Sicherheit mobiler Plattformen

Ein technischer Überblick über Symbian OS und Windows Mobile

Jörg Weber

Rayko Enz

SIC! Software GmbH

www.sic-software.com

1 Einleitung

Mit steigender Verbreitung und immer größerer Funktionsvielfalt eines Smartphones erhöht sich auch die Anforderung an die Sicherheitsfunktionen eines modernen mobilen Betriebssystems.

Dieser Wunsch nach Sicherheit beeinflusst nun auch die Möglichkeiten eines Softwareentwicklers im Hinblick auf seine Aufgaben – er muss die Wünsche der Anwender nach einfacher Benutzung, schneller Programmausführung mit den Sicherheitsaspekten eines mobilen Betriebssystems in Einklang bringen. Der nachfolgende Artikel vergleicht die Sicherheitsmechanismen für die beiden verbreitetsten Betriebssysteme – Symbian OS und Microsoft Windows Mobile.

2 Symbian OS

Das Betriebssystem Symbian OS, hergestellt und vertrieben von dem Softwareunternehmen gleichen Namens, ist mit etwa 70% Marktanteil unter den Smartphones derzeit der absolute Marktführer dieses Segments. Dabei tonangebend in der Entwicklung und weiteren Gestaltung des Symbian-Betriebssystems ist als größter Lizenznehmer die Firma Nokia.

2.1 Dreistufiges Sicherheitsmodell ab Symbian OS v9.1

Durch die hohe Verbreitung und die offene Konzeption von Symbian OS ist das Betriebssystem auch als erste mobile Plattform Opfer virtueller Attacken geworden. Den Anfang machte der Wurm *Cabir* Anfang des Jahres 2004. Dieser war zwar keine eigentliche Gefahr, zeigte jedoch die Verletzbarkeit der damaligen Symbian OS-Versionen 7 und 8.

Auf Druck der Netzbetreiber und professioneller Anwender und nicht zuletzt durch die große Marktdurchdringung von Symbian OS, hat sich Symbian dann Ende 2005 entschieden das Sicherheitskonzept für die Version 9.1 radikal in ein mehrstufiges, mit Zertifikaten steuerbares Rechtesystem umzuwandeln.

Im Detail besteht das neue Sicherheitsmodell mit dem Namen *Symbian Platform Security* aus den drei folgenden drei Konzepten, die in den folgenden Abschnitten erklärt werden:

1. Capabilities
2. Signierung der Installationsdateien mit einem Zertifikat
3. Data-Caging

2.2 Capabilities bestimmen den Systemzugriff

In Symbian OS ab Version 9.1 werden bestimmte, als gefährdet eingestufte Funktionen des Systems über sogenannte *Capabilities* geschützt. Zu diesen Funktionen gehören bestimmte APIs, die beispielsweise bei nicht autorisierter Benutzung Kosten verursachen können oder die Datenintegrität gefährden.

Solche geschützten APIs besitzen immer eine damit verknüpfte Capability. Die API kann nur dann genutzt werden, wenn die Anwendung beziehungsweise der Code Rechte an der entsprechenden Capability besitzt. Das bedeutet, dass eine Anwendung, die Zugriff auf Capabilties benötigt, nur dann ausgeführt werden kann, wenn sie mit einem akzeptierten Zertifikat signiert ist, mit dem diese freigeschaltet werden.

2.2.1 User Capabilities

Die User Capabilities (je nach Quelle auch *Basic Capabilities* oder *Basic Set* genannt) sind die „einfachsten“ Capabilities. Der Name leitet sich von *User Grantable* – also vom Nutzer zustimmbar – ab, da der Nutzer der Anwendung bestimmen kann, ob er sie ausführen will oder nicht. Näheres dazu wird im Abschnitt *Signierung* behandelt.

Die folgenden Capabilities befinden sich in der User-Klasse:

Capability	Beschreibung
LocalServices	Erlaubt Zugriff auf lokale Netzwerk-Dienste, die keine Kosten verursachen können (beispielsweise Infrarot und Bluetooth)
Location	Erlaubt Zugriff auf die Position des Geräts, beispielsweise über die Netzwerkzellen ID
NetworkServices	Erlaubt Zugriff auf remote Netzwerk-Dienste, die Kosten verursachen können (beispielsweise Telefonfunktion, SMS, GPRS/UMTS)
UserEnvironment	Erlaubt Zugriff auf vertrauliche Informationen über den Nutzer und seine direkte Umgebung (diese Capability schützt beispielsweise die Kamera eines Gerätes)
ReadUserData	Erlaubt Lesezugriff auf vertrauliche Nutzerdaten, beispielsweise Adressbuch oder Textnachrichten
WriteUserData	Erlaubt Schreibzugriff auf vertrauliche Nutzerdaten.

2.2.2 Extended Capabilities

Die Klasse der Extended Capabilities beinhaltet sowohl die vorher genannten User Capabilities als auch einige als kritisch eingestufte Capabilities. Der Zugriff auf diese Capabilities wird nur über ein Symbian Signed Zertifikat erlaubt.

Folgende Capabilities beinhaltet die Extended-Klasse:

Capability	Beschreibung
PowerMgmt	Ermöglicht das Abschalten ungenutzter Systemperipherie, das versetzen/aufwecken aus dem Standby-Modus und das Abschalten des Geräts.
ProtServ	Erlaubt einem Server die Registrierung mit einem geschützten Namen. Server ohne ProtServ Capability können sich somit nicht registrieren.
ReadDeviceData	Erlaubt Lesezugriff auf vertrauliche Netzwerkbetreiber-, Gerätehersteller- und Gerätedaten (beispielsweise die PIN für die Gerätesperre)
SurroundingsDD	Erlaubt Zugriff auf Gerätetreiber die Informationen über die Geräteumgebung liefern (beispielsweise GPS)
SwEvent	Ermöglicht die Simulation von Tastatur-/Stifteingaben und diese Eingaben von jedem anderen Programm abzuhören.
TrustedUI	Ermöglicht das Erstellen von Dialogen in einer gesicherten UI-Umgebung (beispielsweise bei der Eingabe von Passwörtern)
WriteDeviceData	Erlaubt Schreibzugriff auf Einstellungen, die das Geräteverhalten bestimmen, beispielsweise Systemuhr, Zeitzone, Alarme.

2.2.3 Manufacturer Approved Capabilities

In der Klasse der Manufacturer Approved Capabilties befinden sich die kritischsten Capabilities, da sie nahezu das gesamte System des betroffenen Gerätes offen legen. Rechte an Capabilities aus dieser Klasse können nur vom Gerätehersteller gewährt werden (daher der Name *Manufacturer Approved*),

Rechte an den beiden Capabilities AllFiles und TCB (für Trusted Computing Base), werden nur nach besonders strengen Kriterien (in der Regel eine Analyse des Quellcodes der Anwendung durch den Gerätehersteller) vergeben, da sie Zugriff auf das komplette System erlauben.

2.3 Die Signierung bestimmt die Zugriffsstufe

Die Signierung mit einem Zertifikat ist das zentrale Element der *Platform Security*. Die Idee hinter diesem Konzept ist, dass eine Anwendung nur dann installiert und ausgeführt werden kann, wenn sie mit einem entsprechenden Zertifikat signiert ist.

Je nachdem, welche Capabilities die Anwendung benötigt, kann sie auf unterschiedliche Arten signiert werden, die in den folgenden Abschnitten erklärt werden.

2.3.1 Signierung mit eigenem Zertifikat

Ein Entwickler kann seine Anwendung mit einem selbst erstellten Zertifikat signieren. Die notwendigen Tools liefert das Symbian SDK mit oder die Entwicklungsumgebung *Carbide.C++* signiert die Anwendung beim Bauen automatisch.

Ein solches Zertifikat und der Signierungsprozess sind kostenlos und sehr schnell, da die Anwendung keinen Umweg über die Symbian-Signierungsstelle (*Symbian Signed*) machen muss. Allerdings erlaubt ein eigenes Zertifikat nur Zugriff auf die oben genannten, einfachen Capabilities der User-Klasse.

Der Nachteil der eigenen Signierung ist jedoch, dass bei der Installation der Anwendung eine Sicherheitswarnung erscheint.

2.3.2 Signierung über *Symbian Signed*

Symbian Signed ist die offizielle Signierungsstelle für Anwendungen, die für Symbian OS entwickelt wurden. Nur hier können Zertifikate erworben werden, die Rechte an sämtliche Capabilities gewähren.

Anwendungen, die ein Symbian Signed Zertifikat besitzen, werden ohne jeglichen Warnhinweis auf dem Gerät installiert und ausgeführt.

Um eine solche Signatur zu erwerben, muss die für die Entwicklung zuständige Firma sich gegenüber Symbian Signed eindeutig identifizieren.

Diese Identifikation wird über die *ACS Publisher ID* vorgenommen Eine solche ID wird nur von einer *Certification Authority* (CA) ausgegeben und ist zwingend notwendig, wenn eine kommerzielle Anwendung über Symbian Signed signiert werden soll.

Zusätzlich wird jede Anwendung von Symbian Signed daraufhin getestet, ob sie sich schädlich auf einem Gerät verhält. Diese Tests werden in unabhängigen, von Symbian autorisierten Testhäusern durchgeführt.

Der Preis eines solchen Tests ist dabei davon abhängig, welche Capabilities die Anwendung benötigt. [1].

2.3.2.1 Zugriff auf Capabilities

So wie sich der Preis des Testens danach richtet, welche Zugriffsrechte die Anwendung benötigt, so verändert sich auch der Symbian Signed Signierungsprozess je nachdem, ob und welche Capabilities die Anwendung benötigt.

Benötigt eine Anwendung keinen Zugriff auf Capabilities oder nur Zugriff auf User-Capabilities, wird die Anwendung anstandslos von Symbian Signed signiert – sofern die Anwendung den entsprechenden Test im Testhaus besteht.

Der Signierungsvorgang ist allerdings schon komplizierter, wenn die Anwendung Zugriff auf Capabilities aus der Extended-Klasse benötigt. Zusätzlich zu dem Funktionstest, muss beim Einreichen der Anwendung beim Testhaus eine Erklärung abgegeben werden, warum die Anwendung überhaupt Zugriff auf diese Capabilities benötigt.

Wesentlich komplexer ist die Signierung, wenn Zugriff auf Manufacturer Approved Capabilities benötigt wird. Hier muss der Gerätehersteller dem Zugriff auf diese Capabilities zustimmen. Bei der Übermittlung der Anwendung zum Testhaus muss der Entwickler auch eine detaillierte Erklärung abgeben, welche APIs und warum er sie ansprechen muss.

Symbian Signed ist aber in jedem Fall die Plattform, über die der Administrative Teil jedes Signierungsprozesses abgewickelt wird.

2.3.3 Signierung über *Symbian Signed Freeware*

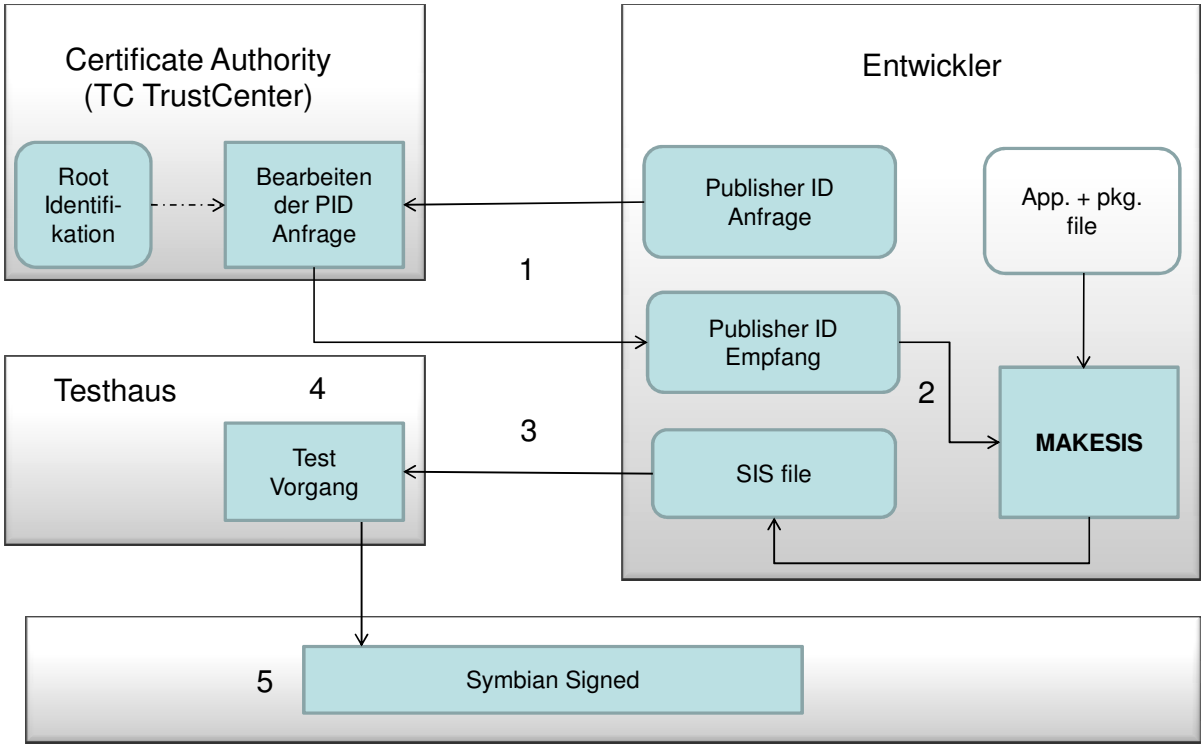
Freeware und Open Source Entwickler haben die Möglichkeit, ihre Anwendungen ohne Kosten von Symbian Signed signieren zu lassen. Da außerdem keine ACS Publisher ID benötigt wird, ist der Signierungsvorgang vollständig kostenlos [2].

2.3.4 Signierung mit Entwickler-Zertifikat

Ein Entwickler-Zertifikat (auch *Developer Certificate* genannt) wird genutzt, um eine Anwendung während der Entwicklungsphase auf einem Gerät zu testen. Zwar kann eine Anwendung während der Entwicklung auch mit einem eigenen Zertifikat signiert und getestet werden, allerdings funktioniert dies nicht, falls die Anwendung mehr als die einfachen User-Capabilities benötigt [3].

2.3.5 Beispiel: Ablauf der Symbian Signed Signierung

Das folgende Ablaufdiagramm zeigt beispielhaft wie der Symbian Signed Signierungsprozess einer Anwendung funktioniert, die keine Manufacturer Approved Capabilities benötigt [4].



1. Der Entwickler beantragt eine ACS Publisher ID bei der Certificate Authority TC TrustCenter. Diese identifiziert die für die Entwicklung verantwortliche Organisation und ist für den folgenden Signierungsprozess zwingend erforderlich.
2. Mithilfe des Symbian-Tools MAKESIS wird die SIS-Programmdatei mit der Publisher ID signiert.
3. Anschließend wird die signierte SIS-Datei an das Testhaus übergeben. Dort wird das Testen der Anwendung nach den Symbian Signed Kriterien durchgeführt.
4. Ist das Testen erfolgreich abgeschlossen, leitet das Testhaus die Anwendung an die Certificate Authority weiter. Die CA signiert abschließend mit dem endgültigen Zertifikat, das von Symbian-Smartphones als offizielles Zertifikat erkannt wird.
5. Anschließend wird der Entwickler informiert, dass die Signierung der Anwendung abgeschlossen ist und sie zum Download auf dem Symbian Signed Portal bereit steht.

Nach Abschluss dieses Vorganges kann die signierte Installationsdatei nicht mehr verändert werden, ohne ihre Gültigkeit zu verlieren.

2.4 Data Caging

Das neue Sicherheitsmodell von Symbian OS beinhaltet auch ein rechtegesteuertes Dateisystem, dass System- und private Daten effektiver schützen soll. Die Anwendung wird sozusagen in ihrem eigenen Bereich des Dateisystems eingesperrt.

Data Caging ist automatisch für alle Anwendungen gesetzt und greift auch für Anwendungen, die TCB beziehungsweise AllFiles Capabilities besitzen, dann jedoch in anderem Umfang. Im Detail besteht das Dateisystem aus drei geschützten Ordnern. Alle anderen Ordner im Telefonspeicher oder auf einer Speicherkarte hingegen sind für jede Anwendung frei zugänglich [5].

3 Windows Mobile

Seit der Einführung von Windows Mobile im Jahr 2001 hat es das mobile Betriebssystem aus dem Hause Microsoft geschafft, sich nach und nach auf dem Markt mobiler Endgeräte zu etablieren. Dies liegt ganz wesentlich daran, dass Windows Mobile immer dann die erste Wahl ist, wenn die Integration mobiler Geräte in eine existierende Microsoft-Systemumgebung im Vordergrund steht. Daher ist Windows Mobile besonders bei Firmenkunden zunehmend beliebt.

3.1 Versionsvielfalt bei Windows Mobile

Windows Mobile existiert seit der Version 2003 in drei unterschiedlichen Ausführungen, die sich auch in ihren Sicherheitsmodellen unterscheiden:

Die folgende Tabelle zeigt die verschiedenen Ausfürhungen und ihre alte und neue Namensgebung von Windows Mobile 5 und 6:

Windows Mobile 5	**Windows Mobile 6**
Windows Mobile für Pocket PC	Windows Mobile 6 Classic
Windows Mobile für Pocket PC Phone Edition	Windows Mobile 6 Professional
Windows Mobile für Smartphone	Windows Mobile 6 Standard

Am stärksten verbreitet sind Pocket PC Phones, also Geräte, die wie ein PDA aussehen und meistens per Stifteingabe bedient werden, aber zusätzlich noch eine Telefonfunktion haben.

3.2 Sicherheitsmodell ab Windows Mobile 2005

Microsoft hat bei der Herstellung von Windows Mobile darauf geachtet, die Entwicklung von Anwendungen so ähnlich wie möglich zu Windows für den Desktop-PC zu gestalten. Einem Entwickler, der bereits Windows-Programme in C++ und mit dem .NET-Framework geschrieben hat, sollte die Umstellung auf die Windows Mobile leicht fallen.

Entwickler müssen allerdings auch das neue Sicherheitsmodell von Windows Mobile verstehen, um Anwendungen zielgerichtet programmieren zu können. Es lässt sich für Entwickler auf drei wesentliche Komponenten zusammenfassen, die bei der Erstellung einer Anwendung wichtig und zu beachten sind:

- *Security Roles* (Sicherheitsrollen)
- *Security Policies* (Sicherheitsrichtlinien)
- Zertifikate und Signierung

3.3 Gerätesicherheit über Security Policies und Security Roles

Windows Mobile bietet den Ansatz, mithilfe von *Security Policies* und *Security Roles* die Gerätesicherheit direkt einstellen zu können. Dadurch kann ein Gerätehersteller oder Netzbetreiber beispielsweise festgelegen, dass nur Anwendungen mit einem gültigen Zertifikat ausgeführt dürfen und dass die Anwendung keine Geräteeinstellungen verändern darf.

3.3.1 Configuration Service Providers (CSPs)

Bei dem Thema Entwicklung für Windows Mobile fällt zwangsläufig der Begriff *Configuration Service Provider* (CSP). CSPs sind nichts anderes als Gruppierungen verschiedener Geräteeinstellungen, Anwendungen und APIs. Insgesamt existieren 43 CSP, welche nahezu alle Betriebssystem- und Gerätefunktionen steuern können [6].

3.3.2 Security Policies

Security Policies regeln die *grundsätzlichen* Sicherheitseinstellungen des Geräts. Eine Security Policy kann erlaubt (*allow*) oder verboten (*deny*) sein. Ein Beispiel wäre die Security Policy 4102, „erlaube oder verbiete die Ausführung unsignierter Anwendungen“.

Insgesamt existieren über zwei Dutzend Security Policies, über welche die Sicherheit sehr detailiert geregelt werden kann [7]. Für Entwickler lässt sich diese große Liste jedoch auf vier maßgebliche Security Policies zusammenfassen:

Tabelle 6

Security Policy	Beschreibung	Standard
Unsigned Applications Policy (4102)	Ausführen von unsignierten Anwendungen	Erlaubt
Unsigned CABS Policy (4101)	Installieren von unsignierten CAB-Dateien	Erlaubt
Unsigned Prompt Policy (4122)	Nachfrage beim Nutzer bei der Installation und Ausführung von unsigniertem Code (alle CAB-, DLL- und EXE-Dateien)	Erlaubt (Nachfrage-Dialog erscheint)
Privileged Applications Policy (4123)	Aktiviert das 1-Tier oder 2-Tier Zugriffsmodell	Pocket PC: 1-Tier Smartphone: 2-Tier

Die Zahl in Klammern hinter dem Namen der Security Policy gibt die *Policy ID* an. Die Spalte *Standard* gibt die üblichen Werkseinstellungen bei Consumer-Geräten an.

3.3.3 Security Roles

Als Security Role wird ein Nutzer mit bestimmten Rechten bezeichnet. Das bedeutet, dass nur ein Nutzer mit den entsprechenden Rechten die Einstellungen der jeweiligen Gerätefunktion ändern kann.

Windows Mobile kennt genau ein Dutzend unterschiedliche Security Roles [8]. Die folgende Tabelle beschränkt sich auf die drei häufigsten:

Tabelle 7

Security Role	Beschreibung
Manager SECROLE_MANAGER	Unbeschränkter Zugriff auf Systemressourcen. Kann die meisten Security Policies umstellen. Ist in der Regel *nicht* der Gerätenutzer!
Authenticated User SECROLE_USER_AUTH	Nur ein authentifizierter Nutzer hat Zugriff auf die entsprechende Systemressource. Der Gerätenutzer hat normalerweise diese Security Role.
Operator SECROLE_OPERATOR	Nur der Netzbetreiber besitzt das Recht an der entsprechenden Systemressource

Die Einstellungen der Security Policies aus der Tabelle 6 können allesamt nur durch die Manager-Role (SECROLE_MANAGER) verändert werden. Das heißt, nur ein Manager kann die Standardeinstellungen der oben genannten Policies umstellen. Ein Entwickler muss also

in Betracht ziehen, dass ein potentieller Nutzer seiner Anwendungen nicht die Sicherheitseinstellungen des Geräts verändern kann.

3.4 Zwei Zugriffsmodelle: One-Tier und Two-Tier

Die meisten Geräte werden jedoch mit denen in Tabelle 6 genannten Standardeinstellungen ausgeliefert. Kurz und knapp zusammengefasst bedeutet dies, dass auch unsignierte Anwendungen auf (fast) allen Geräten ausgeführt werden können – allerdings erscheint dann eine Sicherheitsmeldung, die vor der Installation beziehungsweise Ausführung der Anwendung warnt. Bestätigt der Nutzer diese Meldung, wird die Anwendung installiert beziehungsweise ausgeführt [9].

3.4.1 Zugriffsrechte von Anwendungen (Permissions)

Bevor eine Erklärung der beiden Zugriffsmodelle erfolgt ist es wichtig zu erläutern, wie Anwendungen unter Windows Mobile theoretisch ausgeführt werden können. Dieses Konzept ist grob mit dem Capabilities-Modell von Symbian vergleichbar: durch die Signierung der Anwendung mit einem Zertifikat wird das Zugriffsrecht bestimmt.

Die Möglichkeiten der Ausführung sind:

- Privilegiert
- Normal

Anwendungen, die privilegiert ausgeführt werden, besitzen die höchste Zugriffsstufe. Sie haben vollen Zugriff auf das Dateisystem und die Registry und können jede API ansprechen. Desweiteren besitzen Anwendungen, die privilegiert ausgeführt werden, die SECROLE_MANAGER, also die Manager-Role.

Dagegen besitzen Anwendungen, die normal ausgeführt werden, keinen Zugriff auf kritische, sogenannte *Trusted APIs* und haben ebenfalls keinen Zugriff auf Systemdaten und geschützte Bereiche der Registry.

Wie eine Anwendung jedoch tatsächlich ausgeführt wird, hängt primär vom Zugriffsmodell des Geräts und der Signierung der Anwendung ab.

3.4.2 One-Tier Zugriffsmodell

Die oben genannten Ausführungsmöglichkeiten treffen allerding nicht auf One-Tier Geräte zu. Diese verfügen über keine echte Rechteverwaltung: eine Anwendung kann nur ganz oder gar nicht ausgeführt werden.

3.4.3 Two-Tier Zugriffsmodell

Geräte, die mit dem Two-Tier Zugriffsmodell konfiguriert sind, haben dagegen die Möglichkeit, Anwendungen mit privilegierten oder normalen Rechten auszuführen. Anwendungen, die mit einem bekannten Zertifikat signiert sind werden wie bei One-Tier Geräten ohne Nachfrage beim Nutzer installiert und ausgeführt. Dabei bestimmt das Zertifikat die Zugriffsstufe. Nur Zertifikate von *Privileged Execution Trust Authorities* ermöglichen es, Anwendungen mit privilegierten Rechten auszuführen und zu installieren. Diese Zertifikate sind kostenpflichtig und werden nur nach Erfüllung bestimmten Kriterien bereitgestellt (siehe Abschnitt Signierung).

3.4.4 Zusammenfassung der Sicherheitsstruktur

Fasst man die Abschnitte 3.3 und 3.4 zusammen, ergibt sich die folgende Sicherheitsstruktur, wie Anwendungen ausgeführt werden können. Dies gilt allerdings nur für die Standardeinstellungen der Security Policies, wie sie in der Tabelle 6 aufgelistet sind.

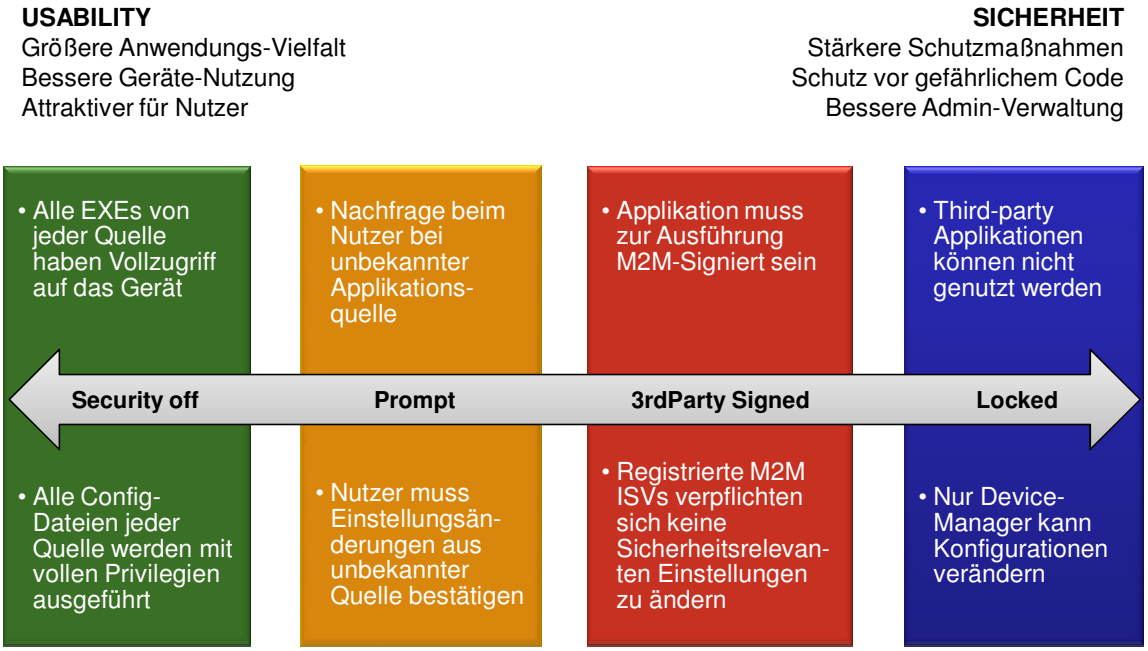
Tabelle 8

Signierung der Anwendung	One-Tier Geräte (Pocket PCs)	Two-Tier Geräte (Smartphones)
Nicht signiert oder unbekanntes Zertifikat	Nachfrage beim Nutzer. Anwendung wird (1) nicht ausgeführt oder (2) mit privilegierten Rechten ausgeführt	Nachfrage beim Nutzer. Anwendung wird (1) nicht ausgeführt oder (2) mit normalen Rechten ausgeführt
Signiert für normale Rechte	Anwendung wird mit privilegierten Rechten ausgeführt.	Anwendung wird mit normalen Rechten ausgeführt.
Signiert für privilegierte Rechte	Anwendung wird mit privilegierten Rechten ausgeführt.	Anwendung wird mit privilegierten Rechten ausgeführt.

Zusammenfassend lässt sich sagen, dass Windows Mobile Geräte in der Werkseinstellung so konfiguriert sind, dass alle Anwendungen installiert und ausgeführt werden können. Ist die Anwendung nicht oder mit einem unbekannten Zertifikat signiert, erscheint beim Nutzer nur eine Nachfrage (*Prompt*), die bestätigt werden muss.

Das bedeutet allerdings nicht, dass alle Geräte die Standardeinstellung nutzen, da jeder Netzbetreiber und Netzwerkadministrator die Sicherheitseinstellungen ändern kann.

Die vier gebräuchlichsten Sicherheitseinstellungen verdeutlicht die folgende Abbildung [10]:



Die oben beschriebene Standardeinstellung der meisten Windows Mobile Geräte ist *Prompt*, also die Nachfrage beim Nutzer, ob eine unsignierte oder aus unbekannter Quelle signierte Anwendung installiert und ausgeführt werden darf.

Berücksichtigt man diese Abbildung, ist dies die zweitniedrigste Sicherheitsstufe – darunter befindet sich nur *Security off*, also das Abschalten fast aller Sicherheitsfunktionen. Auf Geräten mit dieser Einstellung werden alle Anwendungen, also auch unsignierte, ohne jegliche Nachfrage installiert und ausgeführt.

Eine Stufe über Prompt steht *3rd Party Signed*, in manchen Quellen auch *Mobile2Market locked* genannt. Auf Geräten dieser Sicherheitsstufe können nur Anwendungen genutzt werden, die mit einem bekannten Zertifikat signiert sind. Der zuletzt genannte Name bezieht sich auf das Mobile2Market Program von Microsoft, über das (ähnlich wie bei Symbian Signed) Zertifikate erworben werden können. Der Signierungsprozess über Mobile2Market wird ausführlich im nächsten Abschnitt erklärt.

Die höchste Sicherheitsstufe ist *Locked* – gesperrt. Hier wurden alle Mobile2Market Zertifikate entfernt, um das Ausführen ungewollter Drittanwendungen zu verhindern – unsignierte Anwendungen laufen auf solch einem Gerät natürlich auch nicht. Viele Firmen

nutzen die Locked-Einstellung um ihrer Belegschaft die Installation von Fremdsoftware zu verbieten.

3.5 Signierung über Mobile2Market Program

Die Signierung einer Anwendung über eine offizielle Certification Authority garantiert die maximale Kompatibilität der Anwendung. Ausgangsbasis der Signierung von Windows Mobile Anwendungen ist Mobile2Market – Microsofts Gegenstück zu Symbian Signed. Nur mit einem Mobile2Market Zertifikat kann sichergestellt werden, dass die Anwendung auf der größtmöglichen Anzahl an Geräten läuft.

Der Signierungsprozess ist im Vergleich zu Symbian wesentlich einfacher. Benötigt die Anwendung nur normale Rechte, funktioniert der gesamte Signierungsablauf über eine Certificate Authority (CA). Benötigt die Anwendung jedoch auch Zugriff auf privilegierte APIs, muss zuerst die Genehmigung von Microsoft über Mobile2Market eingeholt werden.

3.5.1 *Designed for Windows Mobile* Logos

Ein zusätzliches Angebot von Mobile2Market ist die *Application Logo Certification* einer Anwendung. Dies bedeutet, dass eine Anwendung nach bestimmten Richtlinien entwickelt wird, um danach das entsprechende Logo zu erhalten [11] [12]. Diese Logo ist eine reine werbewirksame Maßnahme. Das Logo wird nur ausgesprochen, nachdem die Anwendung von einem akzeptierten Testhaus getestet wurde.

4 Fazit

Sowohl Symbian als auch Microsoft haben in den letzten Jahren viel für die Sicherheit ihrer Plattformen getan, verfolgen aber unterschiedliche Konzepte. Symbian legt den gesamten Schwerpunkt seines Sicherheitskonzepts auf die Signierung und Schutz von kritischen Gerätefunktion vor gefährlichen Programmen: nur Programme, die durch die Signierung die notwendigen Rechte haben, dürfen auf wichtige APIs zugreifen.

Das Sicherheitskonzept von Windows Mobile dagegen zeigt, dass der Großteil der Kunden Business-Kunden mit einer Microsoft-Infrastruktur sind. Microsoft legt daher besonderen Wert auf eine individuelle und maßgeschneiderte Sicherheitseinstellung der Geräte für Businesskunden.

Denn eins ist sicher: je mehr sich Kunden dem mobilen Markt zuwenden, umso interessanter werden die mobilen Plattformen auch für potentielle Angreifer.

5 Referenzen

[1] Symbian Test Houses
http://wiki.forum.nokia.com/index.php/Test_houses

[2] Freeware Route to Market FAQs
https://www.symbiansigned.com/app/page/overview/freewareFaq

[3] A developer's guide to Symbian Signed
https://www.symbiansigned.com/developerguidetoSymbianSigned.pdf

[4] How do I get my Symbian OS application signed?
how_do_I_get_my_application_signed_2.5.pdf

[5 Data caging
http://www.forum.nokia.com/document/Forum_Nokia_Technical_Library/contents/FNTL/Data_caging.htm

[6] http://msdn2.microsoft.com/en-us/library/ms889539.aspx

[7] http://msdn2.microsoft.com/en-us/library/aa455966.aspx

[8] http://msdn2.microsoft.com/en-us/library/ms890461.aspx

9 Security Model for Windows Mobile 5.0 and Windows Mobile 6
http://www.microsoft.com/technet/solutionaccelerators/mobile/maintain/SecModel/aff7cf7f-0e11-4ef4-8626-f33bd969b35a.mspx?mfr=true

[10] Dave Field, „Windows Mobile Platform Security Drilldown for the Enterprise"
http://msevents.microsoft.com/cui/WebCastEventDetails.aspx?culture=en-US&EventID=1032276839&CountryCode=US

11 Designed for Windows Mobile 6 Standard Software Application Handbook
http://download.microsoft.com/download/1/9/3/19338460-40d9-4b43-909b-f051fe52cc2b/Designed%20for%20WM%206%20Standard_Handbook_May2007_final.pdf

12 Designed for Windows Mobile 6 Professional Software Application Handbook
http://download.microsoft.com/download/1/9/3/19338460-40d9-4b43-909b-

f051fe52cc2b/Designed%20for%20WM%206%20Standard_Handbook_May2007_final.pdf