Software Test Plan

1. Introduction (소개)

1.1. Purpose (목적)

본 Software Test Plan(STP)의 목적은 Database Intrusion Detection System (DB-IDS)이 SRS와 SDS에서 정의된 요구사항을 충족하는지 검증하기 위한 체계적인 테스트 전략과 절차를 제시하는 것이다.

1.2. Scope (범위)

- **테스트 대상**:
  - SQL Proxy/Collector
  - Spring Boot 기반 Detection Engine (패턴 기반, 행동 기반, 권한 기반 탐지)
  - 알림 모듈 (Slack, Email)
  - React 기반 관리자 대시보드
  - 로그 저장소 (SQLite + S3 아카이브)
- **테스트 포함 기능**:
  - SRS에서 정의된 기능 요구사항(FR-1 ~ FR-7)
  - 성능 요구사항(PR-1 ~ PR-6)
  - 보안 요구사항 (TLS, RBAC, 접근제어)
- **테스트 제외 기능**:
  - 자동 차단 기능
  - ML 기반 이상 탐지
  - PostgreSQL/Oracle 멀티 DBMS 지원
  - PDF 리포팅, 다국어 UI 지원

1.3. References (참고자료)

- IEEE Std 829-2008, *Software and System Test Documentation*
- IEEE Std 830-1998, *Software Requirements Specification*
- IEEE Std 1016-1998, *Software Design Description*
- DB-IDS SRS v1.0

- DB-IDS SDS v1.0

2. Test Items (테스트 대상 항목)

DB-IDS 프로젝트에서 테스트할 주요 대상 항목은 다음과 같다. 각 항목은 SRS의 기능 요구사항(FR) 및 SDS의 컴포넌트 설계와 직접적으로 연결된다.

2.1. SQL Proxy / Collector

- 설명: Client Application의 SQL 요청을 가로채어 MySQL로 전달하며, 실행 메타데이터를 수집한다.
- 테스트 포인트: SQL 요청 정상 전달 여부, 로그 데이터 누락 여부, 비정상 인코딩 차단.

2.2. Detection Engine (Spring Boot Service)

- Pattern-based Detection (FR-2): SQL Injection, DROP TABLE, UNION SELECT 등 패턴 탐지
- Behavior-based Detection (FR-3): 쿼리 빈도, 반환 행 수, 실행 시간 기반 행동 이상 탐지
- Authorization-based Detection (FR-4): 비인가 계정의 민감 테이블 접근/INSERT/DELETE 등 권한 탐지
- 테스트 포인트: 정상/이상 SQL 구분 정확성, 오탐/미탐 비율, 이벤트 기록

2.3. Notification Module (Slack & Email)

- 설명: 탐지 이벤트 발생 시 관리자에게 실시간 알림 전송
- 테스트 포인트: 알림 포맷, 전송 시간 SLA(≤10초), 실패 시 3회 재시도 로직, NotificationLog 기록

2.4. Dashboard (React UI)

- 설명: 관리자가 로그 및 탐지 이벤트를 실시간 조회하고, 알림 설정을 관리한다.
- 테스트 포인트: 로그인 인증, 실시간 로그 표시(5초 주기), 이벤트 목록/상세 조회, 통계 그래프, 알림 설정 저장

2.5. Storage (SQLite + Amazon S3)

- SQLite: QueryLog, DetectionEvent, NotificationLog, AdminUser, ArchiveLog 저장
- S3 Archive: 일정 기간이 지난 로그의 압축 아카이빙
- 테스트 포인트: 데이터 무결성, 아카이브 업로드 성공/실패 처리, 30일 보관 정책 반영

3. Features to be Tested / Not Tested

3.1. Features to be Tested (테스트 대상 기능)

다음 기능들은 DB-IDS SRS (3.2 기능 요구사항, 3.3 성능 요구사항, 3.6 시스템 속성)에 정의된 항목으로, 본 테스트 범위에 포함된다.

FR-1. SQL Query Logging

- 모든 SQL 쿼리 기록 (원문/요약/메타데이터)
- 로그 검색, 필터, 다운로드

FR-2. Pattern-based Detection

- SQL Injection, DROP TABLE, UNION SELECT 등 금지 패턴 탐지
- 이벤트 생성 및 알림 전송

FR-3. Behavior-based Detection

- 쿼리 실행 시간, 반환 행 수, 빈도 기반 이상 탐지
- 스코어 계산 및 임계값(≥0.8) 판정

FR-4. Authorization-based Detection

- 비인가 계정의 민감 테이블 접근 및 DML 시도 탐지
- RBAC 정책 위반 이벤트 생성

FR-5. Notification

- Slack/Email 실시간 알림
- SLA: 1~10초 이내 전송, 실패 시 최대 3회 재시도

FR-6. Dashboard (React UI)

- 실시간 로그 조회(5초 주기 갱신)
- 탐지 이벤트 목록 및 상세 확인
- 통계 그래프 (사용자별/시간대별)
- 알림 채널 설정

FR-7. Log Backup & Archive

- 일일 로그 백업 → Amazon S3 업로드
- 실패 시 재시도 및 기록
- 30일 이후 스토리지 클래스로 이동
- **성능 요구사항 (PR-1 ~ PR-6)**
  - 탐지 지연 ≤ 1초, 대시보드 응답 ≤ 2초, 알림 ≤ 10초

    - 초당 1,000건 이상 SQL 처리
    - 동시 100명 사용자 지원
    - MTTR ≤ 1h, 가용성 ≥ 99%
    - 오탐 ≤ 5%, 미탐 ≤ 2%
- **보안 요구사항**
    - 모든 통신 HTTPS (TLS 1.2+)
    - 비밀번호 SHA-256 이상 해시 저장
    - 관리자 계정 5회 실패 시 잠금
    - RBAC 기반 접근 제어

3.2. Features Not to be Tested (테스트 제외 기능)

다음 기능들은 SRS 2.6 (향후 확장 요구사항)에 정의되었으나, 본 프로젝트의 테스트 범위에는 포함되지 않는다.

- 자동 차단 기능
- 머신러닝 기반 이상 탐지
- 멀티 DBMS 지원
- PDF 자동 리포트 생성 기능
- 다국어 지원

4. Test Approach (테스트 전략)

본 절에서는 DB-IDS의 테스트를 수행하기 위한 전반적인 접근 방법을 설명한다. 테스트는 단위(Unit), 통합(Integration), 시스템(System), 성능/보안(Non-functional) 단계로 나누어 진행되며, 각 단계는 점진적으로 요구사항을 검증하도록 설계한다.

4.1. Unit Testing (단위 테스트)

4.1.1. 목적

- 각 모듈(Proxy, Detection Engine, Notifier, Storage 등)이 설계(SDS)에 맞게 독립적으로 동작하는지 검증한다.
- 조기 결함 발견을 통해 통합 테스트 이전에 품질을 확보한다.

4.1.2. 범위 및 항목

- SQL Proxy / Collector

    - SQL 파싱 및 메타데이터 추출 정확성
    - 비정상 SQL 인코딩 차단
- Detection Engine
    - Pattern-based Rule 매칭 정확성 (화이트리스트/블랙리스트)
    - Behavior-based Score 계산 정확성 (임계치 계산)
    - AuthZ Role-Policy 매칭 정확성
- Notifier
    - Slack/Email 전송 함수의 재시도 로직 검증
- Storage (SQLite)
    - QueryLog/DetectionEvent/NotificationLog 삽입 및 무결성
    - SHA-256 비밀번호 해시 검증

4.1.3. 접근 방법

- JUnit5 (Spring Boot 모듈) 기반 유닛 테스트 코드 작성
- Mocking 활용하여 외부 API(Slack, SMTP) 의존성 제거
- 단일 함수/클래스 단위 검증
- 통과 기준: 함수 반환 값, DB 저장 데이터, 로직 분기 기대값 일치

4.1.4. 테스트 케이스

TC ID	대상 모듈	입력	예상 결과	판정 기준
UT-01	Proxy Request Handler	`SELECT * FROM users;`	MySQL로 쿼리 전달, SQLRaw 추출됨	SQLRaw=원문, 상태=성공
UT-02	Proxy Response Handler	ResultSet(10 rows)	RowCount=10, Status=SUCCESS 저장	RowCount 값=10
UT-03	RuleEngine	`DROP TABLE students;`	금지 패턴 탐지 → Event 생성	Event.Type=PATTERN
UT-04	RuleEngine	`SELECT name FROM users;`	정상 쿼리, 탐지 없음	Event=null
UT-05	BehaviorEngine	1초에 200개의 SELECT 요청	Score ≥ 0.8, 이상 탐지 발생	Event.Type=BEHAVIOR
UT-06	BehaviorEngine	행 반환 수=100	Score < 0.8, 탐지 없음	Event=null

TC ID	대상 모듈	입력	예상 결과	판정 기준
UT-07	AuthZEngine	User=READ_ONLY, SQL=`DELETE FROM orders;`	권한 위반 탐지 발생	Event.Type=AUTHZ
UT-08	AuthZEngine	User=DBA, SQL=`DELETE FROM orders;`	정상 통과	Event=null
UT-09	Notifier (Slack)	Event=High Severity	Slack Webhook 호출 성공	NotificationLog.Status=SENT
UT-10	Notifier (Email)	Event=Medium Severity	이메일 전송 성공	NotificationLog.Status=SENT
UT-11	Storage (SQLite)	QueryLog 저장 요청	DB에 정상 insert	Row count 증가
UT-12	Storage (S3 Job)	만료 로그 100건	압축 파일 생성, S3 업로드 성공	ArchiveLog 생성됨

4.2. Integration Testing (통합 테스트)

4.2.1. 목적

- DB-IDS의 주요 컴포넌트(Proxy, Analysis Engine, Storage, Notifier, Dashboard) 간 상호작용이 정상적으로 이루어지는지 검증한다.
- 단위 테스트에서 개별적으로 통과한 기능이 실제 데이터 플로우 상에서도 기대대로 동작하는지 확인한다.

4.2.2. 범위 및 항목

- Proxy ↔ Detection Engine: SQL 메타데이터 전달 및 이벤트 생성 여부
- Detection Engine ↔ Storage: QueryLog와 DetectionEvent 저장 연계
- Detection Engine ↔ Notifier: 이벤트 발생 시 알림 트리거 동작 여부
- Dashboard ↔ API (Spring Boot): 로그/이벤트 API 연동, SSE 실시간 스트리밍

4.2.3. 접근 방법

- Spring Boot Test 환경에서 Embedded DB(SQLite) 사용
- API 호출(Postman)로 시나리오 검증
- Mock SMTP/Slack 서버 구성하여 실제 전송 대신 응답 시뮬레이션

4.2.4. 테스트 케이스

TC ID	시나리오	입력	예상 결과	판정 기준
IT-01	쿼리 로깅 통합	Client → Proxy: `SELECT * FROM users;`	Proxy → Analysis 전달, Storage에 QueryLog 저장	Dashboard API에서 해당 로그 조회 가능
IT-02	패턴 기반 탐지 통합	Client: `DROP TABLE orders;`	Analysis에서 탐지, DetectionEvent 생성	Dashboard 이벤트 목록에 표시
IT-03	행동 기반 탐지 통합	동일 사용자 200TPS 실행	BehaviorEngine 이상 탐지 발생	Event.Type=BEHAVIOR, Notification 전송
IT-04	권한 기반 탐지 통합	READ_ONLY 계정 → `INSERT INTO users;`	권한 위반 탐지, Event 생성	Slack/Email 알림 도착
IT-05	알림 채널 연동	High Severity Event	Slack/Email API 호출 성공, NotificationLog 기록	NotificationLog.Status=SENT
IT-06	알림 채널 장애	Slack API 실패 (Mock)	3회 재시도 후 FAILED 기록	NotificationLog.Status=FAILED
IT-07	대시보드 연동	Admin → Dashboard 이벤트 조회	API가 SQLite에서 이벤트 읽음	UI에 최신 이벤트 표시됨
IT-08	로그 아카이브 연동	오래된 QueryLog 100건	Jobs 모듈이 압축 업로드, ArchiveLog 생성	ArchiveLog.FilePath 생성 확인
IT-09	로그+이벤트 일관성	탐지 Event 발생 시	QueryLog와 Event FK 관계 유지	Event.LogID = QueryLog.LogID

4.3. System Testing (시스템 테스트)

4.3.1. 목적

- DB-IDS 전체 시스템이 요구사항에 따라 동작하는지 검증한다.
- 기능 요구사항(FR-1~FR-7)을 실제 시나리오 중심으로 테스트하여, 관리자가 사용하는 관점에서 정상/이상 동작을 확인한다.

4.3.2. 범위 및 항목

- FR-1 ~ FR-7 기능 요구사항 전체
- 대시보드 로그인 → SQL 실행 → 탐지 이벤트 → 알림 전송 → 백업까지 흐름 검증
- 관리자의 RBAC 기반 접근 제어 확인

4.3.3. 접근 방법

- AWS EC2(Ubuntu 20.04, t3.medium) 환경에서 배포
- 실제 MySQL 8.0 테스트 DB 연결
- 관리자 계정/비인가 계정 시나리오 실행

4.3.4. 테스트 케이스

TC ID	요구사항	시나리오	예상 결과	판정 기준
ST-01	FR-1 쿼리 로깅	Admin 계정 → `SELECT * FROM students;` 실행	QueryLog 저장, Dashboard에서 조회 가능	QueryLog.ExecTime/SQLRaw 일치
ST-02	FR-2 패턴 탐지	`UNION SELECT password FROM users;`	패턴 탐지 Event 생성, Slack/Email 알림 발송	Event.Type=PATTERN, 알림 도착
ST-03	FR-3 행동 탐지	1분간 10,000건 SELECT 실행	Behavior Engine에서 이상 탐지 Event 생성	Event.Type=BEHAVIOR, Severity=HIGH
ST-04	FR-4 권한 탐지	READ_ONLY 계정 → `DELETE FROM orders;`	권한 위반 탐지 Event, 알림 전송	Event.Type=AUTHZ, 알림 도착
ST-05	FR-5 알림	High Severity Event 발생	Slack + Email 각각 알림 도착, NotificationLog 저장	NotificationLog.Status=SENT
ST-06	FR-6 대시보드	Admin 로그인 후 이벤트 페이지 조회	Event 목록/통계 정상 표시	UI=DB 값과 일치
ST-07	FR-6 대시보드 인증 실패	잘못된 PW 입력	로그인 거부, 에러 메시지 표시	인증 실패 메시지 출력
ST-08	FR-7 로그 아카이브	30일 이상 된 로그 존재	Job 실행 시 압축 업로드, ArchiveLog 생성	ArchiveLog.FilePath 존재
ST-09	FR-7 아카이브 실패	S3 접근 불가 상황	업로드 실패, 재시도 후 실패 로그 남김	ArchiveLog 없음, 오류 기록

4.4. Non-functional Testing (비기능 테스트)

4.4.1. 목적

- DB-IDS가 성능, 보안, 신뢰성 측면에서 요구사항을 충족하는지 검증한다.
- 최소한의 MVP 수준 테스트를 통해 시스템이 실제 운영 환경에서도 동작 가능함을 확인한다.

4.4.2. 범위 및 항목

- 성능 (Performance): 탐지 지연 시간, 처리량, 동시 접속자 응답 속도
- 보안 (Security): SQL Injection 탐지, 권한 기반 제어, 관리자 인증 보안성
- 신뢰성 (Reliability): 탐지율(오탐/미탐), 알림 전송 성공률

4.4.3. 테스트 케이스

TC ID	항목	시나리오	예상 결과	판정 기준
NFT-01	성능: 탐지 지연	1000건/초 SELECT 요청	평균 탐지 지연 ≤ 1초	SLA 충족
NFT-02	성능: 처리량	1000 TPS 부하	QueryLog 정상 기록 유지	TPS ≥ 1000
NFT-03	성능: 동시성	Dashboard 100명 동시 접속	응답 ≤ 2초	SLA 충족
NFT-04	보안: SQL Injection	`OR '1'='1'` 쿼리 실행	Event.Type=PATTERN	공격 탐지 성공
NFT-05	보안: 권한 제어	READ_ONLY 계정 → `UPDATE` 실행	Event.Type=AUTHZ	권한 위반 탐지
NFT-06	보안: 관리자 PW	Admin PW DB 저장 값 확인	SHA-256 해시 저장	평문 저장 금지
NFT-07	신뢰성: 탐지율	정상 1000건, 공격 50건 실행	≥ 950건 정상 처리, ≥ 49건 탐지	오탐 ≤ 5%, 미탐 ≤ 2%
NFT-08	신뢰성: 알림 성공률	Event 100건 발생	Slack/Email ≥ 99건 전송 성공	성공률 ≥ 99%

5. Item Pass/Fail Criteria (합격/실패 기준)

DB-IDS 테스트의 합격/실패 기준은 SRS에서 정의된 기능 및 비기능 요구사항을 기반으로 한다. 각 테스트 케이스는 예상 결과와 실제 결과가 일치할 경우 합격(Pass), 불일치할 경우 실패(Fail)로 판정한다.

5.1. 기능 요구사항 (FR) 기준

5.1.1. FR-1 (쿼리 로깅)

- 모든 SQL 쿼리(SELECT, INSERT, UPDATE, DELETE)가 로그에 기록되어야 한다.
- 로그에는 UserID, 실행 시간, SQL 원문/요약, RowCount, 실행 상태가 포함되어야 한다.
- 누락/손상 시 Fail

5.1.2. FR-2 (패턴 기반 탐지)

- SQL Injection, DROP TABLE, UNION SELECT 쿼리를 100% 탐지해야 한다.

- 탐지 이벤트가 즉시 생성되고 알림 전송이 발생해야 한다.
- 미탐 발생 시 Fail

5.1.3. FR-3 (행동 기반 탐지)

- 정의된 임계 조건(실행 ≥5s, 반환행수 ≥10,000, 초당 ≥100 쿼리)에서 반드시 이벤트를 생성해야 한다.
- Score ≥ 0.8일 경우 이벤트가 기록되지 않으면 Fail.

5.1.4. FR-4 (권한 기반 탐지)

- RBAC 정책 위반 시 DetectionEvent가 생성되어야 한다.
- 이벤트 누락 시 Fail

5.1.5. FR-5 (알림 기능)

- 이벤트 발생 후 Slack/Email 알림이 **10초 이내** 도착해야 한다.
- 실패 시 3회 재시도 후 실패 로그가 기록되어야 한다.
- SLA 위반 시 Fail

5.1.6. FR-6 (대시보드)

- 실시간 로그 5초 주기 갱신 및 정상 동작해야 한다.
- 이벤트 목록/상세 조회 ≤2초 응답해야 한다.
- 알림 설정 저장 정상 동작해야 한다.
- UI 오류/지연 시 Fail

5.1.7. FR-7 (로그 백업/아카이브)

- 하루 1회 이상 로그가 S3에 업로드 되어야 한다
- 실패 시 재시도 및 알림 발생해야 한다.
- 아카이브 실패 후 재시도 누락 시 Fail

5.2. 성능 요구사항 (PR) 기준

- 탐지 지연 시간: SQL 탐지 처리 ≤ 1초
- 대시보드 응답 시간: API 요청 후 ≤ 2초
- 알림 SLA: 이벤트 발생 후 알림 전달 ≤ 10초
- 처리량(Throughput): 초당 ≥ 1,000건 SQL 요청 처리 가능
- 동시성: 100명 이상의 동시 사용자 처리 가능

- 자원 사용: 쿼리당 메모리 ≤ 50MB, 일일 로그 ≥ 1GB 저장 가능
- 기준을 충족하지 못하면 Fail.

5.3. 신뢰성 및 보안 기준

- 가용성: 월 기준 99% 이상 유지
- 복구 시간(MTTR): 장애 발생 후 1시간 이내 복구
- 오탐(False Positive): ≤ 5%
- 미탐(False Negative): ≤ 2%
- 알림 전송 성공률: ≥ 99%
- 보안:
  - 모든 통신 HTTPS (TLS 1.2+) 보장
  - 관리자 비밀번호는 SHA-256 이상 해시로 저장
  - 관리자 계정 5회 이상 로그인 실패 시 잠금
- 위 조건 위반 시 Fail.

6. Test Deliverables (테스트 산출물)

본 프로젝트의 테스트 과정에서 생성되거나 제출되는 주요 산출물은 다음과 같다.

6.1. Test Plan (테스트 계획서)

- 본 문서(STP)
- 테스트의 범위, 전략, 일정, 합격/실패 기준을 정의한다.

6.2. Test Cases (테스트 케이스 문서)

- SRS/SDS 요구사항을 기반으로 작성된 케이스 목록
- 각 케이스는 입력 조건, 예상 결과, 실제 결과, 판정(Pass/Fail)을 포함한다.

6.3. Test Logs (테스트 실행 로그)

- 테스트 수행 시 실제 기록된 로그
- 성공/실패 결과, 발생한 오류, 환경 정보 등을 포함한다.
- 예: JMeter 성능 테스트 결과 리포트, Postman 실행 로그.

6.4. System Test Results (시험 결과 보고서)

- 전체 테스트 실행 결과를 집계한 문서

- 테스트 범위 대비 성공률, 주요 결함, 성능 지표 충족 여부, 리스크 평가를 포함한다.

7. Environmental Needs (테스트 환경 요구사항)

DB-IDS의 테스트 수행을 위해 필요한 환경은 다음과 같다.

7.1. 하드웨어 환경

- 테스트 서버
  - vCPU: 2 Core 이상
  - 메모리: 4GB 이상
  - 스토리지: 20GB SSD 이상 (로그 저장용)
  - 네트워크: 1Gbps 가상 NIC
- 클라이언트 환경
  - 일반 노트북/PC, 웹 브라우저 실행 가능

7.2. 소프트웨어 환경

- 운영체제: Ubuntu 20.04 LTS 이상
- DBMS: MySQL 8.0
- 내부 저장소: SQLite 3.35+
- 백엔드 프레임워크: Java 21 + Spring Boot 3.5+
- 프론트엔드 프레임워크: React 19+
- 알림 연동: Slack Webhook, SMTP 서버
- 아카이브 스토리지: Amazon S3

7.3. 네트워크 환경

- 테스트 서버와 MySQL 서버는 동일 VPC/네트워크 내에 배치
- 방화벽 규칙: 포트 3306(MySQL), 443(HTTPS), 587(SMTP) 개방
- 외부 서비스(Slack, S3)와 연결 가능한 인터넷 게이트웨이 필요

7.4. 테스트 도구

- 기능 테스트: Postman (API 호출 검증), 웹 브라우저(대시보드 확인)
- 성능 테스트: Apache JMeter (쿼리 TPS, 지연 시간 측정)
- 로그/결과 관리: 노션 (Test Case 및 Log 기록)

8. Schedule (테스트 일정)

단계	주요 활동	예상 기간	산출물
개발 완료	DB-IDS 기능 구현	~ 10/18	코드 (기능별 MVP)
테스트 실행	기능/성능/보안 테스트 수행, 로그 수집	10/19 ~ 10/21	Test Logs
결과 정리	결함 분석, 테스트 요약 보고서 작성	10/22	System Test Results
제출	최종 결과 보고서 제출	10/23	STP + STR

9. Risks and Contingencies (위험 및 대응)

본 테스트 계획의 성공적 수행에 영향을 줄 수 있는 주요 위험 요소와 대응 방안은 다음과 같다.

위험 요소	설명	대응 방안
개발 지연	기능 구현이 테스트 시작일(10/19)까지 완료되지 않을 경우, 테스트 범위 축소 불가피	핵심 기능(FR-1~FR-3) 우선 구현 후 테스트, 나머지는 문서 기반 시나리오로 보완
환경 구축 실패	VM/DB 세팅, Slack/SMTP 연동 등 환경 준비 지연	로컬 Docker 기반 임시 환경 사용, 외부 알림 연동은 Mock API로 대체
테스트 시간 부족	3일간 모든 기능 및 성능 테스트 수행에 어려움	우선순위 높은 기능(로그/탐지/알림) 위주로 실제 실행, 대시보드 UI는 캡처 중심으로 검증
결함 미해결	테스트 중 발견된 버그를 즉시 수정하기 어려움	System Test Results에 결함 리스트 및 영향도 명시, 개선 계획 첨부
성능/보안 미흡	TPS, 탐지율 목표에 미달 가능성	실제 수치 대신 MVP 수준 검증 + 향후 개선 방안으로 보고서에 기술