

Cnlab/CSI Herbsttagung 2017

Apps und Sandboxen

Agenda

- App-Technologien
- Integrität von Apps
- Schutzmechanismen iOS und Android
- Vergleich mit Standard-PC
- Fazit

Android: Erfolgreich und schon bald am Ende

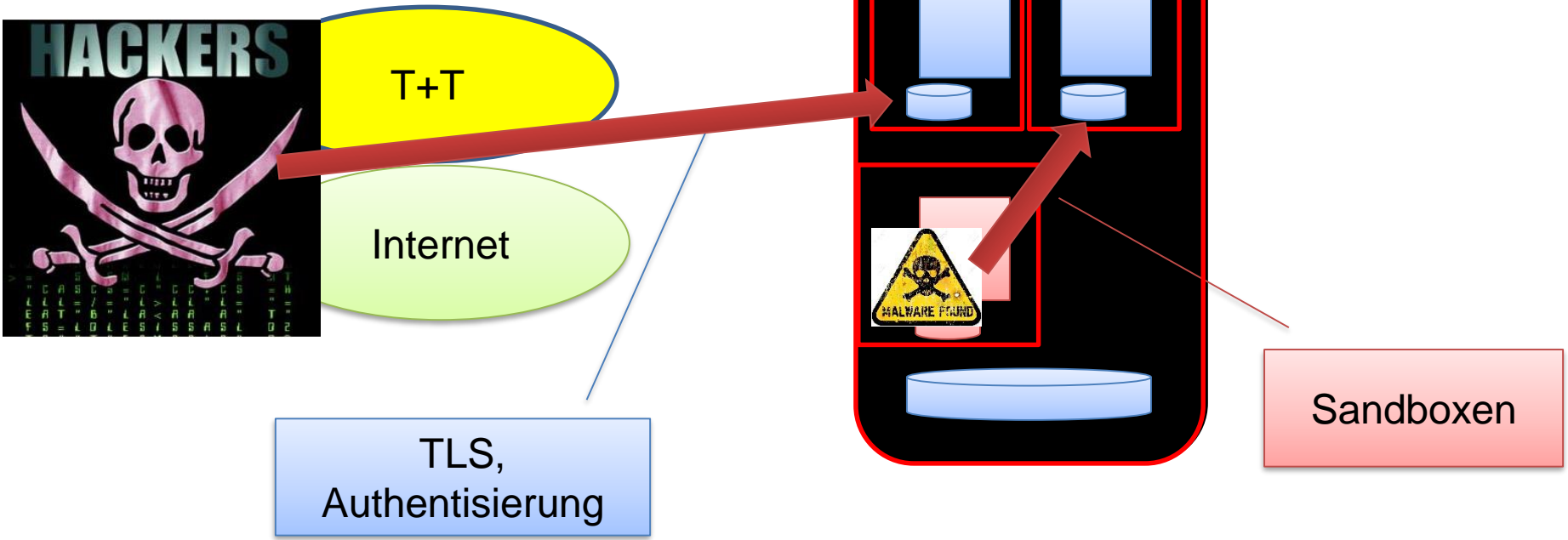
Googles Smartphone-Betriebssystem schleppt so viele Altlasten herum, dass es keine Zukunft hat. Doch mutmasslich ist ein Nachfolger in Arbeit.



TA Online vom 23.8.2017

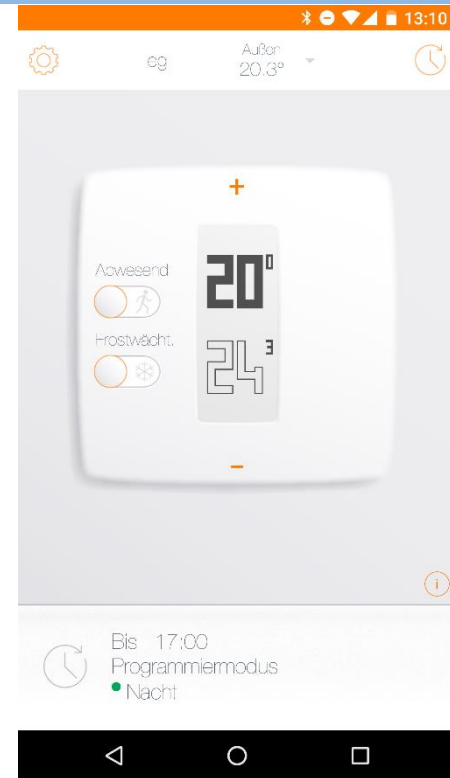
<https://www.tagesanzeiger.ch/digital/computer/android-erfolgreich-und-schon-bald-am-ende/story/12380331>

Warum Sandboxen ?



Standard App: Native App

- Geladen vom Store
- Start via Icon
- Nativer Code (Objective-C, Java)
- Läuft in der **eigenen Sandbox**
- **Voller Integritäts-Check**



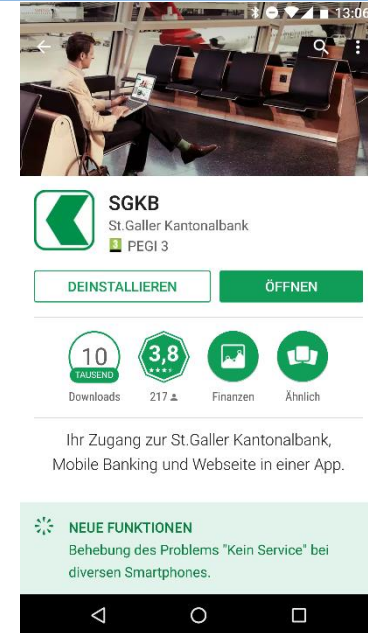
Schwierigkeiten:

- Speziallösung für jede Plattform

Zwischen-Variante: Hybride App

- Geladen vom App-Store
- Start via Icon
- Nativer Code (Rahmen)
- HTML Web-Views (on-line oder Cache)

- Läuft in der **eigenen Sandbox**
- Partieller **Integritäts-Check**



Schwierigkeiten:

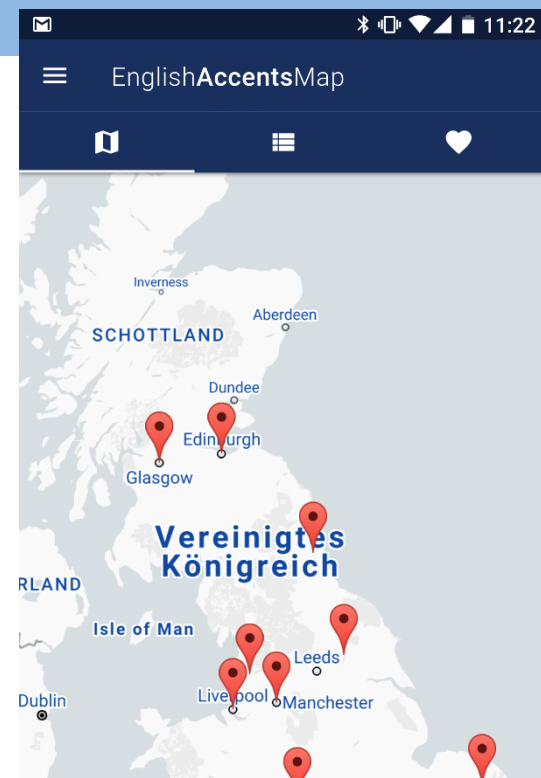
- Web-Views sind nicht voll kontrollierbar

Neuer Trend: Progressive Web App (PWA)

- Start via URL im Browser (optional via Icon)
- HTML-5, CSS3 und JavaScript online
- Browser-Caching für Offline-Funktion
- Läuft in der **Browser-Sandbox**
- **Kein Integritäts-Check**

<https://pwa.rocks/>

<https://englishaccentsmap.com>



Schwierigkeiten:

- Trennung zwischen Apps hängt vom Browser ab
- Fehlende Trennung in der Key Chain
- Fehlendes Zertifikats-Pinning

Vergleich der App-Technologien:


| Kriterium | Native | Hybrid | PWA |
|-------------------------|--------|--------|--------|
| Integritätsschutz | Hoch | Mittel | Tief |
| Kapselung: Sandbox | Hoch | Mittel | Tief |
| Kapselung: Keys | Hoch | Hoch | Tief |
| Anti-Engineering-Schutz | Hoch | Mittel | Tief |
| Flexibilität Peripherie | Hoch | Hoch | Mittel |
| Plattform-Abhängigkeit | Hoch | Mittel | Tief |
| Abhängigkeit vom Store | Hoch | Mittel | Tief |

Beachte: Die Unterschiede bei Funktionalität, Look&Feel, Offline-Fähigkeit sind nur noch gering.

Sandboxen sind nur dann voll wirksam, wenn die Apps sich korrekt verhalten.


Standard-Windows-Anwendungen

Integrität von Apps

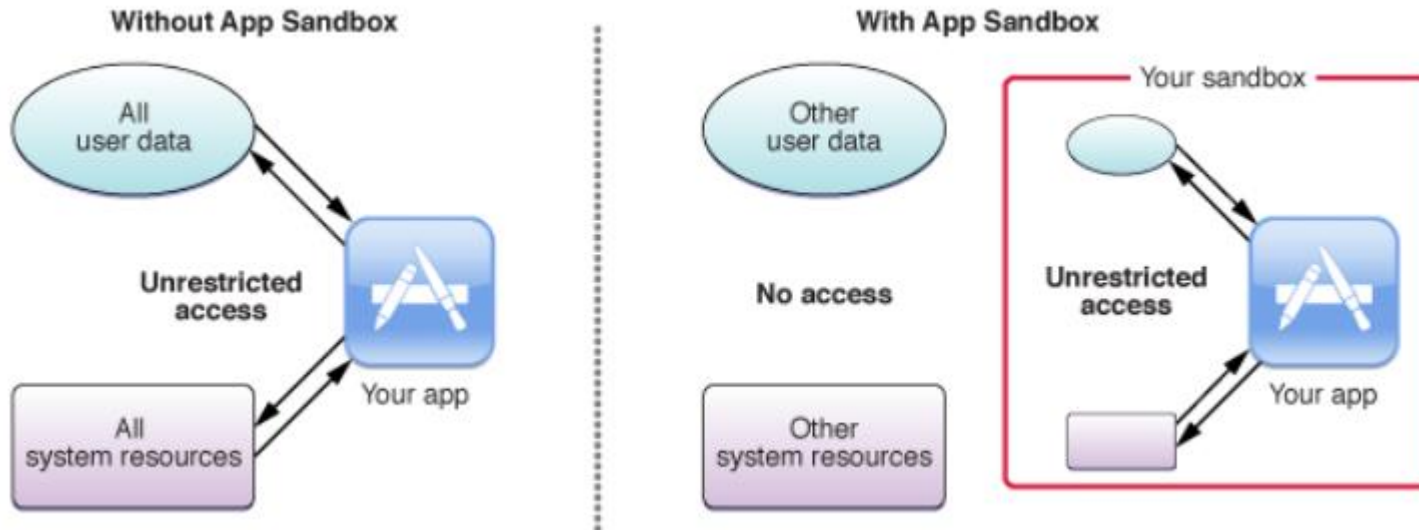
| Thema | iOS | Android |  Windows |
|-------------------------------------|---------------------------------|---------------------------------|---|
| Initialprüfung | Durch Apple | (Durch Anwender) | keine |
| Verteilung | App Store | Play Store oder direkt | Nicht kontrolliert |
| Integritätscheck bei Installation | Signatur von Apple | Eine gültige Signatur | Signatur optional |
| Check bei Update | Signatur von Apple und App-ID | Signatur des Herausgebers | Nicht systematisch |
| Integritätscheck beim Start | Keine | Keine | Keine |
| Zugriff auf Schlüsselspeicher | App-ID | App-ID | Rechte des Run-Users |
| Zugriff auf Inter-App-Kommunikation | App-ID gem. Vorgaben des Autors | App-ID gem. Vorgaben des Autors | Rechte des Run-Users |

Trusted Execution Environment

TEE stellt eine sichere Laufzeitumgebung für Applikationen zur Verfügung.

| Begriff | | iOS und Android |  Windows |
|---------|-------------------------------|---|---|
| TPM | Trusted platform module | <ul style="list-style-type: none"> • Schlüssel-Speicher, • Basis-Krypto-Funktionen • Authentisierung und Autorisierung | In Windows Standard |
| CPU | Prozessor | <ul style="list-style-type: none"> • Prozess-Isolierung | Standard |
| TEE | Trusted execution environment | <ul style="list-style-type: none"> • Peripherie-Isolierung, sichere Vertriebs-Prozesse | nur auf BIOS-Stufe |
| Sandbox | geschützte Run-Time | <ul style="list-style-type: none"> • Eigenes File-System pro App • Restriktives API, • Kontrollierte Inter-App-Kommunikation • eingeschränkter Run-User (iOS) • Eigener Run-User pro App (Android) | rudimentär |
| App | Anwendung | <ul style="list-style-type: none"> • Zusammenfassung der Funktionen (Kapselung) | teilweise |

Die iOS-Sandbox

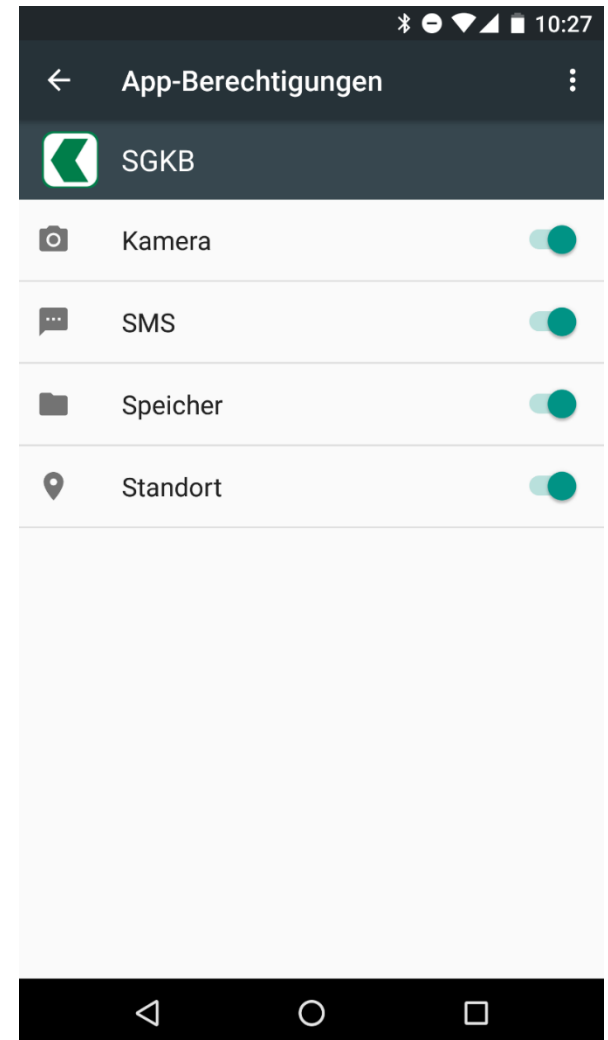
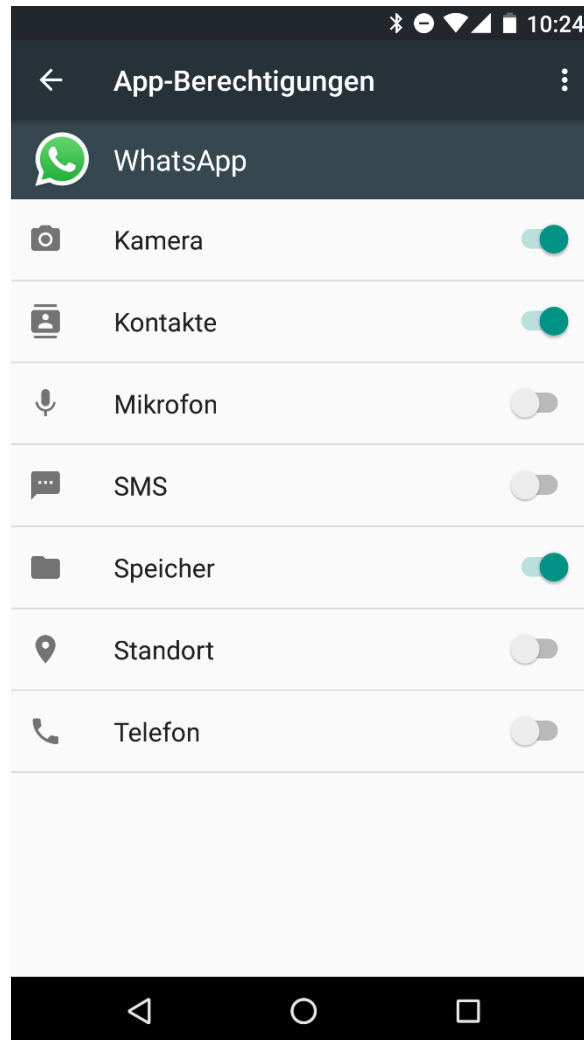


..the App Sandbox strategy is twofold:

1. App Sandbox enables you to describe how your app interacts with the system. The system then grants your app the access it needs to get its job done, and no more.
2. App Sandbox allows the user to transparently grant your app additional access by way of Open and Save dialogs, drag and drop, and other familiar user interactions.

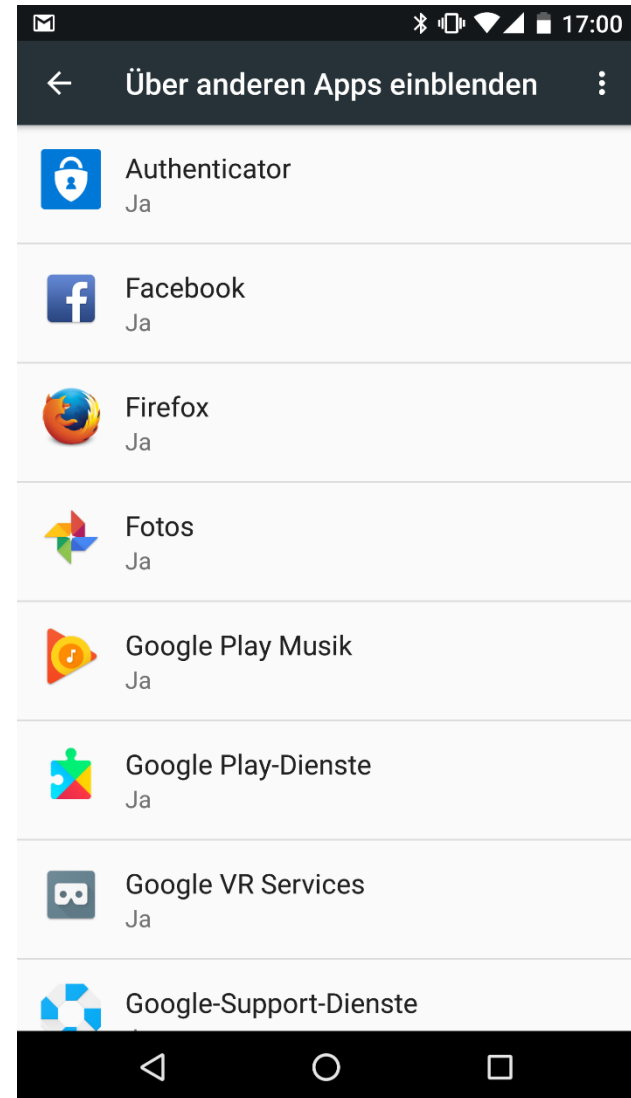
Quelle: developer.apple.com

App-Rechte in Android (2 Beispiele)



Wer hat das «SYSTEM_ALERT_WINDOW»-Recht ?

Einstellungen → Apps → «Zahnrad» →
Spezieller Zugriff → Über ändern Apps
einblenden



Keychain und KeyStore



| Thema | iOS Keychain | Android KeyStore | Bemerkungen |
|--------------|--------------|------------------|--|
| TPM-basiert | X | (X) | <ul style="list-style-type: none"> • Herstellerabhängig |
| App-Bindung | X | X | |
| PIN-Schutz | X | X | |
| FPR-Schutz | X | X | |
| Krypto-Fkt. | X | X | <ul style="list-style-type: none"> • z.B. S/MIME |
| Backup lokal | X | X | <ul style="list-style-type: none"> • PW-Schutz bei iOS • Gerätebindung möglich |
| Backup Cloud | X | X | <ul style="list-style-type: none"> • iOS: Keychain an Gerät gebunden • Android: Kein Key Store im Backup |
| Symm. Keys | X | X | |
| Asymm. Keys | X | X | |
| Strings | X | (X) | |

Thesen

1. Mobile Geräte bieten bessere Sicherheitsfunktionen als Standard-PCs. (Windows, OSX).
2. Apps können sehr sicher gemacht werden.
3. Native und Hybride Apps sind sicherer als PWAs
4. Ein paar Dinge muss man beachten:
 - Alte mobile-Geräte müssen ersetzt werden.
 - Nur Apps vom Store verwenden.
 - Updates sind wichtig.

Danke

Paul Schöbi
paul.schoebi@cnlab.ch
+41 55 214 33 33

6.9.2017