Member-only story

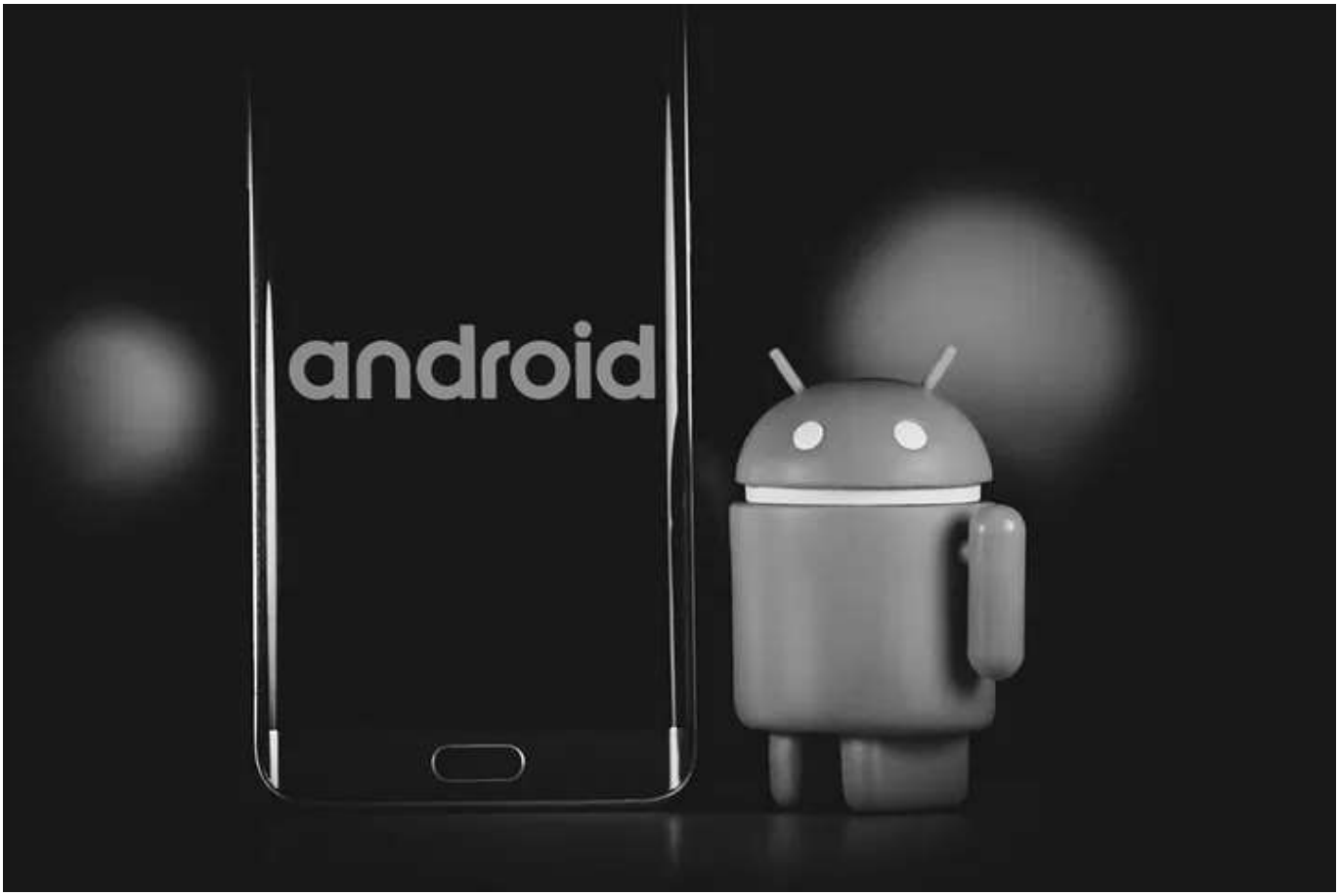
Reverse Engineering and Analyzing Android Apps: A Step-by-Step Guide

Cyber Duck · Follow

6 min read · Dec 26, 2022

276 2

Reverse engineering is the process of taking apart a software application to understand how it works and possibly modify it. This can be useful for a variety of purposes, such as analyzing security vulnerabilities, studying the inner workings of an app, or creating custom modifications. In this article, we will explore some tools and techniques for reverse engineering and analyzing Android apps. If you are interested in how to secure your APK, check out this article.



Decompiling the APK

The first step in reverse engineering an Android app is to obtain its APK (Android Package Kit) file. This is the file that contains all the code and resources of the app, and can be found on the device itself or downloaded from the internet. Once you have the APK file, you can use a tool called "apktool" to decompile it into its component parts. To use apktool, follow these steps:

1. Download and install the latest version of apktool from its website (https://ibotpeaches.github.io/Apktool/).
2. Open a terminal window and navigate to the directory where you have stored the APK file.
3. Use the following command to decompile the APK:

```
apktool d app.apk
```

This will create a new directory with the decompiled code and resources of the app.

Analyzing the Code with JD-GUI

Once you have decompiled the APK, you can use a tool called "JD-GUI" to view the Java source code of the app. JD-GUI is a graphical Java decompiler that allows you to browse the class files and see the code in a more readable format. To use JD-GUI, follow these steps:

1. Download and install JD-GUI from its website (https://github.com/java-decompiler/jd-gui/releases).
2. Open JD-GUI and click "File > Open".
3. Navigate to the directory where you decompiled the APK and select the "classes.dex" file.
4. JD-GUI will decompile the file and show you the Java source code in a tree view. You can use the various features of JD-GUI to browse the code and search for specific classes or methods.

Analyzing the Resources with APK Studio

In addition to the code, the APK file also contains various resources such as images, layouts, and strings. You can use a tool called "APK Studio" to view and edit these resources. APK Studio is an open-source integrated

development environment (IDE) that allows you to edit the XML files and other resources of the app. To use APK Studio, follow these steps:

1. Download and install APK Studio from its website (https://github.com/vaibhavpandeyvpz/apkstudio/releases).
2. Open APK Studio and click “File > Open”.
3. Navigate to the directory where you decompiled the APK and select the “AndroidManifest.xml” file.
4. APK Studio will open the app project in the IDE. You can use the various features of APK Studio to view and edit the resources of the app.

Analyzing the Manifest

The AndroidManifest.xml file is a crucial part of any Android app, as it defines the app’s components, permissions, and overall structure. When reverse engineering an app, it is often useful to analyze the manifest to get a better understanding of the app’s functionality and potential vulnerabilities. Some of the things you might want to look for in the manifest include:

- Permissions: The app might request certain permissions in order to access sensitive data or functionality on the device. You can look at the manifest to see which permissions the app is requesting, and consider whether they are appropriate or potentially risky.
- Activities: The manifest defines the various activities (screens) of the app, and how they are launched. You can use this information to get an

that run in the background and perform tasks without a user interface.

You can analyze the services to see what they are doing and how they might be interacting with other parts of the app or the device.

Analyzing the Code

Once you have decompiled the app's code and viewed it with a tool like JD-GUI, you can start analyzing the various classes and methods to get a better understanding of how the app works. Some things you might want to look for include:

- Network communication: The app might communicate with servers or other devices over the network. You can look for network-related code to see what kind of data the app is sending and receiving, and whether it is using secure protocols or potentially leaking sensitive information.
- Cryptography: The app might use cryptography to protect data or perform other tasks. You can look for code related to encryption and decryption to see how the app is using these techniques and whether there are any vulnerabilities.
- Interactions with the system: The app might use various APIs (Application Programming Interfaces) to interact with the device or other apps. You can look for code related to these APIs to see how the app is using them and whether there are any potential security risks.

Modifying the App

Once you have analyzed the app and understand how it works, you might want to modify it in some way. This could involve changing the code, resources, or manifest of the app to add or remove functionality, or to fix any

issues you have identified. Some tools that can be useful for modifying Android apps include:

- apktool: You can use apktool to modify the decompiled code and resources of the app, and then re-build the APK file with your changes.
- Android Studio: This is a full-featured IDE for Android development, which can be used to create new apps or modify existing ones.
- apk-mitm: This is a tool that allows you to intercept and modify the traffic of an app on the fly, without modifying the APK file itself. This can be useful for testing or debugging purposes.

Keep in mind that modifying an app can have unintended consequences and might violate the terms of service or other agreements. It is important to use these tools responsibly and only modify apps for legitimate purposes.

Analyzing the Resources

In addition to the code, the APK file also contains various resources such as images, layouts, and strings. You can use tools like apktool and APK Studio to view and edit these resources. Some things you might want to look for in the resources include:

- Images: The app might include images that contain sensitive information, or that are used in the user interface in a way that reveals something about the app's functionality.
- Layouts: The app's user interface is defined in XML layout files, which you can view and edit to see how the app is structured and how it functions.

- Strings: The app might include various string resources, such as error messages or labels, that can provide clues about its functionality or vulnerabilities.

Analyzing Network Traffic

When reverse engineering an app, it is often useful to analyze the network traffic it generates. This can help you understand what kind of data the app is sending and receiving, and whether it is using secure protocols or potentially leaking sensitive information. To analyze the network traffic of an Android app, you can use a tool called “Burp Suite”. Burp Suite is a professional-grade web security tool that allows you to intercept, analyze, and modify the traffic of an app or website.

To use Burp Suite to analyze the network traffic of an Android app, follow these steps:

1. Download and install Burp Suite from its website (https://portswigger.net/burp/).
2. Configure your Android device to use a proxy server by going to “Settings > Wi-Fi” and tapping the gear icon next to the connected network.
3. Set the “Proxy” to “Manual” and enter the IP address and port of your computer (e.g., “127.0.0.1:8080”).
4. Open Burp Suite and click “Proxy > Options”.
5. Enable the “Intercept” option and click “Save”.
6. Open the app on your Android device and use it as you normally would.

7. Burp Suite will intercept and display the network traffic generated by the app. You can use the various features of Burp Suite to analyze the traffic and modify it as needed.

Thank you for reading! If you enjoyed the content, please consider following my account for more updates and content.

Reverse Engineering | Cybersecurity | Android | Android App Development

Written by Cyber Duck

Follow

1K Followers

As a cybersecurity professional, I joined Medium to share good cyber habits, IT carreer path, certifications and cyber news.

More from Cyber Duck