

**secunet**

**encrypt. protect. trust.**

AmiEs 2019, Coimbra, Portugal

# **Mobile Security**

## Threats, Risks and Countermeasures

# Content

---

- **Introduction**
- **Differences PC vs. Smartphone**
- **Android vs. iOS – Some facts and figures**
- **Threats and risks**
- **Countermeasures**
- **Cooperation possibilities**

# Introduction



- Dr. Nils Timotheus Kannengießer
  - » Senior consultant at secunet
  - » Dissertation „Improving Copy Protection for Mobile Apps“ at TUM, Munich
  - » working with Android since 2009



# Introduction

## ■ secunet Security Networks AG

- >> more than 500 employees among eleven sites in Germany
- >> Largest shareholder is Giesecke & Devrient
- >> Five divisions with different focus and customers, e.g.,

>> Public Authorities



>> Homeland Security



>> Defense



>> Critical Infrastructures



>> Automotive



# PC vs. Smartphone

# PC vs. Smartphone

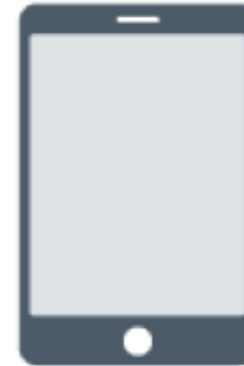
## ■ Differences between a PC and a smartphone

### PC



- lots of scanning tools with system rights
  - User has often **administrator privileges (cf. confirmation dialog)**
  - Applications originate from lots of different locations
- (...)

### Smartphone



- By default: AppStore / Play Store (sometimes 3<sup>rd</sup> parties like Fdroid, Amazon, ...)
  - Antivirus tool is „just an usual app“
  - Typical users have **limited rights**
- (...)

# iOS vs. Android

## Some facts and figures



# iOS vs. Android

## Market Share

- Android has the largest market share.
  - >> Exceptions apply (U.S.)
  - >> Germany: ~68% Android, 31% iOS

## Versions

- Android releases a new system version every few months, adding new and unique features. Manufacturers are responsible for integrating these updates to their devices.
- Apple releases a new iOS version about every year.

## Devices and Screens

- Android is available on thousands of different devices and is suitable for many screen sizes.
- In contrast, iOS offer less than 50 different devices by now.

Ref.  
<https://deviceatlas.com/blog/android-v-ios-market-share#>  
<https://www.opensignal.com/reports/2015/08/android-fragmentation>  
[https://de.wikipedia.org/wiki/Liste\\_von\\_iOS-Ger%C3%A4ten](https://de.wikipedia.org/wiki/Liste_von_iOS-Ger%C3%A4ten)  
<https://developer.android.com/about/dashboardsx>  
<https://www.lifewire.com/ios-versions-4147730>

# iOS vs. Android

## Developer devices

- Usually, the newest version is available on Google's own devices only.
- Nevertheless, emulation is available and various mods due to the fact that Android is open-source.
- In contrast, there are no special developer devices for iOS.

## Architecture

- Android is based on Linux using its user rights management (UIDs/GIDs) for the separation of apps on kernel level for sandboxing.
- Apps are distributed as APK files.
- In contrast, iOS is based on OSX (BSD/UNIX) and apps are distributed as native files secured by sandboxing and a chain of trust (aka secure boot)

## Testing and Release Management

- Each Android app may be signed by a debug key for instant testing. The release on the Play Store with a self-signed certificate takes place in minutes.
- In comparison, testing iOS apps needs a special testing certificate. Also, Apple tests apps severely for issues, which make take hours to days.

Ref.  
<https://developer.android.com/studio/run/emulator>

# iOS vs. Android

---

## Comparison

- » Architecture and security model is similar (Linux/UNIX and sandboxing)
- » Similar performance nowadays, after Google decided to compile apps to native code on each device
- » The amount of different devices and responsibilities is different. Apple tries to protect customers itself, while Android's security largely depends on the device manufacturer.
- » The app publishing and associated security measurements are different.
  - » Almost instant app publishing / Google&Android
  - » Delay of several hours to days / Apple&iOS
- » ...

# iOS vs. Android

## >> Who wins that race?

It is not that easy to answer that question. It depends on a lot of factors like developers, users and ultimately the manufactures. Choose the phone that you prefer 😊 !

**Motherboard**

### Apple Just Released an Emergency Patch for the iPhone

The company released a patch to fix a dangerous bug on Monday, a week after hackers had released a jailbreak for fully patched iPhones.

By [Lorenzo Franceschi-Bicchieri](#)  
Aug 26 2019, 7:52pm [Share](#) [Tweet](#)

Security > 7-Tage-News > 08/2019 > Trojaner-App CamScanner auf mehr als 100 Millionen Android-Geräten...

### Trojaner-App CamScanner auf mehr als 100 Millionen Android-Geräten installiert

In Google Play hat sich eine gefährliche App geschlichen. Wer CamScanner – Phone PDF Creator installiert hat, sollte die App zügig löschen.

?  
None ?!  
?

Ref. [https://www.vice.com/en\\_us/article/d3av8m/apple-emergency-patch-iphone](https://www.vice.com/en_us/article/d3av8m/apple-emergency-patch-iphone)  
<https://www.heise.de/security/meldung/Trojaner-App-CamScanner-auf-mehr-als-100-Millionen-Android-Geraeten-installiert-4508174.html>

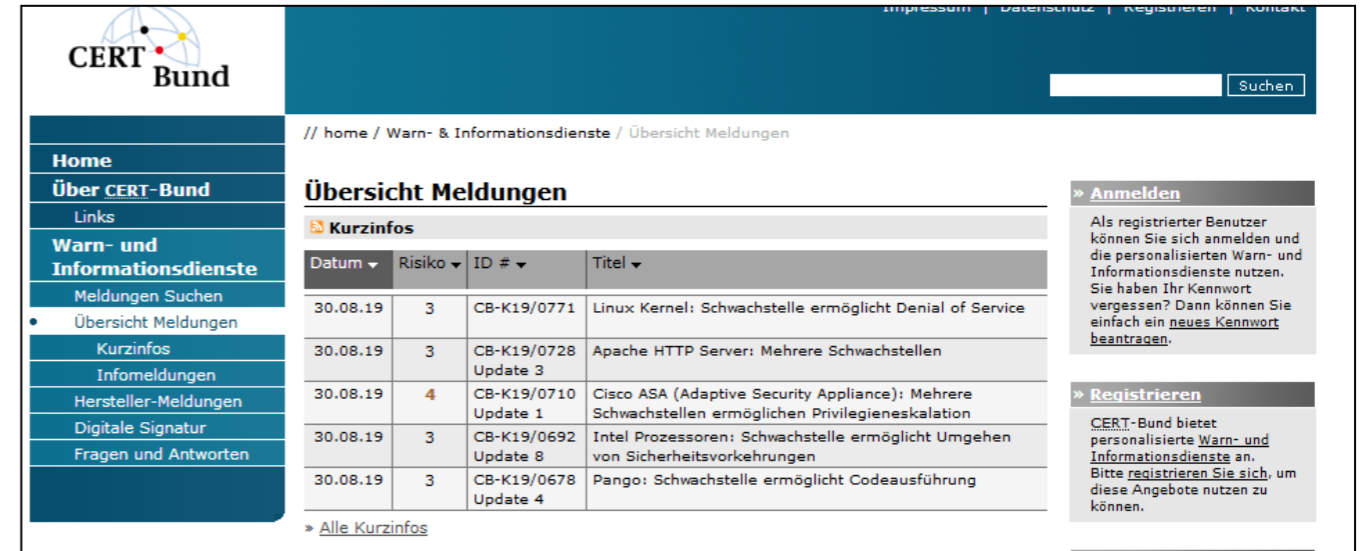
# Threats & risks (a few examples)

# Threats and Risks

## Status quo?

Are there threats and risks?

- <https://www.cert-bund.de/overview>
- <http://www.heise.de/security>
- <https://www.varonis.com/blog/cybersecurity-statistics/>



The screenshot shows the CERT Bund website interface. The main content area is titled "Übersicht Meldungen" (Overview of Alerts). It features a table with columns for "Datum" (Date), "Risiko" (Risk), "ID #", and "Titel" (Title). The table lists several alerts from August 30, 2019, including vulnerabilities in Linux Kernel, Apache HTTP Server, Cisco ASA, Intel Processors, and Pango. The website also includes a navigation menu on the left and a search bar at the top right.

Datum	Risiko	ID #	Titel
30.08.19	3	CB-K19/0771	Linux Kernel: Schwachstelle ermöglicht Denial of Service
30.08.19	3	CB-K19/0728 Update 3	Apache HTTP Server: Mehrere Schwachstellen
30.08.19	4	CB-K19/0710 Update 1	Cisco ASA (Adaptive Security Appliance): Mehrere Schwachstellen ermöglichen Privilegieneskalation
30.08.19	3	CB-K19/0692 Update 8	Intel Prozessoren: Schwachstelle ermöglicht Umgehen von Sicherheitsvorkehrungen
30.08.19	3	CB-K19/0678 Update 4	Pango: Schwachstelle ermöglicht Codeausführung

Ref. <https://www.cert-bund.de/overview>

→ Simple answer is YES !

“Cybersecurity Ventures predicts that a business will fall victim to a ransomware attack every 14 seconds by 2019.”

Ref. <https://cybersecurityventures.com/hackerpocalypse-cybercrime-report-2016/>

# Threats & Risks – “Beautiful“ Headlines 2017/2018/2019 (examples)

- **”1 billion Apple user [...] may have been attacked“**, Src: <https://www.forbes.com/sites/zakdoffman/2019/08/30/google-shocks-1-billion-iphone-users-with-malicious-hack-warning/>
  - **Apple issues emergency update**, Src: <https://www.heise.de/mac-and-i/meldung/Apple-bringt-Notfallupdate-fuers-iPhone-und-weitere-wichtige-Aktualisierungen-4505955.html>
  - **Trojan app CamScanner installed on more than 100 millions Android devices**,  
Src: <https://www.heise.de/security/meldung/Trojaner-App-CamScanner-auf-mehr-als-100-Millionen-Android-Geraeten-installiert-4508174.html>
  - **Researchers find 234 ultrasound spyware apps**, Src: <https://www.heise.de/newsticker/meldung/Tracking-Forscher-finden-Ultraschall-Spyware-in-234-Android-Apps-3704642.html>
  - **Using a master-fingerprint to unlock smartphones**, Src: <https://www.heise.de/newsticker/meldung/Mit-Master-Fingerabdruck-Zugriff-auf-fremde-Smartphones-bekommen-3702411.html>
  - **Developer certificate used to spy on HTTPS connections**, Ref. <https://www.heise.de/mac-and-i/meldung/Malware-mit-Apple-Entwicklerzertifikat-spioniert-HTTPS-Traffic-aus-3699926.html>
- There is a persistent security threat on different levels
- For example, end users want to select devices with regular and immanent updates for most security (and may still be affected by 0-day exploits)

# Threats and Risks – Social Engineering

- Famous figures in history: Kevin Mitnick
  - „*The Art of Deception*“, book
- Examples
  - Attackers distribute a note at a congress with the request to install an app
  - Or, they “loose” an infected USB stick in the parking lot in the hope that the victim is going to plug it in (install trojan)





# Threats and Risks – Social Engineering

- Of course, an attacker may just look and record situations and attack when people are not careful. Some examples:

- » entering passwords, while others are watching

- » Unlock a phone with his own fingerprint when someone is sleeping

- » ...



Sources: LH / Linnemann, secunet

# Threats and Risks – Humans are at risk

- There are lots of associated risks with „layer 8“
  - End users install **repackaged**, infected banking **apps** (there are lots of exploits to infect phones, even when updated regularly)
  
  - End users **enter secret credentials** on faked websites
    - cf. Free WiFi / captive portal → faked login site
  
  - Many use root apps or **rooting-apps without knowing anything** about them (is „super su“ downloaded from X infected?)
  
  - Even administrator and developers are in danger
    - CamScanner
      - **Included library** for commercials was a severe security issue downloading and executing further code (cf. Report Kaspersky)
  
  - Surfing on an infected website (recent iPhone issue), or, earlier this year with Android phones by watching an infected image of a cat – it’s called „**drive-by attack**“.

Sources:  
<https://thehackernews.com/2019/02/hack-android-with-image.html>  
<https://securelist.com/dropper-in-google-play/92496/>  
<https://www.heise.de/ct/artikel/Sicher-unterwegs-Gefahren-fuer-Technik-auf-Reisen-4449707.html>

# Threats and Risks – Developers are at risk

## ■ Developers distributed malware unknowingly

- Android Malware SimBad infected 150 millions smartphones by using 210 apps in Play Store that included their library
- **Sounds familiar ;-)** ?  
CamScanner with 100 million users just had the same issue a few days ago

## ■ Reengineering issues

- APKtool (→ smali code)
- JD Gui (→ Java code)

```
14
15 .super Ljava/lang/Object;
16
17 .method public static main([Ljava/lang/String;)V
18   .registers 2
19
20   sget-object v0, Ljava/lang/System;->out:Ljava/io/PrintStream;
21
22   const-string    v1, "Hello World!"
23
24   invoke-virtual {v0, v1}, Ljava/io/PrintStream;->println(Ljava/lang/String;)V
25
26   return-void
27 .end method
```

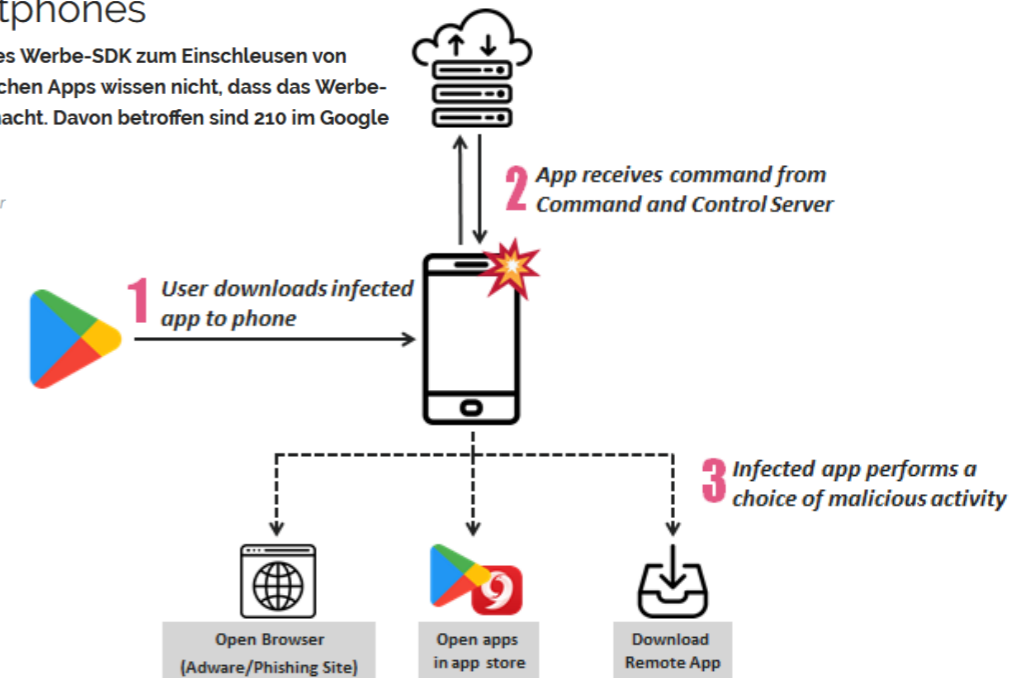
Source: <https://github.com/JesusFreke/smali/blob/master/examples/HelloWorld/HelloWorld.smali>

ZDNet / Sicherheit / Virus

## Android-Malware SimBad infiziert bis zu 150 Millionen Smartphones

Die Hintermänner nutzen ein legitimes Werbe-SDK zum Einschleusen von Schadcode. Die Entwickler der fraglichen Apps wissen nicht, dass das Werbe-SDK ihre Anwendungen zu Adware macht. Davon betroffen sind 210 im Google Play Store angebotene Apps.

von Stefan Beiersmann am 14. März 2019, 07:31 Uhr



Source: <https://www.zdnet.de/88356313/android-malware-simbad-infiziert-bis-zu-150-millionen-smartphones/>

<https://blog.talosintelligence.com/2018/07/Mobile-Malware-Campaign-uses-Malicious-MDM.html>

# Countermeasures

# Countermeasures

- Humans
  - » **Obvious: Try to educate people around you and yourself**
    - » Look up issues on a specific topic or ask professionals for help
      - » For instance, besides events like today, we organize events for companies on typical hacking issues to raise the awareness:



<https://www.secunet.com/en/solutions-services/securitymanagement/awareness-campaigns/live-hacking/>

- » In general, it **depends** really **on the actual use case**, how to increase security. For instance, a government may use a special domain to avoid issues with fraud (\*.gov vs. \*.com)

# Countermeasures

---

- Developers (“Do’s and don’ts”)
  - **Check your requirements carefully** and avoid any external threats, e.g., do not use a huge library, when a small and well-tested (certified?) one fits your requirements
  - **Usage of obfuscation** is always a “cat-and-mouse” game, but they provide some minimal anti-reengineering capabilities to increase the time on the attacker’s side and may be relevant **to protect your product at market release** (cf. dissertation)
  - Use **hardware security** whenever possible (e.g., keystore on Android instead of a file, TEE/smartcards for additional security etc.)
  - Usage of **hardcoded, sensitive information** in your code is a no-go (access keys etc.)
  - **The usage of system libraries** should be carefully **evaluated**. For instance, native Android code ignores those proxy settings. In my own nLVL tests (dissertation), students were not able to intercept the license-communication using proxies.

# Countermeasures

---

- » Moreover, the **chosen programming language** affects you:
  - » Javascript used in WebApps is convenient, but implicates easy reengineering issues often ([obfuscated?] source code in APK?)
  - » Java is a good choice in general, but can be reengineered easily (apktool) and the program logic can often be reconstructed
  - » Instead C/C++ is more difficult, but the resulting code (assembler) is hard to understand and can highly be obfuscated
- » Be careful in **setting up a crypto library**. For instance, using a small prime number instead of a huge one may weaken your encryption security implementation extremely. Also, watch out for updated libraries that may include security fixes.
- » **Avoid sensitive logging** (e.g., do not print passwords etc.) and unnecessary storage in memory (e.g., unlimited caching of password)
- » ...

# Threats, Risks and Countermeasures

---

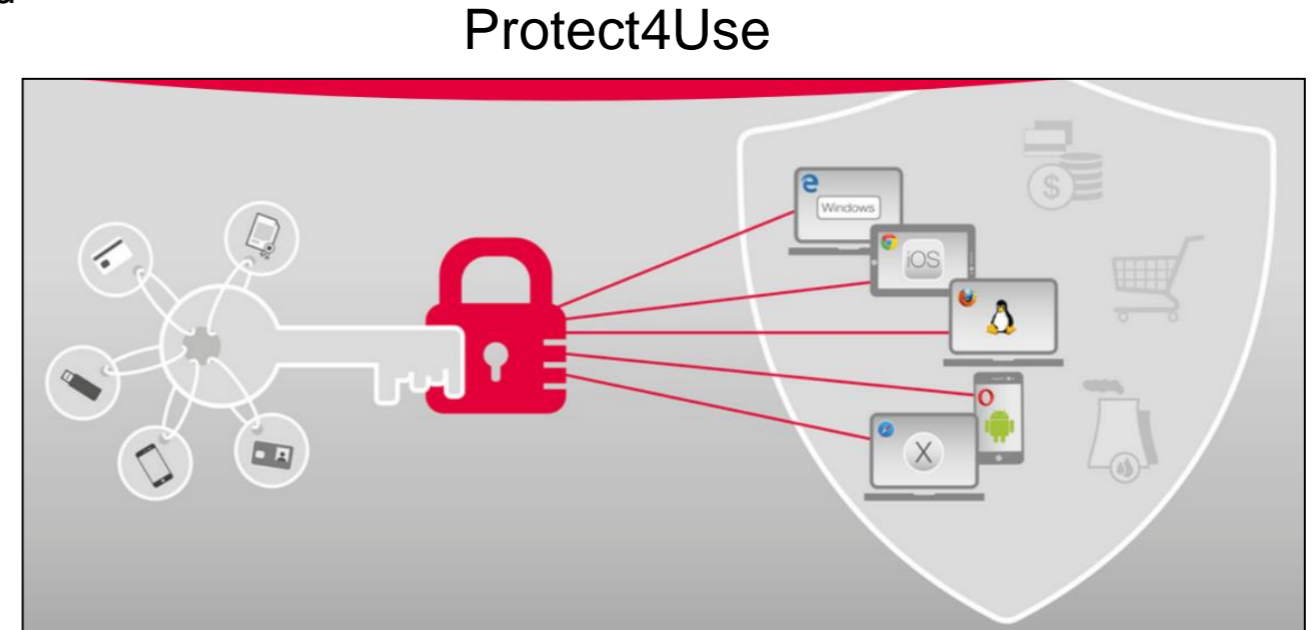
- There are many further aspects that we cannot cover today:
  - Server security, e.g., usage of port knocking to hide SSH services
  - Connection security, e.g., usage of TLS v1.3 ?
  - Web service security, like SQL injections and how to avoid them
  - Advanced anti-reengineering possibilities, like emulation detection
  - Secure hardware, like TEEs and chain of trust
  - Hacking gadgets, like a Trojan creation tool (→ “know your enemy”)
  
  - ... and many more
  
- **Nevertheless, I’d like to invite you to think about a visit in Munich, Germany to join us for lots of fancy projects in one of our divisions.**



# Cooperation possibilities

# Cooperation possibilities

- secunet is available at several locations in Germany
- My working place is located in beautiful Munich (see [www.muenchen.de](http://www.muenchen.de) )
  - Konrad-Zuse-Platz 2-4, 81829 München
- We offer various topics for student theses or just temporary working topics. For instance (recent student project):
  - Awareness App (analyzing installed apps for permissions and calculated score)
- Early ideas to be used as a theses project:
  - Flutter Apps for Protect4Use to have an UI for all five platforms (Android, iOS, Windows, MacOS, Linux)



# Cooperation possibilities

---

- Of course, we offer paid student positions. These require local attendance due to the topics itself and security aspects (often “confidential” level or even higher).
  
- In general, we offer positions for
  - Front-/Backend and full stack developers
  - Consultants (System Architecture / Operations / ISM)
  
- Moreover, secunet offers guidance and products on all kinds of security topics for companies. For instance, the aforementioned live-hacking sessions, besides further solutions like SINA for advanced security requirements.
  
- **Interested?**
  - [Nils.Kannengiesser@secunet.com](mailto:Nils.Kannengiesser@secunet.com)

The logo for secunet, featuring the word "secunet" in a bold, sans-serif font. The letters "secunet" are black, and the letters "net" are red. The background of the slide is white with a large, curved red shape on the left side that tapers towards the top right.

**Dr. Nils Timotheus Kannengießer**

Senior Consultant

secunet Security Networks AG

Kurfürstenstraße 58

45138 Essen

Phone +49 201 5454-2309

[Nils.Kannengiesser@secunet.com](mailto:Nils.Kannengiesser@secunet.com)