Creating a kewl and simple Cheating Platform on Android

Milan Gabor & Danijel Grah
Who Are We

> Just two guys from Slovenia
> Having fun breaking stuff
> Love to play with apps

> BSidesLV, DEF CON Wall of Sheep, BalcCon, Hacktivity, GrrCON, Hackito Ergo Sum, DefCamp, Hek.si
Famous .si people
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
> Analysis (static, dynamic)
> Vaccinating APK, Android
> DEMO
> DEMO
> DEMO
> The end
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.

By Larry Seltzer for Zero Day | November 19, 2013 -- 13:15 GMT (05:15 PST)

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.
DoubleDirect hackers snaffle fandroid and iPhone-strokers' secrets

Windows and Linux seem immune from redirection assault

By John Leyden, 21 Nov 2014

Hackers are running “Man-in-the-Middle” attacks (MitM) against smartphones using a new attack technique, security researchers warn.

The so-called DoubleDirect technique enables an attacker to redirect a victim’s traffic to the attacker’s device. Once redirected, the attacker can steal credentials and deliver malicious payloads to the victim’s mobile device that can not only quickly infect the device, but also spread throughout a corporate network,” according to mobile security firm Zimperium.

Zimperium has detected the DoubleDirect technique in the wild in attacks against the customers of web giants including Google, Facebook, Live.com and Twitter, across 31 countries.

Hackers are also using DoubleDirect technique to gain access to victims’ devices, essentially to steal usernames, emails, and passwords.

DoubleDirect creates a means to run man-in-the-middle attacks targeting smartphone and tablets users on devices running either iOS or Android. Mac OSX users are also potentially vulnerable but Windows and Linux users would appear to be immune because their operating systems don’t accept ICMP redirection packets that carry malicious traffic. A blog post by Zimperium (extract below) explains the mechanism of the attack in greater depth.

DoubleDirect uses ICMP Redirect packets to modify routing tables of a victim's device.
Our story
YES, we can!
We want something that works!
We want to test mobile apps!
> Living inside of APK
> Changing and accessing variables
> Executing code at runtime
> Effectively and easy to use
> Java based
Demo/Video
Java code is obfuscated
> Static analysis
> Dynamical analysis
> What if...?
> Hard time
KEEP CALM AND GET BACK TO BASICS
Testing app/1

> Get the APK
> Unpack
> Decompile
> Check code
> Identify important segments
```java
public class HttpCall {

    private static String SECURITY_TOKEN = "AE94D418E47F3D453248C5C546C8A74214D9C98122C2F94";
    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);
    }

    public String call(String paramString) {
        // Byte code:
    }
```

```java
public void

```
Testing app/ 2

> 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>Extens</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>694</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>
<tr>
<td>76</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>
</tbody>
</table>

---

**GET**

/getserviceSV2_SL/info.php?nuid=354406042390139b4:07:f9:8d:6b:83&uid=354406042390139&agent=android_3&ver=3.1.3&hash=499eebfdd23d007af336cd04f44c50ffcc 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
<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>Extens</th>
</tr>
</thead>
<tbody>
<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>php</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>php</td>
</tr>
<tr>
<td>78</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>
</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

MBBXwfrbrAa1307KDIgf7MZyEZbOhngSroO7Yhdw3Hs8izrSikFh27erHjf1svP3FrEjctH1qnfNIPAgj1b8INXd5Zzjo2KiPnPvAhSpzRAARt83K/jIVB04G6+FkStjDOF/0e9SWYhA9Czwly3kNGUBmfnGaihv10hXAIuHNBDMYSpXARqAdh+Rxl5+3LMnELTP5g8ufTwIlUBiuJ/Lulve2Ns+CGX/erwJEARQb2105ZhawZVvb7TPpvMvZFuCthCjMvTMHdQXjvblazphbllIPqUENGt9iFW8BPe9JycBUGX58NGpgEyj13dVLiDuEXsDyD7x+4n7th+anuDV3NFw4R991T2LltUmdbB7fr8KZshj/TEk7/P1xrghaT7f1oV
Dictionary

➤ Dynamical analysis
➤ Reflection
➤ BeanShell
➤ Combination of static/dynamic
Reflection

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

Java looking Java
BeanShell

- Java Interpreter
- Scripting Language
- Small
- Embeddable / Extensible
- A natural scripting language for Java
I WANT MORE!
DIG DEEPER
Vaccine

Android

- APK
  - Service
  - Beanshell

OS

Vaccine

- Testing component
- Controlling component

Tools

- smali / baksmali
- zip
- adt bundle

adb, TCP
./vaccine -i game.apk
./vaccine -i game.apk
./vaccine -i game.apk
```java
object = object();
object.flag = true;

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

foo = foo();
new Thread(foo).start();
```
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.lo0.ro/
Demo(s)

./vaccine -i android.apk -p 8888
Android DDI: Dynamic Dalvik Instrumentation

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

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

/viris[@]**q***]/
Dictionary

> ADBI, DDI
> Zygote
> Shared libraries
> Hooking
> JNI and native functions
Injecting vaccine at runtime

> Little hacking provided Collin’s examples
> Prepared shared library with DDI framework
> Using hijack from ADBI framework to „hijack“ Zygote
> When Zygote specializes the shared library is loaded into target process and executed
> Shared library contains native code that „replaces“ (hooks) android.app.Activity onStart method
> Native methods loads classes from /data/dalvik-cache/va-classes.dex (Vaccine service, Beanshell)
> Native method gives execution over to original method
> Connect and use Vaccine as before

#viris[@ # @ *]
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)

> SIMPLE and Ultimate cheating platform
Final thoughts

> One script, small GUI tool (never be finished)
> Help testers, researchers (hackers, cheaters)
> Open for suggestions, improvements, comments