출입 통제 시스템: 폐쇄망 HTTPS 통신 설정 가이드

본 문서는 **Ingress 컨트롤러(Nginx/Traefik)**를 사용하여 **HTTPS/SSL Termination**을 구축하는 상세한 3단계 절차를 설명합니다.

폐쇄망 환경에서는 Let's Encrypt와 같은 공인 인증 기관을 사용할 수 없으므로, **"사설 인증 기관(Private CA)"**을 직접 구축하여 인증서를 발급해야 합니다.

1. HTTPS 통신의 목적

`API Gateway`로 전송되는 모든 트래픽을 암호화하는 것은 필수 보안 조치입니다.

1. **관리자 트래픽:** `Admin UI`에서 전송하는 관리자 ID/PW 및 `JWT 토큰`을 보호합니다.
2. **단말기 트래픽:** `IP 리더기`가 전송하는 고유 `Device Token`(비밀 키)을 네트워크 도청(Sniffing)으로부터 보호하여, 시스템 보안이 뚫리는 최악의 상황을 방지합니다.

2. 1단계: 사설 인증 기관(Private CA) 생성 및 서버 인증서 발급

`https://api.sdac.local` (내부 도메인)에서 사용할 SSL 인증서를 자체적으로 발급합니다. 이 작업은 `Node 1` 등 관리자 권한이 있는 서버에서 `openssl` 도구를 사용하여 1회 수행합니다.

A. 사설 Root CA 생성

우리만의 "인증서 발급 기관"을 만듭니다.

```
# 1. Root CA의 개인 키(비밀 키) 생성
openssl genrsa -out MyPrivateCA.key 4096

# 2. Root CA 인증서(공개 키) 생성 (예: 10년 유효)
openssl req -x509 -new -nodes -key MyPrivateCA.key -sha256 -days 365
0 -out MyPrivateCA.crt
# (이 과정에서 "Organization Name: SDAC Corp" 등 내부 CA 정보를 입력합니다)
```

- **결과물:** `MyPrivateCA.key` (CA의 비밀 키, 절대 외부에 노출 금지), `MyPrivateCA.crt` **(CA의 공개 인증서, 모든 클라이언트에 배포해야 함)**

B. 서버 SSL 인증서 발급 (Internal Domain 용)

위에서 만든 `MyPrivateCA`를 사용하여 `api.sdac.local` 도메인용 인증서를 발급(서명)합니다.

```
# 1. 서버의 개인 키(비밀 키) 생성
openssl genrsa -out api.sdac.local.key 2048

# 2. 서버 인증서 서명 요청(CSR) 생성
openssl req -new -key api.sdac.local.key -out api.sdac.local.csr
# (이 과정에서 "Common Name (CN): api.sdac.local"을 정확히 입력해야 합니다)

# 3. (중요) MyPrivateCA로 서버 인증서 서명 (예: 2년 유효)
openssl x509 -req -in api.sdac.local.csr -CA MyPrivateCA.crt -CAkey MyPr
ivateCA.key -CAcreateserial -out api.sdac.local.crt -days 730 -sha256
```

- **결과물: `api.sdac.local.crt` (서버 인증서), `api.sdac.local.key` (서버 비밀 키)**

3. 2단계: Ingress 컨트롤러(Nginx) 설정 (SSL Termination)

`sdac-execution-platform-guide.md`의 `4.D` 역할입니다. 1단계에서 만든 인증서를 사용하여 HTTPS 트래픽을 받고, 암호화를 해제(Termination)한 후 내부 `API Gateway`로 HTTP 트래픽을 전달합니다.

A. 인증서 저장

1단계에서 생성한 `api.sdac.local.crt`와 `api.sdac.local.key` 파일을 `Node 1, 2, 3`의 특정 경로(예: `/etc/ssl/certs/`) 또는 `Docker Secret`으로 안전하게 복사합니다.

B. Nginx 설정 (`nginx.conf` 예시)

`Ingress Nginx` 컨테이너는 이 설정을 사용합니다.

```
server {
    # 1. HTTPS 포트(443)에서 수신
    listen 443 ssl http2;
    listen [::]:443 ssl http2;

    # 2. 내부 도메인 이름 설정
    server_name api.sdac.local;
```

```
    # 3. 1단계에서 발급한 사설 인증서 경로 지정
    ssl_certificate /etc/ssl/certs/api.sdac.local.crt;
    ssl_certificate_key /etc/ssl/certs/api.sdac.local.key;

    # 4. (SSL Termination)
    #    여기서 HTTPS 암호화가 종료됩니다.

    location / {
        # 5. 내부망으로는 암호화가 풀린 HTTP로 전달
        #    'api-gateway'는 sdac-final-architecture.md에 정의된
        #    Spring Cloud Gateway의 Docker Swarm 서비스 이름입니다.
        proxy_pass http://api-gateway:8080;

        # (필수 헤더 설정)
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
    }
}

# (선택 사항) HTTP → HTTPS 자동 리다이렉트
server {
    listen 80;
    server_name api.sdac.local;
    return 301 https://$server_name$request_uri;
}
```

C. API Gateway (Spring Cloud) 설정

`API Gateway` 서비스는 SSL을 전혀 신경 쓰지 않습니다. Nginx로부터 암호화가 해제된 `HTTP/8080` 트래픽을 받도록 설정되어야 합니다.

4. 3단계: 클라이언트 신뢰 설정 (필수 작업)

이 단계가 없으면 폐쇄망 HTTPS 통신이 실패합니다. **모든 클라이언트**가 1단계에서 만든 **`MyPrivateCA.crt` (Root CA 인증서)**를 신뢰하도록 설정해야 합니다.

- **A. `Admin UI` 관리자 PC:**

    - 관리자 PC(Windows, macOS)의 **"신뢰할 수 있는 루트 인증 기관"** 저장소에 `MyPrivateCA.crt` 파일을 **수동으로 설치**해야 합니다.
    - 설치 후, 관리자는 웹 브라우저에서 `https://api.sdac.local` 접속 시 "안전함" 녹색 자물쇠 아이콘을 볼 수 있습니다.
- **B. `IP 리더기` / `ACU` (단말기):**
    - (가장 중요) `sdac-terminal-layer-guide.md`의 `옵션 1` 또는 `옵션 2`의 단말기 하드웨어는 **SSL 통신을 지원해야** 합니다.
    - 단말기(리더기)의 관리자 웹페이지 설정에, `MyPrivateCA.crt` **(Root CA) 인증서를 업로드**하는 기능이 반드시 있어야 합니다.
    - 이 CA 인증서를 업로드해야만, 리더기가 `https://api.sdac.local`에 접속 시 "신뢰할 수 있는 서버"로 인지하고 `Device Token`을 안전하게 전송합니다.