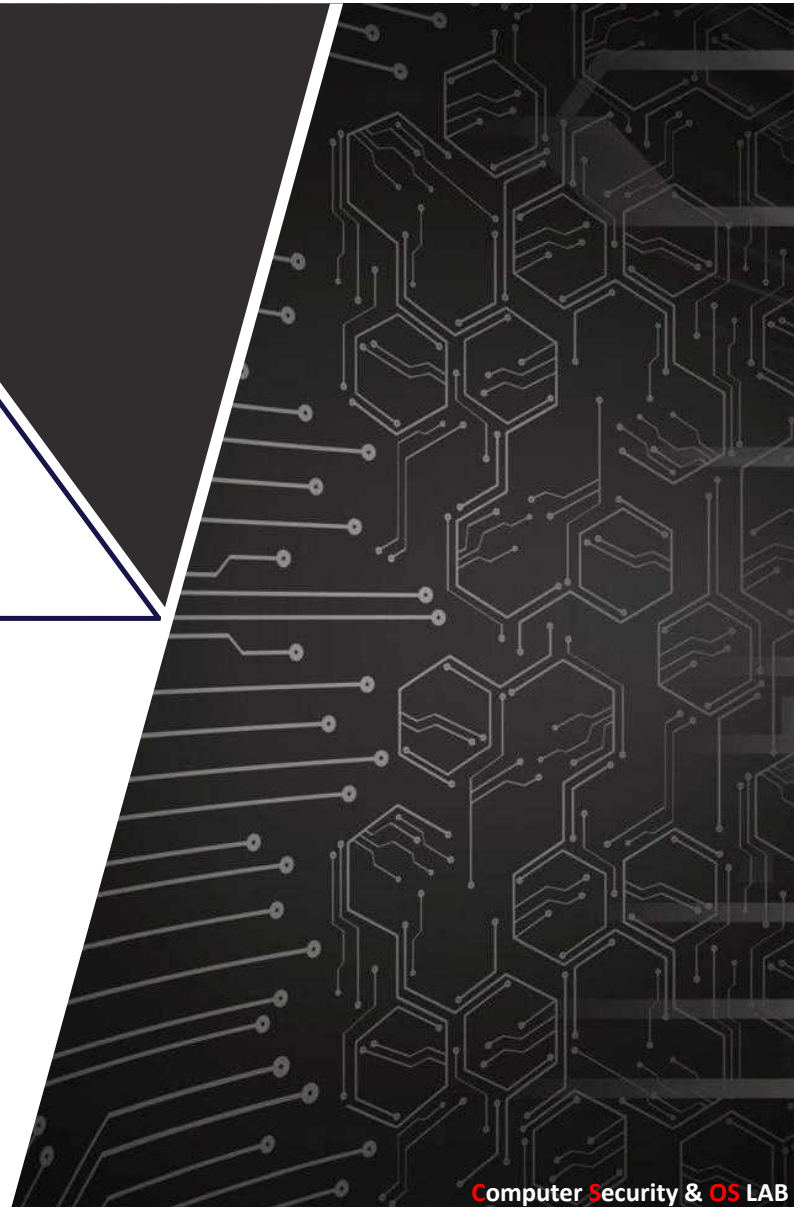
# 루팅 없이 갤럭시 S21 App Data 추출 연구

2024.05.10(금)

모바일시스템공학과 20학번 조민혁




DKU 단국대학교 DANKOOK UNIVERSITY

# INDEX

# 01 연구 계기 및 목표

# 연구 계기

## ❖ 계기

* 5월 1일 Android에서 App Data가 저장되는 방식과 추출에 필요한 명령어를 발표하는 시간을 가짐

* 발표 후, 루팅 없이 App Data를 추출할 수 있는 지를 연구해보라는 피드백을 받음

* 또한, 루팅 없이 App Data를 추출하는 것은 연구실에서 지속적으로 고민해오던 부분임

* 이를 통해, 루팅 없이 App Data를 추출하는 것에 대해 연구를 하기로 결정

# 연구 목표

## ❖ 목표

-> " 루팅 과정 없이 갤럭시 S21의 'Bluelink App Data' 를 추출하는 것"

## ❖ 갤럭시 S21에 대한 정보

* 기종 : Galaxy S21 5G

* Android Version : Android 14

=> 즉, Android 14 수준에서의 App Data 추출을 목표로 함

# 02 App Data 추출 과정

# App Data 추출 과정(1)

## ❖ 고려 사항

1. Android는 App Data를 저장하는 방식이 존재함

* 외부 저장소 : /Android/data/패키지명

* 내부 저장소 : /data/data/패키지명

2. 사용자 정보를 추출해야하기 때문에 내부 저장소로 접근할 필요가 있음

# App Data 추출 과정(2)

## ❖ 루팅이 없을 때 어려운 점

1. 'adb'를 통해 shell에 접속하여 /data 디렉터리에 접근 후 'ls', 'cd', 'cp' 등 많은 파일 관련 명령어가 'Permission Denied' 되어있음

2. 'adb backup' 명령어를 통해 패키지 추출 시 1KB의 쓰레기 값이 발생함

-> 기존 'adb backup' 기능의 보안 취약점을 알고 막아둔 것으로 추정


3. 다이렉트로 패키지 명을 알 수 있기가 쉽지 않음

# App Data 추출 과정(3)

## ❖ Step1) 패키지 명 파악하기

* 'adb shell pm list packages' 명령어 이용

-> 'pm'은 '패키지 관리자'로써 패키지 관련 작업에 도움을 줌

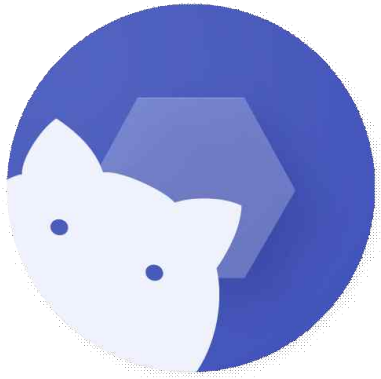
```
PS C:\Users\cgumg> adb shell pm list packages
package:com.sec.android.RilServiceModeApp
package:com.samsung.oda.service
package:com.google.android.overlay.modules.permissioncontroller.forframework
package:com.sec.android.iaft
package:com.skt.massivear
package:com.samsung.android.providers.trash
package:com.samsung.android.app.telephonyui.esimclient
package:com.samsung.android.vtcamerasettings
package:com.samsung.android.app.tips
package:com.samsung.android.app.aodservice
package:com.android.systemui.accessibility.accessibilitymenu
package:com.samsung.android.dsms
package:com.android.dreams.phototable
package:com.samsung.android.mcfds
package:com.samsung.android.smartface.overlay
package:com.samsung.android.knox.kpecore
package:com.samsung.internal.systemui.navbar.sec_gestural_no_hint
```

<실행 화면>

<- 너무 많은 패키지가 존재하고, 블루링크 App으로 의심되는 패키지의 식별이 어려움

# App Data 추출 과정(3)

## ❖ Step1) 패키지 명 파악하기



* 따라서 'Shizuku' App과 'FV 파일 탐색기' App을 통해 패키지명을 파악함

-> 'Shizuku' : Android 기기에서 루팅 없이도 특정 앱들이 루트 권한을 획득할 수 있도록 도와주는 App

-> 'FV 파일 탐색기' : 안드로이드 기기에서 파일 및 폴더를 관리하는 데 사용되는 앱


<확인 결과>

<- 'com.velox.hkmc_tm1k'로 패키지명 확인

# App Data 추출 과정(4)

## ❖ Step2) run-as를 통한 /data/data 접근 시도

* 'adb shell run-as com.velox.hkmc_tm' 명령어 사용

```
PS C:\Users\cgumg> adb shell run-as com.velox.hkmc_tm1k
run-as: package not debuggable: com.velox.hkmc_tm1k
```

-> 해당 패키지에 대해 'not debuggable' 발생

-> 이유 : 설정 파일에서 debug 권한을 주지않음

-> 따라서, APK를 추출하여 App을 Decompile하여 권한 부여 후 Repackaging을 통해 접근하도록 해야겠다고 판단

# App Data 추출 과정(5)

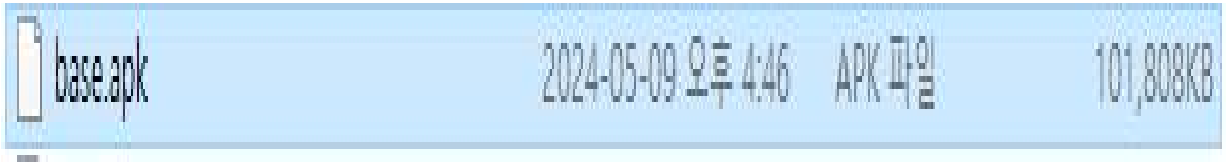
## ❖ Step3) APK 추출하기

-> 'adb shell pm path com.velox.hkmc_tm1k'를 통해 App 경로 확인

```
PS C:\Users\cgumg> adb shell pm path com.velox.hkmc_tm1k
package:/data/app/~~q0yV5fXXF1WaVq-eAMblYA==/com.velox.hkmc_tm1k-my0ALvoxT_EtZ54pOmoY8w==/base.apk
```

-> 'adb pull 패키지 경로'를 통해 apk 추출

```
PS C:\Users\cgumg> adb pull /data/app/~~sQWeZq8Cg_fPbBaFFdRfiQ==/com.velox.hkmc_tm1k-molZlQzjQhfMdNGBKau0iA==/base.apk
/data/app/~~sQWeZq8Cg_fPbBaFFdRfiQ==/com.velox.hkmc_tm1k-m...e pulled, 0 skipped. 37.0 MB/s (104250880 bytes in 2.684s)
```

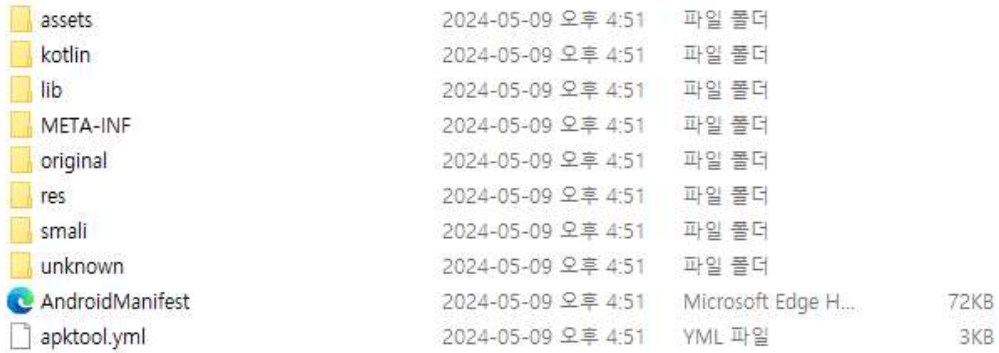


# App Data 추출 과정(6)

## ❖ Step4) Decompile 진행

* 도구 : apktool.jar 이용

-> 'apktool d base.apk -o 저장폴더명'




<디컴파일된 패키지>

# App Data 추출 과정(7)

## ❖ Step5) 설정값 변경

* 변경 전

```
<application android:allowBackup="false" android:appComponentFactory="androidx.core.app.CoreComponentFactory" android:debuggable="false"
```

-> 확인 결과, Backup 기능도 false상태 였음, 따라서 Backup 기능과 Debug 기능을 True로 변경해줌

* 변경 후

```
<application android:allowBackup="true" android:appComponentFactory="androidx.core.app.CoreComponentFactory" android:debuggable="true"
```

# App Data 추출 과정(8)

## ❖ Step6) Compile(Repackaging) 진행

-> 'apktool b 패키지 할 폴더명 -o apk 파일명'

```
PS C:\Users\cgumg> apktool b bluelink_d -o base714.apk
I: Using Apktool 2.9.3
I: Checking whether sources has changed...
I: Smaling smali folder into classes.dex...
I: Checking whether resources has changed...
I: Building resources...
I: Copying libs... (/lib)
I: Copying libs... (/kotlin)
I: Copying libs... (/META-INF/services)
I: Building apk file...
I: Copying unknown files/dir...
I: Built apk into: base714.apk
```

<Compile 과정>


<생성된 APK File>

# App Data 추출 과정(9)

❖ **Step7)** 서명 **진행**

* 키 생성 과정

->도구 : keytool

-> 'keytool -genkey -v -keystore my-release-key.jks -keyalg RSA -keysize 2048 -validity 10000 -alias my-alias'

```
PS C:\Users\cgumg> keytool -genkey -v -keystore my-release-key.jks -keyalg RSA -keysize 2048 -validity 10000 -alias mind
ol
```

<키 생성 명령어>

```
Generating 2,048 bit RSA key pair and self-signed certificate (SHA384withRSA) with a validity of 10,000 days
```

<생성된 키>

# App Data 추출 과정(10)

## ❖ Step7) 서명 진행

* 도구 : zipalign, apksigner

-> 정렬 필요 : 'zipalign -v -p 4 my-app-unsigned.apk my-app-unsigned-aligned.apk'

```
PS C:\Users\cgumg> zipalign -v -p 4 base714.apk my-app-unsigned-aligned.apk
Verifying alignment of my-app-unsigned-aligned.apk (4)...
      49 AndroidManifest.xml (OK - compressed)
   16140 classes.dex (OK - compressed)
   51663 kotlin/annotation/annotation.kotlin_builtins (OK - compressed)
   52316 kotlin/collections/collections.kotlin_builtins (OK - compressed)
   53927 kotlin/coroutines/coroutines.kotlin_builtins (OK - compressed)
   54170 kotlin/internal/internal.kotlin_builtins (OK - compressed)
```

<명령어 입력>


<정렬된 상태로 만들어진 APK File>

# App Data 추출 과정(11)

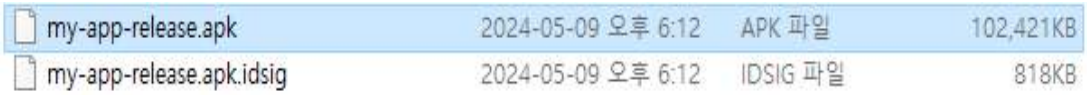
## ❖ Step7) 서명 진행

* 서명 진행

-> apksigner sign --ks my-release-key.jks --out my-app-release.apk my-app-unsigned-aligned.apk

```
PS C:\Users\cgumg> apksigner sign --ks my-release-key.jks --out my-app-release.apk my-app-unsigned-aligned.apk
Keystore password for signer #1:
```

<명령어 입력>




<생성된 2개의 파일>

# App Data 추출 과정(12)

❖ **Step8) /data/data/com.velox.hkmc_tm1k에 접근해보기**

❖ * adb shell run-as com.velox.hkmc_tm1k

```
PS C:\Users\cgumg> adb shell run-as com.velox.hkmc_tm1k

ls
cache
code_cache
databases
files
no_backup
shared_prefs

cd ./databases

ls
com.google.android.datatransport.events
com.google.android.datatransport.events-journal
google_app_measurement_local.db
google_app_measurement_local.db-journal
```

<- 내부저장소에 접근이 가능하고,
'ls' 명령어, 'cd' 명령어 사용 가능한 것 까지 확인 가능

# App Data 추출 과정(13)

## ❖ Step9) App Data 추출하기

* 가능한 경우의 수

CASE 1) 'adb exec-out run-as com.velox.hkmc_tm1k tar c ./ > output.tar' 명령어를 통한 tar file로 추출

-> tar file로 추출은 되지만 압축 해제 과정에서 오류가 발생함

CASE 2) '/data/data/com.velox.hkmc_tm1k'에서 'cp'명령어를 통한 /sdcard 디렉터리로 복사하기

-> Ex1) cp -r databases ../../sdcard

cp: ../../sdcard: Permission denied

-> Ex2) cp -r databases /data/local/tmp 를 통해 /data/local/tmp로 옮긴 후 sdcard로 옮기려 하였지만

'Permission denied' 발생

# App Data 추출 과정(14)

❖ **Step9) App Data 추출하기**

*** 성공 CASE : 'adb backup' 이용**

*** 명령어 : 'adb backup -f my_data.ab -noapk com.velox.hkmc_tm1k'**

*.ab -> Android 디바이스의 데이터를 백업하고 복원하기 위한 특별한 형식 (주로 adb restore와 쓰임)

```
PS C:\Users\cgumg> adb backup -f my_data.ab -noapk com.velox.hkmc_tm1k
WARNING: adb backup is deprecated and may be removed in a future release
Now unlock your device and confirm the backup operation...
PS C:\Users\cgumg>
```

<명령어 입력>


<추출된 ab 파일>

<- 약 20MB정도로 추출된 걸로 보아 성공적으로 data 추출이 되었음을 알 수 있음

# App Data 추출 과정(15)

## ❖ Step10) .ab File .tar로 변환하기

* 도구 : abe.jar

* 명령어 : 'java -jar abe.jar unpack my_data.ab my_bluelink.tar'

```
PS C:\Users\cgumg> java -jar abe.jar unpack my_data.ab my_bluelink.tar
0% 1% 2% 3% 4% 5% 6% 7% 8% 9% 10% 11% 12% 13% 14% 15% 16% 17% 18% 19% 20% 21% 22% 23% 24% 25% 26% 27% 28% 29% 30% 31% 32
% 33% 34% 35% 36% 37% 38% 39% 40% 41% 42% 43% 44% 45% 46% 47% 48% 49% 50% 51% 52% 53% 54% 55% 56% 57% 58% 59% 60% 61% 62
% 63% 64% 65% 66% 67% 68% 69% 70% 71% 72% 73% 74% 75% 76% 77% 78% 79% 80% 81% 82% 83% 84% 85% 86% 87% 88% 89% 90% 91% 92
% 93% 94% 95% 96% 97% 98% 99% 100%
21560832 bytes written to my_bluelink.tar.
```

<명령어 입력>

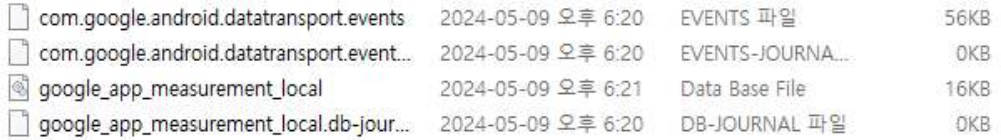

<생성된 my_bluelink.tar>

# App Data 추출 과정(16)

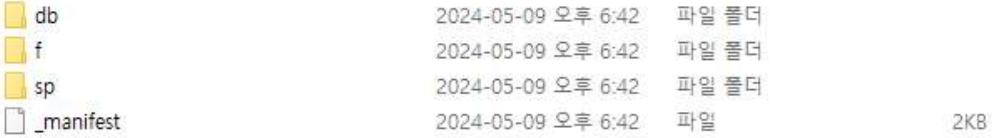
## ❖ Step11) 압축해제 후 App Data 확인


<1>


<2>




<4>




<3>

# 03 아쉬운 점

# 아쉬운 점

1. APP 삭제의 불가피함

-> 디컴파일과 컴파일 과정을 거쳐야 하기에 APP 삭제를 할 수 밖에 없는 점이 아쉬움

2. 컴파일 후 APP 재설치시 APP이 열리지 않음

-> 아직까지 파악하지 못하였지만 APP이 열리지 않는 문제가 발생하긴함

-> 이에 관해서는 좀 더 고민해보거나, 방법이 없다면 APP 삭제 후 Store에서 재설치 하는 방식으로 해야할 듯

# 들어주셔서 감사합니다 !