Event-Korrelation in SIEM-Systemen auf Basis von IF-MAP

Kai-Oliver Detken[1] · Felix Heine[2] · Thomas Rix[1] · Leonard Renners[2]

[1]DECOIT GmbH, Fahrenheitstr. 9, D-28359 Bremen
detken/rix@decoit.de

[2]Hochschule Hannover, Ricklinger Stadtweg 120, D-30459 Hannover
felix.heine/leonard.renners@hs-hannover.de

Zusammenfassung

Heute sind mannigfaltige Sicherheitssysteme wie Firewalls, Virenscanner, Spamfilter und VPN-Gateways in Unternehmensnetzen im Einsatz, die aber normalerweise unabhängig voneinander arbeiten. Viele Schwachstellen können allerdings nur durch die Kombination von Daten verschiedenster Systeme erkannt werden. Dies sollen sog. Security Information and Event Management (SIEM) Lösungen leisten, die in der Lage sind, Meldungen und Warnungen einzelner Sicherheitskomponenten zusammenzuführen und auszuwerten. Die dafür notwendige Korrelation der Ereignisse gestaltet sich allerdings schwierig, da unterschiedliche Formate zum Einsatz kommen. Außerdem stellt die übersichtliche und nachvollziehbare Visualisierung von Korrelationsergebnissen weiterhin eine große Herausforderung dar. Dieser Beitrag stellt eine SIEM-Lösung des Forschungsprojekts SIMU [SIMU15] vor, die auf der IF-MAP Spezifikation der Trusted Computing Group (TCG) basiert und somit auf einer gemeinsamen Datenbasis eine Auswertung vornimmt. Durch die Korrelation der zentral in einem einheitlichen Format vorgehaltenen Daten der unterschiedlichen Überwachungskomponenten, können komplexere Angriffe erkannt werden. Die Basisinformationen, wie auch die korrelierten Ereignisse, werden im Projekt mittels mehrerer fortgeschrittener Visualisierungskomponenten dem Nutzer präsentiert und zur weiteren Bearbeitung im Rahmen der definierten Prozesse bereitgestellt.

1 Einleitung

Während heute mannigfaltige Sicherheitssysteme wie Firewalls, Virenscanner, Spamfilter und VPN-Gateways in Unternehmensnetzen im Einsatz sind, die üblicherweise unabhängig voneinander arbeiten, können viele Schwachstellen nur durch die Kombination von Daten verschiedenster Systeme erkannt werden. Dies sollen sog. „Security Information and Event Management“ (SIEM) Lösungen leisten, die in der Lage sind, Meldungen und Warnungen einzelner Sicherheitskomponenten zusammenzuführen und auszuwerten. Die Bezeichnung SIEM wurde unter anderem in [WIL07] und [SWI06] eingeführt und erläutert. Ein wesentliches Ergebnis der Analyse aus [DRS14] ist, dass ein SIEM-System sich sowohl zur nahezu Echtzeitauswertung als auch zur Langzeiterfassung von Informationen eignet. Die Hauptaufgaben bestehen dabei aus der Konsolidierung der unterschiedlichen Datenformate und -schnittstellen, einer effizienten Auswertung und Einschätzung der Situation und der geeigneten Aufbereitung und Präsentation der Ergebnisse für den Administrator. Eine Zusammenfassung der be-

P. Schartner · N.N (Hrsg.) · DACH Security 2015 · syssec (2015) pp-pp.

sonderen Herausforderungen beim Entwickeln eines Systems mit diesen Funktionalitäten bieten Bhatt et. al. in [BMZ14].

Die konkrete Umsetzung dieser Funktionalitäten gestaltet sich jedoch bei den verfügbaren Produkten auf vielen Ebenen schwierig. Bei der Datenerhebung werden häufig proprietäre Inseln geschaffen, da jeder Hersteller (eine Übersicht bietet bspw. [KNR14]) seine eigene Strategie verfolgt und seine Komponenten möglichst umfassend etablieren möchte. Die besten Erkennungsergebnisse sind aber nur bei flexiblen Systemen zu erwarten, die aufbauend auf einem offenen Datenformat, Informationen nahezu beliebiger Systeme integrieren können. Nur so ist man auch in der Zukunft in der Lage, neue aktuellere Basis-Werkzeuge mit geringem Aufwand in das SIEM-System zu integrieren. Die Auswertung benötigt ein sinnvolles Konzept von der Definition relevanter Situationen, der effizienten Auswertung bis zu dem Umsetzen geeigneter Aktionen als Ergebnis. Auch hier ist die Flexibilität sowohl aus Sicht der Quellsysteme wie auch zur Beschreibung der sicherheitsrelevanten Situationen wichtig, um dynamisch auf künftige Bedrohungen reagieren zu können und die Bedürfnisse schnell an ein sich wandelndes Umfeld anpassen zu können. Auch bei der Präsentation sind einige Punkte zu beachten: Die Nachvollziehbarkeit der Vorgänge muss einerseits im Mittelpunkt stehen, andererseits sollen möglichst viele Informationen zur Verfügung gestellt werden. Geeignete Mechanismen müssen daher genutzt werden, um auch bei der Visualisierung den Nutzer zu unterstützen und so insgesamt eine Verbesserung der Sicherheitssituation zu erreichen.

Im hier zugrundeliegenden Forschungsprojekt wurden genau diese Punkte aufgegriffen um eine neue SIEM-Lösung mit dem Fokus auf kleine und mittelständische Unternehmen zu konzipieren. Die Grundlage stellt hierbei das IF-MAP-Protokoll der TCG dar, welches die gemeinsame Daten- und Kommunikationsbasis für das Projekt bildet. In diesem Paper werden weiterhin die zentralen Konzepte vorgestellt, die die SIEM-Funktionalitäten umsetzen. Hierbei stehen die Korrelation und die visuelle Aufbereitung der Daten im Vordergrund.

Der Rest des Beitrages gliedert sich in die folgenden Abschnitte. Zunächst werden im zweiten Kapitel die technischen Grundlagen zu IF-MAP kurz erläutert. Anschließend behandelt Sektion 3 die Gesamtarchitektur des Systems. Die folgenden Abschnitte 4 und 5 umfassen die zentralen Komponenten zur Auswertung der Informationen und der Darstellung der Ergebnisse. Zum Abschluss beschreiben wir in Kapitel 6 einen beispielhaften Anwendungsfall, um den Einsatz und die Funktionsweise der Komponenten besser zu erläutern. Abschließend werden eine Zusammenfassung der vorgestellten Arbeiten gegeben und in einem Ausblick auf weitere Forschungsarbeiten zukünftige Konzepte und Weiterentwicklungen diskutiert.

2 IF-MAP

IF-MAP ist ein offen spezifiziertes, herstellerunabhängiges Protokoll zum Austausch von Metadaten innerhalb eines Netzwerkes. Es ist zudem ein integraler Bestandteil des Trusted Network Connect (TNC) Frameworks der TCG, kann jedoch auch (wie in diesem Projekt) losgelöst von TNC verwendet werden. Die Grundspezifikation von IF-MAP [TCG14] beschreibt zunächst das Grundprinzip des Daten- und Kommunikationsmodells. IF-MAP beschreibt eine Client-Server-Architektur mit einem zentralen MAP-Server und beliebigen MAP Clients. Als Datenmodell liegt ein ungerichteter Graph vor, der aus *Identifiern* und *Metadaten* besteht. Identifier spiegeln hierbei Entitäten im Netzwerk wieder. Diese Identifier können Beziehungen zueinander haben. Sowohl Identifier als auch die Beziehungen können durch die Metadaten näher beschreiben werden.

Der MAP-Server verwaltet hierbei immer nur den aktuell gültigen Zustand der Informationen. Die Clients hingegen können Informationen veröffentlichen, verändern und abrufen. Zum Veröffentlichen stehen die Mechanismen *update* und *notify* zur Verfügung. Update veröffentlicht Informationen permanent, wohingegen *notify* die Informationen lediglich an die aktuell verbundenen Clients verteilt, ohne dass sie länger in dem MAP-Server verweilen. Mittels der *delete* Operation können Informationen wieder aus dem MAP-Graphen entfernt werden.

Der Abruf von Informationen erfolgt entweder über *subscriptions* oder mittels *search*-Anfragen. Durch *subscriptions* kann ein Client bestimmte Arten von Informationen abonnieren und erhält automatisch Updates vom MAP Server, wenn sich etwas Entsprechendes im Graphen ändert. Search-Anfragen hingegen sind synchrone Aufrufe, die eine Ergebnismenge aufgrund des aktuellen Zustandes und der Suchkriterien direkt und einmalig zurückliefert.

Welche Art von Metadaten letztlich mit IF-MAP transportiert und gesammelt werden, kann durch eine separate Definition eines Metadatenschemas auf die jeweilige Anwendungsdomäne angepasst werden. Die TCG schlägt unter anderem ein Schema speziell für die Anwendung im Bereich der Netzwerksicherheit vor [TCG12]. Dieses wurde hier verwendet, allerdings gemäß des vorgesehenen Erweiterungsmechanismus um SIEM-spezifische Elemente ergänzt.

Diese Ergänzung war notwendig, da der Standard nicht alle Informationen und Aspekte abdeckt, die für die hier verwendeten Analysen notwendig sind. Weiterführende Konzepte wie Services oder die Abbildung von Schwachstellen sind dort noch nicht definiert.

Aufgrund der Struktur der Erweiterungen in IF-MAP ist eine grundsätzliche Kompatibilität zu weiteren Komponenten immer gewährleistet. Dennoch hat die Verwendung des eigenen Schemas zur Folge, dass Komponenten von Dritten sich ebenfalls an die hier verwendeten Erweiterungen halten müssen, oder aber nur einen Teilbereich der Informationen und Funktionen nutzen können.

3 Gesamt-Architektur mit Metadaten-Kommunikation

Die Gesamt-Architektur setzt sich logisch aus zwei Schichten zusammen. Zum einen aus **Netzwerk-Kollektoren** und **Flow-Controllern** und zum anderen aus der sogenannten **SIMU-Engine**. Die Netzwerk-Kollektoren sammeln und veröffentlichen verschiedene Informationen über das Netzwerk, während die Flow-Controller zusätzlich genutzt werden können, um geeignete Maßnahmen bei der Entdeckung besonderer Situationen umzusetzen. Die Kollektoren und Flow-Controller sind daher meist angepasste bestehende Netzwerkkomponenten, die um eine Schnittstelle für IF-MAP erweitert wurden, um ihre Informationen entsprechend zu publizieren oder auf Daten zu reagieren. Die **SIMU-Engine** auf der anderen Seite persistiert, verarbeitet und visualisiert die gesammelten Daten. Hier wird die Erkennung und Darstellung von Vorgängen im Netzwerk umgesetzt. Die Elemente der beiden Schichten kommunizieren im Wesentlichen über IF-MAP und den zentralen MAP-Server. Lediglich innerhalb der SIMU-Engine sind weitere Kommunikationswege notwendig, um beispielsweise die Administration der Regelverarbeitung und der Visualisierung von der Datenmenge, die zur Auswertung genutzt wird, getrennt zu halten. Abb. 1 zeigt die SIMU-Architektur des Projekts, die das IF-MAP-Protokoll als zentrales Protokoll zum Austausch von Metadaten nutzt.

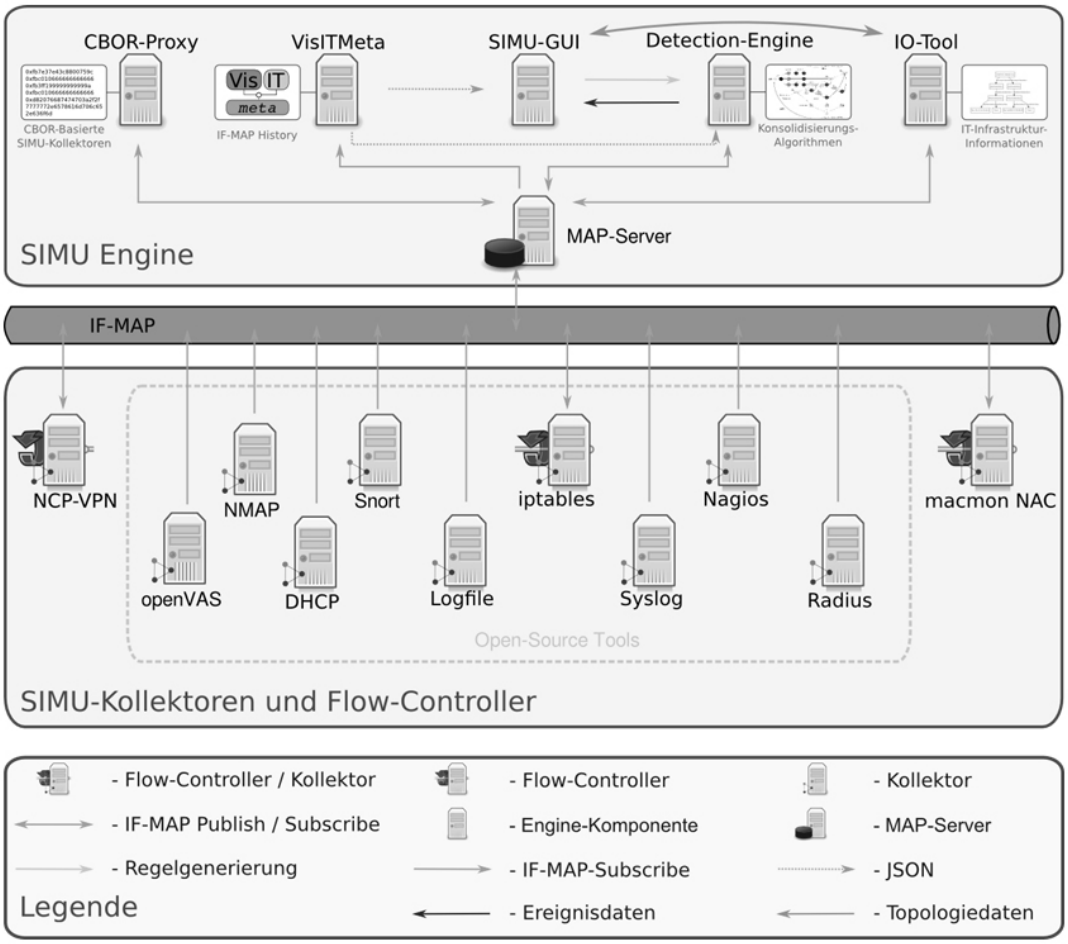

Abb. 1: SIMU-Architektur

Durch die Festlegung auf ein Metadatenschema wird sichergestellt, dass die damit verbundene Semantik eine Vermittlung von Informationen zwischen den verschiedenen IF-MAP-Clients ermöglicht. Die Kollektoren verwenden dafür das in Abb. 2 dargestellte Schema, wobei hier aus Gründen der Übersichtlichkeit nur ein grober und z.T. vereinfachter Überblick gegeben wird, wie die Metadaten(-bündel) der einzelnen Komponenten miteinander in Beziehung stehen können. Ein konkreteres Beispiel wird später im Rahmen des Anwendungsfalls näher betrachtet. Die Übersicht dient lediglich dazu, die Vielfältigkeit der Informationen und gleichzeitig die Mächtigkeit der Abbildung mittels IF-MAP zu verdeutlichen.

Wie zu sehen ist, ergeben sich meist mehrere Möglichkeiten für die Umsetzung der Metadaten: eine Abbildung auf die in [TCG12] definierten Standard-Metadaten (meta: Präfix) oder die Abbildung in eine speziell für das Projekt entwickelte Erweiterung des Metadatenschemas (simu: Präfix). Eine Abbildung in Standard-Metadaten wird bevorzugt, um die Kompatibilität mit anderen Komponenten außerhalb des Projektes zu erhalten. Für bestimmte Informationen bietet sich allerdings die Abbildung in ein eigenes Schema an, da auf diese Weise Information und die Semantik von Beziehungen besser abgebildet werden können.

Eine Besonderheit stellt hierbei das „Event“-Metadatum aus [TCG12] dar. Dies wird vorrangig verwendet, um flüchtige Daten abzubilden. Deshalb wird in [TCG12, Seite 20] empfohlen, solche Metadaten nur über die Notify-Operation zu veröffentlichen. Für das Projekt wer-

den daher zum Teil eigene Metadaten für Ereignisse definiert, die mit der Update-Operation veröffentlicht werden. Dies geschieht aus zwei Gründen:

a. Die Metadaten sollen auch für IF-MAP-Clients zugreifbar sein, die erst nach der Veröffentlichung der Events eine Verbindung zum MAP-Server aufbauen – und auch in der Historie der Daten aufgenommen werden.

b. Ein Überschwemmen des MAP-Servers wird durch eine geeignete Vorverarbeitung, eine Verdichtung und Reduzierung der Ereignisse auf Kollektorseite realisiert.

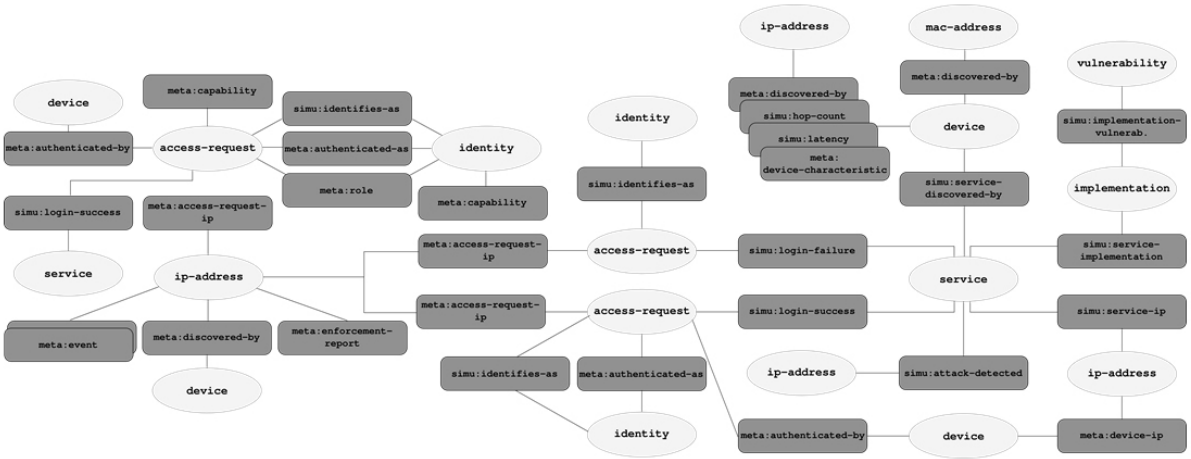

Abb. 2: Überblick des SIMU-Metadatenschemas

4 Visualisierung der Informationen

Die Aufbereitung und Darstellung von sicherheitsrelevanten Informationen werden im Projekt durch zwei verschiedene Komponenten vorgenommen. Während die erste Komponente auf die Visualisierung der Metadaten mit deren Kommunikationsbeziehungen spezialisiert ist, gibt die SIMU-GUI eine Übersicht über die Vorfälle wieder, inkl. des auftretenden Sicherheitsrisikos.

4.1 IF-MAP Graph-Visualisierung

VisITMeta (siehe auch [AHH14]) ist eine Software, die im gleichnamigen BMBF-geförderten Drittmittelprojekt (April 2012 bis März 2015) entwickelt wird und die Speicherung von IF-MAP Graphen und deren Visualisierung zur Verfügung stellt. Mit den Funktionen von VisITMeta können im SIMU-Projekt sowohl die IF-MAP-Rohdaten (veröffentlicht durch die verschiedenen Kollektoren) als auch die Ergebnisse von Korrelationen und anderen Auswertungen angezeigt werden.

Die Software selber teilt sich in zwei Komponenten auf:

a. Datenservice

b. Visualisierungskomponente

Der Datenservice ist ein Dienst, welcher IF-MAP-Daten wie ein gewöhnlicher IF-MAP-Client über eine oder mehrere parallele Abonnements (Subscriptions) erfasst und in einer Graph-Datenbank speichert. Die gespeicherten Daten können dann über eine REST-artige

Schnitt-stelle abgerufen werden. Zu den möglichen Abfragen gehören der aktuelle Graph-Zustand, Zustände zu beliebigen Zeitpunkten und Änderungen zwischen zwei beliebigen Zeitpunkten.

Hierzu wird auch eine Liste der Änderungen verwaltet und bereitgestellt, die Informationen enthält, wann wie viele und welche Änderungen an den Daten vorgenommen wurden. Die gespeicherten Daten zusammen mit der definierten Schnittstelle zum Abfragen der Daten können dann sowohl zur Visualisierung verwendet werden oder aber um darauf Auswertungen durchzuführen, welche die Historie der Daten miteinbeziehen können.

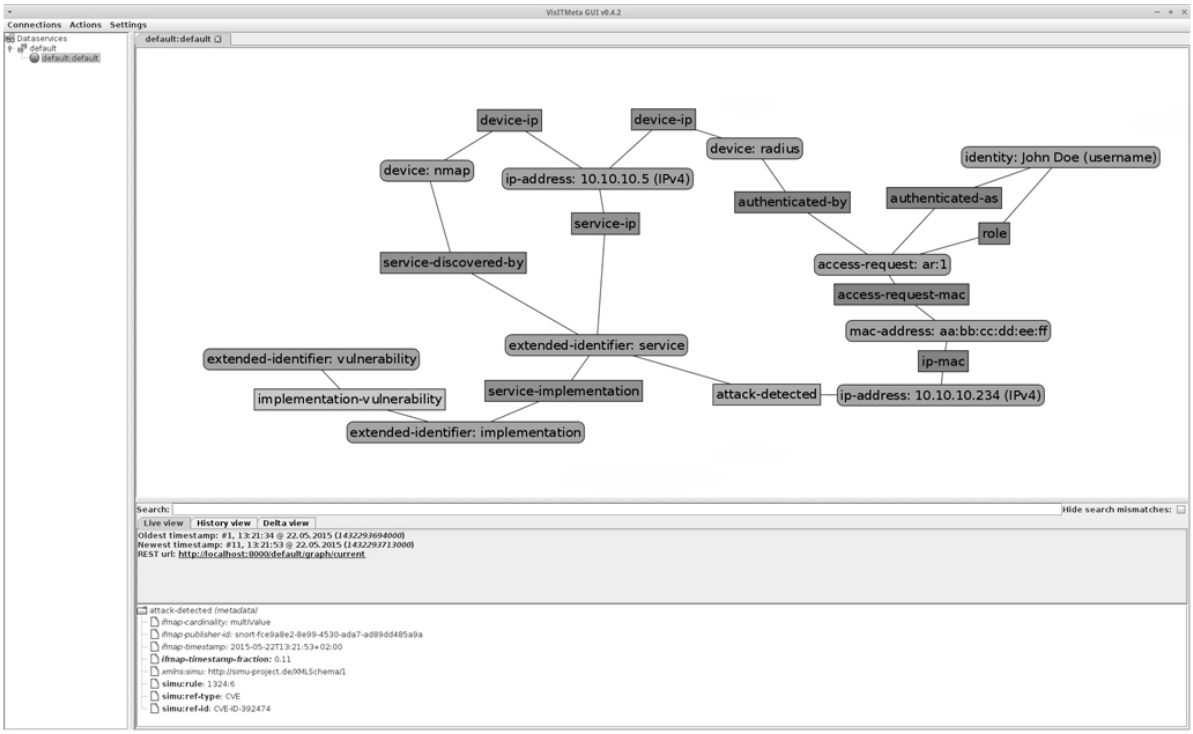

Abb. 3: Benutzeroberfläche von VisITMeta

Die Visualisierungskomponente (Benutzeroberfläche, siehe Abbildung 3) holt sich über die REST-Schnittstelle des Datenservice die benötigten Daten, um den Zustand des Graphen darstellen zu können. Die Visualisierungskomponente hat also keine direkte Verbindung zu einem MAP Server, sondern immer nur zu einem (oder mehreren) Datenservice-Instanzen. Diese Verbindungen können über die GUI konfiguriert werden.

Zur Darstellung der MAP-Daten (siehe Abb. 3) werden Identifier und Metadaten als Knoten sowie Links als Verbindung zwischen diesen Knoten abgebildet. Metadaten auf einem Link erscheinen in der Darstellung als Knoten mit Verbindungen zu den Identifiern an beiden Enden eines Links. Die Visualisierung unterstützt verschiedene Graph-Layouting-Algorithmen (Force-Directed, Spring, Bipartit), die im laufenden Betrieb gewechselt werden können.

Die Navigation durch die Historie der Graph-Daten erfolgt durch die Aufteilung in drei Betriebsmodi: im Live-Betrieb wird immer der aktuelle Stand des Metadatengraphen beobachtet und bei Neuerungen entsprechend angepasst. Ein weiterer Modus ist die Betrachtung des Graphen zu einem gegebenen Zeitpunkt. Über den Delta-Betrieb können die Änderungen zwischen zwei gewählten Zeitpunkten betrachtet werden. Einstellungsmöglichkeiten für die Farbgebung der Elemente und ein Highlighting bei Veränderungen werden eingesetzt um eine

übersichtliche Darstellung und Verfolgung der Ereignisse zu ermöglichen. Eine einfache Suche ermöglicht das Finden und Hervorheben von Elementen aufgrund ihrer Inhalte.

Im unteren Bereich des Anwendungsfensters werden detaillierte Informationen über einzelne Graph-Elemente angezeigt. Hierzu gehören alle durch IF-MAP vorgegebenen Attribute (Zeitpunkt der Veröffentlichung, ID des veröffentlichenden Clients, XML-Schema-Definitionen etc.) und auch die Werte und innere Struktur des konkreten Identifiers oder Metadatums.

4.2 SIMU-GUI

Während VisITMeta die Möglichkeit bietet die IF-MAP Rohdaten des Systems einzusehen ist eine weitere Visualisierungskomponente namens **SIMU-GUI** notwendig, über die das System von IT-Administratoren verwendet werden kann. Diese wurde als Java-Webanwendung umgesetzt und kann mit allen gängigen Browsern verwendet werden. Das Web-Interface ist mit Hilfe von JavaScript und AngularJS als Single-Page-Application implementiert worden. Dieser Ansatz reduziert die Datenmenge, die zwischen Server und Client ausgetauscht werden muss, da das Grundgerüst der Seite nur einmalig geladen werden muss.

Die SIMU-GUI (siehe Abb. 4) ermöglicht die Einsicht und Bearbeitung von erkannten Vorfällen. Weiterhin bietet sie die Möglichkeit den Teil des IF-MAP Metadatengraphen einzusehen, der zur Meldung eines Vorfalls geführt hat. Dadurch lässt sich im Detail nachvollziehen was genau von der Detection Engine erkannt wurde. Ein Vorfall kann außerdem einem Bearbeiter zugewiesen werden, der unter anderem in der Lage ist den Status des Vorfalls zu ändern. Es ist jedoch auch anderen Benutzern möglich die Bearbeitung eines Vorfalls zu kommentieren, wodurch eine gemeinsame Problemlösung vereinfacht wird.

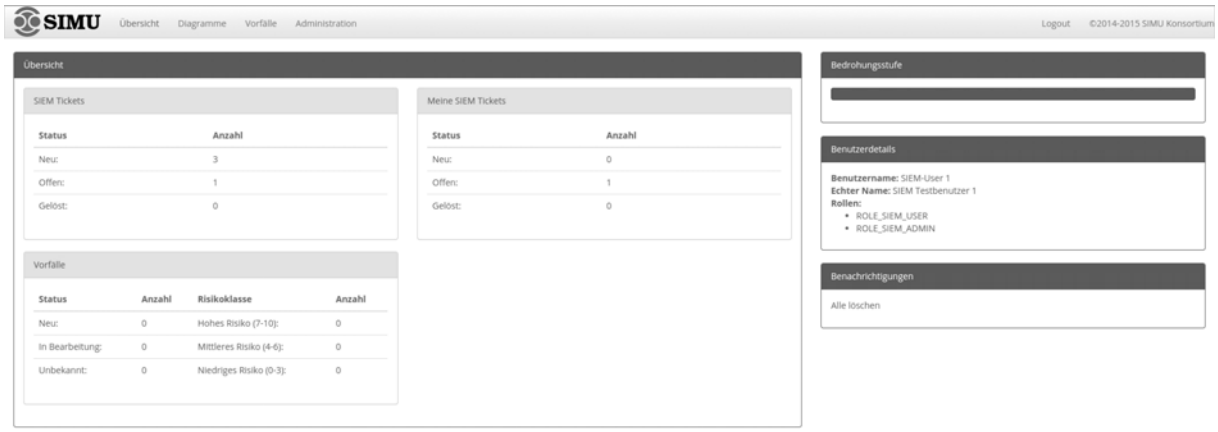

Abb. 4: Oberfläche der SIMU-GUI

Um die genannten Funktionen umzusetzen integriert die SIMU-GUI verschiedene externe Dienste in das System. Diese werden über Konnektoren lose an das GUI-System gekoppelt und sind dadurch austauschbar. Dadurch lässt sich die SIMU-GUI bei Bedarf in bereits bestehende Systeme integrieren und benötigt somit nur wenige Dienste, die zusätzlich im Firmennetzwerk aufgebaut werden müssen. Als externe Dienste werden genutzt:

a. Ticketsystem
b. Benutzerverwaltung
c. Sicherheitsvorfälle

d. VisITMeta Dataservice

Die **Vorfälle** werden aus einer Datenbank gelesen, in der sie von der Detection Engine abgelegt werden. Dieser Weg ist vom SIMU-System vorgegeben, das verwendete DBMS kann jedoch variiert werden. Ebenfalls vorgegeben ist die Verbindung zum **VisITMeta Dataservice**, die benötigt wird um die IF-MAP-Rohdaten zu einem Vorfall aus dem Metadatengraphen auszulesen. Der Dataservice kann dabei auf der gleichen Maschine wie die SIMU-GUI oder aber auch auf anderen Rechnern im lokalen Netzwerk installiert sein. Über ein VPN ließe sich auch auf einen Dataservice an einem anderen Standort zugreifen.

Das **Ticketsystem** und die **Benutzerverwaltung** sind generische Dienste, die von verschiedenen Systemen erfüllt werden können. Dieses Vorgehen hat den Vorteil, dass die SIMU-GUI bei Bedarf mit überschaubarem Aufwand in ein bestehendes Netzwerk integriert werden kann und vorhandene Dienste verwendet werden können. So ist es zum Beispiel nicht notwendig ein zweites Ticketsystem in Betrieb zu nehmen, wenn in der Firma bereits eines in Verwendung ist. Im einfachsten Fall werden beide Aufgaben vom gleichen System übernommen, so dass keine zusätzliche Verknüpfung zwischen SIMU-GUI- und Ticketsystem-Benutzern hergestellt werden muss. Dadurch ist im Idealfall kein Zugriff auf die Benutzeroberflächen der externen Dienste notwendig und alles kann direkt über die SIMU-GUI konfiguriert werden.

Die Tickets des Ticketsystems dienen der Nachverfolgung der Bearbeitung eines Vorfalls. So wird über ein Ticket der Status der Bearbeitung gespeichert, es werden Kommentare und Zeitbuchungen abgelegt und Deadlines für die Behebung definiert. Aus Benutzersicht wird allerdings ausschließlich mit Vorfällen interagiert, welche Informationen tatsächlich in der Vorfalldatenbank und welche im Ticketsystem hinterlegt werden ist für den Anwender nicht wichtig und daher auch nicht ersichtlich. Dies vereinfacht die Bedienung der SIMU-GUI zusätzlich, da prinzipiell nur mit einer einzigen Art von Information, den Vorfällen, umgegangen werden muss. Alles weitere, zum Beispiel die Rohdaten und die Nachverfolgung über das Ticket, wird in die entsprechende Darstellung des Vorfalls integriert.

Zusätzlich ist die Verwendung von **WebSockets** für die Kommunikation zwischen dem Web-Interface und dem Back-end der SIMU-GUI vorgesehen. Neben der üblichen Request-Response-Kommunikation, die auch über eine REST-Schnittstelle möglich wäre, lassen sich durch diese Technik Push-Benachrichtigungen vom Server an den Client senden. Dies wird dazu verwendet um den Anwender nahezu in Echtzeit über neue Vorfälle und andere wichtige Vorgänge im überwachten System über die SIMU-GUI zu informieren.

Die **Datenbank**, in der die erkannten Vorfälle von der Detection Engine abgelegt werden, wird von der SIMU-GUI periodisch auf neue Einträge durchsucht. Wird ein neuer Vorfall gefunden, so wird zunächst im Ticketsystem gesucht ob es bereits ein Ticket zu diesem Vorfall gibt (z.B. beim Neustart des Systems). Wird keines gefunden, so erstellt die SIMU-GUI über den entsprechenden Konnektor ein neues Ticket und verknüpft dieses mit dem Vorfall. Die Detection Engine kann einem Vorfall eine Handlungsempfehlung mitgeben um den Anwender bei der Behebung des Problems zu unterstützen. Ist diese in der Datenbank hinterlegt, so wird sie bei der Erstellung des Tickets als Text eingetragen und kann so bei der Ansicht des Vorfalls über das Web-Interface direkt eingesehen werden.

Neben der Bearbeitung von Vorfällen bietet die SIMU-GUI auch Funktionen, um einen schnellen Überblick über den Zustand des Netzwerks zu bekommen. Es gibt zwei Unterseiten, die eine Zusammenfassung des Systemzustands liefern. Die eine bietet eine tabellarische

Übersicht über den Status unbearbeiteter Vorfälle und deren **Risikoeinschätzung**. Die zweite Seite stellt ähnliche Informationen in Form von Graphen und Diagrammen dar, was zum Beispiel das Erkennen von zeitlichen Entwicklungen gegenüber der tabellarischen Ansicht vereinfacht. Ein weiteres Feature ist eine graphische Darstellung der aktuellen **Bedrohungsstufe** (Threat Level) im Netzwerk. Diese wird aus der Risikobewertung der aktiven, also noch nicht behobenen, Vorfälle errechnet und auf jeder Unterseite der SIMU-GUI angezeigt.

4.3 Darstellung von Regelverstößen

Verstöße gegen die Regeln der Detection Engine können auf zwei verschiedene Arten innerhalb der SIMU-GUI bzw. durch VisITMeta dargestellt werden.

a. **Darstellung der zu einer Regelauslösung gehörenden Teilgraphen:** Hierbei werden nur die Teile des gesamten MAP-Graphen dargestellt, die durch ein Graph-Muster der Detection Engine für das Auslösen einer konkreten Regel erkannt wurden, sozusagen die Instanz eines Graph-Musters. Die Detection Engine kann hierzu der SIMU-GUI diese Informationen mitsenden. Die VisITMeta-Visualisierungskomponente muss hierzu mitgeteilt bekommen, dass sie nur den entsprechenden Teilgraphen abrufen und darstellen soll, z.B. durch Übermittlung eines passenden Filterausdrucks.

b. **Regel-unabhängige Betrachtung:** Alternativ dazu kann auch der gesamte MAP-Graph dargestellt werden. Regelauswertungen werden hier gleichgestellt mit allen anderen MAP-Daten visualisiert. Mit den üblichen Mitteln zur Navigation, Suche und Filterung kann der Benutzer den Graphen betrachten. Diese Darstellung wird durch die gewöhnlichen Ansichten der VisITMeta-Visualisierungskomponente abgebildet.

c. **Kombination beider Ansätze:** Werden beide Ansätze kombiniert, sieht der Benutzer grundsätzlich alle Daten des MAP-Graphen wie in Variante 2, allerdings werden die Graph-Muster-Instanzen hier speziell hervorgehoben. So kann direkt der Kontext der erkannten Graph-Muster-Instanzen in Bezug auf den Rest des Graphen bzw. die direkte Nachbarschaft betrachtet werden.

Unabhängig davon, ob nur Teilgraphen oder der gesamte Graph mit den Regelverstößen betrachtet wird, können für die eigentliche visuelle Darstellung der Regelauswertungen verschiedene grafische Konzepte verwendet werden. Im einfachsten Falle werden die von der Detection Engine publizierten Metadaten wie alle anderen Metadaten auch dargestellt (gleiche Form, Umrandung, etc.). Die GUI kann die Metadaten und Identifier, welche zu einer Regelauswertung gehören alternativ mit einer eigenen, speziell dafür konfigurierten Farbe und Form anzeigen um sie so hervorzuheben und damit für den Benutzer leichter von anderen Elementen unterscheidbar zu machen.

5 Detection Engine

Die **Detection Engine** stellt die Auswertekomponente im SIMU-System dar. In ihr sollen Ereignisse und Zustände analysiert werden, um besondere Situationen auch auf einem höheren Level als einer konventionellen Angriffserkennung, z.B. (N)IDS, feststellen zu können.

In der Architektur ist die Detection Engine bei dem Datenservice von VisITMeta anzusiedeln, da sie dessen Informationen als Grundlage verwendet. Hierdurch ist es möglich, auch die Historie des Graphen zu durchlaufen und so auch Zwischenfälle vor dem Starten der Komponen-

te zu erkennen. Dazu wird ein Modell des Graphen verwaltet, das entsprechend der Historie aller Informationen konstruiert wird. Bei Vorliegen neuer Informationen wird der Graph entsprechend erweitert bzw. angepasst. Für die Erkennung von Vorfällen werden Listener-Klassen implementiert, die die Erkennungslogik für komplexe Vorfälle basierend auf aktuellen und/oder historischen Daten des Graphen implementieren. Diese können auf dem Modell des Graphen registriert werden und erhalten bei Veränderungen eine Benachrichtigung.

Im Augenblick existiert eine Implementierung der Listener, die einen Mustervergleich auf dem Graphen vornimmt. Muster werden dazu selbst als Graph definiert, wobei Identifier und Metadaten, sowie ihre Beziehungen und Attribute beschrieben werden können - Abhängigkeiten zwischen Komponenten und Attributen können hierbei berücksichtigt werden. Dieses Muster wird dann gegen den aktuellen Zustand des Graphen geprüft und für jede Übereinstimmung, die in der Regel zusammen mit dem Muster definierten Aktionen ausgeführt.

Als Reaktion können sowohl neue Metadaten und Informationen in den Graphen eingefügt werden – um dadurch beispielsweise ein automatisiertes „Enforcement“ durchführen zu lassen. Zusätzlich wird das Auslösen von Regeln (also hier das Erkennen eines beschriebenen Angriffsmusters im Graphen) in Form von Vorfällen für die SIMU-GUI aufbereitet und dokumentiert. Die Informationen hierzu werden bei der Erstellung der Regel ebenfalls mit angegeben. Inhalte des Musters können in der textuellen Beschreibung aber verwendet werden, so dass der Vorfall die spezifischen Informationen genau des einen passenden Musters enthält.

Neben dem aktuellen Graph Pattern Matching sind weitere Algorithmen möglich, beispielsweise um kompliziertere Vergleichsverfahren und Analysen umzusetzen. Zusätzlich kann hierbei durch das Listener-Prinzip – also durch das Reagieren auf Änderungen – eine zeiteffiziente Auswertung erfolgen, da nur auf Basis der Änderungen und nur zum Zeitpunkt der Änderungen die Überprüfung durch die Algorithmen erfolgen muss. Außerdem erfolgt der Zugriff durch die Listener zunächst nur lesend. Es kann also eine parallelisierte Auswertung der verschiedenen Regeln und Algorithmen erfolgen.

6 Anwendungsfall als Beispiel

Eine besondere Stärke des SIEM-Ansatzes besteht darin, Informationen aus unterschiedlichen Quellen zur Erkennung von sicherheitsrelevanten Vorfällen kombinieren zu können. Dieses bietet einen erheblichen Mehrwert, da hierdurch z.T. Vorfälle überhaupt erst erkannt oder auch in ihrer Risikostufe besser eingeschätzt werden können. Dies soll im Folgenden durch einen Anwendungsfall exemplarisch aufgezeigt werden.

Der Anwendungsfall beschreibt die Erkennung eines Sicherheitsvorfalls in einem kleineren Firmennetzwerk. In diesem Netzwerk läuft u.a. ein SSH-Server. Zur Zugangskontrolle ist ein RADIUS-Server im Einsatz. Weiterhin sind ein Netzwerkscanner [NMAP], ein Schwachstellenscanner [OVAS] und ein Netzwerk Intrusion Detection System [SNORT] aktiv. Nmap kann ganze Netzbereiche oder spezifische Hosts analysieren und so vorhandene Systeme, laufende Betriebssysteme und angebotene Services (meist aufgrund der Ports) samt Version identifizieren. OpenVAS durchsucht im Netzwerk befindliche Hosts auf Sicherheitslücken. Snort wiederum liest den Netzwerkverkehr mit und überprüft ihn auf der Basis von Signaturen um Angriffsmuster zu erkennen. Die gesamten Informationen werden zentral auf dem MAP-Server gesammelt und so zur Analyse und Visualisierung den weiteren SIEM Komponenten bereitgestellt.

Die einzelnen Schritte stellen sich wie folg dar. Zunächst werden durch Nmap und OpenVAS die Informationen zum Netzzustand publiziert. So erkennt z.B. Nmap den SSH Server und OpenVAS stellt fest, dass die im Einsatz befindliche SSH Version eine bekannte Schwachstelle hat. Dies ist in Abb. 5 im oberen Teil zu erkennen. Die Informationen über den SSH-Server werden am Knoten „simu:service“ angehängt; die von Nmap publizierten Informationen beinhalten u.a. auch die nötigen Details zur Implementierung im Knoten „simu:implementation“. An diesen Knoten hängt OpenVAS die Information zur erkannten Schwachstelle an: „simu:vulnerability“.

Snort überwacht permanent den Netzwerkverkehr und erkennt dabei das Muster eines bestimmten Angriffs. Diese Information wird als Kante „simu:attack-detected“ zwischen dem Service und der verursachenden IP-Adresse eingetragen. Durch die vorher von RADIUS publizierten Informationen im unteren Teil des Graphen kann der hinter dem Angriff stehenden Benutzer „john.doe“ identifiziert werden (Knoten „identity“).

Die Detection Engine hat jetzt alle Informationen, um den Vorfall zu erkennen und seine Bedeutung einzuschätzen. Insbesondere wird erkannt, dass der Server gegenüber dem Angriff tatsächlich verwundbar ist. Dies verleiht dem Vorfall ein hohes Bedrohungspotential, weshalb die Detection Engine eine entsprechend klassifizierte Meldung in den IF-MAP-Graphen publiziert. Die Schwere ist an dem Attribut „significance=important“ an der Kante „meta:unexpected-behavior“ zu erkennen.

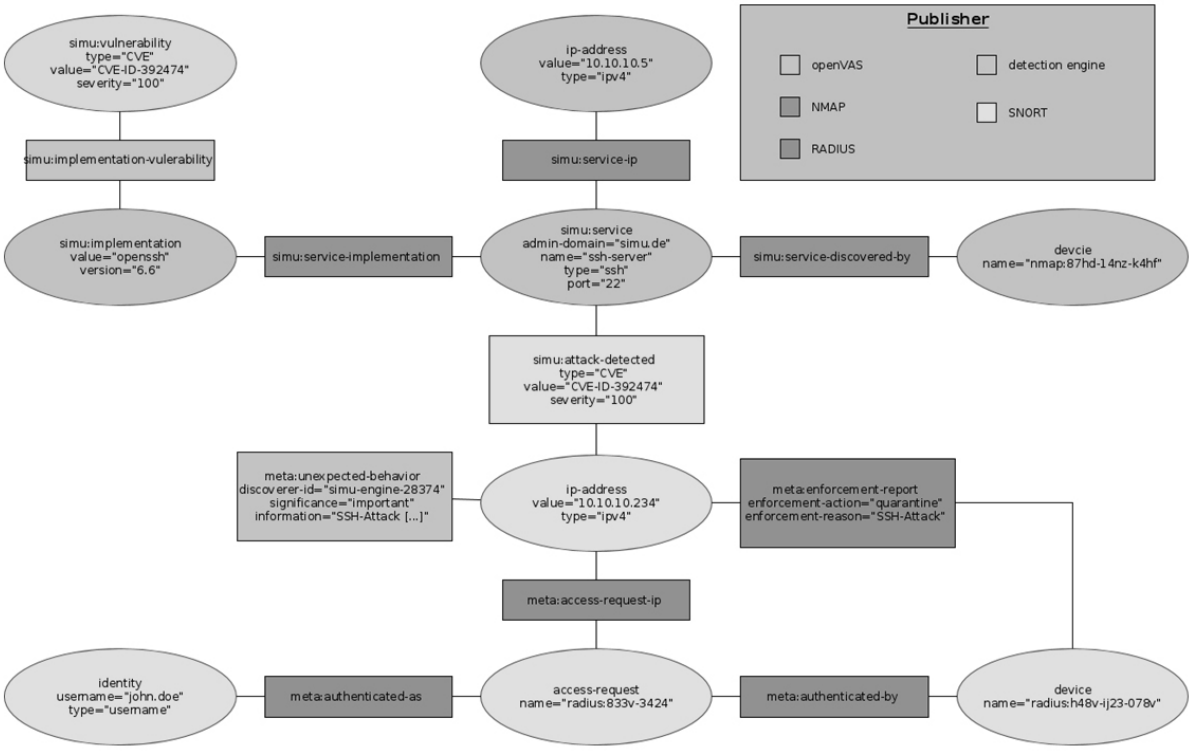

Abb. 5: Metadaten zu dem Anwendungsfall

Die Meldung der Detection Engine führt dann wiederum dazu, dass der Radius-Server den Netzzugang für den entsprechenden Benutzer sperrt. Auch diese Aktion wird letztlich wieder im IF-MAP-Graphen publiziert in Form der Kante „meta:enforcement-report“.

Abschließend kann ein Administrator über die graphische Oberfläche den Vorfall einsehen. Dabei werden die Teile des IF-MAP-Graphen hervorgehoben, die zur Meldung geführt haben, so dass klar ersichtlich ist, welche Konstellation zum Aussperren des Nutzers geführt hat

7 Zusammenfassung

Im beschriebenen Projekt wurde eine neue SIEM-Architektur entwickelt, die auf Basis des IF-MAP-Protokolls der Trusted Computing Group (TCG) in der Lage ist, verschiedene Log-Informationen und Events diverser Sicherheitskomponenten zu konsolidieren und auszuwerten. Dazu wurde ein Metadaten-Schema für ein homogenes Ereignistransportprotokoll ausgearbeitet. Durch die Visualisierung der Kommunikationsbeziehungen über VisITMeta lassen sich unerwünschte Zustände herausfiltern und getrennt von den anderen Daten analysieren. Zudem ist die Detection Engine in der Lage, Angriffsvektoren auszumachen und in nahezu Echtzeit darauf zu reagieren. Hier wird zum einen die Weiterleitung erkannter Vorfälle gemäß der definierten Sicherheitsprozesse unterstützt, die bspw. die Verantwortlichkeiten zur Reaktion dem Administrator überlässt. Es besteht aber auch die Möglichkeit einer automatisierten Ereignisbehandlung durch das Erzeugen weiterer spezifischer Metadaten.

Die wesentlichen Vorteile der Detection Engine bestehen zum einen darin, dass alle im Graphen enthaltenen Informationen der diversen Kollektoren und Flow Controller in die Erkennung von Sicherheitsvorfällen einbezogen werden können, da diese in einem einheitlichen Datenformat in den Graphen geladen werden. Dies ermöglicht eine deutlich bessere und komplexere Detektion von Sicherheitsvorfällen. Zum anderen können neben den aktuellen Metadaten auch beliebige historische Zustände, die vom Data-Service bereitgestellt werden, in die Erkennung der Sicherheitsvorfälle mit einbezogen werden. Zusätzlich besteht durch die flexible Definition der Erkennung von Vorfällen auf Basis des Konzepts der Listener eine sehr flexible und leicht erweiterbare Möglichkeit der Definition relevanter Ereignisse, die zudem durch die Benachrichtigungen nahezu in Echtzeit erkannt werden können.

Durch die übersichtliche Oberfläche der SIMU-GUI können Vorfälle, die von der Detection Engine erkannt werden, einheitlich dargestellt werden. Bei neuen Vorfällen wird das Ticketsystem kontaktiert, so dass auch eine Bearbeitung der Vorfälle schnell möglich und gut dokumentiert ist. Durch die Konsolidierung weiterer sicherheitsrelevanter Daten in der GUI kann eine Art Gesamt-IT-Bedrohungsstufe eines Unternehmens anhand der Risikobewertung ermittelt und mit entsprechenden Farbbalken dargestellt werden. Eine strukturierte und priorisierte Bearbeitung von Sicherheitsvorfällen wird so ermöglicht.

8 Danksagung

Das SIMU-Projekt (www.simu-project.de) ist ein gefördertes BMBF-Projekt mit einer Laufzeit von zwei Jahren, das im Oktober 2013 seine Arbeiten begonnen hat. An dem Projekt sind die Firmen *DECOIT GmbH* (Projektleitung), *NCP Engineering GmbH, macmon secure gmbh* sowie die deutschen Forschungseinrichtungen *Fraunhofer SIT* und *Hochschule Hannover* beteiligt. Daher gilt der Dank den Partnern des Projektes, die durch ihre Beiträge und Arbeiten diesen Bericht erst ermöglicht haben.

Literatur

[AHH14] Ahlers, Heine, Hellmann, Kleiner, Renners, Rossow, Steuerwald: *Replicable security monitoring: Visualizing time-variant graphs of network metadata.* Joint Proceedings of the Fourth International Workshop on Euler Diagrams (ED 2014) and the First International Workshop on Graph Visualization in Practice (GViP 2014) co-located with Diagrams 2014, Hrsg. Jim Burton, Gem Stapleton u. Karsten Klein, Melbourne (Australien) 2014

[BMZ14] Bhatt, S., Manadhata, P.K., Zomlot, L.: *The Operational Role of Security Information and Event Management Systems,* Security & Privacy, IEEE , vol.12, no.5, pp.35,41, Sept.-Oct. 2014

[DRS14] Detken, Rossow, Steuerwald: *SIEM-Ansätze zur Erhöhung der IT-Sicherheit auf Basis von IF-MAP*. D.A.CH Security 2014: Bestandsaufnahme, Konzepte, Anwendungen und Perspektiven, ISBN 978-3-00-046463-8, Hrsg. Peter Schartner u. Peter Lipp, syssec-Verlag, Graz (Österreich) 2014

[DETK14] Kai-Oliver Detken: *Mappen von Sicherheitsereignissen: Einsatz von IF-MAP als Integrationsprotokoll zur Konsolidierung von Systemmeldungen*. NET 09/14, ISSN 0947-4765, NET Verlagsservice GmbH, Woltersdorf 2014

[DETK14b] Kai-Oliver Detken: *Intelligentes Monitoring - SIEM-Lösungen im Einsatz*. NET 06/14, ISSN 0947-4765, NET Verlagsservice GmbH, Woltersdorf 2014

[KNR14] Kavanagh, Nicolett, Rochford: *Gartner Magic Quadrant for Security Information and Event Management Report*. Gartner Report, Juni 2014

[NMAP] NMAP-Projektwebseite: http://www.nmap.org

[OVAS] OpenVAS-Projektwebseite: http://www.openvas.org

[SNORT] SNORT-Projektwebseite: http://www.snort.org

[SIMU15] SIMU-Projektwebseite: http://www.simu-project.de

[SWI06] Swift, David: *A practical application of SIM/SEM/SIEM automating threat identification*. SANS Infosec Reading Room, The SANS, 2006.

[TCG14] Trusted Computing Group: *TNC IF-MAP Binding for SOAP*. Specification Version 2.2, Revision 9, März 2014

[TCG12] Trusted Computing Group: *TNC IF-MAP Metadata for Network Security*. Specification Version 1.1, Revision 8, Mai 2012

[WIL07] Amrit Williams Blog (Observations of a Digitally Enlightened Mind): *The future of SIEM - the market will begin to diverge*. January 2007