Security-Engineering in Software-Entwicklung und Betrieb

Armin Lunkeit

Rohde & Schwarz Cybersecurity
armin.lunkeit@rohde-schwarz.com

Zusammenfassung

Software-Engineering und Security-Engineering sind zwei zentrale Themengebiete der Softwareentwicklung, deren Verzahnung über den gesamten Software-Lebenszyklus für die Entwicklung und den Betrieb sicherer Software von Bedeutung ist. Die Integration zwischen Security-Engineering und des Software-Engineering ist eine der großen Herausforderungen, die es vor dem Hintergrund eines beständigen Kosten- und Zeitdrucks in Entwicklungsprojekten zu meistern gilt. Die oberflächliche oder mangelhafte Bearbeitung von Sicherheitsanforderungen führt in der Konsequenz zu inadäquaten Sicherheitsmaßnahmen, der späten Erkennung von Architektur- und Sicherheitsproblemen, unzureichend hinsichtlich seiner Sicherheitseigenschaften getesteter Software und damit einhergehend zu wirtschaftlichen Implikationen, die aus der späten Behebung dieser Problematiken resultieren [SlHK98], [DeSt00]. Die frühzeitige Identifikation und Integration potentieller und bestehender Sicherheitsanforderungen durch das Security- Requirements-Engineering ist daher ein wichtiger Teil der Anforderungsphase. Hinzu kommt der betriebliche Aspekt, der unter dem Gesichtspunkt der Integration des Security-Engineerings in das Software- Engineering bislang nur unzureichend betrachtet wurde. Während der Entwicklungs- und Testprozess aufgrund klar abgegrenzter Randbedingungen eine Momentaufnahme zur Sicherheit einer Software oder eines Systems generieren, können diese sich in der Betriebsphase verändern.

1 Stand der Technik

Die Unified Modeling Language (UML) ist eine von der Object Management Group (OMG) gepflegte Modellierungssprache [OMG15]. Das Kernkonzept der UML ist die Modellierung statischer Strukturen und dynamischen Verhaltens. Statische Strukturen werden mithilfe von Klassen und diskreten Objekten dargestellt und Beziehungen mit Vererbung und Zuordnung ausgedrückt. Klassen können statische Strukturen beliebiger Sachverhalte abbilden und haben ebenso wie Objekte Attribute. Dynamische Verhaltensbeschreibungen werden mit Zuständen und deren Übergängen, Sequenzen und Aktionen abgebildet. Es wurden verschiedene Erweiterungen der UML vorgeschlagen, um die Abbildung nicht-funktionaler Anforderungen in die UML zu integrieren ([CyPJ01], [Jürj05], [ChPJ09], [ZhGo07]). Eine ausschließliche Modellierung von IT-Sicherheitsanforderungen mithilfe der UML ist nur möglich, wenn es sich um funktionale Sicherheitsanforderungen handelt, sodass eine Nutzung in der frühen Phase der Anforderungsdefinition nur mit Einschränkungen möglich ist. Als Alternative wird in [MoJü06] die Integration zwischen UML und anderen Anforderungsmodellen vorgeschlagen.

Secure Tropos ist eine Erweiterung des Tropos-Frameworks und ist sowohl in der Anforderungs- und Architekturphase als auch in der Testphase nutzbar [MoGi07], [MaMZ07]. Es wird

P. Schartner · N. Pohlmann (Hrsg.) · D•A•CH Security 2018 · syssec (2018) 47-58.

eine grafische Notation genutzt, um Designziele eines Systems zu modellieren. Tropos [BePS05], [SPMG05] definiert unterschiedliche Stereotypen wie *Actor*, *Agent*, *Role* und *Position* und nimmt die Unterscheidung in *Hard Goal* und *Soft Goal* zur Kategorisierung der von einem Akteur verfolgten Ziele vor. Diese Kategorisierung ist mit funktionalen und nicht-funktionalen Anforderungen vergleichbar.

Die *CORAS*-Methode ist ein risikoorientiertes Vorgehen, mit dem auch nicht-technische Sachverhalte untersucht werden können. Es werden acht Schritte von der Vorbereitung für die Analyse über die Identifikation bis hin zur Behandlung vorhandener Risiken definiert. Akzeptable Risiken werden in der Analyse nicht weiterverfolgt, während nicht-akzeptable Risiken behandelt werden müssen. Dies kann die Eintrittswahrscheinlichkeit eines unerwünschten Ereignisses und dessen Auswirkungen entweder in Kombination oder einzeln behandeln, bis das Risiko wieder akzeptabel erscheint.

Zentraler Bestandteil der Methode ist die grafische Analyse. Die CORAS- Sprache zur Beschreibung, Dokumentation und Analyse von Bedrohungen und Risiken wurde ursprünglich als UML-Profil definiert und hat sich zu einer Domänen-spezifischen Sprache (DSL) weiterentwickelt. Es sind verschiedene Diagrammtypen definiert (Asset-, Threat-, Risk- und Treatment-Diagramme), mit deren Hilfe die Risikoanalyse erfolgt.

Die *Common Criteria* sind ein Framework zur Modellierung und Evaluierung der IT- Sicherheit von Produkten [CCM12a], [CCM12b], [CCM12c]. Funktionale Sicherheitsanforderungen bieten eine technologieneutrale Abstraktion an und überlassen die konkrete Umsetzung der Sicherheitsanforderungen dem Entwickler des Produktes. Ausgehend von der Definition des Sicherheitsproblems werden Bedrohungen, zu schützende Werte, Sicherheitspolitiken definiert. Sicherheitsanforderungen werden mithilfe von Security Functional Requirements modelliert; in Korrespondenzanalysen wird die Konsistenz der Definition des Sicherheitsproblems gezeigt. Die Sicherheitsanforderungen an Produkttypen werden in Schutzprofilen (Protection Profile) bzw. in einem Security Target definiert. Aktuelle Beispiele sind die Schutzprofile für das Smart Meter Gateway [BuSI14] und den Konnektor im Gesundheitswesen [BuSI16]. Vordergründig sind die Common Criteria ein Framework für die Evaluierung der Sicherheitseigenschaften eines Produktes. Die Ermittlung von Anforderungen und die Betriebsphase stehen nicht im Vordergrund des Frameworks und müssen durch andere Methoden adressiert werden.

2 Modellbasiertes Security-Engineering

2.1 Kernkonzepte in der Übersicht

Der vorgestellte Ansatz fasst die Konzepte aus [Lunk18] zusammen. Im Schwerpunkt ist die Arbeit auf zwei Teilaspekte des System- bzw. Software-Engineerings ausgerichtet. Der erste Teilaspekt ist auf die Phase der Anforderungsdefinition und die Auswahl geeigneter Sicherheitsmechanismen, der zweite Teilaspekt auf die Betriebsphase ausgerichtet.

Der erste Teilaspekt umfasst die Definition des Sicherheitsproblems und bedient sich dabei eines Metamodells zur Beschreibung der Modellelemente und ihrer Abhängigkeiten. Dieses Metamodell wird für eine tabellarische Analyse genutzt, um eine Aussage zur Eignung ausgewählter Sicherheitsmechanismen zu gewinnen. Dazu ergänzend wird eine grafische Darstellung definiert, mit deren Hilfe das Sicherheitsproblem bestehend aus zu schützenden Werten, deren Sicherheitszielen, Bedrohungen und Sicherheitsmechanismen abgebildet wird. Diese Konzepte

umfassen die Aspekte der Modellierung des Sicherheitsproblems, die Charakterisierung relevanter Bedrohungen und Angriffe und die Ableitung der Sicherheitsanforderungen.

Der zweite Teilaspekt befasst sich mit der Definition eines Metamodells für die betrieblichen Aspekte eines Systems bzw. einer Anwendung. Die Relevanz dieses Ansatzes ergibt sich aus der Fragestellung zu den Abhängigkeiten der Sicherheitsfunktionalität zur Systemumgebung und umfasst die Charakterisierung der Ausführungsumgebung, der Konfiguration und der Sicherheitspolitiken.

2.2 Relation zu bestehenden Ansätzen

Der vorgestellte Ansatz ist in Relation zu den Ansätzen von CORAS[1] und der Common Criteria zu sehen. Der präsentierte Ansatz konzentriert sich im Vergleich zu CORAS stärker auf die Darstellung des Sicherheitsproblems und stellt Sicherheitsziele, Bedrohungen und Sicherheitsmechanismen in den Zusammenhang.

Die Common Criteria sind ein Framework zur Modellierung von Sicherheitsanforderungen. Die Intention dieses Ansatzes ist jedoch die Evaluierung der Sicherheitsfunktionalität eines Produktes. Damit ist diese Methodik nicht ohne weiteres in der Produktentwicklung nutzbar. Diese Lücke wird vom vorgestellten Ansatz behandelt und bietet für die Entwicklungsorganisation eine vereinfachte Methodik.

2.3 Die Fallstudie

Als Fallstudie diente die Entwicklung eines Smart Meter Gateways (SMGW). Smart Meter Gateways verbinden intelligente Messsysteme (Energiezähler) mit dem Energieerzeuger zum Zwecke des Abrufs des Energieverbrauchs, Konfiguration der gebuchten Tarife und in der Endausbaustufe der Steuerung von Erzeugern (Prosumer-Modelle). Die Einführung intelligenter Messtechnik und damit der Smart Meter Gateways ist gemäß Energiewirtschaftsgesetz verpflichtend für Neubauten und Sanierung bei Verbräuchen von mehr als 6000 kW/h. Begleitend zur Gesetzgebung wurde vom Bundesamt für Sicherheit in der Informationstechnik (BSI) eine technische Richtlinie herausgegeben (TR-03109, [BuSI13]), zu der sich Implementierungen der Smart Meter Gateways konform verhalten müssen.

3 Definition des Sicherheitsproblems

Dem vorgestellten Ansatz liegt die Idee zu Grunde, dass Software-Entwickler in der Regel keine IT-Sicherheitsexperten sind. Davon unabhängig werden Methoden benötigt, die im Engineering-Prozess genutzt werden können und nachvollziehbare und transparente Ergebnisse liefern. Die Nutzbarkeit im Engineering steht dabei im Vordergrund. Im ersten Abschnitt werden die genutzten Grundbegriffe vorgestellt, im zweiten Abschnitt ein Metamodell zur Abklärung der Beziehungen zwischen den Modellelementen. Über die Charakterisierung des Angreifers und der Bedrohungen wird die Auswertung des Modells angedeutet. Abschließend wird die Ableitung funktionaler Sicherheitsanforderungen gezeigt.

[1] Insbesondere die grafische Analyse / Modellierung

3.1 Grundbegriffe

Die Ermittlung und Bearbeitung funktionaler Sicherheitsanforderungen benötigt eine Abstraktion des Sicherheitsproblems mithilfe eines Modells. Im Ergebnis der Modellierung sind zu schützende Werte, die geltenden Sicherheitsziele, Bedrohungen, Angriffe und Angreifer und deren jeweiligen Potentiale identifiziert und charakterisiert. Die Modellierung des Sicherheitsproblems erfolgt mithilfe zweier Perspektiven.

- Die sozio-technische Perspektive modelliert die zu schützenden Werte, geltende Sicherheitsziele und Bedrohungen aus Sicht der mit dem System interagierenden Akteure. Das System wird als Black Box betrachtet.
- Die technische Perspektive modelliert die zu schützenden Werte, geltenden Sicherheitsziele und Bedrohungen aus der Systemsicht. Interne Werte, deren Sicherheitsziele und geltenden Bedrohungen werden charakterisiert und in Zusammenhang mit den Zielen externer Akteure gebracht.

Beide Perspektiven sind Stakeholder-Perspektiven. Die sozio-technische Perspektive modelliert die Sicht des Benutzers, d. h. der Benutzer hat ein Interesse an der Realisierung der Anforderungen. Die technische Perspektive geht auf die technische Umsetzung des Systems ein und liefert die Abbildung zum Systemdesign und zur Implementierung. In diesem Falle sind die Projektbeteiligten – speziell das Projektteam – als Stakeholder anzusehen. Die Nutzung des Modells der technischen Perspektive sichert die Erfüllung der in der sozio-technischen Perspektive identifizierten Sicherheitsziele.

Die sozio-technische Perspektive untersucht die Interaktion externer Akteure gegenüber dem technischen System und beschreibt Werte, Sicherheitsziele und Bedrohungen. Externe Akteure identifizieren andere zu schützende Werte und Sicherheitsziele als dies bei einer ausschließlich technisch orientierten Sicht auf das System der Fall ist (technische Perspektive). Der Vorteil der Nutzung der beiden Perspektiven liegt in der umfassenden Analyse aller Teilaspekte des Sicherheitsproblems. Die technische Perspektive greift die Ziele der externen Akteure und der von ihnen umgesetzten Rollen auf. Sie stellt diejenigen Werte und Sicherheitsziele in den Mittelpunkt, die Voraussetzungen zur Erreichung der Ziele externer Akteure sind. Diese Werte und Sicherheitsziele und die damit kor- respondierenden Bedrohungen sind aufgrund der Black-Box-Sicht der sozio-technischen Perspektive des externen Akteurs für diesen irrelevant.

Ein Akteur kann mehrere Rollen einnehmen, ebenso kann eine Rolle von mehreren Akteuren wahrgenommen werden. Die Rolle hat ein Interesse an einem zu schützenden Wert (*Asset*). Die Rolle definiert zudem das Sicherheitsziel (*Security Objective*) für den zu schützenden Wert. Das Sicherheitsziel schützt das Asset. Diese Beziehungen gelten für beide Perspektiven gleichermaßen.

Sicherheitsziele werden mithilfe der Begriffe Authentizität, Integrität, Vertraulichkeit und Verfügbarkeit definiert, sind jedoch nicht auf diese beschränkt. Die Definition weiterer Sicherheitsziele zur Beschreibung des Sicherheitsproblems erfolgt textuell.

Ein zwischen den Perspektiven übergreifendes Konzept ist das Szenario. Szenarien werden verwendet, um den Kontext zu schützender Werte abzugrenzen. Nicht in jedem Szenario sind sämtliche zu schützende Werte beteiligt. Ebenso können die Sicherheitsziele für einen zu schützenden Wert zwischen den Szenarien variieren.

Zur vollständigen Beschreibung des Sicherheitsproblems in den jeweiligen Perspektiven werden Bedrohungen definiert. Bedrohungen subsummieren Angriffe, die gegen einen zu schützenden Wert gerichtet sind. Bedrohungen werden abstrahierter beschrieben und gehen nicht auf die technische Umsetzung ein. Die anfängliche Beschreibung des Sicherheitsproblems nutzt diesen Abstraktionsgrad. Zudem wird der Begriff des Sicherheitsmechanismus eingeführt. Sicherheitsmechanismen werden einzeln oder in Kombination zum Schutz eines Assets gegen Bedrohungen und die zugeordneten Angriffe eingesetzt.

Zur Definition und für die nachfolgende Analyse des zu lösenden Sicherheitsproblems bietet die tabellarische Aufbereitung einen strukturierten Ansatz. Neu eingeführt werden Angriffspotential, Widerstandsfähigkeit und Schutzniveau. Das Angriffspotential charakterisiert die Fähigkeiten eines Angreifers qualitativ. Der Angreifer entwickelt eine oder mehrere Bedrohungen, die mithilfe eines oder mehrerer Angriffe umgesetzt werden. Die Umsetzung eines Angriffs korrespondiert mit den im Angriffspotential charakterisierten qualitativen Fähigkeiten des Angreifers. Diese Charakterisierung ist eine wesentliche Veränderung zu abstrahierten Bedrohungen, da die dem Angreifer zugestandenen Fähigkeiten Eingang in die Analyse finden. Die Widerstandsfähigkeit charakterisiert die Wirksamkeit eines Sicherheitsmechanismus. Beide stehen in einem 1:1 Verhältnis, wobei der Sicherheitsmechanismus einer Bedrohung entgegenwirkt. Die Widerstandsfähigkeit trägt zum Schutzniveau bei, welches einem Angriff im Kontext des zugeordneten Angriffspotentials entgegenwirkt und einen Wert schützt. Der Sicherheitsmechanismus leistet einen Beitrag zur Erreichung eines Sicherheitsziels und stellt so Sicherheitsmechanismen und Sicherheitsziele in einen Kontext.

3.2 Metamodell des Sicherheitsproblems

Um die Wirksamkeit von Sicherheitsmechanismen gegenüber Bedrohungen und Angriffen in der Anforderungs- und Architekturphase zu bewerten, müssen die Verhältnisse innerhalb des Sicherheitsproblems abgebildet werden. Das Sicherheitsproblem stellt Anforderungen, Sicherheitsziele, zu schützende Werte und Sicherheitsmechanismen in einen Kontext mit Bedrohungen, Angriffen und ausgenutzten Schwachstellen. Neben der Ermittlung der Beziehungen zwischen diesen Elementen ist die Identifikation von Potentialen von elementarer Bedeutung. Sicherheitsprobleme bestehen vor dem Hintergrund von Angreifern, die über Fähigkeiten und Expertise zur Ausführung von Angriffen verfügen. Um einen Wert zu schützen, muss das aus den ergriffenen Sicherheitsmaßnahmen erzielte Schutzniveau gleich oder größer dem Potential des einwirkenden Angriffs sein. Zur Abbildung dieser Zusammenhänge werden die Modellelemente Angriffspotential (*Attack Potential*), Widerstandsfähigkeit (*Resistance*) und Schutzniveau (*Protection Level*) eingeführt. Mit dieser Grundlage modelliert das Sicherheitsproblem den Bezug zu den Anforderungen, die zu dessen Lösung erhoben werden.

Das zentrale Element des Modells ist der zu schützende Wert (*Asset*). Zu schützende Werte können Ressourcen und Daten sein. Der zu schützende Wert steht im Kontext der zugeordneten Sicherheitsziele. Aus einem Sicherheitsziel ergeben sich Anforderungen, die funktionaler oder nicht-funktionaler Natur sein können. Dieser Ansatz stellt die Verbindung zwischen zu schützenden Werten, gestellten Sicherheitszielen und Anforderungen her.

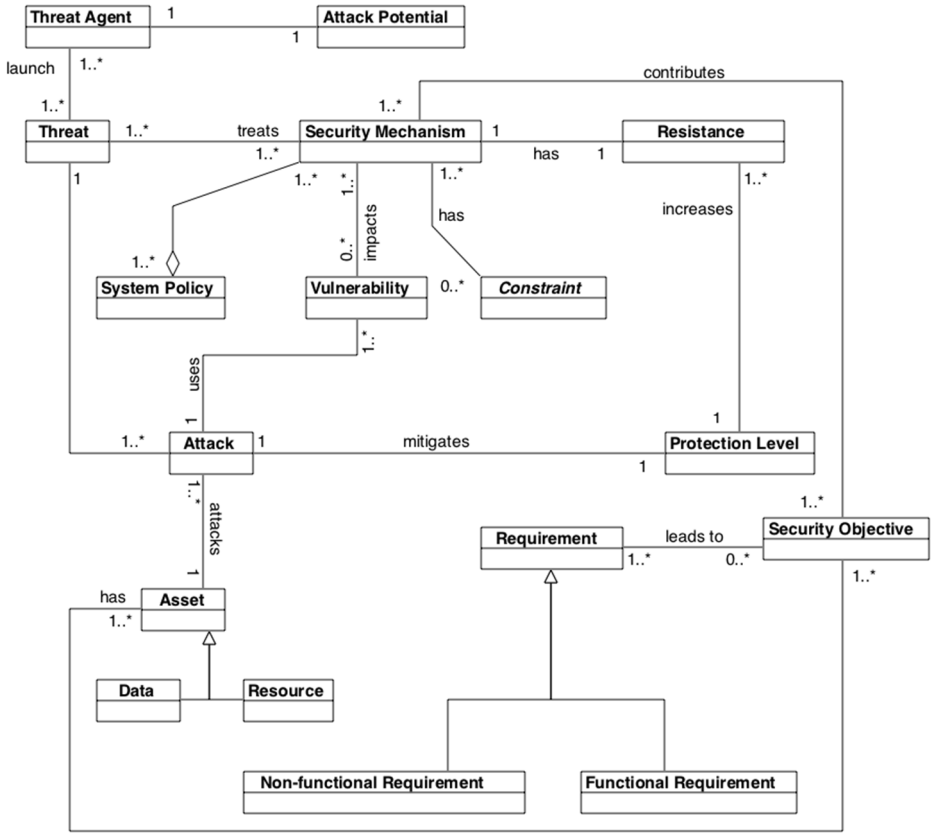

Abb. 1: Metamodell zur Modellierung des Sicherheitsproblems

3.3 Auswertung des Sicherheitsproblems

Auf Grundlage der vorangegangenen Schritte erfolgen Analyse und Auswertung des Sicherheitsproblems. Es wird ermittelt, ob die ausgewählten Sicherheitsmechanismen die identifizierten zu schützenden Werte gegen Bedrohungen und Angriffe adäquat schützen. Zeigt die Analyse, dass alle Bedrohungen und Angriffe in ausreichender Art und Weise behandelt werden, gehen die abgeleiteten Anforderungen in das Software-Engineering ein. Sind die Sicherheitsmechanismen nicht ausreichend, muss entschieden werden, ob die Bedrohungen und Angriffe als akzeptable Risiken angesehen werden oder ob Veränderungen an Sicherheitsmechanismen erforderlich sind.

Diese Entscheidungen werden auf Grundlage einer quantitativen Bewertung gefällt. Dies erfolgt mithilfe einer tabellarischen Analyse. Das Modell berücksichtigt die quantitative Bewertung mithilfe der Elemente Widerstandsfähigkeit (*Resistance*) und Schutzniveau (*Protection Level*). Das erste Element beschreibt individuell für einen Sicherheitsmechanismus dessen Fähigkeiten zur Mitigation eines Angriffs. Das Schutzniveau subsummiert die Widerstandsfähigkeit mehrerer Sicherheitsmechanismen, da die Kombination mehrerer schwächerer Sicherheitsmechanismen geeignet sein kann, einen Angriff abzuwehren.

Neben den Sicherheitsmechanismen sind die einzelnen Angriffe zu bewerten. Aus der Charakterisierung des Angreifers geht das maximale Potential eines Angriffs hervor. Sämtliche Angriffe innerhalb dieses Bereichs sind für einen solchen Angreifer durchführbar. Der Modellansatz erlaubt die Definition mehrerer Angreifer mit verschiedenen Angriffspotentialen. In der Analyse wird der nicht akzeptable Angreifer mit dem höchsten Potential zugrunde gelegt, wenn

die Charakterisierung des Angreifers keine Unterschiede in den zugestandenen Möglichkeiten aufzeigt.

3.4 Ableitung funktionaler Sicherheitsanforderungen

Zur Ableitung funktionaler Sicherheitsanforderungen für das Software-Engineering werden die in der sozio-technischen und technischen Perspektive ermittelten Informationen genutzt. Diese Anforderungsableitung transformiert die vom Security-Engineering definierten Sicherheitsmechanismen in eine für das Software-Engineering nutzbare Beschreibungen, die zu einer Architektur und Implementierung führen. Da die Sicherheitsmechanismen Bedrohungen entgegenwirken, die ihrerseits konkrete Angriffe subsummieren, fließen Maßnahmen gegen bekannte Angriffe in die Formulierung der Sicherheitsanforderungen.

Eine funktionale Sicherheitsanforderung spezifiziert technische Maßnahmen zur Erreichung eines Sicherheitsziels. In diesem Zusammenhang werden die Widerstandsfähigkeit des einzelnen gewählten Mechanismus und das erreichte Schutzniveau gegenüber den zu einer Bedrohung zählenden Angriffen ausgewertet. Die gewählten Mechanismen sind hinreichend und der Angriff ist abgewehrt, wenn die Kombination der genutzten Mechanismen ein Schutzniveau generiert, welches über dem erreichten Angriffspotential liegt. Dieses Vor- gehen wird sowohl während der Analyse der sozio-technischen und technischen Perspektive als auch in der Anforderungsableitung und Designphase genutzt.

4 Betrieb sicherer Anwendungen

Die Betriebsphase nutzt Methoden zur Überwachung der IT-Sicherheit. Voraussetzung dafür sind die Kenntnisse der Anforderungen an die Betriebsumgebung, der Schnittstellen und der sicheren Konfiguration. Kein den Bedrohungen zugeordneter Angriff darf erfolgreich umsetzbar sein. In der Anforderungs- und Designphase ist dies eine Momentaufnahme. In der Betriebsphase kann eine solche Sicherheitsaussage nur dann aufrechterhalten werden, wenn alle im Modell erfassten Randbedingungen weiterhin gültig sind. Eine Momentaufnahme kann nicht berücksichtigen, dass über den Zeitraum der Betriebsphase zu bereits identifizierten Bedrohungen neue Angriffe gefunden oder gänzlich neue Bedrohungen erkannt werden. Der Betrieb sicherer IT-Systeme verlangt die permanente Bewertung der Situation und angemessene Reaktionen. Kern des nachfolgenden Abschnitts ist ein Modell zur Erfassung und Beschreibung der für Sicherheitsmechanismen geltenden betrieblichen Randbedingungen. Deren Kenntnis und das Verständnis ihrer Natur sind die Voraussetzung für den Betrieb sicherer Systeme.

Die Anlässe für einen Wechsel von Komponenten in der Ausführungsumgebung sind vielfältig und werden in dieser Arbeit auch nicht eingehender untersucht. Gründe hierfür sind neue Funktionen, behobene Sicherheitslücken oder Produktabkündigungen. Ebenso ist die Installation zusätzlicher Software zu berücksichtigen. Die Ausführungsumgebung wird in der Modellierung des Sicherheitsproblems Annahmen-basiert abgebildet. Dies hat zur Folge, dass Teile des als sicher angenommenen Systems eine direkte oder indirekte Abhängigkeit von Annahmen aufweisen. Weichen Annahme und Realität voneinander ab, steht die Wirksamkeit des verwendeten Sicherheitsmechanismus in Frage. Die Randbedingungen und deren Einhaltung sind für die erreichte Sicherheit im Betrieb ursächlich. Dazu zählen die Konfiguration des Systems, die Ausführungsumgebung sowie organisatorische Maßnahmen. Weicht eine der Randbedingungen von den zulässigen Werten ab, kann die Behauptung eines sicheren Systems nicht mehr aufrechterhalten werden.

4.1 Metamodell

Die Einhaltung definierter Randbedingungen bestimmt, ob ein System in der Betriebsphase als sicher gelten kann. Als Randbedingungen werden die Konfiguration des sicheren Systems und seiner Ausführungsumgebung, die Ausführungsumgebung selbst sowie externe Sicherheitspolitiken angesehen. Dabei handelt es sich um solche Randbedingungen, welche die Funktionalität eines Sicherheitsmechanismus in direkter oder indirekter Art und Weise beeinflussen.

- *Konfiguration:* Sichere Systeme und auch die Ausführungsumgebung können eine Vielzahl an Konfigurationsparametern mit Einfluss auf die IT-Sicherheit aufweisen.
- *Ausführungsumgebung:* Dies sind Annahmen zum Verhalten, zur Konfiguration und zu den Schnittstellen der Systemumgebung. Die Ausführungsumgebung kann auf Software basieren, Hard- und Software umfassen oder auch abstrakt gehalten sein.
- *Externe Sicherheitspolitiken:* Dies sind Sicherheitspolitiken, die nicht vom untersuchten System umgesetzt werden und einen Einfluss auf die Sicherheitsmechanismen haben.

Das Metamodell setzt Sicherheitsmechanismus und Randbedingungen in eine Beziehung und zeigt so die bestehende Abhängigkeit.

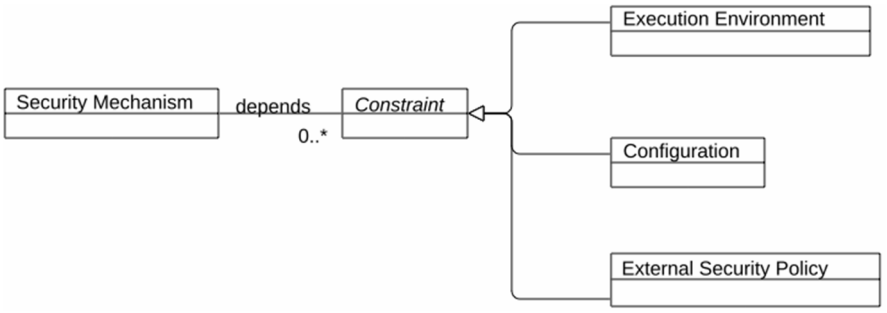

Abb. 2: Betriebliche Randbedingungen

4.2 Charakterisierung der Ausführungsumgebung

Die Ausführungsumgebung ist eine Komposition externer Entitäten (*External Entity*), die selbst wiederum Aggregationen aus Konfiguration (*Configuration*), umgesetzten Sicherheitspolitiken (*External Security Policy*), Attributen (*Attribute*) und Schnittstellen (*Interface*) sind. Die Einführung externer Entitäten trägt dem Umstand Rechnung, dass die Ausführungsumgebung selbst modular sein kann. Schnittstellen können physischer oder logischer Natur sein und werden hinsichtlich der Art ihrer Interaktion mit dem sicheren System unterschieden. Die Charakterisierung der Schnittstellen in dieser Hinsicht ist erforderlich, um die Mächtigkeit und die Auswirkungen von Änderungen zu beurteilen. Grundsätzlich beeinflusst die Interaktion an den Schnittstellen das Verhalten eines Systems und Angriffe können diese nutzen. Dieser Umstand wird mithilfe der Kritikalität (*Criticality*) charakterisiert, die als Attribut einer jeden Schnittstelle zugeordnet wird. Die Kritikalität wird in zwei Typen unterschieden. Als *Interfering* werden Schnittstellen bezeichnet, deren Verhalten die Funktion eines Sicherheitsmechanismus beeinflusst. Schnittstellen, die keinen Einfluss auf die Funktion eines Sicherheitsmechanismus haben, werden als *Non-Interfering* markiert.

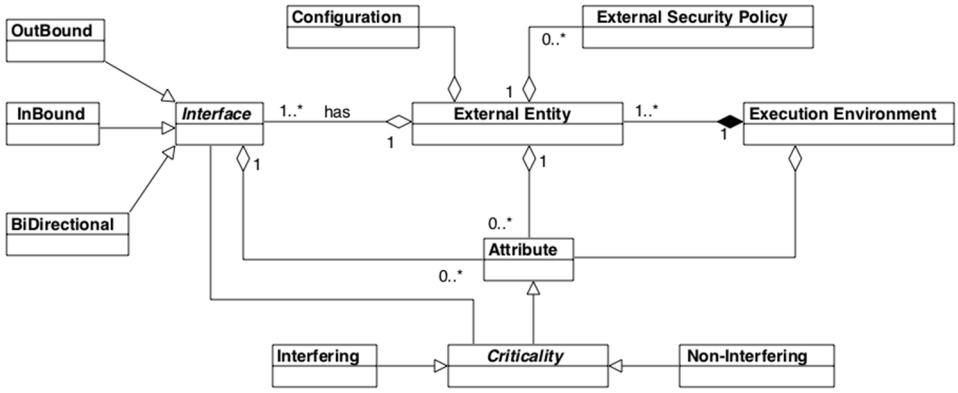

Abb. 3: Charakterisierung der Ausführungsumgebung

4.3 Charakterisierung der Konfiguration

Anpassungen an der Konfiguration eines Systems und der Ausführungsumgebung können Änderungen am Verhalten einzelner Funktionen oder des gesamten Systems nach sich ziehen. Es entsteht das Risiko einer unsicheren Konfiguration und der Umgehung von Sicherheitsmechanismen.

- Eine Konfiguration ist eine Komposition aus Konfigurationsoptionen (*Configuration Item*).
- Jeder Konfigurationsoption ist eine Menge möglicher Werte zugeordnet (*Choice*). Für jede Option wird eine konkrete Auswahl (*Selection*) getroffen.
- Einer Konfigurationsoption können weitere Attribute (*Attribute*) zugeordnet werden.

5 Aspekte der Fallstudie

5.1 Aspekte des Sicherheitsproblems

Auf Grundlage von [BuSI13] und [BuSI14] wurden die zu schützenden Werte ermittelt und den jeweiligen Perspektiven zugeordnet.

Tab. 1: Zuweisung Assets und Perspektiven (exemplarischer Auszug)

Asset	Sozio-Technisch	Technisch
Kommunikationsdaten der Messeinrichtung		x
Verbrauchsdaten	x	
Kommunikationsdaten CLS		x
Identität Verbraucher	x	
Authentisierungsdaten Service-Techniker	x	

Ebenso wurden die Sicherheitsziele für die jeweiligen zu schützenden Werte definiert. Als ein Beispiel wurden für die Authentisierungsdaten des Service-Technikers Integrität, Vertraulichkeit und Authentizität in der sozio-technischen Perspektive und Integrität und Authentizität in der technischen Perspektive definiert. Für die weitere Analyse und Modellierung des Sicherheitsproblems wurden die Szenarien Systemstart, Betrieb, Datenübertragung, Datenspeicherung und Power-Down gewählt.

Für das SMGW sind mehrere Angreifertypen zu definieren:

- Angreifer ohne physischen Zugriff auf das SMGW ($A_{Entfernt}$)
- Angreifer mit physischem Zugriff auf das SMGW ($A_{Physisch}$)

Für beide Angreifertypen müssen die Zielsetzungen durchgeführter Angriffe berücksichtigt werden. Dabei wird deutlich, dass sowohl zwischen Angriffsart, d. h. Angriffen mit und ohne physischen Zugriff auf das SMGW, als auch zwischen Motivationsprofilen zu unterscheiden ist. Es sind unterschiedliche Motivationen vorhanden; die primär wirtschaftlich getriebene Motivation, technisches Interesse/Geltungsdrang und weitere (kriminelle) Motive. Die modellierten Angreifer unterscheiden sich erheblich von denen des Schutzprofils [BuSI14] und der technischen Richtlinie [BuSI13] und legen ein insgesamt höheres Potential der Angreifer insbesondere bei der Durchführung der Angriffe nahe als in [BuSI14] angegeben.

Im Ergebnis führte die Nutzung eines Modells des Sicherheitsproblems zur Definition von Sicherheitsmechanismen und Sicherheitsanforderungen, wie in Tabelle 2 gezeigt.

Tab. 2: Sicherheitsmechanismen und Sicherheitsanforderungen

Sicherheits-mechanismus	Sicherheitsanforderung
Separation der Sicherheitsdomänen	Separation der Sicherheitsdomänen mithilfe von SELinux. Sicherheitsdomänen werden entlang der logischen Kommunikation (HAN, WAN, LMN, CLS) definiert.
Schutz zur Laufzeit	Nutzung von ASLR (address space layout randomization) und SSP (stack smash protection) um Auswirkungen erfolgreicher Angriffe zu minimieren
Minimales Deployment	Ausschließliches Deployment von tatsächlich genutzten Bibliotheken
Defensive Konfiguration	Verwendung einer sicheren Konfiguration, Deaktivierung experimenteller Features

5.2 Betriebliche Aspekte

Das SMGW unterhält Schnittstellen zu drei verschiedenen Entitäten im WAN. Diese sind der Gateway-Administrator (GWA), externe Marktteilnehmer (beispielsweise die Abrechnungsstelle) und ein NTP-Dienst zur Zeitsynchronisation. Die Schnittstellen ins WAN nutzen TCP/IP zur Übertragung und TLS zur Sicherung der Kommunikation.

Die Schnittstellen ins WAN sind bidirektional, Daten werden in beide Richtungen ausgetauscht. Die Kommunikation mit den EMTs beeinflusst die Wirksamkeit der Sicherheitsmechanismen nicht. Es werden Daten ausgetauscht und ggf. auch interpretiert. Änderungen an den Schnittstellen können lediglich dazu führen, dass die gewünschte Funktion nicht mehr zur Verfügung steht.

Bei Verwendung der Funktion zur Zeitsynchronisation besteht eine unmittelbare Abhängigkeit einzelner Sicherheitsmechanismen des SMGWs zur externen Entität NTP- Server. Die Tarifierung erfasster Verbrauchsdaten ist von der Korrektheit der im SMGW genutzten Uhrzeit abhängig. Diese wird über die Schnittstelle zum NTP-Server importiert. Ebenso arbeiten Log- und Protokolleinträge mit Zeitstempeln, die aus der im System gesetzten Uhrzeit gewonnen werden. Über die Schnittstelle zum NTP-Dienst ist das Verhalten der Sicherheitsmechanismen im SMGW beeinflussbar.

6 Resümee

Die durchgehende Nutzung des sicherheitsorientierten Modellansatzes bietet erhebliche Vorteile im Software-Lebenszyklus. Die Identifikation aller für die Entwicklung und den Betrieb bedeutsamen Sicherheitsanforderungen und deren Vereinigung in einem einzigen Modell ist die Grundlage für das Security-Engineering. Die vom zu schützenden Wert ausgehende grafische Analyse integriert Szenarien, Sicherheitsziele, Bedrohungen und Sicherheitsmechanismen in einem Kontextdiagramm. Gemeinsam mit der tabellarischen Analyse bietet diese Form der Problemdarstellung eine gemeinsame Plattform für alle am Entwicklungsprozess beteiligten Parteien.

Die betrieblichen Aspekte der IT-Sicherheit finden bereits im Designprozess des Produktes Berücksichtigung. Die Kenntnisse der Einflussnahme der Ausführungsumgebung, der Konfigurationen und externer Sicherheitspolitiken auf das System sind die Grundlage für den späteren sicheren Betrieb.

Literatur

[BePS05] D. Bertolini, A. Perini, A. Susi, H. Mouratidis: The TROPOS Visual Modeling Language: A MOF 1.4 Compliant Metamodel. In: Agent-oriented software engineering technical forum. Ljubljana, Slovenia, 2005.

[BuSI14] Bundesamt für Sicherheit in der Informationstechnik (BSI): Schutzprofil für die Kommunikationseinheit eines intelligenten Messsystems für Stoff- und Energiemengen Version 1.3. 2014.

[BuSI16] Bundesamt für Sicherheit in der Informationstechnik (BSI): Schutzprofil 1: Anforderungen an den Netzkonnektor (NK-PP) /BSI-PP-CC-0047, Version 3.2.2, 2016.

[CCM12a] Common Criteria Management Board (CCMB: Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and general model, CCMB-2012-09-001, Version 3.1 Revision 4, 2012.

[CCM12b] Common Criteria Management Board (CCMB): Common Criteria for Information Technology Security Evaluation, Part 2: Security functional components, CCMB-2012-09-002, Version 3.1 Revision 4, 2012.

[CCM12c] Common Criteria Management Board (CCMB): Common Criteria for Information Technology Security Evaluation, Part 3: Security assurance components, CCMB-2012-09-003, Version 3.1 Revision 4, 2012.

[ChPJ09] L. Chung, L. Prado, C. S. Julio: On non-functional requirements in software engineering. In: Conceptual modeling: Foundations and applications. Springer (2009) 363-379.

[CyPJ01] L. M. Cysneiros, L. Prado, C. S. Julio: Using UML to reflect non-functional requirements. In: Proceedings of the 2001 conference of the Centre for Advanced Studies on Collaborative research IBM Press (2001) 2.

[DeSt00] P. T. Devanbu, S. Stubblebine: Software Engineering for Security: A Roadmap. In: Proceedings of the Conference on The Future of Software Engineering. ACM ICSE ’00 (2000) 227-239.

[Jürj05] J. Jürjens: Secure Systems Development with UML. Springer (2005).

[Lunk18] A. Lunkeit: Modellbasiertes Security-Engineering in der Softwareentwicklung, PhD Thesis, TU Berlin (2018).

[MaMZ07] F. Massacci, J. Mylopoulos, N. Zannone: Computer-aided support for secure tropos. In: Automated Software Engineering 14, Nr. 3 (2007) 341-364.

[MoGi07] H. Mouratidis, P. Giorgini: Secure Tropos: a security-oriented extension of the Tropos methodology. In: International Journal of Software Engineering and Knowledge Engineering 17, Nr. 02 (2007) 285-309.

[MoJü06] H. Mouratidis, J. Jürjens, J. Fox: Towards a comprehensive framework for secure systems development. In: International Conference on Advanced Information Systems Engineering, Springer (2006) 48-62.

[OMG15] Object Management Group: OMG Unified Modeling Language (OMG UML) Version 2.5, OMG Document Number formal/2015-03.01. Object Management Group, 2015 http://www.omg.org/spec/UML/2.5

[SlHK98] S. A. Slaughter, D. E. Harter, M. S. Krishnan: Evaluating the cost of software quality. In: Communications of the ACM 41, Nr. 8 (1998) 67-73.

[SPMG05] A. Susi, A. Perini, J. Mylopoulos, P. Gi: The Tropos metamodel and its use. In: Informatica 29, Nr. 4 (2005) 401–408.

[ZhGo07] UML profiles for design decisions and non-functional requirements. In: Proceedings of the Second Workshop on Sharing and Reusing Architectural Knowledge Architecture, Rationale, and Design intent IEEE Computer Society (2007) 8.