HOW NOT TO BUILD ANDROID APPS

Jack Mannino

Jack@nvisiumsecurity.com
http://twitter.com/jack_mannino

nVisium SECURITY

Overview

- ❑ Are Things REALLY That Bad?
- ❑ Android Crash Course
- ❑ App Smackdown
- ❑ Q&A

<Me>

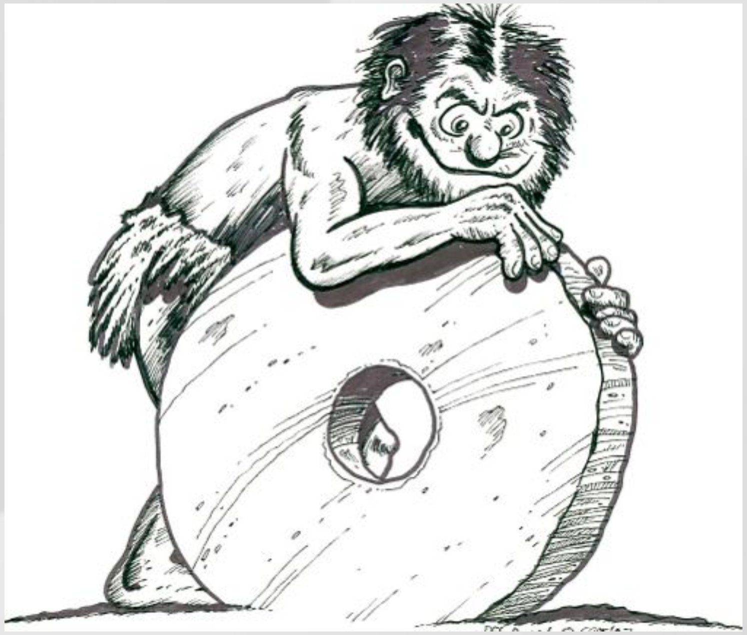
- Company co-founder
  - https://www.nvisiumsecurity.com
- OWASP Mobile Security Project Leader
- LIKES: mobile security, baseball
- DISLIKES: people who can't drive
- For more information, see:
  - http://lmgtfy.com/?q=jack+mannino

</Me>

nVisium SECURITY

Are Things REALLY That Bad?

- Depends on *who* you ask



Are Things REALLY That Bad?

- ❑ Android is “open’
- ❑ Ridiculous amount of malware
- ❑ Ecosystem is very fragmented
- ❑ Lots of W.T.F. in apps

nVisium SECURITY

Are Things REALLY That Bad?

- ❑Ice Cream Sandwich (4.0) solves some problems
- ❑FINALLY adds Address Space Layout Randomization (ASLR)
- ❑Full device encryption, keychain API

“One More Thing”

Unless you root….

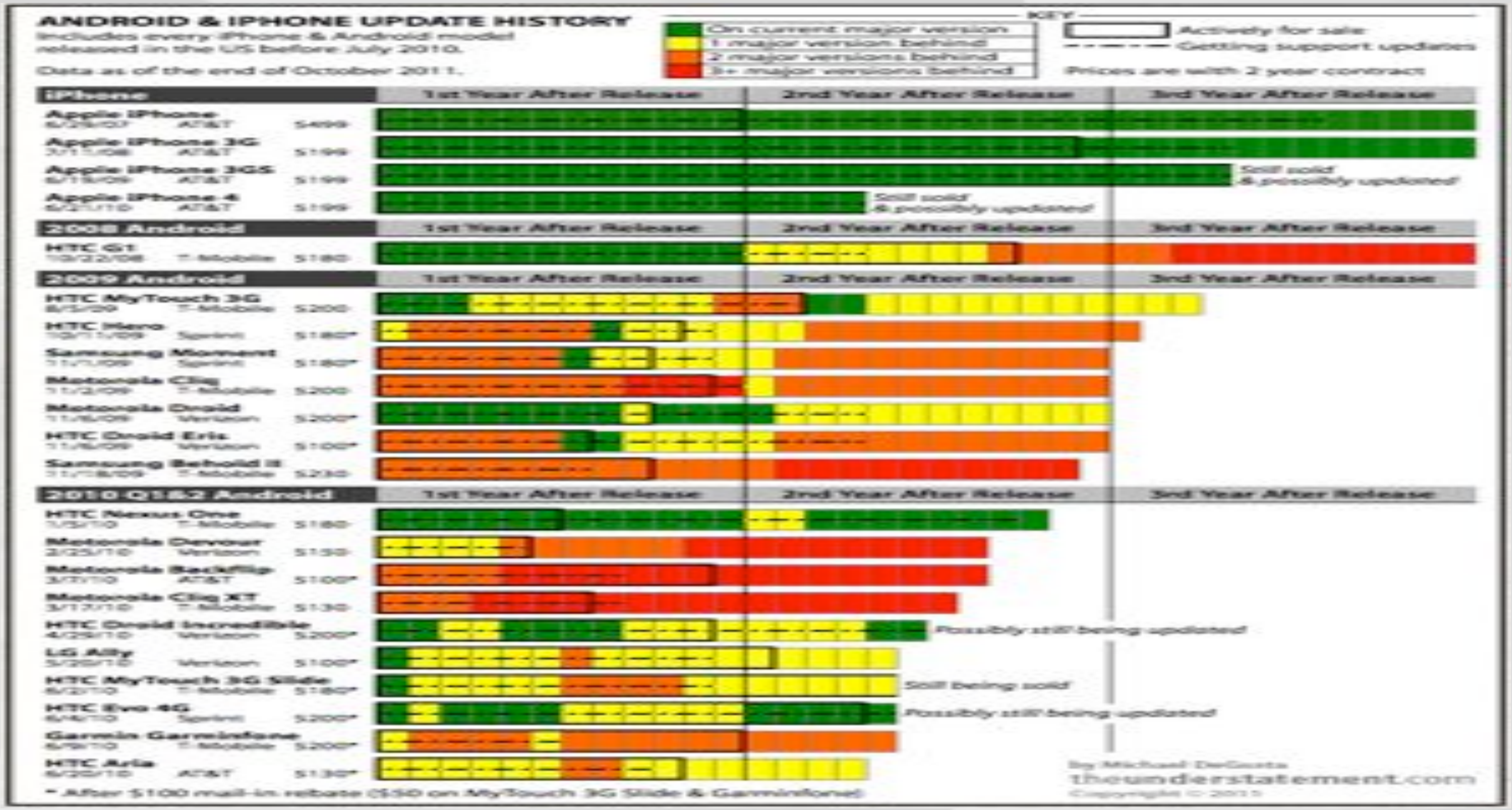
Or get a new device….

Or have a relatively new device….

You probably won’t get an update.

= ☹

Fragmentation!



Does This Affect Development?

- Yeah! Developers have 3 options:

1) Limit their potential users by requiring the latest and greatest OS

2) Distribute multiple versions of an app

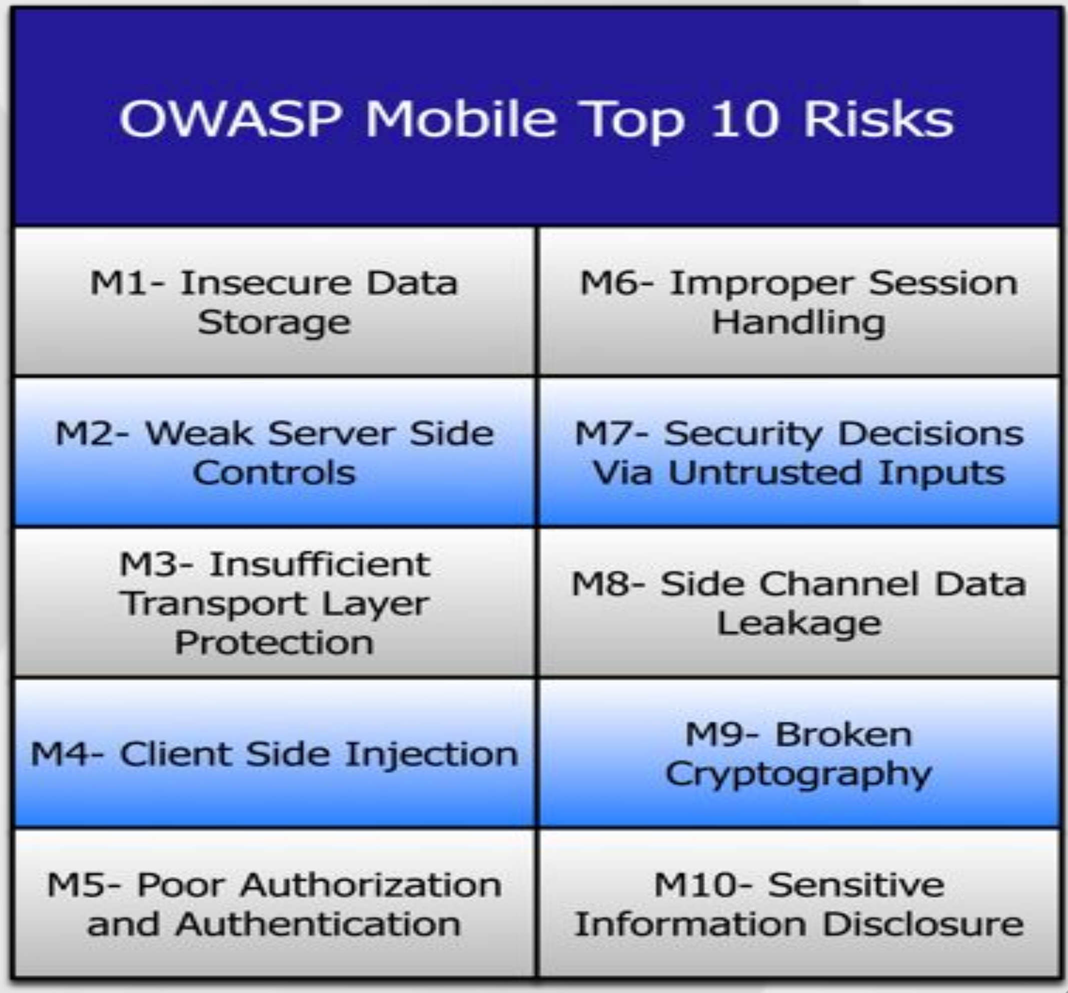
3) Ignore those **fancy** new security features (*most likely*)

nVisium SECURITY

Android Crash Course

- Dalvik VM != security
- Security model is largely user driven
- Applications are granted permissions during installation
- Apps are self-signed (very different than iOS)

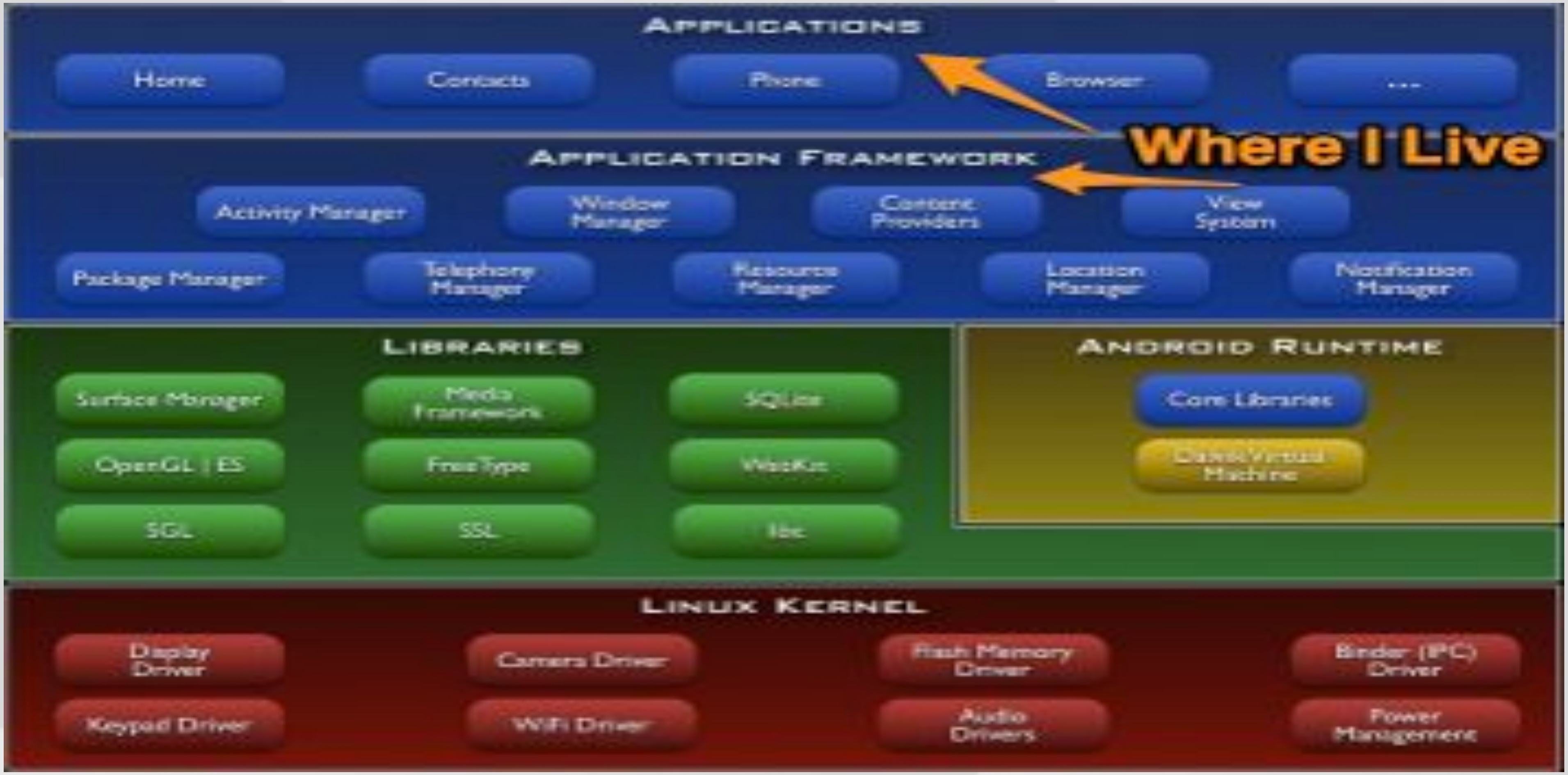
Top Mobile Risks



OWASP Mobile Top 10 Risks	
M1- Insecure Data Storage	M6- Improper Session Handling
M2- Weak Server Side Controls	M7- Security Decisions Via Untrusted Inputs
M3- Insufficient Transport Layer Protection	M8- Side Channel Data Leakage
M4- Client Side Injection	M9- Broken Cryptography
M5- Poor Authorization and Authentication	M10- Sensitive Information Disclosure

nVisium SECURITY

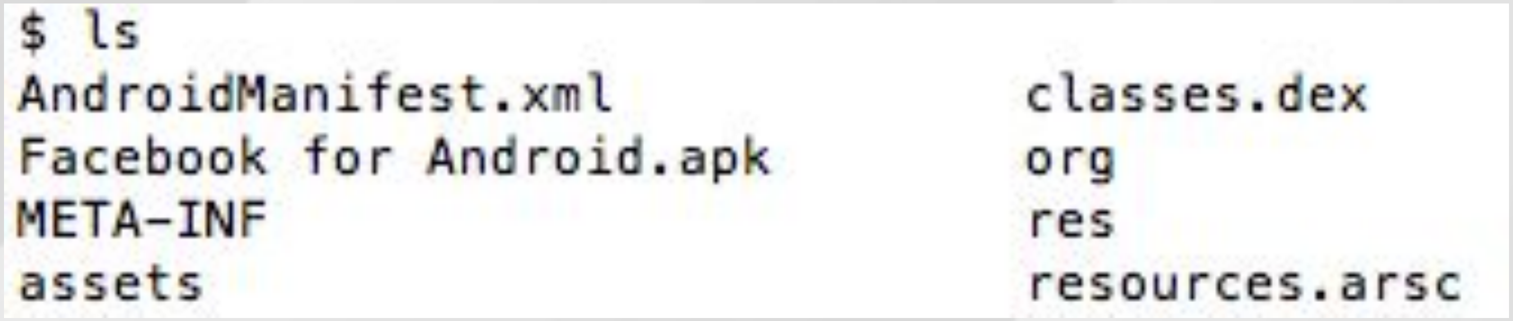
Android Architecture



nVisium SECURITY

Android App Anatomy

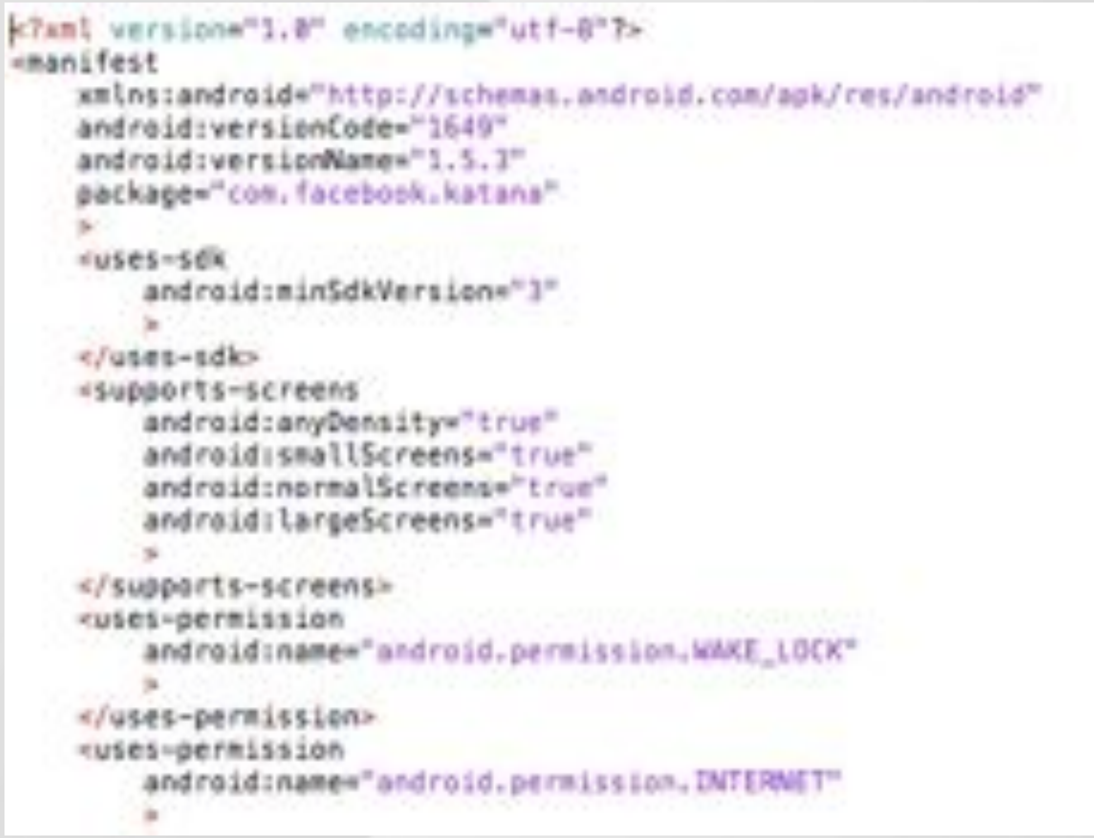
- APK = ZIP format
- classes.dex = app binary
- AndroidManifest.xml = configuration



```
$ ls
AndroidManifest.xml                 classes.dex
Facebook for Android.apk            org
META-INF                            res
assets                              resources.arsc
```

AndroidManifest.xml

- First thing you should look at
- This is where components and permissions are declared

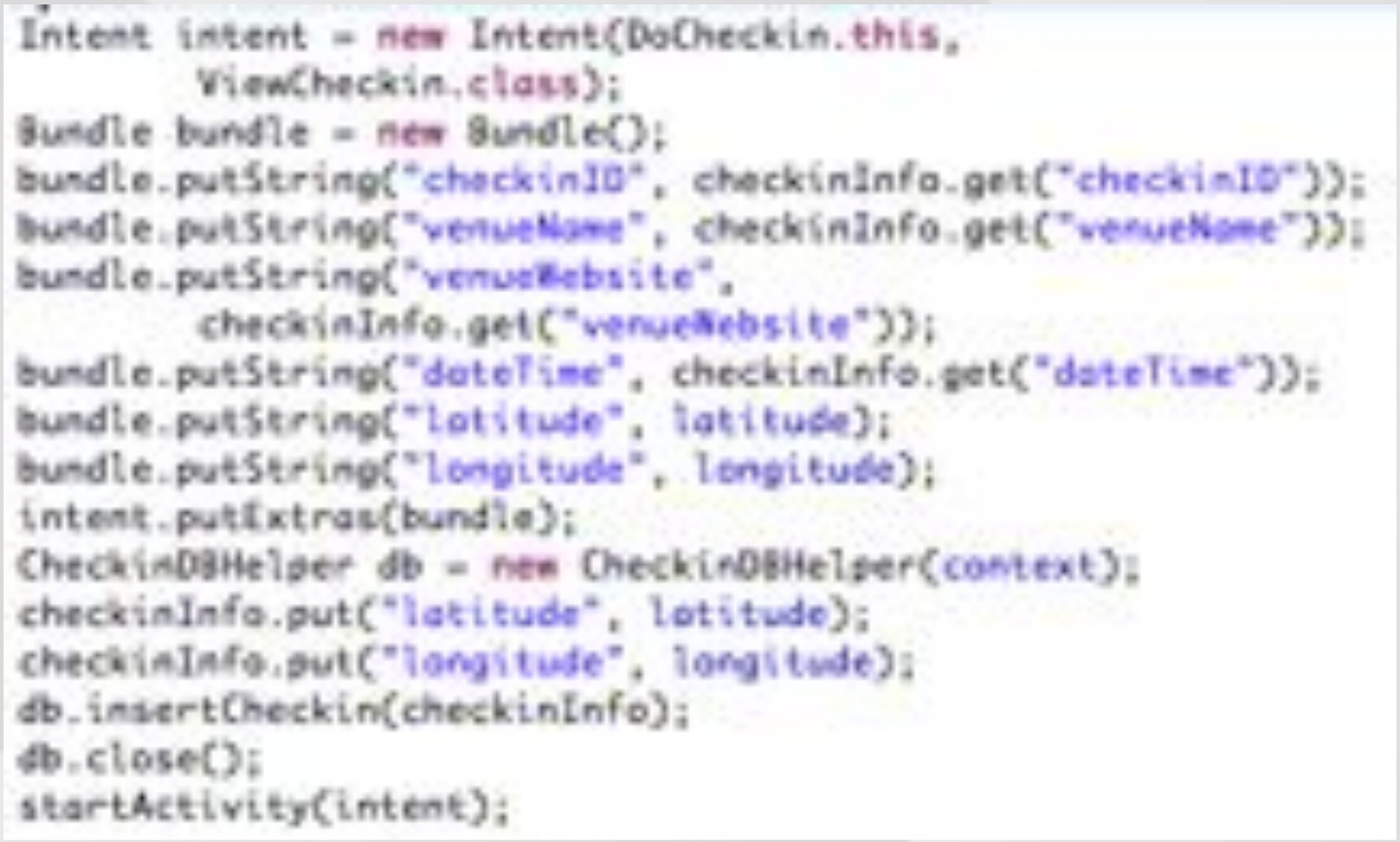


```
<?xml version="1.0" encoding="utf-8"?>
<manifest
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:versionCode="1649"
    android:versionName="1.5.3"
    package="com.facebook.katana"
    >
    <uses-sdk
        android:minSdkVersion="3"
        >
    </uses-sdk>
    <supports-screens
        android:anyDensity="true"
        android:smallScreens="true"
        android:normalScreens="true"
        android:largeScreens="true"
        >
    </supports-screens>
    <uses-permission
        android:name="android.permission.WAKE_LOCK"
        >
    </uses-permission>
    <uses-permission
        android:name="android.permission.INTERNET"
        >
```

Intents

- How components talk to each other
- Explicit or Implicit
- Intent Filters match actions, data type, categories, schemes, etc.
- Filters DO NOT provide security the way you'd think

Intents



```
Intent intent = new Intent(DoCheckin.this,
        ViewCheckin.class);
Bundle bundle = new Bundle();
bundle.putString("checkinID", checkinInfo.get("checkinID"));
bundle.putString("venueName", checkinInfo.get("venueName"));
bundle.putString("venueWebsite",
        checkinInfo.get("venueWebsite"));
bundle.putString("dateTime", checkinInfo.get("dateTime"));
bundle.putString("latitude", latitude);
bundle.putString("longitude", longitude);
intent.putExtras(bundle);
CheckinDBHelper db = new CheckinDBHelper(context);
checkinInfo.put("latitude", latitude);
checkinInfo.put("longitude", longitude);
db.insertCheckin(checkinInfo);
db.close();
startActivity(intent);
```

Components

- ❑ Activities
- ❑ Broadcast Receivers
- ❑ Content Providers (won’t cover it)
- ❑ Services (won’t cover it)

nVisium SECURITY

Activities

- Single screen with a UI
- Life cycle includes onCreate -> onDestroy

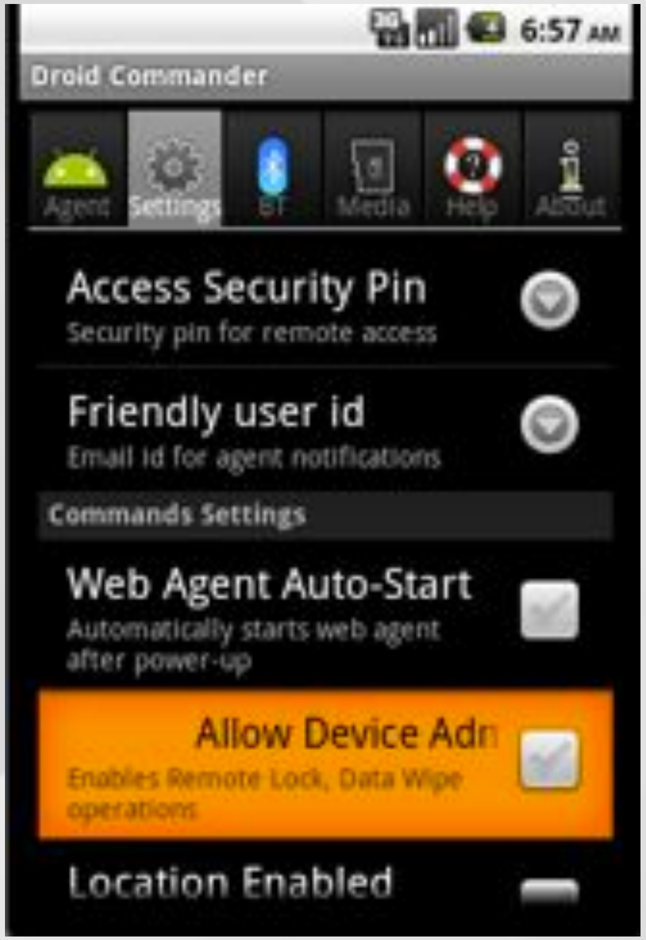
Broadcast Receivers

- Receive intents sent with sendBroadcast(intent) or sendOrderedBroadcast(intent)
- Can require a specific permission by sender
- However, not perfect

Now, Let's Have Fun

- ❑ A few real-world examples
- ❑ Sanitized real-world examples, to protect the "innocent"

Droid Commander


6:57 AM

Droid Commander

Access Security Pin
Security pin for remote access

Friendly user id
Email id for agent notifications

Commands Settings

Web Agent Auto-Start ✓
Automatically starts web agent after power-up

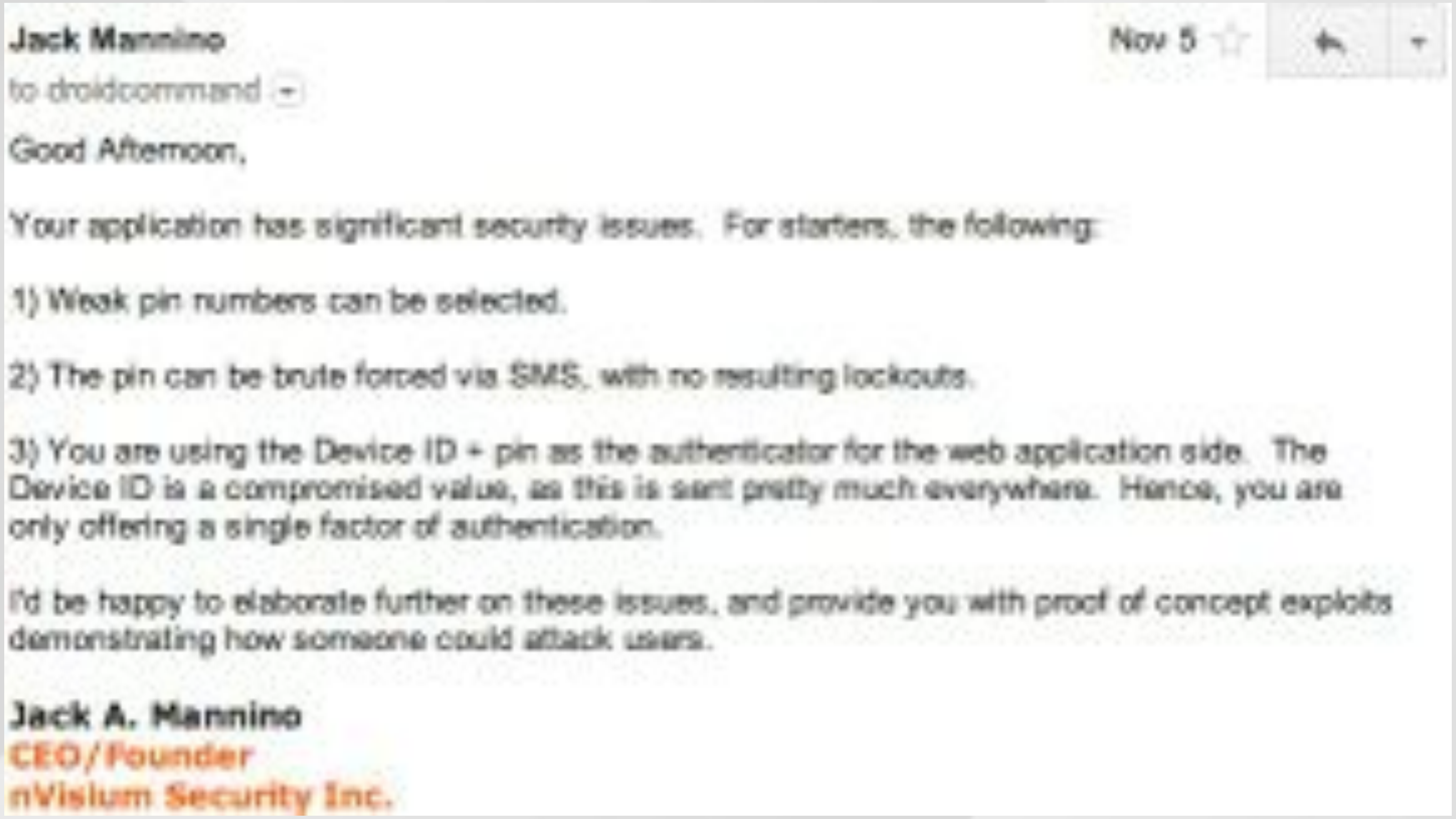
Allow Device Adn ✓
Enables Remote Lock, Data Wipe operations

Location Enabled

- Allows you to remotely control a device via SMS
- Interesting permissions
- BIND_DEVICE_ADMIN
- READ_SMS

nVisium SECURITY

Droid Commander



Jack Mannino Nov 5

to droidcommand

Good Afternoon,

Your application has significant security issues. For starters, the following:

1) Weak pin numbers can be selected.

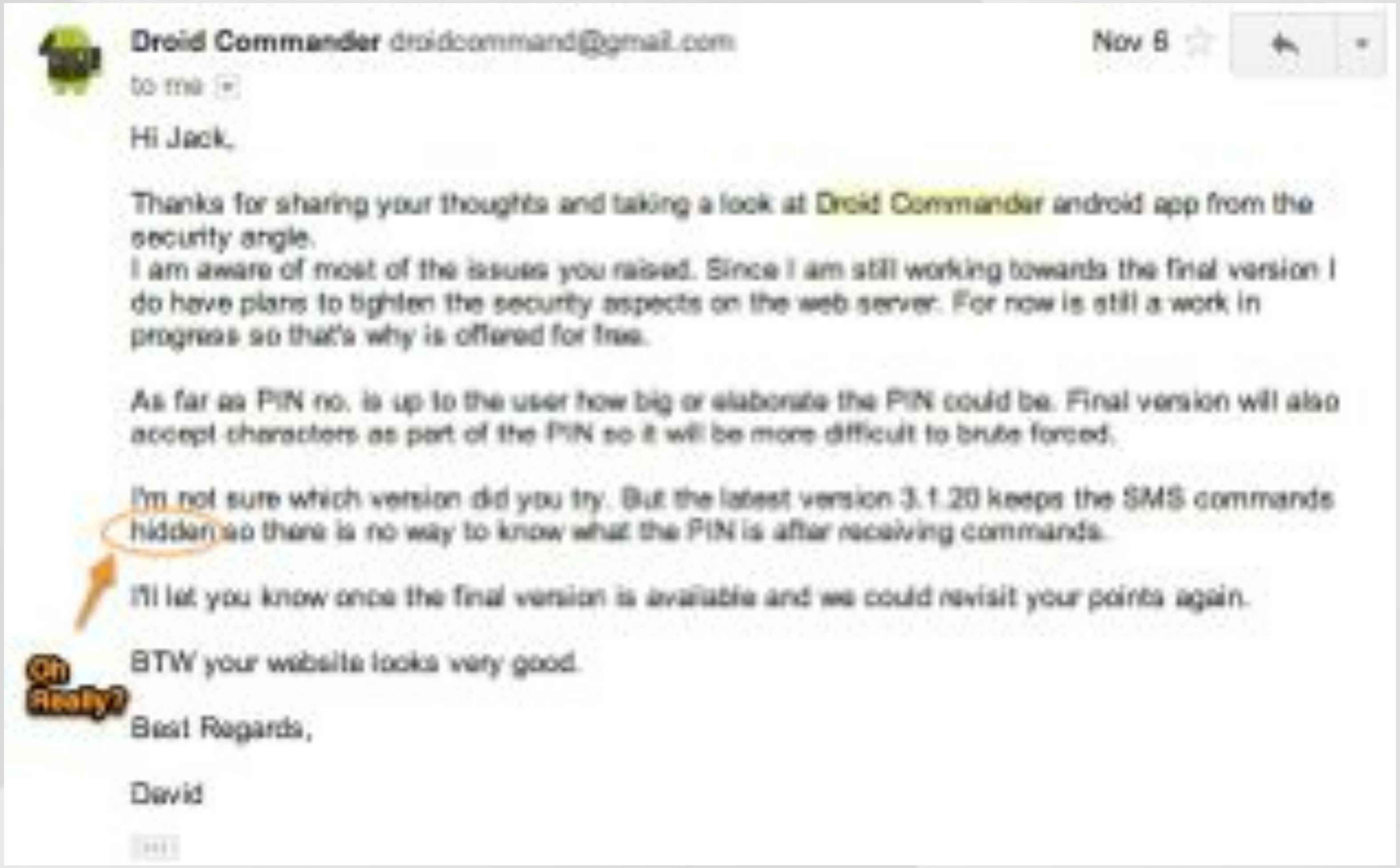
2) The pin can be brute forced via SMS, with no resulting lockouts.

3) You are using the Device ID + pin as the authenticator for the web application side. The Device ID is a compromised value, as this is sent pretty much everywhere. Hence, you are only offering a single factor of authentication.

I'd be happy to elaborate further on these issues, and provide you with proof of concept exploits demonstrating how someone could attack users.

Jack A. Mannino
CEO/Founder
nVisium Security Inc.

Droid Commander



Droid Commander droidcommand@gmail.com

to me

Nov 6

Hi Jack,

Thanks for sharing your thoughts and taking a look at Droid Commander android app from the security angle.
I am aware of most of the issues you raised. Since I am still working towards the final version I do have plans to tighten the security aspects on the web server. For now is still a work in progress so that's why is offered for free.

As far as PIN no. is up to the user how big or elaborate the PIN could be. Final version will also accept characters as part of the PIN so it will be more difficult to brute forced.

I'm not sure which version did you try. But the latest version 3.1.20 keeps the SMS commands hidden so there is no way to know what the PIN is after receiving commands.

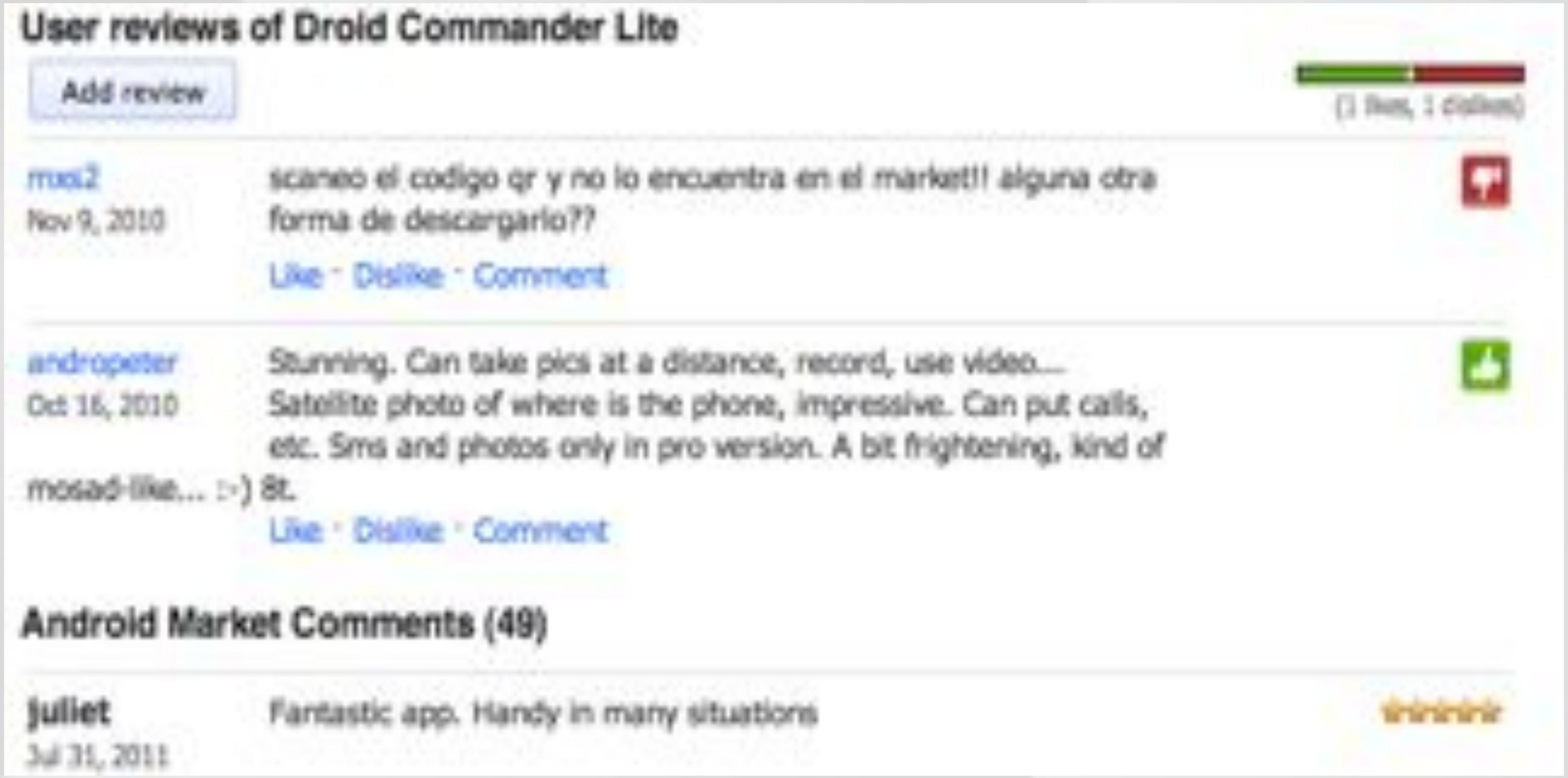
I'll let you know once the final version is available and we could revisit your points again.

BTW your website looks very good.

Best Regards,

David

Heh…"Handy"



User reviews of Droid Commander Lite

Add review

(1 likes, 1 dislikes)

mosi2
Nov 9, 2010
scaneo el codigo qr y no lo encuentra en el market!! alguna otra forma de descargarlo??
Like · Dislike · Comment

andropeter
Oct 16, 2010
Stunning. Can take pics at a distance, record, use video... Satellite photo of where is the phone, impressive. Can put calls, etc. Sms and photos only in pro version. A bit frightening, kind of mosad-like... :-) 8t.
Like · Dislike · Comment

Android Market Comments (49)

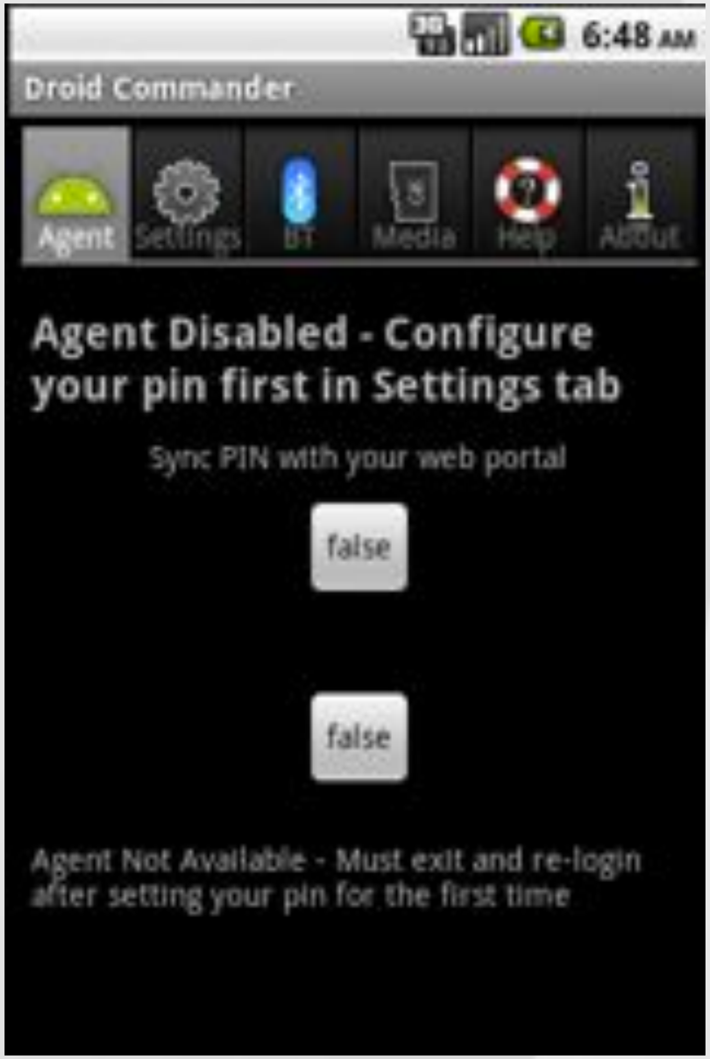
juliet
Jul 31, 2011
Fantastic app. Handy in many situations

Abuse Cases

- Remotely trigger phone calls to a number of your choice
- Eavesdropping
- Take pictures, record calls
- ***REMOTELY WIPE A DEVICE***

nVisium SECURITY

We Need a Pin



Agent Disabled - Configure your pin first in Settings tab

Sync PIN with your web portal

Agent Not Available - Must exit and re-login after setting your pin for the first time

Suggestions

- ❏ 1
- ❏ 11
- ❏ 666
- ❏ 14

How It Works

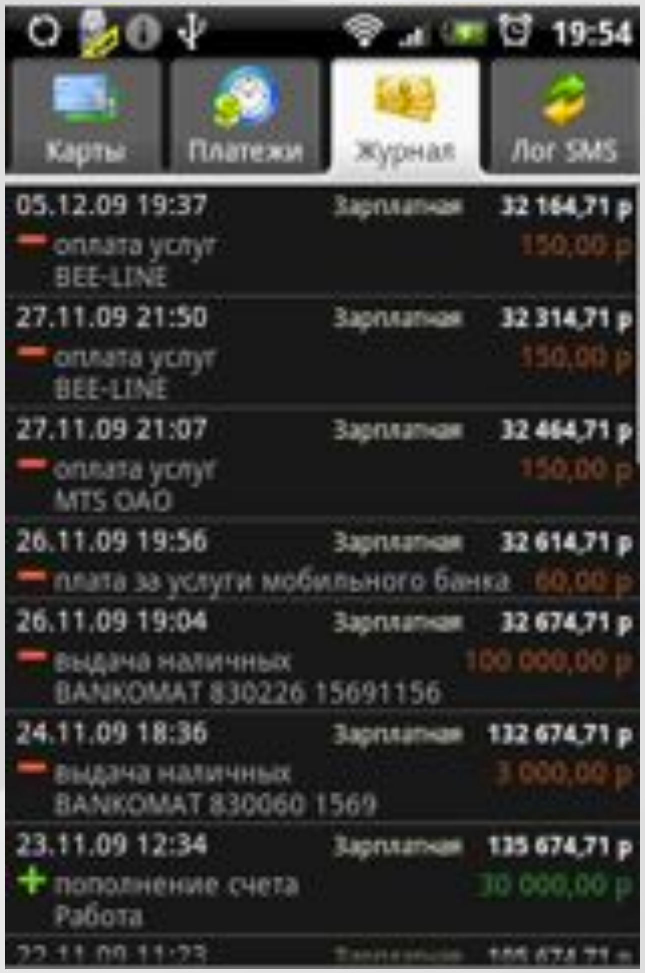
- Send an SMS message to the device
- Syntax: pin:command:data
- 666:Call:9765551234
- 666:wipe:omglulz

Lessons Learned

- ❑ Don’t use this app
- ❑ SMS is a terrible way to enable single factor authentication
- ❑ Weak authentication pin makes it worse
- ❑ Malicious apps can abuse this too

nVisium SECURITY

SMS-Banking



05.12.09 19:37 Зарплатная 32 164,71 р
− оплата услуг BEE-LINE 150,00 р

27.11.09 21:50 Зарплатная 32 314,71 р
− оплата услуг BEE-LINE 150,00 р

27.11.09 21:07 Зарплатная 32 464,71 р
− оплата услуг MTS OAO 150,00 р

26.11.09 19:56 Зарплатная 32 614,71 р
− плата за услуги мобильного банка 60,00 р

26.11.09 19:04 Зарплатная 32 674,71 р
− выдача наличных BANKOMAT 830226 15691156 100 000,00 р

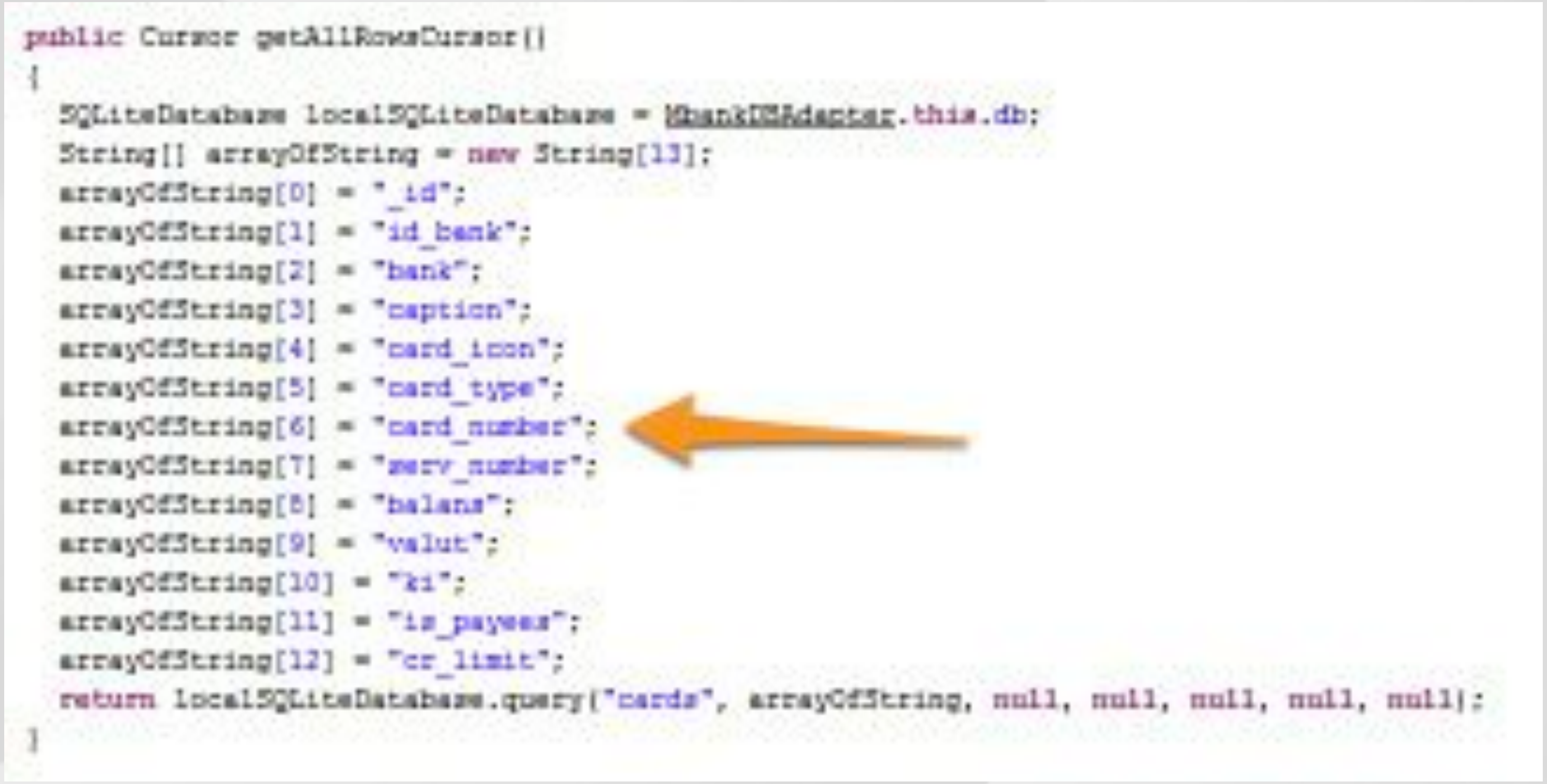
24.11.09 18:36 Зарплатная 132 674,71 р
− выдача наличных BANKOMAT 830060 1569 3 000,00 р

23.11.09 12:34 Зарплатная 135 674,71 р
+ пополнение счета Работа 30 000,00 р

22.11.09 11:23 Зарплатная 105 674,71 р

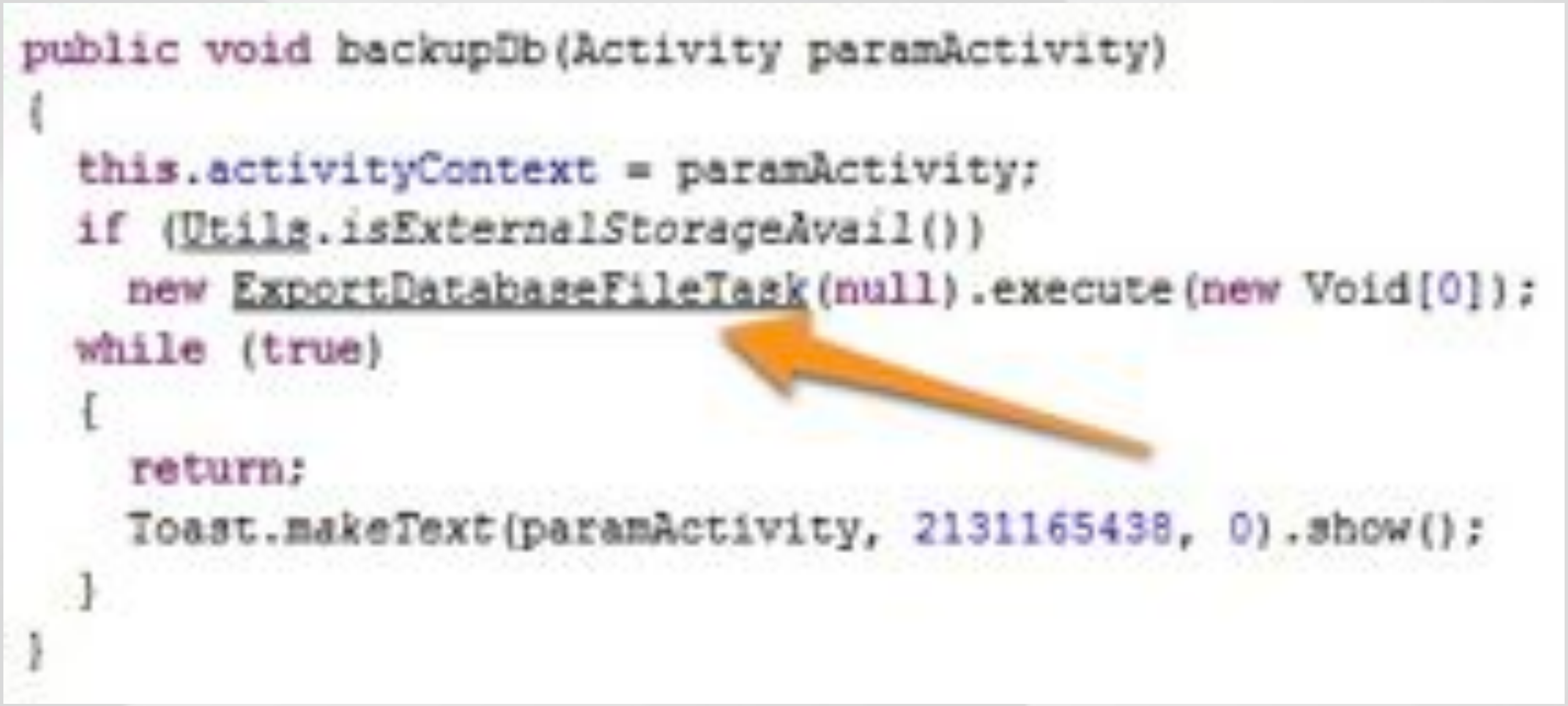
- A Russian app that aggregates multiple Russian banking services

Hide Your Wife, Hide Your Kids



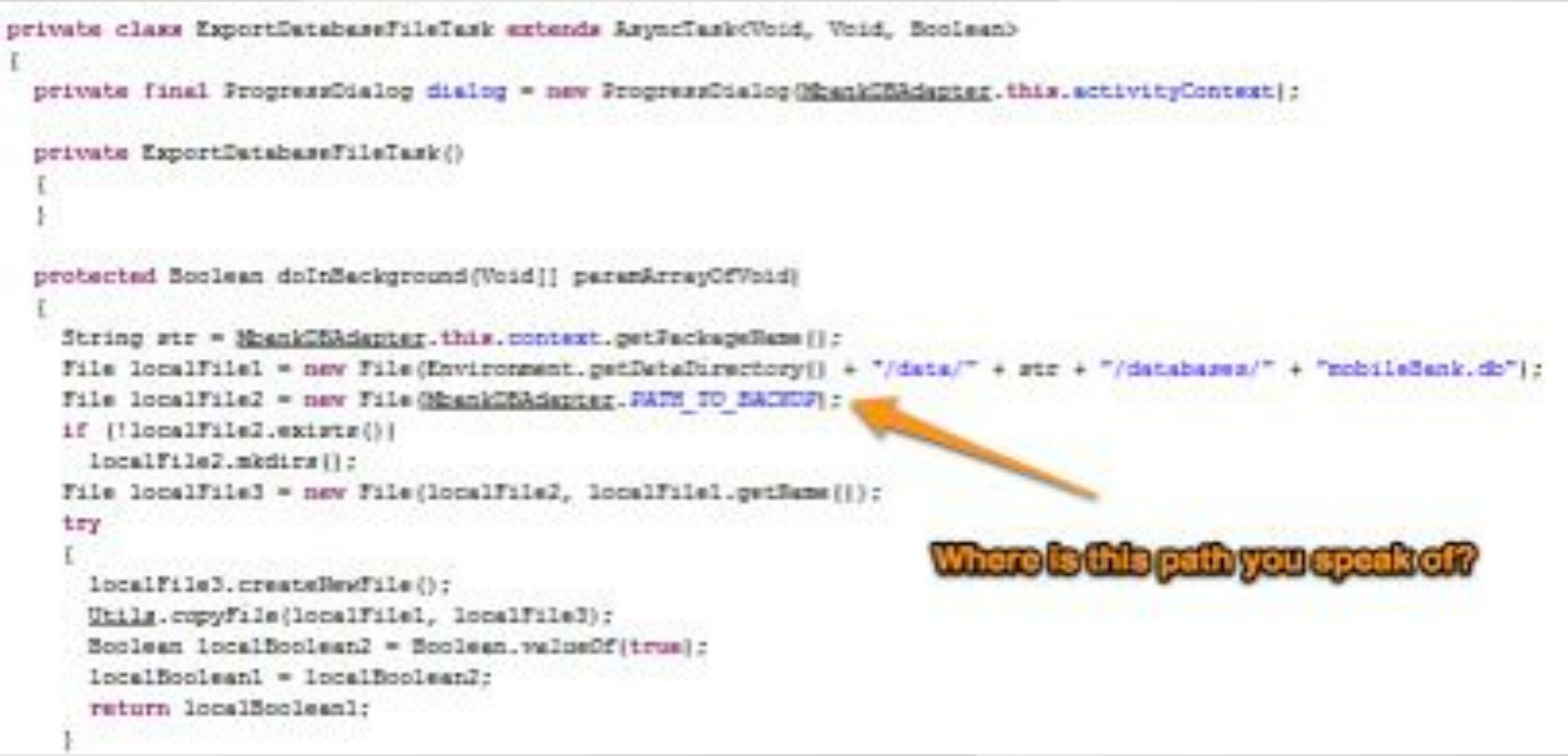
```
public Cursor getAllRowsCursor()
{
  SQLiteDatabase localSQLiteDatabase = MbankDBAdapter.this.db;
  String[] arrayOfString = new String[13];
  arrayOfString[0] = "_id";
  arrayOfString[1] = "id_bank";
  arrayOfString[2] = "bank";
  arrayOfString[3] = "caption";
  arrayOfString[4] = "card_icon";
  arrayOfString[5] = "card_type";
  arrayOfString[6] = "card_number";
  arrayOfString[7] = "serv_number";
  arrayOfString[8] = "balans";
  arrayOfString[9] = "valut";
  arrayOfString[10] = "ki";
  arrayOfString[11] = "is_payees";
  arrayOfString[12] = "cr_limit";
  return localSQLiteDatabase.query("cards", arrayOfString, null, null, null, null, null);
}
```

Can It Get Worse?



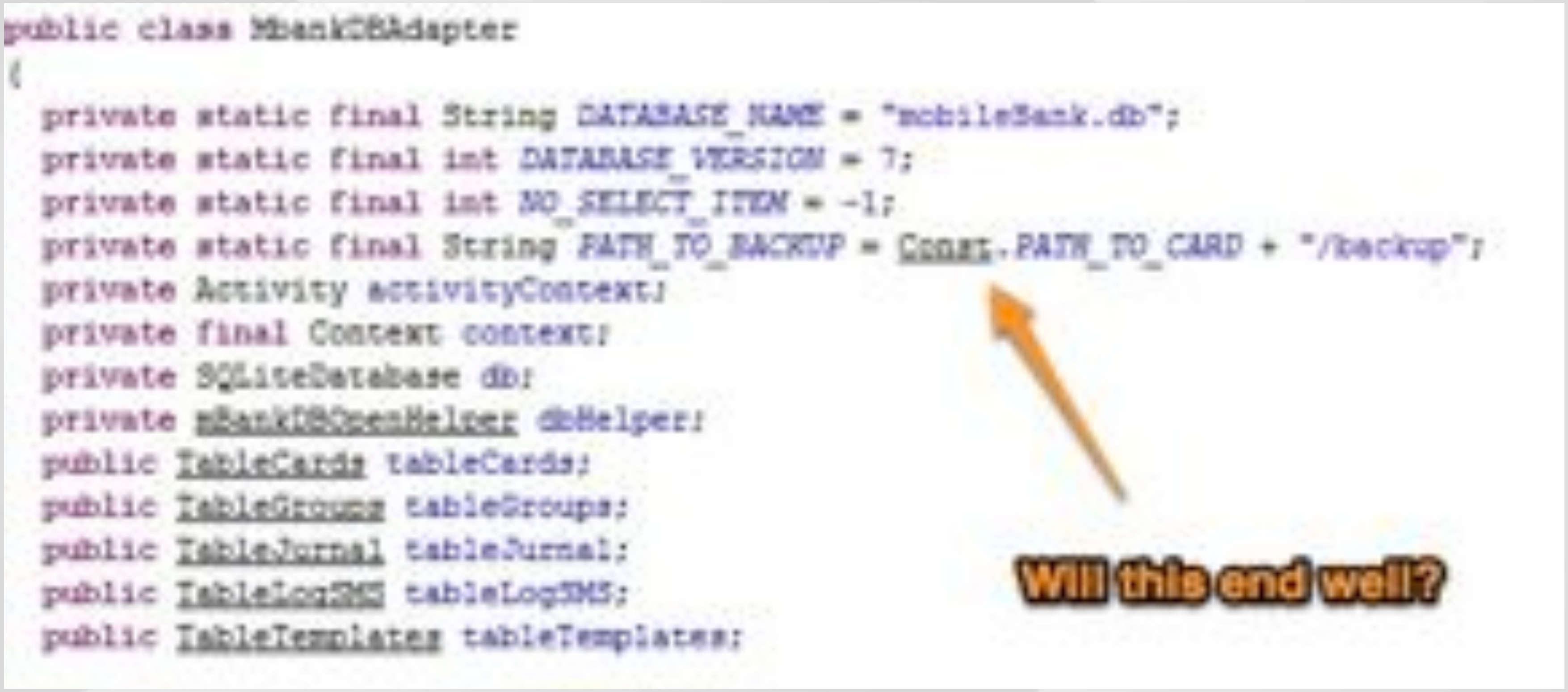
```
public void backupDb(Activity paramActivity)
{
  this.activityContext = paramActivity;
  if (Utils.isExternalStorageAvail())
    new ExportDatabaseFileTask(null).execute(new Void[0]);
  while (true)
  {
    return;
    Toast.makeText(paramActivity, 2131165438, 0).show();
  }
}
```

Ummmm



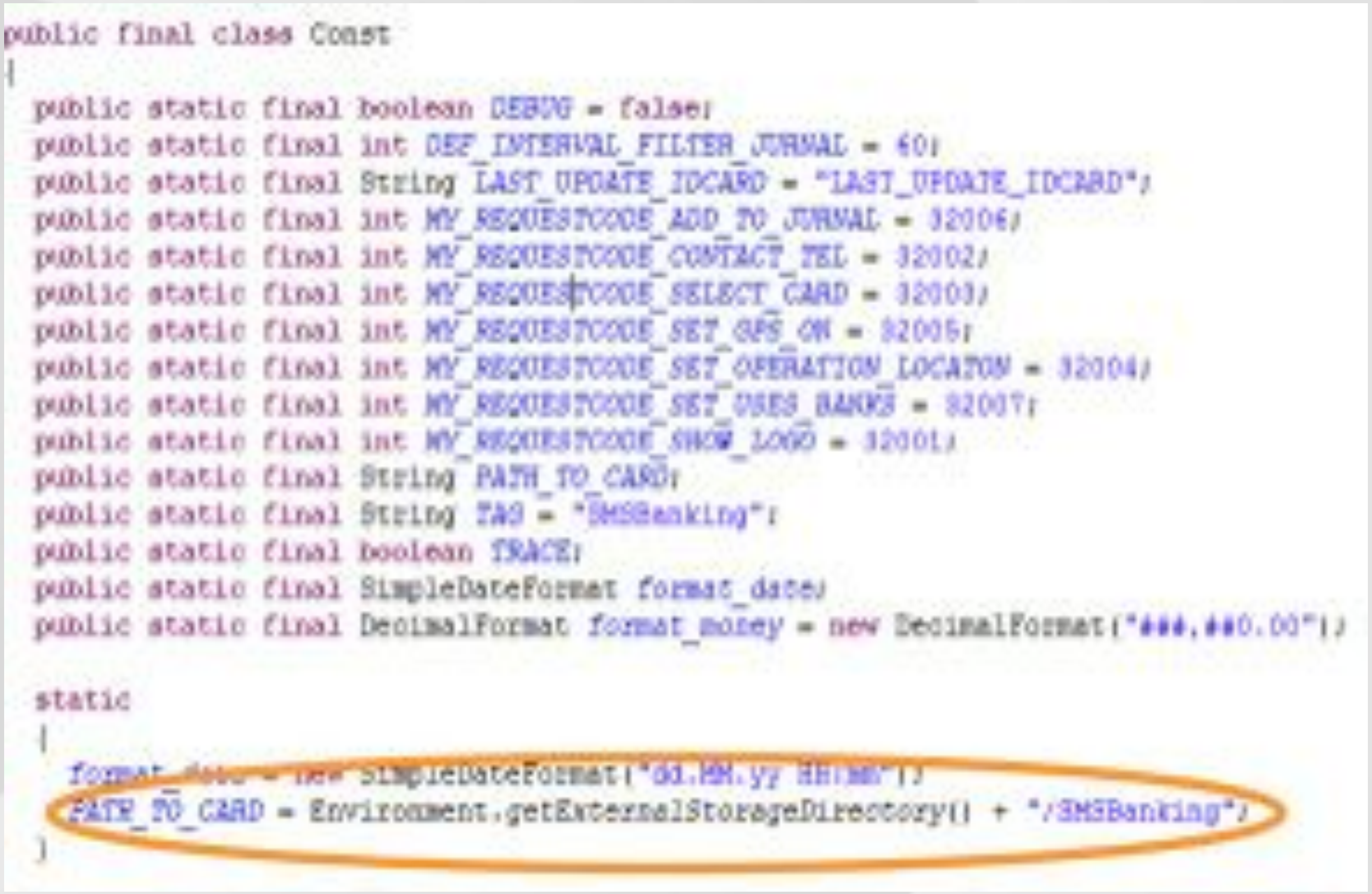
```
private class ExportDatabaseFileTask extends AsyncTask<Void, Void, Boolean>
{
  private final ProgressDialog dialog = new ProgressDialog(MbankCBAdapter.this.activityContext);

  private ExportDatabaseFileTask()
  {
  }

  protected Boolean doInBackground(Void[] paramArrayOfVoid)
  {
    String str = MbankCBAdapter.this.context.getPackageName();
    File localFile1 = new File(Environment.getDataDirectory() + "/data/" + str + "/databases/" + "mobileBank.db");
    File localFile2 = new File(MbankCBAdapter.PATH_TO_BACKUP);
    if (!localFile2.exists())
      localFile2.mkdirs();
    File localFile3 = new File(localFile2, localFile1.getName());
    try
    {
      localFile3.createNewFile();
      Utils.copyFile(localFile1, localFile3);
      Boolean localBoolean2 = Boolean.valueOf(true);
      localBoolean1 = localBoolean2;
      return localBoolean1;
    }
```

Oh Yeah…There



```
public class MbankDBAdapter
{
  private static final String DATABASE_NAME = "mobileBank.db";
  private static final int DATABASE_VERSION = 7;
  private static final int NO_SELECT_ITEM = -1;
  private static final String PATH_TO_BACKUP = Const.PATH_TO_CARD + "/backup";
  private Activity activityContext;
  private final Context context;
  private SQLiteDatabase db;
  private mBankDBOpenHelper dbHelper;
  public TableCards tableCards;
  public TableGroups tableGroups;
  public TableJornal tableJurnal;
  public TableLogSMS tableLogSMS;
  public TableTemplates tableTemplates;
```

Tip: Don't Do This



```
public final class Const
{
  public static final boolean DEBUG = false;
  public static final int DEF_INTERVAL_FILTER_JURNAL = 60;
  public static final String LAST_UPDATE_IDCARD = "LAST_UPDATE_IDCARD";
  public static final int MY_REQUESTCODE_ADD_TO_JURNAL = 32006;
  public static final int MY_REQUESTCODE_CONTACT_TEL = 32002;
  public static final int MY_REQUESTCODE_SELECT_CARD = 32003;
  public static final int MY_REQUESTCODE_SET_GPS_ON = 32005;
  public static final int MY_REQUESTCODE_SET_OPERATION_LOCATON = 32004;
  public static final int MY_REQUESTCODE_SET_USES_BANKS = 32007;
  public static final int MY_REQUESTCODE_SHOW_LOGO = 32001;
  public static final String PATH_TO_CARD;
  public static final String TAG = "SMSBanking";
  public static final boolean TRACE;
  public static final SimpleDateFormat format_date;
  public static final DecimalFormat format_money = new DecimalFormat("###,##0.00");

  static
  {
    format_date = new SimpleDateFormat("dd.MM.yy HH:mm");
    PATH_TO_CARD = Environment.getExternalStorageDirectory() + "/SMSBanking";
  }
```

Lessons Learned

- Storing sensitive stuff in the clear is bad
- Backing it up to external storage is even worse
- Other apps can access this
- Lose phone, game over

Device ID Authentication

- Requires READ_PHONE_STATE permission
- Tons of apps have this permission
- Ad networks track this stuff too

nVisium SECURITY

Device ID Authentication

- Some apps that have this permission:

- 3D Sexy Girls
- Adult Sexy Wallpapers
- Asian Sexy Girl
- Hot Sex Tips
- Male and Female Sex Movies

Device ID Authentication

- ❑Let’s assume that you trust the makers of those apps
- ❑Do you trust their ability to not get owned?

Device ID Authentication

- Demo of a weak implementation
- Pulled from the OWASP GoatDroid Project (I wrote it)
- GoatDroid is a playground for learning about Android security

Authentication Tips

- ❑ Never use device ID, IMEI, IMSI, etc as sole authenticators
- ❑ Out of band measures fail on a single device, for the most part
- ❑ Contextual information (ie- location) can help, but can be spoofed

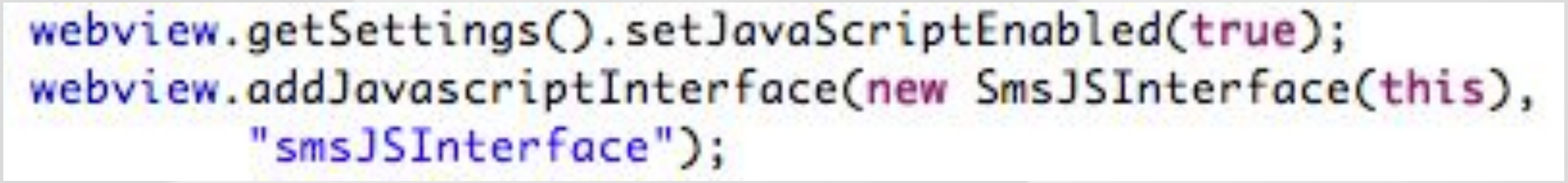
nVisium SECURITY

XSS On Steroids

- XSS…you know, it makes pretty alert boxes
- Arbitrary script execution is fun
- Hybrid native/web apps

XSS On Steroids

- Android allows Java code to be called via WebViews using JavaScript
- Powerful, but dangerous



```
webview.getSettings().setJavaScriptEnabled(true);
webview.addJavascriptInterface(new SmsJSInterface(this),
        "smsJSInterface");
```

XSS On Steroids

- Another demo

Do’s and Don’ts

- Do: contextually output encode untrusted data within a WebView
- Don’t: tie sensitive actions to untrusted data within JavaScript interfaces

Activity Lifecycle Badness

- Other apps can start your app's activities
- Permitted when you export an Activity
- Also permitted if your Activity uses Intent Filters

Activity Lifecycle Badness

- Scenario: your Activity executes something within the onCreate() method
- It also receives data within a Bundle
- Demo time

nVisium SECURITY

Activity Lifecycle Badness

- Another thing you shouldn't do: depend on onDestroy() to execute
- NOT GUARANTED TO BE CALLED
- Often used to "clean up" sensitive data
- Solution: don't do this

nVisium SECURITY

Conclusion

- Nothing I presented today requires a prohibitively high skill level
- These types of issues are extremely common within real-world apps
- Common sense and judgement are equally as important as technical stuff

nVisium SECURITY

Thanks For Listening!

- Follow my daily rants @ http://twitter.com/jack_mannino

- Check out the OWASP Mobile Security Project https://www.owasp.org/index.php/OWASP_Mobile_Security_Project

nVisium SECURITY