

Mobile Testing: Getting The Most Out of Your Device

Stephen Jensen (stephen.jensen@usc-bt.com)

Senior Security Consultant

BT Global Services

Agenda:

- Introduction
- iPhone features, functionality and fun
- Android features, functionality, fun and reversing

Introduction:

- What we can do depends on the devices functionality.
 - Wifi Supported?
 - HTTP Proxy Configuration Supported?
- Why do we care?
 - Because we can!! It's legal now, and we have a right to know what may be touching our personal/sensitive information. Recent research has revealed that more and more mobile applications are malicious in nature. From performing unauthorized purchases to sending personal information from the device to remote servers.

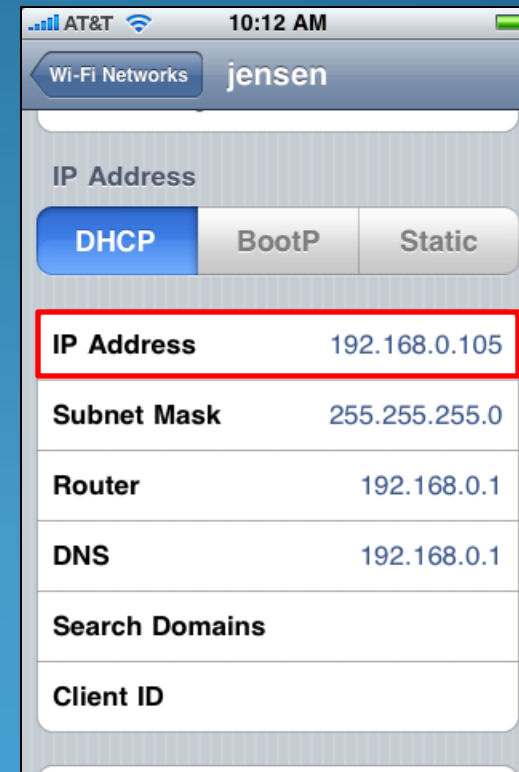
iPhone Jailbreaking:

- Spirit jailbreak utility
- Jailbreakme.com
 - still works on 3G, 3GS & non-updated 4.0
- Cydia (What's this?)
 - Cydia is the “Anti-Apple” store.
 - Where you can install unapproved 3rd party apps on your jailbroken iPhone.
 - How-To's for SSH setup, etc.



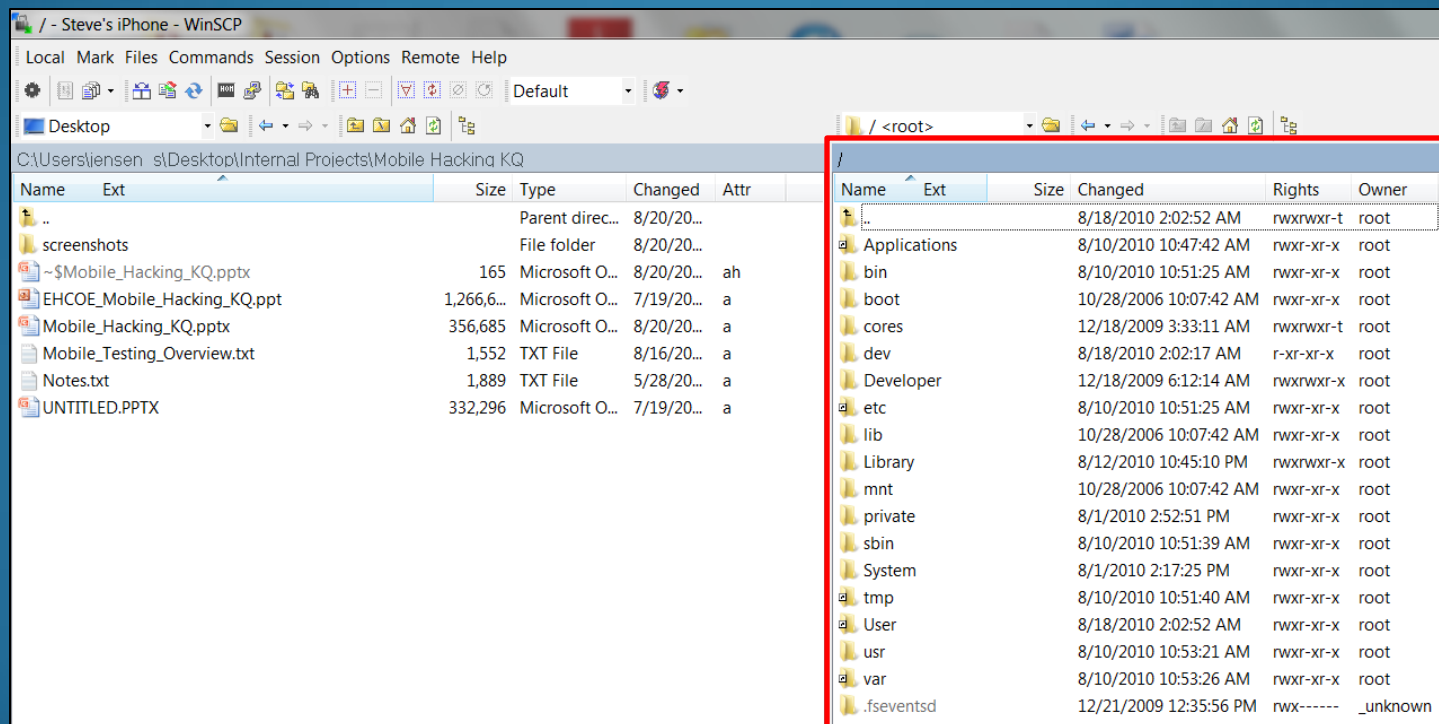
iPhone File System:

- Once SSH is installed on the device the rest is easy.
- Connect iPhone to wifi network.
- Connect laptop to wifi network.
- Get IP Address of iPhone via
 - Settings > Wifi > Details



iPhone File System:

- Can connect via Putty to the device for cmd line viewing.
- Can connect via WinSCP to the device for GUI viewing.
- Default root password on iPhone is “alpine”.

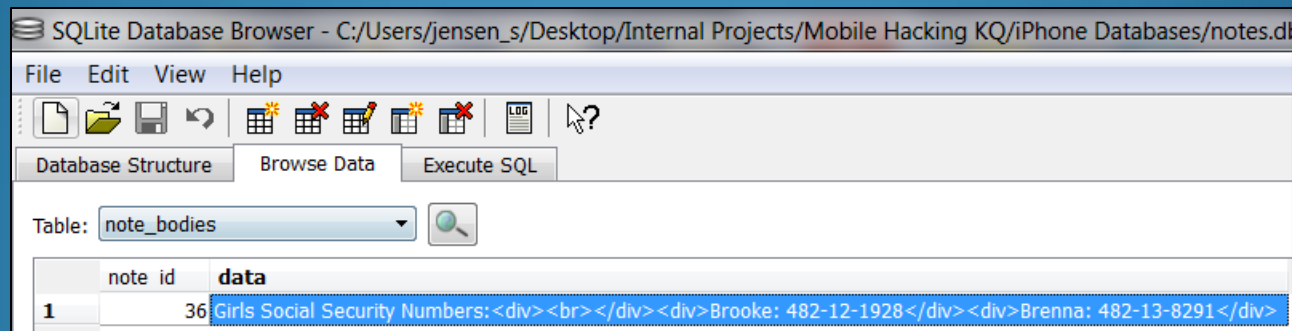


The screenshot displays the WinSCP interface connected to an iPhone. The left pane shows the local desktop with files like screenshots, Mobile_Hacking_KQ.pptx, and Mobile_Testing_Overview.txt. The right pane shows the iPhone root directory with folders like Applications, bin, boot, cores, dev, etc, lib, Library, mnt, private, sbin, System, tmp, User, usr, var, and .fsevents.

| Name | Ext | Size | Changed | Rights | Owner |
|--------------|-----|------|------------------------|-----------|----------|
| .. | | | 8/18/2010 2:02:52 AM | rw-rw-r-- | root |
| Applications | | | 8/10/2010 10:47:42 AM | rw-r--r-- | root |
| bin | | | 8/10/2010 10:51:25 AM | rw-r--r-- | root |
| boot | | | 10/28/2006 10:07:42 AM | rw-r--r-- | root |
| cores | | | 12/18/2009 3:33:11 AM | rw-rw-r-- | root |
| dev | | | 8/18/2010 2:02:17 AM | r-xr-xr-x | root |
| Developer | | | 12/18/2009 6:12:14 AM | rw-rw-r-- | root |
| etc | | | 8/10/2010 10:51:25 AM | rw-r--r-- | root |
| lib | | | 10/28/2006 10:07:42 AM | rw-r--r-- | root |
| Library | | | 8/12/2010 10:45:10 PM | rw-rw-r-- | root |
| mnt | | | 10/28/2006 10:07:42 AM | rw-r--r-- | root |
| private | | | 8/1/2010 2:52:51 PM | rw-r--r-- | root |
| sbin | | | 8/10/2010 10:51:39 AM | rw-r--r-- | root |
| System | | | 8/1/2010 2:17:25 PM | rw-r--r-- | root |
| tmp | | | 8/10/2010 10:51:40 AM | rw-r--r-- | root |
| User | | | 8/18/2010 2:02:52 AM | rw-r--r-- | root |
| usr | | | 8/10/2010 10:53:21 AM | rw-r--r-- | root |
| var | | | 8/10/2010 10:53:26 AM | rw-r--r-- | root |
| .fsevents | | | 12/21/2009 12:35:56 PM | rw-x----- | _unknown |

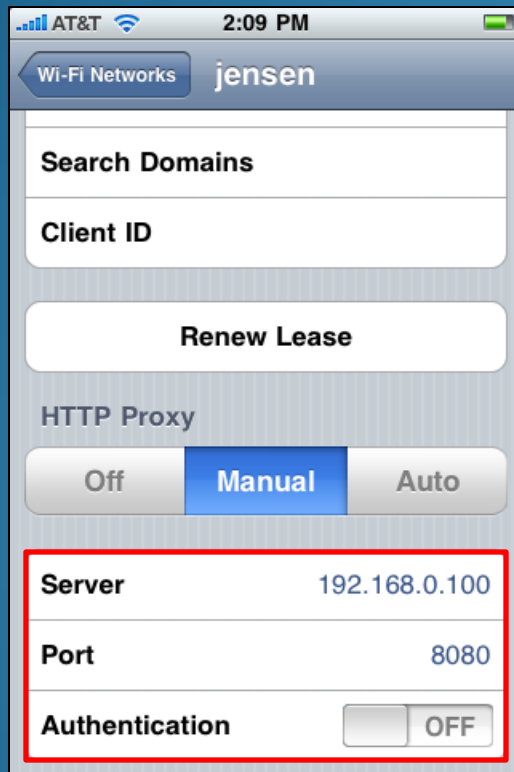
iPhone File System:

- 3rd party applications are located in /User/Applications/
- Data is stored in various databases (.sqlite, .sqlite3 and .db).
 - Below is a display of the “Notes” database via the SQLite Database Browser tool available at <http://sqlitebrowser.sourceforge.net/>



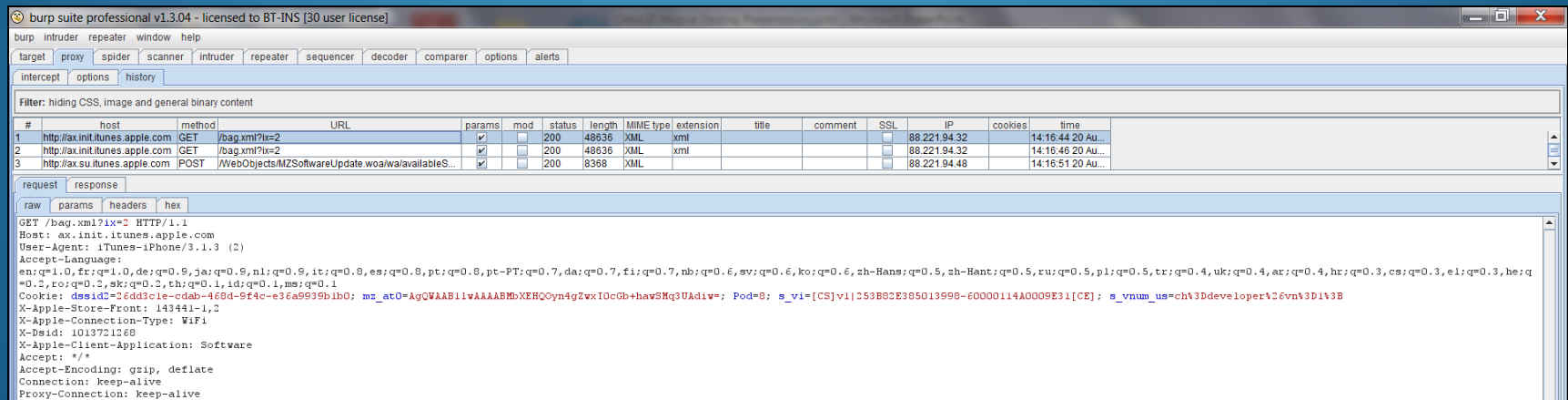
iPhone Traffic:

- iPhone has ability to proxy traffic via wifi and VPN.
- Simply configure HTTP Proxy settings (IP & Port).



iPhone Traffic:

- Start up a proxy and configure it to listen on a designated port.
- Note: If using burp you need uncheck “loopback” and check “support invisible” within the proxy options.



iPhone Application Analysis:

- We can extract the applications off of the device for further analysis.
 - Mach-O file format (<http://en.wikipedia.org/wiki/Mach-O>)
 - ARM Architecture (http://en.wikipedia.org/wiki/ARM_architecture)
 - Written in Objective-C language (<http://en.wikipedia.org/wiki/Objective-C>)
 - Beneficial to have the iPhone SDK (<http://developer.apple.com/iphone/>)
 - Additional information on development and testing is available via the iPhone SDK link above.
 - IDA Pro 5.2 (and later) support analysis of (ARM) iPhone assemblies out of the box.

iPhone Additional Links:

- <http://blog.zynamics.com/2010/04/27/objective-c-reversing-i/>
- <http://networkpx.blogspot.com/2010/01/two-ida-pro-5x-scripts-for-iphoneos.html>
- <http://security.org.my/index.php?/plugin/tag/otool>
- <http://dvlabs.tippingpoint.com/blog/2009/03/06/reverse-engineering-iphone-appstore-binaries>
- <http://www.iphonedownloadblog.com/>

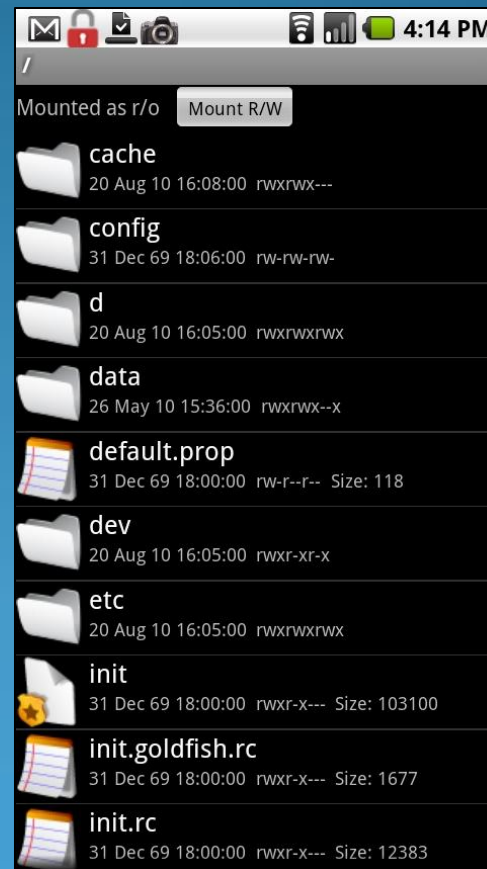
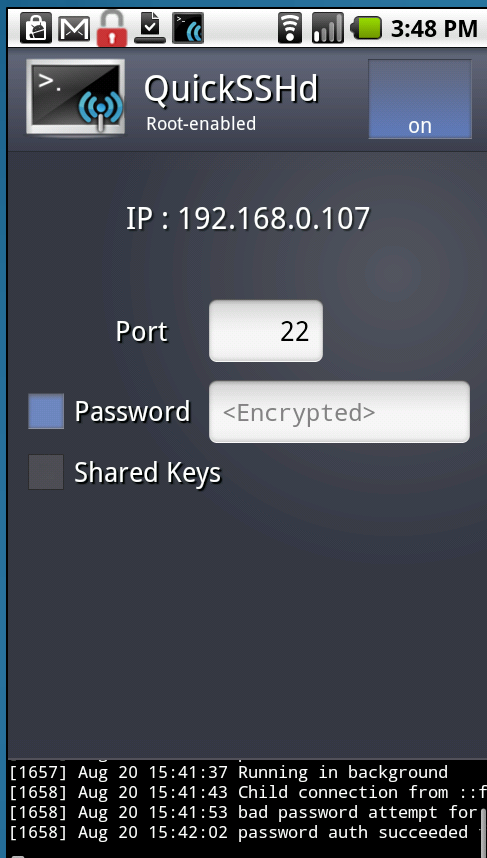
Questions?

Android Rooting:

- Easy Root (Pulled from Google Market within 48 hrs of release).
- Tons of info available on the internet (Nothing is the same!)
- May require rolling back to a previous version.
 - Had to roll back to 2.0.1 from 2.1 (took 20 times before successful!)
- Once rooted, we gain access to “root” only areas of OS.

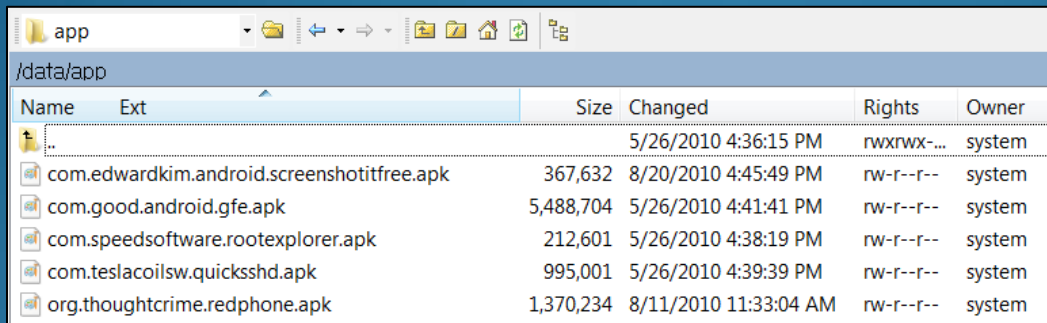
Helpful Android Apps:

- QuickSSHd – allows us to SSH into the phone and access the file system.
- Root Explorer – allows us to view the file system directly from the phone.



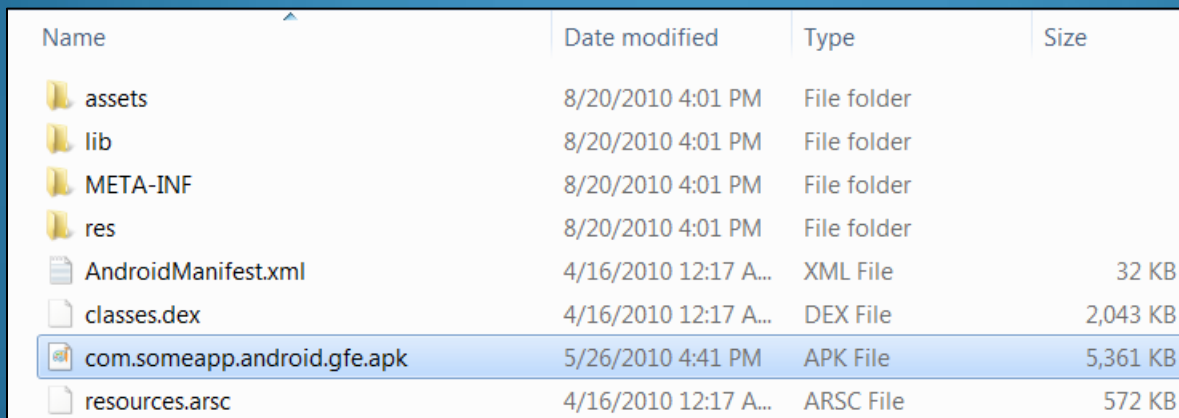
Unpackaging Apps:

- Apps have a .apk extension.
- 3rd party apps are located in /data/app/



| Name | Ext | Size | Changed | Rights | Owner |
|--|-----|-----------|-----------------------|------------|--------|
| .. | | | 5/26/2010 4:36:15 PM | rw-rwx-... | system |
| com.edwardkim.android.screenshotitfree.apk | | 367,632 | 8/20/2010 4:45:49 PM | rw-r--r-- | system |
| com.good.android.gfe.apk | | 5,488,704 | 5/26/2010 4:41:41 PM | rw-r--r-- | system |
| com.speedsoftware.rootexplorer.apk | | 212,601 | 5/26/2010 4:38:19 PM | rw-r--r-- | system |
| com.teslacoilsw.quicksshd.apk | | 995,001 | 5/26/2010 4:39:39 PM | rw-r--r-- | system |
| org.thoughtcrime.redphone.apk | | 1,370,234 | 8/11/2010 11:33:04 AM | rw-r--r-- | system |

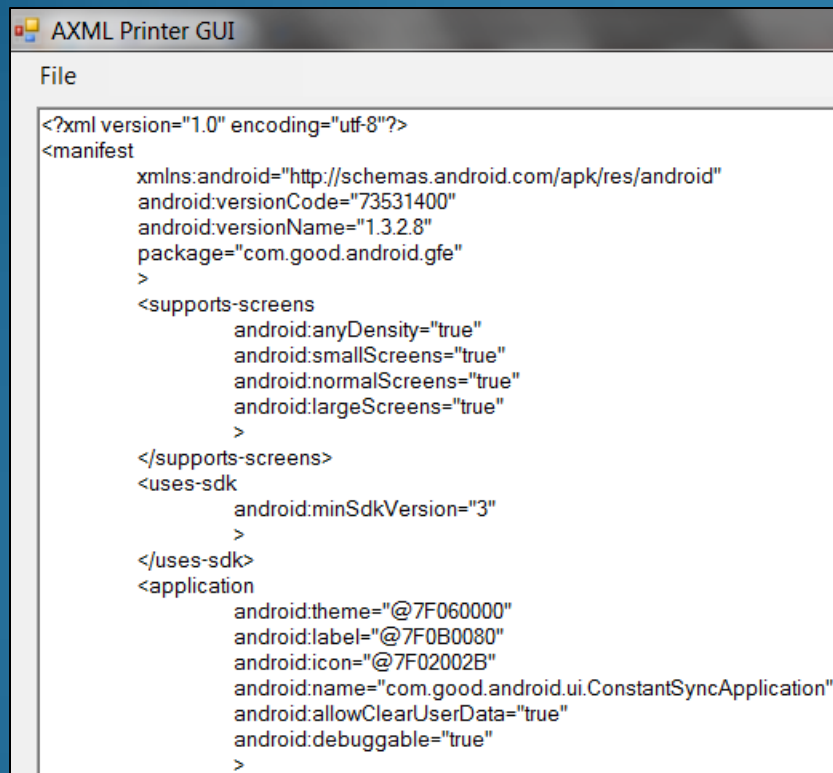
- Apps are simple packages that can be unpackaged using WinZip or 7zip.



| Name | Date modified | Type | Size |
|-----------------------------|----------------------|-------------|----------|
| assets | 8/20/2010 4:01 PM | File folder | |
| lib | 8/20/2010 4:01 PM | File folder | |
| META-INF | 8/20/2010 4:01 PM | File folder | |
| res | 8/20/2010 4:01 PM | File folder | |
| AndroidManifest.xml | 4/16/2010 12:17 A... | XML File | 32 KB |
| classes.dex | 4/16/2010 12:17 A... | DEX File | 2,043 KB |
| com.someapp.android.gfe.apk | 5/26/2010 4:41 PM | APK File | 5,361 KB |
| resources.arsc | 4/16/2010 12:17 A... | ARSC File | 572 KB |

Analyzing the Apps:

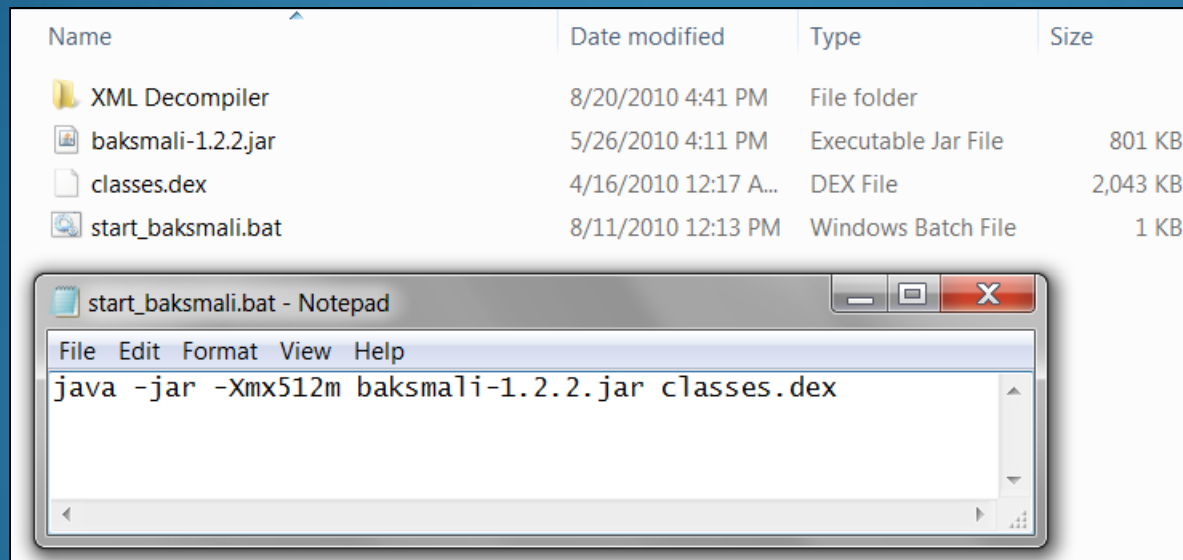
- We can use [AXMLPrinter2](#) to decode encoded XML files, such as the AndroidManifest.xml and other XML resource file.



```
AXML Printer GUI
File
<?xml version="1.0" encoding="utf-8"?>
<manifest
  xmlns:android="http://schemas.android.com/apk/res/android"
  android:versionCode="73531400"
  android:versionName="1.3.2.8"
  package="com.good.android.gfe"
  >
  <supports-screens
    android:anyDensity="true"
    android:smallScreens="true"
    android:normalScreens="true"
    android:largeScreens="true"
    >
  </supports-screens>
  <uses-sdk
    android:minSdkVersion="3"
    >
  </uses-sdk>
  <application
    android:theme="@7F060000"
    android:label="@7F0B0080"
    android:icon="@7F02002B"
    android:name="com.good.android.ui.ConstantSyncApplication"
    android:allowClearUserData="true"
    android:debuggable="true"
    >
```

















Reversing the Apps:

- ARM Architecture
- Dalvik VM, apps are compact dalvik executables (.dex).
- We can use baksmali (<http://code.google.com/p/smali/>) to decompile the classes.dex file back to a more readable files.



Reversing the Apps:

- After running baksmali on the classes.dex file we are presented with an “out” directory that contains our decompiled .smali files.

| Name | Date modified | Type | Size |
|---|-------------------|------------|-------|
|  AddressEntry\$1.smali | 8/20/2010 4:50 PM | SMALI File | 2 KB |
|  AddressEntry.smali | 8/20/2010 4:50 PM | SMALI File | 6 KB |
|  AddressUtil.smali | 8/20/2010 4:50 PM | SMALI File | 6 KB |
|  AlertDialog\$1.smali | 8/20/2010 4:50 PM | SMALI File | 1 KB |
|  AlertDialog\$DialogButtonListener.smali | 8/20/2010 4:50 PM | SMALI File | 1 KB |
|  AlertDialog\$OnClickListenerImp.smali | 8/20/2010 4:50 PM | SMALI File | 5 KB |
|  AlertDialog.smali | 8/20/2010 4:50 PM | SMALI File | 14 KB |
|  APNUtils.smali | 8/20/2010 4:50 PM | SMALI File | 22 KB |
|  AtomicSoftReference.smali | 8/20/2010 4:50 PM | SMALI File | 7 KB |
|  Base16Util.smali | 8/20/2010 4:50 PM | SMALI File | 3 KB |
|  Base64Util.smali | 8/20/2010 4:50 PM | SMALI File | 14 KB |
|  CalendarUtil.smali | 8/20/2010 4:50 PM | SMALI File | 3 KB |
|  CountedOutputStream.smali | 8/20/2010 4:50 PM | SMALI File | 5 KB |
|  CRC16CCITT.smali | 8/20/2010 4:50 PM | SMALI File | 3 KB |
|  DataIO.smali | 8/20/2010 4:50 PM | SMALI File | 5 KB |
|  DataPrepUtil.smali | 8/20/2010 4:50 PM | SMALI File | 5 KB |

Reversing the Apps:

- Reviewing the .smali files can reveal interesting information.

```
Line 10: .field private static final SQL_DELETE_BY_EMAILIDLIST1:Ljava/lang/String; = "delete from Email where email_id in (?)"
Line 12: .field private static final SQL_DELETE_BY_EMAILIDLIST2:Ljava/lang/String; = ")"
Line 14: .field private static final SQL_GET_OLDEST_EMAIL:Ljava/lang/String; = "select email_id from Email where _id in (select _id from Email where folder_id = ? order by receive_date desc limit 1)"
Line 16: .field private static final SQL_GET_PENDING_OUTBOX_EMAIL:Ljava/lang/String; = "select _id,email_id,mailbox_id,folder_id,from_person,to_list,cc_list,bcc_list,to_truncated,cc_truncated from Email where mailbox_id = ? and folder_id = ? and read=0"
Line 18: .field private static final SQL_GET_UPDATED_MEETING_REQUESTS:Ljava/lang/String; = "select _id,email_id,mailbox_id,folder_id,from_person,to_list,cc_list,bcc_list,to_truncated,cc_truncated from Email where mailbox_id = ? and folder_id = ? and read=0"
Line 20: .field private static final SQL_QUERY_GET_BY_EMAILID:Ljava/lang/String; = "select _id,email_id,mailbox_id,folder_id,from_person,to_list,cc_list,bcc_list,to_truncated,cc_truncated from Email where mailbox_id = ? and folder_id = ?"
Line 22: .field private static final SQL_QUERY_GET_EMAIL_ADDRESS_FIELDS:Ljava/lang/String; = "select distinct to_list, cc_list, from_person from Email order by receive_date desc limit 1"
Line 24: .field private static final SQL_QUERY_GET_EMAIL_BY_EMAILID:Ljava/lang/String; = "select * from Email where mailbox_id = ? and folder_id = ? and email_id = ?"
Line 26: .field private static final SQL_QUERY_GET_EMAIL_COUNT_ALL:Ljava/lang/String; = "select count(*) from Email where mailbox_id = ? and folder_id = ?"
Line 28: .field private static final SQL_QUERY_GET_EMAIL_COUNT_UNREAD:Ljava/lang/String; = "select count(*) from Email where mailbox_id = ? and folder_id = ? and read=0"
Line 30: .field private static final SQL_QUERY_GET_EMAIL_HEADERS:Ljava/lang/String; = "select _id, from_person, subject, email_id, read, receive_date, attachment_count, importance, reply_count from Email where mailbox_id = ? and folder_id = ?"
Line 915: const-string v15, " where folder_id = "
```

```
.class public Lcom/someapp/android/security/PasswordKeyGen;
.super Ljava/lang/Object;
.source "PasswordKeyGen.java"

# static fields
.field private static final salt:Ljava/lang/String; = "AQ2#@$12er$9*"

# instance fields
.field private final passPhrase:Ljava/lang/String;

.method public constructor <init>(Ljava/lang/String;)V
    registers 16
    .parameter "passPhrase"
    .annotation system Ldalvik/annotation/Throws;
        value = {
            Ljava/security/spec/InvalidKeySpecException;,
        }
    .end annotation

    .local v1, iterationCount:I
    iput-object p1, p0, Lcom/someapp/android/security/PasswordKeyGen;->passPhrase:Ljava/lang/String;

    .line 44
    new-instance v3, Ljavax/crypto/spec/PBEKeySpec;

    invoke-virtual {p1}, Ljava/lang/String;->toCharArray()[C

    move-result-object v10

    const-string v11, "AQ2#@$12er$9*"

    invoke-virtual {v11, Ljava/lang/String;->getBytes()[B

    move-result-object v11

    invoke-direct {v3, v10, v11, v1}, Ljavax/crypto/spec/PBEKeySpec;-><init>([C[B)V

    .line 47
    .local v8, tb:J
    const-string v10, "PBKWITHSHAAND128BITAES-CBC-BC"

    invoke-static {v10, Ljavax/crypto/SecretKeyFactory;->getInstance(Ljava/lang/String;)Ljavax/crypto/SecretKeyFactory;
```

Helpful Links:

- Video Tutorial of Android Reversing (<http://www.accessroot.com/>)
- Android SDK (<http://developer.android.com/sdk/>)

Questions?

- For future updates you can follow me on twitter (@hack_a_kitten)