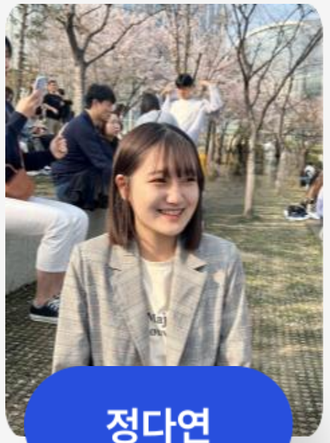
# OSS를 이용한
# ZTNA 인증 서비스 개발

# Contents

프레젠테이션 목차 안내

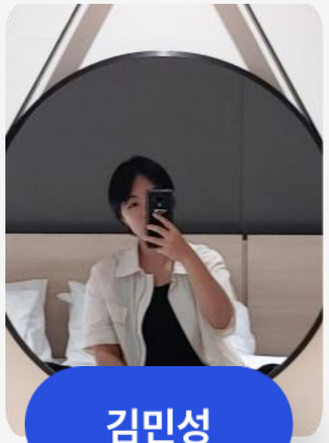
# Role

팀원별 역할 분담



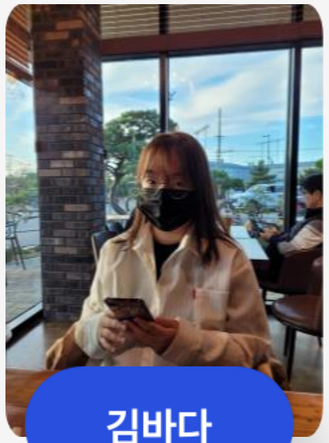
**정다연**

PM
ZTNA 연동 인프라 설계
K8S 인프라 구축



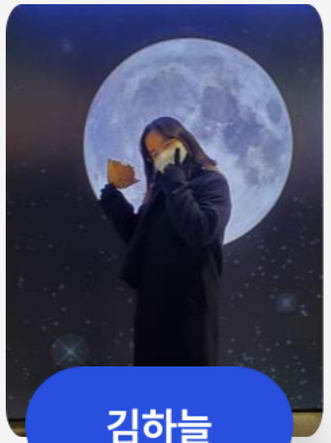
**김민성**

KEYCLOAK 인증서 관리
USER 연동 테스트



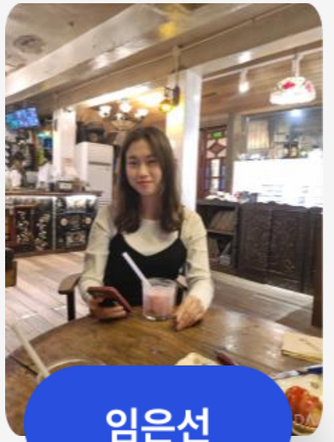
**김바다**

USER FEDERATION
KEYCLOAK THEME 설정



**김하늘**

KEYCLOAK
JWT 개발



**임은선**

KEYCLOAK
OAUTH & OIDC

# Original Develop

기존 개발 목표 사항


**DB 연동**

기존 KeyCloak의 H2 DB가 아닌 기업에서 사용할 수 있는 외부 DB 연동 시스템을 계획하였습니다.

**외부 인증 연동**

해당 서비스 이용 시 외부 OAUTH 연동을 할 수 있도록 외부 인증 연동 기능을 목표하였습니다.

**접속 인가**

KEYCLOAK 서비스 내에서 허가된 사용자만 사용할 수 있도록 인가 서비스를 계획하였습니다.

# Original Develop

기존 개발 목표 사항


**K8S 환경**

다양한 기업에서 사용하고 있는 쿠버네티스 환경에 연동이 가능하도록 K8S 환경에 구축하도록 계획하였습니다.

**대칭키 교환**

로그인 발생시 해당 사용자가 대상 서버에 접근할 경우 미리생성한 16자리 키와 매칭하여 유효성을 검증하도록 계획하였습니다.

**freeRadius**

인증 인가를 위한 프로토콜 서버로, 해당 서버를 이용하여 사용자의 정보를 관리할 수 있도록 추가 개발을 계획하였습니다.

# Custum Plugin

User Federation : MySQL DB 연동

### DB Federation

외부 DB 연동이 가능하도록 연동을 할 수 있는 기능 개발
이후 MySQL 외에도 다른 플러그인까지 이용할 수 있도록
확장 예정

### Custum User Management

ID, EMAIL로 USER를 찾을 수 있도록
USER 관련된 기능 구현
USER 및 EMAIL 검증 기능 구현

**Custum Plugin**

### Event Listener

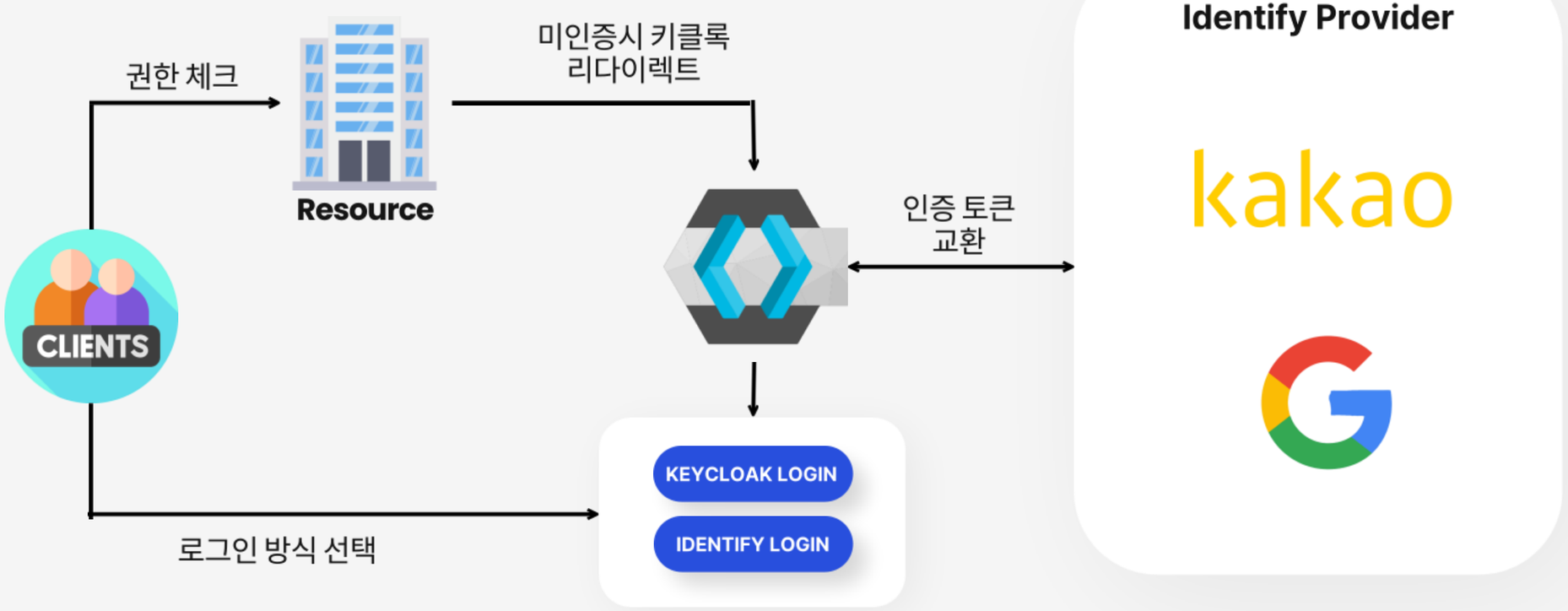
로그인 시 IP 및 타임스탬프를 추출하는 기능 구현
비정상 로그 탐지로 추후 연결 가능

### Custum theme

사용자가 원하는 방식의 다양한 테마를 사용할 수 있도록
theme 기능 구현

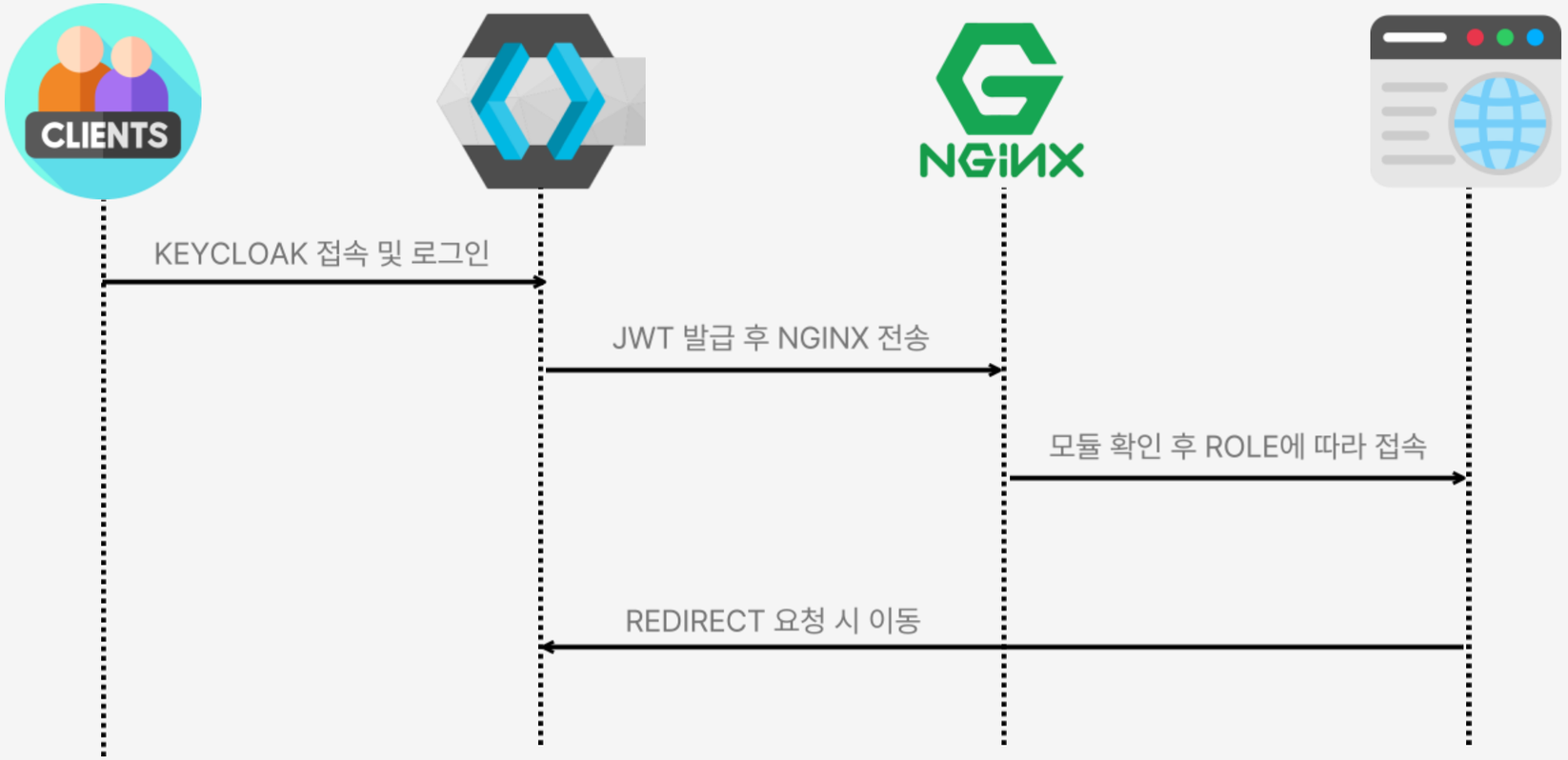
# OAuth & OIDC

KeyCloak을 이용한 외부 인증 연동 통합 관리



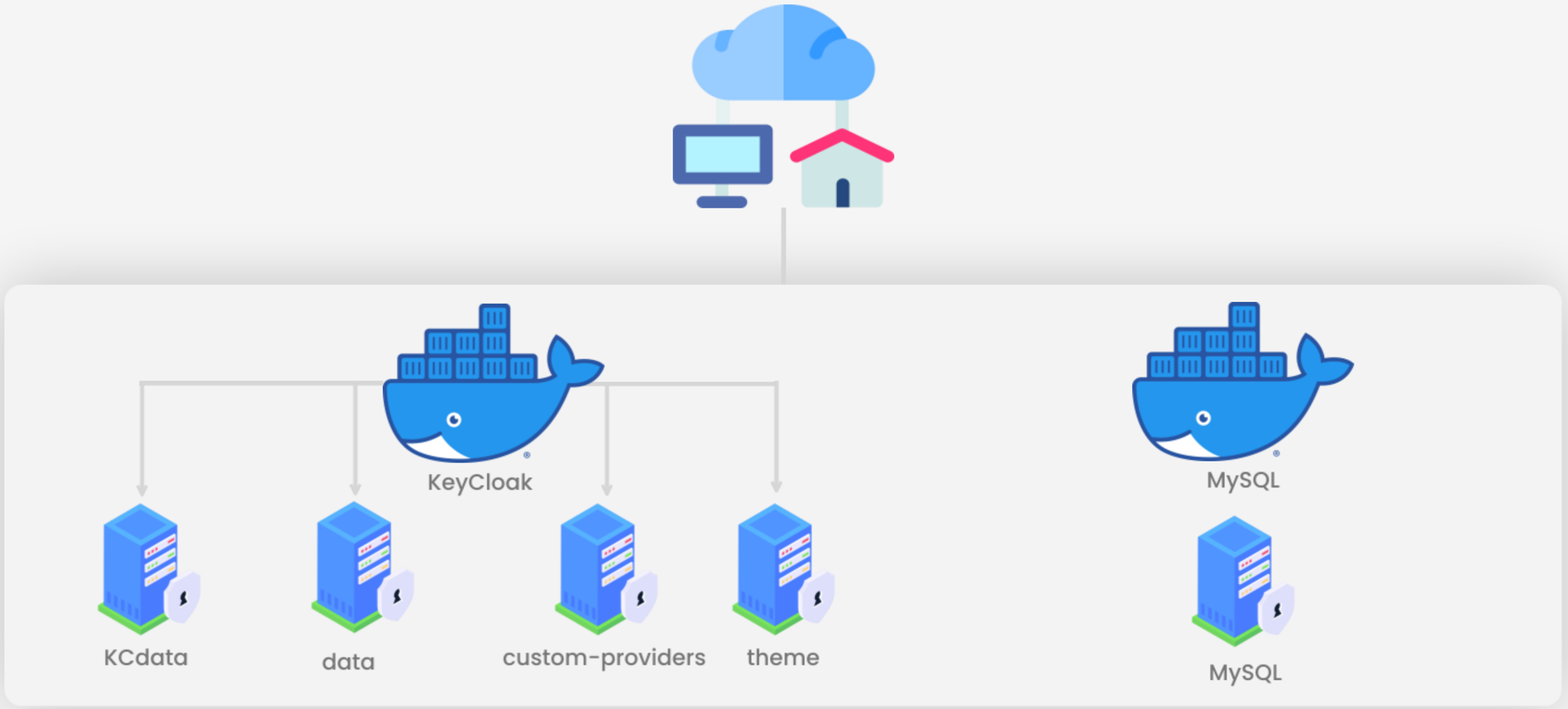
# JWT

Keycloak을 이용한 Application 접속 권한 부여



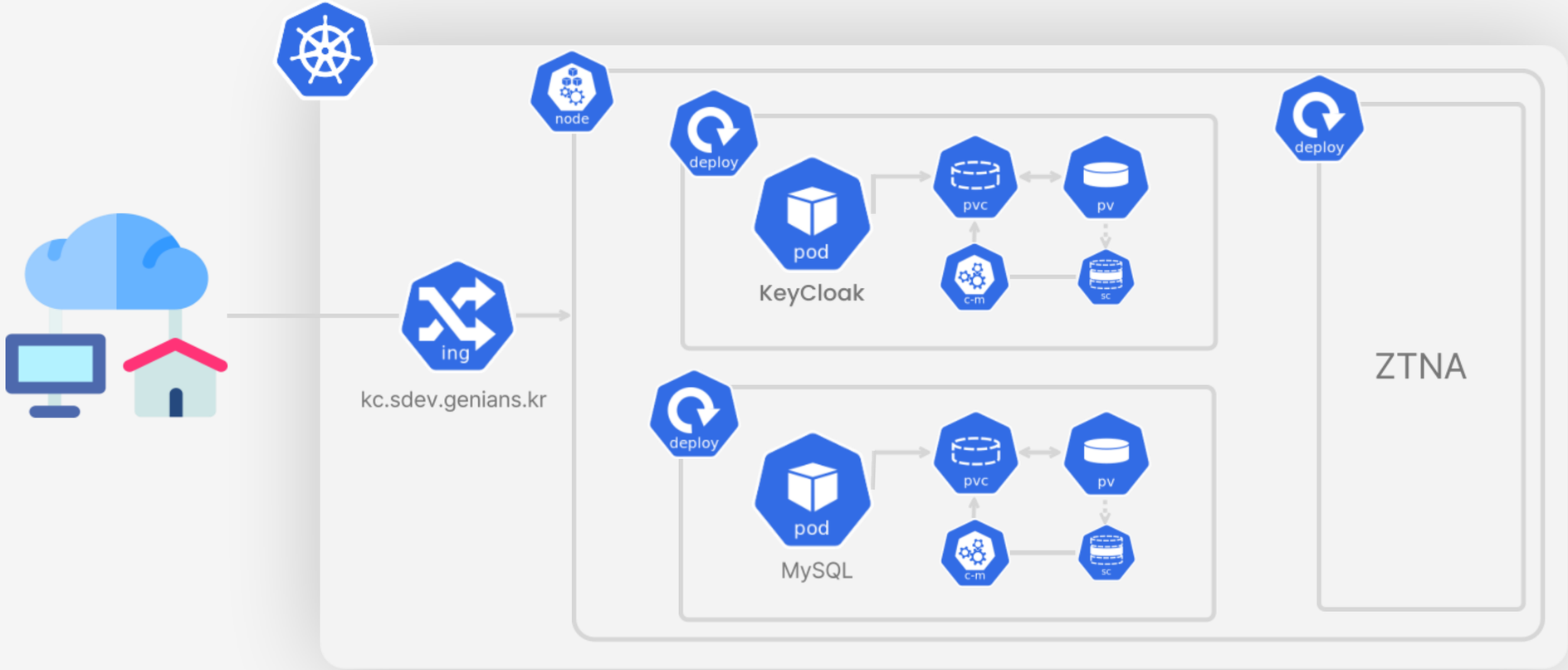
# Infra: docker

ZTNA와의 연동을 위한 도커 인프라 구축



# Infra: kubernetes

ZTNA와의 연동을 위한 쿠버네티스 인프라 구축



# Project Research

연구로 진행된 결과 보고 : 공개키 교환

### 기존 **JWT** 키 교환의 한계

KeyCloak 인증 시 발행 토큰(JWT)은 공개키로 검증이 가능하지만, 토큰이 길다면 전체 토큰 전달에 어려움이 발생하게 된다.

**Key Keywords**

### 공개키 교환방식의 도입

KEYCLOAK 사용자 속성 값에 랜덤한 16자리의 키 생성 이후 토큰 발행 시 해당 토큰에 16자리의 값을 담아 사용자에게 전달 가능

**Conclusion**

즉, Event Listener를 이용하여 로그인이 발생 시 토큰 매칭이 가능하다.
UserId, IpAddress, Key 3가지를 서버로 전달 시 유효성 검증을 할 수 있다.

# Coderay

코드레이 분석 결과

### 1.1 프로젝트 분석 보안경보 현황

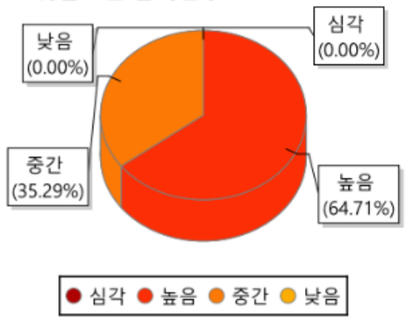

KLOC (Kilo Line Of Code) : 15 (1000 라인 당 15 건의 보안 약점 발생)

| 보안단계 | 5단계(정상) | 4단계(관심) | 3단계(주의) | 2단계(경계) | 1단계(심각) |
|---|---|---|---|---|---|
| 보안경보 구간 | 0% ~ | 25% ~ | 50% ~ | 75% ~ | 100% ~ |
| 위험도 점수 구간 | 0 ~ | 128 ~ | 257 ~ | 385 ~ | 514 ~ |

* 보안경보 구간 : 소스코드의 총 라인수를 KLOC 15 기준으로 환산하여 5단계의 보안경보 구간 도출
* 보안경보 레벨 : 위험도 점수의 총 합이 속하는 보안경보 구간 및 위험도 점수의 구간을 확인하여 보안경보 레벨 도출

### 1.2 프로젝트 분석 위험도 현황

위험도별 탐지건수



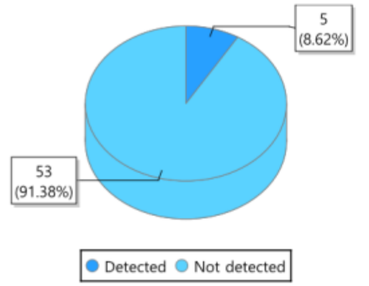

위험도 점수

| 위험도 | 탐지건수 | 위험도 가중치 | 위험도 점수 |
|---|---|---|---|
| 심각 | 0 | 9 | 0 |
| 높음 | 11 | 3 | 33 |
| 중간 | 6 | 2 | 12 |
| 낮음 | 0 | 1 | 0 |
| 탐지 건수 | 17 | | **45** |

* 탐지건수 x 위험도 가중치 = 위험도 점수

### 1. 프로젝트 분석결과 요약

보안약점검출 파일비율




| 분석 정보 | |
|---|---|
| 프로젝트 이름 | OSS를 활용한 SASE 서비스 어플 |
| 분석회차 (분석일) | 2차 (2023-11-28) |
| 점검 기준 | 소프트웨어 개발보안 가이드 49개 항목 |
| 분석 언어 | ANDROID,ASP,ASPX,C/C++,CS,DART,HTML/JDS,JAVA,JAVASCRIPT,JSP,KOTLIN,NEXACRO,PHP,PYTHON,SWIFT,VBSCRIPT,XML |
| 탐지시간 | 00:00:08.475 |

| 분석 파일 정보 | |
|---|---|
| 분석 파일 | 58 건 |
| 분석 라인 | 3,805 라인 |
| 탐지파일 | 5 건 |
| 탐지원인 | 3 건 |
| 탐지결과 | 17 건 |

# Coderay

코드레이 분석 결과

**2.1** [ 4.1. 오류 메시지 정보노출 ]

| 보안약점ID | 위험도 | 보안약점 | 원인패턴 건수 | 탐지건수 |
|---|---|---|---|---|
| CWE-209 | 높음 | 민감한 정보가 포함 된 오류 메시지 생성 | 1 | 11 |

**2.2** [ 4.2. 오류상황 대응 부재 ]

| 보안약점ID | 위험도 | 보안약점 | 원인패턴 건수 | 탐지건수 |
|---|---|---|---|---|
| CWE-390 | 중간 | 조치없이 오류 조건 감지 | 1 | 4 |

**2.3** [ 5.2. 부적절한 자원 해제 ]

| 보안약점ID | 위험도 | 보안약점 | 원인패턴 건수 | 탐지건수 |
|---|---|---|---|---|
| CWE-404 | 중간 | 부적절한 리소스 종료 또는 해제 | 1 | 2 |

# Development Demonstration

개발 시연

# Thank you.