How to do it Wrong: Smartphone Antivirus and Security Applications Under Fire

Stephan Huber, Siegfried Rasthofer, Steven Arzt, Michael Tröger, Andreas Wittmann, Philipp Roskosch, Daniel Magin
Who are we

Stephan
- Mobile Security Researcher at Fraunhofer SIT
- Enjoys teaching students in Android Hacking

Siegfried
- 4th year PhD Student at TU Darmstadt / Fraunhofer SIT
- Enjoys drinking bavarian beer
- @teamsik
Mobile Banking Security

How Can You Protect Yourself?

The likelihood of fraud is no greater than using Your Link but you should follow some similar safety precautions that you would when browsing the internet or accessing your email. There are several security tips and precautions that you can exercise to practice safe mobile banking.

- **Download the App from known sources** – You may download the Dedhamobile app from iTunes® App Store, Android Marketplace, or directly from m.dedhamsavings.com on your mobile device.

- **Protecting your Identity** – never respond to a "phishing" text or email message that requests any account information that you did not initiate. Dedham Savings would never request information in this manner.

- **Anti-virus software** – if it is available to you, we suggest to keep your phone safe at all times to install mobile anti-virus and anti-spyware software on your mobile device and keep it updated.
Spam Protection

Malware Detection Engine

Secure Browsing

Privacy Advisor

Device Configuration Advisor

Premium Features
Kaspersky Internet Security
needs access to

$ In-app purchases

Device & app history

Cellular data settings

Identity

Contacts/Calendar

Location

SMS

Phone

Photos/Media/Files

Camera/Microphone

Wi-Fi connection information

Device ID & call information

ACCEPT
<table>
<thead>
<tr>
<th>App</th>
<th>GooglePlay Downloads</th>
</tr>
</thead>
<tbody>
<tr>
<td>“Pseudo” AV Apps</td>
<td></td>
</tr>
<tr>
<td>AndroHelm</td>
<td>1-5 Mio</td>
</tr>
<tr>
<td>Malwarebytes</td>
<td>5-10 Mio</td>
</tr>
<tr>
<td>ESET</td>
<td>5-10 Mio</td>
</tr>
<tr>
<td>Avira</td>
<td>10-50 Mio</td>
</tr>
<tr>
<td>Kaspersky</td>
<td>10-50 Mio</td>
</tr>
<tr>
<td>McAfee</td>
<td>10-50 Mio</td>
</tr>
<tr>
<td>CM Security</td>
<td>100-500 Mio</td>
</tr>
</tbody>
</table>
#Challenges

- Premium Upgrade for Free?
- Misuse Lost-Device Feature (Ransomware)?
- Remotely Influence Scan Engine Behavior?
- Remote Code Execution?
Premium Upgrade for Free?
(1/2 Examples)
AndroHelm
Free Premium the Simple Way

AndroHelm Security

- AntiVirus app: 9,99 €
- Virenschutz: 10,99 €
- AntiVirus for Tablet: 129,99 €
- AntiVirus Security: 20,85 €
- AntiVirus Android: 99,99 €
Let’s Have a Look at the Free App

Interesting code snippet:

```javascript
... 
this.toast("Thank you for upgrading to PRO!");
//shared pref value set to true
this.prefs.putBoolean("isPro", true);
... 
```

Key/value pair for xml file

```xml
<map>
  <int name="dialogShowTimes" value="1" />
  <boolean name="hasDatabase" value="true" />
  <string name="lastFragment"></string>
  <boolean name="isPro" value="true" />
</map>
```

SharedPreferences at first install:
Changing XML File Without Root

backup
com.androhelm.antivirus.free2

restore
com.androhelm.antivirus.free2

adb

default bridge

tar -xvf mybackup.tar
nano com.androhelm.antivirus.free.preferences.xml

* https://github.com/nelenkov/android-backup-extractor
Premium Upgrade for Free?
(2/2 Examples)
ESET
There are known vulnerabilities for SSL/TLS, but is there an easier way?
One requirement for secure communication is the verification of the SSL certificate!

```java
final class jl implements X509TrustManager {
    ...

    public void checkServerTrusted(X509Certificate[] cert, String s) 
        throws CertificateException {

        //please insert verification here

    } //end of the method
} // end of the class
```
ESET License Verification

ESET Security App

SSL/TLS Protection

ESET Backend

<WTF?>

<Base64 decoded VALUE in HEX: 15 d6 b1 e9 ae f0 8f f2 3e 64 47 ad

<Base64 decoded VALUE in HEX: 15 d6 b1 e9 ae f0 8f f2

<Node NAME="LicenseUsername" VALUE="Fdax6a7wj/I+ZEet" TYPE="STRING"/>

<Node NAME="LicensePassword" VALUE="Fdax6a7wj/I=" TYPE="STRING"/>
Let's do some Crypto Analysis

Classic chosen plaintext attack

<table>
<thead>
<tr>
<th>Plaintext</th>
<th>Cipher (base64)</th>
<th>Cipher (hexbyte)</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>ANY=</td>
<td>0x0 0xd6</td>
</tr>
<tr>
<td>aa</td>
<td>ANa16Q==</td>
<td>0x0 0xd6 0xb5 0xe9</td>
</tr>
<tr>
<td>aaaaa</td>
<td>ANa16bzwml=</td>
<td>0x0 0xd6 0xb5 0xe9 0xbc 0xf0 0x9a 0xf2</td>
</tr>
<tr>
<td>b</td>
<td>A9Y=</td>
<td>0x3 0xd6</td>
</tr>
<tr>
<td>bbbbb</td>
<td>A9a26b/wml=</td>
<td>0x3 0xd6 0xb6 0xe9 0xbf 0xf0 0x99 0xf2</td>
</tr>
<tr>
<td>abc</td>
<td>ANa26b7w</td>
<td>0x0 0xd6 0xb6 0xe9 0xbe 0xf0</td>
</tr>
<tr>
<td>ccc</td>
<td>Ata36b7wmPl=</td>
<td>0x2 0xd6 0xb7 0xe9 0xbe 0xf0 0x98 0xf2</td>
</tr>
<tr>
<td>ddd</td>
<td>Bdaw6bnwn/I=</td>
<td>0x5 0xd6 0xb0 0xe9 0xb9 0xf0 0x9f 0xf2</td>
</tr>
<tr>
<td>eee</td>
<td>BNax6bjwnvl=</td>
<td>0x4 0xd6 0xb1 0xe9 0xb8 0xf0 0x9e 0xf2</td>
</tr>
</tbody>
</table>
Let’s do some Crypto Analysis

Classic chosen plaintext attack

<table>
<thead>
<tr>
<th>Plaintext</th>
<th>Cipher (base64)</th>
<th>Cipher (hexbyte)</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>ANY=</td>
<td>0x0</td>
</tr>
<tr>
<td>aa</td>
<td>ANa16Q==</td>
<td>0x0 0xb5</td>
</tr>
<tr>
<td>aaaa</td>
<td>ANa16bzwmvI=</td>
<td>0x0 0xb5 0xbc 0x9a</td>
</tr>
<tr>
<td>b</td>
<td>A9Y=</td>
<td>0x3</td>
</tr>
<tr>
<td>bbbb</td>
<td>A9a26b/wmfl=</td>
<td>0x3 0xb6 0xbf 0x99</td>
</tr>
<tr>
<td>abc</td>
<td>ANa26b7w</td>
<td>0x0 0xb6 0xbe</td>
</tr>
<tr>
<td>cccc</td>
<td>Ata36b7wmPl=</td>
<td>0x2 0xb7 0xbe 0x98</td>
</tr>
<tr>
<td>dddd</td>
<td>Bdaw6bnwn/I=</td>
<td>0x5 0xb0 0xb9 0x9f</td>
</tr>
<tr>
<td>eeee</td>
<td>BNax6bjwnvl=</td>
<td>0x4 0xb1 0xb8 0x9e</td>
</tr>
</tbody>
</table>
Let’s do some Crypto Analysis

Clean up:

<table>
<thead>
<tr>
<th>Plaintext</th>
<th>Cipher (base64)</th>
<th>Cipher (hexbyte)</th>
</tr>
</thead>
<tbody>
<tr>
<td>aaaa</td>
<td>ANa16bzwmvl=</td>
<td>0x0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0xbc</td>
</tr>
<tr>
<td>bbbb</td>
<td>A9a26b/wmfl=</td>
<td>0x3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0xb6</td>
</tr>
<tr>
<td>cccc</td>
<td>Ata36b7wmPl=</td>
<td>0x2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0xb7</td>
</tr>
<tr>
<td>abc</td>
<td>ANa26b7w</td>
<td>0x0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0xb6</td>
</tr>
<tr>
<td>ddddd</td>
<td>Bdaw6bnwnl=</td>
<td>0x5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0xb0</td>
</tr>
<tr>
<td>eeeee</td>
<td>BNax6bjwnvl=</td>
<td>0x4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0xb1</td>
</tr>
</tbody>
</table>

- 2nd byte is not required
- No chaining
- Looks like a simple substitution
Here Comes the Key

key[0] = ?

a = 0x61 -> ? -> 0x0

<table>
<thead>
<tr>
<th>Letter</th>
<th>Decimal</th>
<th>Hex</th>
<th>1. Cipher</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>97</td>
<td>0x61</td>
<td>0x0</td>
</tr>
<tr>
<td>b</td>
<td>98</td>
<td>0x62</td>
<td>0x3</td>
</tr>
<tr>
<td>c</td>
<td>99</td>
<td>0x63</td>
<td>0x2</td>
</tr>
</tbody>
</table>
Here Comes the Key

key[0] = a = 0x61

\[
\begin{align*}
\text{a} &= 0x61 \\
\text{XOR} &\rightarrow \text{0x0}
\end{align*}
\]

<table>
<thead>
<tr>
<th>Letter</th>
<th>Decimal</th>
<th>Hex</th>
<th>1. Cipher</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>97</td>
<td>0x61</td>
<td>0x0</td>
</tr>
<tr>
<td>b</td>
<td>98</td>
<td>0x62</td>
<td>0x3</td>
</tr>
<tr>
<td>c</td>
<td>99</td>
<td>0x63</td>
<td>0x2</td>
</tr>
</tbody>
</table>
Here Comes the Key

key[0] = a = 0x61

\[
b = 0x62 \rightarrow \text{XOR} \rightarrow 0x3
\]

<table>
<thead>
<tr>
<th>Letter</th>
<th>Decimal</th>
<th>Hex</th>
<th>1. Cipher</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>97</td>
<td>0x61</td>
<td>0x0</td>
</tr>
<tr>
<td>b</td>
<td>98</td>
<td>0x62</td>
<td>0x3</td>
</tr>
<tr>
<td>c</td>
<td>99</td>
<td>0x63</td>
<td>0x2</td>
</tr>
</tbody>
</table>
Here Comes the Key

key[0] = a = 0x61

\[
\begin{align*}
\text{c} & = 0x63 \\
\text{XOR} & \\
\text{c} & = 0x2
\end{align*}
\]

<table>
<thead>
<tr>
<th>Letter</th>
<th>Decimal</th>
<th>Hex</th>
<th>1. Cipher</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>97</td>
<td>0x61</td>
<td>0x0</td>
</tr>
<tr>
<td>b</td>
<td>98</td>
<td>0x62</td>
<td>0x3</td>
</tr>
<tr>
<td>c</td>
<td>99</td>
<td>0x63</td>
<td>0x2</td>
</tr>
</tbody>
</table>
Here Comes the Key

Cipher = 0x0 0xb5 0xbc 0x9a …

aaaa = 0x61 0x61 0x61 0x61 …

\[\text{XOR}\]

Key = 0x61 0xd4 0xdd 0xfb …

<table>
<thead>
<tr>
<th>Letter</th>
<th>Decimal</th>
<th>Hex</th>
<th>1. Cipher</th>
</tr>
</thead>
<tbody>
<tr>
<td>aaaa</td>
<td>97 97 97 97</td>
<td>0x61 0x61 0x61 0x61</td>
<td>0x0 0xb5 0xbc 0x9a</td>
</tr>
</tbody>
</table>
ESET License Verification

ESET Security App

SSL/TLS Protection

<NODE NAME="LicenseUsername" VALUE="Fdax6a7wj/I+ZEet" TYPE="STRING"/>

key = [0x61 0xd4 0xdd 0xfb 0x5b 0x35 0xb7 0x19 0xec 0x2b 0xd9 0x4b 0x7 ...]

* test
#Challenges

- Premium Upgrade for Free?
- Misuse Lost-Device Feature (Ransomware)?
- Remotely Influence Scan Engine Behavior?
- Remote Code Execution?
Misuse Lost-Device Feature (Ransomware)?

(1 Example)

AndroHelm
Misuse Lost-Device Feature

What is a lost-device feature?
• Device Location
• Remote Alarm
• Remote Wipe
• Remote Lock
• …

Can we abuse “Remote Lock“ or “Wipe“?
Remote Communication With Smartphone

Examples:
- Google Cloud Messaging (GCM)
- Push Service Provider
- **SMS Messages**
Androhelm Anti-Theft SMS Protocol

- Anti-theft feature is enabled
- User sends SMS command

Feature not enabled, still possible to bypass the authentication?
Remote Protocol with Activated Anti-Theft

wait for incoming SMS

split at [SPACE]

check password

true

execute command

false

myPass[SPACE]wipe[SPACE]

SMS_PASSWORD := "myPass"
command := "wipe"

//Stored password
pwd := "myPass"
pwd == SMS_PASSWORD?
"myPass" == "myPass"

command := "wipe"
execute(command)
Remote Protocol Deactivated Anti-Theft

- [Image: SMS icon] wait for incoming SMS
- split at [SPACE]
- check password
- true: execute command
- false

Attacker
- [SPACE]wipe[SPACE]somestring
- empty string as pwd
- SMS_PASSWORD := ""
- command := "wipe"
- //default password
- pwd := ""
- pwd == SMS_PASSWORD?
- "" == ""
- command := "wipe"
- execute(command)
#Challenges

- ✔ Premium Upgrade for Free?
- ✔ Misuse Lost-Device Feature (Ransomware)?
- □ Remotely Influence Scan Engine Behavior?
- □ Remote Code Execution?
Remotely Influence Scan Engine Behavior?

(1 Example)

Malwarebytes
Unprotected Signature Updates

Malwarebytes App

Man-in-the-Middle Attacker

Malwarebytes Backend

(signature) update request

TI028Z%th5Y'uX4>dQz… = [Encryption Key]

(signature) update request

TI028Z%th5Y'uX4>dQz… = [Encryption Key]

remove signatures
#Challenges

- ✔ Premium Upgrade for Free?
- ✔ Misuse Lost-Device Feature (Ransomware)?
- ✔ Remotely Influence Scan Engine Behavior?
- Remote Code Execution?
Remote Code Execution?

(1 Example)

Kaspersky
### Zip Directory Traversal

#### Special filename for a zip entry

```
/tmp$ unzip -l zipfile.zip
Archive:  zipfile.zip
Length     Date      Time    Name
---------  ---------- -----   ----
 22  2016-06-28 13:49   ../../../tmp/dir2/badfile.txt
 24  2016-06-28 13:43   file1.txt
---------                     -------
46                     2 files
```
What happens if we unzip?

```
/tmp$ unzip zipfile.zip -d ./dir1/
Archive:  zipfile.zip
warning:  skipped "../" path component(s) in ../../../tmp/dir2/badfile.txt
extracting:  ./dir1/tmp/dir2/badfile.txt
extracting:  ./dir1/file1.txt
```

```
/tmp$ find /tmp/dir1/
/tmp/dir1/
/tmp/dir1/file1.txt
/tmp/dir1/tmp
/tmp/dir1/tmp/dir2
/tmp/dir1/tmp/dir2/badfile.txt
```

Zip Directory Traversal - Concept

disable escaping

/tmp$ unzip zipfile.zip -d ./dir1/
Archive: zipfile.zip
  extracting: ./dir1/../../../tmp/dir2/badfile.txt
  extracting: ./dir1/file1.txt

/tmp$ ls /tmp/dir1/
  file1.txt

/tmp$ ls /tmp/dir2/
  badbile.txt
Kaspersky RCE

- Plaintext, no encryption
- No authentication
- Self-made integrity protection

All important files are signed!

But what is an important file?
Kaspersky RCE

Man-in-the-Middle Attacker

Kaspersky Internet Security App

Kaspersky Backend

GET-Requests of Application:

http://www.kaspersky.com/ucp-ready
http://ipm.kaspersky.com/600eb07a-2926-4407-b014-d3e8c77b0086.zip
http://ipm.kaspersky.com/eeea9321-5eac-4709-9046-8475ee951c82.zip
http://downloads7.kaspersky:labs.com/index/u0607g.xml
...
Finding Attack Vector

App’s folder containing executables

```plaintext
./app_bases/pdm.cfg
./app_bases/pdm.jar  # included in apk file
...                   # contains classes.dex
./app_bases/rootdetector.jar  # signed, can not be manipulated!!
...
./app_ipm/600eb07a-2926-4407-b014-d3e8c77b0086/respond.min.js
./app_ipm/600eb07a-2926-4407-b014-d3e8c77b0086/[Content_Types].xml
./app_ipm/600eb07a-2926-4407-b014-d3e8c77b0086/1000_768.css
./app_ipm/600eb07a-2926-4407-b014-d3e8c77b0086/KISA_EN_Trial.html
./app_ipm/600eb07a-2926-4407-b014-d3e8c77b0086/evil.txt
```

content of our zip archive  

injected file
Finding Attack Vector

App’s folder

```plaintext
./app_bases/pdm.cfg
./app_bases/pdm.jar
... 
./app_bases/rootdetector.jar
... 
./app_ipm/600eb07a-2926-4407-b014-d3e8c77b0086/respond.min.js
./app_ipm/600eb07a-2926-4407-b014-d3e8c77b0086/[Content_Types].xml
./app_ipm/600eb07a-2926-4407-b014-d3e8c77b0086/1000_768.css
./app_ipm/600eb07a-2926-4407-b014-d3e8c77b0086/KISA_EN_Trial.html
./app_ipm/600eb07a-2926-4407-b014-d3e8c77b0086/pdm.jar
```

Can we overwrite this file?

PATH TRAVERSAL!

another injected file
The Exploit

• Overwrite original pdm.jar with manipulated pdm.jar

• Mitm attacker inject/replaces 600eb07a-2926-4407-b014-d3e8c77b0086.zip with following content:

```bash
unzip -l 600eb07a-2926-4407-b014-d3e8c77b0086.zip
```

<table>
<thead>
<tr>
<th>Length</th>
<th>Date</th>
<th>Time</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>2015-09-15</td>
<td>18:57</td>
<td>/data/data/com.kms.free/app_bases/pdm.jar</td>
</tr>
<tr>
<td>4042</td>
<td>2015-08-28</td>
<td>18:49</td>
<td>1000_768.css</td>
</tr>
<tr>
<td>6078</td>
<td>2015-08-28</td>
<td>18:49</td>
<td>AntiVirus_Premium.html</td>
</tr>
</tbody>
</table>
Summary of the Attack

- found unprotected communication
- augment a zip file with traversal file
- overwrite existing file with executable code
- app restart: injected code will be executed
- http-update-request
- advertisement archive
- delivered pdm.jar contains executable code
#Challenges

- ✔ Premium Upgrade for Free?
- ✔ Misuse Lost-Device Feature (Ransomware)?
- ✔ Remotely Influence Scan Engine Behavior?
- ✔ Remote Code Execution?
## Summary

<table>
<thead>
<tr>
<th></th>
<th>AndroHelm</th>
<th>Avira</th>
<th>CM</th>
<th>ESET</th>
<th>Kaspersky</th>
<th>McAfee</th>
<th>MB</th>
</tr>
</thead>
<tbody>
<tr>
<td>DOS</td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>Upgrade</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>Wipe/Lock</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>HTTP</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>Scan Engine</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>Tapjacking</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>RCE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>SSL Vuln</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>Broken Crypto</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>XSS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>x</td>
</tr>
</tbody>
</table>

[ sit4.me/av-advisories ]
Responsible Disclosure Fails

- 6/7 vendors fixed vulnerabilities
- Epic fails during RD
  - Expired public key
  - Certificate was not matching with email address
- Some did not reply - met them at a conference
Lessons learned...

- Big security companies also fail in implementing vulnerable-free apps
- Room for improvement in the RD process
- Vulnerabilities in mobile apps can be also found in the PC counterpart (research by Tavis Ormandy)
sit4.me/av-advisories

**Stephan Huber**
Email: stephan.huber@sit.fraunhofer.de

**Siegfried Rasthofer**
Email: siegfried.rasthofer@sit.fraunhofer.de

Twitter: @teamsik
Website: www.team-sik.org