Enhancing Mobile Malware: an Android RAT Case Study

BSIDES VIENNA 2014
November 22

About

Marco Lancini

Security Consultant, CEFRIEL
@lancinimarco

Roberto Puricelli

Security Consultant, CEFRIEL
@robywankenoby

Introduction

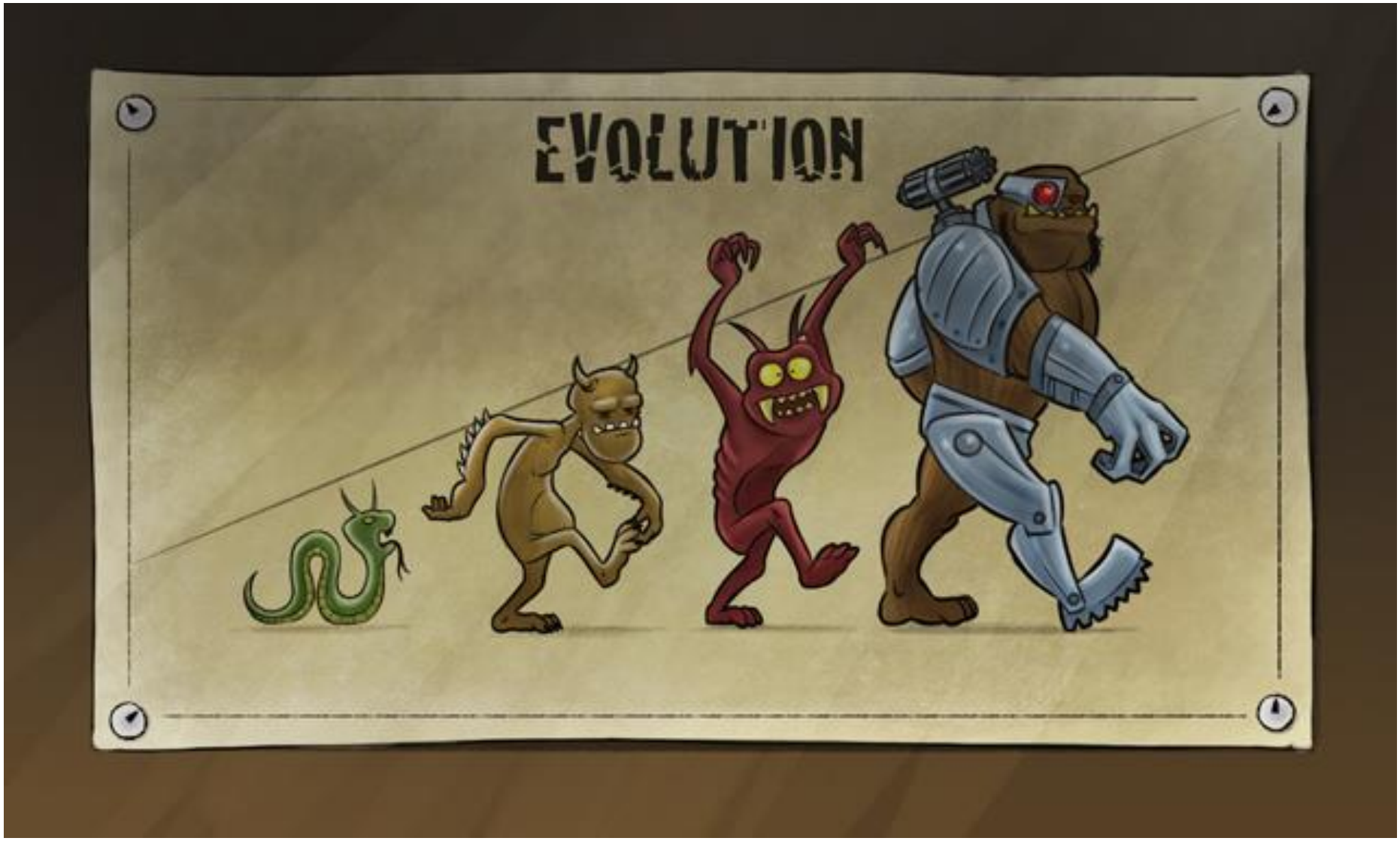
Intro

*Demonstrate how it is possible to **easily create powerful malware**, combining public available attack toolkits and exploits of known vulnerabilities*

*Given the source code of a mobile RAT, it is possible to **extend its features**, adapting and modifying its behavior (hiding malicious features, adding exploits)*

***AndroRAT++**, a proof-of-concept mobile malware, embedded in a legitimate application, that enhances the features of a well-know RAT application*

Mobile malware evolution



Mobile malware evolution

Mobile malware is a (relatively) new trend

- Actually almost 10 years of samples

[1] http://www.sophos.com/en-us/medialibrary/PDFs/other/sophos-mobile-security-threat-report.pdf

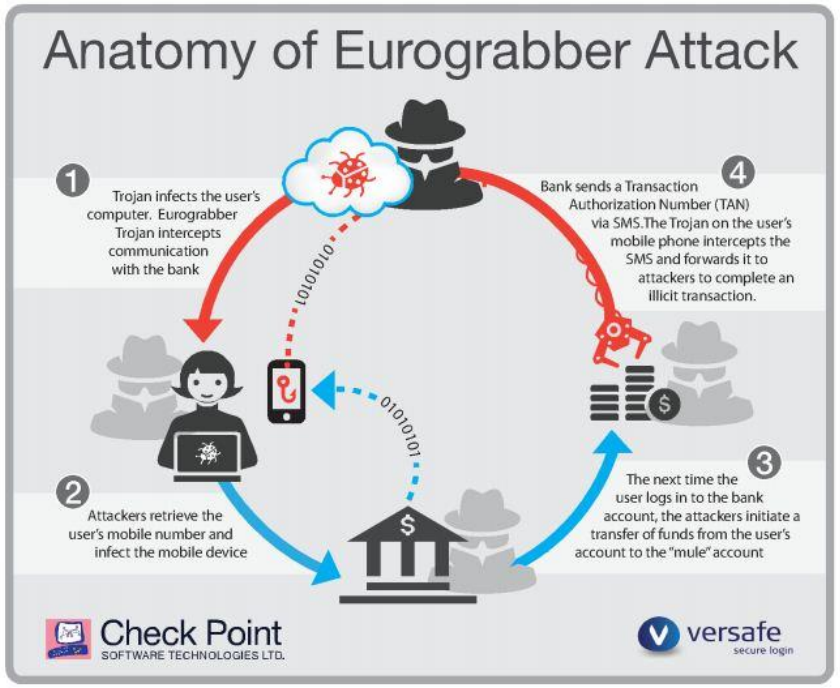
DroidDream

- **Infected 60 different legitimate apps** in the Android Market
- **Breached the Android security sandbox**, installed additional software, and stole data
- Created a botnet

2011

DroidDream
First large attack to Google Play market. Over 50 apps containing a root exploit published to Android Market.

Zitmo

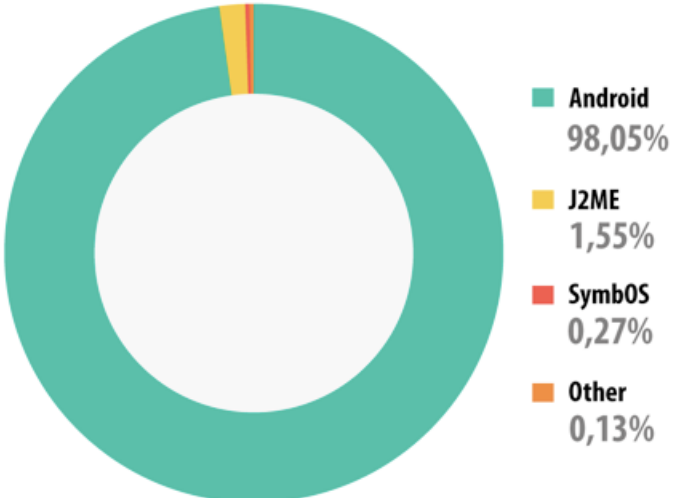


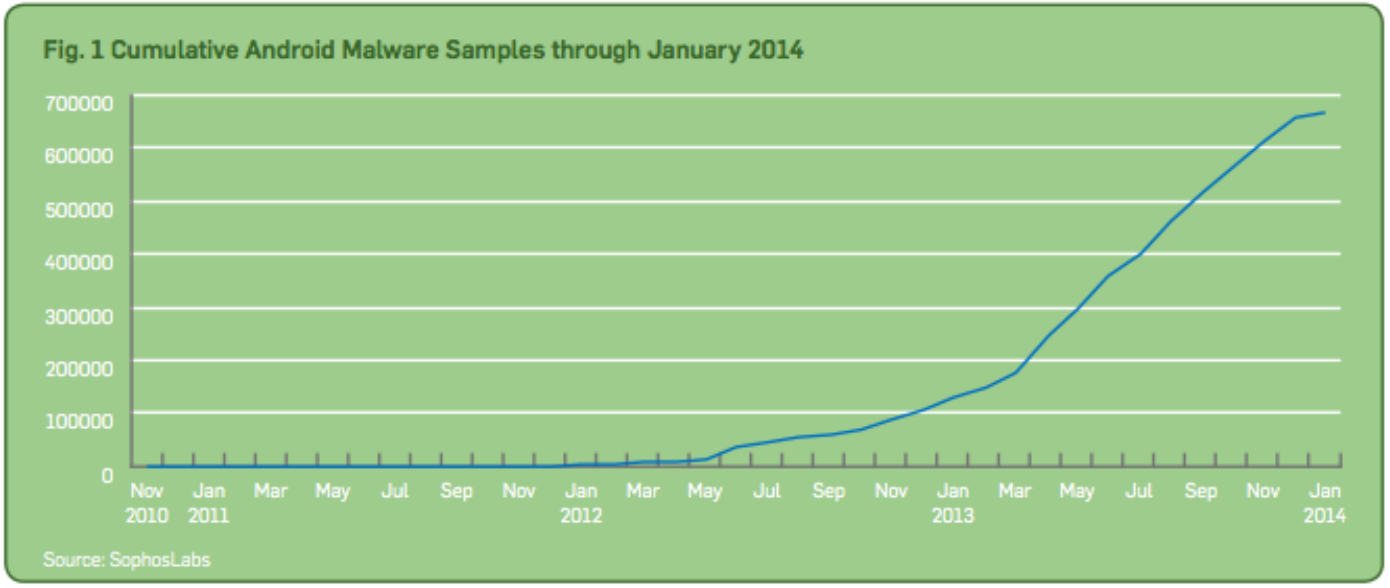
- A.k.a. **Eurograbber**
- Widespread in Europe
- Bypass 2FA (SMS OTP)
- **36M € stolen**

Android is the prime target

Why Android is the most targeted platform?

- Wide-spread
- “Open” philosophy
- Lacks of controls





[1] http://blog.kaspersky.com/mobile-malware-evolution-2013/

How to get compromised?

Social engineering plays a big role in the exploit

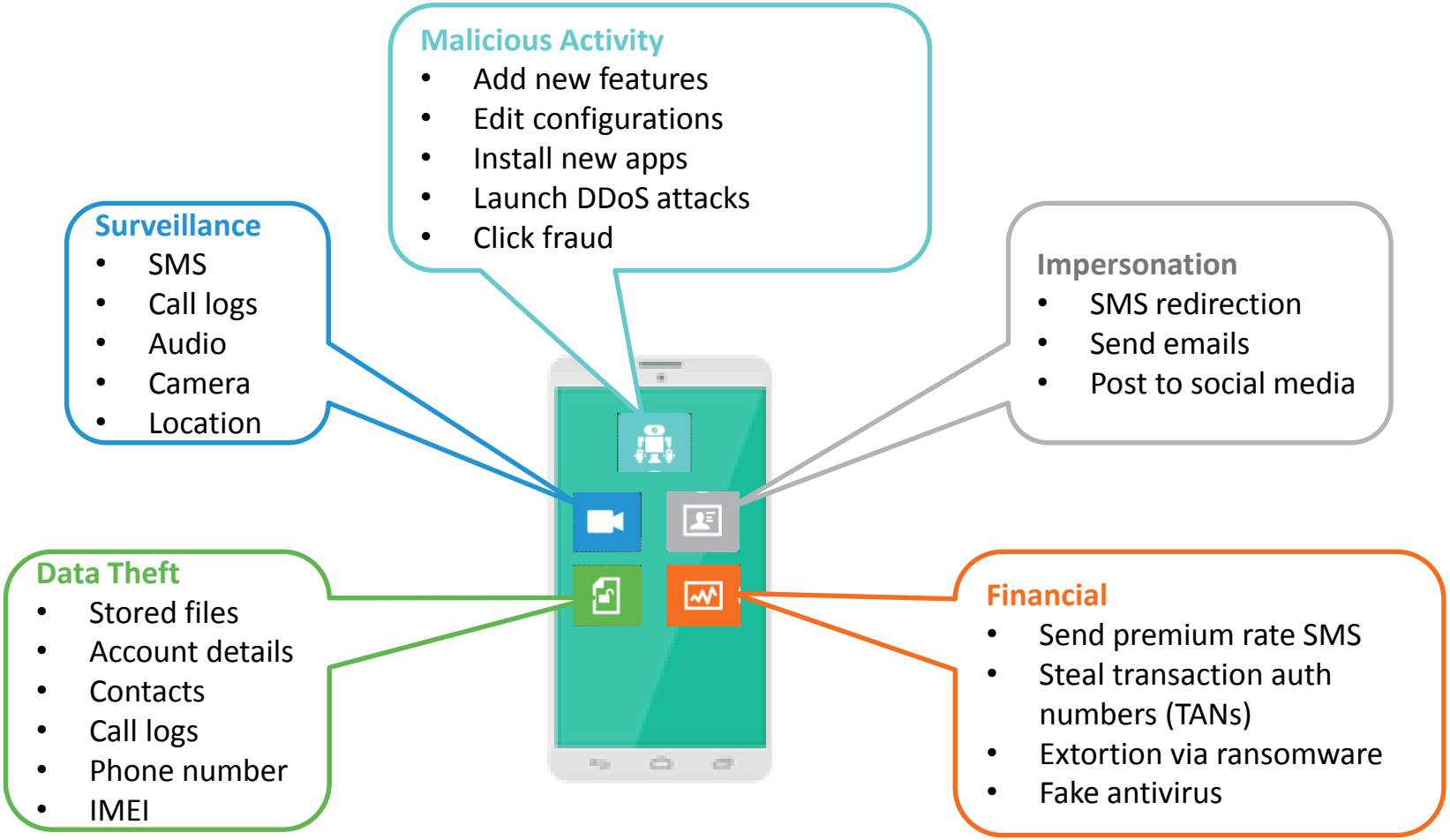
- By installing a trojan app that perform unauthorized operations
- The malware is "**embedded in the app**"

Anzhi Market

Renowned for not making controls over published applications

Used to spread malicious applications disguised as famous ones

What can an attacker do?



[1] https://www.f-secure.com/documents/996508/1030743/Mobile_Threat_Report_Q1_2014_print.pdf

How to build a powerful malware?

The cutting edge of mobile malware

What's new in Android Malware?

HIGHLIGHTS THIS QUARTER

ON ANDROID

With 99% of the new threats that emerged in Q1 2014 designed to run on the Android operating system (OS), it's not surprising the most interesting mobile malware technical developments involved this platform. Here are a few noteworthy advances seen in Android malware in the last few months:

WINDOWS TROJAN HOPS ON ANDROID [1]

A banking-trojan named **Droidpak** that targets Windows PCs also tries to install a mobile banking-trojan on any Android devices connected via USB to the infected machine. Depending on the variant, we detect the mobile banking-trojan used as **Trojan-Spy:Android/Smforw.H** or **Trojan:Android/Gepew.A** or **.B**).

FIRST TOR TROJAN [2]

Trojan:Android/Torsm.A is the first trojan on this platform to leverage the open-source Orbot client for the popular Tor anonymizing network to communicate with its C&C server, making it difficult (if not impossible) for researchers and law enforcement to track and shut down the C&C.

FIRST CRYPTOMINER [3]

Trojan:Android/CoinMiner.A is distributed in a repackaged application. When installed, it essentially hijacks the device to silently mine virtual currency (such as Litecoin) for the malware author. Apart from any data charges incurred, the constant use of the device's hardware may also affect its battery life and eventual lifespan.

FIRST BOOTKIT [4]

Trojan:Android/Oldboot.A is believed to be Android's first bootkit, or malware that affects the earliest stages of the device's bootup routine, making it extremely difficult to detect or remove. The malware is thought to have spread in modified firmware updates, with most infections reportedly seen in China.

PILEUP EXPLOIT [5]

Researchers reported vulnerabilities in the Android OS that could allow an installed malware to silently upgrade its permissions during a system update, and named an exploit of this loophole **Pileup** (as in, "privilege escalation through updating").

DENDROID TOOLKIT [6]

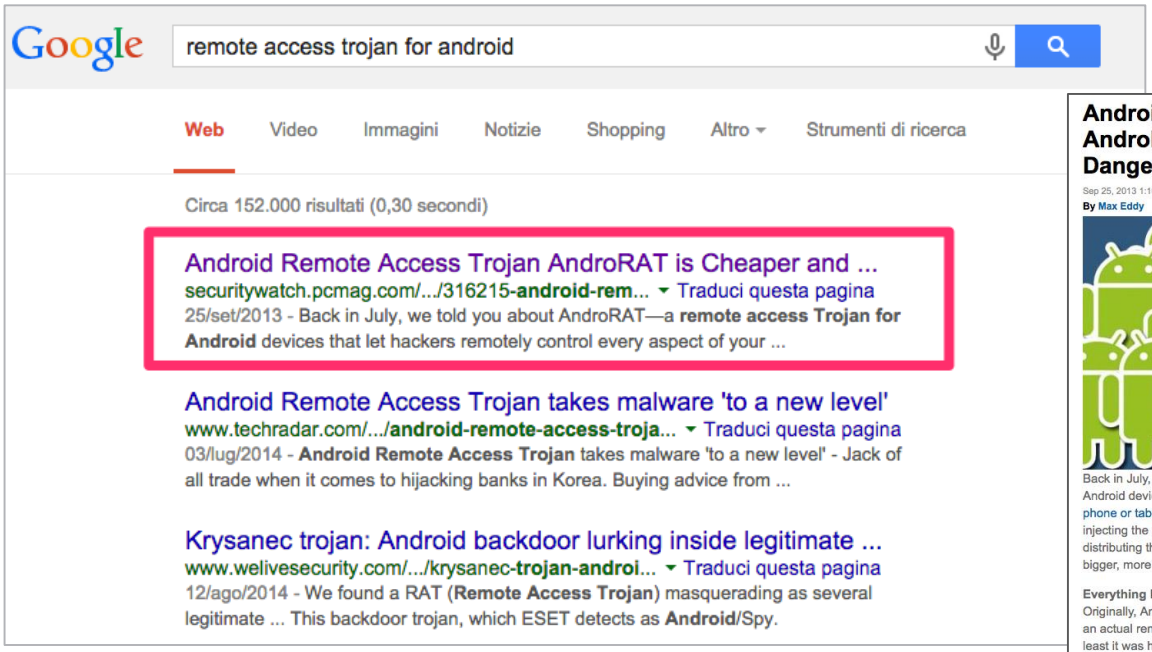
Backdoor:Android/Dendroid.A is a toolkit for creating Remote Access Trojans (RAT) that allow an attacker to create trojans that can remotely access an infected device's audio and video functions. It also creates trojans that can evade Google Play Store security.

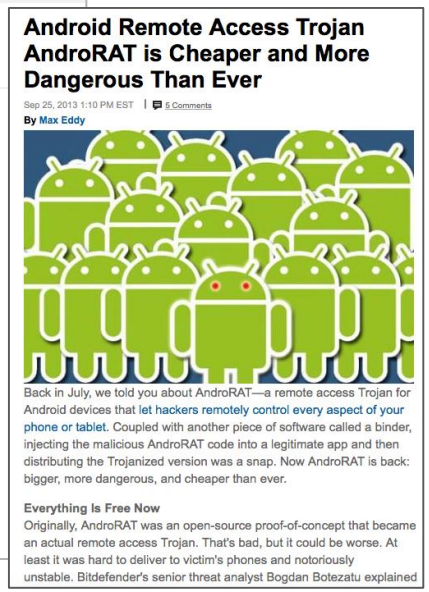
DENDROID TOOLKIT [6]

Backdoor:Android/Dendroid.A is a toolkit for creating Remote Access Trojans (RAT) that allow an attacker to create trojans that can remotely access an infected device's audio and video functions. It also creates trojans that can evade Google Play Store security.

Remote Access Trojan? Interesting, let's Google it…

Remote Access Trojan





I'm feeling lucky...

- First result gave us a possible trojan name

AndroRAT

- Open source proof of concept
- Powerful features
- "Easy like Sunday Morning"!!!!

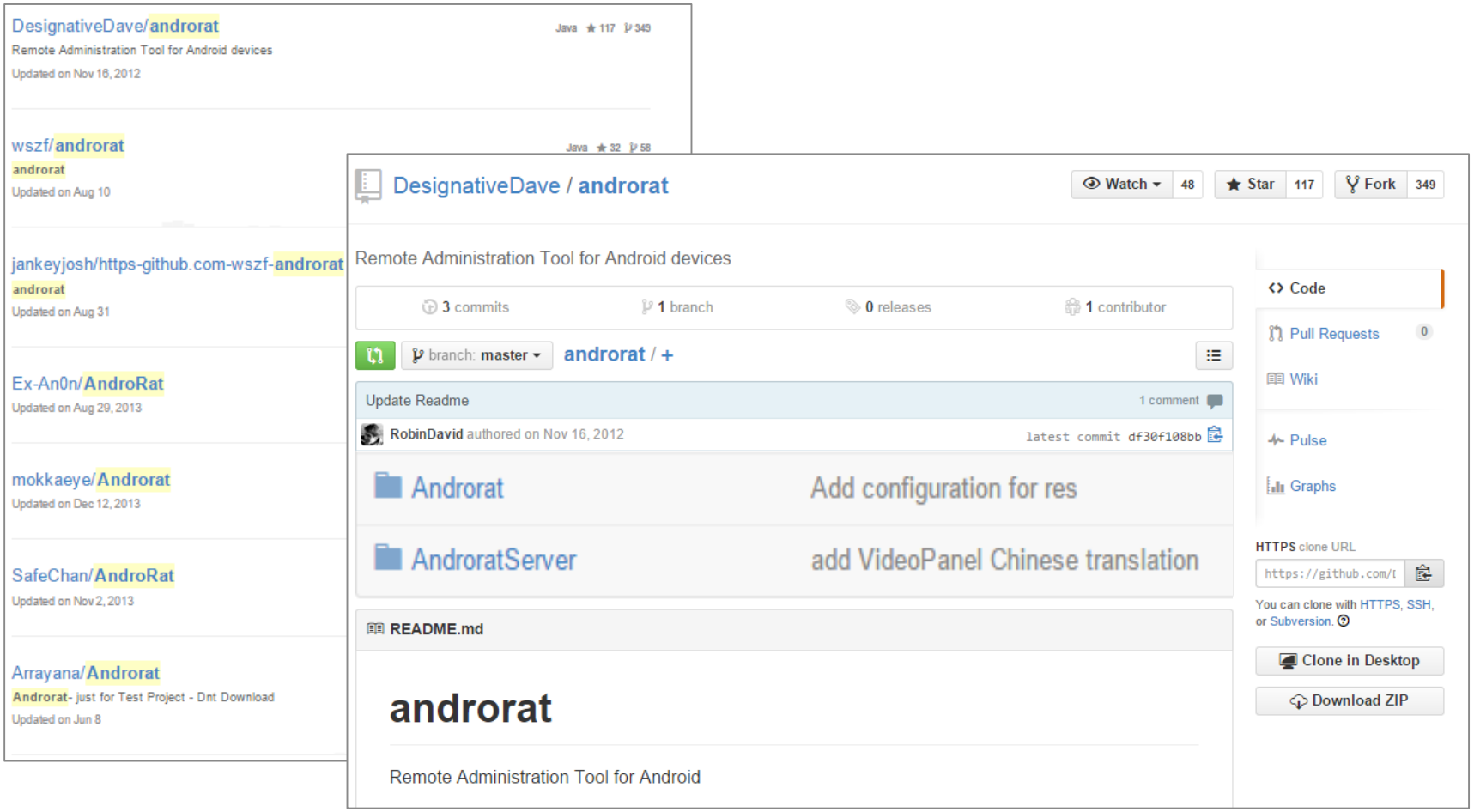
Ok, we just need to find the code…

- Let's try GitHub

AndroRAT Source Code

Still lucky…

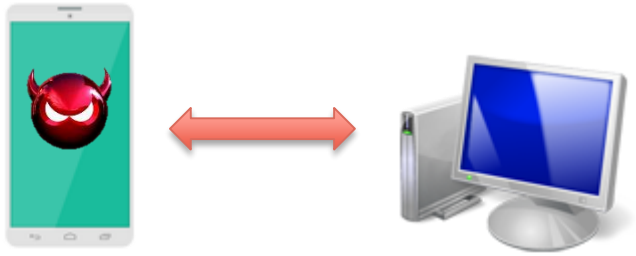
- Lots of different working versions



AndroRAT

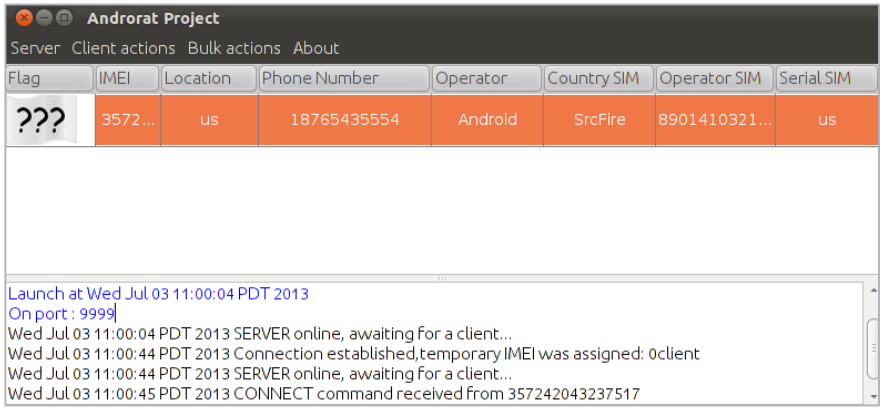
How it works

- ***Java "server"*** application
- ***Android service*** on the phone



The application itself is not so attractive

- We can embed it into another one, it's easy
- A game, or another app could be effective for our target



If we could just ***exploit the certificate validation*** in Android..

Injection of malicious code

If we could just ***exploit the certificate validation*** in Android..

Injection of malicious code

2013
1000 new Android malware samples discovered every day

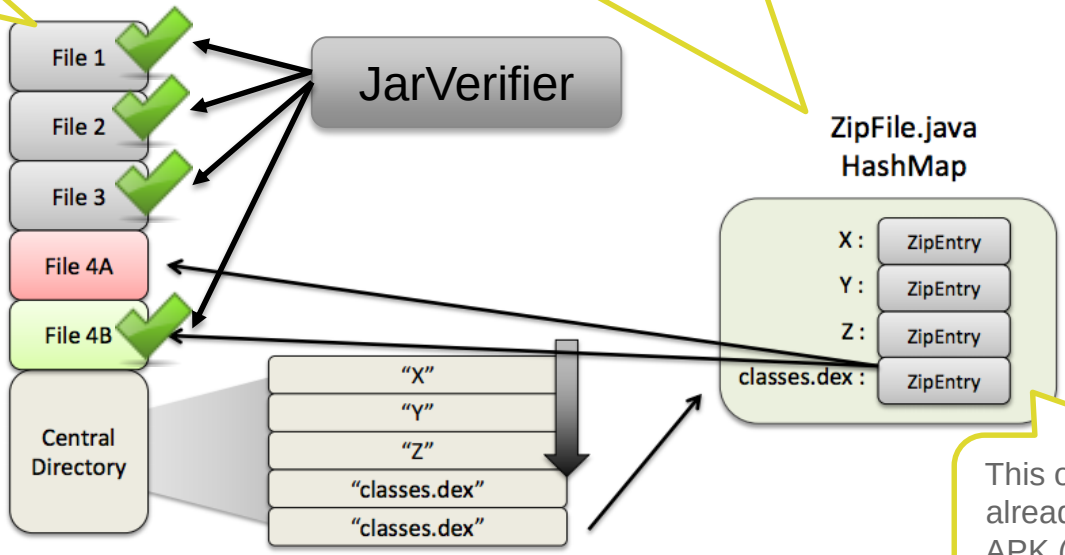
Masterkey
A vulnerability in Android discovered exploiting certificate validation in Android which allows malware to disguise as a legitimate app.

Android Master Key Vulnerability

- Allows to: "***modify APK code without breaking an application's cryptographic signature**, to turn any legitimate application into a malicious Trojan, completely unnoticed by the app store, the phone, or the end user*"
- Android can be tricked into **believing the app is unchanged** even if it has been
- Corrected with Android 4.4

It's possible to decompile an app and to inject code in it

This allows to change any of the resources contained in an APK (manifest, Java classes, graphical assets) and replace them with ones of choice

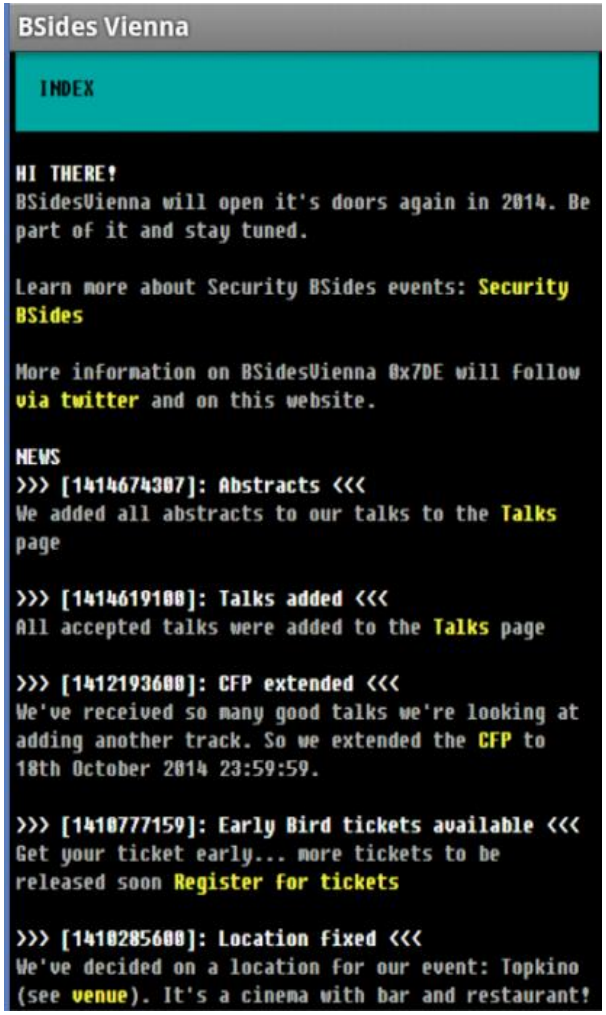


This only applies to resources already existing in the original APK (**new resources cannot be introduced**)

[1] BlackHat US 2013: http://bluebox.com/corporate-blog/bluebox-uncovers-android-master-key/

A real example…

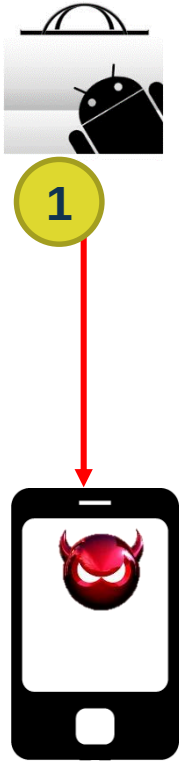
- Let's embed our RAT into a benign application
- The purpose here is to **simulate** the attack, not to do it for real..
  - **AndroRAT** has been injected into a *fake* application of BSides
    - Not available in any store ☺
  - New features were added (**AndroRAT++**)



DEMO Scenario

Scenario

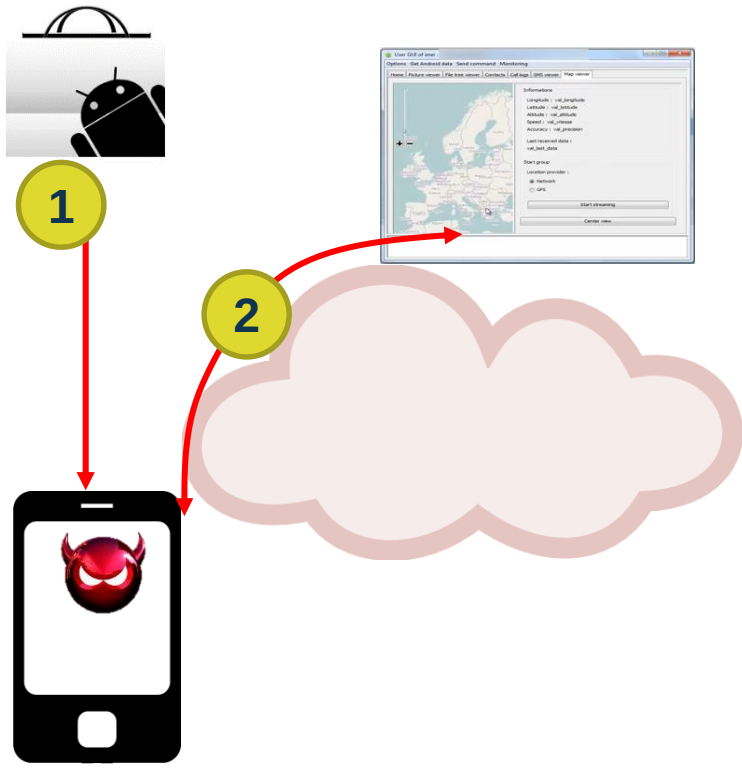
1 *Installation* of a malicious APK



Scenario

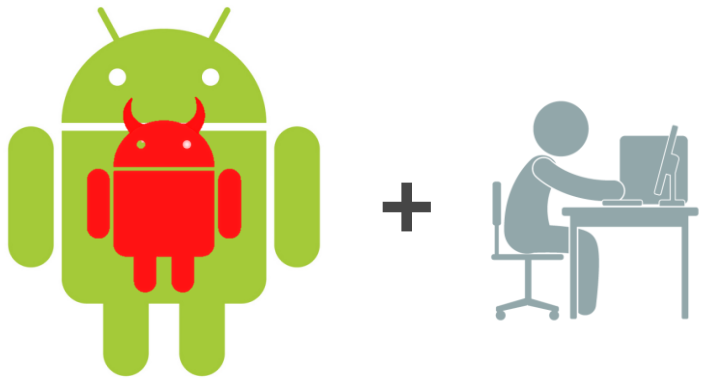
1. *Installation* of a malicious APK
2. *Remote control* of the phone

Androrat + some configurations

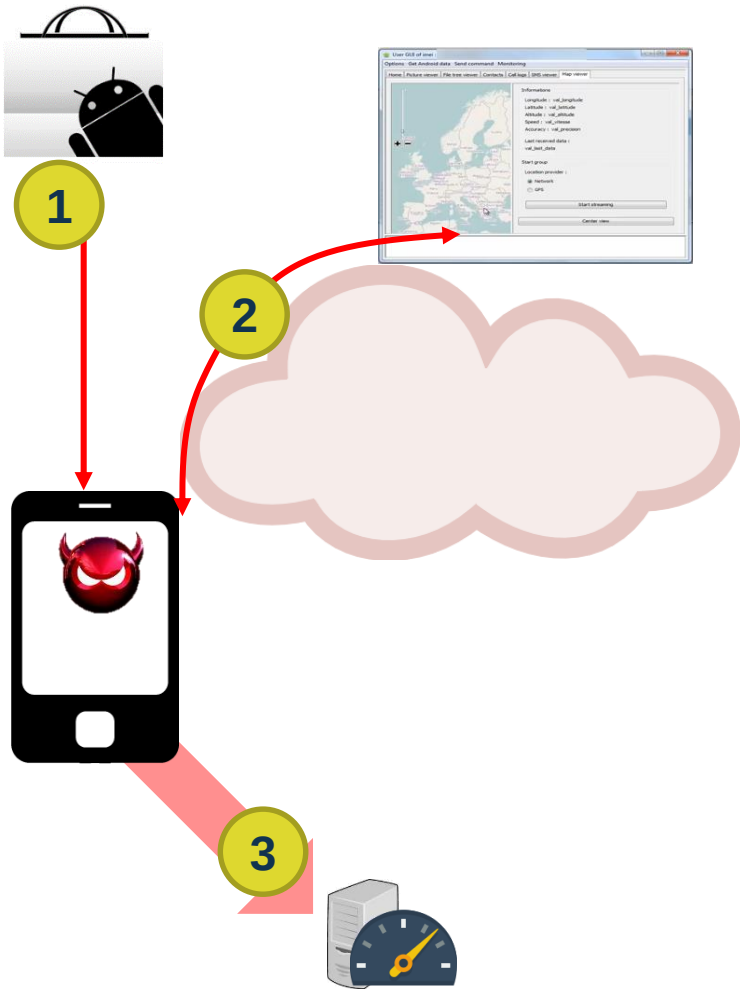


Scenario

1. *Installation* of a malicious APK
2. *Remote control* of the phone
3. Leverage the *botnet*



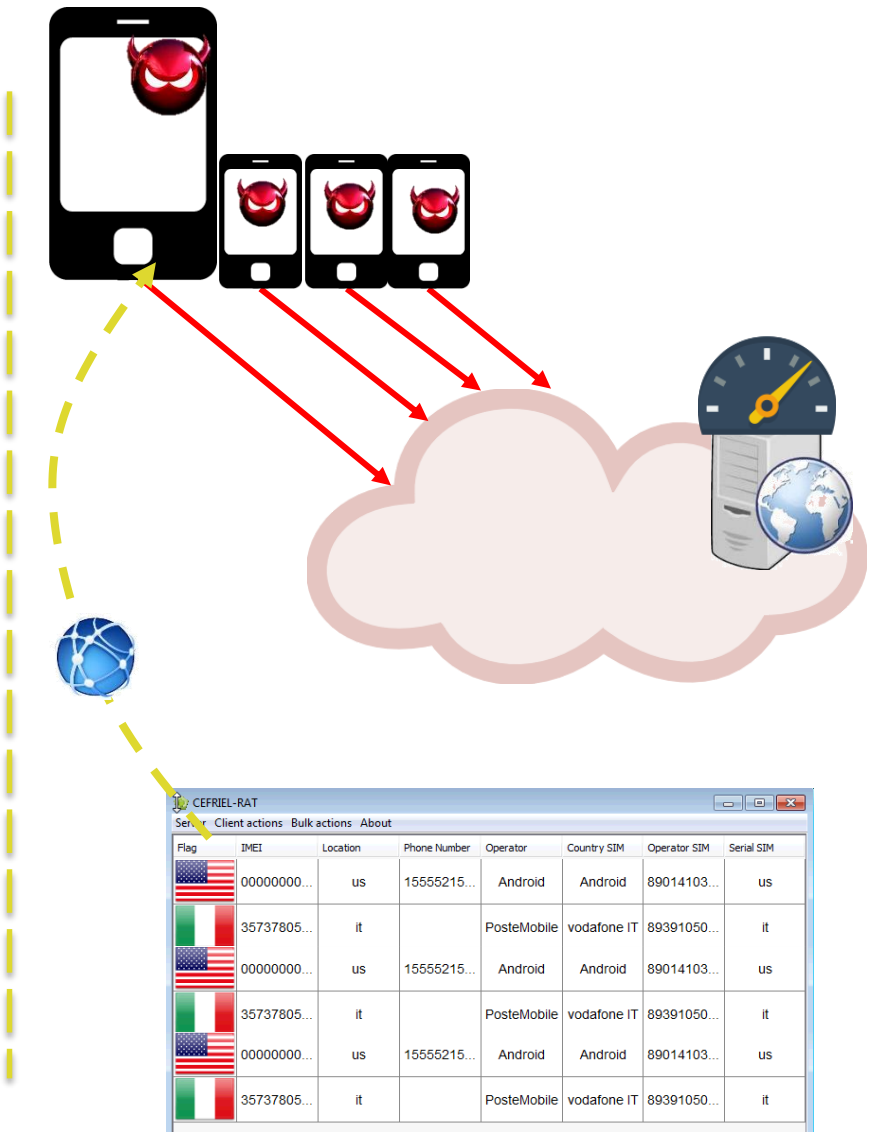
Add some coding



Denial of Service

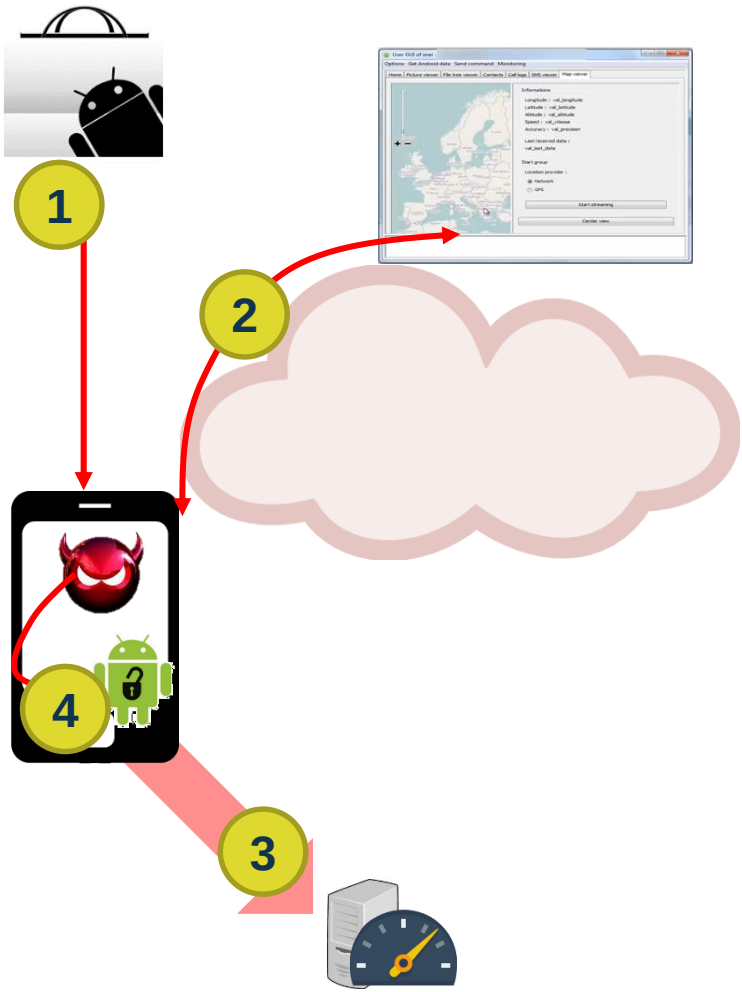
3

- Bulk actions allow to execute a command **on all the controlled devices**
- If the attacker compromises a large number of devices, a **botnet** is created
- The resources of infected devices could be used to carry out attacks on third-party services



Scenario

1. *Installation* of a malicious APK
2. *Remote control* of the phone
3. Leverage the *botnet* (DoS attacks)
4. Privilege *escalation*

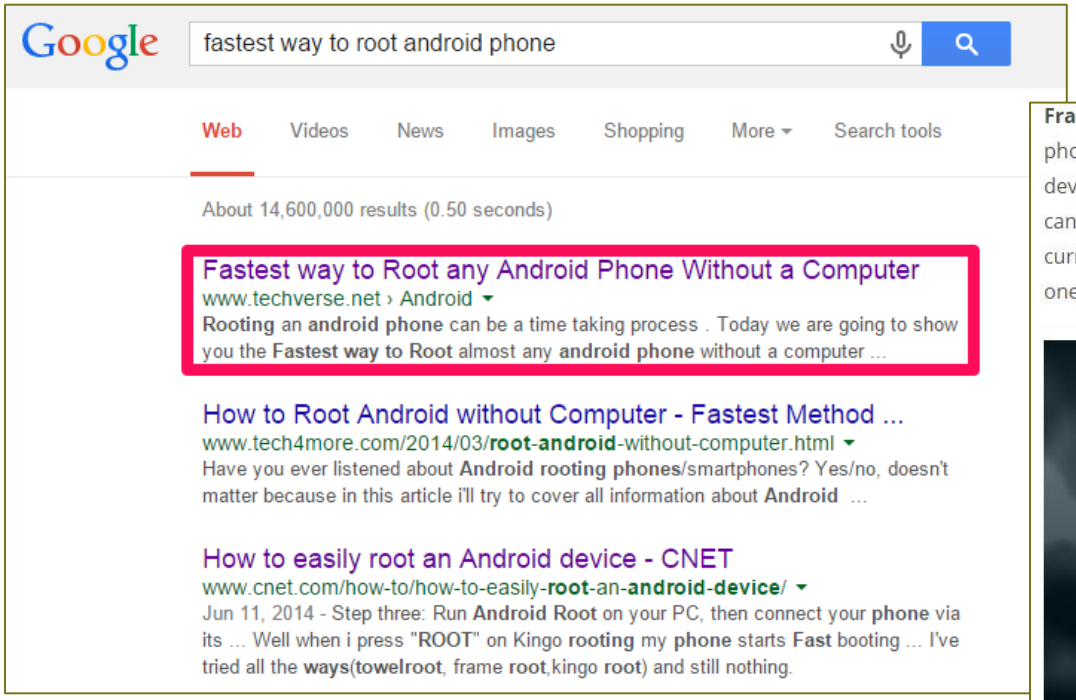


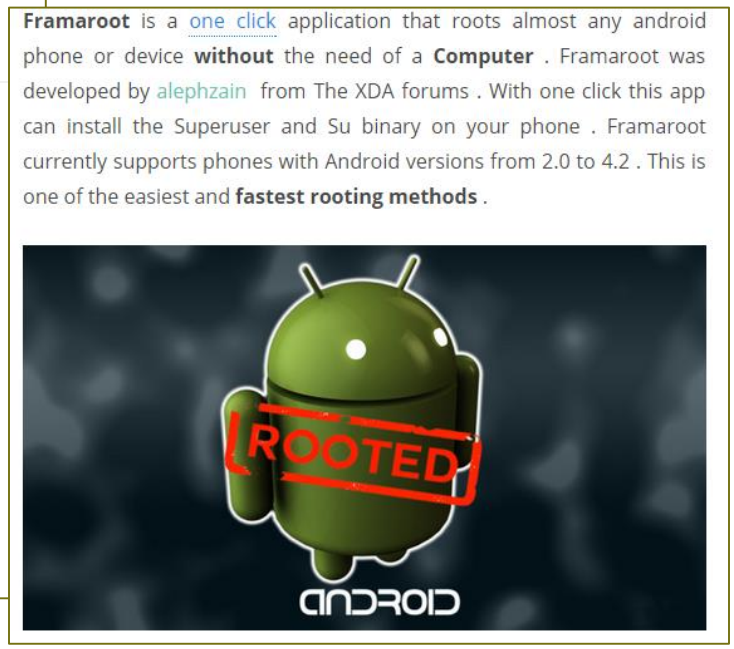
We need more… root power!

…but how? Let's find an easy way

Privilege escalation

4





I'm feeling lucky (AGAIN!!!!)...

- First result gave us an application that can easily root an Android phone

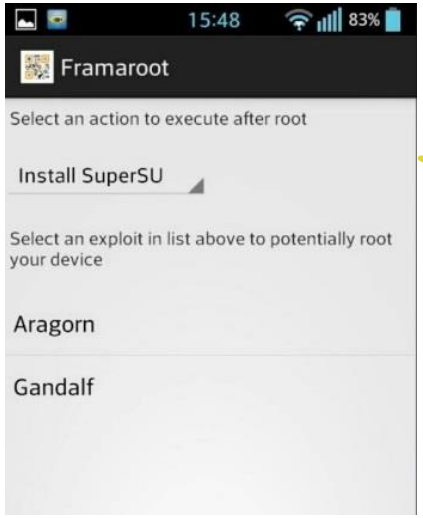
Framaroot

- Not open source, but we can get the APK from XDA
- One-click root
- Works from Android 2.0 to 4.2…good enough!

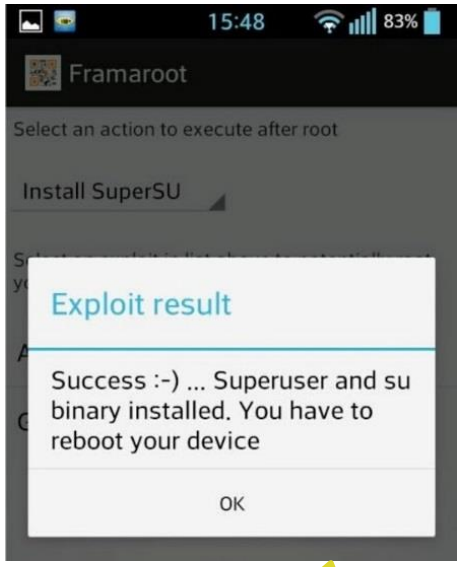
[1] http://forum.xda-developers.com/apps/framaroot/root-framaroot-one-click-apk-to-root-t2130276

Framaroot

- We can also embed the exploits used by Framaroot within the RAT application….
- The embedded version is "**silent**"
- The attacker can root the devices **remotely**



Several exploits are available in Framaroot



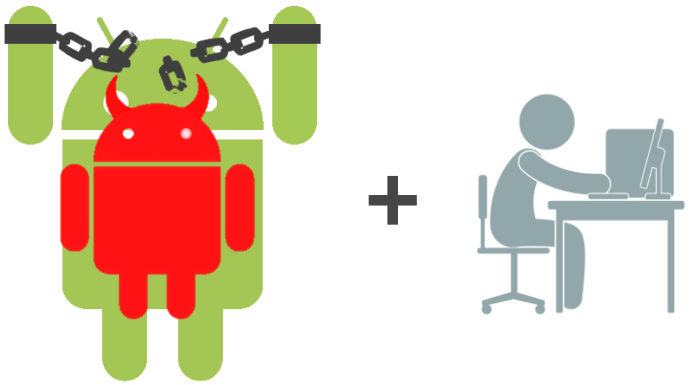
The exploit install an administrative shell

```
p = Runtime.getRuntime().exec("su");
DataOutputStream os = new DataOutputStream(p.getOutputStream());

os.writeBytes(cmd + "\n");
os.flush();

os.writeBytes("exit\n");
os.flush();

exitCode = p.waitFor();
```

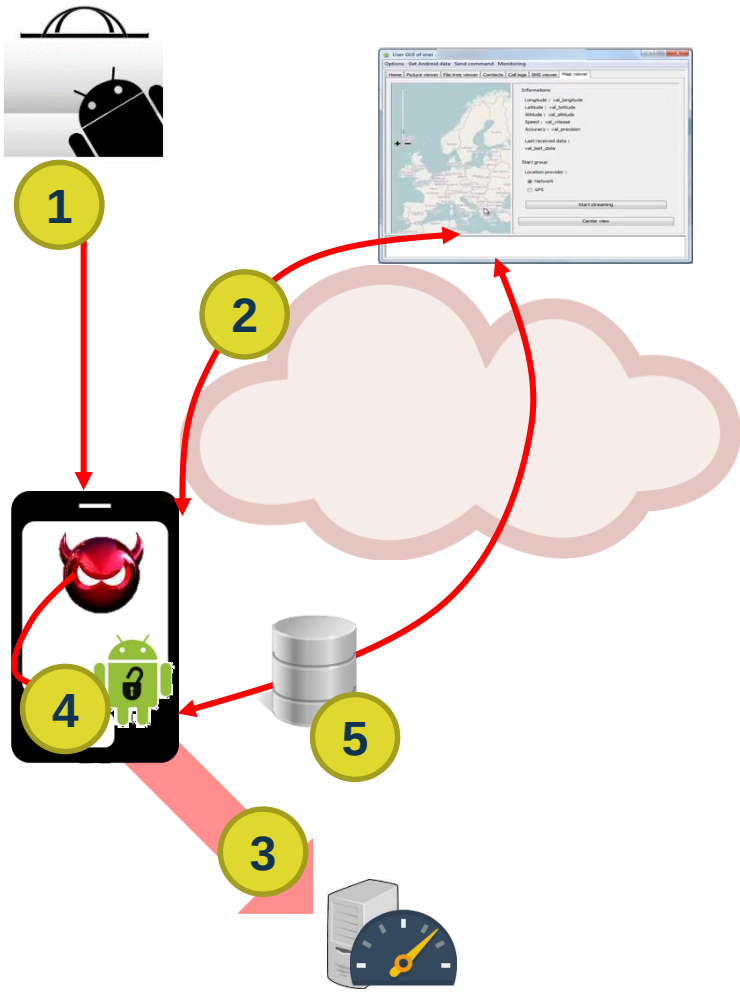
We can now execute system commands from within our code

Scenario

1. *Installation* of a malicious APK
2. *Remote control* of the phone
3. Leverage the *botnet* (DoS attacks)
4. Privilege *escalation*
5. *Exfiltration* of sensitive data

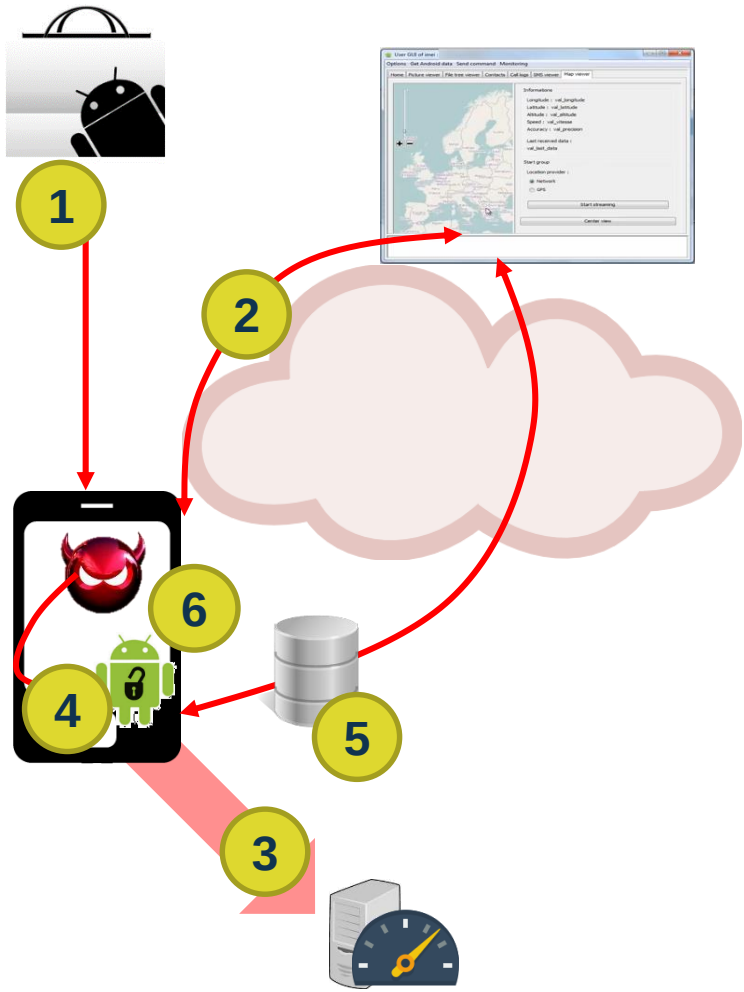


Add some more code…



Scenario

1. *Installation* of a malicious APK
2. *Remote control* of the phone
3. Leverage the *botnet* (DoS attacks)
4. Privilege *escalation*
5. *Exfiltration* of sensitive data
6. *Silent installation* of new applications



Still some code…

Which application to install?

I just have to choose the application…

- The purpose is always to make money

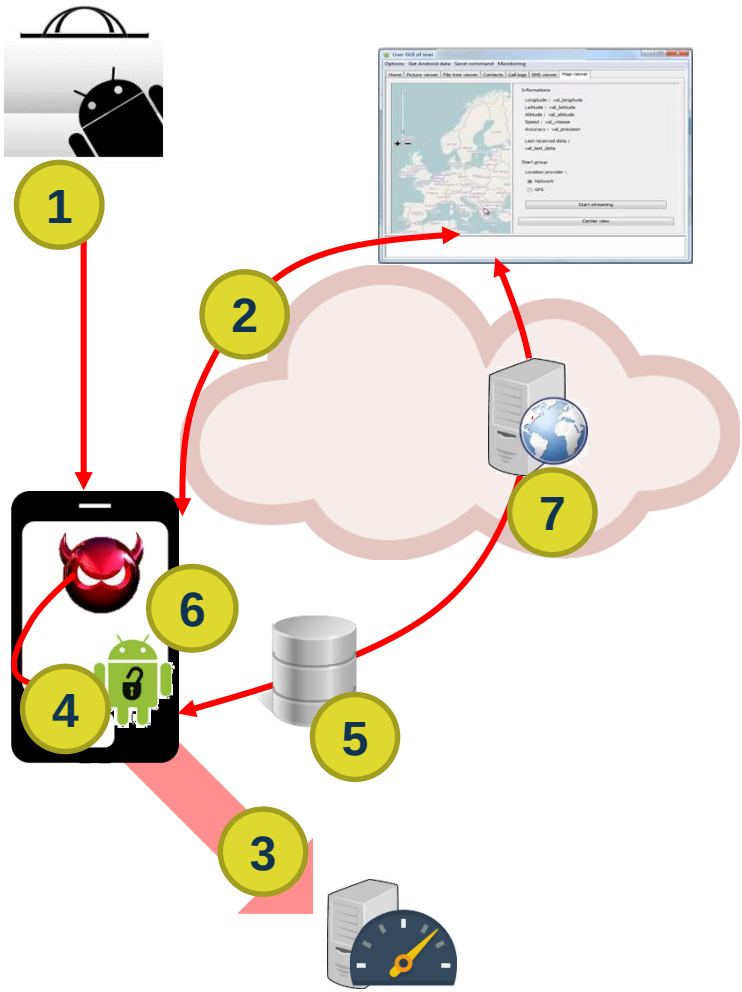
FIRST CRYPTOMINER [3]

Trojan:Android/CoinMiner.A is distributed in a repackaged application. When installed, it essentially hijacks the device to silently mine virtual currency (such as Litecoin) for the malware author. Apart from any data charges incurred, the constant use of the device's hardware may also affect its battery life and eventual lifespan.

```
DataOutputStream os = new DataOutputStream(p.getOutputStream());
os.writeBytes("su -c pm install -r " + appname + "\n");
os.flush();
```

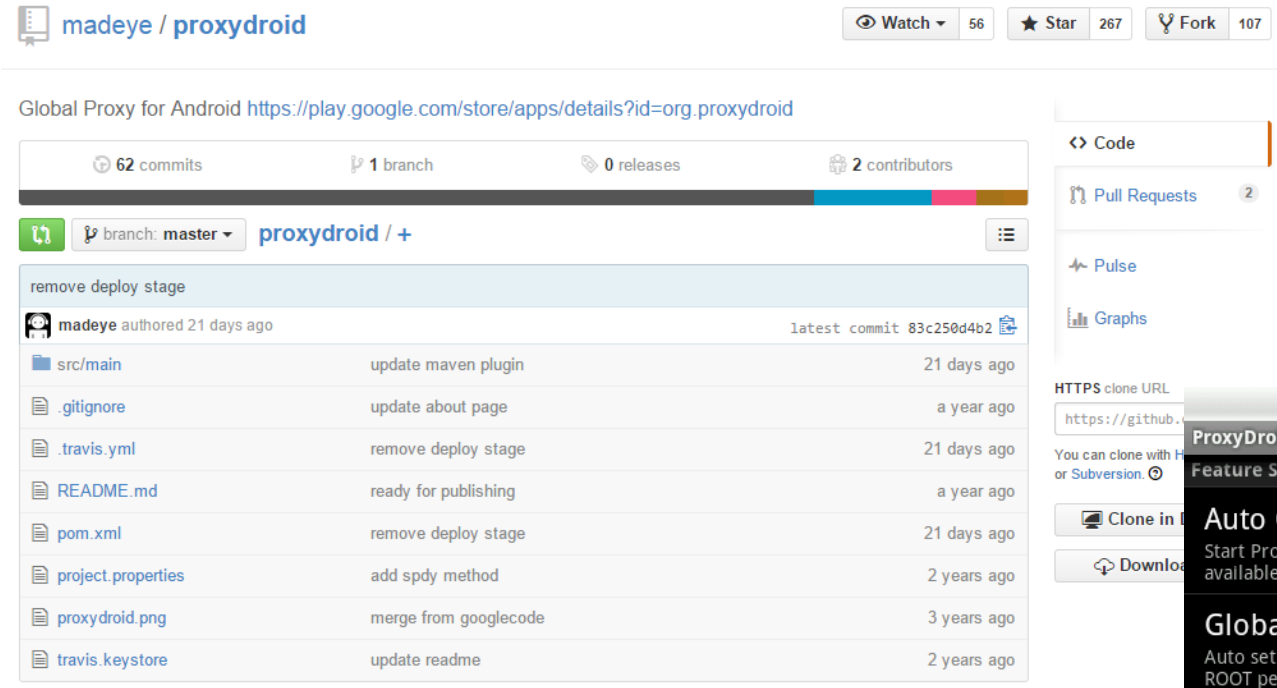
Scenario

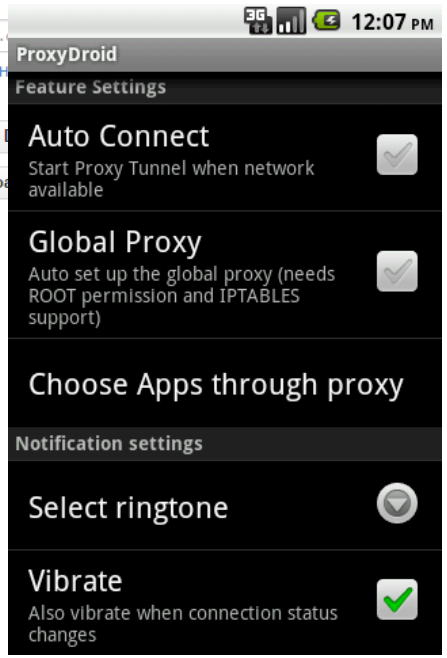
1. *Installation* of a malicious APK
2. *Remote control* of the phone
3. Leverage the *botnet* (DoS attacks)
4. Privilege *escalation*
5. *Exfiltration* of sensitive data
6. *Silent installation* of new applications
7. *Interception* of communications



ProxyDroid

7





ProxyDroid

- Used to set the proxy (HTTP/SOCKS4/SOCKS5) on Android devices
- The app has been modified
  - The GUI has been stripped entirely
  - When launched, sets the proxy and exit
  - The app is installed and run automatically

Conclusions

What we did

Maybe it's just a bit of luck, but we demonstrated that
it's easy to create a powerful Android-based malware…

Take an app

Add malware

Make it bad

Marco Lancini
Security Consultant, CEFRIEL
@lancinimarco

Roberto Puricelli
Security Consultant, CEFRIEL
@robywankenoby