NAMUR	**Ein Informationsmodell für das Automation Security Engineering**	**NE 193**

Vorbemerkung

Bei den NAMUR-Empfehlungen (NE) und-Arbeitsblättern (NA) handelt es sich um Erfahrungsberichte und Arbeitsunterlagen, die die NAMUR-Mitglieder erarbeitet haben.

NAMUR übernimmt keine Gewähr für die Vollständigkeit oder Richtigkeit der NE und NA. Jede Verwendung durch Mitglieder oder sonstige Dritte erfolgt in eigener Verantwortung und auf das eigene Risiko des Verwenders. Schadensersatzansprüche sind ausgeschlossen, soweit diese nicht auf zwingenden gesetzlichen Haftungsvorschriften beruhen. Einzelheiten regeln die Satzung und die Vereinsordnung oder die zwischen NAMUR und Dritten getroffene Vereinbarung.

NE und NA haben nicht den Grad des Konsenses von technischen Normen (z. B. DIN-Normen) oder Richtlinien (z. B. VDI-Richtlinien). Sie stellen lediglich Empfehlungen der NAMUR dar.

Nicht deutsche Ausgaben sind eine Übersetzung. Im Zweifelsfall ist der deutsche Originaltext anzuwenden.

Frühere Ausgaben

(Dies ist die Erstausgabe)

Änderungen

-

Erstellt durch

NAMUR-Arbeitskreis AK 1.3 Informationsmanagement und Werkzeuge

Beteiligte Mitglieder

Dr.-Ing. Andreas Schüller, YNCORIS (Arbeitskreisleiter)

Sarah Fluchs, admeritia

Björn Höper, LTsoft

Thomas Reuter, TGE Marine Engineering

Als externe Experten waren die folgenden Gäste des AK an der Erarbeitung der NE/NA beteiligt:

Emre Taştan, Hochschule Pforzheim

Marco Ehrlich, Technische Hochschule Ostwestfalen-Lippe

Herausgabe erfolgt durch die NAMUR-Geschäftsstelle

NAMUR - Interessengemeinschaft Automatisierungstechnik der Prozessindustrie e.V.
c/o Bayer AG
Gebäude K 9
51368 Leverkusen
Deutschland

Telefon:+49 214/30-71034
E-Mail: office@namur.de
Internet:www.namur.net

Inhaltsverzeichnis

1. Einleitung

Diese NAMUR-Empfehlung definiert ein UML-Informationsmodell für das Security Engineering von Automatisierungssystemen (Automation Security Engineering), also das Analysieren von Security-Problemen, das Treffen von Security-Entscheidungen und das Entwickeln von Security-Lösungen für Automatisierungssysteme.

Das Informationsmodell beinhaltet die für das Automation Security Engineering notwendigen und während des Security Engineering entstehenden Informationen.

2. Ziel

Das Informationsmodell für das Automation Security Engineering hat die folgenden Anwendungsfälle [1]:

- **Informationsaustausch zwischen security-relevanten Planungswerkzeugen:** Security-relevante Informationen gibt es in vielen verschiedenen Softwarewerkzeugen bzw. Dateien: in IT-Administrationstools, wie Asset-Inventare, Konfigurationsmanagement- oder Versionierungstools, in dedizierten Security-Tools wie Anomalieerkennungs- oder Intrusion-Detection-Systemen, aber auch in Engineering-Werkzeugen, die Risikobetrachtungen oder architekturelle Entscheidungen enthalten. Es ist unwahrscheinlich, dass diese verschiedenen Werkzeuge ihre Datenformate in absehbarer Zeit harmonisieren werden, weshalb ein neutrales Informationsmodell zum Austausch der security-relevanten Informationen die pragmatischere Lösung zu sein scheint.
- **Modellbasiertes Security-Engineering und Security by Design:** Ein Informationsmodell ist die Basis, um modellbasiertes Security Engineering zu ermöglichen und flexible Visualisierungen des zu schützenden Systems, seiner Security-Probleme bzw. der Security-Lösungen zu erzeugen. Ein Informationsmodell für das Security-Engineering hilft auch dabei, Security möglichst früh in den Automation-Engineering-Workflow zu integrieren (Security by Design) – so kann man schon Security-Entscheidungen treffen, auch wenn das Detail Engineering (vgl. NA 35) noch nicht abgeschlossen ist.
- **Treffen von Security-Entscheidungen während des Betriebs:** Security-Entscheidungen wie das Patchen einer Schwachstelle oder das Anwenden einer Alternativmaßnahme für Schwachstellen, für die kein Patch verfügbar ist, erfordern jedoch Kontextinformationen aus typischerweise verschiedenen Quellen. Relevant sind zum Beispiel der Schweregrad der Schwachstelle, die bestehenden Risiken und frühere Vorfälle für eine Komponente, die Kritikalität des Versagens oder der Manipulation der Komponente, die Netzwerkexposition der Komponente und die Kritikalität der daran angeschlossenen Komponenten. Diese Informationen sind jedoch wahrscheinlich an verschiedenen Orten gespeichert und müssten zeitaufwändig gesammelt und verarbeitet werden - sofern sie überhaupt verfügbar sind. Ein Informationsmodell hilft, alle Security-relevanten Engineering-Informationen auch in der Betriebsphase noch verfügbar zu haben.
- **Verwaltung von Standardkonfigurationen:** Effizienzgewinne im Betrieb von Security-Lösungen ergeben sich oft aus der Standardisierung von Komponenten und ihren Konfigurationen. Diese Standards müssen maschinenverarbeitbar gespeichert, gepflegt und verwaltet werden. Selten deckt ein Tool alle relevanten Informationen ab. Solche Standardkonfigurationen über viele Tools verteilt zu speichern und zu pflegen ist jedoch fehleranfällig und ineffizient.

Für all diese Anwendungsfälle besteht der Wert des Informationsmodells in der Einigung auf ein Modell, das alle Beteiligten nutzen. Das Ziel der NAMUR-Empfehlung ist, einen Vorschlag für solch ein konsensfähiges Modell zu machen.

3. Anwendungsbereich

Das in dieser NAMUR-Empfehlung beschriebene Informationsmodell ist für das Automation Security Engineering in der Design- und Betriebsphase eines Automatisierungssystems nutzbar. Es kann von Herstellern, Integratoren und Betreibern gleichermaßen verwendet werden und ist branchen- und standortunabhängig.

Der Anwendungsbereich des Informationsmodells ist die Dokumentation der Informationen, die beim Security-Engineering eines Automatisierungssystems verwendet und / oder erzeugt werden. Es dokumentiert also die Security von Automatisierungssystemen. Die Security der Informationen im Informationsmodell wird in dieser NAMUR-Empfehlung *nicht* betrachtet.

4. Modellierungsgrundsätze

Dieses Kapitel fasst grundlegende konzeptionelle Überlegungen zum Informationsmodell für das Automation Security Engineering zusammen.

4.1. Implementierungsunabhängigkeit

Als Modellierungssprache wird UML verwendet, um das Modell implementierungsunabhängig zu halten.

4.2. Methodenneutralität

Für das Security-Engineering im Allgemeinen und die Security-Risikoanalyse im Speziellen gibt es unterschiedliche Philosophien und Methoden. Das vorliegende Modell ist keine Abbildung einer spezifischen Methode. Es ist vielmehr so generisch gehalten, dass es für jede Methode anwendbar ist.

Aus diesem Grund beinhaltet das Modell möglicherweise mehr Klassen und Attribute als notwendig für die Anwendung einer ausgewählten Methode. Klassen oder Attribute, die für eine bestimmte Methode nicht relevant sind, können bei der Anwendung des Modells ignoriert werden. Gleichermaßen ist das Modell erweiterbar, wenn zusätzliche spezifische Anforderungen oder Methoden hinzugefügt werden müssen. Es wurde so konzipiert, dass neue Klassen, Attribute oder Beziehungen in den Bereichen Anlage, Risiken sowie Security eingeführt werden können.

4.3. Anwendung des Modells

Es sind verschiedene Anwendungsformen des Modells denkbar (mit aufsteigendem Abstraktionsgrad):

- **Konkretes Projekt:** Das Modell wird zur Dokumentation der Ergebnisse eines konkreten Security-Engineering-Projektes verwendet. Dafür werden die Klassen des Modells instanziiert, sodass sie einen Teil des konkreten Projekts abbilden: Mit ihnen werden die tatsächliche Anlage, die diese Anlage tatsächlich betreffenden Risiken sowie die tatsächlich identifizierten Anforderungen modelliert.
- **Typenbibliotheken:** Das Modell wird für das Anlegen von Bibliotheken verwendet, die für konkrete Projekte herangezogen werden können. Dafür werden ebenfalls Klassen des Modells instanziiert – allerdings auf Typenebene (z.B. eine Steuerung Siemens S7-1200). Bei der Modellierung eines konkreten Projektes werden diese Bibliothekselemente dann für ein konkretes Element instanziiert (z.B. die Steuerung Siemens S7-1200 mit Seriennummer ABC123 und der IP 192.168.2.13).
- **Abstraktes Modell für eine spezifische Methode:** Die Klassen des Modells werden nicht instanziiert, sondern die Klassen selbst werden modifiziert, um das Modell passender für eine bestimmte Methode oder ein bestimmtes Unternehmen zu machen.

4.4. Modellblöcke

Das vollständige Informationsmodell, wie in Bild 1 im Anhang A dargestellt, ist in drei definierte Abschnitte unterteilt, um eine detaillierte Beschreibung zu ermöglichen:

1) Die Modellierung der abzusichernden Anlage (grün) ist in Bild 2 hinterlegt;
2) Die Modellierung der Security-Risiken (rot) ist in Bild 3 abgebildet;
3) Die Modellierung der Security-Anforderungen (blau) ist in Bild 4 dargestellt.

Die Beschreibung des Modells in den nachfolgenden Kapiteln orientiert sich an diesen Modellblöcken. Zusätzlich zu diesen Blöcken gibt es zwei Klassen, die keinem dieser Modellblöcke zuzuschreiben sind, sondern dazu dienen, grundlegende Modellierungskonzepte darzustellen. Diese Konzepte sind das Rollenkonzept, repräsentiert durch die Klasse "role", und das Quellenkonzept, repräsentiert durch die Klasse "source", und werden in grau dargestellt.

4.5. Allgemeine Attribute

Jede einzelne Klasse aus dem Informationsmodell enthält mindestens die folgenden drei Attribute als Basis:

- UID: Eine eindeutige ID zur Identifizierung innerhalb des aktuellen Anwendungsfalles in der Form eines Textes.
- Name: Möglichst prägnanter und aussagekräftiger Name für die jeweilige Klasse als Text. Für größere Anwendungsfälle sollte ein einheitliches Schema für die Namen genutzt werden.

- Description: Freitextfeld, um die Klasse näher zu beschreiben oder Kommentare zu hinterlassen.

5. Literatur

[1] E. Taştan, S. Fluchs, and R. Drath, ‘Warum wir ein Security-Engineering-Informationsmodell brauchen’, DOI: 10.33968/2022.25

Anhang A UML-Diagramm

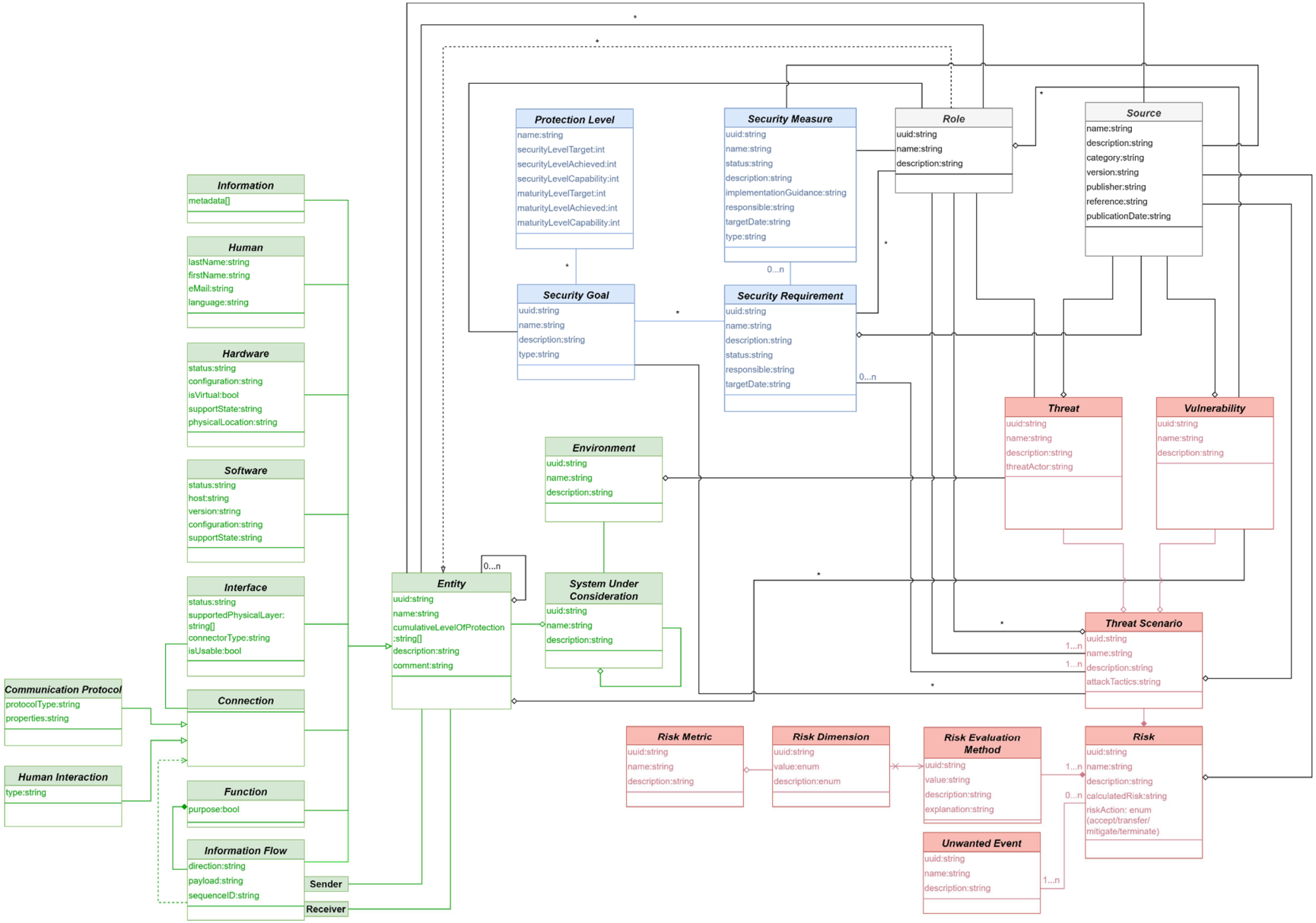

Bild 1: Automation Security Engineering Informationsmodell

Anhang B Detaillierte Modellbeschreibung

I. Modellierung der abzusichernden Anlage

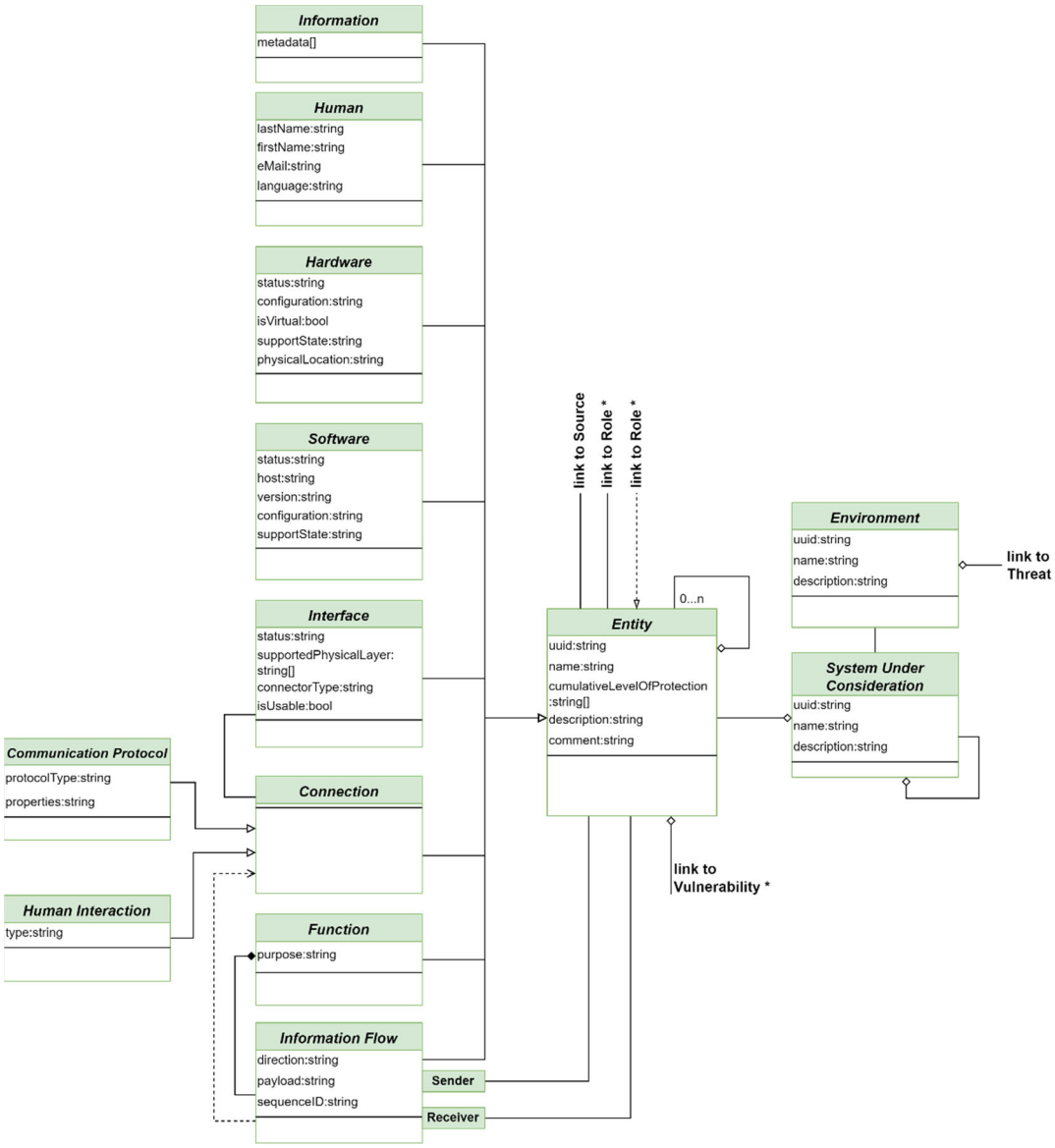

Bild 2: Ausschnitt aus dem Automation Security Engineering Informationsmodell - Abzusichernde Anlage

I.1. Entität

Die abzusichernde Anlage wird im Security-Engineering-Informationsmodell nicht vollständig modelliert. Stattdessen werden nur solche Informationen modelliert, die aus Security-Sicht relevant sind. Welche das sind, kann frei entschieden werden.

Die Klasse „Entität“ bildet diese Entscheidung ab. Der Begriff „Entität“ wird im Modell als allgemeiner Begriff für eine Security-relevante Information verwendet. Entitäten können wiederum aus Entitäten bestehen.

Die abgeleiteten Klassen dienen der genaueren Beschreibung der Entitäten. Da das Modell methodenneutral ist, lässt es Raum für ein weites Verständnis des Entitätsbegriffs. Für die Modellierung können je nach Methode die passenden Klassen ausgewählt und die übrigen ignoriert werden.

I.1.1. Zweck

Die „Entität“ definiert, als Oberklasse, identifizierende und beschreibende Attribute aller Einheiten die security-relevante Informationen enthält

I.1.2. Attribute

- cumulativeLevelOfProtection: Gesamtmaß des Schutzes der Entität
- comment: Freitext für die Weitergabe von individuellen Informationen

I.1.3. Verbindungen

- Link to Role: Die Verbindung zur Rolle wird verwendet, um festzulegen welche Rolle eine Entität im betrachteten Szenario einnimmt
- Link to Source: Beschreibt die Quelle, aus der die Informationen zu der Entität stammen
- Link to Vulnerability: Zeigt auf. welche Schwachstelle (Vulnerability) auf die Entität zutrifft oder mit dieser zusammenhängt

I.2. Role

I.2.1. Zweck

Eine Entität kann aus verschiedenen Gründen Security-relevant sein: Weil sie vor Gefährdungsszenarien geschützt werden muss und vielleicht aufgrund einer Schwachstelle angreifbar ist, weil sie selbst Teil eines Gefährdungsszenarios werden kann, weil für sie Security-Anforderungen definiert werden, weil sie Teil einer Maßnahme ist, die der Anforderungserfüllung dient – oder eine Kombination aus allen dieser Gründe.

Die Klasse „role“ kennzeichnet, aus welchem Grund oder aus welchen Gründen eine Entität Security-relevant ist, indem die Entität verschiedene Rollen einnehmen kann. Durch das Einnehmen mehrerer Rollen kann eine Entität verschiedene Eigenschaften „sammeln“: Je nach Rolle kann beispielsweise ein Mensch unterschiedliche Zugangsberechtigungen oder ein Computer unterschiedliche Software besitzen.

Je nach Rolle müssen die Entitäten anders im Sinne der Security bewertet werden. Ein Computer, der beispielsweise die Rolle „Engineering-Station Sicherheitssteuerung“ einnimmt, unterliegt anderen Security-Anforderungen als ein gewöhnlicher “Büro-PC”.

Die Beziehung zwischen den Objekten Threat scenario und Role sagt aus, dass eine Entität je nach Rolle Täter und Opfer in einem Gefährdungsszenario sein kann. Durch die Beziehung zwischen den Objekten Vulnerability und Role wird ausgedrückt, dass eine Entität je nach Rolle eine Schwachstelle und ein Opfer in einem Gefährdungsszenario sein kann.

Auch die Beziehungen zwischen Entitäten werden über Rollen definiert. Ein Mensch kann eine Entität im Betrachtungsgegenstand sein und über eine Rolle, z.B. die Rolle „Vorgesetzter“, verantwortlich (sowohl im Sinne “responsible als auch” im Sinne “accountable”) für andere Menschen sein.

I.2.2. Attribute

Neben den Standardidentifikationsmerkmalen keine weiteren Attribute.

I.2.3. Verbindungen

- Aggregation of Vulnerabilities: Durch die Einnahme einer Rolle kann eine Entität mehrere relevante Schwachstellen haben
- Link to Threat: Eine Bedrohung ist erst möglich, wenn eine bestimmte Rolle eingenommen wurde
- Link to Threat Scenario: In einem Bedrohungsszenario kann Bezug auf bestimmte Rollen genommen werden
- Link to Security Goal: Je nach eingenommener Rolle einer Entität bestehen bestimmte Security Ziele
- Link to Security Requirement: Je nach eingenommener Rolle einer Entität bestehen bestimmte Security Anforderungen
- Link to Security Measure: Je nach eingenommener Rolle einer Entität besteht die Anforderung an bestimmte Security Maßnahmen
- Link to Entity: Erklärung siehe Zweck

I.3. System under Consideration (SuC)

Alle Entitäten, also alle aus Security-Sicht relevanten Objekte, ergeben zusammen das „System under Consideration“, also den Geltungsbereich für die Security-Betrachtung.

I.3.1. Zweck

Grenzt das betrachtete System gegenüber den nicht betrachteten externen Systemen und der Umgebung ab.

I.3.2. Attribute

Neben den Standardidentifikationsmerkmalen keine weiteren Attribute.

I.3.3. Verbindungen

- Aggregation of SuC: Das „System under Consideration“ kann rekursiv aus weiteren betrachtenden SuC bestehen
- Aggregation of Entitys: Das betrachtete System besteht aus mehreren Entitäten

I.4. Umgebung / Environment

Alles, was nicht zum SuC gehört, wird abstrakt in der Klasse „Umgebung“ / „Environment“ modelliert. Dies ist relevant zur Kennzeichnung von beispielsweise Bedrohungen oder Angreifern, die nicht Teil des Systems sind (in Abgrenzung zu solchen, die es sind, beispielsweise Innentäter).

I.4.1. Zweck

Beschreibung der nicht zum betrachteten System (SuC) gehörenden Elemente, die für die Modellierung relevant sind.

I.4.2. Attribute

Neben den Standardidentifikationsmerkmalen keine weiteren Attribute

I.4.3. Verbindungen

- Link to Threat: Verbindung zu einer Bedrohung, die aus der Umwelt auf das System einwirken können

I.5. Hardware

I.5.1. Zweck

Beschreibt zum betrachteten System gehörende Hardware-Einheiten (beispielsweise Steuerungen oder Computer).

I.5.2. Attribute

- status: Beschreibung des Status der Hardware
- configuration: Aktuelle Konfiguration der Hardware
- isVirtual: Angabe, ob es sich um eine virtualisierte Hardware handelt
- supportState: Angabe des aktuellen Support Status durch den Hersteller oder den Vertreiber
- physicalLocation: Physischer Ort an dem sich die Hardware befinde (bspw. ein technischer Platz)

I.5.3. Verbindungen

s. Entität

I.6. Software

I.6.1. Zweck

Bildet die im betrachteten System verwendeten Softwarekomponenten ab.

I.6.2. Attribute

- status: Freitextbeschreibung des aktuellen Zustands der Softwarekomponente
- host: Angabe des hosts auf dem die Komponente installiert ist
- version: Versionsangabe der Softwarekomponente
- configuration: Freitextangabe der für die Modellierung betrachteten Konfiguration
- supportState: Angabe des aktuellen Support Status durch den Hersteller oder den Vertreiber

I.6.3. Verbindungen

s. Entität

I.7. Information

I.7.1. Zweck

Beschreibt relevante, zwischen den im System under Consideration vorhandenen Entitäten, ausgetauschte Informationen

I.7.2. Attribute

- metadata: Metadaten der Information

I.7.3. Verbindungen

s. Entität

I.8. Human

I.8.1. Zweck

Repräsentation von Personen, welche mit den Entitäten des betrachteten Systems (SuC) interagierenden.

I.8.2. Attribute

- lastName: Familienname der Person
- firstName: Vorname der Person
- eMail: E-Mail-Adresse der Person
- language: Bevorzugte Interaktionssprache der Person

I.8.3. Verbindungen

s. Entität

I.9. Interface

I.9.1. Zweck

Beschreibt Schnittstellen innerhalb des System under Consideration die die Interaktion zwischen verschiedenen Entitäten ermöglichen. Dies können Maschinenschnittstellen (M2M) oder auch Mensch-Maschine-Schnittstellen (HMI) sein.

I.9.2. Attribute

- status: Freitextbeschreibung des Schnittstellenzustands
- supportedPhysicalLayer: Freitextbeschreibung der unterstützten Physical Layer nach ISO-Referenzmodell
- connectorType: Beschreibung des physikalischen Anschlusses (Steckertyp)
- isUsable: Gibt an, ob die Schnittstelle genutzt werden kann

I.9.3. Verbindungen

- Link to Connection: Gibt an von welcher Verbindung die Schnittstelle verwendet wird

I.10. Connection

I.10.1. Zweck

Modelliert die Möglichkeit zum Austausch von Informationen zwischen zwei oder mehr Entitäten des Modells.

I.10.2. Attribute

Neben den Standardidentifikationsmerkmalen keine weiteren Attribute.

I.10.3. Verbindungen

- Link to InformationFlow: Zeigt auf, welcher Informationsfluss durch die Verbindung ermöglicht wird
- Link to Interface: Bildet ab, welche Interfaces für die Verbindung verwendet werden

I.11. **Communication** protocol

I.11.1. Zweck

Als Spezialisierung einer Verbindung bildet das Kommunikationsprotokoll die Verbindung zwischen zwei technischen Systemen mittels eines vorgeschriebenen Formats und Ablaufs ab.

I.11.2. Attribute

- protocolType: Name des eingesetzten Protokolls
- properties: Eigenschaften des eingesetzten Protokolls bei der Betrachtung

I.11.3. Verbindungen

s. Connection

I.12. **Human** interaction

I.12.1. Zweck

Beschreibt die Interaktion eines Menschen (Human) mit den im betrachteten System vorkommenden technischen Systemen. Die Human Interaction ist eine Spezialisierung der Connection.

I.12.2. Attribute

- type: Freitextbeschreibung der Interaktionsart

I.12.3. Verbindungen

I.13. **Information** flow

I.13.1. Zweck

Abbildung des Austauschs von Informationen zwischen zwei Entitäten (Sender und Receiver) des betrachteten Systems. Dies erfolgt in aller Regel zur Erfüllung einer *Function.*

I.13.2. Attribute

- direction: Richtung des Informationsflusses.
- payload: Übertragene Nutzdaten des Informationsflusses
- sequenceID: Identifikation für die Ablaufverfolgung der einzelnen Informationen

I.13.3. Verbindungen

- Link to Sender: Entität welche als Ausgangspunkt des Informationsflusses fungiert
- Link to Receiver: Entität welche als Empfangspunkt des Informationsflusses fungiert

I.14. Function

Eine Funktion ist ein sequenzieller Ablauf von Informationsflüssen zu einem bestimmten Zweck. Ein Beispiel ist die Funktion „SPS-Programmierung“: Der Zweck der Informationsflüsse ist die Programmierung einer SPS. Die Funktion besteht aus zwei Informationsflüssen:

1. Ein Informationsfluss zwischen dem SPS-Programmierer und einem Programmiergerät, aufbauend auf der Verbindung vom Typ „menschliche Interaktion“ (human interaction), bei der der SPS-Programmierer neue SPS-Logik in das Programmiergerät eingibt und

2. ein Informationsfluss zwischen Programmiergerät und SPS, aufbauend auf einer Verbindung vom Typ vom Typ „Kommunikationsprotokoll“ (communication protocol), bei dem ein neues Programm vom Programmiergerät auf die SPS geladen wird.

Die menschlichen Interaktionen können als Attribute auch Berechtigungen beinhalten.

I.14.1. Zweck

Beschreibt den Ablauf von Informationsflüssen (Information Flow) zur Erfüllung einer Aufgabe

I.14.2. Attribute

- purpose: Zweck der in der Funktion beschriebenen Informationsflüsse.

I.14.3. Verbindungen

- Aggregation of InformationFlow: Die Funktion ist durch eine Abfolge von Informationsflüssen beschrieben. Diese sind in der Aggregation abgebildet.

II. Modellierung der Security-Risiken

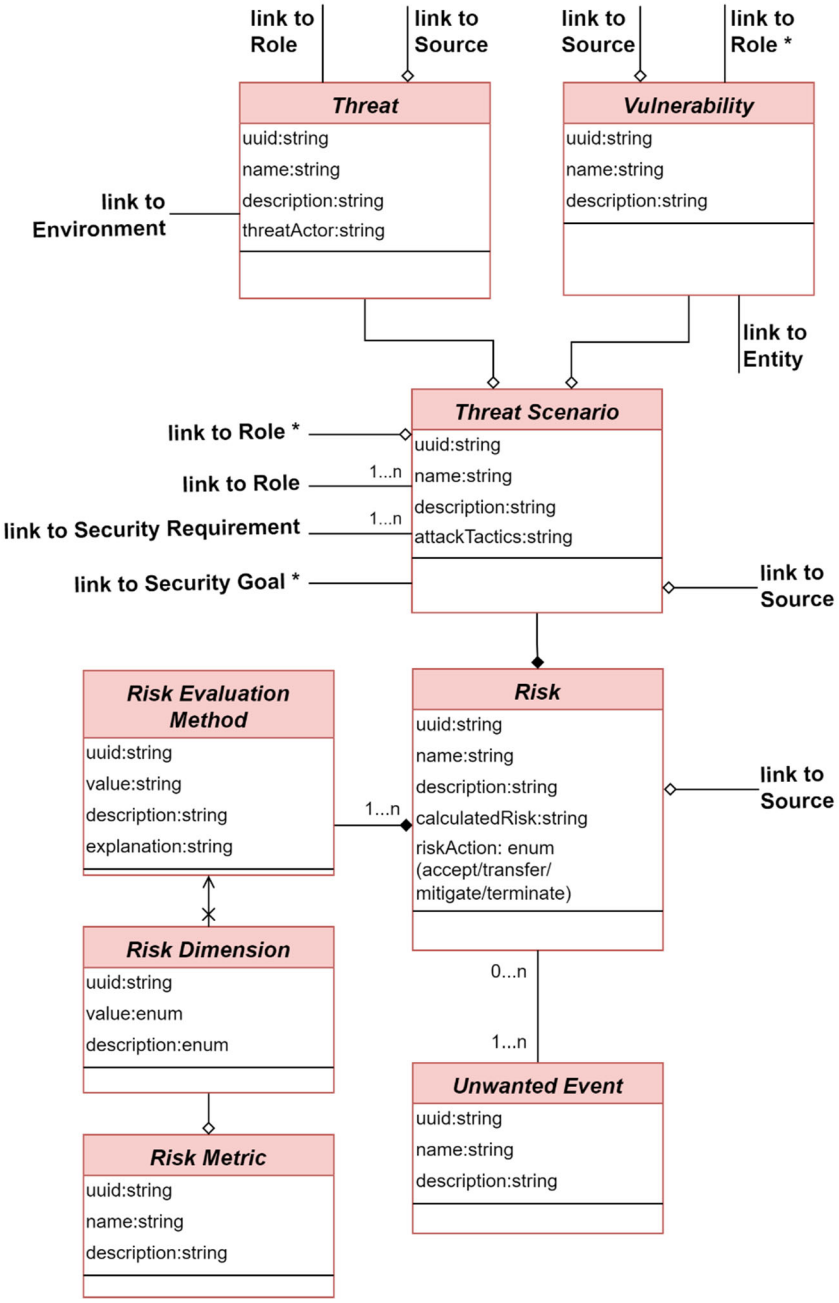

Bild 3: Ausschnitt aus dem Automation Security Engineering Informationsmodell - Security-Risiken

II.1. Unwanted event

II.1.1. Zweck

Das unerwünschte Ereignis (unwanted event) ist ein Zustand, der durch die Security-Betrachtung und die daraus resultierenden Maßnahmen vermieden werden soll. Die Zustände sind anlagenspezifisch.

Unerwünschte Ereignisse können Szenarien aus der funktionalen Sicherheit entsprechen (z.B. das Explodieren eines Kessels), aber auch darüber hinausgehen (z.B. die Beschädigung eines wertvollen Anlagenteils oder das Unterschreiten von Qualitätskriterien für eines der in der Anlage hergestellten Produkte).

II.1.2. Attribute

Neben den Standardidentifikationsmerkmalen keine weiteren Attribute.

II.1.3. Verbindungen

- Link to Risk: Ein Risiko beinhaltet immer ein unerwünschtes Ereignis, welches eintreten kann und zur Beschreibung der Auswirkungen genutzt wird

II.2. Threat scenario

II.2.1. Zweck

Gefährdungsszenarien (threat scenario) sind Szenarien, die eines der unerwünschten Ereignisse herbeiführen. Ein Gefährdungsszenario besteht klassischerweise aus der Kombination einer Bedrohung und einer Schwachstelle.

II.2.2. Attribute

- attackTactics: Kann zum Beispiel für die Modellierung einer Taktik aus dem MITRE ATT&CK Framework verwendet werden

II.2.3. Verbindungen

- Link to Role: Entität kann je nach Rolle Täter und Opfer von Gefährdungsszenario sein
- Link to Security requirement: Stellt dar, welche Security requirements dem Gefährdungsszenario zu Grunde liegen
- Link to Security goal: Stellt dar, welches Security Goal dem Gefährdungsszenario zu Grunde liegt
- Link to Source: Typischerweise werden Gefährdungsszenarien aus Katalogen oder Datenbanken genutzt, um sie dann anzuwenden und zu bewerten
- Link to Threat: Ein Gefährdungsszenario beinhaltet auch immer eine entsprechende Bedrohung zur vollständigen Beschreibung
- Link to Vulnerability: Ein Gefährdungsszenario beinhaltet auch immer eine technische Schwachstelle zur vollständigen Beschreibung

II.3. Threat

II.3.1. Zweck

Eine Bedrohung ist ein Angreifer bzw. ein Ereignis, der oder das im Rahmen eines Gefährdungsszenarios agiert, um ein unerwünschtes Ereignis herbeizuführen.

II.3.2. Attribute

- threatActor: Kann zum Beispiel für die Modellierung eines Angreifers aus der Intel Threat Agency Library (TAL) verwendet werden

II.3.3. Verbindungen

- Link to Source: Typischerweise werden Bedrohungen aus Katalogen oder Datenbanken genutzt, um sie dann anzuwenden und zu bewerten
- Link to Environment: Verbindung zur Umwelt des Systems, welche den Ursprung der Bedrohung darstellt
- Link to Role: Entität kann je nach Rolle der Angreifer in einem Threat oder Teil des Ereignisses sein
- Link to Threat scenario: Detailliert das Gefährdungsszenario weiter mit möglichen Bedrohungen aus

II.4. Vulnerability

II.4.1. Zweck

Eine Schwachstelle ist eine Eigenschaft einer Entität, die im Rahmen eines Gefährdungsszenarios verwendet werden kann, um ein unerwünschtes Ereignis herbeizuführen.

Das Verständnis von „Schwachstelle“ ist im Modell im Sinne einer Angriffsmöglichkeit weit gefasst. Die Schwachstelle kann ein klassischer Software-Fehler sein, der in der Common Vulnerability Enumeration (CVE) Database gelistet ist – aber auch ein intendiertes Feature einer Entität, das nicht als Schwachstelle im engeren Sinne zu bezeichnen ist, aber trotzdem eine Angriffsmöglichkeit bietet. Ein Beispiel ist eine Konfiguration, die das Aktualisieren der SPS-Logik im laufenden Betrieb erlaubt.

II.4.2. Attribute

Neben den Standardidentifikationsmerkmalen keine weiteren Attribute.

II.4.3. Verbindungen

- Link to Source: Typischerweise werden Schwachstellen aus Katalogen oder Datenbanken genutzt, um sie dann auf das System under Consideration anzuwenden und zu bewerten
- Link to Entity: Entität kann unabhängig von der Rolle eine Schwachstelle haben
- Link to Role: Eine Entität kann rollenbezogene Schwachstellen haben

II.5. Risk

II.5.1. Zweck

Sobald ein Gefährdungsszenario eine Risikobewertung erhält, ist es ein Risiko. Da das Modell methodenneutral ist, kann die Methode für die Risikobewertung frei definiert werden. Sie setzt sich zusammen aus einer beliebigen Anzahl von Bewertungsdimensionen (Risk dimension), einer Risikometrik (Risk metric) zur Bewertung der einzelnen Risikodimensionen und der Methode der Risikobewertung (Risk evaluation method), die vorgibt, wie sich aus der Bewertung der einzelnen Risikodimensionen das Risiko ergibt.

II.5.2. Attribute

- calculatedRisk: Finale Bewertung eines Risikos entweder in qualitativer Form (z.B. „low“, „medium“ und „high“) oder in quantitativer Form (z.B. Skala 1-10, Kosten oder Ausfallzeit)
- riskAction: Entscheidung der verantwortlichen Stakeholder wie mit einem Risiko umzugehen ist. Per Definition sind nur die Werte „accept“, „transfer“, „mitigate“ und „terminate“ zugelassen

II.5.3. Verbindungen

- Link to Source: Typischerweise basieren Risiken auf Informationen aus vorher definierten Katalogen oder Datenbanken
- Link to Threat scenario: Jedes Risiko basiert auf einem Gefährdungsszenario und benötigt dieses zur vollständigen Beschreibung
- Link to Risk evaluation method: Jede Risikobewertung basiert auf einer bestimmten Methode, die in diesem Modell für den jeweiligen Nutzer offengelassen worden ist

II.6. Risk dimension

II.6.1. Zweck

Die gängigen Risikodimensionen sind Eintrittswahrscheinlichkeit und Auswirkung, aber das Modell lässt Raum dafür, die Dimensionen anders zu definieren oder zusätzliche Dimensionen einzubeziehen.

II.6.2. Attribute

Neben den Standardidentifikationsmerkmalen keine weiteren Attribute.

II.6.3. Verbindungen

- Link to Risk evaluation method: Die frei wählbaren Risikodimensionen können genutzt werden, um die angewendete Risikobewertungsmethode näher zu beschreiben
- Link to Risk metric: Die Risikodimensionen können quantitativ oder qualitativ ausgestaltet und messbar gemacht werden

II.7. Risk metric

II.7.1. Zweck

Jede Risikodimension hat eine Metrik, nach der die Ausprägung der Dimension bemessen wird. Die Risikometrik entscheidet auch darüber, ob das Risiko qualitativ oder quantitativ bemessen wird. Die Eintrittswahrscheinlichkeit kann beispielsweise in 10 %-Schritten angegeben werden oder als Wert aus der Menge {niedrig, mittel, hoch}.

II.7.2. Attribute

Neben den Standardidentifikationsmerkmalen keine weiteren Attribute.

II.7.3. Verbindungen

- Link to Risk dimension: Risikometriken beschreiben genutzte Risikodimensionen näher und können sie messbar gestalten

II.8. Risk evaluation method

II.8.1. Zweck

Die Risikobewertungsmethode gibt an, wie sich aus der Bewertung der einzelnen Risikodimensionen das Risiko ergibt. Häufig ist dies eine Matrix, die aus der Kombination der Werte der einzelnen Dimensionen einen Gesamtwert berechnet. Stattdessen sind aber beispielsweise auch Entscheidungsbaumverfahren denkbar.

II.8.2. Attribute

- explanation: Textuelle Erklärung und detaillierte Beschreibung der verwendeten Risikobewertungsmethode

II.8.3. Verbindungen

- Link to Risk dimension: Jede Risikobewertungsmethode setzt sich aus verschiedenen und frei wählbaren Risikodimensionen zusammen. Diese beschreiben den Ansatz und den Umfang der jeweiligen Methode
- Link to Risk: Risiken sind die Ergebnisse aus einer definierten Risikobewertungsmethode

III. Modellierung der Security-Anforderungen

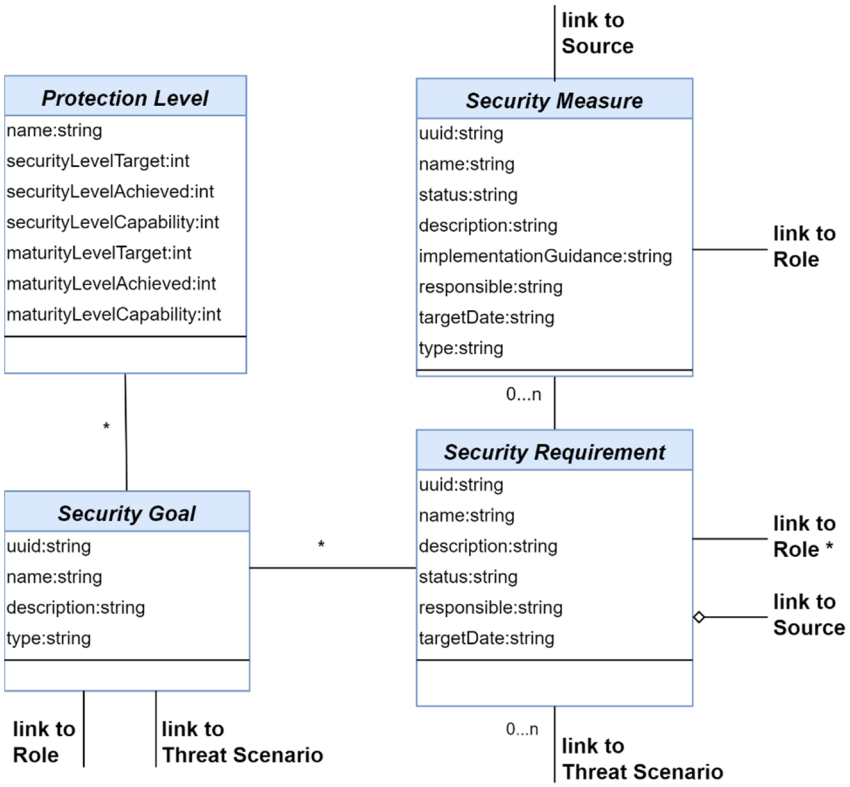

Bild 4: Ausschnitt aus dem Automation Security Engineering Informationsmodell - Security-Anforderung

III.1. Security goal

III.1.1. Zweck

Das Security Goal (Schutzziel) zeigt an, ob und inwiefern eine Entität des System under Consideration in der vorliegenden Security-Betrachtung schutzbedürftig ist.

III.1.2. Attribute

- type: Um das Modell methodenneutral zu halten, kann das Schutzziel auf beliebige Weise modelliert werden. Die Kategorisierung in „Typen“ kann auf verschiedene Arten erfolgen. Üblicherweise werden Schutzziele mit Kombinationen der Ziele Vertraulichkeit, Verfügbarkeit und Integrität beschrieben. Es kann aber auch durch weitere Dimensionen wie Authentizität oder Nichtabstreitbarkeit (non-repudiation) ergänzt oder durch gänzlich eigene Dimensionen definiert werden

III.1.3. Verbindungen

- Link to Role: Stellt dar, für welche Entitäten ein bestimmtes Security Goal gilt. Die Verbindung besteht zur Rolle und nicht direkt zur Entität, weil eine Entität je nach der Rolle, die sie einnimmt, verschiedene Security Goals haben kann (siehe Beschreibung der Klasse „Rolle“)
- Link to Threat Scenario: Ein Threat Scenario kann ein oder mehrere Security Goals beeinträchtigen. Andersherum kann sich das Schutzziel auch aus dem Gefährdungsszenario ergeben
- Link to Security Requirement: Ein Requirement kann ein oder mehrere Security Goals erfüllen bzw. zu seiner Erfüllung beitragen
- Link to Protection Level: Wenn Protection Level definiert werden, kann das Security Goal auch zu diesen eine Verbindung haben, denn es ist denkbar, Protection Level als Vektor zu definieren – mit unterschiedlichen Werten je Security Goal

III.2. Protection level

III.2.1. Zweck

Das Protection Level dient der Quantifizierung des Security Goals.

Die Markierung von Entitäten mit Protection Levels (auch: Security-Leveln oder Schutzniveaus) ist ein gängiges Konzept einiger Methoden (am weitesten entwickelt in der Normenreihe IEC 62443).

III.2.2. Attribute

Bei der Bemessung des Protection Levels ist es relevant, zwischen Soll- und Ist-Zustand zu unterscheiden; daher hat die Protection Level-Klasse dafür eigene Attribute. Der Normenreihe IEC 62443 folgend, ist neben dem Ist-Zustand („achieved“), dem Soll-Zustand („target“) auch noch der theoretisch mögliche Zustand („capability“) als Attribut vorgesehen:

Außerdem haben viele Security Requirements sowohl eine technische als auch eine organisatorische Komponente, weshalb – ebenfalls der Normenreihe IEC 62443 folgend – zwischen einem „security level“ (technisch) und einem „maturity level“ (organisatorisch) unterschieden wird. So ergeben sich in Summe sechs mögliche Werte für die Quantifizierung des Protection Levels:

- securityLevelTarget: Soll-Wert für den technischen Security-Zustand
- securityLevelAchieved: Ist-Wert für den technischen Security-Zustand
- securityLevelCapability: Theoretisch maximal möglicher Wert für den technischen Security-Zustand
- maturityLevelTarget: Soll-Wert für den organisatorischen Security-Zustand
- maturityLevelAchieved: Ist-Wert für den organisatorischen Security-Zustand
- maturityLevelCapability: theoretisch maximal möglicher Wert für den organisatorischen Security-Zustand

Zusätzlich ist es möglich, das Protection Level als Vektor bzw. Matrix zu verstehen, das heißt verschiedene Werte (in allen sechs Attributen) für jedes Security Goal zu vergeben.

III.2.3. Verbindungen

- Link to Security Goal: Da die Protection Level eine Quantifizierung der Security Goals darstellt, kennzeichnet die Verbindung zur Security-Goal-Klasse das gemessene Security Goal.

III.3. Security requirement

III.3.1. Zweck

Ein Security Requirement (Security-Anforderung) beschreibt, welche Eigenschaften eine Entität oder eine Gruppe von Entitäten aus Security-Sicht haben soll. Das Requirement ist unabhängig von der Implementierung; es beschreibt das „Was“, nicht aber das „Wie“.

Eine Anforderung kann auf mehreren Wegen entstehen: Sie kann ein Design-Ziel (als nicht-funktionale Anforderung) darstellen, sie kann notwendig werden zur Behandlung eines Risikos oder für die Erfüllung eines internen oder externen Regulariums. Wenn Security Goals definiert wurden, ist das Security Requirement die Konkretisierung eines Security Goals für eine oder mehrere Entitäten in ein Security Requirement für eine (möglicherweise andere) Entität oder Gruppe von Entitäten.

III.3.2. Attribute

- status: Erfüllungsgrad einer Anforderung
- targetDate: Zieldatum für die Erfüllung
- responsible: klärt, wer die Erfüllung der Anforderung verantwortet

III.3.3. Verbindungen

- Link to Entity: Wie auch das Security Goal hat das Security Requirement eine Verbindung zu einer oder mehreren Entitäten, für die das Security Requirement gilt (je nachdem, welche Rolle sie einnehmen, s. Beschreibung der Klasse „Rolle“)
- Link to Security Measure: Ein Security Requirement kann durch eine oder mehrere Security Measures umgesetzt werden

Die restlichen Verbindungen bilden die möglichen Herkünfte von Security Requirements ab:

- Link to Security Goal: Security Goal(s), das oder die das Requirement konkretisiert
- Link to Threat Scenario: Threat Scenarios, die das vorliegende Requirement erschwert, in den Auswirkungen abmildert oder sogar unmöglich macht
- Link to Source: Modelliert die Herkunft von Security-Requirements aus Regularien, Standards etc.

III.4. Security measure

III.4.1. Zweck

Eine Security Measure (Security-Maßnahme) definiert, wie eine Entität ihre aus Security-Sicht wichtigen Eigenschaften erlangen soll. Während das Security Requirement implementierungsunabhängig das „Was“ beschreibt, klärt die Security Measure das „Wie“, zum Beispiel eine konkrete Konfiguration einer Entität.

III.4.2. Attribute

- status: Erfüllungsgrad einer Anforderung. Für die Dokumentation von Security Measures ist es wichtig, zwischen Soll-Werten und Ist-Werten der Maßnahmen zu unterscheiden. Im Modell wird dies abgebildet, indem die Soll-Werte in der Klasse „Security-Maßnahme“ modelliert werden, Ist-Werte hingegen direkt bei der jeweiligen Entität im Anlagenmodell (unter Bezugnahme auf die Security-Maßnahme)
- targetDate: Zieldatum für die Erfüllung
- responsible: Klärt, wer die Erfüllung der Anforderung verantwortet
- type: Dient der Kategorisierung von Security Measures
- implementationGuidance: Freitextfeld für die Eingabe weiterer Hinweise zur Umsetzung einer Security Measure

III.4.3. Verbindungen

- Link to Entity: Wie auch das Security Goal und das Security Requirement hat die Security Measure eine Verbindung zu einer oder mehreren Entitäten, die die Security Measure implementieren sollen (je nachdem, welche Rolle sie einnehmen, s. Beschreibung der Klasse „Rolle“)
- Link to Security Requirement: Das oder die Security Requirement(s), die durch die Security-Measure konkretisiert und implementiert werden
- Link to Source: Modellierung der Herkunft von Security-Measures aus bestimmten Quellen– Regularien, Standards, Best Practices, etc.

III.5. Source

III.5.1. Zweck

Die Klasse Source (Quelle) dient der Erweiterbarkeit des Modells aus existierenden Modellen oder anderen externen Quellen. Externe Quellen dienen zum Beispiel der Referenzierung von Standards oder Katalogen zum Beispiel für Security-Requirements oder -Measures, Threats oder Vulnerabilties.

Existierende Informationsmodelle können beispielsweise detaillierte Informationen über eine Entität beinhalten, die im Rahmen des Security-Engineering-Modells zwar relevant sind, aber nicht redundant noch einmal modelliert werden sollen.

Auch wenn die Source-Klasse längst nicht nur für die Modellierung der Security-Anforderungen relevant ist, wird sie in diesem Kapitel beschrieben, weil Quellen für Security-Anforderungen der geläufigste Anwendungsfall für

die Source-Klasse sind. Daher ist die Source-Klasse nicht in Bild 3 dargestellt, sondern in der Überblickszeichnung (Bild 1).

III.5.2. Attribute

- category: dient der Strukturierung größerer Quell-Bibliotheken

Attribute, die der eindeutigen Beschreibung einer Publikation dienen:

- version: Versionsnummer der Source
- publisher: Herausgeber der Source
- publicationDate: Veröffentlichungsdatum der Source
- reference: z. B. Link auf einen öffentlichen oder privaten Ablageort

III.5.3. Verbindungen

Die Source-Klasse hat Verbindungen zu allem, wofür es Regularien, Standards oder Kataloge gibt, die als Quelle hinterlegt werden könnten:

- Link to Threat: zeigt Bedrohungen im Modell an, die aus der vorliegenden Quelle stammen
- Link to Vulnerability: Zeigt Schwachstellen im Modell an, die aus der vorliegenden Quelle stammen
- Link to Threat Scenario: Zeigt Gefährdungsszenarien im Modell an, die aus der vorliegenden Quelle stammen
- Link to Risk´: Zeigt Risiken im Modell an, die aus der vorliegenden Quelle stammen
- Link to Security Requirement: Zeigt Security-Anforderungen im Modell an, die aus der vorliegenden Quelle stammen
- Link to Security Measure: Zeigt Security-Maßnahmen im Modell an, die aus der vorliegenden Quelle stammen
- Link to Entity: zeigt Entitäten im Modell an, die aus der vorliegenden Quelle kommen. Dies kann auch bereits existierende Modellierungen für diese Entitäten gelten, – zum Beispiel in Verwaltungsschalen o.ä.