How to Analyze an Android Bot

Kevin McNamee
Nokia Threat Intelligence Lab
@KevMcNamee
Agenda

- Introduction
- Tools
- The Lab
- Demo
- Q&A
Why Analyze Android Malware

- We monitor mobile traffic for malware infections
  - Malware C&C
  - Exploits
  - DDOS
  - Hacking

- Need accurate detection rules
Developing Malware Detection Rules

- **MALWARE SAMPLES**
  - VIRUS VAULT
    - 120K+ ANALYZED PER DAY
    - 30M+ Active samples

- **SANDBOX**
  - MALWARE TRAFFIC LIBRARY

- **RULES DEVELOPMENT**
  - RULES LIBRARY
  - TRAFFIC POLICY
  - ZERO DAY BEHAVIORAL RULES

- **QUALITY TESTING**
  - DEPLOYMENT-SPECIFIC RULE SETS

- **RULE ACTIVATION**
  - RULES REPOSITORY
  - FEEDBACK FROM FIELD TESTS

- **FIELD TESTING IN LIVE NETWORKS**

**NOKIA**

**RSA Conference 2016**
Android Malware Analysis

- So, we built our own Android malware analysis lab
- You will learn
  - What tools are required
  - How to set up the network environment
  - How they are used
- Analysis allows you to:
  - Know what the malware does
  - Understand its threat level
  - Detect and remediate the infection
Android App

- Contained in APK file (zip format)
- Main components include:
  - Manifest
  - Dalvik byte code (classes.dex file)
  - Resources
  - Assets
  - Libraries
Basic Analysis Process

- Explore what’s in APK file
- Decompile DEX and review source
- Run app on phone or AVD & capture network activity
If you are going to analyze apps you have to know a bit about how they are made...

Also provides many of the tools needed for analysis...

- ADB (debugging)
- AVD (simulated phones)
Tools – Apktool

- Tool for reverse engineering Android packages (apk files)
- Extract components
  - Manifest, Resources, Libraries, Assets, Byte-code (Smali)
- Can edit and modify components
- Rebuild modified app
Tools – ADB

- Android Debug Bridge
- Comes with Android Studio
- Provides:
  - Shell access
  - Access to file system
  - Scripted remote control
  - Application Install/Uninstall
Tools – dex2jar

- Converts Dalvik byte code to Java byte code
- First step in de-compiling an Android app.
Tools – Java Decompiler

- Converts Java byte code to source code.
- Doesn’t always work 😞
- Options include:
  - JD-GUI
  - Luyten (Procyon)
Tools – WireShark

- Capture and network traffic
- Analyze network traffic
- Help develop detection rules
The Lab

Control Server

ADB/USB

Malware

wifi

Internet

Packet Capture
Some malware may only function on a real mobile network

You can build your own mobile network.
We have automated the analysis process using:

- Web based user interface
- Real phones and AVDs
- Malware database
- APKtool/Dex2Jar/GD-GUI
- ADB scripting
- Monkey Script
- WireShark
- Interface to Virus Total
<table>
<thead>
<tr>
<th>Vendor</th>
<th>Classification</th>
<th>Vendor</th>
<th>Classification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Comodo</td>
<td>UnclassifiedMalware</td>
<td>Sophos</td>
<td>Andr/Notcom-A</td>
</tr>
<tr>
<td>DrWeb</td>
<td>Android.Proxy.1.origin</td>
<td>VIPRE</td>
<td>Trojan.AndroidOS.Generic.A</td>
</tr>
<tr>
<td>TrendMicro</td>
<td>TROJ_GEN.F47V0319</td>
<td>AntiVir</td>
<td>Android/Proxy.A</td>
</tr>
<tr>
<td>HouseCall</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Kingsoft</td>
<td>Android.Troj.at_Nisev.a.(kcloud)</td>
<td>NANO-Antivirus</td>
<td>Trojan.Nisev.bkqvoe</td>
</tr>
<tr>
<td>F-Prot</td>
<td>AndroidOS/NotCom.A</td>
<td>GData</td>
<td>Android.Trojan.NioServ.A</td>
</tr>
<tr>
<td>ESET-NOD32</td>
<td>a variant of Android/NoCom.A.B</td>
<td>BitDefender</td>
<td>Android.Trojan.NioServ.A</td>
</tr>
<tr>
<td>Ikarus</td>
<td>Trojan.AndroidOS.NotCom</td>
<td>Emsisoft</td>
<td>Android.Trojan.NioServ.A (B)</td>
</tr>
<tr>
<td>Kaspersky</td>
<td>HEUR:Backdoor.AndroidOS.Nisev.b</td>
<td>MicroWorld-eScan</td>
<td>Android.Trojan.NioServ.A</td>
</tr>
<tr>
<td>ClamAV</td>
<td>Andr.Trojan.NotCompatible</td>
<td>AVG</td>
<td>Android/Nise</td>
</tr>
<tr>
<td>Baidu-International</td>
<td>Backdoor.AndroidOS.Nisev.AO</td>
<td>McAfee-GW-Edition</td>
<td>ArtemisI0E8525862F9C</td>
</tr>
<tr>
<td>TrendMicro</td>
<td>ANDROIDOS_NISEV.VTD</td>
<td>Fortinet</td>
<td>Android/Compatible.Altr.bdr</td>
</tr>
<tr>
<td>McAfee</td>
<td>ArtemisI0E8525862F9C</td>
<td>Commtouch</td>
<td>AndroidOS/GenBl.0E8525861Olympus</td>
</tr>
<tr>
<td>K7AntiVirus</td>
<td>Trojan ( 0040f2631 )</td>
<td>K7GW</td>
<td>Trojan ( 0040f2631 )</td>
</tr>
</tbody>
</table>

The detailed VirusTotal report can be viewed [Here](#)
Android APK Analysis

Application: com.android.fixed.update
Version: 1.0

Requested Permissions:
- android.permission.ACCESS_NETWORK_STATE
- android.permissionINTERNET
- android.permission.RECEIVE_BOOT_COMPLETED

Intent Filters (receiver):
- android.intent.action.BOOT_COMPLETED
- android.intent.action.USER_PRESENT

Visual UI Activities:

Application Services:
- FixedUpdate

Broadcast Receivers:
- OnBootReceiver

Information from Manifest
A malware soak test involves passively running a malware sample on a virtual machine and capturing any resulting network traffic.

**Initiate Malware Soak test:**

- **Duration:** 5 minutes
- **VM Host:** Android-1: Android 4.0 (Jelly Bean)

- **Malware Launch:** Automatic
- **DNS:** Actual
- **Listener Ports:** (comma separated list of TCP ports >1024 or IP address:Port)
- **Retain PCAP:** ✓

**Start**

**Run Sample in AVD**
### Existing Packet Capture Files:

<table>
<thead>
<tr>
<th>Date</th>
<th>Source</th>
<th>Details</th>
<th>Grade</th>
<th>Packets</th>
<th>Delete</th>
</tr>
</thead>
<tbody>
<tr>
<td>2013-04-01 06:49:18</td>
<td>By Arvind from Anubis</td>
<td></td>
<td></td>
<td>40</td>
<td></td>
</tr>
<tr>
<td>2013-10-28 16:19:25</td>
<td>Soak</td>
<td>Automated 10 minute soak test ()</td>
<td></td>
<td>553</td>
<td></td>
</tr>
<tr>
<td>2013-10-28 16:30:06</td>
<td>Soak</td>
<td>Automated 15 minute soak test ()</td>
<td></td>
<td>919</td>
<td></td>
</tr>
<tr>
<td>2013-11-28 11:19:08</td>
<td>Soak</td>
<td>Automated 5 minute soak test ()</td>
<td></td>
<td>201</td>
<td></td>
</tr>
<tr>
<td>2013-12-16 16:56:19</td>
<td>AndroidSandbox</td>
<td>Automated Android Sandbox execution (.DNS)</td>
<td></td>
<td>699</td>
<td></td>
</tr>
<tr>
<td>2015-12-04 11:26:48</td>
<td>AndroidSandbox</td>
<td>Interactive Android Sandbox 042b8abd13b6f9f9 execution (.DNS)</td>
<td>A</td>
<td>105</td>
<td></td>
</tr>
</tbody>
</table>

### Upload PCAP File:

- **Select File:**
- **Source:** upload
- **Details:**

**Upload PCAP**
Manual Demo – NotCompatible Proxy Bot
Disassemble APK
Directory structure created by apktool

Disassembled Dex in Smali format
<?xml version="1.0" encoding="utf-8" standalone="no"?>
  <uses-permission android:name="android.permissionINTERNET"/>
  <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
  <uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED"/>
  <application android:debuggable="true">
    <service android:enabled="true" android:name=".FixedUpdate"/>
    <receiver android:enabled="true" android:exported="true" android:name=".OnBootReceiver">
      <intent-filter>
        <action android:name="android.intent.action.BOOTH COMPLETED"/>
        <action android:name="android.intent.action.USER_PRESENT"/>
      </intent-filter>
    </receiver>
  </application>
</manifest>
Unzip APK file

C:\Users\kevinkm\Desktop\Test>unzip VID11219849.apk
Archive:  VID11219849.apk
extracting:  res/raw/data
inflating:  AndroidManifest.xml
extracting:  resources.arsc
inflating:  classes.dex
inflating:  META-INF/MANIFEST.MF
inflating:  META-INF/CERT.SF
inflating:  META-INF/CERT.RSA

C:\Users\kevinkm\Desktop\Test>dir
Volume in drive C is System
Volume Serial Number is C66F-E166

Directory of C:\Users\kevinkm\Desktop\Test
01/05/2016   03:10   PM   <DIR>       .
01/05/2016   03:10   PM   <DIR>       ..
03/11/2013   07:45   PM   2,160   AndroidManifest.x
03/11/2013   07:45   PM   23,748   classes.dex
01/05/2016   03:10   PM   <DIR>       META-INF
01/05/2016   03:10   PM   <DIR>       res
03/11/2013   07:45   PM   572   resources.arsc
12/08/2015   08:40   AM   14,030   VID11219849.apk
4 File(s)   40,510 bytes
4 Dir(s)   7,635,394,560 bytes Free

C:\Users\kevinkm\Desktop\Test>dex2jar classes.dex
0 [main] INFO com.googlecode.dex2jar.v3.Main - version:0.0.7.10-SNAPSHOT
Config file is encrypted using AES

View the Java source

```
package com.android.fixed.update;

import android.content.*;
import java.crypto.spec.*;
import java.security.*;
import java.crypto.*;

class Config
{
    private String CIPHER;
    private String KEY_ALG;
    public Context Owner;
    public int Port1;
    public int Port2;
    public String Server1;
    public String Server2;
    byte[] key;
    int lastShow;
    public String passkey;

    public Config()
    {
        this.passkey = "ZTY4MGE5YQ0";
        this.KEY_ALG = "AES";
        this.CIPHER = "AES/ECB/NoPadding";
        this.Server1 = "";
        this.Server2 = "";
        this.Port1 = 0;
        this.Port2 = 0;
        this.lastShow = 0;
    }
```
switch (unpack.Data.array()[0] & 0xFF) {
    default: {
        this.sendError(0, (byte)2);
        break;
    }
    case 1: {
        this.connectProxy(unpack.chanal, unpack.Data.array());
        break;
    }
    case 3: {
        this.shutdowChanal(unpack.chanal);
        break;
    }
    case 4: {
        this.sendPong();
        break;
    }
    case 253: {
        this.setTimeOut(unpack.Data.array());
        break;
    }
    case 254: {
        this.newReservServer(unpack.Data.array());
        break;
    }
    case 255: {
        this.newServer(unpack.Data.array());
        break;
    }
}
If you don't like Java you can look at the Smali code. It can be modified and the APK can be rebuilt using apktool.
Ping/Pong
C&C packet capture
Proxy Request
Data
NotCompatible - Overview

- Web Proxy Bot ported from Windows to Android environment.
- Allows remote miscreants to anonymously browse the web through the victim’s phone.
- Consumes lots of bandwidth, for example 165MB in two hours over 300K TCP sessions.
Phishing spam is used to lure the victim to an infected web site.

Web site tells you the browser is “not compatible” and provides an update.

The user downloads and installs update.apk.

Malware has no icon or user interface. It is automatically started on BOOT.

You can get rid of the infection by uninstalling the application.
NotCompatible – Operation

- Opens an encrypted configuration file containing the address and port number of the server.
- The bot connects to the server via TCP.
- Sophisticated command and control protocol is then used to multiplex Web proxy services over that connection.
- This provides an anonymous web browsing services to clients.
NotCompatible – Command & Control

- Simple command/response packet format contains both commands and data.
- Channel number can multiplex many connection at once.
- The ping and pong are used as a heartbeat when there is no proxy work to be done.
- Once a proxy request is issued the “raw data” commands are used to transfer the data in either direction.

Packet format:

| 0x04 | chan | type | length | ...data... |

- 0x04 - Protocol Version (1 byte)
- chan - Multiplexor Channel number (2 bytes)
- type - 0x00:Proxy Data, 0x01:Command (1 byte)
- len - Length of the data field (4 bytes)
- data - Is either proxy packet data or a command

Commands:

- Initial handshake: | 00 | 07000v00 |
- Proxy to IP: | 01 | 00 |IP & port|
- Proxy to domain name: | 01 | 01 |len|domain name|
- Response to proxy: | 02 | nnnn |
- End of proxy session: | 03 |
- Ping: | 04 |
- Pong: | 05 |
- Unknown (from victim): | FC | 01 |
- Set Timeout: | FD | timeout |
- Set Reserve Server: | FE | server IP and port |
- Set Primary Server: | FP | server IP and port |
NotCompatible – Uses & Impact

Uses
- Anonymous Web Browsing Service
- Providing Access to Restricted Foreign Content
- Ad-Click Fraud
- Web Site Optimization Fraud
- APT Probing and Exfiltration

Impact
- One user from Finland, roaming in the US, used over 165MBytes in less than two hours of airtime.
- In the lab it averages 100MBytes per hour.
- Causes huge data bills
- Caused the battery to run down quickly
- Who knows what sites your phone in visiting!!!
Summary

- Android malware analysis enables you to:
  - Know what the malware does
  - Understand the threat level
  - Detect and remediate the infection

- You should now know:
  - What tools are required
  - How to set up the network environment
  - How to use the tools
Questions?

Email: kevin.mcnamee@nokia.com
Twitter: @KevMcNamee