IT-Sicherheitsanalyse durch NAC-Systeme mit SIEM-Funktionalität

Kai-Oliver Detken[1] · Carsten Kleiner[2]
Marius Rohde[2] · Marion Steiner[3]

[1]DECOIT GmbH
detken@decoit.de

[2]Hochschule Hannover
{carsten.kleiner | marius.rohde}@hs-hannover.de

[3]IT-Security@Work GmbH
marion.steiner@isw-online.de

Zusammenfassung

Network Access Control (NAC) Systeme dienen in erster Linie zur Absicherung von Unternehmensnetzwerken. Immer öfter werden NAC-Systeme auch im Zusammenspiel mit Anti-Viren-, Intrusion Detection- (IDS) sowie Intrusion Prevention Systemen (IPS) in die IT-Infrastruktur der Unternehmen integriert. NAC-Systeme haben dabei die Aufgabe, die Benutzer- und System-Authentifizierung zu kontrollieren. Allerdings fehlt diesen Lösungen der Gesamtüberblick über die IT-Sicherheit im Unternehmen. So werden keine Statistiken über Attacken erfasst, keine kritischen Events generiert, Compliance-Anforderungen überprüft oder ein Risikomanagement durchgeführt. Dies ist die Domäne von sog. Security Information and Event Management (SIEM) Systemen. Solche Lösungen sind jedoch relativ kostspielig, personalintensiv und komplex in der Handhabung und stellen letztendlich ein weiteres Sicherheitssystem dar, welches zusätzlich verwaltet werden muss. Daher zielt das CLEARER-Projekt[1] darauf ab, bestehende NAC-Systeme um SIEM-Funktionalitäten zu erweitern. Außerdem setzt CLEARER auf den alleinigen Einsatz von Open-Source-Software, um die Lizenzkosten des Systems so gering wie möglich zu halten. Damit wird diese Lösung auch für kleine und mittelständische Unternehmen (KMU) interessant. Des Weiteren wird ein besonderes Augenmerk auf die Nachweisbarkeit der Sicherheitsanalysen gelegt, wie sie bspw. zur Erfüllung von Compliance-Anforderungen benötigt wird.

1 Einleitung

Netzwerkinfrastrukturen ermöglichen diverse Zugangsmöglichkeiten, um unterschiedlichste Endgeräte mit einem Unternehmensnetz verbinden zu können. Zu diesen Endgeräten zählen auch immer mehr mobile Endgeräte, die mit integriert werden sollen. Die Thematik Bring Your Own Device (BYOD) beinhaltet dabei sogar den Einsatz privater Endgeräte für Unternehmensanwendungen und damit auch im Unternehmensnetzwerk. Die willkürliche Nutzung von

[1] http://www.clearer-project.de

P. Schartner · A. Baumann (Hrsg.) · D•A•CH Security 2017 · syssec (2017) 431-443.

Cloud-Diensten (z.B. Dropbox) und Videoanwendungen (z.B. Skype) öffnen weitere Einfallstore, die ein IT-Administrator im Rahmen der Gesamtsicherheit absichern sollte.

Aus diesem Grund wurde der NAC-Ansatz entwickelt. Der Zugang zu einem Unternehmensnetz kann mit NAC dediziert überprüft und ggf. verweigert werden. Das heißt, die wichtigste Aufgabe von NAC ist es, fremde Systeme zu erkennen und nach den Vorgaben der Unternehmensrichtlinien in das Netzwerk zu integrieren oder abzuweisen. Dabei werden Endgeräte während der Phase der Authentifizierung (Authentication) auf Richtlinienkonformität (Policy Compliance) geprüft, um zu gewährleisten, dass nur bekannte und, bezogen auf den Software-Stand, ausreichend aktuelle Endgeräte zugelassen werden. Dies erhöht den Sicherheitsgrad eines Unternehmens enorm.

SIEM-Systeme hingegen sind IT-Sicherheitskomponenten, die Ereignisse in der IT-Infrastruktur erfassen, bewerten und verwerten, um zu alarmieren, sobald ein unerwünschtes Verhalten erkannt wird. Ein solches System kann zusätzlich regelmäßige Reports und Handlungsempfehlungen generieren, um die IT Compliance zu überwachen. IT Compliance stellt dabei neben der Einhaltung regulatorischer Anforderungen insbesondere die Einhaltung der im Unternehmen definierten Regelungen dar. Das Ziel eines SIEM-Systems ist es daher, Angriffe oder Verstöße zu erkennen, schnell darauf zu reagieren sowie umfassende Analysen mit Hilfe des Datenbestandes auch nachträglich durchführen zu können. Konfigurations- und Rechteänderungen werden ebenfalls beobachtet und ggf. gemeldet. Daher verfolgen NAC und SIEM-Systeme komplementäre Ansätze.

Im Projekt CLEARER wird daran geforscht, wie Funktionen von SIEM- und NAC- Systemen miteinander kombiniert werden können, um Sicherheitsvorfälle in Unternehmensnetzen besser zu erkennen und die Einhaltung von IT-Compliance nachweisen, bzw. nachvollziehen zu können. Ein Ziel dabei ist es, einen Zusammenschluss der Sicherheitskomponenten zu erreichen und den hohen Konfigurationsaufwand eines weiteren alleinstehenden Systems zu vermeiden. Dabei steht die ganzheitliche Integration mit zentraler Konfigurationsschnittstelle und Datenhaltung im Vordergrund.

Ein weiteres Ziel dieses Vorhabens ist es, eine Lösung zu entwickeln, mit der die Qualität der Meldungen verbessert und der Aufwand zur Auswertung minimiert werden kann. Dies kann bedeuten, dass die Menge an fälschlich gemeldeten Vorfällen („False Positives“) durch eine bessere Analyse verringert wird. Allerdings können Falschmeldungen nie ganz ausgeschlossen werden. Auch die Anzahl der korrekten Meldungen („True Positives“) ist häufig noch zu hoch, um alle wirklich relevanten (d.h. manuelle Maßnahmen erfordernden) Meldungen detailliert zu betrachten. Um die vorhandenen, limitierten Ressourcen zielgerichtet für die wichtigsten Situationen einzusetzen, wird in diesem Projekt ein Ansatz zur Priorisierung der Vorfallmeldungen, z.B. aufgrund von beteiligten Elementen (Geräten, Nutzern, Diensten etc.) und deren Verhalten, umgesetzt. Dieser soll anpassbare Faktoren, wie z.B. Standort der betroffenen Komponenten, Kritikalität des Prozesses oder Klassifikation des erkannten Vorfalls berücksichtigen und so den Fokus auf besonders kritische Situationen der jeweiligen Umgebung legen.

Für regelbasierte Systeme ist bei der Betrachtung des Gesamtaufwands allerdings ebenfalls der Einrichtungs- und Wartungsaufwand zu bedenken. Feste Regeln können weder der kontinuierlichen Veränderung der IT-Infrastruktur, noch der Anpassung von Sicherheitsrichtlinien, als auch der stetig wachsenden Menge an Angriffsvektoren ausreichend Rechnung tragen. Nur manuell erzeugte Regeln zu nutzen, widerspricht andererseits den Zielen zur Einfachheit der Nutzung. Vielmehr wird eine Lösung benötigt, die selbstlernend Muster und Prioritäten zur Erkennung von Vorfällen anpasst, bzw. dem Nutzer in Zweifelsfällen die relevanten Informationen

zur Festlegung der korrekten Einstufung an die Hand gibt. Die Ergebnisse können dann wiederum in das Regelwerk zurückfließen, um so beim erneuten Auftreten des Vorfalles diesen besser priorisieren oder bearbeiten zu können.

Abschließend muss die zu entwickelnde Lösung auch die Behandlung von Vorfällen umsetzen. Die Unterstützung des Nutzers soll also nicht bei der Datensammlung und -verarbeitung enden, sondern auch die Reaktion auf gefundene Verletzungen mit einbeziehen. Hierzu müssen die bereitgestellten Informationen und Bewertungen sinnvoll strukturiert, aufgearbeitet und präsentiert werden. In klaren Situationen kann bei Entscheidungen sogar ganz auf die Interaktion des Nutzers verzichtet werden und eine automatische Reaktion erfolgen. In unklaren Situationen soll der Entscheidungsprozess beim Nutzer durch Hinweise und Vorschläge unterstützt werden. In jedem Fall erfolgt eine Protokollierung der Bearbeitung. Durch den beschleunigten Prozess können schwerwiegende Konsequenzen vermieden werden und es wird im Allgemeinen die Effizienz des Gesamtsystems gesteigert. Diese Möglichkeiten werden daher auch in diesem Projekt als Anforderung berücksichtigt.

2 Kommerzielle SIEM-Lösungen

Die Gartner 2016 SIEM Magic Quadrant Studie [KaRB16] gibt einen Überblick über verfügbare kommerzielle SIEM-Lösungen. Die Marktanalyse der Studie zeigt, dass der Markt von einigen wenigen großen Herstellern dominiert wird und selbst dort Defizite zur Anbindung von Fremdsystemen, Konfigurierbarkeit, Skalierbarkeit oder in der Erkennung und Hilfestellung von Sicherheitsvorfällen liegen. Zusätzlich fokussieren diese Anbieter vorrangig große Unternehmen, so dass die Zahl der Kandidaten, die für kleinere Unternehmen überhaupt erschwinglich sind, gering ist.

Zu diesen wenigen Kandidaten gehören unter anderem das *Open Source Security Information Management (OSSIM)* des Herstellers *AlienVault*[2] und das Open-Source-basierte SIEM-System der *rt-solutions.de GmbH* [MaSc16]. Eine Installation von OSSIM ist allerdings nur als Single-Server möglich, wodurch der Einsatz des Systems nicht mehr skalierbar ist. Sollten diese Kapazitäten einmal nicht mehr ausreichen, wird somit ein kompletter Umstieg auf ein skalierbares und dadurch zumeist teureres Produkt notwendig. Außerdem unterstützt OSSIM kein Logmanagement, wodurch wichtige Datenquellen nicht angebunden werden können. Hinzu kommt, dass OSSIM inzwischen nicht mehr als reine Open-Source-Lösung erhältlich ist, sondern rein proprietär vom Hersteller weiterentwickelt wird. Daher könnte man dieses System auch nicht als Basis für eine Weiterentwicklung nutzen.

Das Open-Source-basierte SIEM-System der rt-solutions.de GmbH bietet ein umfassendes SIEM-System mit Analyse-Funktionalität, das dem System in CLEARER relativ ähnlich ist. CLEARER bietet jedoch eine garantierte Verarbeitung der Ereignisse durch das Apache-Storm-Framework. Die in dem Open-Source-SIEM eingesetzte frei erhältliche Esper-Lösung kann dies nicht gewährleisten. Insgesamt fokussiert CLEARER den Aspekt der Compliance stärker. Dies zeigt die Anbindung an ein NAC-System und die Möglichkeit der automatisierten Umsetzung von Richtlinien und direkten Reaktion auf Verstöße. Außerdem bietet keines der uns bekannten Open-Source-basierten SIEM-Systeme eine Anbindung an das Protokoll Interface for a Metadata Access Point (IFMAP).

[2] https://www.alienvault.com

3 Anforderungen an die integrierte Datenbasis

Zur Erfüllung der oben beschriebenen Anforderungen müssen Daten unterschiedlicher Quellen zusammengeführt werden. Dies geschieht in einer zentralen Ereignisdatenbank im Zusammenspiel mit dem bereits existierenden IF-MAP Server *irond* und dem *VisITMeta-Dataservice*[3]. Während irond und der VisITMeta-Dataservice für Zustandsinformationen und Netzwerkarchitektur-Informationen verwendet werden, sollen alle erzeugten Ereignisse in einer neuen Ereignisdatenbank gespeichert werden. Die Ereignisdatenbank wird in Verbindung mit dem VisIT-Meta-Dataservice das Langzeitgedächtnis des Gesamtsystems darstellen. Es dient der Unterstützung der Datenkorrelation, -bewertung und Compliance-Einhaltung. Außerdem ist es zur Durchführung eines Auditing erforderlich, da Events in Verbindung mit den Zustandsinformationen die Basis der Compliance-Bewertung bilden. Um eine geeignete Technologie für die Ereignisdatenbank des Projekts auswählen zu können, wurden zunächst Anforderungen definiert und die folgenden Auswahlkriterien ermittelt:

a. **Sicherheit:** Authentifizierung unterschiedlicher Benutzer und Zugriffskontrolle auf Dateien und Ordner über Dateiberechtigungen. Eine Verschlüsselung der Kommunikation unter den Datenbank-Komponenten des CLEARER-Systems ist nicht vorgesehen. Hier wird auf eine ausreichende Abschottung für dedizierte Netzwerk-Segmente gesetzt.
b. **Auditfähigkeit:** Verifizierung der Compliance-Einhaltung, um im Falle eines Angriffs mögliche Compliance-Verletzungen zu erkennen oder ausschließen zu können. Ein umfassendes Logging der Aktivitäten ist damit unerlässlich.
c. **Benutzerrechte:** Eine Rechteverwaltung ist notwendig, damit jeder Benutzer nur die Einstellungen sehen kann, für die er Rechte besitzt.
d. **Schreib-, Lese- und Update-Performance:** Ein hohes Aufkommen an Ereignisdaten wird vorhergesehen. Um diese nahezu in Echtzeit auswerten zu können und alle Ereignisse schnell für Auswertungen verfügbar zu haben, muss die Speichergeschwindigkeit entsprechend hoch sein. Die Performance der Lese- und Update-Geschwindigkeit ist hingegen nicht so kritisch zu bewerten, wie die Schreibgeschwindigkeit.
e. **Flexible Dateninhalte und Schemata:** Da jeder Event-Typ unabhängig von anderen Typen abgelegt wird, sind flexible Dateninhalte nicht notwendig. Dadurch muss das DBMS bei der Selektion eines Event-Typs nicht die gesamten Daten durchsuchen. Durch die Vorgabe des Event-Formats über die Clients, sind die Schemata statisch vorgegeben. Flexible Schemata sind daher auch nicht nötig.
f. **Horizontale Erweiterbarkeit bzw. Skalierbarkeit:** Da die Zahl der datenerzeugenden Sensoren nicht vorhersehbar ist und vermutlich in Zukunft deutlich zunehmen wird, ist durch horizontale Erweiterbarkeit ein hohes Maß an Skalierbarkeit gegeben. Dies ist aufgrund der anzunehmenden Datenmenge und potentiellen Erweiterung des Systems auch notwendig.
g. **Mapping der Eingabe auf gespeicherte Daten:** Das Impedance-Mismatch-Problem soll möglichst klein gehalten, bzw. umgangen werden. Die Speicherung eines z.B. Java-Objekts soll ohne Programmieraufwand in der Datenbank erfolgen können.
h. **Konsistenz:** Die Konsistenz nach dem CAP-Theorem (Consistency, Availability, Partition Tolerance) [GiLy02] ist eine Anforderung mit hoher Kritikalität. Es reicht aus, wenn

[3] http://trust.f4.hs-hannover.de/index.html

gewährleistet wird, dass die Daten im verteilten Speicher nach einer garantierten Zeit konsistent sind.

i. **Verfügbarkeit:** Die Verfügbarkeit nach dem CAP-Theorem wurde in zwei Unterpunkte aufgeteilt: Schreib- und Leseanfragen. Schreibanfragen müssen schnell und garantiert durchgeführt werden. Leseanfragen können hingegen auch länger dauern.

j. **Partitionstoleranzen:** Mit der Partitionstoleranz des CAP-Theorems ist die Ausfallsicherheit des Gesamtsystems gemeint. Um die Compliance der IT-Infrastruktur möglichst für die gesamte Zeit nachweisen zu können, muss die Speicherung von Daten zu jeder Zeit möglich sein. Eine hohe Partitionstoleranz wird somit angestrebt.

Die Anforderungsdefinition basiert auf den Erfahrungen der Projektpartner, auf den Bestimmungen zur *Compliance-Konformität* und auf der Annahme des zu erwartenden Datenaufkommens. Wie im folgenden Abschnitt gezeigt wird, erfüllt das System *Apache Cassandra*[4] die Anforderungen gut, weshalb es für das CLEARER-System ausgewählt wurde.

4 CLEARER-Architektur

Die Architektur (siehe Abbildung 1) des Projekts CLEARER besteht aus diversen Komponenten, wobei an dieser Stelle nur auf die Wichtigsten eingegangen werden kann. Die Darstellung der SIEM-Analysen wird durch zwei verschiedene Graphical User Interfaces (GUI) umgesetzt. Zum einen wird die Komponente *SIEM-GUI+*, zum anderen die *Banana-GUI* (siehe späterer Abschnitt) verwendet.

SIEM-GUI+ wird als zentrale Oberfläche für Anwender des CLEARER-Systems genutzt. Sie bietet dem Anwender dabei die Funktionalität zur Verwaltung anderer Systeme, z.B. Ticketsystem und NAC. Die GUI enthält ein Rollen- und Rechtemanagement, um Administratoren und Benutzer voneinander zu trennen. Das Rollen- und Rechtemanagement wird auch von dem in der GUI integrierten Ticketsystem genutzt, um den Zugriff auf Tickets und Queues einzuschränken. Zur weiteren Verarbeitung können Event-Rohdaten als CSV- oder XML-Format exportiert werden. Eine Compliance-Log Ansicht ermöglicht mithilfe von Filterfunktionen nach Zeitraum oder Typ die einfache Suche nach relevanten Compliance-Vorfällen.

Darüber hinaus wird eine Systemstatus-Ansicht (Dashboard) entwickelt, die zusätzliche Informationen über aktive Sensoren und andere Systemkomponenten bietet. Dazu gehören zum Beispiel auch die Laufzeit-Informationen von *RabbitMQ*[5] und *Apache Storm*[6], sofern diese von externen Systemen auslesbar sind. RabbitMQ ist, wie Abbildung 1 zeigt, die zentrale Ereignisübermittlungs-Komponente. Apache Storm dient als skalierbares Ereignis verarbeitendes Framework. Storm bietet mithilfe von Esper die Möglichkeit einfach und effizient Ereignisse zu korrelieren. Außerdem stellt Storm eine Schnittstelle zur Speicherung und Selektion von Daten in einer Cassandra Datenbank bereit. Geplant ist auch eine Übersicht über die überwachte Netzwerktopologie, soweit sie dem CLEARER-System bekannt ist. Dazu muss die SIEM-GUI+ auf die Zustandsdaten im MAP-Server zurückgreifen. Die SIEM-GUI+ erhält zu diesem Zweck eine IF-MAP-Schnittstelle [TCG12].

[4] http://cassandra.apache.org

[5] https://www.rabbitmq.com

[6] http://storm.apache.org

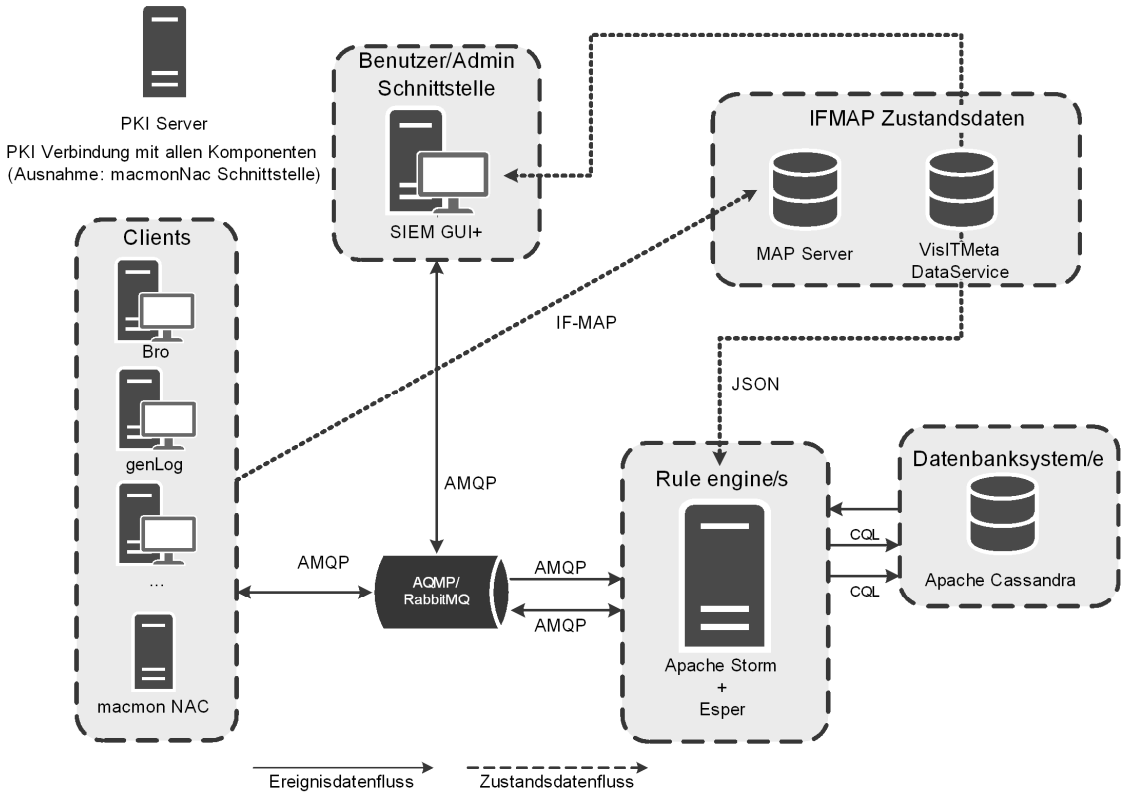

Abb. 1: CLEARER-Architektur in der Gesamtübersicht

Das zu verwendende *Ticketsystem* soll insbesondere die Schnittstelle für eine einfache Konfiguration bereitstellen. Im Idealfall erlaubt diese die gesamte Verwaltung des Ticketsystems. Damit würde die GUI des Ticketsystems ausschließlich zur erstmaligen Installation und Einrichtung benötigt und muss danach nicht mehr verwendet werden. Ziel der Entwicklung sollte es daher sein, dass sämtliche Aufgaben, die Anwender des CLEARER-Systems ausführen müssen (Ticket- und Benutzerverwaltung) über die SIEM-GUI+ durchgeführt werden können. Verschiedene Ticketsysteme (u.a. Redmine[7], RT[8]) wurden evaluiert, letztendlich wird aber eine Eigenentwicklung zum Einsatz kommen. Die Gründe für diese Entscheidung sind zum einen, dass der Overhead, den ein vollwertiges Ticketsystem mitbringt, der einfachen Konfiguration des CLEARER-Systems entgegenwirkt, und zum anderen, dass sich ein fremdes System nicht ohne einen Bruch in das CLEARER-Gesamtsystem einfügen lässt.

Die GUI des verwendeten *NAC-Systems* wird zum Verwalten des Regelwerks genutzt. In welchem Maße die GUI notwendig ist, hängt dabei vom verwendeten System und der angebotenen Schnittstelle ab. Das *macmon secure NAC*[9] erlaubt nur relativ rudimentäre Aufgaben über die angebotene Schnittstelle. Daher wird die macmon GUI für einige Aufgaben keine Verwendung finden. Für die Einbindung von *PacketFence*[10] wird noch geprüft, ob eine Integration der CLI-

[7] http://www.redmine.org

[8] https://bestpractical.com/request-tracker/

[9] https://www.macmon.eu

[10] https://packetfence.org

Schnittstelle möglich ist und in welchem Maße die Aufgaben über die SIEM-GUI+ anstatt über die macmon-eigene NAC-GUI durchgeführt werden können.

Als zweite Haupt-GUI kommt die *Banana-GUI* bei CLEARER zum Einsatz, die eine aufbereitete Sicht auf die Log-Informationen des Gesamtsystems ermöglicht. Die Log-GUI unterscheidet sich von der Statusanzeige der SIEM-GUI+ durch eine historische Sicht auf die Log-Informationen. Hiermit lassen sich z.B. die Stände des Regelwerks in CLEARER zu jedem beliebigen Zeitpunkt in der Historie nachvollziehen. Diese GUI erlaubt es insbesondere auch, die korrekte Funktionsweise des Systems zu analysieren, und somit die Korrektheit der Compliance-Meldungen durch CLEARER nachzuvollziehen. Im Gegensatz zur Statusanzeige der SIEM GUI+ unterstützt die Banana-basierte Log-Ansicht eine freie Konfiguration der angezeigten Diagramme und Berichte. Diese freie Konfiguration unterstützt Audit-Prozesse maßgeblich, da sich das Berichtswesen auf die Anforderungen des Auditors anpassen lässt. Durch die Trennung der operationalen Statusanzeige der SIEM-GUI+ und der historischen Sicht der Log-GUI Banana sind die Anwendungsfälle strikt voneinander getrennt und eine einfachere Abschottung und eventuelle Pseudonymisierung von Daten in der Statusanzeige wird möglich. Hierdurch können betriebsrechtliche Vorschriften in jedem Fall erfüllt werden.

Die technische Umsetzung erfolgt über die *Datastax-Enterprise-Umgebung*[11]. Die Basis der Log-Daten bilden die gespeicherten Ereignisse der Cassandra-Datenbank [CASS17]. Um diese effizient durchsuchbar zu machen, bietet Datastax die Erweiterung *DSE Search* auf Basis von Cassandra und *Solr*[12] an. Solr ist das Konkurrenzprodukt zu ElasticSearch[13]. DSE Search ermöglicht so die Integration von Solr in Cassandra ohne die Verwendung eines zusätzlichen Solr-Clusters zur Datenhaltung. Dieser Aufbau reduziert die Datenredundanzen und ebenfalls den Angriffsvektor des Gesamtsystems, da dieses mit weniger Komponenten auskommt. Auf die Solr-Komponente wird dann die Banana-GUI aufgesetzt, um eine bedienungsfreundliche Oberfläche zur Erstellung von Diagrammen und Berichten zu erhalten.

Da die zentrale Datenbasis alle Ereignisse des Systems beinhaltet, können Berichte in allen Detailierungsgraden erstellt werden. Ein Endereignis-bezogenes Drill-Down auf die Ursprungsereignisse wird somit möglich. Unterschiedliche Berichte könnten beispielsweise beinhalten, wann welche Komponenten inaktiv waren oder in welchen Zeiten der Compliance-Status verletzt wurde. Auch die Anzahl der Warnmeldungen des Systems in einem bestimmten Zeitraum ließe sich erfassen.

Als zentraler Dreh- und Angelpunkt für die Kommunikation zwischen den verschiedenen Systemkomponenten werden im Wesentlichen zwei Systeme verwendet: MAP-Server und RabbitMQ Message Broker. Für die Zustandsinformationen dient der *MAP-Server* als Datenbank. Auf diesen wird via IF-MAP schreibend zugegriffen, um den Systemzustand zu aktualisieren. Für den lesenden Zugriff wird dem MAP-Server der Dataservice von *VisITMeta*[14] nachgeschaltet. Dieser erlaubt den Zugriff auf die Historie des MAP-Graphen und ermöglicht so einen Blick auf vergangene Zustände. Die weitere interne Kommunikation wird über den RabbitMQ-Message-Broker [RABB17] und das Protokoll *Advanced Message Queuing Protocol (AMQP)*[15]

[11] https://www.datastax.com

[12] http://lucene.apache.org/solr

[13] https://www.elastic.co/products/elasticsearch

[14] http://trust.f4.hs-hannover.de/projects/visitmeta.html

[15] http://www.amqp.org

durchgeführt. Insbesondere Ereignisdaten, Korrelationsergebnisse, etc. werden darüber übertragen. Dazu dienen je nach Anwendungsfall verschiedene AMQP-Exchanges und Message-Queues, wodurch eine Nachricht auf einfache Art und Weise an viele interessierte Empfänger geschickt werden kann. Sie müssen sich lediglich beim RabbitMQ die entsprechenden Message-Queues abonnieren.

Um Ereignisdaten zwischen den IF-MAP-Clients und der Korrelationsengine zu übertragen, wird ein gemeinsames Datenaustauschformat benötigt. Dabei wird zwischen dem Format für den Inhalt und dem Format für die Serialisierung unterschieden. Das *Inhaltsformat* beschreibt welche Daten in welcher Struktur übertragen werden; beispielsweise könnten Ereignisdaten die zwei Felder „Zeitstempel des Ereignisses“ und „Informationen über den erkannten Vorfall“ enthalten. Die Informationen über den Vorfall selbst könnten wiederum mehrere Felder wie Typ, ID und eine Bewertung enthalten. Das *Serialisierungsformat* gibt an, wie diese Inhaltsdaten für die Übertragung kodiert werden. Hier könnte man beispielsweise auf textbasierte Formate, wie XML oder JSON, zurückgreifen. Da das CLEARER-System in der Lage sein muss, große Mengen an Ereignisdaten zu verarbeiten, wird auf ein binäres Format zurückgegriffen, das im Gegensatz zu einem textbasierten Format deutlich kompakter ist.

Das Inhaltsformat lehnt sich an das Intrusion Detection Message Exchange Format (IDMEF) aus RFC-4765 [DeCF07] an. Da IDMEF nur für Intrusion Detection Systeme entwickelt wurde, können nicht alle in CLEARER gesammelten Ereignisdaten eins zu eins in IDMEF integriert werden. Eine Erweiterung des Ereignisformats durch das Additional-Data-Feld wäre somit nötig. Der Informationsaustausch über die Definition der hinzugefügten Daten müsste also auch bei Verwendung des IDMEF-Standards im Vorfeld erfolgen. XML hat außerdem den Nachteil, dass es eine große Menge an Verwaltungsdaten benötigt. Um den zu übertragenden Dateninhalt möglichst klein zu halten, wird daher in CLEARER kein XML verwendet. Auch die zu übertragenden Datenmengen werden verringert, indem auf vorgegebene Felder, die innerhalb des CLEARER-Systems nicht benötigt werden, verzichtet wird.

Um ein geeignetes Serialisierungsformat zu finden, wurden primär die beiden Formate *Concise Binary Object Representation (CBOR)* [BoHo13] und *Protocol Buffers (protobuf)*[16] untersucht. Weitere Formate wie ASN.1 (mit Packed Endcoding Rules), MessagePack oder Apache Thrift wurden nicht näher betrachtet. Auf den ersten Blick hat protobuf durch die Code-Generierung gegenüber CBOR den Vorteil, dass keine zusätzlichen Klassen geschrieben werden müssen, die die Datenstruktur für die weitere Verwendung in der Korrelationsengine repräsentieren. Wenn allerdings z.B. IPv4-Adressen als Zahl und nicht als Zeichenkette übertragen werden, muss die Adresse u.U. für die Korrelation wieder in ein Zeichenkettenformat überführt werden. In diesem Fall müsste die von protobuf generierte Klasse doch in eine weitere Wrapper-Klasse verpackt werden, die eine passende Konvertierung vornehmen könnte. Da sowohl die IF-MAP-Clients, als auch die Korrelationsengine in Java geschrieben werden, wird keine Programmiersprachen-übergreifende Portierung benötigt und die nötigen Klassen (für die Verwendung mit CBOR) können auch direkt in Java modelliert werden. Basierend auf diesem Vergleich wurde CBOR als Serialisierungsformat gewählt, da es sich voraussichtlich einfacher in einer reinen Java-Umgebung anbinden lässt. Zusätzlich ist der Größenunterschied binärer Repräsentation, gegenüber einem reinen textbasierten Format, zu vernachlässigen.

Die *Ereigniskorrelation* ist einer der zentralen Bausteine des CLEARER-Systems. Diese arbeitet mit den Rohdaten der verschiedenen Sensoren, erkennt in diesen mit Hilfe eines Regelwerks

[16] https://developers.google.com/protocol-buffers

bestimmte Muster und erzeugt daraus vordefinierte, abstrahierte Informationen. Diese werden in einem weiteren Schritt bewertet und von der Policy-Engine genutzt, um Verletzungen von Compliance-Regeln zu erkennen.

Um die Ereignisdaten vorab filtern zu können und der Korrelation zuzuführen, wird das Framework *Apache Storm* [APST17] verwendet. Storm erlaubt das leistungsstarke Steuern von Datenströmen in einem Rechner-Cluster. Dadurch ist es sehr einfach durch Hinzufügen neuer Knoten zum Cluster horizontal zu skalieren. Apache Storm verwendet die Konzepte *Spouts* (Datenquelle) und *Bolt* (Datenverarbeitung), um eine sogenannte Topologie zu definieren. Innerhalb dieser Topologie können beliebig viele Spouts und Bolts miteinander verknüpft werden, um die Verarbeitung des Datenstroms zu realisieren. Die einzige logische Begrenzung für den Aufbau der Topologie ist, dass es keine zyklischen Verbindungen zwischen Bolts geben darf. Eine iterative Verarbeitung kann jedoch durch die Rückführung von Ereignissen in die Message-Queue und erneutes Abrufen durch das Storm-Cluster erfolgen.

Die in CLEARER zu entwickelnde Topologie wird über die *AMQP Spouts* direkt von *RabbitMQ* mit Daten versorgt. Diese abonnieren bestimmte Queues und werden somit vom Broker mit Ereignisdaten beliefert. Die Spouts erzeugen aus den erhaltenen Daten Storm-Tupel und stellen sie der Topologie zur Verarbeitung bereit. Die Korrelation wird in den Bolts realisiert. Dabei werden die erhaltenen Ereignis-Tupel von den Bolts in ein Datenformat konvertiert, mit dem die Korrelation arbeiten kann. Das Korrelationsframework korreliert diese Ereignisse mit dem zuvor konfigurierten Regelwerk.

Neben den Korrelations-Bolts werden weiterhin Bolts für die Ausgabe der Korrelationsergebnisse an den RabbitMQ-Broker und für die Vorabfilterung der Ereignisse benötigt. Letztere sollten zum Beispiel solche Ereignisse aus dem Strom herausfiltern, die nicht in den zeitlichen Ablauf passen. Diese verspäteten Ereignisse müssen gesondert behandelt werden, da für sie auf historische Ereignisinformationen aus der Ereignisdatenbank zugegriffen werden muss.

Für die Ereigniskorrelation wurden die Werkzeuge Simple Event Correlator (SEC)[17], Esper [EcBr09] und Drools Fusion[18] verglichen. Von diesen hat sich *Esper* als am besten geeignet für die Zwecke des CLEARER-Systems herausgestellt. SEC ist in Perl geschrieben und verwendet hauptsächlich Regular Expression (RegEx) basierte Regeln. Um komplexere Regeln, z.B. auch mehrstufige RegEx-Analysen, zu definieren muss daher Perl-Code geschrieben und in die Regeldefinition eingebettet werden. Dies ist zum einen sehr unübersichtlich und somit schwer zu pflegen und zum anderen führt es eine zusätzliche Regelsprache ein, die beherrscht werden muss.

Esper und Drools Fusion unterscheiden sich von der Funktionsweise stark, ähneln sich aber in der Art wie Regeln definiert und in andere Systeme integriert werden können. Drools Fusion verwendet dabei eine Java-ähnliche aber sehr simple und abstrahierte Domain-Specific Language (DSL) zur Regeldefinition, während Esper auf eine SQL-ähnliche Abfragesprache setzt. Beide Varianten sind sehr mächtig und bieten ausreichend Möglichkeiten, um auch komplexe Regelwerke zu definieren. Drools Fusion ist Bestandteil der Drools Suite, einem Business Rules Management System. Innerhalb dieses Systems ist es in der Drools Engine integriert und lässt sich auch nicht losgelöst davon einsetzen. Diese Engine enthält zusätzlich noch weitere Leistungsmerkmale, die für das CLEARER-System nicht relevant sind. Esper hingegen ist nur als

[17] https://simple-evcorr.github.io

[18] http://drools.jboss.org/drools-fusion.html

Korrelationsengine konzipiert. Aus diesem Grund wird in CLEARER Esper für die Korrelation von Ereignissen eingesetzt.

Außerdem wird Esper als *Bewertungsengine* verwendet. Die im Vergleich zu Apache-Storm [APST17] umfangreicheren Korrelationsmöglichkeiten von Esper erleichtern dabei die Bewertung. Die verlustfreie Übertragung ist mit der Esper-Standardversion allerdings nicht garantiert. Durch die Erweiterung auf die kostenpflichtige Esper HA Version ist die verlustfreie Übertragung aber nicht mehr problematisch. Durch die universelle Ausrichtung und Abfragesprache von Esper ist die Korrelation unterschiedlicher Ereignisse gegeben. Außerdem bietet Esper Zeitfenster und Operatoren zur Korrelation von Ereignissen aufgrund der zeitlichen Reihenfolge an.

Für die Datenbankanbindung an Cassandra sind keine vorgefertigten Software-Plug-Ins für Esper vorhanden. Die Anbindung kann aber durch den Ereignisrückfluss an einen Cassandra Bolt erfolgen. Die Anbindung ist somit in Java-Klassen einfach zu realisieren. Durch die menschenlesbare Abfragesprache von Esper ist die eigenständige Konfiguration der Regeln durch geschulte Kunden möglich. Dadurch wird die Funktionserweiterung, Anpassung und Wartung für den Kunden transparenter und ein besseres Verständnis über das System wird erlangt.

Die einfache Skalierung über die Hinzunahme weiterer Esper-Agenten auf anderen Maschinen ermöglicht die Verarbeitung großer Datenmengen. Die Bewertungsengine benötigt Zugriff auf den korrelierten Ereignisstrom der Esper-Korrelation. Außerdem ist die Anbindung an IF-MAP und Cassandra nötig. Ereignisse müssen in der Cassandra-Datenbank gelesen und geschrieben werden. Ein Update von Ereignissen wird aus Gründen der Compliance-Nachvollziehbarkeit nicht erfolgen, da Veränderbarkeit auch immer Manipulierbarkeit bedeutet. Stattdessen können veränderte Events als neue, zusätzliche Events angelegt werden. IF-MAP-Daten müssen nur gelesen werden. Ob das Anlegen neuer IF-MAP-Daten nötig ist, ergibt sich im weiteren Projektverlauf. Die Weiterleitung der bewerteten Ereignisse an die Policy-Engine muss aber gewährleistet bleiben.

Die technische Umsetzung der *Policy-Engine* wird auch über die CEP-Engine Esper erfolgen. Dabei gestalten sich die Anforderungen im Vergleich zu der Bewertungsengine nicht wesentlich anders. Die Policy-Engine nimmt die bewerteten Ereignisse der Bewertungsengine entgegen. Wie die Bewertungsengine benötigt auch die Policy-Engine den Zugriff auf IF-MAP und Cassandra. Die Ereignisse müssen geschrieben und gelesen werden. IF-MAP-Daten werden in der ersten Überlegung nur gelesen. Der größte technische Unterschied zur Bewertungsengine liegt in der Reaktion auf Ereignisse, auf die eine Aktion erfolgen muss. Außerdem kann die Anzeige des Compliance-Status über eine direkte Anbindung oder den Umweg über die Cassandra-Datenbank realisiert werden.

Unter der Vorgabe, dass das System selber Regeln lernen können soll, muss es auch in der Lage sein, Anomalien bzw. eine Veränderung des Normalzustands im Netzwerk zu erkennen. Dazu muss ein Normalzustand der betrachteten Daten bekannt (fest definiert bzw. vorab trainiert) oder zumindest erfassbar sein. Alles, was von diesem Normalzustand abweicht, ist grundsätzlich als eine Anomalie zu verstehen. Damit nicht jede einzelne Abweichung als Vorfall gemeldet wird, muss die Korrelation oder eine *Zeitreihenanalyse* dafür Sorge tragen, dass die einzelnen Anomalien als aussagekräftige Ereignisse zusammengefasst (oder als Rauschen verworfen) werden. Aus diesem Grund wird eine Zeitreihenanalyse auf Basis von Netzwerkverkehrsdaten ebenfalls innerhalb des Projektes entwickelt und implementiert werden.

5 Schnittstelle zu NAC-Systemen

Das CLEARER-System benötigt eine *Schnittstelle* zu NAC-Systemen, die um SIEM-Funktionalität ergänzt werden sollen. Dies ist notwendig, weil zumindest Endgeräte über NAC gesperrt oder entsperrt werden müssen. Darüber hinaus soll das Regelwerk des NAC-Systems eingesehen und modifiziert werden können sowie die NAC-Konfiguration gesteuert werden. Neben diesen, für die Verwaltung des NAC notwendigen, Schnittstellen benötigt das CLEARER-System Zugriff auf die vom NAC erhobenen Infrastrukturdaten. Diese müssen Informationen über bekannte Systeme im Netzwerk, deren IP- und MAC-Adressen sowie authentifizierte Benutzerkonten enthalten. Mit diesen Daten kann der IF-MAP-Zustandsspeicher gefüllt werden, um Korrelationen und Bewertung von Ereignissen mit den Infrastrukturdaten zu bereichern. Im Falle von macmon secure bietet sich das Tool *macutil* als Schnittstelle an. Dieses kann entweder per Command Line Interface (CLI) oder HTTPS angesprochen werden. Für die Verwendung im CLEARER-System bietet sich nur die HTTPS-Lösung an, da für einen CLI-Zugriff ein Agent auf dem NAC-System notwendig wäre.

Das Tool macutil erlaubt einige Aktionen von außerhalb des NAC auszulösen, darunter fällt zum Beispiel das Sperren oder Entsperren von MAC-Adressen. Da diese Funktion für das CLEARER-System wichtig ist, muss sie in die Gesamtarchitektur integriert werden. Weiterhin bietet macutil die Möglichkeit des Auflösens von IP- zu MAC-Adresse und umgekehrt. Da die meisten Sensoren nur auf IP-Ebene arbeiten, ist diese Funktion ebenfalls wichtig, denn das macmon NAC benötigt MAC-Adressen als Bezeichner für Endgeräte. Abbildung 2 zeigt das erste Konzept zur Integration der macutil-Schnittstelle in das CLEARER-System. Um auch Schnittstellen anderer NAC-Systeme ansprechen zu können, wird diese Funktion in einem *NAC-Aktuator* gekapselt. Dieser verarbeitet die eingehenden Anfragen und wandelt sie in ein für die jeweilige Schnittstelle verständliches Format um. Im Falle von macutil ist dies ein HTTPS-Request, der zudem „HTTP Basic Auth“ für die Authentifizierung verwendet.

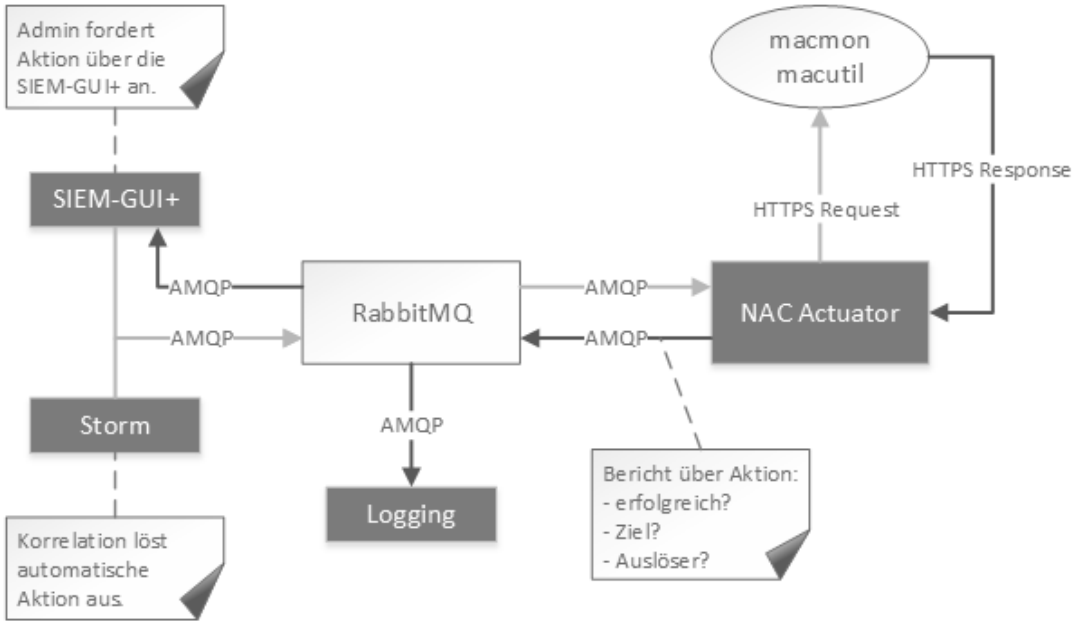

Abb. 2: Erstes Konzept für die Integration von macutil

Anfragen zum Sperren oder Entsperren eines Endgeräts können von zwei Aktoren kommen. Zum einen ist dies die SIEM-GUI+, über die ein Administrator zur Bearbeitung von Vorfällen

Geräte sperren oder entsperren kann. Zum anderen kann, sofern dies gewünscht ist, die Korrelation automatisiert ein *Enforcement* auslösen. Dann kommt die entsprechende Anfrage direkt über einen Bolt aus der Storm-Topologie.

In beiden Fällen wird die Anfrage über RabbitMQ und das AMQP-Protokoll an den *NAC-Aktuator* gesendet. Dieser führt die Aktion über die NAC-Schnittstelle aus und verarbeitet die Rückgabe. In allen Fällen muss ein Eintrag in das Logging-System geschrieben werden, in dem die ausgeführte Aktion, das Ergebnis der Aktion, das Ziel und die auslösende Komponente vermerkt sind. Zusätzlich muss, sofern die Aktion über die SIEM-GUI+ ausgelöst wurde, auch hier eine entsprechende Antwort zur Anzeige in der GUI gesendet werden. Im Falle eines automatischen *Enforcement* ist diese Antwort nicht notwendig, da sich alle benötigten Informationen für eine spätere Fehlersuche im Logging-System befinden. In jedem Fall müssen die hierdurch ausgelösten Events und Zustände ebenfalls wieder direkt Eingang in die CLEARER-Verarbeitung finden.

Wie in Abbildung 2 dargestellt kapselt der *NAC-Aktuator* den Zugriff auf das eigentlich NAC-System, so dass beim Einsatz anderer Produkte lediglich die Schnittstelle zwischen Aktuator und NAC ausgetauscht werden muss.

6 Ausblick

Das IT-Sicherheitsbewusstsein von Unternehmen wächst durch stetig steigende Zahlen von Cyber-Attacken kontinuierlich an, ebenso steigt der Zwang zur Umsetzung von Maßnahmen durch regulatorische oder Kundenanforderungen. Dadurch kommen immer mehr NAC-Lösungen zum Einsatz. Diese decken allerdings nicht die Analyse des IT-Sicherheitsgrades in Firmen ab, so dass diese meistens nicht über das entsprechende Risiko informiert sind. Ebenso besteht eine Lücke bei der Nachweisbarkeit der Erfüllung von Compliance- oder Sicherheitsanforderungen im Rahmen von Audits. Das CLEARER-Projekt soll diese Lücke schließen, indem SIEM-Funktionalität mit NAC-Systemen integriert bereitgestellt wird. Die Entwicklung wird dabei NAC-Unabhängig vorgenommen, so dass unterschiedliche NAC-Hersteller angebunden werden können. In diesem Beitrag wurde die Gesamtarchitektur des CLEARER-Systems vorgestellt, dass diese Integration leisten wird. Ferner wurden für die einzelnen Komponenten des Systems in Frage kommende Technologien evaluiert und Auswahlen getroffen. Aktuell befindet sich das Projekt in der Realisierungsphase, in der die konkrete Integration der genannten Technologien erfolgt. Eine Evaluierung der Ergebnisse ist im Anschluss geplant. Durch den konsequenten Einsatz von Open Source Software können zudem Lizenzkosten eingespart und eine beliebige Schnittstellenanpassung umgesetzt werden. Das Projektergebnis wird daher auch für kleinere Unternehmen interessant werden, die bisher vor den Kosten des Einsatzes kommerzieller IT-Sicherheitstechnologie zurückschreckten.

Danksagung

Das CLEARER-Projekt (www.clearer-project.de) ist ein gefördertes BMWi-Projekt mit einer Laufzeit von zwei Jahren, welches im Mai 2016 seine Arbeiten aufnahm und voraussichtlich im April 2018 enden wird. An dem Projekt sind die Firmen DECOIT GmbH (Projektleitung), IT-Security@Work (ISW) und macmon secure GmbH sowie die deutsche Forschungseinrichtung Hochschule Hannover beteiligt. Als assoziierte Partner nehmen der Hersteller Achtwerk GmbH und der SIEM-Anbieter rt-solutions.de GmbH an dem Projekt teil. Dank gilt den Partnern des Projektes, die durch ihre Beiträge und Arbeiten die erfolgreiche Projektarbeit erst ermöglicht haben.

Literatur

[APST17] Apache Storm: http://storm.apache.org

[BoHo13] C. Bormann, P. Hoffman: *Concise Binary Object Representation (CBOR)*. Internet Engineering Task Force (IETF), RFC-7049, Standards Track, IETF 2013

[CASS17] Webseite der Cassandra-Datenbank: http://cassandra.apache.org

[DeCF07] H. Debar, D. Curry, B. Feinstein: *The Intrusion Detection Message Exchange Format (IDMEF)*. Network Working Group, RFC-4765, Category: Experimental, IETF 2007

[EcBr09] M. Eckert, F. Bry: *Complex Event Processing (CEP)*. Gesellschaft für Informatik e. V., 5. Mai 2009

[GiLy02] S. Gilbert, N. Lynch: *Brewer's Conjeture and the Feasibility of Consistent, Available, Partition-Tolerant Web Services*. ACM SIGACT News, v. 33 issue 2, 2002, p. 51–59

[KaRB16] K. M. Kavanagh, O. Rochford, T. Bussa: *Magic Quadrant for Security Information and Event Management.* Gartner Group, 10. August 2016

[MaSc16] D. Mahrenholz, R. Schumann: *Open-Source-SIEM im Eigenbau*. P. Schartner, P. Lipp. „DACH Security 2016“, syssec 2016

[RABB17] RabbitMQ – Messaging that just works: https://www.rabbitmq.com

[TCG12] TCG Trusted Network Communications: *TNC IF-MAP Metadata for Network Security*. Specification Version 1.1, Revision 9, 7th of May 2012