, Graylog 등 인덱싱 도구를 활용해 중요 로그에 인덱스 값을 추가·관리하고, 전체 로그 관리 체계를 지속적으로 개선해야 한다. 로그 수집 및 분석 시스템과 연동하여 실시간 대응 체계까지 구축하는 것이 바람직하다.

시스템 로그 분석은 수집된 데이터를 기반으로 실시간·상관·시각화 분석 등을 수행해 시스템 활동, 성능, 보안 상태에 대한 통찰력을 제공해야 한다. 통합 로그 시스템과 SIEM 등을 통해 다양한 로그를 상관 분석하고, 이를 기반으로 시스템 개선 활동을 추진해야 하며, 과거 로그 데이터 및 머신러닝을 활용해 오류, 보안 위협, 성능 저하 등을 예측하고 자동화된 대응 시스템까지 구현해야 한다.

전체 로그 관리 체계에서는 전반적 시스템 상태의 인사이트 확보를 위해 요약·세부·비교 등 다양한 유형의 리포트를 주기적으로 자동 생성·관리해야 하며, 리포트는 HTML, PDF, CSV 등 다양한 형식으로 스케줄링이 발송되어야 한다. 자동 분석 기능은 머신러닝 기반으로 적용해, 주요 이벤트와 위험 요인의 실시간 진단 및 즉각적 대응이 가능하도록 하는 등 리포트 관리 체계 역시 지속적으로 고도화해야 한다.

9. 시스템 취약점 관리

시스템 취약점 관리 정책은 시스템 보안 수준을 유지하고 규제 준수를 달성하기 위해 취약점 관리의 목표와 범위를 명확히 정의하고, 취약점 진단·조치·분석·보고 등 전체 관리 프로세스를 체계적으로 수립해야 한다. 취약점 관리 정책에는 취약점의 정의, 심각도 평가 기준, 취약점 진단 방법, 패치 배포 및 코드 수정, 회피 및 완화 방안, 취약점 관련 보고 및 관리 절차가 포함되어야 하며, 상시 최신화된 정책이 모든 시스템에 자동 적용될 수 있도록 SOC, TI 시스템 등과의 연동을 통해 최신 취약점 정보를 수집하고 정책에 자동 반영하는 구조를 마련해야 한다.

취약점 진단 및 조치는 자동·수동 스캐닝, CVE 등 공개 취약점 데이터베이스 활용 등 다양한 방법으로 시스템의 취약점을 신속하고 정확하게 탐지해야 하며, 진단된 취약점에 대해서는 정책에 따라 우선순위를 지정해 패치 배포, 코드 수정, 회피 등 신속한 조치가 이루어져야 한다. 시스템 취약점 관리 시스템을 통해 주기적 진단 및 최신 취약점 반영이 일관되게 이루어져야 하며, 위험도에 따라 실시간 진단·조치 및 패치 자동화가 연계되어야 한다.

취약점 영향도 평가는 발견된 취약점의 원인 분석과 함께 시스템에 미치는 영향을 다각도로 평가해 우선순위를 결정해야 한다. 실제로 악용될 가능성, 시스템 운영에 미치는 영향, 예방·완화 조치의 필요성 등이 객관적으로 판단돼야 한다. 평가 결과에 따라 패치, 회피, 수용 등 다양한 대응 방안을 마련하고, ICAM, TI 시스템 등과 연동해 최신 DB 기반의 실시간 분석과 정책 자동화를 구현해야 한다.

취약점 관리는 단순 탐지·조치에 그치지 않고, SIEM 시스템 등과 연동해 취약점 이벤트를 실시간으로 관리하고 분석할 수 있는 체계 마련이 필요하다. 취약점 관련 보고서, 대응 현황, 완화 계획 등 모든 관리 내역을 자동화 도구로 문서화하고, 머신러닝·빅데이터 기반의 심층 분석을 통해 대응 효과성과 효율성을 지속적으로 개선해야 한다. 취약점 관리 시스템과 백신, 모니터링 시스템 등과의 연계로, 위험 발생 시 자동화된 대응 프로세스가 작동하고, 보고서와 현황 정보가 주기적으로 생성·배포되어야 한다.

10. 시스템 가시성 및 분석

시스템 가시성 확보는 서버의 상태와 성능, 이상 활동, 보안 위협 등을 실시간으로 모니터링하고 분석하여 문제를 조기에 감지하고 신속히 대응할 수 있는 체계를 갖추는 것을 의미한다. 모든 시스템에 대해 모니터링 정책을 적용하고, CPU, Memory, 디스크 사용량 등 리소스 현황과 주요 프로세스 상태, 각종 시스템 메트릭 등 핵심 항목을 지속적으로 수집·분석해야 하며, 이상 탐지 및 알림 방안이 함께 마련되어야 한다. 모니터링 시스템은 시스템 단위뿐만 아니라 전체 인프라에 대한 가시성을 제공하고, 시스템 변동 사항이 발생할 때마다 자동적으로 모니터링 체계에 반영되며, 통합 로그시스템·분석 시스템 등과 연계하여 일관된 운영·관리가 이루어져야 한다.

시스템 분석 능력은 서버에서 수집되는 다양한 데이터의 심층 분석을 통해 시스템 환경에 대한 깊은 이해와 최적화를 실현하는 핵심 기능이다. 단순 데이터 수집에 그치지 않고, 수집된 로그와 데이터를 시각화 도구 등으로 실시간 확인하거나, 장기간 저장·분석하여 시스템 문제를 파악하고 개선책을 도출해야 한다. 데이터는 정제·변환·통합 과정을 거쳐 다양한 분석 기법을 활용해 검토되어야 하며, 시스템 모니터링 및 분석 프로세스를 기반으로 운영 및 보안 정책이 실시간으로 개선·적용되는 구조를 마련해야 한다. 이러한 체계는 정기적 리뷰와 프로세스 개선 활동을 통해 지속적으로 고도화되어야 한다.

11. 정책 및 프로세스

시스템 운영 절차는 조직의 IT 시스템 운영에 필수적인 요소로서, 효율적이고 안정적인 운영을 통해 비즈니스 연속성과 시스템 보안을 동시에 달성하는 데 중요한 역할을 한다. 서버 운영 절차는 시스템 설치, 구성, 모니터링, 유지관리, 문서 관리 등 주요 영역별로 명확하고 일관된 기준을 제공해야 하며, 이러한 운영 절차는 거버넌스 관점에서 수립·관리되어야 한다. 제로트러스트 관점에서는 최소 권한 원칙, 지속적인 인증 및 승인, 네트워크 위치 중심 보안 탈피, 데이터 중심 보안 강화, 보안 사고에 대한 신속한 대응 등 보안 강화를 위한 기준이 운영 절차 전반에 반영되어야 하며, 기존 경계 기반 모델에서 벗어나 실시간 피드백과 반영 체계까지 함께 정립하여야 한다.

시스템 최소 권한 관리 역시 핵심 원칙이다. 시스템 관리자 등 접근이 허용된 인원을 최소화하고, 비인가자의 접근을 원천적으로 통제하는 방안이 필수적이다. 사용자에게는 작업 수행에 필요한 최소한의 권한만을 부여하는 것을 원칙으로 하며, 이는 RBAC(역할 기반 접근제어), ABAC(속성 기반 접근제어) 등 다양한 방식으로 구현되어야 한다. 최소 권한 원칙에 따라 계정 관리 시스템, 인증 관리 시스템 등과 연동해 권한을 관리하고, 작업 수행 이후 권한 회수, 결재 승인 등 절차를 정립해 시스템 전반의 권한 오남용을 방지해야 한다. 특히, 주요 정보 시스템은 별도의 공간이나 장비를 활용해 접근을 추가로 통제하는 등 다중 방어체계를 구축해야 한다.

개인정보 시스템 관리 역시 매우 중요하다. 개인정보가 저장되는 시스템에 대해서는 관리적·기술적 개인정보 보호 정책을 수립하고, 관련 법규 및 컴플라이언스 기준에 부합하는 통제 기능을 적용해야 한다. 개인정보 보호 시스템, 관리 포털 등과 연동해 개인정보가 포함된 모든 시스템을 통합적으로 관리하고, 라이프사이클 전반에 걸쳐 개인정보의 흐름을 자동화·관리하는 체계를 마련해야 한다. 정책 수립 시에는 개인정보 라이프사이클 관리, 흐름 추적, 접근통제 등 각종 보호 조치를 포함해 개인정보 유출 및 오남용 위험을 최소화해야 한다.

위와 같은 주요 요소들을 기반으로, 시스템 필러는 제로트러스트 아키텍처 내에서 조직의 핵심 자산인 서버, 중요 응용 프로그램, 데이터 저장소 등 모든 시스템을 직접적으로 관리하고 보호하는 중심축으로 기능한다. 오늘날 복잡하고 분산된 IT 인프라에서 시스템은 가장 많은 데이터와 업무가 연관되어 있을 뿐 아니라, 보안 사고 발생 시 피해 규모가 크고 파급력도 높은 영역이기 때문에 외부·내부 위협 모두에 철저히 대비해야 하는 대상이다.

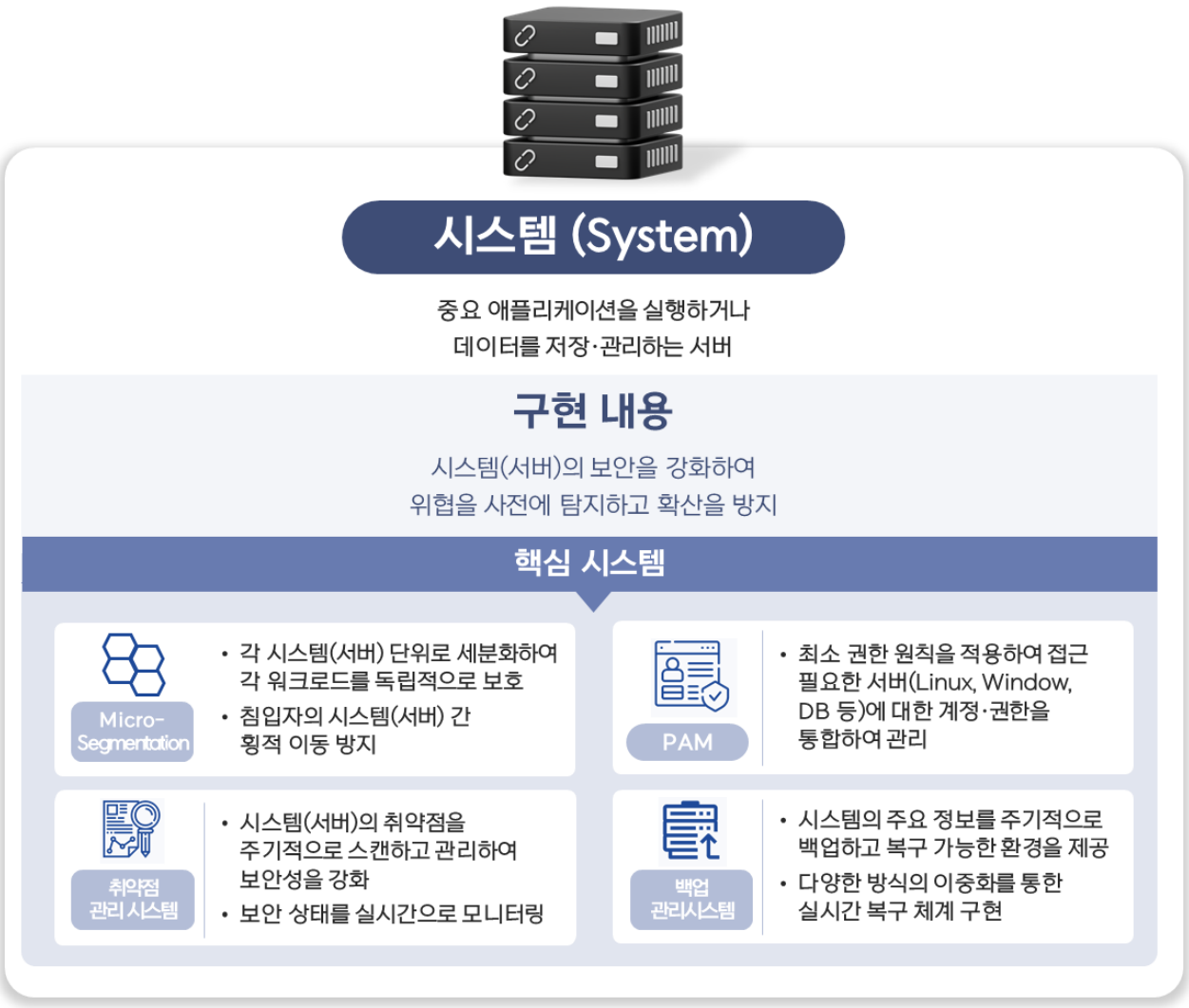
특히 제로트러스트 환경에서는 서버의 위치나 기존 신뢰 관계에 의존한 접근 통제가 더 이상 충분하지 않으며, 각 시스템은 개별적으로 강력한 인증 및 최소 권한 원칙, 실시간 보안 상태 모니터링, 세부적인 접근 권한 관리와 프로세스 통제 등 다양한 보안 메커니즘이 상호 유기적으로 연계되어야 한다. 또한 시스템 인벤토리, 계정 관리, 보안 정책, 취약점 및 패치 관리, 시스템 분리와 정책 자동화, 로그 수집 및 분석, 실시간 가시성 확보 등의 관리적·기술적 요소가 통합적으로 운영될 때 비로소 시스템 전반에 걸친 실질적인 제로트러스트 보안 수준을 달성할 수 있다.

이러한 시스템 필러의 고도화는 조직의 서버 인프라 전체에 제로트러스트 원칙을 일관되게 적용할 수 있는 관리 체계와 기술적 토대를 마련하여, 각 시스템 단위에서 발생할 수 있는 이상 징후를 조기에 탐지하고 신속하게 대응할 수 있는 환경을 실현하게 한다. 또한, 시스템 필러의 효과적 구현은 조직 내 중요한 데이터와 핵심 업무 프로세스를 안정적으로 보호하고, 변화하는 IT 환경과 갈수록 정교해지는 사이버 위협으로부터 조직의 핵심 인프라를 효과적으로 방어하는 데 필수적인 역할을 수행한다.

■ 주요 시스템별 제로트러스트 기능 구현

제로트러스트 환경을 성공적으로 구현하기 위해서는 기술적 방안과 이를 수행할 수 있는 시스템은 필수적이다. 제로트러스트 아키텍처는 "신뢰하지 않고 항상 검증한다"는 원칙을 기반으로 하며, 이를 실현하기 위해 각 시스템 별 상태를 확인하고, 지속적으로 검증하며, 최소 권한 접근을 보장을 수행할 수 있는 시스템은 필수적이다.

아래 주요 시스템 등은 각각 제로트러스트 환경에서 중요한 역할을 담당하며, 이들 시스템은 상호 연계되어 조직의 보안 태세를 강화할 수 있다. 각 시스템 별로 제로트러스트 환경 구현을 위해 수행해야 할 기능과 이를 통해 조직이 얻을 수 있는 보안 강화 효과를 구체적으로 살펴보고자 한다.



출처 : SK 쉴더스, "제로트러스트의 시작:SKZT 로 완성하다"

그림 2. 시스템 필러 주요 시스템

1. PAM (Privileged Access Management, 특권 권한 관리)

시스템 필러에서 PAM 은 가장 중심에 놓이는 보안 시스템이다. 식별자 필러의 IAM(IDP)와 연계되어 제로트러스트 환경에서 "누가, 무엇으로, 어디에, 언제, 얼마나" 접근할 수 있는지를 최종적으로 수행하고 감사하는 역할을 한다. 오늘날의 PAM은 전통적인 서버·DB 중심을 넘어 사내 업무시스템, 네트워크·보안장비, 클라우드 콘솔(AWS/Azure/GCP), 그리고 각종 SaaS 까지 특권 접근 전반을 한 화면에서 통제하도록 설계된다. 즉, 시스템 접근제어와 DB 접근제어를 모두 포괄하되, 동일한 원칙과 절차로 원격접속, 클라우드 관리 콘솔, 웹·API 기반 관리 화면까지 일관되게 다룬다.

이전의 PAM 은 일반적으로 시스템접근제어, DB 접근제어로 사용되고 있고 에이전트를 설치해 세션을 통제·기록하는 방식이었다. 최근에는 관리 범위가 애플리케이션·SaaS·클라우드 인프라로 확대되면서, 브라우저를 통해 접속을 중계하는 웹 기반 구조가 빠르게 자리 잡고 있다. 배스천(중계 게이트웨이)가 모든 세션을 매개하고, 사용자는 로컬 키나 계정 없이도 웹 콘솔에서 필요한 자원으로 "최소 권한" 기준으로 접근한다. 이 구조는 서버·DB 외에도 에이전트 설치가 어려운 자산이나 외부 관리형 서비스까지 같은 사용자 경험으로 통제할 수 있어 실무 환경에서 선호된다.

제로트러스트의 핵심인 "지속적 검증"은 PAM 에서 특권 세션 실시간 재평가로 구현된다. 세션이 시작된 뒤에도 사용자·디바이스 상태, 접속 위치·시간, 수행 명령, 쿼리 패턴 같은 신호를 계속 살펴 위험이 감지되면 MFA 추가 인증을 요구하거나, 권한을 단계적으로 축소하고, 필요 시 세션을 자동 종료한다. 이렇게 접근 허가 이후에도 끊임없이 확인하는 흐름이 서버·DB·SaaS 를 가리지 않고 동일하게 적용된다.

보안 채널과 비밀정보 관리 또한 PAM 의 기본 기능이다. SSH 키와 특권 비밀번호는 시크릿 금고(Secret Vault)에 보관·회전되며, 접속은 키 노출 없이 프록시·터널(SSH/SSL/TLS)로 중개된다. 콘솔 접근, 명령어, 파일 업·다운로드, 쿼리, 데이터 추출 이력 등 모든 행위는 상세 메타데이터와 함께 저장되어, 사고 시점 재현과 규제 감사에 그대로 활용된다.

종합하면, PAM 은 시스템 필러에서 특권 접근의 단일 관문으로 작동한다. IAM(IDP)이 "누구인가"를 보증하면 PAM 은 "무엇을, 어디까지, 어떤 조건에서" 허용할지를 결정·집행하고, 그 결과를 기록하고 감사하는 시스템으로 동작한다. 제로트러스트 환경에서 PAM 시스템은 서버와 DB 는 에이전트, 프록시 방식으로, 애플리케이션·SaaS·클라우드는 웹 기반 중계로, 클라우드 환경과 온프레미스를 아우르는 하나의 통합 시스템으로 동작할 수 있다.

출처 : CyberArk, "CyberArk Next Generation PAM"

그림 3. CyberArk PAM Architecture

2. Micro-Segmentation (마이크로 세그멘테이션)

Micro-Segmentation 은 기존 Macro-Segmentation 보다 한층 더 세분화된 보안 전략으로, 네트워크를 OSI 7 계층(Application 계층) 수준에서 업무, 사용자, 애플리케이션 단위까지 세밀하게 분리하고 최소 권한 원칙에 따라 접근을 제어하는 고도화된 접근 방식이다.

기존의 네트워크 세분화가 주로 IP 주소, 포트 기반, VLAN 등 물리적·논리적 경계에 머물렀다면, Micro-Segmentation 은 서비스·애플리케이션 간의 관계, 목적, 실질적 트래픽 흐름을 중심으로 논리적으로 네트워크를 분할한다. 이를 통해 조직은 네트워크 내부에서 발생할 수 있는 위협이나 공격자의 횡적 이동(Lateral Movement)까지 정밀하게 통제할 수 있다.

Micro-Segmentation 의 구현 방식은 크게 네트워크 기반의 방식과 시스템(서버) 중심의 방식 두 가지로 나뉘는데 시스템 필러에서의 Micro-Segmentation 은 시스템(호스트) 기반의 Micro-Segmentation 으로 봐야 한다.

시스템 기반 Micro-Segmentation 의 시스템은, 엔드포인트(서버, 워크스테이션 등) 단에 에이전트(Agent) 또는 에이전트리스(Agentless) 방식의 전문 솔루션을 설치해, 각 시스템(서버) 별로 세분화된 보안 정책과 접근 제어를 적용하는 방식이다. 이 과정에서 시스템(서버)와 네트워크간 연결 구조(토폴로지)를 시각화 하고, 각 애플리케이션과 서비스 간의 실제 네트워크 트래픽 흐름을 분석해 자동으로 세분화 정책을 생성·관리한다. 최근에는 AI, 머신러닝 기술이 적용되어 시스템간의 이동경로를 최적화하고 최적화하고, 이상징후 탐지, 정책 추천 등 운영 효율성을 높이고 있다.

시스템 필러의 Micro-Segmentation 구현의 핵심은 “구간 방화벽”에서 “시스템 별 “개별 방화벽”으로 보안 단위를 바꾸는 것이다. 과거에는 중요 구간별로 방화벽을 세워 큰 흐름을 막았다면, 이제는 서버마다 개별 방화벽을 두듯 정책을 적용해 내부 침해가 발생해도 옆 서버로 번지는 횡적 이동을 차단해야 한다. 이렇게 경계를 세밀하게 나누면 보안성은 크게 높아지지만, 정책 수가 늘어 관리가 복잡해질 수 있다. 이 부분은 최신 기술의 AI·머신러닝을 활용해 해결해야 한다. 정상 트래픽 패턴 학습, 정책 추천, 중복·불필요 규칙 정리, 정책 변경 전 시뮬레이션, 이상 징후 자동 알림 등을 통해 운영 부담을 줄이고, 정책 품질을 안정적으로 유지해야 한다. 실제 활용사례 및 적용사례를 살펴보면 Micro-Segmentation 구현에서 전용시스템에서 제공하는 AI 은 유의미하게 동작하고 있다.

출처 : Akamai, “Guardicore Segmentation 구현 방법”

그림 4. Micro-Segmentation 적용 절차

3. 시스템(서버) 취약점 관리시스템 (Vulnerability Management System)

시스템(서버) 취약점 관리시스템은 조직 내 서버, 네트워크 장비, 클라우드 인스턴스 전반에서 발견될 수 있는 보안 약점을 지속적으로 탐지·평가·조치해 리스크를 낮추는 핵심 시스템이다. 운영체제, 미들웨어, 애플리케이션, DB, 웹/서비스 프로세스까지 주기적이거나 상시로 스캔하고, 결과를 위험도 기준으로 정렬해 관리할 수 있다. 취약점 조치 진행률, 패치 적용 현황, 미해결 항목과 재발 건수는 대시보드와 리포트로 관리된다.

실무에서는 에이전트 기반(서버 내 설치)과 에이전트리스(원격 인증 스캔) 방식을 함께 쓴다. 인증 스캔을 통해 단순 버전 비교를 넘어 설정 오류, 불필요 서비스, 불필요한 권한, 취약한 암호화 같은 구성 취약점까지 점검한다. 클라우드에서는 일시적으로 생성·종료되는 인스턴스와 오토스케일링 그룹을 놓치지 않도록 태그/라벨과 연동해 자동 등록·스캔하고, 골든 이미지(AMI/템플릿) 자체를 주기적으로 검사한다. 컨테이너 환경은 호스트 OS 와 컨테이너 이미지를 분리해 진단하고, CI/CD 파이프라인 단계에서 이미지 스캔을 시행해 운영 반영 전에 위험을 걸러낸다.

취약점의 우선순위는 단순 CVSS 점수만으로 정하지 않는다. 악용 중(KEV), 악용 가능성(EPSS), 인터넷 노출 여부, 업무 중요도, 데이터 민감도, 확산 가능성 등을 더해 리스크 점수를 계산한다. 이렇게 정렬된 결과를 바탕으로 SOAR 와 연계되어 캠페인 형 패치 작업을 계획하고, 변경 창·롤백 계획·사전/사후 기능 점검을 포함한 표준 조치 플레이북으로 실행한다. 패치가 어려운 경우에는 예외(기간·사유 명시)를 등록하되, 보완 통제(방화벽 차단, WAF 가상패치, 계정/권한 축소, 서비스 격리, 파일 무결성 모니터링)를 자동 적용해 잔여 위험을 낮춘다.

제로트러스트 관점에서 취약점 관리 시스템은 "서버의 신뢰 수준"을 수치화해 다른 보안 장비와 정책 연계를 만든다. 예를 들어, 고위험 취약점이 미조치 상태면 ZTNA(NGFW), PAM 에서 서버 접근을 제한하고, EDR 은 해당 서버를 격리하며, IAM/SSO 는 관련 관리 계정의 세션에 MFA 를 강제하도록 동작하게 할 수도 있다. 이렇게 구축된 취약점 관리 체계는 시스템 필러에서 실시간 취약점 기반 동적 체계로, 제로트러스트 환경을 구현하도록 할 수 있다.

4. 백업관리시스템 (Backup & Recovery Management System)

백업관리시스템은 시스템(서버)에 장애나 침해가 발생했을 때 서비스를 다시 정상화할 수 있도록 백업을 만들어 보관하고 검증해서 실제로 복구까지 수행하는 실행 시스템이다. 파일 몇 개를 따로 저장하는 수준이 아니라 운영체제·애플리케이션·설정·DB 까지 포함한 서버 이미지를 정기적으로 보호하고, 필요하면 바로 마운트해 서비스를 실행시키거나 특정 파일·메일·DB 오브젝트만 골라 되돌릴 수 있게 한다. 제로트러스트 환경에서 백업관리시스템이 이러한 역할을 맡아 온프레미스–가상화–클라우드–SaaS 전반을 한 시스템에서 관리한다.

백업 대상을 자동으로 찾아 등록하고(에이전트/에이전트리스 병행), 애플리케이션 정합성을 보장한 스냅샷을 만든다. 전체를 매번 복사하지 않고 변경된 블록만 전송해 저장소와 네트워크 부담을 줄이며, 중복 제거와 압축으로 보관 효율을 높인다. 백업본은 로컬 저장소와 원격 오브젝트 스토리지 등 여러 위치로 분산하고, 필요(중요) 시스템은 웜스토리지로 보관해 삭제나 변조에 대비한다. 백업이 실제로 복구되는지도 자동 부팅·애플리케이션 점검 절차로 주기적으로 검증하고, 결과는 대시보드와 리포트로 남겨야 한다.

클라우드와 컨테이너 환경에서도 동일한 원칙을 적용해야 한다. 클라우드에서는 태그·라벨과 정책을 연동해 새로 생성된 인스턴스를 자동으로 보호 대상에 포함하고, 스냅샷 API 를 통해 디스크 단위 백업을 오케스트레이션한다. 쿠버네티스는 etcd 와 리소스 매니페스트, 퍼시스턴트 볼륨을 함께 보존해 네임스페이스 단위로 원상 복구할 수 있게 하고, CI/CD 파이프라인과 연결해 배포 전후 스냅샷을 남겨 빠른 롤백을 지원한다. Microsoft 365, Google Workspace, Salesforce 같은 SaaS 데이터도 같은 정책 체계 안에서 보호·복구한다.

백업관리시스템을 고도화하기 위해서는 조직의 재해복구(DR) 전략을 반영하여 구축해야 한다. 개념적으로 접근하면 콜드 사이트처럼 비용을 낮추되 복구 시간이 긴 방식은 백업본과 설정(인프라 코드 포함)을 기반으로 필요 시점에 환경을 띄워 복구할 수 있다. 웜 사이트는 주기적 복제와 스냅샷으로 핵심 서비스만 미리 준비해 중간 수준의 RTO/RPO 를 달성한다. 핫 사이트는 동기 또는 저지연 복제와 자동 페일오버를 사용해 복구 시간을 극단적으로 줄이는 대신 비용이 높다. 백업관리시스템은 이런 다양한 백업 시나리오를 런북(플레이북)으로 자동화하고(순서·의존성·검증 포함), 업무 시간 중에도 격리 영역에서 무중단 DR 리허설을 돌린 뒤, 실제 사고 시 페일오버와 페일보백까지 절차대로 실행할 수 있어야 한다.

제로트러스트 관점의 통제도 필요하다. 백업 콘솔과 삭제·영구폐기 같은 고위험 기능은 SSO/IAM·PAM 과 연동해 MFA 와 승인 절차를 거치게 하고, 백업 전용 네트워크 구간은 ZTNA/NGFW 나 마이크로 세그멘테이션으로 운영망과 분리한다. 백업 시점에는 악성 파일·비정상 패턴 검사로 의심 백업본을 격리 보관하며, 탐지 결과는 SIEM/SOAR 로 전달돼 알림과 후속 조치가 자동화할 수 있다. 실제 복구는 우선 격리된 검증 환경에서 먼저 부팅·검증을 거친 후 운영 인프라에 적용해야 한다.

백업관리시스템은 BCP(비즈니스 연속성)을 보장하고 조직의 정책/지침/절차서를 반영하여 시스템 필러의 가용성과 신뢰성을 보장할 수 있다.

시스템 필러는 제로트러스트 아키텍처에서 주요 리소스가 저장되는 서버 영역을 다룬다. PAM 을 중심으로 Micro-Segmentation, 취약점 관리시스템, 백업관리시스템 등을 활용하여 특권 접근을 중앙에서 통제하고, 서버 간 통신을 세밀하게 분리·제한하며, 취약점을 지속 평가·조치하고, 장애·침해 시 신속 복구까지 하나의 흐름으로 연결한다. 시스템 필러의 주요 시스템들은 IAM, ZTNA, SIEM&SOAR 등 타 필러의 주요 시스템들과 연동되어 온프레미스와 클라우드를 아우르는 제로트러스트 환경에서 신뢰도와 가용성을 지속적으로 유지하고 강화한다.

■ 맺음말

제로트러스트 아키텍처에서 시스템 필러는 해외 기준에서는 존재하지 않는 필러다. 해외 가이드라인이나 기준에서는 시스템(서버) 영역은 디바이스와 엔드포인트 필러에 포함되어 다뤄진다. 국내는 망분리 환경과 온프레미스 중심의 운영으로 시스템(서버)에 대한 영역이 중요하게 여겨지고 운영되고 있기 때문에, KISA 에서 국내 환경에 맞는 제로트러스트 가이드라인을 발간할 때 시스템 필러를 추가·분류하여 포함시켰다.

제로트러스트 환경에서 시스템 필러가 중점적으로 봐야할 점은 온프레미스, 퍼블릭 클라우드, 프라이빗 클라우드 등 다양한 환경에서 사용되는 시스템(서버)들을 통합해 관리하고, 일괄된 보안정책을 부여하는 것이다.

시스템 필러를 일괄적으로 통제하기 위해서는 적절한 정책과 실현시킬 수 있는 시스템이 병행되어야 한다. 시스템 필러의 주요 요소인 시스템 인벤토리, 계정관리, 접근통제, 정책관리, 패치관리 등을 기준으로 관리 기준과 정책을 정의하고, 이를 구현할 수 있는 시스템인 PAM(특권 권한관리), Micro-Segmentation, 취약점 관리, 백업관리 시스템 등을 활용하여 구현해야 한다.

시스템(서버)는 새로 구성되어 사용되는 시스템보다 기존에 구축되고 운영 되는 서버가 훨씬 많기 때문에 제로트러스트 아키텍처 구현을 고려할 때에는 기존 서버들의 호환성 문제가 가장 어려운 부분이기도 하다. OS 와 미들웨어의 종류가 다양한 만큼 시스템(서버) 필러의 제로트러스트 아키텍처를 적용할 때에는 에이전트를 직접 설치하거나 호환되지 않는 시스템은 에이전트리스방식을 혼용하여 고려하고, 지원되지 않는 시스템은 별도의 통제 정책을 만들어서 모니터링하고 관리해야 한다.

결론적으로, 시스템 필러는 국내 환경을 고려해 별도 필러로 분류되었고, 조직의 중요한 리소스들이 저장되어 있는 시스템(서버) 영역을 제로트러스트 기반으로 관리하는 것을 목적으로 둔다. 시스템 필러는 독자적으로 보안 아키텍처를 구현하는 것이 아닌 식별자, 네트워크, 데이터 등의 필러와 유기적으로 연동됨으로써 조직의 환경에 제로트러스트 아키텍처를 구현할 수 있다.

■ 참고 문헌

[1] KISA, “제로트러스트가이드라인 V2.0”, 2024.12

[2] NIST SP 800-207, “Zero Trust Architecture”, 2020.08

[2] NIST SP 1800-35 Final,“Implementing a Zero Trust Architecture:High-Level Document”, 2025.06

[3] NIST SP 800-34, “Contingency Planning Guide for Federal Information Systems”, 2010.11

[4] DoD, “Zero Trust Overlays”, 2024.06

[5] 국가사이버안보센터, “국가 망 보안체계 보안 가이드라인(Draft)”, 2025.01

■ 참고 자료

[1] SK쉴더스, “제로트러스트의 시작:SKZT로 완성하다” – 브로슈어

[2] Gartner, “Best Privileged Access Management Reviews 2025”

[3] Akamai, “What Is Microsegmentation or Micro-Segmentation?”

[4] CyberArk, “What is Privileged Access Management (PAM)? - Definition”

[5] 넷앤드, “HIWARE Privileged Session Management for System”

[6] Arctera, “Arctera™ System Recovery 24 User's Guide”

EQST INSIGHT

2025.08

SK 쉴더스

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

발행인 SK쉴더스 EQST사업그룹
제 작 SK쉴더스 마케팅그룹