**CSCE 548 Project 3**

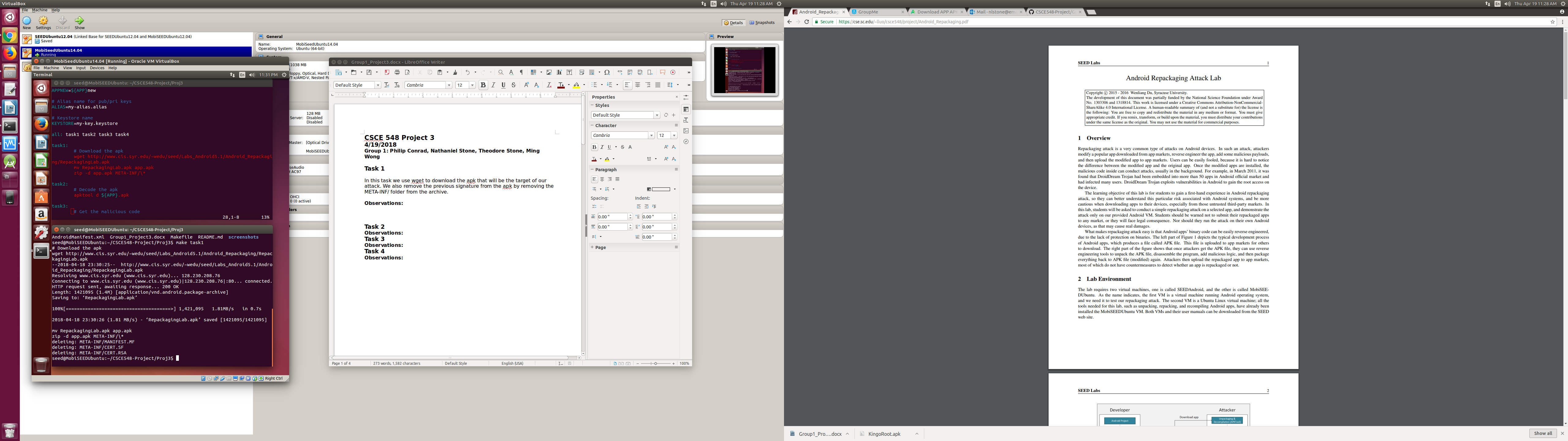
**4/19/2018**

**Group 1: Philip Conrad, Nathaniel Stone, Theodore Stone, Ming Wong**

**Task 1**

In this task we use wget to download the apk that will be the target of our attack. We also remove the previous signature from the apk by removing the META-INF/ folder from the archive.

**Observations:**

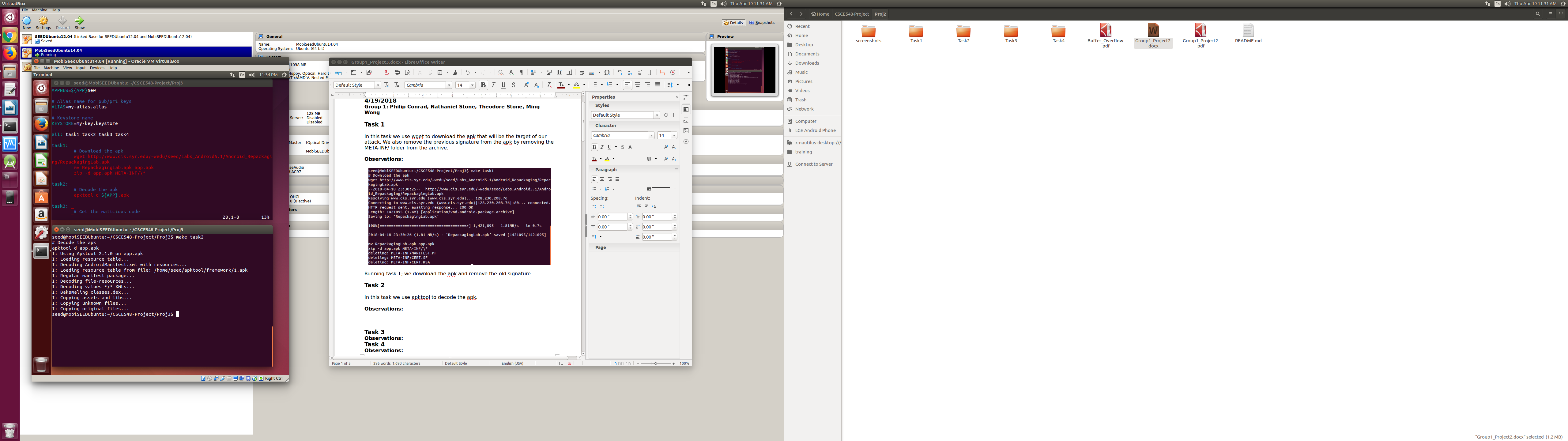


Running task 1; we download the apk and remove the old signature.

**Task 2**

In this task we use apktool to decode the apk.

**Observations:**



Using apktool to decode the apk.

**Task 3**

**Observations:**

**Task 4**

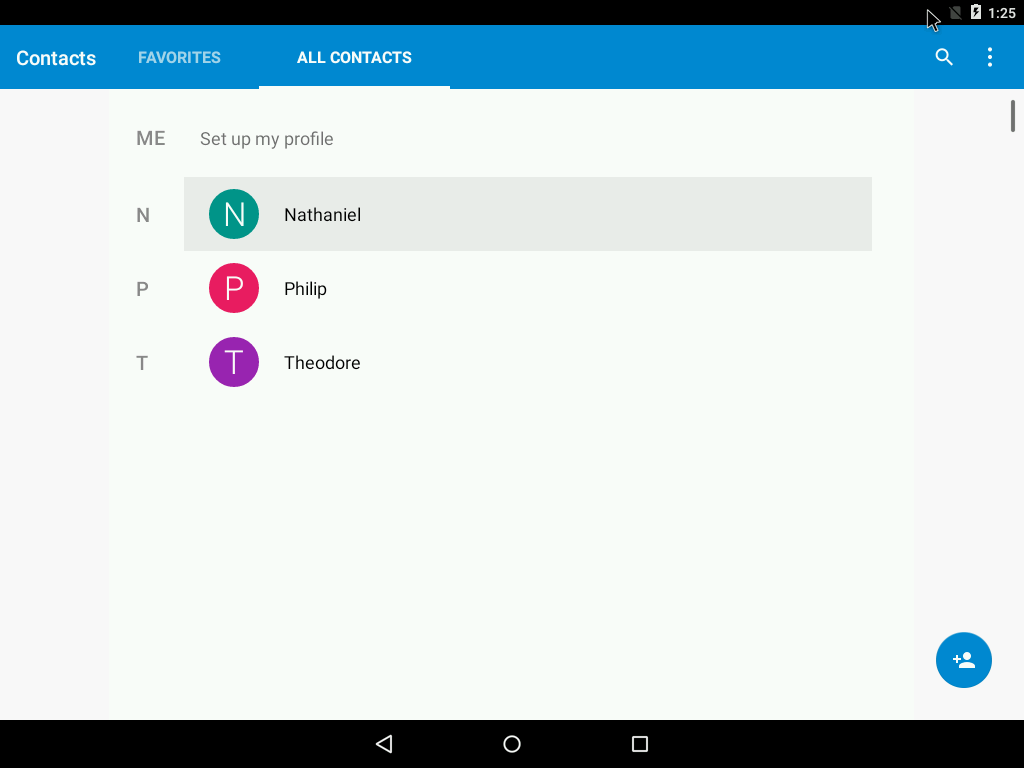
**Observations:**

**Task 5**

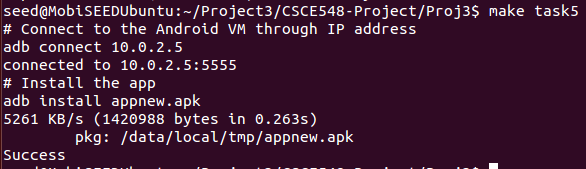
The goal of this task is to install the malicious app onto the android device. We first determine the IP address of the android by typing *netfcg* on the android terminal. After that, we connect our Ubuntu machine to our android emulator to install our newly modified app.

**Observations:**

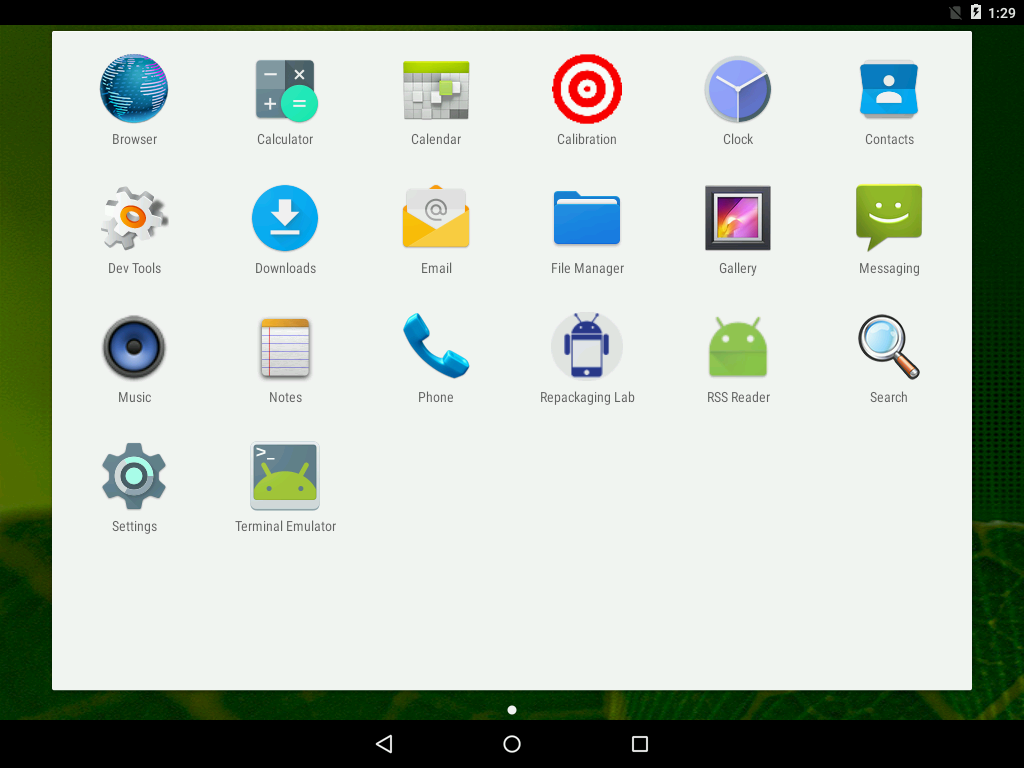
Before we install the app, there are three contacts.



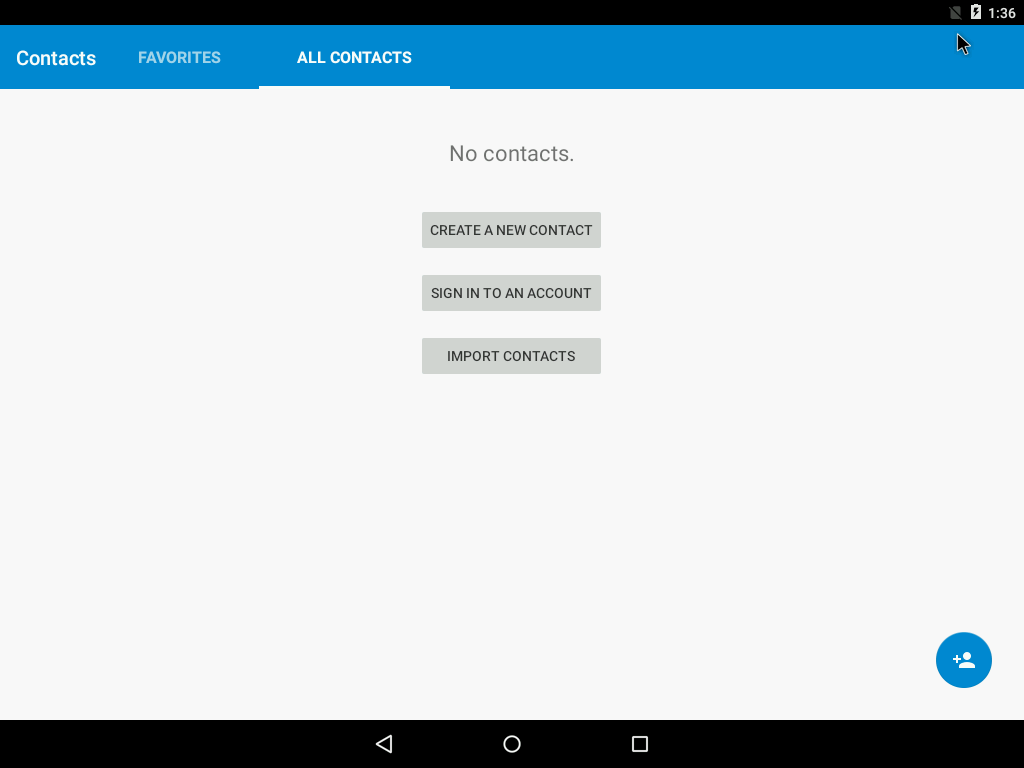
We use these commands to perform our app installation.



We observe that our appnew.apk is installed onto the android device. It still uses the original name of the app, which is Repacking Lab.



When we launch the Repackaging Lab once and restart the android device, we notice all our contacts have been obliterated by our successful repackaging attack.



**Lab Questions**

1. **Why is the repackaging attack not much a risk in iOS devices?**

A certificate authority, often Apple, signs all the code submitted to its App Store and does not allow runtime modification of the app. Code signing ensures the app has not been altered since it was signed, identifies the source of the code, and determines whether the code is trustworthy. In other words, this implies that the app that you submit to the App Store is that version of app offered from the App Store.

On the other hand, Google does not enforce code signing and allows runtime modification of the code. This means an attacker can modify the app after it is on the Google marketplace It is more difficult to repackage an app from Apple App Store because Apple’s code signing can be used to catch a malicious app.

1. **If you were Google, what decisions you would make to reduce the attacking chances of repackaging attacks?**

Google should recommend developers to obfuscate their source code. Google should also forces developers to provide some identity information likes social security number so that developers would be held accountable if their apps conduct malicious activities.

1. **Third-party markets are considered as the major source of repackaged applications. Do you think that using the official Google Play Store only can totally keep you away from the attacks? Why or why not?**

Using Google Play Store might help a bit. However, it does not prevent attackers from submitting a repackaged app to the store that you later download.

1. **In real life, if you had to download applications from untrusted source, what would you do to ensure the security of your device?**

The android system has a built-in security feature called Google Play Protect. Google Play Protect checks on apps from Google play store before you download it. It also scans and warns you about any potentially harmful apps from other sources and removes known harmful apps from your device. You must make sure that Google Play Protect on your device is turned on.

You should check the app permissions and try to give the minimum level permissions. You should turn off Wi-Fi when you are not using your phone.