Technische Hochschule Ingolstadt

PAESSLER
THE NETWORK MONITORING COMPANY

Masterarbeit

Methodik zur Angriffsmodellierung für Security-Tests

zur Erlangung des akademischen Grades
Master of Science

vorgelegt von

Tina Volkersdorfer

Studiengang:	Informatik
Fakultät:	Informatik
Ausgabetermin:	24.07.2019
Abgabetermin:	23.01.2020
Erstprüfer:	Prof. Dr.-Ing. Hans-Joachim Hof
Zweitprüfer:	Prof. Dr.-Ing. Ernst-Heinrich Göldner
Industriepartner:	Paessler AG
Betreuer:	Mathias Hengl (Product Architect)

Danksagung

Ich möchte mich bei allen bedanken, die mich während der Anfertigung meiner Masterarbeit unterstützt haben.

Zunächst möchte ich mich bei Herrn Jörn Paessler bedanken, dass ich diese Arbeit in Kooperation mit der Paessler AG in Nürnberg schreiben durfte. Vielen Dank für die Unterstützung und die Möglichkeiten, informative Veranstaltungen, wie die TestBash Germany 2019 zu besuchen.

Meinem Betreuer Herrn Mathias Hengl möchte ich besonders für seine Tipps, Denkanstöße und der konstruktiven Kritik während der Anfertigung meiner Masterarbeit danken.

Ebenso gilt mein besonderer Dank Herrn Prof. Dr.-Ing. Hans-Joachim Hof für die Betreuung und Unterstützung seitens der Technischen Hochschule Ingolstadt. Bedanken möchte ich mich für die Möglichkeit, dass ich meine Masterarbeit im Umfeld eines noch offenen Forschungsgebiets schreiben durfte. Ich bedanke mich dafür, dass Sie mir zuverlässig bei Fragen beiseite standen, für Ihre Geduld und den richtungsweisenden Gesprächen.

Des Weiteren gilt mein Dank Herrn Prof. Dr.-Ing. Ernst-Heinrich Göldner für die Zweitkorrektur.

Zudem möchte ich mich bei meinen Kollegen der Paessler AG und der Technischen Hochschule Ingolstadt für die hilfreichen Anregungen und interessanten Debatten bei der Erstellung dieser Arbeit bedanken.

Ganz herzlich bedanke ich mich bei meiner Freundin Frau Anke Lukas und meiner Mutter Frau Anja Volkersdorfer für die vielen Stunden des Korrekturlesens.

Schließlich möchte ich mich ganz besonders bei meinem Partner Marco Dietrich für den starken Rückhalt während meiner Masterarbeit bedanken, die in den Zeitraum unseres Umzugs fiel. Meiner Familie danke ich ganz herzlich, die stets ein offenes Ohr für meine Sorgen hatte und mich in allen Lebenslagen unterstützt.

Abkürzungsverzeichnis

AI Artifical Intelligence

APT Advanced Persistent Threat

ATT&CK Adversarial Tactics, Techniques and Common Knowledge

BSI Bundesamt für Sicherheit in der Informationstechnik

CAPEC Common Attack Pattern Enumeration and Classification

CVE Common Vulnerabilities and Exposures

CWE Common Weakness Enumeration

EDR Event Data Recorder

KRITIS Kritische Infrastrukturen

MASSiF Modellbasierte Absicherung von Security und Safety für umfeldbasierte Fahrzeugfunktionen

OBD On-Board-Diagnose

OWASP Open Web Application Security Project

SDL Security Development Lifecycle

SQL Structured Query Language

SQLI Structured Query Language Injection

TTP Tactics, Techniques and Procedures

UML Unified Modeling Language

UMLsec Standard Unified Modeling Language extention mechanism for secure system development

XSS Cross Site Scripting

Inhaltsverzeichnis

1 Einleitung

Die Vernetzung von Systemen nimmt immer weiter zu. Dafür sorgt nicht nur die klassische Informationstechnologie an sich, sondern vielmehr die Industrie. Diese zielen immer mehr auf die stetige Optimierung und Verbesserung durch Digitalisierung ab, wie die zunehmenden Trends Internet of Things, Industrial Internet of Things, oder Artifical Intelligence (AI) zeigen.[30] Das führt zu einer steigenden Komplexität der Systeme und neuen Herausforderungen, die damit verbunden sind.[53] Die neuen Technologien bedeuten gleichzeitig neue Angriffsmöglichkeiten auf IT-Systeme.[53] Beispielsweise können Angreifer mithilfe von AI die Sicherheitsmechanismen von Unternehmen umgehen.[30] Ebenso wird die Einführung von 5G und die damit verbundenen kommunizierenden Geräte dafür sorgen, dass die Angriffsoberfläche noch weiter zunimmt.[30]

Durch die zunehmende Vernetzung steigt auch das benötigte Sicherheitsniveau.[53] Die komplexen, vernetzten Systeme müssen vor unberechtigten Manipulationen geschützt und das Ausspionieren von sicherheitsrelevanten Daten verhindert werden.[53] Die folgenden Informationen dieses Abschnitts sind [54] entnommen. Insbesondere bei Kritischen Infrastrukturen (KRITIS), wie z. B. bei der Stromversorgung, ist das Schadensausmaß bezüglich der gesamten modernen Gesellschaft, bei erfolgreichen Angriffen erheblich. Im Februar 2016 sorgte z. B. ein Ransomware-Trojaner in einem Krankenhaus dafür, dass letztendlich das interne Computer-Netzwerk heruntergefahren werden musste, da das Schadprogramm den internen IT-Betrieb negativ beeinflusste. Speziell die Wiederherstellung der Prozesse und IT-Systeme kostete das Krankenhaus ungefähr 1 Millionen Euro. Daher spielt die IT-Sicherheit in einigen Bereichen, wie bei KRITIS eine sehr große Rolle. Das Bundesamt für Sicherheit in der Informationstechnik (BSI) bietet dahingehend als nationale Cyber-Sicherheitsbehörde seine Unterstützung bei Prävention, Detektion und Reaktion an. Für Unternehmen stehen z. B. die IT-Grundschutz-Kataloge [53] als Nachschlagewerk zur Verfügung. Gemäß § 8a BSIG[1] sind die KRITIS-Betreiber dazu verpflichtet geeignete Methoden und Mechanismen zu etablieren, um Angriffe möglichst frühzeitig zu identifizieren, sodass KRITIS vor Beeinträchtigungen geschützt sind. Das zeigt die hohe Wichtigkeit von Methoden und Techniken zur Gewährleistung von IT-Sicherheit innerhalb des gesamten Softwareentwicklungslebenszyklus für komplexe, vernetzte und insbesondere für sicherheitskritische Systeme.[53]

[1] BSI-Gesetz vom 14. August 2009 (BGBl. I S. 2821), das zuletzt durch Artikel 1 des Gesetzes vom 23. Juni 2017 (BGBl. I S. 1885) geändert worden ist.

Die folgende Ausführung des Paragrafen basiert auf [53, 46]. Zur Gewährleistung von IT-Sicherheit muss auf verschiedenen Ebenen angesetzt werden. Allgemein fordert die Herstellung sicherer Software sowohl die Berücksichtigung von Sicherheit bei den Geschäftsprozessen, inklusive Prozesse zur Entwicklung der Software, als auch die Identifikation von Sicherheitsanforderungen an das zu entwickelnde System. Dahingehend empfiehlt das BSI z. B. einen Sicherheitsprozess im Rahmen eines notwendigen Informationssicherheitsmanagements. Dieses Management ist für die Absicherung von Geschäftsprozessen und dem IT-Betrieb essentiell. Ansonsten kann IT-Sicherheit und das damit verbundene Sicherheitsniveau nicht durchgängig im Unternehmen gewährleistet werden. In diesem Kontext spielen Security-Analysen eine wichtige Rolle. Bedrohungs- und Risikoanalysen sind z. B. maßgebend für die Schutzbedarfsfeststellung bei hohem Schutzbedarf im BSI-Sicherheitsprozess.

Beispielsweise repräsentiert die Risk Rating Methode von Open Web Application Security Project (OWASP) eine Risikoanalyse.[21] Das Ergebnis der Analyse stellt einen Überblick von sicherheitsrelevanten Informationen in Form von Fließtext in Excel-Tabellen für Unternehmen dar.[21] Dieses Wissen kann nicht einfach wiederverwendet werden, da Fließtext kein parametrisierbares und strukturiertes Modell ist, sondern willkürliche Inhalte aufweist. Gleichzeitig muss die Bedrohungs- und Risikoanalyse stets wiederholt angewandt werden, da sie lediglich eine Momentaufnahme der aktuellen Bedrohungslage darstellt.[21] Mit der Existenz neuer Schwachstellen und Angriffe, bzw. spätestens mit Veröffentlichung von Zero-Day-Exploits, müssen die aufwändigen Security-Analysen erneut durchgeführt und die Sicherheitsanforderungen entsprechend angepasst, bzw. Sicherheitsmaßnahmen umgesetzt werden.[14, 53, 18] Darüber hinaus sind z. B. in agilen Softwareentwicklungsprozessen, oder bei den Ansätzen von DevOps geeignete Konzepte gefragt, die wiederholt angewandt bzw. automatisiert werden können.[57, 2] Daher ist ein generisches Modell für Angriffe, Angreifer und der Umgebung wünschenswert, das strukturiert und parametrisierbar ist, sodass es einfach, ohne viel Aufwand angepasst und im Rahmen von Security-Tests (wieder-)verwendet werden kann.

Deshalb soll in dieser Arbeit ein Konzept zur strukturierten und formalen Modellierung von Angriffen entwickelt und umgesetzt werden. Ziel ist es, dass mithilfe der entwickelten Methodik alle bekannten Angriffe modelliert werden können. Eine Methodik zur Modellierung eines Angreifer- und Umgebungsmodells steht dagegen nicht im Fokus. In diesem Zusammenhang werden gewissen Annahmen für die Masterarbeit gemacht.

Die Unterscheidung zwischen Angreifer- und Angriffsmodellen ist notwendig, um eine flexible Reaktion auf neue Angriffe zu gewährleisten. Es soll z. B. möglich sein, das Angreifermodell eines Skript-Kiddies mit verschiedenen Angriffen über ein geeignetes Angriffsmodell zu kombinieren. Dadurch reduziert sich der Aufwand der Angreifer- und Angriffsmodellierung auf lange Sicht, da der Aufwand zur Neuentwicklung wegfällt und die Modelle wiederverwendet werden können.

Für das Konzept eines generischen Angriffsmodells sind zunächst, mithilfe einer Anforderungsanalyse die Rahmenbedingungen für das generische Angriffsmodell zu bestimmen. Die Analyse der verschiedenen Sichtweisen eines Angriffs ist notwendig, um herauszufinden, welche charakteristischen Eigenschaften einen Angriff beschreiben.

Es ist eine Modellierungsmethodik zu bestimmen, die die Grundlage für eine systematische, formale, grafische Darstellung von Angriffen bildet. Dabei stellt sich die Frage, wie aus einem generischen Angriffsmodell, ohne viel Aufwand, spezifische Angriffe abgeleitet werden können. Außerdem sind die Voraussetzungen in Form von relevanten Informationen und Modellelementen festzustellen, die für eine wiederverwendbare Angriffsmodellierung benötigt werden. Im Idealfall soll in Zukunft mithilfe von Werkzeugen, Automatismen und der geeigneten Kombination und Wiederverwendung von Informationen bzw. Modellen, wie Angriffs-, Angreifer-, System- und gegebenenfalls weiteren Modellen des Umfelds auf einen Blick, intuitiv sichtbar sein, welche Teile des Systems bei einem bestimmten Angriff eines bestimmten Angreifers gefährdet sind. Dabei stellt sich die Frage, inwieweit die entwickelte Methodik zur Angriffsmodellierung mit UML-Modellen verknüpft werden kann.

Die Evaluierung der entwickelten Methodik ist mit einigen Schwierigkeiten verknüpft, auf die zunächst eingegangen wird. Als Evaluierungskonzept gilt es, die erarbeitete Methodik zur Angriffsmodellierung auf Basis von zwei relevanten Angriffsszenarien, aus verschiedenen Anwendungsbereichen anzuwenden. Damit sollen die analysierten Anforderungen an die Methodik zur Angriffsmodellierung untersucht werden.

Die Arbeit ist folgendermaßen gegliedert: Zunächst erfolgt die Abgrenzung zu anderen Arbeiten. Anschließend werden einige Grundlagen für das allgemeine Verständnis der Masterarbeit wiederholt. Darunter fällt die Definition von grundlegenden Begriffen und ein kurzer Einblick in verschiedene Aspekte, die in Zusammenhang mit der Modellentwicklung stehen. Mithilfe der Analyse von allgemeinen Anforderungen an das zu entwickelnde Angriffsmodell werden die Rahmenbedingungen für die Methodik zur Angriffsmodellierung gesetzt. Auf Grundlage von identifizierten, charakteristischen Eigenschaften eines Angriffs werden die notwendigen Modellelemente abgeleitet. Es folgt der Kontext und Überblick der Methodik zur Angriffsmodellierung. Danach wird das genaue Vorgehen erläutert. Zum besseren Verständnis sind die Details der Methodik zur Angriffsmodellierung an einem zusammenhängenden Beispiel in Kapitel 6 zu finden. Die Evaluierung des Modells erfolgt anhand verschiedener Aspekte im nächsten Kapitel. Im Fokus stehen die analysierten Anforderungen an die Methodik. Die Arbeit endet mit einer Zusammenfassung und einem Ausblick.

2 Verwandte Arbeiten

In diesem Kapitel wird im Überblick der Unterschied zu vorhanden Arbeiten aufgezeigt. Es gibt bereits eine Vielzahl an Methoden und Verfahren zur Modellierung von Angriffen, die als Basis für die Entwicklung von sicherer Software herangezogen werden. Dabei kann ein Angriff aus verschiedenen Sichtweisen betrachtet werden. Die folgenden Ansätze stellen nur einen Auszug an Möglichkeiten dar.

Mithilfe der prozessorientierten Modellierung wird ein Angriff in Form eines Prozesses mit verschiedenen Phasen dargestellt.[63] Beispielsweise modelliert die Lockheed Martin Cyber Kill Chain einen Angriff anhand von sieben Phasen.[48] Der Angreifer muss jede Phase nacheinander erfolgreich abschließen, um sein Ziel zu erreichen.[63] Das Modell dient als Grundlage zur Analyse von Angriffen, sodass daraus effiziente Abwehrmechanismen abgeleitet werden können.[48] Allerdings ist dieses Modell sehr statisch auf Advanced Persistent Threats (APTs) und Angriffe mithilfe von Schadsoftware ausgelegt.[64] Die Methodik zur Angriffsmodellierung dieser Arbeit soll dagegen unabhängig von APTs sein. Außerdem soll die Modellierung von Angriffen in dynamischer Form erfolgen, sodass sich das Modell mit jeder Iteration ändert, bzw. der neuen Ausgangslage angepasst werden kann.[64]

Unabhängig von der zeitlichen Reihenfolge eines Angriffs bietet die graphbasierte Modellierung eine andere Darstellungs- und Analysemöglichkeit von Angriffen. Der Attack Graph [28] ist Gegenstand im Paper von Kaynar. Anhand von Attack Graphen, bestehend aus Knoten und Kanten, werden die möglichen Wege dargestellt, über die ein Angreifer in ein Netzwerk durch „Privilege Escalation“[1] eindringen kann. Allerdings steigt die Komplexität mit zunehmender Größe des Netzwerks, sodass die Übersichtlichkeit der Graphen verloren geht. Daher soll die zu entwickelnde Methodik der Angriffsmodellierung die Komplexität über Hierarchien regulieren, sodass es eine einfache, abstrakte obere Ebene zur Übersicht gibt. Anhand weiterer detailreicher Ebenen soll ein Angriff spezifiziert werden können. Außerdem soll die Methodik nicht auf das Angriffsziel „Privilege Escalation“ in Umgebung eines Netzwerks beschränkt sein.[28]

Eine spezielle Variante der graphbasierten Modellierung für Angriffe bilden Baumstrukturen. Schneier modelliert Angriffe in Form von sogenannten Attack Trees [47]. Diese optionsorientierte Modellierung stellt eine Untergruppe der graphbasierten Modellierung

[1] Das bedeutet, der Angreifer gewinnt höhere Rechte für ein System oder Netzwerk.[73]

dar. Die Wurzel des Baums entspricht dem Angriffsziel. Dieses Ziel kann über die Blätter und Zwischenknoten erreicht werden. Der Vorteil der einfachen Wiederverwendung solcher Bäume resultiert allerdings daraus, dass keine konkreten Informationen über das Target, dessen Umgebung und weitere Informationen über den Angreifer in die Modellierung mit einfließen. Dagegen soll die Methodik der Angriffsmodellierung in geeigneter Weise Modelle der Umgebung miteinbeziehen, sodass zum einen alle notwendigen Informationen für die Darstellung eines spezifischen Angriffs vorhanden sind. Zum anderen soll das Angriffsmodell mit verschiedenen Angreiferprofilen durchgeführt werden können. Des Weiteren ist der tatsächliche Angriffsablauf aus dem Attack Tree von Schneier nicht ersichtlich. Mehr Details zum Angriff als die Beschriftung bzw. Beschreibung der Knoten und Blätter wird nicht dargestellt. Mithilfe der zu entwickelnden Methodik soll der Ablauf einzelner Angriffsaktivitäten abbildbar sein. Letztendlich gibt es keine formale Spezifikation zur Darstellung der einzelnen Knoten bzw. Blätter des Baums. Die Inhalte sind als menschenlesbare Informationen formuliert. Dagegen gilt es in dieser Arbeit eine geeignete Grundlage zu schaffen für die Unterstützung einer strukturierten Modellierung mithilfe von Automatismen für die Angriffsmodellierung.

Die Multi-Level Modellierung stellt eine weitere Abbildungsmöglichkeit von Angriffen dar. Mithilfe von Tactics, Techniques and Procedures (TTP) kann ein Angriff über die Planung der Vorgehensweise des Angreifers abgebildet werden.[26] Dieser Ansatz kommt aus dem militärischen Bereich. Ausgehend von der sehr abstrakten Ebene (Taktik), über die Technik bis zu einer sehr detaillierten Abbildung (Procedure), wird der Detaillierungsgrad der Verhaltensbeschreibung des Akteurs in drei Ebenen aufgeteilt.[26] Auf dieser Basisstruktur werden Angriffe im sogenannten Adversarial Tactics, Techniques and Common Knowledge (ATT&CK)-Modell von The MITRE Corporation klassifiziert.[61] Anhand dieses Modells können über die Matrix entsprechende Techniken für eine bestimmte Taktik identifiziert werden. Zudem sind daraus verhaltensbasierte Angreiferszenarien und Angreiferprofile ableitbar. Allerdings sind Test und Verifikation für Verteidigungsmaßnahmen ausgelegt. Ziel ist die Anpassung und Verbesserung der Verteidigungsmaßnahmen. Außerdem ist die Handhabung derzeit noch sehr komplex und die Modellierung von Angriffen wird nicht grafisch dargestellt. Folglich liegt eine Schwierigkeit in der konkreten Anwendung von ATT&CK bzw. darin, das Modell auf einer konkreten Umgebung in der Praxis abzubilden.[31] Daher soll die Methodik dieser Arbeit grafisch abbildbar und nicht nur auf Angreifer-Techniken fokussiert sein. Es sollen Informationen verschiedener Abstraktionsstufen, wie die Prozessdarstellung des Angriffs, als auch der konkrete Exploit präsentiert werden können.[61] Es sollen mit geringem Aufwand Modelle der konkreten Umgebung eingebunden werden können. Somit ergibt sich ein Überblick des Zusammenhangs in einem für den Menschen intuitiven Format.

Zusammenfassend wird ersichtlich, dass die aktuelle Modellierung von Angriffen über Insellösungen realisiert ist. Abhängig vom Anwendungsfall existieren verschiedenste Ansätze, einen Angriff abzubilden, wie die prozess-, graph-, optionsbasierte oder Multi-Level Modellierung. Jeder Ansatz fokussiert einen anderen Aspekt eines Angriffs.

Es gibt kein generelles Angriffsmodell, das die verschiedenen Ansätze vereinigt. Auf diese Thematik macht bereits Adepu und Mathur in [1] aufmerksam. Im Hinblick auf physikalische Systeme versucht er das Problem mit einem generischen Angreifermodell und Angriffsmodell zu lösen, indem sowohl Security, als auch Safety Aspekte betrachtet werden.[1] Allerdings ist das Modell nicht auf die systematische bzw. strategische Entwicklung und den Aufbau eines Angriffs aus Sichtweise des Angreifers ausgelegt.[1] Vielmehr dient das generische Modell von Adepu und Mathur dazu, Abwehrmaßnahmen für physikalische Systeme in ihrer Wirkung besser zu verstehen.[1] Die zu entwickelnde Methodik zur Angriffsmodellierung soll dagegen einen allgemeinen Ansatz darstellen, der nicht nur auf physikalische Systeme fokussiert ist. Für eine geeignete Unterstützung von Security-Tests soll über ein generisches Angriffsmodell ein Angriff mit allen notwendigen charakteristischen Informationen präsentiert werden können.

3 Stand der Wissenschaft und der Technik

Für das weitere Vorgehen ist zunächst eine Reihe an relevanten Begrifflichkeiten genauer zu betrachten, denn einige Begriffsdefinitionen unterscheiden sich je nach spezifischem Anwendungsbereich in der IT-Sicherheit. Mit einer kurzen Erläuterung der wesentlichen Grundlagen schließt das Kapitel ab.

3.1 Begrifflichkeiten

Anhand eines beispielhaften Angriffs sollen die notwendigen Definitionen erläutert werden. Das folgende Szenario basiert auf dem Beispielangriff in [18] bezüglich Cross Site Scripting (XSS)[1]. Das Unternehmen KleiderExample GmbH bietet einen Onlineshop als Webanwendung an. Diese Anwendung ist dynamisch aufgebaut und wird über einen Webserver bereitgestellt. Der Webserver entspricht einer Cyber-Enabled Capability.

Definition 3.1 *Unter einer Cyber-Enabled Capability ist auf Grundlage von [71] jegliche Form einer softwarefähigen Technologie zu verstehen. Dabei sind sowohl alle ihre Eigenschaften, inklusive Hardware, Mechanismen für Interaktion, Sensorik und Aktorik zu beachten, als auch sämtliche Informationen zur Entwicklung dieser Technologie in allen Phasen, z. B. Definition, Entwurf, Implementierung, Test, Deployment, Wartung und Entsorgung.[71]*

Der Server ist unter anderem so konfiguriert, dass bei Nichtauffindung einer angeforderten Website eine Fehlermeldung an den Nutzer zurückgesendet wird. Dabei enthält die Fehlermeldung automatisch die vollständige URL, die für diese Meldung verantwortlich war. Die Konfiguration stellt eine XSS-Schwachstelle dar. Das bedeutet, dass der Webserver die vom Nutzer eingegebenen Daten ohne Filterung wieder zurück an den Browser sendet, der die Daten anschließend weiterverarbeitet.

[1] Für mehr Details siehe [18] Kapitel 3 und [29] Kapitel 2.

Definition 3.2 *Eine Schwachstelle (engl. Weakness) liegt nach [71, 53, 18] vor, wenn ein (fehlerhafter) Zustand existiert, der unter bestimmten Voraussetzungen eine Cyber-Enabled Capability angreifbar macht. Angreifbar bedeutet, dass jemand unerlaubt Zugriff auf eine Cyber-Enabled Capability bekommt, oder unerwünschte Funktionen ausführen kann. Dabei lassen sich die Schwachstellen einteilen in fehlerhafte Konzeption, Algorithmen, Implementierung, Konfiguration, Betrieb und technische oder organisatorische Mängel.*

Die Benutzerin Alice hat eine Vorliebe für Kleider und besitzt einen Rechner mit einem Browser. Im Browser von Alice ist JavaScript aktiviert, sodass alle, in einer Website eingebetteten JavaScript-Befehle automatisch ausgeführt werden. Dieser Umstand stellt ebenfalls eine Schwachstelle dar. Ein Angreifer hat als Angriffsziel die Passwörter auf der Festplatte von Alice aufgefasst.

Definition 3.3 *Die folgende Definition basiert auf [5, 45]. Der Angreifer (engl. Attacker) ist die verantwortliche Person (bzw. Personengruppe) für einen Angriff auf ein Ziel(-system). Dabei lässt sich ein Angreifer unter diversen Kriterien betrachten z. B. interner oder externer Angreifer, Einzelperson oder Teil einer Gruppe. Das BSI unterscheidet z. B. folgende Gruppen: Cyberaktivisten, Cyber-Kriminelle, Industriespione, Staatliche Nachrichtendienste, Staatliche Akteure im Cyber-War, Cyber-Terroristen, Skript-Kiddies, Innentäter und IT-Sicherheitsforscher.*

Definition 3.4 *Die Grundlagen für diese Definition stammen aus [39, 77, 53, 75, 18]. Angriff (engl. Attack) ist ein Oberbegriff und besteht aus mindestens einer Aktion, um ein bestimmtes strategisches Ziel zu erreichen. Das strategische Ziel des Angreifers ist in jeglicher Hinsicht eine unerlaubte bzw. unberechtigte Handlung, wie die Schädigung eines Zielobjekts (engl. Target), oder ein nicht autorisierter Zugriff(-sversuch) auf ein Zielobjekt. Das bedeutet gleichzeitig, dass Schutzziele des Targets gebrochen werden, wie z. B. Informationsvertraulichkeit (engl. Confidentiality), Datenintegrität (engl. Integrity), Verfügbarkeit (engl. Availability), Authentizität (engl. Authorization), Zuverlässigkeit (engl. Reliability), Verbindlichkeit (engl. Non-repudiation) oder Zurechenbarkeit (engl. Accountability).*

In diesem Beispiel führt der Angreifer einen XSS-Angriff aus. Der Angreifer versucht über die XSS-Schwachstelle des Webservers von KleiderExample GmbH Schadcode an Alice zu schicken. Die Aufgabe des Schadcodes ist es, die Passwörter von Alice auf der Festplatte ausfindig zu machen und an den Angreifer zu übermitteln. Der Schadcode ist ein Beispiel für einen Exploit.

Definition 3.5 *Die Definition basiert auf den Arbeiten von [71, 53, 18]. Exploit ist der Oberbegriff für diverse Handlungsmittel wie Befehlsfolgen, Funktionalitäten oder der Input für eine Cyber-Enabled Capability, um unerlaubte Handlungen (Angriffe) auszuführen. Durch die Existenz eines Exploits wird eine Schwachstelle zu einer konkreten Verwundbarkeit. Ein Angreifer verwendet einen oder mehrere Exploits, um mindestens eine Verwundbarkeit einer Cyber-Enabled Capability auszunutzen.*

Sobald der Schadcode als Exploit existiert, wird die XSS-Schwachstelle zu einer konkreten Verwundbarkeit der Cyber-Enabled Capability.

Definition 3.6 *Die Definition für Verwundbarkeit (engl. Vulnerability) ist aus [71, 49] entnommen. Eine Vulnerability stellt eine bestimmte Schwachstelle dar, wie eine Sicherheitslücke oder Fehlfunktion, die direkt von einem Angreifer über vorhandene Exploits ausgenutzt werden kann. Bestimmte Stellen einer Cyber-Enabled Capability bieten einen unerwünschten Zugriff auf die existierende Verwundbarkeit an. Der Angreifer versucht über die Vulnerability die Cyber-Enabled Capability zu seinem Vorteil auszunutzen.*

Schließlich gehen wir davon aus, dass der Angreifer die Vorliebe von Alice bereits kennt. Die Informationen über Alice hat der Angreifer über vorherige Angriffshandlungen erhalten, wie beispielsweise über Social Media-Plattformen. Diese stellen einen Zugangspunkt für den Angreifer dar.

Definition 3.7 *Die folgende Definition liegt [3] zugrunde. Zugangspunkte (engl. Access Point) sind Stellen, an denen ein Benutzer (inkl. Angreifer) Zugriff zu Informationen erhält, die nicht für die Absicht eines Angriffs bestimmt sind. Diese Punkte dienen der Informationsbeschaffung und werden diesbezüglich in geeigneter Weise verwendet bzw. analysiert. Insbesondere in der Anfangsphase des Angriffs sind Zugangspunkte das Ziel eines Angreifers, um Informationen über das Target zu erhalten. Suchmaschinen wie Google, der HTTP Header oder jegliche Art von Nachrichten sind Beispiele für potentielle Zugangspunkte. Die Stellen können sowohl Bestandteil der Cyber-Enabled Capability sein, als auch unabhängig davon existieren.*

Um sich Zugang zu den Passwörtern auf der Festplatte von Alice zu beschaffen, durchläuft der Angreifer einen bestimmten Weg bis zu diesem Angriffspunkt. Die Route des Angreifers wird als Angriffsvektor bezeichnet.

Definition 3.8 *Die Definition basiert auf [43, 56]. Unter Angriffsvektor (engl. Attack Vector) ist die genaue Route bzw. der Weg gemeint, um sich Zugang zu dem Zielobjekt zu verschaffen. Dabei versucht der Angreifer mithilfe einer Kombination von Techniken, über einen Zugangs- oder Angriffspunkt Zugriff auf ein Target zu erhalten. Der Angriffsvektor wird durch die verfügbare Angriffsoberfläche, dem Angreifer und seinem Wissen über das Target und dessen Umgebung bestimmt.*

Über die verfügbare Angriffsoberfläche versucht der Angreifer Zugriff auf das Target zu erhalten, um daraufhin eine XSS-Vulnerability auszunutzen.

Definition 3.9 *Die Angriffsoberfläche (engl. Attack Surface) ist auf Basis von [24, 5] die Summe an Angriffspunkten des Targets und dessen Umgebung, die zu einem bestimmten Zeitpunkt vorhanden sind. Ein Angriffspunkt (engl. Attack Point) entspricht einer Stelle, um sich Zugriff zu einer Cyber-Enabled Capability oder Elementen davon zu verschaffen. Dazu zählen insbesondere Interfaces, Services, Protokolle oder der Programmcode. Zunächst versucht ein Angreifer Angriffspunkte zu finden. Das bloße Analysieren von bestimmten Punkten und das Gewinnen von Informationen daraus entspricht zunächst einem Zugangspunkt. Ein Zugangspunkt ist nicht zwangsläufig ein Angriffspunkt (Bestandteil der Angriffsoberfläche). Sobald der (Zugangs-)Punkt beeinflussbar ist, um Zugriff auf eine Cyber-Enabled Capability zu erhalten, wie durch Modifikation der Stelle, wird der Zugangspunkt als Angriffspunkt bezeichnet. Somit bietet ein Angriffspunkt die Möglichkeit, auf existierende Vulnerabilities zuzugreifen.*

Ausgehend von der vorhandenen Angriffsoberfläche, seinem aktuellen Wissen über Alice und dem Angriffsziel, stellt der Angreifer Alice eine glaubhafte Website von KleiderExample GmbH zur Verfügung, die unter anderem einen bösartigen Link aufweist. Daraufhin nutzt Alice den schadhaften Link auf der angeblichen Webanwendung von KleiderExample GmbH in ihrem Browser. Folglich wird von dem Server von KleiderExample GmbH eine spezielle Webseite abgefragt. Allerdings existiert diese Seite nicht, sodass Alice eine Fehlermeldung bekommt. Die Fehlermeldung wird aufgrund der Konfiguration dynamisch erstellt und enthält den scriptbasierten Schadcode des Angreifers. Der bösartige JavaScript-Schadcode wird automatisch im Browser von Alice ausgeführt, sodass die gefundenen Passwörter an den Angreifer zurückgesendet werden. Wir gehen an dieser Stelle davon aus, dass der Angriff erfolgreich war und der Angreifer die Passwörter von Alice erhalten hat.

Mithilfe der Abbildung 3.1 soll der Zusammenhang der Begrifflichkeiten für diese Arbeit noch einmal verdeutlicht werden. Eine Cyber-Enabled Capability kann Schwachstellen enthalten. Sobald ein Exploit für eine Schwachstelle existiert, wird diese Weakness zu einer Verwundbarkeit, die über einen Angriffspunkt zur Verfügung steht. Die zu einem bestimmten Zeitpunkt existierenden Angriffspunkte bilden das Attack Surface

der Cyber-Enabled Capability bzw. des Targets und dessen Umgebung. Ein Angriff wird von einem Angreifer durchgeführt, der zunächst die Angriffspunkte identifizieren bzw. analysieren muss. In diesem Fall werden die Angriffspunkte zunächst als Zugangspunkte bezeichnet. Der Angreifer verwendet Zugangspunkte, um sich Informationen für seinen Angriff zu beschaffen. Die Zugangspunkte können unabhängig vom Target sein, oder Bestandteil davon. Ein Angriff kann über Angriffstechniken präsentiert werden. In den Techniken werden konkrete Exploits für die Ausnutzung von Vulnerabilities verwendet. Der genaue Weg bis zu einem Angriffspunkt kann als Angriffsvektor interpretiert werden. Der Angreifer besitzt ein strategisches Ziel, das er mit dem Angriff erreichen möchte. Das Ziel beeinflusst daher die Vorgehensweise des Angriffs und dessen Bestandteile.

Zum Schutz vor XSS-Angriffen sind auf beiden Seiten Abwehrmaßnahmen möglich. Alice kann beispielsweise die Ausführung von Scriptsprachen im Browser deaktivieren. Für Unternehmen empfiehlt das BSI einen geeigneten Sicherheitsprozess[2], um den notwendigen Schutzbedarf und das gesamten Sicherheitsregelwerk für IT-Infrastrukturen festzulegen.[18, 53]

Definition 3.10 *Die Definition für den Schutzbedarf ist von [18, 53] Der Schutzbedarf entspricht dem Bedarf, der notwendig ist, damit die Vertraulichkeit, die Integrität und Verfügbarkeit eines IT-Systems, von Geschäftsprozessen oder der damit verbundenen Informationen nicht gebrochen werden. Die Höhe des Bedarfs kann in drei Kategorien aufgeteilt werden: normal, hoch, sehr hoch. Die Größe des Schutzbedarfs ist Grundlage für weitere Aktionen im Sicherheitsprozess. Im Kontext dieser Arbeit wird die Definition auf alle möglichen Schutzziele erweitert, die bereits in Abschnitt 3.4 aufgeführt sind.*

Wir gehen von einem hohen Schutzbedarf für den Webserver aus, da KleiderExample GmbH ein kleines Unternehmen darstellt. Ein möglicher finanzieller Schaden von 100.000 Euro ist existenzbedrohend.

Definition 3.11 *Die Definition für Security Policy (Sicherheitsregelwerk) stammt von [18]. Diese fasst die gesamte Sicherheitsstrategie zusammen. Sie besteht aus allen notwendigen Maßnahmen, um einen zuvor identifizierten Schutzbedarf abzudecken.*

Unter anderem gibt es zahlreiche Nachschlagewerke mit wichtigen allgemeinen Sicherheitsgrundfunktionen, die dem Schutzbedarf in vielen Bereichen nachkommt.[18] Auf Seiten des Unternehmens kann die Integration von Eingabefiltern als sichere Programmierung dafür sorgen, dass z. B. nur ein explizit definiertet Input zugelassen wird.

[2] Für weitere Details siehe Kapitel 3 in [51].

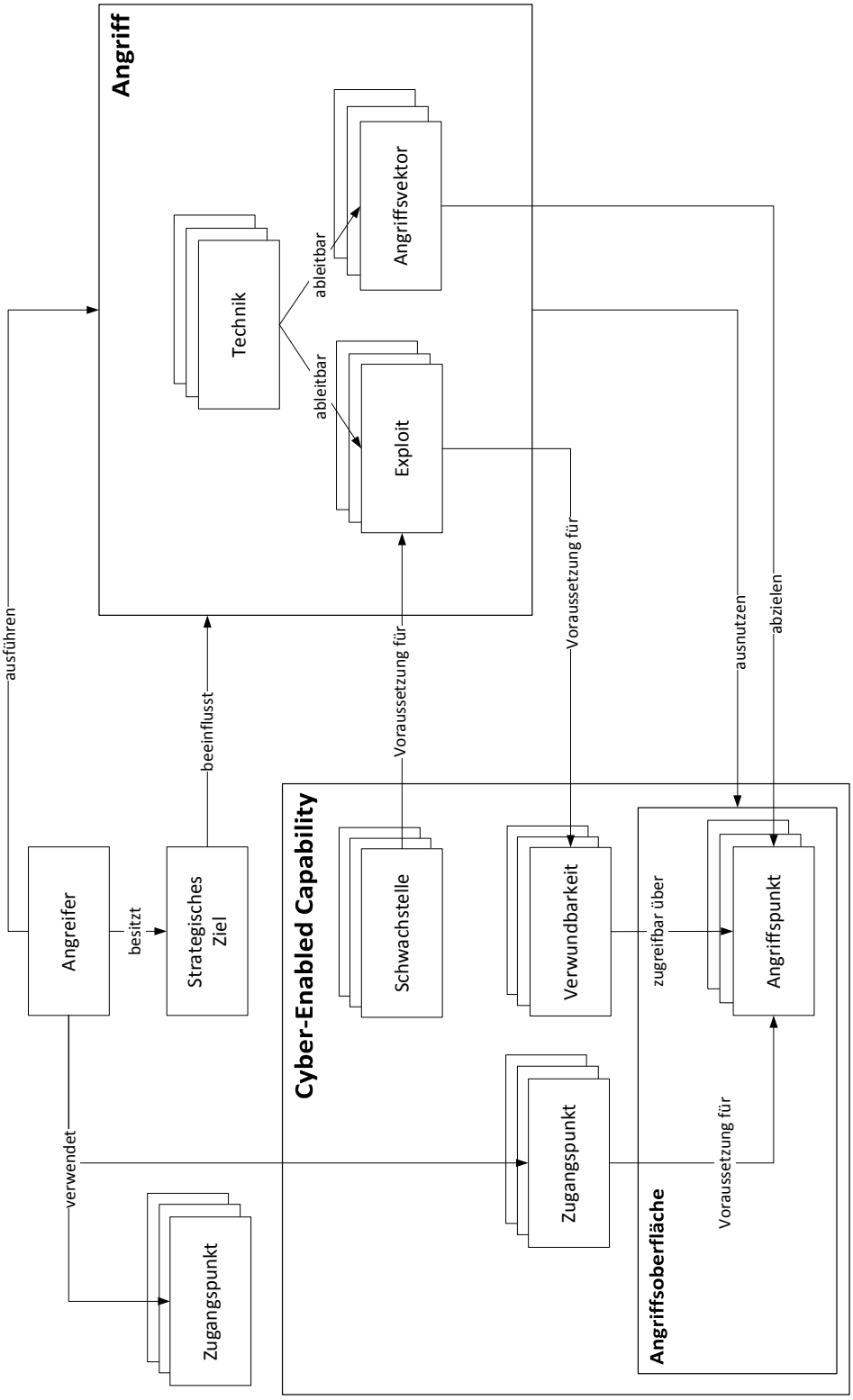

Abbildung 3.1: Zusammenhang der Begrifflichkeiten

3.2 Modellbasierte Entwicklung

Im Hinblick auf die Gewährleistung von IT-Sicherheit bei der Entwicklung von Systemen ist der weitverbreitete Ansatz „penetrate and patch“ nicht die beste Strategie.[27] In diesem Fall werden z. B. Verwundbarkeiten erst sehr spät im Security Development Lifecycle (SDL) entdeckt, was hohe Kosten für die Beseitigung zur Folge hat.[27]

Ein alternativer Ansatz dazu ist die modellbasierte Entwicklung von Software. Dabei sind Modelle ein zentraler Bestandteil der Entwicklung.[6] Das bedeutet, dass sich die Entwickler zunächst auf das Modell konzentrieren und es gegebenenfalls ändern müssen. Erst danach wird das tatsächliche Design des Systems angepasst.[20]

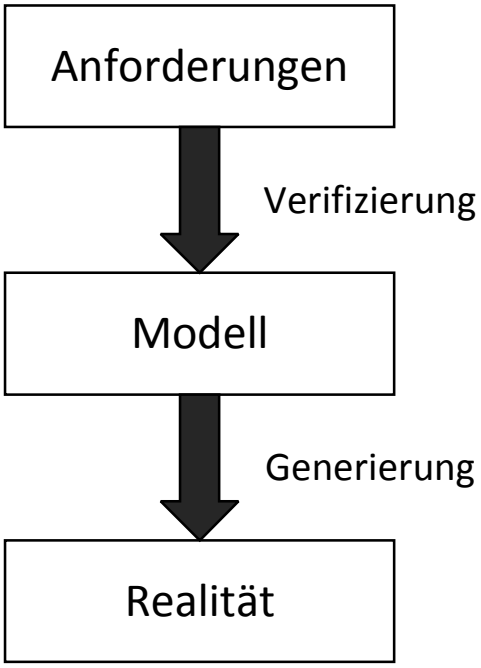

Abbildung 3.2: Prinzip der modellbasierten Entwicklung auf Basis von [27]

Das Prinzip dieses Ansatzes ist in Abbildung 3.2 dargestellt. Zunächst muss ein Modell, basierend auf bestimmten Anforderungen entwickelt werden. Die Modellerstellung kann mithilfe von Werkzeugen unterstützt werden.[20] Das Modell dient als Grundlage zur Ableitung der tatsächlichen Implementierung, beispielsweise in Form von Programmcode. Im Idealfall können zudem Tests anhand des Modells generiert werden.[27]

Jedoch besitzen Modelle den Nachteil, dass sie schnell veralten und somit nicht mehr konsistent mit dem Programmcode sind.[20] Beispielsweise hat zunächst das Schließen einer Sicherheitslücke eine höhere Priorität, als die Aktualisierung des zugehörigen Modells. Allerdings liegt der Fokus in dieser Arbeit nicht auf der Modellierung von Programmcode, sondern auf der Entwicklung einer geeigneten Methodik zur Angriffsmodellierung. Dabei steht die gesamtheitliche Betrachtung eines Angriffs, inklusive Abläufe, Abhängigkeiten von verschiedenen Komponenten der Umgebung und die Darstellung von Techniken im Vordergrund.

Der wesentliche Vorteil von einem Modell ist, dass diese für den Menschen allgemein einfacher zu verstehen sind bzw. ein einfacherer Umgang mit der Komplexität möglich ist, da Modelle etwas bildhaft veranschaulichen, statt der Darstellung in reinem Text.[27]

Auf Grundlage von [27, 19] bieten Modelle folgende weitere Vorteile:

- Einfacher Überblick über das System und dessen Umfeld
- Frühzeitige Beachtung von Sicherheit im SDL
- Nachverfolgbarkeit von Security
- Beweisbarkeit von Security auf Ebene des Designs, z. B. anhand von Modellanalysen zur Vollständigkeit der geforderten Sicherheitsanforderungen
- Erhöhung der Entwicklungsgeschwindigkeit, z. B. durch Generierung von Tests aus dem vorhandenen Modell
- Erhöhung der Code-Qualität
- Einheitlichkeit der Dokumentation (im Vergleich zu Prosatext)
- Einfache Beurteilung/Abschätzung der Grafiken
- Vergleich mit anderen Modellen möglich

3.3 Systemmodelle

Im Bereich der Industrie hat sich für die Modellierung von IT-Systemen und Strukturen insbesondere Unified Modeling Language (UML) von der Object Management Group als modellbasierter Ansatz durchgesetzt.[27] Vorwiegend wird die standardisierte Modellierungssprache zur Kommunikation zwischen Teamkollegen, Stakeholdern und neuer Projektteilnehmer verwendet.[19] Allerdings erfolgt die Modellierung meist in Form von informellen Skizzen.[20] Über die enthaltenen Notationen stellt UML ein Framework zur Modellierung einer allgemeinen Sprache im Bereich der Softwareentwicklung dar.[27] Anhand der verschiedenen Modelltypen und Abstraktionsgrade erhält man einen einfachen Überblick über das System und dessen Umfeld.[19]

UML unterstützt die verschiedenen Teilbereiche bei der Softwareentwicklung über unterschiedliche Modelltypen [27], wie beispielsweise:

- **Use Case Diagramm:** Geforderte Funktionalitäten eines Systems
- **Aktivitätsdiagramm:** Kontrollfluss zwischen Systemkomponenten
- **Klassendiagramm:** Statische Klassenstruktur eines Systems
- **Sequenzdiagramm:** Interaktion (Nachrichtenaustausch) zwischen Komponenten
- **Zustandsdiagramm:** Dynamisches Verhalten von Objekten oder Komponenten
- **Verteilungsdiagramm:** Verteilung der Komponenten in der physischen Umgebung

Dadurch lässt sich ein System aus verschiedenen Sichtweisen abbilden, sodass das gesamte System, dessen Eigenschaften und Abhängigkeiten im Überblick sichtbar sind. UML bietet den Vorteil, dass diese Modellierungssprache weitverbreitet und standardisiert ist. Sowohl in der Industrie, als auch in den Vorlesungen vieler Universitäten kommt die intuitiv verständliche und einfach zu erlernende Sprache zum Einsatz. Grund dafür ist der geringere Aufwand für die Modellerstellung, deren einfache Aneignung und die relativ präzise Definition der Syntax und Semantik, im Vergleich zu anderen Sprachen.[27, 19]

Schließlich ist UML nicht auf objekt-orientierte Systeme beschränkt und weist zudem Mechanismen für Erweiterungen auf.[27] Die Erweiterungsmechanismen bieten zusammen als sogenanntes „Profil“ die Möglichkeit, UML mit neuen Klassen und Attributen zu erweitern, ohne dabei das grundlegende Metamodell von UML zu verändern.[27] Diese Offenheit bietet die Möglichkeit, dass UML zusammen mit gegebenenfalls mehreren (verschiedenen) Erweiterungsprofilen genutzt werden kann.

Speziell bei der Entwicklung von sicherheitsrelevanten Systemen und im Umgang mit Sicherheitsaspekten entwickelte Jürjens die Erweiterung Standard Unified Modeling Language extention mechanism for secure system development (UMLsec) [27] für UML. Anhand dieser Erweiterung können beispielsweise auf Design-Ebene Sicherheitsschwachstellen der UML-Spezifikation identifiziert und entsprechend bewertet werden.[27] Dabei basiert die Nutzung auf etablierten Security Engineering Regeln.[27]

Der Fokus von UMLsec liegt auf der Modellierung von Sicherheitsanforderungen auf Design-Ebene von vernetzten Systemen und deren Kommunikation mit der Umgebung. Es unterstützt die Integration von sicherheitsrelevanten Informationen, wie Sicherheitsanforderungen, -annahmen und -polices in die verschiedenen Modelltypen von UML.[27] Somit wird die Sicherheit von sicherheitsrelevanten Systemen aus verschiedenen Sichtweisen auf das System betrachtet, wie z. B. auf logischer oder physikalischer Ebene.

Dabei wird in UMLsec ein Angreifer mit bestimmten Fähigkeiten betrachtet.[3] Auf Basis des Dolev-Yao-Modells werden die Interaktionen des Angreifers mit der Umgebung präsentiert.[27]

Bei der Modellierung sind gewisse Regeln einzuhalten.[27] Ein fehlerhaftes Modell kann mithilfe von Werkzeugen automatisch erkannt werden. Auf Basis einer formalen Semantik ist eine formale Verifikation von Sicherheitseigenschaften im UMLsec-Modell mit Unterstützung von Werkzeugen möglich.[27] Allerdings ist in UMLsec keine automatisierte bzw. formale Unterstützung für die Identifikation von sicherheitsrelevanten Informationen vorgesehen. Der Schwerpunkt von UMLsec liegt auf der Modellierung von Sicherheitsaspekten in Bezug auf ein bestimmtes System auf Design-Ebene. Angriffe werden nur indirekt, über die Modellierung eines Angreifers und dessen Bedrohungen abgebildet.

3.4 Formale Konzepte

Die Anwendung von (richtig) formalen Modellen ist in der praktischen Umgebung kaum vorzufinden. Sie sind unattraktiv, da entsprechende Schulungen und die Nutzung mit hohen Kosten verbunden sind.[20, 27] Zudem stützen sich viele formale Methoden auf begrenzte Metamodelle.[13]

Formale Methoden basieren im Kontext des Software Engineerings auf mathematischen Konzepten, wobei die Abstraktion ein wesentlicher Bestandteil ist.[12] Unter Abstraktion ist das „Ableiten oder Herausheben des unter einem bestimmten Gesichtspunk Wesentlichen/Charakteristischen/Gesetzmäßigen aus einer Menge von Individuen“[22] zu verstehen. Nach [22] hilft die Abstraktion im Kontext der Modellierung bei der Handhabung von Komplexität und zum besseren Verständnis durch:

- Veranschaulichung von umfassenden Zusammenhängen
- Abbildung von Details
- Konstruktion einer Verknüpfung von verschiedenen Sichtweisen aus Überblick und Detail

[3] Weitere Details dazu sind in [27] zu finden.

Dabei wird in [22] nach vier Abstraktionsarten unterschieden:

- **Generalisierung:** Mehrere Individuen werden über ein umfassendes Individuum repräsentiert, das die gleichartigen Eigenschaften der untergeordneten Einheiten abbildet.

- **Komposition:** Zusammengehörige Individuen werden miteinander zu einer Einheit verknüpft.

- **Benutzung:** Ein Individuum nutzt die Dienstleistung eines oder mehrerer anderer Individuen, um einen Dienst (anderen Individuen) anzubieten. (Schichtenbildung)

- **Klassifizierung:** Anhand von Klassen oder Typen werden Gemeinsamkeiten von Individuen zusammengefasst.

Somit lässt sich Abstraktion über Klassifizierung, Benutzung, Komposition und Generalisierung in Modelle integrieren. Insbesondere in den Bereichen Spezifikation und Verifikation des Software Engineerings finden sich formale Konzepte wieder.[12] Unter Spezifikation versteht man in [12] die Beschreibung eines Systems anhand seiner Eigenschaften, z. B. dessen Funktionsverhalten oder die Performance Charakteristik des Systems. Mithilfe einer mathematischen Syntax und Semantik wird eine formale Spezifikation modelliert. Die verschiedenen Aspekte eines Systems können mit unterschiedlichen Spezifikationssprachen dargestellt werden oder über Aspekte der Architektur und Security Policies. Ein Vorteil von formalen Konzepten ist somit, dass man durch die Anwendung dieser Methoden, wie beispielsweise der formalen Spezifikation, ein grundlegendes Verständnis über das Objekt gewinnt, das man genauer bestimmt.[12] Außerdem sorgt die formale Notation für die Unterbindung von Mehrdeutigkeiten.[13]

Die Verifikation ist der „Nachweis der Richtigkeit einer [. . .] Aussage“[40]. Das bedeutet, dass ein formales Modell, z. B. in Form einer formalen Spezifikation, schematisch analysiert und bewiesen wird.[12] Gleichzeitig bietet es die Grundlage zur automatischen Generierung von Testfällen oder Code.[12] Daher sind formale Modelle aus theoretischer Sicht, aufgrund Konsistenz und Beweisbarkeit wünschenswert, doch in der Praxis lassen sich diese Vorteile nicht einfach umsetzen. Als „gutes“ formales Modell wird ein generelles Modell bezeichnet, das möglichst allumfassend jegliches System abbilden kann. Gleichzeitig ist die Nachfrage für eine genaue und detailreiche Beschreibung vorhanden. Diese gegensätzlichen Anforderungen bilden die Herausforderung bei der Entwicklung von formalen Methoden. Daher erfolgt die Nutzung von formalen Konzepten meist nur als leichte Version und Ergänzung zu anderen Methoden, statt einer vollständigen und strikten formalen Anwendung.[13]

3.5 Quantitative und Qualitative Methoden

Wie bereits in der Einleitung kurz erwähnt, spielen für die Praxis Security-Analysen eine wesentliche Rolle. Z. B. sind Bedrohungs- und Risikoanalysen richtungsweisend für die Identifikation des Schutzbedarfs.[53, 18] Die Risikoanalyse ist notwendig, da eine vollständige Verteidigung gegen alle zuvor identifizierten Bedrohungen nicht möglich ist. Zum einen ist es aus wirtschaftlicher Sicht nicht umzusetzen, zum anderen gibt es keine hundertprozentige Sicherheit.[53]

OWASP Risk Rating Methode Die nachfolgenden Informationen zu der Risk Rating Methode von OWASP sind aus [21, 18] entnommen. Für die Praxis bietet die Risikoanalyse von OWASP eine systematische Identifikation und Bewertung von Risiken an, um anschließend mit vertretbarem Aufwand entsprechende Gegenmaßnahmen umzusetzen. Mit anderen Worten, auf Basis der Risikoanalyse können Verteidigungsmaßnahmen gegen mögliche Angriffe bestimmt werden, die effizient und ressourcenschonend sind. Aus diesem Grund ist der Aspekt Wahrscheinlichkeit ein wesentlicher Bestandteil von Risikoanalysen im Kontext der praktischen Umsetzung von Security. Sie bildet zusammen mit dem Schadensausmaß das Risiko der OWASP Risk Rating Methode. OWASP definiert sowohl Faktoren zur Abschätzung der Wahrscheinlichkeit (Threat Agent Factors, Vulnerability Factors), als auch Faktoren zur Abschätzung des Schadensausmaßes (Technical Impact Factors, Business Impact Factors). Das Gesamtrisiko ergibt sich aus einer Matrix, die in fünf Schweregrade unterteilt ist: „note“, „low“, „medium“, „high“ und „critical“. Voraussetzung für die OWASP-Risikoanalyse ist, dass zuvor mindestens eine Bedrohung identifiziert wurde. Deren Risiko wird anschließend mithilfe der Risikoanalyse ermittelt. Dabei wird die Bedrohung nicht auf Basis von Messwerten priorisiert, sondern nach subjektiver Bewertung (0 bis 9) im Rahmen der gegebenen Faktoren von OWASP. Beispielsweise ergibt sich der Faktor Threat Agent aus den folgenden Merkmalen, wobei jede mit einer entsprechenden Gewichtung verknüpft ist:

- **Skill Level:** Der Angreifer besitzt z. B. keine technischen Fertigkeiten (1) oder er ist ein sachkundiger Benutzter (5) oder er ist im Besitz von Security Penetration Qualifikationen (9).
- **Motive:** Der Angreifer erwartet z. B. keine Belohnung und ist daher kaum motiviert (1) oder der Angreifer erhält eine hohe Entlohnung für die Ausnutzung einer Vulnerability (9).
- **Opportunity:** Der Angreifer benötigt z. B. teure Ressourcen, um den Angriff durchzuführen (0) oder es sind keine speziellen Ressourcen dazu notwendig (9).
- **Size:** Der Angreifer stammt z. B. aus einer kleinen Gruppe von Entwicklern bzw. Administratoren (2) oder er ist einer von vielen anonymen Internetbenutzern (9).

Somit stellt die OWASP Risk Rating Methode sowohl eine qualitative, als auch quantitative Methode dar.

Definition 3.12 *Die folgende Definition basiert auf Informationen von [83]. Quantitative Ansätze befassen sich mit großen Mengen an zahlenmäßigen Ausprägungen eines oder mehrerer charakteristischer Eigenschaften. Dabei werden vorhandene Messwerte zueinander in Verbindung gesetzt. Für eine gute Vergleichbarkeit ist darauf zu achten, dass die Bedingungen bei der Entstehung der konkreten Messwerte möglichst einheitlich sind. Letztendlich soll über Datenreduktion ein Verhalten und deren Zusammenhänge im Überblick präsentiert und prognostiziert werden können. Das bedeutet, quantitative Ergebnisse sind numerische Werte, bzw. anhand von Zahlen darstellbar.*

Definition 3.13 *Für diesen Abschnitt sind die Informationen aus [83] entnommen. Qualitativ bedeutet, dass der Schwerpunkt auf Abbildung der Wirklichkeit mit einem tiefen Informationsgehalt bzw. hoher Inhaltsvalidität liegt, statt über eine zahlenmäßige Ausprägung. Zugunsten einer großen Offenheit und Flexibilität gibt es dementsprechend nur eine grobe Richtung und wenig Einschränkungen zur Vorgehensweise bei qualitativen Methoden. Der Wissensaufbau (Ergebnisse von qualitativen Ansätzen) erfolgt schrittweise und wird mit der Zeit aktualisiert. Letztendlich dienen diese Methoden dazu, ein genaues Bild und Verständnis zu erhalten, indem ein identifiziertes Verhalten anhand von möglichen Ursachen nachvollziehbar wird.*

Aufgrund der durchzuführenden Abbildung der Bedrohung auf verschiedene Faktoren (inklusive deren Gewichtung) lässt sich die OWASP Risk Rating Methode zum einen den qualitativen Methoden zuordnen. Zum anderen sorgt das Prinzip der Gewichtung für die Verknüpfung zu quantitativen Methoden. Die Werte der Skala von 0 bis 9 bestimmen über eine Matrix das Endergebnis. Das Endergebnis entspricht dem Gesamtrisiko, das wiederum einen Wert der folgenden Schweregrade besitzt: „note“, „low“, „medium“, „high“ oder „critical“. Mithilfe dieser definierten Werte ist das Gesamtrisiko vergleichbar, was das Ziel einer quantitativen Methode darstellt.[83]

3.6 Datenbasis zur Modellierung

Unabhängig von dem Modellierungsansatz sind Daten notwendig, die die Grundlage des Modells bilden, bzw. den Input für die Modellierung liefern. Es existieren öffentliche Datenbanken, die einen strukturierten Überblick an Informationen zum Themenbereich von Angriffen geben.

The MITRE Corporation stellt eine Aufzählung an bekannten Schwachstellen (Common Weakness Enumeration (CWE)) [69], Verwundbarkeiten (Common Vulnerabilities and Exposures (CVE)) [68] und Angriffsmustern (Common Attack Pattern Enumeration and Classification (CAPEC)) [67] zur Verfügung.

Die nachfolgende Erläuterung basiert auf den Informationen der Website von CWE [69]. Über die öffentliche Datenbank CWE wird ein strukturierter Überblick von allgemeinen Schwachstellen in Software dargestellt. Die Datenbank soll Informationen zur Identifikation, Milderung und Prävention von Schwachstellen für diejenigen bereitstellen, die diese Inhalte benötigen. Ziel ist es, anhand einer einheitlichen Sprache Schwachstellen zu beschreiben, um diese in der Praxis identifizieren zu können. Beispielsweise ist unter der CWE-Nummer 89 [72] die Schwachstelle bezüglich Structured Query Language Injection (SQLI) zu finden. Dort erhält man zum einen z. B. die allgemeine Information, dass der Vulnerability ein Datenbank-Server als Technologie zugeordnet ist. Zum anderen erhält man die Information, dass die möglichen Auswirkungen eines erfolgreichen Angriffs über SQLI in den Bereichen Vertraulichkeit, Integrität und Zugriffskontrolle zu erwarten sind. Mithilfe von konkreten Beispielen wird versucht, die allgemeine Schwachstelle einer SQLI in „CWE-89“ zu veranschaulichen, da das Einschleusen der Befehle bezüglich Structured Query Language (SQL) von den verwendeten Programmiersprachen abhängig ist, z. B. C#, PHP oder Perl. Ein Schwerpunkt ist die Verteidigung: The MITRE Corporation präsentiert eine Auflistung an Verteidigungsmaßnahmen, geordnet nach Phasen des SDL und verschiedene Methoden zur Entdeckung von SQLI-Schwachstellen.

Die folgenden Informationen sind aus der CVE-Datenbank [68] entnommen. The MITRE Corporation stellt die Datenbank CVE zur Verfügung, die eine Vielzahl an konkreten Vulnerabilities mit einer entsprechenden CVE-ID, kurzer Beschreibung, Referenzen und gegebenenfalls weiteren Details beinhaltet. Diese Informationen können Angriffe genauer spezifizieren. Z. B. wird über die Erläuterung und die Details von „CVE-2019-14313“ ersichtlich, dass ein Angreifer über einen entfernten Zugriff eigene SQL-Kommandos auf dem Target ausführen kann, indem er diese Vulnerability in dem WordPress Plug-in „10Web Photo Gallery Plug-in (vor 1.5.31)“ ausnutzt.[76] Der Fokus liegt auf der Beleuchtung einer spezifischen Verwundbarkeit. Dementsprechend gibt es keine durchgehend definierte Syntax und Semantik, sondern die Inhalte sind teilweise in Fließtext formuliert, sodass die verschiedensten Vulnerabilities und ihre Eigenheiten weitestgehend erfasst werden können.

Eine weitere Informationsquelle ist die öffentliche CAPEC-Datenbank von The MITRE Corporation. Die folgenden Informationen zu CAPEC basieren auf [67, 37, 8]. Über verschiedene Sichtweisen bietet die Datenbank einen strukturierten Überblick an Angriffsmustern. Angriffsmuster beschreiben über gemeinsame Attribute die Ansätze eines Angreifers, um eine Schwachstelle einer softwarefähigen Technologie auszunutzen. Diese generische Darstellung umfasst sowohl die damit verbundenen Verwundbarkeiten, als auch die Herausforderungen und Lösungsmöglichkeiten des Angreifers. Auf der Website können somit z. B. aus der Sicht des Angriffsmechanismus oder der Angriffsdomäne die

benötigten Inhalte schnell gefunden werden, wie beispielsweise Informationen zu SQLI. Anhand von Attributen und Methoden wird aufgezeigt, wie ein Angreifer eine Schwachstelle ausnutzen kann. Beispielsweise wird in dem Angriffsmuster „CAPEC-66: SQLI“ der Execution Flow in drei Schritte aufgeteilt. Beispielsweise erhält man im ersten Schritt die Informationen, dass der Angreifer sogenannte „Spider Websites für alle verfügbaren Links“ zur Durchführung des Angriffs nutzen kann. Anhand dieser Angriffsmuster in CAPEC werden verschiedene Informationen zu einem bekannten Angriff dargestellt, dazu zählen auch die Möglichkeiten zur Abschwächung von Angriffen. Beispielsweise wird zu einer strengen Eingabevalidierung geraten, um SQLI vorzubeugen.

Zusammenfassend bieten die öffentlichen Datenbanken von The MITRE Corporation über verschiedene Sichtweisen mithilfe der bereitgestellten Informationen einen bestmöglichen Überblick zu den Themengebieten. Dadurch können die benötigten Inhalte schnell gefunden werden. Generell erfolgt die Präsentation mithilfe definierter XML-Schemata.[65, 74, 68] Sie sorgen für ein einheitliches Format zur Definition von Angriffsmustern, Schwachstellen und Verwundbarkeiten. Darüber hinaus gibt es eine Verknüpfung über IDs zwischen den Inhalten der Angriffsmuster und Schwachstellen, sodass eine Wiederverwendbarkeit dieser Muster gegeben ist. Konkrete Verwundbarkeiten werden in CAPEC und CWE vereinzelt als Beispiele über die CVE-Nummer referenziert.

4 Analysephase

Für das Modellierungskonzept eines geeigneten Angriffsmodells werden zunächst die Anforderungen an die zu entwickelnde Modellierung analysiert. Anhand von identifizierten charakteristischen Eigenschaften eines Angriffs werden die notwendigen Elemente für das Angriffsmodell abgeleitet und in Beziehung zueinander gesetzt.

4.1 Anforderungsanalyse

Im Folgenden werden die notwendigen Anforderungen an die zu entwickelnde Methodik zur Angriffsmodellierung analysiert. Die erforderlichen Eigenschaften der Modellierungsmethodik basieren auf dem gewünschten Einsatzzweck der Angriffsmodellierung. Tabelle 4.1 enthält eine Zusammenfassung der Anforderungen.

Modellbasiert Ein Angriff soll über ein Modell dargestellt werden. Die Ausführung dieses Abschnitts basiert auf [4, 12, 18, 27]. Neben den Vorteilen, die bereits in Abschnitt 3.2 aufgezeigt sind, liegt die Erwartung darin, dass die Anwendung einer modellbasierten Ansicht des Angriffs zu einer Formalisierung führt, wie das komponenten- und modellorientierte Softwareengineering für den Prozess der Softwareentwicklung in der IT. Mithilfe eines formalen Angriffsmodells sind Angriffe schlüssig in ihrer Darstellung, qualitativ hochwertig, wiederholbar und somit eine geeignete Basis für die Unterstützung durch Automatismen. Abschnitt 3.4 gibt einen Einblick in diese Thematik. Die zu entwickelnde Methodik soll auf eine formale Angriffsmodellierung abzielen, sodass in Zusammenhang mit Security-Tests eine automatisierte Unterstützung möglich ist. Durch diese Eigenschaften sinkt das Risiko von (Folge-)Aktivitäten, die auf dem Angriffsmodell basieren z. B. als Grundlage für die OWASP Risk Rating Methode, Penetrationstests, oder als Basis für eine AI. Der Aufwand zur Erlernung und Nutzung des formalen Angriffsmodells soll durch einen einfachen Aufbau und einer intuitiven Methodik zur Angriffsmodellierung so gering wie möglich gehalten werden. Auf Grundlage der Methodik zur Angriffsmodellierung steht im Sinne einer Spezifikation und Verifikation, wie in Abschnitt 3.4 beschrieben, ein geeignetes Mittel zur Kommunikation von Angriffen zur Verfügung. Somit wird die Kommunikation und Synchronisation im Umgang mit Angriffen verbessert, da das Modell als umfangreiche Dokumentation eines Angriffs fungieren kann, z. B. zur Modellierung

von kryptografischen Angriffen. Ein modellierter Angriff bedeutet, der Angriff ist identifizierbar. Das bildet wiederum die Entscheidungsgrundlage für den Anwendungsbereich der Verteidigung für IT-Sicherheitsexperten. Durch die Existenz eines Angriffsmodells steigt der Aufwand für einen erfolgreichen Angriff, da die Verteidiger anhand des Modells eine Wissensbasis für die Identifikation von Verteidigungsmaßnahmen besitzen.

Wiederverwendbar Wie bereits in Kapitel 2 aufgeführt ist, gibt es die verschiedensten Ansätze, einen Angriff darzustellen. Allerdings sind diese für bestimmte Anwendungsbereiche entwickelt. Z. B. ist die Lockheed Martin Cyber Kill Chain für die Präsentation von APTs ausgelegt. Mithilfe dieser Arbeit soll ein Konzept zur generischen Angriffsmodellierung entwickelt werden, das unabhängig von dem Anwendungsgebiet ist. Das sorgt für weitläufige Einsatzmöglichkeiten des Angriffsmodells und fördert die Wiederverwendbarkeit. Z. B. kann ein Unternehmen, das eine Menge an ähnlicher Software herstellt, das Angriffsmodell mit geringem Aufwand wiederverwenden. Gleichzeitig kann das generische Angriffsmodell unabhängig der Folgeaktivität bzw. des Anwendungsgebiets als Grundlage dienen. Das Angriffsmodell soll wiederverwendbare Bestandteile enthalten, sodass das Modell schnell aktualisiert werden kann. Dadurch wird eine aufwändige Neuentwicklung der Darstellung eines Angriffs vermieden.[17] Des Weiteren soll das Modell über verschiedene Abstraktionsebenen einen Angriff präsentieren können, um die Wiederverwendbarkeit zu unterstützen. Die unterschiedlichen Ebenen des Modells sind erforderlich, damit die relevanten Ziele diverser Anwendungsbereiche und somit die verschiedenen Sichtweisen auf einen Angriff dargestellt werden können.[17] Beispielsweise soll eine Sichtweise einen Überblick des gesamten Angriffs darstellen, der nicht durch unnötige Details zu Befehlsfolgen von konkreten Exploits behindert wird. Dagegen sind für eine andere Zielgruppe des Modells z. B. die Einzelheiten des Vorgehens eines Angreifers gefragt, statt des Gesamtzusammenhangs eines Angriffs.

Ausdrucksstark Das zu entwickelnde Modell soll eine möglichst umfassende Darstellung aller sicherheitsrelevanten Informationen zu Angriff, Angreifer und Zielsystem darbieten. Das bedeutet, mithilfe der Methodik zur Angriffsmodellierung sollen möglichst viele Angriffe abgebildet werden können. Wie bereits Abbildung 3.1 zeigt, steht ein Angriff in Verbindung zu verschiedenen Aspekten. Es soll ein Angriffsmodell entwickelt werden, das alle relevanten Komponenten für einen Angriff angemessen berücksichtigt. Der Aufbau eines umfangreichen Domänenwissens ist notwendig, um zu verstehen, auf welche Art und Weise ein Angriff stattfinden kann.[66] Dabei ist die Verwendung eines ausdrucksstarken Angriffsmodells erforderlich, unabhängig von dem Anwendungsbereich für den Umgang mit Angriffen. Der Einsatz einer umfassenden Abbildung aller charakteristischen Details zu Angriff, Angreifer und der möglichen Zielsysteme kann die IT-Security erhöhen. Beispielsweise kann ein vielseitiges Angriffsmodell die Detektion von Angriffen unterstützen. Zwar wird bereits versucht Angriffe im Unternehmen möglichst frühzeitig, z. B. mithilfe von Security Information and Event Management und Intrusion

Detection Systemen zu erkennen, allerdings gibt es keine Garantie für eine erfolgreiche Identifikation von Angriffen.[55, 53] Somit kann das zu entwickelnde Modell als zusätzliche Informationsquelle dienen, indem potentielle Angriffe aufgezeigt werden.

Systematisch Daneben soll das Modell eine systematische Verwendung aller notwendigen Informationen für die Gesamtbetrachtung eines Angriffs gewährleisten.[17] Z. B. unterstützt der Security-Standard Microsoft SDL [24, 34] einen systematischen und durchgängigen Security-Entwicklungsprozess. Allerdings werden dabei nicht alle Aspekte berücksichtigt, die für die Gesamtbetrachtung eines Systems bezüglich eines Angriffs relevant sind. Die Systemmodelle stehen im Vordergrund.[24, 34] Sämtliche sicherheitsrelevanten Informationen werden größtenteils in der Security Requirements Engineering-Phase des SDL verwendet.[24, 34] In den nachfolgenden Phasen erfolgt keine fortlaufende und gezielte Verwendung bzw. Kombination von sicherheitsrelevantem Angriffswissen mit den notwendigen Systeminformationen.[24, 34] Mithilfe des zu entwickelnden Angriffsmodell sollen die wesentlichen Informationen für die Gesamtbetrachtung eines Angriffs systematisch verwendet werden. Dadurch wird die Durchgängigkeit und Nachverfolgbarkeit von IT-Sicherheit unterstützt. Beispielsweise basieren Penetrationstest auf unstrukturierten Daten, wie auf dem Ergebnis von Bedrohungs- und Risikoanalysen, z. B. der OWASP Risk Rating Methode.[50] Dabei stützt sich die OWASP Risk Rating Methode, wie in Abschnitt 3.5 beschrieben, sowohl auf Security-Informationen, als auch auf Informationen aus Systemmodellen.[21] Allerdings erfolgt keine strukturierte Kombination und Verwendung dieser Informationen.[21] Infolgedessen kann ein systematisches Modell zu Angriff, Angreifer und Angriffsziel als roter Faden für Security-Tests, wie z. B. für Penetrationstests oder für Security-Analysen, wie z. B. der OWASP Risk Rating Methode fungieren. Dadurch ist IT-Sicherheit durchgängig nachweisbar, sodass das Vertrauen in die Software steigt und die Softwarequalität zunimmt.

Simulierbar/Konsistent Mithilfe des zu entwickelnden Konzepts zur Angriffsmodellierung soll eine Analyse bzw. Simulation von Angriffen möglich sein. Beispielsweise kann ein bestimmter Angriff über verschiedene Szenarien simuliert werden. Zum einen sorgt die Simulation für ein besseres Verständnis des Angriffs im Allgemeinen.[17] Zum anderen dient das Ergebnis als zusätzliche Basis für (Folge-)aktivitäten im Sicherheitsprozess eines Unternehmens. Z. B. kann auf Basis des Simulationsergebnisses ein identifizierter Angriff durch geeignete Sicherheitsmaßnahmen abgeschwächt werden. Außerdem sollen sich aus der Simulation zukünftige Angriffsszenarien ableiten lassen. Die Analysierbarkeit bzw. Simulierbarkeit ist maßgebend für die Verifizierung und Validierung des Modells und somit für dessen Konsistenz.[17] Das generische Angriffsmodell soll konsistent sein, da es als Grundlage für weitere Aktivitäten fungiert. Dadurch ist beispielsweise die OWASP Risk Rating Methode nicht mehr lediglich von den Erfahrungswerten der Experten abhängig, wie in Abschnitt 3.5 beschrieben, sondern kann das Angriffsmodell als konsistente Grundlage nutzen. Ein widersprüchliches Modell führt zu einer schlechten

Qualität aller Ergebnisse, die darauf basieren.[33] Zudem ist ein widerspruchsfreies Modell Voraussetzung für eine Werkzeugunterstützung durch geeignete Automatismen.[33] Der Aufwand zur Analyse und Generierung des Modells kann durch Automatisierung bzw. durch Werkzeugunterstützung reduziert werden.[12]

Visualisierbar Die Methodik zur Angriffsmodellierung soll sich auf die Visualisierung von Angriffen fokussieren. Die Ausführung dieses Abschnitts basiert auf [17, 12]. Eine grafische Darstellung des Angriffsmodells erleichtert sowohl die Lesbarkeit und Handhabung, als auch die Abstraktion des Modells. Die Komplexität von Angriffen soll damit handhabbarer sein, denn anhand von grafischen Modellen können sowohl Aktivitäten, als auch die Strukturen besonders gut für den Menschen anschaulich dargestellt werden.[20] Eine grafische Abbildung ist für einen Menschen intuitiver als Prosatext. Grafiken tragen somit zum besseren Verständnis des Angriffs bei. Gleichzeitig soll damit der Aufwand zur Erlernung reduziert und eine einfache Benutzung unterstützt werden. Im Idealfall ist der Zusammenhang der einzelnen Komponenten des grafischen Angriffsmodells einfach ersichtlich, sodass auch dadurch die IT-Sicherheit durchgängig nachweisbar ist und somit die Softwarequalität steigt.

Verständlich Die Methodik zur Angriffsmodellierung soll verständlich, einfach zum Erlernen und unkompliziert in der Benutzung sein.[17, 12] Für die Modellierung eines Angriffs gibt es bereits eine Vielzahl an Möglichkeiten. In Kapitel 2 sind einige Repräsentanten aufgelistet. Wegen Ressourcenknappheit in Zeit und Ausbildung von Entwicklern für die Aneignung von neuen speziellen Methoden soll versucht werden, vorhandene Modellierungsansätze zu verwenden, um den Aufwand zur Erlernung und Anwendung des Angriffsmodells möglichst gering zu halten.[27] Ein Ansatz, der dahingehend verfolgt werden soll, ist der Einsatz von UML-Aktivitätsdiagrammen und Attack Trees für die Angriffsmodellierung. Diese Mittel sind ausgewählt, da sie in den entsprechenden Anwendungsgebieten bereits etabliert sind. Für die Darstellung eines Ablaufs bietet sich das UML-Aktivitätsdiagramm an. Aufgrund der weiten Verbreitung von UML und deren integriertem Erweiterungsmechanismus, wie in Abschnitt 3.3 erläutert, bietet die Modellierungssprache nützliche Voraussetzungen für die Angriffsmodellierung. Es soll analysiert werden, inwieweit ein Angriff als Prozess über ein UML-Aktivitätsdiagramm darstellbar ist. Der Modellierungsansatz Attack Tree aus [47] ist ein bewährtes Mittel für die Abbildung der verschiedenen Möglichkeiten, wie ein Angriffsziel erreicht werden kann. Dieser Ansatz ist intuitiv zu verstehen und kann einfach wiederverwendet werden.[47] Inwieweit Attack Trees eine geeignete Basis für eine generische Methodik zur Angriffsmodellierung bildet, soll ebenfalls untersucht werden.

Tabelle 4.1: Anforderungen an die Methodik zur Angriffsmodellierung auf Grundlage von [17]

1	Modellbasiert
2	Wiederverwendbar
3	Ausdrucksstark
4	Systematisch
5	Simulierbar/Konsistent
6	Visualisierbar
7	Verständlich

4.2 Charakteristische Eigenschaften eines Angriffs

Fokus dieser Arbeit ist die Domäne Angriff, dessen Kontext in Abbildung 3.1 zu sehen ist. Der Angriff steht im Mittelpunkt und ist von Angreifer und Zielobjekt(en) abhängig. Für die Identifikation der notwendigen Angriffsmodellelemente werden die charakteristischen Eigenschaften eines Angriffs analysiert. Anschließend können daraus die relevanten Elemente für die Angriffsmodellierung abgeleitet werden.[17] Die bezeichnenden Eigenschaften eines Angriffs sind so aufzustellen, dass mithilfe einer Spezifizierung der Eigenschaften jeglicher Angriff bestimmt werden kann. Ziel ist ein Modell zur Angriffsillustration mit einem tiefen Informationsgehalt, sodass realistische Angriffe über modellierte Angriffe abgebildet werden können. Der Fokus liegt demzufolge auf qualitativen Eigenschaften, statt auf Merkmalen für quantitative Ansätze, wie in Abschnitt 3.5 erläutert. Als Leitfaden für die Analyse der beschreibenden Eigenschaften eines Angriffs dient der Bottom-up Ansatz. Ein beispielhafter Angriff aus dem Bereich „Webanwendungen“ wird zur Veranschaulichung der Erläuterungen in diesem Abschnitt 4.2 verwendet. Das Beispiel wird an geeigneter Stelle referenziert und gegebenenfalls mit weiteren Details ergänzt.

Beispiel 4.2.1 *Ein Angreifer möchte das öffentliche Ansehen einer Firma ruinieren, die einen Webshop besitzt. Dem Webshop liegt ein Webserver und ein SQL-Server zugrunde.*

Angriff aus Prozesssicht Wie bereits in 3.4 erwähnt, besteht ein Angriff aus vielen Aktionen. Eine einzelne Aktion steht nicht zwangsläufig in Verbindung mit einer unberechtigten Handlung oder einem unautorisierten Zugriff(-sversuch). In Beispiel 4.2.1 möchte der Angreifer einen SQLI-Angriff durchführen. Dafür versucht der Angreifer zunächst Informationen über den Server herauszufinden, indem er beispielsweise den

Server zum Senden einer Fehlernachricht zwingt und diese anschließend nach relevanten Informationen analysiert. Die einzelne Aktion des Angreifers, den Server zum Senden einer Fehlernachricht zu bringen, ist an sich keine unerlaubte Handlung oder ein unautorisierter Zugriff. Angenommen, die Eingabe eines einfachen Anführungszeichens führt zu einer Fehlernachricht des Servers.[16, 32] Es ist davon auszugehen, dass einem Benutzer unabsichtlich Fehler bei der Eingabe unterlaufen, wie z. B. die Eingabe eines einfachen Anführungszeichens. Ausschlaggebend ist die systematische Ausführung von einer Menge an bestimmten Aktionen, die auf das Angriffsziel abzielen. Z. B. kann der Angreifer eine den Webserver zum Senden einer Fehlernachricht erzwingen, diese nach Hinweisen zur verwendeten Datenbank analysieren, eine entsprechende SQL-Query entwickeln, testen und letztendlich den zielgerichteten SQLI-Angriff ausführen. Das bedeutet, ein Angriff wird durch die systematische Ausführung von Angriffsaktionen charakterisiert. Gleichzeitig können einzelne Angriffe zu einem (größeren) Angriff zusammengefasst werden.

Definition 4.1 *Aus Prozesssicht betrachtet ist ein Angriff aus (Wiederholungen von) „kleineren" Angriffen, den sogenannten Angriffsiterationen zusammengesetzt. Eine Angriffsiteration bildet den kleinstmöglichen Angriffsdurchlauf ab, der mit einem Angriffsziel verknüpft ist. Eine Angriffsiteration besteht wiederum aus Aktionen, deren systematische Ausführung zur Erreichung eines Angriffsziels beiträgt. Der Angreifer versucht, mit jeder Iteration eine Schicht tiefer in das System zu gelangen bzw. seinem Angriffsziel näher zu kommen.[60] Mit den Ergebnissen jeder Angriffsiteration baut der Angreifer sein Wissen über das Target und dessen Umgebung auf. Die Überprüfung des aktuellen Angreiferwissens mit dem Angriffsziel stellt den Start- bzw. Endpunkt einer Angriffsiteration dar.*

Beispielsweise entwickelt und experimentiert der Angreifer aus Beispiel 4.2.1 zunächst iterativ eine Query, um eine vorhandene SQLI-Vulnerablity entsprechend seinem Angriffsziel auszunutzen.[8] Jeder dieser experimentellen Tests auf das Target ist eine Angriffsiteration. Dabei versucht der Angreifer sein Wissen über das Target und dessen Umgebung mit jeder Angriffsiteration zu erweitern, um sein Angriffsziel zu erreichen. Das Ziel des Angreifers (Angriffsziel) entspricht einem strategischen Ziel.

Definition 4.2 *In diesem Abschnitt wird der Begriff Strategie auf Basis von [41, 82] definiert und angepasst. Der Begriff Strategie stammt aus dem Militärwesen. Diese legt auf abstrakte Weise die Rahmenbedingungen zur Erreichung eines Ziels und somit das generelle Vorgehen fest. Das strategische Ziel entspricht einem großen, langfristigen Ziel, das an oberster Stelle steht. Es ist mit einem Angriff verknüpft. Dabei wird das strategische Ziel in kleinere Ziele aufgeteilt, die der Angreifer versucht zu erreichen, sodass daraus letztendlich das strategische Ziel realisiert wird. Das bedeutet, der Angriff mit dem strategischen Ziel wird in kleinere Angriffe aufgeteilt, die jeweils mit einem untergeordneten Ziel verknüpft sind.*

Das strategische Ziel des Angreifers in Beispiel 4.2.1 ist die Beschädigung des Images einer Firma, die über das Internet einen Webshop anbietet. Dieses Ziel kann über die Erreichung von aufeinanderfolgenden taktischen Zielen [82] abgebildet werden.

Definition 4.3 *Diese Definition basiert auf Inhalten aus [41, 82] und wird im Kontext dieser Arbeit im Folgenden angepasst. Die Taktik, die ebenfalls aus dem Umfeld des Militärs stammt, beschreibt notwendige Aspekte und Aktivitäten, die Voraussetzung sind, um das übergeordnete strategische Ziel zu erreichen. Das Ziel einer Taktik wird als taktisches Ziel bezeichnet und besteht aus generellen Aussagen, ohne auf die genaue Umsetzung einzugehen. Ein taktisches Ziel ist mit einem Angriff verbunden. Durch die strukturierte Realisierung der taktischen Ziele in einer bestimmten Reihenfolge, bzw. der damit verbundenen Angriffe, versucht der Angreifer sein strategisches Ziel zu erreichen, bzw. den damit verbundenen Angriff auszuführen.*

Beispielsweise unterstützen folgende Taktiken bzw. taktischen Ziele die Erreichung des strategischen Ziels aus dem Beispiel 4.2.1:

- Sammlung von relevanten Informationen über das Target, dessen Umgebung und den damit verbundenen Schwachstellen, um eine geeignete SQL-Query für den Angriff zu entwickeln [8, 16]

- Ausnutzung einer identifizierten Vulnerability durch den dafür entwickelten Exploit (SQL-Query) [8]

- Entwicklung und Einbau einer Hintertür[1] (engl. Backdoor) [53]

Die Umsetzung dieser verschiedenen Taktiken ist nicht willkürlich. Die taktischen Ziele sind mit bestimmten Voraussetzungen gekoppelt. Z. B. setzt das taktische Ziel der Ausnutzung einer Vulnerability Aufklärungs- und Entwicklungsarbeiten voraus, um eine geeignete Schwachstelle zu finden und einen passenden Exploit dafür zu erzeugen. Der Angreifer muss die notwendigen Erkenntnisse bzw. Wissen über die Umgebung und somit für die Ausführung eines Angriffs besitzen. Infolgedessen ist die Ausführungsreihenfolge zielgerichteter Taktiken über mehrere Angriffsiterationen maßgebend für einen Angriff. Die verschiedenen Angriffsiterationen lassen sich über taktische Ziele abbilden, die in Verbindung zueinander einen Angriff mit einem strategischen Ziel präsentieren.

[1] Ein Angreifer hält sich die Möglichkeit eines Systemzugangs für spätere Angriffe offen. Dabei umgeht er Sicherheitsmechanismen.[53]

Umgebung Jeder Angriff zielt auf ein bestimmtes Target ab, wie bereits in 3.4 definiert ist. Die Art des Targets ist dabei sehr vielfältig. Beispielsweise kann ein Mensch das Target einer Social Engineering Attacke sein. Wegen dem bemessenen Rahmen einer Masterarbeit fokussiert sich diese Arbeit lediglich auf Cyber-Enabled Capabilities. Es existiert eine Vielzahl an verschiedensten Cyber-Enabled Capabilities in einem Unternehmen, die angegriffen werden können. Dazu zählen sowohl Webserver, Datenbanken, als auch gekaufte oder selbst entwickelte Softwareprodukte. Daher ist das Target nicht zwangsläufig mit nur einem Zielobjekt bzw. einer Cyber-Enabled Capability verknüpft. Z. B. stellt in dem Beispiel 4.2.1 ein existierender Web- und SQL-Server jeweils eine Cyber-Enabled Capability und somit ein Target dar. Das Beispiel zeigt, dass das Umfeld, in dem ein Angriff stattfindet, bedeutsam ist.[18] Cyber-Enabled Capabilities stehen in einer Umgebung und können mit anderen Cyber-Enabled Capabilities kommunizieren. Jegliche Cyber-Enabled Capability, wie z. B. der Webserver oder die SQL-Datenbank kann Schwachstellen, Fehler bzw. Zugangs- und Angriffspunkte enthalten, über die man das Target angreifen kann. Daran ist ersichtlich, dass jede Cyber-Enabled Capability, deren Umgebung und bereitgestellten Mechanismen, die für Angriffe ausgenutzt werden können oder der dazu beiträgt, die Fähigkeit einer Cyber-Enabled Capability zu beeinflussen, für einen Angriff von Bedeutung ist.[71, 42] Die Umgebung bietet somit gewisse Stellen, über die Angriffe ausgeführt werden können. Auf die Details der Stellen wird im weiteren Verlauf der Arbeit eingegangen.

Angreiferwissen Das Angreiferwissen ist ein wichtiges Attribut als Entscheidungsquelle. Es begrenzt, zusammen mit dem strategischen Ziel des Angreifers, den Angriff bzw. danach richtet sich der Angriff aus. Der Angreifer versucht sich das Wissen über das Target und dessen Umfeld anzueignen. Das bedeutet, das Wissen des Angreifers ändert sich im Laufe der Zeit. Zu Beginn besitzt der Angreifer kaum Informationen über das Target. Er muss sich diese Informationen über seine verfügbaren Möglichkeiten aneignen. Gleichzeitig legen die aktuellen Kenntnisse über das Umfeld die Möglichkeiten und Grenzen für die Informationsbeschaffung bzw. für einen Angriff durch den Angreifer fest.[61, 1] Das aktuelle Wissen eines Angreifers über Target und dessen Umgebung wird für den weiteren Verlauf der Arbeit als Angreiferwissen bezeichnet. Das Angreiferwissen erweitert sich, indem der Angreifer neue Erkenntnisse zu einem Target oder dessen Umgebung nach einer Angriffsiteration gewinnt. Das Angreiferwissen dient daher als Ausgangslage für die nächste Angriffsiteration. Beispielsweise erhält der Angreifer aus dem Beispiel 4.2.1 eine Fehlermeldung des Servers. Nach Analyse der empfangenen Fehlernachricht nimmt der Angreifer an, dass eine „MySQL“-Datenbank mit der Version 5-6-45 als Backendsystem für den Webshop fungiert.[16] Für einen SQLI-Angriff versucht er daraufhin eine geeignete Query entsprechend der „MySQL“-Syntax zu entwickeln und auszuführen. An dieser Stelle ist zu beachten, dass der Angreifer im Aufbau seines Wissens Fehler machen bzw. getäuscht werden kann.[18] Beispielsweise ist die Annahme einer „MySQL“-Datenbank falsch, sodass die erste Angriffsiteration mit einer entwickelten „MySQL“-Syntax Query zum Fehler führt. Diese Information soll ebenfalls im

Angreiferwissen abgebildet werden, sodass seine darauffolgende Iteration dahingehend geplant und ausgeführt wird. Das Angreiferwissen entspricht einer individuellen Grundlage des Angreifers, die sich stetig ändert. Das bedeutet, das Angreiferwissen ist keine (vollständige) Abbildung der tatsächlichen Umgebung.

Technische Sicht auf einen Angriff Ein Angriff kann nicht nur als Prozess dargestellt werden, sondern auch über Techniken, Muster oder in Form eines konkreten Exploit-Codes.

Eine Angriffstechnik beschreibt eine Möglichkeit, „Wie/Mit welchen Mitteln“ ein Angreifer sein aktuelles taktisches Ziel erreichen kann bzw. „Was“ ein Angreifer durch die Ausführung von zielgerichteten Aufgaben erreichen möchte, um sein taktisches Ziel zu erfüllen. [61] Dabei werden Aufgaben festgelegt, ohne dabei auf die genaue Art und Weise der Umsetzung einzugehen.[82] Z. B. kann das taktische Ziel „Informationen über das Target gewinnen (Aufklärungsaktivitäten)“ des Beispiels 4.2.1 über folgende Angriffstechniken erreicht werden:

- Social Engineering [62]
- Google Hacking [62]
- Überprüfung der Unternehmenswebsite [79]
- Passives Monitoring [62, 8, 80]

Der Angreifer wählt z. B. entsprechend seiner Fähigkeiten bzw. Fachkenntnisse und vorhandenen Ressourcen eine geeignete Angriffstechnik aus. Daneben beeinflusst das aktuelle Angreiferwissen die Auswahl einer Angriffstechnik. Es folgt ein Ausschnitt des Angreiferwissens aus dem Beispiel 4.2.1 (zu einem bestimmten Zeitpunkt): [38, 42]

- Relational Database Management System „MySQL“ der Version 5-6-45
- Apache Webserver der Version 2-0-65
- Webshop besteht aus statischen und dynamischen HTML Seiten
- Webshop enthält ein Benutzereingabefeld

Im Rahmen einer Taktik beeinflusst das Angreiferwissen die Auswahl einer geeigneten Angriffstechnik. Ausgehend von der Taktik (Aufklärungsaktivitäten) dient die SQLI-Angriffstechnik z. B. dazu herauszufinden, ob das Target eine Verwundbarkeit für SQLI aufweist. Es wird zum einen angenommen, dass ein Cyber-Krimineller entsprechende

Fähigkeiten und Ressourcen besitzt, die SQLI-Angriffstechnik anzuwenden. Zum anderen hat sich der Cyber-Kriminelle bereits das entsprechende Wissen angeeignet und geht davon aus, dass es eine „MySQL“-Datenbank zur Speicherung, Abfrage und Modifikation von Daten gibt.[8] Außerdem enthält der Webshop ein Benutzereingabefeld, das womöglich in Verbindung zu der „MySQL“-Datenbank steht.[8] Infolgedessen sind die entsprechenden Voraussetzungen für die Anwendung der SQLI-Angriffstechnik gegeben. Das Beispiel zeigt, dass ein Angreifer auf Grundlage seines aktuellen Wissens seinen Angriff plant, geeignete Techniken auswählt und letztendlich den Angriff über die Umsetzung der mit der Technik verbundenen Aufgaben durchführt.

Das „Prüfen nach einer vorhandenen SQLI-Verwundbarkeit“ entspricht einer Aufgabe, die mit der SQLI-Angriffstechnik verknüpft ist. Ein oder mehrere Exploits spezifizieren eine Aufgabe einer Angriffstechnik. Ein Angriff lässt sich infolgedessen über einen oder mehrere konkrete Exploits charakterisieren. Die Auswahl des Exploits innerhalb einer Angriffsiteration basiert neben der gewählten Angriffstechnik auf dem aktuellen Angreiferwissen und seinem (aktuellen) taktischen Ziel. Die Spezifikation des Auswahlverfahrens ist nicht Teil der Masterarbeit.

Durch die Ausführung des ausgewählten Exploits greift der Angreifer sein Zielobjekt an. Dabei können auf einer gewissen Art und Weise, über bestimmte Angriffspunkte vorhandene Vulnerabilities ausgenutzt werden.[58] Dazu benötigt der Angreifer das Wissen über die vorhandenen Vulnerabilities (Sicherheitslücken, Fehlfunktionen) des Targets und dessen Umgebung (Angreiferwissen). Dieses Wissen versucht sich ein Angreifer z. B. über verschiedene Zugangspunkte anzueignen, von denen der Angreifer Kenntnis hat, bzw. die ihm zur Verfügung stehen. Beispielsweise stellen Fehlerseiten/-nachrichten des Servers [7, 16] Zugangspunkte für das Beispiel 4.2.1 dar. Der Angreifer nutzt die erhaltene Fehlernachricht des Servers als Informationsquelle zur Bestimmung der dahinterliegenden Datenbank und somit als Zugangspunkt. Daraus identifiziert er die „MySQL“-Datenbank, sodass er anschließend eine für den Angriff geeignete „MySQL“-Query entwickeln kann.[16] Sobald der Angreifer seine entwickelte „MySQL“-Query über den Zugangspunkt der Benutzereingabe durchführt, wird der Zugangspunkt als Angriffspunkt bezeichnet, wie bereits in 3.4 beschrieben. In Tabelle A.1 soll der Zusammenhang zwischen Zugangs- und Angriffspunkt an weiteren Beispielen verdeutlicht werden. Beispielsweise kann der Programmcode in einem Angriffsszenario über die Zeit hinweg sowohl als Zugangspunkt, als auch als Angriffspunkt bezeichnet werden: Zunächst wird der Programmcode analysiert, um Informationen zu gewinnen (Zugangspunkt), wie z. B. die verwendeten Programmiersprachen. Anschließend kann der Programmcode über Manipulation als Angriffspunkt fungieren, um sich Zugang zu verschaffen, z. B. indem eigene Skripts importiert werden.

Neben der Angriffstechnik und dem aktuellen Wissen des Angreifers ist dessen aktuelles taktisches Ziel ausschlaggebend für die Auswahl des Exploits innerhalb einer Angriffsiteration. Zu einer bestimmten Zeit steht in Beispiel 4.2.1 die Taktik „Aufklärungsaktivitäten“ im Fokus. Das bedeutet, dass zu diesem Zeitpunkt z. B. lediglich

das Prüfen der Existenz einer „MySQL“-Injection-Vulnerability als zielführender Exploit relevant ist. Exploits bezüglich der konkreten Ausnutzung stehen aufgrund des zu diesem Zeitpunkt aktuellen taktischen Ziels (Aufklärungsaktivitäten) nicht im Fokus. Im weiteren zeitlichen Verlauf des Angriffsszenarios aus Beispiel 4.2.1 sind andere Exploits zielführend gemäß einer entsprechenden Taktik.

Verwundbarkeiten, deren Existenz einen Angriff charakterisieren, lassen sich ebenfalls über spezifische Eigenschaften beschreiben. Anhand einer SQLI-Vulnerability aus Beispiel 4.2.1 soll die Charakteristik veranschaulicht werden. Eine Verwundbarkeit wird in dieser Arbeit durch folgende Merkmale charakterisiert:

- **Angriffspunkt:** Um eine existierende Vulnerability auszunutzen, muss es mindestens eine Stelle geben, an dem ein Angreifer Zugriff zu dieser Fehlfunktion bzw. Sicherheitslücke erhält. Ein Angriffspunkt wird zuvor analysiert und in geeigneter Weise von dem Angreifer beeinflusst, sodass die damit verknüpfte Verwundbarkeit für einen Angriff zur Verfügung steht.[1] Zu einer SQLI-Vulnerability gehören als Angriffspunkte beispielsweise Benutzereingabefelder, HTTP Parameter und Netzwerkpakete.[3, 8]

- **Exploit:** Eine Vulnerability ist mit einer Menge an Exploits verknüpft. Für eine konkrete Ausnutzung einer existierenden Verwundbarkeit verwendet ein Angreifer bestimmte Exploits in Form von Befehlsfolgen, Funktionalitäten oder eines Inputs, wie in 3.4 beschrieben. Die SQLI-Vulnerability des Beispiels 4.2.1 steht z. B. in Verbindung mit einem „MySQL“-Exploit zur Identifikation der Existenz einer SQLI-Vulnerability und mit „MySQL“-Exploits zur Bestimmung von Datenbankfeldern und Tabellennamen.

- **Umgebung:** Für eine erfolgreiche Ausnutzung einer Verwundbarkeit müssen gewisse Voraussetzungen bezüglich des Targets und dessen Umgebung gegeben sein. Der Angreifer muss in Kenntnis von den notwendigen Informationen zu dem Target und dessen Umgebung sein, damit er anhand seiner Taktik einen passenden Exploit entwickeln und über einen gegebenen Angriffspunkt die Vulnerability ausnutzen kann.[1] Beispielsweise setzt eine SQLI-Vulnerability das Vorhandensein von SQL-Queries bei einer Anwendung zur Speicherung, Abfrage und Modifikation von Daten voraus, wie z. B. „MySQL“, „Oracle“ oder „Microsoft SQL Server“.[8] Des Weiteren begünstigt eine ungenaue Validierung der Benutzereingabe bezüglich SQL-Queries die Existenz einer SQLI-Vulnerability.[8] Der Angreifer muss daher zunächst herausfinden, inwieweit für eine Vulnerability die zugehörigen Voraussetzungen gemäß der Umgebung gegeben sind. Das Angreiferwissen repräsentiert die aktuellen Kenntnisse des Angreifers für den Vergleich mit den Voraussetzungen einer bestimmten Verwundbarkeit.

Eine Verwundbarkeit wird im Kontext einer Technik, die von dem Angreifer gewählt wird, ausgenutzt. Wie bereits zuvor erläutert, beschreibt eine Angriffstechnik notwendige Aufgaben, um ein entsprechendes taktisches Ziel zu erreichen. Die Aufgaben werden über Exploits spezifiziert. Dabei werden vorhandene Vulnerabilities auf geeigneter Art und Weise miteinbezogen. Die Merkmale der Verwundbarkeit können somit als Eigenschaften eines Angriffs betrachtet werden, da eine Verwundbarkeit Teil eines Angriffs sein kann.

Angreifer Schließlich muss der Angreifer an sich betrachtet werden. Er ist Quelle eines Angriffs und beeinflusst diesen daher maßgebend. Er bestimmt die zuvor erläuterten Aspekte wie z. B. das strategische Ziel, die taktischen Ziele, die Angriffstechniken. Er plant, entwickelt und führt Exploits aus. Zudem sind die nachfolgenden Merkmale essentiell für die Betrachtung eines Angreifers. Jeder Angreifer ist an einer bestimmten Stelle lokalisiert, wodurch er eine Menge an Möglichkeiten besitzt, das Target zu kontaktieren bzw. Informationen darüber zu sammeln. Befindet sich der Angreifer bei dem Target vor Ort, kann er z. B. eine Bluetooth-Schnittstelle als Zugangs- bzw. Angriffspunkt nutzen.[52] In den meisten Fällen ist die Kommunikation über das Internet die erste Anlaufstelle für die Ausführung eines entfernten Angriffs. Jeder Angreifer besitzt gewisse Fähigkeiten, die er für den Angriff mitbringt. Es wird angenommen, dass komplizierte Angriffe oder Angriffe, die mit viel Aufwand in Verbindung stehen, da z. B. umfangreiche Sicherheitsmechanismen eingebaut sind, weder von Skript-Kiddies, noch von einfachen Cyber-Kriminellen bevorzugt verwendet werden.[5] Diese Angreiferprofile suchen vielmehr nach einfachen Angriffspunkten bzw. Vulnerabilities. Sobald einem Angreifer mehrere Angriffstechniken zur Verfügung stehen, wird vorzugsweise die einfachste Möglichkeit verwendet. Gleichzeitig können Angreifer mit gewissen Mustern in Verbindung gebracht werden. The MITRE Corporation verknüpft mithilfe von ATT&CK Angreifer mit bestimmten Techniken und Software.[61] Folglich können Angreiferprofile mit gewissen Angriffsbereichen charakterisiert werden, in dem ein Angreiferprofil bevorzugt mit gewissen Techniken arbeitet. Daneben stehen jedem Angreifer bestimmte Ressourcen und Werkzeuge zur Verfügung. Organisierte Angreiferprofile, wie beispielsweise Cyber-Terroristen oder staatliche Nachrichtendienste sind z. B. im Besitz von vielen leistungsfähigen Rechensysteme, da sie das Geld dafür besitzen. Diese Ressourcen können im Rahmen der zur Verfügung stehenden Zugangs- bzw. Angriffspunkte und Angriffstechniken für Angriffe verwendet werden. Außerdem spielt die verfügbare Zeit als Ressource ein wichtiges Kriterium. Bei Angreiferprofilen in Verbindung mit Industriespionage kann beispielsweise angenommen werden, dass ihnen eine gewisse Zeit zur Verfügung steht, da ihr strategisches Ziel meist im Zusammenhang mit einer langlebigen und unbemerkten Präsenz im System des Opfers verbunden ist.[5]

Eine Zusammenfassung der charakteristischen Merkmale eines Angriffs ist in Tabelle A.1 zu finden. Die Eigenschaften sind mit Beispielen in der dritten Spalte der Tabelle veranschaulicht, wobei es keinen Zusammenhang zwischen den aufgezählten Beispielen in der dritten Spalte gibt. Die Bezeichnungen „A“ und „B“ sind willkürliche Platzhalter. Mehrere Beispiele werden durch ein Semikolon getrennt. Die Charakteristik ist nicht vollständig, insbesondere im Zusammenhang mit dem Angreifer. Fokus der Arbeit ist, eine Methodik zur Angriffsmodellierung zu entwickeln. Die Vollständigkeit der Angriffscharakteristik steht nicht im Mittelpunkt.

4.3 Modellelemente

In diesem Teil der Arbeit sollen die notwendigen Elemente für die Angriffsmodellbildung identifiziert werden. Ausgehend von den zuvor identifizierten Merkmalen eines Angriffs in Abschnitt 4.2 lassen sich die Modellelemente ableiten. Dabei wird nicht auf die Umsetzung und Implementierung der Elemente eingegangen. Fokus ist die Identifikation der notwendigen Elemente für eine geeignete Methodik zur generischen Angriffsmodellierung.

Angreifer Der Angreifer ist als grundlegende Informationsquelle für das Angriffsmodell notwendig. Jeder Angriff wird von einem Angreifer ausgeführt. Demzufolge beeinflusst der Angreifer den Angriff. Ein Angreifer soll über ein Angreiferprofil charakterisiert werden, indem alle notwendigen Attribute bestimmt sind, die die Vorgehensweise im Angriffsmodell beeinflussen. Folgende Merkmale sind aus Abschnitt 4.2 abgeleitet und bilden eine Entscheidungsbasis für die Methodik zur Angriffsmodellierung:

- **Strategisches Ziel:** Das Ziel des Angreifers, was dem strategischen Ziel des Angriffs entspricht, ist die Ausgangslage für den gesamten Angriff. Die grundlegenden, richtungsweisenden Entscheidungen, wie beispielsweise taktische Ziele, Angriffstechnik und der Aufbau seines Wissens basieren auf diesem strategischen Ziel.

- **Scope:** Der Zugangsbereich bestimmt die vorhandenen Möglichkeiten, „wie“ ein Angreifer das Target kontaktieren kann.[52] Der globale Scope beinhaltet globale Schnittstellen, wie z. B. in Verbindung mit dem Internet. Regionale Zugriffspunkte umfassen z. B. Richtfunk-Techniken. Der lokale Scope beinhaltet beispielsweise Bluetooth oder Infrarot. Anhand des Scopes werden die potentiell vorhandenen Zugangs- bzw. Angriffspunkte im Angriffsmodell bestimmt. Befindet sich ein Angreifer z. B. im globalen Scope, kann er keine Zugangspunkte des lokalen Scopes nutzen.

- **Bevorzugter Angriffsbereich:** Angreifer können mit bestimmten Techniken in Verbindung stehen, die sie bei ihren Angriffen häufig verwenden.[31, 70] The MITRE Corporation verknüpft z. B. in [70] die Gruppe „APT18“ mit der Nutzung von „cmd.exe“.[70] Dementsprechend können Angriffstechniken, die die Ausnutzung von „cmd.exe“ beinhalten, für die Auswahl der Angriffstechnik priorisiert werden. Im Zusammenhang mit den bevorzugten Angriffsbereichen stehen wiederum spezielle Zugangs- bzw. Angriffspunkte, die der Angreifer präferiert.

- **Vorhandene Ressourcen:** Jedes Angreiferprofil kann mit bestimmten Ressourcen in Verbindung gebracht werden, von denen ausgegangen wird, dass sie dem Angreifer(-profil) zur Verfügung stehen, wie beispielsweise Ressourcen bzw. Werkzeuge zur Steigerung der Performance, oder zur Tarnung, oder die verfügbare Zeit für einen Angriff.[21] Danach lässt sich die Auswahl an möglichen Angriffstechniken begrenzen. Z. B. ist davon auszugehen, dass ein Cyber-Krimineller schnell an Geld gelangen möchte, sodass eine langwierige Angriffstechnik von diesem Angreiferprofil nicht bevorzugt verwendet wird.

- **Kompetenz:** Die Fachkenntnisse, die mit einem Angreifer(-profil) verbunden sind, sollen eine weitere Entscheidungsquelle für die Methodik zur Angriffsmodellierung bilden. Beispielsweise ist davon auszugehen, dass ein Skript-Kiddie wenig bis kein Know-How bezüglich Kryptographie besitzt.[5] Demnach sollen Skript-Kiddies keine Möglichkeit zur Auswahl von Angriffstechniken basierend auf kryptografischen Analysen erhalten.[11]

Diese Zusammenfassung ist nicht vollständig. Sie präsentiert lediglich einen Analysestand zu der Darstellung eines Angreiferprofils für das entwickelte Konzept der Angriffsmodellierung. Jedes Angreiferprofil definiert sich über die verschiedenen Werte der charakteristischen Eigenschaften eines Angreifers. Daraus ist ersichtlich, dass der Angreifer ein notwendiges Element für die Angriffsmodellierung ist.

Angreiferwissen Das Wissen des Angreifers ist separat vom Angreifer zu betrachten, da es sich mit jeder Angriffsiteration ändert. Der dynamische Wissensaufbau ist erforderlich, um den genauen Verlauf bzw. das Vorgehen eines Angreifers systematisch modellieren zu können. Auf Grundlage des Angreiferwissens plant und entwickelt der Angreifer seinen Angriff. Somit ist das Angreiferwissen ausschlaggebend für die Auswahl einer Angriffstechnik. Nach einem Angriff gewinnt der Angreifer im Idealfall neue Erkenntnisse, die sein Wissen erweitern. Das aktuelle Angreiferwissen beeinflusst die nachfolgenden Aktionen des Angreifers. Beispielsweise gewinnt der Angreifer nach einer Angriffsiteration Zugang zu neuen Verwundbarkeiten, die er in einer darauffolgenden Iteration, mithilfe der verfügbaren Angriffstechniken ausnutzen möchte.

Folgende Informationen sind z. B. je Cyber-Enabled Capability, für die Planung und Entwicklung eines Angriffs auf Basis von [5], für die Methodik zur Angriffsmodellierung relevant:

- Name
- Version
- Typ (z. B. Software, Hardware, Protokoll, Sicherheitsmechanismus)
- Bekannte (zugehörige) Vulnerabilities
- Anwendungsdomäne (z. B. Energieversorgung, Informationstechnik, Kommunikation, Transport/Verkehr, Medizinversorgung, Logistiksystem, Wasserversorgung, Versicherungssystem, Finanzsystem, Monitoringsystem, Gaming)

Auf dieser Grundlage werden die Angriffsmöglichkeiten in Form von Angriffstechniken, Zugangs- bzw. Angriffspunkten und Exploits abgeleitet, die dem Angreifer zu einem bestimmten Zeitpunkt zur Verfügung stehen. Z. B. ist jede Domäne mit typischen Zugangs- und Angriffspunkten verbunden, wie z. B. Daten (Informationen), Speichermedien, IT-Dienste, Software/Anwendungen, Kommunikationskanäle, Schnittstellen, Hardware oder eingebettete Systeme (engl. Embedded-Systems).[5] Je nach der Anwendungsdomäne gibt es diverse Komponenten, die nach den damit in Verbindung stehenden Zugangs- und Angriffspunkten analysiert werden müssen. Während z. B. im Bereich Transport und Verkehr insbesondere die Zugangs- und Angriffspunkte von Embedded-Systems in Betracht gezogen werden sollen, spielen bei Gaming-Plattformen oft die angreifbaren Stellen von Webapplikationen oder einer Cloud eine wesentliche Rolle.[5, 1] Die Anwendungsdomäne(n) des Targets bzw. dessen Umgebung beeinflusst mithilfe einer Menge an relevanten Zugangs- und Angriffspunkten den Aufbau des Angreiferwissens, unabhängig von der Charakteristik des Angreifers. Jeder Zugangs- bzw. Angriffspunkt ist wiederum mit gewissen Angriffstechniken bis hin zu Exploits verbunden. Darauf wird im weiteren Verlauf genauer eingegangen.

Das Wissen des Angreifers muss nicht zwangsläufig mit der tatsächlichen Umgebung übereinstimmen. Der Angreifer kann (als Mensch) etwas fehlerhaft interpretieren oder mit Absicht getäuscht werden. Verschiedenste Sicherheitsmechanismen können im Target und dessen Umgebung vorhanden sein, um Angriffe zu verhindern bzw. den Angreifer zu täuschen. Beispielsweise versuchen Verteidiger einem Angreifer eine reale Umgebung mithilfe von Honeypots vorzutäuschen, um ihn zu beobachten, ohne dass der Angreifer Schaden anrichten kann.[18] Unabhängig davon, ob der Angreifer getäuscht wird oder nicht, versucht er sich ein aktuelles Bild der Umgebung zu machen, um seine Angriffe zu planen und umzusetzen. Daher ist das dynamische Angreiferwissen ein notwendiges Modellelement.

Umgebung Die tatsächliche Darstellung des Targets und dessen Umgebung ist ein weiteres notwendiges Element für die Angriffsmodellierung. Die Umgebung stellt die Ausgangslage für die Simulation eines Angriffs in Form eines Exploits dar. Anhand einer Angriffssimulation werden die Ergebnisse einer Angriffsiteration gewonnen. Diese Ergebnisse bringen das Angreiferwissen auf den neuesten Stand. Auf dieses aktualisierte Fundament baut der Angreifer wiederum seine nächste Angriffsiteration auf, wie in Abschnitt 4.2 erklärt. Durch dieses Konzept ist eine Modellierung des Vorgehens des Angreifers über die Zeit darstellbar. Für eine geeignete Angriffssimulation im Rahmen der entwickelten Methodik sind folgende Aspekte bezüglich der Umgebung von Bedeutung:

- **Asset:** Anhand der Umgebung sollen die Assets abbildbar sein. Ein Asset stellt jegliches Objekt dar, das wertvoll für ein Unternehmen ist und daher geschützt werden muss.[18] Die Höhe des Werts ist für die qualitative Sicht auf einen Angriff nicht relevant.[83] Im Fokus stehen die Eigenschaften, die ein Assets beschreiben bzw. es „wertvoll“ machen. Das strategische Ziel des Angreifers steht in Verbindung mit bestimmten Assets, die der Angreifer anstrebt.

- **Sicherheitsmechanismen:** Neben den Assets spielen die Verbindungen zu anderen Elementen eine wichtige Rolle, wie beispielsweise zu Sicherheitskomponenten bzw. -mechanismen. Die Abbildung von Sicherheitsmechanismen ist notwendig, da sie die Angriffe abwehren können.[53] Infolgedessen können Sicherheitsmechanismen eine Begrenzung für die Angriffstiefe darstellen.

- **Angriffstiefe:** Mithilfe der Angriffstiefe soll sichtbar werden, wie tief der Angreifer bereits in das Target und dessen Umgebung vorgedrungen ist. Es soll sichtbar werden, welche Kenntnisse der Angreifer über das System bereits gewonnen hat und auf welche Stellen er Zugriff besitzt. Auf Basis der tatsächlichen Umgebung soll nach einer Angriffssimulation die aktuelle Angriffstiefe ableitbar sein.

- **Angriffsoberfläche:** Mithilfe von Angriffspunkten des Targets bzw. der Umgebung erhält man Zugriff auf Elemente des Targets und dessen Umgebung. In Summe bilden sie die Angriffsoberfläche, wie in 3.4 definiert. Im Zusammenhang mit der Angriffstiefe soll anhand der Umgebung sichtbar sein, welche Erkenntnisse der Angreifer nach einem Angriff (in Form einer Angriffssimulation) erzielt hat. Zugangspunkte sind nicht zwangsläufig für die Angriffssimulation relevant, wie z. B. Google, die als Suchmaschine zur Informationssammlung verwendet wird. Allerdings sollen Zugangspunkte in geeigneter Weise für die Angriffssimulation abgebildet werden, sobald ein Zugangspunkt – als Bestandteil des Targets bzw. der Umgebung – maßgebend für die Angriffssimulation ist.

- **Vollständigkeit:** Die Umgebung soll vollständig abgebildet sein. Das bedeutet nicht, dass alle möglichen Daten und Informationen des Targets und dessen Umgebung repräsentiert werden müssen. Vielmehr sollen alle Daten zur Verfügung stehen, die für eine Angriffssimulation bezüglich der Angriffsmodellierung erforderlich sind.

Die Umgebung stellt zum einen die Ausgangslage für die Angriffssimulation dar. Zum anderen präsentiert sie die Ergebnisse eines simulierten Angriffs. Somit ist die Umgebung ein relevantes Element für die Angriffsmodellierung. Sie ist notwendig, um die Ergebnisse bzw. den Erfolg eines Angriffs abzubilden.

Exploit Ein weiteres bedeutsames Element für die Angriffsmodellierung ist der Exploit. Ein Exploit repräsentiert in geeigneter Weise eine bestimmte Funktionalität, wobei für dessen Ausführung eine spezifische Stelle ausgenutzt wird. Außerdem ist ein Exploit mit Verwundbarkeiten verbunden, wenn die Ausführung des Exploits die Ausnutzung von Vulnerabilities präsentiert. In diesem Zusammenhang steht ein Exploit immer in Verbindung zu gewissen Angriffstechniken, worauf weiter unten in diesem Kapitel eingegangen wird. Aus dem Element Exploit der Angriffsmodellierung soll ein konkreter Programmcode ableitbar sein. Dieser Code kann zum einen als Input für geeignete Werkzeuge fungieren, wie z. B. für Penetrationswerkzeuge. Zum anderen soll der Exploit für eine Angriffssimulation zur Anwendung kommen. Das Modellelement Exploit ist mit bestimmten Vorbedingungen und Nachbedingungen zu versehen.[28] Exploits sind in einer gewissen Reihenfolge auszuführen, um letztendlich das strategische Angriffsziel zu erreichen, wie in Abschnitt 4.2 verdeutlicht. Die Vorbedingungen entsprechen gewissen Voraussetzungen für die Auswahl des Exploits. Eine Voraussetzung kann die vorherige Ausführung anderer Exploits sein. Beispielsweise muss für die konkrete Ausnutzung einer Verwundbarkeit zunächst getestet werden, ob die Vulnerability existiert. Das bedeutet z. B., dass ein Exploit zum Auffinden einer Verwundbarkeit bereits erfolgreich ausgeführt sein muss, bevor ein Exploit zur Ausnutzung der Verwundbarkeit durchgeführt werden kann. Unter Umständen ist die vorherige Ausführung von bestimmten Exploits nicht notwendig, wenn die erforderlichen Elemente (Voraussetzungen des Exploits) im Angreiferwissen bereits vorhanden sind. Daher ist eine Entscheidung mithilfe des aktuellen Angreiferwissens über das Target und dessen Umgebung notwendig. Das aktuelle taktische Ziel des Angreifers bestimmt neben den Vorbedingungen die Ausführungsreihenfolge der Exploits. Stehen beispielsweise zu einem Zeitpunkt sämtliche Aufklärungsaktivitäten im Fokus, so sind diejenigen Exploits auszuwählen, die mit diesen Informationsbeschaffungsaktivitäten in Verbindung stehen. Exploits, die mit anderen Zielen verknüpft sind sollen zu diesem Zeitpunkt nicht ausgewählt werden, wie z. B. Exploits für die Injektion von Schadsoftware oder bezüglich Spionage-Aktivitäten. Die Nachbedingungen eines Exploits präsentieren die Ergebnisse nach einer Ausführung des Exploits (Angriffsiteration). Zum einen wird das Angreiferwissen gemäß den Ergebnissen aktualisiert, da es als Entscheidungsgrundlage für das weitere Vorgehen dient. Zum anderen können die Ergebnisse eine Voraussetzung bzw. Vorbedingung für die Ausführung weiterer Exploits sein, wie bereits in diesem Abschnitt beschrieben. Daran ist ersichtlich, dass ein Exploit ein wesentliches Element der Angriffsmodellierung ist, da er die Verbindung zwischen der Angriffssimulation und dem systematischen Vorgehen des Angreifers abbildet.

Zugangspunkt und Angriffspunkt Die Zugangs- und Angriffspunkte sind weitere notwendige Elemente für die Methodik zur Angriffsmodellierung. Diese Punkte stellen dem Angreifer Informationen für einen Angriff zur Verfügung und können unter Umständen ausgenutzt werden, um sich Zugang zu Informationen oder Zugriff auf ein System zu verschaffen, wie bereits in Abschnitt 4.2 erläutert. Jeder Punkt kann über bestimmte Angriffstechniken analysiert oder ausgenutzt werden. Beispielsweise stellt die „URL“ einen Zugangs- bzw. Angriffspunkt für die Techniken „URL Encoding“ und „SQLI“ dar.[10] Je nach Angriffstechnik wird die „URL“ auf verschiedene Weise analysiert und modifiziert. Für weitere Details siehe [10]. Somit führt ein bestimmter Zugangs- bzw. Angriffspunkt zu einer begrenzten Menge an möglichen Angriffstechniken. Das bedeutet der Angriff wird durch die Auswahl eines Zugangs- bzw. Angriffspunkts beeinflusst, indem anschließend nur die damit verbundenen Angriffstechniken für den Angriff infrage kommen. Daran ist ersichtlich, dass die Zugangs- bzw. Angriffspunkte als Element für die Angriffsmodellierung von Bedeutung sind. Die Auswahl wird beispielsweise durch den Scope des Angreifers und dessen aktuellem Wissens beeinflusst, wie bereits zuvor in den Paragraf „Angreifer“ und „Angreiferwissen“ erwähnt. Zudem spielen das strategische Ziel und der bevorzugte Angriffsbereich eines Angreifers eine entscheidende Rolle für die Identifikation, wie in Abschnitt 6.2 zu lesen. Dennoch stehen die genauen Vorgehensweisen zur Auswahl einer spezifischen Menge und eines bestimmten Zugangs- bzw. Angriffspunkts nicht im Mittelpunkt dieser Arbeit.

Angriffstechnik Schließlich wird die Angriffstechnik als Element für die Angriffsmodellierung benötigt. Wie bereits in Abschnitt 4.2 erwähnt, stellen Angriffstechniken verschiedene Möglichkeit dar, wie ein taktisches Ziel erreicht werden kann. Jede Angriffstechnik wird durch eine Menge an Zugangs- bzw. Angriffspunkten und Exploits definiert. Daneben kann eine Technik mit Vulnerabilities in Verbindung stehen. Wie in Abschnitt 4.2 beschrieben, ist eine Vulnerability implizit über die Attribute Angriffspunkt, Exploit und Angriffstechnik abbildbar. Die Angriffstechnik ist ein notwendiges Element für die Modellierung, da anhand der Technik die verschiedenen Zugangs- bzw. Angriffspunkte, Exploits und Vulnerabilities miteinander in Beziehung gesetzt werden können. Dadurch wird der Aspekt der Wiederverwendbarkeit des Modellierungskonzepts unterstützt. Die Angriffstechniken, sowie Zugangs- und Angriffspunkte und Exploits können mehrfach miteinander in Verbindung stehen und sollen daher wiederverwendbar sein. Z. B. kann eine „URL“ für die Angriffstechniken „Normal SQLI“, „Blind SQLI“, „URL Encoding“ oder für „Path Traversal“ als Angriffspunkt fungieren.[62, 18, 9, 10]

Eine Angriffstechnik kann wiederum mit mehreren Zugangs- bzw. Angriffspunkten verknüpft sein. Z. B. kann die SQLI-Angriffstechnik sowohl über eine „URL“, als auch über ein „User Input Field“ als Zugangs- bzw. Angriffspunkt verwendet werden.[8] Des Weiteren repräsentiert z. B. Google als Suchmaschine einen sehr mächtigen Zugangspunkt, der für viele Angriffstechniken im Hinblick auf eine allgemeine Informationsrecherche über das Target und dessen Umfeld herangezogen werden kann. Aufgrund der Zusammenhänge

zwischen Angriffstechnik, Exploits und Vulnerability setzt die Angriffstechnik bestimmte Gegebenheiten bezüglich des Targets und dessen Umgebung voraus. In Abschnitt 4.2 wird diese Thematik bereits in der Charakteristik einer Vulnerability erwähnt. Die Auswahl der Angriffstechnik hängt zum einen vom Angreifer ab, zum anderen von dem aktuellen Angreiferwissen über das Target und dessen Umgebung. Der Angreifer besitzt gewisse Fachkenntnisse und hat bestimmte Ressourcen für den Angriff zur Verfügung, die für die Auswahl der Technik relevant sind, wie bereits weiter oben, unter dem Abschnitt „Angreifer“ aufgezeigt. Außerdem ist das Angreiferprofil mit gewissen, bevorzugten Angriffsbereichen charakterisiert, die die priorisierte Auswahl von Angriffstechniken unterstützt. Ebenso beeinflusst das aktuelle Bild bzw. Wissen des Angreifers die Angriffstechniken, die dem Angreifer zu einem bestimmten Zeitpunkt zur Verfügung stehen, wie bereits in dieser Passage erwähnt. Mithilfe des Modellelements Angriffstechnik können die verschiedenen Aspekte eines Angriffs aus technischer Sicht in Verbindung gebracht werden. Gleichzeitig bieten diese Zusammenhänge eine hohe Erweiterbar- und Wiederverwendbarkeit der Modellierungsmethodik.

5 Modellkontext

In diesem Abschnitt wird der Kontext der entwickelten Methodik zur Angriffsmodellierung bzw. des Angriffsmodells, worauf die Methodik basiert erläutert. Das gesamtheitliche Angriffsmodell steht mit verschiedenen Elementen in Verbindung, wie in Abschnitt 4.3 analysiert. Diese Abhängigkeiten werden zunächst beleuchtet, um den Scope des generischen Angriffsmodells und den Fokus innerhalb dieser Arbeit zu bestimmen. Anschließend folgt im Überblick der Aufbau des Modells, bestehend aus verschiedenen Ebenen und deren Zusammenhänge. Daneben spielt die Angriffssimulation für das Konzept zur Angriffsmodellierung eine entscheidende Rolle, auf die abschließend kurz eingegangen wird.

5.1 Abhängigkeiten

Ein Angriff wird von einigen Faktoren beeinflusst, wie z. B. von dem Target, dessen Umgebung, oder von dem Angreifer an sich, wie in Abschnitt 4.2 erläutert. Auf Grundlage der analysierten charakteristischen Merkmale eines Angriffs in Abschnitt 4.2 sind die notwendigen Modellelemente in Abschnitt 4.3 identifiziert. Die Methodik zur Angriffsmodellierung hängt von diesen Elementen ab bzw. baut darauf auf. Daher wird für das weitere Vorgehen der Arbeit angenommen, dass die nachfolgenden Modelle und Sammlungen an Informationen bezüglich der identifizierten Modellelemente bereits existieren. Diese Annahmen basieren auf den aktuellen Arbeiten der Forschungsgruppe von Prof. Dr.-Ing. Hans-Joachim Hof der Technischen Hochschule Ingolstadt und dem derzeit laufenden Projekt „Modellbasierte Absicherung von Security und Safety für umfeldbasierte Fahrzeugfunktionen (MASSiF)[1]“, welches in Zusammenarbeit mit der Technischen Hochschule durchgeführt wird. Der Fokus der Masterarbeit liegt nicht auf den Details dieser Abhängigkeiten in Form von Modellen und Sammlungen, sondern auf der Entwicklung einer gesamtheitlichen Methodik zur Angriffsmodellierung.

[1]Dieses Projekt wird im Rahmen des Programms „KMU-innovativ“ von dem Bundesministerium für Bildung und Forschung unter der Förderungsnummer 16KIS0946 unterstützt.

Angreifermodell Ein Angreifermodell (engl. Attacker Model) soll den Angreifer als Informationsquelle für die Methodik zur Angriffsmodellierung repräsentieren. Das bedeutet, mithilfe eines geeigneten Angreifermodells sollen alle notwendigen Charakteristiken des Angreifers, wie in Abschnitt 4.3 beschrieben, in einem separaten Modell abgebildet und als eine Basis für die Methodik zur Angriffsmodellierung dienen. Die Idee besteht darin, das Angriffsmodell, welches auf dieser entwickelten Methodik basiert, mit verschiedenen Angreiferprofilen auszuführen.[45] Beispielsweise besitzt ein interner Angreifer zusätzliche Angriffsmöglichkeiten, im Vergleich zu einem Angreifer von außen.[45] Mithilfe des Angreifermodells sollen verschiedenen Angreiferprofile abbildbar sein. Das unterstützt den generischen Charakter der Methodik zur Angriffsmodellierung, da verschiedene Angreifer anhand eines Modells darstellbar und wiederverwendbar sind. Daher ist eine Unterscheidung zwischen Angriffen und Angreifer bzw. die Abbildung des Angreifers in einem gesonderten Modell notwendig, das kompatibel zu der entwickelten Methodik zur Angriffsmodellierung ist.

Umgebungs-Wissens-Modell Mithilfe des Umgebungs-Wissens-Modells (engl. Environment Knowledge Model) soll das Angreiferwissen, wie in Abschnitt 4.3 aufgezeigt, separat und dynamisch modelliert werden. Das Umgebungs-Wissens-Modell präsentiert in geeigneter Weise, zu einem bestimmten Zeitpunkt das aktuelle Wissen des Angreifers über das Target und dessen Umgebung, das er angreifen will. Diese Informationen sind notwendig, um das Vorgehen eines Angreifers aufzuzeigen und somit den Angriff modellieren zu können. Der Angreifer versucht die für ihn relevanten Informationen über das Target und dessen Umgebung iterativ aufzuspüren. Dabei kann der Angreifer getäuscht werden bzw. fehlerhafte Schlussfolgerungen ziehen. Außerdem kann der unberechtigte Zugriff auf Informationen durch Sicherheitsmechanismen verhindert werden. Infolgedessen bildet das Umgebungs-Wissens-Modell nicht das reale Target und dessen Umfeld ab, sondern dieses Modell repräsentiert Annahmen, die ein Angreifer über das Target und dessen Umgebung trifft und worauf er seinen Angriff iterativ plant und entwickelt.

Umgebungsmodell Ein Umgebungsmodell (eng. Environment Model) präsentiert die vollständige Umgebung des Targets und ist für die Simulation des modellierten Angriffs notwendig. „Vollständig“ bedeutet in diesem Zusammenhang, dass alle relevanten Informationen für eine Angriffssimulation vorhanden sind, wie bereits in Abschnitt 4.3 erläutert. Im Rahmen einer Angriffsiteration soll ein identifizierter Exploit auf einem passenden Umgebungsmodell simuliert werden können. Mithilfe des Umgebungsmodells sollen die aus der Angriffssimulation resultierenden Ergebnisse sichtbar und in geeigneter Weise in das Umgebungs-Wissens-Modell integriert werden können. Beispielsweise gewinnt der Angreifer mithilfe des Exploits Zugriff auf geheime Informationen oder Kenntnis von neuen Zugangs- und Angriffspunkten, um noch tiefer in das Target und dessen Umgebung vorzudringen. Das aktualisierte Umgebungs-Wissens-Modell stellt über weitere Angriffsiterationen die neue Ausgangslage für die Planung, Entwicklung

und Ausführung von nachfolgenden Angriffen dar. Wie bereits zuvor erläutert, entspricht das Umgebungs-Wissens-Modell weder einer Untermenge des Umgebungsmodells noch dem Umgebungsmodell an sich. Das Umgebungsmodell stellt die vollständige Umgebung des Targets für eine Simulation dar, während das Umgebungs-Wissens-Modell das Angreiferwissen zu einem bestimmten Zeitpunkt abbildet. Im Vergleich zur tatsächlichen Umgebung (Umgebungsmodell) kann das Umgebungs-Wissens-Modell Unterschiede aufweisen. Das bedeutet, Exploits, die auf dem Umgebungs-Wissens-Modell geplant und entwickelt sind, sollen durch die Angriffssimulation auf dem Umgebungsmodell die dazugehörigen Ergebnisse liefern. Dazu zählen auch die Ergebnisse von Täuschungen durch Sicherheitsmechanismen. Des Weiteren soll die Angriffstiefe grafisch abbildbar sein. Z. B. bietet die Angriffstiefe für die Zielgruppe von Sicherheitsbeauftragten die Grundlage zur Ableitung von notwendigen Sicherheitsmechanismen.

Sammlung von Zugangspunkten Es ist eine Sammlung aller bekannter Zugangs- und Angriffspunkte erforderlich. Es soll eine zentrale Stelle geben, die alle bekannten Zugangs- und Angriffspunkte in einer strukturierten Form als Quelle für die Angriffsmodellierung bereitstellt. Dabei sollen alle notwendigen Informationen zu den Punkten enthalten sein, wie die zugehörige Anwendungsdomäne (z. B. eingebettete Systeme) oder der Scope (z. B. lokaler Scope). Einige Informationen zu Zugangs- und Angriffspunkten sind beispielsweise über die CAPEC-Datenbank erhältlich. Allerdings liegt dort der Fokus auf der Präsentation von Angriffsmustern, statt Zugangs- bzw. Angriffspunkten. Die Punkte sind lediglich als zusätzliche Information vereinzelt aus dem Fließtext bzw. über die verschiedenen Views auf die Angriffsmuster aus CAPEC ableitbar. Wie bereits in Abschnitt 4.2 definiert, ist ein Angriffspunkt eine mögliche Erweiterung eines Zugangspunkts, daher steht „Sammlung von Zugangspunkten“ (engl. Access Point Library) für die Sammlung von allen bekannten Zugangs- und Angriffspunkten.

Sammlung von Angriffstechniken Zudem ist die Sammlung von Angriffstechniken (engl. Attack Technique Library) für die entwickelte Methodik von Bedeutung. Alle bekannten Angriffstechniken sollen in geeigneter Form strukturiert an einer zentralen Stelle auffindbar sein. Mithilfe dieser Sammlung werden Angriffe nach Techniken klassifiziert, sodass die Wiederverwendbarkeit und Verständlichkeit der Methodik zur Angriffsmodellierung erhöht wird. Wie bereits in Abschnitt 4.3 erläutert, setzt sich eine Technik unter anderem aus Zugangs- bzw. Angriffspunkten und Exploits zusammen. Anhand einer zentralen Stelle, sollen diese Zusammenhänge in geeigneter Weise abrufbar sein, sodass alle bekannten Techniken nach entsprechenden Attributen, wie z. B. Angriffs- bzw. Zugangspunkt, Exploit oder Vulnerability definiert, ausgewählt und im Angriffsmodell verwendet werden können. Zudem sollen die Angriffstechniken mit wichtigen Merkmalen versehen werden, die für die Methodik zur Angriffsmodellierung erforderlich sind. Dazu zählen z. B. die Voraussetzungen in der Umgebung für die Anwendung der Technik und inwieweit spezielle Kompetenzen oder Ressourcen notwendig sind.

Sammlung von Exploits Die Sammlung von Exploits (engl. Exploit Library) ist eine weitere Quelle an Informationen für die entwickelte Methodik zur Angriffsmodellierung. Sie ist notwendig, um eine Auswahl von Exploits im Angriffsmodell zu präsentieren. An einem zentralen Punkt sollen alle bekannten Exploits in geeigneter Weise strukturiert auffindbar sein. Dabei stehen Exploits in Verbindung zu Angriffstechniken. Ziel ist es, dass die grundlegenden Gerüste von Exploits wiederverwendbar sind, z. B. in Form von gemeinsamen Attributen. Beispielsweise unterscheidet sich das Vorgehen innerhalb eines Exploits für den Test einer SQLI-Verwundbarkeit durch die verschiedenen Syntax-Varianten der Datenbankmanagementsysteme, wie z. B. „MySQL“, „Oracle“, „PostgreSQL“ oder „Microsoft SQL Server“.[32] Eine Gruppierung von Exploits kann daher die Wiederverwendbarkeit unterstützen. Ein Exploit enthält alle notwendigen Attribute für die zweckmäßige Anwendung in der Methodik zur Angriffsmodellierung, wie beispielsweise welche Exploits als Voraussetzung bereits durchgeführt sein sollten, welche Kenntnisse über die Umgebung vorhanden sein müssen und das zugehörige taktische Ziel, worauf der Exploit abzielt.

Diese Modelle und Informationsquellen sind so zu entwickeln, dass sie kompatibel zur entwickelten Methodik zur Angriffsmodellierung sind. Daraus ist ersichtlich, dass das Angriffsmodell nicht isoliert zu betrachten ist, sondern abhängig von diesen Sammlungen an Informationen und Modellen. Sie stellen dem Angriffsmodell die notwendigen Inhalte bereit, die auf geeignete Art und Weise importiert und korrekt interpretiert werden müssen. Die Details zum Vorgehen sind in Kapitel 6 zu finden. Gleichzeitig sollen die Datenquellen wiederverwendbar sein. Diese grundlegende Anforderung bildet das Fundament für das generische Angriffsmodell, wie in Abschnitt 4.1 beschrieben.

5.2 Überblick der Methodik

An dieser Stelle wird die Methodik zur Angriffsmodellierung im Überblick vorgestellt. Ein Angriff kann in „kleinere“ Angriffe aufgeteilt werden. Die genaue Bezeichnung für „kleinere“ Angriffe ist abhängig von der Sichtweise auf den Angriff, wie bereits in Abschnitt 4.2 erläutert. Generell besteht ein Angriff aus Aktionen, wie in 3.4 definiert. Des Weiteren kann ein Angriff auf Prozess-Ebene in Angriffsiterationen und aus technischer Sicht, mithilfe von Techniken bzw. Exploits abgebildet werden. Anhand dieser Einheiten soll jeder Angriff modellierbar sein. Über verschiedene Ebenen bilden die identifizierten Elemente aus Abschnitt 4.3 das gesamtheitliche Angriffsmodell. Dabei stellen die aufgezeigten Abhängigkeiten aus Abschnitt 5.1 die notwendigen Inhalte bereit, deren Existenz angenommen wird. Die Ebenen des Angriffsmodells sind miteinander verknüpft und präsentieren die verschiedenen Sichtweisen auf einen Angriff. Die hierarchische Darstellung von Ebenen bietet für jeden Anwendungsfall einen geeigneten Überblick bezüglich einer bestimmten Sichtweise auf einen Angriff. Durch diese Abstraktion soll die Komplexität von Angriffen handhabbar sein. Beginnend mit der Taktik-Ebene, über die

Ablauf-Ebene, bis hin zur Technik-Ebene, erhält man zunehmend mehr Details über den Angriff. Abbildung 5.1 zeigt die Zusammenhänge der entwickelten Methodik zur Angriffsmodellierung im Überblick, die im Nachfolgenden beschrieben werden. Die Details zu den einzelnen Schichten, dem Vorgehen, der Angriffssimulation und der Zusammenhang mit den Informationsquellen werden in Kapitel 6 der Arbeit erläutert.

Taktik-Ebene Zunächst wird der grobe Verlauf eines Angriffs bezogen auf ein strategisches Ziel in der Taktik-Ebene (engl. Tactic Layer) abgebildet. Wie in Abbildung 5.1 unter Tactic Layer zu sehen ist, erhält man in Form eines einfachen Prozessflussdiagramms einen Überblick des gesamten Angriffs eines Angreifers bezüglich seines strategischen Vorgehens. Die Präsentation des Angriffs erfolgt über mehrere taktischen Ziele, was die Handhabung des Angriffs vereinfacht. Das Prinzip über mehrere Stufen wird auch bei der „Lockheed Martin Cyber Kill Chain“ verwendet, indem der Angriff als Prozess über sieben Phasen aufgeteilt wird.[48] Die Taktik-Ebene gibt die Rahmenbedingungen und die Zielrichtung für die darunterliegenden Ebenen (Ablauf-Ebene, Technik-Ebene) vor, da die Ebenen auf der Strategie des Angreifers basieren. Dabei werden noch keine Aussagen zu der genauen Umsetzung der Taktiken gemacht.

Ablauf-Ebene Jede Taktik der Taktik-Ebene wird, wie in Abbildung 5.1 abgebildet, mithilfe der Ablauf-Ebene (engl. Procedure Layer) spezifiziert. In der Ablauf-Ebene wird eine Taktik durch die Gesamtheit an möglichen Abläufen modelliert. Im Kontext von Entscheidungen und Bedingungen erhält man eine Übersicht der notwendigen Aktionen innerhalb einer Taktik. Dabei wird das Angreiferwissen mit jeder Angriffsiteration dynamisch aufgebaut. Aktuelle Informationen zu dem Target und dessen Umgebung werden systematisch, zu einem bestimmten Zeitpunkt, anhand des Angriffsablaufs auf Basis von UML-Aktivitätsdiagrammen festgehalten. Ein Angriffsablauf ist damit systematisch, iterativ und zielorientiert definierbar, was die Voraussetzung für die Modellierung der darunterliegenden Ebene bildet. Die Ablauf-Ebene fokussiert sich auf die Realisierung jedes taktischen Ziels aus der darüber liegenden Taktik-Ebene. Somit bleibt das Modell in der Taktik-Ebene übersichtlich. Gleichzeitig erhält man, durch eine Ebene darunter, detaillierte Informationen der Aktionen von möglichen Abläufen eines Angriffs, basierend auf einem bestimmten taktischen Ziel.

Technik-Ebene Während in der Ablauf-Ebene die Modellierung der Reihenfolge von Angriffsaktionen im Mittelpunkt steht, werden mithilfe der Technik-Ebene (engl. Technique Layer) die Angriffsmöglichkeiten und die damit verbundene Handlungsmittel aufgezeigt. Eine bestimmte Aktion aus der Ablauf-Ebene wird, wie in Abbildung 5.1 zu sehen, in der Technik-Ebene durch eine zusammenhängende Abbildung von Exploits spezifiziert. Somit ist die Verbindung zwischen dem strategischen Ziel (Taktik-Ebene) bis hin zu einem konkreten Angriff (zu einem bestimmten Zeitpunkt) in Form eins

Exploits geschaffen. Anhand konkreter Exploits entsteht die Verbindung zu existierenden Vulnerabilities, die wiederum konkrete Details zu dem verwundbaren System repräsentieren. Die Darstellung der Technik-Ebene erfolgt auf Grundlage von Attack Trees. Die Zusammenhänge beziehen sich dabei auf einen bestimmten Zugangs- oder Angriffspunkt als Wurzel. Die schwarzen Blätter der Baumstruktur in Abbildung 5.1 stellen die Exploits dar. Eine Traversierung des Baumes ist über verschiedene Angriffstechniken bis hin zu den zugehörigen Exploits möglich. Folglich werden in der Technik-Ebene die verschiedene Exploits über eine Baumstruktur bezüglich eines Zugangs- bzw. Angriffspunkts nach Angriffstechniken klassifiziert. Ziel in dieser Ebene ist die Bestimmung eines Exploits für eine darauffolgende Angriffssimulation. Die Wiederverwendbarkeit und Taversierungsmöglichkeit von Bäumen wird in dieser Ebene ausgenutzt.

Angriffssimulation Schließlich ist die Simulation des zuvor bestimmten Exploits aus der Technik-Ebene notwendig, um zu entscheiden, ob der Angreifer sein strategisches Ziel erreicht hat und somit der Angriff beendet ist. Ein Angriff besteht aus einer Menge an Angriffsiterationen, wie bereits in den einleitenden Sätzen in diesem Abschnitt 5.2 erwähnt. Mit jeder Iteration baut der Angreifer sein Wissen zum Target und dessen Umgebung aus. Das Angreiferwissen bildet die Grundlage für das weitere Vorgehen, wie z. B., ob das strategische Ziel zum jetzigen Zeitpunkt erreicht ist, oder was dafür als Nächstes getan werden muss. Eine Angriffsiteration beinhaltet zum einen die Auswahl eines Exploits (Technik-Ebene) und zum anderen die Simulation dieses Angriffs. Ausgehend von einem zuvor bestimmten Exploit aus der Technik-Ebene soll daraus eine geeignete Form zur Simulation abgeleitet werden können z. B. Exploit-Code, wie in Abbildung 5.1 zu sehen. Zudem liegt die Erwartung darin, geeignete Werkzeuge mithilfe des Exploit-Codes zu parametrisieren, sodass Security-Tests, wie beispielsweise Penetrationstests unterstützt werden können. Für die Simulation eines Angriffs (Exploit) ist ein geeignetes Umgebungsmodell notwendig, wie in Abbildung 5.1 unter „Environment Model“ abgebildet. Im Idealfall gewinnt der Angreifer neue Informationen mithilfe der Angriffssimulation, die den Angreifer näher zu seinem strategischen Ziel bringen bzw. dieses erfüllen. In Abbildung 5.1 stellt der schwarze Pfeil „Results“ die neuen Informationen (Ergebnisse) nach der Simulation eines Angriffs dar. Die gewonnenen Informationen aus der Angriffssimulation werden über die Taktik-Ebene an geeigneter Stelle wieder in das Angriffsmodell integriert. Damit ist der dynamische Wissensaufbau des Angreifers für die Methodik der Angriffsmodellierung abbildbar. Die Angriffssimulation ist für die Terminierung der Methodik zur Angriffsmodellierung wichtig, da anhand deren Ergebnisse das Angreiferwissen (Entscheidungsgrundlage für die Terminierung) aktualisiert wird.

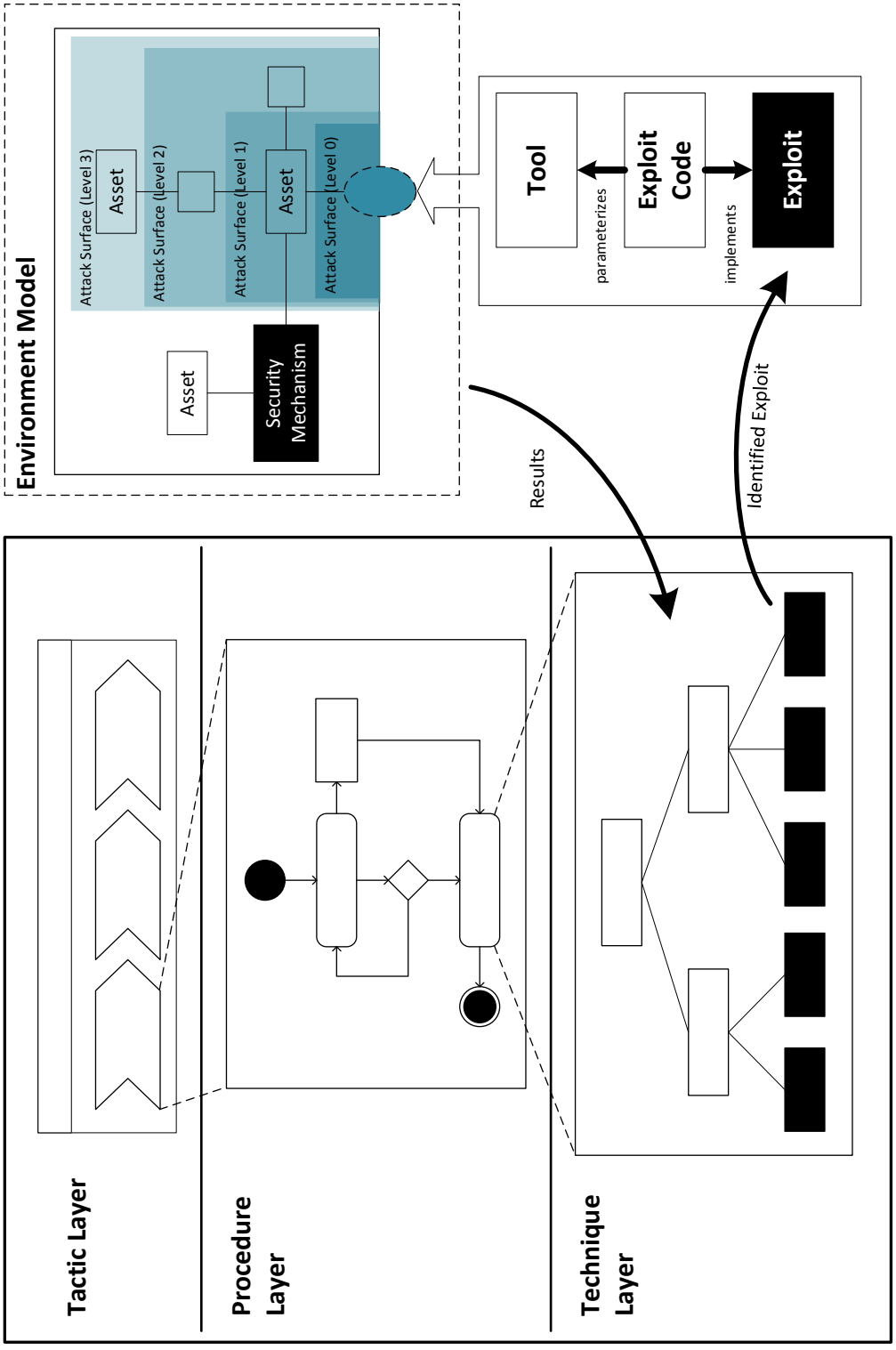

Abbildung 5.1: Überblick des Konzepts zur Angriffsmodellierung

Zusammenfassend betrachtet wird ein Angriff anhand der entwickelten Methodik im Rahmen von drei Ebenen abgebildet. Dabei wird ein Angriff iterativ mithilfe einer Angriffssimulation entwickelt. Die drei aufeinanderfolgenden Ebenen Taktik-, Ablauf- und Technik-Ebene unterscheiden sich bezüglich der Sichtweise auf einen Angriff und den damit verbundenen Detaillierungsgrad. In der Taktik-Ebene wird das strategische Gesamtvorgehen eines Angreifers im Hinblick auf seine Zielrichtung präsentiert. Zur Erreichung eines spezifischen taktischen Ziels werden die einzelnen Angriffsaktionen in der Ablauf-Ebene modelliert. Die Aktionen sind in einer bestimmten Reihenfolge durchzuführen und mit entsprechenden Entscheidungen verknüpft. Die Details einer Aktion der Ablauf-Ebene werden in der Technik-Ebene spezifiziert. Die Technik-Ebene stellt die Grundlage zur Auswahl eines bestimmten Exploits bereit, der anschließend auf einem geeigneten Umgebungsmodell simuliert wird. Die Ergebnisse dieser Angriffssimulation werden an den entsprechenden Stellen des Angriffsmodells integriert. Das aktualisierte Angreiferwissen bildet eine Grundlage für die Entscheidung der Terminierung des Angriffs. Die verschiedenen Modelle und Sammlungen an Informationen stellen die Datenbasis für den Aufbau des Angriffsmodells zur Verfügung und bilden die Entscheidungsgrundlagen für die Methodik zur Angriffsmodellierung. Dazu zählt das Angreifermodell, Umgebungs-Wissens-Modell, Umgebungsmodell und die Sammlungen von Zugangspunkten, Angriffstechniken und Exploits.

6 Methodik zur Angriffsmodellierung

In diesem Kapitel wird das Vorgehen der Methodik zur Angriffsmodellierung in Zusammenhang mit den Ebenen erläutert. Neben der allgemeinen Vorgehensbeschreibung wird wiederholt auf ein zusammenhängendes Beispiel eingegangen, um das systematische Verfahren zu veranschaulichen. Das fortlaufende Beispiel wird in jeder Ebene mit notwendigen Details erweitert, die auf den verfügbaren Informationen aus öffentlichen Datenbanken basieren, die in Abschnitt 3.6 beschrieben sind. Die genaue Umsetzung und die Details zur Implementierung stehen nicht im Fokus dieser Arbeit.

6.1 Taktik-Ebene

Das strategische Ziel eines Angriffs ist ein essentielles Element für das Angriffsmodell. Ohne ein Angriffsziel gibt es keinen Angriff. Diese Information wird durch das Angreifermodell bereitgestellt. Die Art und Weise der Modellierung des Angriffsziels ist nicht Teil dieser Arbeit. Abbildung 6.1 zeigt den allgemeinen Aufbau der Taktik-Ebene. Das strategische Ziel wird in einem einfachen Prozessflussdiagramm über verschiedene Taktiken erreicht, wie in Abschnitt 4.2 definiert. Das Diagramm ist einfach gehalten, sodass keine unnötigen Details die Übersichtlichkeit auf der Ebene stören. Die Taktik-Ebene soll einen Überblick des gesamten Angriffs bezüglich eines strategischen Ziels geben.

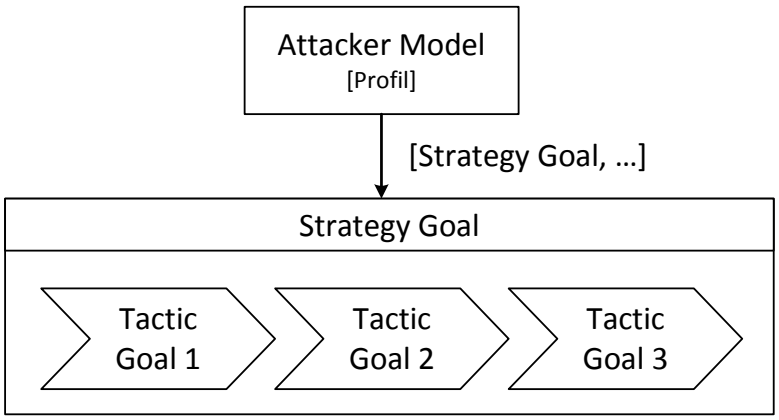

Abbildung 6.1: Modellierung der Taktik-Ebene

Für die Übersichtlichkeit wird jede Taktik in dieser Ebene durch ein taktisches Ziel in dem einfachen Prozessflussdiagramm abgebildet, die in einer geeigneten Reihenfolge ausgeführt werden soll. Diese Eigenschaft ist notwendig, damit das Angriffsmodell flexibel für alle möglichen Taktiken und die damit verbundenen Angriffe erweiterbar ist. Für die Methodik zur Angriffsmodellierung wird angenommen, dass eine Menge an definierten Taktiken bzw. an taktischen Zielen für das Angriffsmodell existiert. Beispielsweise steht das taktische Ziel „Explore“ für sämtliche Aspekte und Aktivitäten, die notwendig sind, um das Target und dessen Umfeld zu entdecken, wie Informationen zu Schwachstellen bzw. Verwundbarkeiten darüber zu sammeln.[8, 74, 63] Dazu zählen Aktionen des Verwundbarkeiten-Managements (engl. Vulnerability Assessment).[44] Das bedeutet, dass das taktische Ziel „Explore“ lediglich die Identifikation von Verwundbarkeiten umfasst. Das Ziel der Ausnutzung von Vulnerabilities wird über ein anderes taktisches Ziel modelliert, z. B. namens Exploit[1].[74] Wegen dem bemessenen Rahmen einer Masterarbeit steht das taktische Ziel Explore im Mittelpunkt.

Die einzelnen Phasen der Cyber Kill Chain von Lockheed Martin lassen sich z. B. als Taktiken interpretieren.[63] Allerdings sind diese auf APTs ausgelegt. Zudem stellen die Phasen eine Mischung aus Prozesssichtweise und der technischen Sicht auf einen Angriff dar. Beispielsweise gehört „Weaponization“ in jegliche Taktik eines Angriffs, da für einen Angriff, unabhängig von dem Ziel, verschiedene Werkzeuge verwendet werden können. Diese Thematik wird über die Technik-Ebene abgedeckt.

Die Auswahl der Taktiken bzw. der taktischen Ziele, um das strategische Ziel zu erreichen, wird durch Informationen aus dem Angreifermodell entschieden. Dafür sind die Taktiken mit geeigneten Vor- und Nachbedingungen verknüpft. Auf Grundlage des einfachen Prozessflussdiagramms wird versucht, die taktischen Ziele in einer geeigneten Reihenfolge, nacheinander zu bewältigen, sodass das übergeordnete, strategische Ziel erfolgreich erreicht wird. Allerdings können sich die Umstände während eines Angriffs stetig ändern. Vor jeder Umsetzung einer Taktik wird daher geprüft, ob das strategische Ziel erreicht ist oder nicht. Ist das strategische Ziel noch nicht erzielt, wird die nächstliegende (zielführende) Taktik identifiziert und realisiert. Das erfolgt anhand der Vorbedingungen einer Taktik. Die Vorbedingungen einer Taktik stellen Voraussetzungen dar, die notwendig sind, um die Taktik umzusetzen. Beispielsweise umfasst das taktische Ziel „Maintain“ sämtliche Aktivitäten und Aspekte, um im System des Opfers unbemerkt zu verweilen.[25] Diese Taktik setzt beispielsweise die taktischen Ziele „Explore“ und „Exploit“ voraus. Der Angreifer muss bereits Zugang zu dem System des Opfers besitzen, um Aktionen zur Beständigkeit im Target zu ergreifen. Die Nachbedingungen einer Taktik entsprechen den Ergebnissen einer durchgeführten Taktik, z. B. ob diese erfolgreich war oder abgebrochen wurde. Beispielsweise wird versucht, eine abgebrochene Taktik im späteren Verlauf des Angriffs erneut durchzuführen, da sich die Ausgangslage geändert hat. Daneben sind weitere beschreibende Attribute für eine Taktik denkbar. Beispielsweise können

[1] Das taktische Ziel Exploit ist nicht zu verwechseln mit dem Exploit(-Code) aus der Begriffsdefinition in Abschnitt 3.1 bzw. 3.4 oder mit dem Modellelement Exploit aus Abschnitt 4.3.

Taktiken mit Schutzzielen verbunden werden. Durch die erfolgreiche Ausführung einer Taktik können die damit verknüpften Schutzziele, z. B. Zuverlässigkeit, Integrität oder Verfügbarkeit bezüglich bestimmter Elemente gebrochen werden. Die Schutzziele können dabei z. B. als Metrik für Security-Tests dienen. Das Schutzziel Verfügbarkeit ist z. B. im Falle von sicherheitskritischen Systemen aufrechtzuerhalten. Die Auswahl der Taktik steht im Mittelpunkt dieser Ebene des Angriffsmodells. Die identifizierte Taktik wird in der Ablauf-Ebene präzisiert. Auf das genaue Verfahren zur Auswahl wird in der Masterarbeit nicht eingegangen.

Beispiel 6.1.1 *Mithilfe eines Beispielszenarios soll in diesem Abschnitt das Vorgehen in der Taktik-Ebene veranschaulicht werden. Die Abbildung 6.2 bildet das folgende Beispiel bezüglich der Taktik-Ebene ab. Ein Angreifer möchte eine Schwachstelle bzw. Verwundbarkeit in einer Webapplikation entdecken. Der Fokus liegt lediglich auf der Aufdeckung von Schwachstellen bzw. Verwundbarkeiten. Sobald der Angreifer eine Schwachstelle bzw. Vulnerability entdeckt hat, ist der Angriff des Beispielszenarios erfolgreich beendet. Das bedeutet, die zielgerichtete Ausnutzung der identifizierten Schwachstellen bzw. Verwundbarkeiten steht bei diesem beispielhaften Angriff nicht im Mittelpunkt. Dieser Angriff steht in der Realität selten allein, sondern ist meistens Bestandteil eines größeren Angriffs. Beispielsweise können die Aufgaben in der „Reconnaissance"-Phase der Lookheed Martin Cyber Kill Chain mit diesem Beispiel verglichen werden.[63] Das angenommene Beispielszenario soll die entwickelte Methodik zur Angriffsmodellierung veranschaulichen und ist daher möglichst einfach gehalten. In diesem Beispiel ist das strategische Ziel des Angriffsmodells die Aufdeckung einer Schwachstelle bzw. Verwundbarkeit einer Webapplikation. Diese Information stammt aus dem Angreifermodell. Bis zu diesem Zeitpunkt ist das strategische Ziel noch nicht erreicht. Mithilfe von mindestens einer bestimmten Taktik will der Angreifer sein strategisches Ziel erreichen. Es wird angenommen, dass dem Angriffsmodell bereits eine Menge an definierten Taktiken zur Verfügung steht. Die Auswahl einer speziellen Taktik basiert auf Informationen aus dem Angreifermodell und den Voraussetzungen der Taktiken. Über ein geeignetes Verfahren fällt die Entscheidung auf das taktische Ziel „Explore", da diese Aufklärungsaktivitäten zielführend für die Aufdeckung einer Schwachstelle des Webservers (strategisches Ziel) sind. Zudem fordert diese Taktik keine Realisierung von anderen Taktiken als Voraussetzung.*

Der Fokus der Taktik-Ebene liegt auf der übersichtlichen Darstellung des abstrakten Ablaufs eines Angriffs, mit einem bestimmten strategischen Ziel. Dabei ist die Anpassungsmöglichkeit auf dem Weg zum strategischen Ziel notwendig, um auf die neuen Gegebenheiten, nach jedem taktischen Ziel, zielorientiert reagieren zu können. Gleichzeitig kann mit diesem einfachen Prozessflussdiagramm die grundsätzliche Vorgehensweise des Angreifers auf der obersten Ebene dargestellt werden, die sich im Verlauf des Angriffs dynamisch, den Umständen entsprechend ändern kann.

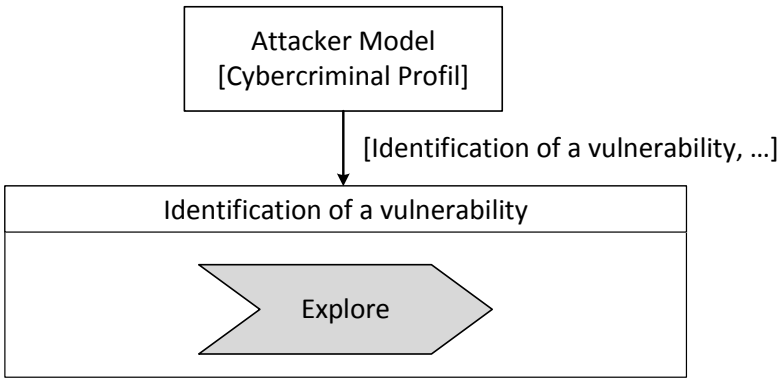

Abbildung 6.2: Der Angriff zur Aufdeckung einer Vulnerability auf Taktik-Ebene

6.2 Ablauf-Ebene

Jede identifizierte Taktik der Taktik-Ebene wird in der Ablauf-Ebene genauer spezifiziert, wie die Abbildung 5.1 zeigt. Die Ablauf-Ebene definiert einen Angriff durch die Gesamtheit an möglichen Abläufen zur Erreichung des taktischen Ziels. Dazu gehören die damit verbundenen Aktionen, deren Reihenfolge und zugehörigen Input- und Output-Objekte. Die Resultate der Ablauf-Ebene beeinflussen die Taktik-Ebene. Wenn z. B. eine Taktik in der Ablauf-Ebene abgebrochen wird und die darauffolgende Taktik die erfolgreiche Durchführung der vorherigen Taktik als Voraussetzung hat, kann diese somit (noch) nicht ausgewählt werden.

Es gibt eine Vielzahl an Sprachen zur Prozessmodellierung, wie z. B. Business Process Model and Notation (BPMN) [23] oder UML-Aktivitätsdiagramme.[17] Wegen des bemessenen Rahmens einer Masterarbeit und den Vorteilen von UML, wie in Kapitel Abschnitt 3.2 erläutert, beschränkt sich die Arbeit auf die Modellierungssprache der UML-Aktivitätsdiagramme. UML ist im Gegensatz zu anderen Sprachen in der Praxis weitverbreitet und gut etabliert. Außerdem soll das erarbeitete Angriffsmodell intuitiv von der Personengruppe verwendet werden können, die UML bereits im Einsatz hat. Laut UML entspricht eine Aktivität einer Verhaltensweise, die durch deren Gesamtheit aller möglichen Abläufe spezifiziert wird.[36] Demzufolge definiert sich die Ablauf-Ebene aus einem taktischen Ziel, den zugehörigen Aktionen, deren Reihenfolge und Input- bzw. Output-Objekte. Folglich wird ein Angriff auf dieser Ebene mithilfe von Informationen zur genauen Vorgehensweise zur Erreichung des taktischen Ziels modelliert.

Das UML-Aktivitätsdiagramm ist für die detaillierte Modellierung der Aktionen eines Angriffs allerdings nur bedingt geeignet. Die Angriffsaktionen auf Ablauf-Ebene stellen keine Aktionen im Sinne der UML dar. In UML ist eine Aktion ein einzelner Schritt der Aktivität, der nicht teilbar ist und unter Zeitaufwand durchlaufen wird.[36] Nicht jede Aktion auf Ablauf-Ebene ist unteilbar oder soll über Fließtext dargestellt werden. Dazu zählt die Angriffsaktion „Test des Targets“ (Test Target), wie Abbildung 6.3

zeigt. „Test Target“ repräsentiert den eigentlichen Angriff und ist daher im Fokus der Arbeit. An dieser Stelle („Test Target“) erfolgt eine Verknüpfung zu den vorhandenen Angriffsmöglichkeiten, in Form verschiedener Angriffstechniken. Die Details zu den Angriffstechniken werden über die Technik-Ebene abgebildet. In diesem Zusammenhang entspricht „Test Target“ einer sogenannten Tree-Action. Die Details der anderen Aktionen von „Explore“, die neben „Test Target“ in Abbildung 6.3 zu sehen sind, stehen im Rahmen dieser Masterarbeit nicht im Mittelpunkt. Es wird angenommen, dass diese Angriffsaktionen in geeigneter Weise die notwendigen Ergebnisse erzeugen.

Auf Ablauf-Ebene soll das genaue Vorgehen des Angreifers bezüglich eines taktischen Ziels modelliert werden können. Das Vorgehen wird durch verschiedene Elemente beeinflusst, wie das Angreifermodell, Umgebungs-Wissens-Modell, Sammlungen von Zugangspunkten, Angriffstechniken und Exploits. Es folgt die Beschreibung der Modellierung des taktischen Ziels „Explore“ auf Ablauf-Ebene, das in Abbildung 6.3 dargestellt wird. Dabei werden die Abhängigkeiten an gegebener Stelle erläutert. Ein Beispiel dazu wird im Anschluss der allgemeinen Beschreibung des Vorgehens präsentiert.

Die erste Angriffsaktion ist die „Identifikation von Zugangspunkten“ (engl. Identification of Access Points). Dazu sind die Informationen aller bekannten Zugangs- bzw. Angriffspunkte notwendig (Access Point Library), das Angreifermodell (Attacker Model) und das aktuelle Angreiferwissen (Environment Knowledge Model). Für den Angreifer gibt es nur eine begrenzte Auswahl an Möglichkeiten bzw. Stellen, das Target anzugreifen bzw. in dessen Umgebung vorzudringen. Somit ist eine sinnvolle Auswahl aus allen bekannten Zugangs- bzw. Angriffspunkten, die in Access Point Library enthalten sind, zu bestimmen. Die Auswahl erfolgt auf Grundlage der vorhandenen Informationen im Environment Knowledge Model und den Attributen des Angreifers aus dem Attacker Model. Zu Beginn eines Angriffs ist das Environment Knowledge Model leer, da der Angreifer noch keine Kenntnisse zu dem Target und dessen Umgebung besitzt. Je weiter fortgeschritten der Angriff ist, desto mehr Informationen sind im Environment Knowledge Model vorhanden, die die Aktion „Identification of Access Points“ beeinflussen. Z. B. spielt die URL als Zugangs- bzw. Angriffspunkt aus der Access Point Library für die Anwendungsdomäne Embedded-System keine Rolle. Gleichzeitig beeinflusst der Angreifer bzw. das Angreiferprofil die Aktion „Identification of Access Points“. Der Scope des Angreifers, wie in Abschnitt 4.3 beschrieben bestimmt z. B., ob dem Angreifer nur globale (z. B. Funkschnittstellen), regionale (z. B. Richtfunk-Schnittstellen) oder auch lokale Zugangs- bzw. Angriffspunkte (z. B. USB-Schnittstellen) zur Verfügung stehen.[52] Das Ergebnis aus der Angriffsaktion „Identification of Access Points“ ist eine Menge an möglichen Zugangs- bzw. Angriffspunkten (engl. Identified Access Points), die dem Angreifer zu diesem Zeitpunkt zur Verfügung stehen. Falls diese Angriffsaktion nicht funktioniert hat, müssen die Fehler auf geeignete Art und Weise abgefangen werden. Um die Anzahl an Wiederholungen zu begrenzen, sind entsprechende Mechanismen zu integrieren. Beispielsweise ist in Abbildung 6.3 ein Entscheidungsknoten mit einem Zähler für die Durchgänge (engl. Counter) von „Identification of Access Points“ verknüpft.

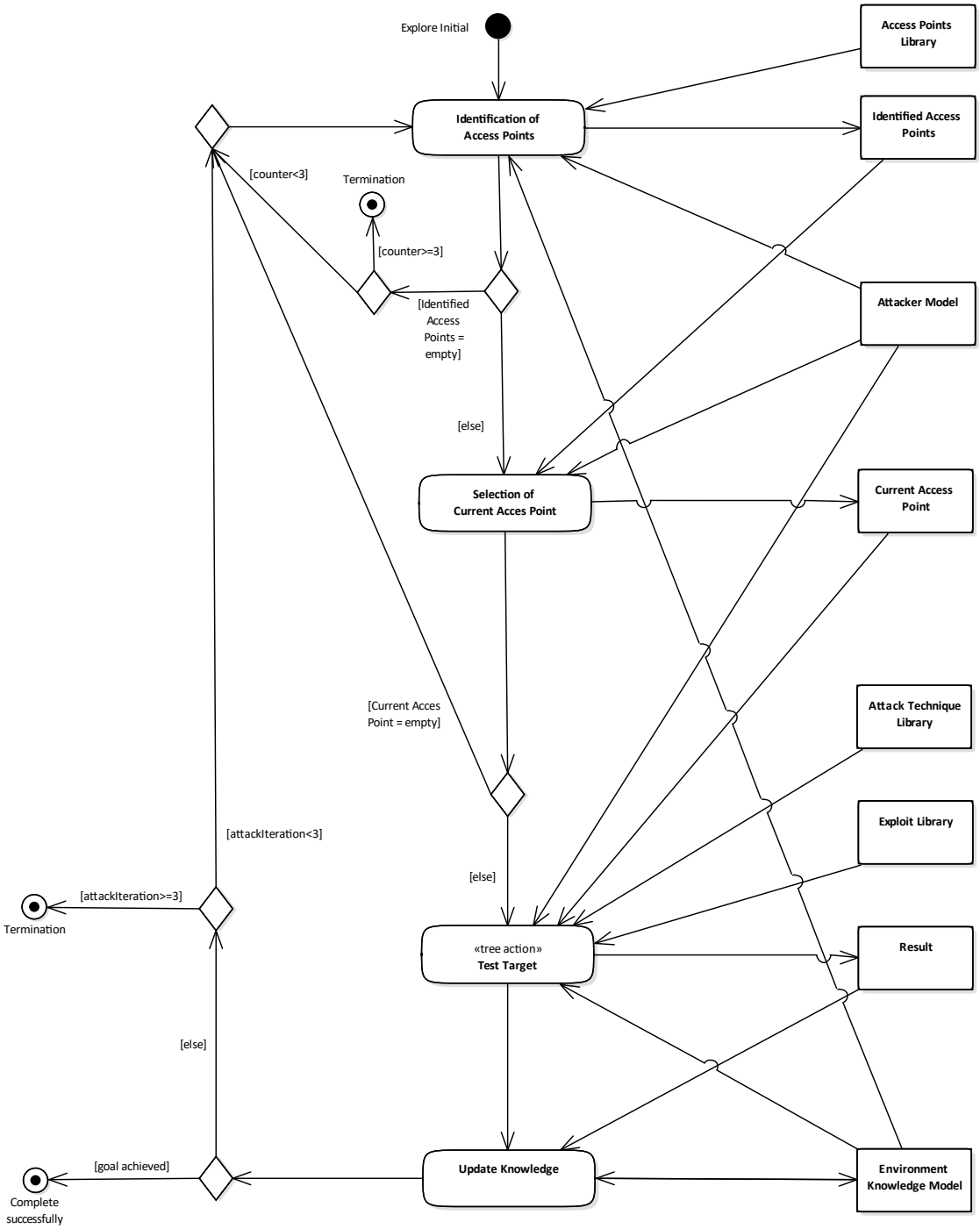

Abbildung 6.3: Modellierung der Ablauf-Ebene bezogen auf das taktische Ziel „Explore“

Falls die „Auswahl an identifizierten Zugangs- bzw. Angriffspunkten“ erfolgreich war, folgt in der nächsten Aktion die „Bestimmung eines Zugangspunkts“ (Selection of Current Access Point). Neben der zuvor identifizierten Menge an Zugangs- bzw. Angriffspunkten ist das Angreifermodell für diese Entscheidung notwendig. Das Angreiferprofil und die damit verbundenen, präferierten Angriffsbereiche bzw. bevorzugten Techniken und das strategische Ziel können die Auswahl priorisieren. Z. B. steht die Angreifergruppe bzw. -profil „APT18“ für Techniken in Zusammenhang mit „Port 80“ und dem „Command Line Interface“ als Zugangs- bzw. Angriffspunkt.[70] Des Weiteren sorgt z. B. die Kontrolle eines Webservers als strategisches Ziel für die Priorisierung von Zugangs- bzw. Angriffspunkten im Bereich eines Webservers. Danach kann die Menge an Zugangspunkten (Identified Access Points) in eine geeignete Reihenfolge gebracht werden. Unabhängig davon, ob das Angreiferprofil mit einer Menge an bevorzugten Zugangs- bzw. Angriffspunkten verknüpft ist, muss in dieser Aktion die Auswahl eines Zugangs- bzw. Angriffspunkts (Current Access Point) in geeigneter Weise erfolgen. Das ist eine Voraussetzung für die Ausführung eines Angriffs in der nächsten Aktion. Über weitere Angriffsiterationen können Angriffe auf Basis der restlichen Zugangs- und Angriffspunkte aus Identified Access Points durchgeführt werden.

An dieser Stelle sind erneut geeignete Mechanismen zu integrieren, die einen Fehlerfall auffangen, wie z. B. in Abbildung 6.3 zu sehen. Wenn ein Zugangs- bzw. Angriffspunkt (Current Access Point) erfolgreich bestimmt wurde, folgt die Angriffsaktion „Test des Targets“ (engl. Test Target) auf Basis des Current Access Point. Dafür sind Informationen aus dem Angreifermodell, der Sammlung von Angriffstechniken (Attack Technique Library), der Sammlung von Exploits (Exploit Library) und dem aktuellen Wissen des Angreifers (Environment Knowledge Model) erforderlich. Die genaue Erläuterung dieser Aktion, zusammen mit den Begründungen, folgt in Abschnitt 6.3.

Anschließend wird das Environment Knowledge Model mit dem Ergebnis (Result) aus der Angriffsaktion „Test Target“ aktualisiert. Diese Aktion wird in der Ablauf-Ebene als „Aktualisierung der Wissensgrundlage“ (engl. Update Knowledge) bezeichnet. Neben den Ergebnissen aus „Test Target“ fungiert das Environment Knowledge Model für diese Aktion als Input- und Output-Objekt. Die Aktualisierung ist für die Modellierung des dynamischen Wissensaufbaus des Angreifers notwendig, wie in Abschnitt 4.3 erläutert.

Zuletzt muss auf einer geeigneten Weise geprüft werden, inwieweit das Ziel „Explore“ zu diesem Zeitpunkt verwirklicht ist. Falls das taktische Ziel „Explore“ noch nicht erreicht ist, kann eine neue Angriffsiteration in der Ablauf-Ebene, z. B. über einen anderen Zugangs- bzw. Angriffspunkt durchgeführt werden. Dafür sind passende Mechanismen zu integrieren, um die Terminierung von Abläufen in dieser Ebene zu garantieren. Beispielsweise kann die Anzahl an Angriffsiterationen, die vom Angreiferprofil abhängig ist, für die Überprüfung, wie in Abbildung 6.3 abgebildet ist, verwendet werden. Es wird angenommen, dass ein Staat als Angreifer mehr Zeit zur Verfügung hat, als ein Skript-Kiddie.[5] Folglich ist die Maximalanzahl an Angriffsiterationen eines Staats (Angreiferprofil) für die Ablauf-Ebene höher, als die eines Skript-Kiddies (Angreiferprofil).

Beispiel 6.1.1 (Fortsetzung) *In diesem Abschnitt wird das Beispiel aus der vorher erläuterten Taktik-Ebene fortgesetzt. Nachdem „Explore“ als Taktik ausgewählt wurde, wird es in der Ablauf-Ebene spezifiziert. Das Vorgehen des Angreifers für seinen Angriff teilt sich in die zuvor erläuterten Angriffsaktionen auf, wie in Abbildung 6.3 abgebildet. Die erste Aktion „Identification of Access Points“ bekommt als Input die Access Points Library mit den Inhalten IP-Adresse, Cookie, Source-Code, Google, HTTP-Header, User Input Field, URL, USB-Schnittstelle, Bluetooth-Schnittstelle, HDMI-Schnittstelle und BUS-Schnittstelle.[7, 8, 52] Aus dieser Menge sind nun die relevanten Punkte für den Angreifer zu identifizieren. Dazu werden die Informationen zu dem Angreiferprofil eines Cyber-Kriminellen und das aktuelle Wissen des Angreifers verwendet. In diesem Fall stehen dem Angreifer lediglich die Zugangs- und Angriffspunkte eines globalen Scopes zur Verfügung. Außerdem beinhaltet das Environment Knowledge Model in diesem Beispiel bereits die Information, dass eine Website in Verbindung zu dem Webserver existiert. Die Anwendungsdomäne entspricht einer Webapplikation. Auf Basis dieser Informationen ergeben sich über ein geeignetes Verfahren folgende Identified Access Points: IP-Adresse, Cookie, Source Code, Google, HTTP-Header, User Input Field, und URL. Nach dieser erfolgreichen Identifikation muss aus dieser Menge ein einzelner Zugangs- bzw. Angriffspunkt ausgewählt werden. Über ein entsprechendes Verfahren in „Selection of Current Access Point“ wird z. B. aufgrund der Prioritäten des Cyber-Kriminellen bzw. dessen bevorzugter Angriffsbereich und dessen strategischen Ziels die URL als Zugangs- bzw. Angriffspunkt für diese Angriffsiteration bestimmt. In der darauffolgenden Aktion „Test Target“ dient die URL als Einstieg. Daneben fließen die Informationen des Angreifermodells, Attack Technique Library, Exploit Library und das Environment Knowledge Model mit ein. Die genaueren Details dazu folgen in Abschnitt 6.3. Über ein angemessenes Verfahren ergibt sich aus der Tree-Action „Test Target“ das Ergebnis (Result), dass eine SQLI-Schwachstelle in Verbindung mit dem Webserver erfolgreich entdeckt wurde. Daraufhin wird das Environment Knowledge Model gemäß diesem Result in „Update Knowledge“ aktualisiert. Eine geeignete Überprüfung, ob das Ziel „Explore“ erreicht wurde fällt positiv aus, da der Angreifer eine Schwachstelle entdeckt hat. Infolgedessen ist das strategische Ziel ebenfalls erfüllt, sodass keine weiteren Taktiken in der Taktik-Ebene notwendig sind und der Angriff bzw. dessen Modellierung beendet ist.*

Zusammengefasst fokussiert sich die Ablauf-Ebene auf die Abbildung der Gesamtheit an möglichen Angriffsiterationen bezüglich eines taktischen Ziels. Der Angriffsablauf ist damit systematisch, iterativ und zielorientiert definiert. Die Modellierung basiert auf UML-Aktivitätsdiagrammen. Dabei sind Informationen aus dem Angreifermodell, Umgebungs-Wissens-Modell und der Sammlung von Zugangspunkten notwendig. Sie stellen die Entscheidungsgrundlage für Modellierung des Vorgehens des Angreifers zur Verfügung. Dadurch kann der dynamische Wissensaufbau realisiert werden und die Wiederverwendbarkeit der Informationen wird unterstützt.

6.3 Technik-Ebene

Mithilfe der Technik-Ebene wird eine Tree-Action aus der Ablauf-Ebene spezifiziert, wie in Abbildung 5.1 im Überblick zu sehen ist. In der Technik-Ebene werden die verschiedenen Möglichkeiten eines konkreten Angriffs abgebildet. Ausgehend von einem bestimmten Zugangs- oder Angriffspunkt werden die Exploits nach Angriffstechniken klassifiziert. Die Modellierungssprache Attack Tree, die bereits in Kapitel 2 aufgeführt ist, dient als Grundlage für die detaillierte Darstellung der Tree-Action in der Technik-Ebene. Wegen der hohen Wiederverwendbarkeit von Bäumen bietet sich die Modellierungssprache an dieser Stelle sehr gut an. Dadurch muss eine Tree-Action nur einmal als Attack Tree definiert werden und kann anschließend immer wieder verwendet oder auch erweitert werden. Für die Darstellung von Angriffsmöglichkeiten sind Attack Trees in verschiedenen Varianten bereits in der Praxis im Einsatz.[35] Infolgedessen ist die Modellierung von Attack Trees intuitiv für die Zielgruppen im Security-Bereich zu verstehen.

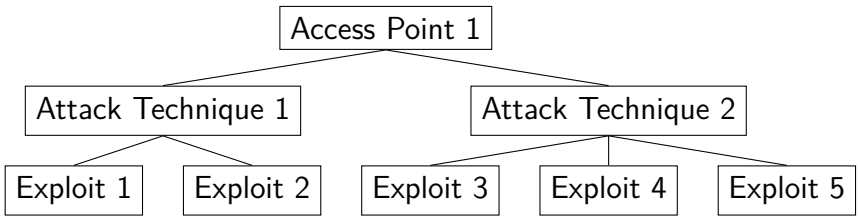

Abbildung 6.4: Modellierung der Technik-Ebene bezogen auf eine Tree-Action

Der Attack Tree nach [47] wird für die Methodik der Angriffsmodellierung in der Technik-Ebene angepasst. Abbildung 6.4 zeigt die allgemeine Baumstruktur der Tree-Action für die Methodik zur Angriffsmodellierung. Die Wurzel bildet in diesem Baum kein Angriffsziel im Sinne von Schneier, sondern einen Zugangs- bzw. Angriffspunkt. Dieser wird zuvor in der Ablauf-Ebene identifiziert (Current Access Point), wie in Abbildung 6.3 abgebildet. Die mit diesem Zugangs- bzw. Angriffspunkt verbundenen Exploits bilden die Blätter des Baumes, die nach Angriffstechniken (innere Knoten) unterteilt sind. Das bedeutet, wie in Abbildung 6.4 zu sehen, eine Angriffstechnik wird durch eine Menge an Exploits spezifiziert, die in einer gewissen Reihenfolge auszuführen sind. Mit anderen Worten ausgedrückt, fasst die Angriffstechnik die relevanten Aspekte der darunterliegenden Exploits bezüglich der Durchführung eines Angriffs zusammen. Somit können Angriffe klassifiziert und wiederverwendet werden. Ein Exploit bezieht sich dabei immer, neben dem internen Knoten, auf die Wurzel, die einen bestimmten Zugangs- bzw. Angriffspunkt abbildet. Der Weg, ausgehend von der Wurzel bis hin zum Blatt, kann als Angriffsvektor interpretiert werden. Gleichzeitig werden im Attack Tree implizit bekannte Vulnerabilities dargestellt: Ausgehend von einem bestimmten Angriffspunkt über die verwendete Technik bis hin zum konkreten Exploit, können diese Informationen und deren Verbindungen zueinander bekannte Verwundbarkeiten repräsentieren, wie in Abschnitt 4.2 erläutert. Die Informationen für den Aufbau der Technik-Ebene

(Baumstruktur) bzw. die Informationen zu den Zusammenhängen zwischen Exploits, Angriffstechniken und Zugangs- bzw. Angriffspunkt werden dem Angriffsmodell über die Exploit Library, Attack Technique Library und Access Point Library zur Verfügung gestellt. Die Libraries müssen in einer geeigneten Weise aufgebaut sein, sodass die entwickelte Methodik zur Angriffsmodellierung unterstützt wird. Dadurch lassen sich die einzelnen Elemente im Baum in geeigneter Weise miteinander kombinieren, dass die Wiederverwendbarkeit der Methodik zur Angriffsmodellierung unterstützt. Zudem sollen mehrere Exploits (Blätter) gruppiert werden können, die relevante Gemeinsamkeiten aufweisen, wie im nachfolgenden Beispiel veranschaulicht. Dieser Aspekt fördert ebenfalls den Aspekt der Wiederverwendbarkeit und Erweiterbarkeit.

Auf dieser grundlegenden Baumstruktur, wie in Abbildung 6.4 abgebildet, soll ein bestimmter Exploit (Blatt) für den konkreten Angriff (Angriffssimulation) ausgewählt werden. In diesem Zusammenhang stellt ein Exploit (Blatt) der Technik-Ebene nicht nur Befehlsfolgen dar, die *direkt* als unerlaubte Handlung zu interpretieren sind, wie in 3.4 definiert. Vielmehr präsentieren die Exploits als Blätter in der Technik-Ebene sämtliche Handlungsmittel, die *in Verbindung* bzw. *im Kontext* eines Angriffs stehen. Dadurch sollen nicht nur die verschiedenen Möglichkeiten im Umgang mit Angriffspunkten modelliert werden können, sondern auch die diversen Variationen in der Nutzung von Zugangspunkten. Die Verwendung eines Zugangspunktes ist nicht zwangsläufig mit unerlaubten Handlungen verbunden, beispielsweise wenn die Stelle öffentlich für jeden Benutzer einsehbar ist. Z. B. präsentiert der Quellcode einer Website einen Zugangspunkt. Ein Zugriff auf den bereitgestellten Source Code ist an sich kein unautorisierter Zugriff oder eine unerlaubte Handlung. Der Quellcode einer Website kann frei mithilfe eines Browsers betrachtet und analysiert werden. Im Kontext eines Angriffs kann diese Quelle dennoch nützliche Informationen für die Planung und Entwicklung eines Angriffs erbringen. Infolgedessen werden in der Technik-Ebene sämtliche Möglichkeiten an Handlungsmitteln (Exploits) betrachtet, die im Kontext eines Angriffs stehen. In dieser Ebene sollen auch über eine geeignete Art und Weise die relevanten Informationen zu den verwendbaren Werkzeugen, bezüglich der Ausnutzung von Zugangs- bzw. Angriffspunkten, mithilfe spezifischer Techniken bzw. Exploits modelliert werden.

Die Auswahl eines konkreten Exploits für die Angriffssimulation findet über die Traversierung der Bäume statt. Über die Wurzel, die einen Zugangs- oder Angriffspunkt darstellt, erfolgt der Einstieg in den Baum, wie in Abbildung 6.4 abgebildet ist. Zunächst muss die Angriffstechnik bestimmt werden. Das entspricht dem Weg über die inneren Knoten des Baums. Die Auswahl der Technik ist zum einen von dem Angreiferprofil (Angreifermodell) und zum anderen von dem aktuellen Wissen des Angreifers (Umgebungs-Wissens-Modell) abhängig, wie bereits in Abschnitt 4.3 erwähnt. Das Profil eines Angreifers stellt z. B. den bevorzugten Angriffsbereich des Angreifers, dessen verfügbare Ressourcen (Zeit, Geld, Performance) und seine Fachkenntnisse bereit, wie in Abschnitt 4.3 und Abschnitt 5.1 beschrieben. Dabei steht beispielsweise dem Staat ein breiteres und tiefgründigeres Know-How zur Verfügung, sodass dieser Angreifer bzw. dieses Angreiferprofil beispielsweise kryptografische Angriffe durchführen kann, die mit der Expertise von komplexen

Sachverhalten verbunden sind.[5] Es wird davon ausgegangen, dass ein Skript-Kiddie diese Angriffe nicht ausführen kann. Folglich ist es für dieses Angreiferprofil nicht möglich, die sehr komplexen Angriffstechniken im Angriffsmodell auszuwählen.[5] Neben den Informationen aus dem Angreifermodell ist das Angreiferwissen für die Auswahl der Technik entscheidungsrelevant, wie in Abschnitt 4.3 erläutert. Das Umgebungs-Wissens-Modell präsentiert die aktuelle Ausgangslage des Angreifers bezüglich des Targets und dessen Umgebung. Auf dieser Grundlage plant der Angreifer seinen Angriff. Die Durchführung von Angriffstechniken steht in Verbindung mit bestimmten Voraussetzungen bezüglich des Targets und dessen Umgebung, wie beispielsweise das Vorhandensein bestimmter Mechanismen oder einer Software- oder Hardwarekomponente. Für das Verfahren zur Auswahl einer Technik müssen diese Voraussetzungen mithilfe des Angreiferwissens vorliegen. Außerdem ist die Anwendungsdomäne, gemäß des aktuellen Angreiferwissens maßgebend für die Identifikation einer relevanten Technik, wie in Abschnitt 4.3 beschrieben. Das genaue Verfahren steht in dieser Masterarbeit nicht im Mittelpunkt.

Nachdem die Angriffstechnik (innerer Knoten) bestimmt ist, muss ein damit verbundener Exploit (Blatt) für die Angriffssimulation gewählt werden. Wie bereits in Abschnitt 4.3 beschrieben, wird die Auswahl durch die vorherige Ausführung anderer Exploits, das aktuelle Angreiferwissen und über das (aktuelle) taktische Ziel des Angreifers bestimmt. Dazu ist jeder Exploit mit entsprechenden Vor- und Nachbedingungen verknüpft, wie in Abschnitt 4.3 erläutert.

Beispiel 6.1.1 (Fortsetzung) *Die Veranschaulichung des Vorgehens in der Technik-Ebene basiert auf dem Beispiel aus der zuvor beschriebenen Taktik- und Ablauf-Ebene. Wegen dem bemessenen Rahmen der Masterarbeit wird im Nachfolgenden lediglich auf die Tree-Action „Test Target“ eingegangen. Diese stellt den eigentlichen Angriff dar und steht somit im Fokus für die zu entwickelnde Methodik einer generischen Angriffsmodellierung. Die Ausgangssituation ist die Ausführung der Angriffsaktion „Test Target“, wie in Abschnitt 6.2 im Beispiel beschrieben. Dafür wurde die „URL“ der Webapplikation als Zugangspunkt (Current Access Point) definiert. Abbildung 6.5 bildet das folgende Beispiel bezüglich der Technik-Ebene ab. Die Abbildung ist nicht vollständig, sondern zeigt nur einen Ausschnitt an möglichen Angriffstechniken und Exploits. Die „URL“ bildet die Wurzel. Aus der Attack Technique Library stammten die verfügbaren Angriffstechniken „URL Encoding“ und „Injection“. Dabei ist „Injection“ eingeteilt in „SQLI“ und „Code Injection“. „SQLI“ lässt sich wiederum in „Normal SQLI“ und „Blind SQLI“ aufteilen. Auf Basis des Cyber-Kriminellen (Angreiferprofil), die damit verknüpften Fachkenntnisse, verfügbaren Ressourcen, seinem bevorzugten Angriffsbereich und dem aktuellen Wissen des Angreifers fällt die Entscheidung über ein geeignetes Verfahren auf „Normal SQLI“ (Angriffstechnik). „Normal SQLI“ ist mit entsprechenden Exploits verbunden, die über die Exploit Library bereitgestellt werden. Nach einem zweckmäßigen Verfahren wird ein Exploit (Blatt) ausgewählt. Die Auswahl basiert auf den Voraussetzungen der verfügbaren Exploits („MySQL Vulnerable Exploit“, „Oracle Vulnerable Exploit“,*

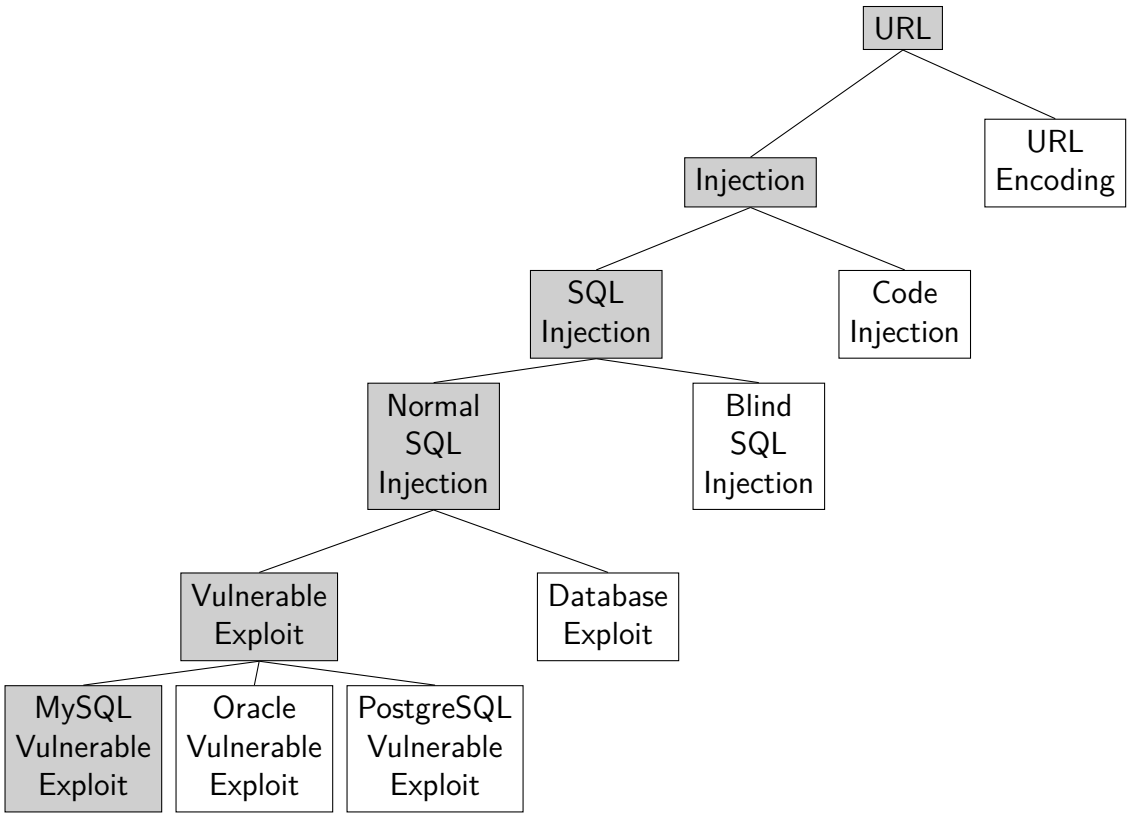

Abbildung 6.5: Die Aufdeckung von Schwachstellen auf Technik-Ebene [32, 10]

„PostgreSQL Vulnerable Exploit“) bzw. auf den Voraussetzungen von Gruppierungen („Vulnerable Exploit“) davon. Daneben ist die Auswahl von dem aktuellen Angreiferwissen und dem taktischen Ziel abhängig. Alle diese Informationen werden der Technik-Ebene bzw. der Tree-Action „Test Target“ bereitgestellt, wie in Abbildung 6.3, aus Sicht der Ablauf-Ebene zu sehen ist. In dieser beispielhaften Abbildung 6.5 ist lediglich ein Ausschnitt an Exploits („Vulnerable Exploit“, „Database Exploit“) abgebildet. „Vulnerable Exploit“ steht für die Exploits zur Feststellung der Verwundbarkeit für „Normal SQLI“. „Database Exploit“ bezeichnet Exploits zur Identifikation von Datenbanken im Rahmen von SQLI als Angriffstechnik. Die Exploits unter „Vulnerable Exploit“ besitzen keine Voraussetzungen bezüglich einer vorherigen Ausführung anderer Exploits. Infolgedessen enthält das aktuelle Umgebungs-Wissens-Modell alle notwendigen Informationen, um den Exploit durchzuführen. Außerdem ist diese Gruppe an Exploits („Vulnerable Exploits“) zielführend für das taktische Ziel „Explore“. „Database Exploit“ ist für andere taktische Ziele relevant, z. B. für Taktiken, die die tatsächliche Ausnutzung fokussieren. Daher fällt die Auswahl auf (die Gruppe) „Vulnerable Exploit“. In diesem Beispiel gibt es eine weitere Unterteilung des Exploits, da nach verschiedenen Datenbankmanagementsystemen unterschieden werden muss. Beispielsweise unterscheiden sich die Exploits von „Vulnerable Exploit“ bezüglich der Syntax.[32] Auf Basis des aktuellen Angreiferwissens, das die Information einer „MySQL“-Datenbank als Backend in der Umgebung des Targets

aufzeigt, wird der Exploit „MySQL Vulnerable Exploit“ ausgewählt. Der Weg, ausgehend von der „URL“ über die Angriffstechniken „Injection“, „SQLI“, „Normal SQLI“ über die Exploit-Gruppierung „Vulnerable Exploit“ bis hin zum Exploit „MySQL Vulnerable Exploit“, kann für diese eine Angriffsiteration als Angriffsvektor mit zusätzlichen Details zu den verwendeten Exploits interpretiert werden. Der Angriffsvektor wird in Abbildung 6.5 anhand der grau hinterlegten Knoten abgebildet.

Folglich wird ein Angriff mithilfe der Technik-Ebene, ausgehend von einem bestimmten Zugangs- bzw. Angriffspunkt, über eine spezielle Angriffstechnik, in Form eines bestimmten Exploits, abgebildet. Die grafische Darstellung in Form einer Baumstruktur ist intuitiv für die Zielgruppe dieser Methodik verständlich. Der Angriffsvektor ist auf einem Blick abbildbar. Des Weiteren unterstützt der individuelle Aufbau eines Baumes die Wiederverwendbarkeit der Methodik zur Angriffsmodellierung. Für die Bestimmung des Exploits sind die Informationen aus dem Angreifermodell, dem Umgebungs-Wissens-Modell und aus den Sammlungen von Angriffstechniken und Exploits notwendig. Nach der Identifikation des Exploits, gemäß einem geeigneten Verfahren, wird dieser anschließend auf einem geeigneten Modell simuliert. Das genaue Vorgehen zur Identifikation einer Angriffstechnik und Exploits ist nicht Bestandteil dieser Masterarbeit.

6.4 Angriffssimulation

Die Angriffssimulation stellt einen wichtigen Beitrag für die Angriffsmodellierung dar. Nach der strukturierten Auswahl eines Exploits, angefangen von der Taktik-Ebene, über die Ablauf-Ebene, bis hin zur Technik-Ebene soll die Ausführung des Exploits auf einem geeigneten Umgebungsmodell simuliert werden. In Abbildung 5.1 wird dieses Vorgehen im Kontext des Umgebungsmodells (Environment Model) abgebildet. Die Simulation des Exploits bildet die Ausführung eines Angriffs in der Praxis ab. Der Angreifer gewinnt dadurch neue Erkenntnisse, die sein Wissen über das Target und dessen Umgebung erweitern, wie bereits in Abschnitt 4.3 erläutert. Dieses Wissen bildet die Grundlage für das weitere Vorgehen des Angreifers. Die Angriffssimulation ist notwendig, um den dynamischen Wissensaufbau des Angreifers für das Angriffsmodell zu modellieren.

Der identifizierte Exploit soll für die Angriffssimulation in eine geeignete Form abgeleitet werden, wie die Abbildung 5.1 veranschaulicht. Die Idee besteht darin, einen brauchbaren Exploit-Code aus dem identifizierten Exploit der Technik-Ebene abzuleiten, wodurch sowohl die Angriffssimulation, als auch Security-Tests unterstützt werden können, wie bereits in Abschnitt 4.3 erwähnt.

Das genaue Vorgehen der Angriffssimulation ist nicht Teil dieser Arbeit. Anhand der Angriffssimulation sollen sich neue Erkenntnisse in Form verschiedener Informationen ergeben, z. B. der Erfolg des simulierten Angriffs (erfolgreich, abgebrochen, neue Informationen/keine neuen Informationen gewonnen). Im Idealfall gewinnt der Angreifer neue Details bezüglich des Targets und dessen Umgebung, neue Zugangs- oder Angriffspunkte. Dafür ist ein geeignetes Umgebungsmodell notwendig, wie bereits in Abschnitt 4.3 beschrieben. Das Umgebungsmodell soll demnach eine vollständige Umgebung des Targets für die Angriffssimulation bereitstellen. Da der Angreifer seinen Angriff gemäß seinen Annahmen aus dem Umgebungs-Wissens-Modell plant und entwickelt, können Unterschiede zwischen der Wissensgrundlage des Angreifers und der Simulationsumgebung bestehen. Die Simulation eines identifizierten Exploits soll die zugehörigen Ergebnisse auf der Simulationsumgebung liefern. Z. B. kann der Angreifer durch vorhandene Sicherheitsmechanismen im Umgebungsmodell getäuscht oder der Angriff kann verhindert werden. Unabhängig davon baut der Angreifer sein Wissen aus diesen Ergebnissen der Angriffssimulation auf. Das bedeutet, die neuen Informationen aus der Angriffssimulation werden über die Technik-Ebene, an geeigneter Stelle in das Angriffsmodell integriert, sodass die nachfolgende Angriffsiteration darauf aufbauen kann.

Zudem soll nach der Angriffssimulation die Angriffstiefe ableitbar sein. Das Umgebungsmodell für die Simulation soll in geeigneter Weise abbilden, wie weit der Angreifer bis zu einem bestimmten Zeitpunkt in die Umgebung eingedrungen ist, wie in Abbildung 5.1 bezüglich der verschiedenen Ebenen der Angriffsoberfläche (Attack Surface) zu sehen. Mit jeder Angriffsiteration versucht der Angreifer tiefer bzw. weiter in das Target bzw. in dessen Umgebung vorzudringen. Die grafische Präsentation der Angriffstiefe im Umgebungsmodell fördert das allgemeine Verständnis zum Angriff, da auf einen Blick die existierende Umgebung, im Hinblick angreifbarer Elemente hervorgehoben wird. Gleichzeitig kann die Angriffstiefe eine geeignete Metrik für Security-Tests darstellen. Beispielsweise wird festgelegt, dass die Durchführung von Security-Tests bis zu den Elementen der Angriffstiefe drei für eine nicht-kritische Webapplikation ausreichend ist.

Beispiel 6.1.1 (Fortsetzung) *An das entwickelte Beispielangriffsszenario, das bereits das Vorgehen der verschiedenen Ebenen im Angriffsmodell veranschaulicht hat, wird hier angeknüpft. Wie bereits in diesem Abschnitt erwähnt, steht das genaue Vorgehen einer Angriffssimulation nicht im Mittelpunkt dieser Arbeit. Fokus sind die Ergebnisse der Angriffssimulation. Nachdem der Exploit „MySQL Vunerable Exploit“ im Kontext einer normalen SQLI bezüglich der „URL“ als Angriffspunkt in der Technik-Ebene identifiziert wurde, wird dieser Exploit auf einem geeigneten Umgebungsmodell simuliert. Das Umgebungsmodell in Abbildung 6.6 ist nicht vollständig und präsentiert lediglich eine Idee, wie die Ergebnisse einer Angriffssimulation für dieses Beispielszenario präsentiert werden können. Unabhängig davon sind die Ergebnisse einer Simulation für die Methodik zur Angriffsmodellierung notwendig. Es wird der Zugangs- bzw. Angriffspunkt („URL“)*

bezüglich des Targets und dessen Umgebung abgebildet. Davon ausgehend ist die damit gewonnene Angriffsoberfläche sichtbar. Die durch den ausgeführten „MySQL Vulnerable Exploit“ existierende SQLI-Verwundbarkeit wird indirekt über die blaue Fläche dargestellt. Somit wird gleichzeitig die aktuelle Angriffstiefe (Level 0) präsentiert. Daneben sind vorhandene Sicherheitsmechanismen und die Assets der Umgebung abgebildet, wie Kundendaten und Systemkonfiguration. Die neuen Informationen, die der Angreifer durch die Ausführung des Angriffs (Simulation des Exploits) gewonnen hat, werden durch die Angriffstiefe begrenzt. Diese sollen anschließend über die Technik-Ebene in das Angriffsmodell an geeigneter Stelle integriert werden. Die Informationen bilden z. B. den Input (Result) für die Aktion „Update Knowledge“ in der Ablauf-Ebene, wie in Abbildung 6.3 abgebildet. Auf dem aktualisierten Angreiferwissen plant der Angreifer seine nächste Angriffsiteration.

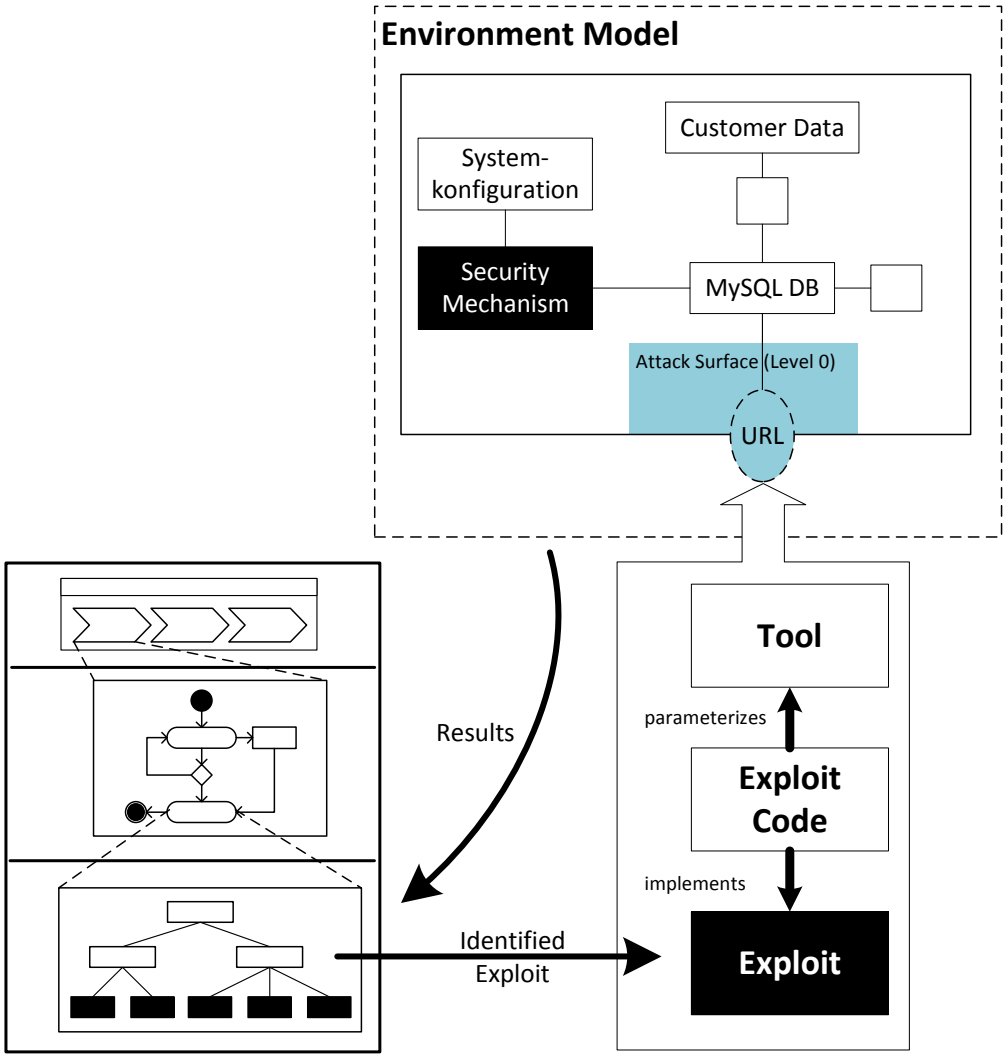

Abbildung 6.6: Die Aufdeckung von Schwachstellen mithilfe der Angriffssimulation

Zusammenfassend wird mithilfe der Angriffssimulation der dynamische Wissensaufbau für die Methodik der Angriffsmodellierung umgesetzt. Dieses Angreiferwissen bildet für jede Angriffsiteration eine Entscheidungsgrundlage für die Modellierung des Vorgehens des Angreifers. Die gewonnen Ergebnisse der Simulation müssen über die Technik-Ebene an den geeigneten Stellen integriert werden. Mithilfe der Angriffstiefe werden die gewonnenen Informationen aus der Simulation des Exploits präsentiert. Demnach ist ein geeignetes Umgebungsmodell für die Angriffssimulation zu gewährleisten.

Anhand des beispielhaften Angriffsszenarios in diesem Kapitel wurde das Vorgehen der entwickelten Methodik zur Angriffsmodellierung veranschaulicht. Die verschiedenen Ebenen präsentieren dabei die verschiedenen Sichtweisen auf einen Angriff. Anhand einer strukturierten Vorgehensweise wird mit jeder Angriffsiteration versucht, einen Exploit (für die Angriffssimulation) zu bestimmen. Die Ergebnisse der Simulation bilden die neue Ausgangslage des Angreifers für die darauffolgende Angriffsiteration.

7 Evaluierung

Mithilfe dieser Masterarbeit soll eine Methodik zur Angriffsmodellierung für die Unterstützung von Security-Tests entwickelt werden. In diesem Kapitel wird die entwickelte Methodik bewertet. Die Evaluierung dieser Methodik ist mit einigen Schwierigkeiten verbunden, die zunächst kurz aufgezeigt werden. Im Kontext der identifizierten Evaluierungskriterien wird die Methodik an zwei Angriffsszenarien angewandt. Wegen dem bemessenen Rahmen der Masterarbeit fokussiert sich die Anwendung der Methodik auf eine Angriffsiteration. Des Weiteren beziehen sich die Angriffsszenarien jeweils auf die Aktivitäten der ersten Phase des Angriffs (taktisches Ziel „Explore“[1]), worauf die gesamte Arbeit konzentriert ist.

An die zu entwickelnde Methodik zur Angriffsmodellierung sind folgende Anforderungen geknüpft, wie in Abschnitt 4.1 analysiert: „Modellbasiert“, „Wiederverwendbar“, „Ausdrucksstark“, „Systematisch“, „Simulierbar/Konsistent“, „Visualisierbar“ und „Verständlich“. Im Rahmen der Evaluierung sollen diese Anforderungen an die Methodik überprüft werden.

7.1 Schwierigkeiten der Evaluierung

Es stellt sich die Frage, wie der Nutzen der entwickelten Methodik bewertet werden kann. In diesem Zusammenhang liegt die Herausforderung in der Identifikation von aussagekräftigen Metriken. Der Nutzen liegt z. B. in der generischen Eigenschaft der entwickelten Methodik. Ein generisches Modell ist sinnvoll, da es unabhängig von dem Anwendungsgebiet eingesetzt werden kann. Infolgedessen kann der Einsatz einer generischen Lösung z. B. die Anwendung mehrerer Insellösungen ersetzen, sodass weniger Aufwand in die Erlernung von verschiedenen Methoden eingesetzt werden muss. Allerdings ist eine generische Methodik schwierig zu evaluieren. Generisch bedeutet im Zusammenhang mit der Methodik zur Angriffsmodellierung, mithilfe eines allgemein gültigen Modells möglichst alle bekannten Angriffe zu spezifizieren.

[1] Andere Taktiken, die im Verlauf eines Angriffs neben „Explore“ ausgewählt werden können, wie „Exploit“, stehen im Rahmen der Masterarbeit nicht im Fokus und werden daher nicht näher erläutert.

Wegen dem begrenzten Rahmen einer Masterarbeit wird die Methodik an zwei beispielhaften Angriffsszenarien angewandt. Das eine Beispielszenario basiert auf Schwachstellen der OWASP Top Ten 2017, die eine hohe Relevanz im Bereich Webanwendungen haben, wie in Abschnitt 7.3 begründet. Das andere Beispielszenario in Abschnitt 7.3 steht im Zusammenhang mit dem aktuell laufenden Projekt „MASSiF“, das bereits in Abschnitt 5.1 erwähnt wird. Weitere notwendige Informationen dazu werden an entsprechender Stelle, im weiteren Verlauf ergänzt. Die Beispielszenarien basieren folglich auf relevanten Grundlagen, was die Auswahl der Beispiele für die Evaluierung begründet.

Eine weitere Herausforderung stellt der Vergleich mit anderen Modellen dar, da noch kein generisches Angriffsmodell in dieser Form existiert. Je nach Anwendungsbereich gibt es einzelne, unabhängige Lösungen für die Handhabung von Angriffen, wie bereits in Kapitel 2 erläutert. Diese stellen gemäß dem Anwendungsbereich spezifische Ergebnisse im Hinblick auf die Gewährleistung von IT-Sicherheit zur Verfügung. Die Ergebnisse der verschiedenen Insellösungen dienen z. B. als Grundlage für die Ableitung von Verteidigungsmaßnahmen. Dahingehend empfiehlt beispielsweise das BSI, als angesehene Cyber-Sicherheitsbehörde, einen Sicherheitsprozess.[51] Der BSI-Sicherheitsprozess definiert in diesem systematischen Vorgehen verschiedene Aspekte als Voraussetzung für die Identifikation von Verteidigungsmaßnahmen.[51] Als Leitfaden für viele deutsche Unternehmen spielt der wirtschaftliche Aspekte im BSI-Sicherheitskonzept eine wesentliche Rolle. Daher werden im BSI-Sicherheitsprozess nicht alle möglichen Angriffe identifiziert. Mithilfe eines Basis-Sicherheitschecks (Soll-Ist-Vergleich[2]) bzw. einer Risikoanalyse wird die Menge auf einen handhabbaren Ausschnitt an relevanten Bedrohungen reduziert.[51] Das Ziel der entwickelten Methodik ist dagegen möglichst alle bekannten Angriffe zu modellieren, bzw. diese modellierten wiederzuverwenden, sodass die Ergebnisse daraus als Grundlage für weitere Aktivitäten fungieren können, wie z. B. für Risikoanalysen. Sowohl die entwickelte Methodik, als auch der BSI-Sicherheitsprozess sollen die Durchgängigkeit von IT-Security unterstützen.[53] Dennoch ist ein Vergleich der entwickelten Methodik mit dem BSI-Sicherheitsprozess bzw. mit Teilen davon nicht einfach möglich, da der Sicherheitsprozess ein weitaus mächtigeres Instrument ist, als die entwickelte Methodik zur Angriffsmodellierung. Zudem ist der Prozess des BSI nicht auf die Identifikation aller bekannter Angriffe ausgelegt. Der BSI-Sicherheitsprozess dient zur Einhaltung eines bestimmten Sicherheitsniveaus im Unternehmen.[51] Dazu zählt auch die Durchführung einer notwendigen Risikoanalyse.[51] Dagegen enthält die Methodik zur Angriffsmodellierung keine Aktivitäten der Risikoanalyse. Sie dient vielmehr als Grundlage und Unterstützung für darauffolgende Security-Analysen und Security-Tests. Ein Vergleich der entwickelten, generischen Methodik mit anderen Ansätzen, die das gleiche Ziel verfolgen ist für eine sinnvolle Beurteilung der Methodik nicht zwangsläufig nützlich. Das wird in diesem Abschnitt sichtbar.

[2]Für weitere Informationen siehe [51].

7.2 Evaluierungskriterien

Im Hinblick auf die zuvor erläuterten Probleme in Abschnitt 7.1 soll die Methodik zur Angriffsmodellierung bezüglich ihrer Anforderungen aus Abschnitt 4.1 evaluiert werden. Dazu sind zunächst geeignete Evaluierungskriterien zu identifizieren.

Es ist anzunehmen, dass eine modellbasierte Lösung einen gleichwertigen Mehrwert wie die Einführung von UML-basierten Lösungen für Softwarearchitekturen erbringt. Z. B. stellt die schlüssige und qualitativ hochwertige Abbildung von Angriffen ein Potential im Hinblick auf den Nutzen für die modellbasierte Methodik zur Angriffsmodellierung dar. Die Anforderung soll über das gleichnamige Kriterium *Modellbasiert* bestimmt werden. Dieses Evaluierungskriterium bezieht sich auf das Ausmaß, in dem der Methodik zur Angriffsmodellierung ein Modell zugrunde liegt. Im Zusammenhang mit den beispielhaften Angriffsszenarien und durch Referenz zu den entsprechenden Stellen in der Masterarbeit soll das Ausmaß aufgezeigt werden.

Mithilfe der nächsten beiden Anforderungen „Wiederverwendbar“ und „Ausdrucksstark“ wird indirekt die Allgemeingültigkeit bzw. der generische Aspekt für die Methodik zur Angriffsmodellierung gefordert. Anhand des Evaluierungskriteriums *Wiederverwendbare Elemente* soll die Anforderung „Wiederverwendbar“ an die Methodik demonstriert werden. Das Kriterium *Wiederverwendbare Elemente* bezieht sich auf das Ausmaß, die Inhalte und Strukturen des Angriffsmodells in Verbindung mit anderen Angriffsszenarien einfach wiederzuverwenden. Das Kriterium soll im Kontext der beispielhaften Angriffsszenarien demonstriert werden. Mithilfe der Methodik zur Angriffsmodellierung sollen möglichst viele Angriffe abgebildet werden können, wie bereits in Abschnitt 4.1 durch die Anforderung „Ausdrucksstark“ bestimmt. Die Ausdrucksstärke bezieht sich auf den Abstraktionsgrad für die Abbildung von Angriffen, bzw. inwieweit alle bekannten Angriffe mithilfe der Methodik modellierbar sind. Es stellt sich die Frage, was nicht mithilfe der entwickelten Methodik zur Angriffsmodellierung abgebildet werden kann und wie dieses Ausmaß messbar ist. Für die Bestimmung der Ausdrucksstärke sollen die Kriterien *Relevante Angriffe* und *Unabhängigkeit zur Anwendungsdomäne* fungieren. *Relevante Angriffe* bezieht sich auf das Ausmaß, inwieweit relevante Angriffe, die z. B. im Zusammenhang mit den Top Ten 2017 von OWASP [78] stehen, mithilfe der Methodik modelliert werden können. Das zweite Evaluierungskriterium für die Ausdrucksstärke *Unabhängigkeit der Anwendungsdomäne* bezieht sich auf die Möglichkeit verschiedene Angriffe, unabhängig von der (Anwendungs-)Domäne des Angriffs abbilden zu können. Dabei stellt sich die Frage, wie die typischen Aspekte und Eigenschaften einer Domäne im Rahmen der Methodik modelliert werden. Die Kriterien sollen im Zusammenhang mit den beiden Angriffsszenarien und dem generellen Aufbau und Vorgehen der Methodik verdeutlicht werden.

„Systematisch“ stellt eine weitere Anforderung an die Methodik zur Angriffsmodellierung dar. Die Systematik ist eine essentielle Grundlage für die Durchgängigkeit und Nachverfolgbarkeit von IT-Security, wie in Abschnitt 4.1 erläutert. Mithilfe des Kriteriums *Systematische Strukturen* soll diese Anforderung aufgezeigt werden. Das Kriterium bezieht sich auf das Ausmaß, inwieweit eine Systematik in der entwickelten Methodik vorhanden ist, sodass ein Angriff nachvollziehbar und wiederholbar modelliert werden kann. Da in diesem Kapitel lediglich die Ergebnisse der beispielhaften Angriffsszenarien aufgezeigt sind[3], wird das Kriterium in Beziehung zu dem allgemeinen Aufbau und der Vorgehensweise der Methodik gesetzt, die Informationen zur Systematik aufweisen.

Die entwickelte Methodik zur Angriffsmodellierung soll konsistent sein. Das ist Voraussetzung für eine Angriffssimulation, für den Einsatz von Automatismen und somit eine geeignete Grundlage für Security-Tests und -Analysen, wie bereits in Abschnitt 4.1 beschrieben. Hierfür ist maßgebend, dass z. B. das Umgebungsmodell (als Grundlage für die Simulation) und die notwendigen Abhängigkeiten im Rahmen der Methodik bereits existieren bzw. definiert sind, wie Angreifermodell, Umgebungs-Wissens-Modell, Sammlung von Angriffstechniken und Exploits. In diesem Zusammenhang soll eine verständliche Syntax und Semantik, die für Maschinen lesbar ist, definiert und umgesetzt werden, sodass Automatismen die Methodik zur Angriffsmodellierung im Hinblick auf Security-Tests unterstützen. Zudem soll die Parametrisierung in einem simulierbaren bzw. konsistenten Angriffsmodell möglich sein. Diese Voraussetzungen sind zu diesem Zeitpunkt nicht gegeben. Die Masterarbeit fokussiert sich auf die Entwicklung einer geeigneten Methodik zur Angriffsmodellierung. Daher wird die Konsistenz der Methodik zur Angriffsmodellierung in diesem Rahmen nicht bewiesen.

Im Hinblick auf eine modellbasierte Lösung soll die Methodik zudem „Visualisierbar“ sein, wie in Abschnitt 4.1 aufgezeigt. Die komplexen Zusammenhänge eines Angriffs sollen über eine visuelle Darstellung die Handhabung von Angriffen und das Verständnis darüber unterstützen. Das Evaluierungskriterium *Visuelle Elemente* soll die Anforderung an die Methodik demonstrieren. Das Kriterium bezieht sich auf das Ausmaß, in dem die Methodik zur Angriffsmodellierung grafische Elemente aufweist bzw. grafisch abbildbar ist. Das Kriterium *Visuelle Elemente* soll im Kontext der beiden Angriffsszenarien dargelegt und bezüglich des allgemeinen Aufbaus und der Vorgehensweise der Methodik betrachtet werden.

Die Verständlichkeit der Methodik ist relevant für die Bewertung des Nutzens der entwickelten Methodik zur Angriffsmodellierung. Im Rahmen dieser Methodik bedeutet die entsprechende Anforderung „Verständlich“ beispielsweise einen einfachen und intuitiven Umgang von komplexen Angriffen in Verbindung mit Angreifern und Systemen. Die Methodik soll dahingehend einfach zu erlernen sein und unkompliziert in der Anwendung, wie in Abschnitt 4.1 gefordert. Beispielsweise kann die Verständlichkeit der Methodik

[3] Die Erläuterungen zum Vorgehen sind analog zu der beschriebenen Vorgehensweise in Kapitel 6 zu finden.

mithilfe von Interviews oder Fragebögen von Personen der Zielgruppe, die die Methodik im Rahmen einer Umfrage nutzten, evaluiert werden.[83] Allerdings müssen dafür bestimmte Voraussetzungen erfüllt sein, wie z. B. die Existenz der abhängigen Elemente der Methodik und weitere notwendige Details für die tatsächliche Implementierung und Umsetzung des Angriffsmodells im Rahmen der Methodik. In diesem Zusammenhang wird die Verständlichkeit in dieser Masterarbeit nicht evaluiert.

7.3 Ergebnisse der Evaluierung

Im Nachfolgenden werden die relevanten Ergebnisse durch die Anwendung der entwickelten Methodik zur Angriffsmodellierung an zwei beispielhaften Szenarien präsentiert. Auf dieser Basis sollen die Anforderungen „Modellbasiert“, „Ausdrucksstark“, „Wiederverwendbar“ und „Visualisierbar“ an die Methodik aufgezeigt werden. Die Details zum Vorgehen der Methodik sind bereits in Kapitel 6 erläutert und stehen daher an dieser Stelle nicht im Fokus. Neben der Ausgangslage des Beispielszenarios werden zunächst wesentliche Hintergrundinformationen aufgezeigt. Mithilfe von Grafiken werden die ausschlaggebenden Elemente im Hinblick der Evaluierung der Methodik dargestellt. Eine beispielhafte Illustration des Umgebungsmodells (für die Angriffssimulation) ist in Abbildung 6.6 abgebildet. Im Kontext der Beispielszenarien ist die Abbildung des Umgebungsmodells nicht vorgesehen, da die Darstellung und Implementierung des Umgebungsmodells nicht Teil dieser Arbeit sind. Der generelle Ablauf von „Explore“ ist in Abbildung 6.3 abgebildet. Das Vorgehen für „Explore“ ist anhand dieser Arbeit festgelegt und somit für jedes Angriffsszenario gleich, das die Taktik „Explore“ verwendet. Daher wird in diesem Kapitel das umfassende Diagramm von „Explore“ nicht erneut abgebildet. Ausschlaggebend dabei sind die Ergebnisse im Rahmen einer Angriffsiteration im Kontext der Angriffsszenarien. Bezüglich der Technik-Ebene werden die möglichen Inhalte der Access Points Library, Exploit Library und Attack Technique Library an dieser Stelle nicht explizit aufgezeigt. Diese Libraries sind unabhängig von dem zu modellierenden Angriff an sich, sodass die Sammlungen für jede Anwendung der Methodik als Datenbasis fungieren. Sie enthalten entsprechend alle bekannten, aktuellen Informationen der jeweiligen Thematik.

Angriffsszenario aus dem Webbereich Für die Evaluierung wird die entwickelte Methodik zur Angriffsmodellierung zunächst an einem weiteren beispielhaften Angriffsszenario aus dem Bereich der Webanwendungen durchgeführt. Das Beispiel beruht auf den Top Ten 2017 von OWASP [78]. Die Ausführung dieses Abschnitts basiert auf den Informationen in [78]. Die OWASP Top Ten 2017 repräsentieren umfassende Informationen zu den zehn größten Risiken für Application Security 2017. Als geschätzte Community dienen diese Ranglisten als Leitfaden für viele Organisationen.[78] Das bedeutet, diese zehn Risiken bzw. Schwachstellen sind für viele Unternehmen und Organisationen von

Bedeutung. Somit wird durch die Anwendung der Methodik im Kontext der relevanten Schwachstellen von OWASP indirekt der Nutzen der Methodik demonstriert. Der Aufbau und die Entwicklung der Methodik zur Angriffsmodellierung basieren bereits auf einem Beispiel aus dem Bereich „Injection“, wie in Kapitel 6 beschrieben. In den Top Ten 2017 von OWASP stehen Injection-Schwachstellen und die damit verbundenen Risiken auf dem ersten Platz. Gemäß der Reihenfolge der OWASP Top Ten 2017 stehen an zweiter Stelle Angriffe im Bereich „Broken Authentication“ im Umfeld von Webanwendungen. Daher wird das Beispielszenario für die Evaluierung im Zusammenhang mit „Broken Authentication“ stehen. Die Ausgangslage dieses Angriffsszenarios setzt sich aus den folgenden Aspekten zusammen:

- Ein Angreifer möchte irgendeine Identität einer berühmten Persönlichkeit auf einer Social Media-Plattform übernehmen.
- Der Angreifer geht davon aus, dass eine Social Media-Plattform eine Anwendungsfunktion bezüglich der Authentisierung aufweist.
- Der Angreifer ist als Cyber-Krimineller ein Experte in seinem Anwendungsgebiet.
- Zudem ist er im Besitz von hilfreichen Ressourcen. Er hat zum einen Zugriff auf Millionen von Benutzerdatensätzen (Username/Passwort). Zum anderen kontrolliert er ein mächtiges Botnetz.
- Die Informationen im Kontext dieses Beispielszenarios stammen aus [78].

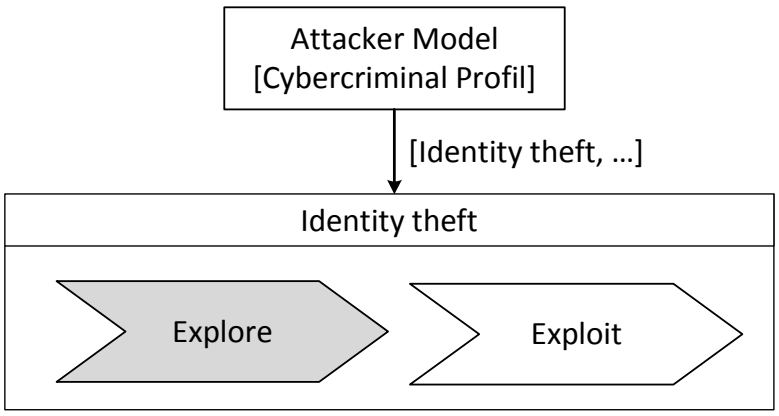

Abbildung 7.1: Der Angriff zur Übernahme einer fremden Identität auf Taktik-Ebene

Die relevanten Ergebnisse bezüglich Taktik- und Technik-Ebene sind nach Anwendung der entwickelten Methodik in Abbildung 7.1 und Abbildung 7.2 illustriert. Es folgt eine kurze Beschreibung dieser Abbildungen. Die Taktik-Ebene in Abbildung 7.1 besteht zu einem bestimmten Zeitpunkt mithilfe eines geeigneten Verfahrens aus den taktischen Zielen „Explore“ und „Exploit“. Gemäß den Informationen aus dem Angreiferprofil versucht der Angreifer, im Rahmen seiner Möglichkeiten sein strategisches Ziel „Übernahme einer anderen Identität“ mithilfe der beiden Taktiken (Explore, Exploit) zu erreichen. Das taktische Ziel „Explore“ wird über ein geeignetes Verfahren ausgewählt und in der Ablauf-Ebene spezifiziert. Bis zu der Tree-Action „Test Target“ aus dem Vorgehen von „Explore“, im Rahmen einer Angriffsiteration ergeben sich folgende Ergebnisse[4]:

- **Identified Access Points:** URL, User Input Field
- **Attacker Model:** Der Angreifer ist ein Cyber-Krimineller. Er besitzt hilfreiche Fachkenntnisse und Zugriff auf nützliche Ressourcen (Millionen von Benutzerdatensätzen, Botnetz)
- **Current Access Point:** User Input Field

Die Abbildung 7.2 stellt einen möglichen Ausschnitt der Spezifikation von „Test Target“ für dieses Beispielszenario dar. In der Technik-Ebene, wie in Abbildung 7.2 abgebildet, wird der Exploit „Vulnerable Exploit“ bezüglich „Credential Stuffing“ im Rahmen eines geeigneten Verfahrens für die Angriffssimulation identifiziert.

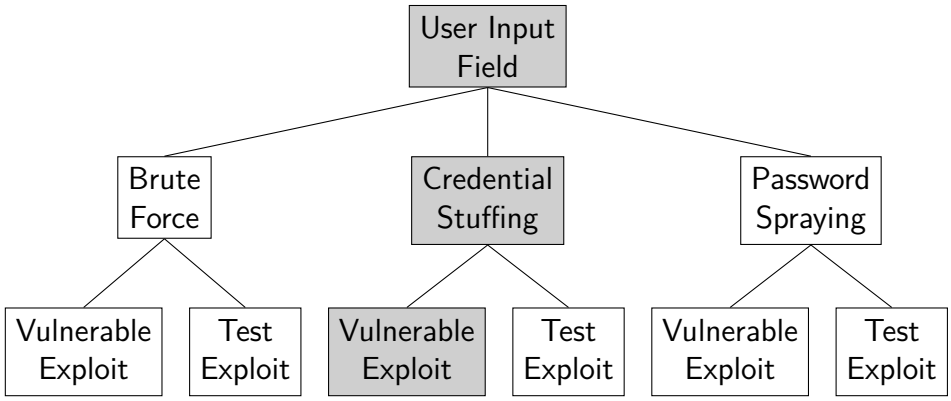

Abbildung 7.2: Technik-Ebene bezüglich des Zugangspunkts User Input Field

[4]Diese repräsentieren einen beispielhaften Ausschnitt an möglichen Ergebnissen.

Es wird angenommen, dass der „Vulnerable Exploit“ im Kontext der Angriffssimulation erfolgreich ausgeführt wird. Die Ergebnisse der Simulation werden wieder an geeigneter Stelle in das Angriffsmodell integriert. Daraus folgen die restlichen Ergebnisse im Kontext von „Explore“ (Ablauf-Ebene) innerhalb dieser Angriffsiteration:

- **Result:** Vulnerable for Credential Stuffing
- **Environment Knowledge Model:** Nach der Angriffssimulation des identifizierten Exploits (Vulnerable Exploit) bezüglich „Credential Stuffing“, wird das Angreiferwissen entsprechend aktualisiert. Das Ergebnis der Angriffssimulation zeigt, dass mindestens eine Social Media-Plattform zur Verfügung steht, worauf Angriffe in Form von „Credential Stuffing“ über den Zugangspunkt „User Input Field“ bezüglich „Credential Stuffing“ möglich sind. Das bedeutet, dass z. B. die Anwendung der Plattform keine Mechanismen enthält, die „Credential Stuffing“ automatisiert erkennt bzw. mit Sicherheitsmechanismen unterbindet.

Mindestens eine Social Media-Plattform, die der Angreifer bezüglich der Anfälligkeit für „Credential Stuffing“ getestet hat, stellt somit die Möglichkeit für einen Zugriff über „User Input Field(s)“ dar. In der Praxis überlegt der Angreifer z. B. zunächst bezüglich seiner Möglichkeiten, welche Technik er für den Angriff verwendet, um anschließend zu testen, ob die Authentifizierung mithilfe dieser Technik angreifbar ist. Das entspricht einer Angriffsiteration im Rahmen von „Explore“ (Taktik).

Angriffsszenario aus dem Bereich Embedded-Systems Neben einem Beispiel aus dem typischen Bereich der Webanwendungen, wird die Methodik zur Angriffsmodellierung für die Evaluierung an einem beispielhaften Angriffsszenario im Kontext des aktuell laufenden Projekts „MASSiF“ angewandt, welches in Abschnitt 5.1 zur Sprache kommt. Das Angriffsszenario basiert auf folgender Ausgangslage:

- Bei einem Autounfall wurde der Front-Airbag ausgelöst.
- Obwohl der Fahrer des Autos nicht angeschnallt war, blieb er nahezu unverletzt.
- Der Fahrer möchte seinen Fehler „Nicht-angeschnallt-zu-sein“ bezüglich der aufgezeichneten Daten im Auto im Kontext des Unfalls verheimlichen[5].
- Der Fahrer besitzt keine Fachkenntnisse, um sein Ziel zu erreichen. Daher übergibt er diesen Auftrag an jemanden (Angreifer), der sich mit dieser Thematik auskennt und die damit in Verbindung stehenden, unerlaubten Handlungen für Geld regelmäßig durchführt.

[5] Andere Faktoren, die in der Realität für die Unfallanalyse ebenso eine wichtige Rolle spielen, werden für dieses minimale Beispiel nicht beachtet.

Das Unfallfahrzeug besitzt einen Event Data Recorder (EDR) auf der Airbag Control Unit. Der EDR stellt eine Funktion zur Erfassung, Speicherung und Abrufbarkeit von Unfalldaten bei Kraftfahrzeugen dar.[59, 81] Der EDR des Unfallautos ist gemäß „Code of Federal Regulations“ [81] der USA entwickelt, wodurch unter anderem das Daten Element „safety belt status, driver“ mit den Werten „on/off“ aufgezeichnet wird. Mithilfe dieses Elements wird im Rahmen des EDR festgehalten, ob der Fahrer angeschnallt war oder nicht.[81] Die Informationen im Kontext des Beispielszenarios zu der Thematik EDR stammen aus [15, 81, 59].

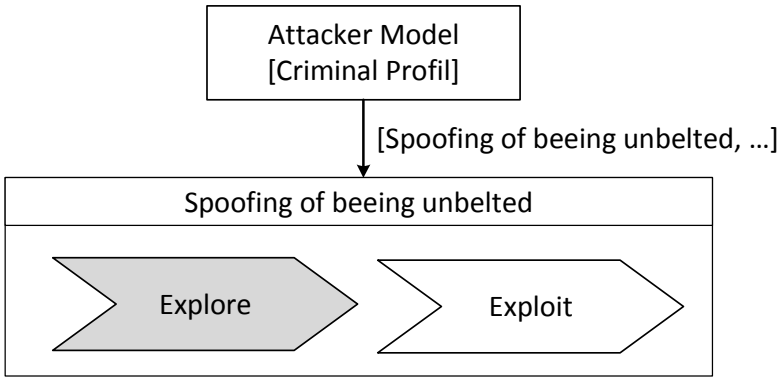

Abbildung 7.3: Der Angriff zur Verschleierung des Nicht-angeschnallt-Seins auf Taktik-Ebene

In Abbildung 7.3 und Abbildung 7.4 sind nach Anwendung der Methodik zur Angriffsmodellierung die relevanten Ergebnisse der Taktik- und Technik-Ebene dargestellt. Im Folgenden werden die beiden Grafiken Abbildung 7.3 und Abbildung 7.4 im Überblick präsentiert.

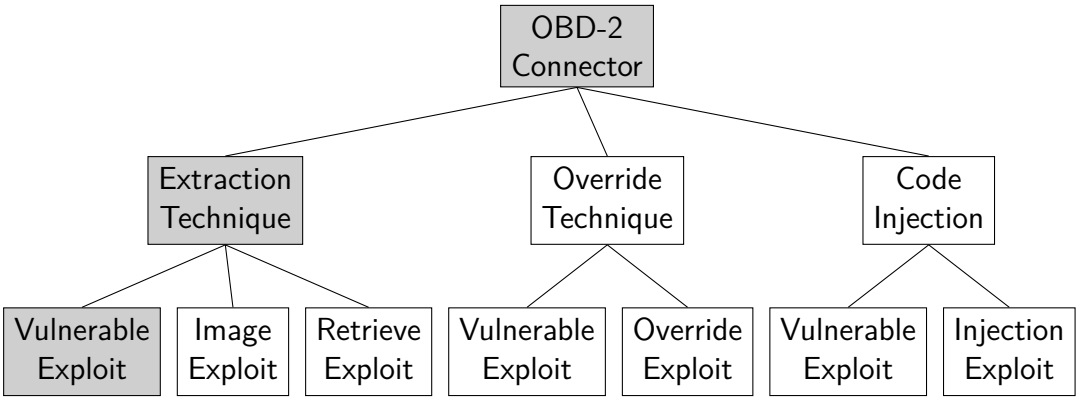

Abbildung 7.4: Technik-Ebene auf Basis des On-Board-Diagnose (OBD)-2 Connectors als Zugangspunkt

Die Taktik-Ebene in Abbildung 7.3 besteht zu einem bestimmten Zeitpunkt mithilfe eines geeigneten Verfahrens aus den taktischen Zielen „Explore“ und „Exploit“. Gemäß den Informationen aus dem Angreiferprofil versucht der Angreifer, als Experte in seinem Anwendungsgebiet, sein strategisches Ziel „Verschleierung des Nicht-angeschnallt-Seins“ mithilfe der beiden Taktiken (Explore, Exploit) zu erreichen. Das taktische Ziel „Explore“ wird über ein geeignetes Verfahren ausgewählt und in der Ablauf-Ebene spezifiziert. Folgende Ergebnisse[6] ergeben sich im Kontext der Ablauf-Ebene innerhalb einer Angriffsiteration bis zur Tree-Action „Test Target“ aus dem Vorgehen von „Explore“:

- **Identified Access Points:** OBD-2 Connector, EPROM
- **Attacker Model:** Der Angreifer ist ein Experte auf seinem Gebiet. Er besitzt die notwendigen Fachkenntnisse und Ressourcen. Außerdem hat er direkten (physikalischen) Zugriff auf das Unfallauto.
- **Current Access Point:** OBD-2 Connector

„Test Target“ wird in der Technik-Ebene spezifiziert. Abbildung 7.4 stellt einen möglichen Ausschnitt der Technik-Ebene für dieses Beispielszenario dar. In der Technik-Ebene, wie in Abbildung 7.4 abgebildet, wird der Exploit „Vulnerable Exploit“ bezüglich „Extraction Technique“ im Rahmen eines geeigneten Verfahrens für die Angriffssimulation identifiziert. Es wird angenommen, dass der „Vulnerable Exploit“ im Kontext der Angriffssimulation erfolgreich ausgeführt wird. Die Ergebnisse der Simulation werden wieder an geeigneter Stelle im Angriffsmodell integriert. Die Ergebnisse der restlichen Aktionen von „Explore“ innerhalb einer Angriffsiteration sind folgende:

- **Result:** Vulnerable for Extraction Technique
- **Environment Knowledge Model:** Nach der Ausführung des Vulnerable Exploits (Angriffssimulation) wird das Angreiferwissen entsprechend aktualisiert. Das Ergebnis der Angriffssimulation zeigt, dass ein Zugriff auf den EDR über den OBD-2 Connector als Zugangspunkt fungiert. Das bedeutet z. B., dass der EDR durch den Unfall nicht beschädigt bzw. zerstört wurde.

Der EDR stellt somit die Möglichkeit für einen Zugriff auf die Extration von Daten über den OBD-2 Connector (CAN Bus) dar. In der Realität überlegt der Angreifer beispielsweise im Rahmen seiner Möglichkeiten, welches Werkzeug er für seinen Angriff verwenden kann, um anschließend zu testen, ob er damit den EDR erfolgreich ansprechen kann. Das entspricht einer Angriffsiteration im Rahmen von „Explore“ der entwickelten Methodik.

[6] Diese repräsentieren einen beispielhaften Ausschnitt an möglichen Ergebnissen.

7.4 Interpretation und Bewertung

Für eine erfolgreiche Evaluierung der Methodik sollen die festgelegten Evaluierungskriterien im Rahmen der Anforderungen erfüllt werden.

Modellbasiert Die Methodik zur Angriffsmodellierung baut auf verschiedene Modellierungskonzepte auf. Die strategische Sichtweise auf den Angriff wird in der Taktik-Eben mithilfe eines einfachen Prozessflussdiagramms modelliert, wie in Abbildung 7.1 bzw. Abbildung 7.3 zu sehen. Die Ablauf-Ebene basiert auf UML-Aktivitätsdiagrammen, wie in Abbildung 6.3 abgebildet. Die technische Sicht gründet auf der Modellierungsmethodik Attack Trees, wie in Abbildung 7.2 und Abbildung 7.4 veranschaulicht. Darüber hinaus soll die Datenbasis für die entwickelte Methodik mithilfe von Modellen bzw. systematischen Strukturen aufgebaut sein, wie das notwendige Angreifermodell, Umgebungsmodell und Umgebungs-Wissens-Modell aus Abschnitt 5.1. Allerdings sind Entwicklung und Implementierung dieser Modelle nicht Teile dieser Arbeit. In diesem Abschnitt wird das Evaluierungskriterium *Modellbasiert* insofern bestätigt, dass die Methodik auf verschiedenen Modellierungsansätzen basiert. Die Grundlage ist geschaffen, um Angriffe systematisch, schlüssig und qualitativ hochwertig abbilden zu können.

Wiederverwendbar Für die Evaluierung der Anforderung „Wiederverwendbar“ soll das Kriterium *Wiederverwendbare Elemente* demonstriert werden. Die unterschiedlichen Abstraktionsebenen (Taktik-, Ablauf-, Technik-Ebene), worauf die Methodik aufgebaut ist, unterstützt die Wiederverwendbarkeit. Für die Arbeit wird davon ausgegangen, dass der entwickelten Methodik definierte Taktiken zur Verfügung stehen, die in einem geeigneten Verfahren in der Taktik-Ebene ausgewählt werden. Beispielsweise zeigen Abbildung 7.1 und Abbildung 7.3 die ausgewählte Taktik „Explore“ (grau hinterlegt), die in den Angriffsszenarien jeweils wiederverwendet wird. Inwieweit die Taktik „Exploit“ im Rahmen der beiden Beispiele wiederverwendet wird, ist nicht festzustellen, da „Exploit“ in der Masterarbeit nicht spezifiziert wird. Die Methodik basiert jedoch darauf, dass die definierten und spezifizierten Taktiken unabhängig des zu modellierenden Angriffs wiederverwendet werden. Wie bereits in Abschnitt 7.3 erläutert, wird „Explore“ bzw. die Spezifikation von „Explore“ (Ablauf-Ebene) aus Abbildung 6.3 z. B. für beide Szenarien in Abschnitt 7.3 wiederverwendet. Der dynamische Wissensaufbau des Angreifers präsentiert ein weiteres wiederverwendbares Konstrukt. Das Wissen des Angreifers wird in jeder Angriffsiteration als Grundlage wiederverwendet und aktualisiert. Die Modellierung des Angriffs kann sowohl auf Basis eines „leeren“ Umgebungs-Wissens-Modells[7] angewandt werden, als auch mit der Annahme, dass der Angreifer bereits im Besitz von gewissen Informationen über das Target und dessen Umgebung ist.

[7] Ein „leeres“ Umgebungs-Wissens-Modell präsentiert das aktuelle Wissen eines Angreifers, wobei dieser noch keine Kenntnisse zu dem Target und dessen Umfeld besitzt.

Neben dem Umgebungs-Wissens-Modell sollen auch Inhalte der anderen Abhängigkeiten aus Abschnitt 5.1 wiederverwendet werden. Die Libraries von Angriffstechniken, Zugangs- bzw. Angriffspunkten und Exploits sollen für jede Angriffsmodellierung notwendige Informationen bereitstellen, für eine einfache und strukturierte Wiederverwendung. Beispielsweise steht die „URL“ und das „User Input Field“ als mögliche Zugangs- bzw. Angriffspunkte sowohl im umfassenden Beispiel zur Erläuterung der Methodik in Kapitel 6 zur Verfügung, als auch im Beispielszenario auf Basis von „Broken Authentication“ in Abschnitt 7.3. Ebenso wird z. B. „Code Injection“ sowohl in Abbildung 6.5, als auch in Abbildung 7.4 als Angriffstechnik bereitgestellt. Das Angreifermodell soll einen Angreifer über wiederverwendbare Angreiferprofile modellieren, sodass diese Profile als wiederverwendbare Inputs für die entwickelte Methodik fungieren. Im Gegensatz dazu werden beispielsweise im Rahmen von Attack Trees keine wiederverwendbaren Angreiferinformationen miteinbezogen, wie in Kapitel 2 beschrieben. Das Umgebungsmodell (für die Simulation) soll mit geeigneten Konstrukten zur Wiederverwendung aufgebaut werden, um sie mit jeder Angriffsiteration (im Kontext eines zu modellierenden Angriffs) zu verwenden und zu aktualisieren. Allerdings steht die Implementierung dieser Modelle und Libraries nicht im Fokus der Arbeit. Neben der Annahme, dass diese Abhängigkeiten (Modelle, Libraries) existieren, wird das Evaluierungskriterium *Wiederverwendbare Elemente* für die entwickelte Methodik erfüllt.

Ausdrucksstark Für die Anforderung „Ausdrucksstark“ sind die damit verbundenen Kriterien *Relevante Angriffe* und *Unabhängigkeit der Anwendungsdomäne* zu prüfen.

Durch die erfolgreiche Anwendung der Methodik zur Angriffsmodellierung an den Beispielen in Abschnitt 7.3 wird auch das Kriterium *Relevante Angriffe* abgedeckt. Wie bereits in Abschnitt 7.3 erläutert, gründet das eine beispielhafte Angriffsszenario auf Schwachstellen im Bereich „Broken Authentication“. Dieser Typ an Schwachstellen steht an zweiter Stelle der OWASP Top Ten 2017.[78] Das zeigt die hohe Relevanz dieser Thematik und den damit verbundenen Angriffen. Folglich zielt die erfolgreiche Anwendung der Methodik an dem beispielhaften Angriffsszenario bezüglich „Broken Authentication“ auf die Erfüllung des Kriteriums *Relevante Angriffe* ab. Das zweite Beispiel in Abschnitt 7.3 basiert auf dem aktuell laufenden Projekt „MASSiF“, wie in Abschnitt 7.1 erwähnt. Im Rahmen von „MASSiF“, das im Bereich Automotive angesiedelt ist, sind unter anderem die Angriffsmöglichkeiten im Kontext des EDR gefragt. Dahingehend bietet die entwickelte Methodik eine geeignete Grundlage, um relevante Informationen dazu zu erhalten. Zudem wird das entwickelte Modellierungskonzept zur Angriffsmodellierung im Rahmen aktueller Arbeiten der Forschungsgruppe von Prof. Dr.-Ing. Hans-Joachim Hof der Technischen Hochschule Ingolstadt als Grundlage verwendet und weiterentwickelt. Daraus lässt sich der entwickelten Methodik zur Angriffsmodellierung ein hoher Nutzen zuschreiben. Der Abschnitt demonstriert das Kriterium *Relevante Angriffe* im Rahmen der beispielhaften Angriffsszenarien.

Die erfolgreiche Durchführung der entwickelten Methodik zur Angriffsmodellierung im Rahmen unterschiedlicher Domänen in Abschnitt 7.3 zeigt, dass die Methodik unabhängig von der Domäne des Angriffs verwendet werden kann. Neben den beispielhaften Angriffen im Webbereich in Kapitel 6 und in Abschnitt 7.3, wird die Methodik für die Evaluierung in Abschnitt 7.3 zudem im Bereich von Embedded-Systems angewandt. In diesen beiden Bereichen gibt es z. B. unterschiedliche Anforderungen im Hinblick auf die Sicherheit an eine Cyber-Enabled Capability. Während bei Webapplikationen die Security im Vordergrund steht, sind bei Embedded-Systems die Safety-Aspekte nicht zu vernachlässigen.[1] In diesem Zusammenhang ist z. B. das generische Angreifer- und Angriffsmodell von Adepu und Mathur auf physikalische Systeme fokussiert, wie in Kapitel 2 erwähnt. Im Kontext der entwickelten Methodik ist die Darstellung der verschiedenen Sichtweisen auf einen Angriff nicht von der Anwendungsdomäne des Angriffs abhängig. Der Angriff bzw. der Angriffsablauf steht vielmehr im Zusammenhang mit dem Angreifer.

Wie in Kapitel 2 zu lesen, ist dagegen die Lockheed Martin Cyber Kill Chain z. B. statisch auf typische APTs ausgelegt und steht im Zusammenhang mit Malware. Die Entwicklung einer Malware ist nicht im Vordergrund der Methodik zur Angriffsmodellierung. Über ein geeignetes Verfahren stehen verschiedene Techniken, zu einem bestimmten Zeitpunkt im Fokus der Methodik. Ein weiterer Modellierungsansatz mit gewissen Einschränkungen aus Kapitel 2 sind Attack Graphen, die hauptsächlich im Rahmen von „Privilege Escalation“ in Umgebung von Netzwerken genutzt werden. Im Evaluierungsbeispiel aus dem Bereich Embedded-System besitzt der Angreifer lokalen (physikalischen) Zugriff auf das Target, was z. B. über den Scope des Angreiferprofils präsentiert werden kann. Die erfolgreiche Anwendung der Methodik an dem beispielhaften Szenario in Abschnitt 7.3, aus dem Bereich Embedded-System zeigt, dass ein Angriff unabhängig von APTs, einer Malware-Nutzung, einer Netzwerkumgebung oder der Technik „Privilege Escalation“ modelliert werden kann.

Die Methodik ermöglicht die Modellierung des Angriffsablaufs, bzw. der genauen Vorgehensweise in Abhängigkeit des Angreifers. Gleichzeitig wird der dynamische Wissensaufbau, im Zusammenhang mit den notwendigen Informationen zu dem Target und dessen Umfeld, für die Modellierung des Angriffs beachtet. Dieses Element ist für die iterative Modellierung des Angriffsvorgehens notwendig. Dagegen bilden beispielsweise optionsorientierte Modellierungen, wie Attack Trees weder die genaue Vorgehensweise eines Angreifers (Angriffsablauf) ab, noch gibt es eine systematische Integration von relevanten Informationen zu Target oder Umgebung, wie in Kapitel 2 beschrieben. Ebenso wird z. B. im generischen Angreifer- und Angriffsmodell für physikalische Systeme von Adepu und Mathur das strategische Vorgehen eines Angreifers und die damit in Verbindung stehenden Informationen nicht beachtet, wie in Kapitel 2 erläutert.

Mithilfe der Methodik sind Exploits abbildbar, die z. B. bei der Angriffsmodellierung im Kontext von TTP nicht aufgefasst werden, wie in Kapitel 2 aufgezeigt. In der entwickelten Methodik sorgt die geeignete Kombination verschiedener Abstraktionsstufen,

in Verbindung mit geeigneten Modellierungen dafür, sowohl die Prozesssichtweise als auch die technische Sicht auf einen Angriff einzubinden. In der Ablauf-Ebene, wie in Abbildung 6.3 abgebildet, sind die einzelnen Aktionen bezüglich des taktischen Ziels „Explore“ sichtbar. Die damit in Verbindung stehende Technik-Ebene präsentiert die Exploits. Im Gegensatz dazu liegt beispielsweise der Fokus der Lockheed Martin Cyber Kill Chain auf der reinen Prozessdarstellung eines Angriffs, wie in Kapitel 2 beschrieben.

Die Methodik bietet die Möglichkeit weitere erforderliche Elemente einzubinden, die nicht in der Masterarbeit identifiziert wurden. Neben dem taktischen Ziel „Explore“ können andere erforderliche taktische Ziele definiert und in der Ablauf-Ebene spezifiziert werden. Das Framework auf Basis eines einfachen Prozessflussdiagramms, UML-Aktivitätsdiagramme und Attack Trees stellen geeignete Mechanismen bereit, relevante Elemente hinzuzufügen.

Ebenso bieten die Sammlungen von Zugangs- bzw. Angriffspunkten, Exploits und Angriffstechniken zu allen bekannten Angriffen eine stetig wachsende Datenbasis an. Neue Angriffe sollen als neue Grundlage in die Sammlungen integriert werden. In diesem Zusammenhang sind die Zugangs- bzw. Angriffspunkte (Access Point Library) von Bedeutung. Mithilfe der Punkte werden domänenspezifische Stellen illustriert, die für die Durchführung eines Angriffs (Angriffsiteration) verwendet werden. Sie stellen den Startpunkt für einen Angriff aus technischer Sicht dar. Daran ist ersichtlich, dass notwendige Elemente, wie domänenspezifische Aspekte, an geeigneter Stelle in die Methodik zur Angriffsmodellierung eingebunden werden können.

Die erfolgreiche Anwendung der Methodik an den Beispielen in Abschnitt 7.3 zeigt, dass das Ergebnis (modellierter Angriff) unabhängig von der Anwendungsdomäne für verschiedene Ziele verwendet werden kann. Beispielsweise können die Ergebnisse als Kommunikationsmittel zwischen verschiedenen Stakeholdern verwendet werden. Ein modellierter Angriff, mithilfe der entwickelten Methodik kann beispielsweise als Grundlage für Security-Analysen, wie der Risikoanalyse verwendet werden. Die umfassende Darstellung potentieller Angriffe bietet eine geeignete Informationsquelle für Risikoanalysen, sodass Sicherheitsmaßnahmen bzw. Abwehrmechanismen auf Grundlage des modellierten Angriffs abgeleitet werden können. Ebenso stellt das systematische Vorgehen eines Angreifers, in Zusammenspiel des strukturierten Aufbaus der Ebenen eine mögliche Vorgehensweise für Penetrationstests zur Verfügung. Im Gegensatz dazu zielt ATT&CK von The MITRE Corporation lediglich auf den Test und die Verifikation von Verteidigungsmaßnahmen gegen weitverbreitete Angriffstechniken ab, wie in Kapitel 2 beschrieben. Eine systematische Darstellung und Einbindung von Exploits, der tatsächlichen Umgebung und die Vorgehensweise eines Angreifers (Prozess) sind in ATT&CK nicht vorgesehen. Im Rahmen dieser Abschnitte wird das Evaluierungskriterium *Unabhängigkeit der Anwendungsdomäne* bestätigt.

Systematisch Mithilfe des Kriteriums *Systematische Strukturen* soll die Anforderung nach einer Systematik für die entwickelte Methodik geprüft werden. Den drei Ebenen der entwickelten Methodik liegt jeweils eine strukturierte Vorgehensweise zugrunde, wie in Kapitel 6 erläutert. Die Taktik-Ebene wird anhand eines einfachen Prozessflussdiagramms aufgebaut. Die verfügbaren Taktiken sind mit gewissen Vor- und Nachbedingungen verknüpft. Mithilfe eines geeigneten Verfahrens werden die relevanten Taktiken gewählt und über die darunterliegende Ablauf-Ebene spezifiziert. Die Gesamtheit aller möglichen Abläufe im Rahmen eines taktischen Ziels basiert auf dem systematischen Vorgehen in einem UML-Aktivitätsdiagramm. Z. B. sind Entscheidungsknoten mit bestimmen Bedingungen verknüpft, sodass sie das Vorgehen systematisch regeln. Über „Test Target“ wird eine Verbindung zu der Technik-Ebene geschaffen. Diese bildet eine strukturierte Darstellung von Angriffstechniken ab und bietet eine ideale Grundlage für ein systematisches Verfahren zur Identifikation eines Exploits.

Der ausgewählte Exploit wird anschließend in einem geeigneten Umgebungsmodell simuliert, sodass die Ergebnisse über die Technik-Ebene in das Angriffsmodell eingepflegt werden können. Diese Systematik ist notwendig, um den dynamischen Wissensaufbau des Angreifers zu gewährleisten, sodass ein Angriff iterativ abgebildet werden kann. Dagegen ist z. B. bei der Lockheed Martin Cyber Kill Chain keine dynamische Systematik für die Modellierung eines Angriffs vorgesehen, wie in Kapitel 2 beschrieben. Dagegen unterstützt das systematische Vorgehen der entwickelten Methodik die nachvollziehbare und wiederholbare Modellierung von Angriffen. Das bedeutet, während z. B. im Microsoft SDL die Modellierung eines Systems im Vordergrund steht, werden mithilfe der entwickelten Methodik Informationen zum Angriff und dem damit verbundenen Target und dessen Umgebung durchgehend, iterativ und systematisch eingebunden.

Neben dem Umgebungsmodell sind auch die anderen Abhängigkeiten (Angreifermodell, Umgebungs-Wissensmodell, Sammlungen von Angriffstechniken, Exploits und Zugangs- bzw. Angriffspunkten) mit einer passenden Systematik zu implementieren, wie in Abschnitt 5.1 beschrieben. Z. B. sollen die Informationen aus den Sammlungen von Angriffstechniken, Exploits und Zugangs- bzw. Angriffspunkten strukturiert, an einer zentralen Stelle abrufbar sein. Die öffentlichen Datenbanken von The MITRE Corporation bieten bereits strukturierte Inhalte zu dem Themenbereich Angriff.[61, 67, 69, 68] Allerdings liegt der Fokus auf der Darstellung aller relevanten Informationen zur Abschwächung von Angriffen. In diesem Zusammenhang sind die notwendigen Informationen für die entwickelte Methodik, wie z. B. Zugangs- bzw. Angriffspunkte nicht einfach, systematisch abrufbar. Diese Informationen befinden sich vorwiegend im Fließtext der Einträge, wie in Abschnitt 5.1 beschrieben. Allerdings sind Implementierung und Umsetzung der Abhängigkeiten aus Abschnitt 5.1 nicht Teile der Masterarbeit. Der Abschnitt zeigt die Erfüllung des Kriteriums *Systematische Strukturen* und somit die damit verbundene Anforderung an die Methodik, mit der Einschränkung, dass die konkrete Implementierung nicht Teil der Arbeit ist und somit nicht evaluiert wird.

Simulierbar/Konsistent Wie bereits erwähnt, ist die tatsächliche Implementierung der Modelle für die Methodik nicht im Fokus der Arbeit und kann daher nicht geprüft werden. Folglich wird aufgrund der fehlenden Implementierung der Methodik und die damit in Verbindung stehenden Modelle und Sammlungen die Anforderung „Simulierbar/Konsistent“ im Rahmen der Masterarbeit nicht bewertet.

Visualisierbar Die Anforderung „Visualisierbar“ wird mithilfe des Kriteriums *Visuelle Elemente* evaluiert. Im Kontext der entwickelten Methodik zur Angriffsmodellierung werden die drei Ebenen (Taktik-, Ablauf-, und Technik-Ebene) in verschiedenen Abstraktionsstufen grafisch abgebildet. Die Taktik-Ebene wird als einfaches Prozessflussdiagramm präsentiert, wie in Abbildung 7.1 bzw. Abbildung 7.3, im Rahmen der Beispielszenarien zu sehen. Die Ablauf-Ebene basiert auf UML-Aktivitätsdiagrammen, wie in Abbildung 6.3 präsentiert. Die Technik-Ebene basiert auf Attack Trees zur grafischen Darstellung, wie in Abbildung 7.2 bzw. Abbildung 7.4 illustriert. Durch diese visuellen Abstraktionsstufen wird die Komplexität eines Angriffs handhabbar und besser verständlich, z. B. im Vergleich zu Beschreibungen von Angriffen in reinem Prosatext. Ein weiteres Beispiel sind Attack Graphen, die zunehmend komplexer werden, abhängig der abzubildenden Netzwerkgröße. Dagegen wird in der Methodik die Komplexität der Graphen in der Technik-Ebene, über die zuvor liegenden Abstraktionsstufen und den (dynamischen) Aufbau der Technik-Ebene auf Basis verschiedener Informationen (z. B. mithilfe der Sammlungen von Zugangspunkten, Angriffstechniken und Exploits) vereinfacht. Die Identifikation des Exploits auf Technik-Ebene kann auf Grundlage von Attack Trees visuell hervorgehoben werden, wie in Abbildung 7.2 bzw. Abbildung 7.4 abgebildet. Beispielsweise ist der Angriffsvektor in Abbildung 7.4 visuell hervorgehoben (graue Knoten), durch den Weg von der Wurzel „OBD-2 Connector“, über die Technik „Extraction Technique“ bis zum identifizierten Exploit „Vulnerable Exploit“.

Sowohl das Umgebungsmodell für die Simulation, als auch das Umgebungs-Wissens-Modell für den dynamischen Wissensaufbau des Angreifers sollen grafisch abbildbar sein. Die genaue Darstellung dieser Modelle ist nicht Teil der Masterarbeit. Dennoch sind die Modelle wesentliche Elemente der Methodik zur Angriffsmodellierung, die das Kriterium *Visuelle Elemente* unterstützen. Die grafischen Bestandteile der Methodik zur Angriffsmodellierung vereinfachen deren Benutzung, wie in Abschnitt 4.1 beschrieben. Im Gegensatz dazu werden z. B. Angriffe mithilfe ATT&CK nicht visuelle präsentiert. In diesem Abschnitt wird das Evaluierungskriterium *Visuelle Elemente* bewiesen, wobei die konkrete Implementierung nicht berücksichtigt wird.

Verständlich Das Evaluierungskriterium *Visuelle Elemente*, das bereits zuvor demonstriert wurde, trägt zur Erfüllung der Anforderung „Verständlich“ bei. Dennoch kann die Anforderung „Verständlich“ im Rahmen der Arbeit nicht bewiesen werden, wie in Abschnitt 7.2 erläutert. Die entwickelte Methodik wird in dieser Arbeit nicht konkret

implementiert bzw. über eine konkrete Syntax und Semantik definiert. Das ist Voraussetzung für die praktische Anwendung der Methodik zur Angriffsmodellierung und deren Evaluierung im Hinblick auf die Verständlichkeit. Wie in den Beispielen aus Abschnitt 7.3 zu sehen, wird ein Angriff zum jetzigen Zeitpunkt überwiegend auf abstrakte Weise aus einer Kombination von Modellelementen und Fließtext präsentiert. Inwieweit die Methodik im Rahmen einer definierten Syntax und Semantik, im Kontext der verschiedenen Ebenen einfach und verständlich ist, kann erst nach Implementierung des Angriffsmodells geprüft werden. Folglich wird im Rahmen der Arbeit die Anforderung „Verständlich“ nicht evaluiert.

7.5 Schlussfolgerungen

Für die Evaluierung sind sieben Anforderungen an die Methodik zur Angriffsmodellierung, die in Tabelle 4.1 aufgelistet sind, untersucht worden. Im Rahmen der Masterarbeit wurden fünf der sieben Anforderungen weiter betrachtet. Konkret werden die Anforderungen „Modellbasiert“, „Wiederverwendbar“, „Ausdrucksstark“, „Systematisch“ und „Visualisierbar“ im Zusammenhang mit den festgelegten Kriterien erfüllt. Dabei gelten gewisse Annahmen, da die konkrete Implementierung der Methodik und die damit verbundenen Abhängigkeiten nicht Teil der Masterarbeit sind. Aufgrund der relevanten Grundlagen, worauf die beispielhaften Angriffsszenarien aufgebaut sind, steht es zu erwarten, dass diese fünf Anforderungen der Methodik, bezüglich der festgelegten Evaluierungskriterien, unabhängig des zu modellierenden Angriffs erfüllt werden. Im Rahmen der Masterarbeit wurden die Anforderungen „Simulierbar/Konsistent“ und „Verständlich“ nicht näher untersucht, da die Voraussetzung der konkreten Implementierung der Methodik und die damit verbundenen Abhängigkeiten nicht gegeben sind.

Folglich ist in dieser Arbeit ein Konzept entwickelt und umgesetzt worden, das eine geeignete Basis für die strukturierte und formale Modellierung von Angriffen bereitstellt. Der modellbasierte Kern der Methodik bietet eine ideale Grundlage für eine Formalisierung der Angriffsmodellierung. Die Basis eines generischen Modells fördert die Wiederverwendbarkeit und Ausdrucksstärke der Methodik, mit dem Ziel, alle bekannten Angriffe zu spezifizieren. Die verschiedenen Abstraktionsstufen stellen verschiedene Sichtweisen auf einen Angriff dar, sodass, unabhängig der Folgeaktivitäten, die relevanten Informationen bezüglich der Modellierung eines Angriffs präsentiert werden. Für eine qualitative Darstellung eines Angriffs ist die entwickelte Methodik im Hinblick auf die Angriffsdomäne von diversen Informationsquellen zu den Bereichen Angreifer, Wissen des Angreifers, tatsächliche Umgebung, bekannte Angriffstechniken, Zugangs- bzw. Angriffspunkte und Exploits abhängig.

Die Methodik stellt für die zielführende und nachvollziehbare Handhabung der riesigen Mengen an Informationen zur Angriffsmodellierung eine systematische Vorgehensweise bereit. Zwar wird die Angriffssimulation in der Masterarbeit nicht entwickelt bzw. evaluiert, trotzdem ist sie als ein maßgebender Bestandteil der entwickelten Methodik identifiziert. Die Angriffssimulation dient als Verknüpfung eines identifizierten Exploits mit den damit verbundenen Auswirkungen im Kontext zum aktuellen Target und dessen Umgebung. Somit wird die iterative und dynamische Modellierung eines Angriffs unterstützt. Der Fokus auf eine grafische Angriffsmodellierung steht in Verbindung zu den verschiedenen Abstraktionsstufen. Die Komplexität eines Angriffs wird somit handhabbar und besser verständlich. Im Vergleich zu den verwandten Arbeiten aus Kapitel 2 bietet die entwickelte Methodik eine systematische Grundlage, Security-Tests zu unterstützen.

7.6 Grenzen der Methodik

Die entwickelte Methodik hat gewisse Grenzen. Die Methodik kann lediglich bekannte Angriffe modellieren. Die Modellierung eines Angriffs setzt das Vorhandensein der relevanten Informationen für den Angriff voraus, um diesen (dynamisch) modellieren zu können. Das bedeutet, ein Angriff muss für die Anwendung der Methodik im Kontext von Zugangs- bzw. Angriffspunkten, Angriffstechniken und Exploits in den entsprechenden Sammlungen (Access Point Library, Attack Technique Library, Exploit Library) zur Verfügung stehen, sodass dieser abgebildet werden kann.

Die Methodik stellt keine Risikoanalyse bezüglich identifizierter Bedrohungen dar. Ziel der Methodik ist es, alle bekannten Angriffe auf Basis eines generischen und strukturierten Angriffsmodells dynamisch zu modellieren, unabhängig von dem Risiko des Angriffs.

Abschließend muss erwähnt werden, dass Modelle eine Abstraktion der Realität sind. Kriha weist in diesem Zusammenhang darauf hin, dass z. B. nicht jeder Sicherheitsmechanismus, der im Modell etabliert ist, auch wirklich in der Praxis anwendbar ist.[29] Im Hinblick auf die entwickelte Methodik bedeutet das z. B., dass ein modellierter Angriff von dem realen Angriff abweichen kann. Das Modell ist begrenzt, sodass unter Umständen nicht alle einflussreichen Informationen für die Angriffsmodellierung miteinbezogen sind. Die Ergebnisse der angewandten Methodik zur Angriffsmodellierung können von den tatsächlichen Möglichkeiten, die ein Angreifer besitzt abweichen. Im Kontext der Methodik wird versucht, ein menschliches Verhalten (Angreiferverhalten) mithilfe eines systematischen Verfahrens bzw. Algorithmus abzubilden. Das ist mit vielen Herausforderungen verbunden, da ein Mensch keine Maschine ist. Dennoch bietet die Methodik eine systematische Hilfestellung zur Angriffsmodellierung und die damit verbundenen Aktivitäten zur Gewährleistung von IT-Sicherheit.

8 Zusammenfassung und Ausblick

In der vorliegenden Masterarbeit wurde ein Konzept entwickelt und umgesetzt, das eine geeignete Grundlage für die strukturierte und formale Modellierung von Angriffen zur Verfügung stellt. Die Methodik basiert auf einem ganzheitlichen Angriffsmodell, mit dem Ziel alle möglichen Angriffe abzubilden. Dabei wird die Komplexität eines Angriffs über Hierarchien greifbar gemacht. Durch eine geeignete Kombination und Anpassung vorhandener Modellierungsansätze wird ein Angriff aus verschiedenen Sichtweisen betrachtet. Die prozessorientierte Modellierung wird mit einer technischen Sicht auf den Angriff verknüpft. Die grundlegende Datenbasis für die entwickelte Methodik, im Hinblick auf einen Angriff und die zugehörigen, spezifischen Elemente der Angriffsdomäne werden in zentralen Libraries strukturiert festgehalten, ergänzt und (wieder-)verwendet. Die tatsächliche Ausführung eines Angriffs wird durch eine geeignete Angriffssimulation repräsentiert. Während die Angriffssimulation auf einem Modell der tatsächlichen Umgebung basiert, gründet die Entwicklung und Planung eines Angriffs auf dem begrenzten Wissen des Angreifers, welches ihm aktuell zur Verfügung steht, im Hinblick auf das Target und dessen Umgebung. Im Rahmen der Masterarbeit liegt der Schwerpunkt auf dem Angriffsablauf zur Sammlung von Informationen über das Target, der zu Beginn eines Angriffs notwendig ist. Die Methodik präsentiert eine iterative Angriffsmodellierung. Mithilfe der Methodik werden nicht nur die Ausgangslage des Angreifers, seine damit verbundenen Angriffsmöglichkeiten und die Ergebnisse des Angriffs präsentiert, sondern auch das zielorientierte und systematische Vorgehen zur Planung und Umsetzung eines Angriffs.

In der Evaluierung konnte gezeigt werden, dass sich die Methodik an zwei sehr unterschiedlichen Domänen anwenden lässt. Sowohl im Bereich von Webanwendungen, als auch im tiefen Embedded-System Kontext genügt die entwickelte Methodik zur Angriffsmodellierung den Ansprüchen. Folglich konnte die Forschungsfrage beantwortet werden. Mithilfe der Methodik lassen sich aus dem generischen Angriffsmodell spezifische Angriffe modellieren. Die Ergebnisse der Methodik stellen eine qualitative Darstellung eines Angriffs dar, die von verschiedenen Zielgruppen bezüglich der Thematik Angriff als Grundlage verwendet werden können. In diesem Zusammenhang dient das Angriffsmodell z. B. als Basis für Risikoanalysen oder für die Ableitung von Sicherheitsmaßnahmen.

Mithilfe der Methodik zur Angriffsmodellierung ist die Entwicklung und Auswirkung eines Angriffs ersichtlich, sodass dieser besser verstanden werden kann. Wo sich die kritischen bzw. angreifbaren Stellen befinden, ist schnell und einfach ersichtlich, bzw.

nachvollziehbar. Folglich sind die Ergebnisse der Methodik für die allgemeine Handhabung, Kommunikation und Dokumentation von Angriffen verwendbar. Darüber hinaus können Security-Test unterstützt werden. Aus dem systematischen und nachvollziehbaren Angreifervorgehen lässt sich z. B. ein Roter Faden für Penetrationstests oder Testfälle und Metriken für Security-Tests ableiten. Zusammenfassend kann durch Anwendung der entwickelten Methodik zur Angriffsmodellierung, im Zusammenhang mit Security-Tests, die Durchgängigkeit und Nachverfolgbarkeit von IT-Sicherheit gefördert werden. Das Vertrauen in die softwarefähige Technologie steigt, was zu einer Verbesserung der Softwarequalität führt.

Im Ausblick soll die Methodik im Rahmen des aktuell laufenden Projekts „MASSiF“, das in Abschnitt 5.1 und in Kapitel 7 erwähnt wird, im Hinblick auf die nicht näher betrachteten Anforderungen „Simulierbar/Konsistent“ und „Verständlich“ weiterentwickelt und evaluiert werden. Dahingehend sind ergänzende Arbeiten notwendig, um die Methodik zu implementieren. Dazu zählt sowohl die Entwicklung und Implementierung des Angreifermodells, des Umgebungsmodells, des Umgebungs-Wissens-Modells, der notwendigen Bibliotheken für Zugangs- bzw. Angriffspunkte, der Angriffstechniken und Exploits, als auch die Spezifikationen der systematischen Vorgehensweisen im Kontext der entwickelten Methodik. Der modellbasierte Kern und das strukturierte Vorgehen der Methodik bietet eine ideale Grundlage für die Unterstützung durch Automatismen. In diesem Zusammenhang wäre nach einer geeigneten Syntax und Semantik für die Elemente der Angriffsmodellierung zu fragen. Ergänzende Arbeiten sind für die Entwicklung und Spezifikation weiterer Taktiken erforderlich, sodass neben „Explore“ auch die tatsächliche Durchführung einer Exploitation im Rahmen einer Taktik modelliert wird. Dahingehend ist z. B. zu klären, wie sich Denial-Of-Service-Angriffe abbilden lassen.

Ein möglicher Geschwindigkeitsvorteil soll untersucht werden. Es stellt sich die Frage, ob die Entwicklungsgeschwindigkeit im Kontext von DevOps erhöht werden kann, indem die entwickelte Methodik z. B. durch Automatismen unterstützt wird.[2, 27] Zudem gilt es zu klären, inwieweit die Möglichkeit einer automatisierten Code- bzw. Test-Generierung auf Grundlage der Exploits des Angriffsmodells realisierbar ist. Im Idealfall kann durch die Möglichkeit der Automatisierung eines parametrisierbaren Modells die Zeit von Beginn der Softwareentwicklung bis hin zur Freigabe reduziert werden.[2, 27] Im Hinblick auf Security-Tests ist neben einem Geschwindigkeitsvorteil, die Nachverfolgbarkeit und Nachweisbarkeit von IT-Security zu erwarten.[27]

Weiterer Forschungsbedarf ergibt sich aus der Idee, die Methodik zur Angriffsmodellierung mit den Aspekten der AI zu verknüpfen. Es wäre in diesem Zusammenhang lohnenswert zu untersuchen, ob eine AI im Zusammenspiel der Methodik entwickelt werden kann, sodass die Ableitung von zukünftigen Angriffsszenarien bzw. Angriffen möglich ist. Folglich könnten neue Angriffe mit der Unterstützung von AI identifiziert werden, sodass die Ergebnisse der entwickelten Methodik nicht mehr lediglich auf bekannte Angriffe beschränkt wäre.

Abbildungsverzeichnis

Tabellenverzeichnis

Literaturverzeichnis

[1] Sridhar Adepu und Aditya Mathur. „Generalized Attacker and Attack Models for Cyber Physical Systems“. In: 2016 IEEE 40th Annual Computer Software and Applications Conference (COMPSAC) (Atlanta, GA, USA, 10.–14. Juni 2016). Herausgegeben von Sorel Reisman. IEEE, 2016, Seiten 283–292. ISBN: 978-1-4673-8845-0. DOI: 10.1109/COMPSAC.2016.122. URL: https://ieeexplore.ieee.org/document/7552024 (besucht am 18. 02. 2019).

[2] Atlassian, Herausgeber. *DevOps. Breaking the Development-Operations barrier*. Sydney, Australien, 2019. URL: https://www.atlassian.com/legal/cloud-terms-of-service (besucht am 18. 10. 2019).

[3] Jim Bird und Jim Manico. *Attack Surface Analysis Cheat Sheet*. Herausgegeben von OWASP Foundation. Bel Air, MD 21014 USA, 2019. URL: https://github.com/OWASP/CheatSheetSeries/blob/master/cheatsheets/Attack_Surface_Analysis_Cheat_Sheet.md (besucht am 15. 07. 2019).

[4] John M. Borky und Thomas H. Bradley. *Effective Model-Based Systems Engineering*. Cham, Schweiz: Springer International Publishing, 2019. ISBN: 978-3-319-95668-8. DOI: 10.1007/978-3-319-95669-5. URL: https://doi-org.thi.idm.oclc.org/10.1007/978-3-319-95669-5 (besucht am 09. 08. 2019).

[5] Bundesamt für Sicherheit in der Informationstechnik - BSI, Herausgeber. *Register aktueller Cyber-Gefährdungen und -Angriffsformen*. Bonn, 2018. URL: https://www.allianz-fuer-cybersicherheit.de/ACS/DE/_/downloads/BSI-CS_026.pdf?__blob=publicationFile&v=5 (besucht am 18. 02. 2019).

[6] Jordi Cabot. *Clarifying concepts. MBE vs MDE vs MDD vs MDA*. Barcelona, 2018. URL: https://modeling-languages.com/clarifying-concepts-mbe-vs-mde-vs-mdd-vs-mda/ (besucht am 02. 05. 2019).

[7] CAPEC Content Team. *CAPEC-170. Web Application Fingerprinting*. Herausgegeben von The MITRE Corporation. 7515 Colshire Drive, McLean, VA 22102-7539, 2018. URL: https://capec.mitre.org/data/definitions/170.html (besucht am 25. 08. 2019).

[8] CAPEC Content Team. *CAPEC-66. SQL Injection*. Herausgegeben von The MITRE Corporation. 7515 Colshire Drive, McLean, VA 22102-7539, 2018. URL: https://capec.mitre.org/data/definitions/66.html (besucht am 23. 07. 2019).

[9] CAPEC Content Team. *CAPEC-126. Path Traversal*. Herausgegeben von The MITRE Corporation. 7515 Colshire Drive, McLean, VA 22102-7539, 2019. URL: https://capec.mitre.org/data/definitions/126.html (besucht am 21. 12. 2019).

[10] CAPEC Content Team. *CAPEC-72. URL Encoding*. Herausgegeben von The MITRE Corporation. 7515 Colshire Drive, McLean, VA 22102-7539, 2019. URL: https://capec.mitre.org/data/definitions/72.html (besucht am 17. 12. 2019).

[11] CAPEC Content Team. *CAPEC-97. Cryptanalysis*. Herausgegeben von The MITRE Corporation. 7515 Colshire Drive, McLean, VA 22102-7539, 2019. URL: https://capec.mitre.org/data/definitions/97.html (besucht am 01. 01. 2020).

[12] Edmund Clarke und Jeanette Wing. „Formal Methods. State of the art and further directions". In: *ACM Computing Surveys* 28.4 (1996), Seiten 626–643. URL: http://www.cs.cmu.edu/~emc/papers/Invited%20Journal%20Articles/state_art_future.pdf (besucht am 09. 04. 2019).

[13] Michael Collins. „Formal Methods". In: Dependable Embedded Systems (1998). Herausgegeben von Carnegie Mellon University, 18–849b. URL: https://users.ece.cmu.edu/~koopman/des_s99/formal_methods/ (besucht am 15. 10. 2019).

[14] Microsoft Corporation, Herausgeber. *Threat Modelling*. Redmond, WA 98052-6399 USA, 2019. URL: https://www.microsoft.com/en-us/securityengineering/sdl/threatmodeling (besucht am 12. 03. 2019).

[15] Crash Data Group Inc, Herausgeber. *CDR is an Acronym for Crah Data Retrival*. PO Box 892885 Temecula, CA 92589, 2020. URL: https://www.crashdatagroup.com/learn-more/ (besucht am 09. 01. 2020).

[16] Drd_. *How to Fingerprint Databases & Reconnaissance for a more successful attack*. Herausgegeben von WonderHowTo. Austin, TX 78750-1223, 2018. URL: https://null-byte.wonderhowto.com/how-to/sql-injection-101-fingerprint-databases-perform-general-reconnaissance-for-more-successful-attack-0184562/ (besucht am 17. 08. 2019).

[17] Andreas Drescher, Agnes Koschmider und Andreas Oberweis. *Modellierung und Analyse von Geschäftsprozessen. Grundlagen und Übungsaufgaben mit Lösungen*. De Gruyter Studium. München, Wien: De Gruyter Oldenbourg, 2017. Kapitel 2, 3. 1275 Seiten. ISBN: 978-3-11-049453-2. URL: https://doi.org/10.1515/9783110494532 (besucht am 15. 07. 2019).

[18] Claudia Eckert. *IT-Sicherheit. Konzepte - Verfahren - Protokolle*. Band 10., erweiterte und aktualisierte Auflage. De Gruyter Studium. München: De Gruyter Oldenbourg, 2018. Kapitel 1, 2, 3, 4, 5, 7, 14. ISBN: 978-3-11-056390-0. DOI: 10.1515/9783110563900. URL: https://doi-org.thi.idm.oclc.org/10.1515/9783110563900 (besucht am 21. 05. 2019).

[19] Ana M. Fernández-Sáez, Michel R. v. Chaudron und Marcela Genero. „An industrial case study on the use of UML in software maintenance and its perceived benefits and hurdles“. In: *Empirical Software Engineering* 23.6 (2018), Seiten 3281–3345. ISSN: 1382-3256. DOI: 10.1007/s10664-018-9599-4. URL: http://dx.doi.org.thi.idm.oclc.org/10.1007/s10664-018-9599-4 (besucht am 20. 09. 2019).

[20] Andrew Forward und Timothy C. Lethbridge. „Problems and opportunities for model-centric versus code-centric software development“. In: Proceedings of the 2008 international workshop on Models in software engineering (Leipzig, Deutschland, 10.–11. Mai 2008). Herausgegeben von Joanne Atlee. ACM Special Interest Group on Software Engineering. ACM, 2008, Seiten 27–32. ISBN: 9781605580258. DOI: 10.1145/1370731.1370738. URL: https://doi-org.thi.idm.oclc.org/10.1145/1370731.1370738 (besucht am 10. 07. 2019).

[21] OWASP Foundation, Herausgeber. *OWASP Risk Rating Methodology*. Bel Air, MD 21014 USA, 2018. URL: https://www.owasp.org/index.php?title=OWASP_Risk_Rating_Methodology&oldid=252633 (besucht am 15. 02. 2019).

[22] Martin Glinz. *Abstraktion*. Informatik 2: Modellierung. Herausgegeben von Universität Zürich Institut für Informatik. Zürich, 2005. URL: https://files.ifi.uzh.ch/rerg/amadeus/teaching/courses/infII_ss05/inf_II_kapitel_13.pdf (besucht am 15. 10. 2019).

[23] Object Management Group, Herausgeber. *Business process model and notation. BPMN*. Version 2.0. Needham, MA 02494, USA, 2011. URL: https://www.omg.org/spec/BPMN/2.0/PDF (besucht am 13. 08. 2019).

[24] Michael Howard und Steve Lipner. *The Security Development Lifecycle*. Best practices, Secure Software Development Series. Redmond, Washington: Microsoft Press, 2006. Kapitel 1, 4, 7, 9, 18. ISBN: 0-7356-2214-0.

[25] Ideal Integrations, Inc., Herausgeber. *The Cyber Kill Chain Model is Obsolete*. 800 Regis Ave. Pittsburgh, PA 15236, 27. Aug. 2019. URL: https://www.idealintegrations.net/the-cyber-kill-chain-model-is-obsolete/ (besucht am 20. 12. 2019).

[26] Chris Johnson u. a. *Guide to Cyber Threat Information Sharing*. Herausgegeben von National Institute of Standards und Technology. NIST Special Publication 800-150. Gaithersburg, MD 20899, Okt. 2016. Kapitel 2. DOI: 10.6028/NIST.SP.800-150. URL: http://dx.doi.org/10.6028/NIST.SP.800-150 (besucht am 23. 07. 2019).

[27] Jan Jürjens. *Secure Systems Development with UML*. Springer Berlin Heidelberg, 2005. Kapitel Vorwort, 1, 2, 3, 4, 6, 7, 8, 9. ISBN: 3-540-00701-6. URL: https://link.springer.com/content/pdf/10.1007%2Fb137706.pdf (besucht am 09. 04. 2019).

[28] Kerem Kaynar. „A taxonomy for attack graph generation and usage in network security“. In: *Journal of Information Security and Applications* 29 (2016). Herausgegeben von Elsevier Ltd, Seiten 2214–2126. ISSN: 2214-2126. DOI: 10.1016/j.jisa.2016.02.001. URL: http://www.sciencedirect.com/science/article/pii/S2214212616300011 (besucht am 27. 08. 2019).

[29] Walter Kriha. *Sichere Systeme. Konzepte, Architekturen und Frameworks*. Herausgegeben von Roland Schmitz. Xpert.press. Berlin: Springer, 2009. Kapitel 1, 2, 3, 10, 14. 639 Seiten. ISBN: 978-3-540-78959-8. URL: https://doi.org/10.1007/978-3-540-78959-8.

[30] Michael Kroker. *Die 7 wichtigsten Trends in der IT-Sicherheit im Jahr 2019*. Herausgegeben von Handelsblatt Media Group GmbH & Co. KG. Düsseldorf, 15. Jan. 2019. URL: https://blog.wiwo.de/look-at-it/2019/01/15/die-7-wichtigsten-trends-in-der-it-sicherheit-im-jahr-2019/ (besucht am 19. 05. 2019).

[31] Peter Loshin. *Challanges and benefits of using the Mitre ATT&CK framework*. Herausgegeben von TechTarget, Inc. Newton, MA 02466 USA, 15. Apr. 2019. URL: https://searchsecurity.techtarget.com/feature/Challenges-and-benefits-of-using-the-Mitre-ATTCK-framework (besucht am 19. 09. 2019).

[32] Netsparker Ltd, Herausgeber. *SQL Injection Cheat Sheet*. United Kingdom, 21. Okt. 2015. URL: https://www.netsparker.com/blog/web-security/sql-injection-cheat-sheet/ (besucht am 28. 08. 2019).

[33] Stefan Marr. *Modellkonsistenz*. Potsdam, 2019. URL: https://stefan-marr.de/pages/modellkonsistenz/ (besucht am 05. 11. 2019).

[34] Microsoft Corporation, Herausgeber. *What are the Microsoft SDL practices?* Redmond, WA 98052-6399 USA, 2019. URL: https://www.microsoft.com/en-us/securityengineering/sdl/practices (besucht am 12. 03. 2019).

[35] Andrew Moore, Robert Ellison und Rick Linger. *Attack Modeling for Information Security and Survivability*. CMU/SEI-2001-TN-001. Pittsburgh, PA 15213-2612, 2001. URL: https://www.researchgate.net/publication/2371562_Attack_Modeling_for_Information_Security_and_Survivability (besucht am 31. 05. 2019).

[36] Object Management Group, Herausgeber. *Unified Modeling Language*. Version 2.5.1. Needham, MA 02494, USA, 2017. Kapitel 15, 16. URL: https://www.omg.org/spec/UML/2.5.1/PDF (besucht am 22. 07. 2019).

[37] Samir Ouchani und Gabriele Lenzini. „Attacks Generation by Detecting Attack Surfaces". In: *Procedia Computer Science* 32 (2014), Seiten 529–536. ISSN: 1877-0509. DOI: 10.1016/j.procs.2014.05.457. URL: http://dx.doi.org.thi.idm.oclc.org/10.1016/j.procs.2014.05.457 (besucht am 27. 08. 2019).

[38] Martin Pollakowski. *Grundkurs MySQL und PHP. So entwickeln Sie Datenbanken mit Open-Source-Software*. Braunschweig, Wiesbaden: Friedr. Vieweg & Sohn Verlagsgesellschaft mbH, 2003. Kapitel 1, 3, 8, 9, 10, 11. ISBN: 978-3-322-93958-6. DOI: 10.1007/978-3-322-93958-6. URL: https://doi-org.thi.idm.oclc.org/10.1007/978-3-322-93958-6 (besucht am 20. 08. 2019).

[39] Cambridge University Press, Herausgeber. *Attack*. Meaning in the Cambridge English Dictionary. Cambridge CB2 8BS United Kingdom, 2019. URL: https://dictionary.cambridge.org/dictionary/english/attack (besucht am 28. 02. 2019).

[40] Bernd-Thomas Ramb und Jean-Paul Thommen. *Verifikation*. Herausgegeben von Springer Fachmedien Wiesbaden GmbH. 19. Feb. 2018. URL: https://wirtschaftslexikon.gabler.de/definition/verifikation-50409/version-273628 (besucht am 15. 10. 2019).

[41] Wolfgang Reinöhl. *Militärische Strategie und Taktik*. Berlin, 12. Okt. 2015. URL: https://www.frag-machiavelli.de/militaer-strategie-taktik/ (besucht am 06. 09. 2019).

[42] Karl Riedling. *Datenbank-basierte Webserver. Struktur eines Datenbank-basierten Webservers*. Herausgegeben von Institut für Sensor- und Aktuatorsysteme Technische Universität Wien. Wien, 3. Nov. 2018. URL: http://karl.riedling.at/webserver/StrukturWeb.pdf (besucht am 21. 08. 2019).

[43] Margaret Rouse. *attack vector*. Definition. Herausgegeben von TechTarget, Inc. Newton, MA 02466 USA, 2012. URL: https://searchsecurity.techtarget.com/definition/attack-vector (besucht am 03. 06. 2019).

[44] Margaret Rouse. *Vulnerability assessment (vulnerability analysis)*. Herausgegeben von TechTarget, Inc. Newton, MA 02466 USA, 2018. URL: https://searchsecurity.techtarget.com/definition/vulnerability-assessment-vulnerability-analysis (besucht am 20. 11. 2019).

[45] Margaret Rouse. *Cyberangreifer*. Definition. Herausgegeben von TechTarget, Inc. Newton, MA 02466 USA, 2019. URL: https://whatis.techtarget.com/de/definition/Cyberangreifer (besucht am 04. 03. 2019).

[46] Niklaus Schild. *Sichere Softwareentwicklung nach dem Security by Desing-Prinzip*. Herausgegeben von Heise Medien GmbH & Co. KG. Hannover, 19. Aug. 2009. URL: https://www.heise.de/developer/artikel/Sichere-Softwareentwicklung-nach-dem-Security-by-Design-Prinzip-403663.html?seite=all (besucht am 20. 05. 2019).

[47] Bruce Schneier. „Attack Trees. Modeling security threats“. In: *Dr. Dobb's Journal* 24.12 (Dez. 1999), Seiten 21–29. URL: https://www.schneier.com/academic/archives/1999/12/attack_trees.html (besucht am 23. 07. 2019).

[48] Oliver Schonschek und Peter Schmitz. *Cyber Kill Chain. Grundlagen, Anwendung und Entwicklung*. Herausgegeben von Vogel IT-Medien GmbH. Augsburg, 2017. URL: https://www.security-insider.de/cyber-kill-chain-grundlagen-anwendung-und-entwicklung-a-608017/ (besucht am 21. 08. 2019).

[49] Robert W. Shirey. *Internet Security Glossary*. Herausgegeben von The IETF Trust. Version 2. Reston, VA 20190, 2007. URL: https://tools.ietf.org/html/rfc4949 (besucht am 29. 05. 2019).

[50] Bundesamt für Sicherheit in der Informationstechnik - BSI, Herausgeber. *Studie Durchführungskonzept für Penetrationstests. Sicherheit*. Bonn, 2003. URL: https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/Studien/Penetrationstest/penetrationstest.pdf;jsessionid=16EE4518EEEEC58F1479AEDE48851E93.2_cid369?__blob=publicationFile&v=3 (besucht am 23. 05. 2019).

[51] Bundesamt für Sicherheit in der Informationstechnik - BSI, Herausgeber. *BSI-Standard 100-2. IT-Grundschutz-Vorgehensweise*. Version 2.0. Bonn, 2008. Kapitel 2, 3, 4, 5, 6. URL: https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/ITGrundschutzstandards/BSI-Standard_1002.pdf?__blob=publicationFile&v=3 (besucht am 21. 10. 2019).

[52] Bundesamt für Sicherheit in der Informationstechnik - BSI, Herausgeber. *Drahtlose Kommunikationssysteme und ihre Sicherheitsaspekte*. Version 3.0. Bonn, 2009. Kapitel A, B, E. 337 Seiten. URL: https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/Broschueren/Drahtlose-Komsysteme.pdf;jsessionid=EBCC663ADF429554FC5B7C9A725F0B34.2_cid351?__blob=publicationFile&v=2 (besucht am 19. 02. 2019).

[53] Bundesamt für Sicherheit in der Informationstechnik - BSI, Herausgeber. *IT-Grundschutz-Kataloge*. 15. Ergänzungslieferung. Bonn, 2016. Kapitel Vorwort, 1, 4, B 1.16, B 2, B 3, G 2.107, M 2.359, M 2.517, M 3.69, M 5.71. URL: https://download.gsb.bund.de/BSI/ITGSK/IT-Grundschutz-Kataloge_2016_EL15_DE.pdf (besucht am 03. 04. 2019).

[54] Bundesamt für Sicherheit in der Informationstechnik - BSI, Herausgeber. *Schutz Kritischer Infrastrukturen. Durch IT-Sicherheitsgesetz und UP KRITIS*. Bonn, März 2017. URL: https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/Broschueren/Schutz-Kritischer-Infrastrukturen-ITSig-u-UP-KRITIS.pdf;jsessionid=024A1FD2FFC9559B6510A80CB23B906F.2_cid360?__blob=publicationFile&v=7 (besucht am 25. 03. 2019).

[55] Bundesamt für Sicherheit in der Informationstechnik - BSI, Herausgeber. *Basismaßnahmen der Cyber-Sicherheit*. Empfehlung: IT im Unternehmen. Version 2.0. Bonn, 2018. URL: https://www.allianz-fuer-cybersicherheit.de/ACS/DE/_/downloads/BSI-CS_006.pdf?__blob=publicationFile&v=4 (besucht am 24. 05. 2019).

[56] Bundesamt für Sicherheit in der Informationstechnik - BSI, Herausgeber. *Glossar der Cyber-Sicherheit*. Bonn, 2019. URL: https://www.bsi.bund.de/DE/Themen/Cyber-Sicherheit/Empfehlungen/cyberglossar/Functions/glossar.html?cms_lv2=9817272 (besucht am 23. 09. 2019).

[57] Markus Siepermann. *Agile Softwareentwicklung*. Herausgegeben von Springer Fachmedien Wiesbaden GmbH. 19. Feb. 2018. URL: https://wirtschaftslexikon.gabler.de/definition/agile-softwareentwicklung-53460/version-276549 (besucht am 18. 10. 2019).

[58] Helmut Siller. *Exploit. Definition*. Herausgegeben von Springer-Verlag GmbH. Heidelberg, 2018. URL: https://wirtschaftslexikon.gabler.de/definition/exploit-53419/version-276511 (besucht am 03. 06. 2019).

[59] Craig Smith. *The Car Hacker's Handbook. a guide for the penetration tester*. 245 8th Street, San Francisco, CA 94103: No Starch Press, Inc., 2016. Kapitel 4. ISBN: 9781593277031. URL: https://docs.alexomar.com/biblioteca/thecarhackershandbook.pdf (besucht am 11. 12. 2019).

[60] Jeffrey Smith und Michael Figueroa. „Reduced realistic attack plan surface for identification of prioritized attack goals“. In: 2013 IEEE International Conference on Technologies for Homeland Security (HST) (Waltham, MA, USA, 12.–14. Nov. 2013). Herausgegeben von IEEE International Conference on Technologies for Homeland Security. Band 14024409. IEEE, 2013, Seiten 716–721. ISBN: 978-1-4799-1535-4. DOI: 10.1109/THS.2013.6699092. URL: https://ieeexplore-ieee-org.thi.idm.oclc.org/stamp/stamp.jsp?tp=&arnumber=6699092 (besucht am 20. 02. 2019).

[61] Blake E. Strom u. a. *MITRE ATT&CK. Design and Philosophy*. Herausgegeben von The MITRE Corporation. 7515 Colshire Drive, McLean, VA 22102-7539, 2018. Kapitel 1, 2, 3, 4. URL: https://www.mitre.org/sites/default/files/publications/pr-18-0944-11-mitre-attack-design-and-philosophy.pdf (besucht am 19. 09. 2019).

[62] Guanyu Su, Fang Wang und Qi Li. „Research on SQL Injection Vulnerability Attack model“. In: 2018 5th IEEE International Conference on Cloud Computing and Intelligence Systems (CCIS) (Nanjing, China, 23.–25. Nov. 2018). Herausgegeben von IEEE. 2018, Seiten 217–221. DOI: 10.1109/CCIS.2018.8691148. URL: https://ieeexplore-ieee-org.thi.idm.oclc.org/document/8691148 (besucht am 22. 07. 2019).

[63] Cho Sungyoung u. a. „Cyber Kill Chain based Threat Taxonomy and its Application on Cyber Common Operational Picture“. In: 2018 International Conference On Cyber Situational Awareness, Data Analytics And Assessment (Cyber SA) (Glasgow, UK, 11.–12. Juni 2018). Herausgegeben von IEEE. 2018, Seiten 1–8. DOI: 10.1109/CyberSA.2018.8551383. URL: https://ieeexplore-ieee-org.thi.idm.oclc.org/stamp/stamp.jsp?tp=&arnumber=8551383 (besucht am 04. 06. 2019).

[64] Aaron Tan. *Cyber kill chain is outdated, says Carbon Black. The chief cyber security officer of Carbon Black calls for a new cognitive paradigm to fend off cyber adversaries that are now attacking in cycles*. Herausgegeben von TechTarget, Inc. Newton, MA 02466 USA, 29. Juli 2019. URL: https://www.computerweekly.com/news/252467482/Cyber-kill-chain-is-outdated-says-Carbon-Black (besucht am 18. 09. 2019).

[65] CWE Team. *Schema Documentation*. Herausgegeben von The MITRE Corporation. Version 6.1. 7515 Colshire Drive, McLean, VA 22102-7539, 23. Sep. 2019. URL: https://cwe.mitre.org/documents/schema/index.html (besucht am 27. 09. 2019).

[66] Erik Tews und Christian Schlehuber. „Quantitative Ansätze zur IT-Risikoanalyse“. In: Regular Research Papers (Wien, Österreich, 19.–21. März 2014). Herausgegeben von Stefan Katzenbeisser, Volkmar Lotz und Edgar Weippl. Band P-251.

Lecture Notes in Informatics (LNI) - Proceedings. Bonn, 2014, Seiten 293–303. ISBN: 978-3-88579-622-0. URL: https://dl.gi.de/bitstream/handle/20.500.12116/20051/293.pdf?sequence=1&isAllowed=y (besucht am 16. 10. 2019).

[67] The MITRE Corporation, Herausgeber. *About CAPEC*. 7515 Colshire Drive, McLean, VA 22102-7539, 4. Apr. 2019. URL: https://capec.mitre.org/ (besucht am 01. 10. 2019).

[68] The MITRE Corporation, Herausgeber. *About CVE Entries*. 7515 Colshire Drive, McLean, VA 22102-7539, 5. Feb. 2019. URL: https://cve.mitre.org/cve/identifiers/index.html (besucht am 26. 09. 2019).

[69] The MITRE Corporation, Herausgeber. *About CWE*. 7515 Colshire Drive, McLean, VA 22102-7539, 29. Apr. 2019. URL: https://cwe.mitre.org/about/index.html (besucht am 26. 09. 2019).

[70] The MITRE Corporation, Herausgeber. *APT18*. 7515 Colshire Drive, McLean, VA 22102-7539, 2019. URL: https://attack.mitre.org/groups/G0026/ (besucht am 23. 12. 2019).

[71] The MITRE Corporation, Herausgeber. *CAPEC Glossary*. 7515 Colshire Drive, McLean, VA 22102-7539, 2019. URL: https://capec.mitre.org/about/glossary.html (besucht am 03. 06. 2019).

[72] The MITRE Corporation, Herausgeber. *CWE-89. Improper Neutralization of Special Elements used in an SQL Command*. SQL Injection. 7515 Colshire Drive, McLean, VA 22102-7539, 19. Sep. 2019. URL: https://cwe.mitre.org/data/definitions/89.html (besucht am 18. 10. 2019).

[73] The MITRE Corporation, Herausgeber. *Privilege Escalation*. 7515 Colshire Drive, McLean, VA 22102-7539, 2019. URL: https://attack.mitre.org/tactics/TA0004/ (besucht am 13. 11. 2019).

[74] The MITRE Corporation, Herausgeber. *The CAPEC Schema. Core Definition*. Version 3.1.0. 7515 Colshire Drive, McLean, VA 22102-7539, 4. Apr. 2019. URL: http://capec.mitre.org/data/xsd/ap_schema_v3.1.xsd (besucht am 27. 08. 2019).

[75] The MITRE Corporation, Herausgeber. *Threat-based Defense*. Cybersecurity. 7515 Colshire Drive, McLean, VA 22102-7539, 2019. URL: https://www.mitre.org/capabilities/cybersecurity/threat-based-defense (besucht am 04. 06. 2019).

[76] The MITRE Corporation, Herausgeber. *Vulnerability Details. CVE-2019-14313*. 7515 Colshire Drive, McLean, VA 22102-7539, 2019. URL: https://www.cvedetails.com/cve/CVE-2019-14313/ (besucht am 06. 09. 2019).

[77] The OWASP Foundation, Herausgeber. *Category:Attack*. Bel Air, MD 21014 USA, 2016. URL: https://www.owasp.org/index.php?title=Category:Attack&oldid=217720 (besucht am 09. 07. 2019).

[78] The OWASP Foundation, Herausgeber. *OWASP Top 10 - 2017. The ten most critical web application security risks*. Bel Air, MD 21014 USA, 2017. URL: https://www.owasp.org/images/7/72/OWASP_Top_10-2017_%28en%29.pdf.pdf (besucht am 18. 12. 2019).

[79] Tutorialspoint, Herausgeber. *Footprinting*. Ethical Hacking. Kavuri Hills, Madhapur, Hyderabad, Telangana, INDIA-500081, 2019. URL: https://www.tutorialspoint.com/ethical_hacking/ethical_hacking_footprinting.htm (besucht am 22. 08. 2019).

[80] Tutorialspoint, Herausgeber. *Sniffing*. Ethical Hacking. Kavuri Hills, Madhapur, Hyderabad, Telangana, INDIA-500081, 2019. URL: https://www.tutorialspoint.com/ethical_hacking/ethical_hacking_sniffing.htm (besucht am 22. 08. 2019).

[81] U.S. Government Publishing Office. „Part 563 - Event Data Recorders“. In: *Code of Federal Regulations* 6 (1. Okt. 2018). URL: https://www.govinfo.gov/content/pkg/CFR-2018-title49-vol6/xml/CFR-2018-title49-vol6-part563.xml (besucht am 13. 12. 2019).

[82] Robby Winchester. *What's in a name? TTPs in Info Sec*. Herausgegeben von Inc. Specter Ops. McLean, VA. 22102, 2017. URL: https://posts.specterops.io/whats-in-a-name-ttps-in-info-sec-14f24480ddcc (besucht am 23. 07. 2019).

[83] Stefanie Winter. *Quantitative vs. Qualitative Methoden*. 15. Mai 2000. URL: http://nosnos.synology.me/MethodenlisteUniKarlsruhe/imihome.imi.uni-karlsruhe.de/nquantitative_vs_qualitative_methoden_b.html (besucht am 17. 10. 2019).

A Weiterführende Ergebnisse der Analysephase

Tabelle A.1: Charakteristik eines Angriffs

Nr.	Eigenschaft	Beispiele
1	Aktion	Eingabe eines einfachen Anführungszeichens in ein Benutzereingabefeld
2	Reihenfolge	Aktion A ist Voraussetzung für Aktion B
3	Angriffsiteration	Angriffsiteration A hat zum Ergebnis, dass eine SQLI-Verwundbarkeit entdeckt wurde; Angriffsiteration B war erfolglos
4	Strategisches Ziel	Das Image von Unternehmen A beschädigen
5	Taktisches Ziel	Sammlung von Informationen über das Target; Ausnutzung identifizierter Vulnerabilities
6	Angriffstechnik	Social Engineering; SQLI; Passives Monitoring
7	Exploit	„MySQL“-Exploit zur Identifikation von Datenbankfeldern; „MySQL“-Exploit zur Identifikation von Tabellennamen
8	Zugangspunkt	Website/Source-Code des Unternehmens A; URL; Programmcode; Cookie
9	Angriffspunkt	URL; Programmcode; Cookie; Benutzereingabefeld;
10	Angriffsoberfläche	Menge der Angriffspunkten
11	Verwundbarkeit	SQLI-Vulnerability; XSS-Vulnerability

Tabelle A.1: Charakteristik eines Angriffs (Fortsetzung)

Nr.	Eigenschaft	Beispiele
12	Angreiferwissen	Source-Code der Website A enthält JavaScript
13	Angreifer	Cyber-Krimineller, der das Ansehen der Firma A schädigen will (Bevorzugter Angriffsbereich: Phishing via E-Mail, Scope: Internet, Kompetenz: durchschnittlich, Ressourcen: Tarnungsressourcen)
14	Umgebung	Relational Database Management System „MySQL“ der Version 5-6-45; Apache Webserver der Version 2-0-65 mit PHP Interpreter als Plug-In; Website A besteht aus statischen und dynamischen HTML Seiten; Source-Code der Website A enthält JavaScript