

사용자 행동 기반
동적 네트워크 접근 제어 연구



저자1 이광훈 201924524

저자2 조유진 202155609

저자3 신해진 202255562

지도교수: 최윤희 교수

목 차

1. 서론	1
1.1. 연구 배경	1
1.2. 기존 문제점	2
1.3. 연구 목표	3
2. 연구 배경	4
2.1. 개발 환경	4
2.1.1. 적용 기술 및 환경	4
2.1.2. 기술 적용 목적	5
2.1.2.1. Virtual box	5
2.1.2.2. Fluentd	5
2.1.2.3. auditd	5
2.1.2.4. mitmproxy	6
2.1.2.5. FastAPI	6
2.1.2.6. Docker	6
2.1.2.7. OpenWRT	6
2.1.2.8. PostgreSQL	6
2.1.2.9. NodeJS/React	7
2.2. 데이터베이스 설계	7
2.3. API 설계	9
2.4. 머신 러닝 알고리즘	11
2.4.1. 결정 트리	11
2.4.2. SMOTE Oversampling	11
2.4.3. RandomForest	12
2.4.4. XGBoost	12
3. 연구 내용	13
3.1. 전체 시스템 설계	13
3.2. 로그 수집 및 저장 모듈	14
3.2.1. 구성요소	14
3.2.2. 사용자 행동 로그 수집 범위 및 방법	15

3.2.2.1. 로그인·로그오프	15
3.2.2.2. 웹 브라우징	15
3.2.2.3. 메일	15
3.2.2.4. 디바이스	15
3.2.2.5. 파일 복사	15
3.2.3. 표준 로그 스키마	15
3.3. 이상 탐지 모듈	16
3.3.1. 이상 탐지 모델 학습	16
3.3.1.1. 데이터 세트	16
3.3.1.2. 주 단위 집계 및 특징 추출	17
3.3.1.3. 모델 학습	18
3.3.2. 이상 탐지 파이프라인 구현	19
3.4. 네트워크 제어 모듈	20
3.5. 모니터링 모듈	21
3.5.1. 조직 인증 및 관리자 회원가입	21
3.5.2. 로그인	21
3.5.3. 대시보드	22
3.5.4. 이상 사용자 탐지 결과 조회	23
3.5.4.1. 이상 사용자 탐지	23
3.5.4.2. 탐지 이력 조회	23
3.5.5. 행동 로그 조회	24
3.5.6. PC 상태 조회 및 관리	25
3.5.7. 이상 사용자 로그인 시 알림	26
4. 연구 결과 분석 및 평가	27
4.1. 로그 수집 에이전트 성능 평가	27
4.1.1. 평상시	27
4.1.2. 일반/고강도 HTTP 트래픽	28
4.1.3. 메일 송신	28
4.1.4. 웹메일 송신	29
4.1.5. USB 장치 연결 및 해제	29

4.1.6. 파일 복사	29
4.1.7. 종합	29
4.2. 이상 탐지 모듈 평가	30
4.2.1. Random Forest 모델 학습 및 성능 평가	30
4.2.2. XGBoost 모델 학습 및 성능 평가	31
4.2.3. Random Forest 와 XGBoost 모델 성능 비교	32
4.2.4. 이상 탐지 파이프라인 평가	33
4.3. 네트워크 제어 모듈 평가	33
4.4. 모니터링 모듈 평가	33
5. 결론 및 향후 연구 방향	34
6. 구성원별 역할	35
7. 개발 일정	36
8. 멘토 의견 반영	37
8.1. OpenWRT 기반 가상 라우터를 통한 실제 네트워크 제어의 현실적 적용 가능성에 대한 검증 필요	37
8.2. 전체 시스템 통합과 성능 평가에 대한 구체적인 결과 제시	37
8.3. 실제 환경에서의 탐지 정확도와 오탐율에 대한 정량적 평가 필요	37
9. 참고 문헌	38

1. 서론

1.1. 연구 배경

팬데믹 이후 기업의 디지털 전환 및 원격 근무의 확대, 클라우드 환경의 도입이 가속화되고 있다. 이에 따라 기업 네트워크의 크기 및 복잡도도 함께 증가하고 있으며, 기업 네트워크에 연결된 엔드포인트 (Endpoint)¹ 수 역시 증가하는 추세다.

이러한 추세는 내부자 위협 (Insider Threat)의 증가로 이어진다[1]. 내부자 위협이란 기업 내부의 시스템 정보와 접근 권한을 가진 부주의하거나 악의적 목적을 가진 직원이 초래하는 보안 위협을 말한다. 내부자 위협으로 인한 보안 사고는 외부자에 의한 공격보다 더 탐지하기 어렵고, 피해 복구에 더 많은 비용이 소요되는 특징을 가진다[2]. 따라서 이러한 유형의 보안 위협에 대처하기 위한 제로 트러스트 모델(Zero Trust Model)의 중요성이 강조되고 있다.

제로 트러스트는 “절대 신뢰하지 말고, 항상 검증하라(Never Trust, Always Verify)”는 원칙을 기반으로 하는 보안 모델이다. 한국 인터넷 진흥원에서 2024년 공지한 제로 트러스트 가이드 라인에서는 제로 트러스트 아키텍처의 기본 원리를 다음과 같이 제시하고 있다[3].

- 모든 종류의 접근에 대해 신뢰하지 않을 것
- 일관되고 중앙 집중적인 정책 관리 및 접근제어 결정, 실행 필요
- 모든 상태에 대한 모니터링, 로그 및 이를 통한 신뢰성 지속적 검증, 제어

이러한 원칙을 바탕으로, 본 연구에서는 사용자의 행동 로그를 머신 러닝 알고리즘 기반으로 분석하고, 이상 사용자를 식별하는 시스템을 구현한다. 기업 내 사용자의 PC에 에이전트를 설치해 각 PC의 상태 및 사용 기록을 지속적으로 수집하고, 머신 러닝 알고리즘을 사용해 사용자 행동을 분석 및 이상 행위를 탐지한다. 또한, 이상 사용자가 로그인한 PC를 원격으로 네트워크에서 격리하는 시스템을 구축한다.

¹ 모바일 디바이스, 데스크톱 컴퓨터, 임베디드 디바이스, 서버 등 네트워크 시스템에 연결하는 물리적 디바이스

1.2. 기존 문제점

네트워크 접근 제어 (NAC, Network Access Control)에 대해 NIST (미국 국립 표준기술연구소, National Institute of Standards and Technology)는 “방화벽에서 제공하는 기능으로, 사용자의 자격 증명과 원격 근무 클라이언트 장치에 수행된 상태 점검 결과를 기반으로 접근을 허용하는 기능”으로 정의한다[4]. NAC의 대표적인 표준 기술로는 802.1x가 있다.

The components of 802.1X

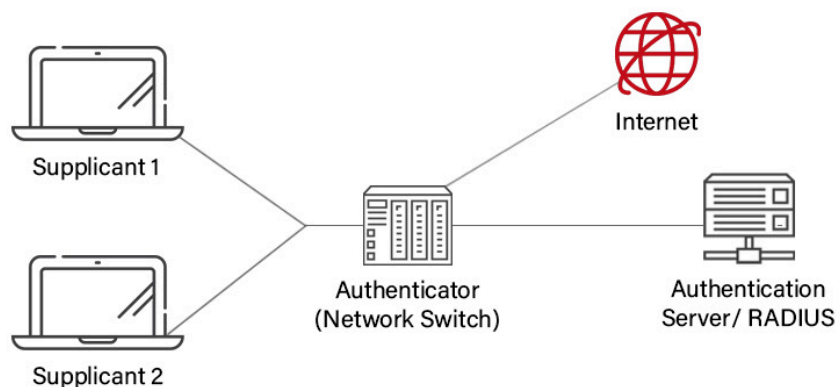


그림 1 802.1x 표준 인증 방식

(출처: www.cloudradius.com)

802.1x는 네트워크 접근 제어를 위한 IEEE 표준 프로토콜로, 유선 및 무선 네트워크에 연결하려는 장치나 사용자가 네트워크에 접근하기 전에 인증을 받도록 하는 포트 기반 인증 방식이다. 802.1x는 요청자(Supplicant), 인증자(Authenticator), 그리고 인증 서버(Authenticator Server)의 세 가지 주요 구성 요소로 동작하며, 인증이 성공하면 포트를 통해 권한이 부여된 디바이스만 네트워크에 접근할 수 있게 한다.

802.1x 프로토콜을 사용한 네트워크 접근 제어 시스템은 인증되지 않은 디바이스의 네트워크 접근을 막고, 중앙 집중화식 관리를 통해 각 디바이스의 접근 권한을 일괄적으로 관리할 수 있다는 장점이 있다. 또한, 네트워크 접속 기록이 저장되어 사후 보안 감사 및 포렌식에도 강점을 가진다. 이러한 강점을 바탕으로 802.1x는 기업, 대학, 공공 기관 등

다양한 조직의 유·무선 네트워크에서 표준적으로 활용되고 있다. 그러나, 이 프로토콜만으론 이미 인증 및 인가가 완료된 디바이스를 악의적인 목적으로 사용하는 내부자로부터의 공격에 대처할 수 없다.

예를 들어, 인증을 정상적으로 받은 사용자가 악의적인 의도를 가지고 민감 데이터를 무단으로 반출하거나, 계정을 탈취한 공격자가 합법적인 사용자를 가장해 내부 네트워크에 접근하는 상황을 가정할 수 있다. 이러한 상황은 802.1x가 보장하는 접속 시점의 인증만으로는 탐지할 수 없다. 따라서 위와 같은 유형의 위협에 대처하기 위해선, 사용자의 행동 로그를 수집하고 행동 패턴을 분석해 이상 사용자를 탐지 및 네트워크로부터 격리하는 새로운 방식의 네트워크 접근 제어 시스템이 요구된다.

1.3. 연구 목표

본 연구의 목표는 특정 조직의 네트워크 상에서, 사용자의 행동 로그를 실시간으로 수집 및 분석해 이상 사용자를 식별한 뒤, 해당 사용자가 사용 중인 PC를 자동으로 네트워크에서 격리하는 보안 시스템을 구축하는 것이다. 이러한 목적을 달성하기 위해 전체 시스템을 서로 다른 역할을 수행하는 4개의 모듈(로그 수집 및 저장 모듈, 이상 탐지 모듈, 네트워크 제어 모듈, 모니터링 모듈)로 구성한다. 각 모듈은 모듈별 특성에 적합한 기술을 적용해 구현한다.

- 로그 수집 및 저장 모듈

각 사용자의 PC로부터 여러 종류의 행동 로그를 수집 및 메인 프레임으로 전송하는 로그 수집 에이전트와, 메인 프레임에서 로그를 수신하고 적절한 형태로 변환해 데이터베이스에 저장하는 로그 수집 서버를 구현한다.

- 이상 탐지 모듈

공개된 오픈 소스 내부자 위협 데이터 세트로부터, 두 종류의 앙상블 트리 모델 기법(Random forest, XGBoost)을 사용해 이상 행위 시나리오를 분류하는 모델을 학습시키고, 각 모델에 대한 성능 평가를 수행한 후 더 나은 성능을 보이는 모델을 선택한다. 선택한 모델을 활용해, 데이터 베이스에 저장된 사용자별 행동 로그로부터 이상 사용자 목록을 식별하는 이상 탐지 파이프라인을 구축한다.

- 네트워크 제어 모듈

가상 PC 및 라우터로 구성된 가상의 조직 네트워크를 구축한다. 가상 라우터에서

주기적으로 현재 연결된 PC의 목록을 스캔해 메인 프레임에 보고하는 데몬²을 개발한다. 또한, 특정 PC에 대한 접근 제어 요청이 발생하면 라우터의 SSH³ 서버에 접속해 방화벽 규칙을 적용하는 자동화 모듈을 개발한다.

- 모니터링 모듈

조직의 보안 관리자가 시스템 전체 현황을 직관적으로 확인할 수 있는 웹 페이지를 제공하는 모듈을 구현한다. 조직 네트워크 보안 통계를 확인할 수 있는 대시보드 페이지, 각 PC의 접근 제한 상태를 확인하고 수동으로 접근을 허용하거나 차단할 수 있는 페이지, 트리 모델 기반 이상 탐지를 수행하고 이전 결과를 확인할 수 있는 페이지 등을 개발한다. 또한, 이상 사용자의 로그온을 탐지할 경우 관리자의 이메일에 경고 알림을 전송하는 기능을 개발한다.

마지막으로 제안하는 시스템의 실효성을 검증하기 위해, 이상 사용자 식별 모델의 성능과 행동 로그 수집을 위한 에이전트 프로그램의 PC 자원 소모량을 정량적으로 평가한다.

2. 연구 배경

2.1. 개발 환경

2.1.1. 적용 기술 및 환경

[표 1]은 시스템 개발에 적용된 기술 및 환경의 버전 정보 목록이다.

번호	분야	적용 기술 및 환경
1	프로그래밍 언어	<ul style="list-style-type: none"> ● Python 3.10.14 ● JavaScript 18.19.1
2	웹 프레임워크	<ul style="list-style-type: none"> ● React 19.1.1 ● FastAPI 0.103.1
3	배포 환경	<ul style="list-style-type: none"> ● Docker 28.3.0
4	데이터베이스	<ul style="list-style-type: none"> ● PostgreSQL 15.4

² 백그라운드(Background)에서 실행되며 특정 서비스나 기능을 지속적으로 제공하는 프로세스

³ Secure Shell, 네트워크를 통해 원격 시스템에 안전하게 접속하고 명령을 수행할 수 있도록 암호화된 통신 채널을 제공하는 프로토콜

5	머신 러닝 라이브러리	<ul style="list-style-type: none"> ● Sckit learn
6	가상 조직 네트워크 구축	<ul style="list-style-type: none"> ● Virtual Box ● Ubuntu LTS 24.04 ● OpenWRT
7	로그 수집 에이전트	<ul style="list-style-type: none"> ● Mitmproxy 12.1.2

표 1 분야 별 사용 기술 및 도구

2.1.2. 기술 적용 목적

2.1.2.1. Virtual box

VirtualBox는 단일 하드웨어에서 여러 개의 독립적인 가상 머신을 생성하고 실행할 수 있는 범용 하이퍼바이저⁴ 기반의 가상화 소프트웨어이다. 본 연구에서는 Virtual Box 기반의 PC 및 라우터 가상 머신들을 생성하고 각각 다른 네트워크 설정(IP 주소, MAC 주소 등)을 부여하여, 실험에 필요한 가상 네트워크 환경을 구축했다.

2.1.2.2. Fluentd

Fluentd는 다양한 소스에서 발생하는 로그를 수집하고 가공하여 전송하는 오픈소스 데이터 수집기로, 플러그인 기반 아키텍처를 통해 Input/Parser → Filter → Buffer → Output 파이프라인을 유연하게 구성할 수 있다. 본 연구에서는 각 에이전트의 로그 스트림을 Fluentd로 집약하고, Input/Parser 단계에서 원시 로그를 태그와 레코드로 구조화하였다. Filter 단계에서 일관된 JSON 스키마로 정규화하고 필드를 〇 보강하며 노이즈를 제거했다. Buffer 단계에서는 이벤트를 청크 단위로 보존하고 재시도 정책을 적용하여 전송 안정성을 확보했으며, Output 단계에서 JSON으로 직렬화하여 중앙 수신 API로 전달하였다.

2.1.2.3. auditd

auditd는 규칙 집합에 따라 시스템 호출·프로세스 실행·파일 접근·인증 등 보안 이벤트를 구조화하여 기록하는 Linux 커널 감사 데몬이다. 본 연구에서는 세션 시작·종료, USB 마운트·언마운트, 파일 복사를 대상으로 한 규칙 집합을 적용하여 보안 관점의 핵심 사용자

⁴ 하나의 물리적 하드웨어에서 여러 개의 가상 머신을 생성 및 관리하여, 각 운영체제가 독립적으로 실행될 수 있도록 지원하는 가상화 소프트웨어

행위 로그를 수집하였다.

2.1.2.4. Mitmproxy

Mitmproxy는 HTTP/HTTPS 트래픽을 중간에서 가로채 분석하고 조작할 수 있는 대화형 오픈소스 프록시⁵로, 자체 인증서 기반 TLS 프록시, WebSocket 지원, Python 애드온을 통한 확장성을 제공한다. 본 연구에서는 이를 엔드포인트와 외부 네트워크 사이의 경유 지점으로 구성하여 웹메일을 포함한 사용자 트래픽을 수집하고, 웹메일 메타데이터와 URL 등 핵심 지표를 추출하였다.

2.1.2.5. FastAPI

FastAPI는 Python 기반의 고성능 웹 프레임워크로, RESTful API와 WebSocket을 통해 데이터를 제공하고 실시간 알림을 처리한다. 본 연구에서는 FastAPI를 활용해 사용자 행동 로그와 이상 탐지 결과를 제공하며, JWT 기반 인증으로 보안성을 강화하였다.

2.1.2.6. Docker

Docker는 컨테이너 기반의 가상화 플랫폼으로, 애플리케이션과 그 실행에 필요한 모든 종속성을 하나의 격리된 환경으로 묶어 패키징한다. 이를 통해 개발, 테스트, 운영 환경 간의 일관성을 보장하여 배포 안정성을 높인다. 본 연구에서는 Docker를 이용해 PostgreSQL 데이터베이스와 pgAdmin⁶을 컨테이너로 구축하여 데이터베이스 초기 설정에 필요한 복잡한 과정을 단순화했다.

2.1.2.7. OpenWRT

OpenWRT는 라우터 시스템을 위해 개발된 리눅스 기반의 오픈 소스 운영체제이다. 사용자가 직접 시스템에 접근하여 패키지를 설치하고 방화벽, 네트워크 인터페이스, DHCP 서버 등의 네트워크 설정을 제어할 수 있는 확장성을 제공한다. 본 연구에서는 OpenWRT를 가상 머신 형태로 설치하여 테스트 베드의 가상 라우터로 활용하였다.

2.1.2.8. PostgreSQL

PostgreSQL은 오픈 소스 객체-관계형 데이터베이스 관리 시스템(ORDBMS)으로, 복잡한 데이터 구조와 대규모 트랜잭션을 안정적으로 처리할 수 있는 기능을 제공한다. 본

⁵ 클라이언트와 서버 사이에서 중계자 역할을 수행하여, 요청과 응답을 대신 전달하거나 필터링, 캐싱, 보안 기능을 제공하는 서버 또는 소프트웨어

⁶ PostgreSQL 데이터베이스를 관리하기 위한 오픈소스 그래픽 사용자 인터페이스(GUI) 도구

연구에서는 PostgreSQL을 활용하여 조직, 사용자, PC, 행동 로그, 이상 탐지 결과 등의 데이터를 저장하고 관리하며, SQLAlchemy⁷를 사용해 데이터베이스와 백엔드 서버의 연동을 구현했다.

2.1.2.9. NodeJS/React

본 연구에서는 React를 핵심 프레임워크로 사용하여 모니터링 시스템의 프론트엔드를 개발하였다. React의 컴포넌트 기반 아키텍처와 상태 관리 기능을 활용하여 실시간으로 수집되는 사용자 행동 로그와 이상 탐지 결과를 시각화하고 네트워크 차단 내역을 모니터링하는 UI를 구현하였다. 또한 NodeJS는 개발 서버 운영과 빌드 프로세스 관리를 위한 런타임 환경으로 사용되었으며, 백엔드 API와의 비동기 통신을 처리하는 역할을 담당했다.

2.2. 데이터베이스 설계

본 연구에서 시스템은 각 모듈이 모두 동일한 하나의 중앙 데이터 베이스와 통신하도록 설계되었다. 이러한 설계는 데이터에 대한 통합 관리를 가능하게 하며, 모든 로그와 탐지 결과가 중앙에서 집계 및 저장됨으로써 조회 연산의 효율성을 올릴 수 있다. 그러나, 다수의 모듈에서 동시 접근이 이루어지기 때문에 정규화, 무결성 제약, 참조 관계 설정 등 엄격한 데이터베이스 설계가 요구된다. 따라서, 정확한 설계와 협업의 용이성을 위해 ERD⁸를 작성한다. [그림 2]는 시스템 중앙 데이터 베이스 테이블의 ERD이며, [표 2]는 DB의 각 테이블에 대한 설명이다.

⁷ 파이썬에서 관계형 데이터베이스를 효율적으로 다룰 수 있도록 지원하는 라이브러리

⁸ Entity-Relationship Diagram, 데이터베이스에 저장되는 객체와 이들 간의 관계를 시각적으로 표현하는 다이어그램

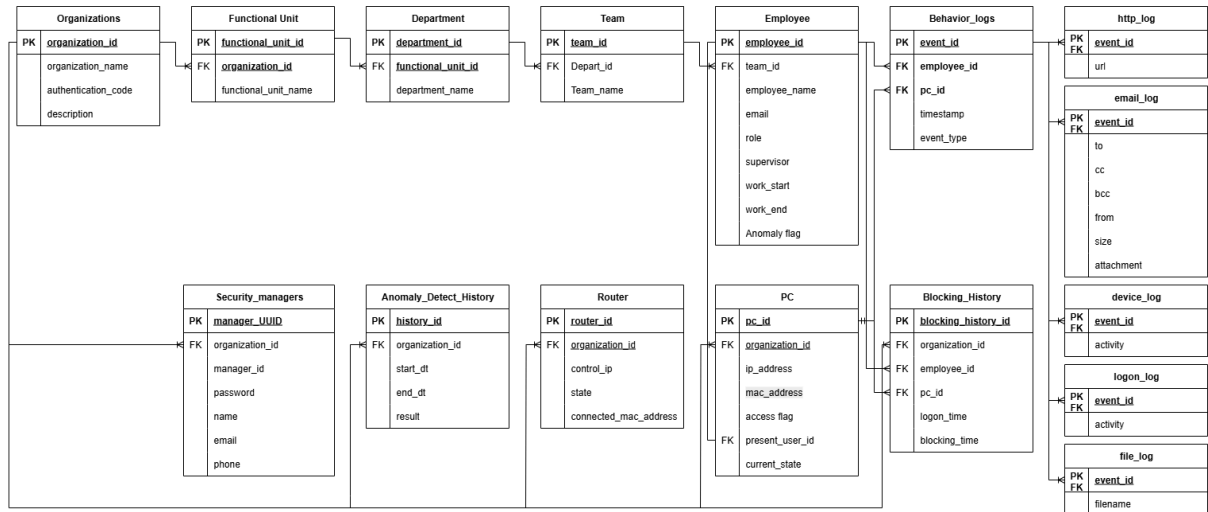


그림 2 ERD 다이어그램

번호	테이블명	설명
1	organization	조직의 id, 이름, 그리고 가입 인증을 위해 필요한 코드를 저장한다.
2	Functional unit	조직의 기능 단위의 이름과 id를 저장한다.
3	Department	조직의 부서 이름과 id를 저장한다.
4	Team	조직의 팀 이름과 id를 저장한다.
5	Employee	조직의 직원 정보(이름, 이메일, 역할, 상사, 입사/퇴사 날짜)와 이상 여부를 저장한다.
6	Behavior log	여러 유형의 행동 로그에서 공통적으로 저장되는 속성(직원 id, pc id, 시각, 행동 유형)을 저장한다.
7	Http log	Http 로그 상세 정보를 저장한다.
8	Email log	이메일 로그 상세 정보를 저장한다.
9	Device log	디바이스 연결 로그 상세 정보를 저장한다.
10	File log	파일 복사 로그 상세 정보를 저장한다.
11	Security manager	조직의 보안 관리자 정보를 저장한다.

12	Anomaly detect history	이상 탐지 결과 이력을 저장한다.
13	Blocking History	이상 사용자의 로그인 PC 차단 기록을 저장한다.
14	Router	조직 네트워크에 존재하는 라우터의 정보와 해당 라우터에 연결된 PC의 MAC 주소 목록을 저장한다.

표 2 데이터 베이스 테이블 목록

2.3. API 설계

본 연구에서 시스템은 웹 기반 관리 페이지와 백엔드 서버 간의 원활한 통신을 위해 RESTful API를 제공하도록 설계되었다. [표 3]은 본 연구에서 구현된 주요 백엔드 서버 API 목록을 정리한 것으로, 각 기능의 목적, 요청 URL, HTTP Method를 명시하여 개발 및 협업 과정의 효율성을 높이고, 향후 유지 보수성을 용이하게 한다.

번호	기능	URL	Method
1	회원가입	/auth/signup	POST
2	로그인	/auth/signin	POST
3	새로고침 토큰 발급	/auth/refresh	POST
4	로그인된 사용자 정보 반환	/auth/me	GET
5	조직 이름 및 설명 반환	/organizations/{organization_id}/name&description	GET
6	조직 인증 코드 검증	/organizations/{organization_id}/verify	POST
7	특정 기간 동안의 이상 탐지	/anomalydetect/{organization_id}	GET
8	이상 사용자 목록 조회	/anomalydetect/{organization_id}/anomaly_employees	GET
9	이상 탐지 이력 조회	/anomalydetect/{organization_id}/get-histories	GET

10	주 별 이상 사용자 수 조회	/anomalydetect/get-anomaly-user-counts-by-week/{organization_id}/	GET
11	특정 PC의 접근 제어	/network_access_control/{organization_id}/{pc_id}/{access_flag}	GET
12	네트워크 차단 이력 조회	/network_access_control/{organization_id}/blocking-network-histories	GET
13	현재 PC 상태 조회	/pcs/pc_state/{organization_id}/	GET
14	로그온 비율 조회	/pcs/get_pc_logon_percent/{organization_id}/	GET
15	전체 행동 로그 조회	/behavior-logs/{organization_id}	GET
16	행동 로그 조회 필터 옵션 제공	/behavior-logs/facets	GET
17	사용자별 행동 로그 조회	/behavior-log/user	GET
18	월별 주간 행동 로그 수 조회	/behavior-log/monthly-type-counts	GET

표 3 백엔드 서버 API 목록

2.4. 머신 러닝 알고리즘

2.4.1. 결정 트리

결정 트리(Decision Tree)는 데이터의 특성 값을 기준으로 분할 규칙을 반복적으로 적용하는 의사 결정 과정을 트리 구조로 표현한 지도 학습(Supervised learning)⁹ 알고리즘이다. 의사 결정 트리는 학습 과정에서 루트 노드에서 시작해 각 내부 노드는 특정 속성에 대한 조건 검사를 수행하고, 분기를 따라가며 데이터를 하위 집합으로 나눈다. 이러한 과정을 통해 최종적으로 리프(leaf) 노드에 도달하면 예측 결과, 클래스 레이블이 할당된다.

결정 트리는 노드의 분할 기준을 결정할 때 지니 불순도(Gini Impurity)를 사용한다. 지니 불순도는 특정 노드에 포함된 데이터 집합의 불순도를 측정하며, 한 노드에 속한 데이터들이 얼마나 다양한 클래스에 걸쳐 분포되어 있는지를 나타낸다. 만약 하나의 클래스만 존재한다면 지니 불순도는 0이 되며, 반대로 클래스가 고르게 섞여 있을 수록 값이 커진다.

$$G = \sum_{k=1}^K \hat{p}_{mk}(1 - \hat{p}_{mk}) = 1 - \sum_{k=1}^K \hat{p}_{mk}^2$$

수식 1 지니 불순도의 정의

[수식 1]은 지니 불순도의 정의식이다. K 는 클래스의 개수, \hat{p}_{mk} 는 노드 m 에서 클래스 k 에 속하는 데이터의 비율을 의미한다. 이 수식은 어떤 데이터가 각 클래스에 속할 확률을 제공해 더한 값을 1에서 뺀 형태로, 값이 작을수록 특정 클래스에 데이터가 집중되어 있음을 나타낸다. 따라서 의사 결정 트리는 각 분할 단계에서 지니 불순도가 가장 크게 감소하는 방향으로 데이터를 분리함으로써, 순수한(단일 클래스만 존재하는) 리프 노드를 형성하는 방향으로 학습된다.

2.4.2. SMOTE Oversampling

SMOTE(Synthetic Minority Over-sampling Technique)는 불균형 데이터 문제를 완화하기 위해 제안된 기법으로, 소수 클래스(minority class)의 데이터를 단순 복제하는 대신 새로운 합성 샘플을 생성한다. 구체적으로는 소수 클래스의 데이터 포인트와 그 주변 이웃 샘플을 선택한 뒤, 이들 사이의 선형 보간(Linear Interpolation)¹⁰을 통해 새로운 데이터를

⁹ 입력 데이터와 정답 레이블을 기반으로 예측 모델을 학습하는 머신 러닝 방법론

¹⁰ 두 점 사이의 값을 직선 비율에 따라 추정하는 보간 기법

만들어낸다. 이를 통해 소수 클래스의 샘플 수를 늘려 모델의 예측 성능을 향상할 수 있다. 본 연구에서는 사용한 데이터 세트 중 악성 사용자의 로그 수 비율이 매우 적은 데이터의 특성을 보완하기 위해 SMOTE를 적용했다.

2.4.3. Random Forest

Random Forest는 의사 결정 트리를 기반으로 한 앙상블 학습(Ensemble Learning) 기법으로, 다수의 의사 결정 트리를 학습시킨 뒤 이들의 예측을 종합해 최종 결과를 도출한다. 각 트리는 데이터 샘플을 부트스트래핑(Bootstrapping)¹¹ 방식으로 무작위 추출하여 학습하며, 분할 과정에서도 일부 특징(feature)을 임의로 선택해 사용한다. 이러한 무작위성은 개별 트리의 상관 관계를 줄여주어, 단일 트리가 갖는 과적합(Overfitting)¹² 문제를 완화시킨다. 최종 클래스는 다수결 투표 방식으로 결정된다.

2.4.4. XGBoost

부스팅(Boosting)은 약한 학습기를 순차적으로 결합해 강한 학습기(Strong Learner)를 만드는 앙상블 학습 기법이다. 초기 모델은 단순한 예측으로 시작해, 각 단계에서 이전 모델이 잘못 예측한 잔차(residual)를 새로운 약한 학습기가 보완하도록 학습한다. 이렇게 반복적으로 학습기를 추가하면서 모델은 점차 복잡한 패턴을 포착하고, 여러 약한 학습기의 예측을 합산해 최종 예측을 출력한다. 학습률 λ 와 반복 횟수에 따라 성능 및 일반화 능력이 조절된다.

XGBoost는 부스팅 알고리즘을 확장 및 최적화한 구현체로 Gradient Boosting의 개념을 기반으로 한다. XGBoost에서는 각 단계에서의 목적 함수(Objective Function)를 [수식 2]와 같이 정의한다.

$$L^{(t)} = \sum_{i=1}^n l(y_i, \hat{y}_i^{(t-1)} - f_t(x_i)) + \Omega(f_t)$$

수식 2 XGBoost의 목적 함수 정의

[수식 2]에서 l 은 예측값과 실제값의 차이를 나타내는 손실 함수(loss function), $\hat{y}_i^{(t-1)}$ 는 이전 단계의 예측값, $f_t(x_i)$ 는 새로 추가되는 약한 학습기(결정 트리), $\Omega(f_t)$ 는 모델 복잡도를 제어하기 위한 정규화 항이다. 정규화항 $\Omega(f_t)$ 는 [수식 3]과 같이 정의된다.

¹¹ 데이터를 복원 추출하여 여러 표본을 생성하고 통계적 추정을 수행하는 재표본화 기법

¹² 학습 데이터에는 높은 성능을 보이나 새로운 데이터에는 일반화 성능이 떨어지는 현상

$$\Omega(f) = \gamma T + \frac{1}{2} \lambda \sum_{j=1}^T w_j^2$$

수식 3정규화 항의 정의

[수식 3]에서 T 는 트리의 리프 노드 개수, w_j 는 각 리프 노드의 가중치, γ, λ 는 정규화 파라미터로 모델의 복잡도를 억제해 과적합을 방지하는 역할을 한다. 이러한 정규화 항을 목적 함수에 적용함으로써, XGBoost는 손실 최소화과 동시에 모델 복잡도 제어를 함께 수행한다. 따라서, 예측 성능과 일반화 성능을 균형 있게 확보할 수 있다.

3. 연구 내용

3.1. 전체 시스템 설계

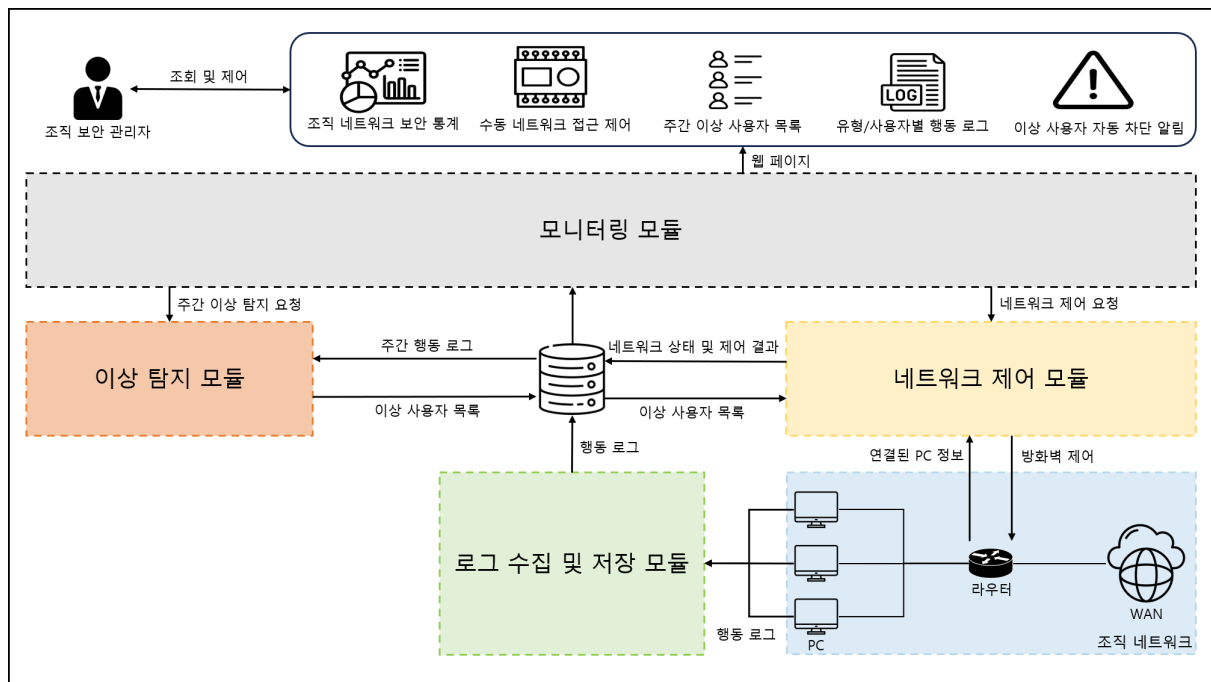


그림 3 전체 시스템 설계

전체 시스템은 서로 다른 역할 및 책임을 가지는 서로 다른 4개의 모듈들로 구성된다. [그림 3]은 각 모듈 간 어떤 정보를 주고받는지 나타낸 시스템 설계도다. 로그 수집 및 저장 모듈은 조직 내 PC에서 발생하는 사용자의 행동 로그를 수집하고 이를 중앙 저장소에 보관한다. 이상 탐지 모듈은 저장된 로그 데이터를 기반으로 사용자 행위를 분석하여 이상 여부를 판별하고, 탐지된 결과를 바탕으로 이상 사용자 목록을 생성한다. 네트워크 제어 모듈은 탐지된 이상 사용자 PC의 네트워크 연결을 허용하거나 차단하는 역할을 수행하며, 라우터와 연동되어 정책을 적용한다. 마지막으로 모니터링 모듈은 보안 관리자가 시스템

전체 현황을 직관적으로 확인할 수 있도록 네트워크 상태, 이상 사용자 기록, 차단 내역 등을 웹페이지로 시각화해 제공한다.

3.2. 로그 수집 및 저장 모듈

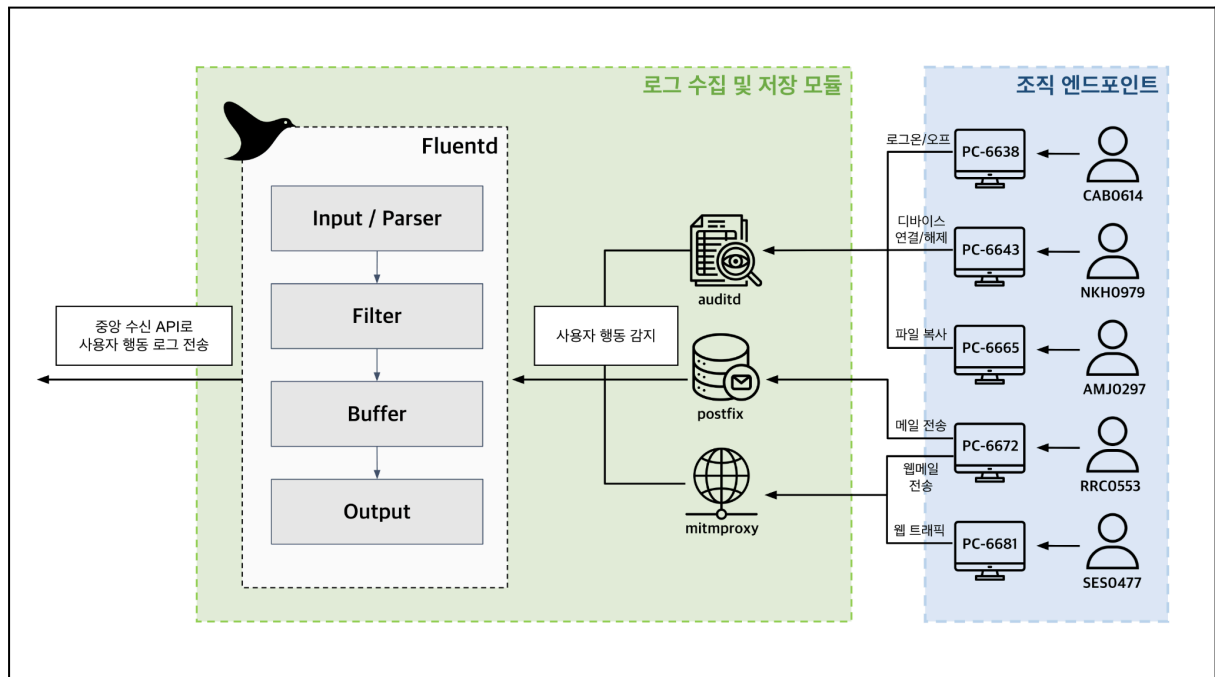


그림 4 사용자 행동 로그 수집 파이프라인

로그 수집 및 저장 모듈은 사용자 PC에서 발생하는 행동 이벤트들을 감지한 다음, 중앙 데이터베이스로 전송하는 모듈이다. 각 PC에 설치된 로그 수집 에이전트가 OS 및 애플리케이션 수준의 이벤트를 포착하면, 수송 계층 애플리케이션(**Fluentd**)이 원본 로그 데이터를 소스별로 파싱한 다음 해석·필터링·전처리·버퍼링한다. 마지막으로 시스템 표준 스키마에 맞춰 필드를 정규화하여 중앙 수신 API로 전송한다. [그림 4]는 사용자 PC에서 발생한 사용자 행동 이벤트가 수집 및 정규화 되어 중앙 데이터베이스로 전달되는 흐름을 나타낸다.

3.2.1. 구성요소

로그 수집 에이전트는 행동 유형 별로 서로 다른 경로를 통해 행동 로그를 수집한다. 로그인·로그오프, USB 마운트·언마운트, USB로의 파일 복사와 같은 커널 수준 이벤트는 **auditd** 규칙으로 캡처하며, 메일은 **Postfix**의 **always_bcc**와 **Maildir** 아카이빙으로 송·수신 메타데이터를 확보한다. 웹메일을 포함한 HTTP 계열 트래픽은 **mitmproxy** 프록시에서 요청 메타를 추출한다. 수송 및 전처리는 **Fluentd**가 맡아 소스별 파서를 통해 원문을 해석하고,

불필요한 시스템 노이즈를 제거한 뒤 시스템 표준 스키마에 맞게 필드를 정규화하여 중앙 수신 API로 전송한다.

3.2.2. 사용자 행동 로그 수집 범위 및 방법

3.2.2.1. 로그인·로그오프

로그인·로그오프는 **auditd**가 생성하는 사용자 세션 이벤트를 활용하여 포착한다. 커널이 기록하는 **USER_START**와 **USER_END**를 신뢰 기준으로 삼고, 시스템 계정·백그라운드 프로세스·그래픽 세션 워커 등 잡음성 프로세스에 기인한 레코드는 사용자 영역 식별자와 실행 파일·프로세스명 단서를 활용하여 일괄 제외하였다.

3.2.2.2. 웹 브라우징

웹 브라우징 트래픽은 **mitmproxy** 기반 프록시를 활용하여 수집한다. 브라우저와 시스템에 프록시를 지정하여 모든 **HTTP(S)** 요청이 프록시를 경유하도록 구성한 뒤, 요청 객체를 프록시 스크립트로 해석하여 **URL**을 추출한다.

3.2.2.3. 메일

메일은 서버 측 메일과 브라우저 기반 웹메일을 구분하여 처리한다. 서버 측 메일은 **Postfix**의 **always_bcc** 기능을 통해 모든 송·수신 메시지에 보관용 숨은 참조(**Bcc**)를 자동 부착하여 사본을 남기고, 이를 통해 메타데이터를 확보한다. 한편 웹메일은 **SMTP/IMAP**이 아닌 **HTTP**로 전송되므로 **mitmproxy**에서 관련 요청 패턴을 식별한 뒤 요청 헤더와 폼 필드를 파싱하여 메타데이터를 추출한다.

3.2.2.4. 디바이스

디바이스 연결 및 해제는 **Linux**의 마운트/언마운트 이벤트를 근거로 감지한다. 사용자 영역에서 일관된 마운트 지점을 확보하기 위해 **udisks2/udiskie** 기반 자동 마운트를 사용하고, 해당 상태 변화를 **auditd**가 기록하도록 감사 규칙을 설정하였다.

3.2.2.5. 파일 복사

파일 복사 이벤트는 디바이스의 마운트 지점으로의 파일 복사 작업을 **auditd** 감사 규칙으로 포착하여 수집하며, 수집된 감사 레코드의 **PATH** 항목에서 경로를 파싱하여 파일명을 추출한다.

3.2.3. 표준 로그 스키마

모든 레코드는 공통 필드와 행동 유형별 확장 필드의 조합으로 저장된다. 공통 필드는 **event_id**, **employee_id**, **pc_id**, **timestamp**, **event_type**으로 구성된다. **event_id**는 각 로그를 고유하게 식별하는 기본 키(Primary Key)이다. **employee_id**는 이벤트 주체가 된 사용자, **pc_id**는 이벤트가 발생한 PC의 식별자, **timestamp**는 이벤트의 실제 발생 시각이다. **event_type**은 행동 유형을 나타내는 분류 필드로, **login**, **http**, **file**, **device**, **email** 중 하나를 값으로 가지며 행동 유형별 로그 테이블과의 조인 기준이 된다. [표 4]는 행동 유형별 확장 필드의 구성을 나타낸다.

행동 유형	확장 필드
login	activity (login or logoff)
http	url
file	filename
device	activity (connect or disconnect)
email	from_addr, to, cc, bcc, size, attachment

표 4 행동 유형별 확장 필드

3.3. 이상 탐지 모듈

이상 탐지 모듈의 개발은 이상 탐지 모델 학습과 이상 탐지 파이프 라인 구현의 두 단계로 나뉜다. 이상 탐지 모델 학습 단계에서는 오픈 소스 데이터 세트를 활용해 이상 탐지를 위한 모델을 학습한다. 이상 탐지 파이프 라인 구현 단계에서는 앞서 학습한 모델을 사용해 새로운 로그 데이터를 받아 전처리 과정을 거친 뒤, 각 사용자 행위가 정상인지 이상인지, 이상이라면 어떤 분류에 해당하는지 예측한다. 예측 결과는 이상 사용자 목록으로 변환되어 데이터 베이스에 저장된다.

3.3.1. 이상 탐지 모델 학습

3.3.1.1. 데이터 세트

데이터 세트는 카네기 멜론 대학에서 개발한 내부자 위협 데이터 세트, **CMU CERT R4.2**를 사용했다[5]. 이 데이터 세트는 1000명의 직원이 존재하는 조직을 모델링하고 있으며, 각

직원의 로그인/로그오프, 파일 복사, **http** 요청, 디바이스 연결, 이메일 송신 등 다양한 사용자 행동 로그를 포함한다. [표 5]는 해당 데이터 세트가 포함하는 행동 유형 별 속성을 나타낸다.

행동 유형	속성
logon	Id, User, Date, PC, Activity (logon or logoff)
http	Id, User, Date, PC, URL, Content
file	Id, User, Date, PC, Filename, Content
device	Id, User, Date, PC, From, To, Cc, Bcc, Size, Attachment, Content
email	Id, User, Date, PC, Activity (connect or disconnect)

표 5 행동 유형 별 속성

또한 데이터 세트엔 정상 행위뿐만 아니라 내부 정보 유출, 기술 절취, 시스템 파괴 등의 시나리오를 수행하는 사용자의 데이터 역시 포함되어 있다. [표 6]은 각 사용자 행위 시나리오의 4가지 분류다.

시나리오 (Class)	상세
0	정상 사용자
1	근무 시간 외에 로그인하여 이동식 드라이브를 연결한 뒤 악성 사이트에 데이터를 업로드 후 퇴사
2	구직 웹사이트를 탐색하고 회사를 떠나기 전에 이전 활동보다 훨씬 더 빈번하게 USB 드라이브를 사용
3	키로거를 다운로드해 USB 드라이브를 통해 상사의 PC 에 설치, 다음날 수집된 키로그를 사용해 상사 계정으로 로그인한 다음 조직 내에 대량의 이메일을 발송 후 퇴사

표 6 행위 시나리오 분류

3.3.1.2. 주 단위 집계 및 특징 추출

원본 데이터를 모델 학습에 알맞은 형태로 변환하는 과정이다. 먼저, 행동 유형별로

나뉘어 있는 원본 데이터를 모델 학습 및 클래스 예측 단위인 주(week) 단위로 집계한다. 다음으로 집계된 주 단위 데이터를 기반으로, 이상 행동을 잘 설명할 수 있는 주요 특징(feature)을 숫자 형태로 추출한다. [표 7]은 모델 학습을 위해 추출되는 특징의 행동 유형별 목록이다.

행동 유형	추출 특징
공통 특징	<ul style="list-style-type: none"> • user(사용자 ID 인덱스) • act(행동 유형) • pc(자신, 공유, 타인, 상사) • time(사용 시간대)
http	<ul style="list-style-type: none"> • http_type(도메인 유형) • url_len(url 길이) • url_depth(url 깊이)
file	<ul style="list-style-type: none"> • file_type(파일 유형) • file_len(파일 크기) • file_nwords(파일 내 단어 수) • disk(disk 위치) • file_depth(경로 깊이)
device	<ul style="list-style-type: none"> • usb_dur(usb 연결 후 지속 시간)
email	<ul style="list-style-type: none"> • n_des(수신자 수) • n_atts(첨부파일 개수) • Xemail(외부 이메일 전송 여부) • n_exdes(외부 수신자 수) • n_bccdes(BCC 수신자 수) • exbccmail(외부 bcc 여부) • email_size(이메일 크기)
label	<ul style="list-style-type: none"> • anomaly(시나리오 분류, 0~3)

표 7 행동 유형별 추출 특징

3.3.1.3. 모델 학습

모델 학습 단계에선 전처리가 완료된 주 단위 사용자 행동 데이터를 입력으로 받아 학습을

수행한다. 모델 학습 알고리즘으로는 두 가지 트리 기반 앙상블 알고리즘인 Randomforest와 XGBoost 분류기를 사용한다. 두 알고리즘을 사용해 학습한 모델의 성능을 평가하고, 둘 중 더 나은 성능을 보이는 모델을 최종적으로 선택한다.

학습 과정에서는 이상 데이터가 적은 불균형 문제를 해결하기 위해 SMOTE(Synthetic Minority Oversampling Technique) 기반의 오버 샘플링 기법을 적용하였다. 소수 클래스에 속하는 이상 사용자 데이터를 인공적으로 생성하여 다수 클래스와의 분포 차이를 줄임으로써 희귀한 이상 클래스를 효과적으로 학습할 수 있도록 하였다. 모델의 입력 변수는 [표 7]의 사용자별 행동 유형에 따라 추출된 주요 특징들로 구성되며, 각 변수들은 표준화 과정을 거쳐 학습에 사용되었다.

최종적으로 선택된 학습 모델은 정상 사용자와 이상 사용자를 구분하며, 이상 행위가 어떤 시나리오(내부 정보 유출, 구직 활동, 시스템 파괴 등)에 해당하는지 예측할 수 있다. 이렇게 학습된 이상 탐지 모델은 이후 파이프라인 단계에서 이상행위를 탐지하기 위해 사용된다.

3.3.2. 이상 탐지 파이프라인 구현

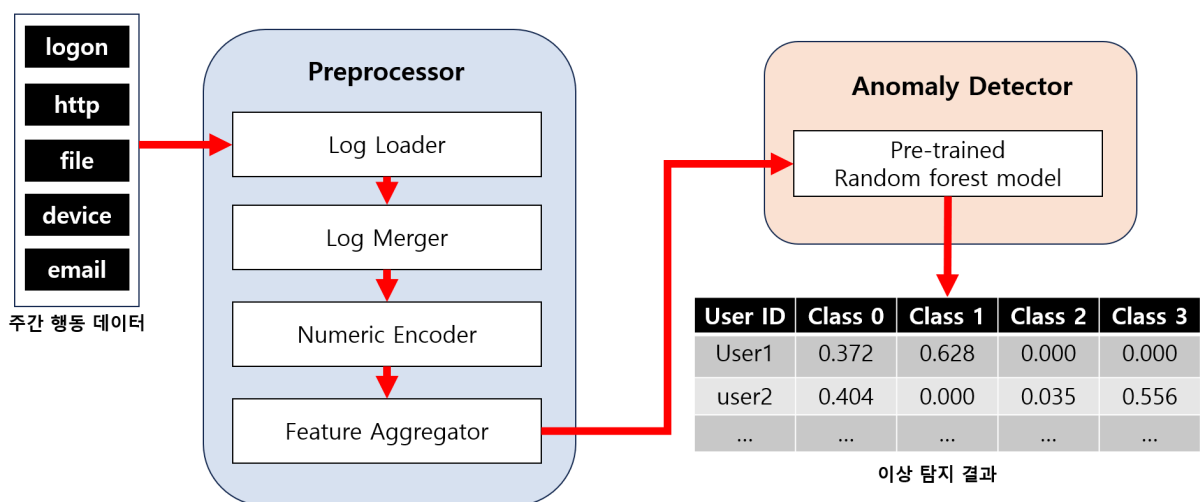


그림 5 이상 탐지 파이프라인 설계

이상 탐지 파이프 라인은 데이터 베이스에서 주 단위 행동 데이터를 조회해, 특정 주에 조직의 PC를 사용한 사용자들의 이상 클래스별 확률 분포를 반환한다. [그림 5]는 이상 탐지 파이프라인의 설계를 나타낸다. 이상 탐지 파이프라인은 전처리 클래스(Preprocessor)와 이상 탐지 클래스(Anomaly Detector)로 나뉜다.

전처리 클래스는 전처리 흐름에 따라 네 개의 세부 클래스를 호출하는 오케스트레이션 클래스다. 먼저, **Log Loader** 클래스는 데이터베이스로부터 특정 주에 행해진 모든 행동

데이터를 데이터프레임(data frame) 변수로 로드하는 클래스다. **Log merger** 클래스는 로드한 데이터프레임의 모든 유형별 행동 데이터를 동일한 포맷의 단일 데이터 프레임으로 병합하는 클래스다. **Numeric Encoder** 는 각 행동 데이터의 속성 중 문자열 속성을 숫자 자료형으로 변환하는 클래스다. 마지막으로, **Feature Aggregator** 클래스는 [표 7]의 특징들을 추출해, 각 행동 데이터 행에 추가하는 클래스다.

전처리 클래스를 거치며 전처리가 완료된 데이터들은 이상 탐지 클래스로 전달된다. 이상 탐지 클래스는 앞서 사전 학습된 모델을 사용해 각 사용자의 행동 패턴을 분석한다. 최종적으로 모델은 각 사용자가 어떤 클래스에 속하는지에 대한 확률 분포를 반환한다. 각 클래스는 [표 6]에서 확인할 수 있다. 정상 클래스 0를 제외한, 나머지 이상 클래스 확률이 0.5 이상을 넘어가는 사용자 목록은 데이터 베이스에 저장된다.

3.4. 네트워크 제어 모듈

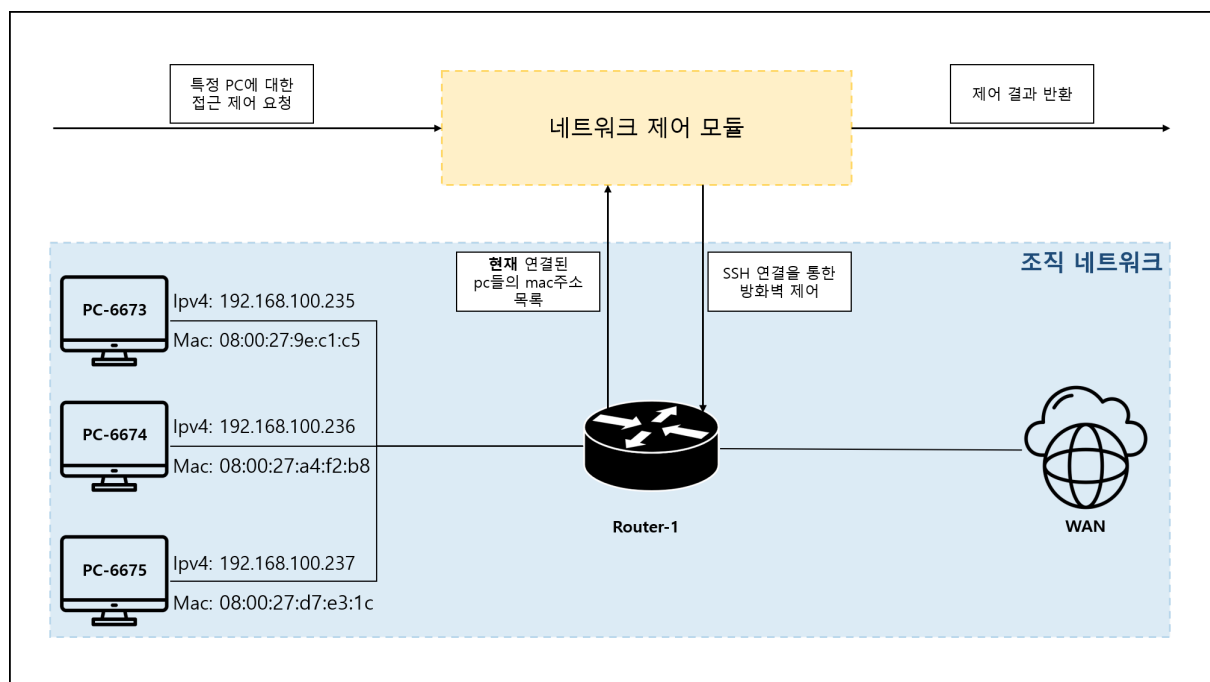


그림 6 네트워크 제어 모듈과 라우터의 연동

네트워크 제어 모듈은 조직 네트워크에 존재하는 라우터로부터 주기적으로 네트워크 상태를 보고 받고, 개별 PC에 대한 접근을 허용하거나 차단하는 역할을 수행한다. 본 연구에서 조직 네트워크는 여러 가상 PC들과 가상 라우터들로 구성되어 있다. [그림 6]은 네트워크 제어 모듈이 라우터와 연동하여 조직 내 PC들의 네트워크 접속을 관리하는 전체 구조를 보여준다. 네트워크 제어 모듈과 라우터는 두 가지 방식으로 통신한다.

먼저, 조직 네트워크 상에 존재하는 모든 라우터는 네트워크 상태를 파악하기 위한 데몬이 설치되어 있다. 이 데몬은 주기적으로 자신의 서브넷에 **ping sweep**¹³을 보내 서브넷 대역의 모든 IP 주소에 대한 연결 상태를 검사하고 **ARP** 테이블¹⁴을 업데이트한다. 다음으로, **ARP** 테이블에서 **REACHABLE** 상태인 디바이스들의 **MAC** 주소를 추출한다. 데몬은 이렇게 수집된 **MAC** 주소 목록을 라우터 고유 ID, 조직 ID와 함께 **JSON** 형태로 구성해 네트워크 제어 모듈의 **FastAPI** 엔드포인트로 전송한다. 네트워크 제어 모듈은 수신한 정보를 데이터베이스의 라우터 테이블에 업데이트한다.

만약 네트워크 제어 모듈이 특정 PC에 대한 네트워크 접근 제어 요청을 받으면, 네트워크 제어 모듈은 먼저 데이터베이스의 PC 테이블을 조회해 특정 PC의 **MAC** 주소를 확인한다. 다음으로, 라우터 테이블을 조회해 요청받은 PC와 연결된 라우터를 식별한다. 만약 해당 PC와 연결된 라우터가 존재하면, 라우터에서 구동 중인 **ssh** 서버에 접속해 요청받은 **MAC** 주소로부터 전송되는 패킷을 허용/차단하는 방화벽 규칙을 적용한다. 위 과정이 성공적으로 완료될 경우, PC의 접근 가능 상태를 업데이트하고 제어 결과를 반환한다.

3.5. 모니터링 모듈

모니터링 모듈은 조직 네트워크의 보안 상태를 실시간으로 관찰하고 관리할 수 있도록 지원하는 기능을 제공한다. 이 모듈은 인증된 보안 관리자가 접근할 수 있는 웹페이지를 제공하며, 조직 인증 및 관리자 회원가입, 로그인, 대시보드 제공, 이상 사용자 탐지 결과 조회, 행동 로그 조회, PC 상태 관리, 이상 사용자 로그인 알림 등 다양한 보안 관리 기능을 제공한다. 이를 통해 관리자는 조직 내 보안 상황을 직관적으로 파악하고, 이상 행위나 보안 위협 발생 시 즉각적으로 대응할 수 있다. 다음 절들은 각 페이지의 기능 및 흐름을 화면과 함께 설명한다.

3.5.1. 조직 인증 및 관리자 회원가입

회원 가입 시, 조직의 고유 ID와 인증 코드를 입력해 조직 인증을 완료해야 한다. 조직 인증이 성공하면, 관리자는 자신의 정보를 입력한 뒤 관리자 계정을 생성할 수 있다.

3.5.2. 로그인

로그인 화면에서 아이디와 비밀번호를 입력하여 시스템에 접근한다. 로그인 성공 시, 대시보드 화면으로 이동한다.

¹³ 네트워크에서 대상 호스트의 도달 가능성과 응답 시간을 확인하는 진단 명령어

¹⁴ IP 주소와 MAC 주소의 매핑 정보를 저장하는 네트워크 캐시 구조

3.5.3. 대시보드

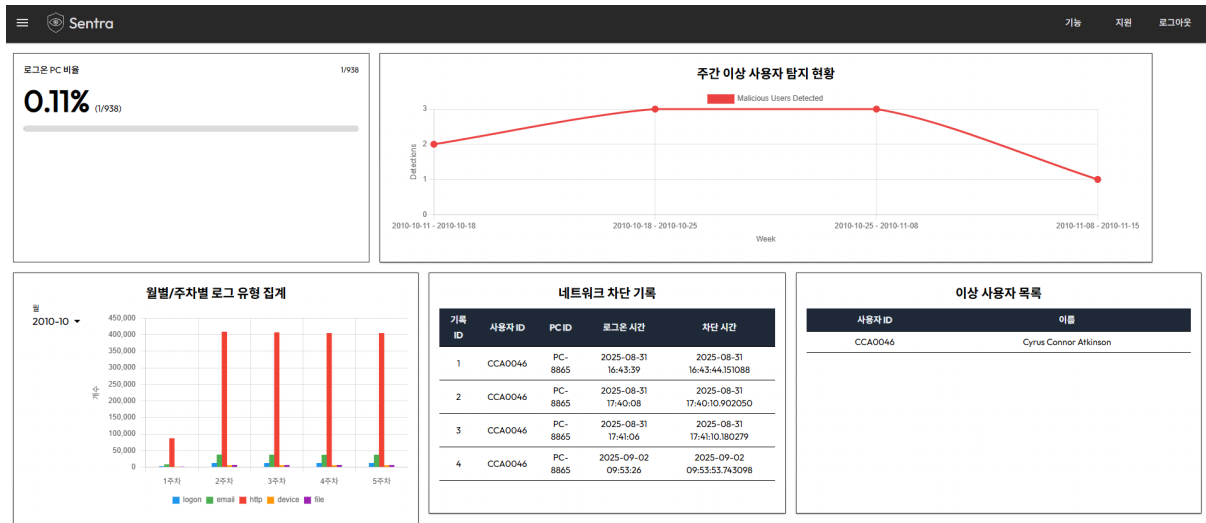


그림 7 대시 보드 페이지

대시보드는 조직 네트워크 보안 상태를 표시한다. 관리자는 5개의 통계 정보를 통해 조직의 현재 보안 상태 및 보안 위험도 변화 추세를 직관적으로 파악할 수 있다.

- 로그인 PC 비율
- 주간 이상 사용자 탐지 현황
- 월별/주차별 로그 유형 집계
- 네트워크 차단 기록
- 이상 사용자 목록

3.5.4. 이상 사용자 탐지 결과 조회

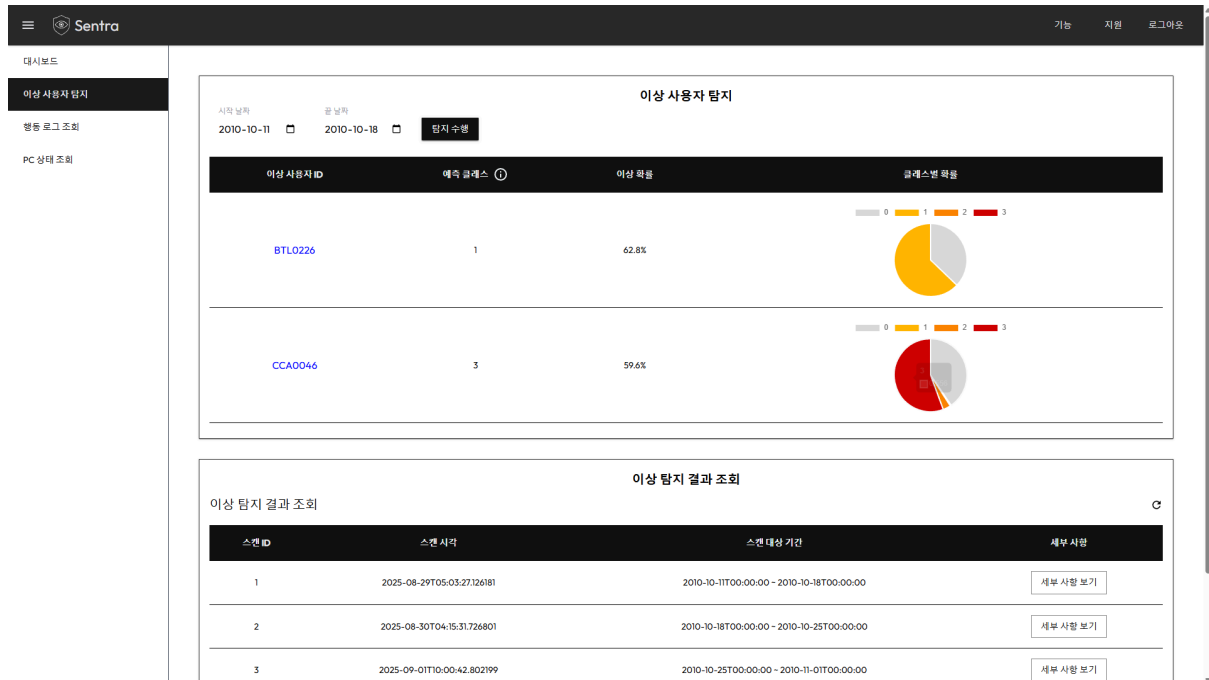


그림 8 이상 사용자 탐지 페이지

왼쪽 네비게이션 바에서 ‘이상 사용자 탐지’ 탭을 클릭하면, 관리자가 특정 기간 동안 시스템이 탐지한 이상 사용자 목록을 확인하고 그 상세 내역을 조회할 수 있다.

3.5.4.1. 이상 사용자 탐지

이상 사용자 탐지 컴포넌트는 조직 내 특정 기간 동안의 사용자 행동 데이터를 분석하여 이상 행동을 보이는 사용자를 탐지한다. 관리자는 탐지 기간을 입력하여 이상 사용자 탐지 작업을 수행할 수 있다.

탐지 작업이 완료되면, 테이블 형태의 탐지 결과가 화면에 표시된다. 탐지 결과는 사용자 별 예측 이상 시나리오 클래스, 이상 확률, 클래스별 확률(파이차트) 정보를 포함한다. 탐지된 사용자 ID를 클릭하면 해당 사용자의 최근 행동 로그를 조회할 수 있다.

3.5.4.2. 탐지 이력 조회

이상 탐지 결과 조회 컴포넌트는 이전에 수행된 이상 탐지 작업의 기록을 조회하는 기능을 제공한다. 탐지 기록은 스캔 ID, 스캔 시간, 스캔 대상 기간으로 구성되며, 관리자는 “세부 사항 보기” 버튼을 통해 탐지 작업의 결과를 상세히 확인할 수 있다.

3.5.5. 행동 로그 조회

☰

Sentra

가능지원로그아웃

대시보드

이상 사용자 탐지

행동 로그 조회

PC 상태 조회

부서

Engineering

팀

MechanicalEngineering

사용자 ID

기간

이벤트 타입: All

loginemailhttpdevicefile

시간	부서	팀	사용자 ID	PC ID	이벤트	세부사항
2010-12-01 02:06:00	Engineering	MechanicalEngineering	HPH0075	PC-2417	login	Logout
2010-12-01 02:01:27	Engineering	MechanicalEngineering	HPH0075	PC-2417	email	Theo...ss@dtaa.com (+1)
2010-12-01 02:00:56	Engineering	MechanicalEngineering	HPH0075	PC-2417	email	HA...2@opt...net (+1)
2010-12-01 02:00:55	Engineering	MechanicalEngineering	HPH0075	PC-2417	email	DTT4@comcast.net
2010-12-01 02:00:00	Engineering	MechanicalEngineering	NJB0243	PC-3752	login	Logout
2010-12-01 02:00:00	Engineering	MechanicalEngineering	TEB0251	PC-6913	login	Logout
2010-12-01 02:00:00	Engineering	MechanicalEngineering	HAD0246	PC-0926	login	Logout
2010-12-01 01:59:00	Engineering	MechanicalEngineering	CSB0080	PC-0731	login	Logout
2010-12-01 01:59:00	Engineering	MechanicalEngineering	MHG0244	PC-9988	login	Logout
2010-12-01 01:59:00	Engineering	MechanicalEngineering	JCV0245	PC-9136	login	Logout

1-10 of 76992 <>

Copyright © 제로트러스트 클라우드 보안 연구센터 All Rights Reserved.

그림 9 행동 로그 필터링 결과 조회

☰

Sentra

가능지원로그아웃

대시보드

이상 사용자 탐지

행동 로그 조회

PC 상태 조회

부서

Engineering

팀

MechanicalEngineering

사용자 ID

기간

이벤트 타입: All

loginemailhttpdevicefile

시간	부서	팀	사용자 ID	PC ID	이벤트	세부사항
2010-12-01 02:01:27	Engineering	MechanicalEngineering	HPH0075	PC-2417	email	Theo...ss@dtaa.com (+1)
2010-12-01 02:00:56	Engineering	MechanicalEngineering	HPH0075	PC-2417	email	
2010-12-01 02:00:55	Engineering	MechanicalEngineering	HPH0075	PC-2417	email	
2010-12-01 01:58:31	Engineering	MechanicalEngineering	HPH0075	PC-2417	http	
2010-12-01 01:58:19	Engineering	MechanicalEngineering	HPH0075	PC-2417	email	
2010-12-01 01:58:17	Engineering	MechanicalEngineering	NJB0243	PC-3752	http	
2010-12-01 01:58:16	Engineering	MechanicalEngineering	NJB0243	PC-3752	http	basecampHQ.com
2010-12-01 01:58:15	Engineering	MechanicalEngineering	NJB0243	PC-3752	http	fidelity.com
2010-12-01 01:58:15	Engineering	MechanicalEngineering	NJB0243	PC-3752	http	fidelity.com
2010-12-01 01:58:13	Engineering	MechanicalEngineering	HPH0075	PC-2417	http	qjc.com

1-10 of 72103 <>

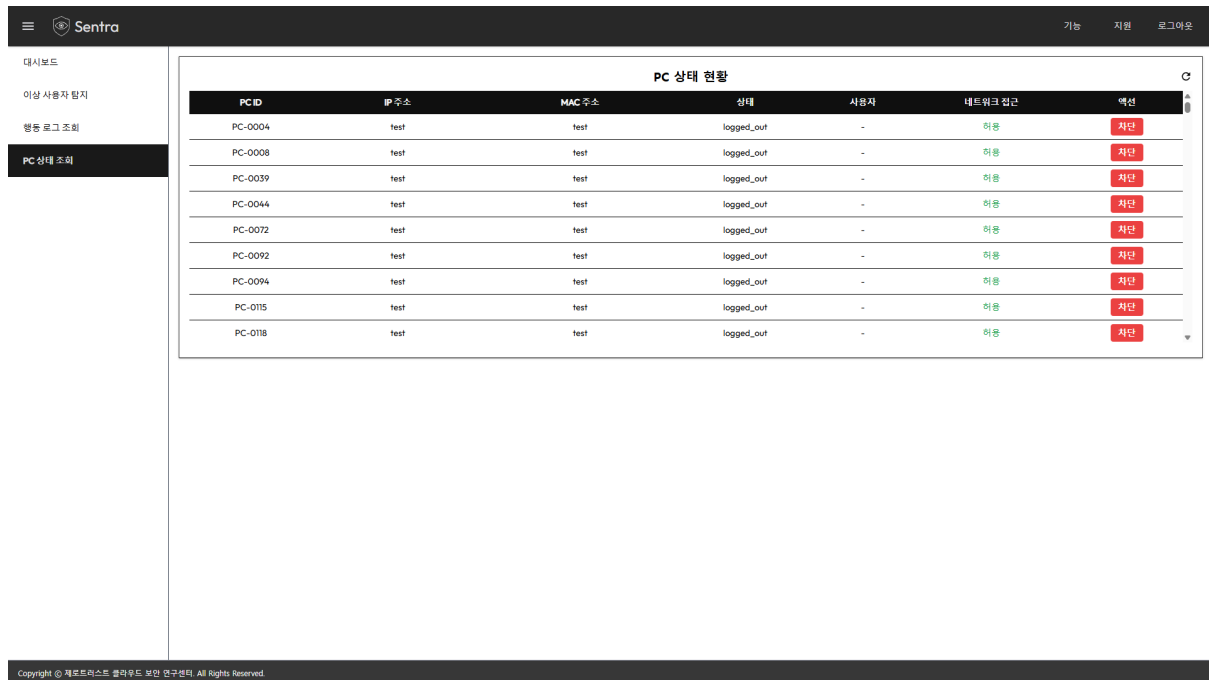
Copyright © 제로트러스트 클라우드 보안 연구센터 All Rights Reserved.

그림 10 email 타입 로그 세부사항 조회

행동 로그 조회 화면은 조직 내 사용자와 PC의 활동 로그를 조건에 따라 필터링하여 조회할 수 있는 기능을 제공한다. 관리자는 부서, 팀, 사용자 ID, 이벤트 타입, 기간의 조건을 설정하여 로그를 필터링하고, 결과를 테이블 형태로 확인한다. 테이블에서 조회되는 로그

중 데이터 타입이 http나 email인 경우, 로그의 세부 사항을 확인할 수 있다. Http 로그는 접속한 URL을 제공하며, email 로그는 발신자, 수신자, 참조(cc), 숨은 참조(bcc), 첨부 파일 개수(att), 이메일 크기 정보를 제공한다.

3.5.6. PC 상태 조회 및 관리



PC ID	IP 주소	MAC 주소	상태	사용자	네트워크 접근	액션
PC-0004	test	test	logged_out	-	허용	차단
PC-0008	test	test	logged_out	-	허용	차단
PC-0039	test	test	logged_out	-	허용	차단
PC-0044	test	test	logged_out	-	허용	차단
PC-0072	test	test	logged_out	-	허용	차단
PC-0092	test	test	logged_out	-	허용	차단
PC-0094	test	test	logged_out	-	허용	차단
PC-0115	test	test	logged_out	-	허용	차단
PC-0118	test	test	logged_out	-	허용	차단

그림 11 pc 상태 조회 및 네트워크 접근 제어

PC 상태 조회 화면은 조직 네트워크 상의 모든 PC의 현재 상태(로그온 상태, 사용자 ID, 네트워크 접근 허용/차단 여부)를 확인하고, 개별 PC에 대한 수동 네트워크 접근 제어 기능을 제공한다.

3.5.7. 이상 사용자 로그인 시 알림

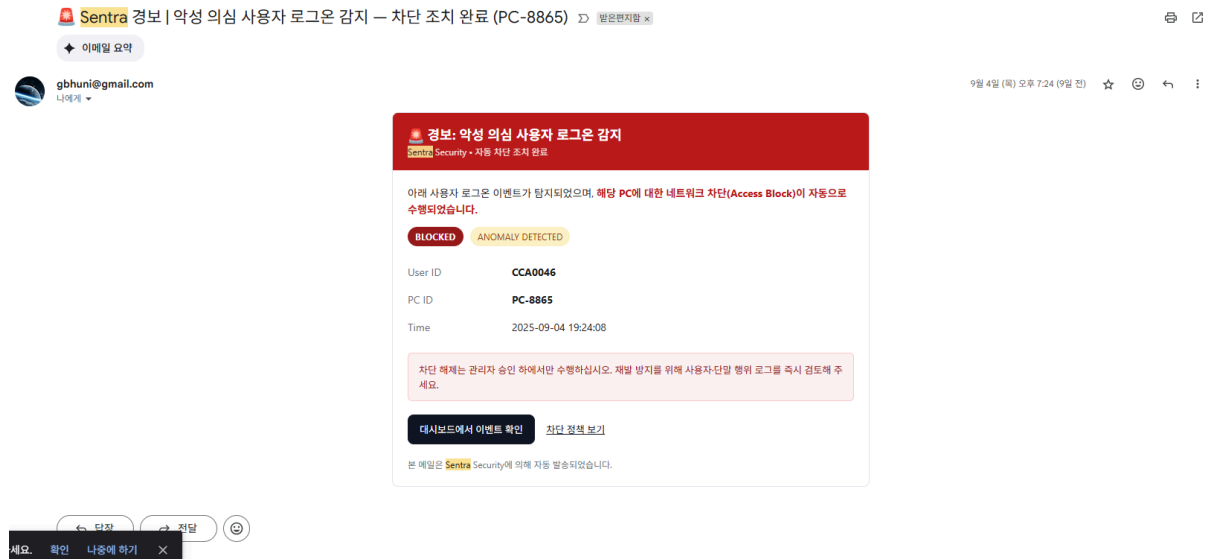


그림 12 이상 사용자 로그인 경고 이메일

이상 사용자 로그온이 감지되면 사전 정의된 정책에 따라 해당 PC의 네트워크 접근이 자동 차단되고, 동시에 **WebSocket**으로 관리자 화면에 실시간 알림이 표시된다. 관리자는 알림을 통해 이상 사용자의 로그온 활동을 즉시 확인하고, 필요 시 후속 조치를 수행한다. 동일 이벤트는 관리자에게 경고 이메일로도 자동 발송되며, 메일에는 사용자 ID, PC ID, 발생 시각 등 주요 필드와 사용자·단말 행위 로그 검토를 위한 대시보드 바로가기가 포함된다.

4. 연구 결과 분석 및 평가

4.1. 로그 수집 에이전트 성능 평가

본 연구에서 개발한 로그 수집 에이전트의 성능을 검증한다. 평가는 Docker Compose¹⁵로 구동되는 `Fluentd(log-collect-td-agent)`, `mitmproxy(log-collect-proxy-agent)`, `Postfix(log-collect-postfix)`, 메일 파싱(`log-collect-mail-agent`) 컨테이너를 중심으로 수행하며, 권한 패치를 담당하는 보조 컨테이너(`fix-audit-perm`, `fix-mail-perm`)도 관찰 범위에 포함한다. 공통 지표로는 CPU 사용률과 메모리 점유율을 사용한다. [표 8]은 본 성능 평가에 사용한 테스트 VM의 할당 자원을 나타낸다.

※ USB 장치 연결/해제와 파일 복사 시나리오는 다른 PC에서 측정했으나, 테스트 VM의 할당 자원을 동일하게 맞추어 비교하였다.

항목	값
CPU 코어 수	4 vCPU
메모리	8 GB RAM (8096 MB)

표 8 테스트 VM 할당 자원

4.1.1. 평상시

CONTAINER ID	NAME	CPU %	MEM USAGE / LIMIT	MEM %
70340999a444	log-collect-td-agent	0.44%	88.17MiB / 7.647GiB	1.13%
021f84319564	log-collect-mail-agent	0.00%	1.293MiB / 7.647GiB	0.02%
c3afb920d55	log-collect-proxy-agent	0.05%	98.25MiB / 7.647GiB	1.25%
b20b4de9bcb7	fix-mail-perm	0.00%	1012KiB / 7.647GiB	0.01%
0dcfe903f4fc	fix-audit-perm	0.01%	1.723MiB / 7.647GiB	0.02%
6c30b669fa4b	log-collect-postfix	0.00%	11.29MiB / 7.647GiB	0.14%
216c9baaba82	fastapi-app-db-1	0.00%	25.27MiB / 7.647GiB	0.32%

그림 13 평상 시 리소스 사용량

평상시 전체 컨테이너는 유휴 상태에 가까웠다. 컨테이너 합산 CPU 사용률은 약 1% 내외, 합산 메모리 점유율은 약 2.0–2.6% 범위에서 안정적으로 유지되었다. `Fluentd`의 `tail`-필터-전송 루틴은 대기 상태였고, `mitmproxy`·`Postfix`·메일 파싱 컨테이너는 입력 이벤트 부재 시 자원 사용이 미미했다.

¹⁵ 여러 컨테이너 기반 서비스를 정의하고 관리하기 위한 도커 오케스트레이션 도구

4.1.2. 일반/고강도 HTTP 트래픽

CONTAINER ID	NAME	CPU %	MEM USAGE / LIMIT	MEM %
70340999a444	log-collect-td-agent	0.22%	86.84MiB / 7.647GiB	1.11%
021f84319564	log-collect-mail-agent	0.00%	1.285MiB / 7.647GiB	0.02%
c3afbf920d55	log-collect-proxy-agent	4.63%	87.9MiB / 7.647GiB	1.12%
b20b4de9bcb7	fix-mail-perm	0.00%	1012KiB / 7.647GiB	0.01%
0dcfe903f4fc	fix-audit-perm	0.77%	1.719MiB / 7.647GiB	0.02%
6c30b669fa4b	log-collect-postfix	0.00%	11.32MiB / 7.647GiB	0.14%
216c9baaba82	fastapi-app-db-1	0.03%	25.27MiB / 7.647GiB	0.32%

그림 14 일반 웹 탐색 시 리소스 사용량

일반적인 웹 브라우징 시에는 mitmproxy가 트래픽을 캡처하고 처리하며 CPU 사용률이 약 4.6%로 소폭 상승했다. 같은 시점 Fluentd는 수집된 로그를 포맷 변환 후 내부 버퍼로 묶어 전송하므로 CPU 사용률 변동폭이 작았고, 두 컨테이너의 메모리 점유율은 1.0–1.3% 범위에서 안정적이었다.

CONTAINER ID	NAME	CPU %	MEM USAGE / LIMIT	MEM %
70340999a444	log-collect-td-agent	1.67%	87.8MiB / 7.647GiB	1.12%
021f84319564	log-collect-mail-agent	0.00%	1.289MiB / 7.647GiB	0.02%
c3afbf920d55	log-collect-proxy-agent	10.09%	98.38MiB / 7.647GiB	1.26%
b20b4de9bcb7	fix-mail-perm	0.00%	1012KiB / 7.647GiB	0.01%
0dcfe903f4fc	fix-audit-perm	1.34%	1.723MiB / 7.647GiB	0.02%
6c30b669fa4b	log-collect-postfix	0.00%	14.71MiB / 7.647GiB	0.19%
216c9baaba82	fastapi-app-db-1	0.00%	25.27MiB / 7.647GiB	0.32%

그림 15 고강도 HTTP 트래픽 처리 시 리소스 사용량

대량 검색 등 트래픽이 집중될 때 mitmproxy의 CPU 사용률은 일시적으로 약 10%까지 상승했다. 그러나 Fluentd는 내부 버퍼링을 통해 추가 지연 없이 로그를 처리했으며, 메모리 점유율도 1.0–1.3%로 안정적이었다.

4.1.3. 메일 송신

CONTAINER ID	NAME	CPU %	MEM USAGE / LIMIT	MEM %
70340999a444	log-collect-td-agent	0.32%	86.79MiB / 7.647GiB	1.11%
021f84319564	log-collect-mail-agent	6.28%	1.285MiB / 7.647GiB	0.02%
c3afbf920d55	log-collect-proxy-agent	0.02%	81.3MiB / 7.647GiB	1.04%
b20b4de9bcb7	fix-mail-perm	0.15%	1012KiB / 7.647GiB	0.01%
0dcfe903f4fc	fix-audit-perm	0.01%	1.723MiB / 7.647GiB	0.02%
6c30b669fa4b	log-collect-postfix	3.57%	16.24MiB / 7.647GiB	0.21%
216c9baaba82	fastapi-app-db-1	0.00%	25.27MiB / 7.647GiB	0.32%

그림 16 메일 송신 시 리소스 사용량

메일 송신 시점에 Postfix는 큐 처리와 로그 기록으로 약 3–4%, 메일 파싱 컨테이너는 메일 디렉터리를 스캔하고 파싱하는 루프 초기에 약 6%의 CPU 사용률 피크를 보였다. 이때 메모리 점유율은 Postfix가 0.25% 내외, 메

파싱 컨테이너가 0.02% 수준으로 매우 작았다.

4.1.4. 웹메일 송신

CONTAINER ID	NAME	CPU %	MEM USAGE / LIMIT	MEM %
70340999a444	log-collect-td-agent	0.82%	86.93MiB / 7.647GiB	1.11%
021f84319564	log-collect-mail-agent	0.00%	1.277MiB / 7.647GiB	0.02%
c3afb920d55	log-collect-proxy-agent	2.29%	90.55MiB / 7.647GiB	1.16%
b20b4de9bcb7	fix-mail-perm	0.00%	1012KiB / 7.647GiB	0.01%
0dcfe903f4fc	fix-audit-perm	0.01%	1.723MiB / 7.647GiB	0.02%
6c30b669fa4b	log-collect-postfix	0.00%	14.71MiB / 7.647GiB	0.19%
216c9baaba82	fastapi-app-db-1	0.03%	25.27MiB / 7.647GiB	0.32%

그림 17 웹메일 트래픽 처리 시 리소스 사용량

웹메일 UI가 열린 상태에서 주기적 폴링과 리소스 로딩으로 발생한 트래픽을 처리하면서 mitmproxy의 CPU 사용률이 약 2~3%로 소폭 상승했다. 동일 구간에서 Fluentd의 CPU 사용률은 약 0.8% 내외로 낮게 유지되었고, 두 컨테이너의 메모리 점유율은 1.0~1.3% 범위에서 안정적이었다.

4.1.5. USB 장치 연결 및 해제

CONTAINER ID	NAME	CPU %	MEM USAGE / LIMIT	MEM %
ba83ca14446f	log-collect-td-agent	7.56%	132.5MiB / 7.663GiB	1.69%
0edbfd1cb2d8	buildx_buildkit_mbuilder0	0.01%	51.2MiB / 7.663GiB	0.65%
47c59f19aea3	log-collect-proxy-agent	0.00%	83.33MiB / 7.663GiB	1.06%
96d624214ea5	log-collect-mail-agent	0.00%	3.93MiB / 7.663GiB	0.05%
148c9f1afcd7	fix-mail-perm	0.00%	4.883MiB / 7.663GiB	0.06%
045bc899a342	fix-audit-perm	3.67%	1.863MiB / 7.663GiB	0.02%
029e67192625	log-collect-postfix	0.00%	23.3MiB / 7.663GiB	0.30%

그림 18 USB 장치 연결 시 리소스 사용량

USB 연결 시 Fluentd의 CPU 사용률이 audit 로그를 파싱하고 필터링 및 전송하는 구간에서 약 7-8%의 단기 스파이크를 보였다. 메모리 점유율은 1.6~1.8%로 안정적이었고, 보조 컨테이너인 fix-audit-perm은 권한 및 소유권 패치 시 때 CPU 사용률이 최대 약 4%까지 일시적으로 반응했다. USB 해제 시에도 유사한 패턴이 관찰되었다.

4.1.6. 파일 복사

CONTAINER ID	NAME	CPU %	MEM USAGE / LIMIT	MEM %
ba83ca14446f	log-collect-td-agent	7.49%	131.9MiB / 7.663GiB	1.68%
0edbfd1cb2d8	buildx_buildkit_mbuilder0	0.00%	51.14MiB / 7.663GiB	0.65%
47c59f19aea3	log-collect-proxy-agent	0.00%	83.33MiB / 7.663GiB	1.06%
96d624214ea5	log-collect-mail-agent	0.00%	3.93MiB / 7.663GiB	0.05%
148c9f1afcd7	fix-mail-perm	0.00%	4.883MiB / 7.663GiB	0.06%
045bc899a342	fix-audit-perm	3.01%	1.863MiB / 7.663GiB	0.02%
029e67192625	log-collect-postfix	0.00%	23.3MiB / 7.663GiB	0.30%

그림 19 파일 복사 시 리소스 사용량

USB로 파일 복사 시 auditd가 다량의 파일 이벤트를 발생시키고 Fluentd가 이를 집계·전송하면서 Fluentd의 CPU 사용률이 약 7–8%의 단기 스파이크를 보였으나, 메모리 점유율은 1.6–1.8% 수준으로 안정적이었다.

4.1.7. 종합

시나리오	주요 컨테이너	CPU 피크(%)	Memory (%)
유휴 상태	합산(전체)	≈1	2.0–2.6
일반 웹 탐색	proxy-agent	≈4.6	1.0–1.3
고강도 HTTP 트래픽	proxy-agent	≈10	1.0–1.3
메일 송신	Postfix / mail-agent	3–4 / ≈6	≈0.25 / ≈0.02
웹메일 송신	proxy-agent / td-agent	2–3 / ≈0.8	1.0–1.3
USB 연결/해제	td-agent	7–8	1.6–1.8
파일 복사	td-agent	7–8	1.6–1.8

표 9 시나리오별 CPU 사용률 및 메모리 점유율 요약

[표 9]는 시나리오별 주요 컨테이너의 CPU 피크와 메모리 점유율 범위를 요약한 표다. 관측된 CPU 사용률 상승은 기능 수행에 따른 단기 반응 수준에 그쳤고, 메모리 점유율은 전 구간에서 낮고 안정적이었다. 이는 해당 PC 에이전트가 일반 사무용 PC 또는 가상머신 환경에서 체감할 만한 성능 저하 없이 로그 수집 및 전송 파이프라인을 안정적으로 유지할 수 있음을 입증한다. 향후 컨테이너별 자원 한도를 보수적으로 설정하면 다양한 운영 환경에서도 동일한 수준의 경량 동작을 보장할 수 있을 것이다.

4.2. 이상 탐지 모듈 평가

이상 탐지 모델의 성능을 평가하기 위해 전체 데이터 세트를 임의로 8 대 2로 분할해 학습 및 검증 세트를 구성했다. [표 10]은 학습 세트와 검증 세트의 클래스 별 분포를 나타낸다.

	0	1	2	3
--	---	---	---	---

Training	53,479	42	203	8
Test	13,371	10	51	2

표 10 학습 세트 및 평가 세트 데이터의 클래스 별 데이터 개수 분포

다음으로, 학습 세트의 데이터 불균형 문제를 해결하기 위해 이상 라벨링 데이터에 대한 증강을 수행한다. [표 11]은 이상 데이터에 대한 SMOTE 증강이 완료된 학습 세트의 클래스 별 개수 분포를 나타낸다.

	0	1	2	3
Training (증강 후)	53,479	500	500	500

표 11 증강이 완료된 학습 세트의 클래스 별 데이터 개수 분포

다음으로, 증강이 완료된 학습 세트를 사용해 Random forest 및 XGBoost 모델 학습을 수행한다.

4.2.1. Random Forest 모델 학습 및 성능 평가

Random Forest를 사용한 모델 학습 시 사용한 하이퍼파라미터 정보는 다음과 같다. 400개의 결정 트리를 사용하였으며, 각 트리의 최대 깊이는 16으로 제한했다. 리프 노드의 최소 샘플 수는 2로 설정하였고, 각 분할에서 고려되는 특징은 전체 특징 수 n 중 n^2 개로 제한했다. 각 클래스의 가중치는 균형 조정 방식으로 적용하였다.

학습이 완료된 Random Forest 모델에 대해, 학습 이전에 분리한 테스트 세트를 사용해 성능 평가를 수행했다. [표 12]는 Random Forest 모델의 예측 결과의 혼동 행렬¹⁶을 나타내며, [표 13]은 동일한 모델의 예측 결과의 정밀도(Precision)¹⁷, 재현율(Recall)¹⁸, F1-score¹⁹, support²⁰를 나타낸다.

	pred_0	pred_1	pred_2	pred_3
--	--------	--------	--------	--------

¹⁶ 분류 모델의 예측 결과를 실제 값과 비교하여 TP, FP, TN, FN으로 정리한 행렬

¹⁷ 모델이 양성으로 예측한 것 중 실제로 양성인 비율을 나타내는 성능 지표

¹⁸ 실제 양성 중에서 모델이 올바르게 양성으로 예측한 비율을 나타내는 성능 지표

¹⁹ 정밀도와 재현율의 조화평균으로 계산되는 성능 지표.

²⁰ 데이터셋에서 각 클래스가 포함하고 있는 실제 샘플의 개수를 나타내는 지표

true_0	13347	0	24	0
true_1	0	10	0	0
true_2	11	0	40	0
true_3	0	0	0	2

표 12 Random forest 모델 예측 결과 혼동 행렬

class	precision	recall	f1-score	support
0	0.9992	0.9982	0.9987	13371
1	1.000	1.000	1.000	10
2	0.6250	0.7843	0.6957	51
3	1.000	1.000	1.000	2.000

표 13 Random forest 모델 예측 결과 정밀도, 재현율, 조화 평균

4.2.2. XGBoost 모델 학습 및 성능 평가

XGBoost을 사용한 모델 학습 시 사용한 하이퍼파라미터 정보는 다음과 같다. 모델은 결정 트리 기반의 부스팅 트리 600개를 사용했으며, 트리 최대 깊이는 6으로 제한했다. 학습률은 0.05로 설정해 개별 트리의 기여도를 완만히 조정했고, 표본 하위 추출과 특징 하위 추출을 각각 0.8로 적용하여 과적합을 억제했다.

학습이 완료된 모델에 대해, 학습 이전에 분리한 테스트 세트를 사용해 성능 평가를 수행했다. [표 14]는 XGBoost 모델의 예측 결과의 혼동 행렬을 나타내며, [표 15]는 동일한 모델의 예측 결과의 정밀도(Precision), 재현율(Recall), F1-score, support를 나타낸다.

	pred_0	pred_1	pred_2	pred_3
true_0	13370	0	1	0
true_1	0	10	0	0
true_2	0	0	51	0

true_3	0	0	0	2
--------	---	---	---	---

표 14 XGBoost 모델 예측 결과 혼동 행렬

class	precision	recall	f1-score	support
0	1.0000	0.9999	1.0000	13371
1	1.000	1.000	1.000	10
2	0.9808	1.00	0.9903	51
3	1.000	1.000	1.000	2.000

표 15 XGBoost 모델 예측 결과 정밀도, 재현율, 조화 평균

4.2.3. Random Forest 와 XGBoost 모델 성능 비교

Random Forest와 XGBoost 모두 이상 사용자 탐지에 있어 의미 있는 성능을 보였다. [표 13]의 결과에서 Random Forest는 정상 사용자(class 0)에 대해 매우 높은 정밀도(0.9992)와 재현율(0.9982)을 기록했으며, 일부 이상 사용자(class 1, class 3) 또한 완벽하게 식별하였다. 그러나 class 2의 경우 정밀도 0.6250, 재현율 0.7843으로 상대적으로 낮은 성능을 보여 특정 유형의 이상 행위를 탐지하는 데 한계가 있음을 확인할 수 있었다. 이는 데이터 불균형과 소수 클래스에서의 일반화 성능 저하가 작용한 결과로 추정된다.

반면 [표 15]의 XGBoost 결과는 전반적으로 더 우수한 성능을 보였다. 특히 Random Forest에서 취약했던 Class 2에 대해 XGBoost 는 정밀도 0.9808, 재현율 1.000, F1-score 0.9903으로 현저히 개선된 탐지 성능을 보였다. 또한 나머지 class에 대해서도 완벽에 가까운 성능을 기록했다. 이러한 결과는 Gradient Boosting 기반의 반복적 최적화 과정이 복잡한 데이터 패턴을 효과적으로 포착하고, 데이터 불균형 상황에서도 강인한 성능을 발휘했음을 보여준다. 따라서 본 연구에서는 XGBoost를 이상 탐지 파이프라인의 최종 모델로 사용했다.

4.2.4. 이상 탐지 파이프라인 평가

이상 탐지 모듈은 실험 결과 높은 정확도와 재현율(recall)을 달성하여, 악성 시나리오 행위자를 효과적으로 식별할 수 있음을 보여주었다. 특히 이상 탐지 시스템에서 오탐(false positive)의 최소화는 실효성 확보에 중요한 요소인데, 본 연구의 모듈은 높은 precision을

유지함으로써 오탐 비율을 줄였다.

반면, 개선이 필요한 부분도 존재한다. 첫째, 현재 탐지는 주(**week**) 단위로 이루어지기 때문에 일(**day**)이나 세션(**session**) 단위에 비해 위협에 대응 속도가 상대적으로 떨어진다. 둘째, 학습 모델의 성능은 실험적으로 **XGBoost** 가 우수했으나, 실제 운영 환경에서는 데이터 증가와 모델 복잡도에 따른 자원 소모와 지연 문제가 추가적으로 발생할 수 있다. 따라서 향후 연구에서는 실시간성 및 경량화 문제를 해결할 필요성이 있다.

4.3. 네트워크 제어 모듈 평가

본 연구에서 제안한 네트워크 제어 모듈은 네트워크 환경 정보를 수집하여 실시간으로 상태를 파악할 수 있다는 점에서 강점을 가진다. 조직 보안 관리자는 이를 통해 조직 네트워크 내 각 **PC**의 접근 권한 정보를 실시간으로 감지할 수 있으며, 이상 탐지 모듈의 오탐으로 인해 차단된 **PC**를 수동으로 허용할 수 있다. 또한 라우터 제어 과정에서 **SSH** 기반의 안전한 연결을 활용함으로써, 명령 전송 시 무결성을 보장할 수 있다. 이는 단순한 모니터링을 넘어, 실제 네트워크 접근 제어 정책을 안정적으로 적용할 수 있는 기반을 제공한다.

그러나, 해당 모듈에는 몇 가지 제약점도 존재한다. 실제 운영 환경에서는 라우터와 메인 프레임 간의 물리적, 논리적 연결을 위한 추가 경로가 필요하므로, 이로 인한 네트워크 구성 복잡성이 증가할 수 있다. 또한 라우터 정보를 스니핑 공격자가 조작하거나, 의도적으로 라우터의 **ARP** 테이블을 조작할 수 있는 가능성도 존재한다. 따라서, 향후 시스템을 실제 운영 환경에 적용하는 과정에선, 이러한 공격을 방지하기 위한 추가적인 기법들이 고려되어야 한다.

4.4. 모니터링 모듈 평가

모니터링 모듈은 중앙 데이터베이스와 연동되어 조직 내 보안 상태를 직관적으로 파악할 수 있도록 설계되었다는 점에서 강점을 가진다. 관리자는 보안 위협 통계 추세 조회, 사용자 행동 로그 조회, 악성 사용자 스캔, 네트워크 상태 확인 및 제어 등의 기능을 웹을 통해 쉽게 사용할 수 있다. 또한, **Websocket** 알림과 이메일 경고를 통해 실시간 이벤트를 즉시 전달하는 기능을 통해 관리자가 보안 위협에 즉각 대응할 수 있도록 지원한다.

5. 결론 및 향후 연구 방향

본 연구에서는 조직 내 사용자 행동 로그를 실시간으로 수집 및 분석하고, 이상 사용자를 식별하여 해당 사용자의 PC를 자동으로 네트워크에서 격리하는 보안 시스템을 구현하였다. 전체 시스템은 로그 수집 및 저장 모듈, 이상 탐지 모듈, 네트워크 제어 모듈, 모니터링 모듈로 구성되었으며, 중앙 데이터베이스와 연동되어 통합적으로 운영되도록 설계되었다.

또한 시스템의 실효성을 검증하기 위해 로그 수집 에이전트의 자원 점유율과 이상 탐지 모듈의 성능에 대한 정량적인 평가를 수행했다. 평가 결과, 로그 수집 에이전트는 경량 환경에서도 안정적인 동작을 보였으며, 이상 탐지 모듈은 XGBoost를 적용해 높은 정확도와 재현율을 달성하였다.

향후 연구에서는 이상 사용자 탐지의 신속성 강화를 핵심 목표로 할 것이다. 첫째, 현재 주 단위로 수행되는 탐지 과정을 세션 단위로 세분화하여 보다 짧은 시간 간격에서 이상 행위를 식별함으로써 실시간 대응 능력을 높일 것이다. 둘째, 새로운 위협 패턴이 등장하는 상황에 대응하기 위해 비지도 학습 기법을 도입하여 레이블이 부족한 데이터에서도 이상 행위를 탐지할 있도록 개선할 것이다. 셋째, 실제 운영 환경에서 성능 부담을 줄이기 위해 성능을 유지하며 모델을 경량화해서, 대규모 로그 데이터가 발생하더라도 안정적인 탐지가 가능하도록 개선할 것이다.

6. 구성원별 역할

팀원	역할
공동	<ul style="list-style-type: none"> - 착수, 중간, 최종 보고서 작성 - 전체 시스템 설계
이광훈	<p>개발 환경 구성</p> <ul style="list-style-type: none"> - Git 레포지터리 생성 및 개발 파이프라인 구축 - 개발 가이드라인 작성 <p>설계</p> <ul style="list-style-type: none"> - 데이터베이스 설계 및 스키마 구현 <p>Backend</p> <ul style="list-style-type: none"> - 백엔드 초기 환경 설정 <p>Frontend</p> <ul style="list-style-type: none"> - 프론트엔드 초기 환경 설정 - 회원 가입, 로그인 기능 개발 - 대시 보드 컴포넌트 개발 - 네트워크 상태 확인 및 제어 페이지 개발 <p>테스트 베드 제작</p> <ul style="list-style-type: none"> - OpenWRT 및 가상 머신을 이용한 가상의 조직 네트워크 구성 <p>네트워크 제어 모듈</p> <ul style="list-style-type: none"> - 가상 라우터 네트워크 모니터링 및 보고 데몬 개발 - 가상 라우터 방화벽 제어를 위한 ssh 자동화 기능 개발 - 악성 사용자 로그인 탐지 및 자동 차단 기능 구현
조유진	<p>로그 수집 모듈</p> <ul style="list-style-type: none"> - 엔드포인트 로그 수집 에이전트 개발 - Docker 기반 배포 자동화 <p>Backend</p> <ul style="list-style-type: none"> - 로그 수집/조회 API 개발 <p>Frontend</p> <ul style="list-style-type: none"> - 행동 로그 조회 페이지 개발 - 대시보드 컴포넌트 개발
신해진	문서 작업

	<ul style="list-style-type: none"> - 포스터 제작 <p>이상 탐지 모듈</p> <ul style="list-style-type: none"> - Random Forest, XGBoost, Mg-uabd 모델 학습 및 성능 테스트 - 백엔드 내에 모듈 병합 및 이상 탐지 파이프라인 구축 <ul style="list-style-type: none"> - 데이터 베이스 조회 - 모델에 적합한 형태로 전처리 - 탐지 결과 데이터 베이스 저장 기능 구현 <p>Frontend</p> <ul style="list-style-type: none"> - 악성 사용자 탐지 및 조회 페이지 구현 - 대시보드 컴포넌트 개발
--	---

7. 개발 일정

5월		6월					7월					8월					9월	
4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2
로그 수집 모듈 개발																		
이상 탐지 모듈 개발																		
네트워크 제어 모듈 개발																		
							중간 보고서											
									API 및 DB 설계									
											백엔드 API 개발							
												프론트엔드 개발						
																디버깅 및 테스트		
																	최종 보고서	

8. 멘토 의견 반영

8.1. OpenWRT 기반 가상 라우터를 통한 실제 네트워크 제어의 현실적 적용 가능성에 대한 검증 필요

[4.3. 네트워크 제어 모듈 평가] 섹션에서 기술한 바와 같이, 네트워크 제어 모듈은 SSH 기반의 보안 채널을 통해 가상 라우터를 원격으로 제어하고, 특정 PC의 네트워크 접근을 차단하는 기능을 구현하는데 성공해 기술의 현실적인 적용 가능성을 입증하였다. 또한, 관리자가 실시간으로 네트워크 상태를 감지하고, 이상 탐지 모듈의 오탐 발생 시 수동으로 차단을 해제할 수 있는 관리 기능을 제공하여 운영상의 안정성을 확보하였다. 그러나 스니핑 공격에 대한 취약점이나 네트워크 구성 복잡성 증가와 같은 제약점 역시 존재하기에, 이는 향후 실제 운영 환경 적용 시엔 보완되어야 한다.

8.2. 전체 시스템 통합과 성능 평가에 대한 구체적인 결과 제시

시스템의 핵심 모듈인 로그 수집 및 저장, 이상 탐지, 네트워크 제어 모듈을 통합 관리하는 모니터링 모듈을 개발하였다. 관리자는 해당 인터페이스를 통해 시스템의 모든 핵심 기능을 통합적으로 제어하고 관찰할 수 있으며, 이를 통해 각 모듈이 유기적으로 연동되는 전체 시스템의 동작 흐름을 완성하였다.

시스템의 통합적인 성능을 평가하기 위해서 핵심 모듈에 대한 정량적 성능 평가를 수행하였다. [4.1. 로그 수집 에이전트 성능 평가] 섹션에서는 로그 수집 에이전트가 유휴 상태, 파일 입출력, 네트워크 통신 등 다양한 탐지 유형에서 에이전트의 리소스 점유율을 측정한 결과, 매우 낮은 수준을 유지하여 사용자 PC의 성능에 거의 영향을 주지 않음을 확인하였다. [4.2. 이상 탐지 모듈 평가] 섹션에서는 이상 탐지 모델의 최종 성능을 사용자 단위의 Confusion Matrix와 Recall, Precision, F1 score 등 정량적인 지표로 제시하였다.

8.3. 실제 환경에서의 탐지 정확도와 오탐율에 대한 정량적 평가 필요

모델의 실제 적용 가능성을 검증하기 위해 본 연구에서 직접 구축한 가상 네트워크 테스트 베드의 로그 수집 시스템과 이상 탐지 모듈을 연동하여 실시간 테스트를 진행하였다. 테스트 베드에서 실제로 생성된 사용자 활동 로그를 이상 탐지 모델에 실시간으로 전송하여 분석한 결과, 모델이 학습용 데이터뿐만 아니라 실제 로그 환경에서도 효과적으로 동작할 수 있음을 확인하였다.

9. 참고 문헌

- [1] 보안뉴스. “내부자 보안사고 급증, 총계 6,803건에 연간 평균 총비용 1,540만 달러,” 보안뉴스, 14 June 2022. Available: <https://www.boannews.com/media/view.asp?idx=107451>. [Accessed: Aug. 26, 2025].
- [2] 선애 김, “내부자 보안 사고 기업 29%, 피해 복구액 14억 이상,” *DataNet*, Nov. 27, 2024. [Online]. Available: <https://www.datanet.co.kr/news/articleView.html?idxno=198095>. [Accessed: Aug. 26, 2025].
- [3] 한국인터넷진흥원, “제로 트러스트 가이드라인 2.0,” 한국인터넷진흥원, 3 Dec 2024. [Online]. Available: <https://www.kisa.or.kr/2060204/form?postSeq=18&page=1#fnPostAttachDownload>. [Accessed: 26-Aug-2025].
- [4] NIST, “Network Access Control (NAC),” Computer Security Resource Center Glossary, NIST SP 800-41 Rev. 1. [온라인]. Available: https://csrc.nist.gov/glossary/term/network_access_control. [Accessed: 26-Aug-2025].
- [5] Software Engineering Institute, “Insider Threat Test Dataset,” Carnegie Mellon University, Pittsburgh, PA, Nov. 28, 2016. DOI: 10.1184/R1/12841247.v1.