MONASH University

FIT2081 Mobile application development - S1 2021



| | |
|---|---|
| **Started on** | Monday, 24 May 2021, 8:38 PM |
| **State** | Finished |
| **Completed on** | Monday, 24 May 2021, 10:44 PM |
| **Time taken** | 2 hours 5 mins |
| **Grade** | **5.86** out of 10.00 (**59**%) |

Print friendly format



Question **1**

Complete

Mark 5.86 out of 10.00

Question 1

**Q1- Briefly explain how could protect your Android application against reverse engineering.**



Reverse engineering cannot be eliminated 100%, so I have to take some methods against it as much as possible.

**Using ProGuard** manipulates Java the way you tell it with your configuration files and the rules they contain. It is a command-line tool that shrinks, optimizes, obfuscates and even pre-verifies the code. It can improve efficiency, reduce the size to protect the Android application.

**Secure the user credentials with extra care** to avoid reverse engineering of the application. The app owners require user credentials. In such cases use a credential object that contains user sign-in information.

**Save important code chunks on the server** This way of preventing apps from reverse engineering is to remove the code from the application and move it to any web service that is encrypted server-side language.

**Hide API keys** Use Private/public key exchange to hide and protect the API key, which can protect the Android application against reverse engineering.

**Hash function** Using the hash function like cryptographic hash function, which can resistant to collisions and not too fast. It also protects Android applications.

Comment:

(Q1=100% Q2=20% Q3=100% )

Question **2**

Complete

Not graded

Question 2

**Q2 - You are developing an application that must have two different releases (apps): demo and full releases. The demo must contain one service only, while the full release must have three extra services.**

**Briefly explain your approach to develop these two releases (apps) such that you don't duplicate the common source codes and resources between them.**

I think using the approach called an incremental developmental model. In other words, the best approach would be to incorporate an agile release management framework. Because elements are broken down into multiple independent software cycle modules. The management, scheduling, planning, controlling, etc. are built through all stages. It applies to the demo and full release of the software.

This method works by first releasing the demo fairly straightforward. This release is also broken into various subsections. Once the demo is complete, the modifications are approved and release is planned. The demo is made to go through acceptance testing and then it is deployed. The complete software or application is then released under real-time monitoring.

Every succeeding release of the system will have the function of the previous release until all planned functionality has been implemented. The full release can be broken into various sprints with one objective each. As the project moves from one to another sprint, we can check the consistency of functions in the application, which does not duplicate the common source codes and resources between them.

Question **3**

Complete

Not graded



Question 3

**Q3 - a) What are the consequences if the private key of an app is lost but not compromised (i.e. no one else acquires it)?**



The existing app on Google Play App Store can't be updated in the future. If you lose or misplace your key, you will not be able to publish updates to your existing app.

If the private key is not compromised but is somehow lost and nobody actually has acquired it, there will be a need to get a new app. You cannot regenerate a previously generated key. This is because there will be issues regarding app updates. One will have to download a different or new model of the app.

Question **4**

Complete

Not graded

Question 4

**b) - What are the consequences if the private key of one of your apps becomes compromised without your knowledge (i.e.**

**someone else acquires it)?**

If the key has been compromised, it should be notified to the certificate authority. This is because with the key anyone can decode the meaning of encrypted data associated with that key. Your authoring identity and the trust of the user can be compromised.

Such a person could also sign and distribute apps under your identity that attack other apps or the system itself, or corrupt or steal user data.

Question **5**

Not answered

Not graded

Question 5



◄ Week 11: Workshop Quiz

Jump to...

Week 2: Pre-reading Quiz ►