Reversing Android Apps

Hacking and cracking Android apps is easy

Tobias Ospelt

DREAMLAB
TECHNOLOGIES

Agenda

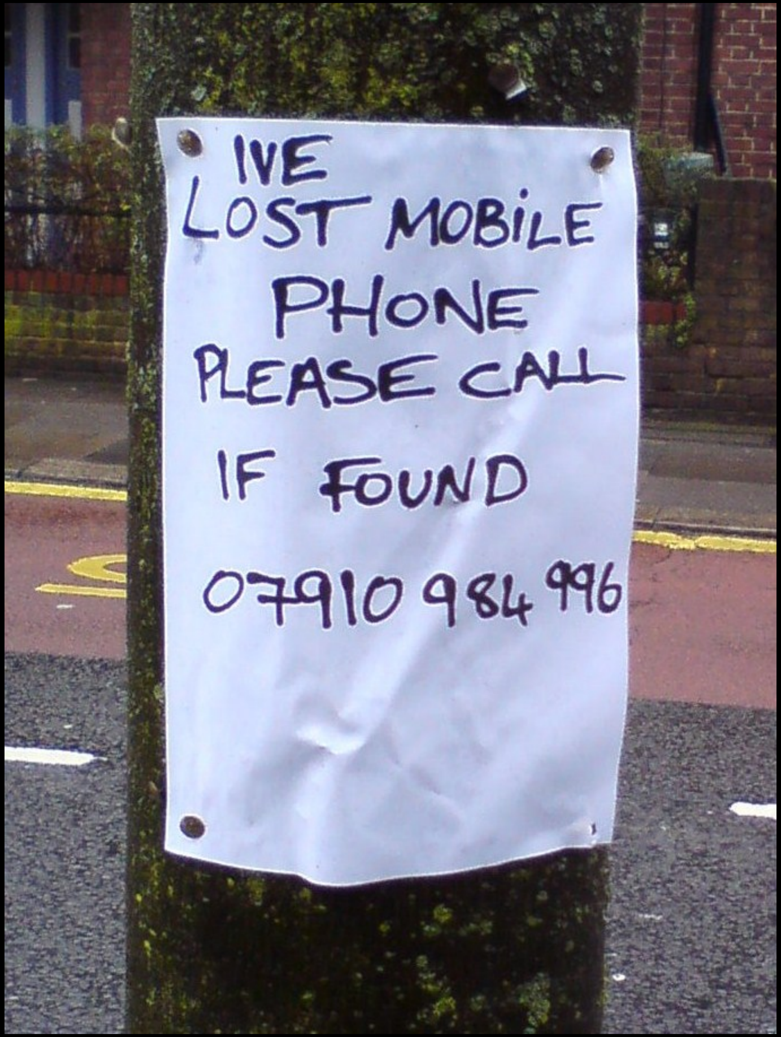
- Issues (in the past)
- Android security / code concept
- Techniques for pentesters / reverse engineers
- My experiences and the general quality of apps

My approach

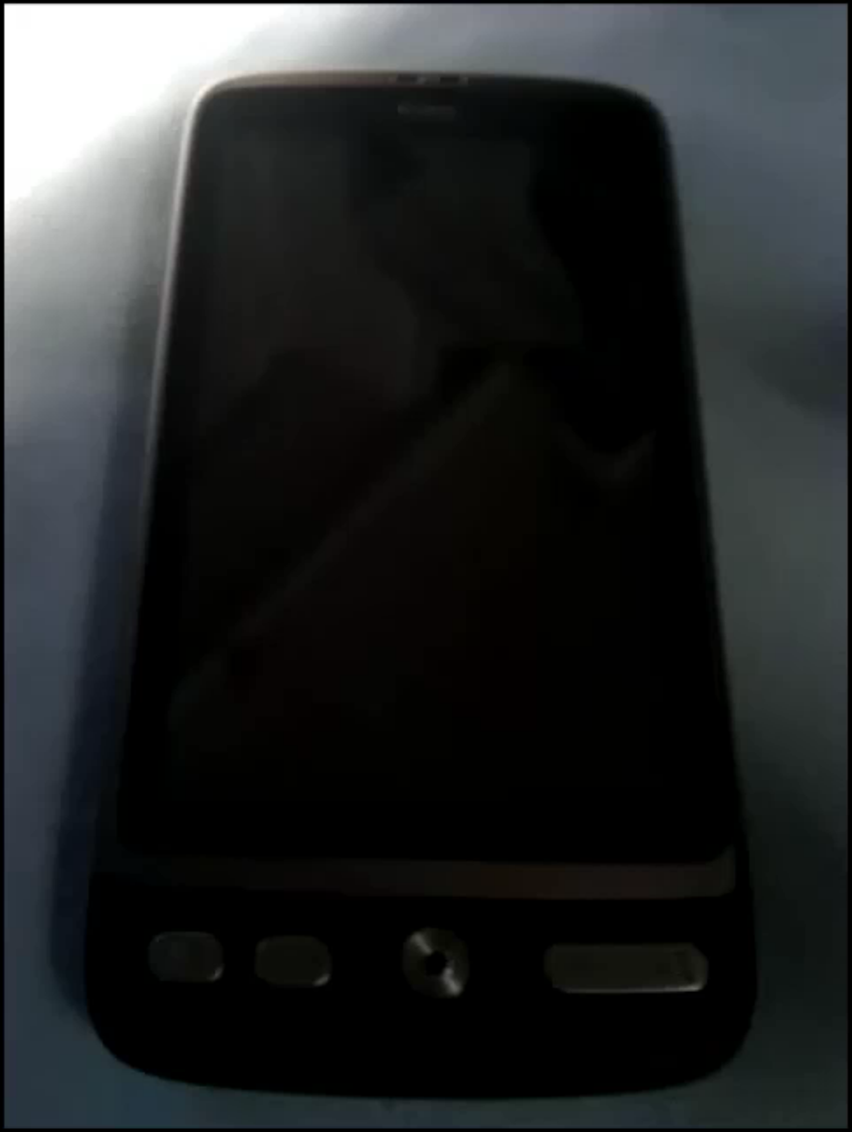
- Bought HTC Desire/Bravo with Android 2.0 (now 2.2.0) in 2010
- Finding security related issues

Issues (in the past?)

Losing phones



Circumventing lock screen



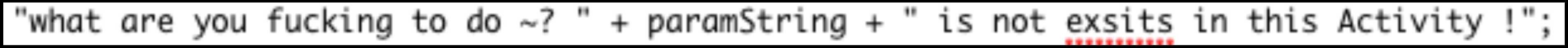
Circumventing lock screen

- Poor lock screen implementation
  - Home button mashing, not all brands<= 2.2
  - Back button during call, not all brands <= 2.0
  - Plug into car dock, unknown
  - Gmail address & password „null“, unknown
- Lock screen not activated
- USB debug on (adb shell)
- Associated Google account
- OpenRecovery, Milestone <= 2.1
- Acquire physical memory (forensic tools)

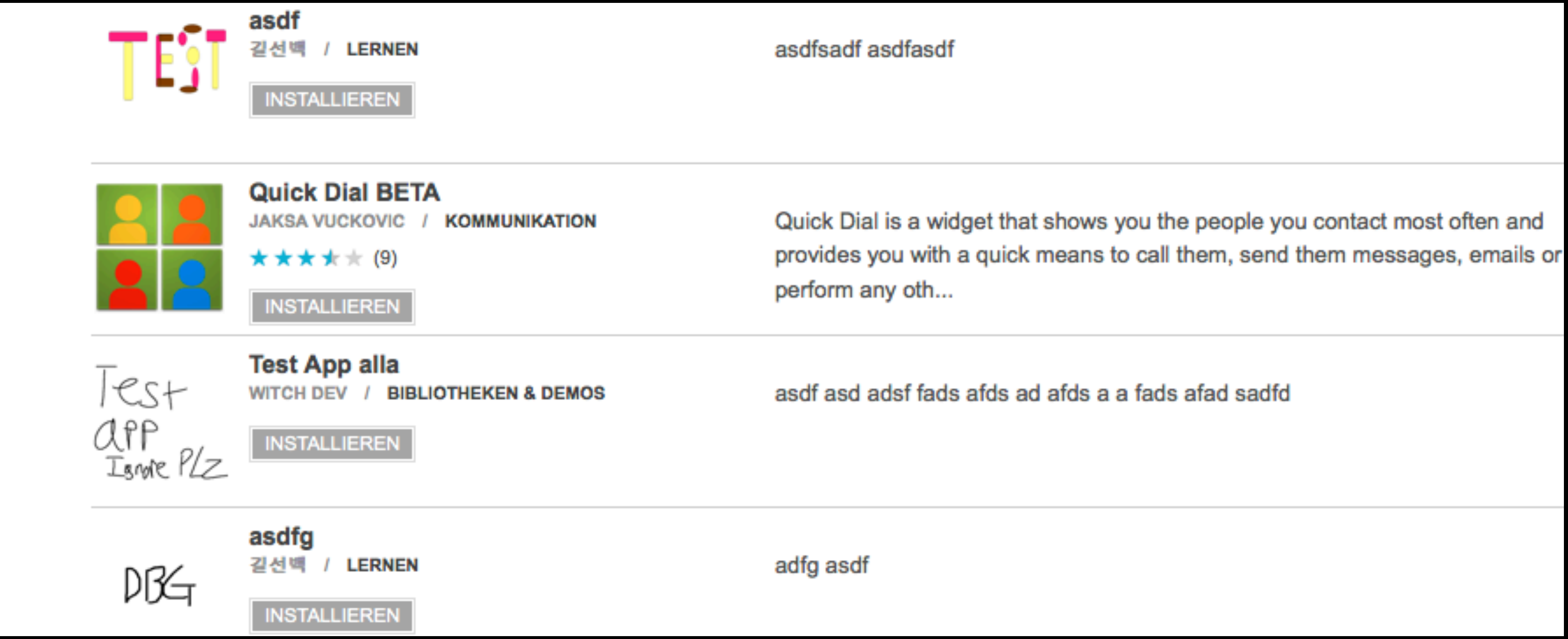
Android or Google?

- Android is Open Source
  - Google is the strong force behind it
- Google Market is not (it‘s Google‘s)
- You can create your own market

Google Market – a feel free environment



```
"what are you fucking to do ~? " + paramString + " is not exsits in this Activity !";
```



asdf
길선백 / LERNEN
INSTALLIEREN
asdfsadf asdfasdf

Quick Dial BETA
JAKSA VUCKOVIC / KOMMUNIKATION
★★★★☆ (9)
INSTALLIEREN
Quick Dial is a widget that shows you the people you contact most often and provides you with a quick means to call them, send them messages, emails or perform any oth...

Test App alla
WITCH DEV / BIBLIOTHEKEN & DEMOS
INSTALLIEREN
asdf asd adsf fads afds ad afds a a fads afad sadfd

asdfg
길선백 / LERNEN
INSTALLIEREN
adfg asdf

Malware

- Malware in the Google Market
  - DroidDream aka Rootcager
- Other malware (often in Chinese markets)
  - Bgserv, Pjabbs, Geinimi, FakePlayer, GingerMaster, Zeus, SpyEye

Bring malware to the mobile

- Convince users (aka put on market)
- XSS on Google Market website
- App without permissions installs apps with permissions
  - Angry Birds extra level malware, fixed
  - Browser vulnerability (cookie stealing), < 2.3.5
  - New technique going to be released in November
    - Oberheide/Lanie, Source Barcelona

Android Browser

- Puts nice little bookmark pics on your SD card

Übersicht

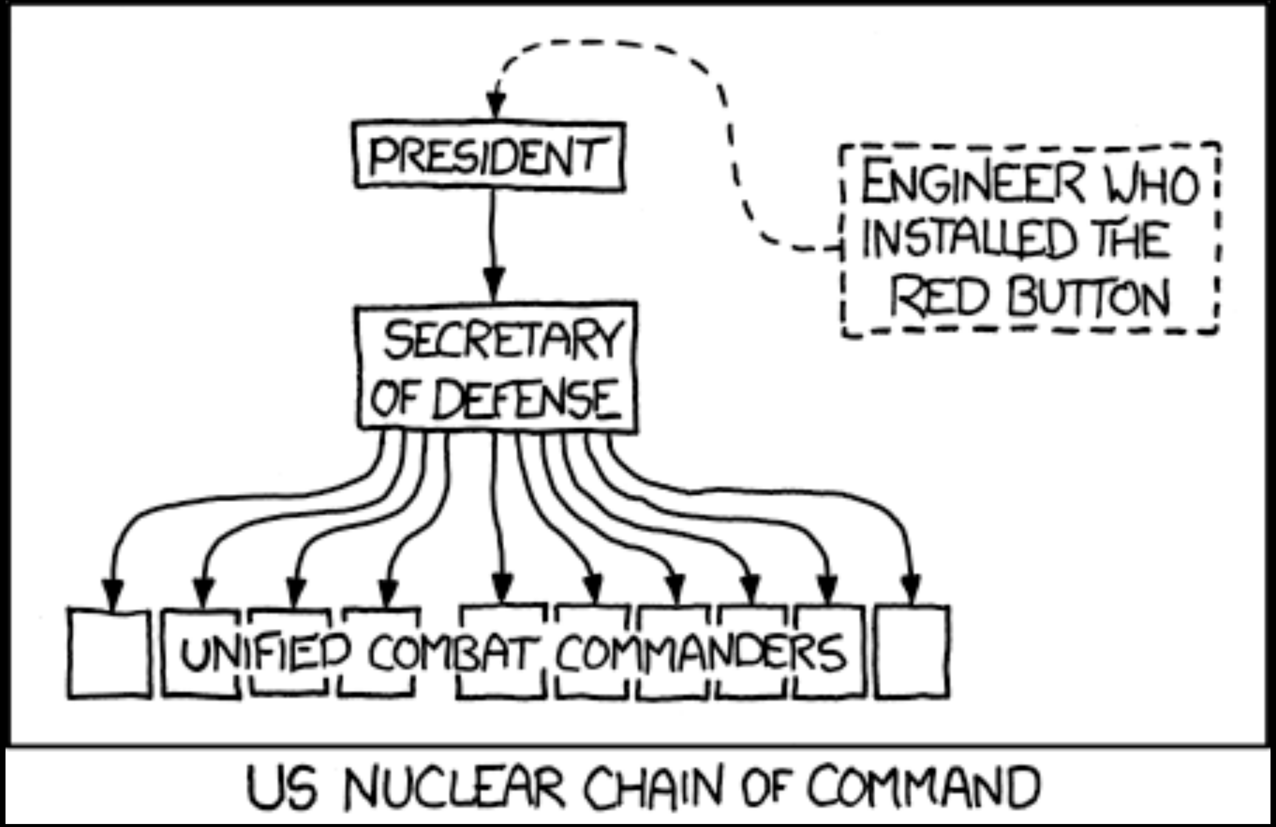
Von:	Winterthur	Datum:	Mi, 22.06.11
Nach:	Bern	Zeit:	07:13 (Ab)

Ab	An	Dauer	Umst.	Belegung	Gleis
07:20	08:44	1:24	1	1. 2.	3
07:28	08:58	1:30	0	1. 2.	5
07:30	08:58	1:28	1	1. 2.	4
07:33	09:25	1:52	1	1. 2.	3

Other issues

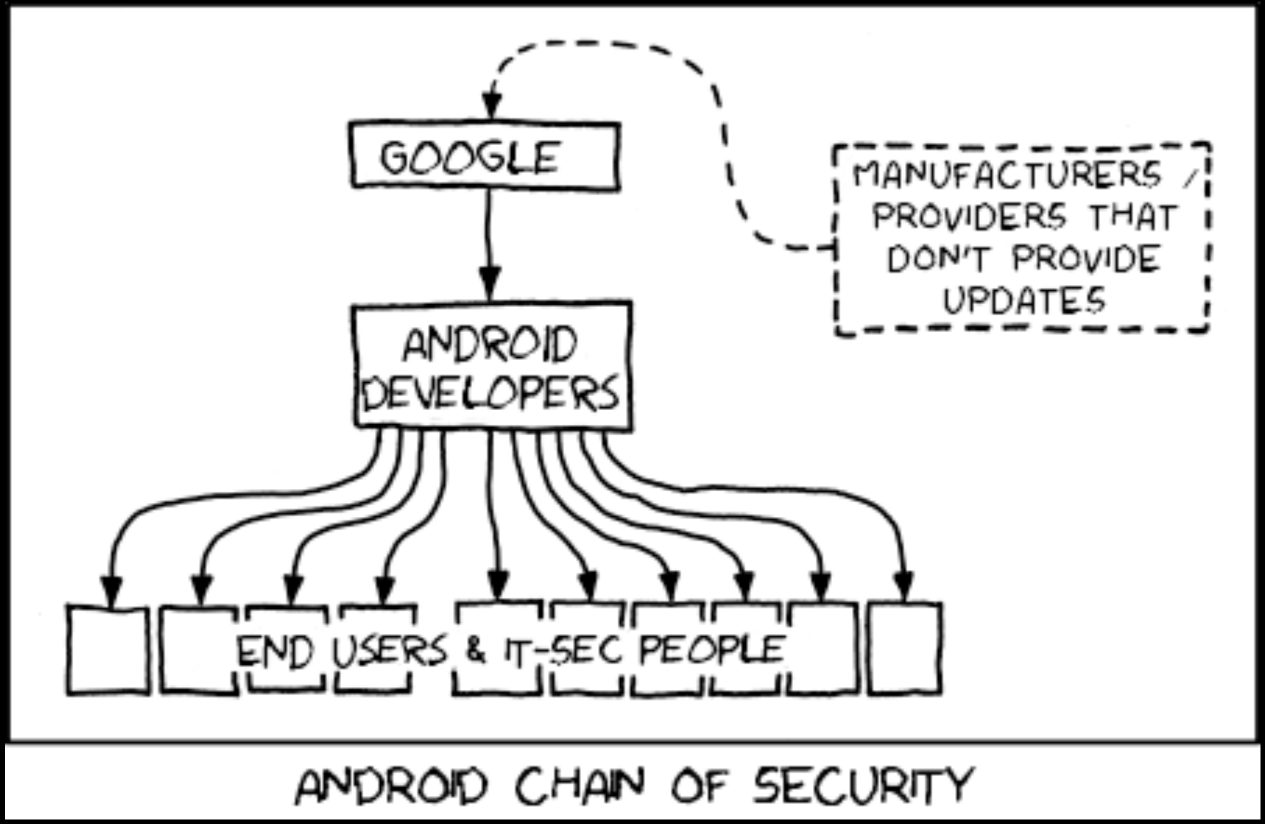
- Facebook-App V. 1.6 is able to read/write/edit SMS/MMS
- Plain authentication tokens, fixed
- SMS receiver incorrect, fixed
- Htclogger, HTC only
- App reversing
- Many more

Nuclear chain of command...


PRESIDENT
SECRETARY OF DEFENSE
ENGINEER WHO INSTALLED THE RED BUTTON
UNIFIED COMBAT COMMANDERS
US NUCLEAR CHAIN OF COMMAND

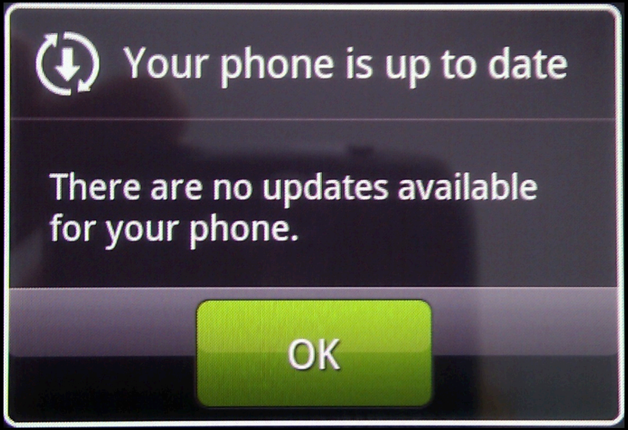
xkcd.com

... is similar to the Android chain of security


GOOGLE
ANDROID DEVELOPERS
MANUFACTURERS / PROVIDERS THAT DON'T PROVIDE UPDATES
END USERS & IT-SEC PEOPLE
ANDROID CHAIN OF SECURITY

My situation

- Bought HTC Desire in 2010
- Still on Android 2.2.0, means:
  - Screen lock circumvention (button mashing)
  - Vulnerable to DroidDream malware
  - Browser vulnerability
    - Cookie stealing / XSS
    - Can be used to install apps


Your phone is up to date

There are no updates available for your phone.

Android security / code concept

Android code

- Write app in Java and HTML/Javascript (Android SDK)
  - The obvious approach
  - Most apps from the Google Market
  - Easy to decompile/disassemble/reassemble
- Write app in ARM native code (Android NDK)
  - Together with Java code
  - ARM Assembler Reverse Engineering and JNI
- Use a framework/generator
  - appmakr.com
  - PhoneGap
  - Others?

Techniques for pentesters / reverse engineers

1. Getting hundrets of Android Apps (apk files)

Obvious download approach

- Open market app on mobile
- Click app and install
- SCP apk file from phone

→ Too slow, not enough space on mobile, etc

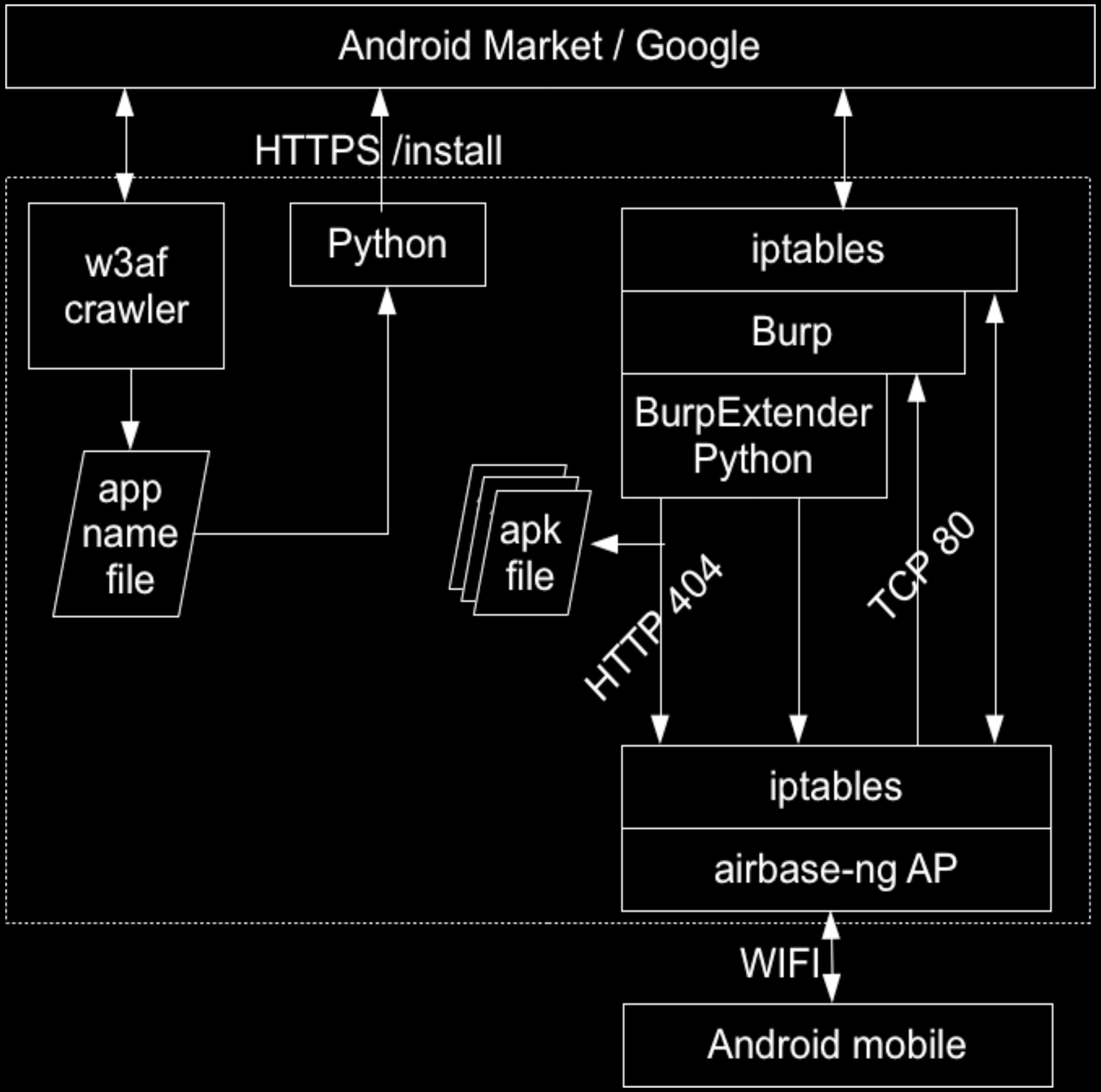
How to download all Android apps

- Connect mobile to laptop Wi-Fi with airbase-ng / dnsmasq
- Use iptables to redirect to local Burp
  - thx Android for not having a proxy option
- BurpExtender to save responses with apk files
- Send mobile a HTTP 404 not found

Install all apps?

- One HTTPS request to market.android.com
- Change the app name
  - com.google.android.youtube
- Modified w3af spider / regex plugin
  - Search for terms A ... ZZ on market.android.com
  - No restrictions (e.g. captcha) as in Google search
- Wrote script that sends HTTPS requests with app name

Download environment



Metadata

- About 300’000 apps in market
- Crawled about 10’000 app names
- Successfully downloaded and decompiled about 3’500 apps (about 15 GB)
  - Took about 3 days to download all these apps

2. Decompile/disassemble

The apktool disassembled structure

- Apk unzipped → apktool disassembled

```
+assets                         +assets
+res                            +res
  +drawable                       +drawable
    -icon.png                       -icon.png
  +layout                         +layout
    -main.xml                       -main.xml
  +values                         +values
    -strings.xml                    -strings.xml
+META-INF
-AndroidManifest.xml            -AndroidManifest.xml
-classes.dex                    +smali
                                  +com
                                    +...
                                -apktool.yml
```

Two approaches

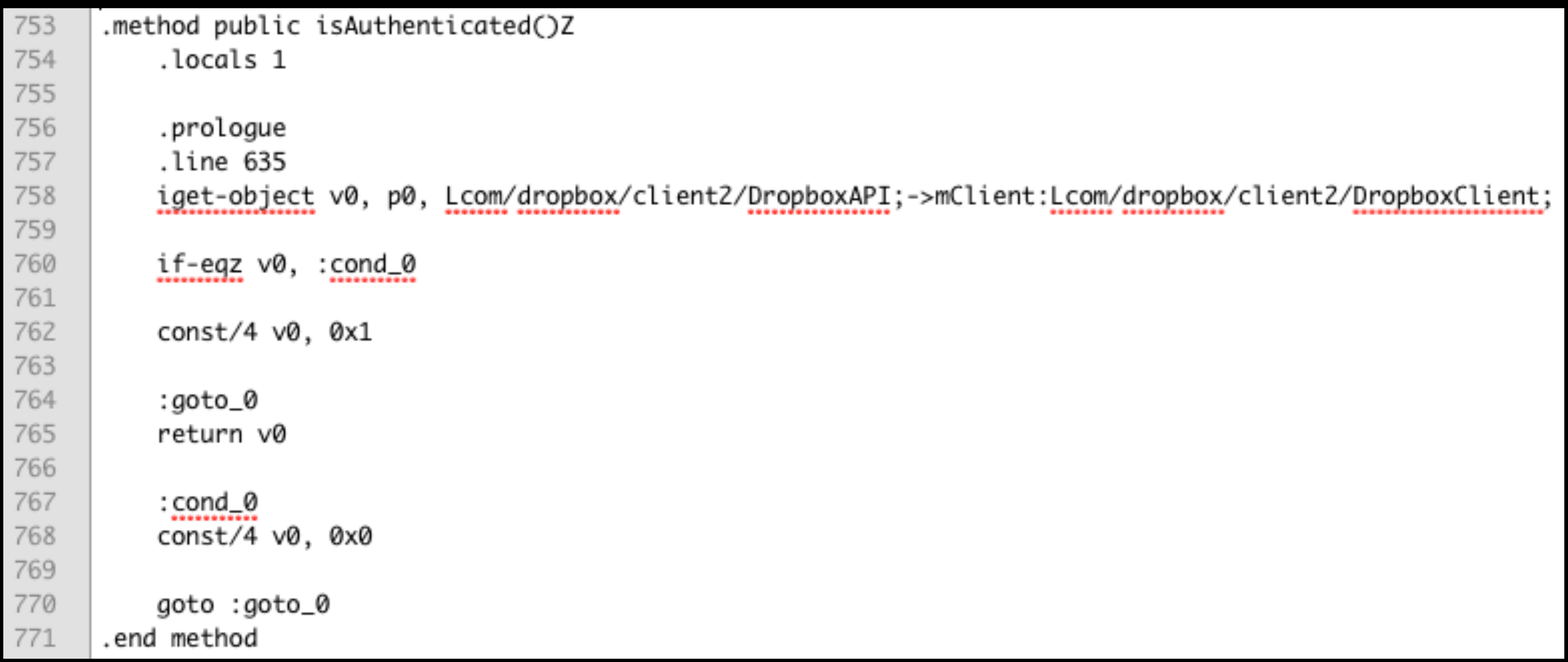
- Disassembling to smali
  - Similar to Jasmin syntax (Java assembler code)
  - Apktool
    - Correct smali code
    - Didn’t use dexdump/dedexer
- Decompiling to Java
  - Dex2Jar + Java-Decompiler
    - Sometimes incorrect Java code

Disassembling how-to

- Apktool

```
me$ java -jar apktool.jar d app.apk output-folder
```

Disassembled example



```
.method public isAuthenticated()Z
    .locals 1

    .prologue
    .line 635
    iget-object v0, p0, Lcom/dropbox/client2/DropboxAPI;->mClient:Lcom/dropbox/client2/DropboxClient;

    if-eqz v0, :cond_0

    const/4 v0, 0x1

    :goto_0
    return v0

    :cond_0
    const/4 v0, 0x0

    goto :goto_0
.end method
```

Reassembling how-to

- Apktool

```
me$ echo "change something"
change something
me$ java -jar apktool.jar b output-folder/ fake.apk
[…]
me$ keytool -genkey -alias someone -validity 100000 -
keystore someone.keystore
[…]
me$ jarsigner -keystore someone.keystore fake.apk someone
me$ adb install fake-app.apk
```

3. Other techniques for pentesters

Heap dump

```
me$ su
me# ps | grep kee
  949 10082     183m S    com.android.keepass
  960 0         1964 S    grep kee
me# kill -10 949
me# grep password /data/misc/heap-dump-tm1312268434-
pid949.hprof
thisisasecretpassword
```

- In Android > 2.3
  - Button in DDMS tool or call android.os.Debug.dumpHprofData(fileName)

Invoking Activities

- Activities are basically user interfaces
  - „one screen“

```
me$ dumpsys package > packages.txt
me$ am start -n com.android.keepass/
com.keepassdroid.PasswordActivity
```

- Fortunately this example doesn‘t work

Tons of other tools

- Androguard
- Apkinspector
  - GUI combining apktool, dex2jar, a Java decompiler, byte code, etc.
- DED
- androidAuditTools
- Smartphonesdumbapps
- Taintdroid (Privacy issues)
- Android Forensic Toolkit
- viaExtract
- More

Experiences when decompiling/ disassembling 3‘500 apps

Finding security related issues

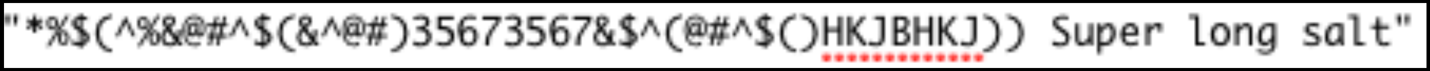
Metadata

- About 3’500 apps
  - 2’300 unique email addresses
  - 1’000 «fuck»
  - Several twitter / facebook / flickr / geocaching API keys

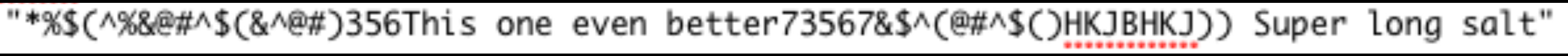
Low hanging fruits

Hashing and encryption – a short best practices refresh

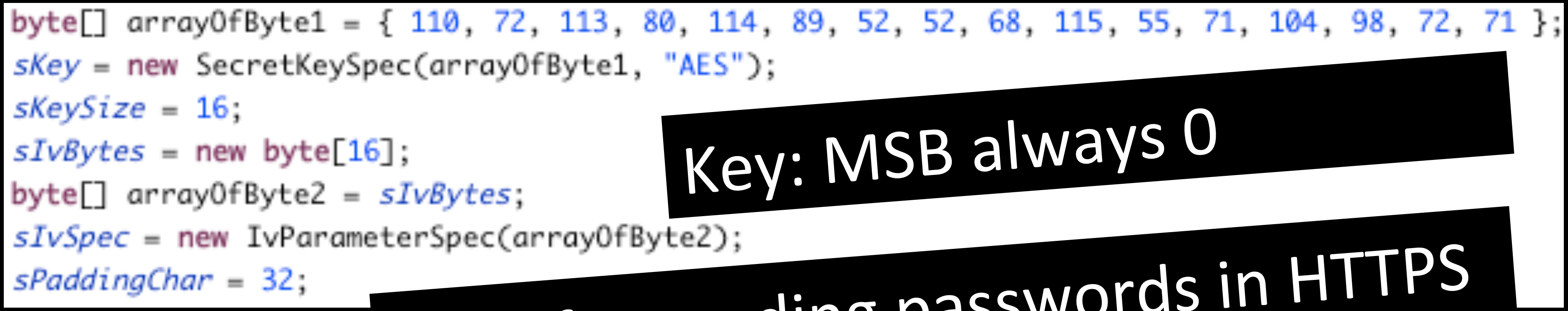
- Secure algorithms/implementations
- Random, long salts/keys
- Hashing
  - Separate salt for every hash
  - Several hashing rounds
    - E.g. hash(hash(... hash(pwd+salt)+salt ...))
- Encryption
  - Keep the key secret



```
"*%$(^%&@#^$(&^@#)35673567&$^(@#^$()HKJBHKJ)) Super long salt"
```



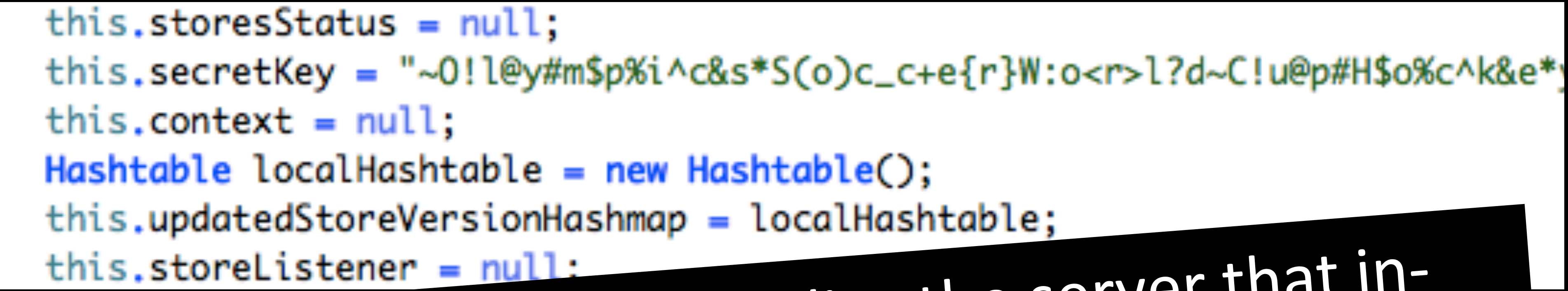
```
"*%$(^%&@#^$(&^@#)356This one even better73567&$^(@#^$()HKJBHKJ)) Super long salt"
```



```
byte[] arrayOfByte1 = { 110, 72, 113, 80, 114, 89, 52, 52, 68, 115, 55, 71, 104, 98, 72, 71 };
sKey = new SecretKeySpec(arrayOfByte1, "AES");
sKeySize = 16;
sIvBytes = new byte[16];
byte[] arrayOfByte2 = sIvBytes;
sIvSpec = new IvParameterSpec(arrayOfByte2);
sPaddingChar = 32;
```

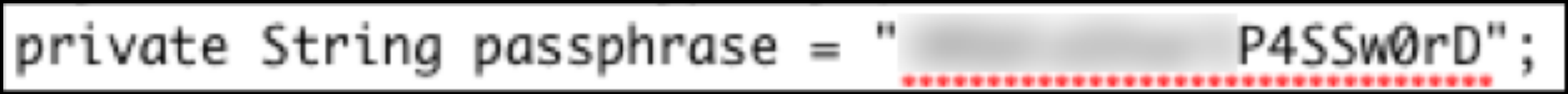
Key: MSB always 0

Used for sending passwords in HTTPS

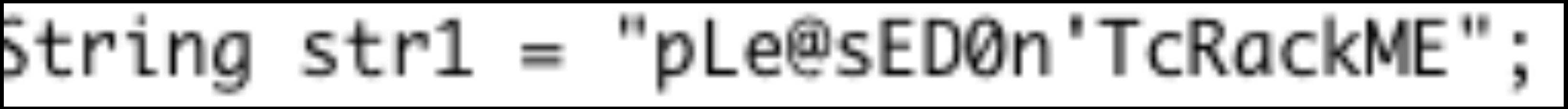


```
this.storesStatus = null;
this.secretKey = "~O!l@y#m$p%i^c&s*S(o)c_c+e{r}W:o<r>l?d~C!u@p#H$o%c^k&e*
this.context = null;
Hashtable localHashtable = new Hashtable();
this.updatedStoreVersionHashmap = localHashtable;
this.storeListener = null;
```

Used to signalise the server that in-game goods were purchased

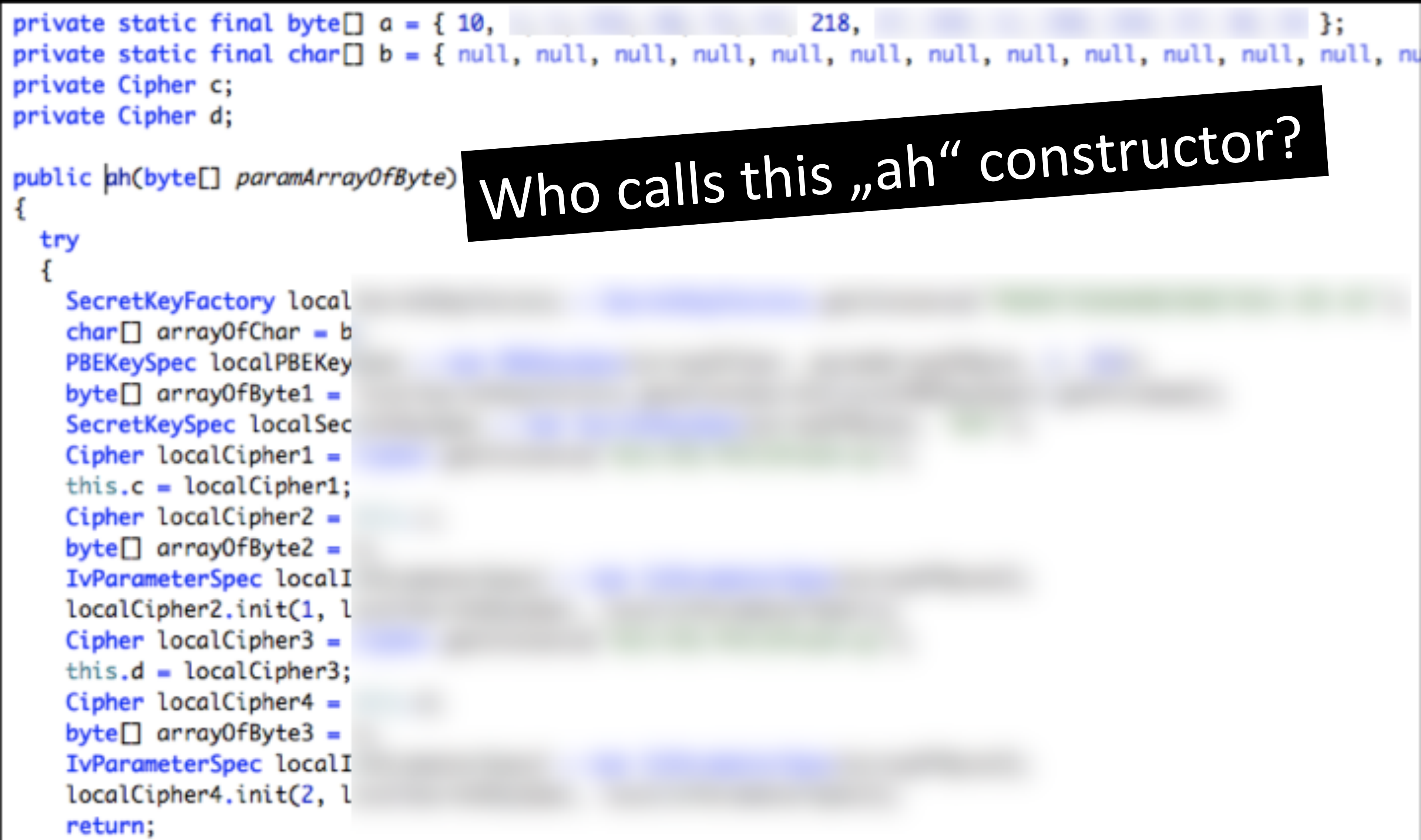


```
private String passphrase = "[redacted]P4SSw0rD";
```



```
String str1 = "pLe@sED0n'TcRackME";
```

Obfuscated code



```
private static final byte[] a = { 10, [redacted] 218, [redacted] };
private static final char[] b = { null, null, null, null, null, null, null, null, null, null, null, null, nu
private Cipher c;
private Cipher d;

public ah(byte[] paramArrayOfByte)
{
  try
  {
    SecretKeyFactory local
    char[] arrayOfChar = b
    PBEKeySpec localPBEKey
    byte[] arrayOfByte1 =
    SecretKeySpec localSec
    Cipher localCipher1 =
    this.c = localCipher1;
    Cipher localCipher2 =
    byte[] arrayOfByte2 =
    IvParameterSpec localI
    localCipher2.init(1, l
    Cipher localCipher3 =
    this.d = localCipher3;
    Cipher localCipher4 =
    byte[] arrayOfByte3 =
    IvParameterSpec localI
    localCipher4.init(2, l
    return;
```
[redacted]

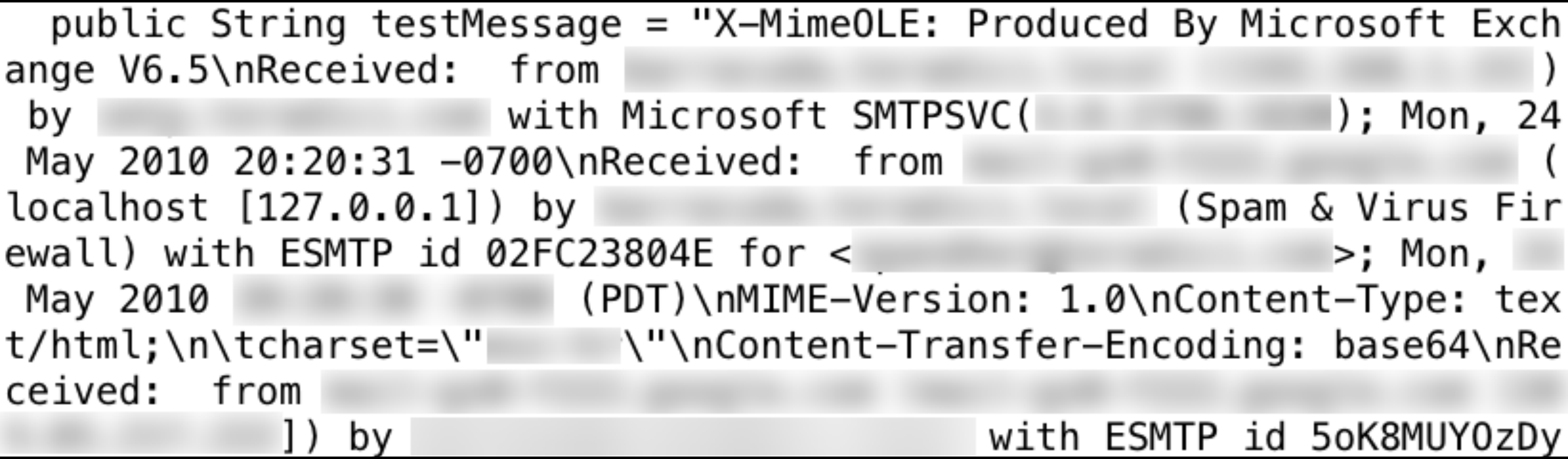
Who calls this „ah“ constructor?

Obfuscated code

- 4 greps later...
- c.f includes the key
  - c.f calls a.bs(key)
    - a.bs calls a.ah(key)
      - a.ah uses the key and locale variables for encryption
- We know all the input data for the encryption routine
- It‘s symmetric crypto
- We can decrypt „it“ (whatever it might be)

TestXXXXX.java

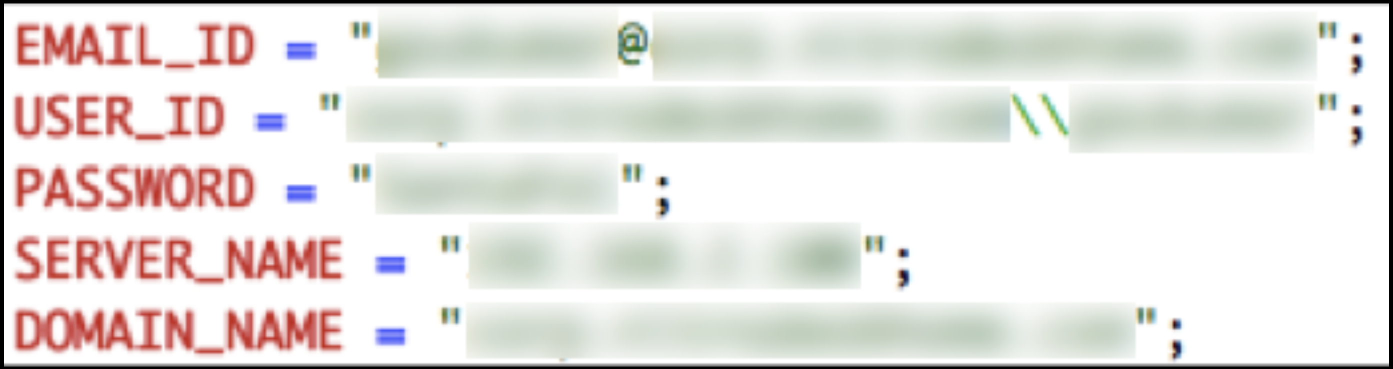
- Yeah, let’s copy/paste a test email!



```
  public String testMessage = "X-MimeOLE: Produced By Microsoft Exch
ange V6.5\nReceived:  from [redacted] )
 by [redacted] with Microsoft SMTPSVC([redacted]); Mon, 24
 May 2010 20:20:31 -0700\nReceived:  from [redacted] (
localhost [127.0.0.1]) by [redacted] (Spam & Virus Fir
ewall) with ESMTP id 02FC23804E for <[redacted]>; Mon, [redacted]
 May 2010 [redacted] (PDT)\nMIME-Version: 1.0\nContent-Type: tex
t/html;\n\tcharset=\"[redacted]\"\nContent-Transfer-Encoding: base64\nRe
ceived:  from [redacted]
[redacted]]) by [redacted] with ESMTP id 5oK8MUYOzDy
```

TestXXXX2.java

- And credentials for the test server...



```
EMAIL_ID = "[redacted]@[redacted]";
USER_ID = "[redacted]\\[redacted]";
PASSWORD = "[redacted]";
SERVER_NAME = "[redacted]";
DOMAIN_NAME = "[redacted]";
```

Some apps I looked at more closely

(it’s getting worse)

App 1 - banking app

- Who really wants banking on the mobile?
- A lot of banking apps! Yay!
- App 1
  - No obfuscation + can easily be recompiled
  - App simply shows the website
  - Hides the URL and SSL cert/lock from the user
  - Can only be used with mTAN

App 2

- Server had self-signed SSL certificate
- SSL MITM Dump:

```
/usernam e=B1436A 13E85D20 F2428D6E 232C2B93
FE....pa ssword=2 C30F3866 016E6C59 52655C06
400BCC6. imei=405 23204606 E450... ...
```

Wow, it's encrypted... Don't we need a key for that?

App 2

- AES key

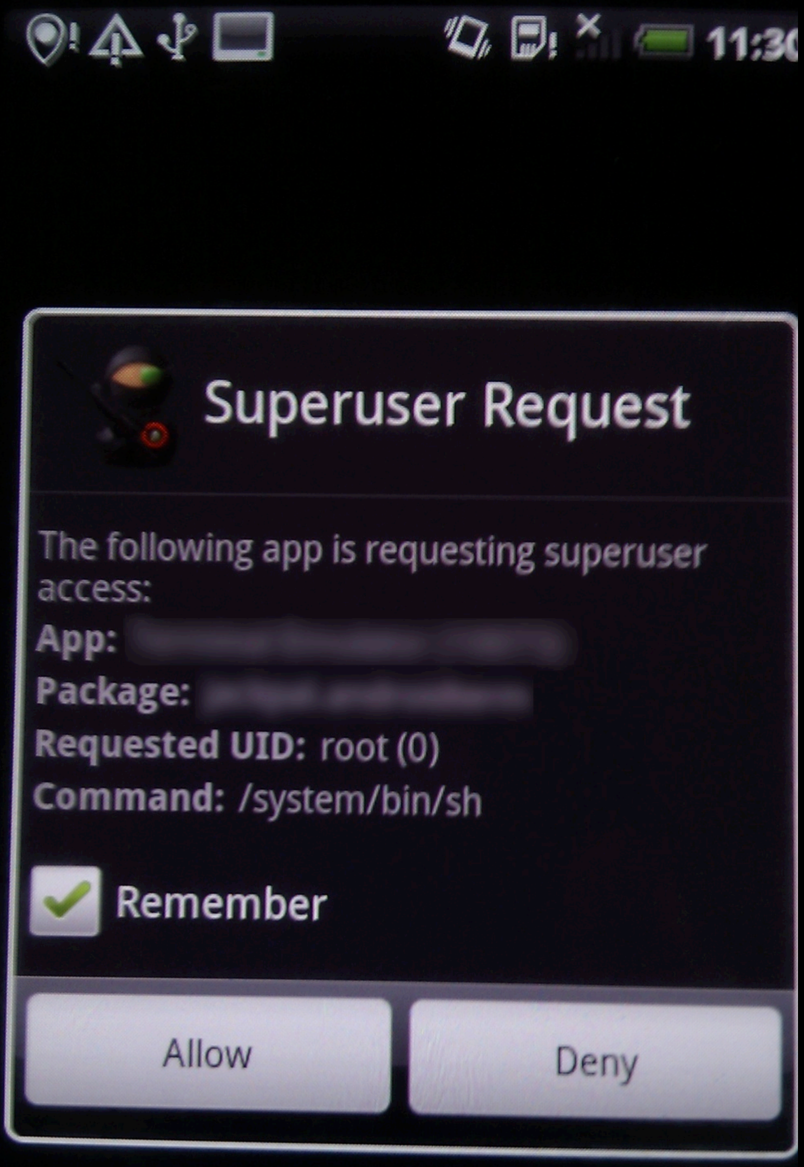
```
public byte[] cryptKey42 = {-31, -21, 4, 24, -21,
54, -63, -40, -38, 61, -47, -115, -95, -36, -142,
64, 53, 120, -85, -96, -69, 85, 81, 16, -36, 80,
-102, 95, -20, 110, 36, -11};
```

App 3 – root detection

```
private boolean deviceRoot(){
     try{
       Runtime.getRuntime().exec("su");
       return true;
     }
     catch (IOException localIOException){
       return false;
     }
}
```

App 3 – Circumventing root detection

- Not necessary



Superuser Request

The following app is requesting superuser access:

App: [redacted]

Package: [redacted]

Requested UID: root (0)

Command: /system/bin/sh

✓ Remember

App 4 – Another root detection

```
public static boolean isDeviceRooted(){
       File f = new File("/system/sbin/su")
       return f.exists()
}
```

App 4 - Removing root detection

```
me$ java -jar apktool.jar d app.apk source
[…]
me$ sed -i "" 's/system\/sbin\/su/system\/sbin\/
CEW1PFSLK/g' source/smali/net/example/checks.smali
me$ java -jar apktool.jar b source/ fake.apk
[…]
me$ keytool -genkey -alias someone -validity 100000
-keystore someone.keystore
[…]
me$ jarsigner -keystore someone.keystore fake.apk
someone
me$ adb install fake.apk
```

App 4 – Was that a good method to remove the root detection?

- Altering the app
  - No updates
- We only want to fail that simple check

App 4 - Prevent root detection

root stays root!

```
me$ adb shell
$ su
# cd /system/bin/; mount -o remount,rw -o rootfs rootfs /;
mount -o remount,rw -o yaffs2 /dev/block/mtdblock3 /system
# echo $PATH
/sbin:/system/sbin:/system/bin:/system/xbin
# mv /system/sbin/su /system/xbin/
```

A special secret key

- 445 apps use the same AES key
  - byte[] a = { 10, 55, -112, -47, -6, 7, 11, 75, -7, -121, 121, 69, 80, -61, 15, 5 }

Google Ads

- Encrypt last known location
  - All location providers (GPS, Wifi, ...)
- Send via the „uule“ JSON parameter
- Notified Google on the 23th of June
  - No response yet
- To be honest I haven‘t seen the „uule“ parameter in my network yet

Google Ads

- Why didn‘t they use asymmetric crypto?

Countermeasures

- Use asymmetric crypto instead of symmetric when transferring data to a server
- Store hashes/session tokens instead of passwords
- Good obfuscation is Security Through Obscurity
- Pentest your apps
- Know the limitations
  - root stays root

References

- http://designora.com/graphics/android-logo/
- http://blog.duosecurity.com/2011/05/when-angry-birds-attack-android-edition/
- http://jon.oberheide.org/blog/2011/03/07/how-i-almost-won-pwn2own-via-xss/
- http://www.h-online.com/open/news/item/Android-apps-send-unencrypted-authentication-token-1243968.html
- https://www.infosecisland.com/blogview/13459-Google-Sued-for-Surreptitious-Android-Location-Tracking.html
- http://www.h-online.com/open/news/item/Android-malware-activates-itself-through-incoming-calls-1253807.html
- http://www.slideshare.net/bsideslondon/bsideslondon-spo#text-version
- https://www.hashdays.ch/assets/files/slides/burns_android_security_the%20fun%20details.pdf
- https://theassurer.com/p/756.html
- http://thomascannon.net/blog/2011/02/android-lock-screen-bypass/
- http://www.symantec.com/content/en/us/about/media/pdfs/symc_mobile_device_security_june2011.pdf?om_ext_cid=biz_socmed_twitter_facebook_marketwire_linkedin_2011Jun_worldwide_mobilesecuritywp
- http://www.xkcd.com/898
- http://www.madaxeman.com/general/2009/11/lost-phone.html
- http://thomascannon.net/projects/android-reversing/
- http://www.infsec.cs.uni-saarland.de/projects/android-vuln/
- http://www.madaxeman.com/general/2009/11/lost-phone.html
- http://www.heise.de/mobil/meldung/Android-verschickt-SMS-an-falsche-Empfaenger-2-Update-1162685.html
- http://blog.duosecurity.com/2011/09/android-vulnerabilities-and-source-barcelona/

Thx!

DREAMLAB
TECHNOLOGIES

- Twitter: floyd_ch
- http://floyd.ch