Frühzeitige modellbasierte Risikoanalyse für mobile, verteilte Anwendungen

Christian Wessel[1] Thorsten Humberg[2] Sven Wenzel[1]
Jan Jürjens[1]

[1]Technische Universität Dortmund, 44227 Dortmund
[2]Fraunhofer-Institut für Software- und Systemtechnik (ISST), 44227 Dortmund

Abstract: Anwendungen, die mobile Komponenten beinhalten, sind besonderen Risiken ausgesetzt. Dabei stellt beispielsweise die Kommunikation bei mobilen, verteilten Anwendungen eine Herausforderung an die Daten- und Abhörsicherheit dar. Im Kontext dieser Anwendungen ist daher eine gesonderte Schutzbedarfsanalyse erforderlich. In diesem Papier wird am Beispiel eines Informationssystems mit mobil angebundenen Clients gezeigt, wie eine durchgängige Sicherheits- und Risikoanalyse in den modellbasierten Entwicklungsprozess eines solchen Systems integriert werden kann. Mögliche Schwachstellen des Systems können so bereits in frühen Entwicklungsphasen erkannt und beseitigt werden.

1 Motivation

War bis vor einigen Jahren noch undenkbar, dass beispielsweise Mobiltelefone zu mehr zu gebrauchen sind als nur für Telefonate, so werden Smartphones, Tablet-PCs und andere mobile Endgeräte heute zunehmend in komplexe IT-Systeme und Geschäftsprozesse integriert. Einhergehend mit diesen neuen Möglichkeiten kommen natürlich auch neue Risiken ins Spiel. Während man ein lokal begrenztes System ausreichend und zuverlässig gegen Angriffe und Ausfälle schützen kann, steht man bei mobil verteilten Systemen vor neuen Herausforderungen, wenn z.B. sensible Daten die sicheren Firmennetze verlassen [Eck06]. Die Übertragung der Daten von/zu einem Außendienstmitarbeiter kann z.B. durch geeignete Verschlüsselung geschützt werden. Dass sie mit dem Verlust/Diebstahl des Geräts aber doch in falsche Hände gelangen können oder das gestohlene Gerät an sich ein Angriffspunkt für die Firmen-IT darstellt, bleibt oft unberücksichtigt. Den o.g. Risiken und Sicherheitslücken kann bereits in der Planungsphase eines IT-Systems entgegen gewirkt werden. Ein Risiko im Sinne dieser Arbeit ist eine sicherheitsrelevante Schwachstelle eines Computersystems oder die ungenügend gesicherte Kommunikation zwischen diesen. Die Sicherheit von Softwaresystemen sollte bereits in der Planungsphase berücksichtigt werden, da das Durchsetzen von Sicherheitsrichtlinien im Nachhinein nur sehr schwer möglich, bzw. unmöglich ist. Insbesondere bei modellgetriebenen Entwicklungsmethoden bieten die existierenden Softwaremodelle eine gute Grundlage für erste fundierte Risiko- und Sicherheitsanalysen. Ausgehend von den bestehenden Arbeiten zur modellbasierten Sicherheits- und Risikoanalyse (Abschnitt 2) schlagen wir im Folgenden eine neue Me-

thodik vor, mit der entsprechende Analysen bereits in besonders frühen Planungsphasen, also noch weit vor dem detaillierten Softwareentwurf, möglich werden, wobei auch Complianceanforderungen berücksichtigt werden. Kern unserer Methodik (Abschnitt 3) ist die Integration von Geschäftsprozessen, die z.B. als BPMN-Modelle [FR10] vorliegen, und der geplanten Verteilung auf verschiedene Systemkomponenten, z.B. mit UML Deployments [OMG05]. Auf Basis dieser einfachen Modelle können bereits grundlegende Risiken lokalisiert werden, wie wir am Beispiel eines IT-Systems für den Direktvertrieb zeigen werden. Abschließend diskutieren wir die vorgeschlagene Methodik und erörtern offene Forschungsfragen (Abschnitt 4).

2 Bestehende Ansätze

Es existieren bereits verschiedene Ansätze zur modellbasierten Sicherheitsanalyse. Zwei bekannte Beispiele, um Sicherheitsanforderungen in Softwaremodellen zu spezifizieren, stellen UMLsec [Jür04] und SecureUML [LBD02] dar. Während SecureUML eine Erweiterung der UML zur Zugriffsverwaltung und -kontrolle (RBAC) darstellt, ermöglicht UMLsec auch die Spezifikation zahlreicher weiterer Sicherheitseigenschaften in UML-Modellen. Existierende Werkzeuge[1] erlauben eine automatisierte Prüfung dieser Eigenschaften, z.B. durch Konsistenzprüfungen. Eine Anwendung von UMLsec zur Analyse mobiler Endgeräte wurde bereits in [Bar06] gezeigt.

Auch zur allgemeinen IT-Risikoanalyse existieren erste Werkzeuge. Ein Beispiel ist der *RiskFinder*, der UML-Modelle auf sicherheitsrelevantes Vokabular hin untersucht und mögliche Gefahrenquellen bzw. Risiken hervorhebt [PHJB11]. Schneider et. al. schlagen in [SKH+11] eine heuristische Suche vor, welche Sicherheitsanalysen auf Grundlage von Bayes-Filtern durchführt. *HeRA* stellt einen Feedback-basierten Ansatz zur Sicherheitsprüfung während der Anforderungsanalyse zur Verfügung [KLM09]. Der Ansatz stellt zwar mächtige Regeln bereit, die auch auf dem verwendeten Vokabular arbeiten, jedoch beziehen sich diese stets auf einzelne Wörter und beziehen keine Textdatenbanken ein.

Es existiert ebenfalls ein Ansatz in [Wol08] um Sicherheitsanforderungen in BPMN-Modellen darzustellen. Diese beziehen sich jedoch auf die Darstellung von Sicherheitsmaßnahmen in einem geschlossenen System. Der in dieser Arbeit vorgestellte Ansatz berücksichtigt ebenfalls die spätere Verteilung der Softwarekomponenten.

3 Frühzeitige modellbasierte Sicherheits- und Risikoanalyse

Prozessmodellierungssprachen, wie z.B. BPMN, dienen der Visualisierung von Geschäftsprozessen oder Arbeitsabläufen. Es können z.B. Dokumente oder Informationen modelliert werden, die während eines Prozesses ausgetauscht werden. Verschiedene Akteure, die an einem Prozess beteiligt sind, können durch sogenannte *Swimlanes* dargestellt werden.

[1] z.B. `http://www.umlsec.de/`

Geschäftsprozessmodelle sind i.d.R. schon vor Erstellung eines IT-Systems vorhanden, insbesondere wenn das System zur Unterstützung dieser Prozesse realisiert werden soll.

Um zu planen, welche Programmkomponenten auf welche Teile des Systems (insb. Hardware) verteilt werden, bieten sich UML Deployment-Diagramme an. Hier steht jedoch nicht die feingranulare Verteilung einzelner Artefakte im Vordergrund, sondern der grundlegende Aufbau (Grobentwurf) eines Systems soll betrachtet werden. So wird z.B. identifiziert, dass es Tablet-PCs für Außendienstmitarbeiter geben wird. Wir können daher annehmen, dass BPMN-Modelle und UML Deployment-Diagramme bereits in sehr frühen Entwicklungsphasen gegeben sind.

Beispiel. Abbildung 1 zeigt ein Beispiel in Anlehnung an den Bestellprozess der fiktiven Direktvertriebsfirma *Eisfrost*. Eine Bestellung kann dabei entweder sofort ausgeführt werden, falls sich die gewünschte Ware im Fahrzeug befindet, oder für die nächste Tour des Verkaufsfahrers vorgemerkt werden. Das dazugehörige Deployment-Diagramm zeigt die Systemstruktur. Dabei handelt es sich lediglich um einen möglichen Entwurf des Systems und stellt noch nicht die endgültige Architektur dar. Der Verkaufsfahrer benutzt für die Bestellung des Kunden vor Ort einen Tablet-PC, um die Kundendaten auszuwählen und die Bestellung entgegen zu nehmen. Der Tablet-PC kommuniziert per Bluetooth mit der *Central Car Unit (CCU)*, die sich im Fahrzeug befindet. Die CCU übernimmt die Kommunikation mit dem *Enterprise Resource Planning* (ERP)-System in der Firmenzentrale, um die Kundendaten abzurufen und die Bonität des Kunden vor der Ausführung der Bestellung zu prüfen.Das Ergebnis der Prüfung wird dann an den Tablet-PC des Verkaufsfahrers gesendet, der auf Grundlage der Daten entscheidet, den Bestellvorgang durchzuführen oder abzubrechen.

Vorgehensweise. Das oben gezeigte Beispiel beinhaltet einige Risiken und Sicherheitsprobleme, die man mit einer systematischen Analyse aufdecken kann. Die hier vorgeschlagene Vorgehensweise zur Untersuchung der Modelle beinhaltet im Wesentlichen drei Schritte:

1. Untersuchung der (physikalischen) Verteilung
2. Untersuchung der Kommunikation
3. Analyse (funktionaler und nicht-funktionaler) Risiken

Die einzelnen Schritte sollen im Folgenden eingehender betrachtet werden.

Untersuchung der Verteilung. Die Verteilung der beteiligten Komponenten auf physikalische Systeme ist von zentraler Bedeutung für die Sicherheit eines Systems. Insbesondere mobile Komponenten unterliegen Gefährdungen, die auf stationäre Systeme nicht zutreffen. Beispielsweise können mobile Systeme verloren gehen oder gestohlen werden. Auch das Abhören oder Manipulieren der Kommunikation der mobilen Komponente stellt

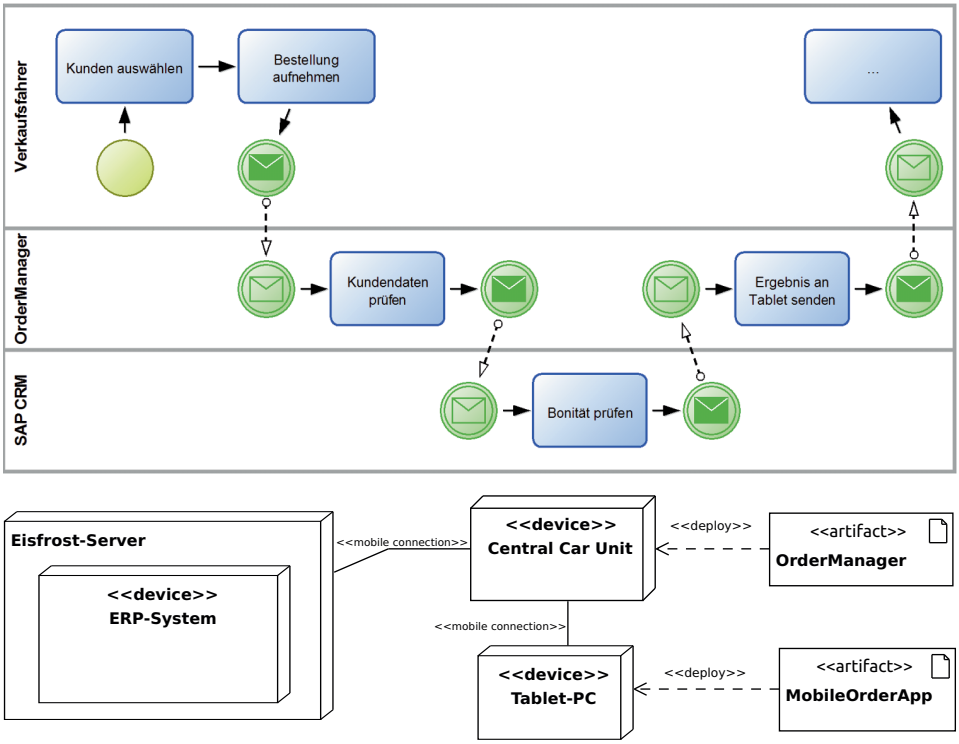

Abbildung 1: Exemplarischer Kundendatenabruf durch ein mobiles System vom zentralen Server

ein erhebliches Risiko dar. Die Verteilung des Systems kann anhand von Deployment-Diagrammen identifiziert werden.

Sobald die Klassifizierung feststeht, kann eine Zuordnung der Komponenten des Deployment-Diagramms auf die Akteure des BPMN-Diagramms vorgenommen werden. Für die Komponenten und Akteure, welche eine erhöhte Aufmerksamkeit hinsichtlich ihrer Gefährdung benötigen, ist es empfehlenswert, diese entsprechend zu visualisieren, um so den Fokus auf diese zu lenken.

Die Zuordnung von Akteuren aus dem BPMN-Modell auf die Komponenten des Deployment-Diagramms kann in Form einer einfachen Tabelle erfolgen. Ein Beispiel für das in Abbildung 1 gezeigte Szenario ist in Tabelle 1 ersichtlich. Mit Hilfe dieser Zuordnung ist es möglich, eine kombinierte Sicht aus Geschäftsprozess und Verteilungsdiagramm zu generieren und die Sicherheitsanalyse darauf auszuführen.

Untersuchung der Kommunikation. Auch die verschiedenen Arten der Kommunikation sind wichtige Merkmale, da diese unterschiedlichen Risiken ausgesetzt sein können. Diese lassen sich ebenfalls aus dem Deployment-Diagramm extrahieren. Ein mögliches Beispiel ist die Unterteilung der Kommunikationsverbindungen in Funk, lokales Netzwerk und Internet. Somit steht für den Nachrichtenaustausch ebenfalls fest, welches Kommu-

Geschäftsprozess	Verteilungsdiagramm
SAP CRM	Eisfrost-Server/ERP-System
OrderManager	OrderManager
Verkaufsfahrer	MobileOrderApp

Tabelle 1: Zuordnung der Akteure des Geschäftsprozesses zu den Komponenten des Verteilungsdiagramms

nikationsmedium dafür benutzt wird. Die besonders schützenswerten Kommunikationskanäle können dann ebenfalls visualisiert werden.

In UMLsec existiert für diesen Zweck den Stereotyp `<<Secure Links>>`. Mit diesem werden Sicherheitsanforderungen für Datenübertragungen gefordert. Jede Verbindung zwischen zwei Knoten kann dann mit dem jeweiligen Leitungstyp, z.B. `<<Internet>>` oder `<<encrypted>>`, annotiert werden. Die übertragenen Daten können z.B. als geheim (`<<secrecy>>`) markiert werden, so dass eine automatische Prüfung erfolgen kann, ob eine Verbindung für entsprechende Daten geeignet ist [Jür04].

Im Beispiel ist ersichtlich, dass eine drahtlose Kommunikation zwischen dem Tablet-PC und der CCU und damit auch mit dem ERP-System stattfindet. Um dieses Schutzbedürfnis im Modell zu manifestieren, schlagen wir eine Erweiterung für BPMN vor, die analog zu UMLsec definiert ist. Dabei geht es in erster Linie aber nicht um die konkrete Kommunikationsart, sondern eher um die Art der Daten, die übertragen werden sollen. Beispielsweise bei personenbezogenen Daten sollte dieses entsprechend im BPMN-Modell annotiert werden, indem der Nachrichtenfluss mit einer besonders gearteten Textannotation versehen werden kann, deren Semantik analog zu einem UMLsec-Stereotyp sein kann. Mit dieser Erweiterung können dann auch Schutzbedarfsanalysen direkt auf dem Geschäftsprozessmodell durchgeführt werden.

Analyse der Risiken. In den vorhergehenden beiden Schritten wurden Gefährdungen betrachtet, die sich aus der Verteilung des Systems und dessen Kommunikation ergeben. Es existieren jedoch weitere Risiken aufgrund der ausgeführten Aktivitäten selbst, deren Identifikation wir wie folgt ermöglichen wollen.

Basierend auf etablierten Standards der IT-Sicherheit, z.B. den Katalogen und Standards des *Bundesamtes für Sicherheit in der Informationstechnik* (BSI)[2] und den ISO-Normen der 27000-Reihe [ISO05], lassen sich sicherheitsrelevante Aktivitäten identifizieren, indem wir das Vokabular der einzelnen Aktivitäten analysieren. Hierzu kann der *RiskFinder* eingesetzt werden [PHJB11]. Wir schlagen jedoch eine Abstraktion von konkreten Notationen vor. So besteht ein zu untersuchender Prozess lediglich aus einer Menge (unabhängiger) *Aktivitäten*, denen jeweils eine Menge von Texten zugeordnet ist. In üblichen Notationen sind dies Titel von Aktivitäten und ggf. ergänzende Kommentare. Dies ist vor allem für die Untersuchung von Geschäftsprozessen geeignet, wo eine Vielzahl formaler und semi-formaler Notationen verwendet werden [Fra11]. Kann einer Aktivität ein Risiko zugeordnet werden, so kann sie entsprechend markiert werden.

[2] `https://www.bsi.bund.de`

Darüber hinaus können auch kritische Strukturen im Prozess aufgedeckt werden, die aus dem Fehlen von Aktivitäten resultieren. Im vorgestellten Beispiel wird bei der Übermittlung der Daten des Kunden keine gesonderte Authentifizierung gefordert. Es muss also davon ausgegangen werden, dass das Login implizit geschieht und die dafür nötigen Zugangsdaten im Gerät gespeichert sind. Bei Verlust des Tablet-PCs oder der CCU, z.B. durch Diebstahl, ist es also möglich, ohne entsprechendes Login auf Firmendaten zuzugreifen.

Weiterhin kann die Risikoanalyse die zuvor betrachtete Untersuchung der Kommunikation unterstützen. So können z.B. Nachrichten, die von Aktivitäten ausgelöst werden, welche schützenswerte Daten verarbeiten, selbst als kritisch gekennzeichnet werden.

4 Diskussion und offene Forschungsfragen

Die Risiko- und Sicherheitsanalyse ist ein unumgänglicher Schritt bei der Entwicklung von Informationssystemen. Insbesondere bei mobilen, verteilten Anwendungen entstehen neue Risiken. In diesem Papier haben wir eine neue Methode zur modellbasierten Analyse solcher Systeme vorgeschlagen, die bereits auf frühphasigen Dokumenten wie Prozessmodellen und groben UML Deployment-Diagrammen angewendet werden kann.

Die Geschäftsprozessmodelle sollten in der Regel schon vorhanden sein und Verteilungsdiagramme lassen sich auch ohne größere Detailkenntnisse entwerfen. Unser Vorschlag ist dabei jedoch nicht auf BPMN-Modelle festgelegt. Für die reine Risikoanalyse sind allein die Bezeichner der Prozessschritte/Aktivitäten ausreichend. Die aktuelle Version des *RiskFinders* wird derzeit dahingehend überarbeitet, dass auch andere Datenquellen möglich sind. Zudem integrieren wir Textdatenbanken zur Auswertung z.B. von synonymen und kookkurenten Begriffen, sodass die Trefferquote zusätzlich präzisiert wird.

Ein Vorteil unseres Ansatzes gegenüber den bestehenden ist dabei, dass die Sicherheitspattern systematisch auf die gefundenen Aktivitäten angewendet werden und damit eine umfassende Sicherheitsanalyse ermöglicht.

Eine weitere nötige Erweiterung des RiskFinders besteht darin, auch das Nichtvorhandensein bestimmter Eigenschaften aufzudecken (vgl. die fehlende Login-Informationen im Beispiel). Dies ist kein triviales Problem, da eine negierte Suche nach Schlüsselwörtern nicht ausreichend ist. Vielmehr muss der Kontext des Nichtvorhandenseins betrachtet werden, um nicht zu viele Falschmeldungen zu generieren.

Die Untersuchung der Kommunikation ist für UML Entwurfsmodelle bereits realisiert worden [Jür04]. Wie oben gezeigt, lässt sich das Konzept problemlos auf Geschäftsprozesse übertragen. Hier ist noch genauer zu evaluieren, wie die Angaben zum Schutzbedarf von Nachrichten sinnvoll in die Modelle integriert werden können. Erweiterungsmechanismen analog zu UML-Stereotypen gibt es in BPMN nicht, die Anmerkungselemente könnten sich evtl. dafür anbieten. Außerdem ist noch eine geeignete Heuristik zu entwerfen, mit der der *RiskFinder* den Schutzbedarf für Datenübertragungen vorschlägt. Insbesondere Transitivitätseigenschaften sind noch zu diskutieren, da nicht immer klar ist, welche Informationen eines Prozessschritts in späteren Schritten weitergenutzt werden. Schlussendlich ist noch eine entsprechende Werkzeugunterstützung für die Konsistenzprüfung zwischen

Schutzbedarf der Nachrichten und verwendetem Kommunikationskanal zu realisieren.

Insgesamt soll die Werkzeugunterstützung dahingehend verbessert werden, dass ein integriertes Werkzeug zur Verfügung steht. Wir denken dabei an eine Integration in das Analysewerkzeug *CARiSMA*[3], um dem Anwender eine ganzheitliche Sicht zu bieten. Hierbei besteht noch Bedarf an einer geeigneten grafischen Benutzerschnittstelle.

Es wäre ebenfalls interessant zu diskutieren, in weit der Ansatz in schon vorhandene Szenarien skaliert, in denen Geschäftsprozessmodelle zur Orchestrierung verwendet werden. Ein verwandter Ansatz zu diesem Thema ist bereits in [Men09] genannt worden.

Literatur

[Bar06] P. Bartmann. *Modellbasierte Sicherheitsanalyse mobiler Endgeräte*. Diplomarbeit, Technische Universität München, 2006.

[Eck06] C. Eckert. *IT-Sicherheit*. Oldenbourg, 2006.

[FR10] J. Freud und B. Rücker. *Praxishandbuch BPMN 2.0*, Jgg. 2. Carl Hanser Verlag, 2010.

[Fra11] Fraunhofer IAO. *Business Process Management Tools 2011*. Fraunhofer-IRB-Verlag, 2011.

[ISO05] ISO. *ISO27001: Information Security Management System (ISMS) standard*, 2005.

[Jür04] J. Jürjens. *Secure Systems Development with UML*. Springer, 1. Auflage, 11 2004.

[KLM09] E. Knauss, D. Lubke und S. Meyer. *Feedback-driven requirements engineering: The Heuristic Requirements Assistant*. ICSE '09, 2009.

[LBD02] T. Lodderstedt, D. A. Basin und J. Doser. *SecureUML: A UML-Based Modeling Language for Model-Driven Security*. UML 2002. Springer-Verlag.

[OMG05] Object Management Group. *Unified Modeling Language Specification v2.0*, 2005.

[PHJB11] M. Peschke, M. Hirsch, J. Jürjens und S. Braun. *Werkzeuggestützte Identifikation von IT-Sicherheitsrisiken*. 2011.

[SKH+11] K. Schneider, E. Knauss, S. Houmb, S. Islam und J. Jürjens. *Enhancing security requirements engineering by organizational learning*. *Requirements Engineering*, 2011.

[Men09] Michael Menzel, Ivonne Thomas and Christoph Meinel. *Security Requirements Specification in Service-Oriented Business Process Management*. *ARES*, 2009.

[Wol08] Christian Wolter and Michael Menzel and Christoph Meinel. *Modelling Security Goals in Business Processes*. *Modellierung*, 2008.

[3] `http://carisma.umlsec.de/`