Mobile Anwendungen durchleuchtet

Sicherheitslücken und Angriffe auf Apps

Dr. Julian Schütte, Fraunhofer AISEC/Breakpoint GmbH

Kurzvorstellung

- Abteilungsleiter "Sichere Dienste und Anwendungen", Fraunhofer AISEC
- Sicherheitsforscher: Statische und dynamische Codeanalyse
- Geschäftsführer Breakpoint GmbH (Fraunhofer Spin-Off-Unternehmen)

  App-Ray: Vollautomatische Sicherheitsanalyse von Apps

  http://www.app-ray.com

TECHNISCHE UNIVERSITÄT MÜNCHEN

App-Ray

Fraunhofer AISEC

- Wie man eine App durchleuchtet
- Wie ist es um die Sicherheit von Apps bestellt?
  - Blick auf den Market
  - Beispiele aus dem Labor
- Was bringen Schutzmaßnahmen?

App-Ray

Fraunhofer AISEC

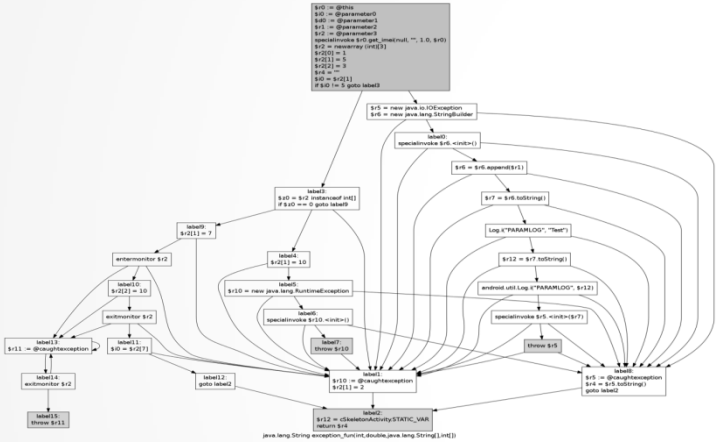
Wie man eine App durchleuchtet: Automatisches Reverse Engineering

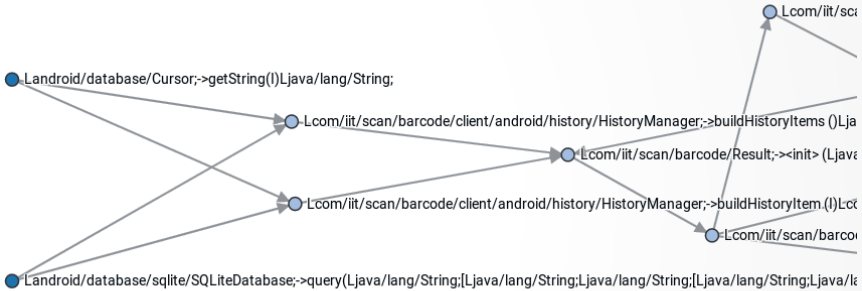
Entpacken → Meta-Daten verstehen → Disassemblieren, Pointer-Rekonstruktion

App-Ray

Fraunhofer AISEC

Wie man eine App durchleuchtet: Automatisches Reverse Engineering



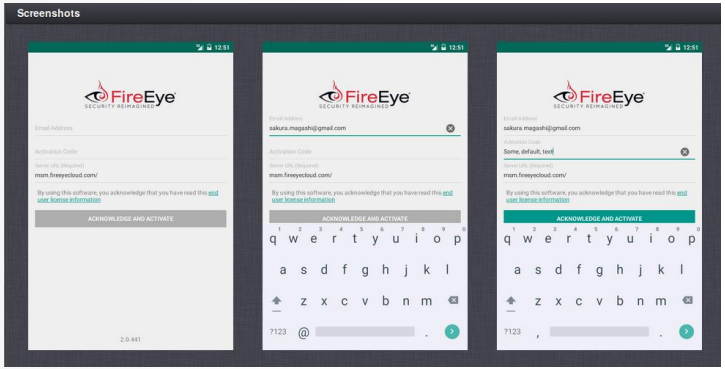


Call Graph rekonstruieren → Datenflüsse und Funktionalität verstehen → 1. Auswertung

App-Ray

Fraunhofer AISEC

Wie man eine App durchleuchtet: Dynamische Analyse



App ausführen & Explorieren → Kommunikation & Verhalten beobachten → 2. Auswertung

App-Ray

Fraunhofer AISEC

Wie man eine App durchleuchtet: Auswertung

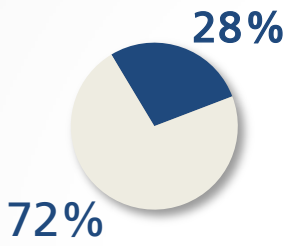
Sicherheits-Report

Datenschutz-Report

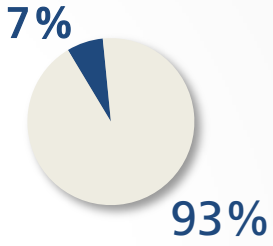
App-Ray

Fraunhofer AISEC

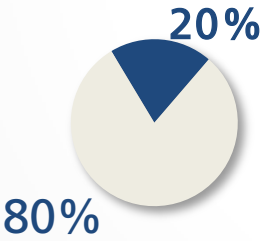
241 Top-Banking-Apps



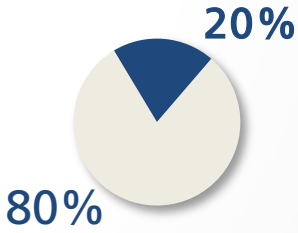
Fehlerhafte Prüfung von TLS-Zertifikaten



Kein Schutz gegen Skimming (Cloak 'n Dagger-Attacke)



Gebrochene Kryptografie

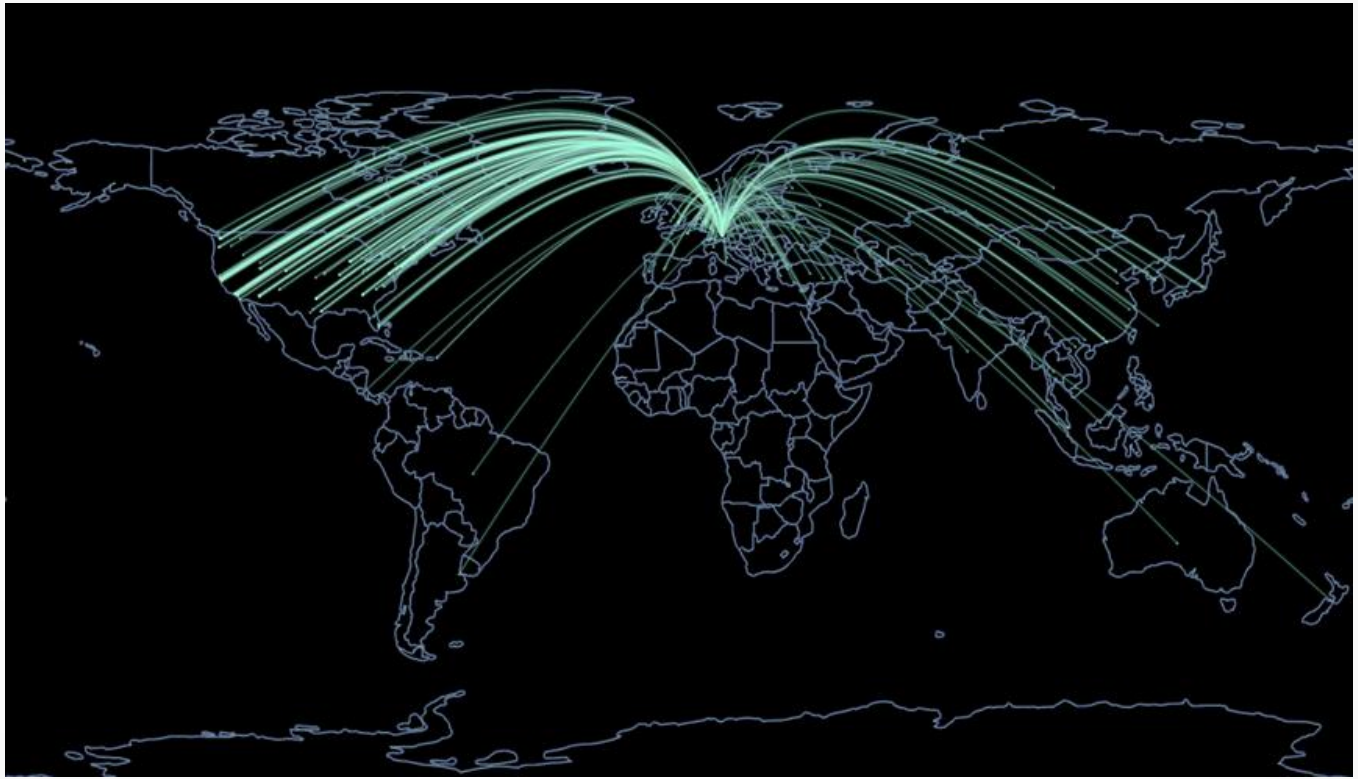


Web-Inhalte erhalten Zugriff auf das System

App-Ray

Fraunhofer AISEC

Kommunikationsverhalten der 10.000 beliebtesten Apps



Persönliche Daten werden an 6841 Server ohne Einwilligung verschickt

App-Ray

Fraunhofer AISEC

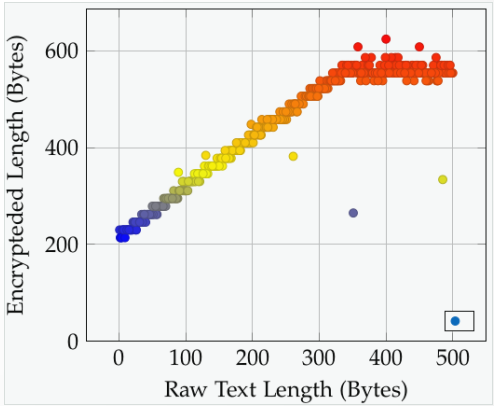
Beispiel WhatsApp (1/2)

- WhatsApp verschlüsselt Nachrichten mit "Ende-zu-Ende-Sicherheit"

Heißt "Ende-zu-Ende-Sicherheit", dass alles vertraulich bleibt?

Schwächen in kryptographischen Protokollen

- Nachrichtengröße und -timing erlaubt Unterscheidung zwischen Nachrichtentypen (Textnachrichten, Synchronisation, Sperren eines Benutzers, …)
- Länge der Nachrichten korreliert stark mit Klartext
- Medien werden nicht direkt übertragen, sondern an Storage-Server mmx-ds.cdn.whatsapp.net
- Meta-Daten an WhatsApp-Server
  - Telefonbuch-sync per Austausch der Tel.-Nr.-Hashes
  - Mitglieder eines Gruppenchats sind WhatsApp bekannt



App-Ray

Fraunhofer AISEC

Beispiel WhatsApp (2/2)

Sicherheitslücke in WhatsApp

- WhatsApp speichert gesendete und empfangene Mediendateien in öffentlich schreibbaren Ordner (Fotos, Voice-Nachrichten, Dokumente)
- Apps speichert Hashes der Dateien, prüft sie aber nicht
- Alle Apps auf dem Gerät können WhatsApp Mediendaten lesen, veröffentlichen, modifizieren & löschen … noch bevor sie in WhatsApp angezeigt werden !

Keine privaten Bilder/Dokumente über WhatsApp verschicken!

Keine privaten Bilder/Dokumente empfangen!

Empfangenen Mediendateien kann nicht vertraut werden!

Demo-Video: https://youtu.be/pN02g_elhb0

App-Ray

Fraunhofer AISEC

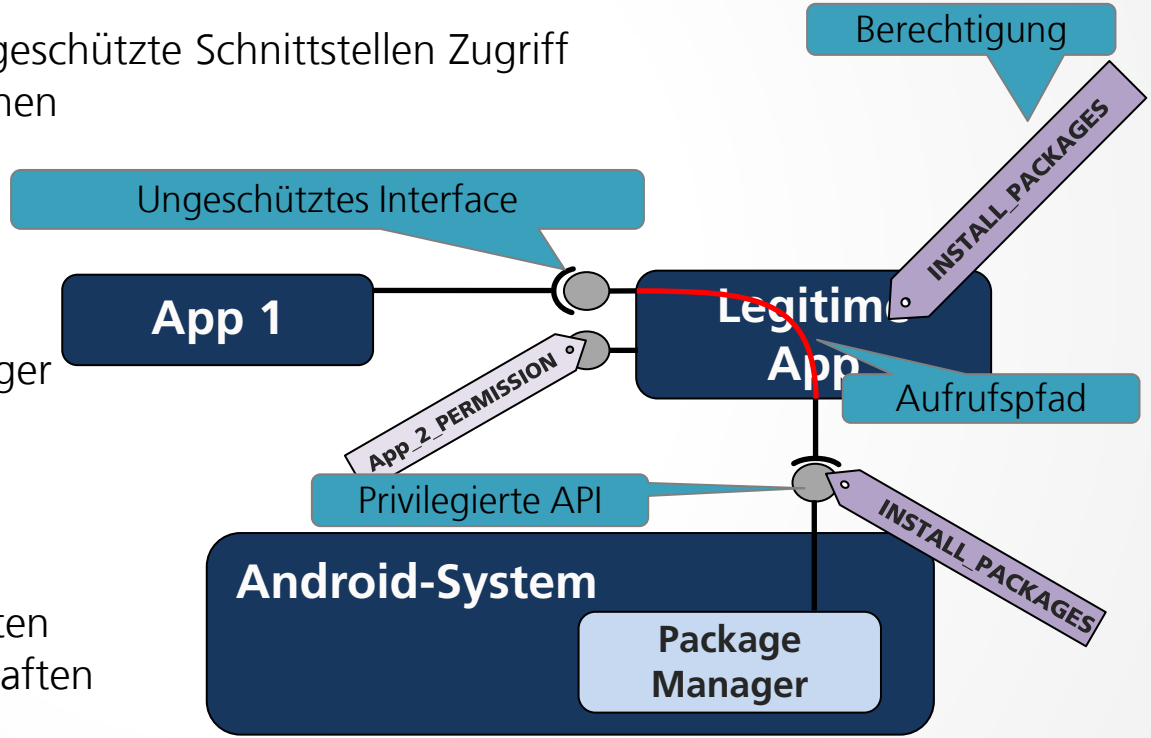
Eine unsichere App kann *alle* Daten auf dem Smartphone gefährden

Capability Leak

- Legitime App erlaubt über ungeschützte Schnittstellen Zugriff auf priviligierte Systemfunktionen

Beispiele

- Samsung Kies
  - Erlaubt Installation beliebiger Apps im Hintergrund
  - Lässt sich nicht entfernen
- Certifi-Gate
  - Fernsteuerung des gesamten Smartphones über fehlerhaften TeamViewer



[1]Certifi-Gate: Front door access to pwning millions of Android devices. Ohad Bobrov, Avi Bashan. BlackHat 2015

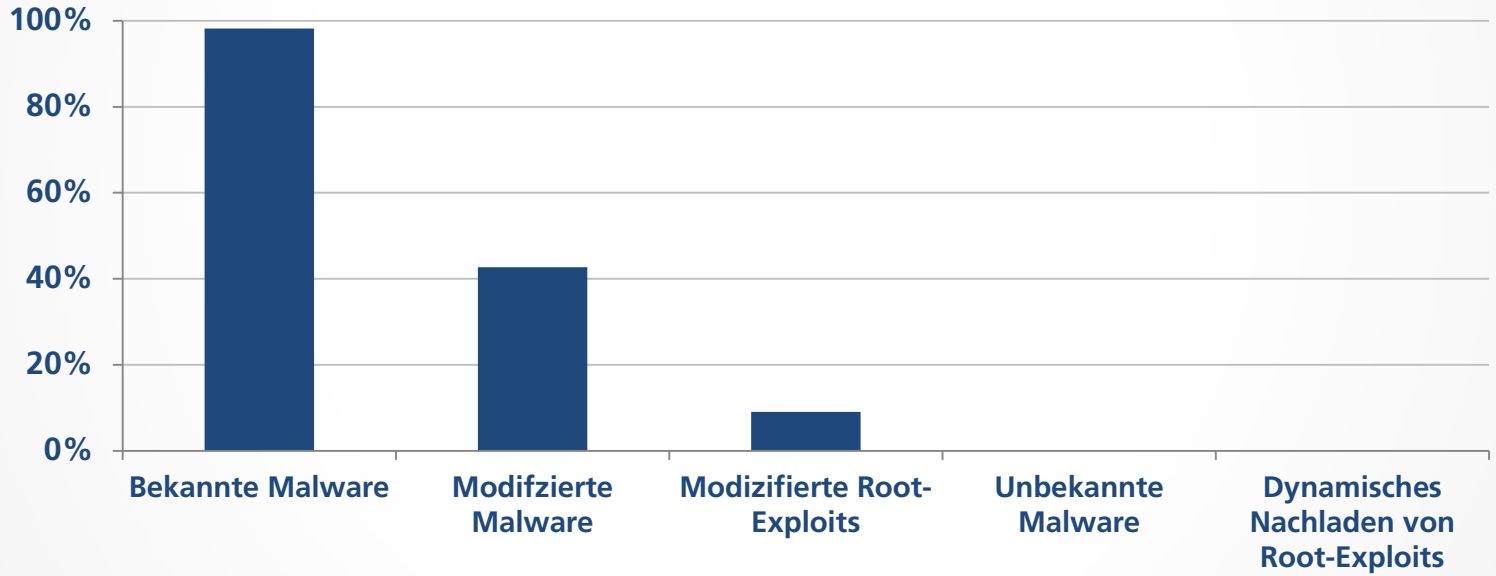
App-Ray

Fraunhofer AISEC

Wirksamkeit von Schutzmaßnahmen

Schutz durch Anti-Virus-Apps?

Studie: "On the Effectiveness of Malware Protection on Android. An Evaluation of Android Antivirus Apps"



- Anti-Virus-Apps unterliegen der gleichen Sandbox wie normale Apps

App-Ray

Fraunhofer AISEC

Schutz durch den Market?

"*Every day, Google Play Protect automatically reviews 50 billion apps*"[1]

"Reviews"?

- "50 billion per day" = 578.703 Apps/Sekunde
- Entpacken einer App (11MB) auf herkömmlichem Rechner: 8 Sekunden
- Erforderliche Rechenleistung ~ das 4.6 Mio.-fache eines normalen Rechners

"Reviews": Abgleiche gegen Datenbank

- Prüfung von Apps bevor sie zum Download angeboten werden, ist nicht näher dokumentiert

[1] https://android-developers.googleblog.com/2018/03/android-security-2017-year-in-review.html

App-Ray

Fraunhofer AISEC

Schutz durch Apple AppStore?

- Publizieren über Apple AppStore: Erfordert Entwicklerzertifikat + App-Review
  - Entwickler sind Apple bekannt und können sanktioniert werden
  - geschätzte Zeit pro App-Review: ~30 Sekunden
- Review kann umgangen werden[1,2]
  - Enterprise-Zertifikate erlauben Installation ohne Review
- Sicherheitsmechanismen auf dem Phone können umgangen werden[1,3]

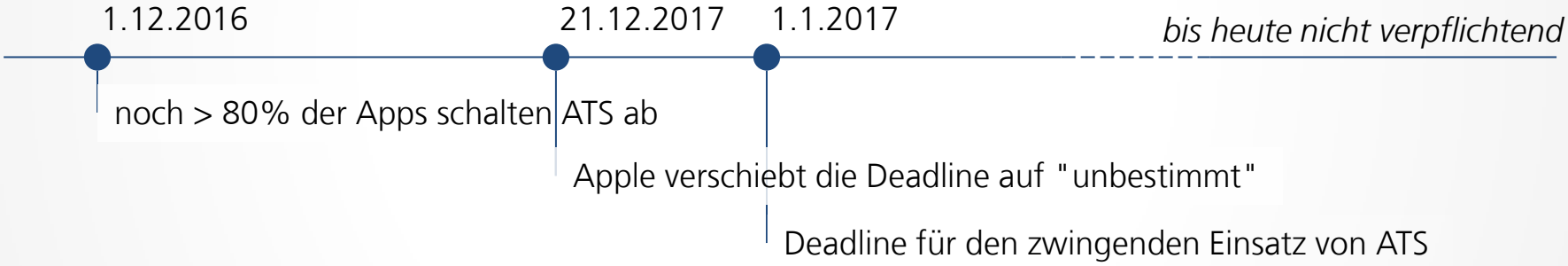
[1] Wang, T.; Lu, K.; Lu, L.; Chung, S.; Lee, W. Jekyll on iOS: when benign apps become evil. In 22nd USENIX Security Symposium, pp.559–572, 2013.
[2] Han, Kywe,Yan,Bao,Deng,Gao,Li,Zhou . Launching Generic Attacks on iOS with Approved Third-Party Applications In ACNS 2013
[3] SandScout: Automatic Detection of Flaws in iOS Sandbox Profiles

App-Ray

Fraunhofer AISEC

iOS Application Transport Security

- ATS zwingt Apps, HTTP**S** zu verwenden → sehr sinnvoller Mechanismus
- Apple will ATS verpflichtend für alle Apps einführen
- Aber dann …

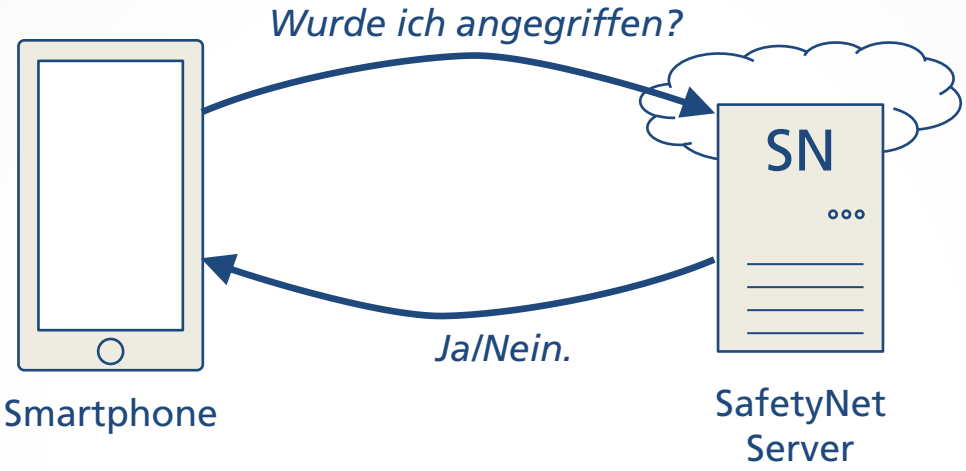



```
<key>NSAppTransportSecurity</key>
<dict>
   <key>NSAllowsArbitraryLoads</key>
   <true/>
</dict>
```


App-Ray

Fraunhofer AISEC

SafetyNet



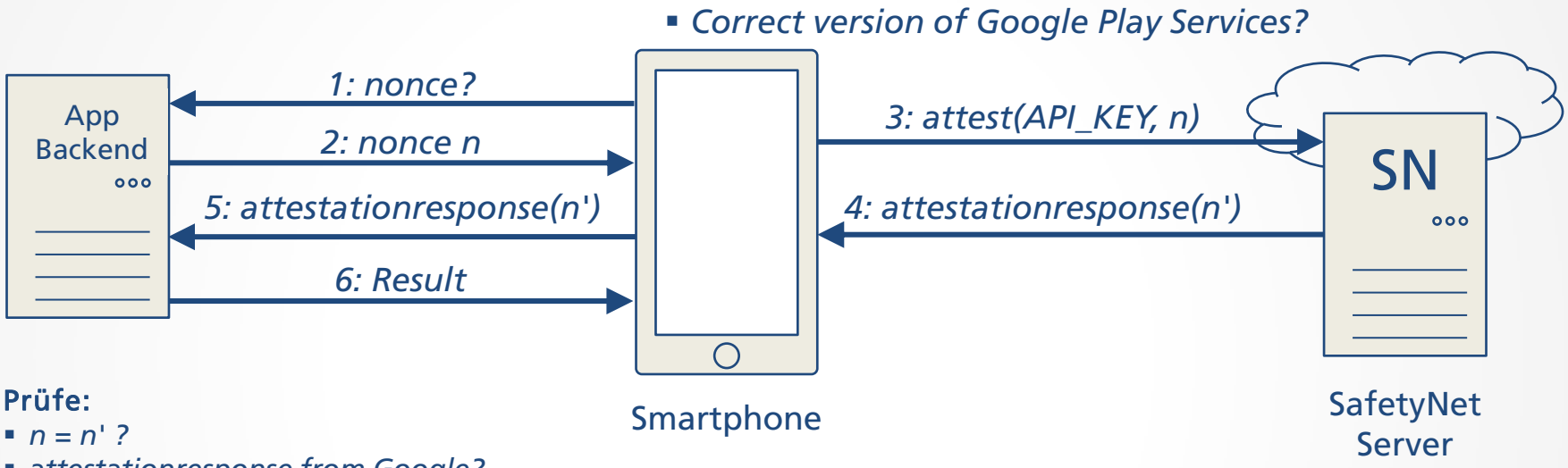
Idee

- Auf dem (angreifbaren) Smartphone werden (viele) Daten gesammelt
- Beurteilung von App und Gerät erfolgt serverseitig
- Angreifer muss Daten fälschen (aber welche?)
- Toll: Entwickler rufen eine API auf und sind "sicher"

App-Ray

Fraunhofer AISEC

SafetyNet: Wie es eigentlich funktioniert



Prüfe:

- *n = n' ?*
- *attestationresponse from Google?*
- *app package name correct?*
- *certificate of APK correct?*
- *timestamp new?*
- *basicIntegrity = true?*
- *ctsProfileMatch = true?*
- *Extract SSL certificate chain from attestationresponse*
- *Validate SSL certificate chain*
- *Check if SSL hostname of leaf certificate = attest.android.com*
- *Use public key of certificate to verify the signature attestationresponse*

App-Ray

Fraunhofer AISEC

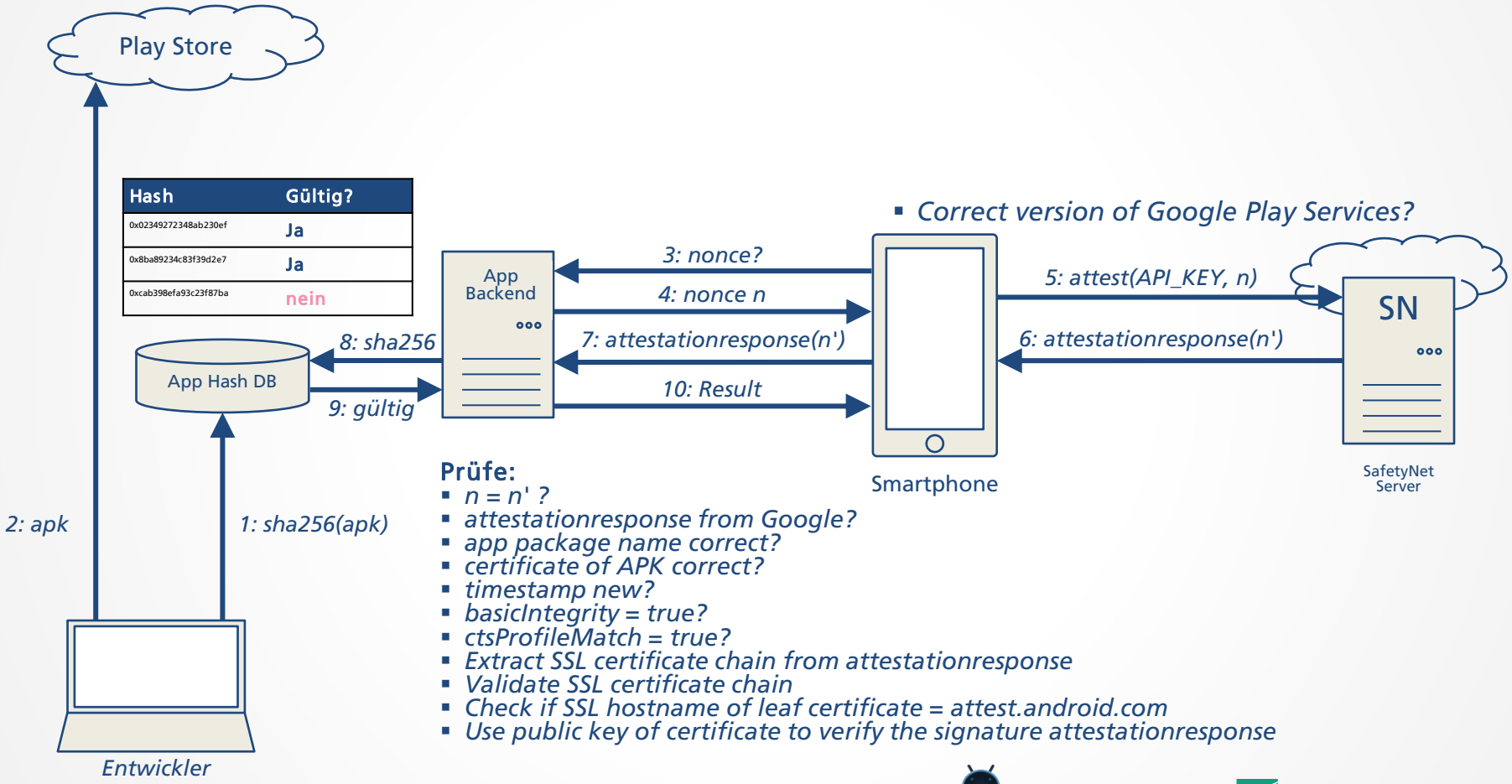
SafetyNet: Fallstricke

- Prüfung von SafetyNet-Antwort in der App ist sinnlos
  - Kann entfernt oder überschrieben werden
- Für Prüfung in Backend muss der Benutzer online sein. Sonst … ?
  - Bsp. Indoor Network Coverage ~30%
- Prüfung von SafetyNet-Antwort über Google-API ist nur für Entwicklungszwecke gedacht
  - Limitiert auf 1000 Anfragen
- Prüfung von SafetyNet-Antwort in eigenem Backend ist kompliziert
  - Was bedeutet "ctsProfile"? Wann ist ein Zeitstempel noch "aktuell"?

App-Ray

Fraunhofer AISEC

SafetyNet: Wie es eigentlich funktioniert (jetzt wirklich)



Prüfe:

- *n = n' ?*
- *attestationresponse from Google?*
- *app package name correct?*
- *certificate of APK correct?*
- *timestamp new?*
- *basicIntegrity = true?*
- *ctsProfileMatch = true?*
- *Extract SSL certificate chain from attestationresponse*
- *Validate SSL certificate chain*
- *Check if SSL hostname of leaf certificate = attest.android.com*
- *Use public key of certificate to verify the signature attestationresponse*

App-Ray

Fraunhofer AISEC

Was bringen Schutzmaßnahmen?

- Einfache Virenscanner lösen kein Problem
- Google & Apple geben sich Mühe
  - die Plattformen werden sicherer und bieten Schutzmaßnahmen an
  - Entwickler werden ermutigt, sie zu verwenden
- Maßnahmen wie SafetyNet & ATS sind durchdacht, kostenlos und bieten Schutz. Aber:
  - Entwickler ~~sind faul~~ stehen unter Zeit- und Kostendruck
  - Schutzmechanismen nehmen einem keine Arbeit ab, sondern erzeugen zusätzliche
  - Fehleranfälligkeit steigt & Benutzererlebnis verschlechtert sich

App-Ray

Fraunhofer AISEC

Fazit

Fast alle Apps haben Sicherheitslücken

- Eine App zu schreiben ist einfach, eine *sichere* App zu schreiben hingegen schwer & teuer
- Bereits eine unsichere App kann *alle* Daten auf dem Telefon gefährden
- Schadhafte Apps können auch aus vertrauenswürdigen Quellen und von namhaften Unternehmen stammen

"*Wenn etwas kostenlos ist, bist Du nicht der Kunde, sondern das Produkt*"

App-Ray

Fraunhofer AISEC

Fazit

Was kann ich tun?

- Gesunder Menschenverstand
  - Überblick verschaffen: Welche Daten liegen wo? Wofür werden Apps genutzt?
  - Sicherheitskonzept erstellen
  - Über einzelne Apps informieren!

- Technologie
  - Geräte-Passwort, Full Disk Encryption einschalten
  - Whitelists. Nicht alles herunterladen & installieren
  - Berufliche Daten vom Rest des Systems trennen (Container-Lösungen)

App-Ray

Fraunhofer AISEC