# CSEC 467 Lab 2

## Instructions

1. Create a virtual machine to run MobSF and run MobSF.
    a. This can be a basic VM that uses the dockerized version of MobSF, a Mobexler VM, or any other VM capable of running MobSF. It does not matter too much as long as MobSF is running.
    b. Note: You will need to export PDFs from MobSF, which may require an extra setup.

2. Download the following applications from APKPure (or acquire the APKs via some other mechanism).
    a. RIT Mobile
    b. Tigersafe
    c. One other application of your choice.
        i. You must verify that you are not prohibited from reverse engineering the mobile application by checking the terms of service and the maker's website for any additional terms of service.

3. Upload these APKs to your MobSF instance. **<u>DO NOT PERFORM DYNAMIC ANALYSIS ON THESE APPS.</u>**
4. Answer the critical thinking questions based on the MobSF static analysis.
5. Generate PDF reports for each of the applications and submit those with your assignment.

## Critical Thinking Questions

1. Document how you determined that you were not prohibited from reverse engineering the third application that you selected.
    a. Mihon is an open-source application that discovers, downloads, and reads manga comics from various sources, with features like progress tracking, multiple reading modes, and customization options; essentially, it functions as a platform to access and organize manga content on your phone that I personally use regularly

2. Were the mobile applications developed by the same team? Justify your answer.
   No, the mobile applications were not developed by the same team.
    a. **RITMobile**
        i. The **RITMobile** app is developed by Rochester Institute of Technology (RIT), as indicated by the X.509 certificate subject and issuer:
            1. **Organization**: Rochester Institute of Technology
            2. **Organizational Unit**: Information & Technology Services
            3. **Country**: USA (New York)
    b. **Tigersafe**
        i. The **Tigersafe** app, however, is developed by **CutCom Software Inc**., as indicated by its certificate:
            1. **Organization**: CutCom Software Inc.
            2. **Location**: Kingston, Ontario, Canad
    c. **Mihon**
        i. The **Mihon** app appears to be developed by **WorkshopOfAntsyLich**

3. Briefly describe any risks attached to the code signing certificates used for the RIT mobile applications.
    a. **RITMobile**
        i. The app is signed with **SHA1withRSA**, which is vulnerable to hash collision attacks
        ii. The certificate is valid from 2011 to 2039, which is an unusually long period. A compromised private key could remain a security risk for decades
    b. **Tigersafe**
        i. The application uses v1 signature scheme, making it vulnerable to the Janus vulnerability on Android 5.0-8.0 if signed only with v1

ii. Like RITMobile, it is signed with SHA1withRSA, which is vulnerable to hash collisions

c. **Mihon**

i. The application is signed using SHA256withRSA, which is more secure than SHA1.

ii. However, it allows user-installed certificates, increasing the risk of man-in-the-middle (MITM) attacks

4. What permissions do each of the RIT mobile applications request? For each permission, provide a reason that the permission was likely needed for the application's functionality. This must be specific to the application's functionality and not just a general description of what the permission is used for.

   a. **RITMobile**

      i. `ACCESS_FINE_LOCATION` & `ACCESS_COARSE_LOCATION`: Needed for location-based services, such as campus maps and bus tracking.

      ii. `CAMERA`: Possibly used for QR code scanning or photo uploads.

      iii. `INTERNET`: Required for fetching real-time data from RIT services.

      iv. `USE_BIOMETRIC`: Likely for secure logins

   b. **Tigersafe**

      i. `ACCESS_FINE_LOCATION` & `ACCESS_COARSE_LOCATION`: Essential for location-based safety alerts.

      ii. `SYSTEM_ALERT_WINDOW`: Allows emergency alerts to be displayed over other apps.

      iii. `CAMERA`: Could be for emergency reporting with photos.

      iv. `WRITE_EXTERNAL_STORAGE`: Likely for saving reports or media for emergency situations

      v.

   c. **Mihon**

      i. `WRITE_EXTERNAL_STORAGE`: Required for saving downloaded content.

      ii. `REQUEST_INSTALL_PACKAGES`: Could be for installing updates or external plugins.

      iii. `INTERNET`: Necessary for fetching data from online sources.

      iv. `POST_NOTIFICATIONS`: Allows the app to send updates

5. Briefly review the source code for the RIT mobile applications. Are there any obfuscated classes? If so, what applications have obfuscation?

   a. **RITMobile**

      i. The app includes obfuscated class names like `a1/b.java`, `c/i.java`, and `com/parse/ParseRequest.java`, indicating the use of code obfuscation tools

   b. **Tigersafe**

        i. Uses R8 without markers, which suggests that code shrinking and obfuscation are applied
    c. **Mihon**
        i. Also uses R8 without markers, along with Anti-VM and Anti-Debug checks, indicating some level of obfuscation

6. Examine the "Security Analysis" section of the MobSF output for the RIT applications. Are there any findings that are consistent across both mobile applications? If there are, why might this/these findings be common?
    a. **RITMobile**
        i. **Cleartext Traffic Enabled:** RITMobile and Tigersafe allow unencrypted HTTP traffic, which is a security risk
    b. **Tigersafe**
        i. **Unprotected Exported Components:** Multiple apps have exported activities/services that could be accessed by other applications, increasing the attack surface.
    c. **Mihon**
        i. **Weak Cryptography:** Both RITMobile and Tigersafe use SHA1withRSA, which is vulnerable to hash collisions

    These findings may be common due to legacy coding practices and lack of modern security updates.

7. Examine the "Code Analysis" section of the MobSF output for the RIT applications. Are there any findings that you believe are false positives? If so, briefly explain - with evidence - why these are false positives.
    a. **RITMobile**
        i. **RITMobile logs sensitive information:** While logging is flagged as a security issue, it may only be logging non-sensitive app events.
    b. **Tigersafe**
        i. **Tigersafe may request root privileges:** While the app contains references to JailMonkey (a root detection library), it does not necessarily mean it seeks root access
    c. **Mihon**
        i. Did not find any flagged issues that appeared to be clear false positives or true positives based on the report's contents

8. Examine the "Code Analysis" section of the MobSF output for the RIT applications. Are there any findings that you believe are true positives? If so, briefly explain - with evidence - why these are true positives.
    a. **RITMobile**
        i. No clear false or true positives were detected in the Mihon application based on the MobSF reports.
    b. **Tigersafe**

   i. Tigersafe allows cleartext traffic: This means data can be intercepted, making MITM attacks easier

c. **Mihon**

   i. **Mihon allows the installation of external packages:** This increases the risk of **malicious installations**

9. For each of the RIT mobile applications, provide a general assessment of the risk associated with the use of that application. You should assess whether the application is low, medium, or high risk and justify your position. You should write approximately one paragraph per application. You do not have to assess risk from a purely security mindset. You can also factor privacy into your assessment.

   a. **RITMobile - Medium Risk**

      i. While useful for campus navigation and updates, it has security concerns such as cleartext traffic and exported activities that could be exploited. However, it does not exhibit any highly dangerous behaviors

   b. **Tigersafe - Medium to High Risk**

      i. As a safety app, Tigersafe's security should be top-notch, but it has cleartext traffic, unprotected exported activities, and potentially dangerous permissions like `SYSTEM_ALERT_WINDOW` which could be misused

   c. **Mihon - Medium Risk**

      i. Since Mihon is open-source, it benefits from community scrutiny, but permissions like package installation and external storage access introduce privacy risks