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

Fraunhofer SIT

team [SIK]

Who are we

Stephan

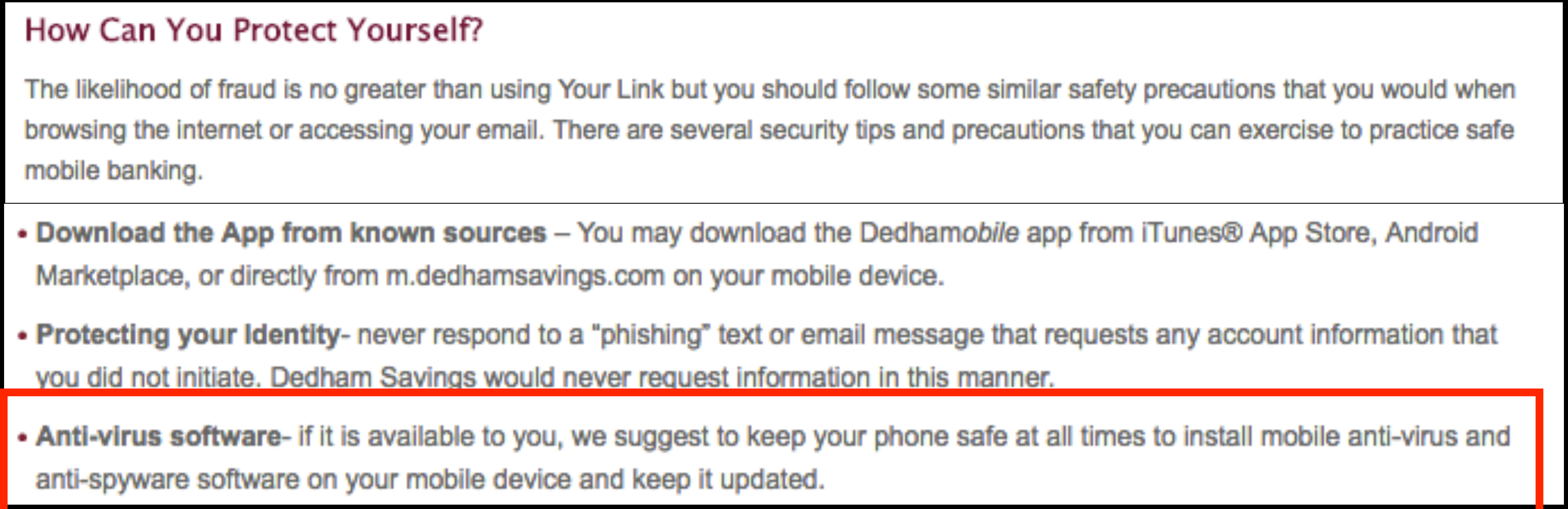
- Mobile Security Researcher at Fraunhofer SIT
- Enjoys teaching students in Android Hacking

Siegfried

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

team [SIK]

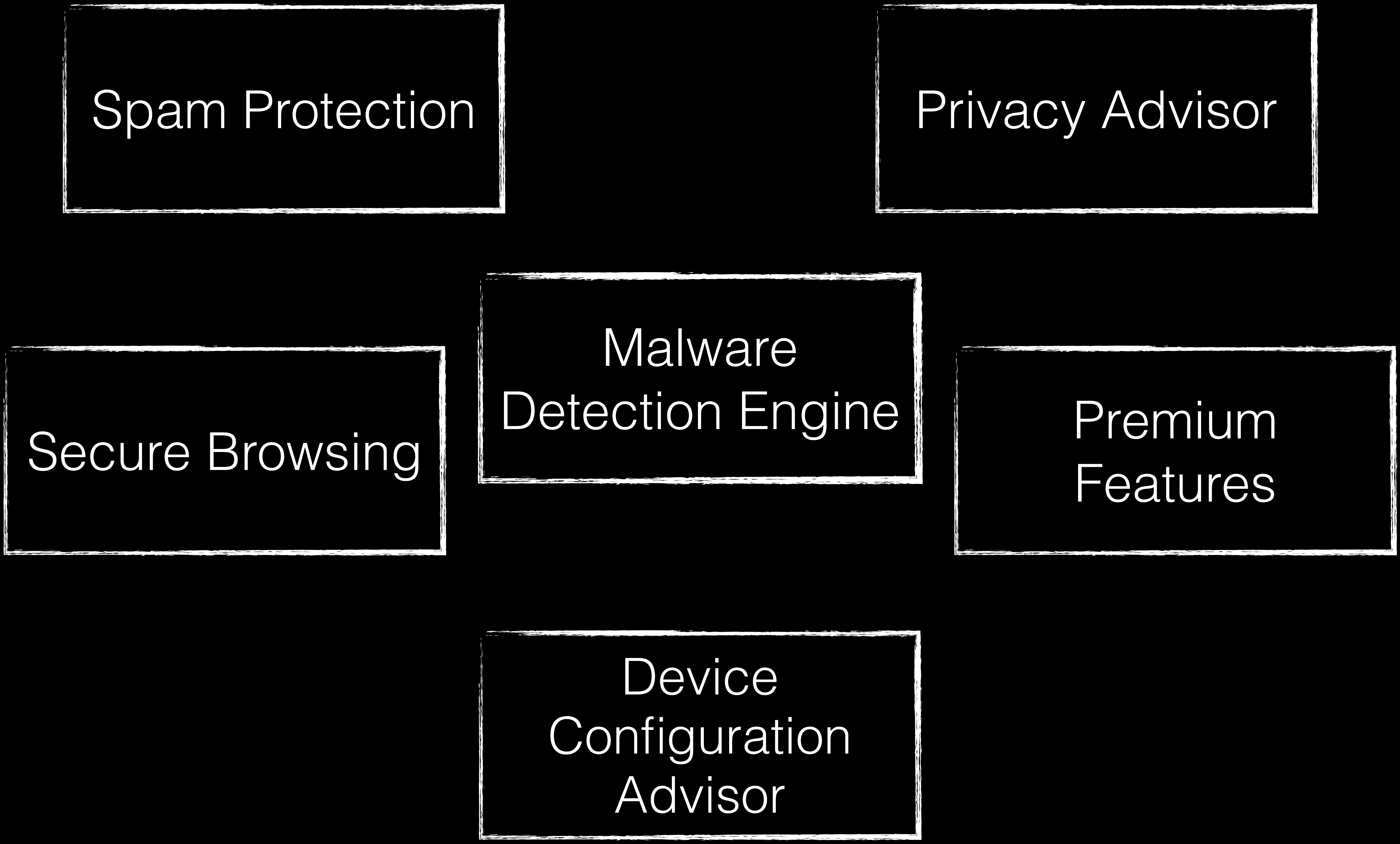
Mobile Banking Security



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 Dedham*obile* 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

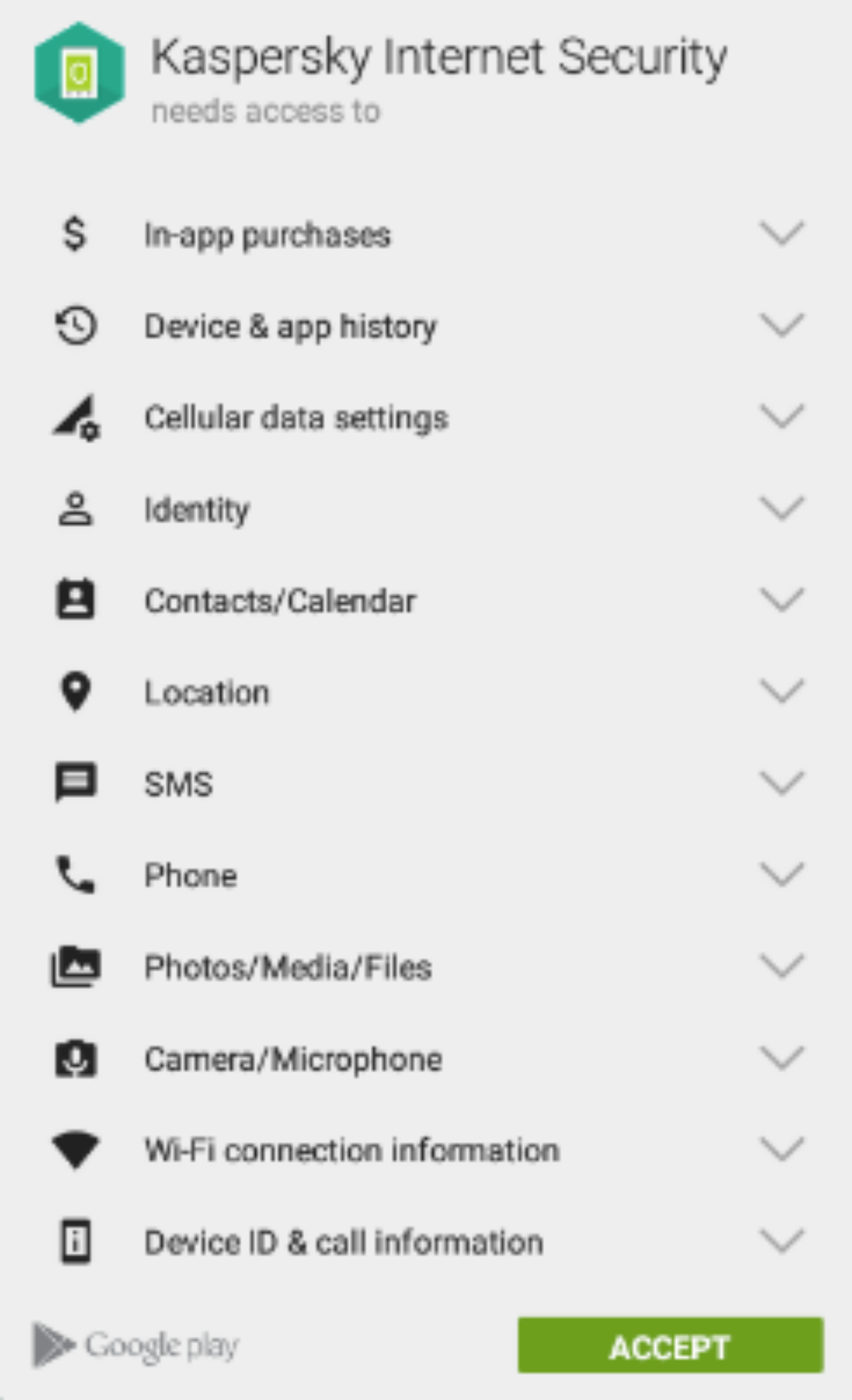
Privacy Advisor

Malware Detection Engine

Secure Browsing

Premium Features

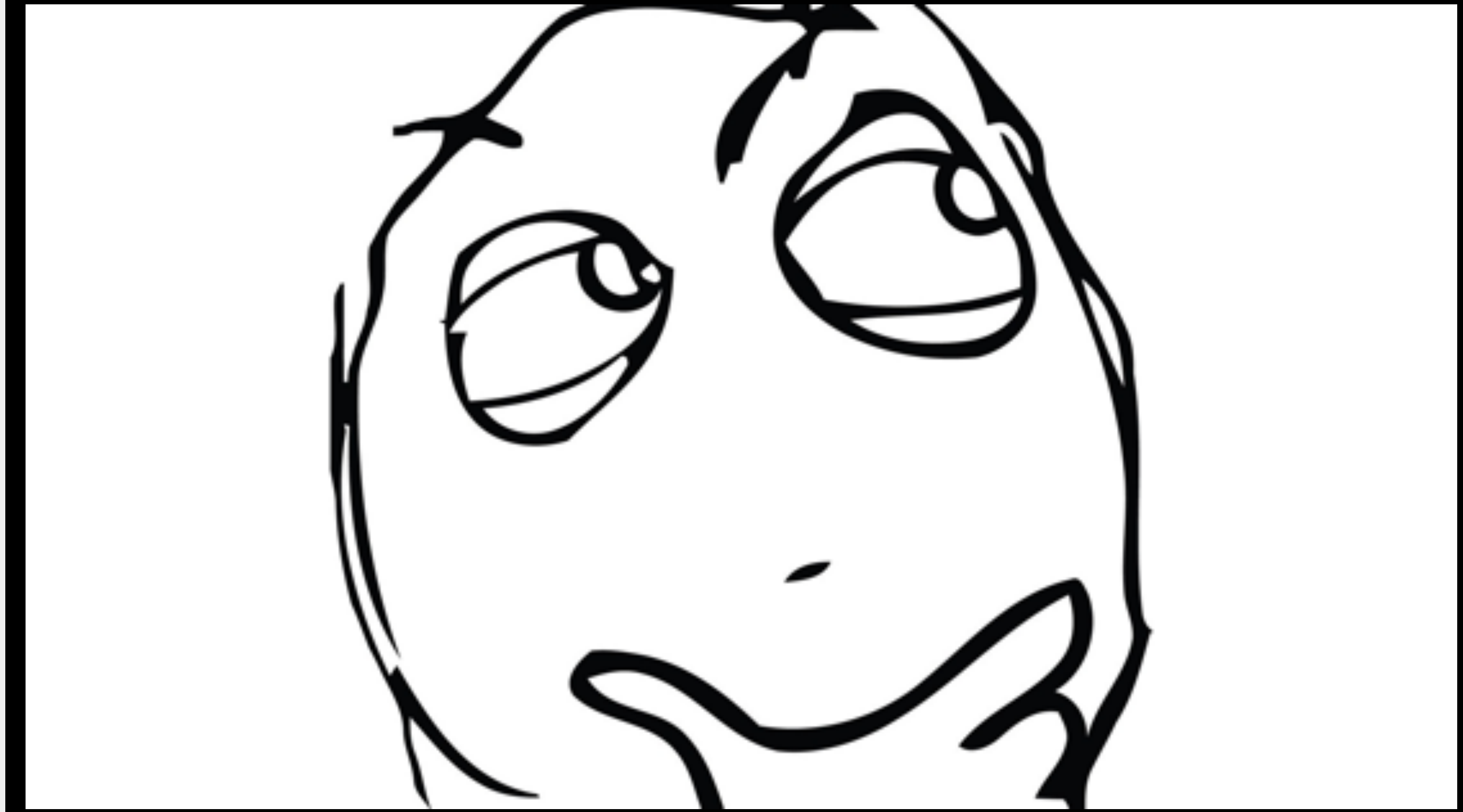
Device Configuration Advisor



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



App	GooglePlay Downloads
“Pseudo“ AV Apps	
AndroHelm	1-5 Mio
Malwarebytes	5-10 Mio
ESET	5-10 Mio
Avira	10-50 Mio
Kaspersky	10-50 Mio
McAfee	10-50 Mio
CM Security	100-500 Mio

#Challenges

□ Premium Upgrade for Free?

□ Misuse Lost-Device Feature (Ransomware)?

□ Remotely Influence Scan Engine Behavior?

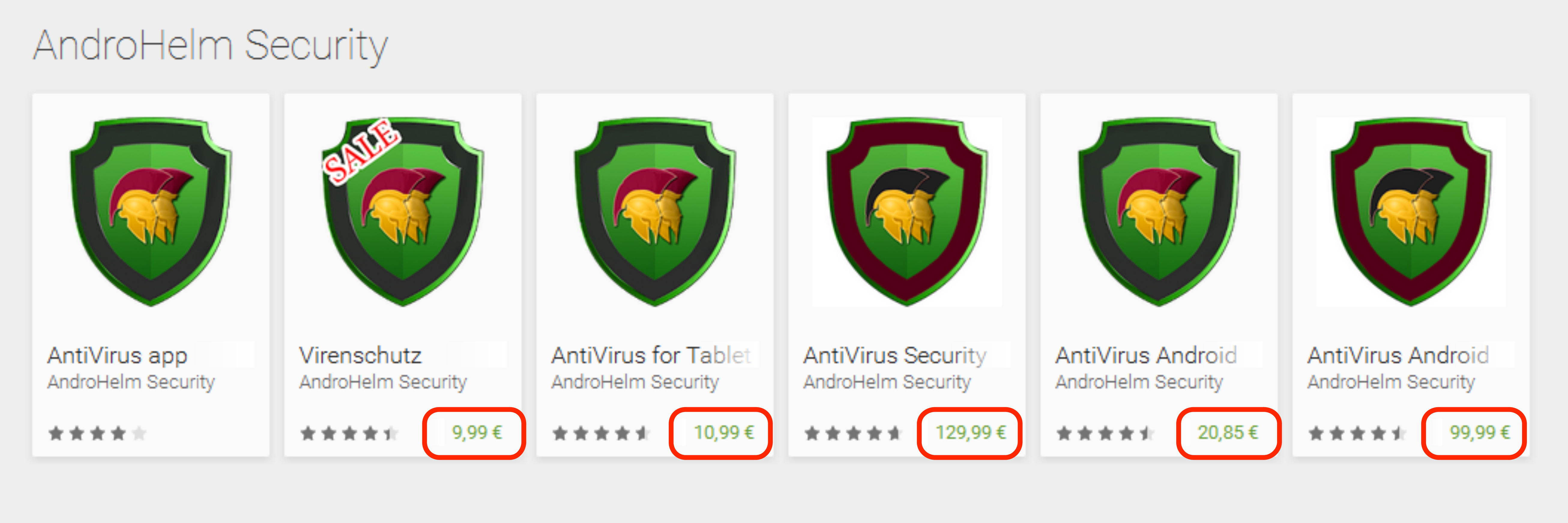
□ Remote Code Execution?

Premium Upgrade for Free?

(1/2 Examples)

AndroHelm

Free Premium the Simple Way



AndroHelm Security

AntiVirus app	AndroHelm Security	
Virenschutz	AndroHelm Security	9,99 €
AntiVirus for Tablet	AndroHelm Security	10,99 €
AntiVirus Security	AndroHelm Security	129,99 €
AntiVirus Android	AndroHelm Security	20,85 €
AntiVirus Android	AndroHelm Security	99,99 €

Let‘s Have a Look at the Free App

Interesting code snippet:

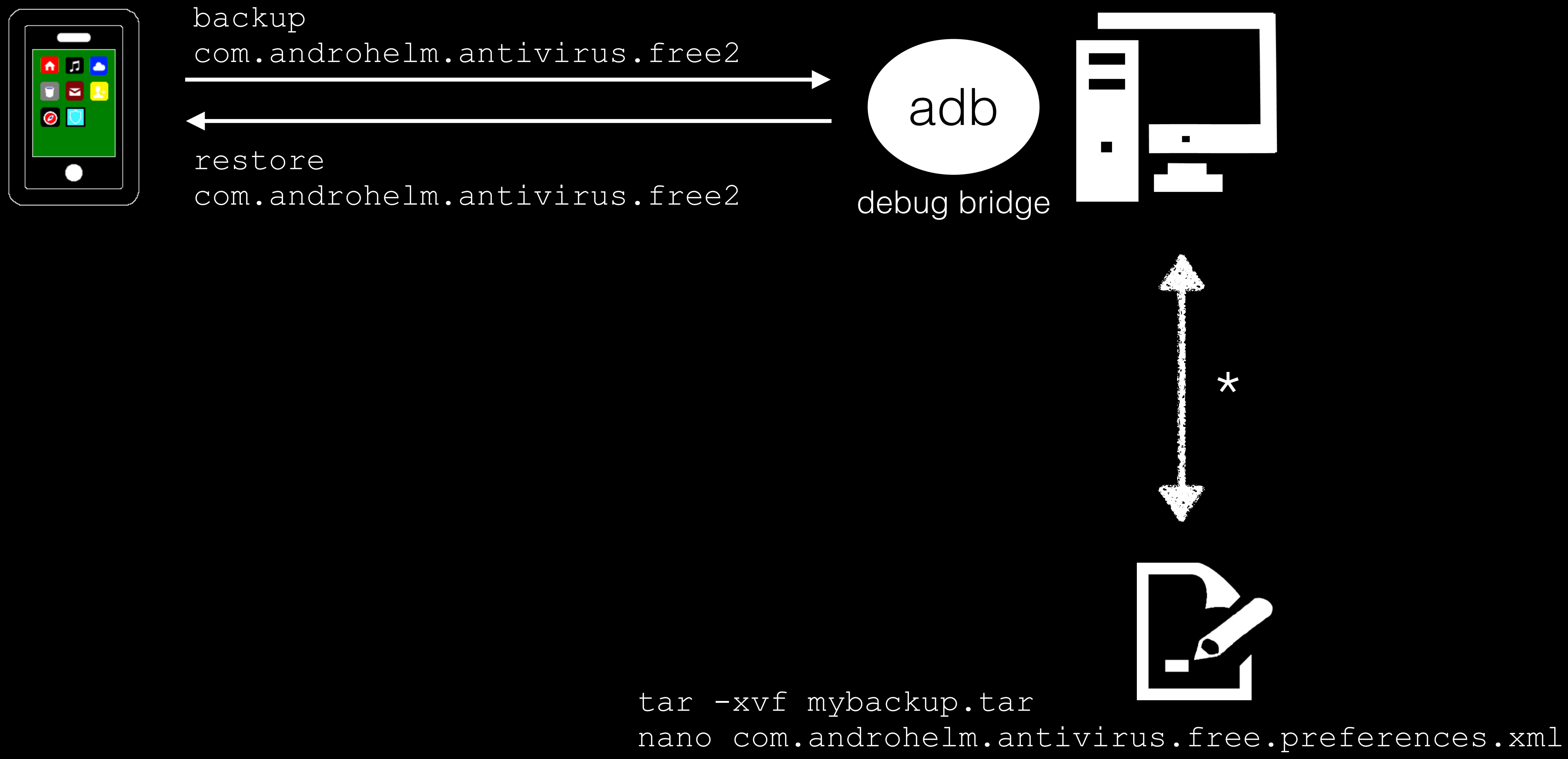
```
…
this.toast("Thank you for upgrading to PRO!");

//shared pref value set to true
this.prefs.putBoolean("isPro", true);
…
```

key/value pair for xml file

SharedPreferences at first install:

```
<?xml version='1.0' encoding='utf-8' standalone='yes' ?>
<map>
    <int name="dialogShowTimes" value="1" />
    <boolean name="hasDatabase" value="true" />
    <string name="lastFragment"></string>
    <boolean name="isPro" value="true" />
</map>
```

Changing XML File Without Root



```
backup
com.androhelm.antivirus.free2
```

```
restore
com.androhelm.antivirus.free2
```

adb

debug bridge

*

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

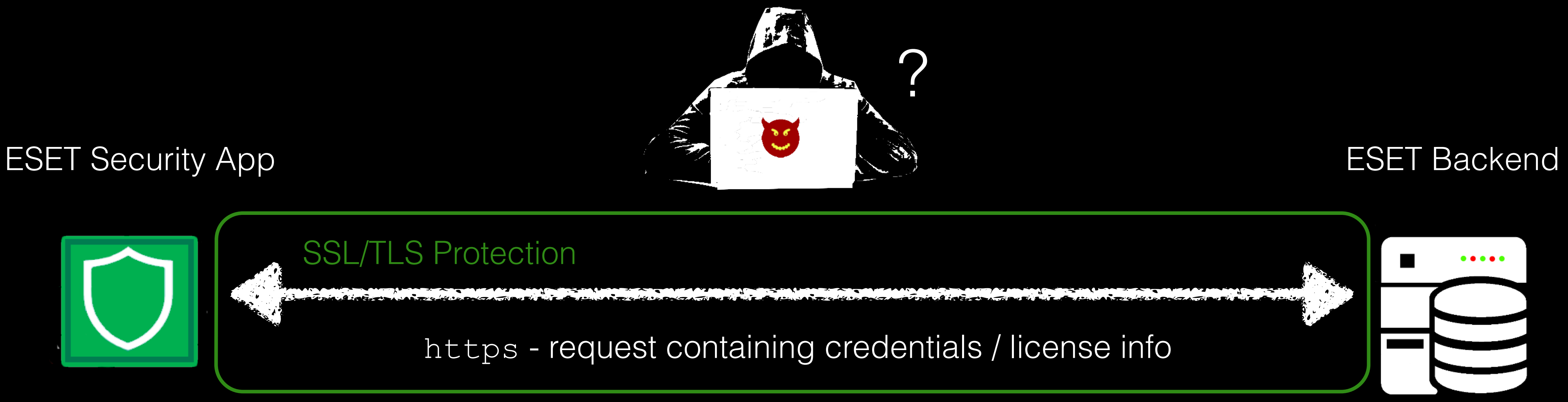
* https://github.com/nelenkov/android-backup-extractor

Premium Upgrade for Free?

(2/2 Examples)

ESET

ESET License Verification



ESET Security App

?

ESET Backend

SSL/TLS Protection

`https` - request containing credentials / license info

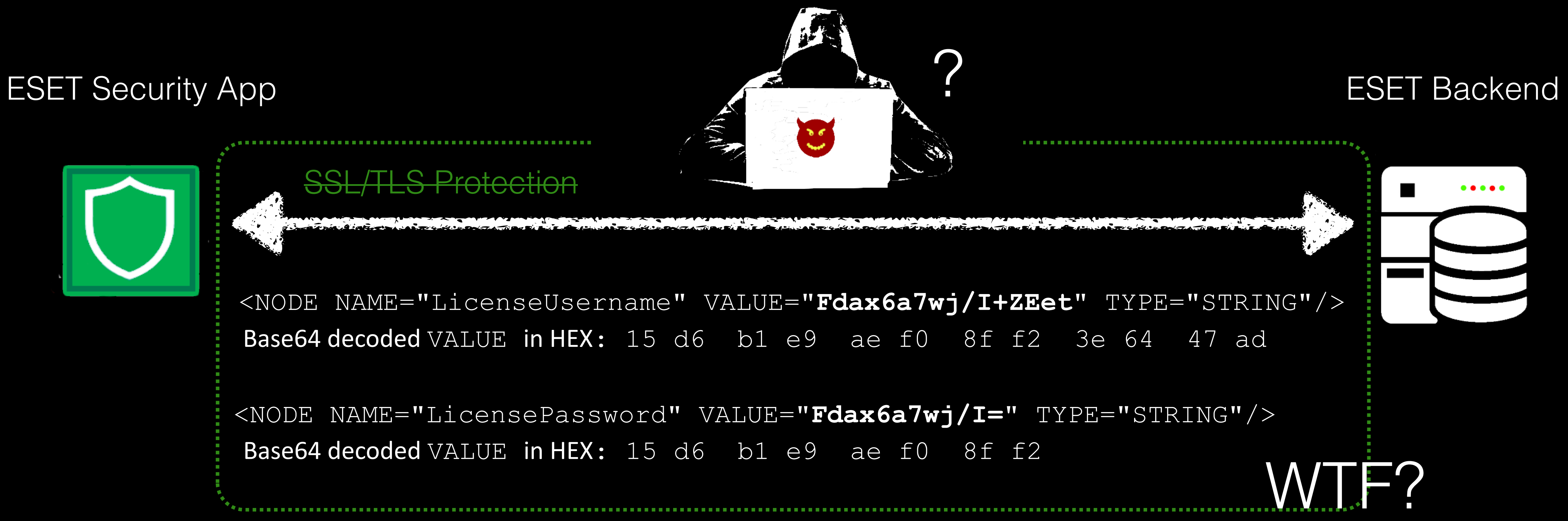
There are known vulnerabilities for SSL/TLS, but is there an ***easier*** way?

One requirement for secure communication is the verification of the SSL certificate!

```
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
```

BROKEN!

ESET License Verification



ESET Security App

?

ESET Backend

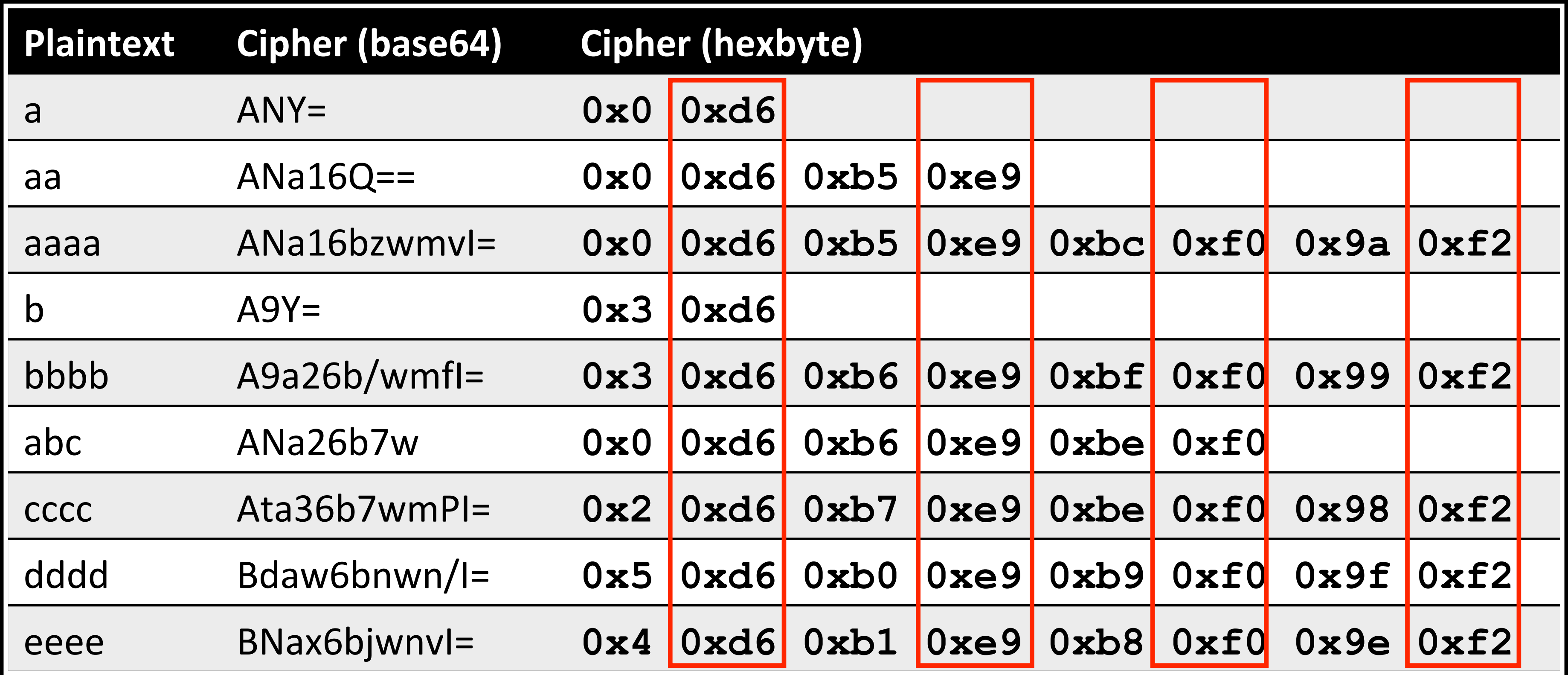
~~SSL/TLS Protection~~

```
<NODE NAME="LicenseUsername" VALUE="Fdax6a7wj/I+ZEet" TYPE="STRING"/>
Base64 decoded VALUE in HEX: 15 d6  b1 e9  ae f0  8f f2  3e 64  47 ad

<NODE NAME="LicensePassword" VALUE="Fdax6a7wj/I=" TYPE="STRING"/>
Base64 decoded VALUE in HEX: 15 d6  b1 e9  ae f0  8f f2
```

WTF?

Let’s do some Crypto Analysis

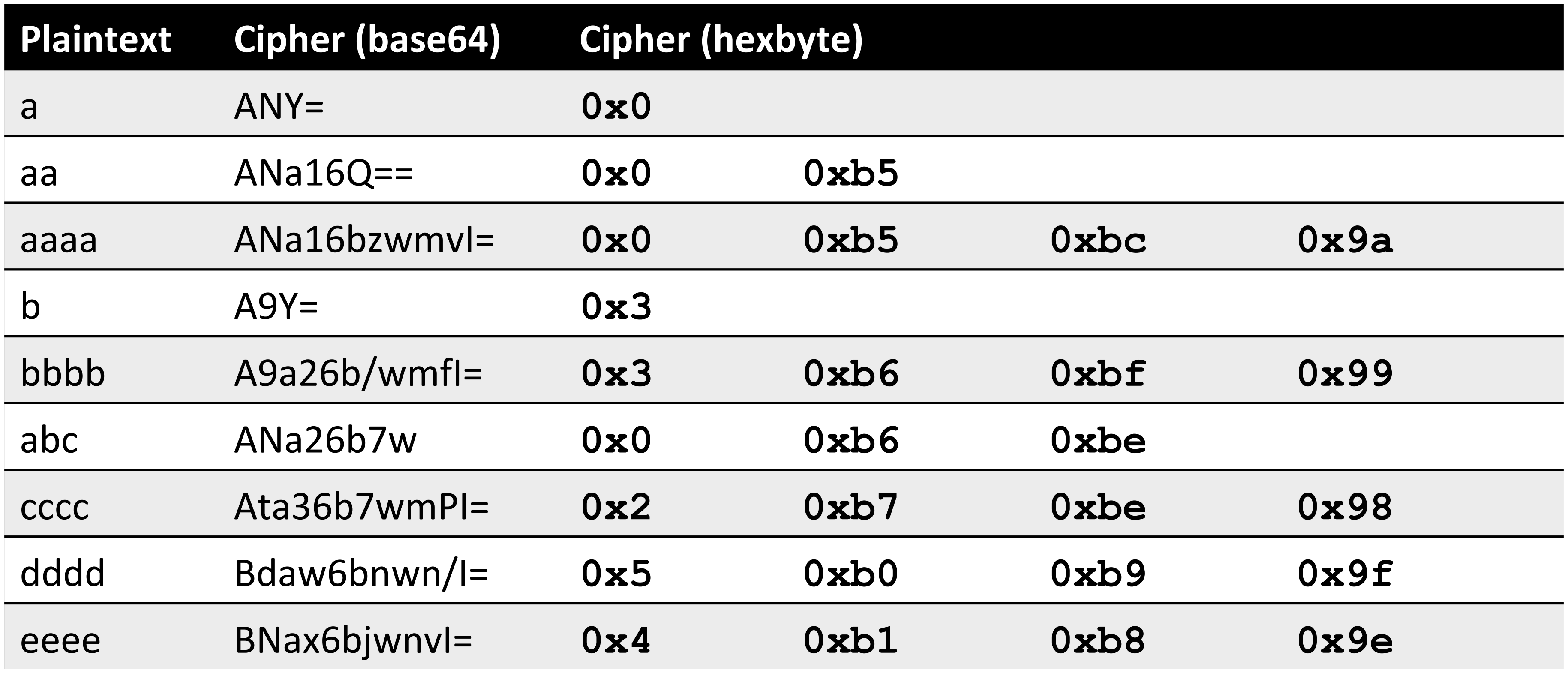
Classic chosen plaintext attack



Plaintext	Cipher (base64)	Cipher (hexbyte)							
a	ANY=	0x0	0xd6						
aa	ANa16Q==	0x0	0xd6	0xb5	0xe9				
aaaa	ANa16bzwmvI=	0x0	0xd6	0xb5	0xe9	0xbc	0xf0	0x9a	0xf2
b	A9Y=	0x3	0xd6						
bbbb	A9a26b/wmfI=	0x3	0xd6	0xb6	0xe9	0xbf	0xf0	0x99	0xf2
abc	ANa26b7w	0x0	0xd6	0xb6	0xe9	0xbe	0xf0		
cccc	Ata36b7wmPI=	0x2	0xd6	0xb7	0xe9	0xbe	0xf0	0x98	0xf2
dddd	Bdaw6bnwn/I=	0x5	0xd6	0xb0	0xe9	0xb9	0xf0	0x9f	0xf2
eeee	BNax6bjwnvI=	0x4	0xd6	0xb1	0xe9	0xb8	0xf0	0x9e	0xf2

Let’s do some Crypto Analysis

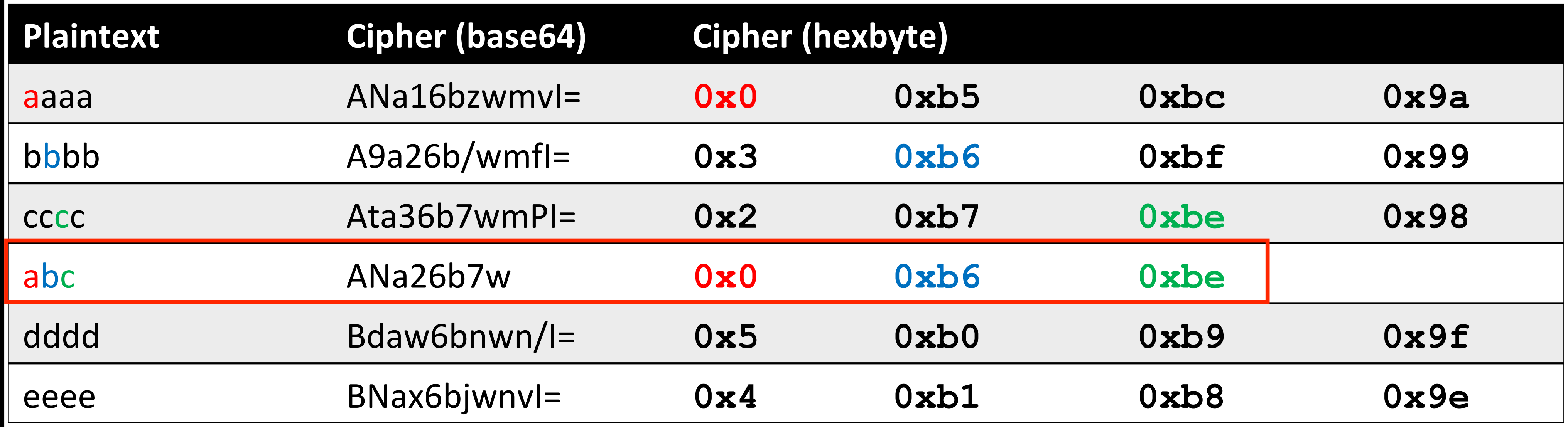
Classic chosen plaintext attack



Plaintext	Cipher (base64)	Cipher (hexbyte)			
a	ANY=	0x0			
aa	ANa16Q==	0x0	0xb5		
aaaa	ANa16bzwmvI=	0x0	0xb5	0xbc	0x9a
b	A9Y=	0x3			
bbbb	A9a26b/wmfI=	0x3	0xb6	0xbf	0x99
abc	ANa26b7w	0x0	0xb6	0xbe	
cccc	Ata36b7wmPI=	0x2	0xb7	0xbe	0x98
dddd	Bdaw6bnwn/I=	0x5	0xb0	0xb9	0x9f
eeee	BNax6bjwnvI=	0x4	0xb1	0xb8	0x9e

Let’s do some Crypto Analysis

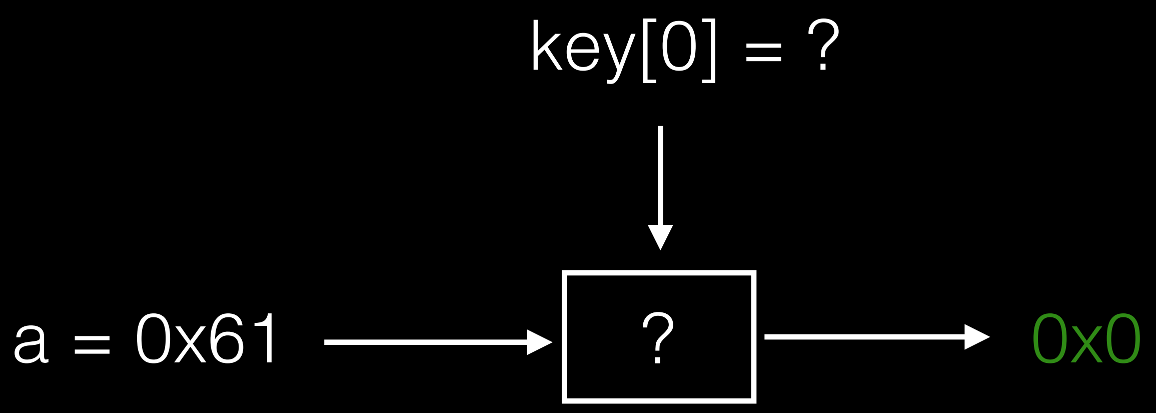
Clean up:



Plaintext	Cipher (base64)	Cipher (hexbyte)			
aaaa	ANa16bzwmvI=	0x0	0xb5	0xbc	0x9a
bbbb	A9a26b/wmfI=	0x3	0xb6	0xbf	0x99
cccc	Ata36b7wmPI=	0x2	0xb7	0xbe	0x98
abc	ANa26b7w	0x0	0xb6	0xbe	
dddd	Bdaw6bnwn/I=	0x5	0xb0	0xb9	0x9f
eeee	BNax6bjwnvI=	0x4	0xb1	0xb8	0x9e

- 2nd byte is not required
- No chaining
- Looks like a simple substitution

Here Comes the Key



key[0] = ?

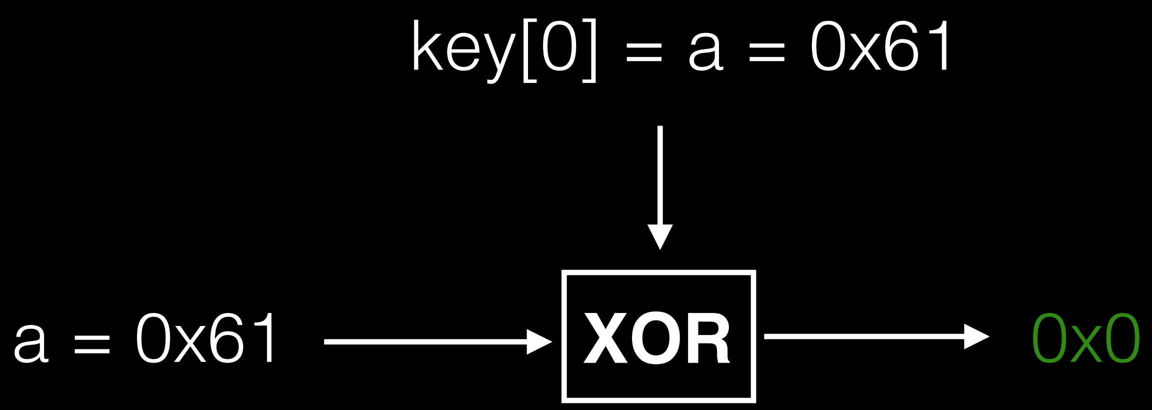
a = 0x61

?

0x0

Letter	Decimal	Hex	1. Cipher
a	97	0x61	0x0
b	98	0x62	0x3
c	99	0x63	0x2

Here Comes the Key



key[0] = a = 0x61

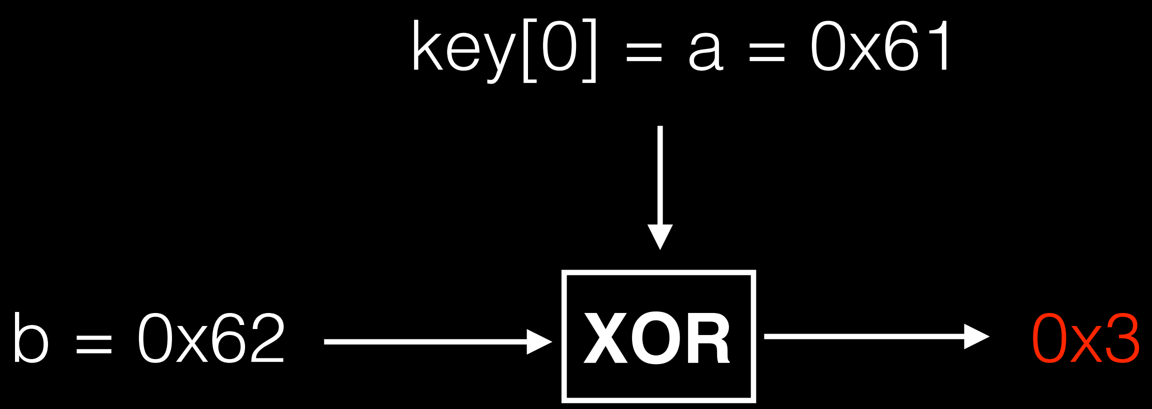
a = 0x61

XOR

0x0

Letter	Decimal	Hex	1. Cipher
a	97	0x61	0x0
b	98	0x62	0x3
c	99	0x63	0x2

Here Comes the Key



key[0] = a = 0x61

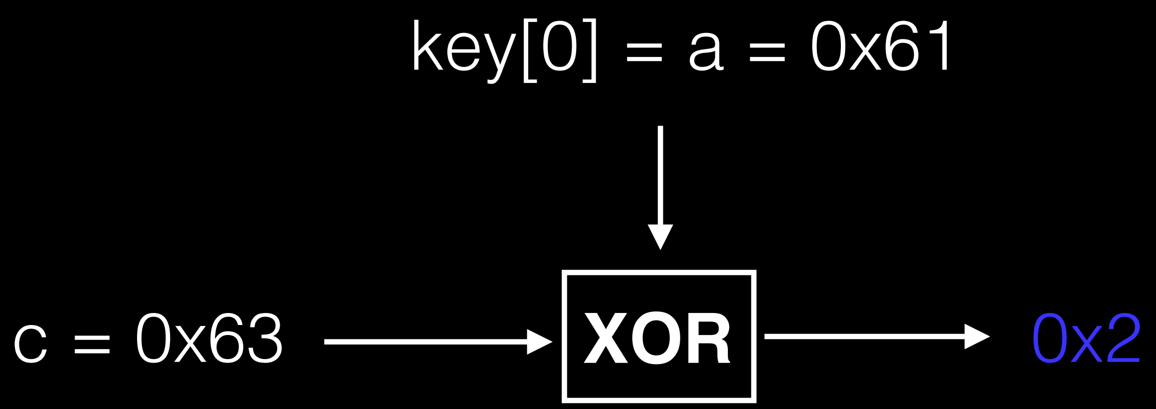
b = 0x62

XOR

0x3

Letter	Decimal	Hex	1. Cipher
a	97	0x61	0x0
b	98	0x62	0x3
c	99	0x63	0x2

Here Comes the Key



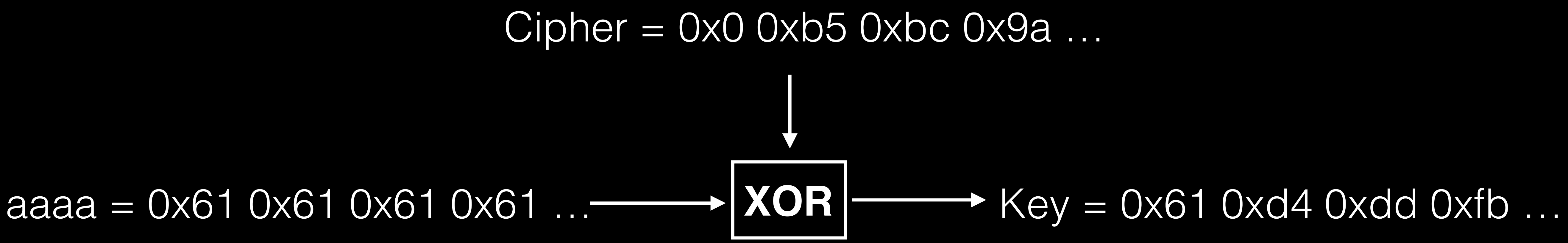
key[0] = a = 0x61

c = 0x63

0x2

Letter	Decimal	Hex	1. Cipher
a	97	0x61	0x0
b	98	0x62	0x3
c	99	0x63	0x2

Here Comes the Key

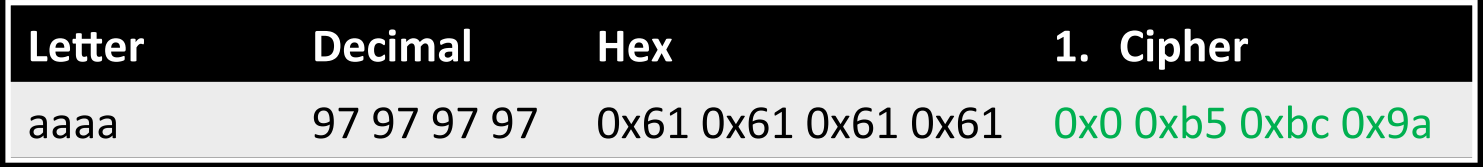


Cipher = 0x0 0xb5 0xbc 0x9a …

aaaa = 0x61 0x61 0x61 0x61 …

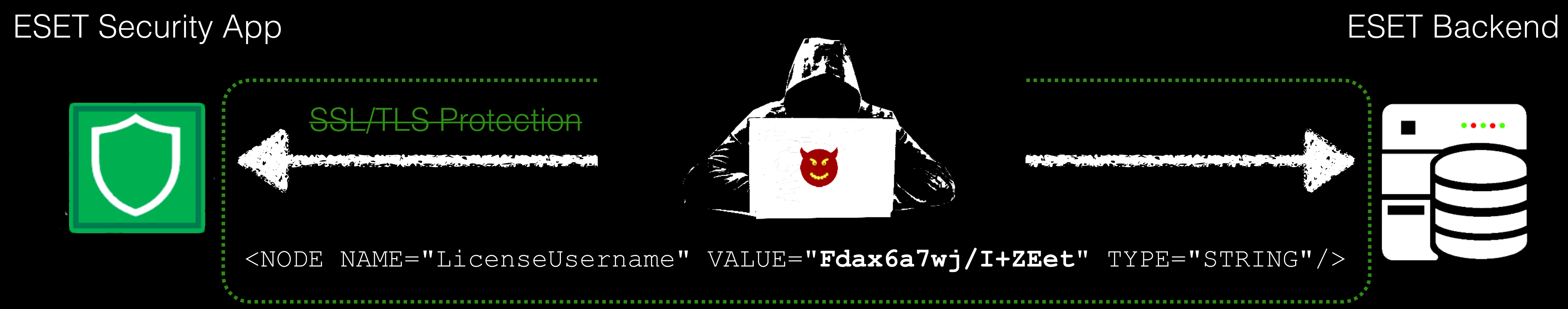
XOR

Key = 0x61 0xd4 0xdd 0xfb …



Letter	Decimal	Hex	1. Cipher
aaaa	97 97 97 97	0x61 0x61 0x61 0x61	0x0 0xb5 0xbc 0x9a

ESET License Verification



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

key = [0x61 0xd4 0xdd 0xfb 0x5b 0x35 0xb7 0x19 0xec 0x2b 0x42 0xd9 0x4b 0x7 …]

$\otimes$

`Fdax6a7wj/I+ZEet`

↓

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

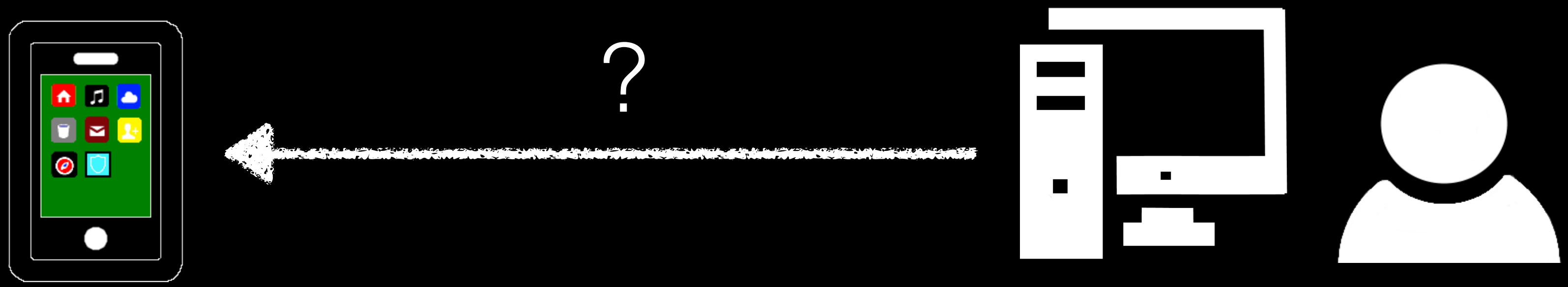
What is a lost-device feature?

- Device Location
- Remote Alarm
- Remote Wipe
- Remote Lock
- …

Kartendaten © 2016 GeoBasis-DE/BKG (©2009),Google

Can we abuse “Remote Lock“ or “Wipe“?

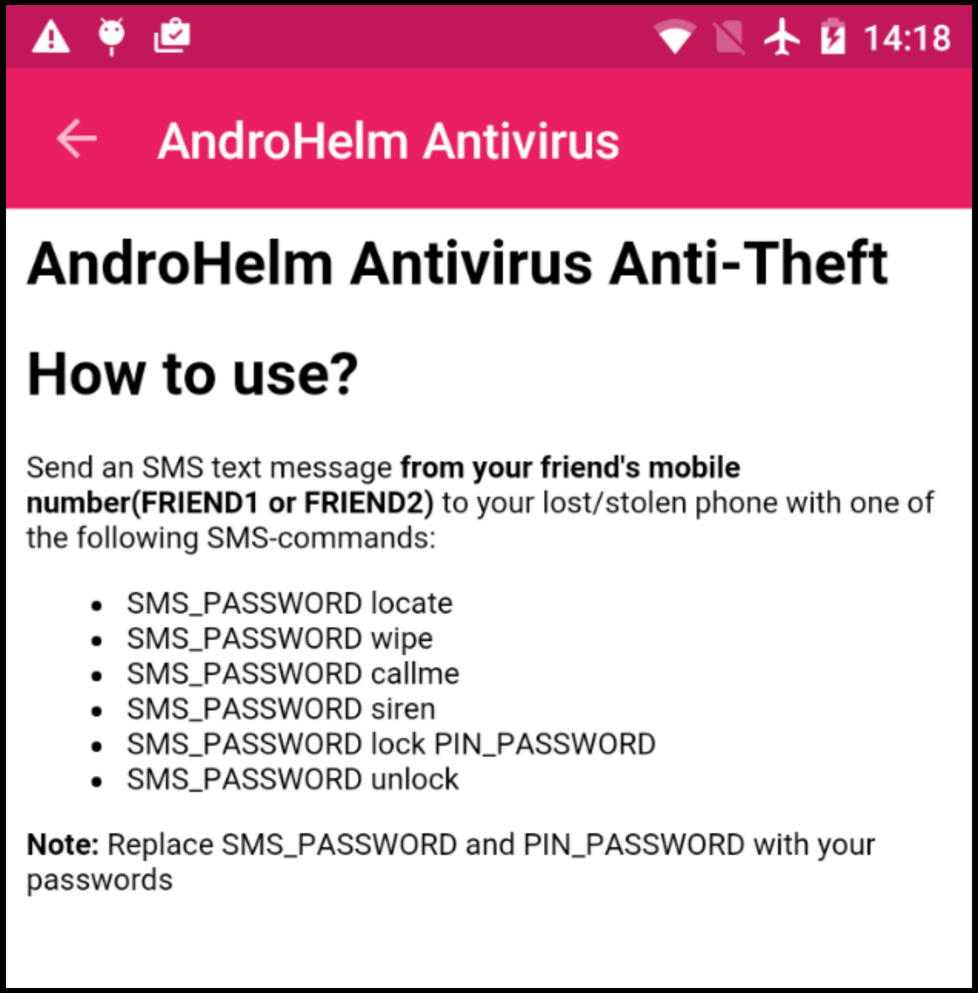
Remote Communication With Smartphone



Examples:

- Google Cloud Messaging (GCM)
- Push Service Provider
- **SMS Messages**

Androhelm Anti-Theft SMS Protocol



AndroHelm Antivirus Anti-Theft

How to use?

Send an SMS text message **from your friend's mobile number(FRIEND1 or FRIEND2)** to your lost/stolen phone with one of the following SMS-commands:

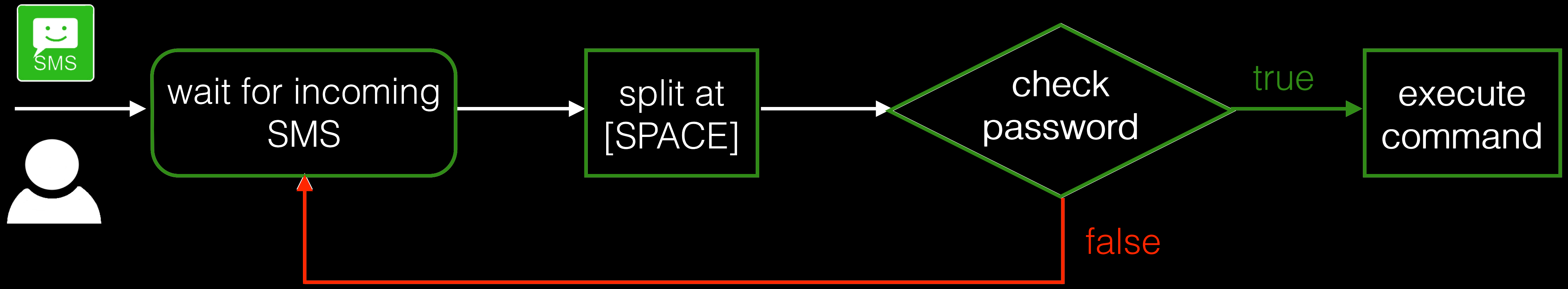
- SMS_PASSWORD locate
- SMS_PASSWORD wipe
- SMS_PASSWORD callme
- SMS_PASSWORD siren
- SMS_PASSWORD lock PIN_PASSWORD
- SMS_PASSWORD unlock

Note: Replace SMS_PASSWORD and PIN_PASSWORD with your passwords

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

Feature not enabled, still possible to bypass the authentication?

Remote Protocol with Activated Anti-Theft



SMS

wait for incoming SMS

split at [SPACE]

check password

true

execute command

false

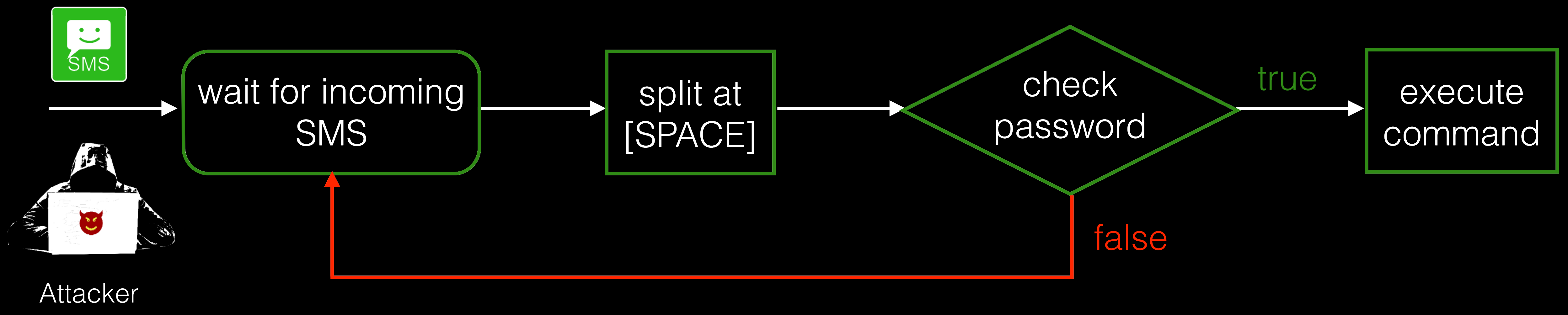
```
myPass[SPACE]wipe[SPACE]
```

```
SMS_PASSWORD := “myPass“
command := “wipe“
```

```
//Stored password
pwd := “myPass“
pwd == SMS_PASSWORD?
“myPass“ == “myPass“
```

```
command := “wipe“
execute(command)
```

Remote Protocol Deactivated Anti-Theft



```
[SPACE]wipe[SPACE]somestring
```

empty string
as pwd

```
SMS_PASSWORD := ““
command := “wipe“
```

SMS_PASSWORD
is empty

```
//default password
pwd := ““
pwd == SMS_PASSWORD?
 ““ == ““
```

```
command := “wipe“
execute(command)
```

#Challenges

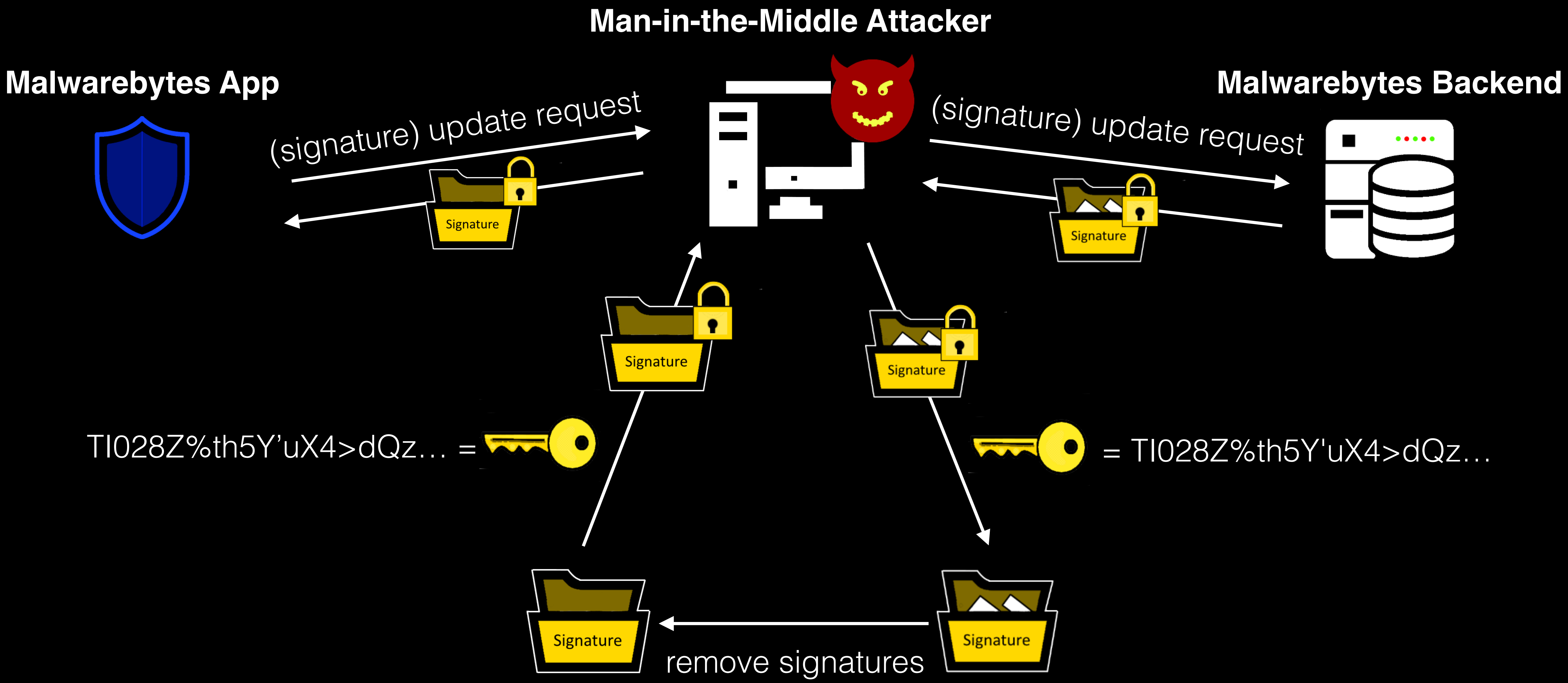
- ☑ Premium Upgrade for Free?
- ☑ Misuse Lost-Device Feature (Ransomware)?
- □ Remotely Influence Scan Engine Behavior?
- □ Remote Code Execution?

Remotely Influence Scan Engine Behavior?

(1 Example)

Malwarebytes

Unprotected Signature Updates



Man-in-the-Middle Attacker

Malwarebytes App

Malwarebytes Backend

(signature) update request

(signature) update request

TI028Z%th5Y’uX4>dQz… =

= TI028Z%th5Y'uX4>dQz…

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?

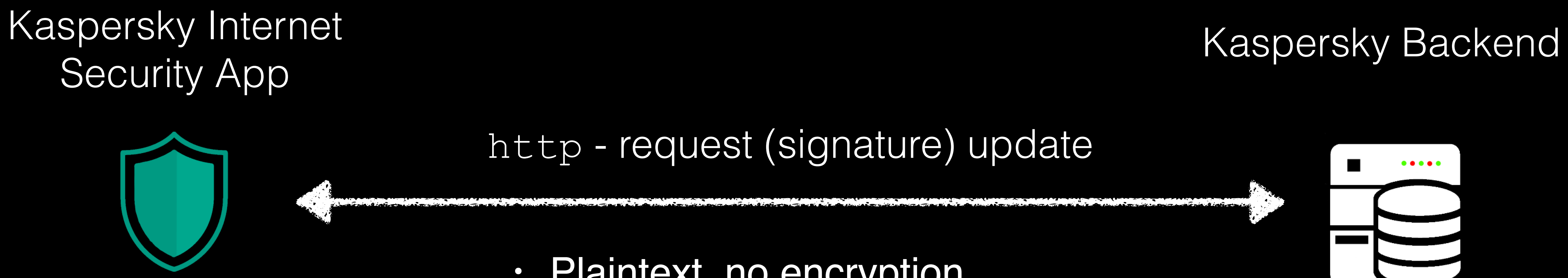
```
/tmp$ unzip zipfile.zip -d ./dir1/
      Archive:  zipfile.zip
      warning:  skipped "../" path component(s) in ../../../tmp/dir2/badfile.txt
       extracting: ./dir1/tmp/dir2/badfile.txt
       extracting: ./dir1/file1.txt

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

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



Kaspersky Internet Security App

`http` - request (signature) update

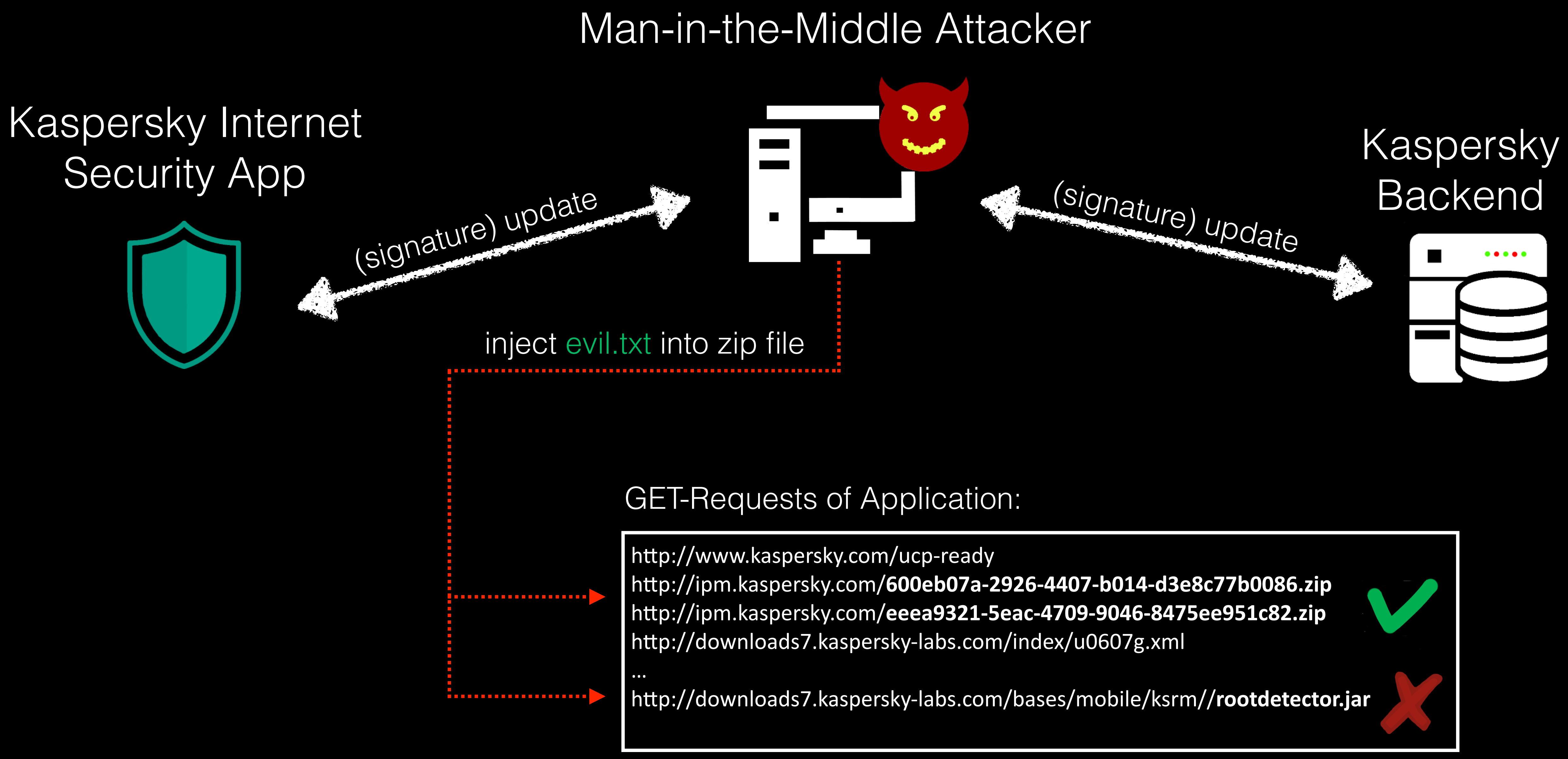
Kaspersky Backend

- 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

(signature) update

(signature) update

inject evil.txt into zip file

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
…
http://downloads7.kaspersky-labs.com/bases/mobile/ksrm//**rootdetector.jar** ✗

Finding Attack Vector

App’s folder containing executables

```
./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/evil.txt
```

included in apk file
contains classes.dex

signed, can not be manipulated!!

content of our zip archive

injected file

Finding Attack Vector

App's folder

PATH TRAVERSAL!

```
./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?

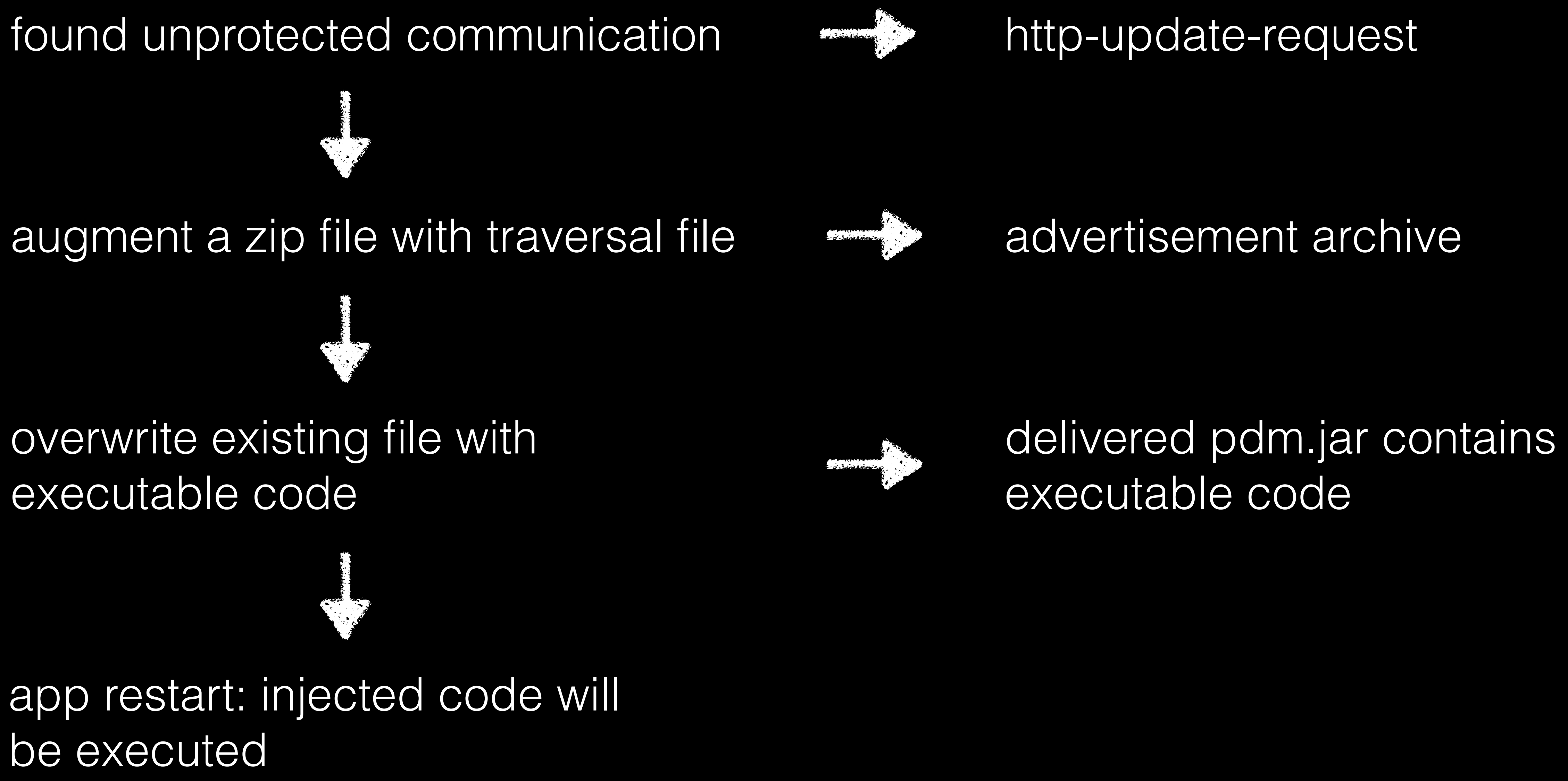
another injected file

The Exploit

- Overwrite original pdm.jar with manipulated pdm.jar
- Mitm attacker inject/replaces 600eb07a-2926-4407-b014-d3e8c77b0086.zip with following content:

```
unzip -l 600eb07a-2926-4407-b014-d3e8c77b0086.zip
Archive:  600eb07a-2926-4407-b014-d3e8c77b0086.zip
  Length      Date    Time    Name
---------  ---------- -----   ----
       16  2015-09-15 18:57   ../../../../../../../../../../../../../
   ../../../../../../../../../data/data/com.kms.free/app_bases/pdm.jar
    4042  2015-08-28 18:49   1000_768.css
    6078  2015-08-28 18:49   AntiVirus_Premium.html
```

Summary of the Attack



found unprotected communication → http-update-request

augment a zip file with traversal file → advertisement archive

overwrite existing file with executable code → delivered pdm.jar contains executable code

app restart: injected code will be executed

#Challenges

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

Summary

	AndroHelm	Avira	CM	ESET	Kaspersky	McAfee	MB
DOS	x	x				x	
Upgrade	x			x			
Wipe/Lock	x						
HTTP		x	x		x		x
Scan Engine		x	x				
Tapjacking			x				
RCE			x		x		
SSL Vuln				x			
Broken Crypto				x			x
XSS						x	

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

Lessens 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