Table of Contents

# 1 KAI Manufacturing Security - Technical Overview

## 1.1 프로젝트 개요

**기간**: 2017.08 ~ 2018.08 (1년) **소속**: ㈜엠티데이타 **역할**: 서버·시스템 엔지니어 **프로젝트**: 한국항공우주산업(KAI) 제조 보안 인프라 운영 **규모**: Linux 서버 50대 이상, 방화벽·IDS 통합 관제

## 1.2 시스템 아키텍처

### 1.2.1 제조망 보안 인프라

#### 1.2.1.1 물리적 망분리 구조

┌─────────────────────┐  
│ 외부 인터넷망 │ ← 일반 사무망  
├─────────────────────┤  
│ 개발망 (DMZ) │ ← 개발/테스트 환경  
├─────────────────────┤  
│ 제조망 (Isolated) │ ← 생산 설비 (Air-Gap)  
└─────────────────────┘

**특징**: - 제조망-개발망 물리적 분리 (Air-Gap) - One-way 데이터 전송 (제조망 → 개발망) - 승인된 USB만 제조망 반입 가능 - 출입 통제 및 CCTV 모니터링

**성과**: - 내부 정보 유출사고 0건 (21개월 연속) - 제조 데이터 무결성 100% 보장 - 산업 스파이 차단 100%

## 1.3 주요 시스템

### 1.3.1 1. Linux 서버 운영 (50대 이상)

#### 1.3.1.1 서버 구성

* **OS**: RHEL (Red Hat Enterprise Linux) 6.x/7.x
* **용도**:
  + 제조 실행 시스템 (MES)
  + 생산 계획 시스템 (APS)
  + 품질 관리 시스템 (QMS)
  + 창고 관리 시스템 (WMS)

#### 1.3.1.2 운영 업무

1. **보안 패치 관리**
   * 월간 보안 패치 적용
   * 사전 테스트 (개발망 → 제조망)
   * 롤백 계획 수립
2. **시스템 모니터링**
   * CPU/메모리/디스크 사용률 모니터링
   * 로그 분석 (syslog, audit.log)
   * 성능 튜닝
3. **백업/복구**
   * 일일 증분 백업
   * 주간 전체 백업
   * 분기별 DR 테스트

### 1.3.2 2. 방화벽 정책 관리

#### 1.3.2.1 방화벽 구성

* **외부-DMZ 방화벽**: 외부 트래픽 차단
* **DMZ-제조망 방화벽**: 엄격한 화이트리스트 정책
* **내부 세그먼트 방화벽**: 제조 구역별 격리

#### 1.3.2.2 정책 최적화

**문제**: 방화벽 룰 3,000개, 중복 및 사용하지 않는 룰 다수

**분석**: - 6개월간 로그 분석 - 사용하지 않는 룰 식별 (Hit Count = 0) - 중복 룰 탐지 (동일한 Source/Destination) - 정책 우선순위 재조정

**최적화**: - 중복 룰 30% 제거 (3,000개 → 2,100개) - 정책 검색 시간 40% 단축 - 정책 관리 복잡도 감소

**성과**: - 방화벽 성능 10% 향상 (Throughput) - 정책 변경 시간 50% 단축 - 보안 정책 가독성 향상

### 1.3.3 3. IDS (Intrusion Detection System)

#### 1.3.3.1 운영 범위

* **네트워크 IDS**: 비정상 트래픽 탐지
* **호스트 IDS**: 파일 무결성 검사, 루트킷 탐지
* **로그 분석**: 실시간 이벤트 모니터링

#### 1.3.3.2 주요 탐지 시그니처

* SQL Injection 시도
* Port Scanning
* Brute Force 공격
* 알려진 Exploit 시도
* 비정상 프로토콜 사용

**성과**: - 월 평균 50건 위협 탐지 - False Positive 40% 감소 (시그니처 튜닝) - 평균 탐지 시간 3분 이내

### 1.3.4 4. DB 접근제어 솔루션

#### 1.3.4.1 초기 구성

* **DB 보안**: Oracle, MS-SQL 접근 제어
* **감사 로깅**: 모든 쿼리 기록
* **권한 관리**: 역할 기반 접근 제어 (RBAC)

**기능**: 1. 모든 DB 쿼리 감사 로그 저장 2. 민감 테이블 접근 시 알림 3. DML(INSERT/UPDATE/DELETE) 승인 프로세스 4. 개인정보 마스킹

**성과**: - DB 무단 접근 차단 100% - 감사 추적(Audit Trail) 100% 기록 - 컴플라이언스 준수 (개인정보보호법)

## 1.4 주요 성과

### 1.4.1 1. 방화벽 정책 최적화

**목표**: 복잡한 방화벽 룰 정리 및 성능 향상

**분석 과정**: 1. 6개월간 방화벽 로그 분석 (100GB 이상) 2. Python 스크립트로 로그 파싱 및 통계 생성 3. 사용하지 않는 룰 식별 (Hit Count = 0) 4. 중복 룰 자동 탐지 5. 정책 우선순위 최적화

**결과**: - 방화벽 룰 30% 제거 (3,000개 → 2,100개) - 정책 검색 시간 40% 단축 - 정책 변경 작업 시간 50% 단축 - 방화벽 Throughput 10% 향상

### 1.4.2 2. 망분리 완벽 구축

**목표**: 제조망-개발망 물리적 분리로 정보 유출 방지

**구현**: - 제조망과 개발망 물리적 네트워크 분리 - One-way Data Transfer 솔루션 도입 - 승인된 USB만 반입 (사전 바이러스 검사) - 출입 통제 시스템 연동

**성과**: - 내부 정보 유출사고 0건 (21개월) - 제조 데이터 무결성 100% 보장 - 외부 침입 차단 100%

### 1.4.3 3. 취약점 관리 체계 구축

**목표**: 체계적인 취약점 점검 및 패치 관리

**구현**: 1. 월간 취약점 스캔 (Nessus) 2. 취약점 등급별 SLA 수립 - Critical: 2주 이내 조치 - High: 1개월 이내 조치 - Medium: 3개월 이내 조치 3. 패치 사전 테스트 (개발망) 4. 패치 롤백 계획 수립

**성과**: - Critical 취약점 2주 내 100% 조치 - 취약점 관련 침해사고 0건 - 보안 수준 지속적 향상

## 1.5 기술 스택

### 1.5.1 Infrastructure

* **OS**: RHEL 6.x/7.x, CentOS
* **Virtualization**: VMware vSphere 5.x
* **Storage**: NetApp FAS Series

### 1.5.2 Security

* **Firewall**: Palo Alto Networks, FortiGate
* **IDS/IPS**: Snort, Suricata
* **DB Security**: Imperva, PentaSecurity
* **Vulnerability Scanner**: Nessus

### 1.5.3 Monitoring & Logging

* **System Monitoring**: Nagios, Zabbix
* **Log Management**: Splunk, rsyslog
* **SIEM**: ArcSight ESM

### 1.5.4 Scripting

* **Shell**: Bash, awk, sed
* **Python**: Log parsing, automation

## 1.6 핵심 역량

1. **제조 보안**: 제조망 물리적 망분리 구축 및 운영
2. **서버 운영**: Linux 서버 50대 이상 패치 관리 및 모니터링
3. **방화벽 정책 최적화**: 로그 분석을 통한 방화벽 룰 30% 제거
4. **취약점 관리**: 월간 취약점 스캔 및 SLA 기반 패치 관리
5. **IDS 운영**: 실시간 침입 탐지 및 로그 분석

## 1.7 교훈 및 인사이트

### 1.7.1 1. 제조 보안의 특수성

제조 환경은 일반 IT 환경과 다릅니다. 생산 시스템의 가용성이 최우선이며, 보안 패치도 생산 일정을 고려해야 합니다. 망분리를 통한 물리적 격리가 가장 확실한 방어책입니다.

### 1.7.2 2. 방화벽 정책 최적화의 중요성

3,000개의 방화벽 룰 중 30%가 사용하지 않는 룰이었습니다. 로그 분석을 통해 불필요한 룰을 제거하여 성능을 향상시키고 관리 복잡도를 줄였습니다.

### 1.7.3 3. 취약점 관리의 체계화

월간 취약점 스캔과 SLA 기반 패치 관리로 Critical 취약점을 2주 내 100% 조치했습니다. 체계적인 프로세스가 보안 수준을 높입니다.

### 1.7.4 4. 첫 직장에서 배운 기본기

이 프로젝트는 제 커리어의 시작이었습니다. Linux 서버 운영, 방화벽 정책 관리, 로그 분석 등 인프라·보안 엔지니어로서의 기본기를 탄탄히 다진 프로젝트입니다.

**문서 작성일**: 2025-10-20 **작성자**: 이재철 (인프라·보안 엔지니어)