

Lab #1

Reverse Engineering a Malicious Android App

CSC435: Computer Security
Spring 2024



WARNING: IMPORTANT NOTICE!

The app you will be working on is an **actual real-life** scam app. Here are some DO's and DON'T's.

Do **NOT**:

- Install and run this APK on any android device.
- Give out ANY actual/true information during the dynamic analysis tasks.
- Distribute and send this file to anyone who might fall for the scam.

Do:

- Follow instructions step by step to ensure cyber safety.
- Run the application **ONLY** on the android emulator.

General instructions:

- Your submission should include a detailed lab report that includes the procedures/steps taken to complete the lab, included but not limited to screenshots, answers to the questions, and a final page that includes a summary of what each person did in this project, along with any difficulties/issues faced in this lab and how you resolved them.
- Each member is supposed to work on the lab and is responsible about everything that is covered in this lab.
- Each group will be assigned demos. All group members should be present. The lab grade is individual to every member based on their performance in the lab demo. Members of the same group may get different grades, based on their performance.
- No time extension will be given. Late submissions will NOT be allowed.
- Any form of cheating or plagiarism will not be tolerated.
- Provide any references used, if any, in your report. Any information used without references will not be considered.
- Only ONE member of the group should submit a report on blackboard. You DO NOT have to submit a printed copy.
- In your report, include all MAC addresses and IP addresses of the machine you are working on. Any report without these will result in a 0.
- If the machine used in the demonstration does not match with the screenshots shown in your report, you will be given a 0.

Background

A brief introduction and overview

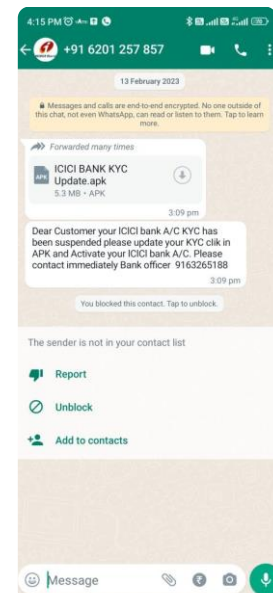
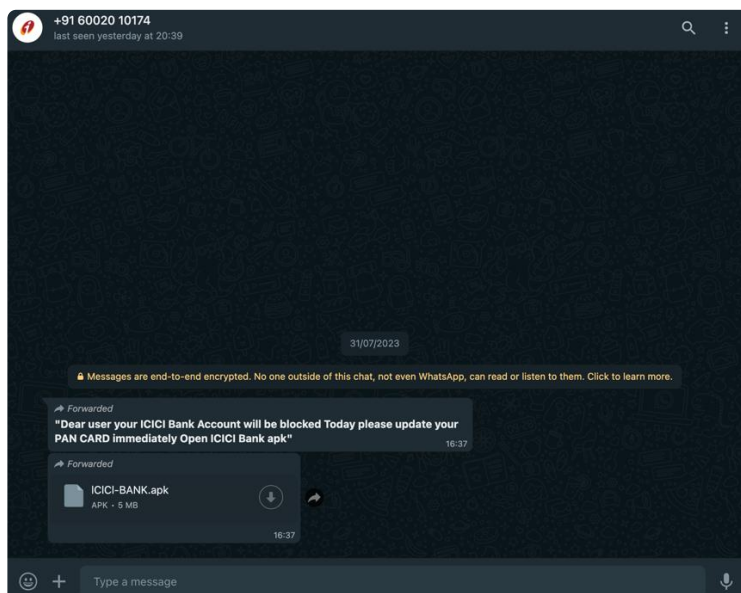
“The ICICI Bank is an Indian multinational bank and financial services company, with its headquarters in Mumbai, India. It offers a wide range of banking and financial services for corporate and retail customers through a variety of delivery channels and specialized subsidiaries in the areas of investment banking, life/non-life insurance, venture capital and asset management.”

(Source: [Wikipedia](#))

In 2023, many users received a WhatsApp text message, claiming that they are the ICICI bank, saying that the account will be blocked, and that they should update their card immediately by opening the APK file. (Read more: [News18](#))



The image below shows an example of what an *actual user* received via WhatsApp on December 7th, 2023. Notice how the profile picture of the scammer/hacker is the ICICI bank logo, to make it seem official.



After the ICICI bank knew of this, they immediately spread awareness via their social media pages, e-mail newsletters, and more, informing their users about this scam, and how to prevent it. You can see some of the awareness below:

X (formerly Twitter): <https://twitter.com/ICICIBank/status/1630907036396552192?lang=en>

LinkedIn: https://www.linkedin.com/posts/icici-bank_beware-do-not-install-apps-apk-files-from-activity-7036642885336731648-yoT6/

Facebook: <https://www.facebook.com/icicibank/videos/avoid-the-trap-beware-of-fake-app-beatthecheats-again-with-tabu/2055753588095335/>

YouTube: https://www.youtube.com/watch?v=G_GfQ5U6DJs

Part One: Preliminary Research and Study (15 pts)

In this part, you are asked to answer a few questions, and understand a few concepts before investigating the actual APK.

Grades of this part will depend on your answers to the following questions (5 points per question).



Questions to be answered in your report!

Briefly answer each of the following questions, in no more than 100 words:

1. What is an APK file? What does it stand for? How does it work? Which programming language is usually used to write an APK file?
2. What is meant by the term ‘*Sideload*ing’? Define it and state the security concerns that come with it. Which operating system(s) offer(s) sideloading?
3. WhatsApp offers a file sharing feature, where users can share files of almost any type. Does this raise a security concern? Should WhatsApp limit the file types one can send? In case of an attack that happened by sharing files on their platform, will WhatsApp be responsible for that? Discuss.

Part Two: Setting up the Environment (15 pts)

You should also ensure that you have all the required tools safely and ready to use.

In this part, we will be setting up the environment in preparation for the static and dynamic analysis in the upcoming parts. **Grades of this part will depend on your screenshots and step-by-step walkthroughs in your report.**



General System Requirements

The recommended system requirements of this lab are:

Note: you may find your way to set up the environment differently if you don't meet the requirements below, but that's up to you!

Operating System	<ul style="list-style-type: none">• macOS Sanoma (14) or Ventura (13)• Windows 11 or 10• Linux Debian 11, Ubuntu 22.04LTS, or Fedora Workstation 38
Processor (CPU)	<ul style="list-style-type: none">• PC: Intel or AMD• Mac: Intel or Apple Silicon
Free Space	At least 3GB for the entire thing

Important note for PC users: your device may have virtualization off by default. In that case, Genymotion may not work on your PC instantly. Therefore, you should enable virtualization in BIOS. Here's an [article](#) that can help you.

Section A: Installing the Android Emulator

This section will focus on installing and setting up the android emulator.



What is an Emulator?

An emulator is a software program or hardware device that enables a computer system (the host) to behave like another computer system (the guest) by mimicking its hardware, software, and functionalities. Essentially, it allows the host system to run programs or execute operations that are typically designed for the guest system.

In our case, since we are examining an APK file, we need an android emulator.

We will use **Genymotion Desktop** as our emulator.

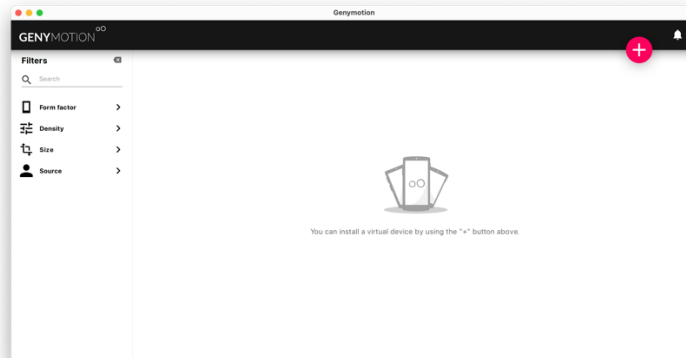
To install and set up the emulator, follow the steps below:

1. Download Genymotion Desktop from their official site, according to your operating system:

<https://www.genymotion.com/product-desktop/download/>

Note: Genymotion requires VirtualBox in order to run emulators. If you already have VirtualBox you should be good to go! If not, you would have to install it, and you will be prompted to do so in the installation.

2. Run through the installation process. Once you're done, you should see a screen like the one below:



3. To create the emulator, simple click on the **+** button on the top right corner.
4. In this guide, we will be emulating a Google Pixel 3a. It's up to you to choose any other phone, but let it run android 11.0. So, choose the following specification:
 - a. **Phone:** Google Pixel 3a
 - b. **Android Version:** 11.0
 - c. **System Hardware:** *based on your preferences. (You can leave them as default)*



Once done, you should see the following entry:

Type	Name	Android API	Resolution	Density	Size on disk	Source	Status	Actions
	Google Pixel 3a	11.0 - API 30	1080 x 2220	420	1.82 MB	Genymotion	Off	

5. You're now all set! You have your emulator ready!

Section B: Installing the Necessary Apps

In this lab, you will be using two main necessary apps: **Visual Studio Code** and **Burp Suite**.



What are these Apps?



Visual Studio Code

Visual Studio Code is a source-code editor and Integrated Development Environment (IDE) developed by Microsoft for Windows, Linux and macOS. VS Code will be used to view and analyze the code of the malicious app.



Burp Suite

Burp Suite, developed by PortSwigger, is a software security application used for penetration testing of web applications. Burp Suite will be used to view network traffic, and track/read HTTP/HTTPS packets.

To install these applications, use the following links from their official websites and run through the installation process: (If you already have these apps, you can skip this part)

- Visual Studio Code: <https://code.visualstudio.com/download>
- Burp Suite: <https://portswigger.net/burp/communitydownload>

Section C: Installing platform-tools for Using ADB Commands



What are ADB commands?

“Android Debug Bridge (adb) is a versatile command-line tool that lets you communicate with a device. The adb command facilitates a variety of device actions, such as installing and debugging apps. adb provides access to a Unix shell that you can use to run a variety of commands on a device. It is a client-server program that includes three components:

- **A client**, which sends commands. The client runs on your development machine. You can invoke a client from a command-line terminal by issuing an adb command.
- **A daemon (adb)**, which runs commands on a device. The daemon runs as a background process on each device.
- **A server**, which manages communication between the client and the daemon. The server runs as a background process on your development machine.”

Source: <https://developer.android.com/tools/adb>

We will use ADB commands in order to modify network settings and install the CA certificate so that burp suite can listen and intercept traffic coming from the app in the android emulator.



What is a CA certificate?

“In cryptography, a certificate authority is an entity that stores, signs, and issues digital certificates. A digital certificate certifies the ownership of a public key by the named subject of the certificate.”

Read more [here](#).

Android has given us a software development kit (SDK) called platform-tools which we can use to run the ADB commands from the terminal (for Mac/Linux) or from the command prompt (for Windows).

You can download this SDK from these links based on your platform:

- Mac: <https://dl.google.com/android/repository/platform-tools-latest-darwin.zip>
- Linux: <https://dl.google.com/android/repository/platform-tools-latest-linux.zip>
- Windows: <https://dl.google.com/android/repository/platform-tools-latest-windows.zip>

Once done, extract the zip file. You can keep it in your downloads folder.



Checking that ADB is installed and working

To make sure that ADB is installed and working properly, we will run a command to see the available devices:

1. Open Genymotion and click on the play button (▶) to start your emulator.
2. Open terminal/command prompt.
3. To use the ADB commands, you must be inside the `platform-tools` folder. Assuming it is in downloads, change the directory accordingly.

```
cd Downloads/platform-tools/
```

4. Once in, you can use ADB commands.

- a. For Mac and Linux users: (use `./adb`)

```
./adb devices
```

- b. For Windows users: (use `adb`)

```
adb devices
```

5. You should be able to see your device under the list of devices attached.

Part Three: Static Analysis (35 pts)

Static analysis refers to examining files without executing the code. In this stage, your task is to just read and analyze the code, files, helper functions, URLs, etc.

Grades of this part will depend on your screenshots and step-by-step walkthroughs in your report along with the answers to the questions.

Create a folder in your user folder named “Malware Analysis”. Download the APK from blackboard and place it in this folder. Unzip the file using the password **CSC435**.

Section A: Decompiling the Malware

As a first essential step, we need to view the code by using a decompiler.

Use any decompiler of your choice to decompile the APK file (JADX-GUI is a recommendation) or you can choose an online decompiler like <https://www.decompiler.com> (**not recommended**).

After you decompile, you should see two folders named `resources` and `sources`.



Questions to be answered in your report!

1. What is a decompiler? What does it do?
2. Do you notice anything odd about the name of the file? Why do you think the developer did this?

Section B: Examining the App's Resources

We can view the resources of the app by visiting the `resources` folder.



Questions to be answered in your report!

3. Look through the resources used by the app, especially the images. What conclusions can you draw out? Take a look at some images, and attach them into your report, and analyze.

Section C: Reading and Analyzing the Codes

After you decompiled the APK, read and analyze the source codes of the app and answer the following questions.



Questions to be answered in your report!

Answer the questions with justification and screenshots.

4. What is the name of the main application?
5. What platform did the attacker use to create this app? What programming language was used to write the code?
6. What can you notice about the names of the code files? Figure out what the actual names of the files are.
7. What is the domain (URL) that this malware is communicating with?
8. What is URL `whois` analysis? Explain briefly. Then conduct a `whois` analysis on this URL. What can you deduce?
9. What does the malware mainly do? In which file did you find the results? Does it use any encryption/encoding techniques throughout the process? Explain.
10. This malware sends an SMS message to a certain phone number. What is that phone number? What is its content?
11. What is the type of this malware? Explain.

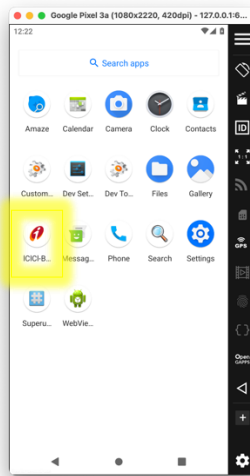
Part Four: Dynamic Analysis (35 pts)


Dynamic malware analysis executes suspected malicious code usually in a sandbox environment, virtual machine or emulator. This closed system enables security professionals to watch the malware in action without the risk of letting it infect their system or escape into the enterprise network.

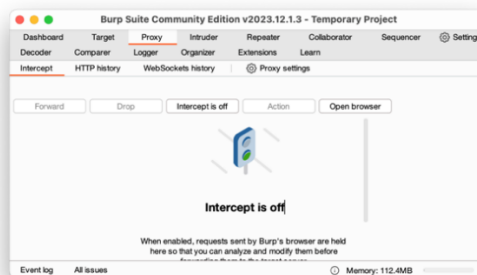
In this part, we will actually run the app and see exactly what it does. We shall then validate whether our deductions and analysis in the static part are correct or not. **Grades of this part will depend on your screenshots and step-by-step walkthroughs in your report along with the answers to the questions.**

To do so:

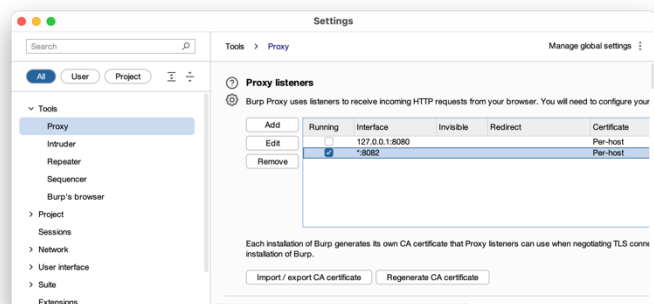
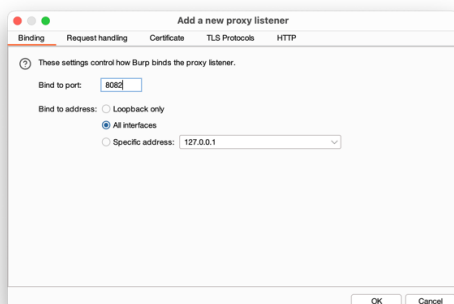
1. Open Genymotion and launch your emulator.
2. Drag and drop the APK file onto your emulator. This should install the APK, and you should see the app as shown in the screenshot below.



3. Next start up Burp Suite to set up the proxy in order to start listening to network traffic. Go to the 'Proxy' tab and click on  Proxy settings.



4. You should see a proxy listener by default, listening to 127.0.0.1:8080. Unselect this listener and click on the 'Add' button to create a new listener. Bind this listener to any unused port of your choice. Let's choose 8082. Bind it to all interfaces.



5. Click on OK.
6. We now need to generate the CA certificate so that Android and Burp Suite can communicate. To do that, click on `Import / export CA certificate`, and choose to export certificate in DER format. Select the destination that you want to save it to (we'll choose desktop), and name it (we'll choose `burp.cer`).
7. For android to understand and use this certificate, it should be in a certain format. To convert the certificate, we shall use `openssl`.



What is OpenSSL?

"OpenSSL is an open-source command line tool that is commonly used to generate private keys, create CSRs, install your SSL/TLS certificate, and identify certificate information."

Read more [here](#).

a. If you have OpenSSL on your machine (you can check by running `openssl version -a`)

First, we need to have the certificate in PEM format then output the `subject_hash_old`. To do that, follow the commands:

```
cd Desktop
openssl x509 -inform DER -in burp.cer -out burp.pem
openssl x509 -inform PEM -subject_hash_old -in burp.pem |head -1
```

Then, rename the file with the output hash from the last command. The hash will most probably be `9a5ba575`. Android wants the certificates to be in `.0` extension. So simply execute the command:

```
mv burp.pem 9a5ba575.0
```

b. If you don't have OpenSSL on your machine

You can simply visit the website: <https://www.cryptool.org/en/cto/openssl>

This website allows you to execute the commands online.

Upload the certificate file, then execute the commands:

```
openssl x509 -inform DER -in burp.cer -out burp.pem
openssl x509 -inform PEM -subject_hash_old -in burp.pem |head -1
```

Then download the new file and rename it to `9a5ba575.0`.

8. Once you have the certificate in `.0` format, move it into the platform-tools file. This is because we want to use ADB commands in order to push and install the certificate into the android emulator.
9. We then need to execute the following commands to install the certificate:
 - a. First, ensure that your emulator is rooted. This is because we need to have root privileges in order to install the certificate on the system level. Installing the certificate on the user level will not work as new android versions don't allow apps to trust user-level certificates. The below command should give you that the device is already rooted, as all emulators on Genymotion are rooted by default.

```
cd Downloads/platform-tools/
./adb root
```
 - b. Second, we need to run the `remount` command. This will allow us to view, read, and write into the android file system. Without remounting the device, you can only read the files.

```
./adb remount
```
 - c. Once remounted, we can push the file into the android device.

```
./adb push 9a5ba575.0 /sdcard/
```


- d. We then need to enter the shell in order to properly have the file in its right location.

```
./adb shell

# mv /sdcard/9a5ba575.0 /system/etc/security/cacerts/
# chmod 644 /system/etc/security/cacerts/9a5ba575.0
# exit
```

- e. Now that the certificate is installed, we need to configure the Android device to route its network traffic through Burp Suite and set up port forwarding so that Burp Suite can intercept and inspect the traffic properly. For more info, go [here](#).

```
./adb shell settings put global http_proxy localhost:3333
./adb reverse tcp:3333 tcp:8082
```

Now that everything is ready, we can officially start our dynamic analysis.

10. Start interception (by having 'Intercept is on') on Burp Suite. This should start intercepting network traffic, and you can read any packet that you select.
11. Launch the malicious app (giving it all its permissions), and start filling in the form, while observing the packets on Burp Suite.

Remember not to enter any personal information.

Note: On Burp Suite, drop all packets after analysis in order for them not to be forwarded to the attacker.



Questions to be answered in your report!

1. How many packets are sent? What is their destination.
2. What is inside these packets?
3. What protocol is being used: HTTP or HTTPS? What can you deduce?
4. Can you confirm that the app sends an SMS message? What is their content?
5. What happens after you submit the form and try to re-enter the app? Why do you think this happens?
6. Do your findings comply with your static analysis? Explain.