Fraunhofer SIT

Appicaptor Security Index

Ergebnisauszug für iOS-Apps

Fraunhofer-Institut für
Sichere Informationstechnologie (SIT)

24. September 2014

Investition in Ihre Zukunft

HESSEN

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

Gefördert im Rahmen von

EC SPRIDE

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

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 dem Dienst-Smartphone installierten Apps kommen auch in Kontakt mit Unternehmensdaten. Mitarbeiter können jedoch die daraus resultierenden Risiken meist nur unzureichend einschätzen, und auch den Administratoren fehlen häufig klare App-Auswahlkriterien. So wird aufgrund der enormen Menge an Apps, die in Unternehmen Einsatz finden (können), ein kosten- und ressourceneffizienter Prozess zur Sicherheitsbewertung benötigt, der auf objektiven Sicherheitskriterien beruht.

Um einen Einblick in das gegenwärtige Sicherheitsniveau bei Apps zu geben, werden im Folgenden Ergebnisauszüge von iOS-App-Massentests vorgestellt. Dazu wird zunächst eine kleine Auswahl relevanter Sicherheitskriterien für den Unternehmenseinsatz beschrieben. Die Auswertung der Kriterien erfolgte automatisiert mit dem am Fraunhofer SIT entwickelten Analysewerkzeug Appicaptor. Im Anschluss werden die Ergebnisse der Analyse beschrieben.

2 App-Tests

2.1 App-Auswahl

Für die Analyse wurden zwei App-Testmengen gebildet. Die erste Testmenge *TOP* besteht aus den 4.600 Apps, die laut Apples App Store am 14. Juni 2014 zu den jeweils 200 kostenlosen Top-Apps der 23 App-Kategorien zählten.

Die zweite Testmenge *RAND* besteht aus 10.000 kostenlosen iOS-Apps, die zufällig aus einer Liste von ca. 760.000 kostenlosen iOS-Apps ausgewählt wurden. Die Liste der kostenlosen Apps wurde aus den insgesamt ca. 1,1 Millionen iOS-Apps erstellt, die im Juli 2014 dem Appicaptor-Framework bekannt waren.

2.2 Blacklist-Kriterien

App Blacklisting ist eine Methode, um die Installation und Ausführung unerwünschter Apps auf Unternehmensgeräten zu verhindern. Solche Apps sind nicht nur diejenigen, von denen Sicherheitsbedrohungen oder Schwachstellen bereits bekannt sind, sondern auch solche, die innerhalb der Organisation im Hinblick auf die spezifische Unternehmens-IT-Sicherheitsrichtlinie als unangemessen betrachtet werden.

Generell sind viele Appicaptor-Testergebnisse direkte Erkenntnisse, die autark genutzt werden können, um die Sicherheitsqualität der App zu bewerten. Viele der Testergebnisse bringen jedoch ihren vollen Nutzen, wenn sie gemeinsam mit anderen Appicaptor-Testergebnissen interpretiert werden. Mit integrierten Appicaptor-Funktionen können effizient Regelsätze modelliert werden, mit deren Hilfe Apps unternehmensspezifisch bewertet werden und gekennzeichnet wird, 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.

In der Folge wird ein kleiner Auszug von Testkriterien und Appicaptor-Ergebnissen vorgestellt, die direkt zur App-Bewertung verwendet werden können und die aufgrund des signifikanten Vorkommens für die Ergebnisdarstellung ausgewählt wurden.

2.2.1 Unvollständige Absicherung

Blacklist-Regel: Fehlende Nutzung schützender Compiler Flags

Wird eine App ausgeführt, definieren unterschiedliche interne App-Merkmale die maximale Sicherheitsstufe, die die App erreichen kann. In diesem Zusammenhang kann der App-Entwickler Einstellungen vornehmen, um die Angriffsfläche der App zu reduzieren. Hierzu stehen dem iOS-Entwickler mehrere Kompilierungsoptionen (*Position Independent Executable (PIE)* als Teil von Address Space Layout Randomization (ASLR), *Stack Smashing Protection* und *Automatic Reference Counting (ARC)*) zur Verfügung, um das resultierende App Binary zu härten, z.B. gegen Attacken auf Basis von Pufferüberlauf- und Speicherkorrumpierungsproblemen.

Appicaptor wertet durch Reverse Engineering die Kompilierungsoptionen aus, die der App-Entwickler während der App-Entwicklung und -Bereitstellung angewendet hat. Abhängig von der eingesetzten Entwicklungsumgebung sind standardmäßig viele der Kompilierungsoptionen aktiviert. Das Deaktivieren bestimmter dieser Optionen gibt Hinweise über das (Sicherheits-)Know-how des Programmierers, denn das Deaktivieren vereinfacht oft die Programmierung, ermöglicht jedoch den Gebrauch von unsichereren Programmiertechniken. Zusätzlich werden aufgrund der fehlenden Kompilierungsoptionen die Aufwände für einen erfolgreichen Angriff auf die App deutlich verringert und heben damit das allgemeine Angriffsrisiko.

Apps in denen dieser Schutz nicht vollständig aktiviert ist, entsprechenden damit häufig nicht den Anforderungen an die Sicherheitsqualität für den Unternehmenseinsatz.

2.2.2 SSL-Implementierungsfehler

Blacklist-Regel: Enthält fehlerhafte Zertifikatsprüfung

Unsere Untersuchungen zeigen, dass nur ein kleinerer Anteil der Smartphone-Apps vollständig autark ohne externe Ressourcen auskommt. Die Mehrheit der Apps interagiert mit externen Entitäten über Netzwerk- oder lokaler Kommunikation. Wenn eine Smartphone-App mit Servern interagiert und diese Kommunikation sensible Daten beinhaltet, sollte in der App die Kommunikationssicherheit in der richtigen Art und Weise umgesetzt werden.

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 werden 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.

Eine dieser Änderungen, die das Sicherheitsniveau senkt, ist die teilweise oder vollständige Deaktivierung der Zertifikatsvalidierung. Wenn solche Änderungen in den Release-Versionen einer App erkannt werden können, ist dies ein guter Indikator für mangelnde Sicherheitskenntnisse des App-Entwicklers und/oder ein Hinweis auf einen gefährlich unsicheren Entwicklungsprozess. Denn eine solche Änderung eröffnet Möglichkeiten für Man-in-the-Middle Attacken, bei denen der Angreifer die ausgetauschten Daten mitlesen und manipulieren kann. Die Vertraulichkeit, Authentizität und Integrität der Daten der Kommunikation sind daher bei der festgestellten unsicheren Zertifikatsvalidierung grundlegend gefährdet.

2.2.3 Lücken im Verbindungsschutz

Blacklist-Regel: HTTP und HTTPS zum selben Server

Immer wieder wird sowohl geschützte als auch ungeschützte Kommunikation innerhalb einer App für die Kommunikation mit dem *selben* Server verwendet. 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 potenziellen 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. 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.

Darüber hinaus ist es denkbar, dass vergessen wurde, ein 'S' an eine 'HTTP'-URL hinzuzufügen, da die zugrunde liegenden Schnittstellen nicht zwischen einer ungeschützten und geschützten Kommunikation unterscheiden und damit kein Fehlverhalten sichtbar wird. Ungeschützte Kommunikation und damit auch die enthaltenen Autorisierungsgeheimnisse gegenüber Servern können jedoch z.B. bei öffentlichen WLANs ganz einfach mitgelesen werden. So könnte ein Angreifer die Kontrolle über die Verbindung erlangen und damit auch Zugriff auf die Dienstkontodaten des App-Benutzers mit allen gespeicherten Informationen erhalten.

Um diese Gefährdungen in Form eines potenziellen Datenlecks zu detektieren, sucht und dokumentiert Appicaptor für Apps jede HTTP-Nutzung zu Servern, mit denen die jeweilige App auch über HTTPS kommuniziert. Serverseitige

Schutzmaßnahmen der Sitzungscookies wie Secure- und HTTP-Only-Flags werden dabei nicht berücksichtigt. Richtig eingesetzt bieten sie zwar einen Schutz des Cookies gegen direktes Mitlesen oder Abrufen, können aber nicht die Manipulation der App verhindern und damit bleibt die Möglichkeit für Angreifer bestehen, bereits autorisierte Verbindungen zum Server zu missbrauchen.

2.2.4 Umfangreiches Tracking

Blacklist-Regel: Mehr als 5 Unternehmen erhalten Nutzungsdaten

Werbe-, Diagnose- und Tracking-Frameworks sammeln unterschiedliche Daten hinsichtlich der Smartphone-, App- und Internet-Nutzung. Dazu werden durch App-Entwickler Code-Elemente von Drittanbietern eingebettet, die ohne Wissen des Benutzers lokal gesammelte Daten direkt an Dritte weitergeben. So können umfangreiche Datensammlungen hinsichtlich der Nutzer aufgebaut werden.

Diese Frameworks können zu Privatsphärenverletzungen und Datenlecks durch Profilgenerierung führen. Neben der auch im Web anzutreffenden Informationsweitergabe durch Werbe- und Tracking-Netzwerke erfolgt dies bei Apps immer öfter auch über Diagnose-Dienste für Entwickler, die über das Nutzungsverhalten, den Speicherverbrauch und Fehlersituationen informieren und dazu unbemerkt vom Nutzer Informationen einsammeln. Die erhobenen Informationen enthalten eindeutige Identifikationsmerkmale, die auch mit Informationen aus der Nutzung von anderen Web-Diensten und den dort hinterlegten Identitäten verknüpft werden können. Der einzelne Entwickler erhält dabei lediglich die Informationen zu seiner App, der Anbieter dieser häufig kostenlosen Dienstleistung erhält jedoch die Nutzungsdaten aller Apps, die das entsprechende Diagnosemodul, Tracking- oder Werbenetzwerk enthalten. So erhält jeder Dienstleister ein detailliertes Bild darüber, wer wann wo welche App in welchem Kontext nutzt und so möglicherweise auch beispielsweise einen Einblick in Tagesablauf, Lebenssituation und Aufenthaltsorte von Mitarbeitern, wobei diese Informationen prinzipiell auch gegen Unternehmen einsetzbar sind.

Appicaptor kann die unterschiedlichen Module in den Apps identifizieren. Da für diese Analyse jedoch keine allgemeingültige Bewertung der Modul-Anbieter vorliegt wird das Risiko über die Anzahl der enthaltenen Module abgebildet. Für die Analyse wurden Apps erfasst, die Informationen an mehr als 5 Unternehmen schicken, welche mit der eigentlichen App-Funktion nichts zu tun haben. Der unternehmensspezifische App-Bewertungsprozess sollte beurteilen, ob die in die App integrierten Werbe-, Diagnose- und Tracking-Frameworks konform zu den Unternehmensregeln sind, bezogen auf den Schutzbedarf der Einsatzumgebung, und zu der bereitgestellten Funktionalität bzw. den zugänglichen Daten passen.

2.2.5 Unsichere Verschlüsselung

Blacklist-Regel: Verwendung von ECB-Modus für Daten

Appicaptor bewertet und listet die kryptographischen Funktionen auf, die innerhalb der App genutzt werden. Der Fokus dieses Tests ist es, die Nutzung von veralteten und schwachen Algorithmen, die nicht mehr für neue Anwendungen verwendet werden sollten, zu detektieren oder eine unsachgemäße Verwendung von kryptographischen Funktionen zu erkennen. Die hier ausgewählten Ergebnisse beziehen sich auf eine nicht empfohlene Verwendung des kryptographischen Modus ECB (Electronic Code Book Mode) bei der Nutzung von Blockverschlüsselungsalgorithmen.

Durch die fehlende Blockverkettung im ECB-Modus bleiben einheitliche und großflächige Bereiche, welche sich über mehrere Blöcke erstrecken, im Chiffrat immer noch erkennbar. Damit bietet der ECB-Modus eine einfache Angriffsfläche für statistische Analysen. Zu den grundsätzlichen Problemen gehört, dass der ECB-Modus die Häufigkeit von Blöcken im unverschlüsselten Text durch die fehlende Blockverkettung nicht ausreichend schützt.

2.2.6 Data Protection deaktiviert

Blacklist-Regel: Dateien erhalten nicht den Standardschutz

Aktuelle iOS-Geräte bieten Hardware-Verschlüsselung (iOS Data Protection) zum Schutz lokal gespeicherter Daten als zusätzliche Schutzschicht, insbesondere wenn ein Gerät verloren wird.

Apps, die eine Speicherung sensibler Daten im Dateisystem von iOS-Geräten vornehmen, sollten immer die iOS Data Protection verwenden. Wenn eine neue Datei auf einem iOS-Gerät erstellt wird, kann der Entwickler die neu anzulegende Datei einem spezifischem Schutzniveau zuordnen oder das Standard-Schutzniveau verwenden. Um festzustellen gegen welche Angriffe die Daten geschützt sind, wertet Appicaptor alle Dateigenerierungs- und -veränderungsprozesse innerhalb der App aus und überwacht die Zuordnung der Datenschutzklassen auf diese Dateien.

Häufig ist die Standardklasse ausreichend, wenn eine App auf iOS 7 oder höher installiert wurde (*NSFileProtectionCompleteUntilFirstUserAuthentication*). Wird durch eine spezifische Konfiguration innerhalb der App die verwendete Schutzklasse der Data Protection so gewählt, dass das Nutzergeheimnis nicht in die Verschlüsselung eingeht (*NSFileProtectionNone*) wird damit dieser wichtige Schutz deaktiviert. Gemäß vieler Unternehmens-IT-Sicherheitsrichtlinien, die eine Verschlüsselung von Unternehmensdaten mit einem Nutzergeheimnis fordern, ist das Deaktivieren dieses Mechanismus als Hinweis zu werten, dass die App die Unternehmensdaten nicht konform schützt und damit die App als nicht vertrauenswürdig eingestuft werden muss.

3 Ergebnisse

3.1 Appicaptor-Charts

Die hier präsentierten Appicaptor-Charts (siehe Abbildung 3.1, 3.2) 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 einer der beschriebenen Schwächen ein.

Um den Anteil von verschiedenartiger Schwachstellen innerhalb einer App ablesen zu können, schließt sich an das Balkendiagram ein Kreisbalken an. Der farblich hervorgehobene Teil stellt den Anteil der Schwachstelle an den ebenfalls gefundenen anderen Schwachstellen dar. Die Bereiche am Ende jedes Kreisbalkens stellen dabei den Anteil der Apps an der Gesamtzahl dar, die nicht die darüber dargestellten Schwächen aufweisen. Unterhalb des so resultierenden Kreissegmentes befinden sich die Angaben zu der Verteilung der Apps die gleichzeitig die Schwäche 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 4.600 Top iOS-Apps (siehe Abbildung 3.1), enthalten beispielsweise 350 Apps SSL-Implementierungsfehler (7,6%), nicht aber Lücken im Verbindungsschutz und keine unvollständige Absicherung. Von diesen 350 Apps enthalten aber 152 (3,3%) ebenfalls ein umfangreiches Tracking und 133 (2,9%) setzen unsichere Verschlüsselung ein, jeweils bezogen auf die Gesamtzahl von 4.600 getesteten Apps. Zur Kategorie der SSL-Implementierungsfehler zeigt das Diagramm weiterhin, dass 2,5% der Apps (115 Apps) gleichzeitig auch die Schwachstellen der unvollständigen Absicherung aufweisen.

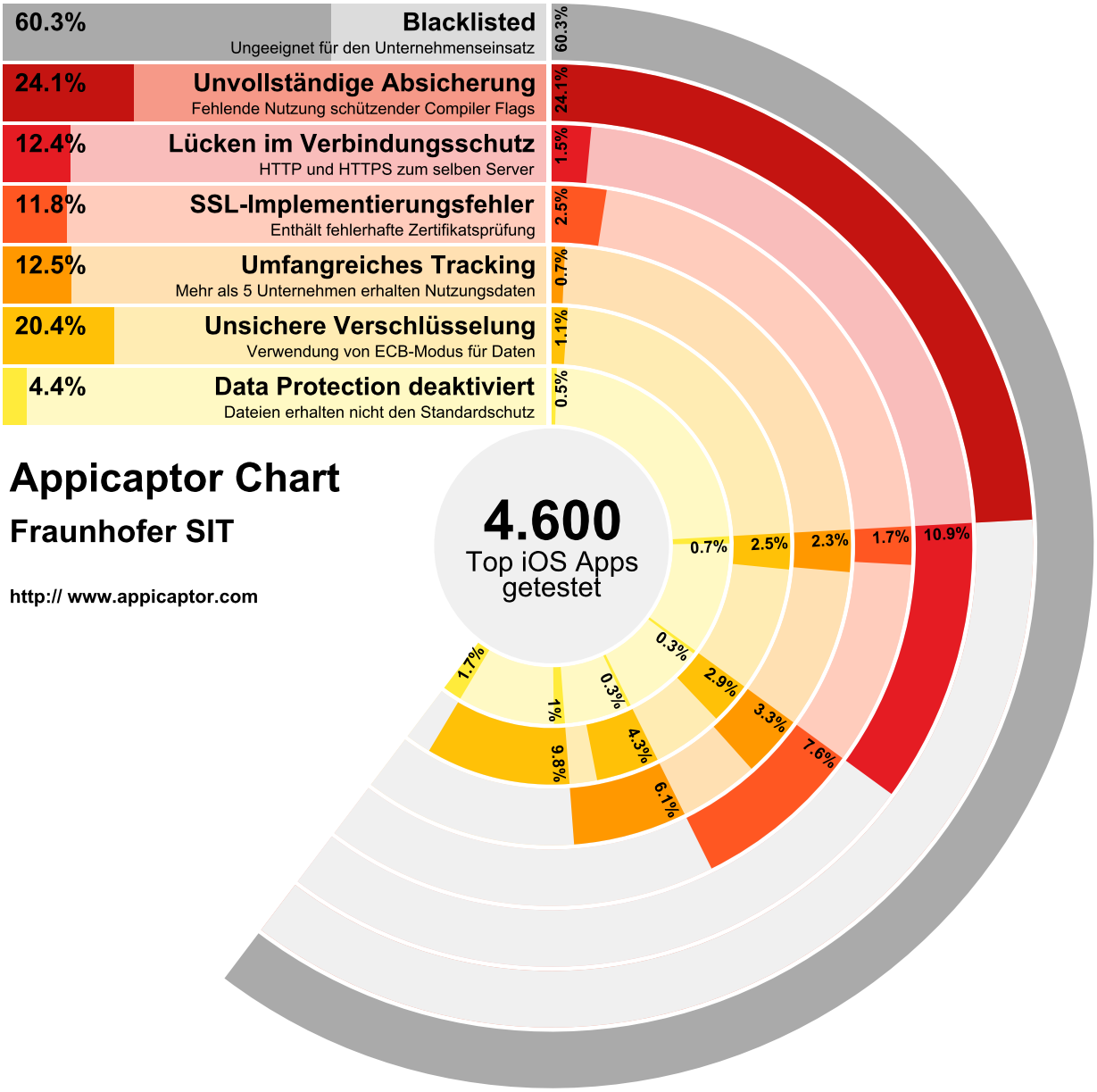

Abbildung 3.1: Blacklisting-Ergebnisse der 4.600 kostenlosen Top iOS-Apps (Appicaptor, Juli 2014)

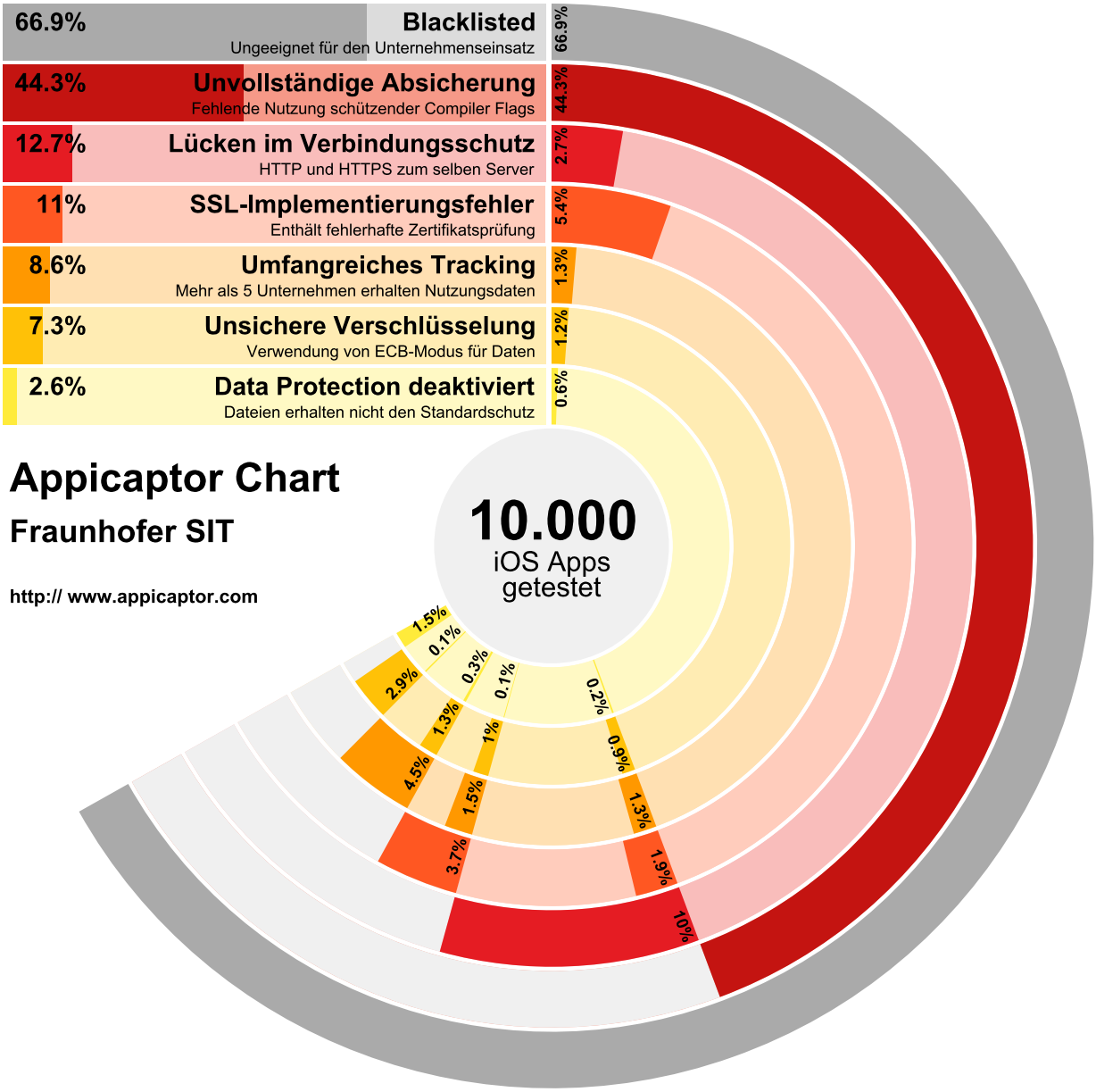

Abbildung 3.2: Blacklisting-rgebnisse für 10.000 zufällig ausgewählte kostenlose iOS-Apps (Appicaptor, Juli 2014)

3.2 Auswertung

Bei den Ergebnissen fällt zunächst erfreulich auf, dass die Gesamtschwachstellenrate der TOP-Testmenge (siehe Abschnitt 2.1) mit 60,3% etwas geringer ausfällt als die der RAND-Testmenge (siehe Abschnitt 2.1) mit 66,9% und damit die verbreitetsten Apps geringfügig weniger der getesteten Eigenschaften aufweisen, die häufig Unternehmensanforderungen verletzen.

Ebenso ist die fehlende vollständige Absicherung der Apps der TOP-Testmenge mit 24,1% deutlich geringer vertreten als in der RAND-Testmenge (44,3%).

Dies kann zum Teil mit einem höheren Aktualisierungsgrad der Top-Apps erklärt werden, da aktuellere Entwicklungsumgebungen (SDKs) den Schutz durch Compiler Flags standardmäßig aktiviert haben. Zur Erläuterung zeigt Abbildung 3.4, dass 18,3% der Apps der RAND-Testmenge mit einer Entwicklungsversion von iOS 5.0 oder kleiner und 54,2% mit der von iOS 7.0 oder höher erstellt wurden. Demgegenüber stehen bei den Apps der TOP-Testmenge 5,6% und 74,9% (siehe Abbildung 3.3). Die TOP-Testmenge ist somit mit signifikant aktuelleren SDKs erstellt worden.

Weitere hier nicht dargestellte Auswertungen haben ergeben, dass in der TOP-Testmenge 10,3% eine SDK Version größer gleich 7.0 nutzen und dennoch keine vollständige Absicherung nutzen. Dies bedeutet, dass die Entwickler in diesen Fällen absichtlich den (für Unternehmens-Apps notwendigen) Schutz deaktiviert haben.

Zudem sind die Mindestvoraussetzungen an die installierte iOS-Version bei der TOP-Testmenge signifikant höher. 47,5% setzen hier eine iOS-Version von 6.0 oder höher voraus, während es in der RAND-Testmenge nur 27,4% sind. Die höheren Anforderungen an die iOS-Version haben damit auch einen Einfluss auf die Gesamtsicherheit, da in den iOS-Versionen 6 und 7 die Schutzfunktionen weiter verbessert wurden, wovon die Sicherheit für Unternehmensdaten profitieren kann.

Durch die hohe Popularität der Apps der TOP-Testmenge ist zum anderen wohl aber auch der höherer Entwicklungsaufwand für die bessere Absicherung motivierbar. Diese höhere Popularität spiegelt sich aber auch in einem höheren Anteil an Trackingaktivitäten der TOP-Testmenge (12,5%) im Vergleich zur RAND-Testmenge (8,6%) wider.

Noch nicht abschließend geklärt ist die Frage, warum der Anteil der unsicheren ECB-Nutzung bei der TOP-Testmenge (20,4%) deutlich höher ist als die der RAND-Testmenge (7,3%). Mögliche Erklärungen sind eine populäre Drittanbieter-Bibliothek oder generell die häufigere Anforderung zur Verschlüsselung in der TOP-Testmenge, deren Umsetzung jedoch nicht vollständig korrekt implementiert wurde.

Kein signifikanter Unterschied ergibt sich hinsichtlich der getesteten SSL-Implementierungsfehler zwischen der TOP-Testmenge (11,8%) und der RAND-Testmenge (11%). Auch bei den populäreren Apps besteht somit für etwa jede neunte App die Gefahr für Man-in-the-Middle-Angriffe, die in der Folge zu Datenlecks und zu Missbrauch von entwendeten Unternehmenspasswörtern führen können.

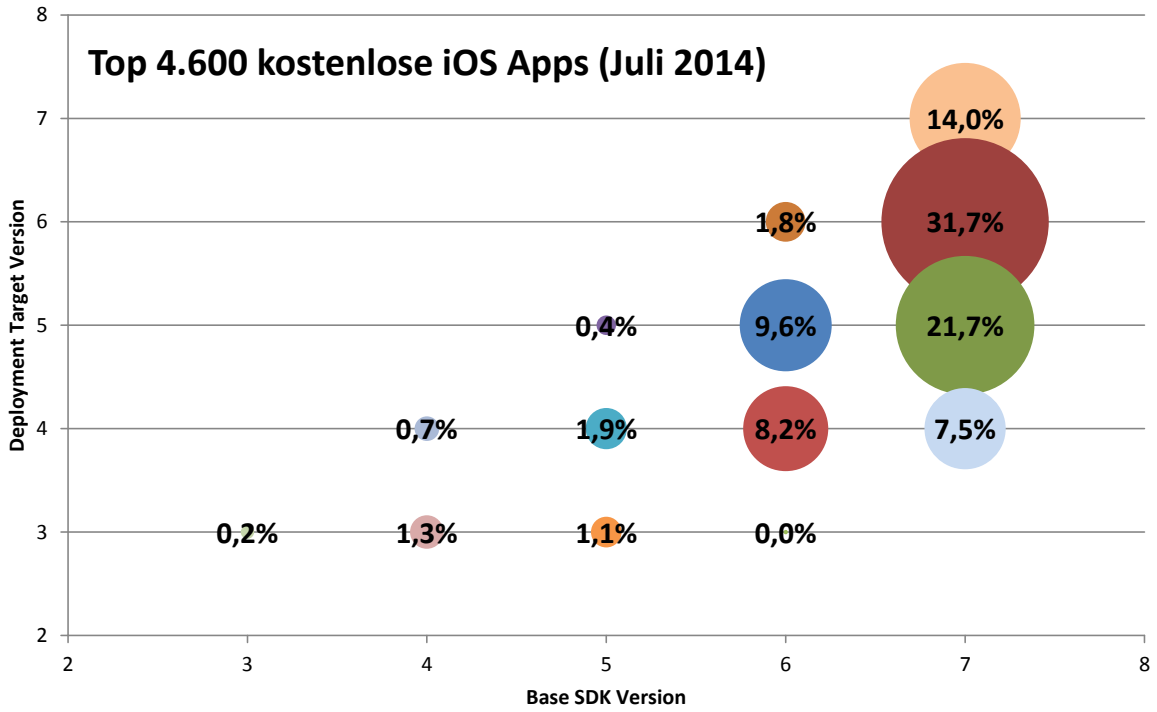

Abbildung 3.3:
Versionsverteilung der Zielplattformen (Deployment Target Version) und der Entwicklungsversion (Base SDK) der kostenlosen Top 4.600 iOS-Apps (Appicaptor, Juli 2014)

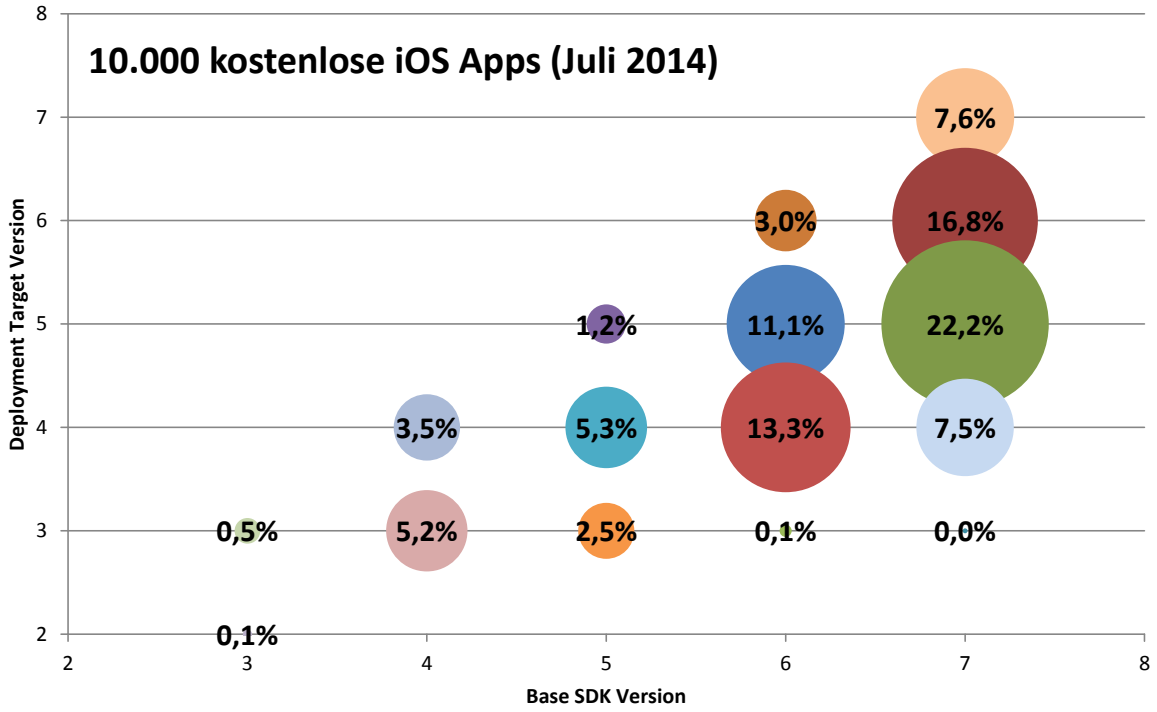

Abbildung 3.4:
Versionsverteilung der Zielplattformen (Deployment Target Version) und der Entwicklungsversion (Base SDK) von 10.000 zufällig ausgewählten iOS-Apps (Appicaptor, Juli 2014)