Auf dem Weg verTAN

Über die Unsicherheit von App-basierten TAN-Verfahren im Onlinebanking

Vincent Haupert

28. Dezember 2015

Lehrstuhl für IT-Sicherheitsinfrastrukturen
Friedrich-Alexander-Universität Erlangen-Nürnberg

Zwei-Faktor-Authentifizierung im Onlinebanking

- Onlinebanking ist etabliert
- Seit seiner Einführung 1980 ein Zwei-Faktor-Verfahren
  1. Benutzername/Passwort
  2. TAN-Verfahren
- Über die Jahre aus unterschiedlichen Motiven immer neue TAN-Verfahren entstanden

Zwei-Faktor-Authentifizierung im Onlinebanking

1

Login in Onlinebanking-Portal mit Benutzername/Passwort

$\Rightarrow$ Überweisung einreichen

Zwei-Faktor-Authentifizierung im Onlinebanking

1

Login in Onlinebanking-Portal mit Benutzername/Passwort

$\Rightarrow$ Überweisung einreichen

2

Bestätigung der Überweisung mit einem TAN-Verfahren.

Zwei-Faktor-Authentifizierung im Onlinebanking

1

Login in Onlinebanking-Portal mit Benutzername/Passwort

⇒ Überweisung einreichen

1) 3548533
2) 1041931
3) 4213370
4) 9438589
5) 3087452

2

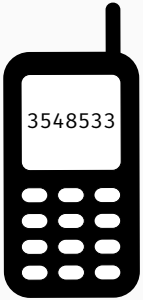
Bestätigung der Überweisung mit einem TAN-Verfahren.

Zwei-Faktor-Authentifizierung im Onlinebanking

1

Login in Onlinebanking-Portal mit Benutzername/Passwort

$\Rightarrow$ Überweisung einreichen



2

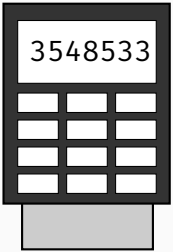
Bestätigung der Überweisung mit einem TAN-Verfahren.

Zwei-Faktor-Authentifizierung im Onlinebanking

1

Login in Onlinebanking-Portal mit Benutzername/Passwort

$\Rightarrow$ Überweisung einreichen

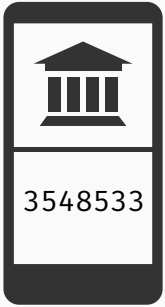


2

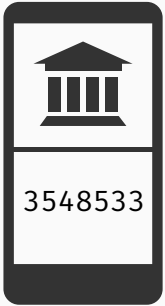
Bestätigung der Überweisung mit einem TAN-Verfahren.

Mobilebanking

App-basierte TAN-Verfahren und Mobilebanking



App-basierte TAN-Verfahren und Mobilebanking

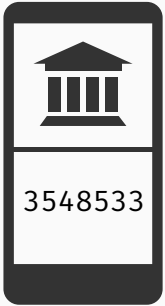


1

Login in der Banking-App mit Benutzername/Passwort

⇒ Überweisung einreichen

App-basierte TAN-Verfahren und Mobilebanking



1

Login in der Banking-App mit Benutzername/Passwort

⇒ Überweisung einreichen

2

Wechsel zur TAN-App, TAN empfangen und in die Banking-App übertragen

App-basierte TAN-Verfahren und Mobilebanking

DKB-pushTAN-App

HVB Mobile B@nking

ING-DiBa SmartSecure

VR-SecureGo

S-pushTAN-App

App-basierte TAN-Verfahren und Mobilebanking

- Malware in den offiziellen App-Stores der mobilen Betriebssysteme ist keine Fiktion

App-basierte TAN-Verfahren und Mobilebanking

- Malware in den offiziellen App-Stores der mobilen Betriebssysteme ist keine Fiktion
- 2014: Dominik Maier zeigt, dass sich der Google Play Store nicht effektiv gegen Malware schützen kann

App-basierte TAN-Verfahren und Mobilebanking

- Malware in den offiziellen App-Stores der mobilen Betriebssysteme ist keine Fiktion
- 2014: Dominik Maier zeigt, dass sich der Google Play Store nicht effektiv gegen Malware schützen kann

App-basierte TAN-Verfahren und Mobilebanking

- Malware in den offiziellen App-Stores der mobilen Betriebssysteme ist keine Fiktion
- 2014: Dominik Maier zeigt, dass sich der Google Play Store nicht effektiv gegen Malware schützen kann
- 2015: App *Brain Test* im Google Play Store mit über 100 000 Downloads
  $\Rightarrow$ rootet Gerät und lädt Schadcode nach

App-basierte TAN-Verfahren und Mobilebanking

DKB-pushTAN-App

HVB Mobile B@nking

ING-DiBa SmartSecure

VR-SecureGo

S-pushTAN-App

pushTAN-Angriff

pushTAN-Angriff

Mögliche technische Angriffe

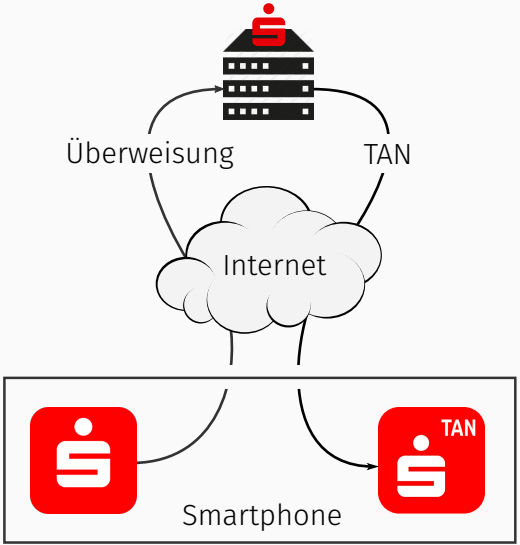
- *S-pushTAN-App* kopieren
- Protokoll der *S-pushTAN-App* reversen und eigenen Client entwickeln
- Transaktionsmanipulation

pushTAN-Angriff

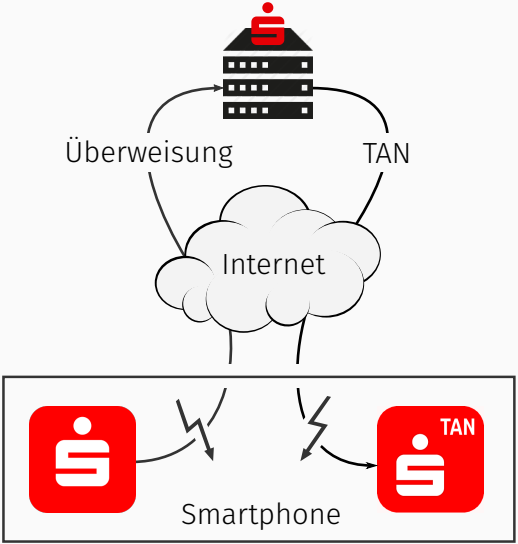
Mögliche technische Angriffe

- *S-pushTAN-App* kopieren
- Protokoll der *S-pushTAN-App* reversen und eigenen Client entwickeln
- Transaktionsmanipulation

Transaktionsmanipulation



Transaktionsmanipulation



pushTAN-Verfahren: Sicherheitsmerkmale

Sparkassen-App

`root`-Erkennung

S-pushTAN-App

`root`-Erkennung
Anti-Debugging
Device-Fingerprinting
String-Encryption
Packed Library
Anti-Hooking
...

pushTAN-Verfahren: Sicherheitsmerkmale

Sparkassen-App

S-pushTAN-App

`root`-Erkennung

...

S-pushTAN-App: Schutzmaßnahmen

- Alle Sicherheitsmerkmale werden von der nativen Bibliothek *Promon Shield* umgesetzt
- Die Bibliothek selbst ist verschlüsselt und obfuskiert
  - Wird als Erstes mit `System.load()` geladen
  - Verschiebt Strings im Java-Code in die Bibliothek

    ⇒ `public static final String fsh_650 = null`

    ⇒ `public static native String getStr(int i)`
  - Erkennt `root`, Debugger, Repackaging, ...

S-pushTAN-App: Schutzmaßnahmen

- Alle Sicherheitsmerkmale werden von der nativen Bibliothek *Promon Shield* umgesetzt
- Die Bibliothek selbst ist verschlüsselt und obfuskiert
  - Wird als Erstes mit `System.load()` geladen
  - Verschiebt Strings im Java-Code in die Bibliothek

    ⇒ `public static final String fsh_650 = null`

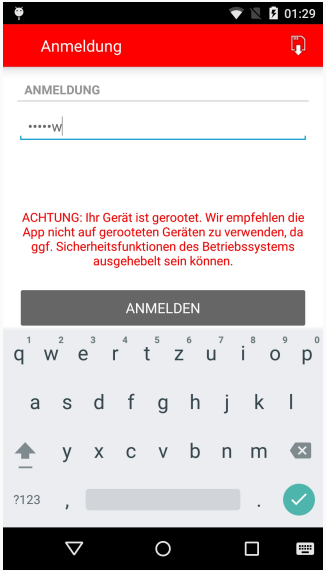
    ⇒ `public static native String getStr(int i)`
  - Erkennt `root`, Debugger, Repackaging, ...
  - ... behandelt die Erkennung aber nicht selbst

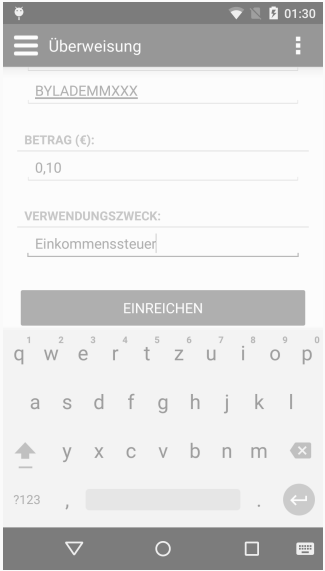
```
public interface ShieldCallbacks {
    void debuggerStatus(boolean z);
    void emulatorStatus(boolean z);
    void keyboardStatus(boolean z);
    void nativeCodeHooksStatus(boolean z);
    void repackagingStatus(boolean z);
    void rootingStatus(boolean z, int i);
    void screenreaderStatus(boolean z);
}

public class aqh implements ShieldCallbacks {
    public void rootingStatus(boolean z, int i) {
        if (z || i > anl.b())
            a(anm$f.rooted_texthinweis, i);
    }
    /* ... */
}
```

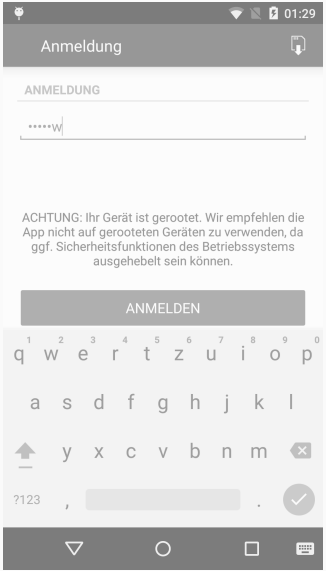
Demo Transaktionsmanipulation

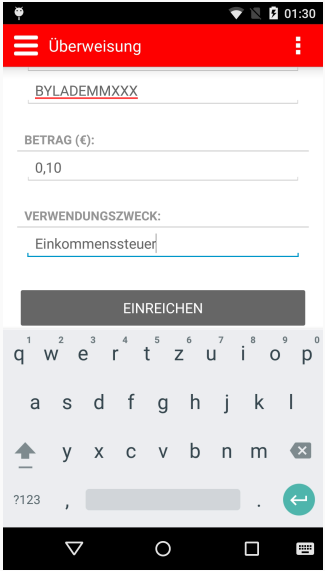
Transaktionsmanipulation pushTAN-Verfahren



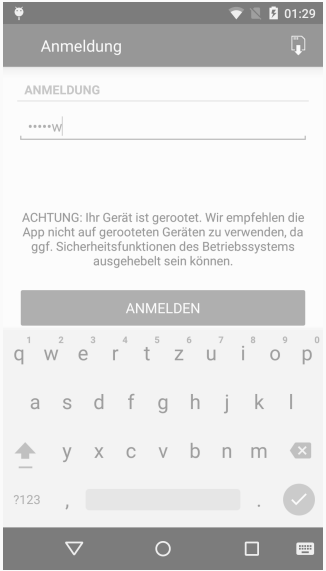


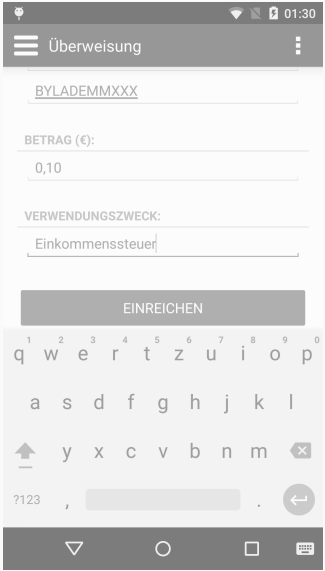
Transaktionsmanipulation pushTAN-Verfahren

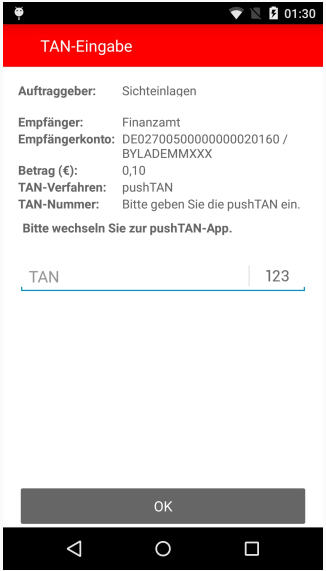




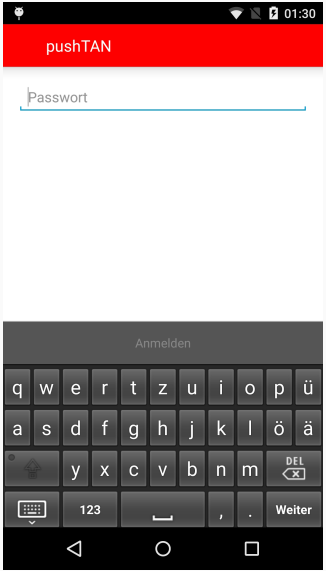
Transaktionsmanipulation pushTAN-Verfahren

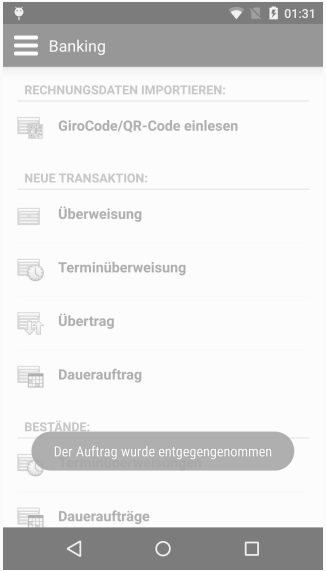




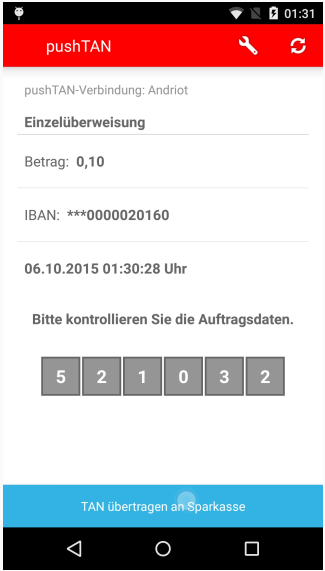


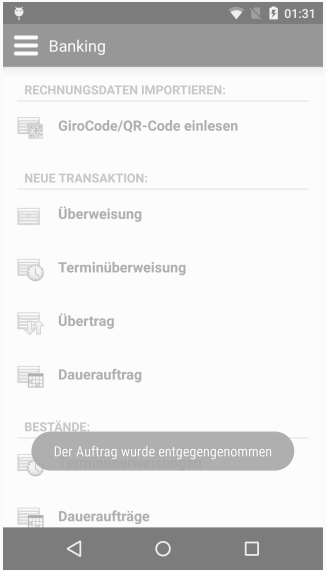
Transaktionsmanipulation pushTAN-Verfahren



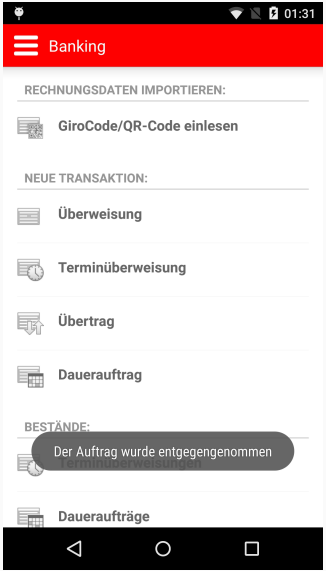


Transaktionsmanipulation pushTAN-Verfahren

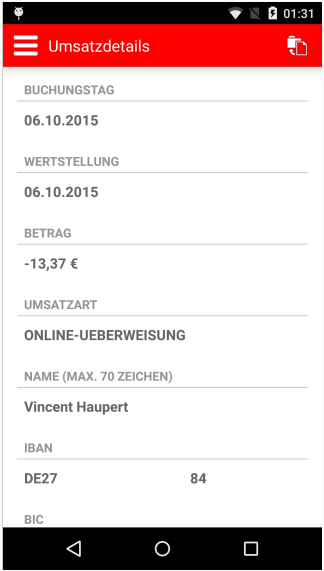


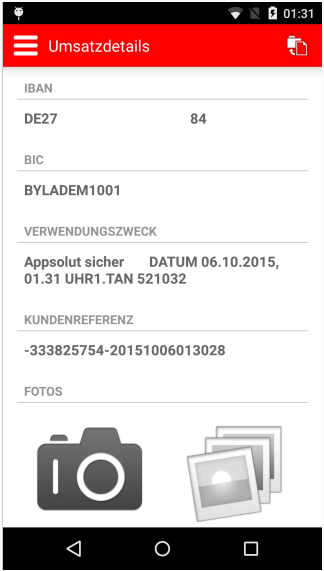


Transaktionsmanipulation pushTAN-Verfahren



Transaktionsmanipulation pushTAN-Verfahren





Reaktion der Sparkasse

„Die beschriebenen unter Laborbedingungen durchgeführten Manipulationen betreffen veraltete Versionsstände der S-pushTAN-App. [...] Ab der Version 1.0.5, die seit 16.10.2015 im Google-Playstore verfügbar ist, ist das beschriebene Angriffsverfahren zur Manipulation nicht mehr möglich.“

— Stellungnahme des DSGV, 25. Okt ’15

Reaktion der Sparkasse

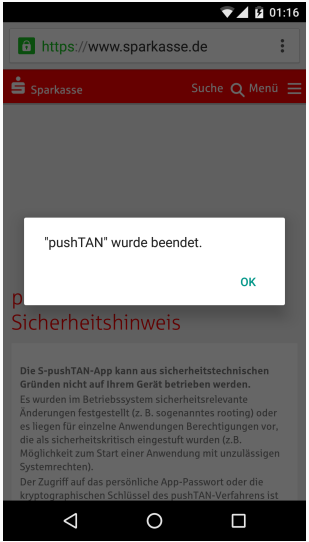
„Diese Version macht unsere bestehenden Lösungsansätze [...] zunächst unwirksam [...]. Dennoch möchten wir darauf hinweisen und davor warnen, dass auch diese Version der S-pushTAN-App – und alle zukünftigen Versionen – mit entsprechendem Mehraufwand gebrochen werden kann.“

— Ausarbeitung Haupert & Müller, 22. Okt ’15

Die S-pushTAN-App ab Version 1.0.5

Die S-pushTAN-App ab Version 1.0.5

- Keine Java-Callbacks mehr
  ⇒ stürzt stattdessen ab und zeigt eine Seite im Browser an
  ⇒ `root`-Erkennung kann nicht mehr trivial deaktiviert werden
- Bekannte Hooking Frameworks werden erkannt
  ⇒ Xposed funktioniert nicht mehr ohne Weiteres



Um den Angriff auf die neue Version zu portieren muss

1. die `root`- und
2. die `Xposed`-Erkenunng

umgangen werden

Wie funktioniert die root-Erkennung?

Die S-pushTAN-App ab Version 1.0.5: root

```
access("/dev/com.koushikdutta.superuser.daemon")
access("/system/app/Superuser.apk")
access("/system/bin/.ext/.su")
access("/system/bin/su")
access("/system/etc/.has_su_daemon")
access("/system/etc/.installed_su_daemon")
access("/system/etc/init.d/99SuperSUDaemon")
access("/system/xbin/daemonsu")
access("/system/xbin/su")
```

Die S-pushTAN-App ab Version 1.0.5: root

```
access("/dev/com.koushikdutta.superuser.daemon")
access("/system/app/Superuser.apk")
access("/system/bin/.ext/.su")
access("/system/bin/vince")
access("/system/etc/.has_su_daemon")
access("/system/etc/.installed_su_daemon")
access("/system/etc/init.d/99SuperSUDaemon")
access("/system/xbin/daemonsu")
access("/system/xbin/vince")
```

Die S-pushTAN-App ab Version 1.0.5: root

```
access("/dev/com.koushikdutta.superuser.daemon")
access("/system/app/Superuser.apk")
access("/system/bin/.ext/.su")
access("/system/bin/vince")
access("/system/etc/.has_su_daemon")
access("/system/etc/.installed_su_daemon")
access("/system/etc/init.d/99SuperSUDaemon")
access("/system/xbin/daemonsu")
access("/system/xbin/vince")
```

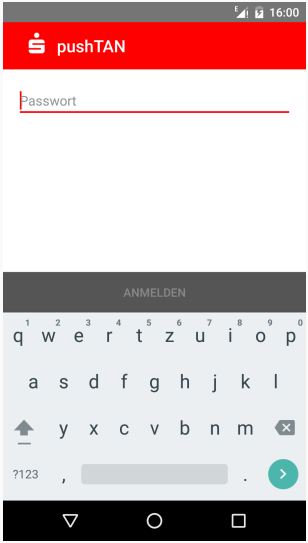
In Oslo unbekannt: PATH

Die S-pushTAN-App ab Version 1.0.5: root

```
access("/dev/com.koushikdutta.superuser.daemon")
access("/system/app/Superuser.apk")
access("/system/bin/.ext/.su")
access("/system/bin/vince")
access("/system/etc/.has_su_daemon")
access("/system/etc/.installed_su_daemon")
access("/system/etc/init.d/99SuperSUDaemon")
access("/system/xbin/daemonsu")
access("/system/xbin/vince")
```

In Oslo unbekannt: PATH

⇒ Systemless su



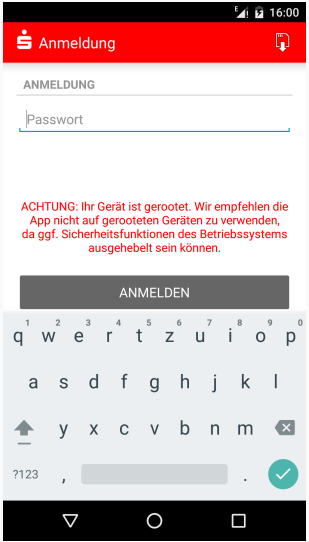
Die S-pushTAN-App ab Version 1.0.5: root

```
access("/dev/com.koushikdutta.superuser.daemon")
access("/system/app/Superuser.apk")
access("/system/bin/.ext/.su")
access("/system/bin/vince")
access("/system/etc/.has_su_daemon")
access("/system/etc/.installed_su_daemon")
access("/system/etc/init.d/99SuperSUDaemon")
access("/system/xbin/daemonsu")
access("/system/xbin/vince")
```

In Oslo unbekannt: **PATH**

⇒ Systemless **su**

Die *Sparkassen*-App macht es besser



Wie funktioniert die Xposed-Erkennung?

Die S-pushTAN-App ab Version 1.0.5: Xposed

```
access("/data/data/de.robv.android.xposed.installer")
access("/system/bin/app_process.orig")
access("/system/bin/app_process32_xposed")
access("/system/bin/app_process64_xposed")
access("/system/framework/XposedBridge.jar")
access("/system/lib/libxposed_art.so")
access("/system/xposed.prop")
```

Die S-pushTAN-App ab Version 1.0.5: Xposed

~~access("/data/data/de.robv.android.xposed.installer")~~
~~access("/system/bin/app_process.orig")~~
~~access("/system/bin/app_process32_xposed")~~
~~access("/system/bin/app_process64_xposed")~~
access("/system/framework/XposedBridge.jar")
access("/system/lib/libxposed_art.so")
access("/system/xposed.prop")

Die S-pushTAN-App ab Version 1.0.5: Xposed

~~access("/data/data/de.robv.android.xposed.installer")~~
~~access("/system/bin/app_process.orig")~~
~~access("/system/bin/app_process32_xposed")~~
~~access("/system/bin/app_process64_xposed")~~
access("/system/framework/SpkminBridge.jar")
access("/system/lib/libspkmin_art.so")
access("/system/spkmin.prop")

Die S-pushTAN-App ab Version 1.0.5: Xposed

~~access("/data/data/de.robv.android.xposed.installer")~~
~~access("/system/bin/app_process.orig")~~
~~access("/system/bin/app_process32_xposed")~~
~~access("/system/bin/app_process64_xposed")~~
access("/system/framework/SpkminBridge.jar")
access("/system/lib/libspkmin_art.so")
access("/system/spkmin.prop")

Reicht noch nicht ganz:

open("/proc/self/exe") ⇒ sucht nach „[X|x]posed“

Die S-pushTAN-App ab Version 1.0.5: Xposed

~~access("/data/data/de.robv.android.xposed.installer")~~
~~access("/system/bin/app_process.orig")~~
~~access("/system/bin/app_process32_xposed")~~
~~access("/system/bin/app_process64_xposed")~~
access("/system/framework/SpkminBridge.jar")
access("/system/lib/libspkmin_art.so")
access("/system/spkmin.prop")

Reicht noch nicht ganz:

open("/proc/self/exe") ⇒ sucht nach „[X|x]posed“

⇒ letztendlich neu kompiliert

Video: S-pushTAN-App Version 1.0.7

Fazit

App-basierte TAN-Verfahren sind konzeptionell schwach

- Die Schutzmechanismen der *S-pushTAN-App* lassen sich in der aktuellen Version zumindest nicht mehr trivial deaktiveren
- Es bleibt ein Katz-und-Maus-Spiel, das die Sparkasse am Ende immer verlieren wird
- Die `root`-Erkennung, etc. bringen der Sparkasse hauptsächlich verärgerte Nutzer ein, statt gegen echte Angriffe zu schützen

Ende

Offizielle Stimmen

„Die TAN sollte auf keinen Fall auf demselben Smartphone generiert werden, auf dem das Online-Banking stattfindet. Hat ein Betrüger das Smartphone gehackt, so kann er dadurch auf beide Verfahren zugreifen.“

— BaFin Journal, Aug ’15

„Allerdings muss das App-basierte Sicherungsverfahren und das tatsächliche Online-Banking unabhängig voneinander – also über verschiedene Geräte (i.S.v. verschiedene Kanäle) – erfolgen.“

— FAQ zu MaSI, 28. Okt ’15

Bewertungen im Google Play Store

amilo 9000 16. Oktober 2015

★☆☆☆☆

Schwer enttäuscht Ich freue mich schon auf unsere Kunden, denen ich S-pushTan empfohlen habe, die jetzt gerootet sind. Selbst ich bin geschädigt und habe keine Erklärung für diesen Mist seit dem 16.10.2015.

D anny 16. Oktober 2015

★☆☆☆☆

Kein Öffen möglich sofortiger Absturz

Friederich Loheide 16. Oktober 2015

★☆☆☆☆

Stürzt direkt ab Stürzt direkt ab... Angeblich wäre mein Handy gerootet, was aber nicht stimmt

Klaus Kendel 16. Oktober 2015

★☆☆☆☆

Seit Update geht's nicht mehr, stürzt sofort ab. Bitte beheben.

Oliver M 16. Oktober 2015

★☆☆☆☆

Seit dem heutigem Update wird Behauptet das mein Handy gerootet ist. Die App stürzt ab und ich werde auf eine Seite verwiesen mit genau dieser Behauptung.

Jürgen Fischer 16. Oktober 2015

★☆☆☆☆

Funktioniert nicht mehr Stürzt seit Update beim Start ab. Habe keine Lust mehr, mich auf die Willkür dieser App zu verlassen - lasse schleunigst wieder auf smsTan umstellen.