



Fakultät für Ingenieurwissenschaften

## **Master-Thesis**

Evaluation von IT-Systemen zur Automatisierung der  
standardkonformen Reaktion bei IT-Sicherheitsvorfällen

*Evaluation of IT-Systems Concerning the Automation of  
Standard-Based IT-Security Incident Response*

Abschlussarbeit zur Erlangung des Grades eines

## **Master of Engineering**

der Hochschule Wismar

eingereicht von:

Christoph Lobmeyer

[REDACTED]

Studiengang IT-Sicherheit und Forensik

Matrikelnummer:

[REDACTED]

Erstgutachterin:

Prof. Dr.-Ing. A. Raab-Düsterhöft

Zweitgutachter:

Prof. Dr.-Ing. habil. A. Ahrens

[REDACTED] 28. Mai 2020

# Abstract

Die vorliegende Masterthesis evaluiert IT-Systeme, die zur Automatisierung des IT-Sicherheitsvorfallsprozess entwickelt wurden, hinsichtlich der Erfüllung von Anforderungen, wie sie sich aus Standards zur Beschreibung des IT-Sicherheitsvorfallsprozesses ergeben.

Dazu beschreibt diese Arbeit zunächst die Anforderungen, entwickelt anschließend Testszenarien bzw. Testfälle und wendet diese dann auf zwei exemplarische IT-Systeme an.

Das Ergebnis der Arbeit ist, dass obwohl nicht alle Anforderungen erfüllt werden, diese Systeme dennoch eine Unterstützung in der Vorfallsreaktion darstellen können.

The main aim of this master thesis is to evaluate IT systems, designed to automate the IT-Security Incident Response Process. This evaluation is done regarding their capability to fulfill requirements described in well-know standards.

First these requirements are described and explained, then test scenarios and cases are developed. Finally these tests are performed using two typical IT-systems serving as examples.

The result of this thesis is, that although the systems do not fulfill each requirement, they are capable to support the Incident Response Process.

# Vorwort

*Mit der hier vorliegenden Arbeit endet mein Masterstudium der IT-Sicherheit und Forensik. Ich möchte mich bei allen Personen bedanken, die mich bei diesem Projekt begleitet und unterstützt haben.*

*Zuallererst möchte ich mich bei meiner Frau Lisa bedanken, die mir mit vielen frischen Ideen, Zeit für Diskussionen und viel Liebe zur Seite steht. Ich liebe dich.*

*Dann bedanke ich mich bei meiner Familie, die mich sowohl in diesem Studium, als auch sonst begleitet und ein offenes Ohr hat.*

*Abschließend möchte ich mich noch bei meiner Erstgutachterin Frau Prof. Dr.-Ing. Antje Raab-Düsterhöft und meinem Zweitgutachter Herrn Prof. Dr.-Ing. habil. Andreas Ahrens bedanken, die es mir ermöglicht haben, die Arbeit über dieses Thema zu verfassen, und die mich mit sehr guten Hinweisen unterstützt haben.*

*Abschließend noch eine Anmerkung: Diese Arbeit verwendet durchgehend die weibliche Form. Die Aussagen beziehen sich dennoch auf alle Geschlechter.*

# Inhaltsverzeichnis

<b>1. Einleitung</b>	<b>10</b>
1.1. Motivation . . . . .	10
1.2. Abgrenzung . . . . .	12
<b>2. Theoretische Grundlagen</b>	<b>13</b>
2.1. Umgang mit IT-Sicherheitsvorfällen . . . . .	13
2.1.1. IT-Sicherheitsvorfallsreaktionsprozess . . . . .	16
2.1.2. Beteiligte Parteien bei der Vorfallsreaktion . . . . .	25
2.1.3. Technische Schnittstellen im Reaktionsprozess . . . . .	28
2.2. Anforderungsanalyse . . . . .	29
2.3. Softwaretests . . . . .	31
2.3.1. Testbeschreibung . . . . .	32
2.3.2. Testdurchführung . . . . .	34
<b>3. Anforderungen</b>	<b>35</b>
3.1. Vorgehensweise . . . . .	35
3.2. Rollen . . . . .	36
3.2.1. Vorfallsteammitglied . . . . .	36
3.2.2. Vorfallsteamleiterin . . . . .	36
3.2.3. Detektionssystem . . . . .	36
3.2.4. Threat-Sharing-Plattform . . . . .	37
3.3. Anforderungen . . . . .	37
<b>4. Testszenarien und Testfälle</b>	<b>63</b>
4.1. Testszenarien . . . . .	63
4.1.1. Benutzerverwaltung und -anmeldung . . . . .	63

4.1.2.	Vorfallserstellung . . . . .	64
4.1.3.	Vorfallsanzeige und -editierung . . . . .	65
4.1.4.	Vorfallsanalyse . . . . .	66
4.1.5.	Vorfallsreaktion . . . . .	66
4.1.6.	Threat-Sharing . . . . .	67
4.1.7.	Dokumentation . . . . .	68
4.1.8.	Kommunikation . . . . .	68
4.1.9.	Nachbereitung . . . . .	69
4.2.	Testfälle . . . . .	69
4.3.	Abhängigkeiten der Testfälle . . . . .	81
<b>5.</b>	<b>Auswahl der IT-Systeme</b>	<b>82</b>
5.1.	TheHive/Cortex . . . . .	82
5.2.	Splunk Phantom . . . . .	84
<b>6.</b>	<b>Entwicklung einer Laborumgebung</b>	<b>86</b>
6.1.	Aufbau der Laborumgebung . . . . .	86
6.2.	Integration der Untersuchungsgegenstände . . . . .	89
6.2.1.	Integration von TheHive/Cortex . . . . .	89
6.2.2.	Integration von Splunk Phantom . . . . .	90
6.3.	Vorbereitung der Tests . . . . .	90
6.3.1.	Vorbereitung von TheHive/Cortex . . . . .	91
6.3.2.	Vorbereitung von Splunk Phantom . . . . .	92
<b>7.</b>	<b>Testdurchführung</b>	<b>94</b>
7.1.	TheHive/Cortex . . . . .	95
7.2.	Splunk Phantom . . . . .	105
7.3.	Untersuchung der Ergebnisse . . . . .	115
<b>8.</b>	<b>Fazit und Ausblick</b>	<b>119</b>
8.1.	Zusammenfassung . . . . .	119
8.2.	Diskussion des Untersuchungsansatzes . . . . .	120
8.3.	Ausblick . . . . .	121

<b>9. Literaturverzeichnis</b>	<b>123</b>
<b>10. Ehrenwörtliche Erklärung</b>	<b>128</b>
<b>11. Thesen</b>	<b>129</b>
<b>A. Anhang</b>	<b>130</b>

# Abbildungsverzeichnis

2.1.	IT-Sicherheitsvorfallsprozess nach [6]	17
2.2.	Möglichkeiten der Detektion: netzwerkbasiert (grün bzw. rot), system- basiert (blau)	18
2.3.	Abgrenzung von Observables zu Indicators of Compromise	20
2.4.	Übersicht der CSIRT-Services (Grafik aus [7, S. 24])	26
2.5.	Softwareentwicklung im V-Modell XT (nach [26])	33
3.1.	Funktionen innerhalb des Vorfallsreaktionsprozess (nach [7, S. 67])	37
4.1.	Abhängigkeiten der Testfälle (TF)	81
5.1.	Datenfluss in TheHive/Cortex [31]	83
5.2.	Datenfluss in Splunk Phantom [32]	85
6.1.	Aufbau der Laborumgebung	87
6.2.	Konfiguration des VMware Subnets	88
6.3.	Konfiguration von Splunk Phantom	91
6.4.	Konfiguration der Benutzer in TheHive	92
6.5.	Konfiguration der Tenants in Splunk Phantom	93
7.1.	Testerfüllung nach Anwendungen	115
7.2.	Testerfüllung nach Testszenarien	118
A.1.	Erstellter Vorfall in TheHive	135
A.2.	Erstellung eines Case Templates in TheHive	136
A.3.	Abarbeiten von Tasks in TheHive	136
A.4.	Abarbeiten von Tasks in TheHive	136
A.5.	Abarbeiten von Tasks in TheHive	137

A.6. Teilen eines Vorfalls in TheHive . . . . .	137
A.7. Export eines Events zu MISP in TheHive . . . . .	137
A.8. Export von TheHive in MISP . . . . .	138
A.9. Konfiguration von Splunk Phantom . . . . .	139
A.10. Dashboard von Splunk Phantom . . . . .	140
A.11. Erstellter Vorfall in Splunk Phantom . . . . .	140
A.12. Ansicht eines Playbooks in Splunk Phantom . . . . .	141
A.13. Ansicht eines Workbooks in Splunk Phantom . . . . .	141
A.14. Export von Splunk Phantom in MISP . . . . .	142



# Tabellenverzeichnis

3.1. Übersicht über die Userstories . . . . .	39
4.1. Testszenario Benutzerverwaltung und -anmeldung: Abgedeckte Anforderungen . . . . .	64
4.2. Testszenario Vorfallserstellung: Abgedeckte Anforderungen . . . . .	64
4.3. Testszenario Vorfallsanzeige und -editierung: Abgedeckte Anforderungen . . . . .	65
4.4. Testszenario Vorfallsanalyse: Abgedeckte Anforderungen . . . . .	66
4.5. Testszenario Vorfallsreaktion: Abgedeckte Anforderungen . . . . .	67
4.6. Testszenario Threat-Sharing: Abgedeckte Anforderungen . . . . .	67
4.7. Testszenario Dokumentation: Abgedeckte Anforderungen . . . . .	68
4.8. Testszenario Kommunikation: Abgedeckte Anforderungen . . . . .	69
4.9. Testszenario Nachbereitung: Abgedeckte Anforderungen . . . . .	69

# 1. Einleitung

In den vergangenen Jahren sind immer wieder weitreichende IT-Sicherheitsvorfälle wie zum Beispiel NotPetya und WannaCry im Jahr 2017, der Bundestagshack im Jahr 2015 und der Bundeshack im Jahr 2018 öffentlich bekannt geworden. Auch in Studien lässt sich die steigende Zahl von IT-Sicherheitsvorfällen belegen [1]. Dem gegenüber steht ein Mangel an Fachkräften in der IT allgemein und im Bereich der IT-Sicherheit im Speziellen [2]. Da die Bearbeitung von IT-Sicherheitsvorfällen wiederum eine Spezialisierung im Bereich der IT-Sicherheit ist, ist davon auszugehen, dass der Fachkräftemangel dort noch gravierender ist.

## 1.1. Motivation

Steigende Vernetzung (z.B. von industriellen Steuerungsanlagen) und steigenden Möglichkeiten der Angreiferinnen (z.B. „APT for hire“ [3]) führen zu einer erhöhten Gefahr für IT-Systeme. Somit wird es zukünftig eine elementare Aufgabe in der Informatik sein, IT-Systeme nicht nur möglichst sicher zu gestalten, sondern auch darauf vorbereitet zu sein, dass diese Sicherheit nicht vollständig erreicht werden kann. Das heißt, dass neben der Informationssicherheitsdomäne *Prävention* auch die Domänen *Detektion* und *Reaktion* berücksichtigt werden müssen. Nachdem IT-Sicherheit mittlerweile nicht mehr nur von Konzernen als wichtiges Thema erkannt wird, sondern auch kleine und mittelständische Unternehmen ihre Bemühungen verstärken [4], müssen auch diese Organisationen auf IT-Sicherheitsvorfälle reagieren, obwohl sie möglicherweise nicht über das Fachpersonal und die finanziellen Ressourcen verfügen.

Es stellt sich also die Frage, wie die steigende Anzahl der IT-Sicherheitsvorfälle mit der auch zukünftig absehbar knappen Ressource Personal bearbeitet werden kann.

Ein möglicher Lösungsansatz ist die Automatisierung oder Teilautomatisierung des IT-Sicherheitsvorfallsreaktionsprozesses durch die Unterstützung von darauf spezialisierten IT-Systemen [5].

Zur technischen Unterstützung dieses Prozesses gibt es bereits verschiedene IT-Produkte, die entweder kommerziell von Anbieterinnen von IT-Sicherheitslösungen oder als Open Source durch die internationale Community der Vorfallsbearbeiterinnen zur Verfügung gestellt werden. Eine Beschreibung, welche Anforderungen mit diesen IT-Systemen erfüllt werden, ist jedoch nicht veröffentlicht. In welchem Maße diese aus der Praxis entstandenen Werkzeuge dennoch in der Lage sind, bei der Reaktion auf einen IT-Sicherheitsvorfall zu unterstützen, lässt sich daher zur Zeit nicht standardisiert bzw. vergleichbar evaluieren.

Einige dieser Systeme konzentrieren sich auf die Analyse von bestimmten Dateitypen, andere unterstützen bei der Erstellung von Schadsoftwaresignaturen. Weitere Tools haben die Aufgabe, bereits bestehende Protokollierungs- und Netzwerkdaten zu analysieren und bei erkannten IT-Sicherheitsproblemen Alarmer zu erzeugen. Da in dieser Arbeit jedoch IT-Systeme untersucht werden, die die Bearbeitung von bereits erkannten Vorfällen unterstützen und automatisieren sollen, fallen IT-Systeme, die sich auf die Detektion und Analyse von IT-Sicherheitsvorfällen oder von einzelnen Artefakten beschränken, aus der Betrachtung heraus.

Diese Arbeit untersucht also die folgende These: **„Die verfügbaren IT-Systeme zur Automatisierung der IT-Sicherheitsvorfallsreaktion unterstützen Organisationen in der Bearbeitung von IT-Sicherheitsvorfällen, wie sie in [6] und [7] theoretisch definiert ist.“**

Die Untersuchung der Hypothese wird in dieser Arbeit in drei Schritten bearbeitet:

1. Die Erarbeitung von Anforderungen aus der fachlichen Beschreibung des IT-Sicherheitsvorfallsreaktionsprozesses (siehe Kapitel THEORETISCHE GRUNDLAGEN und ANFORDERUNGEN).
2. Die Entwicklung einer Methodik zur Evaluation dieser Anforderungen durch die Entwicklung von Testfällen und die Beschreibung der Laborumgebung zur Durchführung dieser Testfälle (siehe Kapitel TESTSZENARIEN UND TEST-

FÄLLE, AUSWAHL DER IT-SYSTEME sowie ENTWICKLUNG EINER LABOR-UMGEBUNG).

3. Die Beschreibung des Umsetzungsstandes der erarbeiteten Anforderungen bzw. Testfälle bezogen auf die ausgewählten Testobjekte und die Identifizierung von möglichen Umsetzungslücken (siehe Kapitel TESTDURCHFÜHRUNG und FAZIT UND AUSBLICK).

## 1.2. Abgrenzung

Die auch im Titel der Arbeit angesprochene Automatisierung bezieht sich nicht auf eine vollständige Automatisierung der Vorfallsreaktion, darf also nicht mit einem sich selbst betreibenden System verwechselt werden, das auf alle IT-Sicherheitsvorfälle gleichermaßen angemessen reagieren kann. Vielmehr geht es um die (Teil-)Automatisierung des Prozesses zur IT-Sicherheitsvorfallsreaktion. Der Prozess wird zwar immer noch von Menschen kontrolliert, diese werden jedoch in ihrer Aufgabenerfüllung durch die in dieser Arbeit evaluierten IT-Systeme unterstützt. Im FAZIT UND AUSBLICK werden Möglichkeiten betrachtet, wie zukünftig ein höherer Grad der Automatisierung erreicht werden kann.

Aufgrund des beschränkten Umfangs der Arbeit, kann die Evaluation nicht für alle verfügbaren Automatisierungssysteme erfolgen, stattdessen werden zwei exemplarische Varianten überprüft. Die in dieser Arbeit entwickelten Evaluationskriterien lassen sich jedoch auch auf weitere Automatisierungslösungen anwenden.

Da der Fokus der Arbeit auf der Frage liegt, inwieweit die IT-Systeme Organisationen bei der Vorfallsreaktion entlasten können, ohne selbst noch weiteren Aufwand zu erzeugen, werden die hier vorgestellten IT-Systeme so genutzt, wie sie vom Hersteller zur Verfügung gestellt werden. Das heißt, sie werden nicht angepasst oder durch weitere IT-Systeme in ihrem Funktionsumfang erweitert.

## 2. Theoretische Grundlagen

Um das in der Einleitung beschriebene Ziel der Evaluation von IT-Systemen zur Automatisierung der Bearbeitung von IT-Sicherheitsvorfällen zu erreichen, werden in diesem Kapitel zunächst die fachlichen Grundsätze beschrieben. Zunächst wird dargestellt, welche Aktivitäten überhaupt als IT-Sicherheitsvorfälle angesehen werden können, wie solche Vorfälle beschrieben werden können und wie Teams zur Bearbeitung von IT-Sicherheitsvorfällen aufgestellt sind bzw. sein sollten. Anschließend werden mit der Anforderungsanalyse und dem Testmanagement zwei Werkzeuge aus dem Fach der Softwareentwicklung vorgestellt. Diese werden genutzt, um die Anforderungen an ein IT-System zur Unterstützung von Vorfallsteams abzuleiten und im weiteren Verlauf zu überprüfen.

### 2.1. Umgang mit IT-Sicherheitsvorfällen

Auch wenn im Rahmen der Präventionsmaßnahmen innerhalb eines ISMS (Informationssicherheitsmanagementsystem) versucht wird, die Angriffsfläche von IT-Systemen, IT-Anwendungen und IT-Netzwerken soweit zu reduzieren, dass diese möglichst sicher sind, ist es doch möglich, dass eine Angreiferin die vorgesehenen Sicherheitsmechanismen überwindet und sich unberechtigten Zugriff verschafft. [8]

#### **Definition**

*Als Angreiferin wird in dieser Arbeit eine unberechtigte dritte Partei definiert, die sich Zugriff zu einem IT-System verschafft. Die Aktivitäten, die eine Angreiferin zur Erreichung ihres Ziels ausführt, werden demzufolge als Angriff bezeichnet.[9]*

Die jeweilige betrachtete Organisation muss also Ressourcen in die drei Bereiche der IT-Sicherheit investieren:

1. Prävention: Das Absichern der Infrastruktur zur Reduktion der Angriffsfläche und einem damit einhergehenden geringeren Risiko
2. Detektion: Das Erkennen von Angriffen bzw. Angriffsversuchen auf die eigene IT-Infrastruktur bzw. von der eigenen IT-Infrastruktur ausgehend
3. Reaktion: Die Etablierung von Prozessen zur Vorbereitung auf IT-Sicherheitsvorfälle und zur Herstellung der Fähigkeit auf diese Vorfälle angemessen zu reagieren

Zwischen der Detektion und der Reaktion besteht dabei ein besonders starker Zusammenhang: Diese beiden Aspekte bedingen sich gegenseitig und können auch als Teil eines vollständigen Prozesses zur Vorfallsreaktion betrachtet werden, wie noch im Weiteren deutlich werden wird. [8]

Durch den technischen Aufbau von IT-Systemen und -netzwerken ist es für eine Angreiferin in der Regel nicht möglich, das avisierte Ziel (etwa das Kopieren von Informationen oder die Beeinträchtigung des Betriebs) ohne Umwege zu erreichen. Stattdessen müssen im Laufe eines Angriffs mehrere Schritte durchlaufen werden, die in [10] als so genannte *Killchain* beschrieben werden:

- Initial Access: In diesem ersten Schritt versucht die Angreiferin Zugriff auf einen ersten Teil der Infrastruktur ihres Ziels zu erlangen. Beispiele hierfür wären der Versand einer E-Mail, mit der Zugangsdaten von Benutzerkonten abgefragt werden sollen (Phishing) oder die Ausnutzung einer Schwachstelle um Zugang zu einem IT-System zu erlangen.
- Execution: In dieser Phase gelingt es der Angreiferin auf einem System schadhafte Software auszuführen. Dieser Schadcode könnte zum Beispiel durch einen im ersten Schritt versandten E-Mailanhang eingebracht werden.
- Persistence: Um nach dem Neustart des Rechners oder dem Abmelden des aktuellen Benutzers weiterhin auf einem infizierten System aktiv sein zu können, versucht die Angreiferin dafür zu sorgen, einen permanenten Zugang zu erhalten. Dies ist beispielsweise durch das Erstellen von eigenen, legitim aussehenden Benutzerkonten oder der Infektion von Standardprogrammen, die bei jedem Systemstart ausgeführt werden, möglich.

- **Privilege Escalation:** Um mehr Bewegungsfreiheit in der IT-Landschaft des Opfers zu haben, versucht die Angreiferin ihre Rechte zu erweitern, z.B. um ein Benutzerkonto mit Administratorberechtigungen zu erhalten.
- **Defense Evasion:** Da Angreiferinnen davon ausgehen können, dass auf Ihren Zielsystemen Mechanismen zur Detektion von Angriffen vorhanden sind, versuchen sie Maßnahmen zu ergreifen, damit sie nicht erkannt werden. So könnten sie beispielsweise ausschließlich eigentlich legitime Programme auf den IT-Systemen verwenden, die sie für ihre Angriffsschritte missbrauchen.
- **Credential Access:** Im weiteren Verlauf versucht die Angreiferin, weitere Zugangsberechtigungen zu erlangen, damit sie sich weiter unerkant durch das Netzwerk bewegen kann.
- **Discovery:** Um interessante Systeme oder Benutzerkonten zu finden, untersucht die Angreiferin die vorgefundene Umgebung auf ihre Rahmenparameter und Besonderheiten. So kann sie sich einerseits gezielt zu interessanten Informationen bewegen, andererseits aber auch die Systemumgebung besser kennenlernen und ggf. speziellere Angriffstechniken entwickeln.
- **Lateral Movement:** Hat die Angreiferin interessante Ziele ausgemacht, versucht sie sich vom initialen Zugangspunkt dorthin zu bewegen, wo die nächsten Schritte ausgeführt werden sollen. Dies kann das tatsächliche Zielsystem sein, aber auch die Konzentration auf Etappenziele zur Verbesserung der Angriffssituation ist möglich.
- **Collection:** Davon ausgehend, dass die Angreiferin es auf Informationen abgesehen hat, ist dies der Schritt, in dem sie die für sich relevanten Informationen sammelt und dafür sorgt, dass diese in einem weiteren Schritt kopiert bzw. verarbeitet werden können. Ein einfaches Beispiel wäre die Sammlung von Dateien, die auf USB-Sticks gespeichert sind oder die Kopie von Daten, die auf dem Zielsystem geöffnet werden.
- **Exfiltration:** Die Angreiferin bewegt die zuvor gesammelten Daten dorthin, wo sie direkten Zugriff darauf hat. Das kann zum Beispiel erfolgen, in dem ein zum Internet geöffneter Dateiserver genutzt wird.

- **Command and Control:** Mittels der Command-and-Control-Mechanismen steuert die Angreiferin ihre Aktivitäten in der Zielumgebung aus der Ferne. Dies kann z.B. über die Steuerung per Webserver oder über die Nutzung von im System eingebauten Remote-Management-Tools <sup>1</sup> umgesetzt werden.
- **Impact:** Impact ist der Teil des Angriffs, bei dem die Angreiferin die ursprünglich beabsichtigten Aktivitäten durchführt. Beispiele wären das Entwenden von Dokumenten (siehe Collection und Exfiltration) oder auch das Verschlüsseln von Daten zur anschließenden Erpressung von Lösegeld.

### Definition

*Ein IT-Sicherheitsvorfall ist demnach die Verletzung mindestens eines der IT-Sicherheitsschutzziele (Verfügbarkeit, Vertraulichkeit, Integrität) durch die oben genannten Aktivitäten. Für diese Arbeit wird dies jedoch auf erfolgreich durchgeführte Aktivitäten eingeschränkt. Aktivitäten, die von Präventionsmechanismen blockiert wurden, stellen nach dieser engen Definition keinen IT-Sicherheitsvorfall dar, ziehen somit keinen Reaktionsprozess nach sich und werden daher in dieser Arbeit nicht weiter betrachtet. [11] [6, S. 6]*

### 2.1.1. IT-Sicherheitsvorfallsreaktionsprozess

Als Grundlage für die Beschreibung des IT-Sicherheitsvorfallsreaktionsprozesses wurden mit [6] und [7] die beiden Standards ausgewählt, die auch in der wissenschaftlichen Diskussion immer wieder zur Beschreibung dieses Prozesses herangezogen werden, wie zum Beispiel [8] feststellt. An der Beschreibung dieses Prozesses orientiert sich im Folgenden auch die Aufnahme von Anforderungen an eine Anwendung zur Unterstützung und Automatisierung des Vorfallsreaktionsprozesses. Ersterer Standard legt den Fokus eher auf die Beschreibung des eigentlichen Vorfallsreaktionsprozesses, letzterer auf die Beschreibung der benötigten Teamfunktionen eines Teams zur Bearbeitung von IT-Sicherheitsvorfällen.

---

<sup>1</sup>z.B. PSEXec, ein von Windows-Administratoren verwendetes Programm



Wie in Abbildung 2.1 erkennbar, besteht der Prozess zur Bearbeitung von IT-Sicherheitsvorfällen aus vier Phasen, die grundsätzlich hintereinander ablaufen, sich jedoch auch wiederholen können.

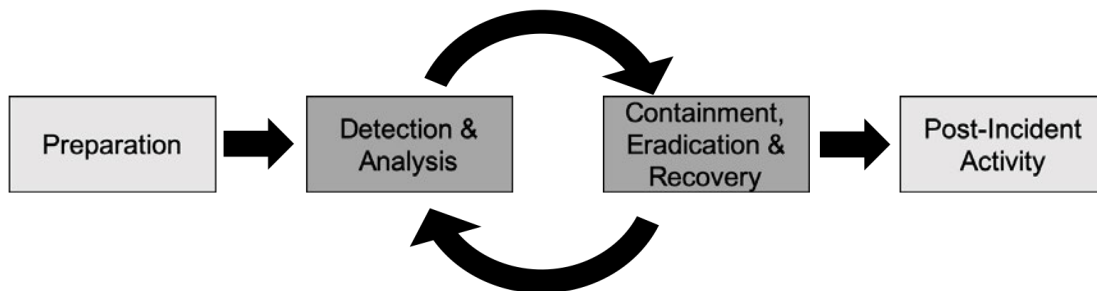


Abbildung 2.1.: IT-Sicherheitsvorfallsprozess nach [6]

### Vorbereitung (engl. Preparation)

In der Phase der Vorbereitung bereitet sich die Organisation auf die Behandlung von IT-Sicherheitsvorfällen vor.

Als erster Schritt steht hier die Klärung, welcher Teil einer Organisation überhaupt für die Behandlung von Vorfällen verantwortlich ist. Die Verantwortlichen müssen für die übrigen Teile der Organisation (z.B. den IT-Helpdesk oder die Anwenderinnen) erreichbar sein, sodass sie über IT-Sicherheitsvorfälle informiert werden können. Neben dieser grundsätzlichen Klärung der Verantwortlichkeit ist auch die organisatorische Vorbereitung auf IT-Sicherheitsvorfälle notwendig: Es muss geklärt werden, welche Möglichkeiten und Befugnisse zur Reaktion den Zuständigen gegeben sind, wie Eskalationswege genutzt werden können und auf welche weiteren Ressourcen der Organisation (z.B. zusätzliches Personal, finanzielle Ressourcen) die Zuständigen zugreifen dürfen. [6, S. 21ff]

Auch die technische Vorbereitung fällt in diese Phase: Analysemöglichkeiten für die Untersuchung von möglicherweise infizierten IT-Systemen müssen geschaffen werden, sichere Dateiablagen hergestellt und Möglichkeiten zur Durchführung von Datensicherungen oder das Mitschneiden von Netzwerkverkehr vorbereitet werden. Hinzu kommt die Bereitstellung von *Baselines*, also der Beschreibung, wie sich die IT-Umgebung verhält, während es keine (bekannte) Angriffsaktivität gibt [6, S. 23].

In einigen Prozessmodellen (z.B. [12]) wird zwischen der strategischen Vorbereitung und der unmittelbaren Vorbereitung einer Vorfallsuntersuchung unterschieden. Die

strategische Vorbereitung bezieht sich auf die Klärung grundsätzlicher Fragen, wie z.B. die Frage nach der Zuständigkeit; die unmittelbare Vorbereitung beinhaltet beispielsweise die Vorbereitung der Analyseumgebung, kurz bevor es zur eigentlichen Analyse kommt.

### Detektion und Analyse (engl. Detect and Analyze)

IT-Sicherheitsvorfälle können auf unterschiedliche Arten und Weisen detektiert, also erkannt werden. Je nach Angriffsmechanismus, bzw. Phase des Angriffs sind hierzu verschiedene Mechanismen notwendig. Zeichen eines IT-Sicherheitsvorfalls können technisch sowohl auf Endpunkten bzw. Knoten (Clients oder Server) und im Netzwerk, also auch in den Kommunikationsbeziehungen der Knoten untereinander erkannt werden (siehe Abbildung 2.2). Allerdings ist es auch möglich, dass Endanwenderinnen oder Administratorinnen im Rahmen ihrer Arbeit Auffälligkeiten feststellen, die zur Entdeckung eines IT-Sicherheitsvorfalls führen. [6, S. 27f]

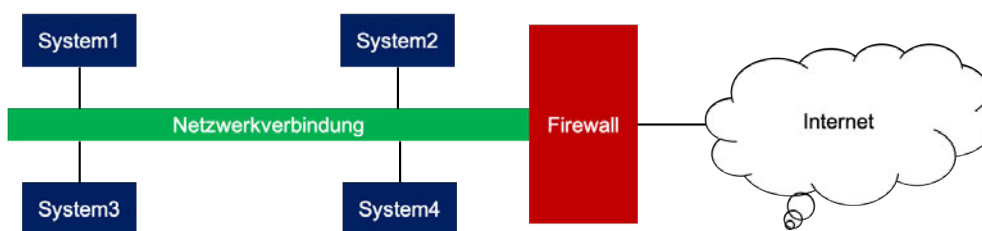


Abbildung 2.2.: Möglichkeiten der Detektion: netzwerkbasiert (grün bzw. rot), systembasiert (blau)

#### Definition

*Wird ein IT-Sicherheitsvorfall gemeldet, der sich im Laufe der Untersuchung als Falschmeldung herausstellt, bezeichnet man diese Meldung als false-positive. Die Anzahl der False-Positives sollte möglichst gering sein, da die Zuständigen für die Bearbeitung der IT-Sicherheit sonst überlastet werden. [8]*

Welche Mechanismen zur Detektion eingesetzt werden können, ist von der betroffenen IT-Landschaft abhängig. Da sich diese Arbeit nicht mit der Detektion beschäftigt, ist in diesem Kontext unerheblich, ob hostbasierte (z.B. Virens Scanner oder Endpoint-Detection-and-Response-Systeme) oder netzwerkbasierte (Proxyserver, Firewall, Intrusion-Detection oder Intrusion-Prevention) Erkennungsmechanismen zum Einsatz

kommen. Einen Überblick, welche Mechanismen in Organisationen eingesetzt werden können, findet sich in der Literatur, beispielsweise in [13].

Wurde ein IT-Sicherheitsvorfall erkannt, so schließt sich zunächst die Analyse der Meldung an. Die Analyse beginnt an der Stelle, wo der IT-Sicherheitsvorfall erkannt worden ist. Von dort ausgehend werden weitere Untersuchungen vorgenommen, weil dort schon die ersten Informationen über die Angreiferin vorliegen. Welche Schritte der Analyse sich daran anschließen, hängt davon ab, wodurch der Sicherheitsvorfall erkannt wurde. Wurde eine Netzwerkanomalie festgestellt, wäre es eine Option, zu untersuchen, von welchem IT-System diese Anomalie ausging. Dieses System könnte man dann weiteren Untersuchungen unterziehen, beispielsweise indem das System mit Mitteln der IT-Forensik untersucht wird oder in dem die Nutzerin des Systems nach Auffälligkeiten auf dem System befragt wird. [7, S. 79] [6, S. 28ff]

Während der Analyse finden Analystinnen Hinweise auf die Angreiferinnen, in so genannten Artefakten. [7, S. 85] In jüngeren Veröffentlichungen unterscheidet man diese Hinweise in *observables* und *Indicators-of-Compromise (IoC)* [14].

### Definition

*Artefakte sind Dateien, Protokolle und sonstige Eigenschaften von IT-Systemen, die auf Aktivitäten der Angreiferin hinweisen. Eine zurückgelassene Schadcode-datei oder ein Protokoll in dem Aktivität der Angreiferin sichtbar wird, wäre ein solches Artefakt. Artefakte können dazu genutzt werden, um aus ihnen weitere Hinweise auf Verhalten von Angreiferinnen (IoC und observables) zu finden.*

### Definition

*Als observables werden beobachtbare Merkmale, die im Zusammenhang mit einem Angriff auftreten können, bezeichnet. Eine Untermenge der observables sind die Indicators of Compromise (IoC). Dabei handelt es sich um observables, die nur von Angreiferinnen benutzt werden, also keinen legitimen Anwendungszweck haben (siehe Abbildung 2.3).*

Observables können beispielsweise IP-Adressen, Hashwerte von Dateien oder Speicherbereichen sein, bestimmte Konstellationen von aufgerufenen Bibliotheken

oder URLs. Ein IoC hingegen wäre beispielsweise ein Benutzerkonto, von dem bekannt ist, dass eine Angreiferin dieses für den Angriff angelegt hat. Um IoC und observables schneller zu identifizieren, hilft es, die eigene Systemumgebung möglichst gut zu kennen. Sowohl der regelmäßige Austausch mit dem IT-Betrieb, die regelmäßige Untersuchung der vorliegenden Daten als auch das bereits beschriebene Baselineing sind hierfür wichtig.

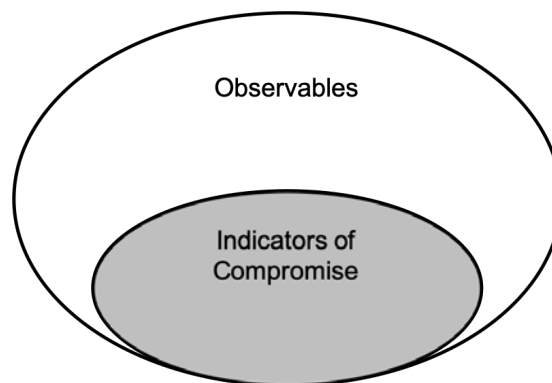


Abbildung 2.3.: Abgrenzung von Observables zu Indicators of Compromise

Ziel der Detektions- und Analysephase ist, den IT-Sicherheitsvorfall zu validieren, um False-Positives auszuschließen [6, S. 29] und um ihn seiner Schwere und Auswirkung auf die Organisation entsprechend einzustufen. Die Einstufung eines IT-Sicherheitsvorfalls erfolgt nach den Dimensionen Auswirkung auf die Funktionsfähigkeit, Auswirkung auf Informationen und Wiederherstellbarkeit nach dem Vorfall. [6, S. 32f] Diese Dimensionen können miteinander kombiniert werden, um eine möglichst starke Aussagekraft zu erhalten.

Das Ergebnis dieser Phase ist also entweder ein erledigter IT-Sicherheitsvorfall (bei False-Positives) oder ein bestätigter IT-Sicherheitsvorfall, der initial dokumentiert und bewertet ist. Falls nötig, werden zum Abschluss dieser Phase auch schon Eskalations- und Benachrichtigungsmechanismen ausgelöst [6, S. 33f], damit zusätzliche Personen über den Vorfall informiert werden und die Behandlung des IT-Sicherheitsvorfalls unmittelbar beginnt. Abbildung 2.1 zeigt, dass sich die Detektions- und Analysephase mit der folgenden Eindämmungs-, Behebungs- und Wiederherstellungsphase immer wieder abwechseln kann.

### **Eindämmung, Behebung und Wiederherstellung (engl. Containment, Eradication and Recovery)**

In der Phase der Eindämmung soll die Ausbreitung der Angreiferin gestoppt werden. Das bedeutet insbesondere, dass nicht noch weitere Systeme, Ressourcen bzw. Netzwerkbereiche von dem Vorfall getroffen werden. Die Gegenmaßnahmen müssen so umfassend sein, dass sie alle betroffenen Ressourcen berücksichtigen. Andernfalls wäre die Eindämmung nicht erfolgreich – die Angreiferin könnte sich weiter ausbreiten. Wie die Eindämmung durchgeführt wird, hängt von dem jeweiligen Vorfall ab [6, S. 35f]:

- Schaden bzw. Diebstahl von Ressourcen: Ist es wichtig herauszuarbeiten, welche Daten manipuliert oder entwendet wurden oder ist eine schnelle Wiederherstellung wichtiger?
- Notwendigkeit der Beweissicherung: Soll später in einem juristischen Verfahren ermittelt werden, wer den Sicherheitsvorfall verursacht hat, so müssen gerichts-feste Beweise, die auf die Identität der Angreiferin hinweisen könnten, gesichert werden. Die Spuren dürfen dann möglichst nicht manipuliert werden.
- Erreichbarkeit der Anwendungen: Können Dienste oder betroffene Systeme für den Zeitraum der Eindämmung eingeschränkt werden? Falls nein, kann dies die Möglichkeiten der Eindämmung einschränken.
- Dauer der Maßnahmen: Einige Maßnahmen können nur für kurze Zeit durchgeführt und aktiviert werden, da sonst beispielsweise ein zu hoher wirtschaftlicher Schaden entstehen würde.

Für manche Organisationen ist es relevant, die Quelle des Angriffs und nicht nur dessen Auswirkungen zu identifizieren. Insbesondere im zwischenstaatlichen Kontext, bei der Strafermittlung oder im Zusammenhang mit nachrichtendienstlich relevanten Fällen ist dies von Bedeutung, da durch diese sogenannte *Attribution* auch außenpoli-tische Maßnahmen opportun werden könnten. [15, S. 22f]

Nachdem der Vorfall eingedämmt wurde, müssen die nächsten Schritte ergriffen werden, die die Angreiferin vollständig aus dem Netzwerk entfernen sollen, bzw. ihr

auch die Möglichkeit nehmen sollen, erneut Zugriff auf das IT-Netzwerk zu erhalten. Entscheidend bei der Behebung ist es, dafür so sorgen, dass alle identifizierten Systeme abgesichert werden. Dies kann erreicht werden, indem die Werkzeuge der Angreiferin entfernt bzw. unbrauchbar gemacht werden oder aber in dem das gesamte System neu aufgesetzt wird. Ersteres bietet den Nachteil, dass immer noch ein Restrisiko besteht, dass die Angreiferin doch noch eine Zugriffsmöglichkeit hat, die bei der Analyse und Detektion übersehen wurde. Allerdings ist es in manchen Situationen aus praktischen Erwägungen heraus nicht möglich, das gesamte Netzwerk neu aufzubauen. In diesem Schritt sind jedoch nicht nur infizierte IT-Systeme zu berücksichtigen. Es ist auch möglich, dass die Angreiferin die Möglichkeit erhalten hat, auf ein Netzwerk Zugriff zu erhalten, ohne dafür ein spezifisches IT-System zu benötigen. [6, S. 37f] So könnte die Angreiferin beispielsweise einen zusätzlichen Benutzeraccount angelegt haben, mit dem sie sich unbemerkt weiter im Netzwerk bewegen kann. Auch andere Offenlegungen von Zugängen sind denkbar, zum Beispiel wenn die Angreiferin in der Lage war, auf den Domänencontroller, die zentrale Verwaltungsinstanz einer Microsoft-Active-Directory-Umgebung zuzugreifen. In diesem Fall hätte sie die Möglichkeit, sich als beliebige Benutzerin in dieser Umgebung auszugeben. [10]

Bei der Wiederherstellung schließlich kommt es darauf an, dass der Normalzustand der Umgebung wiederhergestellt wird. Während der vorherigen Schritte können für die Funktion der Organisation benötigte Betriebsmittel in Mitleidenschaft gezogen werden, etwa weil ein Server für eine wichtige Anwendung neu aufgebaut werden musste. Auch könnte es im Rahmen der Eindämmung zu Einschränkungen für Benutzerinnen gekommen sein, die dazu geführt haben, dass die Organisation nicht mehr so effizient arbeiten konnte wie zuvor. Das ist beispielsweise der Fall, wenn in diesem Schritt neue IT-Sicherheitsregeln eingeführt werden. Diese sind zu Beginn ungewohnt und müssen erst gelernt und gelebt werden. Ziel dieses Schritts ist sowohl die Einschränkungen durch den IT-Sicherheitsvorfall als auch die Einschränkungen durch die Behandlung des IT-Sicherheitsvorfalls zu überwinden. Im besten Fall steht am Ende eine Organisation, die mit gestärkten Präventions- und Detektionsmechanismen den aktuellen bestehenden und zukünftigen Bedrohungen für IT-Netzwerke

widerstehen kann. [6, S. 37f]

Die Phase der Eindämmung, Behebung und Wiederherstellung ist insgesamt also darauf ausgelegt, dass die in der vorherigen Phase Detektion und Analyse gewonnenen Erkenntnisse genutzt werden, um das betroffene IT-Netzwerk zu bereinigen und zum Regelbetrieb zurückzuführen.

Die Phasen Analyse und Detektion bzw. Eindämmung, Behebung und Wiederherstellung wechseln sich immer weiter ab (siehe Abbildung 2.1). Das heißt, kommt es z.B. während der Eindämmung zu neuen Detektionen der Angreiferin, so fließen diese auch in die Eindämmung ein. Zudem werden neue Analyseergebnisse auch fortwährend in die Planung für die Behebung und Wiederherstellung einfließen. Ab einem gewissen Punkt kann es sinnvoll sein, nicht mehr einzelne IT-Systeme als betroffen anzunehmen, sondern davon auszugehen, dass ganze Netzwerkbereiche betroffen sind und darauf auch die Behebungs- und Wiederherstellungsmaßnahmen abzielen.

Bei der Behandlung von unterschiedlichen IT-Sicherheitsvorfällen können sich bestimmte Schritte immer wieder ähneln. Daher kann es sinnvoll sein, bei der Behandlung von IT-Sicherheitsvorfällen auf so genannte *Standard Operating Procedures* (SOP) zu setzen, um die Fehlerrate bei der Behandlung von IT-Sicherheitsvorfällen (ausgelöst durch repetitive Tätigkeiten) zu reduzieren.

### Definition

*Eine Standard Operating Procedure (SOP) ist eine Beschreibung von standardisierten Aktivitäten, die dazu genutzt werden, die Varianz in der Durchführung von Prozessen zu reduzieren.*

SOPs kommen ursprünglich aus dem Bereich der Flugsicherheit und dienen dazu, Maßnahmen, die immer wieder standardisiert durchgeführt werden, zu beschreiben, damit sie dann in einer höheren Qualität durchgeführt werden können. Das leitet sich aus der Überlegung ab, dass sich bei wiederholenden Aufgaben im Laufe der Zeit Fehler einschleichen. Dennoch lösen sich IT-Sicherheitsvorfälle nicht einfach durch die Befolgung von SOPs: Selbst wenn einzelne Schritte sich gleichen oder ähneln, ist die

Expertise von Analystinnen gefragt, wenn es um die Reaktion auf zuvor noch nicht berücksichtigte bzw. bekannte Phänomene geht. [16] [6, S. 6] [7, S. 28]

### **Nachbereitung (engl. Post-Incident Activity)**

Das Ziel der Nachbereitung ist, aus dem zuvor bearbeiteten IT-Sicherheitsvorfall zu lernen und so dafür zu sorgen, dass die IT-Sicherheit einer Organisation stetig steigt, ganz im Sinne des kontinuierlichen Verbesserungsprozesses.

Bei dieser Aufbereitung wird zunächst versucht darzustellen, was bei dem jeweiligen IT-Sicherheitsvorfall zu welchem Zeitpunkt passiert ist und zwar sowohl bezogen auf den Angriff, als auch bezogen auf die Analyse und Reaktion der Organisation. Es wird dabei sowohl die Arbeit des Vorfallteams, als auch die Zusammenarbeit mit externen Stellen, wie der Polizei, anderen IT-Sicherheitsteams und Dienstleisterinnen, auf den Prüfstand gestellt. Damit wird herausgearbeitet, wie sich die Organisation auf zukünftige IT-Sicherheitsvorfälle vorbereiten kann. [6, S. 38]

Außerdem kann es sinnvoll sein, nach der Vorfallsbearbeitung einige Kennzahlen für die Behandlung des Vorfalls zu sammeln, um Vergleichbarkeit zwischen den Vorfällen zu erzeugen und somit die Verbesserung des Organisation über einen gewissen Zeitraum hinweg zu messen. [7, S. 81]

Die Phase der Nachbereitung geht am Schluss auch wieder in die Vorbereitung über, nachdem die Einzelaktivitäten für einen Vorfall abgeschlossen sind und wieder die generelle Vorbereitung auf die nächsten Vorfälle durchzuführen ist.

### **Threat-Intelligence**

Die im Schritt der Analyse des IT-Sicherheitsvorfalls gewonnenen Observables und Indicators of Compromise sind zusammengefasst Teil der Threat-Intelligence.

#### **Definition**

*Threat Intelligence ist die Beschreibung von Wissen über Angreiferinnen, unter anderem anhand von Indicators of Compromise und observables, aber auch durch bekannte Modi-operandi oder Verhaltensweisen. [17]*

Je mehr über die Angreiferin bekannt ist (also je mehr Threat-Intelligence vorhanden ist), desto besser kann das Netzwerk auf weitere infizierte Systeme hin durchsucht



werden und desto hilfreicher sind die gesammelten Informationen auch für andere Organisationen, mit denen man diese Threat-Intelligence gegenseitig austauscht. [17, S. 5] Nicht jede Organisation kann für jede Angreiferin die gleichen Informationen sammeln, sodass sie darauf angewiesen sind, diese Informationen innerhalb ihrer Gemeinschaft zu teilen. Teilnehmerinnen dieser Gemeinschaft können sowohl andere Teilbereiche der eigenen Organisation sein, aber auch befreundete Organisationen, staatliche Stellen oder Organisationen, die sich zum Austausch über Threat-Intelligence gegründet haben. [18] [19]

Innerhalb dieser peer-groups werden Informationen getauscht, wenn davon ausgegangen wird, dass die Informationen auch für andere hilfreich sind und keine Probleme bzw. Nachteile für die eigene oder andere befreundete Organisationen entstehen. [20] [7, S. 132f]

### 2.1.2. Beteiligte Parteien bei der Vorfallsreaktion

Nicht nur der eigentliche Prozess der Vorfallsreaktion ist eine Grundlage für das Verständnis eines IT-Sicherheitsvorfallteams, sondern auch die organisatorischen Rahmenbedingungen, sowie die Schnittstellen des Teams. Die bei der Vorfallsreaktion beteiligten Entitäten sollen hier im folgenden beschrieben werden.

#### CSIRT/Vorfallsteam

Definition
<i>Das CSIRT (engl. Computer Security Incident Response Team), oder im Rahmen dieser Arbeit auch Vorfallsteam genannt, hat die Aufgabe, innerhalb einer Organisation die Behandlung von IT-Sicherheitsvorfällen zu koordinieren. [7, S. 2]</i>
<i>Im Rahmen dieser Arbeit wird als CSIRT ein internes Team einer Organisation verstanden, das diese Aufgabe wahrnimmt. [7, S. 12]</i>

Das Vorfallsteam bzw. CSIRT kann sowohl als ständig arbeitende Organisationseinheit, aber auch als ad-hoc Team einberufen werden. Auch hybride Ansätze (also ein festes Team als Ausgangspunkt mit Erweiterung um zusätzliches Personal im Bedarfsfall) sind hierbei denkbar. [7, S. 24] Das Team dient als Informationsdrehscheibe und stellt so sicher, dass die nötigen Schritte zur Behandlung des IT-Sicherheitsvorfalls unternommen werden bzw. benötigte Dritte eingebunden werden. [7, S. 103] Letztendlich

liegt es in der Verantwortung des CSIRT den oben beschriebenen Reaktionsprozess mit Leben zu füllen. Hierfür hat das CSIRT verschiedene Gruppen von Dienstleistungen (siehe Abbildung 2.4), die es ihrer Organisation anbietet, bzw. die dabei helfen das Leistungsangebot des CSIRT zu beschreiben. [7, S. 23] Diese Dienstleistungen lassen sich in die Kategorien reaktive Dienstleistungen, proaktive Dienstleistungen und Qualitätssicherungsdienstleistungen aufteilen. [7, S. 24]



Abbildung 2.4.: Übersicht der CSIRT-Services (Grafik aus [7, S. 24])

Die hier näher betrachtete Dienstleistung Dienstleistung *Vorfallsbearbeitung* (engl. *Incident Handling*) ist Teil der reaktiven Dienstleistungen. Reaktive Dienstleistungen beginnen immer mit einem von außen eintreffenden Ereignis.

### Organisation

Die Organisation, in der ein IT-Sicherheitsvorfall stattfindet, kann ebenfalls als eine beteiligte Partei gesehen werden. Je nachdem wie das CSIRT in die Organisation eingebunden ist, schwankt der Einfluss, den das Team hat zwischen dem Recht Anweisungen in Bezug auf den Umgang mit dem Vorfall zu geben oder lediglich Hinweise und Beratung anzubieten. Die Organisation, die von einem Vorfall betroffen ist, hat in der Regel ein Interesse, dass der Vorfall behandelt wird, ohne dass es negative Auswirkungen auf die Leistungserbringung gibt. Um Risiken und den Fortschritt der Vorfallsbehandlung beurteilen zu können, hat die Organisation ein Interesse daran, regelmäßige

Informationen über den Fortschritt der Untersuchung zu erhalten. Vorausgesetzt die Organisation ist an einer Verbesserung der eigenen IT-Sicherheitsmechanismen interessiert, benötigt die Organisation einen Abschlussbericht, um für die Zukunft weitere Gegenmaßnahmen zu stellen. [7, S. 15] Auch wenn dieses Interesse nicht immer besteht, hilft ein Abschlussbericht dem CSIRT dabei, weitere Ressourcen (finanziell, personell, technisch) zu erhalten. Nicht nur nach der eigentlichen Vorfallsbehandlung, sondern auch bei relevanten Zwischenschritten, also z.B. wenn weitere Ressourcen benötigt werden oder aus Sicht der Organisation, wenn z.B. meldepflichtige Sachverhalte entdeckt worden sind, können solche Berichte hilfreich sein.

### **Externe CSIRTs**

Bei der Vorfallsbearbeitung kann die Zusammenarbeit mit externen Teams, also anderen CSIRTs notwendig sein. Entweder, weil das eigene CSIRT aufgrund regulatorischer Vorgaben dazu verpflichtet ist oder weil es bestimmte (Teil-)dienste nicht selbst erbringen kann. Im Falle eines Vorfalls kann es ein Interesse der eigenen Organisation und damit des eigenen CSIRTs sein, die gewonnene Threat-Intelligence mit befreundeten bzw. bekannten CSIRTs zu teilen, etwa weil dies regulatorisch vorgeschrieben ist oder weil sich das eigene CSIRT dadurch eine höhere Bekanntheit oder Vertrauen in der CSIRT-Gemeinschaft erhofft. Durch das Teilen von Threat-Intelligence können wiederum auch andere CSIRTs dazu ermuntert werden, dem eigenen CSIRT Threat-Intelligence zur Verfügung zu stellen. Auch ergriffene Gegenmaßnahmen, bzw. Erkenntnisse wie mit bestimmten Angreiferinnen umgegangen wurde, z.B. welche Schutzmaßnahmen ergriffen werden konnten, sind für den Austausch zwischen CSIRTs interessant. [7, S. 19ff]

Unabhängig davon welche Daten ausgetauscht werden sollen, ist ein Vertrauensverhältnis zwischen den verschiedenen Teams notwendig. Dieses kann jedoch erst im Laufe der Zeit aufgebaut werden. Wichtig ist dabei, dass es sich um eine Beziehung handelt, von der beide Organisationen profitieren. Der o.g. Austausch von Informationen ist eine Möglichkeit dieses Vertrauen aufzubauen.

### **Behördliche Stellen**

Im Falle der Vorfallsbearbeitung kann es passieren, dass die eigene Organisation Meldepflichten und Vorgaben unterliegt, die dazu führen, dass offizielle Stellen über die Behandlung eines IT-Sicherheitsvorfalls zu informieren sind. Das ist besonders relevant, wenn eine Strafverfolgung des möglichen Täters geplant ist. Die Strafverfolgung wird durch die Ermittlungsbehörden durchgeführt. Hierfür müssen diese aber zunächst erstmal informiert werden. Hierbei kann es ebenfalls sinnvoll sein, die gewonnene Threat-Intelligence, aber auch zusammenfassende Berichte über zuvor bereits gewonnene Erkenntnisse zu teilen. [7, S. 19ff] Zumindest im Gebiet der europäischen Union ist es zudem erforderlich, über IT-Sicherheitsvorfälle zu informieren, wenn diese datenschutzrechtliche Relevanz haben. Dort sind nicht die Details zu der Angreiferin relevant, sondern vielmehr die Frage, welche Personen von einem Datenschutzverstoß betroffen waren, sowie die Frage, ob technische oder organisatorische Maßnahmen des Datenschutzes nicht ausreichend implementiert waren. [21, Art. 33]

### **2.1.3. Technische Schnittstellen im Reaktionsprozess**

In den einzelnen Phasen des Vorfallsprozesses kommen verschiedene Anwendungen zum Tragen. Da sich diese Arbeit explizit mit der eigentlichen Behandlung des Vorfalls beschäftigt und sich dabei an den CSIRT-Services orientiert, sollen nicht die Anwendungen als solche beschrieben werden. Der Schwerpunkt wird stattdessen auf die vom Vorfallssystem genutzten Schnittstellen gelegt.

### **Detektionsmechanismen**

Wie beschrieben, können IT-Sicherheitsvorfälle von technischen Detektionsmechanismen erkannt werden. Bei einer erfolgten Detektion wird in der Regel eine Meldung auf Basis von erkannten Anomalien erzeugt, die dann im weiteren ausgewertet werden muss. Die Meldung kann beispielsweise in einem Protokoll auf dem Detektionssystem gespeichert werden oder in ein zentrales Management eingebracht werden. Alarmmeldungen können auch per automatisierter Schnittstelle an andere Systeme (z.B. Ticketssysteme oder Intrusion Prevention Systeme) weitergeleitet werden, sodass dort die Meldungen zusammenfließen. Für diese Arbeit wird nicht genauer betrachtet, wel-

che Arten von Detektionssystemen es gibt bzw. wie deren Güte ist. Vielmehr wird im weiteren Verlauf der weiteren Arbeit davon ausgegangen, dass es sich bei erkannten Meldungen um True-Positives (also erfolgreiche Erkennungsleistungen) handelt, die dem System zur Automatisierung der Vorfallsbehandlung bekannt geworden sind. [13]

### Threat-Intelligence-Plattformen

Wie schon beschrieben, gehört das Sammeln und der Austausch von Threat-Intelligence zu den Aufgaben eines Vorfallsreaktionsteams. Nicht nur um diese Informationen mit anderen Teams zu teilen, sondern auch, um in der Lage zu sein, Informationen zu Angreiferinnen systematisch zu sammeln und aufzubereiten, können spezielle IT-Anwendungen eingesetzt werden. Für die Unterstützung dieser Aufgabe haben sich im Laufe der Zeit verschiedenen Anwendungen herauskristallisiert. In diesen Anwendungen werden IoC und Observables gruppiert, kategorisiert und so aufbereitet, dass es möglich ist, die Entwicklung und das Vorkommen von Angreiferinnen zu untersuchen und zu identifizieren. Da diese Schnittstelle für die Vorfallsreaktion so wesentlich ist (da dadurch besser identifiziert werden kann, wodurch sich die Angreiferin eigentlich auszeichnet), muss sie auch bei der Beschreibung der Automatisierung der Vorfallsreaktion berücksichtigt werden. [22] Im europäischen Raum ist vor allem das Open-Source-Projekt *MISP (Malware Information Sharing Platform)* verbreitet, was möglicherweise auch daran liegt, dass die Software vom Luxemburgischen CIRCL (Computer Incident Response Center Luxembourg) entwickelt und als Open-Source-Software herausgegeben wird. [23]

## 2.2. Anforderungsanalyse

Da in der vorliegenden Arbeit Anforderungen für Systeme zur Unterstützung und Automatisierung der Vorfallsreaktion identifiziert und bestehende Systeme auf die Erfüllung dieser Anforderungen hin untersucht werden sollen, wird zunächst erläutert, wie bei der Erarbeitung von Anforderungen vorgegangen wird: Für das Identifizieren von Anforderungen gibt es Standards und Normen aus dem Bereich des Software-Engineering. Die aus diesem Gebiet betroffene Teildisziplin ist die *Requirement-*

*Analyse*, bzw. die *Systemanalyse*, die aus der Anforderungsermittlung und der anschließenden Spezifikation besteht. [24, S. 7ff]

### Definition

*Eine Anforderung (engl. Requirement) ist eine Bedingung oder Fähigkeit, die von einer Benutzerin (Person oder System) zur Lösung eines Problems oder zur Erreichung eines Ziels benötigt wird. Unterscheiden lassen sich funktionale Anforderungen, die beschreiben, was ein Produkt tun soll sowie nicht-funktionale Anforderungen, die Randbedingungen oder Qualitätskriterien beschreiben, die ein Produkt erfüllen muss.*

Die Anforderungen sollten eindeutig, also zur Vermeidung von Missverständnissen nicht interpretierbar sein. Außerdem sollten sie konsistent, verständlich und vollständig sein. Zudem ist wichtig, dass die Anforderungen testbar bzw. prüfbar sind, damit festgestellt werden kann, ob die Anforderungen tatsächlich durch die Software erfüllt worden sind. [24, S. 11ff] Da in dieser Arbeit nicht ein zu entwickelndes System spezifiziert wird, sondern vielmehr bestehende IT-Systeme auf ihre Übereinstimmung mit den Standards zur Vorfallsreaktion hin überprüft werden, werden die Anforderungen in dieser Arbeit absichtlich nicht zu spezifisch beschrieben: Vielmehr geht es um die Beschreibung von Funktionen, ohne die genaue Beschreibung zu liefern, wie die Funktion implementiert werden muss.

Zur Entwicklung von Anforderungen gibt es unterschiedliche Strategien. So können die Anforderungen einerseits *Bottom-Up*, also von sehr detaillierten Anforderungen, hin zu abstrakteren Clustern von Anforderungen gewonnen werden, es lässt sich jedoch auch ein *Top-Down-Ansatz* nutzen, bei dem ausgehend von abstrakteren Anforderungen, immer detailliertere Anforderungen gewonnen werden. [24, S. 79ff]

Da in dieser Arbeit nicht die Art und Weise einer konkreten Implementierung, sondern vielmehr die zu lösende Aufgabe der Automatisierung des Vorfallsreaktionsprozesses im Vordergrund steht, wird der *Top-Down-Ansatz* gewählt. Ausgehend von der generellen Beschreibung dieser Aufgabe, werden in einem weiteren Schritt detaillierte Anforderungen herausgearbeitet.

Zur Beschreibung der Anforderungen wird in dieser Arbeit auf Userstories gesetzt.

### Definition

*Die Userstories beschreiben in natürlicher Sprache, wofür eine gewisse Funktion benötigt wird. Dadurch wird der Kontext und die Interaktion zwischen System und Nutzerin deutlich. [24, S. 90ff] Zur Beschreibung der Userstory ist es nötig, dass die anfordernde Person, das Ziel der Anforderung (Akzeptanzkriterium) und der Nutzen (Beschreibung der Userstory) definiert ist. [24, S. 91 aus Cohn04]. Zu jeder Userstory wird hier eine eindeutige Kennziffer, ein Namen, sowie ein Quellenverweis dokumentiert, um zu identifizieren, auf welcher Basis die Userstory entstanden ist. Für den Zweck dieser wissenschaftlicher Arbeit wird auch die Herleitung der Userstory aus der angegebenen Quelle beschrieben, damit diese nachvollziehbar bleibt.*

Für die hier betrachtete Problemstellung bietet sich das Verfahren an, da durch die Vielzahl an Schnittstellen der Kontext und die Perspektive der Anforderung sehr wichtig ist. Zudem wird hier ein bereits bestehender Prozess modelliert, der im Rahmen einer Userstory formuliert werden kann.

In der Regel wird die Einhaltung eines Standards als nicht-funktionale Anforderungen gewertet. [24, S. 217ff] Damit wird in dieser Arbeit gebrochen: Die Standards dienen als Quelle für die Beschreibung von Prozessschritten innerhalb der IT-Sicherheitsvorfallsreaktion. Da hier Systeme evaluiert werden, die diesen Prozess automatisieren, handelt es sich bei den Anforderungen aus den Standards in diesem Spezialfall um funktionale Anforderungen. Eine nicht-funktionale Anforderung wäre beispielsweise eine Anforderung, die darauf abzielt einen Standard oder eine Norm zur benutzerfreundlichen Benutzerführung einzuhalten. Dort, wo es für die Funktion der Anwendung gemäß der Standards zur Beschreibung des Vorfallsreaktionsprozesses erforderlich ist, werden nicht-funktionale Anforderungen in dieser Arbeit als Teil der jeweils betroffenen funktionalen Anforderungen berücksichtigt.

## 2.3. Softwaretests

Das Ziel von Softwaretests besteht darin, die Qualität von Softwareprodukten zu testen, bzw. ein Produkt gemäß der Spezifikation abzunehmen. Die Freiheit von Fehlern einer Software kann durch Tests nicht überprüft werden. Hierfür wäre eine formale

Verifikation notwendig. Letztendlich dienen sie damit der Risiko- und Haftungsreduktion.

Softwaretests sind Teil des Prozesses zum Testmanagement und damit auch Teil des Software-Engineering in dem auch die zuvor beschriebene Anforderungsanalyse angesiedelt ist. Softwaretests laufen in mehreren Phasen ab: In der Planung wird die Durchführung des gesamten Testprozesses vorbereitet. Es werden Vorgehensweisen definiert und Verantwortlichkeiten festgelegt. In der zweiten Phase, der Spezifikation, wird definiert, wie die Tests durchgeführt werden. Es werden also Testfälle beschrieben, mit denen die Anforderungen überprüft werden. Im dritten Schritt werden die Tests anhand der zuvor beschriebenen Testanweisungen durchgeführt. Ergebnisse und Abweichungen vom erwarteten Verhalten werden anschließend im nächsten Schritt protokolliert. Am Schluss des Testprozesses steht die Auswertung. Dabei werden die Protokolle mit den Anforderungen abgeglichen, um zu klären, wie die untersuchte Software die Anforderungen erfüllt. In der Regel sollte der Testprozess parallel zum Softwareentwicklungsprozess stattfinden bzw. in diesen eingebettet sein. In dieser Arbeit ist das nicht möglich: Es werden bestehende IT-Systeme untersucht, ob sie aus Standards abgeleitete Anforderungen erfüllen, indem die Anforderungen und die Tests im Nachhinein definiert werden.

Für die Durchführung des Testprozesses haben sich verschiedene Testschulen entwickelt, mit denen verschiedene Herangehensweisen verknüpft sind. Für die hier vorliegende Arbeit wird die *Agile School* herangezogen. Diese findet in der agilen Entwicklung Anwendung, um zu überprüfen, ob die Userstories erfüllt werden. [25, S. 2f, S. 13f, S. 34, S41, S47ff]

### 2.3.1. Testbeschreibung

Damit ein Testfall formal korrekt beschrieben ist, muss er einige Eigenschaften aufweisen: Die Tests müssen reproduzierbar sein und sie sollten geplant sein, d.h. es sollte nicht spontan bzw. ohne Vorbereitungsstrategie getestet werden. Tests müssen sich ebenfalls dem Kriterium der Wirtschaftlichkeit unterwerfen, sie sollten also nicht um jeden Preis durchgeführt werden.



Tests lassen sich in verschiedene Ebenen einteilen. Da in dieser Arbeit die funktionalen Anforderungen in ihrer Gesamtheit überprüft werden sollen, kommt hierfür der *Systemtest* in Frage (siehe Abbildung 2.5).

### Definition

*Beim Systemtest wird das Gesamtsystem gegen die gesamten Anforderungen (funktional und nicht-funktional) getestet. Dafür wird der Test in einer Umgebung ausgeführt, die weitestgehend mit der Produktivumgebung übereinstimmt. [25, S. 71f]*

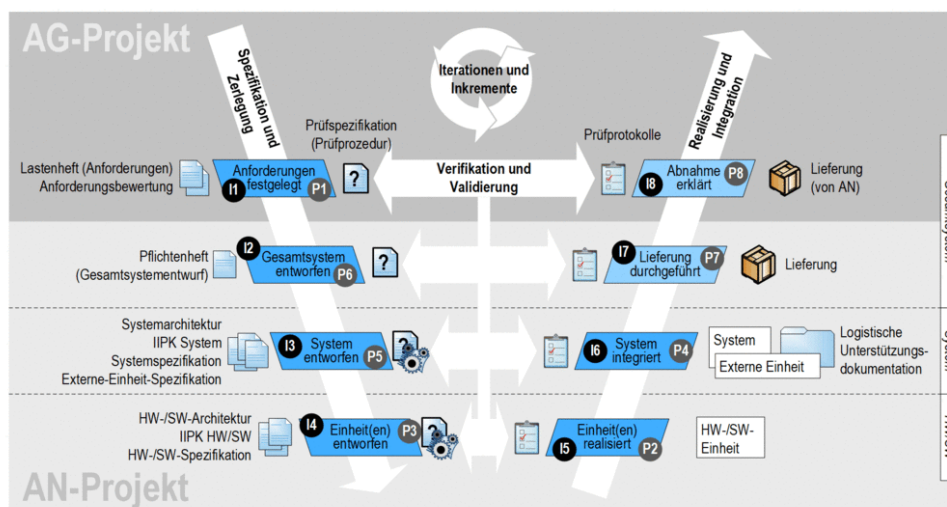


Abbildung 2.5.: Softwareentwicklung im V-Modell XT (nach [26])

Gäbe es für die hier definierten Anforderungen einen tatsächlichen Kunden bzw. eine bekannte Benutzerinnengruppe, käme auch die Durchführung eines zusätzlichen Abnahme- bzw. User-Acceptance-Tests in Frage, bei dem geprüft wird, ob das Produkt von den designierten Anwenderinnen benutzt werden kann (also akzeptiert wird). [25, S. 73]

Testfälle lassen sich in statische und dynamische Testfälle unterteilen. Statische Testfälle basieren darauf, Testfälle auf Basis des Quellcodes durchzuführen (als eine spezielle Form eines Code-Reviews). Dynamische bzw. funktionsorientierte Testfälle basieren auf der Funktionsspezifikation des Testobjekts, also z.B. der hier betrachteten Anwendung. In dieser Arbeit werden dynamische Testfälle genutzt: Wie bereits oben beschrieben geht es darum, die Anforderungen an die Systeme zu überprüfen und zwar anhand der Beschreibung der erwarteten Funktionalität. Da der Quellcode der einzel-

nen IT-Systeme nicht bekannt ist, werden die Tests als Blackbox-Test (im Gegensatz zu einem Whitebox-Test bei bekanntem Quellcode) durchgeführt.

### Definition

*Ein Testfall im Rahmen dieser Arbeit ist immer ein Testfall, der innerhalb eines Systemtests dynamisch als Blackbox-Test durchgeführt wird.*

Bei der Beschreibung der Testfälle ist es grundsätzlich wichtig, dass diese eindeutig beschrieben sind, und keinen Platz für Interpretationen lassen. Andernfalls könnte es passieren, dass ein Test falsch positiv ist, also eine Anforderung als erfüllt angesehen wird, obwohl die Anforderung eigentlich nicht erfüllt wurde. Werden während der Testdurchführung bestimmte Parameter benötigt, müssen diese ebenfalls in der Testbeschreibung dokumentiert werden, um die Reproduzierbarkeit der Testergebnisse zu gewährleisten. [25, S. 151ff] Wie schon bei der Beschreibung der Anforderungen muss davon in dieser Arbeit etwas abgewichen werden: Es geht um die Erfüllung der grundsätzlichen Funktionsanforderungen, nicht um die Art und Weise wie eine Funktion implementiert ist. Daher werden auch an dieser Stelle bewusst Unschärfen in Kauf genommen.

### 2.3.2. Testdurchführung

Bei der Durchführung des Testfalls werden die Testanweisungen exakt so durchgegangen, wie in der Testanweisung beschrieben. Beim ersten Durchlauf des Tests wird gleichzeitig ein Review der Testanweisung vorgenommen. Sind bestimmte Testanweisungen nicht durchführbar, so würde dies spätestens beim ersten Durchlauf auffallen. Das Ergebnis des Tests bzw. aufgetretene Fehler müssen dokumentiert werden, um im Anschluss herausarbeiten zu können, warum ein Fehler aufgetreten ist: Es könnte sowohl der Testfall falsch bzw. unvollständig beschrieben sein, es wäre aber auch möglich, dass eine Anforderung nicht oder nicht vollständig umgesetzt ist. [25, S. 179ff]

## 3. Anforderungen

Nachdem im Kapitel THEORETISCHE GRUNDLAGEN sowohl die inhaltliche als auch die methodische Ausgangslage beschrieben wurde, werden in diesem Kapitel die Anforderungen an ein IT-System zur Automatisierung der standardkonformen Reaktion bei IT-Sicherheitsvorfällen herausgearbeitet. Diese als Userstories formulierten Anforderungen werden dann in den folgenden Kapiteln als Grundlage für die Evaluierung der IT-Systeme genutzt. Sie beschreiben den benötigten Funktionsumfang, den eine Anwendung erfüllen muss, um den IT-Sicherheitsvorfallprozess standardgemäß zu erfüllen.

### 3.1. Vorgehensweise

Diese Arbeit orientiert sich an den beiden für die Beschreibung des IT-Sicherheitsvorfallreaktionsprozesses maßgeblichen Standards [6] und [7]. Wie erwähnt, werden die Anforderungen in dieser Arbeit als Userstories aufgenommen, um der Rollenverteilung und der Tiefe bei funktionalen Anforderungen gerecht zu werden. Um die Userstories zu entwickeln, wurden die Standards dabei auf Vorgaben zur Bearbeitung von IT-Sicherheitsvorfällen hin untersucht. Diese Vorgaben wurden dann analysiert (siehe auch Zeile „Herleitung“ bei der Beschreibung der Userstories) und hinsichtlich der Relevanz bewertet. Die eigentliche Userstory wurde dann aus Sicht der jeweiligen zuständigen Rolle beschrieben, die diese Funktion in der Anwendung nutzen würde und einer der CSIRT-Funktionen zugeordnet.

Wird im Folgenden von dem „IT-System“ gesprochen, ist dabei zunächst ein fiktives IT-System gemeint, zu der an dieser Stelle die Anforderungen aufgenommen werden, um die Bearbeitung von IT-Sicherheitsvorfällen zu unterstützen und zu automatisieren.

### 3.2. Rollen

Um eine Userstory beschreiben zu können, ist es notwendig, auch einen User, also eine Endanwenderin zu identifizieren, aus deren Sicht die Anforderung beschrieben ist. Im Rahmen der Analyse der genutzten Standards und des darin beschriebenen Reaktionsprozesses sind dabei verschiedene Rollen identifiziert worden, die in der Behandlung von IT-Sicherheitsvorfällen relevant sind. Diese Rollen werden im folgenden beschrieben und können sowohl Personen als auch andere IT-Systeme sein.

#### 3.2.1. Vorfallsteammitglied

Die Mitglieder des Vorfallteams bzw. CSIRT sind für die Analyse und Bearbeitung eines IT-Sicherheitsvorfalls verantwortlich. Je nach Team- oder Vorfallsgröße können auch mehrere Personen an der Lösung eines IT-Sicherheitsvorfalls beteiligt sein. Dieses Team ist nicht zu verwechseln mit einem Team, das für die Behandlung eines konkreten Vorfalls betraut ist, sondern ist letztendlich ein Sammelbegriff für die Personen, die in der Vorfallsreaktion insgesamt arbeiten.

#### 3.2.2. Vorfallsteamleiterin

Die Vorfallsteamleiterin hat die Aufgabe, für das gesamte Team und alle von diesem Team bearbeitenden Vorfälle verantwortlich zu sein. Sie hat das Interesse ihr Team weiterzuentwickeln und steht gleichzeitig in der Gesamtverantwortung gegenüber der Organisation. Außerdem muss sie nachvollziehen können, ob die aktuell vorhandenen Ressourcen des Teams ausreichend sind. Zudem dient sie als Eskalationsinstanz, sowohl intern (bei Fragen der einzelnen Teammitglieder), aber auch extern (aus Richtung der Organisation).

#### 3.2.3. Detektionssystem

Das Detektionssystem ist der Mechanismus, der mögliche IT-Sicherheitsvorfälle erkennen soll. Bei einem Detektionssystem kann es sich sowohl um ein spezielles IT-System handeln, das explizit für diese Funktion aufgebaut wurde (z.B. eine Antiviren-Lösung), aber es kann sich im Sinne dieser Arbeit auch um ein System des IT-

Helpdesks handeln, in dem Benutzerinnen ihre Verdachtsfälle melden. Aus diesem System werden dann Meldungen vom IT-Helpdesk an die in dieser Arbeit evaluierten IT-Systeme zur Vorfallsbearbeitung weitergeleitet.

#### 3.2.4. Threat-Sharing-Plattform

Die Threat-Sharing-Plattform verwaltet die gesamte Threat-Intelligence und hat neben der Funktion des Wissensmanagement zu Observables und IoC auch die Aufgabe, Kontakt zu den Threat-Sharing-Plattformen anderer Organisationen zu haben, um ständig Informationen über aktuelle Bedrohungen und Angreiferinnen auszutauschen.

### 3.3. Anforderungen

Um die Anforderungen zu gliedern, werden die einzelnen Funktionen des CSIRT-Dienstes Incident-Response (bzw. Vorfallsbearbeitung) zu Hilfe genommen. Anhand dieser Funktionen lassen sich die Aktivitäten, die im Falle eines Vorfalls von einem IT-Sicherheitsvorfallsteam durchgeführt werden müssen, einteilen. Die Grafik 3.1 zeigt die verschiedenen Dienste und ihre Interaktion miteinander.

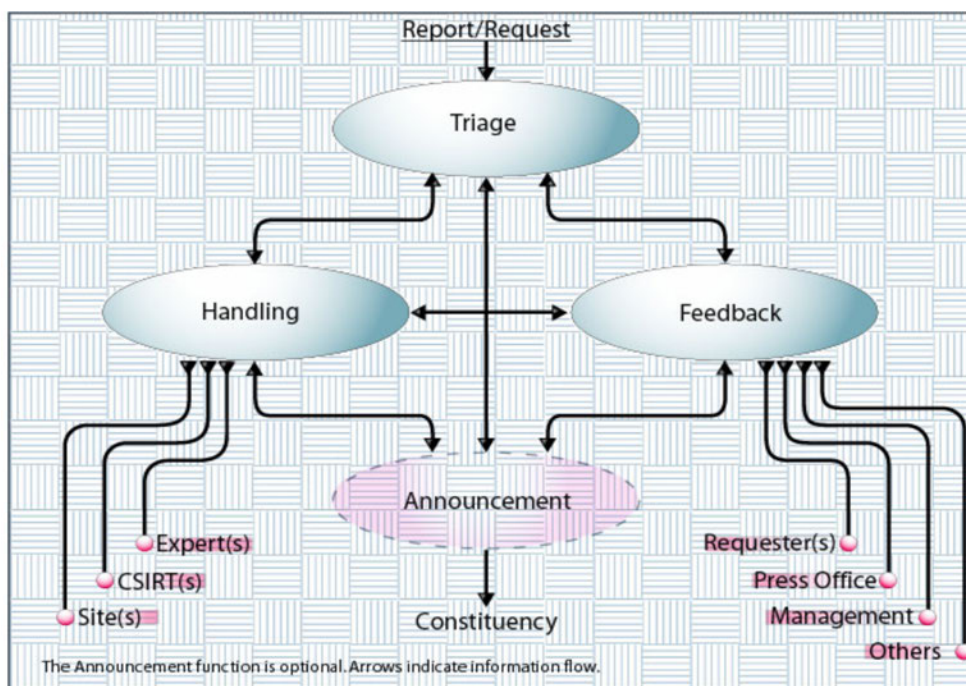


Abbildung 3.1.: Funktionen innerhalb des Vorfallsreaktionsprozess (nach [7, S. 67])

- Triage: Als Eingangskanal in die Vorfallsbearbeitung wird hier geprüft, ob es sich tatsächlich um einen Vorfall handelt bzw. ob die eingehende Meldung bereits zu einem bestehenden Vorfall gehört. Zudem wird hier eine Einstufung hinsichtlich der Priorität der Vorfallsbearbeitung vorgenommen.
- Vorfallsbearbeitung: Hier wird der eigentliche Vorfall bearbeitet, es werden also Analysen durchgeführt und Eindämmung, Behebung sowie Wiederherstellung geplant. Verantwortlich für die eigentliche Vorfallsbearbeitung ist das Vorfallsteammitglied.
- Kommunikation: Die Kommunikation, z.B. mit der eigenen Organisation oder auch mit anderen Teams ist Teil der Vorfallsreaktion. Verantwortlich für diese Kommunikation ist ebenfalls das Vorfallsteammitglied.
- Steuerung: Dies ist keine eigentliche Funktion des Dienstes zur Vorfallsreaktion, sondern eine Querschnittsaufgabe, die letztendlich jede Organisation betrifft. In dieser Arbeit werden hier Aufgaben zusammengefasst, die in der Verantwortung der Vorfallsteamleiterin liegen und die letztendlich ihre Aufgaben als Führungsperson widerspiegeln.

Wie bereits im Kapitel THEORETISCHE GRUNDLAGEN erwähnt, sollten Anforderungen strukturiert aufbereitet werden. Daher werden die Anforderungen nachfolgend aufgeführt und nach den zuvor beschriebenen CSIRT-Diensten gegliedert. Zudem wird die Quelle und die Herleitung der Anforderung beschrieben, um den Kriterien einer wissenschaftlichen Arbeit genüge zu tun. Da die Userstories aus der Perspektive der anfordernden Person bzw. des anfordernden Systems geschrieben sind, sind sie in der „Ich“-Form beschrieben. Die Userstories sind dabei nicht in einer spezifischen Reihenfolge (z.B. nach der Reihenfolge der Prozessdurchführung) sortiert, sondern sind gemäß der vergebenen Userstory-ID aufgelistet. Eine Übersicht über die Zuordnung der einzelnen Userstories zu den Diensten des IT-Sicherheitsvorfallsprozesses liefert Tabelle 3.1 in Verbindung mit Abbildung 3.1.

Tabelle 3.1.: Übersicht über die Userstories

<b>Triage</b>	<b>Vorfallsbearbeitung</b>	<b>Kommunikation</b>	<b>Steuerung</b>
Initiale Beurteilung eines Vorfalls (13) Zuordnung von einem Vorfallsteammitglied zu einem Vorfall (14) Verknüpfen von Vorfällen (15) Manuelle Eskalation von Vorfällen (16) Triage-Queue (17)	Benachrichtigung bei Zuordnung zu einem Vorfall (18) Benachrichtigung bei neuen Vorfällen (19) Protokollierung von Aktionen (20) Bearbeiten von Stammdaten des Vorfalls (21) Anzeigen eines Vorfalls (22) Aufnehmen von Observables / IOC (23) Hinzufügen von Artefakten (24) Analyse von Artefakten (25) Definieren von SOP (26) Automatisches Ausführen von SOP (27) Verweisen auf weitere Dokumente (28) Auflistung von Vorfällen (29) Anzeige des Vorfallszeitstrahls (30) Dokumentation der Vorfallsbearbeitung (31) Bearbeiten von Vorfällen anhand von SOP (32)	Vorbereitung des Abschlussberichts (1) Aufbereitung des Vorfalls für Externe (2) Weitergabe von Threat-Sharing-Daten (3) Eingabe von Threat-Sharing-Daten (4) Weitergabe von Analysetätigkeiten (5) Vorbereitung der Nachbereitungsbesprechung (6)	Pflegen von Vorfallsteammitgliedern (7) Einschränkung des Zugriffs auf Vorfälle (8) Automatische Eskalation von Vorfällen (9) Anzeige von Metriken (10) Zugriffsbeschränkung der Anwendung (11)

#### Userstory 1 (Vorbereitung des Abschlussberichts)

**Beschreibung:**

„Als Vorfallsteammitglied kann ich alle Daten, die zu einem Vorfall gehören herunterladen, sodass ich sie für die Erstellung eines Berichts nutzen kann. Beim Download soll ein Template des Berichts erstellt werden, das bereits mit den bekannten Daten vorausgefüllt ist (Name, Priorität, Timeline, etc.).“

**Akzeptanzkriterium:**

Der Download der Daten in ein maschinenlesbares Format (z.B. JSON oder XML) ist möglich.

**Betroffene Rollen:**

Vorfallsteammitglied

**Betroffene CSIRT-Funktion:**

Kommunikation

**Herleitung:**

Der Abschlussbericht ist einerseits der Tätigkeitsnachweis eines CSIRT, andererseits aber auch im Fall der Strafverfolgung notwendig. Da in dem System ohnehin alle Informationen zu dem Vorfall vorliegen, ist es sinnvoll, diese Daten auch als Grundlage für den Abschlussbericht heranzuziehen.

**Quelle:**

[6, S. 39]

#### Userstory 2 (Aufbereitung des Vorfalls für Externe)

**Beschreibung:**

„Als Vorfallsteammitglied kann ich eine Zusammenfassung des Vorfalls für Externe eintragen. Diese Zusammenfassung ist dazu gedacht, den Vorfall allgemeinverständlich zusammenzufassen und die nächsten Schritte zu beschreiben. Externe können somit über die Entwicklungen auf dem Laufenden gehalten werden. Habe ich die Beschreibung, die Zusammenfassung und die kommenden Schritte beschrieben, kann ich diese Informationen exportieren (z.B. als PDF) oder per E-Mail verschicken.“

**Akzeptanzkriterium:**



Es gibt drei Textfelder (zur allgemeinverständlichen Beschreibung, zur Zusammenfassung der letzten Schritte und zur Zusammenfassung der kommenden Schritte). In diesen Feldern kann formatierter Text eingetragen werden, um die einzelnen Informationen bereitzustellen. Anschließend können diese Informationen exportiert oder aber direkt verschickt werden.

**Betroffene Rollen:**

Vorfallsteammitglied

**Betroffene CSIRT-Funktion:**

Kommunikation

**Herleitung:**

Die Kommunikation an externe Personen (extern im Sinne von außerhalb des Vorfallsteams) ist eine ständige Aufgabe eines CSIRT. Insbesondere Mitarbeitende des Managements und andere Personen in der Organisation (z.B. Kommunikationsabteilung) müssen über den Vorfall bei entsprechender Relevanz informiert werden. Da es sich dabei nicht unbedingt um technisches Personal handelt, sollte eine auf diese Zielgruppe angepasste Information bereitgestellt werden.

**Quelle:**

[6, S. 10, S. 11, S. 16, S. 34]

#### Userstory 3 (Weitergabe von Threat-Sharing-Daten)

**Beschreibung:**

„Als Vorfallsteammitglied kann ich die bekannten IoC/Observables an die Threat-Sharing-Plattform weitergeben. Die Informationen können mit einer Einstufung (hinsichtlich der Vertraulichkeit) versehen werden, damit sie nicht unberechtigt weitergegeben werden.“

**Akzeptanzkriterium:**

Eine Integration mit einer Threat Sharing Plattform zur Abfrage von IoC ist möglich.

**Betroffene Rollen:**

Vorfallsteammitglied

**Betroffene CSIRT-Funktion:**

Kommunikation

**Herleitung:**

Um der Rolle von Threat-Sharing bei der Arbeit eines CSIRT gerecht zu werden, muss es die Möglichkeit geben, Informationen weiterzugeben oder auch aus Threat Intelligence Datenbanken abzufragen.

**Quelle:**

[6, S. 45, S. 48, S. 49, S. 50] und [7, S. 132, S. 143]

**Userstory 4 (Eingabe von Threat-Sharing-Daten)****Beschreibung:**

„Als Threat Sharing Plattform kann ich dem IT-System zur Vorfallsbearbeitung IoC oder Observables übermitteln. Das System fügt meine mit dem IoC verknüpften Information automatisch einem Vorfall hinzu, falls dort diese IoC bereits bekannt sind. Die weiteren IoC werden dann als externe IoC markiert, um sie von den eigenen gewonnenen IoC unterscheiden zu können.“

**Akzeptanzkriterium:**

Eine Schnittstelle für Threat Sharing Plattformen existiert, mit der diese eigenen Daten in das hier betrachtete System einspeisen können.

**Betroffene Rollen:**

Threat Sharing Plattform

**Betroffene CSIRT-Funktion:**

Kommunikation

**Herleitung:**

Um der Rolle von Threat-Sharing bei der Arbeit eines CSIRT gerecht zu werden, muss es die Möglichkeit geben, Informationen weiterzugeben.

**Quelle:**

[6, S. 50]

**Userstory 5 (Weitergabe von Analysetätigkeiten)****Beschreibung:**

„Als Vorfallsteammitglied möchte ich Artefakte zur Analyse an andere Teams weiterleiten können. Diese Artefakte werden über eine definierte Schnittstelle an eine andere Organisation weitergeleitet, von wo aus dann eine weitere Analyse

möglich ist.“

**Akzeptanzkriterium:**

Das System kann einen Vorfall an eine andere Organisation weitergeben.

**Betroffene Rollen:**

Vorfallsteammitglied

**Betroffene CSIRT-Funktion:**

Kommunikation

**Herleitung:**

Nicht alle Analysetätigkeiten können von dem eigenen CSIRT selbst durchgeführt werden. Daher ist es notwendig, dass Vorfälle bzw. zu analysierende Artefakte auch an andere, befreundete Teams weitergegeben werden können.

**Quelle:**

[7, S. 87]

#### Userstory 6 (Vorbereitung der Nachbesprechung)

**Beschreibung:**

„Als Vorfallsteammitglied möchte ich die Notizen, die mit dem Vermerk „Relevant für Nachbereitung“ erstellt wurden, in einer Übersicht angezeigt bekommen, um diese als Vorbereitung für die Nachbesprechung eines Vorfalls nutzen zu können.“

**Akzeptanzkriterium:**

Die zuvor als „Relevant für Nachbereitung“ markierten Notizen werden angezeigt.

**Betroffene Rollen:**

Vorfallsteammitglied

**Betroffene CSIRT-Funktion:**

Kommunikation

**Herleitung:**

Die Vorbereitung der Nachbesprechung ist Grundlage für die Verbesserung der Vorfallsbehandlung.

**Quelle:**

[6, S. 37]

#### Userstory 7 (Pflegen von Vorfallsteammitgliedern)

**Beschreibung:**

„Als Vorfallsteamleiterin möchte ich die Mitglieder meines Teams im IT-System zur Vorfallsbearbeitung eintragen können. Dabei sollen sowohl ihre Rollen (Vorfallsteamleiterin, Vorfallsteammitglied) und ihre Kontaktdaten, als auch Zugangsdaten für das System (Benutzername, Passwort) hinterlegt werden.“

**Akzeptanzkriterium:**

Nachdem ein neues Teammitglied angelegt wurde, erscheint es mit seinen Daten (Name, Kontaktdaten, Rollen) in der Liste der Teammitglieder. Zudem kann sich das neue Teammitglied in dem System anmelden.

**Betroffene Rollen:**

Vorfallsteamleiterin

**Betroffene CSIRT-Funktion:**

Steuerung

**Herleitung:**

Einerseits ist vorgesehen, dass die Teammitglieder des Vorfallsteams erreichbar sind, wofür entsprechende Kontaktdaten hinterlegt werden müssen. Andererseits sind einige der vorfallsbezogenen Daten so sensibel, dass der Zugang zu dem System nur einem beschränkten Nutzerkreis zur Verfügung gestellt werden darf.

**Quelle:**

[6, S. 8, S. 22]

#### Userstory 8 (Einschränkung des Zugriffs auf einen Vorfall)

**Beschreibung:**

„Als Vorfallsteamleiterin und Vorfallsteammitglied kann ich Vorfälle als lesebeschränkt markieren, sodass sie nur noch für Vorfallsteamleiterinnen angezeigt werden sowie für Personen, die dem Vorfall als Mitarbeiterin zugeordnet sind.“

**Akzeptanzkriterium:**

Lesebeschränkte Vorfälle werden für unberechtigte Personen nicht mehr in der Liste der Vorfälle angezeigt und können auch nicht mehr manuell ohne Berechtigung aufgerufen werden.

**Betroffene Rollen:**

Vorfallsteamleiterin und Vorfallsteammitglied

**Betroffene CSIRT-Funktion:**

Steuerung

**Herleitung:**

Da Informationen zu Vorfällen sensible Daten enthalten können, kann es notwendig werden, den Zugriff auf bestimmte Vorfälle einzuschränken.

**Quelle:**

[6, S. 31]

#### Userstory 9 (Automatische Eskalation von Vorfällen)

**Beschreibung:**

„Als Vorfallsteamleiterin möchte ich einen Zeitraum festlegen, innerhalb dessen Vorfälle bearbeitet (d.h. triagiert) werden müssen. Wird ein Vorfall nicht innerhalb dieses Zeitraums initial bearbeitet, erhalte ich eine Benachrichtigung.“

**Akzeptanzkriterium:**

Nach Ablauf der Frist bei einem neuen Vorfall werden in dem System registrierte Vorfallsteamleiterinnen (z.B. per E-Mail) benachrichtigt.

**Betroffene Rollen:**

Vorfallsteamleiterin

**Betroffene CSIRT-Funktion:**

Steuerung

**Herleitung:**

Für den Fall, dass Vorfälle für einen gewissen Zeitraum nicht bearbeitet wurden, sollte die Teamleiterin darüber informiert werden, damit sie ggf. Gegenmaßnahmen treffen kann.

**Quelle:**

[6, S. 16, S. 33]

#### Userstory 10 (Anzeige von Metriken)

**Beschreibung:**

„Als Vorfallsteamleiterin kann ich mir Metriken zu den Aktivitäten in dem System anzeigen lassen: Nummer der bearbeiteten Vorfälle innerhalb eines Zeitfensters,

die Zeit pro Vorfall (Meldung bis Abschließen), die Anzahl der beteiligten Analystinnen, die Arten der Artefakte, die Dauer der einzelnen Analysen pro Artefakt“

**Akzeptanzkriterium:**

Die o.g. Metriken werden der Vorfallsteamleiterin angezeigt, auch wenn sie in le-sebeschränkten Vorfällen aufgetreten sind.

**Betroffene Rollen:**

Vorfallsteamleiterin

**Betroffene CSIRT-Funktion:**

Steuerung

**Herleitung:**

Zur Weiterentwicklung des CSIRT ist es hilfreich, Kennzahlen über die Schwerpunkte der Arbeit des CSIRTs zu erhalten. Diese Kennzahlen werden aus der regulären Bearbeitung der Vorfälle gewonnen.

**Quelle:**

[6, S. 40] und [7, S. 80, S. 81]

#### Userstory 11 (Zugriffsbeschränkung der Anwendung)

**Beschreibung:**

„Als Vorfallsteammitglied kann ich auf das System nur zugreifen, wenn ich mich mit einem gültigen Benutzerkonto angemeldet habe (siehe Userstory 7, Pflegen von Vorfallsteammitgliedern)“

**Akzeptanzkriterium:**

Zugriff auf das System, mit Ausnahme des Login-Formulars, ist ohne Benutzerkonto nicht möglich.

**Betroffene Rollen:**

Vorfallsteammitglied

**Betroffene CSIRT-Funktion:**

Steuerung

**Herleitung:**

Einige der vorfallsbezogenen Daten sind so sensibel, dass der Zugang zu dem System nur einem beschränkten Nutzerkreis zur Verfügung gestellt werden darf.

**Quelle:**

[6, S. 31]

**Userstory 12 (Anlegen eines Vorfalls)****Beschreibung:**

„Als Detektionssystem möchte ich IT-Sicherheitsvorfälle melden können, die dann als vermutlicher IT-Sicherheitsvorfall aufgenommen werden. Diese noch nicht eingeschätzten Vorfälle werden der Triage-Queue (siehe Userstory 17, Triage-Queue) zugeordnet.“

**Akzeptanzkriterium:**

Nachdem ein IT-Sicherheitsvorfall gemeldet wurde, ist dieser in dem System sichtbar. Der gemeldete IT-Sicherheitsvorfall enthält neben der Quelle auch die inhaltlichen Informationen, die zur Meldung des Vorfalls geführt haben sowie, falls anwendbar, die bereits erkannten Artefakte.

**Betroffene Rollen:**

Detektionssystem

**Betroffene CSIRT-Funktion:**

Triage

**Herleitung:**

Die Erstellung eines Vorfalls ist der Beginn des IT-Sicherheitsvorfallsreaktionsprozesses. Da auch fortgeschrittene Detektionssysteme (unabhängig ob menschlich oder technisch) Meldungen erzeugen können, die als Falsch-Positiv bewertet werden müssen, sind neue Meldungen zunächst als Verdachtsfall zu dokumentieren.

**Quelle:**

[6, S. 29, S. 44]

**Userstory 13 (Initiale Beurteilung eines Vorfalls (Triage))****Beschreibung:**

„Als Vorfallsteammitglied möchte ich einen Vorfall beurteilen können. Hierzu muss ich die Auswirkung des Vorfalls auf die Funktion des Unternehmens und auf die Informationen des Unternehmens vermerken können. Zudem muss ich die

Wiederherstellbarkeit (des Normalzustands) angeben können. Die Kombination dieser Werte (Auswirkungen und Wiederherstellbarkeit) ergibt die Priorität des Vorfalls. Um sicher zu stellen, dass nur verifizierte IT-Sicherheitsvorfälle weiter bearbeitet werden, kann ich einen Vorfall als falsifiziert markieren, wenn dieser sich als False-Positive herausstellt.“

**Akzeptanzkriterium:**

Wird die Auswirkung mit dem Maß der Wiederherstellbarkeit kombiniert, ergibt dies die Priorität des Vorfalls. Ein weiteres Feld steht zur Markierung des Vorfalls entweder als „Verifizierte Meldung“ oder als „Falsifizierte Meldung“ zur Verfügung.

**Betroffene Rollen:**

Vorfallsteammitglied

**Betroffene CSIRT-Funktion:**

Triage

**Herleitung:**

Da insbesondere bei beschränkten Bearbeitungskapazitäten nicht alle IT-Sicherheitsvorfälle in der gleichen Intensität bearbeitet werden können, muss eine Priorisierung vorgenommen werden, um die Vorfälle nach Priorität abarbeiten zu können. Ohne diese Funktion würden evtl. wichtige IT-Sicherheitsvorfälle nicht fristgerecht bearbeitet werden können.

**Quelle:**

[6, S. 32, S. 33] und [7, S. 69, S. 66, S. 70]

#### Userstory 14 (Zuordnen von Vorfallsteammitgliedern zu einem Vorfall)

**Beschreibung:**

„Als Vorfallsteammitglied kann ich einem oder mehreren Vorfällen zugeordnet werden bzw. mich selbst zuordnen, falls diese nicht sichtbarkeitsbeschränkt (siehe Userstory 8, Einschränkung des Zugriffs auf Vorfälle) sind.“

**Akzeptanzkriterium:**

Vorfallsteammitglieder können zugeordnet werden. Nachdem sie zugeordnet wurden, wird dies im Vorfall dokumentiert. Außerdem können zu einem Vorfall zugeordnete Mitglieder auf den Vorfall zugreifen, selbst wenn dieser zugriffsbe-



schränkt sein sollte.

**Betroffene Rollen:**

Vorfallsteammitglied

**Betroffene CSIRT-Funktion:**

Triage

**Herleitung:**

Die Bearbeitung von Vorfällen erfolgt je nach Komplexität eines Vorfalls von einem oder mehreren Personen. Die Zuordnung zu einem Vorfall sorgt dafür, dass die Person die jeweiligen Aufgaben in dem Vorfall wahrnehmen kann, selbst wenn der Vorfall zugriffsbeschränkt sein sollte.

**Quelle:**

[6, S. 13]

#### Userstory 15 (Verknüpfung von Vorfällen)

**Beschreibung:**

„Als zu einem Vorfall zugeordnetes Teammitglied oder als Vorfallsteamleiterin kann ich Vorfälle miteinander verknüpfen.“

**Akzeptanzkriterium:**

Bei verknüpften Vorfällen werden die anderen (verknüpften Vorfälle) im jeweils anderen Vorfall angezeigt. Zudem erhalten alle zugeordneten Teammitglieder lesenden Zugriff auf den anderen Vorfall.

**Betroffene Rollen:**

Vorfallsteammitglied

**Betroffene CSIRT-Funktion:**

Triage

**Herleitung:**

Es ist möglich, dass sich zwei oder mehrere unabhängig voneinander gemeldete Vorfälle als zusammengehörig herausstellen. In diesem Fall müssen die Vorfälle miteinander verbunden werden, damit sie gemeinsam bearbeitet werden können.

**Quelle:**

[6, S. 31] und [7, S. 70, S. 90]

#### Userstory 16 (Manuelle Eskalation eines Vorfalls)

**Beschreibung:**

„Als Vorfallsteamleiterin möchte ich Vorfälle unabhängig von ihrer zuvor festgelegten Priorität (siehe Userstory 13, Initiale Beurteilung des Vorfalls (Triage)) eskalieren können, sodass die Priorität für die Vorfallsteammitglieder höher angezeigt wird als sie eigentlich laut der Triage wäre. Diese Eskalation soll auch als solche im Vorfall angezeigt werden.“

**Akzeptanzkriterium:**

Die Vorfallsteamleiterin kann die Priorität von Vorfällen festlegen bzw. anpassen. Dies wird im Protokoll angezeigt.

**Betroffene Rollen:**

Vorfallsteamleiterin

**Betroffene CSIRT-Funktion:**

Triage

**Herleitung:**

Wie in den Standards dargestellt, kann es nötig sein, gewisse Vorfälle managementseitig zu eskalieren. Dies ist zum Beispiel notwendig, wenn bestimmte Informationen vorliegen, die bei der ursprünglichen Triage noch nicht vorlagen, bzw. weil aus Leitungssicht bestimmte Prioritäten anders zu legen sind.

**Quelle:**

[6, S. 16] und [7, S. 47, S. 66, S. 128, S. 129, S. 130]

#### Userstory 17 (Triage-Queue)

**Beschreibung:**

„Als Vorfallsteamleiterin möchte ich, dass Vorfälle, die noch nicht triagiert wurden (siehe Userstory 13, Initiale Beurteilung eines Vorfalls (Triage)) immer von einer vorher definierten Gruppe von Vorfallsteammitgliedern bearbeitet werden können.“

**Akzeptanzkriterium:**

Neue Vorfälle werden automatisch in einem separaten Bereich angezeigt, sodass die für die Triage vorgesehenen Teammitglieder diese Ersteinschätzung durchführen können.

**Betroffene Rollen:**

Vorfallsteamleiterin

**Betroffene CSIRT-Funktion:**

Triage

**Herleitung:**

Die Triage ist der Eingangskanal in den Dienst IT-Sicherheitsvorfallsbearbeitung. Daher muss sichergestellt werden, dass alle neuen IT-Sicherheitsvorfälle zunächst auf ihre Validität hin untersucht werden.

**Quelle:**

[7, S. 26, S. 69]

#### Userstory 18 (Benachrichtigung bei Zuordnung zu einem Vorfall)

**Beschreibung:**

„Werde ich als Vorfallsteammitglied einem Vorfall zugeordnet, erhalte ich darüber eine Benachrichtigung (z.B. per Mail).“

**Akzeptanzkriterium:**

Die Benachrichtigung über die Zuordnung erfolgt nach der Zuordnung nur, wenn sie nicht durch das Team Mitglied selbst durchgeführt wurde.

**Betroffene Rollen:**

Vorfallsteammitglied

**Betroffene CSIRT-Funktion:**

Vorfallsbearbeitung

**Herleitung:**

Um eine zeitnahe Bearbeitung der Vorfälle zu gewährleisten, muss es möglich sein, dass sich Teammitglieder über neu eingetroffene Vorfälle informieren lassen.

**Quelle:**

[6, S. 13, S. 22]

#### Userstory 19 (Benachrichtigung bei neuen Vorfällen)

**Beschreibung:**

„Als Vorfallsteammitglied kann ich einstellen, bei welchen eingetroffenen Kriterien ich eine Benachrichtigung über neue IT-Sicherheitsvorfälle erhalte. Wird ein

neuer Vorfall angelegt, der den Kriterien entspricht, erhalte ich eine Benachrichtigung (z.B. per E-Mail) und kann den Vorfall aufrufen.“

**Akzeptanzkriterium:**

Vorfallsteammitglieder erhalten bei Erfüllung der festgelegten Kriterien eine Benachrichtigung (z.B. per E-Mail), wenn ein neuer Vorfall im System aufgenommen wurde.

**Betroffene Rollen:**

Vorfallsteammitglied

**Betroffene CSIRT-Funktion:**

Vorfallsbearbeitung

**Herleitung:**

Um eine zeitnahe Bearbeitung der Vorfälle zu gewährleisten, muss es möglich sein, dass sich Teammitglieder über neu eingetroffene Vorfälle informieren lassen.

**Quelle:**

[6, S. 33] und [7, S. 43]

#### Userstory 20 (Protokollierung von Aktionen)

**Beschreibung:**

„Als Vorfallsteammitglied werden alle Aktionen, die ich innerhalb eines IT-Sicherheitsvorfalls durchführe (z.B. Statusänderungen, Analysen, Zuordnung von Teammitgliedern) in einem dem Vorfall zugehörigen Aktivitätsprotokoll dokumentiert.“

**Akzeptanzkriterium:**

Durchgeführte Aktionen sind mit der jeweiligen ausführenden Person im Zeitstrahl des Vorfalls (siehe Userstory 30, Anzeige des Vorfallszeitstrahls) sichtbar.

**Betroffene Rollen:**

Vorfallsteammitglied

**Betroffene CSIRT-Funktion:**

Vorfallsbearbeitung

**Herleitung:**

Sämtliche Aktionen, die im Zusammenhang mit dem Vorfall durchgeführt wer-

den, müssen protokolliert werden, um im Nachhinein möglichst einfach einen Bericht zum Vorfall erstellen zu können. Insbesondere im Zusammenhang mit den Automatisierungsfunktionen ist es notwendig, nachvollziehen zu können, welche Tätigkeiten im Laufe des Vorfalls durchgeführt wurden.

**Quelle:**

[6, S. 30, S. 31, S. 38]

#### Userstory 21 (Bearbeiten von Stammdaten eines Vorfalls)

**Beschreibung:**

„Als Vorfallsteammitglied kann ich folgende Felder (Stammdaten) des Vorfalls bearbeiten: Titel, Aktueller Status, Zusammenfassung, Tags, Einschätzung der Kritikalität (siehe Userstory 13, Initiale Beurteilung des Vorfalls (Triage))“

**Akzeptanzkriterium:**

Die o.g. Felder sind für zugeordnete Teammitglieder und für die Vorfallsteamleiterin bearbeitbar.

**Betroffene Rollen:**

Vorfallsteammitglied

**Betroffene CSIRT-Funktion:**

Vorfallsbearbeitung

**Herleitung:**

Um einen Vorfall zu identifizieren, benötigt dieser gewisse Stammdaten. Hierzu zählen unter anderem der Titel und der Status, sowie die Kritikalität. Da sich diese Informationen im Laufe des Prozesses ändern können, muss es möglich sein, diese Daten zu ändern.

**Quelle:**

[6, S. 31] und [7, S. 78]

#### Userstory 22 (Anzeigen eines Vorfalls)

**Beschreibung:**

„Als Vorfallsteammitglied kann ich die Daten eines IT-Sicherheitsvorfalls anzeigen lassen, außer der Zugriff ist beschränkt (siehe Userstory 8, Einschränkung des Zugriffs auf Vorfälle): Eindeutige Nummer des Vorfalls, Tags zu dem Vorfall,

Kontaktdaten von Verantwortlichen, Titel, Quelle, Aktueller Status, Zeitpunkt der Erstellung, Zusammenfassung, Zugeordnete Teammitglieder, Aktivitätsprotokoll (siehe Userstory 30, Anzeige des Vorfallszeitstrahls), Einschätzung der Kritikalität (siehe Userstory 13, Initiale Beurteilung des Vorfalls (Triage)), Zusätzliche Dokumente (siehe Userstory 28, Verweisen auf weitere Dokumente), Observables / Indicators of Compromise (siehe Userstory 23 Aufnehmen von Observables / IoC), Artefakte (siehe Userstory 24, Hinzufügen von Artefakten), hinterlegte SOP (siehe Userstory 26 Definieren von SOP)“

**Akzeptanzkriterium:**

Die o.g. Datenfelder sind mit den jeweiligen Daten befüllt und werden korrekt, d.h. mit den für den Vorfall eingetragenen Werten angezeigt.

**Betroffene Rollen:**

Vorfallsteammitglied

**Betroffene CSIRT-Funktion:**

Vorfallsbearbeitung

**Herleitung:**

Die Darstellung eines Vorfalls ist notwendig, um sich die Informationen zum Vorfall anzeigen lassen zu können, bzw. um einen Einstiegspunkt in die weiteren Aktivitäten bezogen auf einen Vorfall zu bieten.

**Quelle:**

[6, S. 31] und [7, S. 71, S. 92]

#### Userstory 23 (Hinzufügen von Observables / IoC)

**Beschreibung:**

„Als Vorfallsteammitglied möchte ich Observables oder IoC für den IT-Sicherheitsvorfall aufnehmen. Diese IoC sollen anschließend in den Daten des jeweiligen Vorfalls hinterlegt werden. Dabei kann ich markieren, ob es sich bei den Observable um einen IoC handelt.“

**Akzeptanzkriterium:**

Die möglichen Observables / IoC, die eingetragen werden können, entsprechen den üblichen Datentypen, die in Threat-Sharing-Systemen (z. B. MISP) verarbeitet werden können.

**Betroffene Rollen:**

Vorfallsteammitglied

**Betroffene CSIRT-Funktion:**

Vorfallsbearbeitung

**Herleitung:**

Observables bzw. in der spezielleren Form IoC sind ein wichtiger Faktor in der Bearbeitung des Vorfalls, da durch diese die Angreiferin identifiziert und möglicherweise auch immer wieder erkannt werden kann. Daher ist es nötig, dass alle Personen, die an einem Vorfall arbeiten, über die aktuell bekannten IoC und Observables informiert sind.

**Quelle:**

[6, S. 27, S. 29, S. 31, S. 45, S. 47, S. 48, S. 49, S. 50] und [7, S. 85]

#### Userstory 24 (Hinzufügen von Artefakten)

**Beschreibung:**

„Als Vorfallsteammitglied möchte ich auf betroffenen IT-Systemen gefundene Artefakte in die das hier beschriebene System laden können, damit diese zentral zur Analyse zur Verfügung stehen. Ich kann das Artefakt anderen Teammitgliedern zuordnen, damit diese wissen, dass sie das Artefakt analysieren sollen. Zudem ist es notwendig, die Quelle des Artefakts angeben zu können.“

**Akzeptanzkriterium:**

Artefakte werden mit ihren Metadaten als zum jeweiligen Vorfall zugehörig angezeigt. Neben dem eigentlichen Artefakt wird auch die Quelle angezeigt.

**Betroffene Rollen:**

Vorfallsteammitglied

**Betroffene CSIRT-Funktion:**

Vorfallsbearbeitung

**Herleitung:**

Artefakte, die aufgrund von Angriffsverhalten entstanden sind, bieten weitere Informationsquellen für die Ableitung von IoC/Observables, aber auch zur weiteren Identifizierung von betroffenen Systemen.

**Quelle:**

[6, S. 23, S. 29, S. 31]

**Userstory 25 (Automatische Analyse von Artefakten)****Beschreibung:**

„Als Vorfallsteammitglied möchte ich, dass Artefakte von dazu passenden Analysetools automatisch analysiert werden können. Dazu soll es möglich sein, für bestimmte Typen von Artefakten Analysemethoden zu hinterlegen, die von dem System automatisch ausgeführt werden. Bei den Analysemethoden kann es sich um ausgelagerte Dienste handeln, die z.B. über Plugins für das System angesprochen werden können.“

**Akzeptanzkriterium:**

Es besteht eine Schnittstelle, die es ermöglicht, externe Analysesysteme an das Sicherheitsvorfallsmanagementsystem anzubinden. Diese Analysesysteme werden vom System automatisch aufgerufen, wenn sie erkennen, dass ein zu einem Artefakt passendes Analysesystem angebunden ist. Das Ergebnis der Analyse wird entweder vergleichbar zu einem Kommentar eines Vorfallsteammitglied eingetragen, oder falls zusätzliche Indicators of Compromise / Observables identifiziert wurden als solche dem Vorfall hinzugefügt.

**Betroffene Rollen:**

Vorfallsteammitglied

**Betroffene CSIRT-Funktion:**

Vorfallsbearbeitung

**Herleitung:**

Die Analyse von Artefakten ist ein unmittelbar mit der Bearbeitung von Vorfällen verbundener Schritt. Hier wird viel Zeit und Expertise eingesetzt, um die Details aus den Artefakten zu extrahieren. Um diesen Schritt wo möglich zu parallelisieren und zu automatisieren, muss das System eine Unterstützung hierfür anbieten.

**Quelle:**

[6, S. 29, S. 28] und [7, S. 26, S. 29, S. 85]



#### Userstory 26 (Definieren von SOP)

**Beschreibung:**

„Als Vorfallsteammitglied möchte ich für jede Phase eines Vorfalls (Detektion und Analyse sowie Eindämmung, Behebung und Wiederherstellung) standardisierte Arbeitsschritte (SOP) eintragen können, die bei einem Vorfall angezeigt werden, wenn dieser zuvor festgelegte Parameter erfüllt. Innerhalb eines Vorfalls kann ich vermerken, ob ich die vorher definierten Schritte abgearbeitet habe, bzw. ob sie in diesem Fall nicht zutreffend waren.“

**Akzeptanzkriterium:**

Es besteht eine Möglichkeit, innerhalb des Systems vorher definierte Maßnahmen-schritte zu dokumentieren und festzulegen, bei welchen eingetretenen Kriterien diese Hinweise in einem Vorfall angezeigt werden.

**Betroffene Rollen:**

Vorfallsteammitglied

**Betroffene CSIRT-Funktion:**

Vorfallsbearbeitung

**Herleitung:**

Bestimmte IT-Sicherheitsvorfälle werden immer wieder nach dem gleichen Schema bearbeitet. Daher sollten diese Schemata (Standard Operating Procedures) auch in dem System hinterlegt werden können. Einerseits als Unterstützung für die Vorfallsteammitglieder, andererseits auch als Grundlage für die Automatisierung von (Teil-)Schritten.

**Quelle:**

[6, S. 9, S. 35, S. 37]

#### Userstory 27 (Automatisches Ausführen von SOP)

**Beschreibung:**

„Als Vorfallsteammitglied möchte ich, dass bestimmte, vorab definierte Aktionen ausgeführt werden, wenn ein Vorfall über bestimmte, vorher festgelegte Attribute verfügt. Hierzu soll es eine Schnittstelle geben, damit die Menge der zur Verfügung stehenden Aktionen erweitert werden kann. Die Beschreibung, in welchem

Fall welche Aktion durchgeführt werden soll, sollte durch eine Skriptsprache anpassbar sein.“

**Akzeptanzkriterium:**

Es besteht eine Schnittstelle, mit der Reaktionsschritte ausgelöst werden können. Diese Schnittstelle kann zusätzliche Plugins ansprechen, die dann wiederum die eigentliche Aktion (z.B. blockieren einer IP-Adresse in einer Firewall) ausführt. Im Frontend ist es für Vorfallsteammitglieder möglich, je nachdem welche Kriterien von dem Vorfall erfüllt sind, bestimmte Aktionen auszuführen. Dabei ist es auch möglich Blöcke von Aktionen durchzuführen, bzw. weitere Aktionen von den Ergebnissen anderer Aktionen abhängig zu machen.

**Betroffene Rollen:**

Vorfallsteammitglied

**Betroffene CSIRT-Funktion:**

Vorfallsbearbeitung

**Herleitung:**

Gleichartige IT-Sicherheitsvorfälle bzw. Teile von ihnen werden immer wieder nach dem gleichen Schema bearbeitet. Insofern ist es sinnvoll, diese Bearbeitung auch zu automatisieren. Da es nicht von vornherein absehbar ist, welche Automatisierungsmöglichkeiten benötigt werden und sich IT-Infrastrukturen auch untereinander sehr unterscheiden können, ist hier das Angebot einer Schnittstelle zur Implementation von eigenen Reaktionsschritten notwendig.

**Quelle:**

[6, S. 9, S. 35, S. 37]

#### Userstory 28 (Verweis auf weitere Dokumente)

**Beschreibung:**

„Als Vorfallsteammitglied, das einem Vorfall zugeordnet ist, kann ich weitere Dokumente (z.B. Dokumentationen von forensischen Datensicherungen oder Berichte über vergleichbare IT-Sicherheitsvorfälle) hinzufügen.“

**Akzeptanzkriterium:**

Die hinzugefügten Dokumente sind in der Ansicht für den Vorfall sichtbar und können aufgerufen werden.

**Betroffene Rollen:**

Vorfallsteammitglied

**Betroffene CSIRT-Funktion:**

Vorfallsbearbeitung

**Herleitung:**

Das System kann nicht jeden Anwendungsfall für kollaboratives Arbeiten oder Dokumentation abbilden. Insbesondere sollte keine Parallelstruktur zu bestehenden Systemen (z.B. Dokumentenmanagement oder Wissensmanagement) geschaffen werden. Um trotzdem auf dortige Quellen verweisen zu können, ist es nötig, dass es eine Möglichkeit gibt, auf andere Dokumente (z.B. per Link) verweisen zu können.

**Quelle:**

[6, S. 31]

#### Userstory 29 (Auflistung von Vorfällen)

**Beschreibung:**

„Als Vorfallsteammitglied möchte ich mir Vorfälle nach verschiedenen Kriterien auflisten lassen können, z.B. um zu prüfen, welchen Vorfall ich als nächstes bearbeiten muss. In einer solchen Liste müssen mindestens folgende Werte angezeigt werden: eindeutige Nummer des Vorfalls, Titel des Vorfalls, Zeitpunkt der Erstellung, Tags, Status. Vorfälle, die in der Sichtbarkeit eingeschränkt sind und auf die ich nicht zugreifen kann (siehe Userstory 8, Einschränkung des Zugriffs auf Vorfälle) werden mir nicht angezeigt“

**Akzeptanzkriterium:**

Das System verfügt über eine Seite, in der die Vorfälle entsprechend der angegebenen Kriterien als Liste angezeigt werden können.

**Betroffene Rollen:**

Vorfallsteammitglied

**Betroffene CSIRT-Funktion:**

Vorfallsbearbeitung

**Herleitung:**

Die Auflistung der Vorfälle soll einen Überblick über die im System aufgenom-

menen Vorfälle liefern. Außerdem dient sie als Eingangsportal in einen einzelnen Vorfall. Zur Reduzierung von Suchaufwand wird hierzu auch eine Suche implementiert.

**Quelle:**

[7, S. 43]

#### Userstory 30 (Anzeige des Vorfallszeitstrahls)

**Beschreibung:**

„Als Vorfallsteammitglied möchte ich mir einen Zeitstrahl des IT-Sicherheitsvorfalls anzeigen lassen können. In diesem Zeitstrahl werden die Kommentare (siehe Userstory 31, Dokumentation der Vorfallsbearbeitung) sowie die durchgeführten Tätigkeiten (siehe Userstory 20, Protokollierung der Aktionen) in einem Zeitstrahl angezeigt.“

**Akzeptanzkriterium:**

Im Vorfallszeitstrahl werden alle Informationen angezeigt, die über einen zeitlichen Marker verfügen. Dabei wird die ausführende Person (falls möglich) ebenso angezeigt.

**Betroffene Rollen:**

Vorfallsteammitglied

**Betroffene CSIRT-Funktion:**

Vorfallsbearbeitung

**Herleitung:**

Der Zeitstrahl ist beim Verständnis eines IT-Sicherheitsvorfalls von hoher Bedeutung. Hier lässt sich nachvollziehen, welche Aktivitäten der Angreiferin zu welchem Zeitpunkt stattgefunden haben (z.B. auch Lateral Movement). Gleichzeitig lässt sich hier auch die eigentliche Behandlung des Vorfalls (die weit nach der eigentlichen Angreiferaktivität liegen kann) nachvollziehen.

**Quelle:**

[6, S. 37]

#### Userstory 31 (Dokumentation der Vorfallsbearbeitung)

**Beschreibung:**

„Als Vorfallsteammitglied kann ich innerhalb des Vorfalls Kommentare eintragen. Zu jedem Kommentar kann ich manuell einen Zeitstempel angeben, der anzeigt, zu welchem (historischen) Zeitpunkt eine Aktion der Angreiferin stattgefunden hat. Wenn ich markieren möchte, dass eine Information für die Nachbesprechung relevant ist, kann ich das mit der Auswahl einer Eigenschaft „Relevant für die Nachbereitung“ markieren.“

**Akzeptanzkriterium:**

Es existiert für jeden Vorfall eine Sektion, in der ein dem Vorfall zugeordnetes Vorfallsteammitglied die Schritte der Vorfallsbearbeitung dokumentieren kann. Die eingegebenen Informationen werden bei der Anzeige des Vorfalls im zeitlichen Verlauf (siehe Userstory 30, Anzeige des Vorfallszeitstrahls) angezeigt. Auch die Möglichkeit der Markierung als „Relevant für die Nachbereitung“ ist möglich.

**Betroffene Rollen:**

Vorfallsteammitglied

**Betroffene CSIRT-Funktion:**

Vorfallsbearbeitung

**Herleitung:**

Die Schritte zur Vorfallsbearbeitung müssen dokumentiert werden, einerseits um nachvollziehen zu können, welche Aktivitäten der Angreiferin zu welchem Zeitpunkt identifiziert wurden, andererseits auch, um im Nachhinein sagen zu können, welche Aktionen von der Angreiferin und welche vom Vorfallsreaktionsteam durchgeführt wurden.

**Quelle:**

[6, S. 37]

#### Userstory 32 (Bearbeiten von Vorfällen anhand von SOP)

**Beschreibung:**

„Als Vorfallsteammitglied möchte ich markieren können, dass ich vordefinierte Schritte der SOP (siehe Userstory 26, Definieren von SOP) abgearbeitet habe.

Dazu kann ich in einer zugeordneten SOP anklicken, dass ich einen Schritt ausgeführt habe. Dies wird auch im Zeitstrahl des Vorfalls angezeigt.“

**Akzeptanzkriterium:**

Die durchgeführten SOP-Schritte werden im Vorfallszeitstrahl (siehe Userstory 30, Anzeige des Vorfallszeitstrahls) angezeigt und können nicht erneut als durchgeführt markiert werden.

**Betroffene Rollen:**

Vorfallsteammitglied

**Betroffene CSIRT-Funktion:**

Vorfallsbearbeitung

**Herleitung:**

Bestimmte IT-Sicherheitsvorfälle werden immer wieder nach dem gleichen Schema bearbeitet. Daher sollten diese Schemata (Standard Operating Procedures) auch in dem System hinterlegt werden können, einerseits als Unterstützung für den Vorfallsbearbeiter, andererseits auch als Grundlage für die Automatisierung von (Teil-)Schritten. Das Abarbeiten der SOPs muss dann auch von Vorfallsteammitgliedern dokumentiert werden können.

**Quelle:**

[6, S. 9, S. 35, S. 37]

## 4. Testszenarien und Testfälle

Nachdem im vorherigen Kapitel die ANFORDERUNGEN an ein System zur Automatisierung des IT-Sicherheitsvorfallsreaktionsprozesses erarbeitet wurden, wird im folgenden Schritt die Grundlage der Evaluierung von IT-Systemen zur Automatisierung der IT-Sicherheitsvorfallsreaktion gelegt: Durch das Beschreiben von Testszenarien und Testfällen werden die Evaluationskriterien beschrieben.

### 4.1. Testszenarien

Als Zwischenschritt zur Entwicklung der einzelnen Testfälle werden zunächst Testszenarien entwickelt. Dies dient zur Beschreibung von verschiedenen Arbeitsschritten und um zu verhindern, dass jeder Testfall einzeln für sich steht. Vielmehr soll durch die Gruppierung in Testszenarien der Arbeitsablauf in der Anwendung abgebildet werden, wohingegen jeder Testfall dann einen konkreten Arbeitsschritt umfasst. Die Beschreibung der Szenarien basiert hierbei nicht auf den anhand der CSIRT-Funktionen identifizierten Anforderungen, sondern wird eben anhand der verschiedenen Arbeitsabläufe gegliedert. Für jedes Szenario ist angegeben, welche Anforderungen damit überprüft werden. Da ein Testszenario auch mehrere Userstories überprüfen kann wird jeweils die Identifikationsnummer (ID) der Userstory, der Titel der Userstory und die von der Userstory betroffene Rolle angegeben.

#### 4.1.1. Benutzerverwaltung und -anmeldung

**Zweck des Szenario:** Durch dieses Szenario werden die Funktionen geprüft, die im Zusammenhang mit der Erstellung von Benutzerkonten und dem Zugang zu der Anwendung stehen. Zudem werden hier auch die Zugangsbeschränkungen zu einzelnen Fällen geprüft.

Tabelle 4.1.: Testszenario Benutzerverwaltung und -anmeldung: Abgedeckte Anforderungen

ID	Titel	Rollen
7	Pflegen von Vorfallsteammitgliedern	Vorfallsteamleiterin
11	Zugriffsbeschränkung der Anwendung	Vorfallsteammitglied

**Beschreibung des Szenario:** Die Vorfallsteamleiterin erstellt zwei Benutzerkonten für Vorfallsteammitglieder. Diese beiden Benutzerkonten sind jeweils in der Lage, sich mit den zuvor vergebenen Daten bei der Anwendung anzumelden. Eine Benutzung der Anwendung bzw. eine Anmeldung mit falschen Benutzerdaten ist nicht möglich.

### 4.1.2. Vorfallserstellung

**Zweck des Szenario:** Der gesamte Vorfallsreaktionsprozess dreht sich um die Bearbeitung von erkannten IT-Sicherheitsvorfällen. Bevor ein IT-Sicherheitsvorfall bearbeitet werden kann, muss dieser Vorfall jedoch zunächst angelegt und hinsichtlich der Kritikalität eingestuft werden. Das Szenario deckt die Anforderungen zum Anlegen des Vorfalls, der initialen Klassifikation des Vorfalls (die so genannte Triage), sowie die Funktionalitäten der Teamleitung zur Eskalation zur Verfügung.

Tabelle 4.2.: Testszenario Vorfallserstellung: Abgedeckte Anforderungen

ID	Titel	Rollen
9	Automatische Eskalation von Vorfällen	Vorfallsteamleiterin
12	Anlegen eines Vorfalls	Detektionssystem
13	Initiale Beurteilung eines Vorfalls (Triage)	Vorfallsteammitglied
16	Manuelle Eskalation eines Vorfalls	Vorfallsteamleiterin
17	Triage-Queue	Vorfallsteamleiterin
19	Benachrichtigung bei neuen Vorfällen	Vorfallsteammitglied
14	Zuordnung von Vorfallsteammitgliedern zu einem Vorfall	Vorfallsteammitglied
18	Benachrichtigung bei Zuordnung zu einem Vorfall	Vorfallsteammitglied



**Beschreibung des Szenario:** Durch ein Detektionssystem werden zwei neue Vorfälle erstellt. Über beide neue Vorfälle sollen die dafür registrierten Benutzerinnen eine Benachrichtigung erhalten. Einer der beiden Vorfälle wird von einem Vorfallsteammitglied initial beurteilt, in dem sie sich den Vorfall in der Triage-Queue auswählt. Nach der Einordnung werden dem Vorfall zwei Benutzerinnen als zuständige Teammitglieder zugeordnet. Diese beiden Benutzerinnen sollen über die Zuordnung informiert werden. Nach der Triage und der Zuordnung von Personen wird durch eine Testperson mit der Rolle Vorfallsteamleiterin der Vorfall manuell eskaliert. Nach Ablauf einer vorkonfigurierten Frist erhält die Vorfallsteamleiterin eine automatische Benachrichtigung über den zweiten Vorfall, der nicht bearbeitet wurde.

### 4.1.3. Vorfallsanzeige und -editierung

**Zweck des Szenario:** Die Anzeige der Benutzeroberfläche ist der Einstiegspunkt in die Bearbeitung. In dem hier erläuterten Szenario werden daher alle Anforderungen überprüft, die sich mit der eigentlichen Darstellung und Auflistung von Vorfällen und der Bearbeitung der Stammdaten beschäftigen.

Tabelle 4.3.: Testszenario Vorfallsanzeige und -editierung: Abgedeckte Anforderungen

ID	Titel	Rollen
8	Einschränkung des Zugriffs auf einen Vorfall	Vorfallsteamleiterin oder Vorfallsteammitglied
15	Verknüpfung von Vorfällen	Vorfallsteammitglied
21	Bearbeitung von Stammdaten eines Vorfalls	Vorfallsteammitglied
22	Anzeigen eines Vorfalls	Vorfallsteammitglied
29	Auflistung von Vorfällen	Vorfallsteammitglied

**Beschreibung des Szenario:** Als Vorbedingung müssen die Benutzerkonten und einige Vorfälle bereits im System vorhanden sein. Zunächst lässt sich eine Benutzerin in einer beliebigen Rolle die aktuell laufenden Vorfälle und die abgeschlossenen Vorfälle anzeigen. Aus der Liste der laufenden Vorfälle wird anschließend ein Vorfall ausgewählt, zu dem die Testperson zugeordnet ist. Der Vorfall wird mit seinen Stammdaten

angezeigt; allerdings werden die Stammdaten durch die Person im weiteren Verlauf verändert und gespeichert. Der Vorfall wird jetzt mit den aktualisierten Stammdaten angezeigt. Zum Abschluss des Szenario werden zwei Vorfälle miteinander verknüpft.

### 4.1.4. Vorfallsanalyse

**Zweck des Szenario:** Die Analyse ist der erste Schritt in der eigentlichen Bearbeitung eines Vorfalls und wird wie beschrieben im Rahmen der Vorfallsuntersuchung immer wieder durchlaufen. In diesem Testszenario werden daher alle Anforderungen überprüft, die sich auf die Analyse von Indicators of Compromise oder die Untersuchung von einzelnen Artefakten beziehen. Auch die Automatisierbarkeit der Analyse wird an dieser Stelle überprüft.

Tabelle 4.4.: Testszenario Vorfallsanalyse: Abgedeckte Anforderungen

ID	Titel	Rollen
5	Weitergabe von Analysetätigkeiten	Vorfallsteammitglied
23	Hinzufügen von Observables / IoC	Vorfallsteammitglied
24	Hinzufügen von Artefakten	Vorfallsteammitglied
25	Analyse von Artefakten	Vorfallsteammitglied

**Beschreibung des Szenario:** Als Vorbedingung müssen die Benutzerkonten und einige Vorfälle bereits im System vorhanden sein. In einem bestehenden Vorfall wird ein Artefakt (z.B. eine Sicherung des Arbeitsspeichers oder eine Datei) hinzugefügt. Desweiteren werden verschiedene IoC (Prüfsummen, Hostnames, IP-Adressen) in den Vorfall mit aufgenommen. Artefakte und IoC/Observables werden anschließend, veranlasst von der Testperson, automatisiert analysiert. Eines der Artefakte wird an eine externe Organisation zur weiteren Analyse weitergeleitet.

### 4.1.5. Vorfallsreaktion

**Zweck des Szenario:** Die Durchführung von Reaktionsmaßnahmen (also Eindämmung, Behebung und Wiederherstellung) ist bei der Vorfallsbearbeitung die zweite Stufe im sich wiederholenden Reaktionszyklus. Dieses Testszenario beinhaltet daher

die Bearbeitung von Vorfällen anhand von vorher definierten SOP, sowie die Automation dieser zuvor definierten Vorfallsreaktion.

Tabelle 4.5.: Testszenario Vorfallsreaktion: Abgedeckte Anforderungen

ID	Titel	Rollen
26	Definieren von SOP	Vorfallsteammitglied
27	Automatisches Ausführen von SOP	Vorfallsteammitglied
32	Bearbeiten von Vorfällen anhand von SOP	Vorfallsteammitglied

**Beschreibung des Szenario:** Als Vorbedingung müssen die Benutzerkonten und einige Vorfälle bereits im System vorhanden sein. In der Benutzeroberfläche für das System wird durch die Testperson eine neue SOP angelegt. Diese SOP wird einem neuen Vorfall zugeordnet. Anschließend wird eine weitere SOP angelegt, die automatisierte Komponenten enthält, die auf jeden Fall zutreffen werden (z.B. die reine Existenz eines Vorfalls und daran gekoppelt der Versand einer E-Mail). Anschließend wird ein neuer Vorfall erstellt und die neu erstellte, automatisierte SOP wird abgearbeitet.

##### 4.1.6. Threat-Sharing

**Zweck des Szenario:** Die Anforderungen, die sich aus der Notwendigkeit ergeben, Informationen zwischen Vorfallsreaktionsteams zu teilen, werden in diesem Testszenario überprüft. Dazu gehört sowohl das Abfragen von Threat-Intelligence-Datenbanken als auch das Aufnehmen von Meldungen aus diesen Datenbanken.

Tabelle 4.6.: Testszenario Threat-Sharing: Abgedeckte Anforderungen

ID	Titel	Rollen
3	Weitergabe von Threat-Sharing-Daten	Vorfallsteammitglied
4	Eingabe von Threat-Sharing-Daten	Threat-Sharing-Plattform

**Beschreibung des Szenario:** Als Vorbedingung müssen die Benutzerkonten und einige Vorfälle bereits im System vorhanden sein. Außerdem müssen in dem Vorfall bereits IoC aufgenommen worden sein. Die Testperson sendet die bestehenden IoC an eine Threat-Intelligence-Plattform. Daten, die in dieser Plattform bestehen, werden

mit den weiteren Informationen, die zu diesem Datensatz vorhanden sind angereichert und anschließend an das System zurückgegeben. Die Threat-Sharing-Plattform sendet zusätzlich IoC, die zum Teil bereits in einem Vorfall bekannt sind. Die weiteren Daten werden in dem Vorfall ergänzt.

### 4.1.7. Dokumentation

**Zweck des Szenario:** Alle zuvor beschriebenen Aktionen und Teilprozesse müssen im Rahmen des Vorfallsbearbeitungsprozesses dokumentiert werden. Die Erfüllung der damit verbundenen Anforderungen werden in diesem Szenario auf ihre Umsetzung hin überprüft.

Tabelle 4.7.: Testszenario Dokumentation: Abgedeckte Anforderungen

ID	Titel	Rollen
20	Protokollierung von Aktionen	Vorfallsteammitglied
28	Verweis auf weitere Dokumente	Vorfallsteammitglied
30	Anzeige des Vorfallszeitstrahls	Vorfallsteammitglied
31	Dokumentation der Vorfallsbearbeitung	Vorfallsteammitglied

**Beschreibung des Szenario:** Als Vorbedingung müssen die Benutzerkonten und einige Vorfälle bereits im System vorhanden sein. Zudem müssen in dem Vorfall die Aktionen aus den Szenarien „Vorfallsanalyse“ und „Vorfallsreaktion“ durchgeführt worden sein. In dem Vorfall erstellt die Testperson Kommentare und fügt Verweise zu weiteren Dokumenten an. Zudem werden Kommentare geschrieben, die die Reaktion auf den Vorfall beschreiben. Die Testperson kann sich anschließend alle im Vorfall stattgefundenen Aktionen in einem Zeitstrahl anzeigen.

### 4.1.8. Kommunikation

**Zweck des Szenario:** Dieses Szenario beinhaltet die Überprüfung der Anforderung, Information an externe Stellen (außerhalb des Reaktionsteams, also auch die betroffene Organisation) bereitzustellen.

**Beschreibung des Szenario:** Als Vorbedingung müssen die Benutzerkonten und einige Vorfälle bereits im System vorhanden sein. Die Testperson schreibt in der Rolle des

Tabelle 4.8.: Testszenario Kommunikation: Abgedeckte Anforderungen

ID	Titel	Rollen
2	Aufbereitung des Vorfalls für Externe	Vorfallsteammitglied

Vorfallsteammitglied eine Zusammenfassung über den Vorfall, den sie anschließend per E-Mail an eine externe Stelle verschicken kann.

### 4.1.9. Nachbereitung

**Zweck des Szenario:** Am Ende der Vorfallsbearbeitung steht die Durchführung von Nachbereitungsmaßnahmen, wie die Erstellung von Untersuchungsberichten, das Durchführen von Abschlussgesprächen und die Kontrolle der aufgewendeten Ressourcen in Form der Überprüfung von Kennzahlen.

Tabelle 4.9.: Testszenario Nachbereitung: Abgedeckte Anforderungen

ID	Titel	Rollen
1	Vorbereitung des Abschlussberichts	Vorfallsteammitglied
6	Vorbereitung der Nachbesprechung	Vorfallsteammitglied
10	Anzeige von Metriken	Vorfallsteamleiterin

**Beschreibung des Szenario:** Als Vorbedingung müssen die Benutzerkonten und einige Vorfälle bereits im System vorhanden sein. Die Testperson lädt sich in der Rolle des Vorfallsteammitglieds alle Informationen zu einem Vorfall herunter. Anschließend exportiert sie sich die Kommentare, die als „Relevant für die Nachbesprechung“ markiert wurden. Die Vorfallsteamleiterin lässt sich die Metriken zu den abgeschlossenen Vorfällen anzeigen.

## 4.2. Testfälle

Die Beschreibung der Testfälle basiert auf der oben aufgezeigten Herleitung der Szenarien. Den jeweiligen Szenarien sind hier die eigentlichen Testfälle zugeordnet, in denen das Testvorgehen detailliert beschrieben wird sowie die erwarteten Ergebnisse

dokumentiert werden. Wie bereits in den Grundlagen beschrieben, wird darauf verzichtet, die Testfälle zu detailliert (quasi als Klick-Anleitung) darzustellen, da zu diesem Zeitpunkt der Arbeit immer noch ein abstraktes System zur Automatisierung der Bearbeitung von IT-Sicherheitsvorfällen beschrieben wird. Im weiteren Verlauf werden die Evaluationskriterien dann auf zwei tatsächlich existierende IT-Systeme angewendet. Innerhalb der Überschrift ist in Klammern das betroffene Testszenario angegeben.

### Testfall 1 (Benutzerverwaltung und -anmeldung)

**Testvoraussetzung:**

Die Testperson verfügt über ein Benutzerkonto mit den Berechtigungen einer Vorfallsteamleiterin, sodass neue Benutzerkonten angelegt werden können.

**Testbeschreibung:**

Die Testperson sucht im Menü die Funktion zur Erstellung eines neuen Benutzerkontos. Anschließend legt sie jeweils ein Benutzerkonto pro Rolle (Vorfallsteamleiterin, Vorfallsteammitglied) an. Hierzu gibt sie (mindestens) einen Benutzernamen, ein Passwort und die jeweilige Rolle bzw. die dazugehörigen Berechtigungen an. Anschließend lässt sie sich die bestehenden Benutzerkonten anzeigen. Dann meldet sie sich mit dem Account ab und greift auf das System zu. Danach meldet sie sich mit jedem der neuen Benutzerkonten einmal an und wieder ab.

**Bezug:**

Userstory 7 und Userstory 11

**Priorität:**

erforderlich

**Details:**

Das Rechtemanagement (die Granularität der Benutzerkonten und Berechtigungen) wird in diesem Testfall nicht geprüft.

**Sollergebnis:**

Die neuen Benutzerkonten erscheinen in der Liste der Benutzerkonten des Systems und können zur Anmeldung am System benutzt werden. Die Benutzung ist ohne Anmeldung nicht möglich.

### Testfall 2 (Vorfallserstellung)

**Testvoraussetzung:**

Die Testperson verfügt über ein Vorfallsteammitglied-Benutzerkonto, das Vorfälle initial beurteilen darf und ein weiteres Benutzerkonto mit der Berechtigungen einer Vorfallsteamleiterin. Außerdem bestehen noch zwei weitere Benutzerkonten mit der Rolle Vorfallsteammitglied. Es besteht eine Simulation eines Detektionssystems, mit dem Vorfälle im hier betrachteten System erzeugt werden können.

**Testbeschreibung:**

Die Testperson löst durch die Simulation des Detektionssystems die Übergabe von einem IT-Sicherheitsvorfall an das hier betrachtete System aus. Anschließend meldet sie sich mit einem Benutzerkonto mit der Berechtigung für die Bearbeitung der Triage-Queue im System an und wählt einen der beiden erstellten Vorfallsmeldungen aus. Die Testperson beurteilt nun den Vorfall anhand der Dimensionen „Auswirkungen auf Funktionserbringung “ und „Auswirkungen auf Informationen“, woraus der Wert Kritikalität (oder vergleichbar) entsteht, der dem Vorfall vergeben wird. Die Testperson ordnet dem Vorfall zwei Vorfallsteammitglieder zu. Anschließend meldet sie sich ab, meldet sich mit dem Benutzerkonto der Vorfallsteamleiterin an und vergibt eine neue Kritikalität für den Vorfall.

**Bezug:**

Userstory 12, Userstory 13, Userstory 17, Userstory 14, Userstory 16

**Priorität:**

erforderlich

**Details:**

Keine Anmerkungen

**Sollergebnis:**

Der vom Detektionssystem gemeldete Vorfall erscheint im hier betrachteten System und kann mit einer Priorität versehen werden. Dem Vorfall können Teammitglieder zugeordnet werden. Die Kritikalität lässt sich auch im Nachhinein verändern.

### Testfall 3 (Vorfallserstellung)

**Testvoraussetzung:**

Die Testperson verfügt über ein Benutzerkonto mit der Rolle einer Vorfallsteamleiterin. Es besteht eine Simulation eines Detektionssystems, mit dem Vorfälle in dem hier betrachteten System erzeugt werden können. Es ist ein Zeitfenster konfiguriert, nachdem ein Vorfall automatisch eskaliert wird.

**Testbeschreibung:**

Die Testperson löst die Übergabe von zwei verschiedenen IT-Sicherheitsvorfällen durch die Simulation des Detektionssystems an das hier betrachtete System aus. Die Testperson wartet auf den Ablauf der Eskalationszeit für den zweiten Vorfall.

**Bezug:**

Userstory 9

**Priorität:**

erforderlich

**Details:**

Keine Anmerkungen

**Sollergebnis:**

Die Testperson erhält eine Benachrichtigung über den Ablauf der Frist.

### Testfall 4 (Vorfallserstellung)

**Testvoraussetzung:**

Durchführung von Testfall 2

**Testbeschreibung:**

Die Testperson prüft, ob sie eine Benachrichtigung über den neuen Vorfall und eine Benachrichtigung über die Zuordnung erhalten hat.

**Bezug:**

Userstory 19 und 18

**Priorität:**

erforderlich

**Details:**

Keine Anmerkungen



**Sollergebnis:**

Die Testperson wurde über den neuen Vorfall und über die Zuordnung benachrichtigt.

**Testfall 5 (Vorfallsanzeige und -editierung)****Testvoraussetzung:**

Die Testperson verfügt über ein Vorfallsteammitglied-Benutzerkonto, das Vorfälle initial beurteilen darf und ein weiteres Benutzerkonto mit den Berechtigungen einer Vorfallsteamleiterin. Außerdem bestehen noch zwei weitere Benutzerkonten mit der Rolle Vorfallsteammitglied. Es bestehen in dem System zwei bereits beurteilte Vorfälle.

**Testbeschreibung:**

Die Testperson meldet sich am System als Vorfallsteammitglied an. Sie sucht die Schaltfläche zur Auflistung von IT-Sicherheitsvorfällen. Anschließend wählt sie einen der Vorfälle aus. In dem ausgewählten Vorfall bearbeitet sie einige der Stammdaten (Titel und Beschreibung). Anschließend verknüpft sie den einen Vorfall mit dem anderen Vorfall.

**Bezug:**

Userstory 15, Userstory 21, Userstory 22, Userstory 29

**Priorität:**

erforderlich

**Details:**

Das Berechtigungsmanagement wird an dieser Stelle nicht geprüft.

**Sollergebnis:**

Der Testperson werden die zuvor erstellten Vorfälle angezeigt, die Daten des Vorfalls lassen sich anzeigen und bearbeiten. Die Verknüpfung von zwei Vorfällen funktioniert.

**Testfall 6 (Vorfallsanzeige und -editierung)****Testvoraussetzung:**

Durchführung von Testfall 5

**Testbeschreibung:**

Die Testperson meldet sich im System als Vorfallsteamleiterin an und markiert einen Vorfall als lesebeschränkt. Anschließend meldet sie sich ab und meldet sich mit einem Benutzerkonto an, welches dem gerade geänderten Vorfall nicht zugeordnet ist. Sie versucht sich den gerade geänderten Vorfall anzeigen zu lassen. Anschließend meldet sie sich ab und meldet sich mit einem Benutzerkonto an, welches dem Vorfall zugeordnet ist und versucht sich den Vorfall anzeigen zu lassen.

**Bezug:**

Userstory 8

**Priorität:**

erforderlich

**Details:**

Keine Anmerkungen

**Sollergebnis:**

Die Testperson kann sich mit einem nicht zugeordneten Benutzerkonto den angepassten Vorfall nicht anzeigen lassen. Der Zugriff mit einem zugeordneten Benutzerkonto funktioniert jedoch.

### Testfall 7 (Vorfallsanalyse)

**Testvoraussetzung:**

Die Testperson verfügt über ein Vorfallsteammitglied-Benutzerkonto. Es besteht ein bereits beurteilter Vorfall, dem sie zugeordnet ist.

**Testbeschreibung:**

Die Testperson meldet sich im System als Vorfallsteammitglied an. Sie lässt sich den angelegten IT-Sicherheitsvorfall anzeigen. Dort fügt sie ein Artefakt hinzu und ordnet es einem Benutzerkonto zu.

**Bezug:**

Userstory 24

**Priorität:**

erforderlich

**Details:**

Keine Anmerkungen

**Sollergebnis:**

Die Person kann Artefakte hochladen und einem Benutzerkonto zur Analyse zuordnen.

**Testfall 8 (Vorfallsanalyse)****Testvoraussetzung:**

Durchführung von Testfall 7

**Testbeschreibung:**

Die Testperson meldet sich im System als Vorfallsteammitglied an. Sie lässt die Artefakte automatisiert analysieren.

**Bezug:**

Userstory 25

**Priorität:**

erforderlich

**Details:**

Keine Anmerkungen

**Sollergebnis:**

Die Person kann Artefakte automatisiert untersuchen lassen.

**Testfall 9 (Vorfallsanalyse)****Testvoraussetzung:**

Die Testperson verfügt über ein Vorfallsteammitglied-Benutzerkonto. Es besteht ein bereits beurteilter Vorfall, dem sie zugeordnet ist.

**Testbeschreibung:**

Die Testperson meldet sich im System als Vorfallsteammitglied an. Sie fügt verschiedene Observables und IoC (Prüfsummen, Hostnames, IP-Adressen) hinzu.

**Bezug:**

Userstory 23

**Priorität:**

erforderlich

**Details:**

Keine Anmerkungen

**Sollergebnis:**

Die Testperson kann die IoC hinzufügen.

**Testfall 11 (Vorfallsreaktion)****Testvoraussetzung:**

Die Testperson verfügt über ein Vorfallsteammitglied-Benutzerkonto.

**Testbeschreibung:**

Die Person meldet sich an und sucht die Funktion zur Erstellung von SOP. Für jede Phase der Vorfallsbehandlung (Detektion und Analyse sowie Eindämmung, Behebung und Wiederherstellung) erstellt die Person beispielhafte Handlungsanweisungen (SOP).

**Bezug:**

Userstory 26

**Priorität:**

erforderlich

**Details:**

Keine Anmerkungen

**Sollergebnis:**

Die Testperson kann Handlungsanweisungen für die verschiedenen Phasen eintragen sowie die Parameter, bei denen diese Handlungsanweisungen eingebunden werden sollen.

**Testfall 12 (Vorfallsreaktion)****Testvoraussetzung:**

Durchführung von Testfall 11, zusätzlich ein bereits beurteilter Vorfall, der die in Testfall 11 erstellten Kriterien für die Zuordnung einer SOP erfüllt.

**Testbeschreibung:**

Die Testperson meldet sich an, öffnet den Vorfall und aktiviert bzw. kommentiert die Aktionen, die durch die SOP vorgegeben werden.

**Bezug:**

Userstory 32

**Priorität:**

erforderlich

**Details:**

Keine Anmerkungen

**Sollergebnis:**

Die Testperson kann Schritte einer Handlungsanweisungen als durchgeführt bzw. entbehrlich markieren und zusätzlich Kommentare einfügen.

### Testfall 13 (Vorfallsreaktion)

**Testvoraussetzung:**

Das System verfügt laut Dokumentation über eine Schnittstelle zur Automatisierung der Vorfallsreaktion. Eine beispielhafte Funktion ist mittels dieser Schnittstelle implementiert (Versand einer E-Mail bei Eintreffen von bestimmten Kriterien).

**Testbeschreibung:**

Die Testperson löst die Meldung eines Vorfalls aus, die den Kriterien für die automatische Reaktion unterfällt.

**Bezug:**

Userstory 27

**Priorität:**

erforderlich

**Details:**

Keine Anmerkungen

**Sollergebnis:**

Die zuvor für die Vorfallsautomation vorgesehene Aktion wird bei der Erstellung des neuen Vorfalls ausgelöst.

### Testfall 14 (Threat Sharing)

**Testvoraussetzung:**

Das System ist an eine Threat-Sharing-Plattform angebunden, die Testperson verfügt über ein Vorfallsteammitglied-Benutzerkonto, es gibt einen beurteilten Vorfall. In diesem Vorfall sind IoC bzw. Observables enthalten und die Testperson ist diesem Vorfall zugeordnet.

**Testbeschreibung:**

Die Testperson meldet sich an, ruft den zuvor erstellten Vorfall auf und schickt die Observables/IoC an die Threat-Sharing-Plattform.

**Bezug:**

Userstory 3

**Priorität:**

erforderlich

**Details:**

Keine Anmerkungen

**Sollergebnis:**

Die Daten werden als Ereignis an die Threat-Sharing-Plattform exportiert.

### Testfall 15 (Threat Sharing)

**Testvoraussetzung:**

Das System ist an eine Threat-Sharing-Plattform angebunden, die Testperson verfügt über ein Vorfallsteammitglied-Benutzerkonto, es gibt einen beurteilten Vorfall, in dem IoC und Observables enthalten sind und die Testperson ist diesem Vorfall zugeordnet.

**Testbeschreibung:**

Die Testperson löst eine Meldung von der Threat-Sharing-Plattform an das System aus, in der auch IoC enthalten sind, die bereits in dem Vorfall dokumentiert sind.

**Bezug:**

Userstory 4

**Priorität:**

erforderlich

**Details:**

Keine Anmerkungen

**Sollergebnis:**

Die in der Meldung enthaltenen Daten werden im System dem bestehenden Vorfall zugeordnet.

### Testfall 16 (Vorfalldokumentation)

**Testvoraussetzung:**

Die Testperson verfügt über ein Vorfallsteammitglied-Benutzerkonto, es existiert ein bereits beurteilter Vorfall, in dem auch schon Aktionen (Analyse, Reaktion) vorgenommen wurden.

**Testbeschreibung:**

Die Testperson meldet sich an, ruft den zuvor erstellten Vorfall auf und verfasst Kommentare. Zudem fügt sie einen Link hinzu, der auf ein externes Dokument verweist. Zudem passt sie einige Daten in dem Vorfall an (z.B. Titel, Tags). Anschließend ruft sie den Vorfallszeitstrahl auf.

**Bezug:**

Userstory 20, Userstory 29, Userstory 30, Userstory 31, Userstory 33

**Priorität:**

erforderlich

**Details:**

Keine Anmerkungen

**Sollergebnis:**

Im Zeitstrahl werden alle durchgeführten Aktionen angezeigt.

### Testfall 17 (Vorfallskommunikation)

**Testvoraussetzung:**

Die Testperson verfügt über ein Vorfallsteammitglied-Benutzerkonto, es existiert ein bereits beurteilter Vorfall.

**Testbeschreibung:**

Die Testperson meldet sich an und schreibt eine Zusammenfassung für den Vorfall, die nächsten Schritte und die letzten Schritte. Anschließend exportiert die Testperson den Bericht oder verschickt ihn per Mail.

**Bezug:**

Userstory 2

**Priorität:**

erforderlich

**Details:**

Keine Anmerkungen

**Sollergebnis:**

Die Testperson kann einen Management-Bericht verschicken.

### Testfall 18 (Nachbereitung)

**Testvoraussetzung:**

Die Testperson verfügt über ein Vorfallsteammitglied-Benutzerkonto und es existiert ein bereits beurteilter Vorfall.

**Testbeschreibung:**

Die Testperson meldet sich an, ruft den zuvor erstellten Vorfall auf und verfasst einen Kommentar, der als „Relevant für die Nachbereitung“ markiert wird. Anschließend exportiert sie den Vorfall für die Nachbesprechung und für den Abschlussbericht.

**Bezug:**

Userstory 1, Userstory 6

**Priorität:**

erforderlich

**Details:**

Keine Anmerkungen

**Sollergebnis:**

Die Notizen für die Nachbesprechung werden in einem speziellen Nachbesprechungsexport angezeigt. Für den Abschlussbericht werden alle Aktionen und insbesondere die Timeline exportiert.

### Testfall 19 (Nachbereitung)

**Testvoraussetzung:**

Die Testperson verfügt über ein Vorfallsteamleiterin-Benutzerkonto, es existieren mehrere bearbeitete Vorfälle.

**Testbeschreibung:**

Die Testperson meldet sich an, ruft die Funktion zur Anzeige der Metriken auf.

**Bezug:**

Userstory 10



**Priorität:**

erforderlich

**Details:**

Keine Anmerkungen

**Sollergebnis:**

Die Metriken „Anzahl der bearbeiteten Vorfälle pro Zeiteinheit“, „Zeit pro Vorfall“, „Anzahl der beteiligten Vorfallsteammitglieder“, die „Arten der Artefakte“ und die „Dauer der einzelnen Analysen pro Artefakt“ werden angezeigt.

## 4.3. Abhängigkeiten der Testfälle

Wie in einigen Testfällen ersichtlich, basieren bestimmte Testfälle explizit auf anderen Testfällen bzw. haben die gleichen Voraussetzungen, die erfüllt sein müssen. Diese Voraussetzungen sind teilweise auch schon durch die Ergebnisse aus anderen Testfällen erfüllt. Insofern besteht eine Abhängigkeit zwischen den Testfällen untereinander, die auch bei der Durchführung der Tests genutzt werden kann. Durch die Beschreibung der Abhängigkeit lässt sich der Testaufwand reduzieren, da die Testergebnisse zur Vorbereitung von weiteren Tests nutzbar sind.

In Abbildung 4.1 ist dargestellt, wie die Testfälle voneinander abhängig sind. Dies wird im weiteren Verlauf zur Testplanung und zur Durchführung der Tests genutzt.

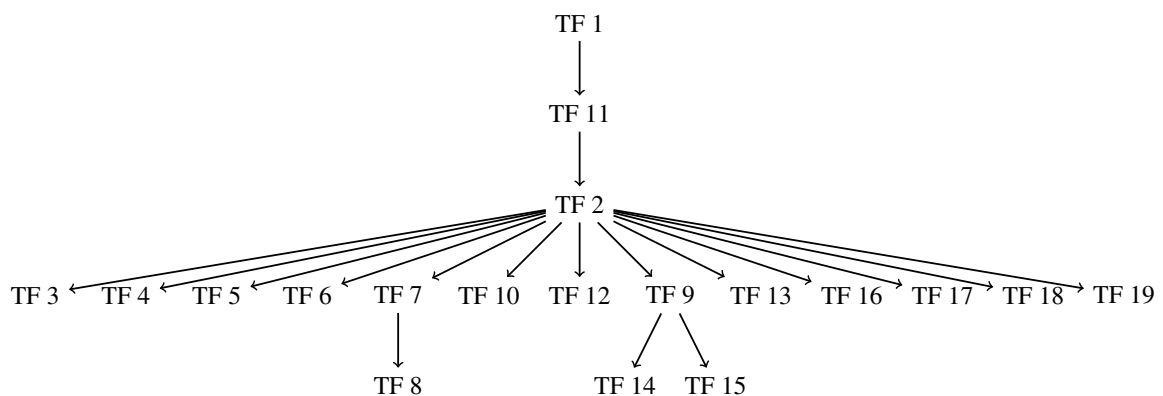


Abbildung 4.1.: Abhängigkeiten der Testfälle (TF)

## 5. Auswahl der IT-Systeme

Nachdem zunächst im gleichnamigen Kapitel die ANFORDERUNGEN und im Kapitel TESTSZENARIEN UND TESTFÄLLE die Kriterien zur Evaluation von IT-Systemen zur Automatisierung der IT-Sicherheitsvorfallbearbeitung dargestellt wurden, müssen im nächsten Schritt die zu untersuchenden IT-Systeme identifiziert werden. Für diese Arbeit werden zwei exemplarische IT-Systeme evaluiert; mit den zuvor beschriebenen Anforderungen und Testfällen wäre es aber auch möglich noch weitere IT-Systeme zu testen. Die hier betrachteten IT-Systeme, die dafür entwickelt wurden den Prozess der IT-Sicherheitsvorfallreaktion zu unterstützen und zu automatisieren, werden als *Security Orchestration, Automation and Response (SOAR)* vertrieben. Diese Lösungen werden von Herstellern sowohl kommerziell (z.B. [27] und [28]) als auch als community-getriebene Open-Source-Lösungen ([29]) angeboten. Für diese Arbeit sollte sowohl eine kommerzielle Lösung als auch eine Open-Source-Lösung betrachtet werden.

### 5.1. TheHive/Cortex

Als Open-Source-Lösung wurde die Anwendung *TheHive* ausgewählt, eine Software zur Verfolgung von IT-Sicherheitsvorfällen, die aus dem gleichnamigen Open-Source-Projekt stammt. TheHive benutzt zur Automatisierung die Anwendung *Cortex*, die ebenfalls aus dem TheHive-Projekt stammt. Als Open-Source-Software lassen sich TheHive und Cortex kostenlos über GitHub herunterladen. Die aktuell als stabil bezeichnete Version ist Version 3.4, die Anfang September 2019 veröffentlicht wurde und seitdem mehrere Bugfixes erhalten hat. Für diese Arbeit wird jedoch die aktuell in Entwicklung befindliche Version 4.0.0 ausgewählt: Mit dieser Version wurden einige grundsätzliche Änderungen hinsichtlich Softwaredesign und Anpassbarkeit vorgenommen. Zudem handelt es sich um die Version, in die auf absehbare Zeit der größte

Aufwand für die Weiterentwicklung gesteckt wird, auch wenn TheHive 3 noch zwei Jahre nach dem Release von TheHive 4 unterstützt werden wird. [30]

Zum Zeitpunkt der Testdurchführung war die Version TheHive 4 RC1 die aktuellste Version<sup>1</sup> von TheHive. Auch wenn diese Version noch nicht für den Produktivbetrieb freigegeben ist, so wurde sie doch schon als ausreichend stabil angesehen, um hier untersucht zu werden. Für die Automatisierung von Analysen und Reaktionen wurde wie schon erläutert die Anwendung Cortex verwendet, die derzeit in der Version 3.0.1 zur Verfügung steht. [29] In Abbildung 5.1 ist dargestellt, wie der Arbeitsablauf in TheHive und Cortex realisiert wird.

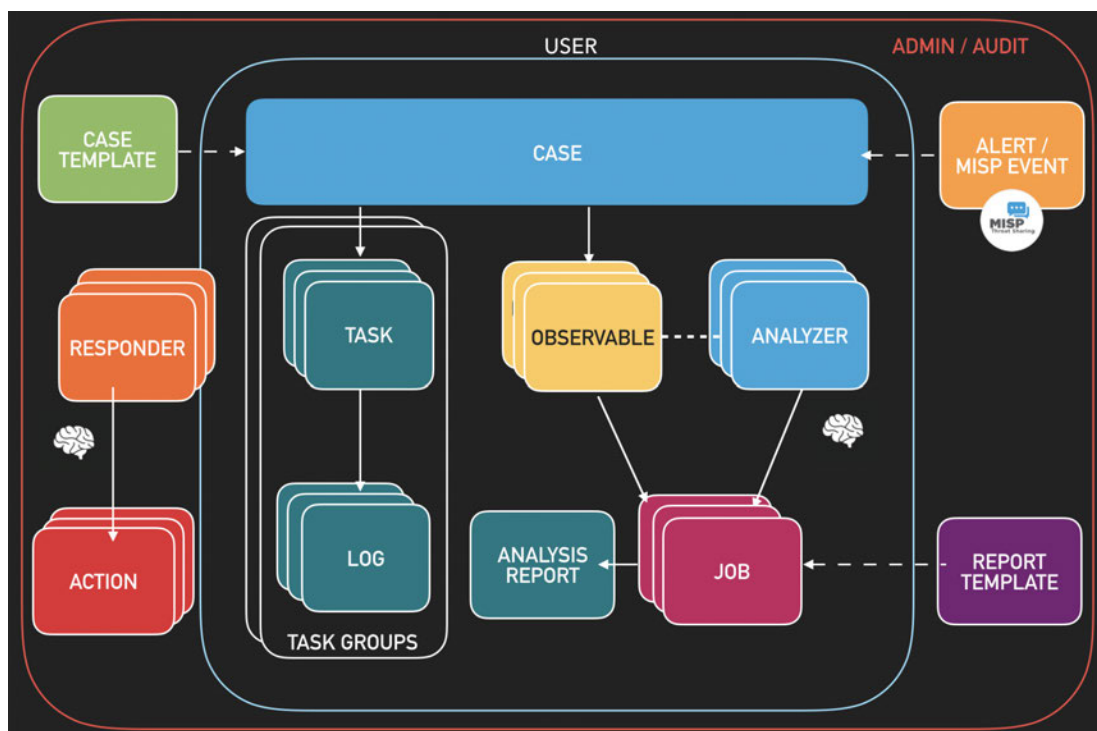


Abbildung 5.1.: Datenfluss in TheHive/Cortex [31]

Über externe Anwendungen können innerhalb von TheHive *Alerts* erzeugt werden. Diese Alerts beinhalten zum Beispiel Observables, Indicators of Compromise und weitere Informationen, zum Beispiel über die Quelle der Meldung. Aus Alerts können dann neue *Cases*, also IT-Sicherheitsvorfälle erzeugt werden, bzw. die Alerts können zu bestehenden Cases hinzugefügt werden. Cases können *Tasks* enthalten, die beschrei-

<sup>1</sup>Nach Abschluss der Tests wurde TheHive 4 RC2 und während der Fertigstellung der Arbeit TheHive4 RC3 herausgebracht. An Stellen wo dies einen Unterschied in der Testdurchführung gemacht hätte, wird dies innerhalb der Arbeit aufgegriffen.

ben, welche Aktionen innerhalb eines Vorfalls durchgeführt werden müssen. Außerdem enthalten die Tasks *Observables*, die mit *Analyzer* untersucht werden können. Zur Automatisierung der Reaktion dienen sogenannte *Responder*, mit denen weitere Systeme, wie Firewalls oder die Benutzerverwaltung angesprochen werden können.

## 5.2. Splunk Phantom

Zur Evaluierung der kommerziellen Lösungen wurde das SOAR-System *Splunk Phantom* des Herstellers Splunk ausgewählt, da der Hersteller dieser Software eine kostenlose Community-Edition zum Download bereitstellt. Diese Version enthält zwar alle Funktionen der Vollversion, ist allerdings hinsichtlich der Menge der verarbeiteten Ereignisse limitiert. Dies stellt jedoch kein Problem für die Arbeit dar, da für die Tests ohnehin sehr wenige Ereignisse erzeugt und untersucht werden müssen. Laut Hersteller umfasst Splunk Phantom die Funktionalitäten Infrastruktursteuerung, Automation und Case Management. Als aktuellste Version steht von Splunk Phantom die Version 4.8 zur Verfügung, die auch im Rahmen der Masterthesis verwendet wurde. [28] In Abbildung 5.2 ist ersichtlich, wie der Arbeitsablauf in Splunk Phantom abgebildet wird.

Mittels sogenannter *Apps* kann der Funktionsumfang von Splunk Phantom erweitert werden, da diese Apps *Actions* zur Verfügung stellen, mittels derer auf andere IT-Systeme (z.B. Benutzerverwaltung oder Firewall) zugegriffen werden kann. Die Vorstufe eines IT-Sicherheitsvorfalls heißt in Splunk Phantom *Container*. Ein solcher Container enthält Informationen über einen möglichen IT-Sicherheitsvorfall. Der Container kann auch *Artifacts* enthalten, die Informationen über den Vorfall, z.B. in Form von Dateihashes, IP-Adressen oder E-Mailheadern aufgenommen werden. Die Container können zu einem *Case* hochgestuft werden, wodurch ein bestätigter IT-Sicherheitsvorfall dargestellt wird. Um die Bearbeitung von IT-Sicherheitsvorfällen zu automatisieren, können *Playbooks* geschrieben werden, die basierend auf den aus den Apps entnommenen Actions beschreiben, wie mit IT-Sicherheitsvorfällen umgegangen wird. *Workbooks* dienen als Vorlagen, die einerseits Playbooks für die Bearbeitung von IT-Sicherheitsvorfällen enthalten; andererseits aber auch festlegen, wie der Vorfall bear-

beitet wird, also welcher Arbeitsablauf für die Behandlung eines IT-Sicherheitsvorfalls zu durchlaufen ist.

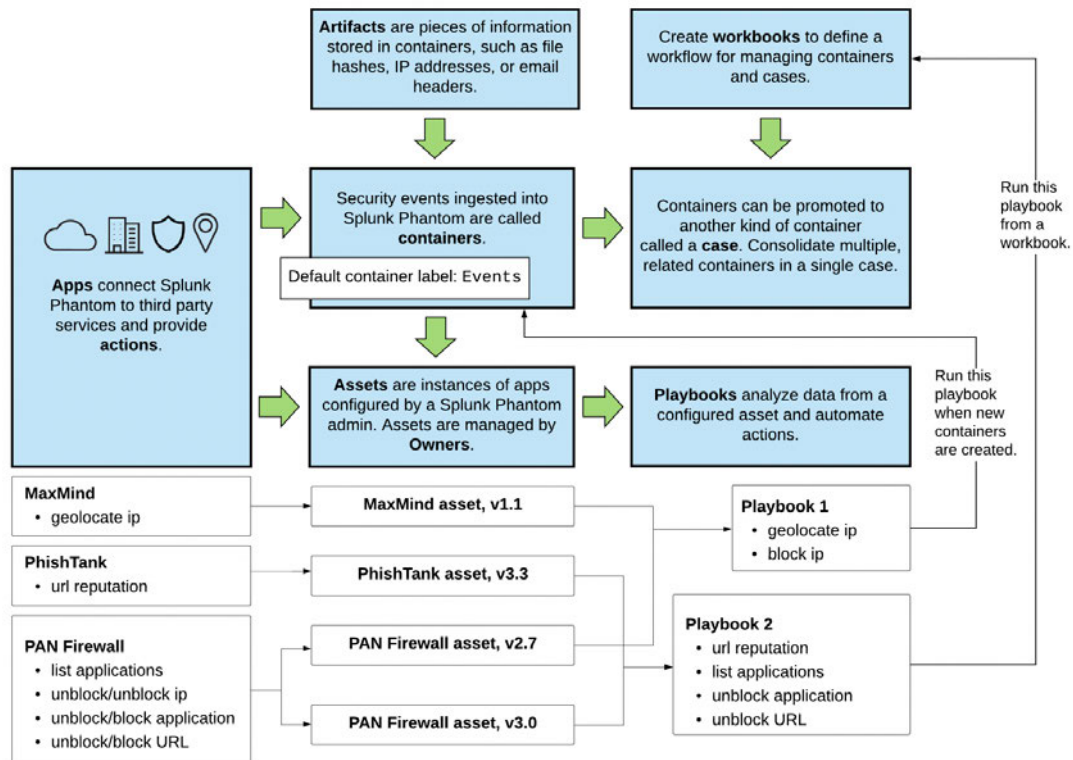


Abbildung 5.2.: Datenfluss in Splunk Phantom [32]

## 6. Entwicklung einer Laborumgebung

Nachdem nun im vorangegangenen Kapitel AUSWAHL DER IT-SYSTEME die IT-Systeme zur Evaluierung ausgewählt werden, muss im nächsten Schritt eine Untersuchungs- bzw. Laborumgebung zur Evaluation gemäß der in TESTSZENARIEN UND TESTFÄLLE aufgestellten Kriterien geschaffen werden. In diesem Kapitel wird beschrieben, wie diese Umgebung aufgebaut wird.

### 6.1. Aufbau der Laborumgebung

Das Testlabor wird innerhalb einer Virtualisierungsplattform realisiert. Dies hat den Vorteil, dass keine zusätzliche Hardware beschafft werden muss. Zudem ermöglicht die Nutzung von virtuellen Maschinen das Erzeugen von so genannten Snapshots, also einer Sicherung der virtuellen Maschine zu einem definierten Zeitpunkt, mit der Möglichkeit zu diesem Zustand jederzeit wieder zurückgehen zu können. Für den Zweck der Automatisierbarkeit der Testumgebung bietet das den Vorteil, dass nach der Durchführung eines jeden Testfalls zum Originalzustand des Systems zurückgesprungen werden kann. Hierdurch lassen sich Tests besser reproduzieren, falls dies durch neue Versionen oder weitere Anforderungen und Testfälle notwendig werden sollte.

Der Aufbau der Laborumgebung ist in Abbildung 6.1 skizziert. Im Rahmen dieser Arbeit wurde VMware Fusion als Virtualisierungsumgebung eingesetzt. VMware Fusion wurde auf einem MacBook Pro mit 16 GB Arbeitsspeicher und 2,5 GHz i7 Dual-Core Prozessor installiert, sodass auch ausreichend Arbeitsspeicher zur Verfügung stand, um mehrere IT-Systeme parallel zu virtualisieren.

Um das Labornetzwerk bereitzustellen, wurde innerhalb von VMware Fusion ein virtuelles Netzwerk 172.16.6.0/24 erstellt, sodass dieses Netzwerk über 253 mögliche IP-Adressen für Gastsysteme (nachdem eine IP-Adresse für den Host abgezogen wur-

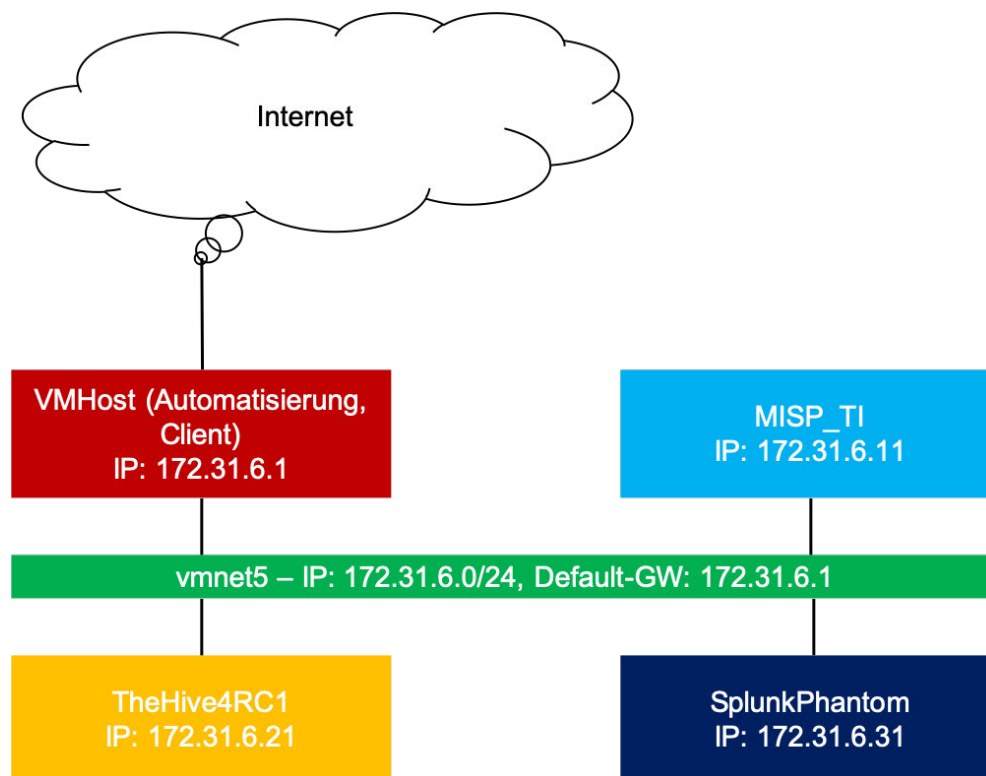


Abbildung 6.1.: Aufbau der Laborumgebung

de) verfügt. Zudem wurde dem Netzwerk der Zugang zum Internet über das Hostsystem per NAT<sup>1</sup> ermöglicht.

Für die Durchführung der Tests werden insgesamt drei virtuelle Maschinen benötigt: Jeweils ein System wird für den jeweiligen Untersuchungsgegenstand, also TheHive und Splunk Phantom benötigt sowie ein weiteres System für die Bereitstellung der Threat-Intelligence-Plattform MISP.

Um zu verhindern, dass mehrere Gastsysteme die gleiche IP-Adresse nutzen möchten, vergibt VMware Fusion die IP-Adressen standardmäßig per DHCP. Dies musste in der Konfiguration des virtuellen Netzwerks so angepasst werden, dass für jedes vorgesehene IT-System eine feste IP-Adresse vergeben wird:

```
1 # [...] Standardkonfiguration wird hier nicht dargestellt
2
3 host MISP_TI {
4     hardware ethernet 00:50:56:3A:73:80;
5     fixed-address 172.31.6.11;
```

---

<sup>1</sup>NAT (Network Address Translation): Ermöglicht den Zugang zu externen Netzwerken durch die Übersetzung von internen IP-Adressen zu einer extern erreichbaren IP-Adresse

Die virtuelle Maschine mit dieser Konfiguration wird eine benutzerdefinierte Netzwerkverbindung verwenden.

☒ Verbindung virtueller Maschinen in diesem Netzwerk zu externen Netzwerken zulassen (mithilfe von NAT)

☐ IPv6 aktivieren

IPv6-Präfix: fd15: ::/64

Port-Weiterleitung

Port des...	Typ	IP-Adresse der VM	Beschreibung

+ -

☒ Host-Mac mit diesem Netzwerk verbinden

☒ Adressen in diesem Netzwerk über DHCP zur Verfügung stellen

Teilnetz-IP: 172.31.6.0

Teilnetzmaske: 255.255.255.0

MTU: Systemkonfiguration

Abbildung 6.2.: Konfiguration des VMware Subnets

```

6 }
7 host TheHive4RC1 {
8     hardware ethernet 00:50:56:24:DB:98;
9     fixed-address 172.31.6.21;
10 }
11 host SplunkPhantom {
12     hardware ethernet 00:50:56:28:9F:C2;
13     fixed-address 172.31.6.31;
14 }

```

Listing 6.1: Ausschnitt aus VMware Konfigurationsdatei vmnet5/dhcpd.conf

Für die Integration von MISP wurde die auf der Seite des Projekts verfügbare Schulungsumgebung [33] als virtuelle Maschine eingebunden. Dieses System wurde noch dahingehend angepasst, dass der Hostname auf `misp` angepasst wurde und dass der Webserver so konfiguriert wurde, dass die Anwendung direkt vom Hostsystem und innerhalb des Testnetzwerks aufrufbar gemacht wurde. Um einige Testdaten nutzen zu können, wurde eine öffentliche Quelle für Threat-Intelligence-Daten [34] hinzugefügt. Außerdem wurde die Organisation Test-MT angelegt, in der die Testdaten für die



Arbeit angelegt werden. Für jeden Testgegenstand wurde zudem ein eigenes Benutzerkonto angelegt, dass dieser Organisation zugeordnet wurde:

- *misp.thehive@lab.[domain.tld]*<sup>2</sup>: Benutzerkonto für die Integration zwischen MISP und TheHive
- *misp.phantom@lab.[domain.tld]*: Benutzerkonto für die Integration zwischen MISP und Splunk Phantom

Beide Benutzerkonten wurden mit der Rolle *Sync-User* angelegt und ein API-Key pro Benutzer angelegt, mit dem der Zugriff auf Schnittstellen für andere IT-Systeme (in diesem Fall die Untersuchungsgegenstände) möglich ist.

Die Simulation des Detektionssystems wurde auf dem Hostsystem ausgeführt. Hierfür wurde das Werkzeug `curl` genutzt, mit dem Aufrufe über das HTTP-Protokoll ausgeführt werden können. Da beide Testobjekte über eine HTTP-Schnittstelle erreichbar sind, konnten über diesen Weg Alarmmeldungen erzeugt werden (siehe Listings A.3 und A.4 ab Seite 134).

## 6.2. Integration der Untersuchungsgegenstände

Um die IT-Systeme testen zu können, müssen diese in die Laborumgebung eingebunden werden. Je nach System ist dieser Integrationsschritt unterschiedlich komplex. Im Folgenden sind die dafür notwendigen Schritte für beide Systeme beschrieben.

### 6.2.1. Integration von TheHive/Cortex

Wie bereits erläutert, benötigt die Incident Response Plattform von TheHive als Hintergrundsystem zur Automatisierung die ebenfalls von dem Projekt entwickelte Anwendung *Cortex*. Beide Komponenten wurden auf dem gleichen System installiert, da auch die zweite in der Arbeit betrachtete Anwendung Splunk Phantom als eine vollständige Appliance bereitgestellt wird.

Als Basis für TheHive/Cortex wurde das Betriebssystem Ubuntu in der Version 20.04 verwendet. Dabei wurde der Installationsassistent verwendet und die Standardeinstellungen ausgewählt. Als zusätzliche Software wurde noch ein SSH-Server installiert,

---

<sup>2</sup>Die tatsächliche Domain wurde für den Druck gegen einen Platzhalter ersetzt.

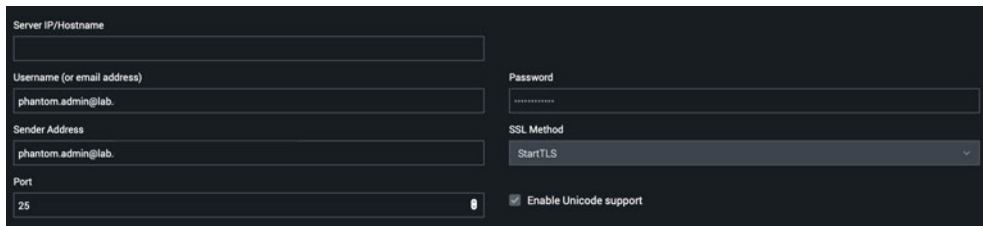
um auf das System auch aus der Ferne zugreifen zu können. Anschließend wurde zunächst TheHive anhand der Installationsanleitung [35] installiert. Da es sich bei dieser Installation um einen Test handelt, wurde auf die Nutzung des verteilten Filesystems *hadoop* verzichtet und ein lokaler Datenspeicher eingerichtet. Im nächsten Schritt erfolgte die Installation von Cortex [36] und die anschließende Integration von Cortex in TheHive anhand der Dokumentation zur Integration von weiteren Systemen [37]. Abschließend wurde nach dieser Anleitung vorgegangen, um auch die Integration mit MISP abzuschließen. Ein Fehler, der in der Anleitung und im Konfigurationsbeispiel enthalten ist, stellte hier eine Schwierigkeit bei der Konfiguration dar, konnte aber mithilfe eines Problembeitrags auf GitHub [38] behoben werden. Zur Aktivierung der Konfiguration musste noch das selbstsignierte Zertifikat des MISP-Servers dem Speicher für vertrauenswürdige Zertifikate hinzugefügt werden, bevor die Integration abgeschlossen war. Die vollständige Darstellung der TheHive-Konfigurationsdatei `application.conf` befindet sich auf Seite 132 im Anhang.

#### 6.2.2. Integration von Splunk Phantom

Für die Installation von Splunk Phantom konnte die vorbereitete Appliance genutzt werden. Diese Appliance wurde in VMware Fusion importiert und gemäß der Benutzeranleitung [39] und [40] installiert. Nach Abschluss der Installation wurde das Passwort für den Administrationsbenutzer angepasst, die Anwendung zur Integration von MISP installiert und der Versand von E-Mails vorbereitet (siehe Abbildung A.9). Da die Appliance von Splunk Phantom dazu gedacht ist, auch potentielle Kundinnen zu gewinnen, ist der Prozess der Erstinstallation sehr einfach gehalten. Beim ersten Start lassen sich außerdem Testdaten importieren, sodass die Anwendung direkt getestet werden kann. Hierauf wurde jedoch verzichtet, da für die Evaluierung eigene Testbeispiele festgelegt wurden.

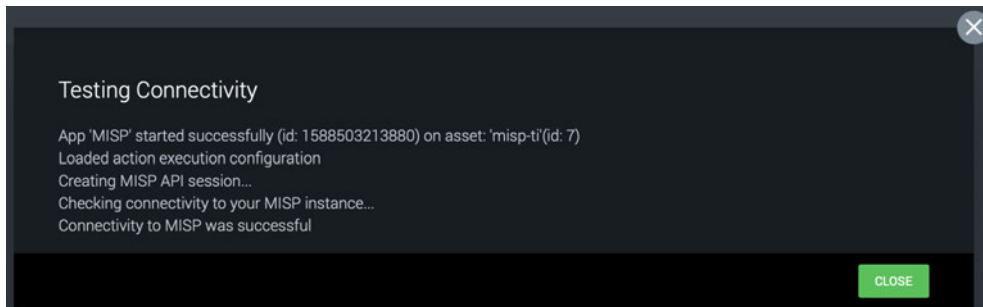
### 6.3. Vorbereitung der Tests

Neben der Integration mussten die Testobjekte auch für die Tests vorbereitet werden. Hierbei sollte die Ausgangssituation für die erfolgreiche Durchführung der Tests geschaffen werden. Das betrifft sowohl die Erzeugung von Testdaten, als auch das An-



The screenshot shows a configuration window for mail settings. It includes fields for 'Server IP/Hostname', 'Username (or email address)' (filled with 'phantom.admin@lab.'), 'Password', 'Sender Address' (filled with 'phantom.admin@lab.'), and 'Port' (filled with '25'). There is a dropdown for 'SSL Method' set to 'StartTLS' and a checkbox for 'Enable Unicode support' which is checked.

(a) Mailanbindung



(b) Verbindung zu MISP

Abbildung 6.3.: Konfiguration von Splunk Phantom

legen von Benutzerkonten und das Vorbereiten von Testdaten in den zu integrierenden Systemen.

### 6.3.1. Vorbereitung von TheHive/Cortex

Nach der Erstinstallation von TheHive ist ein Benutzerkonto mit Administratorinnen-rechten angelegt, das nur Organisationen anlegen kann. Dieses Konto darf keine Analysen durchführen oder Fälle anlegen, allerdings kann mit diesem Konto eine Organisation und das dazugehörige Administratorinnenkonto angelegt werden. Da die Komponente Cortex in dieser Arbeit nur als Automatisierungsplattform für TheHive dient, mussten dort keine weiteren Benutzerkonten angelegt werden.

Im Rahmen der Vorbereitung wurde eine Organisation mit dem Namen *TestOrg* angelegt. Zudem wurde auch ein *Organisations-Administrator*-Konto angelegt, dass diese Organisation verwalten darf. Um auch die Testfälle zum Teilen einzelner Vorfällen zwischen verschiedenen Organisationen testen zu können, wurde auch eine weitere Organisation mit dem Namen *SecondTestOrg* inklusive Organisations-Administrator angelegt. Die vordefinierte Rolle der Organisations-Administratorin entspricht der IT-Sicherheitsvorfallsteamleiterin in dieser Arbeit.

Als weitere Maßnahmen zur Testvorbereitung wurde noch ein Benutzerkonto mit der Rolle *Analyst* angelegt, das für die Erzeugung von Meldungen durch das simulierte Detektionssystem genutzt wird.

Nach Abschluss dieser Schritte sind also zwei Organisationen innerhalb von TheHive angelegt: *TestOrg* als Organisation, die für die Durchführung der Tests genutzt wird und *SecondTestOrg* als Organisation, die für die Testfälle zur Weitergabe der Analyse an andere Organisationen verwendet werden soll.

Somit bestanden zu diesem Punkt drei Benutzerkonten innerhalb von TheHive:

- *admin* als Systemadministratorin, mit der Berechtigung weitere Organisationen und Benutzerinnen anzulegen.
- *testorgadmin* als Administratorin für die Testorganisation TestOrg
- *testorgapi* als Benutzerkonto zum Erzeugen von Vorfällen per HTTP-API

Für das Benutzerkonto *testorgadmin* wurde noch ein Passwort vergeben, für das Benutzerkonto *testorgapi* wurde ein API-Key als Authentifizierungsmerkmal erzeugt.

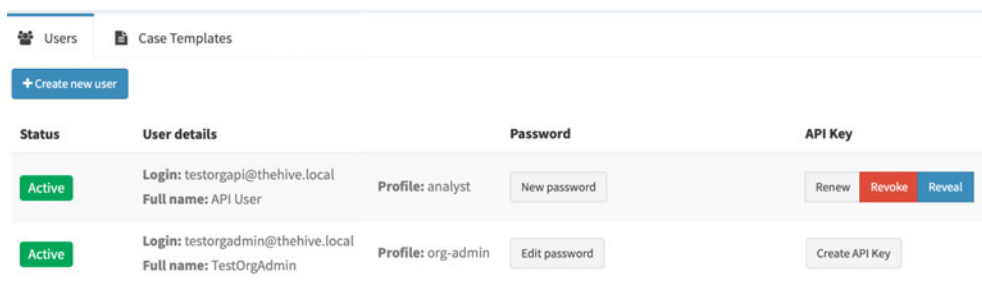
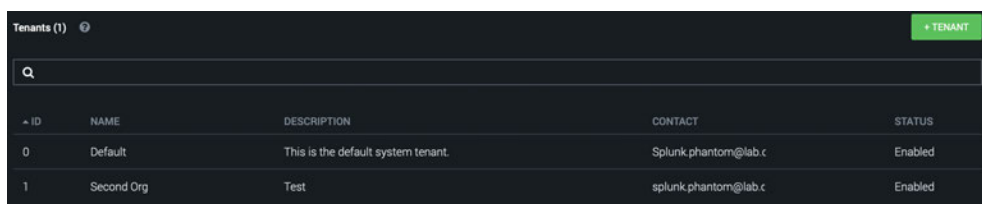


Abbildung 6.4.: Konfiguration der Benutzer in TheHive

#### 6.3.2. Vorbereitung von Splunk Phantom

Wie bereits geschildert, wurde von Splunk Phantom die vorgefertigte Appliance installiert. Diese enthält standardmäßig ebenfalls ein Administratorinnenkonto. Mit diesem Konto können alle in Splunk Phantom implementierten Aktionen durchgeführt werden, sodass dieses Konto auch hier für die Rolle der Vorfallsteamleiterin genutzt werden kann. Zudem wird noch ein Benutzerkonto mit den Berechtigungen *Automation* erstellt, welches für die Anbindung des simulierten Detektionssystems verwendet wird.

Um die Weitergabe von Fällen zwischen Organisationen zu testen, musste noch die Funktion *Multi Tenancy* aktiviert werden. Somit können mehrere Organisation innerhalb einer Instanz abgebildet werden, was auch für Splunk Phantom dazu genutzt werden sollte, die Testfälle zur Weitergabe von Vorfällen an andere Organisationen durchführen zu können. Auch hier wurden zwei Organisationen eingerichtet (siehe Abbildung 6.5).



ID	NAME	DESCRIPTION	CONTACT	STATUS
0	Default	This is the default system tenant.	Splunk.phantom@lab.c	Enabled
1	Second Org	Test	splunk.phantom@lab.c	Enabled

Abbildung 6.5.: Konfiguration der Tenants in Splunk Phantom

In Splunk Phantom sind nach Abschluss der Vorbereitungen ebenfalls insgesamt drei Benutzerkonten hinterlegt:

- *admin* als Systemadministratorin, mit der Berechtigung weitere Tenants und Benutzerinnen anzulegen, sowie mit Rechten für die Analyse- und Reaktionsfunktionen.
- *defaultadmin* als Systemadministrator, jedoch mit Zugriff nur auf den *default*-Tenant, sowie mit Rechten für Analysten.
- *Detection* als Benutzerkonto zum Erzeugen von Vorfällen per HTTP-API

## 7. Testdurchführung

Nachdem in den letzten Kapiteln zunächst die ANFORDERUNGEN sowie die TEST-SZENARIEN UND TESTFÄLLE für die Evaluierung von IT-Systemen zur automatischen IT-Sicherheitsvorfallsbehandlung aufgezeigt wurden, sind in den vergangenen beiden Kapiteln die Testobjekte ausgewählt (AUSWAHL DER IT-SYSTEME) und in eine Laborumgebung eingebunden (ENTWICKLUNG EINER LABORUMGEBUNG) worden. Die Dokumentation der Evaluationsergebnisse ist Inhalt dieses Kapitels. Es teilt sich dafür in drei Abschnitte auf: Zunächst wird die Anwendung der Testfälle auf die beiden Testobjekte TheHive/Cortex und Splunk Phantom dokumentiert, ehe dann im dritten Abschnitt die gewonnenen Daten ausgewertet werden.

Bei der Testdurchführung wurde für jeden Test der Status der Testerfüllung beurteilt. Ein Test kann dabei folgende Zustände haben.

- *Bestanden:* Dieser Test wurde ohne Einschränkungen bestanden.
- *Sinngemäß Bestanden:* Die geprüfte Funktion kann unter Zuhilfenahme von weiteren Funktionen wie angefordert umgesetzt werden.
- *Teilweise Bestanden:* Die Funktion ist nur teilweise, also nicht vollständig im System enthalten bzw. kann nur über große Umwege umgesetzt werden.
- *Nicht Bestanden:* Die Anforderungen aus dem Test wurden nicht erfüllt. Die dahinterliegende Funktion wurde nicht umgesetzt.

Das Maß der Erfüllung ergibt sich aus der Beurteilung, inwieweit der jeweilige Testgegenstand die im Testfall geforderte Funktion abbilden konnte. Das Ergebnis ist ebenfalls für jeden Testfall dokumentiert.

Zur besseren Lesbarkeit ist die Dokumentation der Testfälle für beide Untersuchungsgegenstände in unterschiedlichen Farben ausgeführt. [ID] steht in den Beispielen für die fortlaufende Identifikationsnummer jedes Testfalls und [Testszenario] für das jeweils zugrundeliegende Testszenario (siehe jeweils Kapitel TESTSZENARIEN UND TESTFÄLLE).

<b>Testfall [ID] ([Testszenario])</b>
Testdokumentation für TheHive/Cortex

<b>Testfall [ID] ([Testszenario])</b>
Testdokumentation für Splunk Phantom

## 7.1. TheHive/Cortex

Zunächst wurden die Testfälle auf den Untersuchungsgegenstand TheHive in Verbindung mit Cortex angewandt. Die dabei entstandenen Screenshots sind im Anhang ab Seite 135 ersichtlich.

<b>Testfall 1 (Benutzerverwaltung und -anmeldung)</b>
<b>Soll-Ergebnis:</b> Die neuen Benutzerkonten erscheinen in der Liste der Benutzerkonten des Systems und können zur Anmeldung am System benutzt werden. Die Benutzung ist ohne Anmeldung nicht möglich. <b>Ist-Ergebnis:</b> Alle angelegten Benutzerkonten konnten zur Anmeldung benutzt werden. Das Anlegen der Benutzerkonten wurde über das Menü Organisation -> Users -> Create new user vorgenommen. <b>Bestanden:</b> Bestanden <b>Fehler (Falls nicht bestanden):</b> Kein Fehler <b>Kommentar:</b> Keine Anmerkungen

**Testdurchführung:**

Lobmeyer, 09.05.20 17:00

**Testfall 2 (Vorfallserstellung)****Soll-Ergebnis:**

Der vom Detektionssystem gemeldete Vorfall erscheint im hier betrachteten System und kann mit einer Priorität versehen werden. Dem Vorfall können Teammitglieder zugeordnet werden. Die Kritikalität lässt sich auch im Nachhinein verändern.

**Ist-Ergebnis:**

Neue eingehende Meldungen können über die Alarm-Funktion eingebunden werden. Der Alarm kann dann klassifiziert und als neuer Vorfall aufgenommen werden.

**Bestanden:**

Teilweise bestanden

**Fehler (Falls nicht bestanden):**

Die Zuordnung ist immer nur einer Person möglich.

**Kommentar:**

Ein Softwarefehler [41], der verhindert, dass man die Zuordnung von Personen verändern kann, wurde nicht negativ ausgelegt, da er inzwischen behoben ist.

**Testdurchführung:**

Lobmeyer, 09.05.20 17:45

**Testfall 3 (Vorfallserstellung)****Soll-Ergebnis:**

Die Testperson erhält eine Benachrichtigung über den Ablauf der Frist.

**Ist-Ergebnis:**

Es existiert keine zeitgesteuerte Benachrichtigungsfunktion.

**Bestanden:**

Nicht bestanden

**Fehler (Falls nicht bestanden):**

Eine Funktion zur zeitgesteuerten Benachrichtigung ist nicht vorhanden.



**Kommentar:**

Diese Funktion ließe sich möglicherweise mit von TheHive unterstützten Webhooks und einer externen Software realisieren.

**Testdurchführung:**

Lobmeyer, 10.05.20 14:50

**Testfall 4 (Vorfallerstellung)****Soll-Ergebnis:**

Die Testperson wurde über den neuen Vorfall und über die Zuordnung benachrichtigt.

**Ist-Ergebnis:**

Es existiert keine Benachrichtigungsfunktion für die externe Benachrichtigung bei zugeordneten Aufgaben.

**Bestanden:**

Nicht bestanden

**Fehler (Falls nicht bestanden):**

Eine Funktion für diese Benachrichtigung ist nicht vorhanden.

**Kommentar:**

Innerhalb des Systems werden Benutzerinnen über zugeordnete Aufgaben informiert. Diese Funktion ließe sich möglicherweise mit von TheHive unterstützten Webhooks und einer externen Software realisieren.

**Testdurchführung:**

Lobmeyer, 10.5.20 14:55

**Testfall 5 (Vorfallsanzeige und -editierung)****Soll-Ergebnis:**

Der Testperson werden die zuvor erstellten Vorfälle angezeigt, die Daten des Vorfalls lassen sich anzeigen und bearbeiten. Die Verknüpfung von zwei Vorfällen funktioniert.

**Ist-Ergebnis:**

Nach dem Login kann ein Vorfall über die Startseite aufgerufen werden. Innerhalb des Vorfalls können die Daten bearbeitet werden.

**Bestanden:**

Bestanden

**Fehler (Falls nicht bestanden):**

Kein Fehler

**Kommentar:**

Das Verknüpfen der zwei Fälle funktioniert in der Version noch nicht, wurde in älteren Versionen jedoch implementiert und soll auch für die vorliegende Version noch implementiert werden<sup>a</sup>

**Testdurchführung:**

Lobmeyer, 10.05.20 15:10

<sup>a</sup>Im RC2 wurde die Funktion nachgereicht.

**Testfall 6 (Vorfallsanzeige und -editierung)****Soll-Ergebnis:**

Die Testperson kann sich mit einem nicht zugeordneten Benutzerkonto den angepassten Vorfall nicht anzeigen lassen. Der Zugriff mit einem zugeordneten Benutzerkonto funktioniert jedoch.

**Ist-Ergebnis:**

Es ist keine Funktion zur Beschränkung der Leserechte vorhanden.

**Bestanden:**

Nicht bestanden

**Fehler (Falls nicht bestanden):**

Eine Funktion für diese lesebeschränkung ist nicht vorhanden.

**Kommentar:**

Keine Anmerkungen

**Testdurchführung:**

Lobmeyer, 10.05.20 15:15

**Testfall 7 (Vorfallsanalyse)****Soll-Ergebnis:**

Die Person kann Artefakte hochladen und einem Benutzerkonto zur Analyse zuordnen.

**Ist-Ergebnis:**

Die Zuordnung von einzelnen Artefakten zur Analyse ist nicht möglich. Allerdings können über die Aufgabenverwaltung einzelne Aufgaben (wie etwa die Analyse eines spezifischen Artefakts) über den Dialog zur Verwaltung der Aufgaben innerhalb der Anzeige eines Vorfalls einem Benutzerkonto zugeordnet werden.

**Bestanden:**

Sinngemäß Bestanden

**Fehler (Falls nicht bestanden):**

Kein Fehler

**Kommentar:**

Keine Anmerkungen

**Testdurchführung:**

Lobmeyer, 10.05.20 17:06

**Testfall 8 (Vorfallsanalyse)****Soll-Ergebnis:**

Die Person kann Artefakte automatisiert untersuchen lassen.

**Ist-Ergebnis:**

Über Cortex können Analysetools angebunden werden, mit denen Observables (auch Dateien) untersucht werden können. In dem Testfall wurde eine Textdatei hochgeladen, anschließend konnte ein Analyzer zur Identifizierung des Dateityps mit dem Aktivieren einer Schaltfläche gestartet und das Artefakt dadurch untersucht werden.

**Bestanden:**

Bestanden

**Fehler (Falls nicht bestanden):**

Kein Fehler

**Kommentar:**

Keine Anmerkungen

**Testdurchführung:**

Lobmeyer, 10.05.20 17:22

### Testfall 9 (Vorfallsanalyse)

**Soll-Ergebnis:**

Die Testperson kann die IoC hinzufügen.

**Ist-Ergebnis:**

Siehe Testfall 8. Artefakte werden als Datei-Observable aufgenommen.

**Bestanden:**

Bestanden

**Fehler (Falls nicht bestanden):**

Kein Fehler

**Kommentar:**

Keine Anmerkungen

**Testdurchführung:**

Lobmeyer, 10.05.20 17:24

### Testfall 11 (Vorfallsreaktion)

**Soll-Ergebnis:**

Die Testperson kann Handlungsanweisungen für die verschiedenen Phasen eintragen sowie die Parameter, bei denen diese Handlungsanweisungen eingebunden werden sollen.

**Ist-Ergebnis:**

Über die Funktion Organisation -> Case Templates können neue Templates, mit darin enthaltenen Tasks (Aufgaben zur Vorfallsbearbeitung) angelegt werden. Diese Case Templates lassen sich nutzen, um SOP zu hinterlegen. Die Konfiguration von Parametern, die erfüllt sein müssen, damit das Template automatisch zugeordnet wird, ist nicht möglich.

**Bestanden:**

Teilweise bestanden

**Fehler (Falls nicht bestanden):**

Kein Fehler

**Kommentar:**

Die Berechtigung zum Anlegen von Case Templates ist standardmäßig nicht für

Nicht-Administratorinnen freigegeben. Dies kann aber angepasst werden.

**Testdurchführung:**

Lobmeyer, 09.05.20 17:11

### Testfall 12 (Vorfallsreaktion)

**Soll-Ergebnis:**

Die Testperson kann Schritte einer Handlungsanweisungen als durchgeführt bzw. entbehrlich markieren und zusätzlich Kommentare einfügen.

**Ist-Ergebnis:**

Aufgaben (Tasks) können durchgeführt und kommentiert werden. Zudem wird die Durchführung der Tasks mit einem Zeitstempel versehen. Einzelne Aufgabenschritte können auch gelöscht werden.

**Bestanden:**

Bestanden

**Fehler (Falls nicht bestanden):**

Kein Fehler

**Kommentar:**

Keine Anmerkungen

**Testdurchführung:**

Lobmeyer, 10.05.20 16:30

### Testfall 13 (Vorfallsreaktion)

**Soll-Ergebnis:**

Die zuvor für die Vorfallsautomation vorgesehene Aktion wird bei der Erstellung des neuen Vorfalls ausgelöst.

**Ist-Ergebnis:**

Über die Komponente Cortex können Aktionen ausgeführt werden. Diese müssen jedoch manuell ausgelöst werden.

**Bestanden:**

Teilweise bestanden

**Fehler (Falls nicht bestanden):**

Eine automatische Auslösung ist nicht vorgesehen.

**Kommentar:**

Die Funktion der automatischen Auslösung kann durch Drittsoftware und mit der Hilfe von Webhooks umgesetzt werden.

**Testdurchführung:**

Lobmeyer, 10.05.20 16:37

**Testfall 14 (Threat Sharing)****Soll-Ergebnis:**

Die Daten werden als Ereignis an die Threat-Sharing-Plattform exportiert.

**Ist-Ergebnis:**

Über die Funktion „Export“ in der Fallansicht kann ein Vorfall mit den IoC/Observables an eine angeschlossene MISP-Instanz geschickt werden.

**Bestanden:**

Bestanden

**Fehler (Falls nicht bestanden):**

Kein Fehler

**Kommentar:**

Keine Anmerkungen

**Testdurchführung:**

Lobmeyer, 10.05.20 17:37

**Testfall 15 (Threat Sharing)****Soll-Ergebnis:**

Die in der Meldung enthaltenen Daten werden im System dem bestehenden Vorfall zugeordnet.

**Ist-Ergebnis:**

Über die API kann MISP angesprochen werden, sodass neue MISP-Ereignisse automatisch als Alarm angezeigt werden und dem Vorfall zugeordnet werden können.

**Bestanden:**

Bestanden

**Fehler (Falls nicht bestanden):**

Kein Fehler

**Kommentar:**

Keine Anmerkungen

**Testdurchführung:**

Lobmeyer, 10.05.20 17:39

**Testfall 16 (Vorfalldokumentation)**

**Soll-Ergebnis:**

Im Zeitstrahl werden alle durchgeführten Aktionen angezeigt.

**Ist-Ergebnis:**

Über die Übersichtsseite eines Vorfalls kann der Livestream angezeigt werden, der alle Aktionen, die in dem Vorfall vorgenommen wurden, dokumentiert.

**Bestanden:**

Bestanden

**Fehler (Falls nicht bestanden):**

Kein Fehler

**Kommentar:**

Keine Anmerkungen

**Testdurchführung:**

Lobmeyer, 10.05.20 16:54

**Testfall 17 (Vorfallskommunikation)**

**Soll-Ergebnis:**

Die Testperson kann einen Management-Bericht verschicken.

**Ist-Ergebnis:**

Diese Funktion ist nicht vorhanden.

**Bestanden:**

Nicht bestanden

**Fehler (Falls nicht bestanden):**

Eine Benachrichtigung und allgemeinverständliche Zusammenfassung des Falls ist nicht vorgesehen.

**Kommentar:**

Es können zusätzliche Felder (Custom Fields) genutzt werden, um die Informationen für den Vorfall einzutragen. Hierbei ist aber ebenfalls kein Export möglich.

**Testdurchführung:**

Lobmeyer, 10.05.20 16:56

### Testfall 18 (Nachbereitung)

**Soll-Ergebnis:**

Die Notizen für die Nachbesprechung werden in einem speziellen Nachbesprechungsexport angezeigt. Für den Abschlussbericht werden alle Aktionen und insbesondere die Timeline exportiert.

**Ist-Ergebnis:**

Diese Funktion existiert nicht

**Bestanden:**

Nicht bestanden

**Fehler (Falls nicht bestanden):**

Die Funktion ist nicht Bestandteil des Systems.

**Kommentar:**

Es wäre möglich, hierfür ebenfalls ein Custom-Field einzubauen. Auch eine so gestaltete Berichtsfunktion existiert nicht.

**Testdurchführung:**

Lobmeyer, 10.05.20 17:00

### Testfall 19 (Nachbereitung)

**Soll-Ergebnis:**

Die Metriken „Anzahl der bearbeiteten Vorfälle pro Zeiteinheit“, „Zeit pro Vorfall“, „Anzahl der beteiligten Vorfallsteammitglieder“, die „Arten der Artefakte“ und die „Dauer der einzelnen Analysen pro Artefakt“ werden angezeigt.

**Ist-Ergebnis:**

Über die Funktion „Dashboards“ können benutzerdefinierte Dashboards konfiguriert werden.

**Bestanden:**

Bestanden



**Fehler (Falls nicht bestanden):**

Kein Fehler

**Kommentar:**

Keine Anmerkungen

**Testdurchführung:**

Lobmeyer, 10.05.20 17:44

## 7.2. Splunk Phantom

Im nächsten Schritt wurden die Testfälle mit Splunk Phantom durchgeführt. Auch dabei wurden Screenshots aufgenommen, die ab Seite 139 dargestellt sind.

### Testfall 1 (Benutzerverwaltung und -anmeldung)

**Soll-Ergebnis:**

Die neuen Benutzerkonten erscheinen in der Liste der Benutzerkonten des Systems und können zur Anmeldung am System benutzt werden. Die Benutzung ist ohne Anmeldung nicht möglich.

**Ist-Ergebnis:**

Alle angelegten Benutzerkonten konnten für den Login verwendet werden. Das Anlegen von Benutzern wurde über die Funktion Administration -> Users -> User vorgenommen

**Bestanden:**

Bestanden

**Fehler (Falls nicht bestanden):**

Kein Fehler

**Kommentar:**

Keine Anmerkungen

**Testdurchführung:**

Lobmeyer, 16.5.20 15:07

### Testfall 2 (Vorfallserstellung)

**Soll-Ergebnis:**

Der vom Detektionssystem gemeldete Vorfall erscheint im hier betrachteten System und kann mit einer Priorität versehen werden. Dem Vorfall können Teammitglieder zugeordnet werden. Die Kritikalität lässt sich auch im Nachhinein verändern.

**Ist-Ergebnis:**

Der vom Detektionssystem gemeldete Vorfall erscheint als Container. Dieser Container kann zum Vorfall hochgestuft werden, was dazu führt, dass ein Workbook zugeordnet werden kann. Zudem kann die Kritikalität angepasst und der Vorfall einem Benutzerkonto zugeordnet werden.

**Bestanden:**

Teilweise bestanden

**Fehler (Falls nicht bestanden):**

Die Zuordnung ist immer nur zu einer Person möglich.

**Kommentar:**

Keine Anmerkungen

**Testdurchführung:**

Lobmeyer, 16.05.20 15:57

**Testfall 3 (Vorfallserstellung)****Soll-Ergebnis:**

Die Testperson erhält eine Benachrichtigung über den Ablauf der Frist.

**Ist-Ergebnis:**

Es existiert keine Funktion, die eine automatische Benachrichtigung vorsieht.

**Bestanden:**

Nicht bestanden

**Fehler (Falls nicht bestanden):**

Eine Funktion zur zeitgesteuerten Benachrichtigung ist nicht vorhanden.

**Kommentar:**

Diese Funktion könnte sich durch automatisch ablaufende Abfragen oder durch die Benutzung von beim Erstellen eines Vorfalls automatisch ausgeführten Play-

books manuell einbauen lassen.

**Testdurchführung:**

Lobmeyer, 16.05.20 16:05

### Testfall 4 (Vorfallserstellung)

**Soll-Ergebnis:**

Die Testperson wurde über den neuen Vorfall und über die Zuordnung benachrichtigt.

**Ist-Ergebnis:**

Es existiert keine Benachrichtigungsfunktion zur Benachrichtigung bei zugeordneten Vorfällen oder Aufgaben.

**Bestanden:**

Nicht bestanden

**Fehler (Falls nicht bestanden):**

Eine Funktion zur Benachrichtigung ist nicht vorhanden.

**Kommentar:**

Diese Funktion könnte sich durch automatisch ablaufende Abfragen oder durch die Benutzung von beim Erstellen eines Vorfalls automatische ausgeführten Playbooks manuell einbauen lassen.

**Testdurchführung:**

Lobmeyer, 16.05.20 16:08

### Testfall 5 (Vorfallsanzeige und -editierung)

**Soll-Ergebnis:**

Der Testperson werden die zuvor erstellten Vorfälle angezeigt, die Daten des Vorfalls lassen sich anzeigen und bearbeiten. Die Verknüpfung von zwei Vorfällen funktioniert.

**Ist-Ergebnis:**

Nach dem Login können die Vorfälle über die Funktion „Cases,, aufgerufen werden. Nach Auswahl des Falls können die Informationen über den Vorfall ausgewählt und bearbeitet werden. Das Verknüpfen von Vorfällen ist nicht möglich.

**Bestanden:**

Teilweise bestanden

**Fehler (Falls nicht bestanden):**

Die Verknüpfung von Vorfällen ist nicht möglich.

**Kommentar:**

Keine Anmerkungen

**Testdurchführung:**

Lobmeyer, 16.05.20 16:16

### Testfall 6 (Vorfallsanzeige und -editierung)

**Soll-Ergebnis:**

Die Testperson kann sich mit einem nicht zugeordneten Benutzerkonto den angepassten Vorfall nicht anzeigen lassen. Der Zugriff mit einem zugeordneten Benutzerkonto funktioniert jedoch.

**Ist-Ergebnis:**

Es ist keine Funktion zur Beschränkung der Leserechte auf einen Vorfall vorhanden.

**Bestanden:**

Nicht bestanden

**Fehler (Falls nicht bestanden):**

Eine Funktion für diese lesebeschränkung ist nicht vorhanden.

**Kommentar:**

Keine Anmerkungen

**Testdurchführung:**

Lobmeyer, 16.05.20 16:18

### Testfall 7 (Vorfallsanalyse)

**Soll-Ergebnis:**

Die Person kann Artefakte hochladen und einem Benutzerkonto zur Analyse zuordnen.

**Ist-Ergebnis:**

Das Hochladen von Dateien ist möglich. Dateien können jedoch nicht speziell zur Analyse zugeordnet werden.

**Bestanden:**

Teilweise bestanden

**Fehler (Falls nicht bestanden):**

Keine Funktion zur Zuordnung zu Benutzerkonten.

**Kommentar:**

Keine Anmerkungen

**Testdurchführung:**

Lobmeyer, 16.05.20 17:46

**Testfall 8 (Vorfallsanalyse)****Soll-Ergebnis:**

Die Person kann Artefakte automatisiert untersuchen lassen.

**Ist-Ergebnis:**

Die Analyse von Daten ist mit der Nutzung von Apps möglich.

**Bestanden:**

Bestanden

**Fehler (Falls nicht bestanden):**

Kein Fehler

**Kommentar:**

Splunk Phantom verfügt über eine Vielzahl an Apps, mit denen der Funktionsumfang erweitert werden kann. Viele Anbieter von IT- oder IT-Sicherheitslösungen entwickeln diese Apps selbst, um Schnittstellen zu Splunk Phantom bereitstellen zu können.

**Testdurchführung:**

Lobmeyer, 16.05.20 18:02

**Testfall 9 (Vorfallsanalyse)****Soll-Ergebnis:**

Die Testperson kann die IoC hinzufügen.

**Ist-Ergebnis:**

Das Hinzufügen von Observables/IoC ist als Artefakt möglich.

**Bestanden:**

Bestanden

**Fehler (Falls nicht bestanden):**

Kein Fehler

**Kommentar:**

Keine Anmerkungen

**Testdurchführung:**

Lobmeyer, 26.05.20 18:05

### Testfall 11 (Vorfallsreaktion)

**Soll-Ergebnis:**

Die Testperson kann Handlungsanweisungen für die verschiedenen Phasen eintragen sowie die Parameter, bei denen diese Handlungsanweisungen eingebunden werden sollen.

**Ist-Ergebnis:**

Innerhalb von Splunk Phantom gibt es zwei Funktionen, mit denen das Ziel erreicht werden kann: Einerseits können sogenannte Workbooks erzeugt werden, die einzelne Unterschritte beinhalten, andererseits können auch Playbooks erstellt werden, die jedoch den Fokus auf die spätere Automatisierung einer Vorfallsreaktion legen. Das Erstellen von Workbooks wiederum benötigt besondere Rechte und soll das Abarbeiten von verschiedenen Playbooks erleichtern.

**Bestanden:**

Bestanden

**Fehler (Falls nicht bestanden):**

Kein Fehler

**Kommentar:**

In diesem Testfall findet keine Überprüfung des Berechtigungsmanagements statt.

**Testdurchführung:**

Lobmeyer, 16.05.20 15:27

### Testfall 12 (Vorfallsreaktion)

**Soll-Ergebnis:**

Die Testperson kann Schritte einer Handlungsanweisungen als durchgeführt bzw.

entbehrlich markieren und zusätzlich Kommentare einfügen.

**Ist-Ergebnis:**

Im Dashboard für den Vorfall werden die nächsten Tasks angezeigt. Diese können zugeordnet und bearbeitet werden. Für einen Vorfall können Tasks auch als entbehrlich markiert werden. Außerdem können zu Tasks Notizen erfasst werden.

**Bestanden:**

Bestanden

**Fehler (Falls nicht bestanden):**

Kein Fehler

**Kommentar:**

Keine Anmerkungen

**Testdurchführung:**

Lobmeyer, 16.5.20 16:50

### Testfall 13 (Vorfallsreaktion)

**Soll-Ergebnis:**

Die zuvor für die Vorfallsautomation vorgesehene Aktion wird bei der Erstellung des neuen Vorfalls ausgelöst.

**Ist-Ergebnis:**

Über das System konnte ein Testplaybook erstellt werden, das dazu führt, dass eine E-Mailbenachrichtigung für neue Vorfälle aktiviert wurde. Über eine Skriptsprache und Plugins ist die Erweiterung möglich.

**Bestanden:**

Bestanden

**Fehler (Falls nicht bestanden):**

Kein Fehler

**Kommentar:**

Splunk Phantom verfügt über eine Vielzahl an Apps, mit denen der Funktionsumfang erweitert werden kann. Viele Anbieter von IT- oder IT-Sicherheitslösungen entwickeln diese Apps selbst, um Schnittstellen zu Splunk Phantom bereitstellen zu können.

**Testdurchführung:**

Lobmeyer, 16.05.20 17:02

**Testfall 14 (Threat Sharing)****Soll-Ergebnis:**

Die Daten werden als Ereignis an die Threat-Sharing-Plattform exportiert.

**Ist-Ergebnis:**

IoC lassen sich per Action als Event an angebundene MISP-Instanzen senden, bzw. bestehenden Events hinzufügen.

**Bestanden:**

Bestanden

**Fehler (Falls nicht bestanden):**

Kein Fehler

**Kommentar:**

Keine Anmerkungen

**Testdurchführung:**

Lobmeyer, 16.05.20 17:02

**Testfall 15 (Threat Sharing)****Soll-Ergebnis:**

Die in der Meldung enthaltenen Daten werden im System dem bestehenden Vorfall zugeordnet.

**Ist-Ergebnis:**

Über die App können immer wieder Abfragen an eine MISP-Instanz abgesetzt werden. Mit diesem Ergebnis können dann neue Fälle erzeugt werden.

**Bestanden:**

Teilweise bestanden

**Fehler (Falls nicht bestanden):**

Kein Fehler

**Kommentar:**

Keine Anmerkungen



**Testdurchführung:**

Lobmeyer, 16.05.20 18:16

**Testfall 16 (Vorfallsdokumentation)****Soll-Ergebnis:**

Im Zeitstrahl werden alle durchgeführten Aktionen angezeigt.

**Ist-Ergebnis:**

Ein Zeitstrahl wird im Dashboard des Vorfalls angezeigt.

**Bestanden:**

Bestanden

**Fehler (Falls nicht bestanden):**

Kein Fehler

**Kommentar:**

Keine Anmerkungen

**Testdurchführung:**

Lobmeyer, 16.05.20 17:10

**Testfall 17 (Vorfallskommunikation)****Soll-Ergebnis:**

Die Testperson kann einen Management-Bericht verschicken.

**Ist-Ergebnis:**

Diese Funktion ist nicht vorhanden. Eine E-Mail müsste manuell erzeugt und mit den nötigen Informationen befüllt werden.

**Bestanden:**

Nicht bestanden

**Fehler (Falls nicht bestanden):**

Eine Benachrichtigung und allgemeinverständliche Zusammenfassung des Falls ist nicht vorgesehen.

**Kommentar:**

Es könnten zusätzliche Felder (Custom Fields) hierfür genutzt werden. Hierbei ist aber ebenfalls kein Export möglich.

**Testdurchführung:**

Lobmeyer, 16.05.20 17:12

**Testfall 18 (Nachbereitung)****Soll-Ergebnis:**

Die Notizen für die Nachbesprechung werden in einem speziellen Nachbesprechungsexport angezeigt. Für den Abschlussbericht werden alle Aktionen und insbesondere die Timeline exportiert.

**Ist-Ergebnis:**

Kommentare können nicht als relevant für die Nachbereitung markiert werden. Eine Report-Funktion besteht zwar, hier kann aber nicht explizit gefiltert werden.

**Bestanden:**

Nicht bestanden

**Fehler (Falls nicht bestanden):**

Kein Fehler

**Kommentar:**

Keine Anmerkungen

**Testdurchführung:**

Lobmeyer, 16.5.20 17:15

**Testfall 19 (Nachbereitung)****Soll-Ergebnis:**

Die Metriken „Anzahl der bearbeiteten Vorfälle pro Zeiteinheit“, „Zeit pro Vorfall“, „Anzahl der beteiligten Vorfallesammitglieder“, die „Arten der Artefakte“ und die „Dauer der einzelnen Analysen pro Artefakt“ werden angezeigt.

**Ist-Ergebnis:**

Es lassen sich die vorkonfigurierten Dashboards nutzen, diese beinhalten jedoch nicht die geforderten Werte. Die Anpassung dieses Dashboards ist jedoch nicht vorgesehen.

**Bestanden:**

Teilweise bestanden

**Fehler (Falls nicht bestanden):**

Kein Fehler

**Kommentar:**

Keine Anmerkungen

**Testdurchführung:**

Lobmeyer, 16.05.20 18:23

## 7.3. Untersuchung der Ergebnisse

Zur Untersuchung der Tests wurden die Testfälle und der Erfüllungsstatus zusammengefasst und zunächst die Verteilung des Erfüllungsstatus in Form jeweils eines Kreisdiagramms dargestellt:

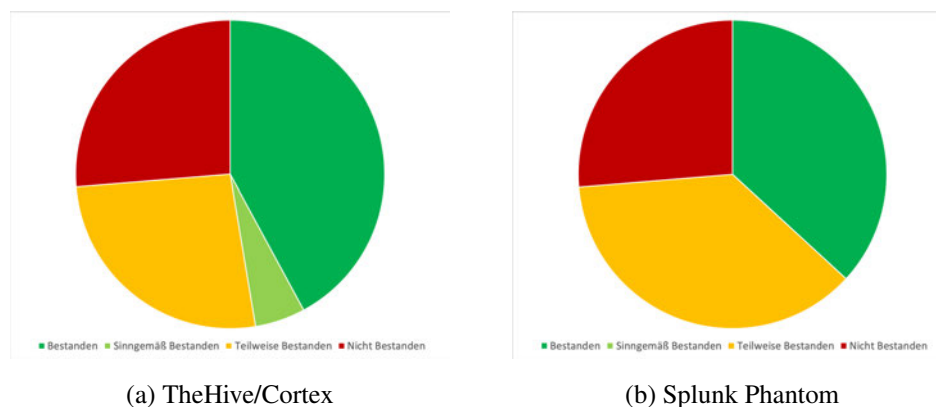


Abbildung 7.1.: Testerfüllung nach Anwendungen

Beide Anwendungen haben nicht alle Testfälle bestanden: Jeweils knapp über ein viertel der Tests wurden mit „Nicht bestanden“ bewertet. Erst auf den zweiten Blick lässt sich unter den Testfällen, die als „(Sinngemäß/Teilweise) Bestanden“ gewertet wurden ein Unterschied in der Erfüllung der Testfälle ermitteln. TheHive erfüllt die Anforderungen zu einem höheren Grad als die Anwendung Splunk Phantom: Während Splunk Phantom die Anforderungen nur bei zwei Fünftel der Tests mit „(Sinngemäß) Bestanden“ erfüllt, lässt sich diese Einschätzung bei TheHive auf knapp die Hälfte ausweiten. Wie im Kapitel TESTSZENARIEN UND TESTFÄLLE beschrieben, wurden die Testfälle nach Testszenarien, also nach zusammenhängenden Aktionen sortiert. Wertet man die Erfüllung der Testszenarien aus, so ergibt sich das in Abbildung 7.2 dargestellte Bild. Auf den ersten Blick ist hier sichtbar, dass beide Tools bei den gleichen Testszenarien nicht alle enthaltenen Testfälle und damit auch der damit geprüften Anforderun-

gen erfüllen können. Beide Anwendungen können nicht alle geforderten Funktionen der Vorfallerstellung, der Vorfallsanzeige und -editierung, der Vorfallskommunikation und der Nachbereitung erfüllen. Beide Anwendungen haben zudem den gleichen (Nicht-)Erfüllungsgrad in diesen Bereichen.

Diese Quote hätte sich bei beiden Systemen durch die Anpassung der Funktionsweise erhöhen können: Da im Rahmen dieser Arbeit jedoch geprüft werden sollte, ob diese Funktionen schon standardmäßig beinhaltet sind, wurde wie schon erwähnt auf die Anpassung (engl. *Customizing*) verzichtet. Durch das Customizing hätten weitere benutzerdefinierte Felder hinzugefügt werden können, die insbesondere weitere Daten für den Vorfall hätten beinhalten können.

Aber auch eine weitere Gemeinsamkeit lässt sich hier als Ursache für die Nichterfüllung einiger Testfälle ableiten: Die Zugangsbeschränkung für Vorfälle ist in beiden Anwendungen nicht vorgesehen. Es ist zudem nicht möglich, mehr als eine Person als Besitzer eines Vorfalles hinzuzufügen. Beide Einschränkungen folgen hier der gleichen Logik: Verantwortlich für die Bearbeitung (bzw. die Delegation von Tätigkeiten in einem Vorfall) ist immer eine Person. Diese Person legt durch organisatorische Maßnahmen fest, welche Aktivitäten von wem vorgenommen werden dürfen. Das zeigt sich auch in der fehlenden Option hinsichtlich der Leseinschränkung: Offensichtlich gehen beide Hersteller davon aus, dass innerhalb einer Organisation lesebeschränkungen nicht sinnvoll sind.

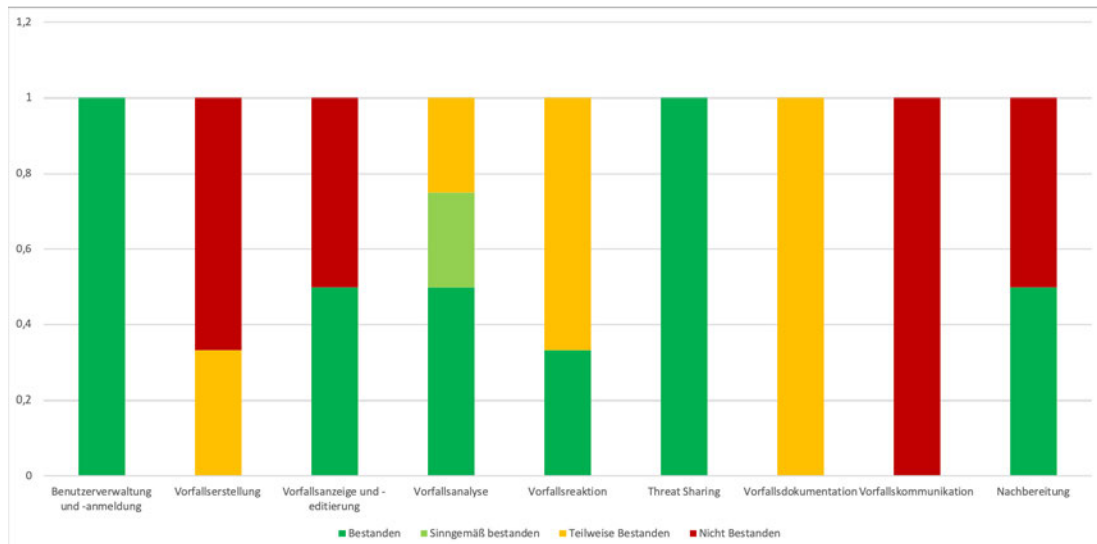
Unterschiede zwischen den IT-Systemen lassen sich im Bereich der Vorfallsreaktion finden: Die dortigen Anforderungen beziehen sich insbesondere auf die Durchführung von automatisierten Reaktionen und der Unterstützung in den Phasen Eindämmung, Behebung und Wiederherstellung. Hier ist Splunk Phantom stärker als TheHive, was sich insbesondere auf die starke Unterstützung der Automatisierung von Reaktionen durch die in der Software eingebaute Erstellung der Playbooks und der vorhandenen Plugins begründen lässt. Splunk Phantom hat in diesem Bereich eine deutliche Stärke, was auch durch die Fokussierung der Benutzerschnittstelle auf diesen Aspekt deutlich wird.

Im Gegensatz dazu liegt die Stärke von TheHive in der Dokumentation, der Analyse von IT-Sicherheitsvorfällen und in der Integration von MISP. Zudem bietet TheHive den Vorteil, dass die Entwicklung in einer offenen Community stattfindet, sodass die

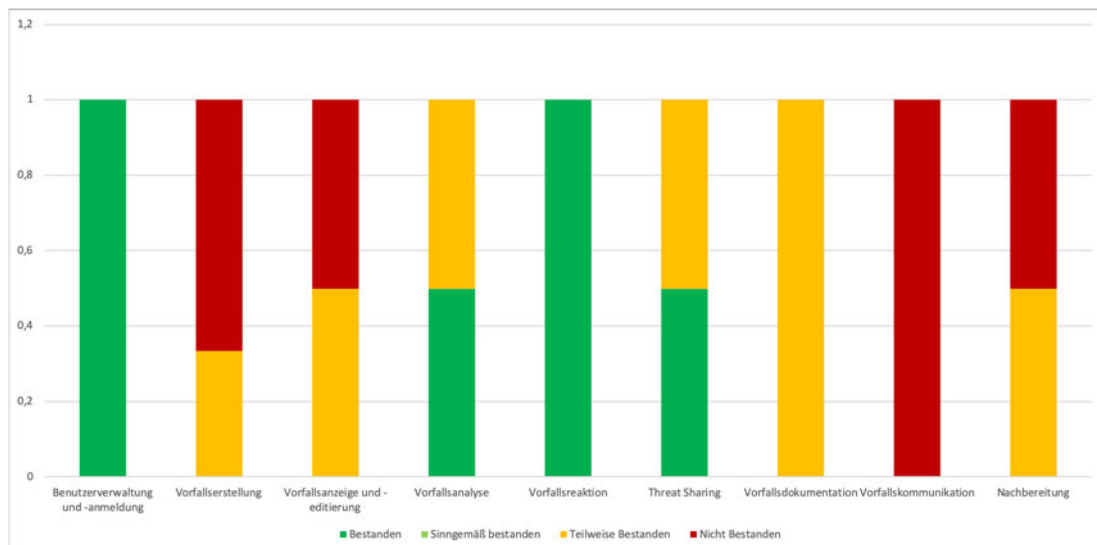
Anwendung auch durch Organisationen im Funktionsumfang erweitert werden können.

Den Bereich Kommunikation haben beide IT-Systeme gleichermaßen nicht erfüllt. Hintergrund dafür könnte sein, dass es mit *Customer-Relation-Managementsystemen* bzw. *Ticketsystemen* schon Lösungen gibt, die auf die Kommunikation mit Externen, bzw. Beteiligten innerhalb der Organisation ausgelegt und von den Herstellern als unabhängig von Systemen zur Automatisierung der Vorfallsreaktion gesehen werden könnten.

Der Unterschied im Ergebnis zwischen den beiden Systemen ist geringer ausgefallen als erwartet. Beide erfüllen zu einem fast gleichen Teil die einzelnen Anforderungen aus den Standards, aber beide zu einem relativ geringen Maß. Hieraus lässt sich die Frage ableiten, inwieweit die Standards überhaupt als Grundlage dienen, bzw. dienen könnten und sollten, um für die Hersteller als Inspiration für Anforderungen an ihre jeweiligen IT-Systeme zu dienen. Außerdem zeigt sich hierbei der Unterschied zwischen den eher theoretisch-orientierten Standardwerken und Produkten, die auf Basis von praktischer Arbeit entwickelt wurden.



(a) TheHive/Cortex



(b) Splunk Phantom

Abbildung 7.2.: Testerfüllung nach Testszenarien

## 8. Fazit und Ausblick

In diesem Kapitel sollen die vorherigen Ergebnisse zusammengefasst, der Lösungsansatz diskutiert und mögliche weitere Forschungsfragen angerissen werden.

### 8.1. Zusammenfassung

Ausgangslage war die Fragestellung, inwieweit bereits bestehende IT-Systeme bereits in der Lage sind, Organisationen bei der standardkonformen Reaktion auf IT-Sicherheitsvorfälle zu unterstützen.

Hierfür wurden zunächst auf Basis der beiden anerkannten Standards ([6] und [7]) Anforderungen ausgearbeitet, davon ausgehend, dass in dem Standard beschriebene Anforderungen auch als Funktion in einem System zur Automatisierung der Vorfallsreaktion vorhanden sein sollten. Um diese Anforderungen überprüfen zu können, wurden die Anforderungen in Arbeitsabläufen gruppiert, die als Testszenarien bezeichnet wurden. Um die Testszenarien für eine Testperson überprüfbar zu gestalten, wurden anschließend Testfälle aus diesen Szenarien abgeleitet. Nach der Auswahl der Testobjekte wurden diese Testobjekte in einer Laborumgebung aufgebaut. Diese Laborumgebung stellt sicher, dass die Umgebungsvariablen innerhalb des Systems gleich bleiben – und somit eine Reproduzierbarkeit der Testergebnisse gegeben ist. Schlussendlich wurden für beide Testobjekte die Tests durchgeführt und im Kapitel TESTDURCHFÜHRUNG dokumentiert und ausgewertet.

Zusammenfassend lässt sich die Frage nach der Möglichkeit einer **standardkonformen** Automatisierung des IT-Sicherheitsvorfallsreaktionsprozesse nur teilweise positiv beantworten.

Sowohl das IT-System TheHive, als auch das IT-System Splunk Phantom waren in der Lage, einen Teil der Anforderungen zu erfüllen. Beide IT-Systeme sind in der Lage, Vorfälle zu erstellen, die für diese Vorfälle durchgeführten Aktivitäten zu protokollieren und somit Unterstützung für eine Organisation zu bieten. Ebenfalls ist es in beiden Systemen - wenn auch zu einem unterschiedlichen Grad - möglich, Teilschritte, wie etwa die Analyse oder die Eindämmung zu automatisieren. Diese Automation unterscheidet sich in beiden Systemen jedoch hinsichtlich ihrer Granularität: Während Splunk Phantom hier offensichtlich, auch ausweislich der Dokumentation, der zusätzlich verfügbaren Apps und der durchgeführten Tests, seinen Schwerpunkt setzt, so liegt die Stärke von TheHive eher in der Abbildung eines Workflows für menschliche Analysten und die Integration in Threat-Intelligence-Systeme. Zu berücksichtigen ist außerdem, dass beide Aufwand erzeugen: Im Falle von Splunk Phantom ist dieser finanzieller Natur, da das System ab einer gewissen Größe kostenpflichtig ist, im Falle von TheHive heißt das, dass Personal zur Verfügung stehen muss, welches sich innerhalb der Organisation mit der Wartung und Weiterentwicklung, bzw. Fehlerbeseitigung beschäftigen kann.

Neben der Klärung der formulierten These, hat diese Arbeit auch noch einen weiteren Beitrag geleistet: Aus den Standards wurden Anforderungen an IT-Systeme zur Automatisierung der Vorfälle abgeleitet. Diese Anforderungen und die daraus abgeleiteten Testfälle lassen sich auch für die Evaluation von anderen IT-Systemen zur Vorfälleautomatisierung nutzen. Außerdem werden die hier getesteten IT-Systeme stetig weiterentwickelt, sodass eine Überprüfung auch späterer Versionen weiter möglich ist.

Insofern können auch die drei definierten Teilziele als erfüllt angesehen werden.

## 8.2. Diskussion des Untersuchungsansatzes

Auch wenn diese Arbeit inhaltlich die in der Aufgabenstellung gesteckten Fragen beantwortet hat, so bleibt auch noch Raum für Diskussion: Für diese Arbeit wurden Stan-



dards ausgewertet, die zum Teil schon im Jahr 2003 herausgegeben worden und somit über 15 Jahre alt sind.

Auch wenn sich die grundlegenden Prozessschritte der Vorfallsbehandlung sich seitdem nicht fundamental gewandelt haben und die Werke nach wie vor ihre Gültigkeit haben, so hat es jedoch in der Zwischenzeit auch Entwicklungen gegeben. Einerseits hat sich das Thema IT-Sicherheit seit dieser Zeit deutlich aus einer Nische der Forschung herausbewegt, andererseits haben sich auch neue Ansätze im Bereich der Detektion und Reaktion entwickelt. Ein Beispiel ist das so genannte *Threat-Hunting*, also das dauerhafte aktive Untersuchen von IT-Systemen und -Netzwerken auf Angriffsspuren. Auch Philosophien, bei denen die Verteidigerinnen davon ausgehen, dass ihr Netzwerk eigentlich schon von einer Angreiferin übernommen wurde, fassen immer mehr Fuß.

Für die Frage nach Optimierungsmöglichkeiten (bzw. Automatisierungsmöglichkeiten) heißt das, dass neben der Berücksichtigung der Standards auch die tatsächlichen Bedürfnisse von Vorfallsreaktionsteams in die Entwicklung von IT-Systemen einfließen müssen: Die Standards existieren nicht als Selbstzweck, sondern sollen Anhaltspunkte liefern, wie Vorfallsteams bzw. CSIRTs ihre Arbeit strukturieren und aufbauen können. Insofern heißt die unvollständige Erfüllung der beiden Testobjekte wahrscheinlich nicht, dass die Testobjekte für die Unterstützung der Vorfallsreaktion untauglich sind: Vielmehr können wir davon ausgehen, dass die Herstellerinnen ihren Fokus auf Funktionen legen, die sie für am Markt für aussichtsreich halten, bzw. die ihnen selber als sinnvoll erscheinen.

### 8.3. Ausblick

In einem weiteren Schritt könnten die Ergebnisse dieser Arbeit also um eine qualitative Umfrage erweitert werden, in der beispielsweise IT-Sicherheitsvorfallsteams beide hier betrachteten IT-Systeme testen bzw. in ihren täglichen Alltag einbringen und vor und nach der Einführung der Software Fragen zum Aufwand der Vorfallsbearbeitung geben. Eine andere Möglichkeit wäre, in einem Testlabor (zum Beispiel bei einer Schulungsanbieterin für die Durchführung von Schulungen im Bereich der IT-Sicherheitsvorfallsreaktion) Teams beide Produkte miteinander vergleichen zu

lassen.

Neben der Evaluation dieser Systeme gibt es auch noch weitere Forschungsthemen, die im Bereich der Automatisierung von IT-Sicherheitsvorfallsreaktionen akut werden könnten. Einerseits gibt es einige interessante Veröffentlichungen [42], die daran arbeiten, erkannte Angreiferinnentechniken üblichen Reaktionsmechanismen gegenüber zu stellen. Das wäre die Grundlage, für eine tatsächlich automatisch vorgenommene Reaktion, weil dies bedeutet, dass ein IT-System auf einen erkannten Alarm mit einer Gegenreaktion reagieren kann (wie zum Beispiel [43] für die Simulation von Angreiferinnen). Das Problem könnte bei diesem Vorgehen jedoch sein, dass dieser Ansatz einem Schachspiel gleicht, also ein Zug auf den nächsten folgt. In der Realität ist es allerdings so, dass die Angreiferin schon unerkannt mehrere Schritte vorgenommen haben könnte, ohne das die Verteidigerin das gemerkt hat. Um die Schachmetapher nochmals zu bemühen: Die Angreiferin hat also mehrere Züge Vorsprung, ehe die Verteidigerin überhaupt erst reagieren kann. Neben dieser Schrittweisen Reaktion könnte auch ein anderer Lösungsansatz weiter erforscht werden: Nämlich indem sich die Mechanismen des Machine-Learning zunutze gemacht werden: Solch ein System könnte möglicherweise anhand von erlernten Reaktionsmaßnahmen auf bekannte Angriffe lernen, wie auf solche IT-Sicherheitsvorfälle reagiert werden muss. Vergleichbare Ansätze existieren bereits für den Bereich der Detektion von IT-Sicherheitsvorfällen.

In jedem Fall müssen die Möglichkeiten für den Schutz von Organisationen und damit auch Detektions- und Reaktionsmechanismen erweitert werden, um das bereits in der EINLEITUNG skizzierte Ressourcenproblem zu lösen und damit eine Chancengleichheit zwischen Verteidigerin und Angreiferin zu etablieren.

## 9. Literaturverzeichnis

- [1] SAUERMAN, Michael ; GESCHNONNECK, Alexander: e-Crime in der deutschen Wirtschaft / KPMG AG. 2019. – Study
- [2] SCHÜTZE, Julia: *Warum dem Staat IT-Sicherheitsexpert:innen fehlen*. Stiftung Neue Verantwortung, 2018
- [3] BING, Christopher ; SCHECKTMAN, Joel: Inside The UAE's Secret Hacking Team of American Mercenaries. In: *Reuters Online* (2019), Januar. <https://www.reuters.com/investigates/special-report/usa-spying-raven/>, Abruf: 1. Januar 2020
- [4] BUNDESAMT FÜR SICHERHEIT IN DER INFORMATIONSTECHNIK: *Was tun im IT-Notfall?* [https://www.bsi.bund.de/DE/Presse/Pressemitteilungen/Presse2019/Notfallkarte\\_260919.html](https://www.bsi.bund.de/DE/Presse/Pressemitteilungen/Presse2019/Notfallkarte_260919.html). Version: 9 2019, Abruf: 30. Dezember 2019
- [5] TRACY, Richard P.: IT Security Management and Business Process Automation: Challenges, Approaches, and Rewards. In: *Information Systems Security* 16 (2007), Nr. 2, S. 114–122
- [6] CICHONSKI, Paul ; MILLAR, Tom ; GRANCE, Tim ; SCARFONE, Karen: *Computer Security Incident Handling Guide*. Revision 2. National Institute of Standards and Technology, U.S. Department of Commerce, 2012 (Recommendations of the National Institute of Standards and Technology)
- [7] WEST-BROWN, Moira J. ; STIKVOORT, Don ; KOSSAKOWSKI, Klaus-Peter ; KILLCRECE, Georgia ; RUEFLE, Robin ; ZAJICEK, Mark: *Handbook for Computer Security Incident Response Teams*. CarnegieMellon University, Software Engineering Institute, 2003

- [8] WERLINGER, R. ; MULDER, K. ; HAWKEY, K. ; BEZNOSOV, K.: Preparation, detection, and analysis: the diagnostic work of IT security incident response. In: *Information Management and Security* 18 (2010), Nr. 1, S. 26–42
- [9] FRAUNHOLZ, D. ; ANTON, S.D. ; SCHOTTEN, H. D.: Introducing GAMfIS: A generic attacker model for information security. In: *25th International Conference on Software, Telecommunications and Computer Networks (SoftCOM)* Bd. 25 IEEE, IEEE, 2017
- [10] MITRE: *MITRE ATT&CK*. <https://attack.mitre.org>. Version: 12 2019, Abruf: 30. Dezember 2019
- [11] NG, Boon-Yuen ; KANKANHALLI, Atreyi ; XU, Yunjie (.: Studying users' computer security behavior: A health belief perspective. In: *Decision Support Systems* 46 (2009), March, Nr. 4, S. 815–825
- [12] BUNDESAMT FÜR SICHERHEIT IN DER INFORMATIONSTECHNIK: Leitfaden IT-Forensik / Bundesamt für Sicherheit in der Informationstechnik. 2011 (March). – Forschungsbericht
- [13] PATEL, Ahmed ; QASSIM, Qais ; WILLS, Christopher: A survey of intrusion detection and prevention systems. In: *Information Management & Computer Security* 18 (2010), Nr. 4, S. 277–290
- [14] SYKOSCH, Arnold ; OHM, Marc ; MEIER, Michael: Hunting Observable Objects for Indication of Compromise. In: *ARES 2018: Proceedings of the 13th International Conference on Availability, Reliability and Security* Bd. 13, 2018, S. 1–8
- [15] STEFFENS, Timo: *Auf der Spur der Hacker*. Springer Vieweg, 2018
- [16] YOON, Jungsang ; DUNLAP, Stephen ; BUTTS, Jonathan ; RICE, Mason ; RAMSEY, Benjamin: Evaluating the readiness of cyber first responders responsible for critical infrastructure protection. In: *International Journal of Critical Infrastructure Protection* 13 (2016), June, S. 13–27
- [17] ROBERTS, Scott J. ; BROWN, Rebekah: *Intelligence Driven Incident Response*. O'Reilly, 2017

- [18] DEUTSCHER CERT-VERBUND: *Überblick*. <https://www.cert-verbund.de>.  
Version: April 2020, Abruf: 19. April 2020
- [19] CYBER SECURITY SHARING & ANALYTICS E.V.: *CSSA*. <https://www.cssa.de>.  
Version: April 2020, Abruf: 19. April 2020
- [20] FIRST: *Information Exchange Policy (IEP) - Version 1.0*. <https://www.first.org/iep/>.  
Version: April 2020, Abruf: 19. April 2020
- [21] EUROPÄISCHE UNION: *Datenschutz-Grundverordnung*. April 2016
- [22] SAUERWEIN, Clemens ; SILLABER, Christian ; MUSSMANN, Andreas ; BREU, Ruth: Threat Intelligence Sharing Platforms: An Exploratory Study of Software Vendors and Research Perspectives. In: *Wirtschaftsinformatik 2017: Track 9 - Information Privacy and Information Security*, University of St. Gallen, 2017
- [23] WAGNER, Cynthia ; DULAUNOY, Alexandre ; WAGENER, Gérard ; IKLODY, Andras: MISP - The Design and Implementation of a Collaborative Threat Intelligence Sharing Platform. In: *WISCS '16: Proceedings of the 2016 ACM on Workshop on Information Sharing and Collaborative Security*, 2016, S. 49–56
- [24] HRUSCHKA, Peter: *Business Analysis und Requirements Engineering*. 2. Carl Hanser Verlag, 2019
- [25] WITTE, Frank: *Testmanagement und Softwaretest, Theoretische Grundlagen und praktische Umsetzung*. Springer Vieweg, 2016
- [26] VEREIN ZUR WEITERENTWICKLUNG DES V-MODELL XT E.V.: *Das V-Modell XT - Das deutsche Referenzmodell für Systementwicklungsprojekte*. <http://ftp.tu-clausthal.de/pub/institute/informatik/v-modell-xt/Releases/2.3/Dokumentation/V-Modell-XT-HTML/index.html>.  
Version: Mai 2020, Abruf: 24.05.2020
- [27] PALO ALTO NETWORKS: *Cortex XSOAR - Security Orchestration, Automation and Response (SOAR)*. <https://www.paloaltonetworks.com/cortex/xsoar>.  
Version: Mai 2020, Abruf: 17.05.2020

- [28] SPLUNK: *Phantom*. [https://www.splunk.com/en\\_us/software/splunk-security-orchestration-and-automation.html](https://www.splunk.com/en_us/software/splunk-security-orchestration-and-automation.html). Version: Mai 2020, Abruf: 17.05.2020
- [29] THEHIVE PROJECT: *TheHive Project*. <https://thehive-project.org>. Version: Mai 2020, Abruf: 27.05.2020
- [30] THEHIVE PROJECT: *TheHive 4 is Here, Finally!* <https://blog.thehive-project.org/2020/03/02/thehive-4-is-here-finally/>. Version: März 2020, Abruf: 17.05.2020
- [31] THEHIVE PROJECT: *Workflow in TheHive*. <https://raw.githubusercontent.com/TheHive-Project/TheHive/master/images/thehive-workflow.png>. Version: Mai 2020, Abruf: 26.05.2020
- [32] SPLUNK PHANTOM: *Splunk Phantom Overview v8*. [https://docs.splunk.com/File:Splunk\\_Phantom\\_Overview\\_v8.png](https://docs.splunk.com/File:Splunk_Phantom_Overview_v8.png). Version: Mai 2020, Abruf: 26.05.2020
- [33] COMPUTER INCIDENT RESPONSE CENTER LUXEMBOURG: *Latest MISP Appliances*. <https://www.circl.lu/misp-images/latest/>. Version: Mai 2020, Abruf: 17.05.2020
- [34] MISP PROJECT: *MISP Default Feeds*. <https://www.misp-project.org/feeds/>. Version: Mai 2020, Abruf: 17.05.2020
- [35] THEHIVE PROJECT: *TheHive Manual - Installation with .deb*. [https://github.com/TheHive-Project/TheHiveDocs/blob/master/TheHive4/Installation/Install\\_deb.md](https://github.com/TheHive-Project/TheHiveDocs/blob/master/TheHive4/Installation/Install_deb.md). Version: April 2020, Abruf: 3. Mai 2020
- [36] THEHIVE PROJECT: *Cortex Installation Guide*. <https://github.com/TheHive-Project/CortexDocs/blob/master/installation/install-guide.md>. Version: Dezember 2019, Abruf: 3. Mai 2020
- [37] THEHIVE PROJECT: *TheHive Manual - Connectors*. <https://github.com/TheHive-Project/TheHiveDocs/blob/master/TheHive4/Administration/Connectors.md>. Version: März 2020, Abruf: 3. Mai 2020

- [38] THEHIVE PROJECT ON GITHUB: *GitHub Issue: Change application.conf for MISP connection*. <https://github.com/TheHive-Project/TheHiveDocs/issues/150>. Version: März 2020, Abruf: 3. Mai 2020
- [39] SPLUNK: *Phantom Documentation - Installation*. <https://docs.splunk.com/Documentation/Phantom/4.8/Install/InstallOVA>. Version: Januar 2020, Abruf: 3. Mai 2020
- [40] SPLUNK: *Phantom Documentation - Login*. <https://docs.splunk.com/Documentation/Phantom/4.8/Install/Login>. Version: Februar 2020, Abruf: 3. Mai 2020
- [41] THEHIVE PROJECT ON GITHUB: *GitHub Issue: Assignee is not changeable*. <https://github.com/TheHive-Project/TheHive/issues/1243>. Version: Mai 2020, Abruf: 25.05.2020
- [42] ATC PROJECT: *ATC RE&CT*. <https://atc-project.github.io/atc-react/>. Version: Mai 2020, Abruf: 27.05.2020
- [43] APPLEBAUM, Andy ; MILLER, Doug ; STROM, Blake ; KORBAN, Chris ; WOLF, Ross: Intelligent, automated red team emulation. In: *Proceedings of the 32nd Annual Conference on Computer Security Applications* Bd. 32, 2016, S. 363–373

## 10. Ehrenwörtliche Erklärung

Ich versichere hiermit ehrenwörtlich, dass ich meine vorliegende Abschlussarbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel - insbesondere keine im Quellenverzeichnis nicht benannten Internet-Quellen - benutzt habe.

Die Arbeit wurde vorher nicht in einem anderen Prüfungsverfahren eingereicht und die eingereichte schriftliche Fassung entspricht derjenigen auf dem elektronischen Speichermedium.

Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter Angabe der Quellen kenntlich gemacht.

 28. Mai 2020 .....



# 11. Thesen

1. IT-Sicherheitsvorfälle stellen eine Bedrohung für Organisationen dar. Dies wird durch einen Mangel an IT-Spezialisten, insbesondere im Bereich der IT-Sicherheit verschärft.
2. Neben der Prävention und der Detektion ist auch der Reaktionsprozess ein wichtiger Faktor, um die IT-Sicherheit einer Organisation zu verbessern.
3. IT-Systeme zur Automatisierung der Vorfallsreaktion können den Mangel an Personal erträglicher machen.
4. Die Standards NIST SP 800-61R2 und CMU/SEI-2003-HB-002 helfen dabei, die Bedarfe von IT-Sicherheitsvorfallsteams (CSIRTs) zu beschreiben.
5. Bestehende IT-Systeme eignen sich teilweise, diese Bedarfe zu decken und somit zu einer Entlastung der Teams zu führen.
6. Nicht alle Anforderungen der Standards müssen durch die betrachteten IT-Systeme erfüllt werden, um dennoch eine Entlastung für Teams zu bieten.

# A. Anhang

## Konfigurationsdateien

In der Datei `dhcpd.conf` wird ein virtuelles Netzwerk konfiguriert, sodass anschließend die dort enthaltenen Hosts miteinander kommunizieren können.

```
1 # Configuration file for ISC 2.0 vmnet-dhcpd operating on vmnet5.
2 #
3 # This file was automatically generated by the VMware configuration
  program.
4 # See Instructions below if you want to modify it.
5 #
6 # We set domain-name-servers to make some DHCP clients happy
7 # (dhclient as configured in SuSE, TurboLinux, etc.).
8 # We also supply a domain name to make pump (Red Hat 6.x) happy.
9 #
10
11
12 ##### VMNET DHCP Configuration. Start of "DO NOT MODIFY SECTION"
   #####
13 # Modification Instructions: This section of the configuration file
   contains
14 # information generated by the configuration program. Do not modify
   this
15 # section.
16 # You are free to modify everything else. Also, this section must
   start
17 # on a new line
18 # This file will get backed up with a different name in the same
   directory
19 # if this section is edited and you try to configure DHCP again.
```

---

```

20
21 # Written at: 05/03/2020 11:59:06
22 allow unknown-clients;
23 default-lease-time 1800;                # default is 30 minutes
24 max-lease-time 7200;                    # default is 2 hours
25
26 subnet 172.31.6.0 netmask 255.255.255.0 {
27     range 172.31.6.128 172.31.6.254;
28     option broadcast-address 172.31.6.255;
29     option domain-name-servers 172.31.6.2;
30     option domain-name localdomain;
31     default-lease-time 1800;                # default is 30 minutes
32     max-lease-time 7200;                    # default is 2 hours
33     option netbios-name-servers 172.31.6.2;
34     option routers 172.31.6.2;
35 }
36 host vmnet5 {
37     hardware ethernet 00:50:56:C0:00:05;
38     fixed-address 172.31.6.1;
39     option domain-name-servers 0.0.0.0;
40     option domain-name "";
41     option routers 0.0.0.0;
42 }
43 ##### VMNET DHCP Configuration. End of "DO NOT MODIFY SECTION"
44 #####
45 host MISP_TI {
46     hardware ethernet 00:50:56:3A:73:80;
47     fixed-address 172.31.6.11;
48 }
49 host TheHive4RC1 {
50     hardware ethernet 00:50:56:24:DB:98;
51     fixed-address 172.31.6.21;
52 }
53
54 host SplunkPhantom {
55     hardware ethernet 00:50:56:28:9F:C2;
56     fixed-address 172.31.6.31;

```

57 }

### Listing A.1: VMware Konfigurationsdatei dhcpd.conf

Die Konfigurationsdatei `application.conf` beinhaltet alle notwendigen Parameter, um eine Instanz von TheHive zu konfigurieren.

```
1 ###
2 ## Documentation is available at https://github.com/TheHive-Project/
   TheHiveDocs/TheHive4
3 ###
4
5 ## Include Play secret key
6 # More information on secret key at https://www.playframework.com/
   documentation/2.8.x/ApplicationSecret
7 include "/etc/thehive/secret.conf"
8
9 ## Database configuration
10 db.janusgraph {
11   storage {
12     ## Cassandra configuration
13     # More information at https://docs.janusgraph.org/basics/
       configuration-reference/#storagecql
14     backend: cql
15     hostname: ["127.0.0.1"]
16     # Cassandra authentication (if configured)
17     // username: "thehive"
18     // password: "password"
19     cql {
20       cluster-name: thp
21       keyspace: thehive
22     }
23   }
24 }
25
26 ## Attachment storage configuration
27 storage {
28   ## Local filesystem
29   provider: localfs
30   localfs.directory: /opt/thp_data/files/thehive
```

```

31 }
32
33 ## CORTEX configuration
34 # More information at https://github.com/TheHive-Project/TheHiveDocs
    /TheHive4/Administration/Connectors.md
35 # Enable Cortex connector
36 play.modules.enabled += org.thp.thehive.connector.cortex.
    CortexModule
37 cortex {
38   servers: [
39     {
40       name: "Cortex-Internal"
41       url: "http://localhost:9001" # URL of Cortex instance
42       auth {
43         type: "bearer"
44         key: "[REDACTED]" # Cortex API key
45       }
46       ws {} # HTTP client configuration (SSL
    and proxy)
47     }
48   ]
49 }
50
51 ## MISP configuration
52 # More information at https://github.com/TheHive-Project/TheHiveDocs
    /TheHive4/Administration/Connectors.md
53 # Enable MISP connector
54 play.modules.enabled += org.thp.thehive.connector.misp.MispModule
55 misp {
56   interval: 1 hour
57   servers: [
58     {
59       name = "local" # MISP name
60       url = "https://misp" # URL or MISP
61       auth {
62         type = "key"
63         key = "[REDACTED]" # MISP API key
64       }
65       ws

```

---

```

66     {}                                # HTTP client configuration (SSL
    and proxy)
67 }
68 ]
69 }

```

Listing A.2: TheHive Konfigurationsdatei application.conf

## Erzeugen von Meldungen mittels curl

```

1 curl -XPOST -H 'Authorization: Bearer [REDACTED]' -H 'Content-Type:
    application/json' http://172.31.6.21:9000/api/alert -d '{
2   "title": "Testalarm No 1",
3   "description": "First Alarm Test",
4   "type": "detection",
5   "date": 1589029747566,
6   "source": "DetectionSystemMockup",
7   "sourceRef": "1234"
8 }'

```

Listing A.3: Erzeugung einer Alarmmeldung in TheHive mittels curl

```

1 curl -k -u ":[REDACTED]" https://172.31.6.31/rest/container \
2 -d '{
3   "artifacts": [ ],
4   "custom_fields": {},
5   "data": { },
6   "description": "This is another example.",
7   "label": "events",
8   "name": "TestIncident No2",
9   "run_automation": false,
10  "container_type": "default"
11 }'

```

Listing A.4: Erzeugung einer Alarmmeldung in Splunk Phantom mittels curl

## Screenshots

## Case # 1 - Testalarm No 1 - Besserer Titel

 Created by Teammitglied Nummer 1  Sat, May 9th, 2020 17:44 +02:00  **1 alert**

 Details

 Tasks **0**

 Observables **0**

### Summary

<b>Title</b>	Testalarm No 1 - Besserer Titel
<b>Severity</b>	<b>L</b>
<b>TLP</b>	TLP:AMBER
<b>PAP</b>	PAP:AMBER
<b>Assignee</b>	Teammitglied Nummer 1
<b>Date</b>	Sat, May 9th, 2020 17:44 +02:00
<b>Tags</b>	<i>Not Specified</i>

### Additional information

*No additional information have been specified*

### Description



First Alarm Test. Jetzt ist der Text auch noch ausführlicher.

Abbildung A.1.: Erstellter Vorfall in TheHive

**Case basic information**

**Template name** \* TestTemplate für Cases  
This name should be unique

**Title prefix**  
This is used to prefix the case name

**Severity** 4  
This will be the default case severity

**TLP** TOP SECRET  
This will be the default case TLP

**P&P** NO P&P  
This will be the default case P&P

**Tags**  
These will be the default case tags

**Description** \* Default TestTemplate für Cases

**Tasks (4)**

- [Analyse] Erfledige ersten Schritt Edit Delete
- [Analyse] Erfledige zweiten Schritt Edit Delete
- [Containment] Sperrung Benutzerkonto Edit Delete
- [Containment] Wechsle das Passwort Edit Delete

**Custom fields (0)**

No custom fields have been added. [Add a custom field](#)

[Delete case template](#) \* Required field [Export case template](#) [Save case template](#)

Abbildung A.2.: Erstellung eines Case Templates in TheHive

Group	Task	Date	Assignee	Actions
Analyse	Erfledige ersten Schritt <small>Started a minute ago</small>	Sun, May 10th, 2020 16:24 +02:00	Teammitglied Nummer 1	Close
Analyse	Erfledige zweiten Schritt			Start
Containment	Sperrung Benutzerkonto			Start
Containment	Wechsle das Passwort			Start

Abbildung A.3.: Abarbeiten von Tasks in TheHive

Group	Task	Date	Assignee	Actions
default	Analyze Testobservable		Teammitglied Nummer 1	Delete Start

Abbildung A.4.: Abarbeiten von Tasks in TheHive





### Testalarm No 1








Event ID	1164
UUID	5eb81cc1-2174-4f4e-afc6-0310ac1f060b 
Creator org	MT-Test
Owner org	MT-Test
Email	misp.thehive@lab.
Tags	 
Date	2020-05-09
Threat Level	Low
Analysis	Initial
Distribution	Your organisation only  
Info	Testalarm No 1
Published	No
#Attributes	0 (0 Object)
First recorded change	
Last change	2020-05-10 17:24:49
Modification map	
Sightings	0 (0) - restricted to own organisation only. 

Abbildung A.8.: Export von TheHive in MISP

Let's configure a few administrative settings

### BASIC SETTINGS

Please set a new administrative password  
.....

Please confirm your password  
.....

Your Company name  
TestCompany

System time zone  
UTC

Contact email for system maintenance  
e.g. admin@splunk.com

Base URL for this Phantom instance  
https://172.31.6.128/

Phantom requires an email server to send users email for action approvals, when SLAs are breached, and when items that they are tracking change. You can configure a server now or skip this and configure one later.

Configure

### ASSET SETTINGS

Asset name  
smtp

Server IP/Hostname

SSL Method  
None

▶ ADDITIONAL INFORMATION *(optional)*

SAVE AND CONTINUE

Abbildung A.9.: Konfiguration von Splunk Phantom

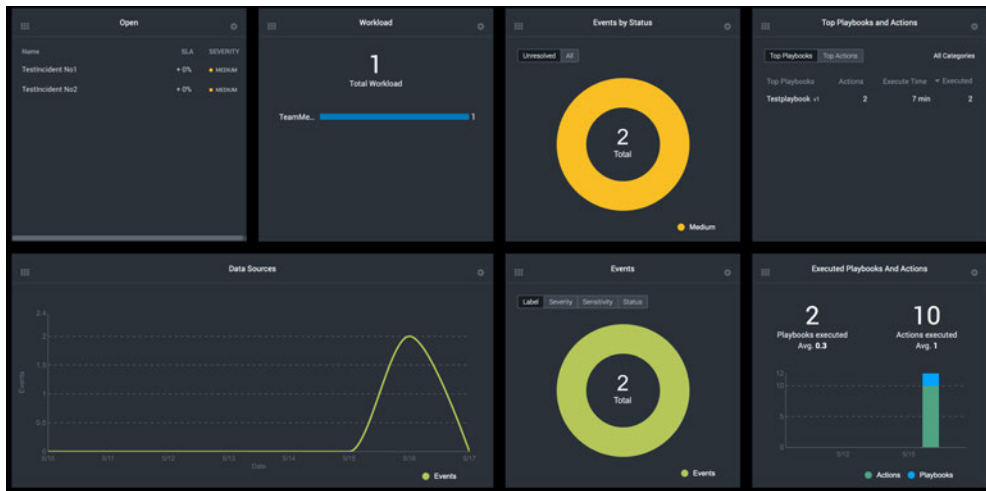


Abbildung A.10.: Dashboard von Splunk Phantom

The 'Edit events' form contains the following fields:

- Event Name:** TestIncident No1
- Label:** events
- Advanced:**
  - Tenant:** Default
  - Status:** New
  - Owner:** TeamMember1
  - Severity:** Low
  - Sensitivity:** TLP-Amber
  - SLA Expires:** 05/17/2020 01:49 pm
  - Description:** This is an example.
  - Tags:**

Buttons: CANCEL, SAVE

Abbildung A.11.: Erstellter Vorfall in Splunk Phantom

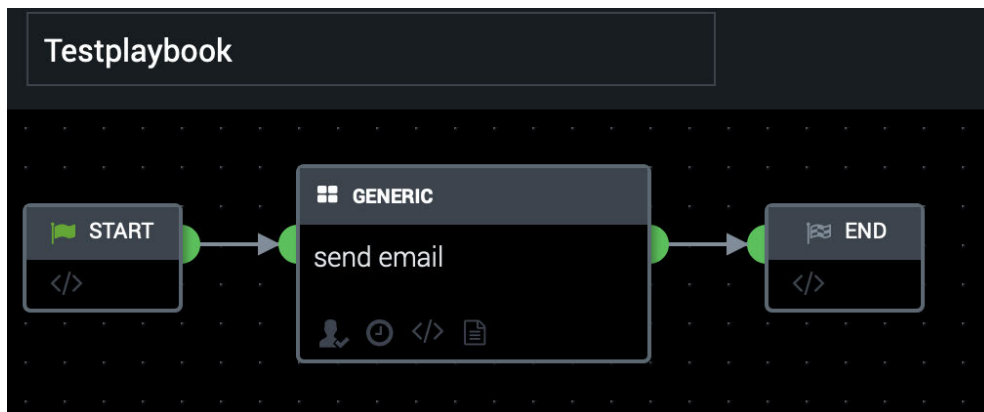


Abbildung A.12.: Ansicht eines Playbooks in Splunk Phantom

The screenshot shows a 'Workbook' in Splunk Phantom titled 'NIST 800-61'. It has a 'Set as default' button and an 'EDIT' button. The workbook is organized into three main sections, each with a 'Phase SLA' dropdown and a list of tasks. The tasks are organized into columns: TASK NAME, SLA, ACTIONS, PLAYBOOKS, and OWNER.

Section	Phase SLA	TASK NAME	SLA	ACTIONS	PLAYBOOKS	OWNER
Detection	-	Determine if an incident has occurred				
		Analyze precursors and indicators				
		Look for correlating information				
		Perform research				
		Confirmed incident				
Analysis and Containment	-	Determine functional impact				
		Determine information impact				
		Determine recoverability effort				
		Prioritize incident				
		Report incident				
		Contain incident				
Eradicate	-					

Abbildung A.13.: Ansicht eines Workbooks in Splunk Phantom









<b>Event ID</b>	1165
<b>UUID</b>	5ec00fd1-9bac-45c7-a4b3-089fac1f060b 
<b>Creator org</b>	<a href="#">MT-Test</a>
<b>Owner org</b>	<a href="#">MT-Test</a>
<b>Email</b>	misp.phantom@lab.i
<b>Tags</b>	   
<b>Date</b>	2020-05-16
<b>Threat Level</b>	Low
<b>Analysis</b>	Initial
<b>Distribution</b>	This community only   
<b>Info</b>	TestEvent created by Splunk Phantom

Abbildung A.14.: Export von Splunk Phantom in MISP