[한양대학교 ERICA System&IoT Security Lab]

**Android Reversing Study 학습 내용 정리 노트**

성명 : 지훈

1. 학습 내용 요약  
     
   - 안드로이드 리버스 엔지니어링의 개념과 환경 설정, 시작 방법에 대해 학습  
   - 말웨어 리버스 엔지니어링의 시작점 찾기와 리버싱 방법  
   - 안드로이드 취약점 리버싱에서의 시작점 찾기와 방법  
   - 바인드를 사용하는 앱에서의 리버싱의 시작점 찾기와 방법
2. 학습 내용 세부설명

Android 애플리케이션 : APK 파일 형식 - ZIP 파일

APK 콘텐츠

1. AndroidManifest.xml

2. META-INF : 인증서 존재

3. classes.dex

애플리케이션용 Dalvik 바이트코드

실행할 JAVA or KOTLIN 코드

4. lib/

기본 라이브러리, CPU 관련 디렉토리

5. assets/

앱에 필요할 수 있는 기타 파일

추가 기본 라이브러리 및 DEX 파일

기본 위치에 추가 코드를 포함하지 않음

-> Dalvik의 추가 코드를 '숨기기' 할 때 발생

Android Java -> DEX(Dalvik Executable) 바이트 코드 형식

==> 이전 버전의 자바 : Davlik 가상 머신

==> 최신 자바 : Android 런타임(ART)

Java or Kotlin -> DEX Bytecode

Dex Bytecode -> SMALI (어셈블리어) -> Decompiled Java

디컴파일된 Java 출력 의심 -> Smali 출력 검사

DEX -> Smali : baksmali 도구(디스어셈블러)

Smali -> DEX : Smali 도구

애플리케이션 진입점

1. Launcher Activity

2. Services

UI 없이 백그라운드 실행

Intents가 기본 방법

--> startServiceAPI 호출 -> onStart 실행

3. Broadcast Receivers

메시징 시스템

수신기 등록

1. 메니페스트

2. registerReceiver() API 호출

4. Exported Componenets

"내보내기" 된 서비스 및 액티비티

Reverse의 시작점 설정을 위한 방침

1. 목표 설정

2. API 호출

관심 있는 동작에 필요한 API를 찾기

3. 앱 진입점

실행할 수 있는 코드에 집중

4. 암호 해독 방법

뒤죽박죽 섞인 문자열이나 이진 배열이 동일한

메서드에 전달되는지 확인하기

앱의 취약점을 찾을 때

Runtime.exec() 검색

ProcessBuilder() 검색

Native code:system() 검색

바인더를 사용하는 경우

바인더를 기준으로 바인더가 정의한 메소드를 따라감

1. 진행한 실습 내용  
     
   - 직접 Premium 메시지를 보내는 코드를 찾아내는 실습  
   - 애플리케이션의 취약점을 찾는 코드 실습  
   - 바인드를 이용한 앱을 리버싱하는 실습
2. 새로 배우게 된 점  
     
   - 기본적인 안드로이드 리버싱을 시작하는 방법에 대해 강조했고, 그것을 잘 숙지할 수 있었음  
   - 목적, 그리고 앱의 특성마다 리버싱을 시작하는 시작점이 다르다는 것을 배움  
   - 구체적으로 어떤 메커니즘을 따라 리버싱이 진행되는지에 대해 배울 수 있었음.