안드로이드 악성앱 언패커 제작

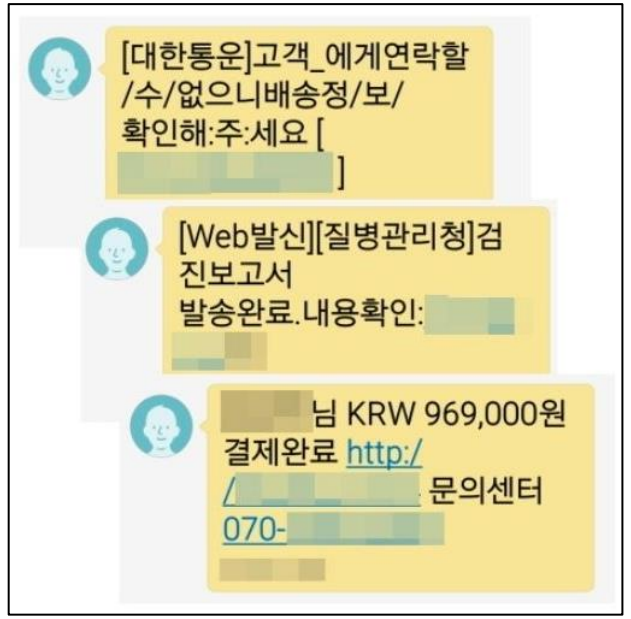
- 중간 발표 -

서동훈, 전유민

Contents

프로젝트 배경

질병관리청 사칭



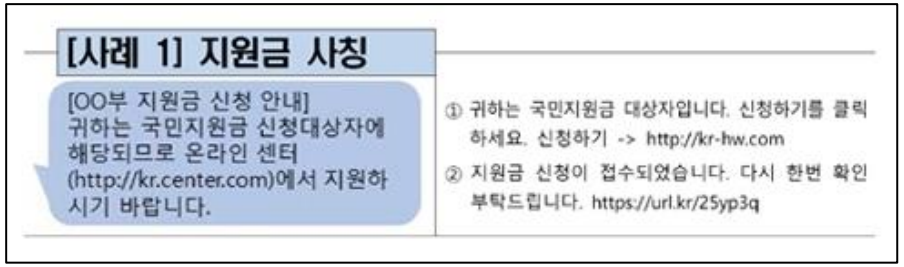
택배 사칭

서울경찰청 위장 악성 앱 설치 유도하는 스미싱 유포

보이스피싱 예방앱 '피싱아이즈' 위장 악성앱 설치 유도... 사용자 권한 획득

[보안뉴스 원병철 기자] 최근 서울지방경찰청을 위장한 스미싱이 유포되고 있어 사용자들의 주의가 요구된다. 이스트시큐리티 ESRC(시큐리티 대응센터)는 이번 스미싱은 보이스피싱 예방앱인 '피싱아이즈'를 위장한 악성 앱을 활용해 스마트폰의 권한을 요구한다고 밝혔다.

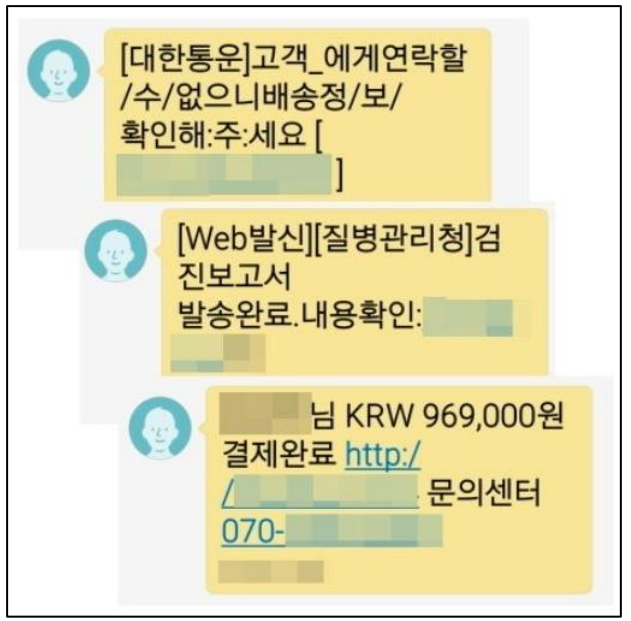
경찰청 위장 앱



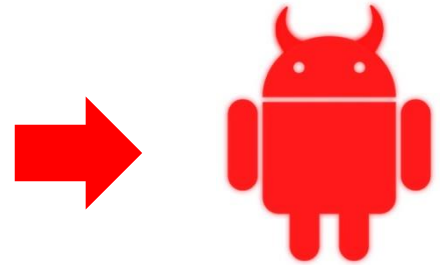
지원금 사칭

프로젝트 배경

질병관리청 사칭



택배 사칭



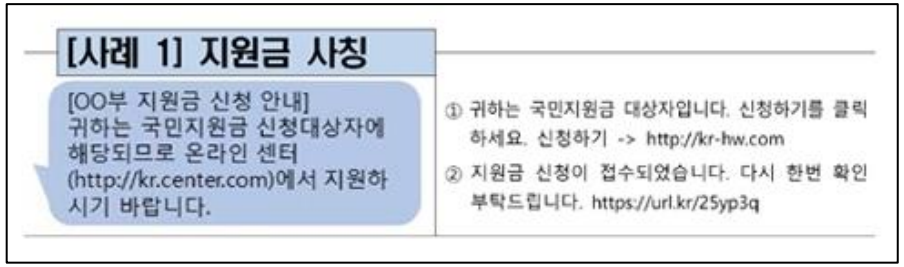
악성 앱

서울경찰청 위장 악성 앱 설치 유도하는 스미싱 유포

보이스피싱 예방앱 '피싱아이즈' 위장 악성앱 설치 유도... 사용자 권한 획득

[보안뉴스 원병철 기자] 최근 서울지방경찰청을 위장한 스미싱이 유포되고 있어 사용자들의 주의가 요구된다. 이스트시큐리티 ESRC(시큐리티 대응센터)는 이번 스미싱은 보이스피싱 예방앱인 '피싱아이즈'를 위장한 악성 앱을 활용해 스마트폰의 권한을 요구한다고 밝혔다.

경찰청 위장 앱

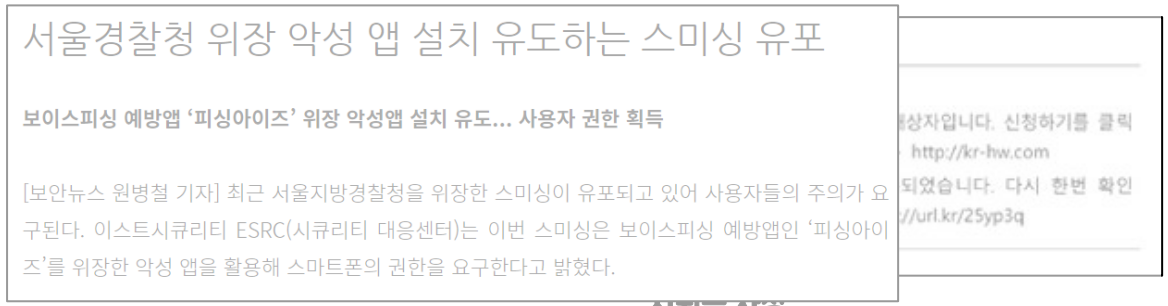


지원금 사칭

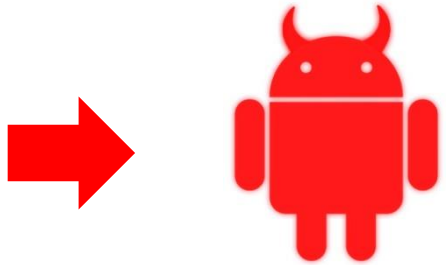
프로젝트 배경

질병관리청 사칭

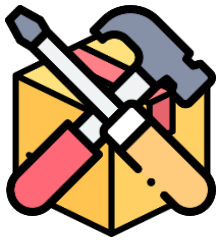
택배 사칭



경찰청 위장 앱



악성 앱



패킹

프로젝트 계획

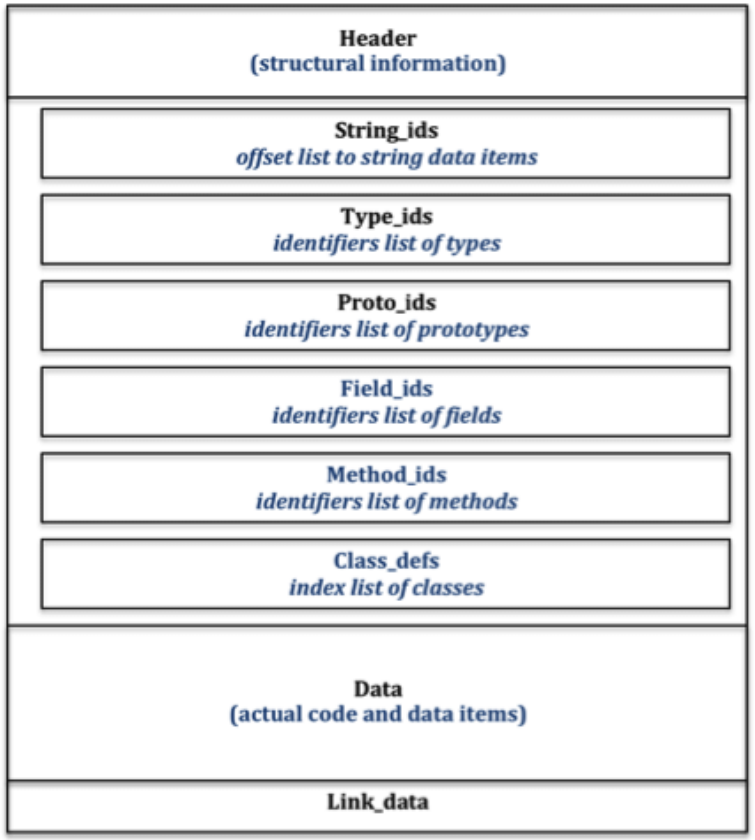
주차	1 주차	2 ~ 3 주차	4 ~ 6 주차	산출물
준비 단계	자료 조사			
분석 단계		패커 및 패킹앱 분석		분석 보고서
			언패커 제작	언패커 도구

안드로이드 패킹

패킹 (Packing)

- 역공학과 같은 피해를 방지하고자 원본 덱스를 암호화하여 리소스에 포함하고 원본 덱스 자리에는 복호화를 수행하는 코드를 대신 넣어주는 것

안드로이드 패킹

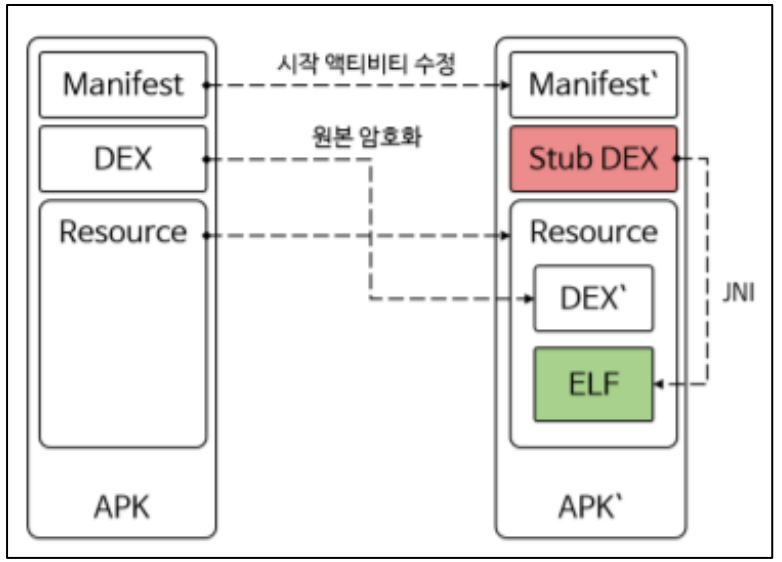


Dex file structure

- Dex 구조는 이미 공개되어 있다.
- 바이트코드로 구성되어 있어 역공학에 취약
- 난독화로 보완하였지만 현재는 쉽게 파회 가능

안드로이드 패킹

- 구현에 따라 원본 덱스의 위치는 바뀐다.
- 스텁 덱스도 결국은 바이트 코드이기 때문에 역공학에 취약
- 이를 보완하기 위해 JNI를 사용하여 네이티브 단계에서 진행
- 런타임 보호 기법 실행 -> 원본 덱스 복호화 -> 복호화된 덱스를 통해 클래스 로더 생성 -> 스텁 클래스 로더를 원본 클래스 로더로 치환 -> 원본 클래스 정보를 불러온 후 기존의 흐름대로 진행



기본적인 패킹앱 구조

Ref. Android Code Protection via Obfuscation Techniques:Past, Present and Future Directions

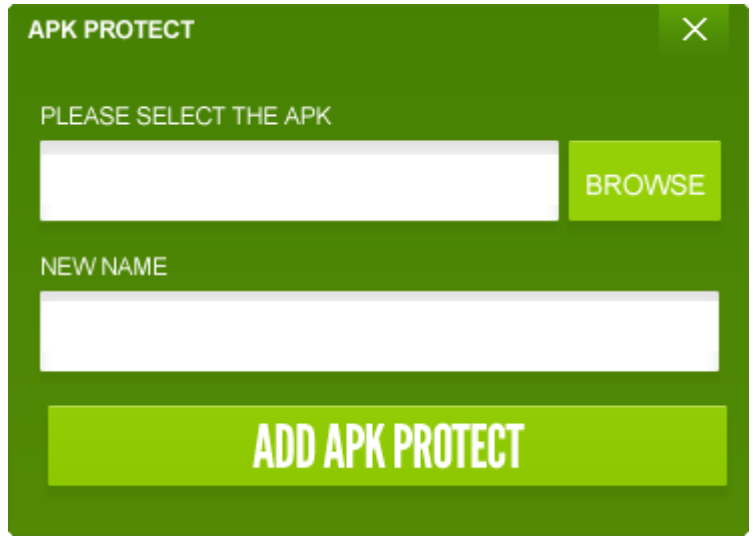
패커 종류

Packer Protection Techniques								
Packer	Code Obfuscation	Dynamic Code Loading	Dynamic Code Modification	Debugger Detection	Append shared Libraries	Additional Class insertion	DVM Support	ART Support
APKProtect	✔	✔	✔	✔	✔	✔	✔	✗
Ali	✔	✔	✔	✔	✔	✔	✔	✗
Baidu	✔	✔	✔	✔	✔	✔	✔	✔
Bangle	✔	✔	✗	✔	✔	✔	✔	✔
Ijiami	✔	✔	✗	✔	✔	✔	✔	✔
HoseDex2jar	✔	✔	✗	✔	✗	✗	✔	✗
Pangxie	✔	✗	✗	✗	✗	✔	✔	✗

패커 종류

APK Protect : APK Protect에서 개발한 제품으로 안티 디버그, 안티 디컴파일, 안티 디스어셈블리를 지원하는 도구이다.

- 패킹 방법 : Base64 인코딩을 통해 문자열을 암호화하여 코드 난독화를 수행한다.
- 기능
  - 디버거 탐지
  - 앱 암호화
  - 코드 리플렉션
  - 안티 디버깅
  - 안티 디스어셈블리
  - 안티 디컴파일



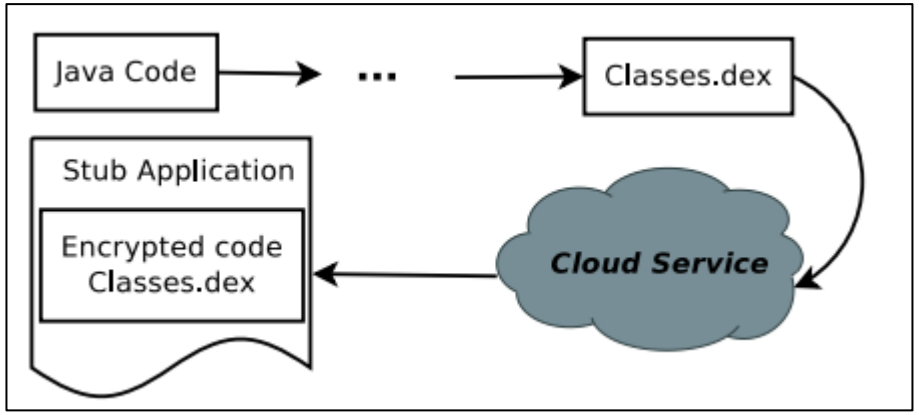
APK Protect 실행 화면

Ref. Android Code Protection via Obfuscation Techniques:Past, Present and Future Directions

패커 종류

HoseDex2Jar

- 패킹 방법 :.dex를 암호화하고 암호화된 파일을 다시 패키징한 후 해당 패키지 안에 112 바이트 크기의 헤더를 포함시켜 실행하는 패커
- 기능
  - .dex 리패키징
  - 코드 암호화
- https://github.com/strazzere/dehoser



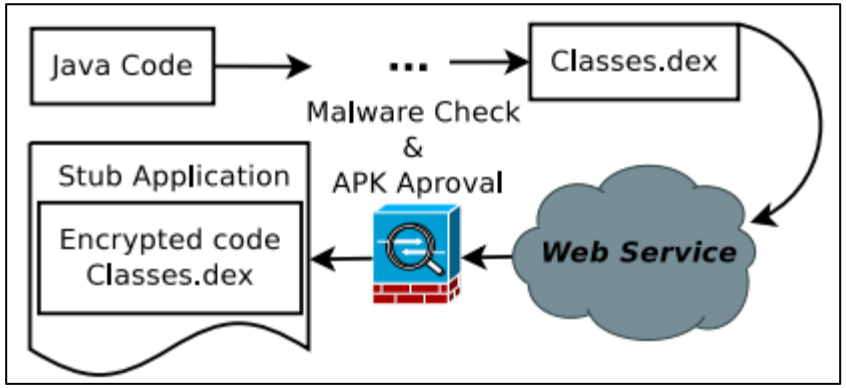
HoseDex2Jar 패킹 과정

Ref. Android Code Protection via Obfuscation Techniques:Past, Present and Future Directions

패커 종류

Bangcle : Bang Bang Security에서 개발한 온라인 APK 패킹 도구

- 패킹 방법 : 개발자는 Bangcle에 등록하고 관련 도구를 사용하여 업로드하면 업로드된 패키지는 APK 이름을 변경하고, 새로운 assets, 네이티브 라이브러리를 삽입하여 Manifest를 수정하는 식으로 패킹 작업이 진행된다.
- 기능
  - 온라인 APK Wrapping
  - 리버스 엔지니어링 저항
  - 안티 디버깅
  - 변조 방지
  - 디컴파일 방지
- https://www.bangcle.com/



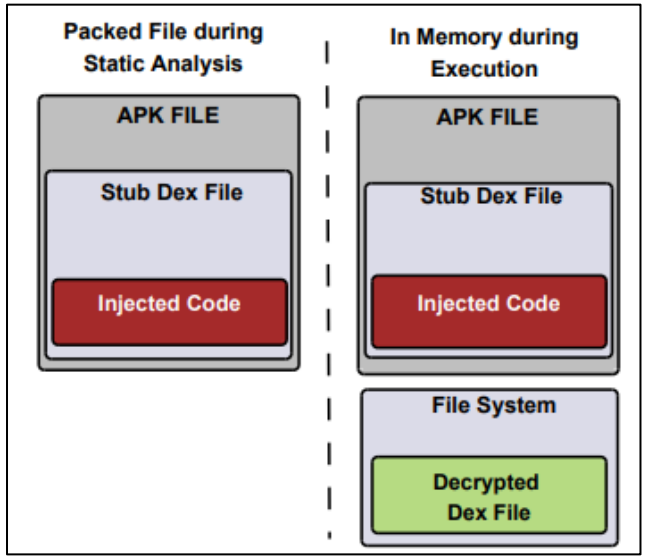
Bangcle 패킹 과정

Ref. Android Code Protection via Obfuscation Techniques:Past, Present and Future Directions

패커 종류

PANGXIE : 안티 디버깅과 안티 템퍼링 기술을 지원하는 PoC 패커

- 패킹 방법 : APK 내부 번들로 제공되는 .dex를 암호화한다.
- 기능
  - 안티 디버깅
  - 변조 방지



런타임 실행에 따른 코드 언패킹 과정

Ref. Android Code Protection via Obfuscation Techniques:Past, Present and Future Directions

패커 종류

ijiami : iJiami에서 제공하는 앱 보안 기능

- 기능
  - 안티 리버싱
  - 변조 방지
  - 안티 디버깅
  - 데이터 유출 방지
  - 페이지 데이터 보호
  - 데이터 전송 보호
- https://www.ijiami.cn/enappProtect

패커 종류

왜 상용화된 패커의 조사 및 분석을 진행하는가..?

패커 종류

악성 앱에서도 상용화된 패커 혹은 관련 서비스를 사용하여 보안 탐지를 우회하거나 분석을 힘들게 만든다.

스미싱과 보이스피싱 등을 결합한 형태로 악성 앱들이 유포되고 있습니다. 해당 앱들은 주로 1 금융 관련 앱을 사칭하였으나 최근에는 국가기관, 2 금융 사칭 등으로 다양한 형태로 나타나고 있습니다. 기기 및 개인정보를 탈취하고 금융 정보 탈취를 목적으로 기기의 통화 상태를 감시합니다.

특히, 해당 앱은 분석을 어렵게 하기 위해서 중국 Qihoo 360사의 패킹 기술을 적용하였습니다.

본 분석 보고서에서는 "Trojan.Android.KRBanker"를 상세 분석하고자 합니다.

악성코드 상세 분석

1. 패킹 특징

중국 Qihoo 360사의 패킹 된 앱은 일반 앱과는 다른 부분이 존재합니다. 앱의 권한과 컴포넌트 관련 정보를 볼 수 있는 매니페스트를 보면 일반 안드로이드 앱에서는 볼 수 없는 항목인 "android:qihoo"부분이 추가되어 있는데 이는 디컴파일을 방해합니다. "assets" 폴더에는 파일의 무결성과 동적 패킹에 관여하는 ".appkey", "libjiagu.so" 파일이 포함되어 있습니다. 또한, 아래 [그림 2]를 보면 패키지명이 "com.android.hellod3"이지만, 패킹 된 덱스 코드에서는 해당 부분을 찾을 수 없어 정적 분석으로는 실제 악성 행위와 관련된 코드를 볼 수 없습니다.

이스트시큐리티에서 작성한 악성코드 분석 보고서

Ref. https://blog.alyac.co.kr/1994

자료 정리

패커

Packer Protection Techniques								
Packer	Code Obfuscation	Dynamic Code Loading	Dynamic Code Modification	Debugger Detection	Append shared Libraries	Additional Class insertion	DVM Support	ART Support
APKProtect	✔	✔	✔	✔	✔	✔	✔	✗
Ali	✔	✔	✔	✔	✔	✔	✔	✗
Baidu	✔	✔	✔	✔	✔	✔	✔	✔
Bangle	✔	✔	✗	✔	✔	✔	✔	✔
Ijiami	✔	✔	✗	✔	✔	✔	✔	✔
HoseDex2jar	✔	✔	✗	✔	✗	✗	✔	✗
Pangxie	✔	✗	✗	✗	✗	✔	✔	✗

Table III: Comparing packer protection methods

- APK Protect : APK Protect에서 개발한 제품으로 anti-debug, anti-decompile and anti-disassembly를 지원하는 도구이다. 패커는 Base64 인코딩를 통해 문자열을 암호화 하는 코드 난독화를 수행한다. 해당 과정은 자바 리플렉션(Java reflections)을 사용하여 동적으로 코드를 가져온다.
  - 기능
    - 디버거 탐지 (Debugger detection)
    - 앱 암호화 (App encryption)
    - 코드 리플렉션 (Code reflections)
    - 안티 디버깅 (Anti-debugging)
    - 안티 디스어셈블리 (Anti-disassembly)
    - 안티 디컴파일 (Anti-decompilation)

  > 개발사 홈페이지 링크가 http://www.apkprotect.com/ 인데 접속 시 도메인 호스팅 관련 업체 사이트로 이동하는 것 보니 아마 현재는 없는 회사인듯

- HoseDex2Jar : dex를 암호화하고 암호화된 파일을 다시 패키징하고, 해당 패키지 안에 112 크기의 헤더 바이트를 포함시켜 실행하는 패커이다.
  - 기능
    - .dex 재포장 (.dex repackaging)

배경

Android Packing

What is Packing..?

패킹에 대해서 조사해보면 각자 살짝은 차이가 있는 의미로 사용하고 있는 모습을 확인할 수 있다.

- 배포를 하기 위해 파일의 크기를 줄이거나 보안성을 높이기 위해 실행파일을 압축하는 기술
- 실행 파일을 압축하고 압축된 데이터를 압축 해제 코드와 합쳐 하나의 실행 파일로 만드는 기술

이와 같이 상황마다 살짝 차이가 있지만 다르게 사용을 하고 있지만 대부분은 두 가지(데이터의 압축, 보안 강화) 의미를 포함하고 있다. 데이터 압축은 말 그대로 데이터를 압축하여 크기를 줄이는 것에 의미를 가지고 있으며, 보안 강화는 역공학을 통해 받을 수 있는 피해를 방지하는 것을 의미한다.

그렇다면 안드로이드에서는 왜 패킹이 필요할까?

이를 알기 위해서 먼저 안드로이드 앱 구조에 대해 알아낼 필요가 있었다.

Android APK Structure

APK(Android Application Package)는 안드로이드의 소프트웨어와 미들웨어 배포에 사용되는 패키지 파일이며, '.apk'확장자를 가진다. 또한 ZIP파일 기반인 JAR을 기반으로 하는 압축 파일의 종류중 하나이다.

이러한 압축 파일인 APK는 다음과 같은 구조를 가지고 있다.

- AndroidManifest.xml : 애플리케이션에 대한 주요 정보(패키지 이름, 애플리케이션 구성 요소, 실행하는데 필요한 권한 및 액세스시 필요한 권한, 호환성)가 포함되어 있다.
- META-INF : 인증 정보가 포함된 폴더
- assets : 앱 실행에 필요한 자원이 모여있는 디렉토리로 주로 동영상, 일부 문서 템플릿와 같이 용량이 큰 데이터를 가지고 있으며, 빌드가 되지 않는다.
- res : 앱 실행에 필요한 자원이 모여있는 디렉토리로 빌드시 설치 파일에 포함되어 설치된다.
- kotlin : 애플리케이션이 코틀린으로 작성된 경우만 생성되며, 코틀린과 관련된 데이터가 포함되어있다.
- lib : 라이브러리 파일이 저장되어 있는 디렉토리
- resources.arsc : 컴파일된 리소스를 포함한다.
- classes.dex : Dalvik이 인식할 수 있도록 자바로 짜여진 코드가 컴파일되어 바이트 코드로 변환 된

노션을 통해서 지속적으로 진행 과정 작성하여 정리중

향후 계획

주차	1 주차	2 ~ 3 주차	4 ~ 6 주차	산출물
준비 단계	자료 조사			
분석 단계		패커 및 패킹앱 분석		분석 보고서
			언패커 제작	언패커 도구

Q & A