Solution for Scenario 3: Mobile Malware

1. Starting from the clue about the installation of Skype we investigate the installed packages. Looking trough the application list (Launcher → Device group → Settings → Apps → Downloaded). Skype is installed. A Network Watchdor is installed but it does not appear in the Launcher. It might be a system app.
   We enter adb shell by running `adb shell`. In this new shell `pm list packages`. We see the list of installed packages. The fact that there are two skype packages looks strange. We will download them both. To see what is the path of the corresponding apk: `pm list packages -f | grep skype`. We exit the adb shell: `exit`. To get the files from the linux shell we run adb pull with the paths obtained int the previous step:
   ```
   adb pull /data/app/com.skype.raider-1.apk
   adb pull /data/app/com.skype.rayder-1.apk
   ```
   We unzip the files, each in it's own folder:
   ```
   unzip com.skype.raider-1.apk -d com.skype.raider
   unzip com.skype.rayder-1.apk -d com.skype.rayder
   ```
   In each of the folders we go into the META-INF subfolder. We use check the certificate to see who issued it. Use `keytool -printcert -file CERT.RSA`. We see that com.skype.raider certificate is issued by Skype and com.skypte.rayder's is Android Debug.
   You have got the malware: **com.skype.rayder**.

2. Using `adb logcat` some strange log messages about files being uploaded. While `adb logcat -b radio` shows SMS beeing sent. ***So SMS and Internet traffic are the two channels of communication used for exfiltration.***

3. In the `adb logcat -b radio` log we see a ***SMS being sent*** to a short number. This usually are ***premium taxed numbers.***

4. using `aapt d badging <apkfile>` and `aapt d xmltree <apkfile> AndroidManifest.xml` you get the content of the manifest, providing an overview of the application. We can see the names of the services(.WatchDogService), providers (none), receivers(.Receiver) and activities (one activity: .Activity) of the application.

5. Application has SEND_SMS and INTERNET permissions. So SMS and Internet traffic are the two channels of communication used for exfiltration, confirming our initial assertion from point 2.
   Using `dex2jar <apkfile>` and `jd-gui <jarfile>` we can see the code of the application.
   Based on the imports, ***two protocols, http and ftp, are used for web traffic.***

6. We can see clearly in the Service code (file WatchDogService.java) the username and password are in clear as fields of the class.

7. Analyzing the code of the Service we can conclude that the documents (extensions ending with .doc, .ppt, etc) are exfiltrated trough FTP. The method isInteresting returns true if a filename has standard document extension or ends with password. The list of interesting files is exfiltrated by the fuction upload().
   Also the contacts are read and POST-ed via HTTP on the attackers server. There is a

LocationListener that gets notified each time the target moves. It handles the notification by posting the targets location to the attacker server.

8. Steps to clean the malware. Must take note of the Device Administrator issue. Go to Launcher→Settings → Personal group → Security → Device Administration group → Device Administrators. Uncheck Clear Sound. Proceed to uninstall the application Launcher → Device group → Settings → Apps → Downloaded. Select Network WatchDog. Tap Uninstall. Confirm by tapping OK. The malware should be uninstalled now and the phone clean.

9. Advisory: never install apks from untrusted sources. To make it harder to do this accidentally only install apks from the official Play Market. Disable *Unkown sources* and enable *Verify apps* (in *Launcher→Settings → Personal group → Security → Device Administration group*).