# Smartphone Forensics

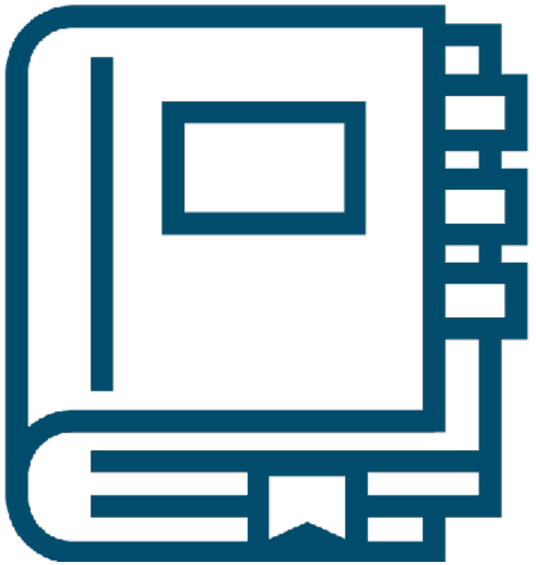
Dr. Michael Spreitzenbarth


SECURITY SAYS YOUR EMPLOYEE LOCATOR DEVICE ISN'T TURNED ON.
Dilbert.com DilbertCartoonist@gmail.com
MY WHAT?
I THINK YOU CALL IT A SMART-PHONE.
5-27-11 ©2011 Scott Adams, Inc. / Dist. by Universal Uclick
I MIGHT HAVE SOME QUESTIONS.
PUT THEM IN A TEXT TO YOUR-SELF. I'LL READ THEM LATER.

# Agenda and Dates

# Agenda

- 2018-05-11: Mobile Device Forensics
- 2018-05-25: Android and Forensic Investigation of this OS
- 2018-06-01: Apple iOS and Forensic Investigation of this OS
- **2018-06-08: Mobile Malware & Hacking Android Apps**

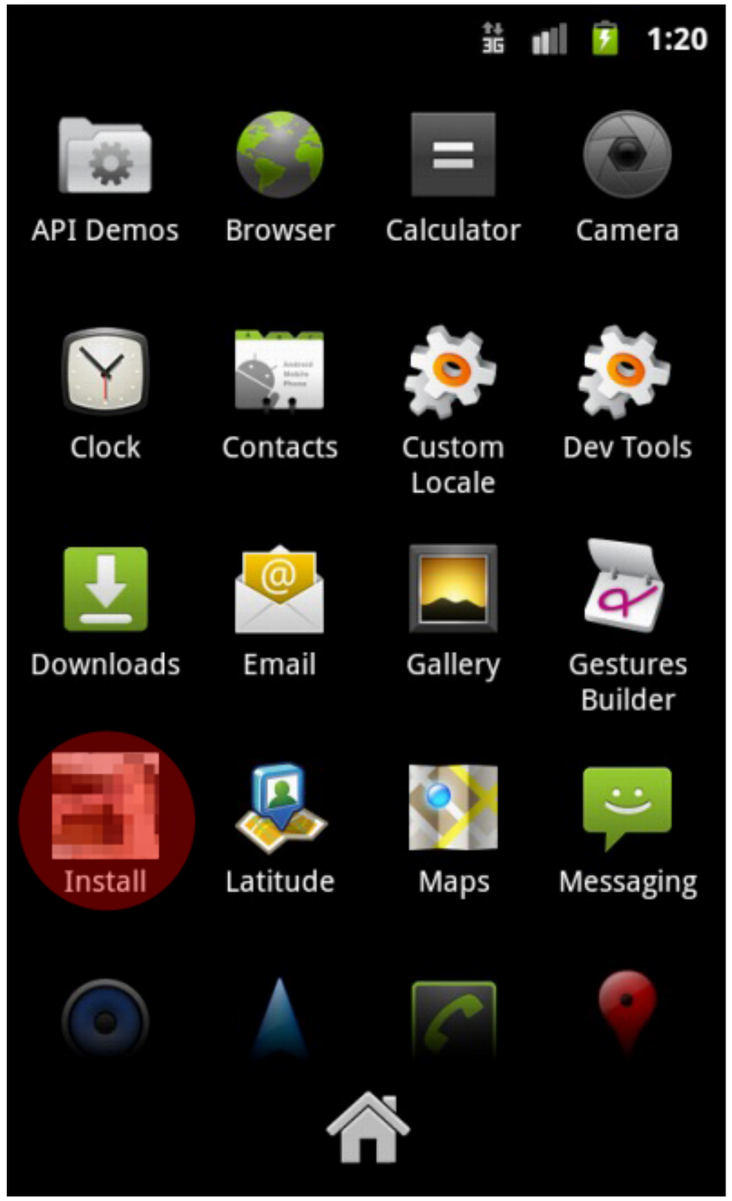


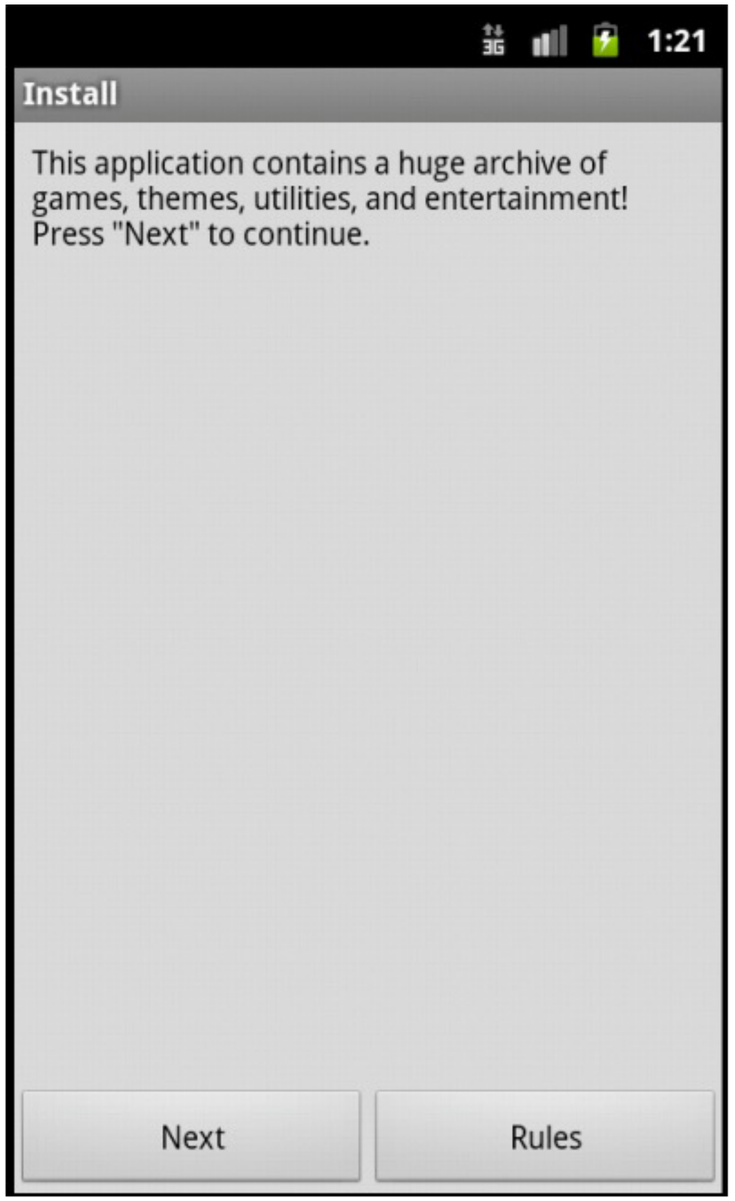
# Mobile Malware

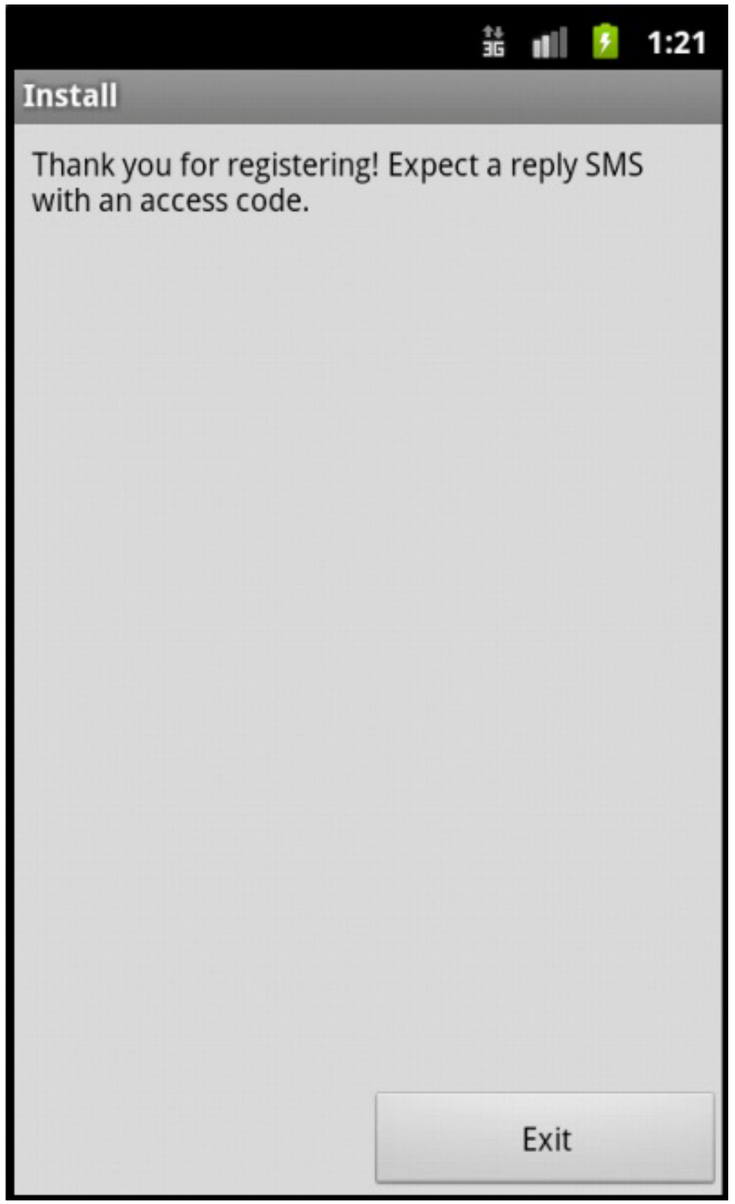
# Types of Mobile Malware

| PUA | Fraudware | Malware |
| --- | --- | --- |
| often called „hacking tools“ | trick fraud apps | clearly malicious |
| can be dangerous for the device integrity and the user | apps send expensive SMS messages | is hiding its actions |
| will often be used for pentesting purposes | within the general terms and conditions it is often stated what the app is doing | often also called trojan or virus |
| can be used to monitor traffic on the smartphone | | many versions/families steal personal data |
| | | latest versions also have the capability to encrypt phones |

# Fraudware


1:20
API Demos
Browser
Calculator
Camera
Clock
Contacts
Custom Locale
Dev Tools
Downloads
Email
Gallery
Gestures Builder
Install
Latitude
Maps
Messaging

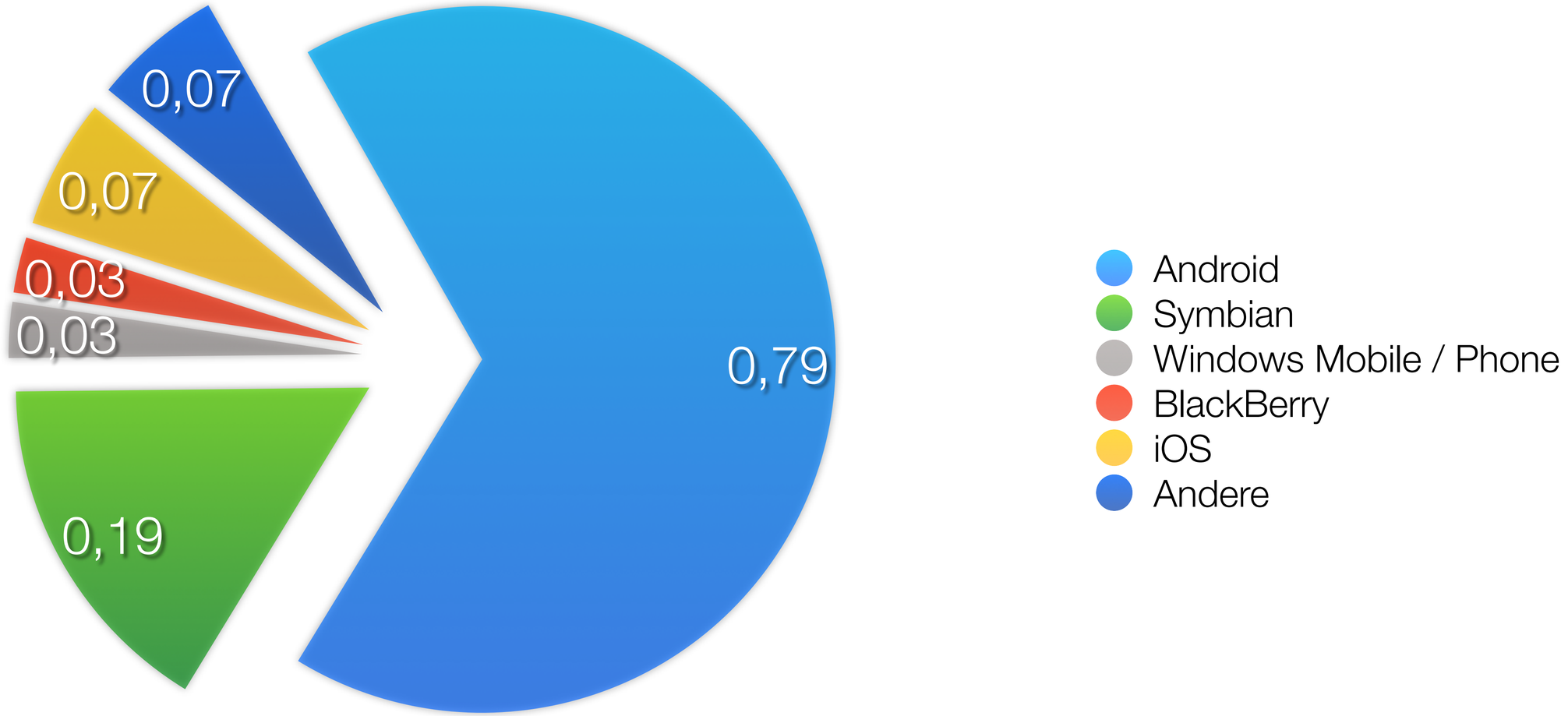

1:21
Install
This application contains a huge archive of games, themes, utilities, and entertainment! Press "Next" to continue.
Next
Rules


1:21
Install
Thank you for registering! Expect a reply SMS with an access code.
Exit

# Threats and Distribution

| Distribution | Threats |
| --- | --- |
| Using unofficial app stores | Data espionage |
| Piggyback on legitimate apps | Sending premium SMS messages |
| Upgrades of legitimate apps | Calling premium numbers |
| Social-Engineering / Phishing | Sending SPAM-Emails or -SMS messages |
| Through side-loading from malicious websites or through ad-networks | Gaining root access to the device |
| | Connecting the device to a malicious botnet |
| | Tracking a device through GPS |

# Overview


0,07
0,07
0,03
0,03
0,79
0,19
Android
Symbian
Windows Mobile / Phone
BlackBerry
iOS
Andere

# The Beginning

- Android HippoSMS Malware:
  - is hooking the legitimate SMS app
  - is sending premium SMS messages to 1066****** (located in China)
  - afterwards it registers a ContentObserver to detect incoming SMS messages
  - this observer is deleting all incoming SMS messages from numbers starting with 1066 (which belong to the premium SMS provider it sends messages to)

# Soon after it became more mature

- Android Rootsmart/Bmaster Malware
  - is using obfuscation
  - connects to a C&C server to receive commands and send statistics
  - gaining root on the smartphone by loading additional code after the initial installation
  - installing additional malicious apps
  - at the time this malware family has been detected more than 150.000 smartphones had been connected to the backend
  - the estimated revenue was between 10.000 and 20.000 USD per day
  - this malware family was undetected for several months
  - the idea for this malware was a research paper from two US researchers

# Businessmodels

- monitor calls
- read all SMS and MMS messages
- read all saved and received emails
- get the current GPS position of the device
- access calendar and address book
- access the browser history
- monitor WhatsApp and Skype
- record noises

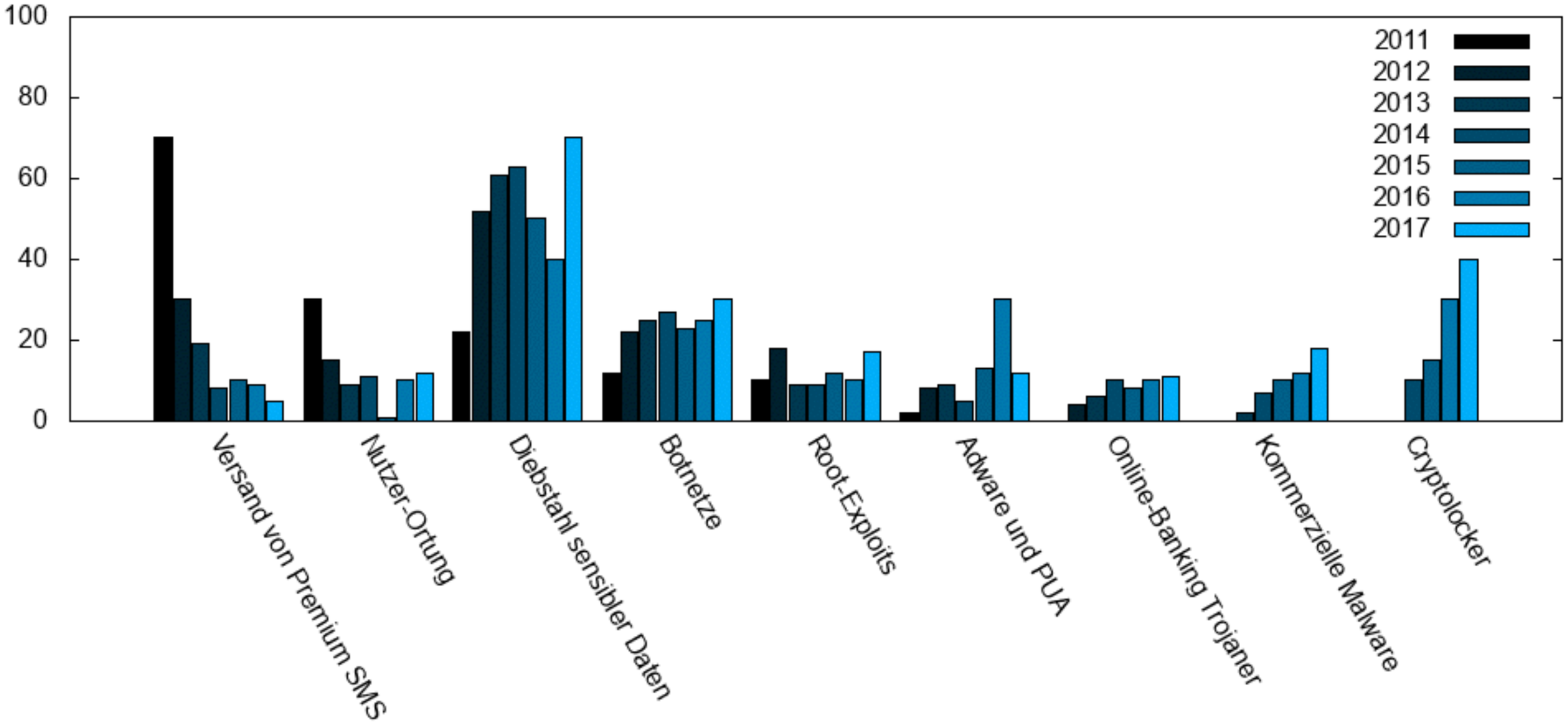

ab €29.99


ab €62.49

# State of the Art Malware

- Android Oldboot.B Malware
  - first bootkit for Android (is changing parts of the boot.img and init.rc to load itself after every restart)
  - is deleting the app used for infection and is running in RAM only
  - huge parts of the code are obfuscated or even encrypted
  - is able to detect and evade sandboxes
  - is able to detect and disable AV solutions
  - connects to C&C servers for command and control
  - hiding its configuration within pictures

# Android Malware Overview


100
80
60
40
20
0
2011
2012
2013
2014
2015
2016
2017
Versand von Premium SMS
Nutzer-Ortung
Diebstahl sensibler Daten
Botnetze
Root-Exploits
Adware und PUA
Online-Banking Trojaner
Kommerzielle Malware
Cryptolocker

# Which kind of Malware to find during forensic investigations

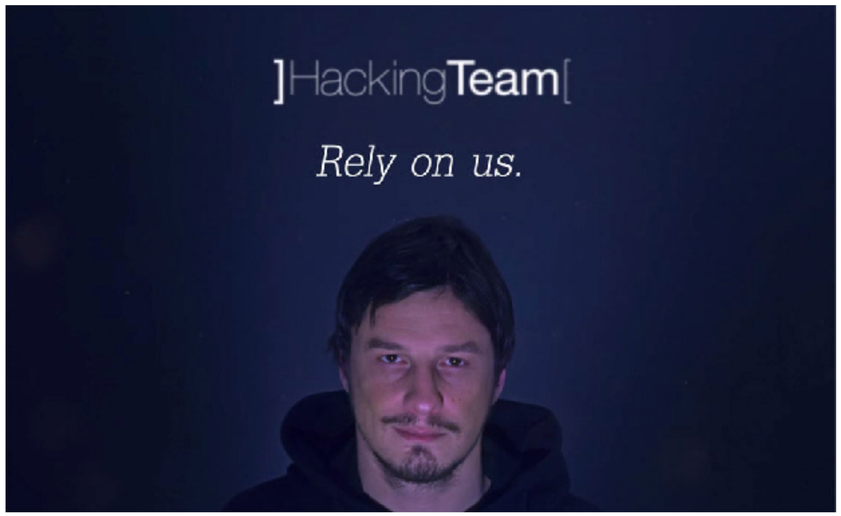
# What could be on a device?

- **Cryptolocker** are very real - also on mobile devices

- currently Android devices are the only target
- Malware is using known vulnerabilities or tricking the user into granting device administrator rights
- If this is not working, the device will just try to change the screenlock
- In Most of the cases the device is locked and no access to the device or stored data is possible
- Latest versions also leak huge amount of stored data before they lock/ encrypt the device

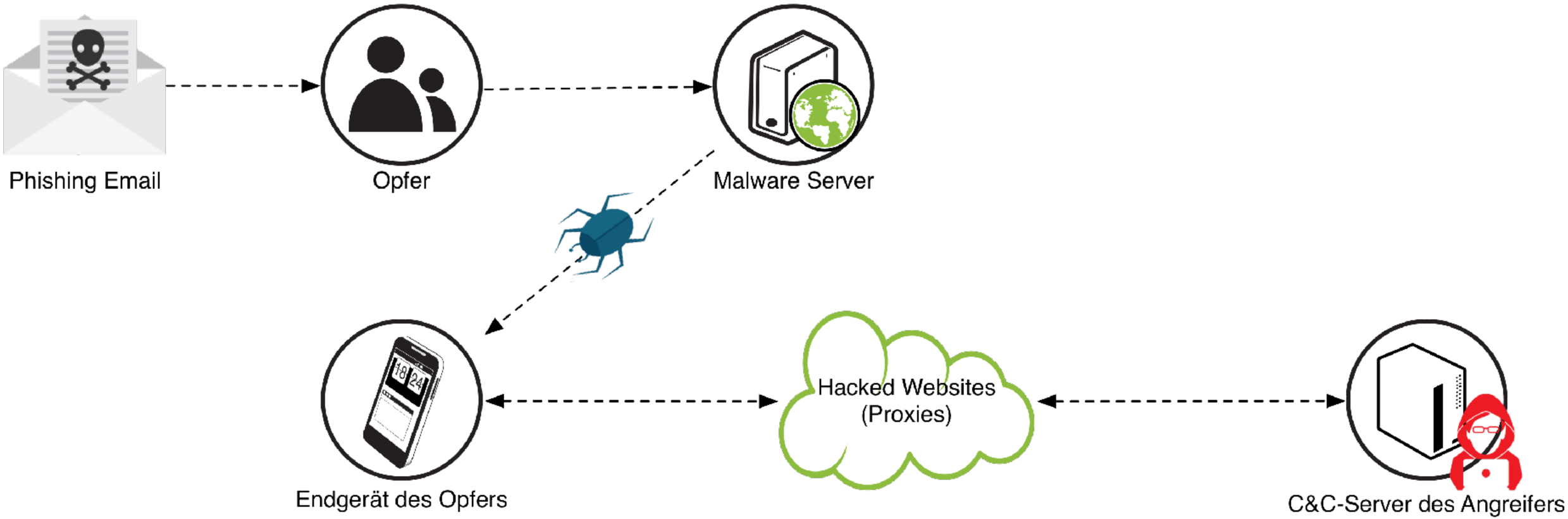
# What could be on a device?

- **Targeted Attacks** are seen more and more often on mobile devices. Most of the time because those devices are less protected than notebooks.

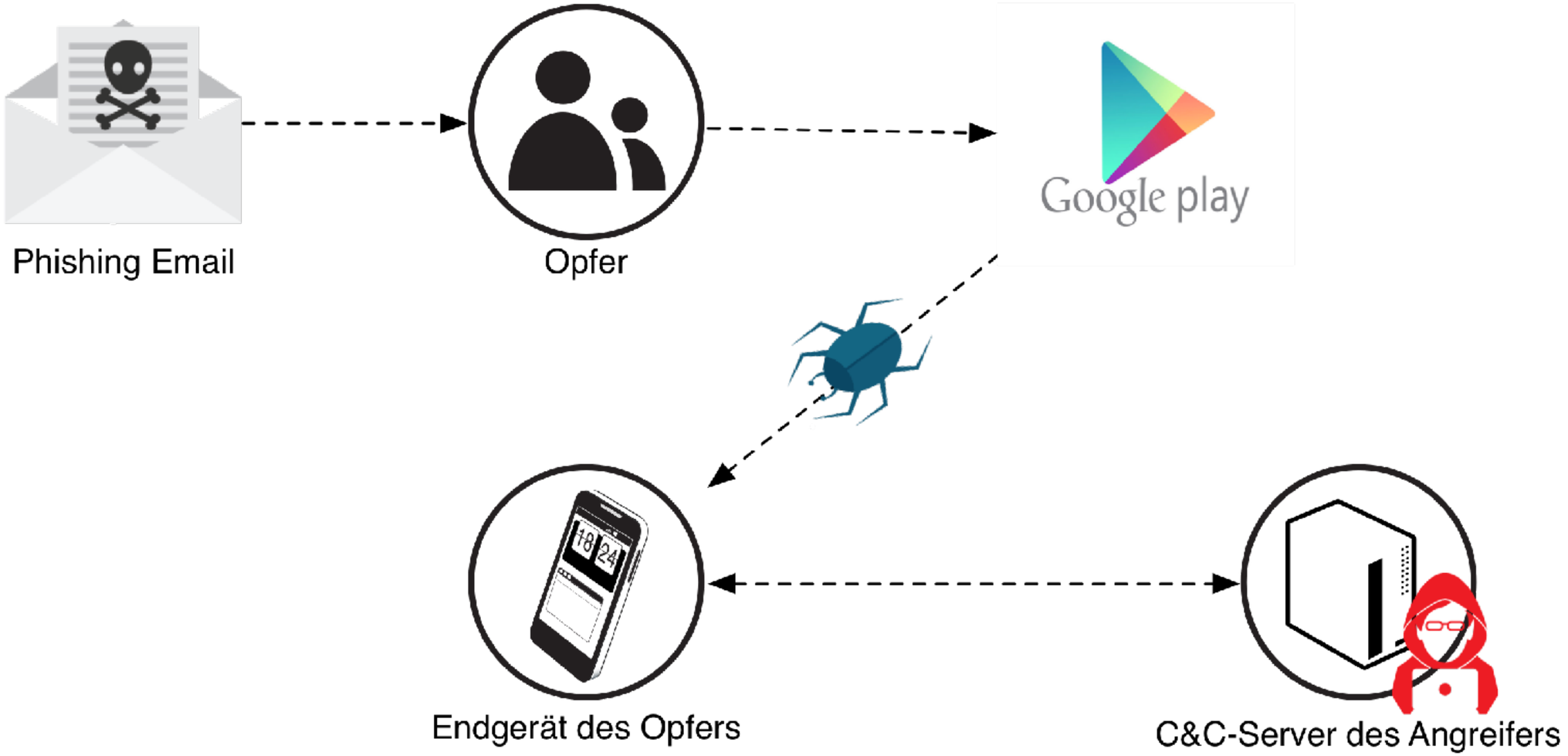



- There are a huge amount of spyware and RATs available on the market for very cheap prices.
- Those apps are often sold officially as remote maintenance tools or to protect your kids

# Mobile Incept



# Hacking Team



# OmniRAT

### File Manager

You can access, rename, delete, download and execute files and folders of the clients file system. You can also upload files and create new directories.

### Notification Manager

You can send advanced notifications to the client.

### No root required

OmniRAT does not need root permissions in order to work.

### Record Microphone

You can record the clients microphone.

### SMS Manager

You can view, delete and send sms.

### Camera

You can take pictures with the clients camera.

### Record Display

You can record the clients display. (Works only with Android 5 or higher!)

### Installed Apps Information

You can retrieve the installed apps on the clients device.

### Location Information

You can retrieve detailed information about the clients location.

# OmniRAT

### Deviceadmin Manager

You can access special deviceadmin functions. For example: Disable Camera, Wipe device or change PIN.

### Call Manager

You can make calls to any number, end running calls, block incoming and outgoing calls.

### Download Manager

You can download files from the internet through the android DownloadManager.

### Clipboard Manager

You can view and set the clients clipboard.

### Install Apps

You can install any apk on the clients device.

### Call & SMS Surveillance

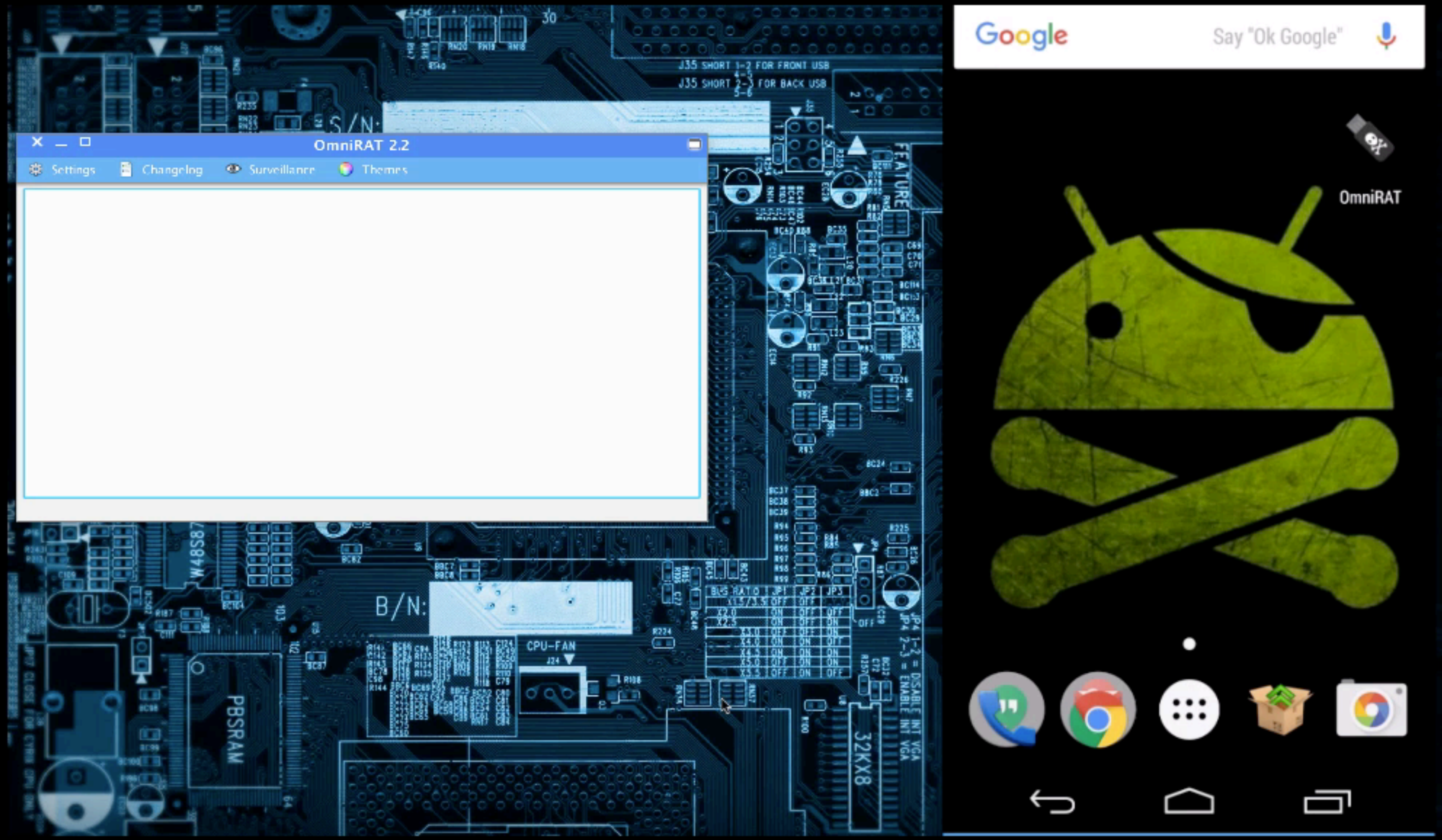
You can monitor incoming SMS & incoming and outgoing calls.

### Account Manager

You can view and manage all available accounts on the clients device.

### Browser Manager

You can view and delete the clients browser history, bookmarks and searches.

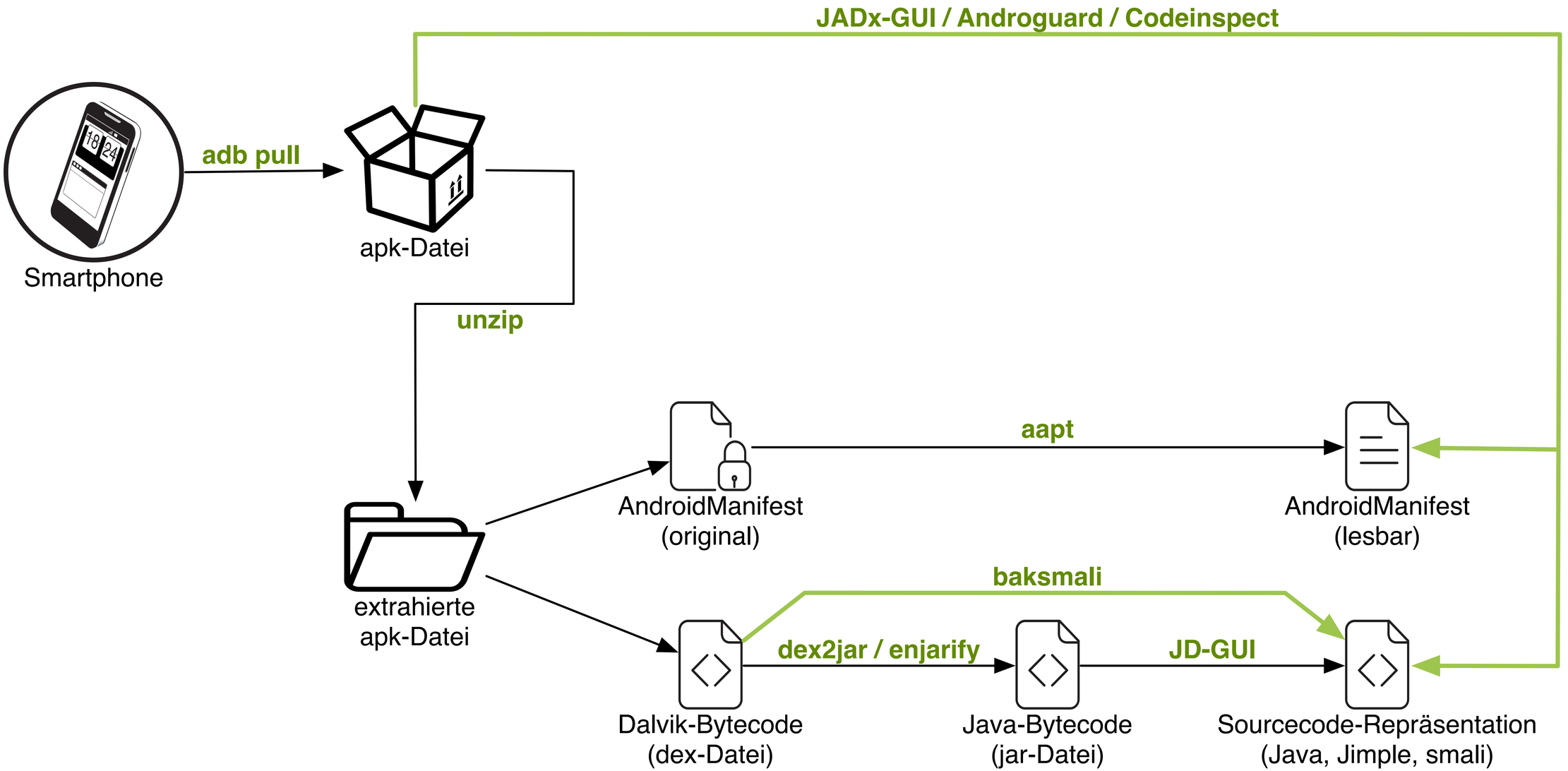

OmniRAT 2.2
Settings
Changelog
Surveillance
Themes
Google
Say "Ok Google"
OmniRAT

# Analyzing Android-based Malware

# Sandboxes and Analysis Techniques

| Static | Dynamic | Hybrid |
| --- | --- | --- |
| apps will be decompiled | App will be executed in a manipulated environment | static analysis is used as input for the dynamic one |
| sourcecode will be analyzed | the user interaction will be simulated | user interaction is way better with the input from the static analysis |
| suspicious functions/ methods will be searched | function calls will be recorded | |
| other content of the app will be analyzed (icons, linked libraries, etc.) | network traffic will also be recorded | |

# Static Analysis



# Androguard

- Androguard is a full python tool to play with Android files.
- Some of the main features are:
  - Android’s binary xml parser/viewer
  - Decompiler for DEX/ODEX files
- https://github.com/androguard/androguard

# Jadx

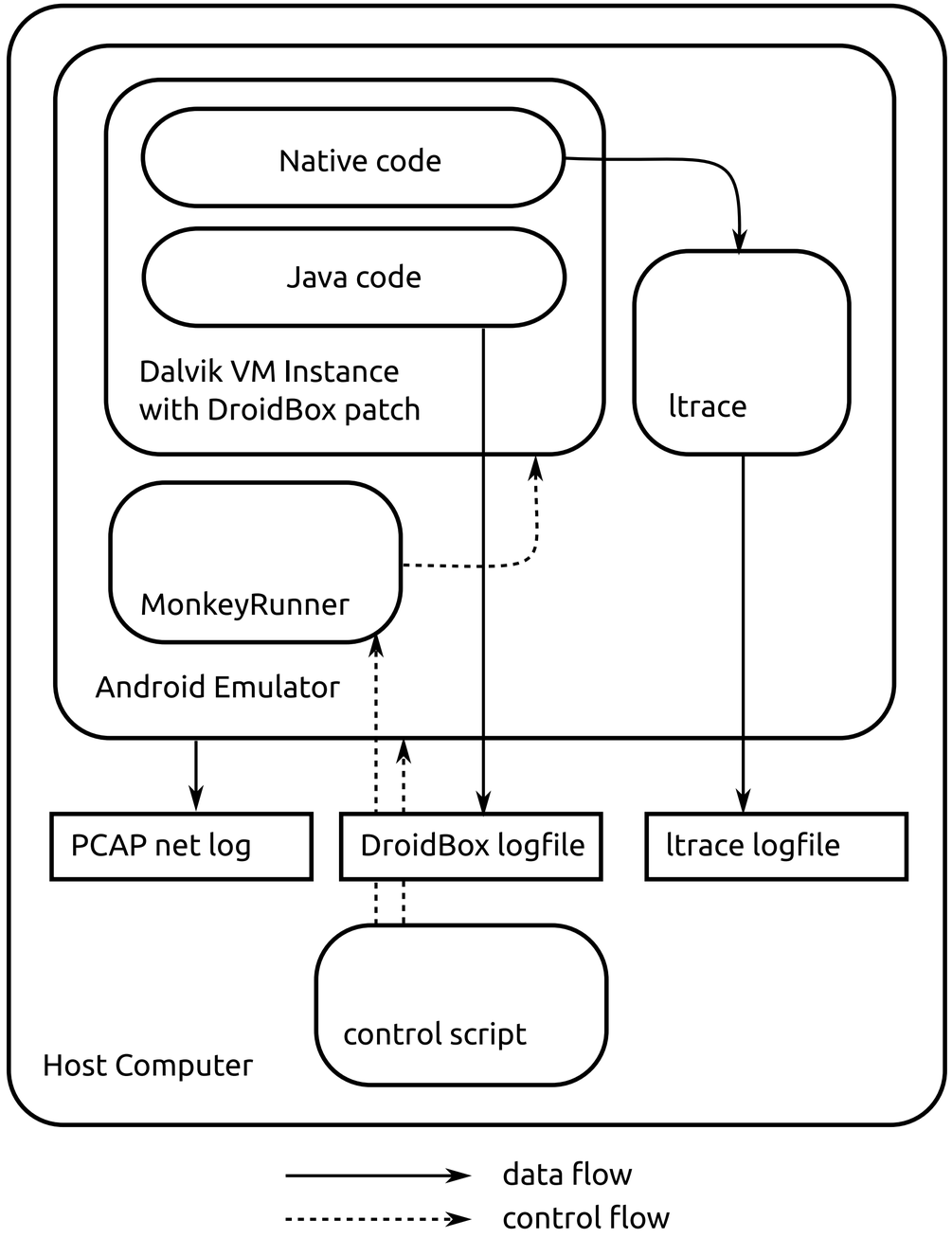
- Dex to Java decompiler
- Command line and GUI tools to receive Java source code from Android Dex and APK files.
- https://github.com/skylot/jadx/

# Codeinspect

- This is a powerful and flexible tool for analyzing Android applications in a static and dynamic manner.
- You can work on binary-only apps just as you would on source code, including debugging, refactorings, and code modification.
- You can import arbitrary Android apps into a workspace built on top of the well-known Eclipse IDE, navigate the code in our semantic-rich Jimple intermediate languages, set breakpoints, change the code as you wish, and then run it on a phone or an emulator to inspect the runtime behavior.
- https://codeinspect.sit.fraunhofer.de/

# Mobile Sandbox

- Hybrid system using the Android emulator
- One of the first systems available but currently outdated and not maintained anymore



# Cuckoo Sandbox

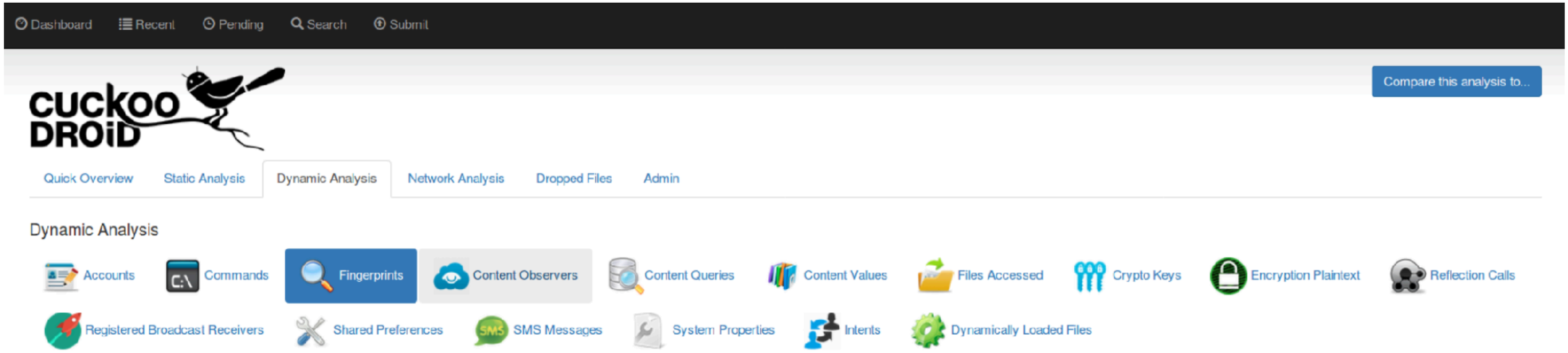
- Cuckoo Sandbox is the leading open source automated malware analysis system.

- You can throw any suspicious file at it and in a matter of minutes Cuckoo will provide a detailed report outlining the behavior of the file when executed inside a realistic but isolated environment.

# Cuckoo Sandbox

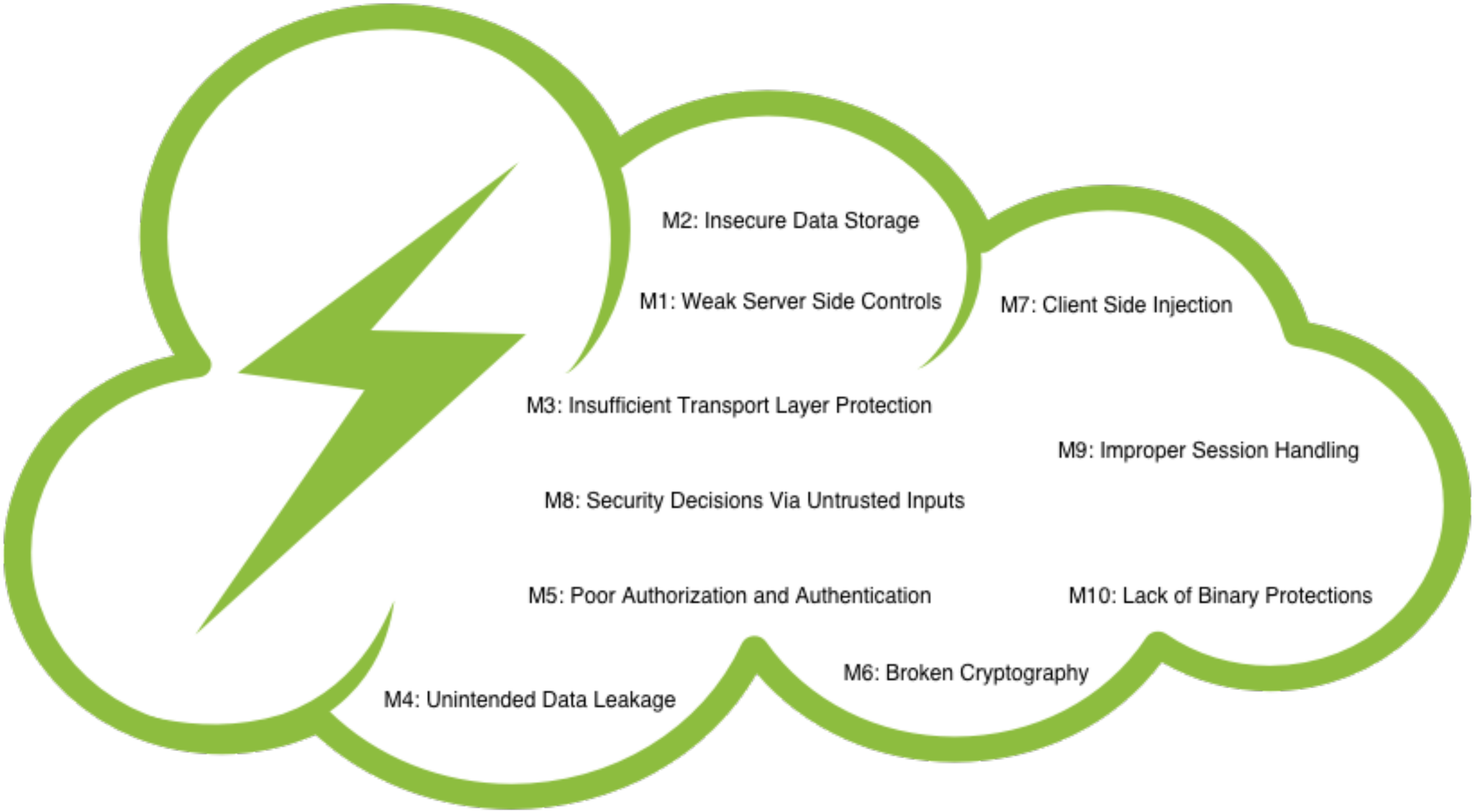
- There is a dedicated version of cuckoo for Android called cuckoo droid
- It has additional tools inside:
  - Androguard
  - Google Play Unofficial Python API
  - APKiD - Android Application Identifier for Packers, Protectors, Obfuscators and Oddities - PEiD for Android
  - https://github.com/idanr1986/cuckoodroid-2.0

# Cuckoo Sandbox

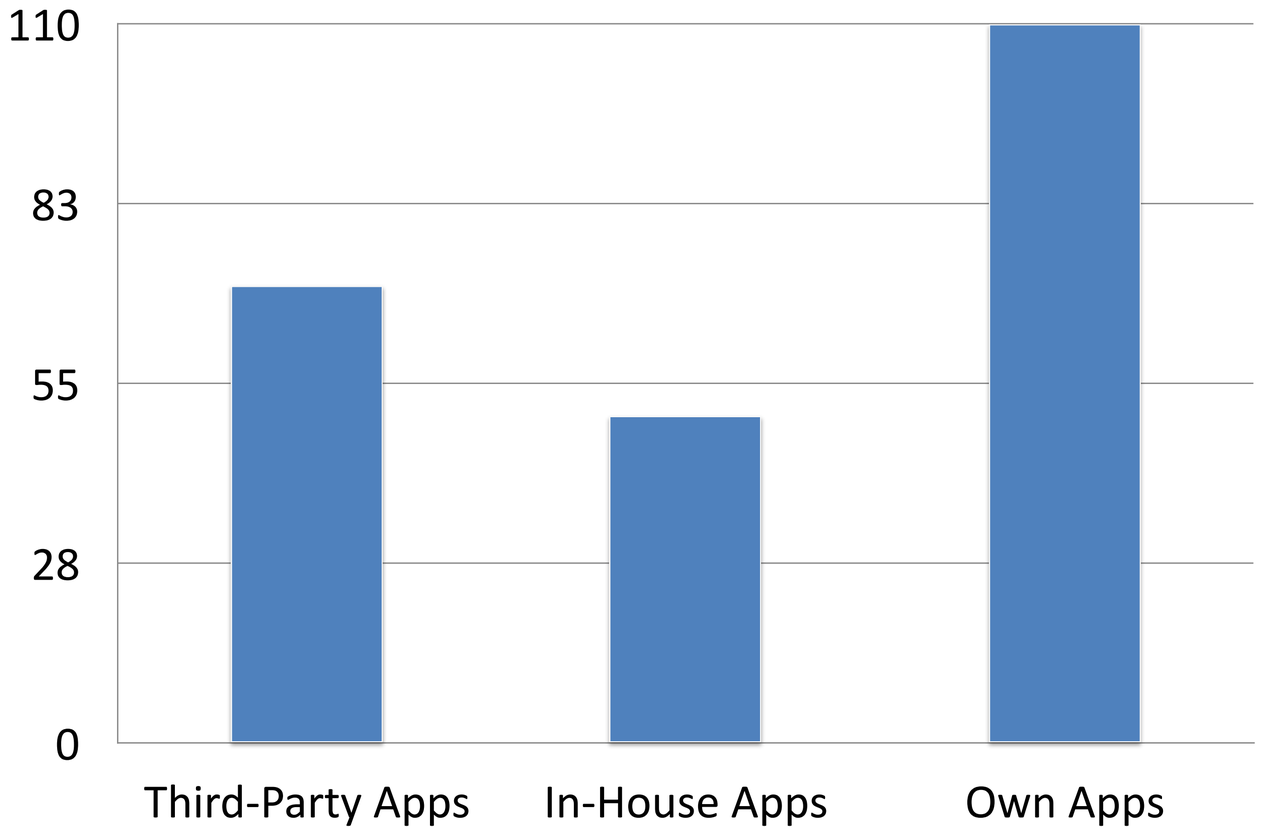


# Mobile App Hacking - Introduction

# OWASP Top10

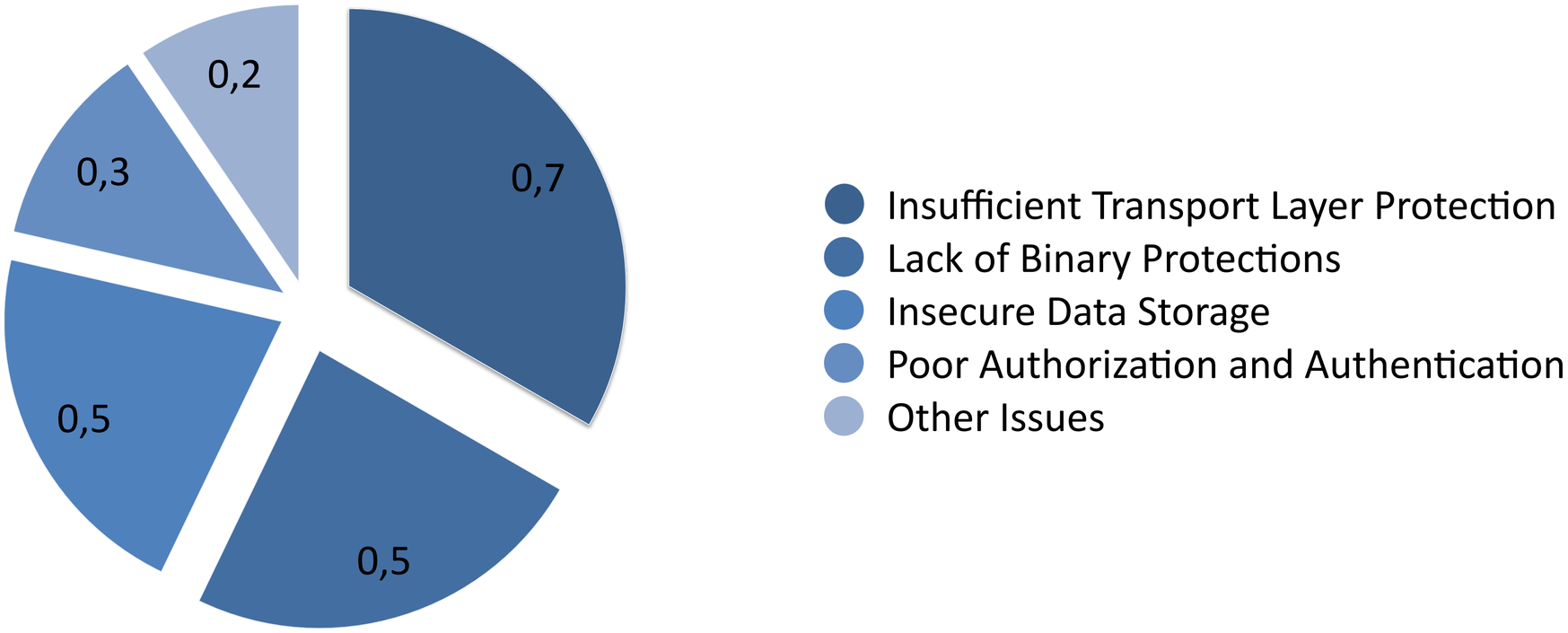

M2: Insecure Data Storage
M1: Weak Server Side Controls
M7: Client Side Injection
M3: Insufficient Transport Layer Protection
M9: Improper Session Handling
M8: Security Decisions Via Untrusted Inputs
M5: Poor Authorization and Authentication
M10: Lack of Binary Protections
M6: Broken Cryptography
M4: Unintended Data Leakage

# One Year of internal Testing in Review




- About 220 tested Apps in total
- Found more than 400 critical vulnerabilities or flaws
- 12 third-party apps had been forbidden to use within the own company
- 24 third-party apps had been limited in scope within the own company
- About ¾ of the tested apps needed another round of fixing by the developers

# How the OWASP Top10 really look like



# How to Attack an App

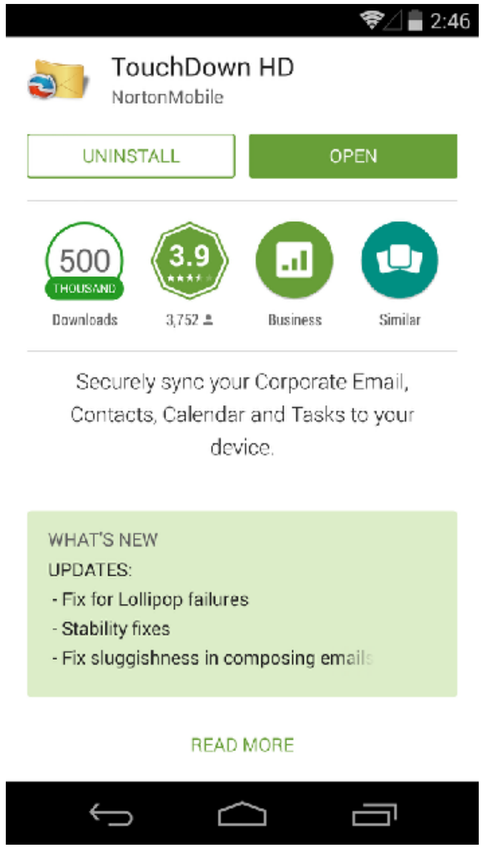
- Forensics
- Reverse Engineering
- Binary Instrumentation
- Network Interception

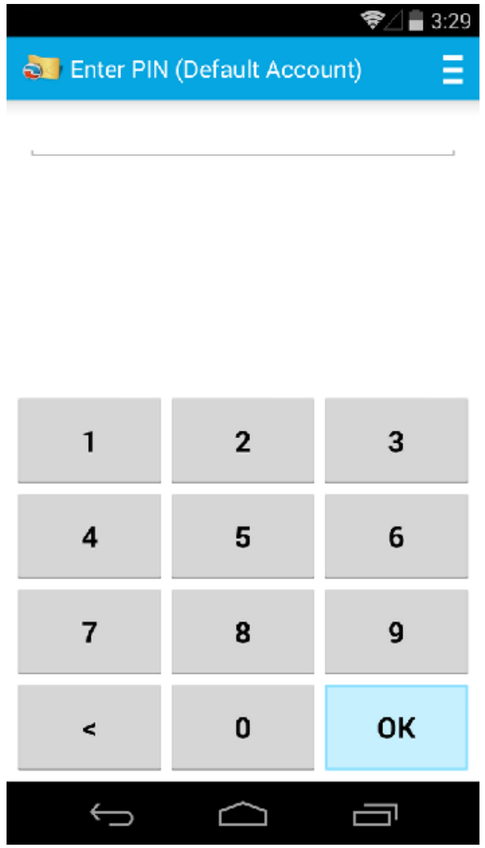
# Forensics

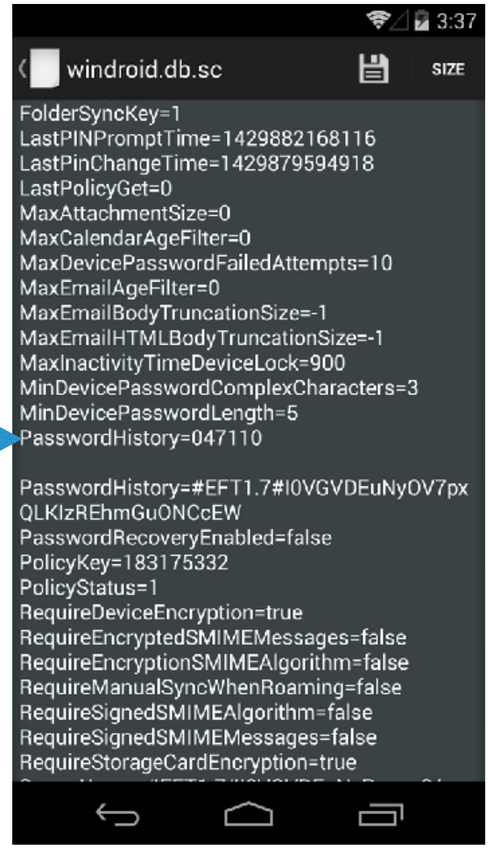
# Forensics

- Everything that you learned in the last weeks

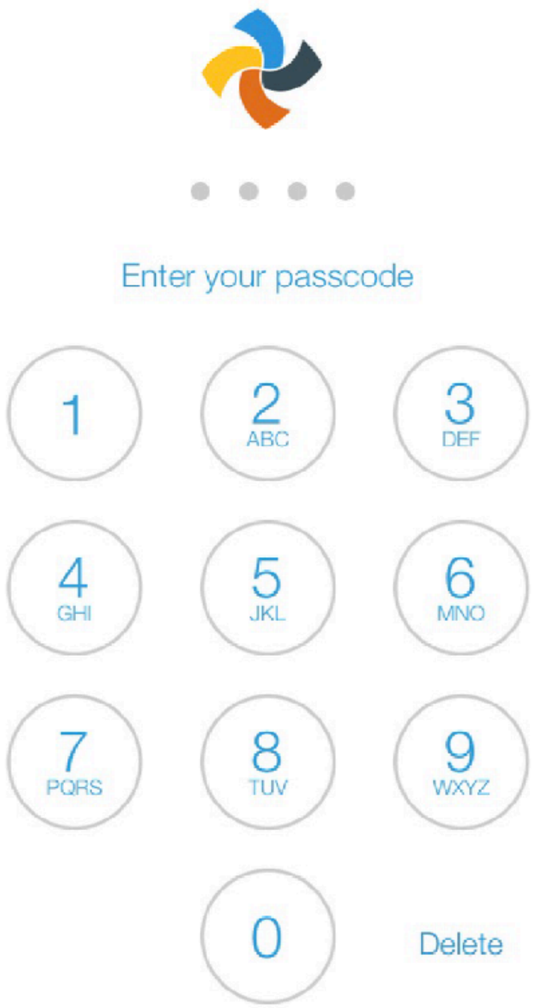
# Get Hardcoded Keys



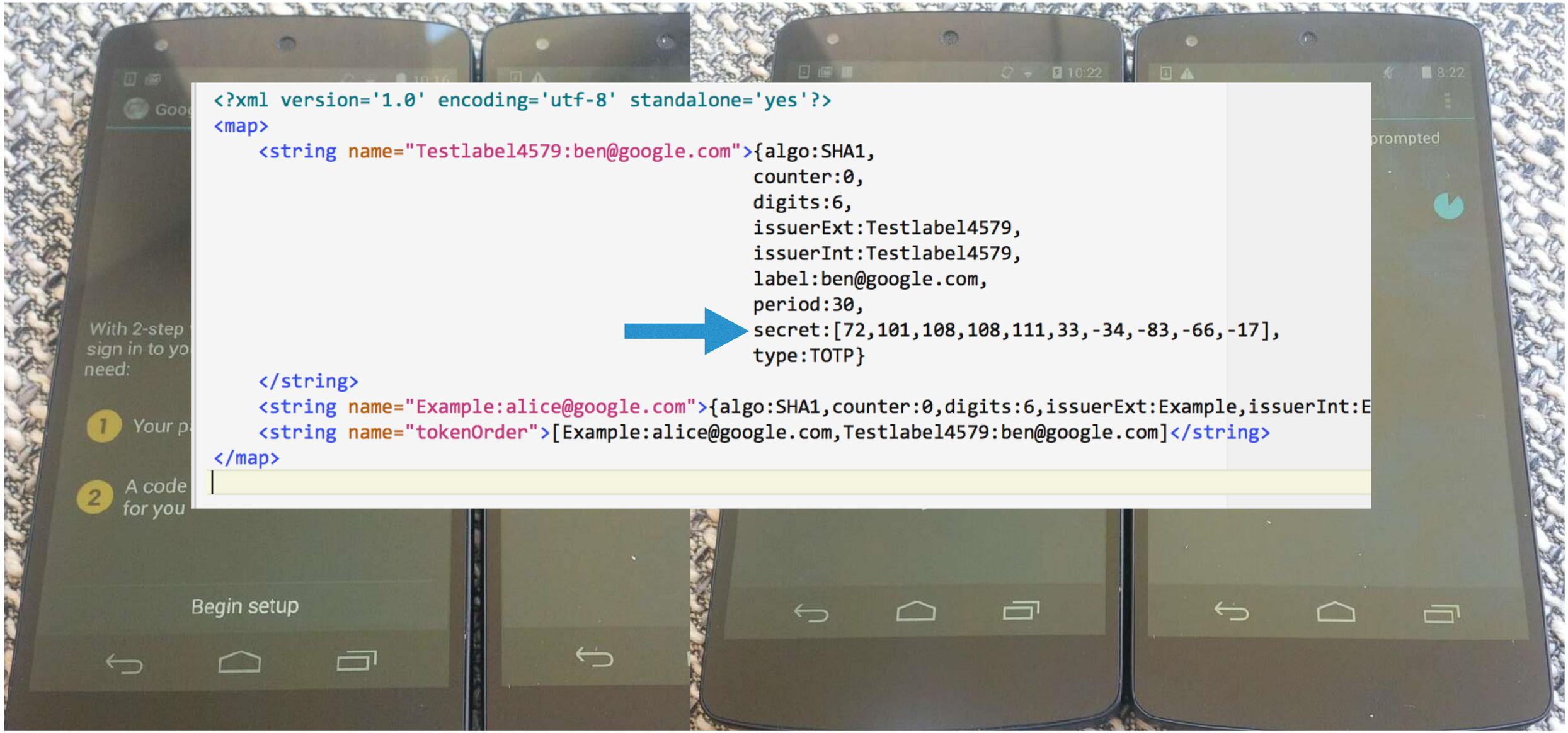



# Manipulate App Config




```
<?xml version='1.0' encoding='utf-8' standalone='yes' ?>
<map>
    <int name="mdm_access_policy" value="1" />
    <int name="share_link_expiration_policy" value="1" />
    <int name="open_in_policy" value="2" />
    <int name="passcode_failure_policy_limit" value="10" />
    <int name="passcode_failure_policy_status" value="2" />
    <int name="mobile_sync_limit_reset_day" value="1" />
    <int name="synchronization_policy" value="3" />
    <int name="passcode_enforcement_policy" value="2" />
    <int name="min_share_link_password_length" value="6" />
    <long name="mobile_sync_transfer_limit" value="1073741824" />
    <int name="share_link_expire_in_days" value="8" />
    <int name="mobile_sync_limits_policy" value="1" />
    <int name="share_link_password_complexity_policy" value="1" />
    <int name="share_link_password_protected_policy" value="1" />
    <int name="mobile_app_access_policy" value="1" />
    <int name="remote_wipe_syncpoint_policy" value="2" />
    <int name="share_link_policy" value="2" />
    <int name="synchronization_storage_policy" value="1" />
</map>
```

# Circumvent 2FA/OTP



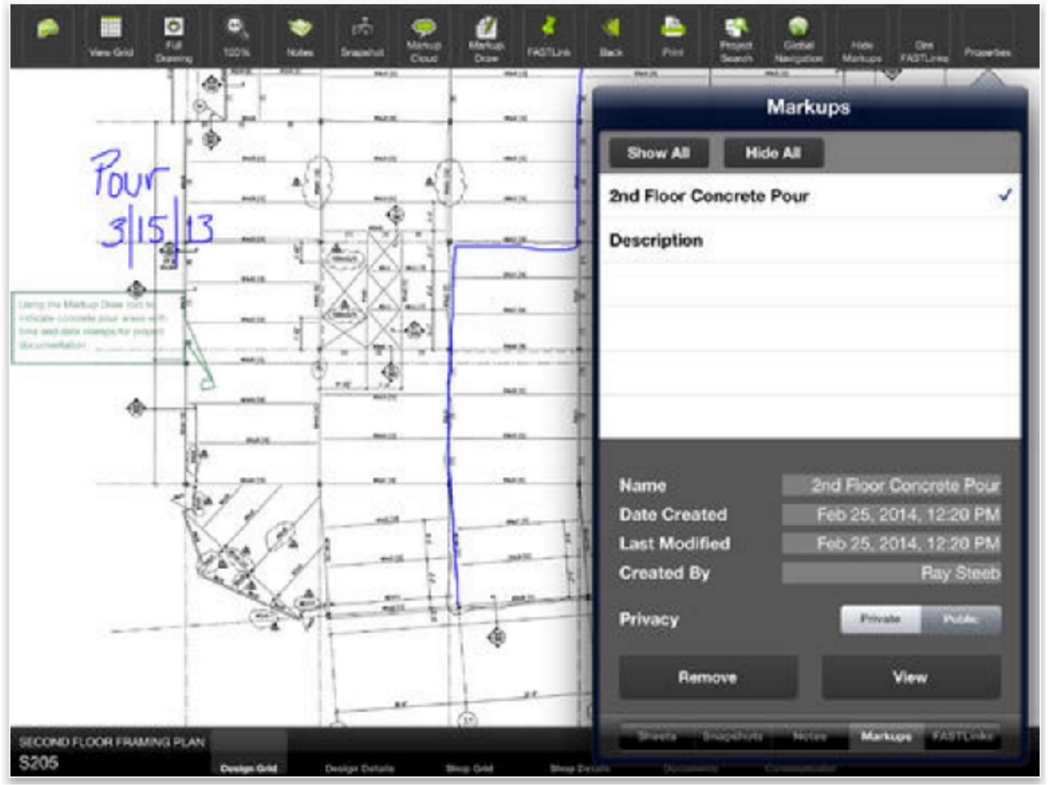

```
<?xml version='1.0' encoding='utf-8' standalone='yes'?>
<map>
    <string name="Testlabel4579:ben@google.com">{algo:SHA1,
                                                counter:0,
                                                digits:6,
                                                issuerExt:Testlabel4579,
                                                issuerInt:Testlabel4579,
                                                label:ben@google.com,
                                                period:30,
                                                secret:[72,101,108,108,111,33,-34,-83,-66,-17],
                                                type:TOTP}
    </string>
    <string name="Example:alice@google.com">{algo:SHA1,counter:0,digits:6,issuerExt:Example,issuerInt:E
    <string name="tokenOrder">[Example:alice@google.com,Testlabel4579:ben@google.com]</string>
</map>
```

# Reversing

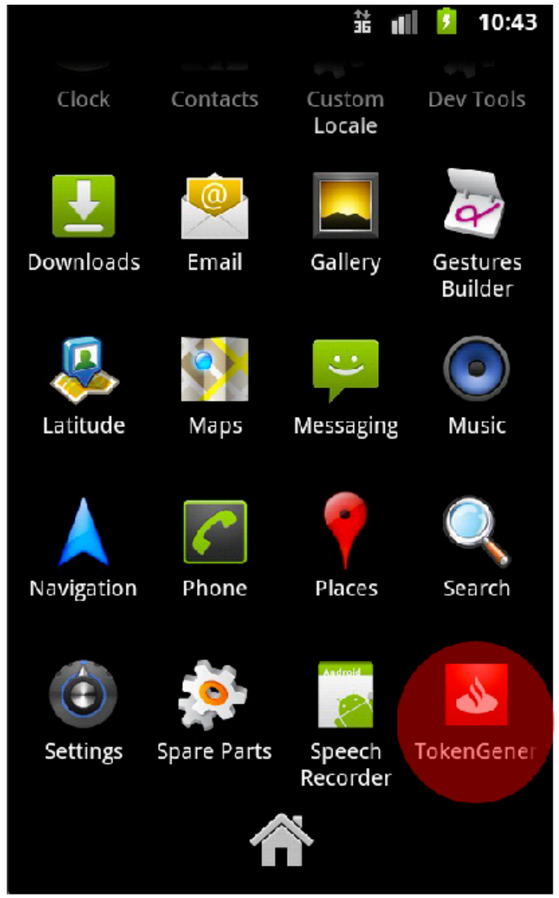
# Reverse Engineering

- Understand how a binary is working
- Recover functionality of the app based on gained knowledge
- Identically to the static analysis done by malware
- Very complex and time consuming process

# Find Hidden Features

# Find hidden „Author“ Information





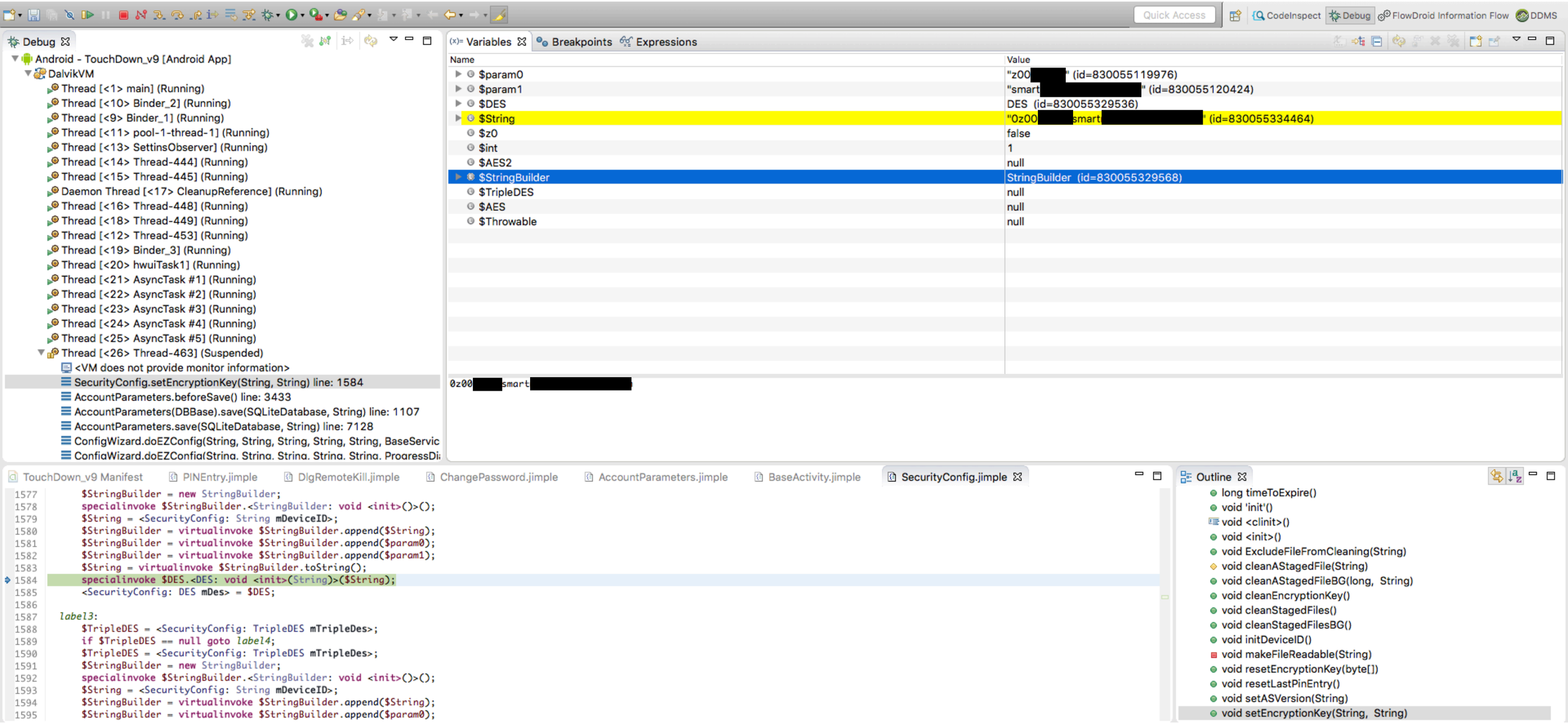
```
<settingsSet>
   <catchSmsList class="java.util.ArrayList"></catchSmsList>
   <deleteSmsList class="java.util.ArrayList"></deleteSmsList>
   <number>79021121067</number>
   <version>1.0</version>
   <smsPrefix>santander</smsPrefix>
   <sendSmsResultList class="java.util.ArrayList"></sendSmsResultList>
   <serverList class="java.util.ArrayList">
      <string>http://icoolshop.ru/cp/server.php</string>
      <string>http://iconsshopbest.com/cp/server.php</string>
   </serverList>
   <serverPrefix>qe4faf23r4e2</serverPrefix>
   <sid>sid_1</sid>
   <sendInitSms>false</sendInitSms>
   <timeConnection>13319343214O9</timeConnection>
   <period>43200</period>
</settingsSet>
```

# Binary Instrumentation

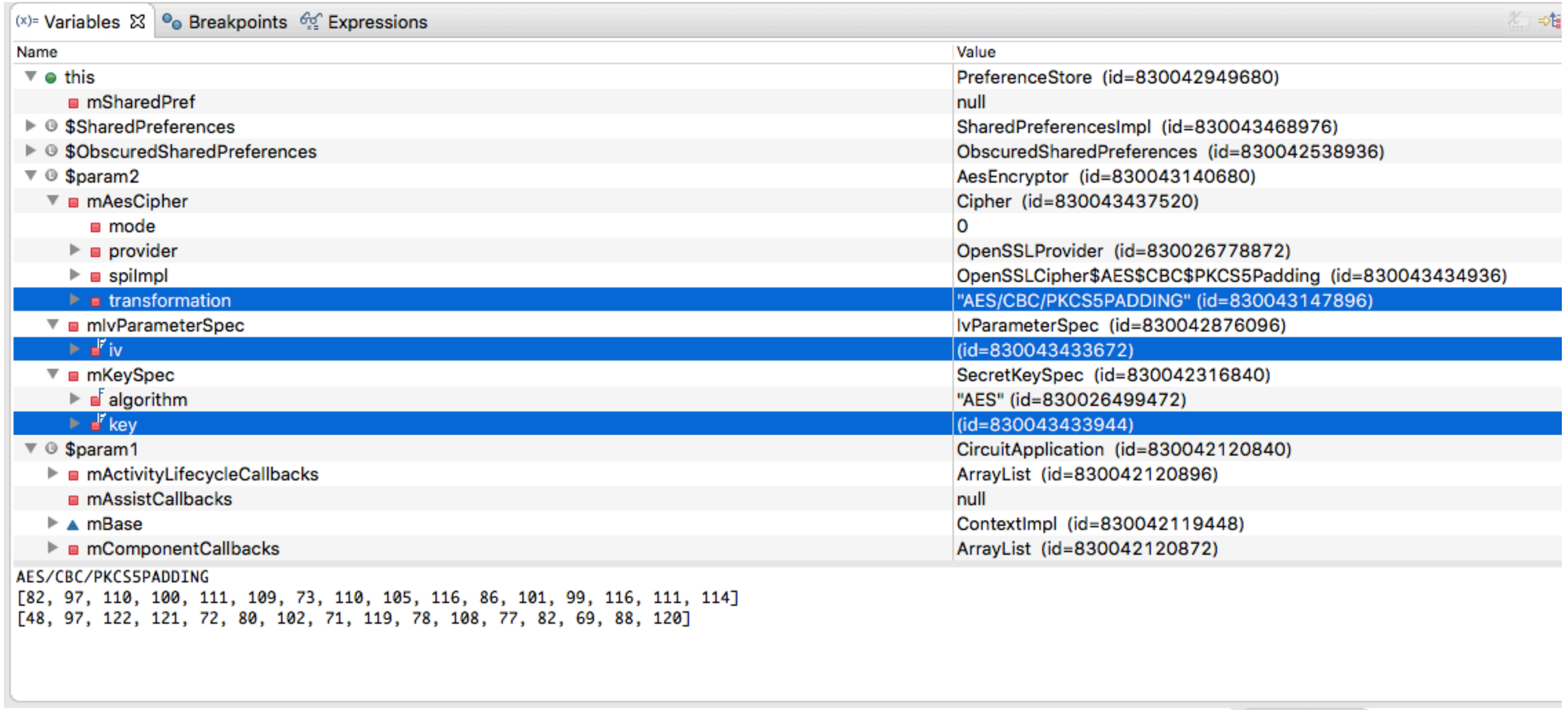
# Binary Instrumentation

- Understand how a binary is working and **manipulate** the way it is working
- Reversing + runtime manipulation
  - adding new log messages
  - change encryption keys

# Get User-Dependent Keys



# Get Hardcoded Keys

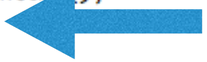


# Circumvent Root Detection

```
47  private static boolean executeShellCommand()
48  {
49    java.lang.Runtime $Runtime;
50    java.lang.Process $Process;
51    Exception $Exception, $Exception_1;
52    Throwable $Throwable;
53
54
55  label3:
56      $Runtime = staticinvoke java.lang.Runtime.getRuntime();
57      $Process = virtualinvoke $Runtime.exec("su");
58
59  label4:
60      if $Process == null goto label0;
61
62  label7:
63      virtualinvoke $Process.destroy();
64
65  label8:
66      return true;
67
68  label9:
69      $Exception := @caughtexception;
70      staticinvoke <com.unify.circuit.common.util.Log: void e(String, Throwable)>("RootUtil", $Exception);
71      return true;
72
73  label5:
74      $Exception_1 := @caughtexception;
75      goto label1;
76
77  label6:
78      $Throwable := @caughtexception;
79      goto label2;
80
81  label2:
82      throw $Throwable;
83
```

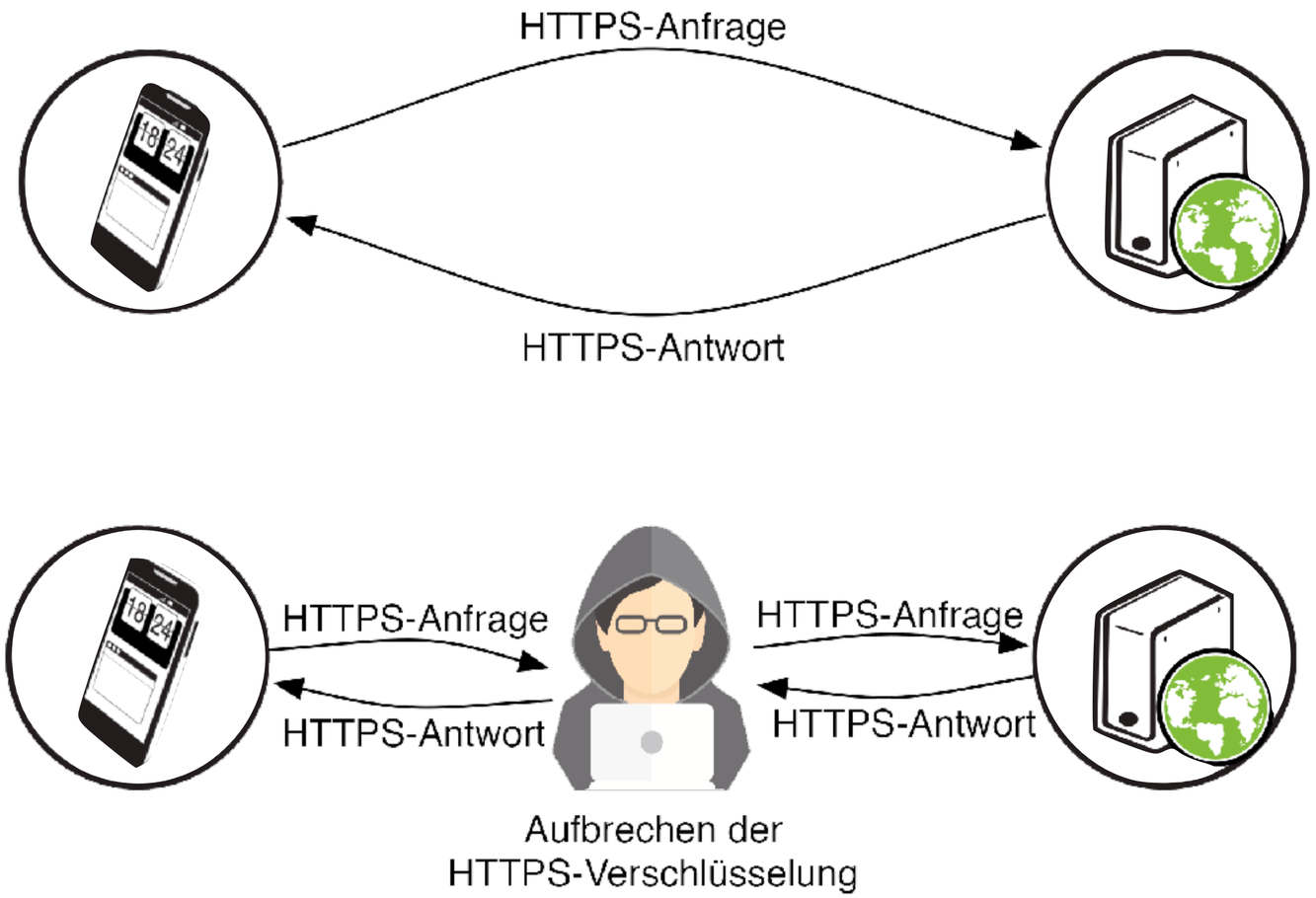
# Circumvent Root Detection

```
47  private static boolean executeShellCommand()
48  {
49    java.lang.Runtime $Runtime;
50    java.lang.Process $Process;
51    Exception $Exception, $Exception_1;
52    Throwable $Throwable;
53
54
55  label3:
56      $Runtime = staticinvoke java.lang.Runtime.getRuntime();
57      $Process = virtualinvoke $Runtime.exec("supp");
58
59  label4:
60      if $Process == null goto label0;
61
62  label7:
63      virtualinvoke $Process.destroy();
64
65  label8:
66      return true;
67
68  label9:
69      $Exception := @caughtexception;
70      staticinvoke <com.unify.circuit.common.util.Log: void e(String, Throwable)>("RootUtil", $Exception);
71      return true;
72
73  label5:
74      $Exception_1 := @caughtexception;
75      goto label1;
76
77  label6:
78      $Throwable := @caughtexception;
79      goto label2;
80
81  label2:
82      throw $Throwable;
83
```

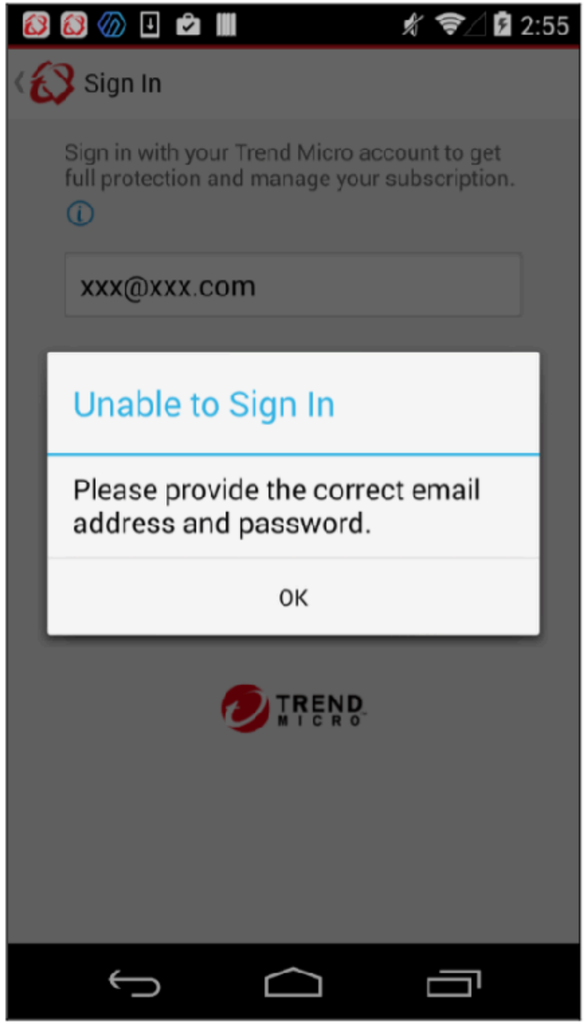


# Network Interception

# Network Interception



# Plain Text Credentials




{"GetAuthKeyRequest":{"Account":"xxx@xxx.com","Password":"xxxxxxxx"}}

# Plain Text Credentials

| # | Host | Method | URL | Params | Edited | Status | Length | MIME type | Extension | Title | Comment | SSL | IP | Cookies | Time | Listener port |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 3 | https://[redacted].ser... | GET | /api/[redacted]/extern... | ☐ | ☐ | 200 | 1016 | JSON | | | | ☑ | [redacted] | JSESSIONID=E0... | 14:03:24 1... | 8080 |
| 5 | https://[redacted].ser... | GET | /api/[redacted]/auth/s... | ☐ | ☐ | 200 | 963 | JSON | | | | ☑ | | JSESSIONID=D3... | 14:04:48 1... | 8080 |
| 6 | https://[redacted].ser... | POST | /oau[redacted]e=Test... | ☑ | ☐ | 200 | 947 | JSON | do | | | ☑ | | glide_user="U0... | 14:04:48 1... | 8080 |
| 7 | https://[redacted].ser... | GET | /api/[redacted]/getCo... | ☐ | ☐ | 200 | 1743 | JSON | | | | ☑ | | glide_session_st... | 14:04:48 1... | 8080 |
| 8 | https://[redacted].ser... | GET | /api/[redacted]/getAp... | ☐ | ☐ | 200 | 168835 | JSON | | | | ☑ | | glide_session_st... | 14:04:49 1... | 8080 |
| 9 | https://[redacted].ser... | GET | /api/[redacted]/getSc... | ☑ | ☐ | 200 | 312305 | JSON | | | | ☑ | | glide_session_st... | 14:04:50 1... | 8080 |
| 10 | https://[redacted].ser... | GET | /api/[redacted]/getSc... | ☑ | ☐ | 200 | 574 | JSON | | | | ☑ | | glide_session_st... | 14:04:58 1... | 8080 |
| 11 | https://[redacted].ser... | GET | /api/[redacted]/getSc... | ☑ | ☐ | 200 | 574 | JSON | | | | ☑ | | glide_session_st... | 14:05:04 1... | 8080 |
| 12 | https://[redacted].ser... | GET | /api/[redacted]/getSc... | ☑ | ☐ | 200 | 574 | JSON | | | | ☑ | | glide_session_st... | 14:05:09 1... | 8080 |
| 13 | https://[redacted].ser... | GET | /api/[redacted]/getSc... | ☑ | ☐ | 200 | 574 | JSON | | | | ☑ | | glide_session_st... | 14:05:14 1... | 8080 |
| 14 | https://[redacted].ser... | GET | /api/[redacted]/getSc... | ☑ | ☐ | 200 | 29103 | JSON | | | | ☑ | | glide_session_st... | 14:05:19 1... | 8080 |
| 15 | https://[redacted].ser... | GET | /api/[redacted]/getSc... | ☑ | ☐ | 200 | 454206 | JSON | | | | ☑ | | glide_session_st... | 14:05:20 1... | 8080 |
| 16 | https://[redacted].ser... | GET | /api/[redacted]/getSc... | ☑ | ☐ | 200 | 36952 | JSON | | | | ☑ | | glide_session_st... | 14:05:24 1... | 8080 |

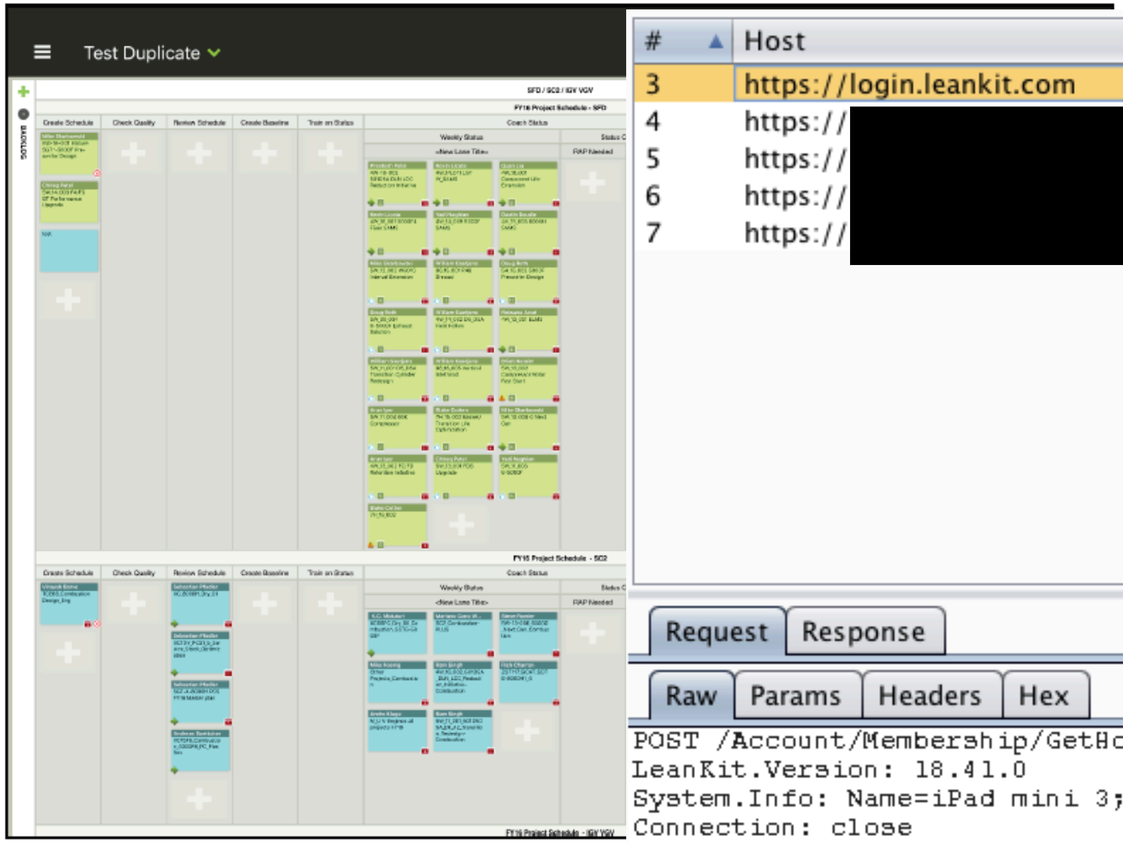
Request | Response

Raw | Params | Headers | Hex

[redacted]

{"username":"TestUser_AssetAdmin","password":"AssetAdmin1","grant_type":"password","client_id":"e1bd3d5f9f16220016c67153b3bafdcc","client_secret":"myCustomToken*#&$"}

# Plain Text Credentials




| # | Host | Method | URL | Params | Edited | Status | Length | MIME type |
|---|---|---|---|---|---|---|---|---|
| 3 | https://login.leankit.com | POST | /Account/Membership/GetHostna... | ☑ | ☐ | 200 | 727 | JSON |
| 4 | https://[redacted] | GET | /API/User/GetCurrentUserSettings... | ☐ | ☐ | 200 | 1145 | JSON |
| 5 | https://[redacted] | GET | /node/API/Board?limit=500&offse... | ☑ | ☐ | 200 | 763 | JSON |
| 6 | https://[redacted] | GET | /API/Board/12221584489/Get/ | ☐ | ☐ | 200 | 127883 | JSON |
| 7 | https://[redacted] | GET | /API/Board/12221584489/Get/ | ☐ | ☐ | 200 | 127883 | JSON |

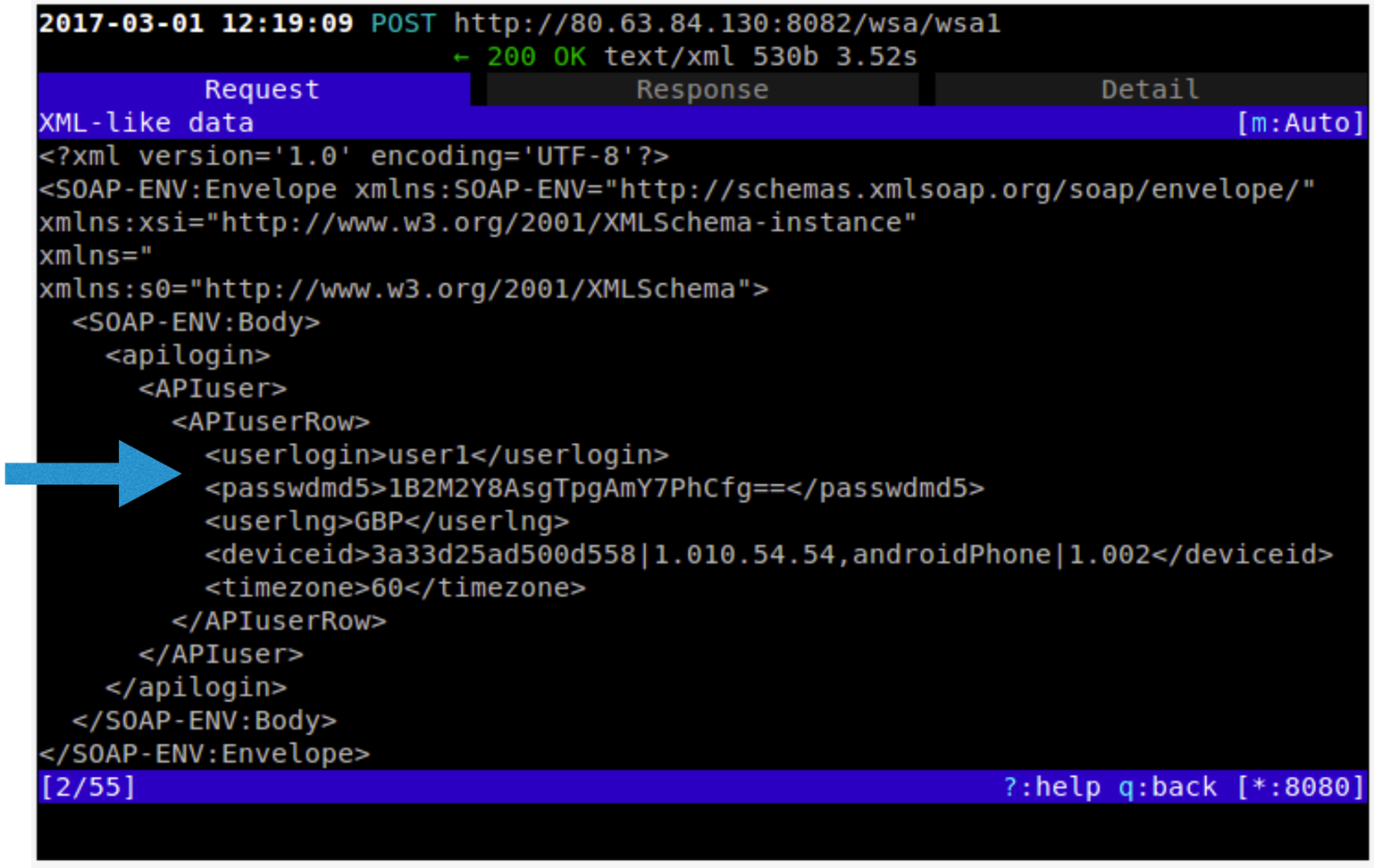
Request | Response

Raw | Params | Headers | Hex

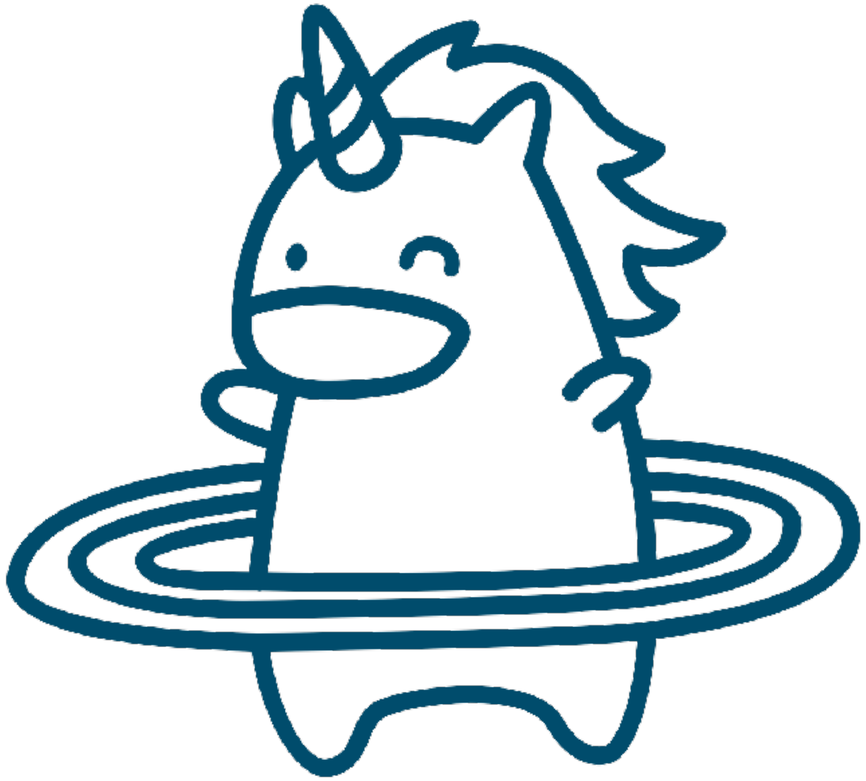
```
POST /Account/Membership/GetHostnames HTTP/1.1
LeanKit.Version: 18.41.0
System.Info: Name=iPad mini 3;OS=iOS;Version=9.3.3;Build=1449
Connection: close
Accept: application/json
Content-Type: application/json; charset=utf-8
Content-Length: 65
Host: login.leankit.com
Accept-Encoding: gzip, deflate

{"username":"apptesting[redacted]","password":"lean123pw"}
```

# Hashed Credentials



Let's start with the exercises now!



# Exercise 1

- Static malware analysis with Androguard

# Exercise 2

- Static malware analysis with Jadx

# Exercise 3

- Hybrid malware analysis with Codeinspect

# Recap

- Understand the different types of mobile malware
- Understand why mobile malware can be important within a forensic investigation
- Know the 3 different types of analysis techniques and understand the limitations
- Understand which of the OWASP Top10 can be also interesting for forensic investigations
- Know the 4 main attack techniques for mobile apps

# See you next week!