Getting Started with Android Malware Analysis

Keheira Henderson

Backpack Media

Agenda

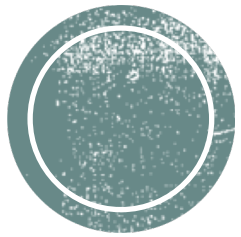
- Who am I?
- My Android Background
- Malicious App Demo
- Analysis Demo
- Present Your Findings
- Questions

Who am I?

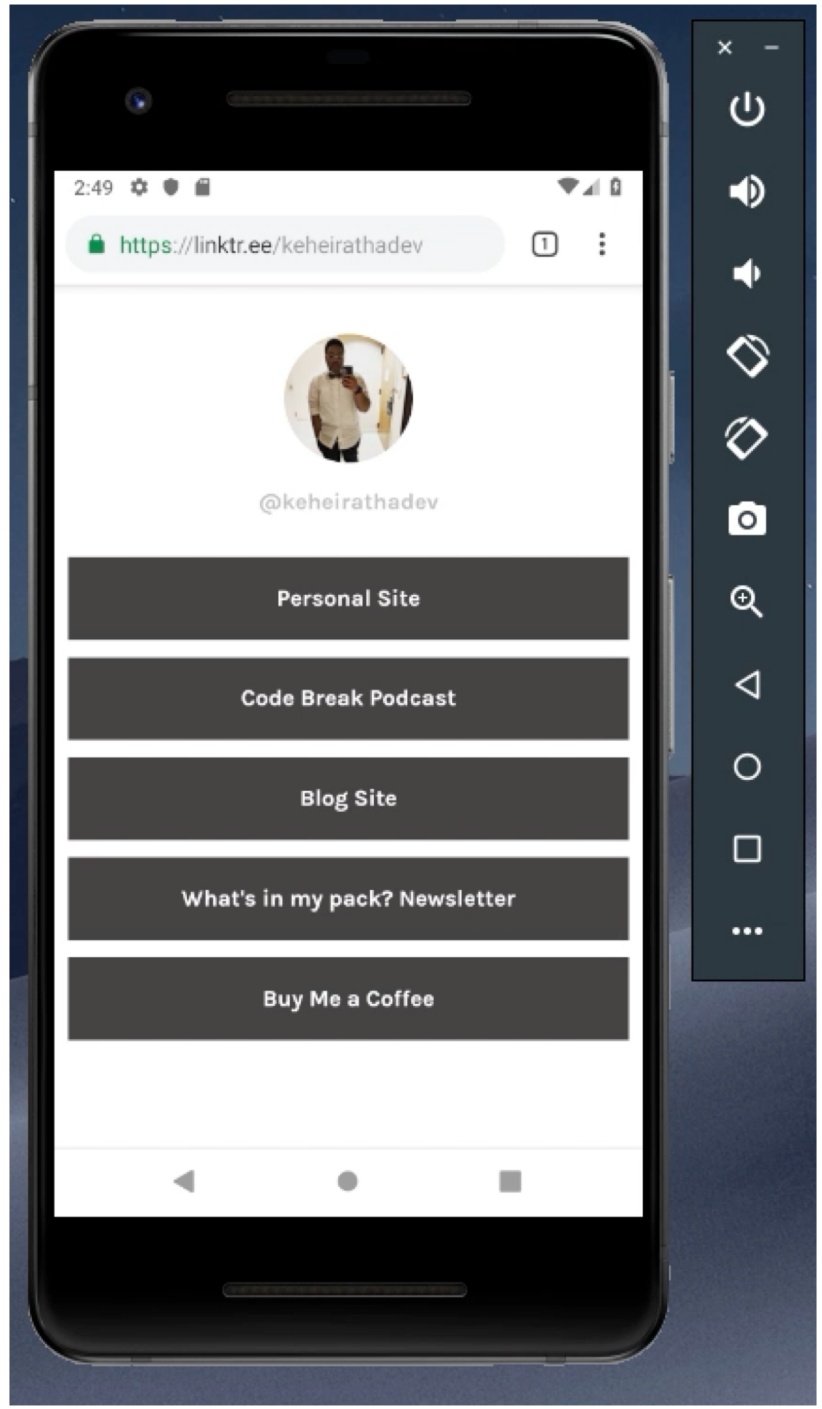
- Key-air-uh
- Nashville Native
- BS in Computer Engineering
- Mobile Developer
- Masters student in Network Security

Android Background

- Worked on some work regarding Droidsheep Guard
- Published a paper an how to teach viruses & malware
- Worked on SACH (Secure Android Code Helper)
- Android Dev specialist at work



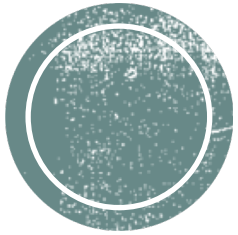
App Demo


2:49
https://linktr.ee/keheirathadev
@keheirathadev
Personal Site
Code Break Podcast
Blog Site
What's in my pack? Newsletter
Buy Me a Coffee

GET https://bsidesnash19.herokuapp.com/api/search Send Save

Pretty Raw Preview JSON


```
[
    {
        "_id": "5c907f23427add1a7fc5bd90",
        "params": "bsidesNash19",
        "deviceId": "@keheirathadev",
        "__v": 0
    },
    {
        "_id": "5c99d2d1256c4900171e1623",
        "params": "demo",
        "deviceId": "demo",
        "__v": 0
    },
    {
        "_id": "5cad4aeb538aea0017659106",
        "params": "PSR1.180720.075",
        "deviceId": "Android SDK built for x86",
        "__v": 0
    },
    {
        "_id": "5cad4b4c538aea0017659107",
        "params": "PSR1.180720.075",
        "deviceId": "Android SDK built for x86",
        "__v": 0
    },
    {
        "_id": "5cb178e0291b06001729d808",
        "params": "PQ2A.190205.002",
        "deviceId": "Pixel 2",
        "__v": 0
    }
]
```

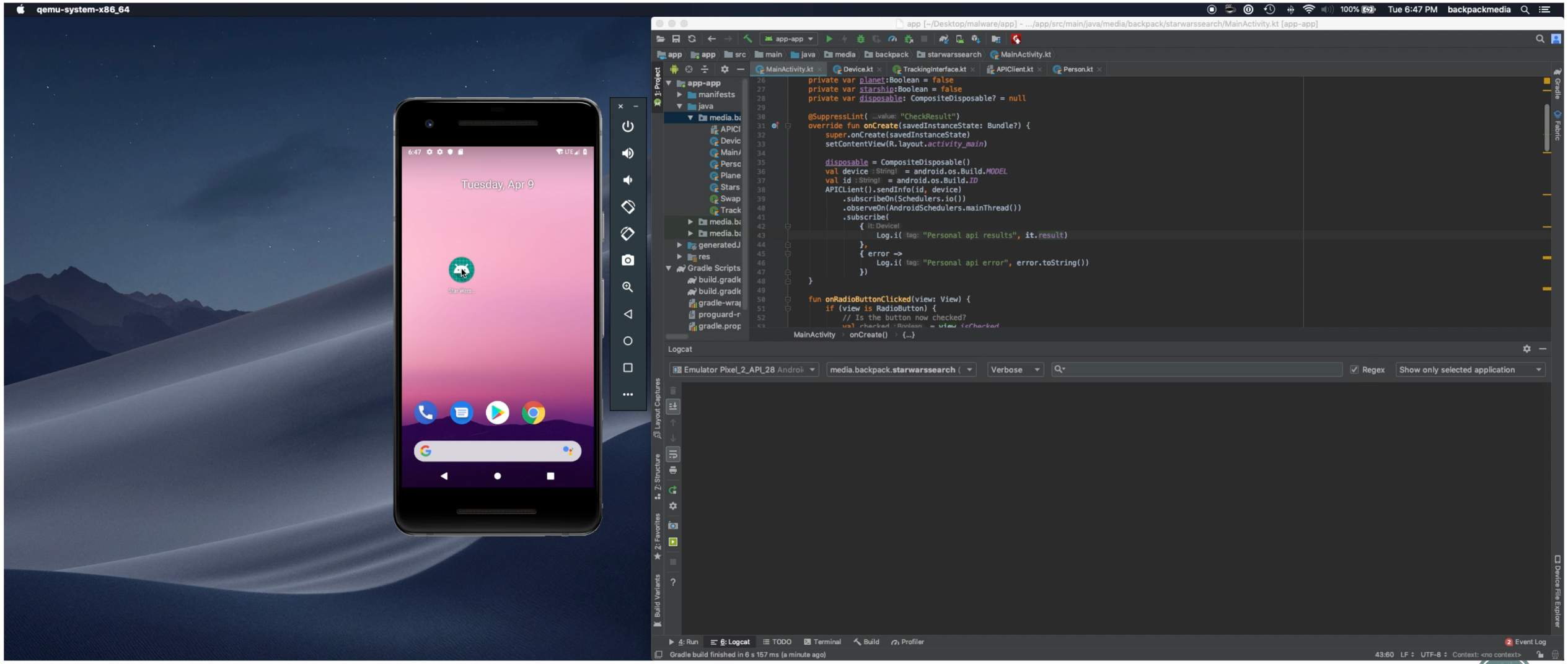


Tool Demo

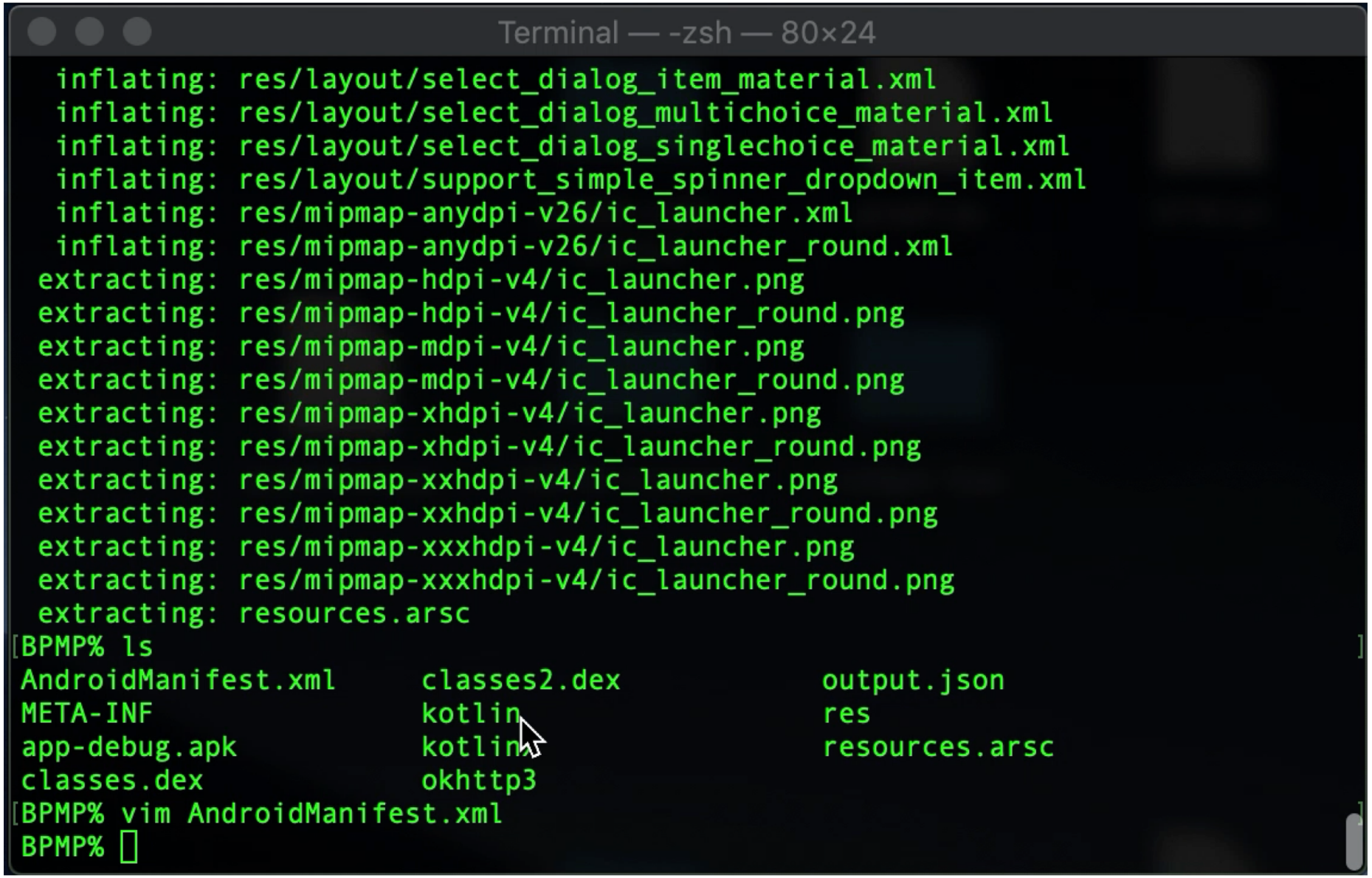
Hardware/software

- Computer (8gb preferred)
- Android Studios
- Homebrew (if using mac)
- APKtool
- Burp Suite
- Dex2Jar
- Java Decompiler GUI

ADB Logcat


qemu-system-x86_64
Tue 6:47 PM backpackmedia
app [~/Desktop/malware/app] - .../app/src/main/java/media/backpack/starwarssearch/MainActivity.kt [app-app]
MainActivity.kt
Device.kt
TrackingInterface.kt
APIClient.kt
Person.kt
Logcat
Emulator Pixel_2_API_28 Androi
media.backpack.starwarssearch
Verbose
Regex
Show only selected application
Tuesday, Apr 9
Gradle build finished in 6 s 157 ms (a minute ago)

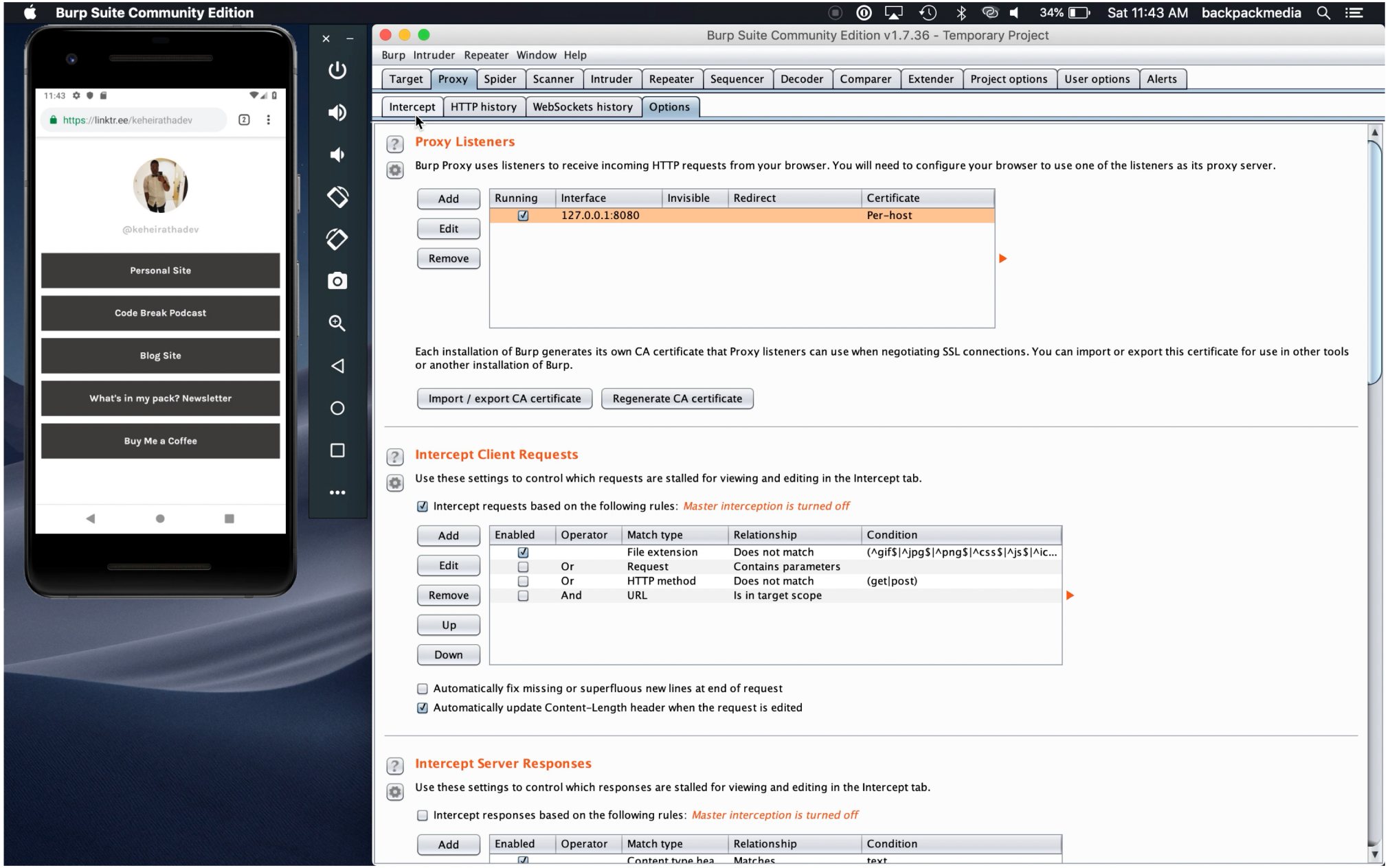
APK Tool


Terminal — -zsh — 80×24
inflating: res/layout/select_dialog_item_material.xml
inflating: res/layout/select_dialog_multichoice_material.xml
inflating: res/layout/select_dialog_singlechoice_material.xml
inflating: res/layout/support_simple_spinner_dropdown_item.xml
inflating: res/mipmap-anydpi-v26/ic_launcher.xml
inflating: res/mipmap-anydpi-v26/ic_launcher_round.xml
extracting: res/mipmap-hdpi-v4/ic_launcher.png
extracting: res/mipmap-hdpi-v4/ic_launcher_round.png
extracting: res/mipmap-mdpi-v4/ic_launcher.png
extracting: res/mipmap-mdpi-v4/ic_launcher_round.png
extracting: res/mipmap-xhdpi-v4/ic_launcher.png
extracting: res/mipmap-xhdpi-v4/ic_launcher_round.png
extracting: res/mipmap-xxhdpi-v4/ic_launcher.png
extracting: res/mipmap-xxhdpi-v4/ic_launcher_round.png
extracting: res/mipmap-xxxhdpi-v4/ic_launcher.png
extracting: res/mipmap-xxxhdpi-v4/ic_launcher_round.png
extracting: resources.arsc
BPMP% ls
AndroidManifest.xml
META-INF
app-debug.apk
classes.dex
classes2.dex
kotlin
kotlin
okhttp3
output.json
res
resources.arsc
BPMP% vim AndroidManifest.xml
BPMP%

DEX2Jar and JD-GUI

```
[BPMP% /Users/backpackmedia/Desktop/malware/dex2jar-2.0/d2j-dex2jar.sh -f  /Users]
/backpackmedia/Desktop/malware/app/app/build/outputs/apk/debug/app-debug.apk
dex2jar /Users/backpackmedia/Desktop/malware/app/app/build/outputs/apk/debug/app
-debug.apk -> ./app-debug-dex2jar.jar
com.googlecode.d2j.DexException: not support version.
        at com.googlecode.d2j.reader.DexFileReader.<init>(DexFileReader.java:151
)
        at com.googlecode.d2j.reader.DexFileReader.<init>(DexFileReader.java:211
)
        at com.googlecode.dex2jar.tools.Dex2jarCmd.doCommandLine(Dex2jarCmd.java
:104)
        at com.googlecode.dex2jar.tools.BaseCmd.doMain(BaseCmd.java:288)
        at com.googlecode.dex2jar.tools.Dex2jarCmd.main(Dex2jarCmd.java:32)
BPMP%
```

Burp Suite


Burp Suite Community Edition
Sat 11:43 AM backpackmedia
11:43
https://linktr.ee/keheirathadev
@keheirathadev
Personal Site
Code Break Podcast
Blog Site
What's in my pack? Newsletter
Buy Me a Coffee
Burp Suite Community Edition v1.7.36 - Temporary Project
Burp Intruder Repeater Window Help
Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Alerts
Intercept HTTP history WebSockets history Options
Proxy Listeners
Burp Proxy uses listeners to receive incoming HTTP requests from your browser. You will need to configure your browser to use one of the listeners as its proxy server.
Add
Edit
Remove
Running Interface Invisible Redirect Certificate
127.0.0.1:8080 Per-host
Each installation of Burp generates its own CA certificate that Proxy listeners can use when negotiating SSL connections. You can import or export this certificate for use in other tools or another installation of Burp.
Import / export CA certificate
Regenerate CA certificate
Intercept Client Requests
Use these settings to control which requests are stalled for viewing and editing in the Intercept tab.
Intercept requests based on the following rules: Master interception is turned off
Add
Edit
Remove
Up
Down
Enabled Operator Match type Relationship Condition
File extension Does not match (^gif$|^jpg$|^png$|^css$|^js$|^ic...
Or Request Contains parameters
Or HTTP method Does not match (get|post)
And URL Is in target scope
Automatically fix missing or superfluous new lines at end of request
Automatically update Content-Length header when the request is edited
Intercept Server Responses
Use these settings to control which responses are stalled for viewing and editing in the Intercept tab.
Intercept responses based on the following rules: Master interception is turned off
Add
Enabled Operator Match type Relationship Condition

Key Items to Look for

- Permissions

```
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE"/>
<uses-permission android:name="android.permission.CHANGE_WIFI_STATE"/>
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-configuration android:name="android.permission.ACCESS_WIFI_STATE" />
```

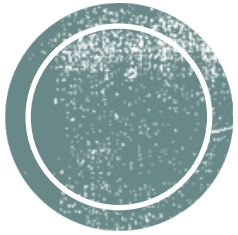
- Client

```
class APIClient{
    private val SWAPI_URL = "https://swapi.co/api/"
    private val TRACKING_URL = "https://bsidesnash19.herokuapp.com/api/"
    private val swapiService: SwapiInterface
    private val trackingService: TrackingInterface

    init{
        val retrofit : Retrofit = Retrofit.Builder()
            .baseUrl(SWAPI_URL)
            .addCallAdapterFactory(RxJava2CallAdapterFactory.create())
            .addConverterFactory(GsonConverterFactory.create())
            .build()
        swapiService = retrofit.create(SwapiInterface::class.java)

        val retrofit2 : Retrofit = Retrofit.Builder()
            .baseUrl(TRACKING_URL)
            .addCallAdapterFactory(RxJava2CallAdapterFactory.create())
            .addConverterFactory(GsonConverterFactory.create())
            .build()
        trackingService = retrofit2.create(TrackingInterface::class.java)
    }
```

- Logcat/ADB

How do I report findings?

1. See if the company has a security team and email
    1. Be nice
    2. Be forward
    3. Don’t seem malicious
    4. Don’t forget to let them know how it was reproduced
2. Write a paper/blog
    1. Intro -> Scope -> Summary
    2. Date Found
    3. Screenshots
    4. Phone Specs
    5. Steps to reproduce
3. Just tweet tips with good hashtags



Questions?

https://linktr.ee/keheirathadev

Resources

- Hacker101 Android quickstart: https://www.hacker101.com/sessions/android_quickstart
- Cybrary: https://www.cybrary.it/course/ethical-hacking/
- Apktool: https://ibotpeaches.github.io/Apktool/documentation/
- Dex2jar: https://github.com/pxb1988/dex2jar
- Java Decompiler GUI: http://java-decompiler.github.io/
- Example report: https://hackerone.com/reports/283063
- SACH: https://ieeexplore.ieee.org/document/7925374