Vaccinating Android

BalCCon2k14 edition

#!/viris[@#q*]
/WhoAreWe

> Just two guys from Ex-Yu
> Having fun breaking stuff
> Love to play with apps
> Specialized in app security
> Only 6 hours to get here

#!/viris[@#q*]
Famous .si people
FBI, Slovenian and Spanish Police Arrest Mariposa Botnet Creator, Operators

Washington, D.C.
July 28, 2010

The FBI, in partnership with the Slovenian Criminal Police and the Spanish Guardia Civil, announced today significant developments in a two-year investigation of the creator and operators of the Mariposa Botnet. A botnet is a network of remote-controlled compromised computers.

The Mariposa Botnet was built with a computer virus known as “Butterfly Bot” and was used to steal passwords for websites and financial institutions. It stole computer users’ credit card and bank account information, launched denial of service attacks, and spread viruses. Industry experts estimated the Mariposa Botnet may have infected as many as 8 million to 12 million computers.

“In the last two years, the software used to create the Mariposa botnet was sold to hundreds of other criminals, making it one of the most notorious in the world,” said FBI Director Robert S. Mueller, II. “These cyber intrusions, thefts, and frauds undermine the integrity of the Internet and the businesses that rely on it; they also threaten the privacy and pocketbooks of all who use the Internet.”
Agenda

> Android mobile apps
> Short 101 APK
> Analysis (static, dynamic)
> Vaccinating APK, Android
> DEMO(s)
> The end
The diagram illustrates the various layers of the Android operating system:

**Applications**
- Home
- Dialer
- SMS/MMS
- IM
- Browser
- Camera
- Alarm
- Calculator
- Contacts
- Voice Dial
- Email
- Calendar
- Media Player
- Photo Album
- Clock

**Application Framework**
- Activity Manager
- Window Manager
- Content Providers
- View System
- Notification Manager
- Package Manager
- Telephony Manager
- Resource Manager
- Location Manager

**Libraries**
- Surface Manager
- Media Framework
- SQLite
- WebKit
- Libc
- OpenGL ES
- Audio Manager
- FreeType
- SSL

**Android Runtime**
- Core Libraries
- Dalvik Virtual Machine

**Hardware Abstraction Layer**
- Graphics
- Audio
- Camera
- Bluetooth
- GPS
- Radio (RIL)
- WiFi

**Linux Kernel**
- Display Driver
- Camera Driver
- Bluetooth Driver
- Shared Memory Driver
- Binder (IPC) Driver
- USB Driver
- Keypad Driver
- WiFi Driver
- Audio Drivers
- Power Management
HP research finds vulnerabilities in 9 of 10 mobile apps

Summary: Obvious security vulnerabilities are disturbingly common in corporate mobile apps. If HP can find them, so can malicious actors.

Tests run by HP Fortify, the company’s enterprise security arm, indicate that 90% of mobile apps have at least one security vulnerability.

The company used their Fortify On Demand for Mobile product to test the security posture of 2,107 applications published by 601 companies on the Forbes Global 2000. Only iOS apps were tested, but HP says that there is good reason to believe the same problems exist in any Android counterparts.

Overall, the problems fell into one of four categories. The analysis showed that 86% of apps that accessed potentially private data sources, such as address books or Bluetooth connections, lacked sufficient security measures to protect the data from access.

86% of apps tested lacked binary hardening protection. This refers to a group of techniques, many implemented simply with checkboxes at compile time, which protect against certain attacks, like buffer overflows, path disclosure and jailbreak detection.
Researchers compile list of Android apps that allow MitM attacks

Posted on 05 September 2014.

Around 350 Android apps that can be downloaded from Google play and Amazon stores fail to properly validate SSL certificates for HTTPS connections, and thus open users to Man-in-the-Middle attacks if they use them on insecure and open networks, a researcher with the CERT Coordination Center at the Software Engineering Institute at Carnegie Mellon University warned.

The vulnerable apps have been discovered via automated testing using the CERT Tapioca testing appliance, and the researchers keep a list of these updated - among them are OKCupid's official app, (ironically) a number of security apps, but most worryingly, a number of e-commerce (sucha as an eBay app for German users) and e-banking apps.

The list is not yet complete. The setup created by the researchers tests only one application at a time, and the testing started only a few weeks ago.
Things

> There is a (big) need for testing mobile apps
> Mobile app development feels like late 90’s development
> Our experience?
101 APK, Android

> APK? WTF?
> Get APK
> Decompile and analyze code
> Test
> Exploit

/viris[@#q*]
APK?

Android application package file (APK) is the package file format used to distribute and install application software and middleware onto Google's Android operating system, and certain other operating systems, such as Blackberry 10 Devices with the OS version 10.2.1.

Wikipedia
Android Applications

- .apk (Android Package) format
- Nothing more than a zip
- Written exclusively in Java, with native libraries in C/C++.
- Composed of components like Activities, Services, Broadcast Receivers, etc.
Getting APK

- Copy from the phone
- Copy from the backup
- Adb pull
- Download from untrusted source ;)

#viris[@#*]
Decompile

> Pull from phone.

adb pull /data/app(or app-private)/app1.apk
unzip app1.apk
dex2jar classes.dex
jogui classes2jar.jar

or convert to smali and then analyse the code

adb pull /data/app/app1.apk
unzip app1.apk
java -jar baksmali.jar -o C:\pentest\app classes.dex
Tools used for reversing APK

- Dex2Jar
- JD-GUI
- (Back)smali
- APKTool

http://www.decompileandroid.com/
Short demo
What to check?

› Transport security
  » Plaintext Traffic
  » Improper session handling
  » Validate SSL certificates

› Compiler protection

› UIWebviews
  » Data validation
  » Analyze UIWebView implementations

› Insecure data storage
  » SQLite DB
  » File caching
  » Checking log files
What to check? (cont)

➤ Logging
  » Custom logs
  » Crash reports logs and files

➤ Binary analysis
  » Disassemble/decompile the application
  » Detect obfuscations
  » Detect anti-debugging protections

➤ Client side injections

➤ Third party libraries

#/viris[@# q #]
Testing app

> Start simulator with proxy
> Install app in emulator or device
> Use Wireshark, Fiddler &/|| Zap &/|| Burp to monitor network
> Run app
> See logs, dump, crashes, files
### Request

<table>
<thead>
<tr>
<th>#</th>
<th>Host</th>
<th>Method</th>
<th>URL</th>
<th>Params</th>
<th>Modified</th>
<th>Status</th>
<th>Length</th>
<th>MIME Type</th>
<th>Extent</th>
</tr>
</thead>
<tbody>
<tr>
<td>71</td>
<td><a href="http://kelimeavisl.fugo.mobi">http://kelimeavisl.fugo.mobi</a></td>
<td>GET</td>
<td>/servicesV2_SL/info.php?nuid=...</td>
<td></td>
<td></td>
<td>200</td>
<td>905</td>
<td>text</td>
<td>php</td>
</tr>
<tr>
<td>73</td>
<td><a href="http://mob.adwhirl.com">http://mob.adwhirl.com</a></td>
<td>GET</td>
<td>/getInfo.php?appid=f3743c9b9c1...</td>
<td></td>
<td></td>
<td>200</td>
<td>588</td>
<td>JSON</td>
<td>php</td>
</tr>
<tr>
<td>74</td>
<td><a href="http://i.w.inmobi.com">http://i.w.inmobi.com</a></td>
<td>POST</td>
<td>/showad.asm</td>
<td></td>
<td></td>
<td>200</td>
<td>1541</td>
<td>XML</td>
<td>asm</td>
</tr>
<tr>
<td>75</td>
<td><a href="http://met.adwhirl.com">http://met.adwhirl.com</a></td>
<td>GET</td>
<td>/exmet.php?appid=f3743c9b9c1...</td>
<td></td>
<td></td>
<td>200</td>
<td>119</td>
<td>HTML</td>
<td>php</td>
</tr>
</tbody>
</table>

### Request Details:

```plaintext
GET
&hash=499eebdf23d007af336cd04f44c50ff6 HTTP/1.1
User-Agent: Dalvik/1.6.0 (Linux; U; Android 4.2.2; GT-I9000 Build/JDQ39E)
Host: kelimeavisl.fugo.mobi
Connection: Keep-Alive
Accept-Encoding: gzip
```
### Filter: Hiding CSS, image and general binary content

<table>
<thead>
<tr>
<th>#</th>
<th>Host</th>
<th>Method</th>
<th>URL</th>
<th>Params</th>
<th>Modif.</th>
<th>Status</th>
<th>Length</th>
<th>MIME type</th>
<th>Ext.</th>
</tr>
</thead>
<tbody>
<tr>
<td>71</td>
<td><a href="http://kelmeavisl.fugo.mobi">http://kelmeavisl.fugo.mobi</a></td>
<td>GET</td>
<td>/servicesV2_SL/info.php?nuid=...</td>
<td></td>
<td></td>
<td>200</td>
<td>634</td>
<td>text</td>
<td>php</td>
</tr>
<tr>
<td>72</td>
<td><a href="http://adserver.fugo.mobi">http://adserver.fugo.mobi</a></td>
<td>GET</td>
<td>/ads/geomap.php?platform=and...</td>
<td></td>
<td></td>
<td>200</td>
<td>255</td>
<td>text</td>
<td>php</td>
</tr>
<tr>
<td>73</td>
<td><a href="http://mob.adwhirl.com">http://mob.adwhirl.com</a></td>
<td>GET</td>
<td>/getInfo.php?appid=f3743c9b9c1...</td>
<td></td>
<td></td>
<td>200</td>
<td>588</td>
<td>JSON</td>
<td>php</td>
</tr>
<tr>
<td>74</td>
<td><a href="http://i.w.inmobi.com">http://i.w.inmobi.com</a></td>
<td>POST</td>
<td>/showad.asm</td>
<td></td>
<td></td>
<td>200</td>
<td>1541</td>
<td>XML</td>
<td>as</td>
</tr>
<tr>
<td>77</td>
<td><a href="http://met.adwhirl.com">http://met.adwhirl.com</a></td>
<td>GET</td>
<td>/exmet.php?appid=f3743c9b9c1...</td>
<td></td>
<td></td>
<td>200</td>
<td>119</td>
<td>HTML</td>
<td>as</td>
</tr>
<tr>
<td>78</td>
<td><a href="http://kelmeavisl.fugo.mobi">http://kelmeavisl.fugo.mobi</a></td>
<td>GET</td>
<td>/servicesV2_SL/info.php?nuid=...</td>
<td></td>
<td></td>
<td>200</td>
<td>905</td>
<td>text</td>
<td>php</td>
</tr>
</tbody>
</table>

### Content-Length: 448

Date: Sat, 30 Nov 2013 11:14:15 GMT

X-Varnish: 1695575935 1695575798

Age: 1

Via: 1.1 varnish

Connection: keep-alive

```
MBBXwfrbrAa1307KDIgf7MZYeEZbOhng5Rg07Yhdw3Hs8izrSikFh27erHjff1svP3FrelJctH1qnfNIPAgj81NXd5Zzj02KIPnAvhpzRRAAnT83K/JjVBO4G6+FkstjJDOF/0e9SWYhA9Czwyly3kNGUBmfrNGaih10hXAlUHNBDMYsPaXARadh+rxIL5+3LMMnELTPs8uFRWuItUbiu1j/Ulve2Ns+CGX/erwJEARQb2105ZhaWzQVb7TPpvMVZFutCtJCMvTLMHDQXjvbJiapbllIPqUNGT9ifW8BPBe9jycBUGX58NGpgEygj13dVlDExsDyD7x+4n7th+anuDv3NFv4R991T2LItUmdB7fr8KZshJ/TEk7/P1xrgha7fioY
```
## General Analysis Information

<table>
<thead>
<tr>
<th>PROPERTY</th>
<th>VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Filename</td>
<td>08CFFA8F55BE4BBED2704355876B618F_2.zip</td>
</tr>
<tr>
<td>App package name</td>
<td>com.android.services</td>
</tr>
<tr>
<td>Version</td>
<td>351</td>
</tr>
<tr>
<td>Minimum SDK Version</td>
<td>7</td>
</tr>
<tr>
<td>Target SDK Version</td>
<td></td>
</tr>
<tr>
<td>Checksum</td>
<td>08cfa8f55be4bbbed2704355876b618f</td>
</tr>
<tr>
<td>Developer</td>
<td>[Bad Signature]</td>
</tr>
<tr>
<td>Shared UID</td>
<td></td>
</tr>
</tbody>
</table>

## Program statistics

### Classes per package
- 13.1%: 2,572 classes
- 2.38%: 226 classes
- 4.76%: 89 classes
- 1.19%: 22 classes
- 8.33%: 76 classes
- 14.29%: 64 classes
- 3.57%: 16 classes
- 14.29%: 64 classes
- 20.24%: 101 classes

Classes per package show legend.
Security Researcher Accidentally Crashes Google Play When Testing POC App

Code Analyzer: C, C++, Java

Turkish security researcher Ibrahim Balic claims to have found an Android vulnerability that could lead to memory corruption. While testing his findings, he may have crashed Google Play a couple of times.

According to the expert, Android 2.3, 4.2.2 and 4.3 are certainly affected, but he believes that all versions of the operating system are vulnerable.

He has found that executing a malformed APK file leads to a denial-of-service (DOS) condition and the device freezes. Balic wanted to test his theory against Bouncer, the Android anti-malware system developed by Google, so he uploaded a malformed APK file to Google Play.

Shortly after, he started getting errors on Google Play. In addition, during the time he performed his tests, many people reported being unable to upload applications to Google’s app market.

“I think it was probably because of testing my PoC exploit on Google Play,” Balic noted in a blog post.
Static analysis

- You need to know how read Java code
- Cannot see all runtime replies
- Obfuscated, renamed?
- Identify important segments in code
public class HttpCall
{
    private static String SECURITY_TOKEN = "AE94DFKMA6F4U94MNDF324SF3ADASCAR4GASDF94";
    private CookieStore cookieStore = new BasicCookieStore();
    private HttpClient httpclient = new DefaultHttpClient();
    private HttpContext localContext = new BasicHttpContext();

    public HttpCall()
    {
        this.localContext.setAttribute("http.cookie-store", this.cookieStore);
    }

    // ERROR //
    public String call(String paramString)
    {
        Byte code:
        // 0: new 52 org/apache/http/client/methods/HttpPost
        // 3: dup
        // 4: aload_1
        // 5: invokespecial 55 org/apache/http/client/methods/HttpPost:<init> (Ljava/lang/String;)V
        // 8: astore_2
        // 9: aload_2
        // 10: ldc 57
        // 12: getstatic 18 com/ttech/turkcell/sdk/util/HttpCall:SECURITY_TOKEN Ljava/lang/String;
        // 15: invokevirtual 61 org/apache/http/client/methods/HttpPost:setHeader (Ljava/lang/String;Ljava/lang/String);
        // 18: aload_0
        // 22: astore_3

        //...
Dynamical analysis

- Monitoring/changing traffic with proxy
- Debugging
- Reflection
Reflection

> "Reflection" is a language's ability to inspect and dynamically call classes, methods, attributes, etc. at runtime.

> Java looking Java
Debugging vs Reflection

> Higher level view
> Better idea how application works
> Java like access to objects, methods, variables
> Interaction with application
Features

▶ Access all variables
▶ Change values of variables
▶ Call methods
▶ Use variables and scripts
▶ Use full BeanShell
▶ Write Java code
BeanShell

Java Interpreter
Scripting Language
Small
Embeddable / Extensible
A natural scripting language for Java
What do we see..

- Authentication PINs in system logs
- Session identifiers and credentials cached in WebView
- Inappropriate data stored in local SQLite databases
- Internal IP’s
- Hardcoded usernames, passwords
- Testing cases left inside code

#/viris[@ #а #*]
I WANT MORE!
Vaccine

> Repackaging if injecting in APK
> Service injection
> Injecting Beanshell
> Connection and Dynamical analysis
Vaccine (bash)script

Preparing the APK

» Copy APK
» Unzip
» Baksmali classes.dex - smali source code
» Adding smali source of service
» Smalilng source - classes.dex
» Changing AndroidManifest.xml
» Replacement of classes.dex and AndroidManifest.xml
» Removing signature
» Signing
» Installing the mobile application
» Starting the service
» Connecting and showing UI
Vaccine

> Accessing objects and fields
> Executing methods
> Using objects, variables in Java source and Beanshell scripts
> ...

#viris[@#qa*]
object = object();
object.flag = true;

foo() {
  run()
      
  while(object.flag)
    print("Running...");
    Thread.sleep(2000);

  return this;
}

foo = foo();
new Thread(foo).start();

#/viris[@qqq]
Demo(s)

./vaccine.sh -i android.apk -p 8888
Disclaimer

This presentation was created for educational purposes. We will not take any responsibility for any action you cause using the information shown in this presentation. Please do not contact us with blackhat type hacking requests. Thanks!

Original taken from: http://www.loo.ro/

#viris[@#&*]
Demo(s)

./vaccine.sh -i android.apk -p 8888

#/viris[@#q*]
Android DDI: Dynamic Dalvik Instrumentation

30th Chaos Communication Congress
Hamburg, Dec. 29th, 2013

Collin Mulliner
collin[at]mulliner.org  twitter: @collinrm

#viris[@* ]
Injecting vaccine at runtime

- Little hacking provided Collin’s examples
- Instead of changing APK, we “hijack” running process (in our case zygote)
- Inject shared library into process
- Hook `android.app.Activity` onStart method
- Injects Vaccine service and additional BeanShell classes when app is started
- Use vaccine as before
Demo

> Is it possible to inject Vaccine into Google apps at runtime?
Pros/cons APK Android

> APK
  » No need for rooted phone
  » Untrusted sources
  » Download, modify, upload

> Android
  » No need for APK modification
  » Rooted phone
  » Injecting shared libs (more skills needed)
CHALLENGES AHEAD
Possible usage

- Not only for Android
- Reflection is still NOT dead
- Tested with Oracle Foms
- Have idea to use it with other Java apps/applets (Minecraft maybe)

> Ultimate cheating platform

#/viris[@#q*]
Final thoughts

> One script, small GUI tool (never be finished)
> Help testers, researchers (hackers, cheaters)
> Open for suggestions, improvements, comments
Questions?
www.github.com/viris

@MilanGabor

@alm8i

Thank You!!