

spezielle Programme im Internet gesucht. Ursprünglich handelt es sich bei Spam um Pressfleisch in Dosen, Spiced Porc And Meat. Der Begriff Spam-Mail geht angeblich auf einen Sketch von Monty Python zurück, in dem ein Restaurantgast nur Gerichte bestellen konnte, wenn er gleichzeitig auch Spam geordert hat. In Übertragung auf das Internet bedeutet das, dass ein Benutzer die Vorzüge des Internets und insbesondere des sehr kostengünstigen Versendens von E-Mails nur nutzen kann, wenn auch bereit ist, Spam-Müll in Kauf zu nehmen.

Schäden

Untersuchungen z.B. von Nucleus Research (USA) gehen davon aus, dass jeder Mitarbeiter in einem Unternehmen im Durchschnitt täglich 13,3 Spam-Mails erhält, für die er 6,5 Minuten benötigt, um diese zu lesen und danach zu löschen. Das summiert sich jährlich auf etwa 25 Arbeitsstunden pro Kopf und macht etwa 1,4 Prozent der gesamten Arbeitszeit aus. Insgesamt ergeben sich durch die Spam-Mails erhebliche Verluste an Produktivität, die zu sehr hohen gesamtwirtschaftlichen Kosten führen. So kommt 2004 das US-Forschungsinstitut Ferris Research in einer Studie über Spam-Mails zu dem Schluss, dass allein in den USA Einbußen bis zu zehn Milliarden Dollar jährlich aufgrund spamverseuchter E-Mails anfallen. Neben den Kosten, die direkt durch diese Mails verursacht werden, müssen auch die Kosten betrachtet werden, die für die Entwicklung und Aufrechterhaltung von Antispamsystemen aufzubringen sind. Ferris Research schätzt, dass US-Unternehmen allein im Jahr 2004 dafür bis zu 120 Millionen Dollar aufbringen mussten.

Abwehr

Zur Abwehr der Spam-Flut werden Spam-Filterungen eingesetzt. Zur Filterung von Spam stehen im Internet frei verfügbare Filterprogramme zur Verfügung, wie beispielsweise SpamProbe (vgl. <http://spamprobe.sourceforge.net/>) für Unix-Derivate oder auch für Windows-Systeme unter der Cygwin-Umgebung. In Unternehmen erfolgt die Spam-Filterung häufig über zentral administrierte Mailrelays, die neben einer Spam-Filterung auch in der Regel eine Virenerkennung durchführen. Neben der Spam-Erkennung durch Spam-Filterungsprogramme muss über Regelwerke festgelegt werden, wie mit Mails, die unter Spam Verdacht stehen, umgegangen wird. Häufig beschränken sich die Filter darauf, in der Subject-Zeile der Mail diese als mögliche Spam zu markieren und ggf. nicht an den Empfänger weiterzuleiten, sondern sie in einer Spam-Quarantäne-Datei für den Empfänger zu sammeln und regelmäßig, z.B. einmal wöchentlich, dem Empfänger eine Mail zu schicken, in der die Subject-Zeilen und Absenderadressen aller als Spam eingestufter Mails zusammengefasst werden. Manche Unternehmen gehen jedoch auch soweit, Spam-Mails direkt zu löschen.

Rechtslage

Beide Vorgehen, also das Ändern der Subject-Zeile und das Löschen sind jedoch rechtlich bedenklich. Gemäß §303a des Strafgesetzbuches (StGB)

Fazit

macht sich jemand strafbar, der rechtswidrig Daten löscht, unterdrückt, unbrauchbar macht oder verändert. Ob die Subject-Zeile als geschützter Bestandteil der Mail anzusehen ist, wird in juristischen Kreisen noch stark diskutiert. Die DFN Forschungsstelle Recht in Münster rät davon ab, Subject-Zeilen zu manipulieren. Das Löschen von Spam ist laut dieser Forschungsstelle ebenfalls rechtlich nicht unbedenklich, falls in Unternehmen die private Nutzung der E-Mails zugelassen oder geduldet wird. Das Fernmeldegeheimnis ist zu beachten, und gemäß §206 Abs. 2 Nr. 2 des Strafgesetzbuches müssen grundsätzlich alle Mails ohne Verzögerung zugestellt werden. Ein Nutzer kann jedoch sein Einverständnis zum Löschen von Spam erklären.

Das Filtern von Spam-Mails ist aber auch problematisch, weil dadurch in das Fernmeldegeheimnis eingegriffen wird, das nicht nur den Inhalt, sondern auch die Umstände der Telekommunikation schützt. Nach §206 StGB kann derjenige mit einer Freiheitsstrafe bis zu fünf Jahren bedroht werden, der „unbefugt einer anderen Person eine Mitteilung über Tatsachen macht, die dem Post- und Fernmeldegeheimnis unterliegen und die ihm als Inhaber oder Beschäftigtem eines Unternehmens bekannt geworden sind, das geschäftsmäßig Post- oder Telekommunikationsdienste erbringt.“ Auch hier muss also für das Filtern die Einwilligung des E-Mail Empfängers eingeholt werden.

Nach wie vor steigen die Angriffe durch Schadsoftware weiter an. Im Gegensatz zu früher ist hierbei jedoch der Anteil an Viren und Würmer erheblich zurückgegangen und der Anteil an Trojanischen Pferden liegt bei über 70 Prozent. E-Mails sind noch immer häufig genutzte Verbreitungswege für Schadsoftware. Als Ausgangspunkt für die Verbreitung dienen aber zunehmend URLs und gefälschte Web-Seiten, die vermeintlich interessante Downloads anbieten, so dass sich Benutzer sehr häufig auf diesem Weg einen Trojaner auf ihrem System installieren. Spam-Mails oder aber Massen-Emails mit verseuchtem Anhang könnten durch die konsequente Nutzung von Signaturen eingedämmt werden. Es ist kritisch zu fragen, ob hier nicht auch die Politik und die Gesetzgebung gefordert sind, um über Regelungen zur Eindämmung der Flut nachzudenken.

2.7 Mobiler Code

Mit der Zunahme des verteilten Berechnens u.a. durch Agentensysteme und insbesondere mit der Verbreitung der plattformunabhängigen Programmiersprache Java wächst der Einsatz von so genanntem mobilen Code. Im Zusammenhang mit Smartphones sind mobile Apps in den letzten Jahren äußerst populär geworden. Die Frage der Sicherheit tritt dabei verstärkt zu Tage, da vermehrt Apps auf mobilen Endgeräten in Umlauf gebracht werden.

spezielle Programme im Internet gesucht. Ursprünglich handelt es sich bei Spam um Pressfleisch in Dosen, Spiced Porc And Meat. Der Begriff Spam-Mail geht angeblich auf einen Sketch von Monty Python zurück, in dem ein Restaurantgast nur Gerichte bestellen konnte, wenn er gleichzeitig auch Spam geordert hat. In Übertragung auf das Internet bedeutet das, dass ein Benutzer die Vorzüge des Internets und insbesondere des sehr kostengünstigen Versendens von E-Mails nur nutzen kann, wenn auch bereit ist, Spam-Müll in Kauf zu nehmen.

Schäden

Untersuchungen z.B. von Nucleus Research (USA) gehen davon aus, dass jeder Mitarbeiter in einem Unternehmen im Durchschnitt täglich 13,3 Spam-Mails erhält, für die er 6,5 Minuten benötigt, um diese zu lesen und danach zu löschen. Das summiert sich jährlich auf etwa 25 Arbeitsstunden pro Kopf und macht etwa 1,4 Prozent der gesamten Arbeitszeit aus. Insgesamt ergeben sich durch die Spam-Mails erhebliche Verluste an Produktivität, die zu sehr hohen gesamtwirtschaftlichen Kosten führen. So kommt 2004 das US-Forschungsinstitut Ferris Research in einer Studie über Spam-Mails zu dem Schluss, dass allein in den USA Einbußen bis zu zehn Milliarden Dollar jährlich aufgrund spamverseuchter E-Mails anfallen. Neben den Kosten, die direkt durch diese Mails verursacht werden, müssen auch die Kosten betrachtet werden, die für die Entwicklung und Aufrechterhaltung von Antispamsystemen aufzubringen sind. Ferris Research schätzt, dass US-Unternehmen allein im Jahr 2004 dafür bis zu 120 Millionen Dollar aufbringen mussten.

Abwehr

Zur Abwehr der Spam-Flut werden Spam-Filterungen eingesetzt. Zur Filterung von Spam stehen im Internet frei verfügbare Filterprogramme zur Verfügung, wie beispielsweise SpamProbe (vgl. <http://spamprobe.sourceforge.net/>) für Unix-Derivate oder auch für Windows-Systeme unter der Cygwin-Umgebung. In Unternehmen erfolgt die Spam-Filterung häufig über zentral administrierte Mailrelays, die neben einer Spam-Filterung auch in der Regel eine Virenerkennung durchführen. Neben der Spam-Erkennung durch Spam-Filterungsprogramme muss über Regelwerke festgelegt werden, wie mit Mails, die unter Spam-Verdacht stehen, umgegangen wird. Häufig beschränken sich die Filter darauf, in der Subject-Zeile der Mail diese als mögliche Spam zu markieren und ggf. nicht an den Empfänger weiterzuleiten, sondern sie in einer Spam-Quarantäne-Datei für den Empfänger zu sammeln und regelmäßig, z.B. einmal wöchentlich, dem Empfänger eine Mail zu schicken, in der die Subject-Zeilen und Absenderadressen aller als Spam eingestufter Mails zusammengefasst werden. Manche Unternehmen gehen jedoch auch soweit, Spam-Mails direkt zu löschen.

Rechtslage

Beide Vorgehen, also das Ändern der Subject-Zeile und das Löschen sind jedoch rechtlich bedenklich. Gemäß §303a des Strafgesetzbuches (StGB)

Fazit

macht sich jemand strafbar, der rechtswidrig Daten löscht, unterdrückt, unbrauchbar macht oder verändert. Ob die Subject-Zeile als geschützter Bestandteil der Mail anzusehen ist, wird in juristischen Kreisen noch stark diskutiert. Die DFN Forschungsstelle Recht in Münster rät davon ab, Subject-Zeilen zu manipulieren. Das Löschen von Spam ist laut dieser Forschungsstelle ebenfalls rechtlich nicht unbedenklich, falls in Unternehmen die private Nutzung der E-Mails zugelassen oder geduldet wird. Das Fernmeldegeheimnis ist zu beachten, und gemäß §206 Abs. 2 Nr. 2 des Strafgesetzbuches müssen grundsätzlich alle Mails ohne Verzögerung zugestellt werden. Ein Nutzer kann jedoch sein Einverständnis zum Löschen von Spam erklären.

Das Filtern von Spam-Mails ist aber auch problematisch, weil dadurch in das Fernmeldegeheimnis eingegriffen wird, das nicht nur den Inhalt, sondern auch die Umstände der Telekommunikation schützt. Nach §206 StGB kann derjenige mit einer Freiheitsstrafe bis zu fünf Jahren bedroht werden, der „unbefugt einer anderen Person eine Mitteilung über Tatsachen macht, die dem Post- und Fernmeldegeheimnis unterliegen und die ihm als Inhaber oder Beschäftigtem eines Unternehmens bekannt geworden sind, das geschäftsmäßig Post- oder Telekommunikationsdienste erbringt.“ Auch hier muss also für das Filtern die Einwilligung des E-Mail Empfängers eingeholt werden.

Nach wie vor steigen die Angriffe durch Schadsoftware weiter an. Im Gegensatz zu früher ist hierbei jedoch der Anteil an Viren und Würmer erheblich zurückgegangen und der Anteil an Trojanischen Pferden liegt bei über 70 Prozent. E-Mails sind noch immer häufig genutzte Verbreitungsweg für Schadsoftware. Als Ausgangspunkt für die Verbreitung dienen aber zunehmend URLs und gefälschte Web-Seiten, die vermeintlich interessante Downloads anbieten, so dass sich Benutzer sehr häufig auf diesem Weg einen Trojaner auf ihrem System installieren. Spam-Mails oder aber Massen-Emails mit verseuchtem Anhang könnten durch die konsequente Nutzung von Signaturen eingedämmt werden. Es ist kritisch zu fragen, ob hier nicht auch die Politik und die Gesetzgebung gefordert sind, um über Regelungen zur Eindämmung der Flut nachzudenken.

2.7 Mobiler Code

Mit der Zunahme des verteilten Berechnens u.a. durch Agentensysteme und insbesondere mit der Verbreitung der plattformunabhängigen Programmiersprache Java wächst der Einsatz von so genanntem mobilen Code. Im Zusammenhang mit Smartphones sind mobile Apps in den letzten Jahren äußerst populär geworden. Die Frage der Sicherheit tritt dabei verstärkt zu Tage, da vermehrt Apps auf mobilen Endgeräten in Umlauf gebracht wer-

den, die Schadcode mit sich führen. Wir gehen in diesem Abschnitt deshalb abschließend (vgl. Seite 86 ff) auf Sicherheitsprobleme speziell im Zusammenhang mit mobilen Endgeräten und Apps ein.

2.7.1 Eigenschaften

Im Gegensatz zu den in diesem Kapitel bereits eingeführten Viren, Würmern und Trojanischen Pferden wird mobiler Code nicht vordringlich zum Zwecke eines Angriffs auf einen entfernten Rechner eingesetzt. Das Konzept enthält aber ein großes Bedrohungspotential, weswegen an dieser Stelle auf allgemeine Sicherheitsprobleme im Zusammenhang mit mobilem Code eingegangen wird.

Definition 2.4 (Mobiler Code, aktiver Inhalt)

mobiler Code

Unter dem Begriff des mobilen Codes oder des aktiven Inhalts lassen sich alle Programme zusammenfassen, deren Code auf einem entfernten, potentiell nicht vertrauenswürdigen Rechner generiert wurde und die auf Gastrechnern (engl. host) ausgeführt werden.



Mobiler Code, oder ein mobiler Agent als spezielle Ausprägung davon, wandert von einem Rechnerknoten in einem Netz zu einem anderen und wird dort ausgeführt. Damit unterscheidet sich das zugrunde liegende Ausführungsmodell grundlegend vom herkömmlichen Client-Server Modell zur Durchführung verteilter Berechnungen. Beim Client-Server Modell werden Nachrichten, also Daten, als Eingabe- und Ausgabeparameter für Dienstaufrufe (z.B. Remote Procedure Call (RPC) oder Remote Method Invocation (RMI)) über ein unsicheres Transportmedium übertragen. Der ausführbare Code selbst, also der angebotene Dienst, bleibt jedoch permanent auf dem Rechner, auf dem er installiert wurde.

Die heutzutage allgegenwärtigen mobilen Apps für Android oder iPhone Geräte, die von einer App-Market-Plattform auf ein lokales Gerät heruntergeladen werden, sind bekannte Beispiele für mobilen Code.

2.7.2 Sicherheitsbedrohungen

Im Zusammenhang mit dem Einsatz von mobilem Code treten im Wesentlichen drei Klassen von Sicherheitsbedrohungen auf, die sowohl den Code selbst als auch die Ausführungsumgebung seines Gastrechners betreffen. Mobiler Code wird über ein Transportmedium übertragen, so dass er auf diesem Transportweg vielfältigen Bedrohungen ausgesetzt ist. Wird er auf einem Gastrechner zur Ausführung gebracht, so ist der mobile Code Bedrohungen seitens dieses Rechners ausgesetzt. Natürlich stellt der mobile Code

Bedrohungen

auch seinerseits eine Bedrohung für seinen Gastrechner dar. Dieser muss sicherstellen können, dass bei der Ausführung des Codes keine unautorisierten Zugriffe erfolgen, Ressourcen nicht unzulässig beansprucht oder Viren und Trojanische Pferde auf den Gastrechner übertragen werden. Die drei Bedrohungsbereiche sind in Abbildung 2.8 skizziert und werden im Folgenden genauer untersucht.

Bedrohungen des mobilen Codes

Passiven Angriffen ist der mobile Code sowohl beim Transport über ein unsicheres Medium als auch bei seiner Ausführung ausgesetzt. Aus der Sicht des Codes ist sicherzustellen, dass er nur von berechtigten Subjekten ausgeführt wird und dass bei der Ausführung seine lokalen, sensiblen Daten nicht unautorisiert an unberechtigte Dritte weitergegeben werden. Beispielsweise könnte bei einer interpretativen Ausführung des mobilen Codes auf dem Gastrechner der Interpreter der Ausführungsumgebung Zugriff auf lokale Kommunikationsschlüssel erhalten und diese zu Missbrauchszielen speichern. Falls der Gastrechner von einem externen Angreifer kontrolliert wird, so kann diese Information auch direkt an den Angreifer weitergeleitet werden.

passive Angriffe

Durch aktive Angriffe kann der Code beim Transfer oder im Verlauf der Ausführung modifiziert, oder die Ausführung des Codes kann absichtlich vorzeitig beendet werden. Dies entspricht einem Denial-of-Service (DoS) Angriff. Auf diese Weise kann zum Beispiel ein Virus oder ein Trojanisches Pferd integriert oder bei mobilen Agenten der vorgesehene Ausführungspfad verändert werden, so dass der Agent gezielt auf einen Rechner umgelenkt oder von einem vorgesehenen Gastrechner absichtlich ferngehalten wird. Bedrohungen durch Manipulationsangriffe auf den mobilen Code ergeben sich insbesondere auch im Kontext von Electronic Commerce. Hat z.B. ein Agent

aktive Angriffe

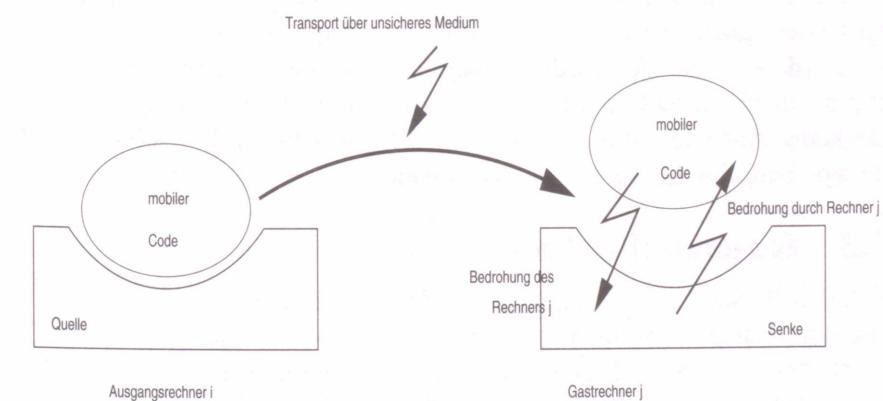


Abbildung 2.8: Bedrohungsszenarien für mobilen Code

die Aufgabe, Preise für ein Produkt aufzusammeln, so kann ein Gastrechner die bereits gespeicherten Preise von Konkurrenten löschen oder modifizieren bzw. verhindern, dass der Code den Rechner eines preisgünstigeren Kontrahenten besucht.

Bedrohungen für den Gastrechner

Mobiler Code stellt eine erhebliche Bedrohung für seinen Gastrechner dar, falls keine Maßnahmen ergriffen werden, den Zugriff auf die lokalen Ressourcen des Gastsystems zu beschränken. Durch passive Angriffe wie den lesenden Zugriff auf Dateien können Informationen gewonnen und über eine Netzwerkverbindung an den Quellrechner des Codes gesendet werden. Ein bekanntes Beispiel hierfür ist das Lesen bzw. Transferieren der Passworddatei des Gastrechners zum Rechner des Angreifers, so dass dort gezielt ein Passwort-Crackangriff durchführbar ist. Passive Angriffe können auch das Abhören der Kommunikation zwischen anderen mobilen Agenten, die auf dem System ausgeführt werden, betreffen. Auf diese Weise lassen sich Informationen über deren Aktivitäten sammeln und für einen Angriff ausnutzen.

aktive Angriffe

Mittels aktiver Angriffe kann der mobile Code versuchen, einen unautorisierten lesenden oder modifizierenden Zugriff auf die Dateien oder direkt auf die Speicherbereiche seines Gastsystems zu erlangen. Aktive Angriffe betreffen auch den Bereich des Denial-of-Service. Die Ausführung von mobilem Code kann dazu führen, dass die CPU monopolisiert wird, so dass andere Anwendungen blockiert werden. Weitere Ausprägungen von Angriffen auf die Verfügbarkeit von Ressourcen betreffen das „Lähmen“ des Gastrechners durch das Erzeugen einer Vielzahl von geöffneten Fenstern oder durch das Ausschöpfen der vorhandenen Speicherkapazität, so dass das System nur noch mit Verwaltungsmaßnahmen beschäftigt ist (so genanntes Thrashing) und die Berechnungen der Anwendungen nicht fortschreiten.

Probleme ergeben sich weiterhin aus Maskierungsangriffen seitens des mobilen Codes. Dabei ist zu unterscheiden zwischen der Vortäuschung einer falschen Identität und der Vortäuschung einer falschen Funktionalität. Zu beachten ist auch bei mobilem Code, dass Maßnahmen wie das Signieren von Code keine Auskunft über die Funktionalität geben. Auch signierter Code kann eine bedrohliche Funktionalität beinhalten!

2.7.3 Gegenmaßnahmen

Durch den Einsatz von Verschlüsselungstechniken kann auf relativ einfache Weise der Transport mobilen Codes abgesichert werden, jedoch müssen für eine interpretative Ausführung des Codes die benötigten Daten der Ausführungsumgebung unverschlüsselt vorliegen. Der Schutz dieses Programmcodes ist somit durch kryptografische Verfahren nicht zu gewährleisten. Eine

Maskierung

Verschlüsseln



Lösungsmöglichkeit besteht darin, mobilen Code nur zu solchen Systemen zu versenden, die vertrauenswürdig sind und die ihre Identität korrekt nachgewiesen haben (vgl. Kapitel 10). Mit den Möglichkeiten, die das Trusted Computing auf der Basis des Trusted Platform Modules (TPM) eröffnet (vgl. Abschnitte 11.4), stehen nunmehr auch weitergehende Mechanismen zur Verfügung, um die Vertrauenswürdigkeit von potentiellen Gastrechnern zu prüfen. Über das so genannte Remote-Attestation Protokoll, das ein TPM-Chip unterstützt (siehe Seite 621), ist es möglich, einen Gastrechner nach seiner Systemkonfiguration zu befragen. Da dies jedoch nur ein rudimentärer Mechanismus dafür ist, Vertrauen in die unmanipulierte Arbeitsweise eines Gastrechners aufzubauen, kann man auch vorab auf Gastrechnern spezielle Ausführungsumgebungen installieren, in denen dann der mobile Code zur Ausführung gebracht wird. Diese Vorgehensweise ist aus der Welt der Agentensysteme oder aber auch der DRM-Systeme (Digital-Rights-Management) bekannt.

Schutz des Gastrechners

Wie schon bei den Software-Anomalien, so ist auch bei der Abwehr von Bedrohungen durch mobilen Code eine Beschränkung der Zugriffsrechte des ausführbaren Codes unumgänglich. Eine sehr restriktive Vorgehensweise ist die so genannte Sandbox. Die Sandbox definiert die Ausführungsumgebung des Codes als einen isolierten und isolierenden Speicherbereich, innerhalb dessen der Code freien Zugriff besitzt, aber Zugriffe auf Objekte außerhalb des Bereichs kaum oder nur unter sehr strengen Kontrollen möglich sind. Die Nutzung von Virtualisierungstechniken stellen eine Verallgemeinerung der Sandbox-Technik dar und ermöglichen es, Subsysteme, die beispielsweise nur wenige Komponenten umfassen, in einer virtuellen Maschine ablaufen zu lassen, und diesen Bereich von anderen sensiblen Bereichen des Gastsystems zu isolieren.

Zur Abwehr von Maskierungsangriffen muss der Code eindeutig identifizierbar sein. Dazu werden in der Praxis vielfach Authentizitätsnachweise eingesetzt. Hierunter fällt auch die so genannte Authenticode-Technik (vgl. Seite 420), die von Microsoft als Schutz vor bedrohlichen ActiveX-Controls empfohlen und eingesetzt wird. Diese Technik bietet jedoch nur einen sehr schwachen Schutz, da lediglich der Ursprung einer mobilen Einheit nachgewiesen und überprüft werden kann. Die Überprüfung basiert darüber hinaus auf Zertifikaten, die man sich für geringes Entgelt ausstellen lassen kann. Ein signiertes und zertifiziertes ActiveX-Control oder Java-Applet erlaubt jedoch keinerlei Aussage über die Vertrauenswürdigkeit des Codes, der ausgeführt wird. Er kann Viren und Trojanische Pferde enthalten und eine absichtliche oder unabsichtliche Bedrohung des Systems darstellen. Ein Signieren von

vertrauenswürdige
Gastrechner

Sandbox

Authenticode

mobilem Code als einzige Abwehrmaßnahme ist somit völlig unzureichend und eher kontraproduktiv, da er Benutzer in einer falschen Sicherheit wiegt.

Nachweise, dass der mobile Code eine spezifizierte Funktionalität und nur diese erfüllt, sind sehr viel schwieriger zu erstellen und nachzuprüfen als Ursprungsnachweise. Mit der Technik des Proof-Carrying Codes [136] wurde ein interessanter Ansatz entwickelt. Die grundlegende Idee dabei ist, dass ein Gastrechner zunächst eine Sicherheitsstrategie festlegt und veröffentlicht. Der ausführbare, mobile Code wird vom Erzeuger mit einem Beweis ausgestattet, der nachweist, dass der Code die spezifizierte Strategie erfüllt. Vor der Ausführung dieses Codes kann dann der Gastrechner den Beweis überprüfen und damit ohne aufwändige, zur Laufzeit durchzuführende Kontrollen sicherstellen, dass bei der Ausführung des mobilen Codes keine Strategie-Verletzung auftreten wird. Zu den Eigenschaften, die auf diese Weise nachweisbar sind, gehören Aussagen darüber, dass der Code nur Speicheradressen eines bestimmten Adressbereichs verwendet oder dass keine Pufferüberläufe auftreten können, da die Einhaltung von Bereichsgrenzen kontrolliert wird. Die Erstellung von Beweisen ist jedoch keinesfalls eine triviale Aufgabe und nur in Ausnahmefällen automatisch durchführbar. Deshalb wird diese Technik wohl erst dann auf eine breite Akzeptanz stoßen, wenn einfach handhabbare Entwicklungsumgebungen zur Konstruktion solcher erweiterter Code-Bausteine zur Verfügung stehen.

2.7.4 Mobile Apps

Mobile Apps¹⁴ sind bereits heute im Consumer-Bereich sehr weit verbreitet. Derzeit wird noch ein großer Teil der Apps als rein lokale Anwendungen auf mobilen Geräten ausgeführt, jedoch ist die Entwicklung hin zu smarten mobilen Apps deutlich zu sehen. Kennzeichnend für diese fortgeschrittenen Apps ist ihre Eigenschaft, Dienste zu nutzen, die über das Internet oder ein privates Netzwerk bereitgestellt werden. Viele dieser Dienste werden bereits heute in einer Cloud betrieben und ermöglichen es dem Nutzer so mit, einen konsistenten Datenbestand auf unterschiedlichen mobilen und stationären Geräten zu führen. Eine App stellt dabei die Client-Seite eines Cloud-Dienstes dar, so dass es für den Nutzer keine Rolle mehr spielt, von wo und mit welchem Gerät der Zugriff auf seine Daten erfolgt. Mit diesen Eigenschaften werden smarte Apps zunehmend auch für den Geschäftsbe reich attraktiv. Als Business-Apps werden sie ein integraler Bestandteil von Geschäftsprozessen werden.

¹⁴ Der Abschnitt basiert auf einem Teil eines Buchkapitels, das die Autorin zusammen mit Christian Schneider verfasst habe. Das Buch ist unter dem Titel *Smart Mobile Apps* im Springer Verlag erschienen.

Sicherheitsbedrohungen

Smartphones und Tablet-PCs werden von Nutzern als eine Art persönliche Informationszentralen verwendet. Die Geräte sind *always on* und werden von vielen Benutzern stets mitgeführt. Von zentraler Bedeutung für die Sicherheit der Anwendungen, und damit natürlich auch der Apps, ist die Systemsicherheit der mobilen Plattform, auf der die Anwendungen zur Ausführung gelangen. Der zweite sicherheitskritische Bereich wird durch die Anwendungsebene definiert. Sie bildet die direkte Ausführungsumgebung für die Apps.

Bedrohungen der Mobilen Plattformen

Mobile Plattformen unterliegen einer höheren Gefährdung als stationäre. Sie werden an verschiedenen Orten und damit in unterschiedlich vertrauenswür digen Umgebungen eingesetzt. Sie gehen schneller verloren oder werden gestohlen als etwa ein PC. Gleichzeitig kann sich ein Unbefugter Zugang zu einem mobilen Gerät verschaffen. Die kritischste Bedrohung für mobile Plattformen ist jedoch, dass solche personalisierten Geräte oftmals nicht oder nur unzureichend von einer unternehmensweiten Sicherheitsstrategie erfasst werden. Meistens werden sie von den Benutzern selbst adminis triert. Die Erfahrung aus der Welt der Desktop-Betriebssysteme lehrt jedoch, dass noch immer vielen Anwendern ein hinreichendes Sicherheitsbewusst sein fehlt: Zugriffspermen und Passwortschutz werden deaktiviert, da sie als störend empfunden werden, Sicherheitswarnungen werden ignoriert, Software-Updates werden, wenn überhaupt, sehr unregelmäßig eingespielt. In solchen Fällen vergrößert sich entsprechend das bereits vorhandene Bedrohungspotential mit einer unnötig großen Angriffsfläche der mobilen Plattform.

Viele mobile Betriebssysteme müssen für das Einspielen von Updates mit einem Rechner verbunden werden. Sogenannte *Over-the-Air* (OTA)-Updates, also Firmware-Updates ohne PC-Verbindung, sind bisher nur für Android, Windows Phone 7 und Symbian 3 verfügbar, unter gewissen Voraussetzungen auch für BlackBerry OS. Ältere Windows Mobile und Symbian-Versionen sowie Apples iOS bieten bisher nur Betriebssystem-Updates mit Unterstützung eines PCs. Googles Android-Plattform, bietet zwar OTA-Updates, leidet aber unter einem anderen Problem: Mit den schnellen Release-Zyklen von neuen Android-Versionen durch Google kön nen viele Hersteller von mobilen Geräten nicht mithalten. Sie stellen deshalb nur für einen kurzen Zeitraum Firmware-Updates für ihre Produkte zur Verfügung, und das auch nur mit deutlicher Verzögerung zur Android-Version¹⁵. Erschwerend kommt hinzu, dass die Mobilfunkbetreiber viele Smartphones

¹⁵ vgl. Namestnikov, Yury: Malware-Report, 2011. Kaspersky Lab, <http://www.viruslist.com/de/>

mit speziellen Anpassungen vertreiben. Diese benötigen jedoch ihrerseits speziell angepasst Firmware-Updates, die in der Regel wiederum noch später als die Updates der Hardware-Hersteller zur Verfügung gestellt werden. Die Konsequenz ist, dass auf den meisten Android-Geräten veralteter System-Software installiert ist. Beispielsweise wurde Android 2.3 im Dezember 2010 veröffentlicht, doch nach Angaben von Google war ein halbes Jahr später auf 64,6 Prozent aller Geräte noch Version 2.2 installiert. Im Mai 2011 liefen insgesamt nur 8,1 bzw. 0,3 Prozent aller Android-Geräte mit der aktuellen Version 2.3.4 für die Smartphone-Klasse beziehungsweise Version 3.1 für die Tablet-Klasse. Das heißt, über 90 Prozent aller Android-Geräte verwenden eine veraltete Betriebssystemversion, die möglicherweise bekannte aber noch nicht geschlossene Sicherheitslücken enthält.

Bedrohungen der Anwendungsebene

Malware-App

Die meisten Apps werden über einen speziellen App-Marktplatz bekannt gemacht und vertrieben. Für die mobilen Endgeräte von Apple (iPhone, iPad und iPod) ist Apple's AppStore sogar der einzige Weg, um neue Anwendungen zu beziehen und auf den Geräten zu installieren. Der Nutzer muss also der Qualitätskontrolle der Marktplatzbetreiber vertrauen. D.h. der Nutzer muss darauf vertrauen, dass die über den Marktplatz angebotenen Apps keine Sicherheitslücken oder Schad-Funktionen aufweisen. Da die entsprechenden Kontrollverfahren nicht offen gelegt sind und Apps keine standardisierte Zertifizierungsprozedur durchlaufen müssen, kann über die Qualität der durchgeführten Kontrollen keine verbindliche Aussage getroffen werden. So musste Apple in der Vergangenheit wiederholt Apps aus dem Store zurückrufen, da Sicherheitslücken entdeckt wurden. Für den offenen Android-Marktplatz sieht die Situation sogar noch schlechter aus. Der bereits zitierte Malware-Report von Kaspersky Lab von 2011 zeigt auf, dass allein im ersten Quartal 2011 über 50 Apps mit trojanischen Funktionalitäten aus dem Googles Android-Market entfernt werden mussten. Laut dem Bericht ist es nur eine Frage der Zeit, bis mobile Botnetze im großen Stil auftauchen, wie wir sie bisher nur aus der Welt der Desktops kennen.

Die Problematik des Vertrauens in Closed-Source-Software besteht zwar im Prinzip auch bei Software, die auf Desktop-Betriebssystemen installiert wird. Auf einer mobilen Plattform wird dieses Problem jedoch dadurch verschärft, dass beispielsweise Smartphones viele verschiedene Datenquellen aggregieren und diese über eine einheitliche Schnittstelle leicht für andere Anwendungen zugänglich machen. Darüber hinaus können schadhafte Apps ihrem Besitzer direkt beträchtliche Kosten verursachen, in dem etwa SMS-Nachrichten an Mehrwertdienste gesendet oder Sonderrufnummern angerufen werden. Wenn sich das Smartphone durch die NFC-Technologie in Zukunft auch noch als mobile Geldbörse etabliert, wie es derzeit von Goo-

gle mit großen Partnern aus dem Finanzsektor als *Google Wallet* für seine Android-Plattform forciert wird (vgl. <http://www.google.com/wallet/>), wird sich diese Problematik voraussichtlich weiter verschärfen.

Eine App kann aber umgekehrt auch nicht zuverlässig feststellen, ob sie in einer sicheren, nicht modifizierten Umgebung ausgeführt wird. Eine durch Schadcode manipulierte Ausführungsumgebung könnte die App missbrauchen, um gezielt Daten auszuspähen, zu manipulieren oder über die App Zugriffe auf die Daten anderen Benutzer, andere Geschäftsprozesse etc. zu gelangen. Es stellt sich die Frage, inwieweit die Daten einer Anwendung vor unberechtigten Zugriffen durch eine andere Anwendung geschützt sind. Hier ist zunächst die Zugriffskontrolle des mobilen Betriebssystems zu betrachten. Je nach Hersteller sind hier sehr starke Unterschiede zu verzeichnen. Die Größe der zu schützenden Einheit, also die Granularität der Kontrolle, spielt dabei eine wesentliche Rolle. Je geringer die verfügbare Granularität, desto größer sind die zu schützenden Einheiten und desto mehr Zugriffsrechte erhält eine App, auch wenn sie diese nicht im vollen Umfang benötigen würde. Da neue Software von unbekannten Autoren heutzutage auf einfachste Weise aus dem App-Marktplatz auf die mobilen Geräte gelangt, sollte die Isolation der Apps untereinander und die Beschränkung der Rechte auf das Nötigste oberstes Gebot sein.

Aber auch legitime Zugriffe können missbraucht werden und damit unerwünschte Folgen haben. Beispielsweise gleicht die offizielle Facebook-App, die für alle populären mobilen Plattformen verfügbar ist, die Kontaktliste auf Smartphones mit den Freunden auf Facebook ab, wozu die App natürlich Zugriff auf die Kontakte des Geräts benötigt. Allerdings lädt die App auch alle Kontakte in das Phonebook hoch, die nicht bei Facebook sind, ohne dass der Benutzer dies unterbinden kann. Auf diese Weise gelangt Facebook an viele Kontaktdaten von Nicht-Mitgliedern und kann diese über die Telefonnummern als Identifikatoren verknüpfen, um das soziale Netzwerk auch über die Mitglieder hinaus zu erweitern. Dieses Beispiel verdeutlicht, warum die Aggregation verschiedener Datenquellen auf einem Gerät und die Verfügbarkeit dieser Daten über eine systemweite Schnittstelle eine besondere Bedrohung darstellt.

Gegenmaßnahmen, Lösungsansätze

Die Hersteller von mobilen Betriebssystemen, Endgeräten und klassischen Sicherheitslösungen sind sich der zuvor beschriebenen Bedrohungen natürlich bewusst und versuchen diesen auf unterschiedliche Weise zu begegnen. Im Folgenden wird auf einige Maßnahmen exemplarisch eingegangen.

Unsichere Umgebung

Android

Google folgt auf seiner Android-Plattform der Philosophie, dass den Apps und ihren Autoren grundsätzlich nicht vertraut werden kann. Entsprechend restriktiv sind hier die Vorgaben für die Zugriffsrechte, so dass jede App zunächst nur ihre eigenen Daten lesen und schreiben darf. Um auf systemweite Daten, wie etwa die Kontakte oder den Kalender, zuzugreifen, muss eine App die dafür vorgesehene Schnittstelle zum Android-System verwenden. Diese Schnittstellen unterliegen einem speziellen Kontrollmechanismus. Damit der App diese Zugriffe vom System gewährt werden, muss sie schon bei der Installation angegeben haben, dass sie diese Berechtigung erwünscht. Die dem Vorgehen zugrundeliegenden allgemeinen Prinzipien des *default deny* und des *need-to-know* sind im Grundsatz sehr positiv. Allerdings definiert Android annähernd 200 verschiedene Rechte, die den Zugriff auf die persönlichen Daten regeln, oder aber auch auf Geräte wie Mikrofon, Kamera und GPS-Empfänger. Daneben vergibt Android Rechte zur Verwendung von Systemfunktionen, wie das Senden von SMS-Nachrichten und das Initiieren von Telefongesprächen. Diese Berechtigungen muss der Benutzer bestätigen, bevor die App installiert wird. Dieses Vorgehen schafft eine gewisse Transparenz, verlagert aber damit das Problem auf den Benutzer, der ein entsprechendes Problembewusstsein haben muss, um eine sinnvolle Entscheidung treffen zu können.

iOS

Apples iOS verfügt nicht über solche feingranularen Berechtigungen, sondern verfolgt eine andere Strategie zur Zugriffskontrolle. Anders als auf der Android-Plattform erlaubt Apple die Installation von Apps ausschließlich über den Apple AppStore. Dadurch gibt es für iOS-Geräte einen zentralen Punkt, an dem Apple regulierend eingreifen kann. Alle Programme, die in den AppStore eingestellt werden, müssen sich zunächst einem Review-Prozess unterziehen. Dabei analysiert Apple die verwendeten Systemfunktionen und testet das Programm rudimentär. Verwendet eine App Systemfunktionen, die sie zur Erbringung ihrer Funktionalität aber gar nicht benötigt, wird das Programm abgelehnt und nicht im AppStore aufgenommen. Hier übernimmt also der Hersteller in gewisser Weise die Verantwortung, dass eine App nur die Zugriffsrechte erhält, die sie benötigt. Die skizzierten Maßnahmen wie Zugriffskontrollen und Transparenz verhindern jedoch nicht, dass Apps mit Schadfunktionen auf ein mobiles Endgerät gelangen können. Das haben auch die Hersteller von Sicherheits-Software erkannt und bieten mobile Varianten ihrer Security-Suiten an. Der Funktionsumfang dieser Produkte reicht vom obligatorischen Viren-Scanner über ein entferntes Löschen der Daten (Remote-Wipe) bei Verlust oder Diebstahl des Endgeräts bis hin zu Rufnummernsperrlisten und SMS-Spam-Filtern.

3 Internet-(Un-)Sicherheit

Kapitelüberblick

Nach einer knappen Einführung in Abschnitt 3.1 gibt der Rest des Kapitels einen Überblick über wesentliche Sicherheitsprobleme im Zusammenhang mit den bekannten Internet-Protokollen. Abschnitt 3.2 stellt die Internet-Protokollfamilie zusammen mit dem OSI- und dem TCP/IP-Referenzmodell vor. Die für das Verständnis der Sicherheitsprobleme wichtigsten Eigenschaften der Internet-Protokolle werden erklärt, bevor deren Sicherheitsprobleme in Abschnitt 3.3 diskutiert werden. Abschnitt 3.4 beschäftigt sich mit weit verbreiteten Diensten wie DNS und WWW und erläutert Sicherheitsprobleme, die bei deren Nutzung auftreten können. Abschließend gehen wir in Abschnitt 3.6 auf eine Auswahl von Analysetools ein, die zur Härtung und Sicherung von Systemen sinnvoll einsetzbar sind.

3.1 Einführung

Informationstechnologie durchdringt heutzutage in gleichem Maße die privaten wie geschäftlichen Bereiche unseres Lebens. Mit der ständig zunehmenden Vernetzung von IT-Systemen und deren Anbindung an öffentliche Netze wie das Internet gewinnen Fragen der Informationssicherheit zunehmend an Bedeutung. Sicherheitsbedrohungen ergeben sich durch das unautorisierte Lesen elektronischer Nachrichten, die sensible, personenbezogene Informationen oder Geschäftsdaten beinhalten können, durch das unautorisierte Verändern von gesendeten Daten, so dass gefälschte Informationen beim Empfänger ankommen, oder auch durch das Maskieren und Vortäuschen einer falschen Absenderidentität, so dass der Kommunikationspartner im Vertrauen auf diese Identität vertrauliche Informationen preisgibt. Aufgrund der Bedeutung des Internets konzentriert sich dieses Kapitel auf Sicherheitsfragen im Zusammenhang mit den für das Internet festgelegten Kommunikationsprotokollen – der Internet-Protokollfamilie.

Wurzel

Die Ursprünge des Internets gehen auf die Entwicklung des ARPA-Netzes (Advanced Research Projects Agency) zurück, das in den frühen 70er Jahren durch das amerikanische Verteidigungsministerium initiiert wurde. Die für das ARPA-Netz entwickelten Protokolle (Internet Protocol (IP), Internet Control Message Protocol (ICMP), Transmission Control Protocol (TCP)) sind die Grundlage der heutigen Internetprotokolle. Der 1.1.1983 gilt als