**QARK:**

QARK is an interactive tool developed for Linux systems. The tool can be downloaded from its github page and run using the command \textit{python qarkMain.py}. Upon running the command, QARK provides two choice for analysing the application. Either input the apk file or the Source code. The user is then prompted to enter a choice of path to the apk file or pull an existing apk from a site. Once the apk file is provided, QARK inspects the manifest file. It determines the attack surfaces from the AndroidManifest.xml and notifies the user if any vulnerabilities are found.

In the case of TSB application, the following warning has been generated.

The following activity are exported, but not protected by any permissions. Failing to protect activity could leave them vulnerable to attack by malicious apps. The activity should be reviewed for vulnerabilities, such as injection and information leakage.

**com.halifax.halifax.GrappleActivity**

POTENTIAL VULNERABILITY - The following receiver are exported and protected by a permission, but the permission can be obtained by malicious apps installed prior to this one. More info: https://github.com/commonsguy/cwac-security/blob/master/PERMS.md. Failing to protect receiver could leave them vulnerable to attack by malicious apps. The receiver should be reviewed for vulnerabilities, such as injection and information leakage.

**co.uk.apptivation.nga.library.googlecloudmessaging.GcmBroadcastReceiver**

**Drozer:**

Drozer is another static analysis tool that helps to identify vulnerabilities in mobile applications.

The main requirements for installing drozer are

1. Java Development Kit (JDK) 1.6
2. Python 2.7
3. Android SDK

The installation package comes with a drozer Agent.apk which is installed in the mobile device or emulator using

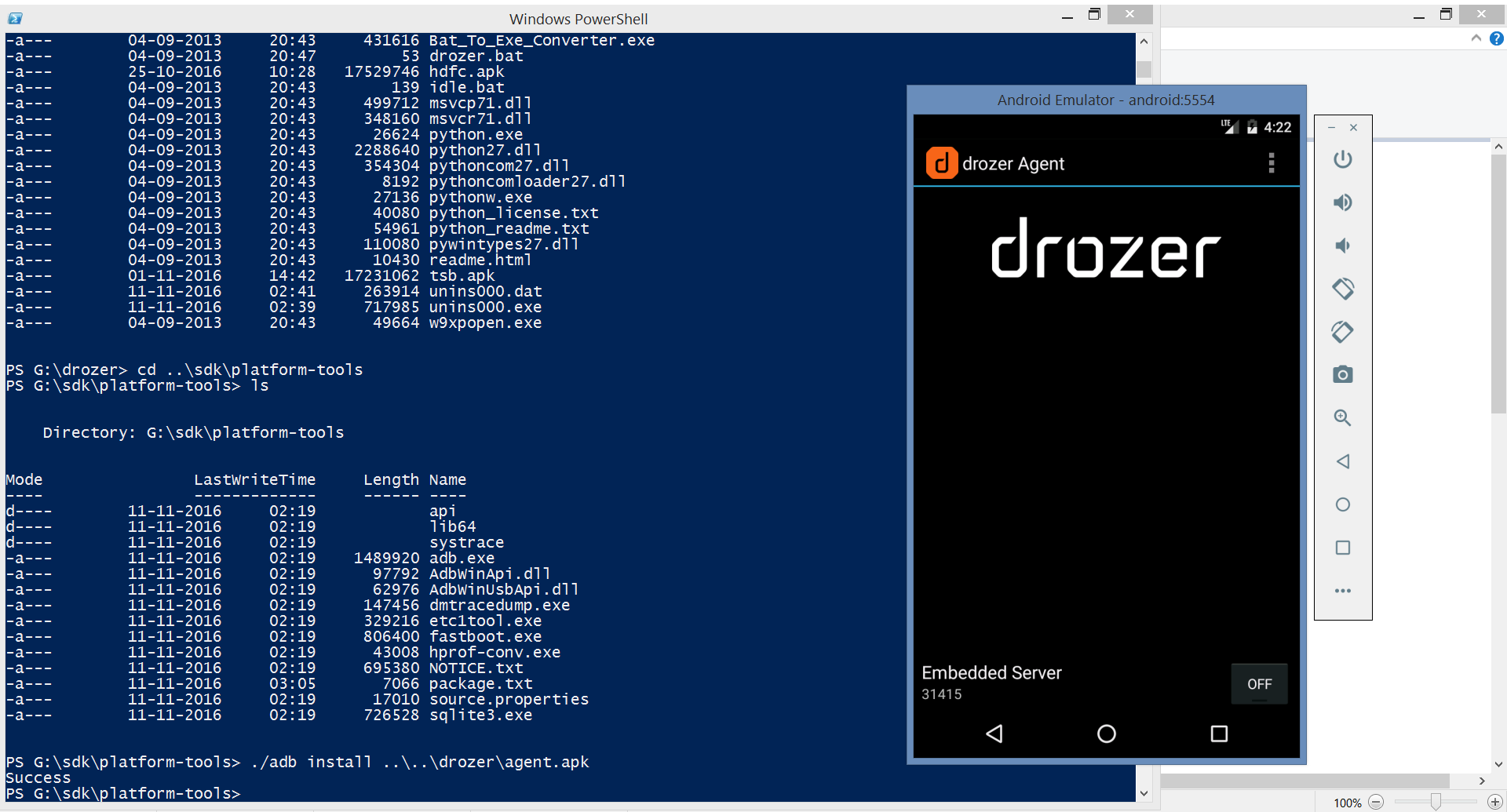
$ adb install agent.apk

For this experiment, an android emulator was used and the Agent.apk was installed. Drozer console is installed in the PC and the Agent is installed in the emulator. The Agent and the PC are connected by a TCP socket opened by the Agent using port 31415.

$ adb forward tcp:31415 tcp:3141

Then the agent and the PC are connected by the command

$ drozer console connect



The command

dz> run app.package.list

provides the list of all the application packages installed in the device or emulator in this case. The following is the analysis of TSB banking app.

dz> run app.package.list -f tsb

uk.co.tsb.mobilebank (TSB)

**Information about the application package:**

dz> run app.package.info -a uk.co.tsb.mobilebank

Package: uk.co.tsb.mobilebank

Application Label: TSB

Process Name: uk.co.tsb.mobilebank

Version: 10.1

Data Directory: /data/user/0/uk.co.tsb.mobilebank

APK Path: /data/app/uk.co.tsb.mobilebank-1/base.apk

UID: 10054

GID: [3003]

Shared Libraries: null

Shared User ID: null

Uses Permissions:

- com.android.vending.CHECK\_LICENSE

- android.permission.READ\_PHONE\_STATE

- android.permission.INTERNET

- android.permission.ACCESS\_NETWORK\_STATE

- android.permission.ACCESS\_WIFI\_STATE

- android.permission.READ\_CONTACTS

- android.permission.WRITE\_EXTERNAL\_STORAGE

- android.permission.ACCESS\_FINE\_LOCATION

- android.permission.ACCESS\_LOCATION\_EXTRA\_COMMANDS

- android.permission.ACCESS\_COARSE\_LOCATION

- com.google.android.providers.gsf.permission.READ\_GSERVICES

- uk.co.tsb.mobilebank.permission.MAPS\_RECEIVE

- android.permission.VIBRATE

- android.permission.GET\_ACCOUNTS

- android.permission.WAKE\_LOCK

- com.google.android.c2dm.permission.RECEIVE

- uk.co.tsb.mobilebank.C2D\_MESSAGE

- android.permission.READ\_EXTERNAL\_STORAGE

Defines Permissions:

- uk.co.tsb.mobilebank.permission.MAPS\_RECEIVE

- uk.co.tsb.mobilebank.C2D\_MESSAGE

**The ‘app.package.attacksurface’ command identifies the attack surfaces of the app:**

dz> run app.package.attacksurface uk.co.tsb.mobilebank

Attack Surface:

1 activities exported

1 broadcast receivers exported

0 content providers exported

0 services exported

**To see which activities are exported by TSB:**

dz> run app.activity.info -a uk.co.tsb.mobilebank

Package: uk.co.tsb.mobilebank

com.halifax.halifax.GrappleActivity

Permission: null

**To see content providers exported by the app:**

dz> run app.provider.info -a uk.co.tsb.mobilebank

Package: uk.co.tsb.mobilebank

No matching providers.

dz> run app.service.info -a uk.co.tsb.mobilebank

Package: uk.co.tsb.mobilebank

No exported services.

**Injection vulnerabilities:**

dz> run scanner.provider.injection -a uk.co.tsb.mobilebank

Scanning uk.co.tsb.mobilebank...

Not Vulnerable:

No non-vulnerable URIs found.

Injection in Projection:

No vulnerabilities found.

Injection in Selection:

No vulnerabilities found.

dz> run scanner.provider.traversal -a uk.co.tsb.mobilebank

Scanning uk.co.tsb.mobilebank...

Not Vulnerable:

No non-vulnerable URIs found.

Vulnerable Providers:

No vulnerable providers found.

**Androbugs:**

|  |  |  |
| --- | --- | --- |
| **TSB banking app** | | |
|  | **Checking** | **Result** |
|  |  |  |
| **Critical** | Runtime Command Checking | app uses critical function 'Runtime.getRuntime().exec("...")' |
|  | App Sandbox Permission Checking | Found Base64 encoding "String(s)" (Total: 1) |
|  | SSL Connection Checking | URLs that are NOT under SSL (Total:6) |
|  | SSL Implementation Checking | handler.proceed(); should not be used. inside those methods in extended "WebViewClient", which allows the connection even if the SSLCertificate is invalid (MITM Vulnerability) |
|  | WebView RCE Vulnerability Checking |  |
|  |  |  |
| **Warning** | Dynamic Code Loading | Dynamic code loading(DexClassLoader) found |
|  | External Storage Accessing | External storage access found |
|  | <Sensitive\_Information> Getting IMEI and Device ID | This app has code getting the "device id(IMEI)" but there are problems with this "TelephonyManager.getDeviceId()" approach |
|  | <Sensitive\_Information> Getting ANDROID\_ID | This app has code getting the 64-bit number "Settings.Secure.ANDROID\_ID" |
|  | SSL Certificate Verification Checking | make sure this app has the conditions to check the validation of SSL Certificate |
|  | WebView Local File Access Attacks Checking | Found "setAllowFileAccess(true)" or not set(enabled by default) in WebView. The attackers could inject malicious script into WebView and exploit the opportunity to access local resources. This can be mitigated or prevented by disabling local file system access. |
|  | WebView Potential XSS Attacks Checking | Found "setJavaScriptEnabled(true)" in WebView, which could exposed to potential XSS attacks |
|  |  |  |
| **Notice** | AndroidManifest Adb Backup Checking | ADB Backup is ENABLED for this app (default: ENABLED) |
|  | Android SQLite Databases Vulnerability Checking | This app is using Android SQLite databases but it's "NOT" suffering from SQLite Journal Information Disclosure Vulnerability |
|  | File Unsafe Delete Checking | Please make sure do not use "file.delete()" to delete essential files |
|  | <Debug><Hacker> Codes for Checking Android Debug Mode |  |
|  | <Hacker> APK Installing Source Checking | This app has code checking APK installer sources(e.g. from Google Play, from Amazon, etc.). It might be used to check for whether the app is hacked by the attackers. |
|  | <Signature><Hacker> Getting Signature Code Checking | This app has code checking the package signature in the code. It might be used to check for whether the app is hacked by the attackers |
|  | Native Library Loading Checking | Native library loading codes(System.loadLibrary(...)) found |
|  |  |  |
| **Info** | <Command> Executing "root" or System Privilege Checking | Did not find codes checking "root" permission(su) or getting system permission |
|  | <Database> SQLiteDatabase Transaction Deprecated Checking | Ignore checking SQLiteDatabase:beginTransactionNonExclusive" because your set minSdk >= 11 |
|  | <Database> Android SQLite Databases Encryption (SQLite Encryption Extension (SEE)) | This app is "NOT" using SQLite Encryption Extension (SEE) on Android (http://www.sqlite.org/android) to encrypt or decrpyt databases |
|  | <Database> Android SQLite Databases Encryption (SQLCipher) | This app is "NOT" using SQLCipher(http://sqlcipher.net/) to encrypt or decrpyt databases |
|  | <Debug> Android Debug Mode Checking | DEBUG mode is OFF(android:debuggable="false") in AndroidManifest.xml |
|  | Fragment Vulnerability Checking | Did not detect the vulnerability of "Fragment" dynamically loading into "PreferenceActivity" or "SherlockPreferenceActivity" |
|  | <Framework> Framework - MonoDroid | This app is NOT using MonoDroid Framework (http://xamarin.com/android) |
|  | <Hacker> Base64 String Encryption |  |
|  | <Database><Hacker> Key for Android SQLite Databases Encryption | Did not find using the symmetric key(PRAGMA key) to encrypt the SQLite databases (It's still possible that it might use but we did not find out) |
|  | <KeyStore><Hacker> KeyStore File Location | Did not find any possible BKS keystores or certificate keystore file (Notice: It does not mean this app does not use keysotre) |
|  | <KeyStore><Hacker> KeyStore Protection Checking | Ignore checking KeyStore protected by password or not because you're not using KeyStore |
|  | <Hacker> Code Setting Preventing Screenshot Capturing | Did not detect this app has code setting preventing screenshot capturing. |
|  | HttpURLConnection Android Bug Checking | Ignore checking "http.keepAlive" because you're not using "HttpURLConnection" and min\_Sdk > 8 |
|  | <KeyStore> KeyStore Type Checking | KeyStore 'BKS' type check OK |
|  | Google Cloud Messaging Suggestion | Nothing to suggest |
|  | Master Key Type I Vulnerability | No Master Key Type I Vulnerability in this APK |
|  | AndroidManifest Dangerous ProtectionLevel of Permission Checking | No "dangerous" protection level customized permission found (AndroidManifest.xml) |
|  | AndroidManifest PermissionGroup Checking |  |
|  | <Implicit\_Intent> Implicit Service Checking |  |
|  | AndroidManifest "intent-filter" Settings Checking |  |
|  | AndroidManifest Normal ProtectionLevel of Permission Checking |  |
|  | AndroidManifest Exported Lost Prefix Checking | No exported components that forgot to add "android:" prefix |
|  | AndroidManifest ContentProvider Exported Checking | No exported "ContentProvider" found (AndroidManifest.xml) |
|  | Codes for Sending SMS | Did not detect this app has code for sending SMS messages (sendDataMessage, sendMultipartTextMessage or sendTextMessage) |
|  | <System> AndroidManifest sharedUserId Checking | This app does not use "android.uid.system" sharedUserId |
|  | <SSL\_Security> SSL Implementation Checking (Verifying Host Name in Custom Classes) | Self-defined HOSTNAME VERIFIER checking OK |
|  | <SSL\_Security> SSL Implementation Checking (Verifying Host Name in Fields) | Critical vulnerability "ALLOW\_ALL\_HOSTNAME\_VERIFIER" field setting or "AllowAllHostnameVerifier" class instance not found |
|  | <SSL\_Security> SSL Implementation Checking (Insecure component) | Did not detect SSLSocketFactory by insecure method "getInsecure" |
|  | <SSL\_Security> SSL Implementation Checking (HttpHost) | DEFAULT\_SCHEME\_NAME for HttpHost check: OK |
|  | <SSL\_Security> SSL Certificate Verification Checking |  |
|  | Unnecessary Permission Checking | Permission 'android.permission.ACCESS\_MOCK\_LOCATION' sets correctly |
|  | Accessing the Internet Checking | This app is using the Internet via HTTP protocol |
|  | AndroidManifest System Use Permission Checking | No system-level critical use-permission found |
|  | App Sandbox Permission Checking | No security issues "MODE\_WORLD\_READABLE" or "MODE\_WORLD\_WRITEABLE" found on 'openOrCreateDatabase' or 'openOrCreateDatabase2' or 'getDir' or 'getSharedPreferences' or 'openFileOutput' |
|  |  |  |