- 2025 전기 졸업과제 중간보고서 -

사용자 행동 기반 동적 네트워크 접근 제어 시스템 연구

부산대학교 정보컴퓨터공학부

지도교수 최윤호

팀명 CtrlAltDefend

팀원 201924524 이광훈

202255562 신해진

202155609 조유진

목차

1. 서론

최근 사이버 보안 위협은 외부 침입뿐만 아니라 내부 사용자에 의한 이상 행위로도 확산되고 있으며, 특히 조직 내부의 사용자 행동을 기반으로 한 보안 위협 탐지는 점점 더 중요해지고 있다. 이에 따라 본 과제에서는 사용자 행동 기반 동적 네트워크 접근 제어 시스템을 제안하고, 구현하고자 한다.

해당 시스템은 사용자 단말기에서 발생하는 다양한 행동 로그(로그온/오프, 메일 발신, 웹 브라우징, 디바이스 사용 등)을 실시간으로 수집하고, 이를 기반으로 이상 행위를 탐지한 뒤, 이상 사용자로 식별된 경우 해당 사용자가 로그온한 PC의 네트워크 접근을 자동 또는 수동으로 제어하는 구조로 구성된다. 이를 통해 내부자의 악의적 행위 또는 실수로 인한 보안 사고를 사전에 차단하고, 관리자는 전체 네트워크 상태를 실시간으로 시각화된 대시보드에서 직관적으로 모니터링할 수 있도록 지원한다.

본 보고서에서는 과제 초기 설계를 바탕으로 수정된 요구사항 분석, 시스템 구조의 세부 설계 변경 사항, 각 구성원의 진척도, 핵심 시스템의 구현 현황과 중간 결과를 상세히 다룬다. 또한, 향후 보완이 필요한 요소 및 남은 개발 계획에 대해서도 구체적으로 기술한다.

2. 요구조건 및 제약사항 분석에 대한 수정사항

1) 기존 요구조건 및 수정사항

3-1. 기능적 요구사항

기능		설명
로그 수집 시스템	로그 수집	로그 수집 에이전트는 사용자 PC에서 발생하는 로그온/로그오프, 메일 발신, 웹 브라우징, 파일 복사, 디바이스 연결 등 이벤트 로그를 실시간으로 수집한다.
	로그 전송	**로그 수집 에이전트는 사용자 행동 이벤트 발생 시마다 이벤트 로그를 서버로 실시간으로 전송한다.**

	로그 저장	로그 수집 서버는 로그 수집 에이전트로부터 전송받은 이벤트 로그들을 데이터베이스에 저장한다.
	PC 상태 갱신	**로그온 이벤트 로그 발생 시 데이터 베이스의 PC 상태 정보(log on 여부, log on 중인 사용자)를 갱신한다.**
	이상 사용자 로그인 식별	**시스템은 이상 사용자의 PC 로그온 이벤트를 감지하면 해당 PC에 대한 네트워크 접근 제어를 네트워크 접근 제어 시스템에 요청한다.**
이상 행위 탐지 시스템	**이상 사용자 탐지**	**시스템은 사전 학습된 머신러닝 모델을 사용해, 매주 각 사용자의 주 단위 행동 로그의 이상 여부를 판별한다.**
	이상 사용자 목록 저장	**시스템은 이상으로 판별된 사용자의 정보를 데이터베이스에 업데이트 한다.**
	분석 결과 저장	시스템은 탐지된 이상 행위의 탐지 시각, 탐지된 사용자 및 PC, 이상 확률, 로그 요약 및 원인 이벤트와 같은 정보를 데이터베이스에 저장한다.
네트워크 접근 제어 시스템	**이상 사용자 로그인 제어**	**시스템은 로그 수집 시스템 서버로부터 이상 사용자가 로그인한 PC의 접근 제어 요청을 받아 해당 PC의 네트워크 접근을 차단한다.**
	제어 결과 저장	시스템은 접근 차단의 결과를 데이터베이스에 저장한다.
모니터링 시스템	**관리자 회원 가입**	**관리자는 모니터링 시스템에 정보(아이디, 비밀번호, 이메일, 이름, 조직명)을 입력해 회원 가입 할 수 있다.**

	관리자 로그인	관리자는 아이디, 비밀번호를 입력해 로그인할 수 있다.
	관리자 로그아웃	관리자는 로그아웃할 수 있다.
	인증 토큰 발급	관리자의 로그인 세션은 인증 토큰의 유효기간 동안만 유지 되며, 유효 기간이 만료되면 자동으로 로그아웃 된다.
	PC 상태 조회	관리자는 전체 네트워크 구조와 네트워크 상에 존재하는 각 PC의 정보(PC ID), 상태(PC on/off, 네트워크 접근 제한/허용)를 확인할 수 있어야 한다.
	네트워크 토폴로지 시각화	네트워크 구조 및 각 PC의 정보는 그래프로 시각화 되어 관리자에게 표시 된다.
	PC 로그 조회	관리자는 네트워크 상에 존재하는 각 PC의 이벤트 로그를 PC별, 유형별, 기간별로 조회할 수 있다.

	실시간 접근 제어 알림	모니터링 시스템은 특정 PC에 대한 자동 접근 제어 발생 시 다음과 같은 방식으로 관리자에게 알림을 제공한다.
		• 알림 수단
		  ○ 실시간 팝업 알림
		  ○ 관리자 이메일 전송
		• 전달 정보
		  ○ 접근 제어 발생 시각
		  ○ 제어 대상 PC

	접근 제어 기록 확인	관리자는 특정 PC에 대한 접근 제어 기록(제어 시각, 제어 유형, 성공 여부)를 확인할 수 있다.

	접근 제한 PC 로그 조회	관리자는 접근 제한이 발생한 PC의 로그를 유형별(이메일, http, 디바이스, 파일 복사, log/off)로 조회할 수 있다.
	수동 접근 제어	관리자는 특정 PC에 대해 수동으로 접근 제어를 할 수 있다.

3-2. 비기능적 요구사항

항목	설명
가용성	시스템은 24시간, 365일 상시 운영될 수 있어야 한다.
성능	로그 수집 및 전송, NAC 조치 시 실시간성이 보장되어야 한다.
사용성	관리자 대시보드는 이상 행위 탐지, NAC 조치 내역, 실시간 팝업 알림 등을 직관적으로 표시해야 한다.
확장성	시스템은 에이전트 수 또는 로그량 증가 시에도 성능 저하 없이 처리 가능해야 한다.

2) 기존 제약사항 및 수정사항

제약 사항	대책
머신러닝 모델 구축에 필요한 데이터 세트를 구하기 어렵다.	공개 데이터 세트를 통해 CMU CERT 데이터 세트를 사용한다.
사용할 데이터 세트 내에 악성 사용자의 행동 샘플 수가 정상 행동 샘플에 비해 현저히 적어 데이터 불균형 문제가 발생할	오버 샘플링 기법을 사용해 학습 시 클래스 불균형을 보정한다.

수 있다.	
실제 사용자 환경을 구성하기 위해 다수의 PC가 필요하다.	VMware와 같은 호스트 가상 머신을 사용해 네트워크 토폴로지를 구성한다.
네트워크 접근 제어를 위한 실제 라우터 또는 네트워크 장비 제어가 제한된다.	오픈 소스 라우터(OpenWrt)를 활용해 제어 가능한 테스트 환경을 구성한다.
실시간 데이터 수집 및 처리 성능 확보가 어렵다.	로그 전송 시 카프카와 같은 비동기 방식 전송 기술을 사용한다.
탐지된 이상행위가 실제 악성 행위가 아닐 경우, 오탐(false positive) 가능성이 존재한다.	모델 개발 과정에서 적절한 threshold 값을 설정하고, 관리자가 수동으로 접근 제어할 수 있는 기능을 추가한다.
서버에 수집되는 로그의 저장용량이 빠르게 증가할 수 있다.	일정 시간이 지난 로그 파일은 삭제한다.
내부 네트워크 구조의 복잡성으로 인해 접근 제어 대상의 식별이 어려울 수 있다.	접근 제어 대상 엔드포인트들을 ip - mac 주소 쌍으로 관리한다.
사용자 행동 로그 포맷이 서로 달라 전처리에 어려움이 발생한다.	자동 전처리 모듈(Log parser)를 개발한다.
시스템 테스트 시 실제 이상 행위를 모두 재현하는 데 어려움이 있다.	재현할 수 있는 이상 행위 시나리오 몇 가지를 구성하여 테스트를 진행한다.

3. 설계 상세화 및 변경 내역

1) 전체 설계 구조

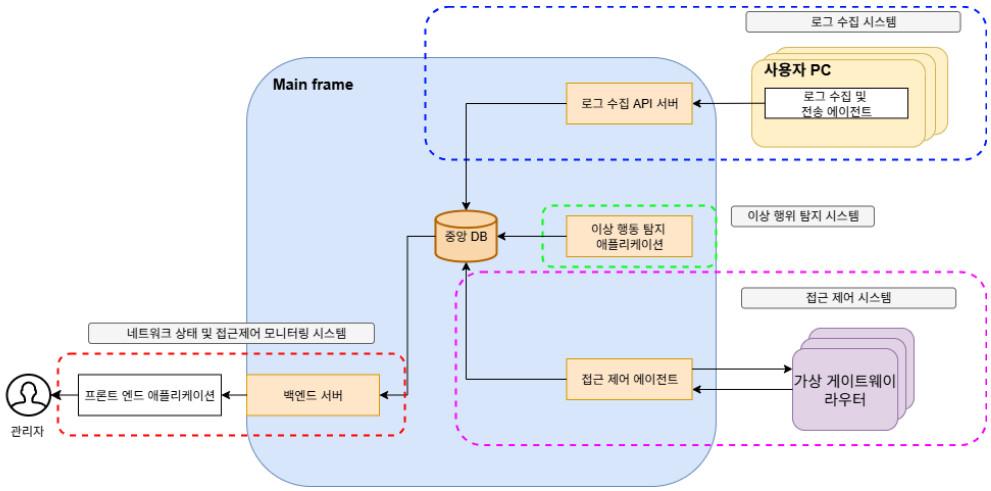


그림 1. 사용자 행동 기반 동적 네트워크 접근 제어 설계

그림 1은 사용자 행동 기반 동적 네트워크 접근 제어 시스템의 전체 구조를 표현한 것이다. 초기 설계와 비교해 다음과 같은 개선 사항이 반영되었다.

1. 데이터베이스 구조 통합

   기존 설계에서는 각 모듈(로그 수집, 이상 행위 탐지, 접근 제어 등)이 서로 독립된 데이터베이스 인스턴스를 사용하고 있었으나, 설계 복잡도와 초기 구현 효율성을 고려하여 중앙 단일 DB 구조로 통합하였다. 이는 개발 및 테스트 단계에서의 데이터 연계 편의성과 유지 보수성을 높이기 위한 목적이다.

2. 시스템 간 API 명세 통합

   로그 수집, 이상 행위 탐지, 접근 제어 등 각 구성 요소 간의 연동을 위해 RESTful API 인터페이스 표준화를 적용하였다. 이를 통해 시스템 간 종속성을 줄이고, 모듈 교체 및 확장 시 독립적인 운영이 가능하도록 설계하였다.

3. 관리자 개입 경로 확보

   네트워크 상 이상 행위를 트리 기반 이상 탐지 모델을 통해 실시간으로 분석할 뿐만 아니라, 외부 관리자 시스템(프론트엔드

애플리케이션)을 통해 수동으로 접근 제어를 요청할 수 있는 구조를 추가함으로써, 자동화 탐지 시스템의 한계를 보완하였다.

2) 데이터 베이스 설계

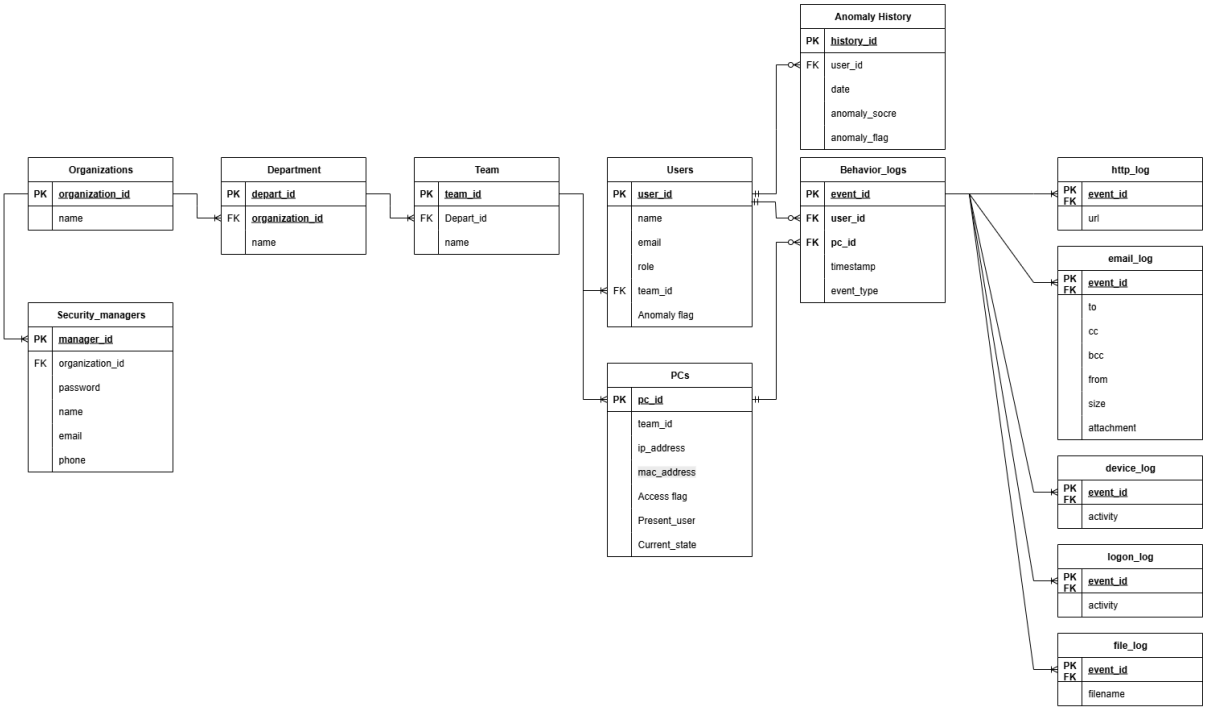


그림 2. 데이터베이스 스키마

그림 2는 초기 데이터 베이스 설계를 엔터티 관계 다이어그램(ERD)로 나타낸 것이다. 사용자의 조직 소속, 이상행위 이력, 행위 로그 및 개별 이벤트 정보까지 연계하여 통합적인 보안 분석이 가능하도록 구성되어 있다. 다음은 각 테이블의 목적에 대한 설명이다.

1. 조직 및 사용자 계층 구조

   Organizations → Department → Team 구조를 통해 조직의 다단계 계층을 반영하였다. 각 사용자는 Users 테이블에 저장되며, team_id를 통해 상위 조직 단위까지 추적할 수 있다. 또한 Security_managers 테이블은 조직별 관리자 정보를 저장하고, 인증 및 접근 제어 권한 부여에 활용된다.

2. 사용자 및 단말기 상태 정보 관리

   Users 테이블은 사용자 기본 정보 외에 Anomaly_flag를 통해 현재 이상행위 여부를 기록한다. PCs 테이블은 각 PC의 네트워크 정보(ip_address, mac_address), 현재 사용자(Present_user), 접근 가능 상태(Access_flag) 등을 저장하며, 이를 통해 실시간 접근 제어가 가능하도록 설계하였다.

3. 이상행위 탐지 이력 (Anomaly_History)

Anomaly_History 테이블은 트리 기반 모델을 통한 이상 탐지 결과를 기록한다.

anomaly_score와 anomaly_flag를 통해 정량적 스코어링 기반의 이력 관리가 가능하며,

timestamp와 함께 저장되어 사용자별 이상 발생 주기 및 패턴 분석에도 활용된다.

4. 행위 로그 기록 및 세부 이벤트 연동

Behavior_logs는 사용자와 PC 간의 행위 로그를 저장하며, event_type에 따라 세부 로그 테이블과 연동된다. http_log, email_log, device_log, login_log, file_log는 각각의 이벤트 유형에 대한 세부 정보를 별도로 저장하는 구조이며,

이는 로그 데이터의 정규화와 확장성을 동시에 확보하기 위함이다.

3) 기능별 시나리오

우리의 행동 기반 동적 네트워크 접근 제어 설계는 서로 다른 역할을 수행하는 여러 시스템 간의 상호작용으로 동작한다. 이 절에서는 핵심 기능별로 시스템이 어떻게 동작하는지에 대한 시나리오를 제시한다.

1-1. 악성 사용자 여부 판별 시나리오

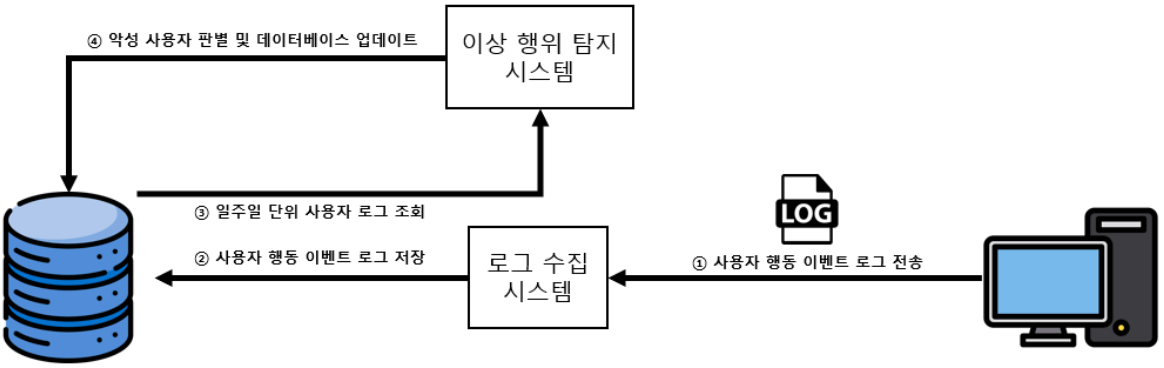


그림 3. 악성 사용자 여부 판별 시나리오

그림 3은 내부 사용자의 행위 로그를 기반으로 악성 여부를 판단하고, 이를 데이터베이스에 기록하는 전체 흐름을 설명한다. 주요 단계는 다음과 같다.

1. 사용자 행동 이벤트 로그 전송

   사용자가 시스템에 로그인하거나, 특정 행동을 수행할 경우, 해당 이벤트 로그가 로그 수집 시스템으로 전송된다.

2. 사용자 행동 이벤트 로그 저장

   로그 수집 시스템은 전송받은 로그를 실시간으로 저장하며, 사용자 식별자와 함께 행동 이벤트를 기록한다.

3. 주 단위 사용자 로그 조회

   일정 시간 동안 수집된 로그 데이터를 기준으로, 단위 사용자(세션 단위 또는 일정 기간)의 로그 전체를 조회한다.

4. 악성 사용자 판별 및 데이터베이스 업데이트

   이상 행위 탐지 시스템은 조회된 사용자 로그를 기반으로 기존에 학습된 트리 기반 모델을 통해 이상 여부를 판별한다. 악성 사용자로 판별될 경우, 해당 결과를 데이터베이스에 기록하고, 이후 대응 절차에 활용될 수 있도록 업데이트한다.

1-2. 악성 사용자 로그온 탐지 및 PC에 대한 네트워크 접근 차단 시나리오

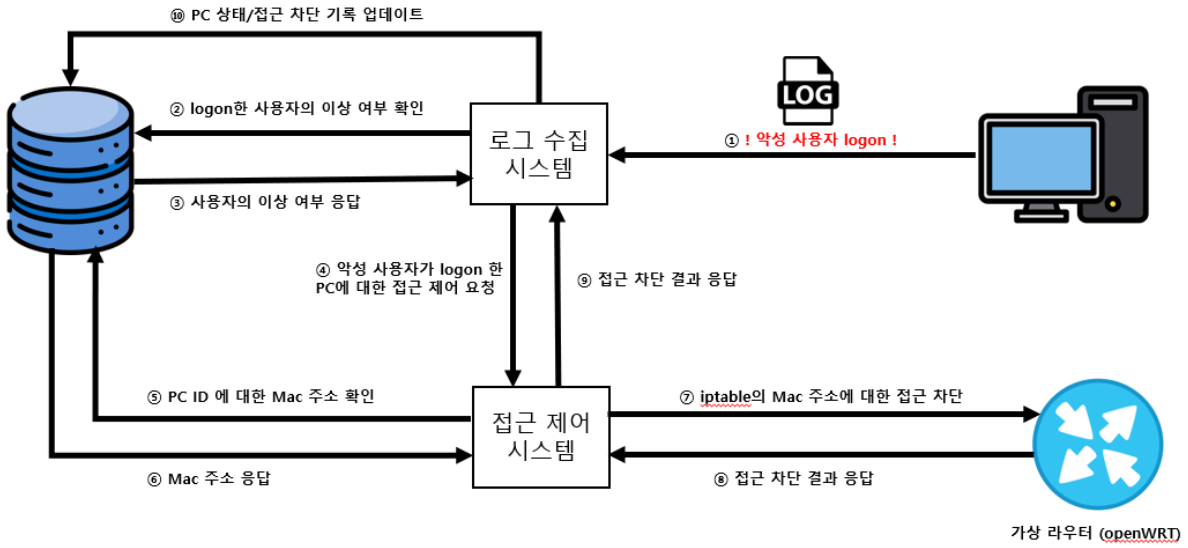


그림 4. 악성 사용자 탐지 및 접근 제어 과정

그림 4는 이상행위 탐지 시스템을 통해 악성 사용자를 탐지한 후, 해당 사용자가 로그온한 PC에 대해 네트워크 접근을 차단하는 전체 흐름을 설명한다. 주요 단계는 다음과 같다.

1. 악성 사용자 logon 이벤트 수신

   사용자가 시스템에 로그인하면, 해당 이벤트가 로그 수집 시스템에 전달된다.

2. 로그온 사용자의 이상 여부 확인

   로그 수집 시스템은 로그인한 사용자가 악성 사용자 목록에 포함되어 있는지를 확인하기 위해 데이터베이스 또는 탐지 시스템에 질의한다.

3. 사용자의 이상 여부 응답

   데이터베이스는 해당 사용자의 이상 여부에 대한 판단 결과를 로그 수집 시스템에 응답한다.

4. 악성 사용자에 대한 접근 제어 요청

   악성 사용자로 판별된 경우, 로그 수집 시스템은 해당 사용자가 로그온한 PC에 대해 네트워크 접근 차단 요청을 접근 제어 시스템에 전달한다.

5. PC ID에 대한 Mac 주소 확인

   접근 제어 시스템은 네트워크 차단을 위해 해당 PC의 고유 식별자인 Mac 주소를 조회하기 위해 데이터베이스에 질의한다.

6. Mac 주소 응답

   데이터베이스는 요청된 PC ID에 대응하는 Mac 주소 정보를 접근 제어 시스템에 응답한다.

7. iptables를 통한 접근 차단 실행

   접근 제어 시스템은 openWRT 기반 라우터에 명령을 전달하여 해당 Mac 주소에 대한 네트워크 접근을 iptables를 통해 차단한다.

8. 접근 차단 결과 응답 수신

   라우터는 차단 명령에 대한 실행 결과를 접근 제어 시스템에 응답한다.

9. 차단 결과 전달

접근 제어 시스템은 차단 결과를 로그 수집 시스템에 전달한다.

10. PC 상태 및 접근 차단 기록 업데이트

로그 수집 시스템은 차단 결과 및 현재 PC 상태를 데이터베이스에 기록하여, 향후 이력 관리 및 보안 대응에 활용한다.

1-3. 관리자에 의한 수동 네트워크 접근 허용 및 차단 시나리오

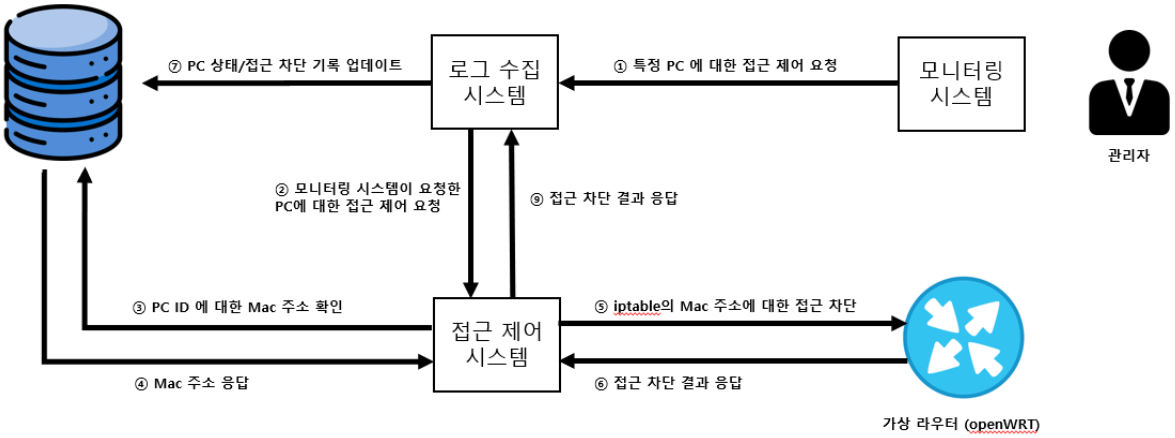


그림 5. 수동 네트워크 접근 허용/차단 흐름

그림 5는 보안 관리자가 모니터링 시스템을 통해 특정 PC의 이상 행위를 탐지했을 때, 해당 PC의 네트워크 접근을 수동으로 차단(또는 허용)하는 흐름을 설명한다. 이는 자동 탐지 외에 수동 판단 기반의 보안 대응 체계를 보완하는 시나리오이다. 주요 단계는 다음과 같다:

1. 특정 PC에 대한 접근 제어 요청

관리자는 모니터링 시스템을 통해 특정 PC에서 이상 행위 또는 정책 위반을 탐지한 경우, 해당 PC에 대한 접근 차단(또는 허용) 요청을 로그 수집 시스템에 전달한다.

2. 모니터링 시스템이 요청한 PC에 대한 접근 제어 요청

로그 수집 시스템은 해당 요청을 처리하여 접근 제어 시스템에 전달한다.

3. PC ID에 대한 Mac 주소 확인

접근 제어 시스템은 차단 명령을 실행하기 위해, 요청된 PC ID에 해당하는 Mac 주소를 확인하고자 데이터베이스에 질의한다.

4. Mac 주소 응답

   데이터베이스는 해당 PC의 Mac 주소를 접근 제어 시스템에 응답한다.

5. iptables를 통한 접근 차단/허용 수행

   접근 제어 시스템은 openWRT 기반의 가상 라우터에 접근 차단 또는 허용 명령을 전달한다. 이때 Mac 주소를 기반으로 iptables에 규칙이 설정된다.

6. 접근 차단 결과 응답

   가상 라우터는 명령 실행 결과를 접근 제어 시스템에 응답한다.

7. 접근 차단 결과 응답 처리

   접근 제어 시스템은 해당 결과를 로그 수집 시스템에 다시 전달한다.

8. PC 상태 및 접근 차단 기록 업데이트

   로그 수집 시스템은 차단/허용 결과와 현재 PC 상태를 데이터베이스에 기록하여, 전체 보안 로그 이력에 반영한다.

4. 구성원별 진척도

이름	진척도
이광훈	● 프론트엔드 UI 프로토타입 설계 및 구현 ● 백엔드 데이터 베이스 스키마 설계 ● 윈도우 로그 수집 에이전트 개발
신해진	● 이상 행위 탐지 모델 학습용 데이터 전처리 ● 이상 행위 탐지 모델 개발
조유진	● 로그 수집 에이전트 개발 ● 로그 수집 시스템 서버 개발

5. 과제 수행 내용 및 중간 결과

5.1 로그 수집 및 저장 시스템

- 로그 수집 에이전트

로그 수집 에이전트는 Fluentd를 활용하여 다양한 사용자 행동 이벤트 로그를 통합 수집하도록 설계되었다. 수집 대상은 로그온/로그오프, 디바이스 연결 및 해제, 파일 복사, 메일 발신, 웹 브라우징 등이며, 각 항목별 수집 방식은 다음과 같다.

로그온/로그오프 로그

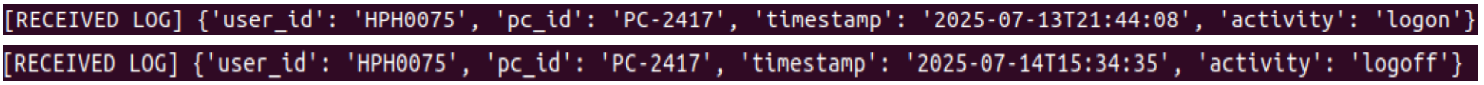


그림 6. 로그온(위)/로그오프(아래) 로그

로그온/로그오프와 이하 디바이스 연결, 파일 복사는 auditd를 통해 모니터링한다. auditd(Audit Daemon)는 리눅스 시스템에서 발생하는 다양한 이벤트에 대한 감사 정보를 수집하여 로그 파일에 저장한다. 사용자의 로그온/로그오프 이벤트는 감사 로그 중 USER_START와 USER_END 이벤트를 대상으로 필터링한다.

디바이스 로그

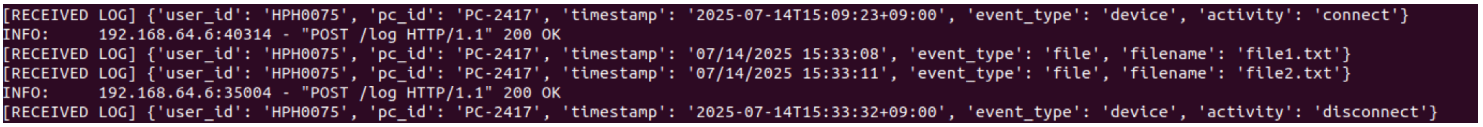


그림 7. 디바이스 연결/해제 및 파일 복사 로그

USB 장치의 연결(mount)와 해제(umount)는 /etc/audit/rules.d/ 디렉토리에 파일을 생성해 auditd 감사 규칙을 추가 정의하여 모니터링한다. 두 이벤트에 키워드를 각각 부여하고, Fluentd가 audit 로그를 tail 방식으로 읽으면서 해당 키워드가 포함된 로그만 필터링하여 수집한다.

파일 복사 로그

USB 장치로 파일이 복사되는 이벤트는 디바이스 로그와 유사하게 auditd 감사 규칙을 추가 정의하여 모니터링한다. USB 장치가 마운트된 디렉토리에 쓰기 작업이 발생하면 usb_copy 키워드가 로그에 부여되며, Fluentd에서 해당 키워드가 포함된 이벤트 중에서 실제 파일 복사 명령어(cp)만을 대상으로 추가 필터링하여 단순한 쓰기 작업이나 다른

프로그램에 의한 변경이 아닌, 사용자의 명시적인 복사 행위만을 추적할 수 있도록 한다.

메일 로그

메일 로그는 두 가지 방식으로 수집된다. 첫 번째는 내부 메일 서버(Postfix)를 활용한 방식이며, 두 번째는 웹메일 발송 시 발생하는 HTTP 요청을 mitmproxy로 가로채는 방식이다.

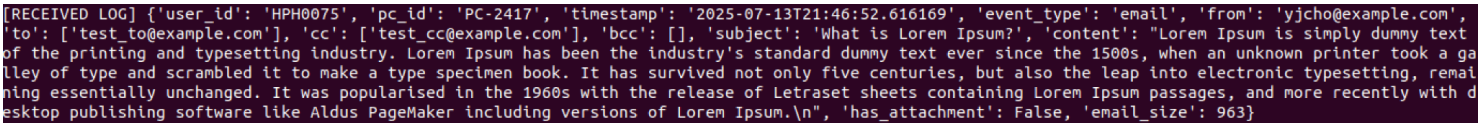
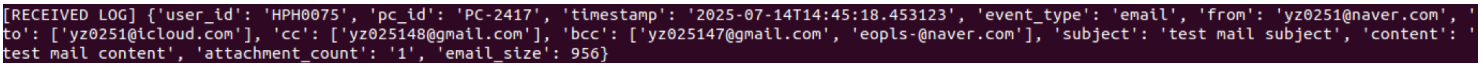


그림 8. Postfix 메일 발신 로그

Postfix 기반 로그 수집에서는 Postfix의 always_bcc 기능을 활용하여 모든 메일을 하나의 특정 계정으로 복사하도록 설정하였다. 해당 계정의 메일은 Maildir 형식으로 저장되며, Python 스크립트를 통해 주기적으로 Maildir의 신규 메일을 파싱하여 발신자, 수신자, 참조, 숨은 참조, 제목, 본문, 첨부파일



유무, 크기 등의 정보를 추출한다.

그림 9. 웹메일 발신 로그

웹메일 로그 수집은 mitmproxy를 사용하여 브라우저 트래픽을 가로채는 방식이다. 웹메일 도메인에 대해 특정 API 경로로 POST 요청이 들어오면 해당 요청을 분석하여 메일 송신 정보를 추출한다.

웹 브라우징 로그

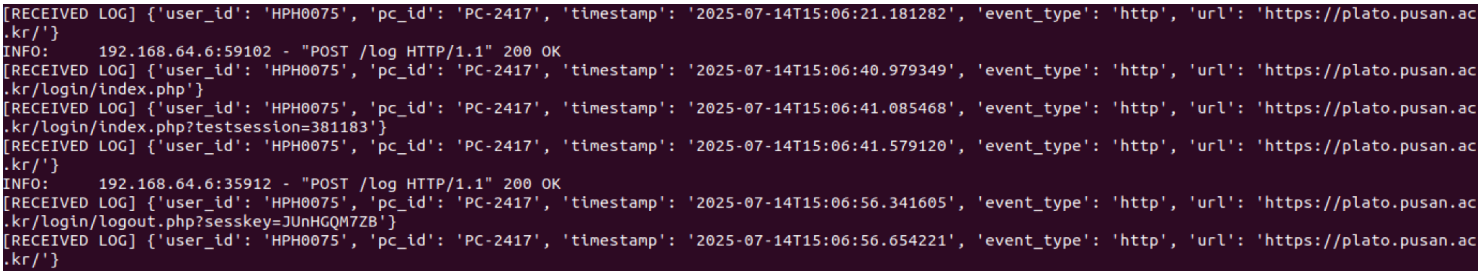


그림 10. 웹 브라우징 로그

웹 브라우징 로그는 웹메일 로그 수집과 유사하게 mitmproxy를 활용하여 브라우저 트래픽을 가로채는 방식으로 수집한다.

- 로그 수집 시스템 서버

  FastAPI 기반의 로그 수집 시스템 서버는 Fluentd로부터 전달되는 이벤트 로그를 수신하여 데이터베이스에 저장한다.

- 향후 과제 수행 계획

  현재는 로그 수집 기능이 개별 스크립트로 분리되어 있으나, 이를 단일 로그 수집 에이전트로 통합하여 사용자 단말에 배포 가능한 형태로 구성한다.
  수집된 모든 로그가 공통 필드를 포함하는 Behavior_logs 테이블에 저장되며, event_type 필드를 기준으로 이벤트 유형별 세부 테이블과 연동되는 구조로 데이터베이스를 설계에 맞게 구현한다.
  로그온 이벤트 로그 발생 시 PC 상태 정보를 실시간으로 갱신하는 기능과, 특정 PC에서 이상 사용자의 로그온 이벤트가 감지되면 네트워크 접근 제어 시스템에 해당 PC에 대한 제어 요청을 전송하는 기능을 서버에 도입한다.

5.2 이상 행위 탐지 시스템

MG-UABD 논문에서 제시한 구조를 CERT R4.2 공개 데이터셋에 기반하여 구현하고 테스트하였다.

전처리 단계 설계

로그 데이터를 사용자-일(user-day) 단위로 집계하여 주요 행동 특성을 추출한 뒤, 이를 날짜 기준으로 통합해 개별 사용자의 일별 활동 데이터를 구성한다. 이후 주차 정보를 생성하고 사용자-주(user-week) 단위로 요약하여 CAD(coarse-grained detection)에 적합한 형태로 정비한다. 사용자 프로필은 시점별로 가장 가까운 값을 기준으로 병합하고, 주요 범주형 항목은 수치형으로 인코딩하여 학습이 가능하도록 한다. 마지막으로 insiders.csv를 활용해 사용자-주 단위 데이터에 이상 여부를 라벨링함으로써 CAD 모델 학습과 테스트를 위한 기반 데이터를 완성한다.

```python
def extract_logon_features(df):   #user-day 단위로 행동 특성 추출

    df["offhour"] = (df["hour"] < 8) | (df["hour"] >= 18)

    agg = df.groupby(["user_id", "date"]).agg(
        logon_count=("activity", lambda x: (x == "Logon").sum()),
        logoff_count=("activity", lambda x: (x == "Logoff").sum()),
        offhour_logon_count=("activity", lambda x: ((x == "Logon") & (df.loc[x.index, "offhour"])).sum()),
        unique_pc_count=("pc", lambda x: x.nunique()),
        logon_label=("label", lambda x: 1 if (x == 1).any() else 0)
    ).reset_index()
    return agg
```

그림 11. 행동 특성 추출

```python
def aggregate_user_week(df_user_day):  # user-week 단위로 로그 병합

    columns_to_exclude = ["user_id", "date", "ios_year", "iso_week", "year_week"]
    columns_for_sum = [col if col not in columns_to_exclude for col in df_user_day.columns]

    agg_dict = {col: 'sum' for col in columns_for_sum}

    df_user_week = df_user_day.groupby(["user_id", "year_week"]).agg(agg_dict).reset_index()
    df_user_week["label"] = (df_user_week["label"] > 0).astype(int)
    return df_user_week
```

그림 12. 주 단위로 로그 병합

이상 행위 탐지 시스템 구조 구현

이상 행위 탐지 시스템은 CAD→FAD의 2단계로 구성된다. CAD 단계에서는 전처리 단계에서 완성한 데이터를 모델 훈련용과 테스트용으로 분할한 뒤 훈련용 데이터를 사용해 랜덤 포레스트 모델을 학습한다. 테스트용 데이터를 학습된 모델의 입력으로 하여 이상 사용자 후보군 리스트를 . 일부 오탐지를 감수하더라도 가능한 많은 의심 사용자를 탐지해야 한다.

```python
def train_and_predict_cad(X_train, y_train, X_test, y_test, df_test, threshold=0.5):
    # CAD 모델 학습 및 이상 사용자로 의심되는 ID 리스트 생성

    clf = RandomForestClassifier(n_estimators=100, random_state=42, class_weight="balanced")
    clf.fit(X_train, y_train)

    y_prob = clf.predict_proba(X_test)[:, 1]
    y_pred = (y_prob >= threshold).astype(int)

    df_test_with_pred = df_test.copy()
    df_test_with_pred["pred_label"] = y_pred
    suspicious_user_ids = df_test_with_pred[df_test_with_pred["pred_label"] == 1]["user_id"].unique().tolist()
    return suspicious_user_ids, y_pred
```

그림 13. CAD 모델

FAD 단계에서는 CAD 결과에서 탐지된 이상 사용자 후보군에 한해 원본

데이터를 다시 불러오고, 날짜 단위 행동 데이터를 정제하여 개인별 행동 특성을 새롭게 구성한다. 이때 변동성, 비율 등 정교한 통계치를 포함한다. 사용자별로 개별 랜덤 포레스트 모델을 학습한다. 또한, 학습 데이터의 불균형을 완화하기 위해 SMOTE 기반 오버샘플링 기법을 적용한다.

```python
def train_and_evaluate_fad(df_fad_labeled, test_ratio=0.3, random_state=42, verbose=True):
    # FAD 모델 학습 및 테스트, 평가
    user_models = {}
    user_predictions = {}
    user_ground_truths = {}
    for user_id, df_user in df_fad_labeled.groupby("user_id"):
        df_user = df_user.sort_values("date")
        # (중략)
        drop_cols = ["user_id", "date", "label"]
        X_train = df_train.drop(columns=drop_cols, errors="ignore")
        y_train = df_train["label"]
        X_test = df_test.drop(columns=drop_cols, errors="ignore")
        y_test = df_test["label"]
        # (중략)
        model = RandomForestClassifier(random_state=random_state)
        model.fit(X_train, y_train)
        y_pred = model.predict(X_test)

        is_anomalous = (y_pred == 1).any()
        user_predictions[user_id] = int(is_anomalous)
        is_true_anomaly = (y_test == 1).any()
        user_ground_truths[user_id] = int(is_true_anomaly)
        user_models[user_id] = model
        # (중략)
    return user_models
```

그림 14. FAD 모델

향후 과제 수행 계획

📈 [FAD 사용자 단위 평가 결과]

	precision	recall	f1-score	support
0	0.0714	1.0000	0.1333	1
1	0.0000	0.0000	0.0000	13
accuracy			0.0714	14
macro avg	0.0357	0.5000	0.0667	14
weighted avg	0.0051	0.0714	0.0095	14

그림 15. 모델 평가 지표

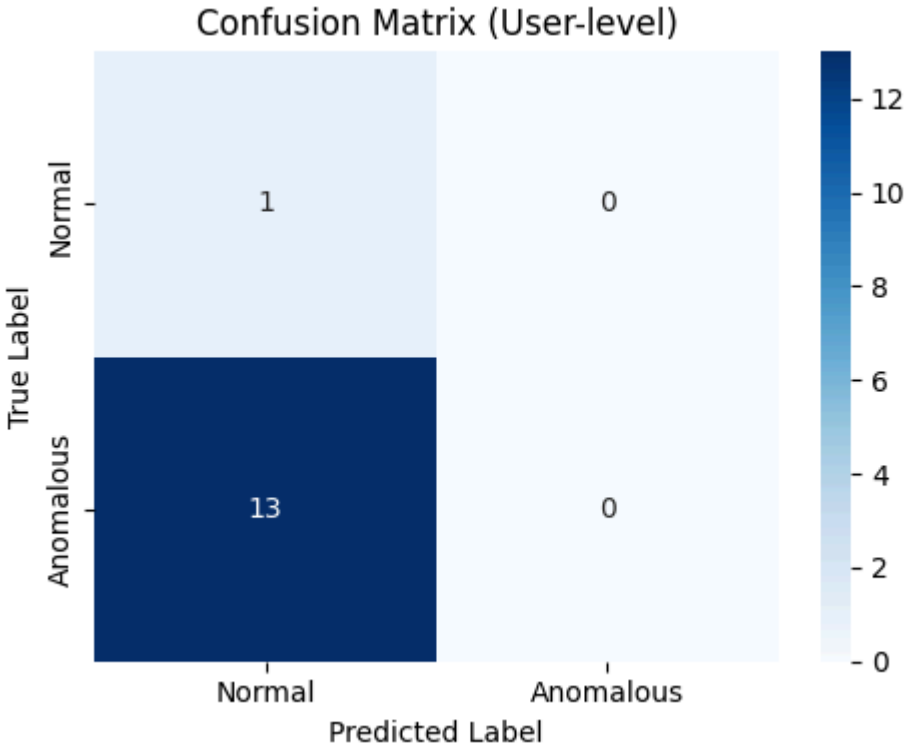


그림 16. 모델의 confusion matrix

앞으로는 구현한 CAD→FAD 모델을 다양한 조건에서 그림 16. 모델의 confusion matrix

테스트하고, 정량적인 성능 지표(F1-score, recall, precision 등)를 산출한다. 지금까지 학습한 모델을 테스트한 결과, 성능이 안 좋기 때문에 행동 특성(feature) 구성을 조정하거나 학습/테스트 데이터 분할 방식 변화 등을 통해 모델 성능 개선 작업을 진행한다. 이후에는 CERT R4.2 전체 로그 데이터를 학습에 활용하여 최종 모델을 구축하고, 로그 수집 에이전트와 연동되는 형태의 이상 사용자 탐지 인터페이스를 개발한다.

5.3 모니터링 시스템

모니터링 시스템은 네트워크에 연결된 각 사용자 단말기의 상태를 시각적으로 모니터링하고, 이상 사용자의 행위 및 네트워크 접근 제어 상황을 실시간으로 파악할 수 있는 관리자용 웹 기반 보안 대시보드이다. 시스템의 설계 목표는 관리자에게 직관적이고 신속한 보안 통제 기능을 제공하는 것이다.

초기 구현 단계에선 웹페이지 레이아웃 설계, 회원 가입 및 로그인 페이지 등의

기능을 완료하였다(그림 16, 17, 18). 향후 기능으로는 다음과 같은 모듈이 추가될 예정이다.

- 네트워크 토폴로지 시각화(React flow 기반)
- 단말기별 로그 조회 필터링 기능
- 자동 접근 제어 발생 시 실시간 팝업/이메일 알림 기능
- 접근제어 기록 확인 및 수동 제어 기능

그림 17. 초기 레이아웃 페이지

그림 18. 로그인 페이지

그림 19. 회원가입 페이지