e

H_SCOPE

```
목표 : 앱의 위,변조를 탐지하는 무결성 검증 과정을 앱에 추가 및 검증
```

개발 환경 및 사용 툴

```
OS : Linux (WSL v1.2.5.0)
tool : apktool (v2.4.1, with smali v2.3.4 and baksmali v2.3.4)
target : 2048.apk (v 1.3.3, tpcstld.twozerogame, 안드로이드 플레이스토어에서 추출)
lang : C, JAVA
SDK : min 14 ~ max 28
```

23.05.17

- 데모 앱 : genOTP.apk
- 진행 상황 : 메인에서 애플리케이션 이름을 입력하면 H register origin app.c 에서 해당 앱을 열고 해시값을 구한다.
- Application class smali 주입 시 고려사항
    1. 이미 Application class를 상속받는 클래스, 메소드가 존재할 수 있다
        - 순서를 바꿔주는 과정이 필요
    2. AndroidManifest.xml 파일 또한 수정이 필요.
        - 자동화를 진행한다면 AndroidManifest.xml의 패턴마다 파싱 과정이 달라져야 한다
    3. SDK 버전 체크
        - 리컴파일(리패키지) 과정에서 SDK 버전마다 컴파일러가 달라 고려해야한다

23.05.18

- 진행 상황
    - 리눅스 환경에서 apktool 2.5.0 를 설치하고 디컴파일, 리컴파일, 리사이닝을 테스트 했다. 앱 수정 없이 디컴파일, 리컴파일, 리사이닝 과정을 거쳐도 원본과 다른 해시값을 가진다.

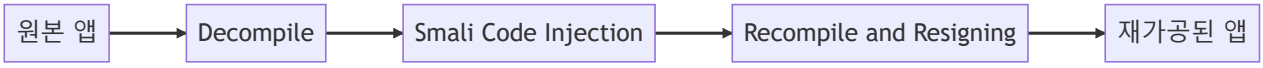
      위 다이어그램 순서로 실행 시 리컴파일, 리사이닝 과정에서 키도 바뀌고 앱 해시값도 달라진다. 리컴파일, 리사이닝 이후 과정의 해시값을 구할 순 있지만, 이후 과정의 해시값은 원본에 대한 해시값이 아니기 때문에 원본 무결성의 성격이 아니게 된다.

      새롭게 고안해본 방법은 최초 클라리언트가 의뢰를 할 때, 원본 앱과 원본 앱의 해시값을 같이 주는 것 이다. 해시 알고리즘은 상호 약속이 되어있다는 조건이다. 원본 앱과 원본 앱의 해시값을 같이 준다면 서버는 받은 앱의 해시값을 새로 구하고 받은 해시값과 비교한다.

위 과정이 추가된다면, 최초 과정에서 클라이언트가 전송하는 앱이 원본임을 증명할 수 있다. 이후 서버에서 받은 앱 또한 원본임을 증명할 수 있다. 이후 리컴파일, 리사이닝 과정 후 해시값을 저장한 뒤, 실행 시 비교한다면 무결성을 검증 할 수 있을 것이다.

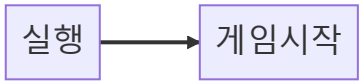
23.05.19

- 앱 재가공 과정

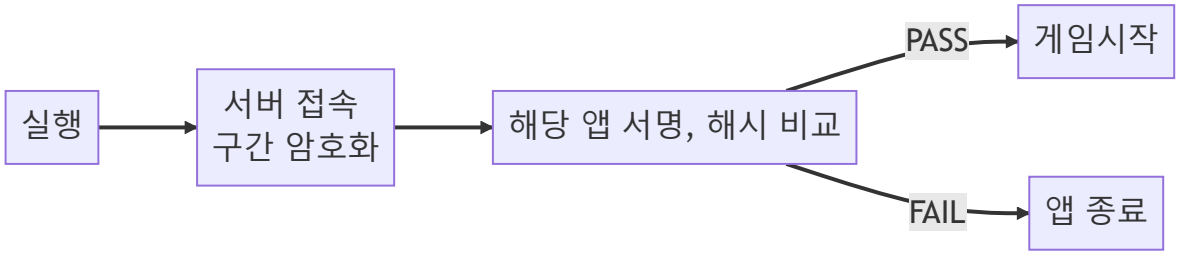


- 원본 앱과 재가공 앱 실행 순서 비교

1. 기존 앱



2. 재가공 앱



- 진행 상황
  - 리눅스에서 사용하는 apktool과 윈도우에서 사용하는 apkeasytool, 두 환경에서 디컴파일 및 리컴파일을 진행해보았다. 리눅스에서 사용하는 apktool을 통해 디컴파일을 한 경우 AndroidManifest.xml 파일이 윈도우에서 열리지 않았다. 운영체제가 달라 열리지 않는 듯 했다.

    프로젝트 주제에 걸맞게 전제조건과 내가 수행할 부분의 선을 명확히 했고 총 과정의 절차가 수행되는지 우선적으로 테스트 해보는 것을 목적으로 삼았다.

    타켓 APP에 토스트 메세지를 띄우는 스말리 코드를 주입해보았다. 추후 소켓으로 연결 할 라이브러리와 JNI를 만들어 주입하거나, 자바에서 소켓을 열어 무결성 검증을 할 서버에 접속을 테스트 해볼것이다.

23.05.22

- 진행상황
  - 리눅스 apktool 문제 해결, 버전 문제였음
  - 앱 재공정 과정 이후 재공정 앱 해시, 재공정 앱 키 해시 데이터를 저장
  - 재공정 시 데모 기준으로 작성된 신규 smali 코드와 수정된 Androidmanifest.xml 파일 주입
  - 주입후 재공정 마친 앱 정상 작동 확인 ("this is test" 문구 토스트)

- 재공정 과정

    1. 실행 전 대상 앱이 올바른 경로에 위치해야함

    2. DATABASE 텍스트 파일 생성 및 초기화

    3. H_solution 진행

        1. 대상 앱 디컴파일
        2. 디컴파일 경로에 준비된 코드 주입
        3. 대상 앱 리컴파일
        4. 대상 앱 리사이닝

    4. 생성된 재공정 이후 앱의 해시값, 재공정 앱의 키 해시값을 DATABASE 텍스트 파일에 저장

- 문제점

    - 리사이닝에 사용하는 서버의 키스토어가 셀프사인이기 때문에 앱 설치시 경고 문구가 뜸
    - 재공정 이후 앱의 키 값을 어떻게 가져와야 할지 고민중