Fraunhofer
SIT

Appicaptor Security Index 9/2015

Ergebnisauszug für iOS & Android Apps

Fraunhofer-Institut für
Sichere Informationstechnologie (SIT)

6. Oktober 2015

„Investition in Ihre Zukunft'

Investitionen für diese Entwicklung wurden
von der Europäischen Union aus dem Europäischen Fonds
für regionale Entwicklung und vom Land Hessen kofinanziert.

Fraunhofer SIT Kontaktperson
Dr. Jens Heider
Fraunhofer-Institut für Sichere Informationstechnologie (SIT)
Rheinstraße 75, 64295 Darmstadt, Germany
E-Mail: jens.heider@sit.fraunhofer.de
Telefon: +49 (0) 61 51/869-233
Fax: +49 (0) 61 51/869-224

Inhaltsverzeichnis

1 Einleitung **4**

2 App-Tests **5**
2.1 App-Auswahl 5
2.2 Risikokategorien 6
2.2.1 Privatsphärenrisiken 6
2.2.2 Kommunikationsrisiken 7
2.2.3 Externe Risiken 8
2.2.4 Risiken für Datenlecks 8
2.2.5 Risiken verdächtiger Funktionalität 8
2.2.6 Kryptographische Risiken 9
2.2.7 Risiken für Credentials 9

3 Ergebnisse **10**
3.1 Appicaptor Risiko Charts 10
3.2 Implementierungsfehler 13
3.3 Tracking 17

1 Einleitung

Mit dem freien Zugang zu App-Märkten ist eine neue Situation im Umgang mit Unternehmenssoftware entstanden. Die Auswahl der Apps für das Dienst-Smartphone kann direkt durch den Mitarbeiter mittels der verfügbaren App-Märkte erfolgen. Eine eingehende Überprüfung der Eignung der Apps für den Unternehmenseinsatz findet zumeist nicht statt.

Die auf Smartphones installierten Apps kommen im Arbeitsalltag in Kontakt mit schützenswerten Unternehmensdaten. Mitarbeiter können jedoch die daraus resultierenden Risiken meist nur unzureichend einschätzen und den Administratoren fehlen häufig klare Auswahlkriterien für einen App-Freigabeprozess. Es fehlt dann ein kosten- und ressourceneffizientes Verfahren zur Sicherheitsbewertung von Apps, das auf objektiven Sicherheitskriterien beruht und den enormen Menge an Apps, die in Unternehmen Einsatz finden (könnten), gerecht werden kann.

Um einen Einblick in das gegenwärtige Sicherheitsniveau beliebter Apps zu geben, werden im folgenden Ergebnisauszüge von iOS & Android App-Massentests vorgestellt. Dazu wird zunächst eine kleine Auswahl relevanter Sicherheitskriterien für den Unternehmenseinsatz beschrieben, deren Auswertung automatisiert mit dem am Fraunhofer SIT entwickelten Analysewerkzeug *Appicaptor*[1] erfolgte.

Appicaptor erstellt als Dienstleistung zu jeder App einen individuellen Testbericht. Die Analyse läuft dabei vollständig automatisch. Bei Feststellung von Sicherheitslücken oder einer unsicheren Verwendung schützenswerter Daten protokolliert das System diese Risiken. Da Apps regelmäßig überarbeitet werden und sich immer wieder neue Erkenntnisse zu Schwachstellen und Implementierungsfehlern ergeben, wiederholt Appicaptor regelmäßig die Tests und bewertet die Sicherheitseigenschaften dadurch stets anhand des aktuellen technischen Wissens.

Mithilfe von Appicaptor können Firmen entweder eine Whitelist oder eine Blacklist erstellen. Eine Whitelist enthält unbedenkliche Apps, die Mitarbeiter auf den Smartphones nutzen können. Eine Blacklist enthält die Apps, die nicht die IT-Sicherheitsrichtlinien des Unternehmens erfüllen. Weiterhin können Unternehmen selbst entwickelte Apps oder Apps aus firmeneigenen App-Stores regelmäßig automatisiert auf Schwachstellen überprüfen lassen.

[1] siehe: `http://www.appicaptor.de`

2 App-Tests

Für die Analyse der Apps durch Appicaptor werden die vollständigen App-Binärdaten automatisiert aus den App-Märkten heruntergeladen und in eine Zwischensprache zurückübersetzt. Für den vorliegenden Bericht stammen die Apps aus den deutschen App-Märkten. Nach dem Download erfolgt eine Vielzahl automatisierter Tests und Analysen, für die das Appicaptor Framework Werkzeuge auf unterschiedlichen Ebenen zur Verfügung stellt und mit dem weiterhin die Einzelbefunde zu konsistenten Ergebnissen korreliert werden. Für den vorliegenden Bericht wurden statische Analyseverfahren verwendet. Damit wird sichergestellt, das der gesamte App-Code untersucht wird. Es erfolgt damit eine gewisse Überapproximation die alle Möglichkeiten der App bewertet, unabhängig davon, welche Bedingungen zum Auslösen einer Funktion notwendig sind.

2.1 App-Auswahl

Die Auswahl der analysierten Apps erfolgte anhand der Einstufung der Beliebtheit durch die App-Märkte. Es wurden pro Plattform jeweils die beliebtesten 200 Apps der folgenden 10 Kategorien monatlich mit Appicaptor analysiert.

iOS App Store Kategorien	**Android Play Store Kategorien**
• Productivity	• Productivity
• Utilities	• Tools
• Business	• Business
• Socialnetworking	• Social
• Finance	• Finance
• News	• News and Magazines
• Lifestyle	• Lifestyle
• Entertainment	• Entertainment
• Travel	• Travel and Local
• Weather	• Communication

2.2 Risikokategorien

Generell sind viele Appicaptor Testergebnisse direkte Erkenntnisse, die autark genutzt werden können, um die Sicherheitsqualität der App zu bewerten. Viele der Testergebnisse bringen jedoch noch mehr Nutzen, wenn sie gemeinsam mit anderen Appicaptor Testergebnissen interpretiert werden. Mit integrierten Appicaptor-Funktionen können effizient Regelsätze modelliert werden, mit deren Hilfe unternehmensspezifisch Apps bewertet und gekennzeichnet werden, ob diese Apps konform mit der jeweiligen Unternehmens-IT-Sicherheitsrichtlinie sind. Die unternehmensspezifischen Sicherheitsanforderungen werden auf Appicaptor Regelsätze abgebildet, die definieren welche Kombination von App-Eigenschaften eine vertrauenswürdige bzw. nicht vertrauenswürdige App kennzeichnet.

Für die vorliegende Analyse der Apps wurden die automatisiert detektierbaren Ergebnisse anhand der verschiedenen Risiken kategorisiert. Je nach Art des Befunds, kann dieser auf mehrere Risiken deuten. Im Folgenden werden die für die Appicaptor Charts ausgewählten Kategorien und Beispiele für die zugrundeliegenden Befunde im Detail beschrieben.

2.2.1 Privatsphärenrisiken

In dieser Analyse werden Befunde im Bezug auf eine unangemessene Preisgabe von Benutzerinformationen als Privatsphärenrisiken klassifiziert. Im Folgenden werden einige Beispiele erläutert:

- Werbung/Tracking: Die App benutzt mehr als 5 Werbe- oder Tracking-Anbieter und verbreitet dadurch persönliche Daten in Kombination mit dem Kontext der App-Nutzung.
- Nicht plausibler Sensorzugriff: Die Nutzung von Smartphone-Sensoren (z.B. Mikrofone, GPS, Kamera, etc.), außerhalb der für den detektierten App-Typ üblichen Gebrauch, birgt ein Risiko bezüglich des Zugriffs auf sensible persönliche Daten.
- Informationspreisgabe: Die Offenlegung von Standort-Daten oder Informationen über Web-Suchanfragen durch *ungeschützte* Kommunikation mit einem Dienstleister (z.B. Google), sodass neben dem Dienstleister auch Dritte die übermittelten Informationen einsehen können.
- Benutzer-Identifikation: Die App versucht Zugang zu Informationen zu erlangen, anhand welcher ein Benutzer eindeutig identifiziert werden kann, wie beispielsweise Telefonnummer oder eindeutige Gerätenummern. Der Zugriff widerspricht dabei der App-Beschreibung (z.B. Taschenlampe).

- (iOS) Nutzung undokumentierter APIs: Die App beinhaltet Code, der Funktionen unveröffentlichter bzw. undokumentierter APIs aufruft. Diese Art der Programmierung beinhaltet das Risiko, dass diese APIs kritische Funktionen bereitstellen, die lediglich von Apps von Apple Inc. genutzt werden sollten. Apple erklärte, dass Apps die solche APIs benutzen vom App Store ausgeschlossen werden (siehe Abschnitt 2.5 App Store Review Guidelines[1]). Dennoch findet Appicaptor regelmäßig Apps mit dieser Funktionalität im iOS App Store.

2.2.2 Kommunikationsrisiken

Kommunikationsrisiken beziehen sich auf den fehlenden, schwachen oder fehlerhaften Schutz der Geheimhaltung und Integritätssicherung von Informationen während eines Informationsaustauschs mit externen Quellen. Gründe für die Einstufung einer App in dieser Kategorie sind beispielsweise folgende:

- SSL Schwachstelle: Die App beinhaltet unsicheren Code zum Schutz der Kommunikation mit SSL/TLS. Oftmals ist unsicherer Code die Ursache für fehlerhaften Schutz vor Man-in-the-Middle Angriffen. Die Umsetzung korrekter Secure Socket Layer (SSL) oder Transport Layer Security (TLS) Kommunikation kann in der App-Entwicklung prinzipiell einfach mit den Standardfunktionen des Smartphone-Betriebssystems durchgeführt werden. In der Entwicklungsphase einer Smartphone-App wird jedoch die SSL/TLS-Konfiguration oder ihre Prozesse häufig modifiziert, um das Debugging oder die Funktion in einer Testumgebung ohne gültige Zertifikate zu ermöglichen. Dies wird benötigt, wenn die Test-Umgebung oder -weitaus schlimmer- die Produktivumgebung kein Server-Zertifikat verwendet, das durch eine Certificate Authority (CA) unterzeichnet wurde. App-Entwickler lösen dieses Problem durch die Deaktivierung oder Änderung der SSL/TLS-Sicherheitsmaßnahmen.
- Ungeschützte Kommunikation: Die Verwendung des ungeschützten HTTP-Protokolls zur Übertragung von Parametern oder zur Abfrage von Inhalten von Servern, welche eigentlich fähig wären eine geschützte HTTPS-Kommunikation aufzubauen. Oft wird von den Entwicklern argumentiert, dass der ungeschützte Zugriff über HTTP nicht problematisch sei, da die übertragenen Informationen nicht vertraulich wären. Dies berücksichtigt jedoch nicht, dass ein Angreifer jede ungeschützte Kommunikation nicht nur lesen sondern auch manipulieren kann. Dies gibt einem potentiellen Angreifer die Möglichkeit, Server-Anfragen oder -Antworten zu verändern (oder mit eigenen Funktionen zu ergänzen) und damit die auswertende App-Umgebung zu einem anderen Verhalten (im Bezug auf das Verhalten mit unmodifizierten Daten) zu bewegen. Dies kann u.a.

[1] siehe: `https://developer.apple.com/app-store/review/guidelines/#functionality`

verwendet werden, um das Vertrauen des Benutzers in eine App auszunutzen, bspw. durch eine hinzugefügte Dialogbox mit Passwortabfrage, deren Eingaben an den Angreifer gesendet werden.

- Implementierungsfehler: Fehlender oder mangelhafter Schutz gegen Injection-Angriffe. Angreifer können dann Daten manipulieren, die durch den Implementierungsfehler als Programminstruktionen verstanden werden und das Verhalten der App ändern können.

2.2.3 Externe Risiken

Apps werden mit externen Risiken markiert, wenn

- ein Angreifer von außerhalb des Smartphones in der Lage ist eine App-Schnittstelle, -Funktionalität oder -Kommunikation auszunutzen, ohne physischen Zugriff auf das Gerät erlangt oder das Betriebssystem zuvor manipuliert zu haben oder
- Schutzmaßnahmen durch den Entwickler deaktiviert wurden, die einen Angriff hätten abwehren können.

2.2.4 Risiken für Datenlecks

Daten, die während der Verarbeitung oder im gespeicherten Zustand unzureichend geschützt sind, verursachen Risiken für Datenlecks. Daher werden Apps auf Schwachstellen und schwachen Implementierungen hin untersucht, die zu unsicherer Datennutzung führen. Beispiele dafür sind:

- Die Verarbeitung von Unternehmensdaten ohne Nutzung von Verschlüsselungsfunktionen der Plattform.
- Der Verzicht auf die Nutzung von sicheren Schlüsselspeichern.
- Eine unsichere Nutzung der Zwischenablage.

2.2.5 Risiken verdächtiger Funktionalität

Apps die Code-Muster beinhalten, die nicht direkt als bösartig eingestuft werden können, allerdings auf ungewollte bzw. versteckte Funktionalitäten hinweisen, werden als verdächtig markiert. Dies bezieht sich unter anderem auf

- Techniken zum Verbergen von Kommunikationsendpunkten durch anonymisierende Proxys,
- die übermäßige Benutzung von Werbung- und Tracking-Bibliotheken (mehr als 10) oder

- Zugriff auf Daten, die nicht dem üblichen Verhalten des App-Typ entsprechen.

2.2.6 Kryptographische Risiken

Die korrekte Benutzung von kryptographischen Funktionen ist wichtig, um das benötigte Sicherheitslevel sicherzustellen. Kryptographische Risiken beschreiben Schwächen bei Apps die beispielsweise

- bekannte unsichere Implementierungen von kryptographischen Funktionen aufweisen,
- veraltete bzw. schwache Algorithmen verwenden oder
- zu kurze kryptographische Schlüssel benutzen.

2.2.7 Risiken für Credentials

Die sichere Handhabung von Credentials durch die Implementierung ist essenziell für die allgemeine Sicherheit von Apps. Sollten Credentials von Apps ungeschützt gespeichert oder übertragen werden, kann dies zu Zugriffen durch nicht autorisierte Gruppen führen. Sollten Muster für die unsichere Behandlung von Credentials entdeckt werden, werden diese Apps mit Risiken für Credentials eingestuft.

3 Ergebnisse

3.1 Appicaptor Risiko Charts

Die hier präsentierten Appicaptor Charts (siehe Abbildung 3.1, 3.4) stellen auf der linken Seiten den prozentualen Anteil der beschriebenen Eigenschaft an der Gesamtmenge der getesteten Apps als Balkendiagram dar. In die Gesamtbewertung als ungeeignet für den Unternehmenseinsatz fließen dabei alle Apps mit mindestens eines der beschriebenen Risiken ein.

Um den Anteil von verschiedenartiger Risiken innerhalb einer App ablesen zu können, schließt sich an das Balkendiagramm ein Kreisbalken an. Der farblich hervorgehobene Teil stellt den Anteil der Risiken an den ebenfalls gefundenen anderen Risiken dar. Die Bereiche am Ende jedes Kreisbalkens stellen dabei den Anteil der Apps an der Gesamtzahl dar, die nicht die darüber dargestellten Risiken aufweisen. Unterhalb des so resultierenden Kreissegmentes befinden sich die Angaben zu der Verteilung der Apps die gleichzeitig das Risiko des äußersten farbigen Kreissegments aufweisen. So addieren sich letztlich die einzelnen Anteile wieder zum Gesamttortenstück der als ungeeignet für den Unternehmenseinsatz gekennzeichneten Apps auf.

Von den getesteten 2.000 iOS Apps (siehe Abbildung 3.1), bestehend aus den jeweils 200 Top-Apps der 10 App-Kategorien im iOS App-Store (siehe Abschnitt 2.1), enthalten beispielsweise 262 Apps Risiken für Datenlecks (13,1%) und gleichzeitig externe Risiken (13,1%), nicht aber Privatsphärenrisiken. Von diesen 262 Apps enthalten aber 232 (11,8%) ebenfalls Kommunikationsrisiken und 6 (0,3%) enthalten zudem verdächtige Funktionalität, jeweils bezogen auf die Gesamtzahl von 2.000 getesteten Apps.

Der Appicaptor Chart für Android in Abbildung 3.4 ließt sich auf die selbe Weise. Allerdings ist zu beachten, dass die prozentualen Anteile der Risiken nicht direkt mit denen der iOS Analyse verglichen werden können, da die Anzahl der Testfälle pro Risikokategorie nicht gleich groß ist. Die Ursache dafür liegt in den Unterschieden der iOS und Android Programmierung. Die automatisierte Analyse von iOS Apps ist komplexer aufgrund der fehlenden guten Unterstützung einer Zwischenrepräsentation des analysierten App-Codes und einer signifikant schlechteren Werkzeugunterstützung der Analyse verglichen zu den unzähligen (wissenschaftlichen und kommerziellen) Arbeiten im Bereich der Android Sicherheitsanalysen. Testfälle für Android lassen sich damit schneller realisieren und sind damit auch für weniger häufige Schwächen wirtschaftlich umsetzbar.

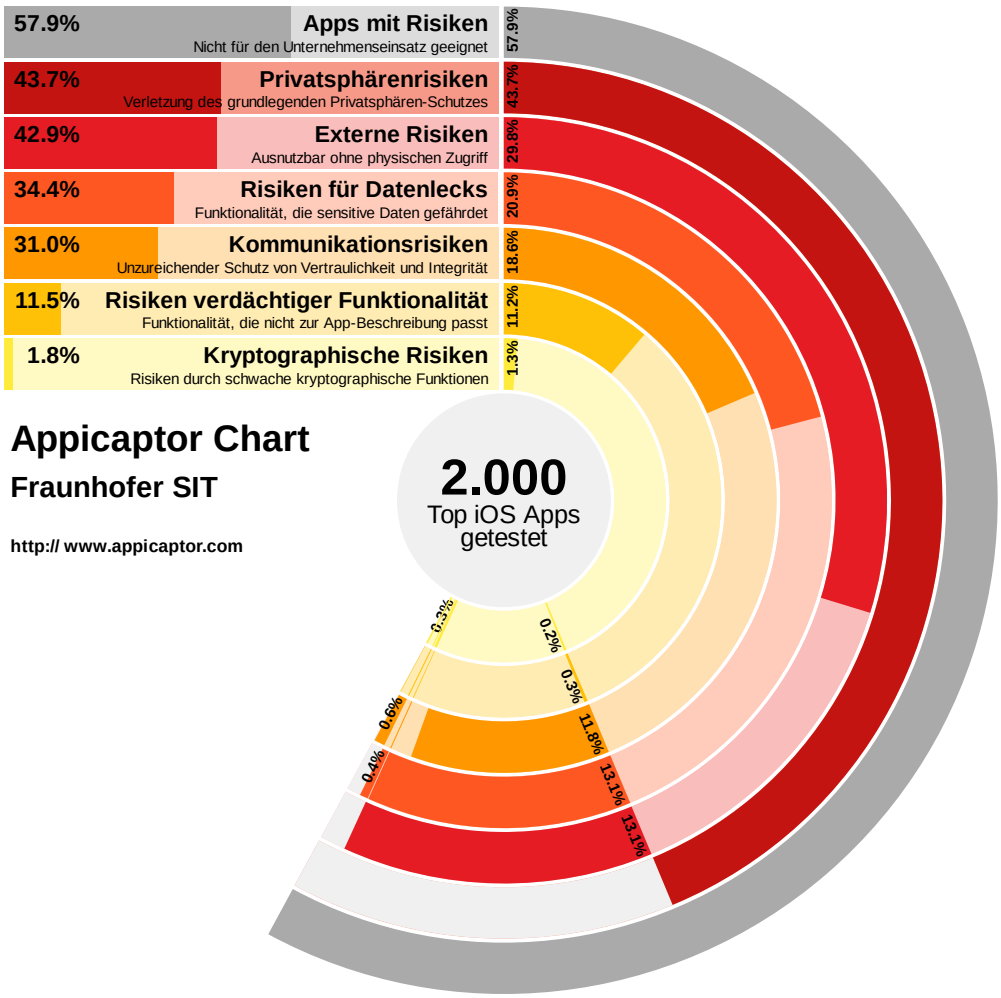

Abbildung 3.1: Risiken der kostenlosen Top 2.000 iOS Apps (Appicaptor, September 2015)

Durch diese Unterschiede in der Analyse sollten die dargestellten Ergebnisse von iOS und Android Apps nicht verglichen werden und können nicht für eine Aussage genutzt werden, welche der Plattformen sicherer ist. Allerdings zeigen die Ergebnisse sehr deutlich, dass in sicherheitsrelevanten Umgebungen Smartphone Apps nur nach gründlicher Analyse der Implementierung eingesetzt werden sollten, da die große Mehrzahl der Apps nicht einmal grundlegende Sicherheitsanforderungen erfüllen.

iOS

Zunächst lässt sich positiv für iOS Apps feststellen, dass die Strategie von Apple, bei (einigen wichtigen) Sicherheitsfunktionen/-einstellungen von einem Opt-In- zu einem Opt-Out-Modell zu wechseln, zu einer Verbesserung der Gesamtsicherheit geführt hat. So sind beispielsweise die für die Härtung von Apps wichtigen Compiler-Flags in den aktuellen SDKs bereits standardmäßig aktiviert. Wie der Vergleich von Abbildung 3.2 und 3.3 zeigt, sind inzwischen 86,24% der Apps mit der Base SDK Version 8 erstellt worden und die Mehrzahl der Apps setzt

das sicherere iOS 7 Betriebssystem voraus. Im Vergleich zum Vorjahr war die Verteilung von Base-SDK und Ziel-Plattform deutlich mehr gestreut. Apple hat es somit in diesem Jahr geschafft mehr Entwickler zur Aktualisierung zu bewegen.

Die Häufigkeit von Privatsphärenrisiken liegt in etwa gleich auf mit den externen Risiken. Allerdings ermöglicht der Appicaptor Chart auch eine Aussage über die Koexistenz dieser Risiken in Apps. Lässt man beispielsweise die Privatsphärenrisiken für das eigene Unternehmen vollständig außer Acht, reduziert sich das Gesamtrisiko nur um 13,9% auf 44,0%. Zu beachten ist zudem, das 858 Apps direkt ohne physischen Zugriff in der aktuellen Analyse angreifbar waren. Welchen Nutzen ein Angreifer daraus ziehen kann, hängt wesentlich von der Funktion der angegriffenen App ab, da das Sandbox-Prinzip den Schaden auf den Kontext der App begrenzen kann. Daher ist es wichtig festzustellen, mit welchen Daten und Zugriffsrechten eine kompromittierbare App arbeiten kann, um das Schadenspotential für das Unternehmen abzuleiten. Die vorliegende Risikobewertung geht von einer worst-case Betrachtung aus und legt zugrunde, dass jeweils unternehmenskritische Daten entsprechend der detektierten App-Funktionalität mit dieser verarbeitet werden. Darüber hinaus hat ein Angreifer aber immer die Möglichkeit die Reputation einer verwundbaren App für Social Engineering Angriffe auszunutzen.

Kommunikationsrisiken wurden mit 31% nach wie vor erschreckend häufig festgestellt. Dies ist insbesondere deswegen bemerkenswert, weil es durch die angebotenen Standardverfahren zur Absicherung das am leichtesten zu vermeidende Sicherheitsrisiko darstellt. Es wird sich zeigen müssen, ob der von Apple angestrebte Wechsel zum HTTPS-Zwang und geänderte Schnittstellen sowie die kostenlose Verfügbarkeit von einfach zu erlangenden SSL-Zertifikaten durch die *Let's Encrypt* Initiative[1] zu einer Verbesserung der Situation beitragen können.

Gerade für Unternehmen zeigt sich zudem auch, dass das Risiko für Datenlecks mit 34,4% hoch ausfällt. So wiesen 688 Apps im September 2015 Eigenschaften auf, durch die Daten während der Verarbeitung oder im gespeicherten Zustand unzureichend geschützt wurden. Beispielsweise wurde die standardmäßige Verschlüsselung der Dateien durch das Betriebssystem bei Apps deaktiviert die Office Daten verarbeiten oder es wurde kein sicherer Schlüsselspeicher für Nutzergeheimnisse verwendet. Dadurch ergibt sich ein Risiko im Lost-Device-Szenario. Ebenso ergaben sich aber auch Risiken für Datenlecks durch einen schlechten Schutz der Kommunikation, wenn in der Analyse der Kommunikation kritischer Daten festgestellt werden konnte.

Die dargestellten kryptographischen Risiken fallen vergleichsweise sehr gering aus. Nur bei 36 iOS Apps aus der getesteten Menge (1,8%) konnten automatisiert kryptographische Risiken festgestellt werden. Daraus lässt sich jedoch nicht der Umkehrschluss ableiten, dass die große Mehrzahl an Apps fehlerfreie kryptografische Maßnahmen umsetzen. Vielmehr ist aus manuellen Code-Reviews

[1] siehe: `https://letsencrypt.org/`

bekannt, dass hier immer wieder durch kleine Unachtsamkeiten oder fehlendem Sicherheits-Know-how gravierende Implementierungsfehler gemacht werden, die jedoch bisher nicht automatisiert detektierbar sind, da Fehler in diesem Bereich sehr unterschiedlich implementiert werden können.

Android

Der Appicaptor Chart für Android Apps zeigt (Abbildung 3.4), das viele Apps mehrere Risiken gleichzeitig aufweisen. So sind 1.446 Apps (72,3%) ohne physischen Zugriff angreifbar. Ein Großteil ist dabei auf die fehlerhafte Absicherung der Kommunikation zurückzuführen.

Durch den hohen Anteil an Kommunikationsrisiken ergibt sich für Unternehmen auch das Risiko für Datenlecks von 70,8%, zusätzlich zu den Risiken durch einen fehlenden Schutz für persistente Daten bei einem Lost-Device-Szenario. So sind beispielsweise Apps, die Office Daten verarbeiten erfolgversprechende Ziele für Angreifer, wenn die Kommunikation nicht geschützt wird oder die Serverauthentizität nicht korrekt validiert wird.

Auch hier bietet der Appicaptor-Chart Einen Einblick in die Co-Existenz der Risiken. Werden beispielsweise die Privatsphärenrisiken völlig außer Acht gelassen, reduziert sich das Gesamt-Risiko lediglich um 2,3% auf 75,8%. Wird zudem auch das Risiko durch verdächtige Funktionalität außer Acht gelassen, so reduziert sich das Gesamt-Risiko erneut nur um 1,4% auf 74,4%.

3.2 Implementierungsfehler

Durch die kontinuierliche Analyse mit Appicaptor ist es möglich, auch über das zeitliche Entwicklerverhalten in Bezug auf Implementierungsfehler Aussagen zu treffen. Ein Beispiel ist die Nutzung der weit verbreiteten externen AFNetworking Bibliothek unter iOS. Für diese wurde am 27. März 2015 eine Schwachstelle für die Version 2.5.1 veröffentlicht. Zudem wurde am 22. April 2015 veröffentlicht, dass die Standardkonfiguration der Versionen 2.1 bis 2.5.2 auch verwundbar sind. Abbildung 3.5 zeigt nun den Verlauf der Versionen ab Mai 2015 in den Top 2.000 der kostenlosen iOS Apps, für die eine Verwundbarkeit automatisiert verifiziert werden konnte. Viele der verwundbaren Versionen in ursprünglich ca. 20% der Apps wurden demnach nach und nach ausgetauscht. Die verwundbaren Versionen stellen aber auch ein halbes Jahr später mit 11,5% noch ein großes Risiko für Angriffe über das Netzwerk dar. Zudem zeigt sich, dass sich die Änderungsrate weiter abflacht, da nicht mehr über die Schwäche berichtet wird und offensichtlich wenig für eine Aktualisierung aus funktionalen Gründen spricht. Dies ist auch an den Versionen 2.0 und kleiner zu erkennen, die nur sehr langsam aus den Apps verschwinden.

Die Verwundbarkeit der AFNetworking Bibliothek hat allerdings nicht zu einem Rückgang in der Verwendung geführt. Abbildung 3.6 zeigt im Gegenteil, dass

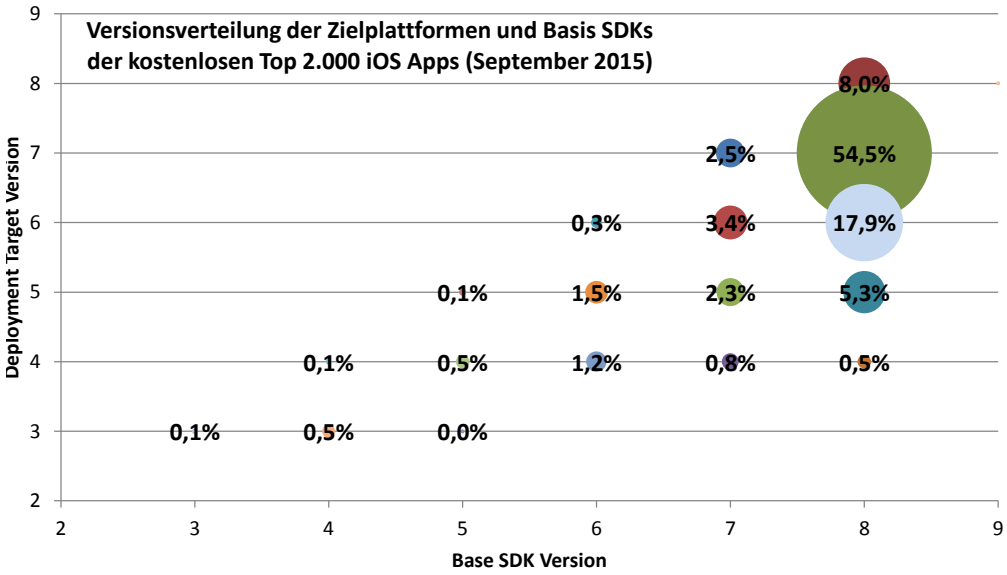

Abbildung 3.2:
Die große Mehrzahl der Apps in den Top 2.000 der kostenlosen iOS Apps wird gegenwärtig mit aktuellen SDK-Versionen erstellt. Dadurch werden die neuen Standard-Sicherheitsmaßnahmen besser genutzt als im letzten Jahr, sofern sie nicht vom Entwickler wieder deaktiviert wurden. (Appicaptor, September 2015)

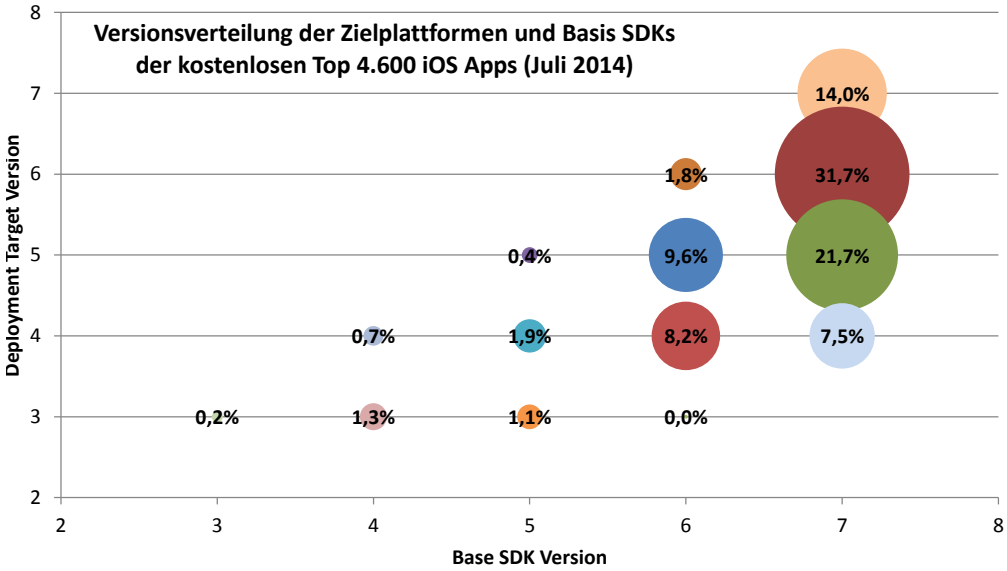

Abbildung 3.3:
Im Juli 2014 zeigt die Verteilung der Versionen von Base SDK und Deployment Targets noch eine deutlich größere Streuung. Die Entwickler der kostenlosen Top 2.000 iOS Apps haben demnach ihre Entwicklungsumgebungen aktualisiert und setzen nun für die Apps neuere Betriebssystemversionen voraus, was der Sicherheit zugutekommen kann. (Appicaptor, September 2015)

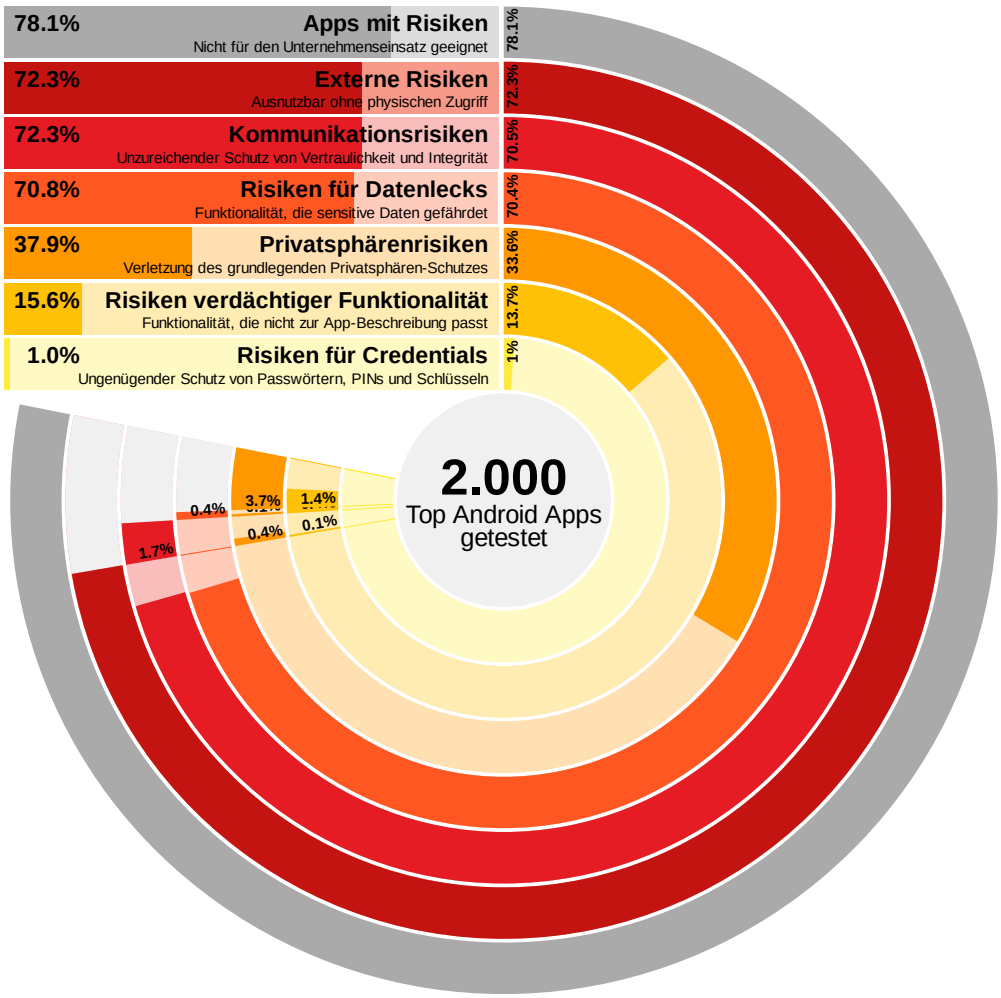

Abbildung 3.4: Risiken der kostenlosen Top 2.000 Android Apps (Appicaptor, September 2015)

die Beliebtheit der Bibliothek seit der Veröffentlichung der Verwundbarkeit insgesamt eher zugenommen hat, und das obwohl Apple bei den eigenen Netzwerkbibliotheken inzwischen eine einfach zu nutzende Funktionalität zur Verfügung stellt. AFNetworking vereinfacht zwar gegenüber den Systembibliotheken die Umsetzung von Certificate Pinnung, allerdings zeigt die Appicaptor Analyse, dass mit 0,64% die wenigsten Entwickler Certificate Pinning mit AFNetworking auch tatsächlich nutzen.

Dies zeigt wie stark Entwickler Abhängigkeiten zu externen Bibliotheken eingehen, die dazu führen, dass häufig selbst bekannte Verwundbarkeiten über eine lange Zeit bestehen bleiben. Durch dieses große Fenster in dem bekannte Verwundbarkeiten ausgenutzt werden können, steigt auch der Nutzen für Angreifer, weil der investierte Aufwand zur Ausnutzung länger gewinnbringend genutzt werden kann und durch die große Anzahl verwundbarer Apps die Erfolgsaussichten steigen. Im Gegensatz zu Systembibliotheken, die nach dem (Betriebssystem-)Patch automatisch allen Apps wieder sicher zur Verfügung stehen, stellen damit externe Bibliotheken bei dem gegenwärtigen Entwicklerver-

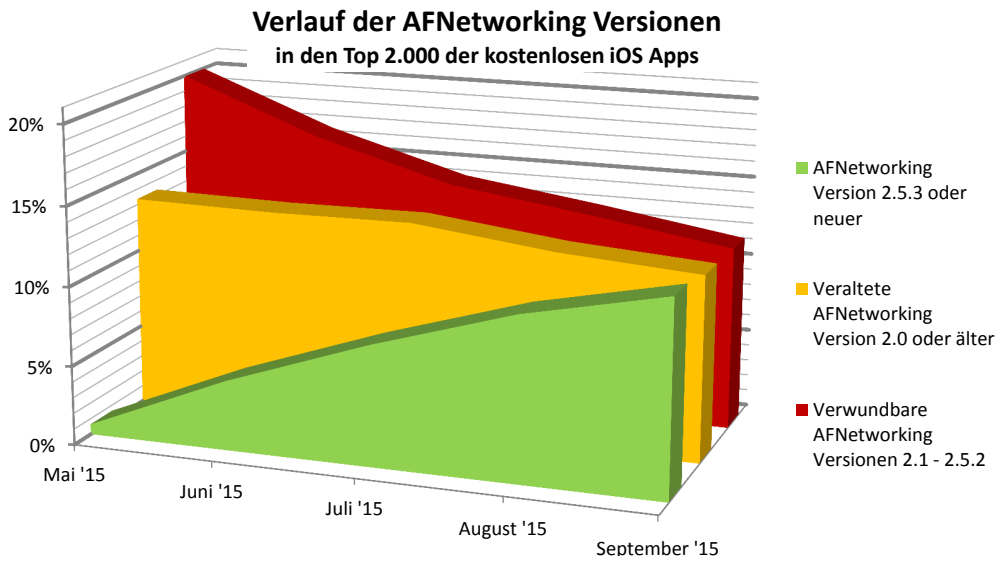

Abbildung 3.5:
Auch 6 Monaten nach Veröffentlichung der AFNetworking Verwundbarkeit sind noch 11,5% der kostenlosen Top 2.000 iOS Apps über ihre Kommunikation angreifbar. Einmal integrierte Versionen werden kaum aktualisiert. Die veralteten Version 2.0 und älter haben nur um 1,7% abgenommen. (Appicaptor, September 2015)

halten ein großes Risiko dar.

Aber auch bei anderen Implementierungsschwächen steigt gegenwärtig das Risiko. So ist beispielsweise die fehlerhafte Server-Zertifikatsprüfung von 43,7% im Februar 2015 auf 46,9% im September 2015 bei den Top 2.000 der kostenlosen Android Apps gestiegen (siehe Verlauf in Abbildung 3.7). Trotz der zahlreichen Berichterstattungen zu den Verwundbarkeiten scheint das Problem somit gegenwärtig tendenziell noch größer zu werden.

Bemerkenswert ist auch die starke Zunahme der Implementierungsschwäche bei der Kommunikation zum selben Server sowohl gesichert (per HTTPS) als auch ungesichert (per HTTP) durchgeführt wird. Dies birgt in zweierlei Hinsicht Risiken für den Nutzer. Zum Einen werden Cookies, die über eine sichere Verbindung aufgebaut wurden nun über die unsichere Verbindung automatisch mitgeschickt, auch wenn dies von dem Entwickler so gar nicht vorgesehen ist. Zum Anderen ermöglicht die Manipulation der ungesicherten Antwort auch den Zugriff auf alle Inhalte der von dieser Domain geladenen Inhalte und der verarbeiteten Daten. Der erste Angriff könnte serverseitig durch das Setzen der Secure- und HttpOnly-Flags verhindert werden. Hiermit werden Cookies einer Domäne vom Versand über HTTP ausgeschlossen (Secure-Flag) und der Zugriff von aktiven Inhalten (z.B. JavaScript) auf die Cookies dieser Domain verhindert (HttpOnly-Flag). Richtig eingesetzt entsteht so zwar ein Schutz des Cookies gegen direktes Mitlesen oder Auslesen, nicht aber gegen die Manipulation der App und damit bleibt die Möglichkeit für Angreifer bestehen, bereits autorisierte Verbindungen zum Server zu missbrauchen.

Die Frage wodurch dieser klare Trend in der Zunahme herrührt, ist nicht ab-

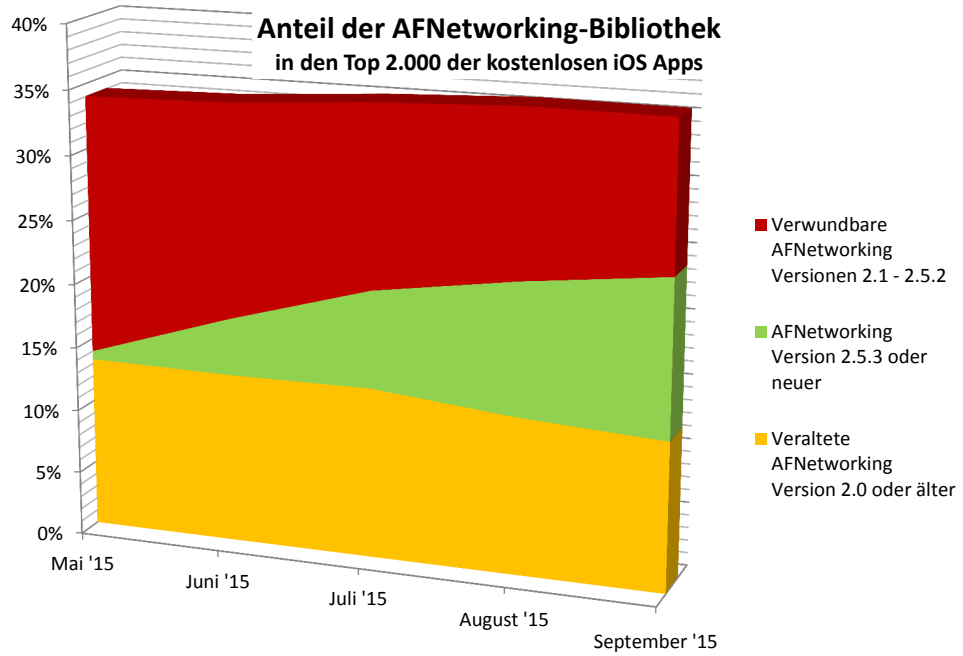

Abbildung 3.6:
Die Verbreitung der AFNetworking Bibliothek hat trotz Verwundbarkeit eher zugenommen. (Appicaptor, September 2015)

schließend geklärt. Naheliegend ist aber die Annahme, dass die URLs in den betroffenen Apps in unterschiedlichen Verantwortungsbereichen liegen, also beispielsweise in unterschiedlichen externe Bibliotheken verwendet werden, die untereinander und gegenüber dem Haupt-Code verschiedene Funktionen in Verbindung mit dem selben Server erfüllen und daher keine einheitliche Umstellung auf HTTPS erfolgt oder diese sogar überhaupt nicht möglich ist. In diesem Fall ist dann auch der Einsatz der beschrieben Flags problematisch, da die Flags immer auf eine ganze Domain wirken und viele Bibliotheken mit JavaScript auf die Cookies zugreifen müssen oder ohne den Versand der Cookies per HTTP nicht mehr funktionieren. Auch hier zeigt sich somit die Abhängigkeit der Entwickler bei externen Bibliotheken die der Sicherheit abträglich ist.

3.3 Tracking

Auch in diesem Jahr setzt sich der Trend zu mehr Werbe- und Trackingprovidern in Apps fort. Der Anteil der Apps ohne detektierte Interaktion mit diesen Providern ist bei den iOS-Apps von etwa 26% im Juli 2014 auf aktuell 18,6% gesunken. Bei Android liegt dieser Anteil aktuell bei 22,1%. Zudem ist der Anteil der iOS Apps mit mehr als 5 Werbe- und Trackingnetzwerken von 12,5% auf 22,4% gestiegen. In den getesteten Android Apps lag dieser Anteil bei 6,2%.

Durch die gestiegene Nutzung der Werbe- und Trackingprovider in Apps erhalten immer mehr Provider Einblick in das App-Nutzungsverhalten der Anwender

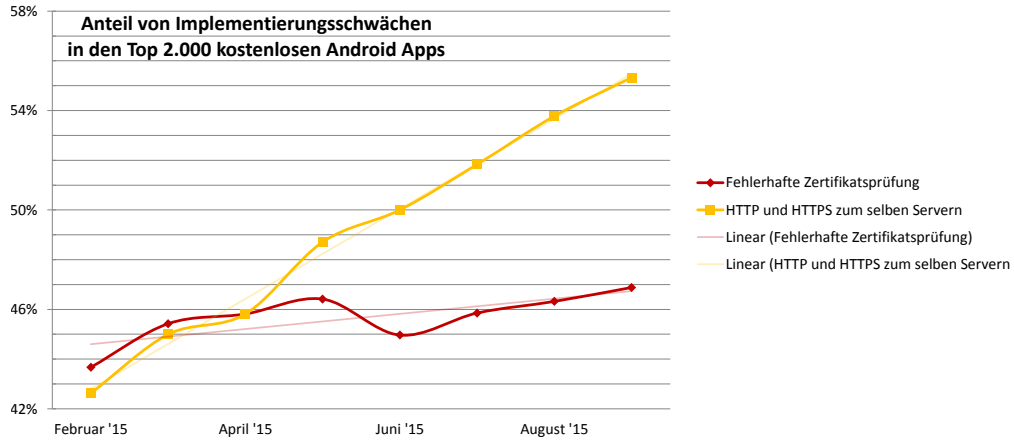

Abbildung 3.7:
Verlauf der Anteile von Implementierungsschwächen in den Top 2.000 kostenlosen Android Apps. Sowohl die fehlerhafte Zertifikatsprüfung als auch die gemischte Nutzung von HTTP und HTTPS zum selben Server nimmt zu. (Appicaptor, September 2015)

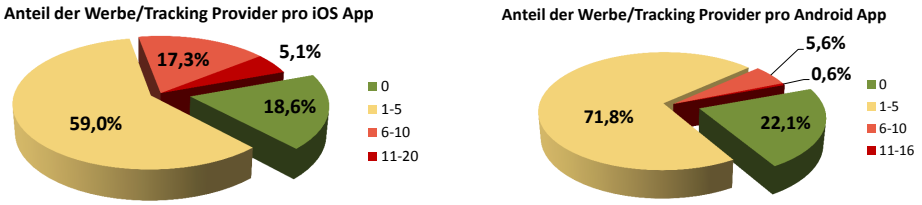

Abbildung 3.8:
Anteil der Werbe/Tracking-Provider pro iOS/Android App in den Top 2.000 kostenlosen Apps der jeweiligen Plattform. In der Mehrzahl der Apps wurden bis zu 5 Werbe/Tracking-Provider gefunden. Allerdings wurden auch Apps gefunden, die mit bis zu 20 solcher externen Dienstleistern in Verbindung stehen. (Appicaptor, September 2015)

und pro Provider können mehr Details der Nutzer gesammelt werden. Bei einer geschäftlichen Nutzung können somit auch immer mehr Rückschlüsse über aktuelle Abläufe im Unternehmen gezogen werden. Aus Sicht der Nutzer steigt damit die Wahrscheinlichkeit, dass ein Provider unter Einbeziehung von eindeutigen IDs, dem App-Kontext und der App-Aufrufhistorie konkrete Rückschlüsse über den Ablauf im Unternehmen erstellen kann. Da ein Großteil der Kommunikation mit den Providern immer noch unverschlüsselt stattfindet, stehen diese Informationen aber auch allen anderen zur Verfügung, die Zugriff auf das genutzte Netzwerk haben.