Echtzeit-Sicherheit für Unternehmensnetze durch Konsolidierung von Metadaten

Best-Practice-Report „public“

DOKUMENTINFORMATIONEN	
TYP	Bericht
TITEL	Best-Practise-Dokument
DATUM	27.03.2013
ARBEITSPAKET	AP5
FÖRDERKENNZEICHEN	16BY1050-16BY1054

ESUKOM-KONSORTIUM	
KOORDINATOR	DECOIT GmbH
PARTNER 1	NCPe GmbH Network Communications Products engineering
PARTNER 2	macmon secure gmbh
PARTNER 3	Fachhochschule Hannover
PARTNER 4	Fraunhofer-Gesellschaft zur Förderung der angewandten Forschung e. V.

DOKUMENTSTATUS		
AKTION	**DURCH**	**DATUM**
EINGEREICHT	DECOIT GmbH	31.09.2012
AP-LEITER	DECOIT (AP5.1 und AP5.2)	
GENEHMIGT	DECOIT GmbH	18.12.2012

ÄNDERUNGSSHISTORIE			
DATUM	**VERSION**	**AUTOR**	**KOMMENTAR**
01.08.2012	0.1	Dennis Dunekacke	Aufsetzen des Dokumentes
02.08.2012	0.2	Dennis Dunekacke	Kapitel 1,2,3 und 4 angefangen
03.08.2012	0.3	Dennis Dunekacke	Kapitel weiter ausgebaut, Anhang „IF-MAP-Issues eingefügt
14.08.2012	0.4	Dennis Dunekacke	Kapitel 4 weiter ausgebaut
16.08.2012	0.5	Prof. Dr. Kai-Oliver Detken	Kapitel 2 komplett aktualisiert
04.09.2012	0.5a	Jens Lucius	NCP VM eingetragen
14.09.2012	0.6	Dennis Dunekacke	Kapitel 3 erweitert
17.09.2012	0.7	Dennis Dunekacke	Kapitel 5 angefangen, Kapitel 3 überarbeitet/erweitert
18.09.2012	0.8	Dennis Dunekacke	Kapitel 4 aktualisiert/geändert
27.09.2012	0.9	Dennis Dunekacke	Kapitel 4 erweitert und umgestellt, Tabelle in Abschnitt 4.1 erweitert
27.09.2012	1.0	Dennis Dunekacke	Kapitel 5 aktualisiert
28.09.2012	1.1	Dennis Dunekacke	Kapitel 4 um Anwendungsfall 1 erweitert
31.10.2012	1.2	Dennis Dunekacke	Kapitel 4 um Anwendungsfall 2 erweitert, die anderen Anwendungsfälle um die Enforcement-Komponente ergänzt
28.11.2012	1.3	Dennis Dunekacke	Nochmal einzelne Textpassagen in Kapitel 4 ergänzt, Kapitel 6 angefangen und fertiggestellt, Durchsicht des Dokuments, Glossar eingefügt
18.12.2012	1.4	Prof. Dr. Kai-Oliver Detken	Finale Version angelegt und komplette Durchsicht des Berichts
27.03.2013	1.5	Prof. Dr. Kai-Oliver Detken	Nochmalige Erweiterung des Berichts und endgültige finale Version angelegt

KONTAKTINFORMATIONEN		
NAME	**ORGANISATION**	**EMAIL**
Prof. Dr. Kai-Oliver Detken	DECOIT GmbH	detken@decoit.de
Dennis Dunekacke	DECOIT GmbH	dunekacke@decoit.de
Jens Lucius	NCP GmbH	jens.lucius@ncp-e.com
Prof. Dr. Josef von Helden	Hochschule Hannover	josef.vonhelden@hs-hannover.de
Jürgen Westerkamp	macmon secure gmbh	juergen.westerkamp@macmon.eu
Nicolai Kuntze	Fraunhofer SIT	nicolai.kuntze@sit.fraunhofer.de

Inhalt

1 Einleitung

Dieses Dokument beschreibt die Erfahrungen und die daraus resultierenden Best-Practice-Richtlinien, welche sich innerhalb des vom Bundesministerium für Bildung und Forschung (BMBF) geförderten Verbundprojektes ESUKOM im Rahmen der Entwicklungsarbeiten ergeben haben. Außerdem wird aufgezeigt, wie der finale Demonstrator, welcher alle im Projekt entwickelten Komponenten umfasst, konzipiert und technisch umgesetzt wurde. Abschließend werden die Ergebnisse der einzelnen Tests aufgeführt und zusammengefasst.

Innerhalb dieses Dokuments wird in Kapitel 2 zunächst eine kurze, allgemeine Übersicht über die Inhalte und Ziele des ESUKOM-Projekts gegeben. Im darauf folgenden Kapitel 3 werden die aufgetretenen Probleme und Unklarheiten bezüglich der Implementierung der IF-MAP-Spezifikation aufgezeigt sowie die Ansätze, welche zur Lösung dieser Probleme zur Anwendung kamen. In Kapitel 4 folgt eine Beschreibung des finalen Demonstrators, welcher im Rahmen des Projekts umgesetzt wurde. Dabei werden die Konzepte und eingesetzten Technologien erläutert, auf deren Grundlage die Realisierung des Demonstrators letztendlich erfolgte sowie eine kurze Anleitung zum Aufsetzen und zur Benutzung des Demonstrators gegeben. Das Kapitel 5 zeigt die einzelnen Ergebnisse der durchgeführten Tests auf, in Kapitel 6 folgt eine abschließende Zusammenfassung der erzielten Ergebnisse des Projekts.

2 Beschreibung des ESUKOM-Projektes

Das Gesamtziel des ESUKOM-Vorhabens ist die Konzeption und Entwicklung einer Echtzeit-Sicherheitslösung für Unternehmensnetze, die basierend auf der Konsolidierung von Metadaten arbeitet. Dabei soll insbesondere der durch mobile Endgeräte wie Smartphones erzeugten Bedrohungslage Rechnung getragen werden. ESUKOM setzt auf die Integration vorhandener Sicherheitslösungen (kommerziell und Open Source) basierend auf einem einheitlichen Metadatenformat gemäß der IF-MAP-Spezifikation der Trusted Computing Group (TCG). Das bedeutet konkret, dass man als Grundlage für die Arbeitsweise eine gemeinsame Datenbasis verwendet, die den gesamten, aktuellen Zustand eines Unternehmensnetzes abbildet. Dabei sollen alle vorhandenen Sicherheitstools die Möglichkeit besitzen, aus dieser Datenbasis die für ihre Funktionsweise relevanten Daten zu beziehen und selbst neue Daten zu veröffentlichen. Erst dadurch wird es möglich, auf verschiedenste Bedrohungen in Echtzeit zu reagieren. Das ESUKOM-Vorhaben verfolgte hinsichtlich der Integration verschiedener Sicherheitsmechanismen den Ansatz ein wohl definiertes Modell zur Beschreibung von Metadaten zu erstellen, das nicht alleine auf der Analyse von Logdateien beruht.

Das Ziel des Vorhabens war daher zusammenfassend die Konzeption und prototypische Entwicklung einer ganzheitlichen Sicherheitslösung, die durch die Integration vorhandener Sicherheitstools und die Konsolidierung von Metadaten die Sicherheit eines Netzwerkes in Echtzeit gewährleistet. Als technologische Basis diente die IF-MAP-Spezifikation der Trusted Computing Group (TCG).

2.1 IF-MAP-Spezifikation

Die technologische Basis für das ESUKOM Projekt ist das IF-MAP (Interface for Metadata Access Point) Protokoll der TCG. Dabei handelt es sich um ein offenes, Hersteller-unabhängiges Client-Server Netzprotokoll zum Austausch von beliebigen, in XML codierten Metadaten. IF-MAP ist ein substantieller Bestandteil des Trusted Network Connect (TNC) Frameworks. Die ursprüngliche Motivation für IF-MAP war die Integration von vorhandenen, sicherheitsrelevanten Infrastrukturdiensten wie Firewalls, Virtual Private Networks (VPN) und Intrusion-Detection-Systemen (IDS). Durch die Integration dieser Dienste erhält man eine ganzheitliche Sicht auf den aktuellen Status eines Netzwerkes, was Vorteile bei der Administration und der Erkennung von Bedrohungen verspricht. Die erste Version (1.0) von IF-MAP erschien im Mai 2008. Die Version 2.1 (Revision 15) wurde im Mai 2012 veröffentlicht[1].

Die grundlegende Architektur von IF-MAP ist in Abbildung 1 dargestellt. Die zentralen Komponenten unterteilen sich in einen MAP-Server und eine Menge von IF-MAP-Clients. Der MAP-Server innerhalb des zu schützenden Netzwerkes ist dafür verantwortlich, den aktuellen Zustand des Netzwerkes abzubilden. Dieser Zustand wird anhand eines vorgegebenen Formates für Metadaten beschrieben und kann (sicherheitsrelevante) Informationen wie angemeldete Benutzer, verwendete IP-Adressen oder erkannte Anomalien enthalten.

Über die standardisierte IF-MAP-Schnittstelle können Metadaten von diesem Server abgefragt oder neue Metadaten veröffentlicht werden. Innerhalb des MAP-Servers werden die veröffentlichten Metadaten in Form eines Graphen verwaltet. Damit bietet sich die Möglichkeit, an zentraler Stelle eine Gesamtsicht auf den aktuellen Status eines

[1] http://www.trustedcomputinggroup.org/resources/tnc_ifmap_binding_for_soap_specification

Netzwerkes zu etablieren. Durch Korrelation der vorhandenen Metadaten können außerdem sicherheitsrelevante Informationen abgeleitet werden.

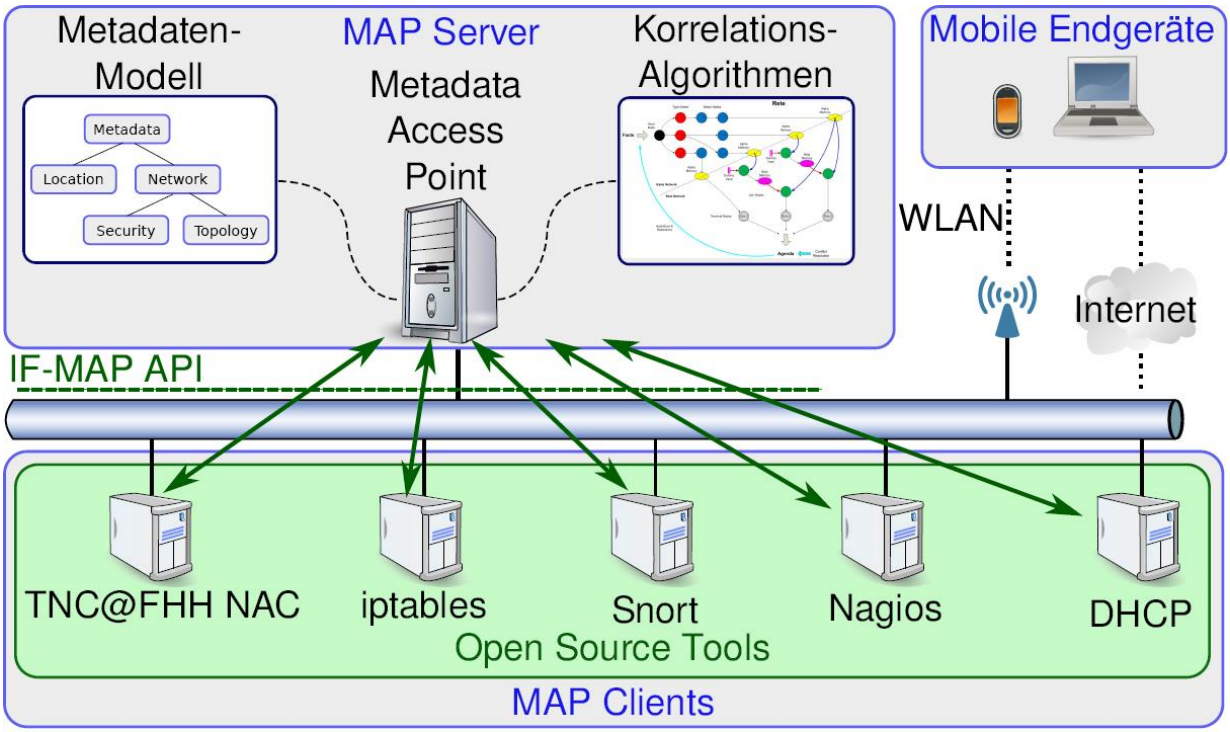


Abbildung 1: IF-MAP-Architektur

Systeme, die mit dem MAP-Server kommunizieren, werden als IF-MAP-Clients bezeichnet. Die Kommunikation basiert auf einem Publish-Search-Subscribe-Modell, bei dem sowohl synchron als auch asynchron Metadaten ausgetauscht werden können:

1. Neue Metadaten werden über die Publish-Operation veröffentlicht.
2. Nach vorhandenen Metadaten kann per Search-Operation gesucht werden.
3. Über die Subscribe-Operation können sich IF-MAP-Clients asynchron über Änderungen der im MAP-Server gespeicherten Metadaten informieren lassen. Dabei wird seitens des IF-MAP-Clients spezifiziert, welche Art von Metadatenänderungen relevant ist. Nur solche Änderungen haben eine Benachrichtigung durch den MAP-Server zur Folge.

Technologisch basiert IF-MAP auf einer Reihe von etablierten Standardtechnologien. Als Framework zur Übertragung der Metadaten kommt das SOAP-Protokoll in Kombination mit HTTP(S) zum Einsatz. Das Format der Metadaten ist durch XML-Schemata beschrieben. Auf diese Weise können etablierte Sicherheitssysteme, die um IF-MAP-Client-Funktionen erweitert worden sind, beliebige Metadaten über den aktuellen Status des Netzwerkes austauschen. In der Abbildung ist dies u.a. für die Systeme iptables, DHCP und Snort dargestellt. Auf diese Weise können ansonsten isolierte Daten (IDS Events, DHCP Lease Informationen) verknüpft werden.

Durch den Einsatz der IF-MAP-Architektur in dem geschilderten Szenario wäre es möglich, kompromittierte Endgeräte gezielt aufzuspüren und ggf. weitere Maßnahmen einzuleiten. Dies kann beispielsweise die Sperrung oder Isolierung dieser Geräte sein. Ein wichtiger Aspekt hierbei ist, dass durch den Einsatz der IF-MAP-Architektur ein kompromittiertes Endgerät auch nach den Authentifizierungs- und Autorisierungsvorgang

anhand seines Verhaltens erkannt und somit flexibel auf die jeweilige Bedrohungslage reagiert werden kann. [DDB11]

2.2 Mehrwert durch IF-MAP-Einsatz

Durch die zunehmende Professionalisierung und Komplexität von Schadsoftware, insbesondere in den Bereichen der Industrie- und Wirtschaftsspionage, wird es zunehmend schwerer, eventuell manipulierte Endgeräte bereits bei der Authentifizierung korrekt zu erkennen und zu isolieren, z.B. wenn die entsprechende Schadsoftware erst nach einen erfolgreichen Anmeldevorgang aktiv wird. Die Autoren solcher Schadsoftware bedienen sich dabei zunehmend ausgefeilter Mechanismen, welche eine Erkennung der Manipulation zusätzlich erschwert. Als Beispiel sei an dieser Stelle nochmal der eingangs erwähnte „Stux-Net“-Schädling zu nennen, der sich dabei unterschiedlicher Mechanismen bedient, welche zum Teil so ausgeklügelt sind, das selbst die Analyse des Schadsoftware mit einen sehr hohen Aufwand verbunden ist [HEIS10].

Nachfolgend soll Stichpunktartig ein möglicher Ablauf dargestellt werden, in welchem ein zunächst unentdecktes, manipuliertes Endgerät aufgrund seiner Aktivitäten aufgespürt und gesperrt bzw. isoliert wird. Hierbei soll vor allem der Mehrwert aufgezeigt werden, der durch den Einsatz eine IF-MAP-Architektur entsteht:

a. Das kompromittierte Endgerät erbittet Zugang zum Unternehmens-Netzwerk. Die auf dem Gerät befindliche Schadsoftware ist dabei noch inaktiv und wird während der Authentifizierung/Autorisierung und Überprüfung der Integrität des Endgerätes (beispielswiese durch den Einsatz von Trusted Network Connect, TNC) nicht korrekt erkannt. Das Gerät erhält demzufolge Zugang. Die entsprechenden Verbindungsdaten (MAC/IP-Adresse des Endgerätes, Zeitpunkt des Zugriffes, Nutzerrollen und Berechtigung, etc.) werden an den MAP-Server übertragen und in den entsprechenden Meta-Daten-Graphen eingetragen.
b. Nach einem gewissen Zeitraum aktiviert sich die Schadsoftware und beginnt damit ihre Tätigkeit aufzunehmen. Einige mögliche Beispiele für solche Aktivitäten wurden bereits im vorhergehenden Abschnitt geschildert.
c. Die eingeleiteten Aktivitäten des manipulierten Endgerätes werden von einem IF-MAP-Client (z.B. ein Intrusion-Detection-System wie Snort) bemerkt. Dieses veröffentlicht die entsprechende Meldung und Daten an den MAP-Server („Publish“).
d. Der MAP-Server nimmt die entsprechenden Daten entgegen und fügt diese zum jeweiligen Metadaten-Graphen hinzu. Anschließend werden die IF-MAP-Clients, welche sich für eine Benachrichtigung beim Ändern bestimmter Metadaten beim MAP-Server registriert haben („Subscribe“), von diesen Ereignissen in Kenntnis gesetzt und erhalten die entsprechenden Informationen.
e. Die benachrichtigten IF-MAP-Clients können daraufhin weitere Maßnahmen einleiten, welche wiederum von den jeweiligen Informationen abhängig sind. Beispiele hierfür sind unter anderem die Blockierung des Datenstroms durch eine Firewall, Sperrung des Zugriffes auf das Unternehmensnetz durch einen Policy Enforcement Point (PEP), z.B. in Form eines Switches oder eines VPN-Gateways, Isolierung des Endgerätes in eine Quarantänezone etc. Hierbei müssen die entsprechenden IF-MAP-Clients ihre Entscheidung nicht nur anhand der

eingehenden Informationen treffen, sondern können zusätzliche Daten vom MAP-Server abfragen („Search“) und zur Entscheidungsfindung hinzuziehen.

Nach diesen Vorgängen wird die unerlaubte Aktivität des Endgerätes unterbunden und das Endgerät wird vom Unternehmensnetz isoliert. Abschließend können auf Grundlage der gesammelten Informationen die unterbundenen Aktivitäten sowie deren Details protokolliert und entsprechende Meldungen an die verantwortlichen Systemadministratoren generiert werden. [DDB11]

2.3 Ableitung von Kernanforderungen

Basierend auf den fachlichen Szenarien sowie den Bedrohungsaspekten wurden Kernanforderungen im ESUKOM-Projekt abgeleitet, welche die Basis für die weiteren Arbeiten bildeten:

1. **Anomalie-Erkennung:** Normalverhalten und Grenzverhalten muss sich über das Sammeln möglichst viele Daten identifizieren lassen. Grenzüberschreitungen lassen sich dann durch Korrelation mit dem sonstigen Systemverhalten ins Verhältnis setzen.
2. **Smartphone Awareness:** Durch Smartphone-Awareness soll gewährleistet werden, dass Komponenten und Dienste innerhalb der IT-Infrastruktur eines Unternehmens erkennen können, ob es sich bei angebundenen Geräten um ein Smartphone handelt. Darüber hinaus soll ebenfalls ermittelt werden können, ob die angebundenen Smartphones hinsichtlich Verwendung und Softwarezustand den Vorgaben des Unternehmens genügen.
3. **Single-Sign-Off:** Darunter versteht man infrastrukturseitige Komponenten, welche die unabhängige Authentifizierung eines Nutzers gegenüber verschiedener Anwendungen und Diensten durch eine einmalige, zentrale Authentifizierung ersetzen. Dem Anwender bleibt es somit erspart, verschiedenste Daten zur Authentifizierung, wie Kombinationen aus Benutzernamen und Passwörtern oder Zertifikaten, vorzuhalten
4. **Secure Evidence:** Darunter versteht man die Erzeugung eines möglichst gerichtsfesten Beweises über einen Vorgang. Dazu ist es notwendig, den Zustand des Erzeugers des Beweises, im folgenden Evidence Generator (EG) genannt, sicher zu erfassen. Dafür kann IF-MAP als Publish-Subscribe-basiertes System zur Verbreitung von Metainformationen im Netzwerk eingesetzt werden.
5. **Identity Awareness:** Der Begriff bezeichnet die Fähigkeit von Netzwerkkomponenten, ihre Funktionsweise abhängig von der Identität des anfragenden Benutzers anzupassen. Das Ziel ist die Konfiguration von Komponenten flexibler gestalten zu können. Über IF-MAP soll auch eine identitätsbasierte Konfiguration unterstützt werden.
6. **Location-based Services:** Darunter versteht man Dienste und Anwendungen, die den aktuellen Aufenthaltsort des Benutzers für die Bereitstellung und Verarbeitung von Daten nutzen. Anhand des Aufenthaltsortes kann dann auch auf den regulären Zugriff geschlossen werden (z.B. wenn ein WLAN-Zugang außerhalb des Firmennetzes genutzt wird).
7. **Erkennen von MalApp-basierten Angriffen:** Im Rahmen der Bedrohungsanalyse haben sich Malicious Application (MalApps) als Ursache für viele der beschriebenen Bedrohungen herausgestellt. Im Rahmen des ESUKOM-

Projektes sollen deshalb MalApps erkannt und so den Bedrohungen entgegen gewirkt werden können.

8. **Real-time Enforcement:** Bei dieser Anwendung handelt es sich um die automatisierte Umsetzung von reaktiven Maßnahmen, die durch den Metadata Access Point (MAP) kommuniziert werden und möglicherweise auch mit Hilfe von IF-MAP-Anwendungen ausgelöst werden können.

Diese Kernanforderungen bestehen aus einer Menge von Anwendungsmöglichkeiten, deren technologische Grundlage der Austausch von Metadaten über das IF-MAP-Protokoll ist. Diese Anwendungen können (einzeln oder kombiniert) in verschiedenen Anwendungsfällen einen Mehrwert erbringen. Zudem ermöglichen sie es, den identifizierten Bedrohungen effektiv entgegenzuwirken. Durch diese Abstraktion der ESUKOM-Anwendungen lassen sich die erforderlichen Kern-Funktionen unabhängig von einer konkreten Fachlichkeit benennen. Auf diese Weise kann den universellen Einsatzmöglichkeiten des IF-MAP-Protokolls Rechnung getragen werden.

Die Beschreibung, der definierten ESUKOM-Anwendungen unterteilt sich in vier Abschnitte:

1. Als erstes wurde ein Überblick über die generelle Idee der ESUKOM-Anwendung im Projekt seitens der Partner gegeben.
2. Anschließend wurde dargestellt, welche Art von Metadaten für die Realisierung der Anwendung erforderlich ist. So konnte herausgearbeitet werden, welche der im Konsortium existierenden Sicherheitslösungen die entsprechenden Metadaten bereitstellen könnten und welche Metadaten darüber hinaus erzeugt werden müssten.
3. Weiterhin wurde exemplarisch dargestellt, wie die jeweilige ESUKOM-Anwendung in den fachlichen Szenarien eingebettet werden können. Es wurde insbesondere untersucht, auf welche Weise Metadaten gesammelt und verknüpft werden können und welche Rückschlüsse sich aus den so korrelierten Metadaten schließen lassen.
4. Abschließend wurde untersucht und skizziert, wie die dem Konsortium zur Verfügung stehenden Sicherheitslösungen in diesem Anwendungsfall Einsatz finden können und welche der Metadaten von diesen Tools erzeugt werden können.

Es soll dabei jedoch beachtet werden, dass es sich um eine repräsentative Teilmenge der möglichen Zielanwendungen von IF-MAP in ESUKOM handelt. Anhand eines generischen Szenarios wurden in ESUKOM dann die relevanten Kernanforderungen exemplarisch umgesetzt, wie in dem Demonstrator-Kapitel noch beschrieben wird.

3 Probleme bei der Umsetzung anhand der IF-MAP-Spezifikation

Während der Entwicklungsarbeiten im ESUKOM-Projekt traten bei der Implementierung des IF-MAP-Protokolls an einigen Stellen Probleme und Unklarheiten auf, welche vor allem aus der komplexen Interaktion zwischen den verschiedenen Komponenten resultierten. Sämtliche auftretende Probleme sowie die gewählten Lösungsansätze wurden in einem separaten Dokument festgehalten, welches auch der TCG vorgelegt wurde und im Anhang gefunden werden kann („Dokument IF-MAP-Issues vom 09.05.2012“). Der nachfolgende Abschnitt soll eine kurze Zusammenfassung der dort geschilderten Probleme und der zum Einsatz gekommenen Lösungen aufzeigen.

3.1 Multiple Device-Knoten für einen Endpunkt

Innerhalb der IF-MAP Spezifikation ist ein Beispiel aufgeführt, in welchem von einem Endpunkt eine Verbindung auf Layer 2 sowie auf Layer 3-Ebene hergestellt wird [IFM-ME12, Abschnitt 9.12]. Im Rahmen des ESUKOM-Projektes trat ein ähnlicher Fall auf, in welchem eine NAC-Lösung (macmon) und ein VPN-Server (NCP VPN Gateway) jeweils Metadaten für eine erkannte Verbindung auf beiden Ebenen (Layer 2 und Layer 3) innerhalb des Metadatengraphen veröffentlichen. Dadurch entstehen zwei voneinander unabhängige Access-Request-Knoten, welche jedoch beide mit den gleichen Device-Knoten verbunden werden müssen. Da sich beide IF-MAP-Clients auf den gleichen Device-Identifier beziehen müssen, kann es in diesem Fall zu einer Art „Race-Condition“ kommen: Während der erste IF-MAP-Client seine Daten (inkl. der Access-Request-Device-Metadaten) noch normal im Graphen veröffentlichen kann, muss der zweite IF-MAP-Client seine Metadaten zwischen seinen eigenen Access-Request-Knoten und den vorher angelegten Device-Knoten veröffentlichen. Da der dem Device-Knoten zugewiesene Name jedoch gewisse Anforderungen bzgl. der Einzigartigkeit genügen muss [IFM-SO12] und dementsprechend zu einen bestimmten Anteil zufällig generiert wird, bleibt die Frage, wie der zweite IF-MAP-Client diesen Knoten zuverlässig ermitteln kann, um seine eigenen Metadaten an der richtige Stelle zu veröffentlichen.

Zur Lösung dieses Problems wurde innerhalb des ESUKOM-Projektes der folgende Ansatz verfolgt: Der VPN-Server sowie die NAC-Lösung publizieren ihre Authenticated-As-Metadaten innerhalb der Verbindung zwischen ihren jeweiligen Access-Request- und Identity-Knoten. Dieser Identity-Knoten repräsentiert dabei den Hostnamen des Endgerätes, was durch das Setzen eines entsprechenden Other-Type-Definition Attributes festgelegt wird. Bevor jetzt einer der beiden IF-MAP-Clients seine Metadaten sendet wird zunächst eine Suchabfrage nach anderen Access-Request-Knoten durchgeführt, welche mit den gleichen Hostname-Identity-Knoten verbunden sind. Sollte dieser vorhanden sein, kann der IF-MAP-Client auf diese Weise feststellen, welcher Device-Knoten vom ersten IF-MAP-Client benutzt wurde.

3.2 Hersteller-spezifische Attribute für Standart-Metadaten

Ein weiteres Problem, welches innerhalb des Projektes auftrat, war die Verwendung von Hersteller-spezifischen Attributen, durch welche die IF-MAP-Standart-Metadaten erweitert werden können. Innerhalb des Projektes wurden diese Attribute für die Metadaten vom VPN-Server benutzt: Wenn ein mobiles Endgerät auf ein Netzwerk zugreift und sich dabei über den VPN-Server verbindet, werden von diesem die entsprechenden Metadaten über dieses Endgerät an den MAP-Server gesendet. Diese

Daten beinhalten dabei zwei unterschiedliche IP-Adress-Knoten, welche mittels Access-Request-IP-Links mit den entsprechenden Access-Request-Knoten verbunden sind. Da innerhalb der IF-MAP-Spezifikation nicht genau festgelegt wurde, wie eine Unterscheidung zwischen den beiden IP-Adress-Knoten vorgenommen werden kann (ISP- oder VPN-Adresse), welche jedoch für die Umsetzung einiger Anwendungsfälle von Nöten war, wurde zunächst beschlossen die Differenzierung anhand eines neuen Attributes zu gewährleisten. Da das Hinzufügen eines solchen Attributes auf XML-Ebene laut Schema-Definition kein Problem bei der Validierung des XML-Schemas darstellen sollte wurde sich zunächst auf diesen Lösungsansatz geeinigt.

Im Rahmen des Plug-Fests im Frühjahr 2012 kam es dabei jedoch aufgrund einer fehlgeschlagenen Schema-Validierung zu Problemen während eines Tests gegen einen der dort vertretenden MAP-Server. Obwohl die Hauptursache des Problems auf eine fehlende Schema-Validierungs-Datei zurückzuführen war, führte dieses Ereignis zu Diskussionen bzgl. der Frage, ob dieses Vorgehen nicht doch einen Bruch mit der IF-MAP-Spezifikation darstellt. Aufgrund der Aussagen innerhalb der Spezifikation („*IF-MAP standard metadata schema SHOULD NOT be extended directly with vendor-specic extensions*“) wurde vom ESUKOM-Konsortium beschlossen, einen anderen Ansatz zur Lösung dieses Problems zu benutzen: Anstatt die bestehenden Standart-Metadaten zu erweitern wurden Hersteller-spezifische Metadaten zu den jeweiligen IP-Adress-Knoten hinzugefügt. In der neusten Version der IF-MAP-Spezifikation wurde die Aussage bzgl. der Erweiterung von IF-MAP-Standard Metadaten auch klarer formuliert *(„clients are prohibited from extending the standard metadata schema directly with vendor-specific extensions“*) und zeigt, dass diese Entscheidung richtig war.

3.3 Such-Operationen ohne Wurzel-Knoten

Für eine Such- oder Subskriptions-Operation muss im Kontext von IF-MAP stets ein bestimmter Knoten als Ausganspunkt für die Abfrage angegeben werden. Dieses Verhalten macht Abfragen wie beispielsweise „Gib mir alle derzeitig existenten IP-MAC-Metadaten“ unmöglich. Diese Tatsache kann ein Problem darstellen, wenn ein IF-MAP-Client darauf angewiesen ist eine große Anzahl (oder alle) Metadaten innerhalb des MAP-Servers abzufragen, wie es beispielsweise bei der Visualisierungskomponente „irongui“ der Fall ist. Um ein solches Verhalten zu gewährleisten wurde eine sogenannte „dump“-Operation innerhalb des „irond“ MAP-Servers implementiert, welche es einen IF-MAP-Client ermöglicht, sämtliche Metadaten abzufragen. Allerdings ist diese Operation nicht innerhalb der IF-MAP-Spezifikation definiert und stellt somit einen offensichtlichen Bruch mit der Spezifikation dar.

Wie sich später herausstellte, werden in der kommenden Version der IF-MAP-Spezifikation (Revision 2.1) sogenannte „Custom-Identifiers“ eingeführt, welche auch komplexere Abfragen erlauben und mit deren Hilfe der gleiche Mechanismus bereitgestellt wird. Somit kann die benötigte Funktion realisiert werden ohne dabei gegen die Spezifikation zu verstoßen.

3.4 Fehlen gerichteter Kanten

Das IF-MAP-Datenmodell basiert auf einem ungerichteten Graphen. Allerdings könnte sich unter bestimmten Umständen die Verwendung von gerichteten Kanten innerhalb des Graphen als sehr nützlich erweisen: Während bei manchen Standart-Metadaten-Typen sich die Verbindungs-Richtung anhand ihres Namens ableiten lassen (z.B. Authenticated-By oder Authenticated-As), existieren einige Anwendungsfälle, in denen

dieses nicht möglich ist. Innerhalb des ESUKOM-Projektes wurden zu diesem Zweck eigene Knoten und Hersteller-spezifische Metadaten definiert, welche die Abbildung einer hierarchischen Struktur innerhalb des Metadatengraphen anhand ihres Namens ermöglichen. Ein anderer Ansatz wäre die Einführung von Referenzen innerhalb der Metadaten-Kanten. Diese könnten als Zeiger auf die verbundenen Knoten dienen und somit die Richtung angeben, in Welcher zwei Knoten miteinander in Beziehung stehen.

Somit lässt sich abschließend sagen, das die Möglichkeit, die Richtung einer Beziehung anzugeben, zwar für einige Anwendungsfälle nötig werden kann, sich allerdings auch leicht durch die genannten Ansätze selbst realisieren lässt ohne dabei einen Bruch mit der Spezifikation zu verursachen.

3.5 Fazit

Das IF-MAP-Protokoll stellte eine ideale Basis für die Erfüllung der Anforderungen des ESUKOM-Projektes dar. Eine Integration verschiedener Netzkomponenten hätte man zwar grundlegend auch über alternative Ansätze realisieren können. Denkbar wäre zum Beispiel die Nutzung eines zentralen Dienstes zum Sammeln von Log-Meldungen in einem wohldefinierten Format gewesen. Die Kombination aus der Menge der unterstützten Funktionen, der Erweiterbarkeit und der durch die offene Spezifikation gegebenen Interoperabilität ist bei alternativen Ansätzen in dieser Form allerdings nicht vorhanden.

Zudem wird IF-MAP von der TCG stark vorangetrieben und ist inzwischen auch bei den Netzwerkherstellern wie Juniper Networks, Enterasys Networks und Cisco Systems angekommen, wodurch eine weite Verbreitung zukünftig wahrscheinlich ist.

4 Beschreibung des Demonstrators

Der Demonstrator für das Testen der einzelnen Anwendungsfälle wird in zwei unterschiedlichen Varianten zur Verfügung gestellt:

a. In Form einer VSA (Virtual Security Appliance)
b. Als Zip-Archiv, welches die einzelnen virtuellen Maschinen für die jeweiligen Komponenten enthält

Für die Umsetzung als VSA wurde dabei auf die Erkenntnisse des ebenfalls vom Bundesministerium für Bildung und Forschung (BMBF) geförderten VISA-Projekts als Basis zurückgegriffen. Dieses beschäftigt sich unter anderem mit der Entwicklung eines Frameworks für den vereinfachten und passgenauen Einsatz von Sicherheitsanwendung auf Basis von Virtual Security Appliances (VSA). Ziel des Projektes ist es, durch die Nutzung von Virtualisierungstechnologien das Management von IT-Infrastrukturen, insbesondere der Sicherheitskomponenten, zu erleichtern und zu unterstützen. Ebenso sollte eine Möglichkeit geschaffen werden, ein automatisiertes Aufsetzten der einzelnen Komponenten auf Basis einer VSA zu ermöglichen. Weitere Informationen zum VISA-Projekt können auf der VISA-Projektwebseite http://www.visa-project.de gefunden werden.

Neben der technischen Umsetzung der einzelnen Komponenten des Demonstrators in einer der eingangs genannten Variationen wurde zusätzlich begleitendes Material in Form von Videos entwickelt und zur Verfügung gestellt, um die konkreten Vorgänge und Abläufe anschaulich darzustellen. Beides kann auf der Projekt-Webseite des ESUKOM-Projektes http://www.esukom.de gefunden werden.

Der Demonstrator wurde auf Basis des Anwendungsszenarios aus [ESAP312] entwickelt. Innerhalb des Szenarios wurde eine fiktive IT-Infrastruktur eines Krankenhauses beschrieben, welche in zwei Netzsegmente unterteilt ist: ein unsicheres, externes Netz, in welchem Dienste laufen die von außen erreichbar sind (DMZ), sowie ein sicheres internes Netz. Innerhalb des internen Netzes befindet sich ein Server mit sensiblen Patientendaten. Für den Zugriff auf diese Daten gelten die folgenden Sicherheitsrichtlinien:

1. Zugriffe aus dem externen Netz dürfen nur durchgeführt werden, wenn der entsprechende Client sich erfolgreich am VPN-Gateway authentifiziert hat. Darüber hinaus muss der Client bestimmte Privilegien besitzen, welche ihn für den Zugriff auf die Daten autorisieren.
2. Zugriffe von mobilen Clients aus dem internen Netz erfolgen über einen der beiden Access Points (Cafeteria & Patient Rooms) und dürfen nur an den Patientendaten-Dienst weitergeleitet werden, wenn der Client einer bestimmten Nutzer- oder Gerätegruppe zugeordnet (mobiles Gerät eines Arztes) und über den Access Point innerhalb der Patienten-Räume mit den Netzwerk verbunden ist.
3. Sollte auf dem mobilen Endgerät eine potentiell schädliche Anwendung installiert sein, muss dieses Gerät aus dem internen Netz ausgesperrt werden.
4. Wenn ein (autorisiertes) mobiles Endgerät auf den Patienten-Daten-Dienst zugreift, darf das Verhalten nicht von dem „Normal-Verhalten“ des Nutzers bzw.

des Endgerätes abweichen. Sollte es Abweichungen von diesem geben, so wird dieses als Anomalie erkannt und das Endgerät wird gesperrt. Als Anomalie gilt beispielsweise das häufige und wahllose Zugreifen auf die Patientendaten innerhalb eines bestimmten Zeitfensters.

Die Umsetzung dieser Richtlinien wurde dabei anhand von vier unterschiedlichen Anwendungsfällen auf ihre Umsetzbarkeit hin überprüft. Die verschiedenen Anwendungsfälle dienen dabei dem Test der Implementierung verschiedener Kernanforderungen, welche für eine beispielhafte Umsetzung aus einer Menge von Anforderungen als die wichtigsten identifiziert und ausgewählt wurden. Alle erarbeiteten Kernanforderungen sind in Kapitel 2.3 aufgeführt. Im Rahmen der Anwendungsfälle wurden die folgenden Anforderungen umgesetzt:

- Real-Time-Enforcement (Anwendungsfall 1, 2, 3, 4)
- Smartphone-Awareness (Anwendungsfall 2)
- Location-Based Services (Anwendungsfall 2)
- MalApp-Detection (Anwendungsfall 3)
- Anomalie-Erkennung (Anwendungsfall 4)

Die einzelnen, detaillierten Anwendungsfälle sowie deren Umsetzung werden in Kapitel 5 noch genauer beschrieben.

4.1 Aufbau der virtuellen Umgebung

Der Aufbau der virtuellen Umgebung baut auf der Netzwerkstruktur des generischen Krankenhaus-Szenarios auf, welches in [ESAP312] definiert wurde. Abbildung 2 zeigt den Aufbau der (virtuellen) Infrastruktur auf.

Im Gegensatz zur in [ESAP312] gezeigten Infrastruktur wurden für die virtuelle Umgebung einige kleinere Änderungen und Optimierungen durchgeführt:

- Die iptables-Firewall zwischen den beiden Netzwerksegmenten wurde für die Umsetzung der Anwendungsfälle nicht mehr benötigt, für die Vermittlung zwischen den beiden Netzen kommt das NCP VPN-Gateway zum Einsatz.
- Der abgebildete DHCP-Server im internen Netz kommt nicht zur Anwendung. Stattdessen wird der DHCP-Dienst des Access Points für mobile Clients eingesetzt.
- Der Bürodaten-Server wird für die Umsetzung der Anwendungsfälle nicht benötigt und wurde nicht innerhalb des Prototyps umgesetzt.
- Die Detection Engine und der MAP-Server laufen auf innerhalb einer (virtuellen) Maschine.
- Der FreeRADIUS-Server kommt nicht zur Anwendung, da dieser für das Szenario keine Relevanz hat und dessen benötigte Funktionen bereits durch macmon abgedeckt werden.

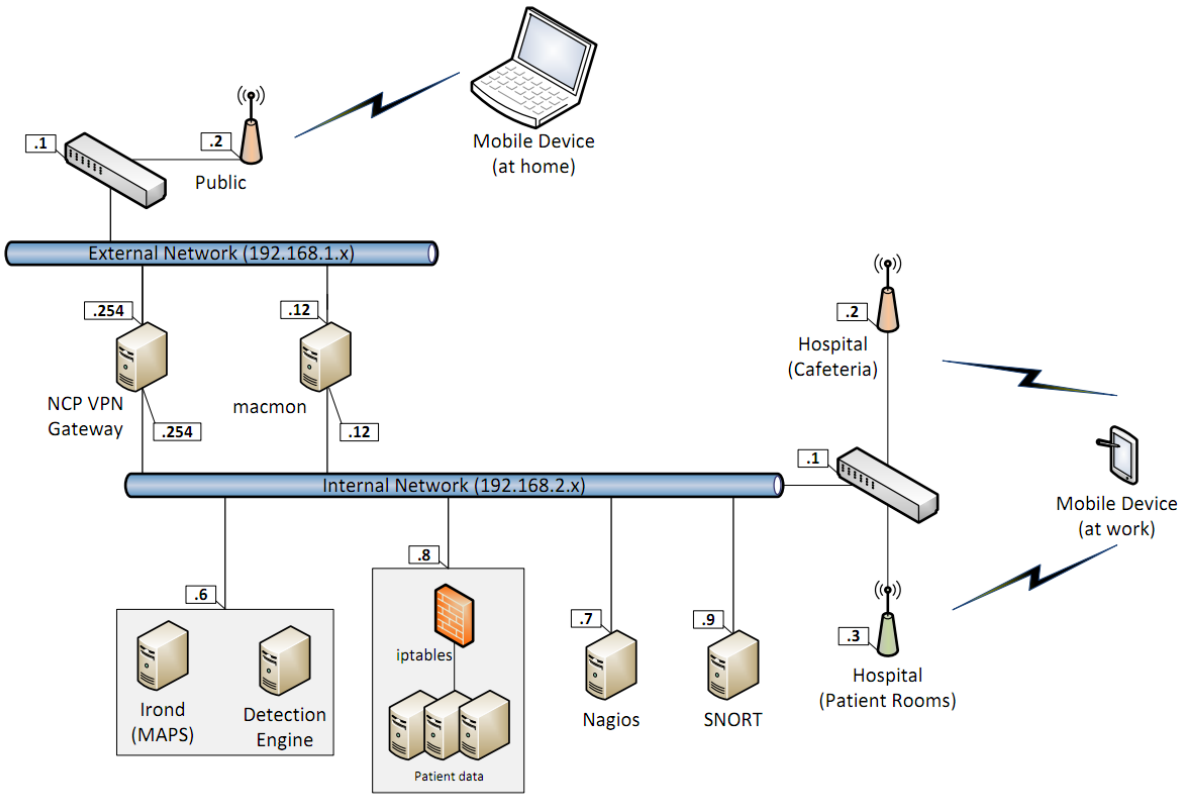


Abbildung 2: Infrastruktur des Krankenhausszenarios

Auf den in Abbildung 2 aufgeführten virtuellen Maschinen kommen die folgenden Komponenten zum Einsatz:

1. Virtuelle Maschine für das VPN-Gateway, zu welchem beliebige VPN-Clients (z.B. für Android) eine sichere Verbindung aufbauen können. Sollte eine solche Verbindung zustande kommen, werden von der IF-MAP-Erweiterung des NCP VPN-Servers verschiedene Metadaten bzgl. der Verbindung an den MAP-Server gesendet und der Client in das interne Netz weitergeleitet. Ebenso wird durch die virtuelle Maschine ein DHCP-Dienst für dieses Netzwerksegment bereitgestellt.
2. Virtuelle Maschine für die NAC-Lösung macmon, welche in beiden Netzsegmenten Zugriffe durch sich verbindende Clients überwacht und entsprechende Metadaten über diese innerhalb des Metadatengraphen publiziert.
3. Virtuelle Maschine für den zentralen MAP-Server. Zusätzlich befindet sich auf dieser Maschine die Detection Engine, welche für die Konsolidierung der Metadaten innerhalb des Graphen zum Einsatz kommt.
4. Datenserver für die (sensiblen) Patientendaten. Für den Zugriff auf die Patientendaten steht ein prototypischer Patienten-Dienst („Hospital-Mockup-Service“) zur Verfügung, auf welchen mittels Browser zugegriffen werden kann. Sobald ein Zugriff erfolgt, wird durch diesen Dienst ein IF-MAP-Event generiert und an den MAP-Server publiziert. Der Patientendienst ist zusätzlich durch eine iptables-Firewall geschützt, welche den Zugriff auf den Dienst für die anfragenden Clients steuert. Die zugehörige Instanz des iptables-IF-MAP-Clients auf dieser Maschine führt dabei die Regeln für die iptables-Firewall aus.
5. Die Nagios-Maschine dient zur Überwachung der einzelnen Systeme innerhalb des internen Netzes. Sollte eine dieser Maschinen nicht länger erreichbar sein,

wird dieses durch Nagios erkannt und mit Hilfe des auf der Maschine laufenden Nagios-Client als Event an den MAP-Server gesendet.

6. Virtuelle Maschine, auf welcher das Intrusion Detection System Snort und der zugehörige Snort-IF-MAP-Client läuft. Diese Maschine scannt das Netzwerk auf sicherheitskritische Ereignisse und publiziert diese an den MAP-Server.

Die nachfolgende Tabelle zeigt nochmal eine Übersicht über oben aufgeführten virtuellen Maschinen:

Nr.:	VM-Name:	IP-Adresse:	Einsatzzweck:	Login-Daten
1	NCP VPN Gateway	eth0: 192.168.1.254 eth1: 192.168.2.254	Zugriffsteuerung, Weiterleitung extern -> intern	root/esukom
2	macmon	eth0: 192.168.1.12 eth1: 192.168.2.12	NAC-Lösung	root/macmon
3	irond, irondetect, irongui	eth2: 192.168.2.6	MAP-Server, Korrelation der Metadaten	tncuser/esukom123
4	Patient data, iptables	eth0: 192.168.2.8	Patienten-Daten Dienst, Zugriffskontrolle	root/decoit
5	nagios	eth0: 192.168.2.7	Monitoringn	root/decoit
6	snort	eth0: 192.168.2.9	Intrusion-Detection	root/decoit

Tabelle 1: virtuelle Maschinen des Demonstrators

4.2 Aufsetzen der virtuellen Umgebung

Im folgenden Abschnitt wird das Aufsetzen der virtuellen Umgebung des Demonstrators anhand der einzelnen virtuellen Maschinen mittels VirtualBox beschrieben. Dafür sind zunächst die folgenden Komponenten erforderlich:

- Drei WLAN Access Points
- Zwei SNMP-fähige Switche
- Oracle VirtualBox zum Einbinden der virtuellen Maschinen
- Ein Smartphone, auf welchem Android 4.0 oder höher installiert ist

Bei einer Benutzung des Demonstrators in der Variante als VSA (siehe auch Kapitel 4) müssen die meisten der im folgendem aufgeführten Konfigurationsschritte nicht mehr durchgeführt werden.

4.2.1 Importieren der virtuellen Maschinen

Zunächst müssen alle virtuellen Maschinen in Virtual Box importiert werden. Hierzu wählt man in VirtualBox im Menü „Maschine“ den Punkt „Hinzufügen“ aus und wählt die entsprechende .vbox-Datei aus (Abbildung 3).

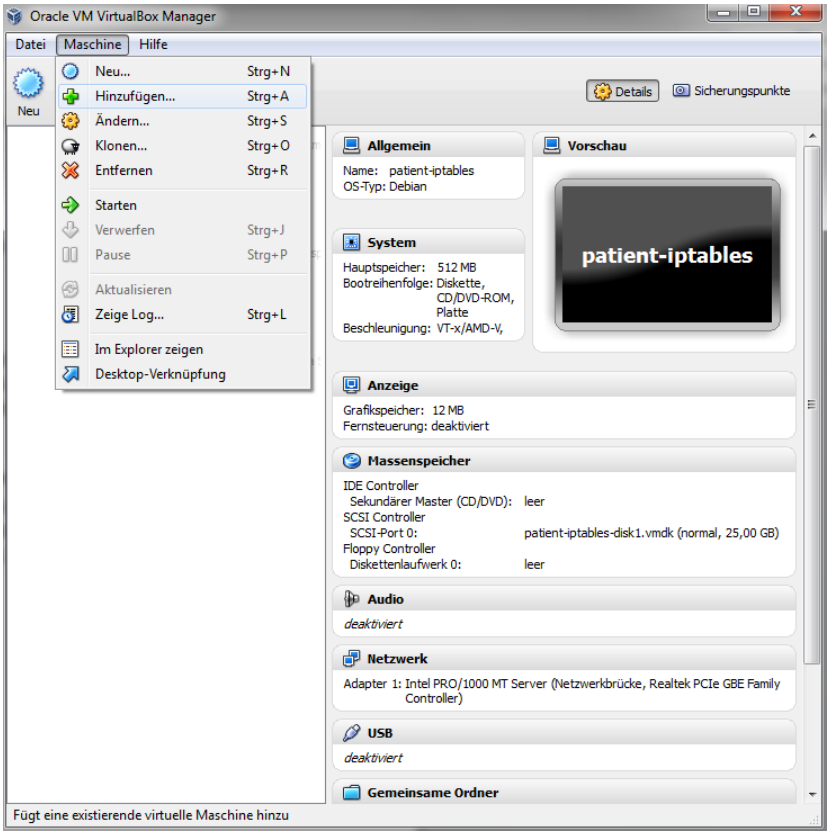


Abbildung 3: Import der virtuellen Maschinen in VirtualBox

Nach dem Hinzufügen aller benötigten virtuellen Maschinen müssen noch deren Netzwerkschnittstellen konfiguriert werden (Abbildung 4).

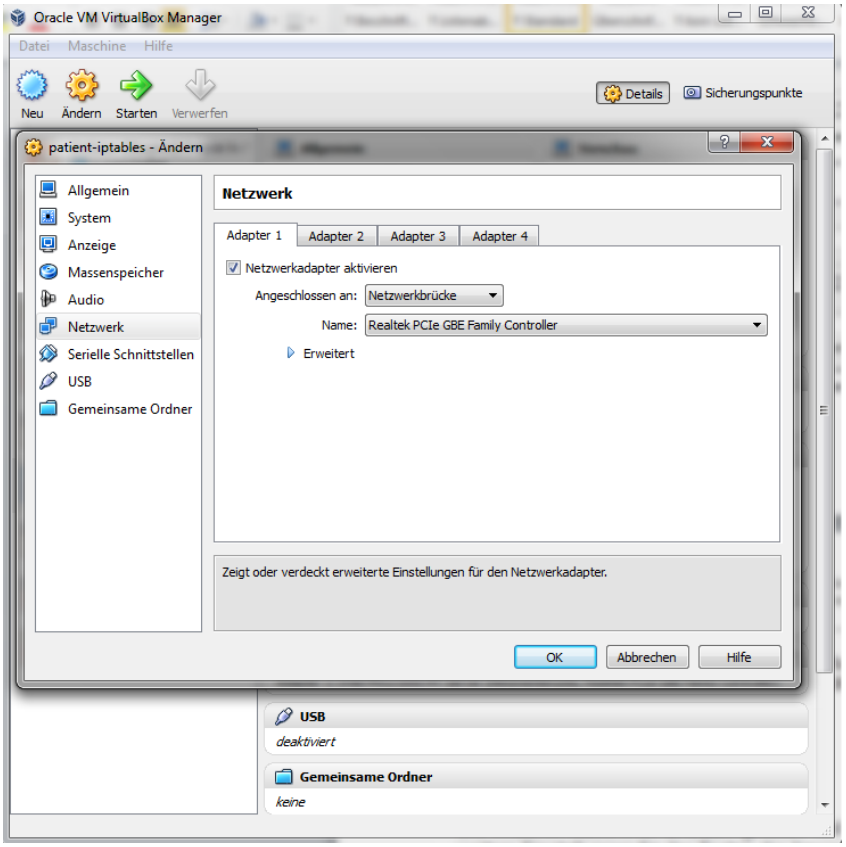


Abbildung 4: Konfiguration der Netzwerkschnittstellen

4.2.2 Konfiguration der Komponenten

Die virtuellen Maschinen sind zum größten Teil bereits vorkonfiguriert und können ohne weitere Einstellungen für das Testen der Anwendungsfälle eingesetzt werden. Trotzdem soll in diesem Abschnitt nochmal kurz auf die wichtigsten Punkte der Konfiguration eingegangen werden, auch um für die am häufigsten auftretenden Probleme bei der Inbetriebnahme Lösungswege aufzuzeigen. Bei weiterführenden Fragen sei an dieser Stelle auf die jeweiligen README-Dateien der einzelnen Komponenten hingewiesen, welche in der Regel die wichtigsten Konfigurations-Schritte genauer beschreiben. Diese können im Download-Archiv des Demonstrators oder der einzelnen Komponenten auf der Projektwebseite (www.esukom.de) gefunden werden.

Autorisierung der IF-MAP-Clients durch den MAP-Server:

Insgesamt stehen 2 Authentifizierungsverfahren für IF-MAP-Clients innerhalb des MAP-Servers „irond“ zur Verfügung: Einmal die Basic-Authentication, welche auf einer Benutzername/Passwort Kombination im Klartext aufbaut, sowie eine zertifikatsbasierte Authentifizierung. Für das Aufsetzen des Demonstrators genügt die erste Methode. Alle Benutzername-Passwort-Kombinationen finden sich innerhalb der Datei `/irond/basicauthusers.properties`. Hier können bei Bedarf auch neue Credentials hinzugefügt werden. Wenn neue Clients hinzugefügt werden ist auch bei der Basic-Authentication ein Zertifikat auf beiden Seiten erforderlich.

Die MAP-Server Installation, welche auf der vorkonfigurierten Maschine zu finden ist, verfügt bereits über alle notwendigen Zertifikate für die einzelnen Komponenten des Demonstrators. Sollten neue IF-MAP-Clients hinzukommen, müssen die entsprechenden Zertifikate erstellt und in die jeweiligen Keystores eingefügt werden.

Konfiguration der Detection Engine:

Die Detection Engine kann über die Datei `/irondetect/configuration.properties` konfiguriert werden. Neben mehreren allgemeinen Einstellungen kann hier die Policy festgelegt werden, mit welcher die Detection-Engine arbeitet (zu finden unter der Eigenschaft `irondetect.policy.filename`). Innerhalb des Demonstrators sind insgesamt drei dieser Policys von Bedeutung:

1) demonstrator_af2.pol → 2.Anwendungsfall, Location-Based-Services
2) demonstrator_af3.pol → 3.Anwendungsfall, MalApp-Detection
3) demonstrator_af4.pol → 4.Anwendungsfall, Anomalie-Erkennung

Je nach zu testenden Anwendungsfall muss die entsprechende Policy für diesen aktiviert sein. Hierzu können die nicht zum Einsatz kommenden Policys auskommentiert werden, wie in Abbildung 5 zu sehen ist.

```
# Filename of policy
#irondetect.policy.filename = /policy/oberseminar.pol
#irondetect.policy.filename = ./oberseminar.pol
#irondetect.policy.filename = ./demonstrator_af1.pol
#irondetect.policy.filename = ./demonstrator_af2.pol
irondetect.policy.filename = ./demonstrator_af3.pol
#irondetect.policy.filename = ./demonstrator_af4.pol
```

Abbildung 5: Auszug aus irondetect-Konfiguration

Eine Ausnahme bildet dabei der erste Anwendungsfall, bei welchem die Detection Engine nicht benötigt wird und dementsprechend auch keine Änderungen in der Konfiguration vorgenommen werden müssen.

Konfiguration des Android-IF-MAP-Clients:

Die Konfiguration des Android-IF-MAP-Clients kann innerhalb der Anwendung im Setup-Reiter durchgeführt werden. Die folgenden Einstellungen werden für den Betrieb mit dem Demonstrator empfohlen:

- enable auto-start → deaktiviert
- enable auto-connect → deaktiviert
- enable auto-update → deaktiviert
- don’t send app informations → deaktiviert
- enable Location-Tracking → deaktiviert
- use Esukom-Metadata → aktiviert
- don’t send Google-Apps Informations → aktiviert
- Basic Authentication → aktiviert
- Connection Type → aktiviert
- allow unsafe SSL → aktiviert
- Username → android
- Password → android
- Server IP-Address → 192.168.2.6
- Server Port-Number → 8443

Die nicht aufgeführten Optionen können entweder auf ihren Standarteinstellungen belassen oder frei gewählt werden.

Skript zum Publizieren von Standart-Metadaten:

Als Hilfestellung für die letzten beiden Anwendungsfälle steht ein Skript zur Verfügung, welches dabei hilft, den Testaufwand zu reduzieren. Mit Hilfe dieses Skriptes wird ein Satz von Standart-Metadaten an den MAP-Server gesendet. Diese Daten entsprechen den Informationen, welche ansonsten vom NCP-Gateway und von macmon publiziert worden wären (Abbildung 5).

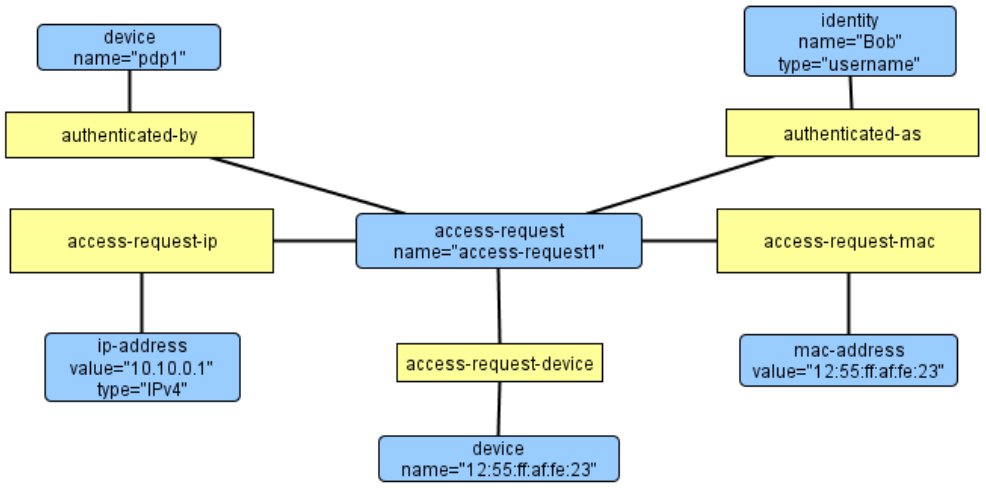


Abbildung 6: Standart-Metadaten

Das Hilfsskript befindet sich im gleichen Verzeichnis wie „irond“ und kann mittels `./tc-ifmap-Skripts.sh -a` aufgerufen werden. Vorher muss jedoch zunächst eine Änderung innerhalb des Skripts durchgeführt werden: Die Device-Eigenschaft muss den Wert der MAC-Adresse des Android-Smartphones zugewiesen bekommen, wie in Abbildung 7 zu sehen ist.

```
#!/bin/bash

INIT_MAPS=0
SIG_BEHAVES_LIKE_SMARTPHONE=0
SIG_MALAPP_INSTALLED=0
SIG_OPEN_PORTS=0
SIG_REQUEST_SERVICE_ACCESS=0
PUBLISH_SNORT_FEATURES=0
DEVICE=a0:0b:ba:cb:ef:cf
```

Abbildung 7: Skript zum Publizieren der Standart-Metadaten (Auszug)

Patientendaten-Dienst:

Der Patienten-Daten-Dienst befindet sich im Verzeichnis `/usr/esukom/patient-service/` der entsprechenden virtuellen Maschine. Der Dienst ist dabei ein Prototyp, welcher keine richtigen Daten bereitstellt, sondern lediglich Zugriffe erkennt und entsprechende Metadaten an den MAP-Server sendet. Der Zugriff auf den Patientendienst erfolgt mittels HTTP (entweder aus einem Webbrowser heraus oder mit Hilfe von Tools wie z.B. curl), die URL für den Aufruf ist folgendermaßen aufgebaut:

http://192.168.2.8:8888/patient/47711?device=aa:bb:cc:dd:ee:ff

Wichtig für diesen Anwendungsfall ist hierbei der letzte Parameter der URL (device). Über diesen Parameter wird angegeben, welches Gerät auf diesen Dienst zugegriffen hat. Der Wert selbst spiegelt den Wert des Device-Identifiers des Endgerätes innerhalb des Metadatengraphen wider, also die MAC-Adresse des Gerätes. Das Starten des Patientendienstes erfolgt durch den Aufruf `java -jar hospital-mock-0.0.1.jar`.

Zugriff auf Webinterfaces zur Konfiguration:

Einige der eingesetzten Komponenten innerhalb des Demonstrators bieten die Möglichkeit, über Webinterfaces konfiguriert zu werden oder über diese Informationen über deren aktuellen Status anzuzeigen. Die nachfolgende Tabelle zeigt auf, über welche URL diese erreichbar sind sowie die Credentials, welche für den Zugriff benötigt werden.

Komponente:	URL:	Login/Passwort:
NCP-VPN-Server	https://192.168.1.254:20112	Administrator / esukom
Nagios	https://192.168.2.4/nagios3	nagiosadmin / nagios
Snort/Acidbase	http://192.168.2.2/acidbase/	admin / admin

Tabelle 2: Erreichbare IP-Adressen der eingesetzten Komponenten

4.3 Testen der Anwendungsfälle

Nach dem die virtuelle Umgebung aufgesetzt und alle relevanten Einstellungen vorgenommen wurden, können die einzelnen Anwendungsfälle anhand des Demonstrators getestet werden. Die dafür erforderlichen Einzelschritte sollen in diesem Abschnitt kurz erläutert werden.

4.3.1 Anwendungsfall 1

Bestandteil des ersten Anwendungsfalls ist das Key-Feature „Realtime Enforcement“. Konkret geht es darum, das sich ein Gerät über den Access Point „Public“ mit den externen Netz verbindet, über das VPN-Gateway eine Verbindung in das interne Netz aufbaut und auf den Patientendaten-Dienst zugreift. Der Zugriff auf diesen Dienst wird dabei vom iptables-IF-MAP-Client gesteuert, welcher prüft ob innerhalb des Metadaten-Graphen die entsprechenden Capability-Metadaten (welche vom NCP VPN-IF-MAP-Client publiziert werden) existieren. Sollte dies der Fall sein wird der Zugriff gestattet, andernfalls wird das Endgerät gesperrt. Folgende Komponenten sind zur Durchführung dieses Anwendungsfalls erforderlich:

- MAP-Server (irond)
- NCP VPN-Gateway mit IF-MAP-Erweiterung
- Android-Smartphone mit installierten NCP VPN-Client
- Patient-Data-Service
- iptables-IF-MAP-Client (Allowance-Komponente)

Folgende Schritte sind zur Durchführung dieses Anwendungsfalls erforderlich:

1. MAP-Server starten

2. Virtuelle Maschine für das NCP VPN-Gateway starten. Nach einer kurzen Zeit wird vom VPN-IF-MAP-Client automatisch eine Verbindung mit den MAP-Server hergestellt

3. Auf der Patienten-Dienst-Maschine den iptables-IF-MAP-Client und den Patientendienst starten (siehe Abschnitt 4.2.2)

4. Mit den Smartphone eine Verbindung mit dem externen Netz über den Access Point „Public“ aufbauen

5. Über den NCP VPN-Client für Android eine Verbindung mit dem internen Netz aufbauen. Folgende Profile sind bereits vorhanden und können zum Verbindungsaufbau benutzt werden:
   - IPsec: ipsec1/ipsec1 – ipsec5/ipsec5
   - L2TP: l2tp1/l2tp1 – l2tp5/l2tp5

6. Nachdem die Verbindung erfolgreich zustande gekommen ist, werden die Metadaten der Verbindung durch den VPN-IF-MAP-Client veröffentlicht.

7. Mit den Webbrowser des Smartphones eine Verbindung zum Patientendienst aufbauen (siehe Abschnitt 4.2.2)

8. Sobald die Verbindung durch den iptables-IF-MAP-Client erkannt wurde, wird von diesem eine Suchabfrage auf die für den Zugriff benötigten Privilegien durchgeführt (die durch den VPN-IF-MAP-Client publizierten Capability-Metadaten mit den Wert „Arzt“). Sollten diese vorhanden sein, wird der Zugriff durch das automatische Einfügen einer iptables-Regel durch den iptables-IF-MAP-Client erlaubt.

4.3.2 Anwendungsfall 2

Bestandteil des zweiten Anwendungsfalls ist der Zugriff auf sensible Daten in Abhängigkeit des aktuellen Standorts des Benutzers bzw. des mobilen Endgerätes (Kernanforderung „Location Based Services“). Konkret geht es darum, dass der Zugriff auf die sensiblen Daten nur gestattet wird, wenn dieser von einem bestimmten WLAN-Access-Point aus erfolgt (Access-Point innerhalb der Patientenräume). Sollte der Zugriff von einem anderen Access-Point aus erfolgen, wird dieses von der Detection Engine als Regelbruch erkannt und ein Enforcement des jeweiligen Endgerätes eingeleitet. Die folgenden Komponenten sind zur Umsetzung dieses Anwendungsfalls erforderlich:

- MAP-Server (irond)
- Detection Engine (irondetect)
- Android-Smartphone mit installierten Android-IF-MAP-Client
- macmon mit IF-MAP-Erweiterung
- Patient-Data-Service
- iptables-IF-MAP-Client (Enforcement-Komponente)

Folgende Schritte sind zum Testen des Anwendungsfalls erforderlich:

1. MAP-Server starten

2. Standart-Metadaten per Skript veröffentlichen

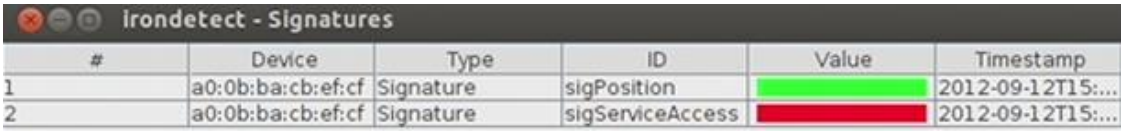
3. Detection Engine mit Hilfe des Start-Skripts ausführen. Hierbei ist darauf zu achten, dass die für den Anwendungsfall richtige Policy aktiviert ist (siehe Abschnitt 4.2.2). Nach dem Starten öffnet sich die grafische Benutzeroberfläche, in welche später die erkannten Ereignisse angezeigt werden.

4. Patienten-Dienst starten

5. Enforcement-Komponente starten

6. Mit den Smartphone eine Verbindung mit den Access-Point „Cafetaria“ herstellen.

7. Den Android-IF-MAP-Client starten, eine neue Session mit den MAP-Server beginnen und Gerätedaten publizieren. Die Detection Engine sollte nun anzeigen, das die zu überprüfende Signatur „Position“ bereits positiv getestet wurde (sprich: das mobile Endgerät über den Access-Point „Cafetaria“ mit den internen Netz verbunden ist). Abbildung 8 zeigt einen entsprechenden Screenshot.

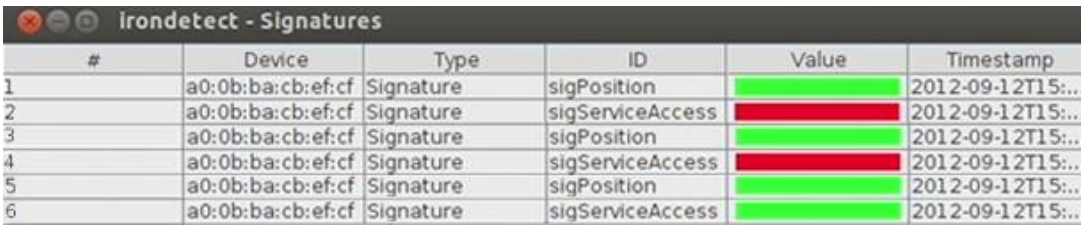


#	Device	Type	ID	Value	Timestamp
1	a0:0b:ba:cb:ef:cf	Signature	sigPosition		2012-09-12T15:...
2	a0:0b:ba:cb:ef:cf	Signature	sigServiceAccess		2012-09-12T15:...

Abbildung 8: Positive Erkennung der Positions-Signatur durch die Detection Engine

8. Anschließend muss mit den Smartphone auf den Patienten-Dienst zugegriffen werden. Für den Zugriff auf den Dienst wird am besten ein Browser verwendet, die aufzurufende URL ist im Abschnitt 4.2.2 zu finden.

9. Die Detection Engine erkennt jetzt zusätzlich, dass ein Zugriff auf den Patienten-Dienst erfolgte (siehe Abbildung 9). Durch die Erfüllung dieser und der in Punkt 6 geschilderten Signatur wird dadurch durch die Detection Engine ein Alert-Event an den MAP-Server veröffentlicht. Die iptables-Enforcement-Komponente wird per Subskription über dieses Event benachrichtigt und führt ein Enforcement des entsprechenden Endgerätes durch.



#	Device	Type	ID	Value	Timestamp
1	a0:0b:ba:cb:ef:cf	Signature	sigPosition		2012-09-12T15:...
2	a0:0b:ba:cb:ef:cf	Signature	sigServiceAccess		2012-09-12T15:...
3	a0:0b:ba:cb:ef:cf	Signature	sigPosition		2012-09-12T15:...
4	a0:0b:ba:cb:ef:cf	Signature	sigServiceAccess		2012-09-12T15:...
5	a0:0b:ba:cb:ef:cf	Signature	sigPosition		2012-09-12T15:...
6	a0:0b:ba:cb:ef:cf	Signature	sigServiceAccess		2012-09-12T15:...

Abbildung 9: Positive Erkennung beider Signaturen durch die Detection Engine

4.3.3 Anwendungsfall 3

Der dritte Anwendungsfall behandelt die Erkennung von potentiell schadhafter Software auf einem mobilen Endgerät. Folgende Komponenten sind für den Test des Anwendungsfalls mindestens erforderlich:

- MAP-Server (irond)
- Detection Engine (irondetect)
- Android-Smartphone mit installierten Android-IF-MAP-Client
- Skript zum Publizieren der Standart-Metadaten
- Iptables-IF-MAP-Client (Allowance-Komponente)

Mit Hilfe dieser Komponenten kann der dritte Anwendungsfall komplett getestet werden. Die im folgendem aufgeführten Einzelschritte sind hierfür notwendig:

1. MAP-Server starten

2. Standart-Metadaten per Skript veröffentlichen

3. Detection Engine starten, dabei auf die Aktivierung der richtigen Policy achten (siehe Abschnitt 4.2.2)

4. Enforcement-Komponente starten

5. Mit den Smartphone eine Verbindung zum internen Netz über einen der beiden Access Points herstellen, den Android-IF-MAP-Client starten (Einstellungen beachten, siehe Abschnitt 4.2.2), eine neue Session mit den MAP-Server beginnen und die Gerätedaten publizieren.

6. Sobald alle Metadaten vom Android-IF-MAP-Client veröffentlich wurden, werden diese durch die Detection Engine ausgewertet. Innerhalb dieses Anwendungsfalls wird eine App als potentiell schädlich erkannt wenn sie die folgenden Bedingungen erfüllt:
   - Sie muss mit den folgenden Berechtigungen ausgestattet sein: RECEIVE_BOOT_COMPLETED (automatisches Starten beim Start des Betriebssystems, INTERNET (Zugriff auf das Internet), CAMERA (Zugriff auf die Kamera des Endgerätes).
   - Zusätzlich muss sie aus einer unbekannten Quelle installiert worden sein (also nicht aus dem offiziellen Google Play-Store)

Als Testfall dient dabei der Android-IF-MAP-Client selbst, welcher alle diese Voraussetzungen erfüllt. Innerhalb der GUI der Detection Engine sollte nach einer kurzen Zeit angezeigt werden, dass eine schadhafte Software auf dem Endgerät erkannt wurde (Abbildung 10). Anschließend wird für das Endgerät ein Enforcement mittels iptables durchgeführt.

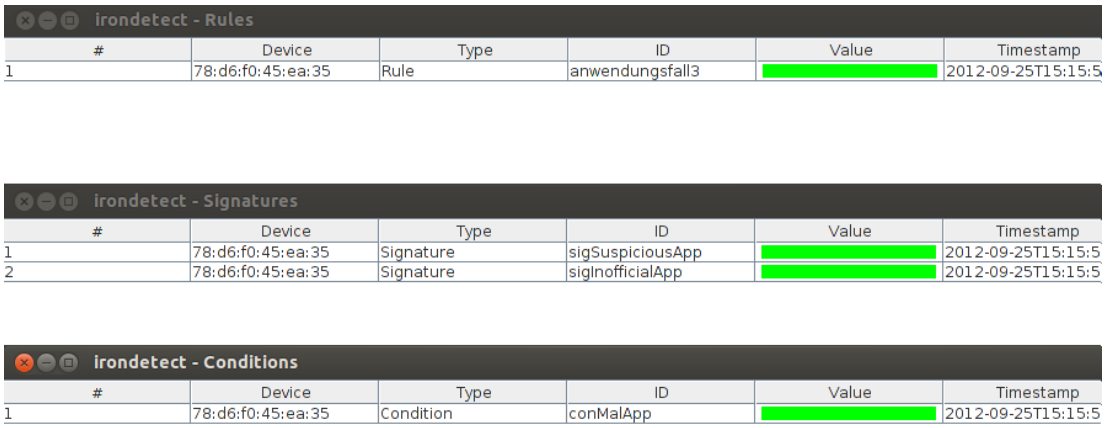


Abbildung 10: Erkennung einer schadhaften App durch irondetect

4.3.4 Anwendungsfall 4

Der vierte Anwendungsfall beschäftigt sich mit der Erkennung von Anomalien, also Abweichungen von einem als „Normal“ definierten Verhaltens. Folgende Komponenten kommen bei diesem Anwendungsfall zum Einsatz:

- MAP-Server (irond)
- Detection Engine (irondetect)
- Patient-Data-Service
- Skript zum Publizieren der Standart-Metadaten
- Iptables-IF-MAP-Client (Allowance-Komponente)

Die folgenden Ausführungsschritte sind zum Testen dieses Anwendungsfalls erforderlich:

1. MAP-Server starten

2. Standart-Metadaten mittels Hilfs-Skript veröffentlichen

3. Detection Engine starten (Policy beachten)

4. Patienten-Dienst starten

5. Als nächstes muss mit den Smartphone, welches wieder über einen der beiden Access Points innerhalb des internen Netzes verbunden ist, auf den Patienten-Dienst zugreifen. Das häufige Aufrufen dieses Dienstes innerhalb eines kurzen Zeitfensters führt dazu, dass die Detection Engine eine Anomalie erkennt.

6. Die Auswertung der Metadaten sowie die Erkennung der Anomalie werden, wie bereits erwähnt, durch die Detection Engine ausgeführt. Diese sollte nach mehreren Zugriffen innerhalb einer kurzen Zeitspanne wie in Abbildung 11 aufgeführt aussehen.

Abbildung 11 zeigt sämtliche Zugriffe des Clients auf den Patientendienst auf. Die Zugriffe welche in Rot markiert sind stellen normale Zugriffe dar, welche noch keine Anomalie darstellen. Nach einigen Zugriffen, welche innerhalb einer gewissen Zeitspanne durchgeführt wurden, setzt der Zeitpunkt ein, in welchem die Detection-Engine das Zugriffsverhalten als Anomalie einordnet (zu erkennen an der grünen Markierung). Anschließend wird ein entsprechender Event von der Detection Engine an den MAP-Server gesendet und ein Enforcement des jeweiligen Endgerätes mittels iptables durch die Enforcement-Komponente ausgeführt.

irondetect - Rules

#	Device	Type	ID	Value	Timestamp
1	78:d6:f0:45:ea:35	Rule	anwendungsfall4		2012-09-25T16:10:0
2	78:d6:f0:45:ea:35	Rule	anwendungsfall4		2012-09-25T16:10:1
3	78:d6:f0:45:ea:35	Rule	anwendungsfall4		2012-09-25T16:10:1
4	78:d6:f0:45:ea:35	Rule	anwendungsfall4		2012-09-25T16:10:1
5	78:d6:f0:45:ea:35	Rule	anwendungsfall4		2012-09-25T16:10:1
6	78:d6:f0:45:ea:35	Rule	anwendungsfall4		2012-09-25T16:10:1
7	78:d6:f0:45:ea:35	Rule	anwendungsfall4		2012-09-25T16:10:1
8	78:d6:f0:45:ea:35	Rule	anwendungsfall4		2012-09-25T16:10:1
9	78:d6:f0:45:ea:35	Rule	anwendungsfall4		2012-09-25T16:10:1
10	78:d6:f0:45:ea:35	Rule	anwendungsfall4		2012-09-25T16:10:1
11	78:d6:f0:45:ea:35	Rule	anwendungsfall4		2012-09-25T16:10:1
12	78:d6:f0:45:ea:35	Rule	anwendungsfall4		2012-09-25T16:10:1
13	78:d6:f0:45:ea:35	Rule	anwendungsfall4		2012-09-25T16:10:1
14	78:d6:f0:45:ea:35	Rule	anwendungsfall4		2012-09-25T16:10:1
15	78:d6:f0:45:ea:35	Rule	anwendungsfall4		2012-09-25T16:10:1
16	78:d6:f0:45:ea:35	Rule	anwendungsfall4		2012-09-25T16:10:2
17	78:d6:f0:45:ea:35	Rule	anwendungsfall4		2012-09-25T16:10:2
18	78:d6:f0:45:ea:35	Rule	anwendungsfall4		2012-09-25T16:10:2
19	78:d6:f0:45:ea:35	Rule	anwendungsfall4		2012-09-25T16:10:2
20	78:d6:f0:45:ea:35	Rule	anwendungsfall4		2012-09-25T16:10:2
21	78:d6:f0:45:ea:35	Rule	anwendungsfall4		2012-09-25T16:10:2

irondetect - Anomalies

#	Device	Type	ID	Value	Timestamp
1	78:d6:f0:45:ea:35	Anomaly	anoPatientAccess		2012-09-25T16:10:0
2	78:d6:f0:45:ea:35	Anomaly	anoPatientAccess		2012-09-25T16:10:1
3	78:d6:f0:45:ea:35	Anomaly	anoPatientAccess		2012-09-25T16:10:1
4	78:d6:f0:45:ea:35	Anomaly	anoPatientAccess		2012-09-25T16:10:1
5	78:d6:f0:45:ea:35	Anomaly	anoPatientAccess		2012-09-25T16:10:1
6	78:d6:f0:45:ea:35	Anomaly	anoPatientAccess		2012-09-25T16:10:1
7	78:d6:f0:45:ea:35	Anomaly	anoPatientAccess		2012-09-25T16:10:1
8	78:d6:f0:45:ea:35	Anomaly	anoPatientAccess		2012-09-25T16:10:1
9	78:d6:f0:45:ea:35	Anomaly	anoPatientAccess		2012-09-25T16:10:1
10	78:d6:f0:45:ea:35	Anomaly	anoPatientAccess		2012-09-25T16:10:1
11	78:d6:f0:45:ea:35	Anomaly	anoPatientAccess		2012-09-25T16:10:1
12	78:d6:f0:45:ea:35	Anomaly	anoPatientAccess		2012-09-25T16:10:1
13	78:d6:f0:45:ea:35	Anomaly	anoPatientAccess		2012-09-25T16:10:1
14	78:d6:f0:45:ea:35	Anomaly	anoPatientAccess		2012-09-25T16:10:1
15	78:d6:f0:45:ea:35	Anomaly	anoPatientAccess		2012-09-25T16:10:1
16	78:d6:f0:45:ea:35	Anomaly	anoPatientAccess		2012-09-25T16:10:2
17	78:d6:f0:45:ea:35	Anomaly	anoPatientAccess		2012-09-25T16:10:2
18	78:d6:f0:45:ea:35	Anomaly	anoPatientAccess		2012-09-25T16:10:2
19	78:d6:f0:45:ea:35	Anomaly	anoPatientAccess		2012-09-25T16:10:2
20	78:d6:f0:45:ea:35	Anomaly	anoPatientAccess		2012-09-25T16:10:2
21	78:d6:f0:45:ea:35	Anomaly	anoPatientAccess		2012-09-25T16:10:2

irondetect - Conditions

#	Device	Type	ID	Value	Timestamp
1	78:d6:f0:45:ea:35	Condition	conPatientAccess		2012-09-25T16:10:0
2	78:d6:f0:45:ea:35	Condition	conPatientAccess		2012-09-25T16:10:1
3	78:d6:f0:45:ea:35	Condition	conPatientAccess		2012-09-25T16:10:1
4	78:d6:f0:45:ea:35	Condition	conPatientAccess		2012-09-25T16:10:1
5	78:d6:f0:45:ea:35	Condition	conPatientAccess		2012-09-25T16:10:1
6	78:d6:f0:45:ea:35	Condition	conPatientAccess		2012-09-25T16:10:1
7	78:d6:f0:45:ea:35	Condition	conPatientAccess		2012-09-25T16:10:1
8	78:d6:f0:45:ea:35	Condition	conPatientAccess		2012-09-25T16:10:1
9	78:d6:f0:45:ea:35	Condition	conPatientAccess		2012-09-25T16:10:1
10	78:d6:f0:45:ea:35	Condition	conPatientAccess		2012-09-25T16:10:1
11	78:d6:f0:45:ea:35	Condition	conPatientAccess		2012-09-25T16:10:1
12	78:d6:f0:45:ea:35	Condition	conPatientAccess		2012-09-25T16:10:1
13	78:d6:f0:45:ea:35	Condition	conPatientAccess		2012-09-25T16:10:1
14	78:d6:f0:45:ea:35	Condition	conPatientAccess		2012-09-25T16:10:1
15	78:d6:f0:45:ea:35	Condition	conPatientAccess		2012-09-25T16:10:1
16	78:d6:f0:45:ea:35	Condition	conPatientAccess		2012-09-25T16:10:2
17	78:d6:f0:45:ea:35	Condition	conPatientAccess		2012-09-25T16:10:2
18	78:d6:f0:45:ea:35	Condition	conPatientAccess		2012-09-25T16:10:2
19	78:d6:f0:45:ea:35	Condition	conPatientAccess		2012-09-25T16:10:2
20	78:d6:f0:45:ea:35	Condition	conPatientAccess		2012-09-25T16:10:2
21	78:d6:f0:45:ea:35	Condition	conPatientAccess		2012-09-25T16:10:2

Abbildung 11: Erkennung einer Anomalie durch irondetect

4.4 Ergänzendes Material

Als ergänzendes Anschauungsmaterial wurden zusätzlich zu dem Demonstrator Videos für die einzelnen Anwendungsfälle erstellt, welche die Abläufe und die Kommunikation der beteiligten Komponenten visualisieren und den Betrachter dabei helfen sollen, eine Übersicht sowie ein weitergehendes Verständnis für die Abläufe innerhalb des Demonstrators zu entwickeln. Dabei wurde für jeden einzelnen der in [ESAP312] definierten Anwendungsfälle jeweils ein Video erstellt.

Folgende Videos wurden von der DECOIT GmbH erstellt:

a. ESUKOM PlugFest 2012 Demo-Video: TCG TNC Spring PlugFest 2012
b. ESUKOM Demo-Video (1): Real-time Enforcement
c. ESUKOM Demo-Video (2): Location Based Services
d. ESUKOM Demo-Video (3): Anomaly Detection
e. ESUKOM Demo-Video (4): MalApp-Detection

Abbildung 12: ESUKOM-Video “Real-time Enforcement”

Sämtliches ergänzendes Material kann auf der ESUKOM-Webseite (www.esukom.de) gefunden werden. Darüber hinaus enthalten die beiden Demonstratoren (VSA und ZIP-Archiv mit den einzelnen virtuellen Maschinen) jeweils eine Kurz-Dokumentation, welche dabei helfen kann, für die häufigsten auftretenden Probleme eine Lösung zu finden.

5 Zusammenfassung der Testergebnisse

Die abschließenden Tests wurden im Rahmen des letzten abgehaltenen Projekt-Workshops der ESUKOM-Partner durchgeführt. Innerhalb des Workshops wurde der finalen Prototypen implementiert und anhand mehrerer vordefinierter Anwendungsfälle auf seine Funktionalität hin überprüft. Die getesteten Anwendungsfälle beinhalten eine Auswahl unterschiedliche Kernanforderungen, welche im Rahmen des Projektes erarbeitet wurden:

- Real-Time Enforcement
- Smartphone-Awareness
- Location Based Services
- Anomalie-Erkennung
- Malware-Erkennung

Um diese Anforderungen in einen beispielhaften Kontext zu bringen, wurde ein fiktives Gesamtszenario definiert, in welchem eine beispielhafte IT-Infrastruktur innerhalb eines Krankenhauses beschrieben wird. Unterschiedliche Benutzergruppen (Ärzte sowie andere Mitarbeiter) greifen dabei auf verschiedene Ressourcen zu, wobei der Zugriff auf diese Ressourcen durch vorher festgelegte Sicherheitsrichtlinien reglementiert ist. Eine detaillierte Beschreibung des umgesetzten Demonstrators kann in Kapitel 4 gefunden werden. Anhand des Prototypens wurden exemplarisch sämtliche der im folgendem aufgeführten Anwendungsfälle auf ihre technische Umsetzung hin überprüft.

5.1 Anwendungsfall 1

Der erste Anwendungsfall beinhaltet den Zugriff eines mobilen Endgerätes vom externen Netz auf geschützte Ressourcen im internen Netz (in diesem Fall sensible Informationen, welche durch einen zusätzlichen Dienst bereitgestellt werden) und beinhaltet die Kernanforderung „Real-Time-Enforcement“. Damit ein mobiles Endgerät auf diese Informationen zugreifen darf, muss zunächst eine Verbindung vom externen in das interne Netz mit Hilfe einer VPN-Verbindung hergestellt werden. Abbildung 13 zeigt diesen Ablauf schematisch auf, wobei die roten Linien den Verlauf der Kommunikation darstellen.

Nach der erfolgreichen Herstellung der Verbindung werden daraufhin die Metadaten des mobilen Endgerätes (z.B. die ISP- sowie VPN-IP-Adresse) vom NCP VPN-Server innerhalb des Metadatengraphen veröffentlicht. Bei diesem Vorgang werden zusätzlich die dem Benutzer-Account zugeordneten Privilegien in Form von Capability-Metadaten publiziert. Im zweiten Schritt (Zugriff auf den Dienst mit den sensiblen Informationen) werden diese Privilegien vom iptables-IF-MAP-Client überprüft (siehe Abbildung 13, die zu überprüfenden Metadaten sind rot markiert).

Sollten die für den Dienst benötigten Privilegien innerhalb des Metadatengraphen gefunden werden, wird für das entsprechende Endgerät eine iptables-Regel ausgeführt, welche den Zugriff auf den Dienst freischaltet.

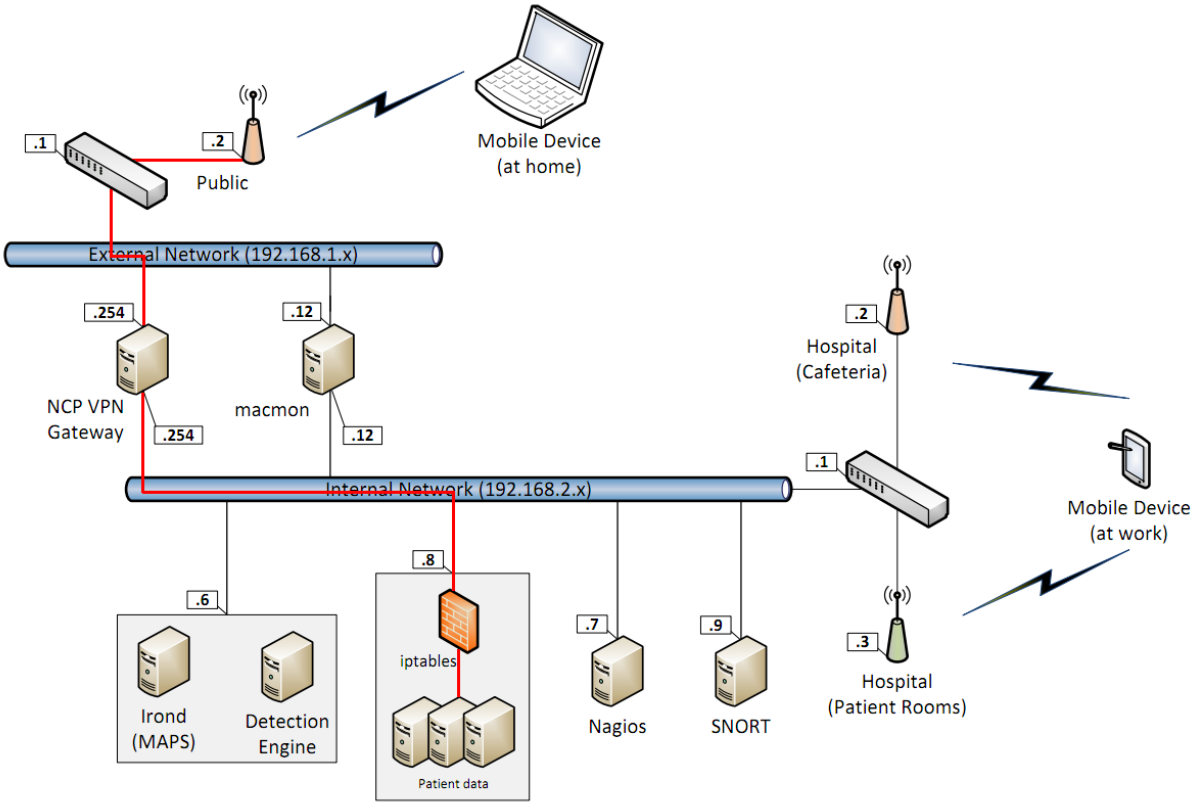


Abbildung 13: Zugriff aus dem externen auf das interne Netz

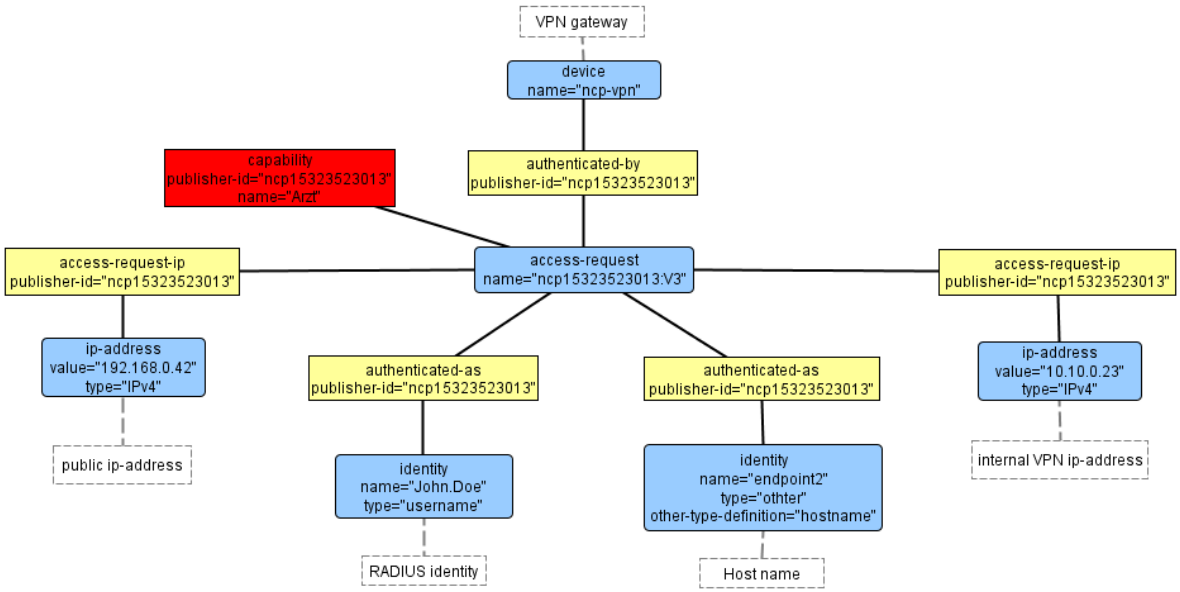


Abbildung 14: Durch iptables-IF-MAP-Client abgefragte Metadaten (1)

5.2 Anwendungsfall 2

Im Rahmen des zweiten Anwendungsfalls wird die Umsetzung der Kernanforderungen „Location-Based-Services“, „Smartphone-Awareness“ und „Real-Time-Enforcement“

überprüft. Hierzu verbindet sich ein mobiles Endgerät über einen der Access Points im internen Netz mit dem Patientendienst. Innerhalb der Sicherheitsrichtlinien für diesen Zugriff wurde festgelegt, dass dieser nur von mobilen Endgeräten durchgeführt werden darf, welche a) die entsprechenden Capability-Metadaten besitzen und b) über den Access Point innerhalb der Patientenräume mit den internen Netz verbunden sind (und nicht über den Access Point innerhalb der Cafeteria). Die Kommunikationswege sind in Abbildung 15 abgebildet.

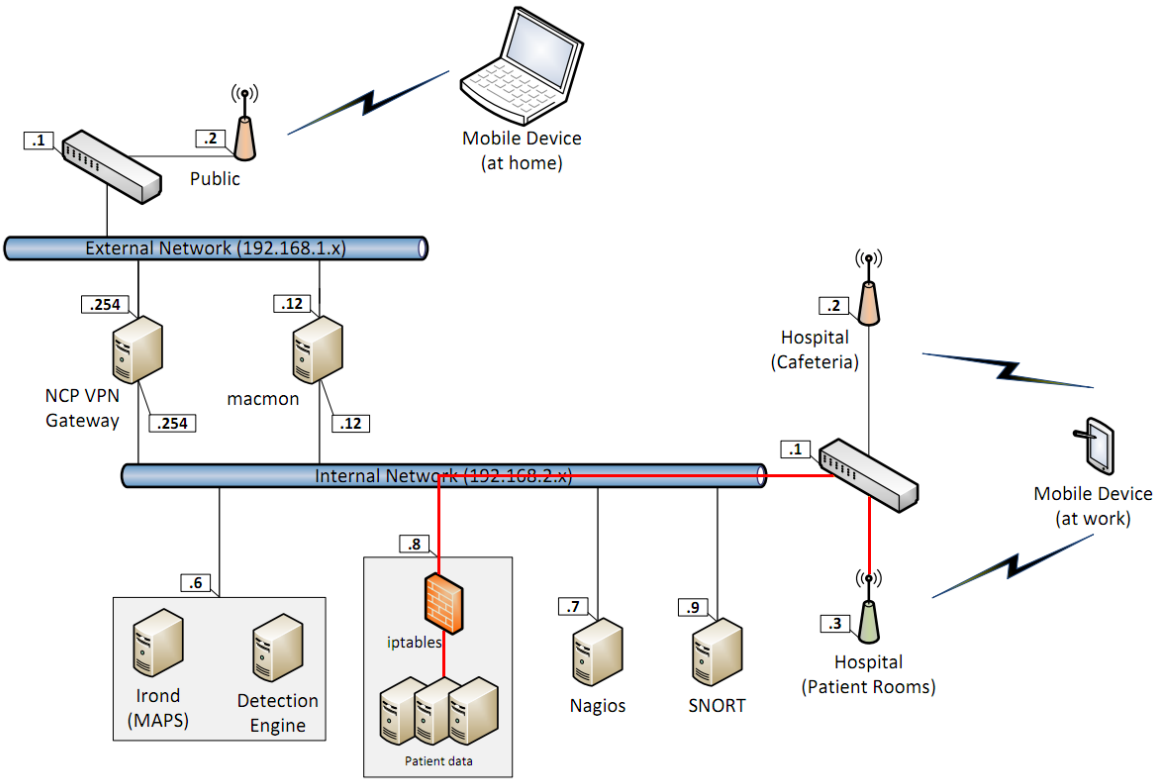


Abbildung 15: Zugriff über einen WLAN-Access-Point

Die Initiale Abfrage der Capability-Metadaten verläuft in diesem Fall analog zur Abfrage im 1. Anwendungsfall, allerdings werden hier die Capability-Metadaten durch macmon im Metadatengraphen veröffentlich. Sobald ein Zugriff erfolgt werden entsprechende Metadaten vom Daten-Dienst innerhalb des Metadatengraphen veröffentlicht. Die Detection Engine wertet diese Ergebnisse aus und prüft, ob der entsprechende Zugriff von einen Gerät aus erfolgte, welches über den Access Point im Patientenbereich mit dem internen Netz verbunden ist. Wenn dies nicht der Fall sein sollte, werden von der Detection Engine entsprechende Alert-Metadaten veröffentlich, welche wiederum durch den iptables-IF-MAP-Client abgefragt werden und ein Enforcement des Endgerätes zur Folge haben. Die von der Detection Engine abgefragten Metadaten des macmon-IF-MAP-Client bzgl. des aktuell benutzten Access Points sind in Abbildung 16 aufgeführt, die abgefragten Metadaten sind dabei rot markiert.

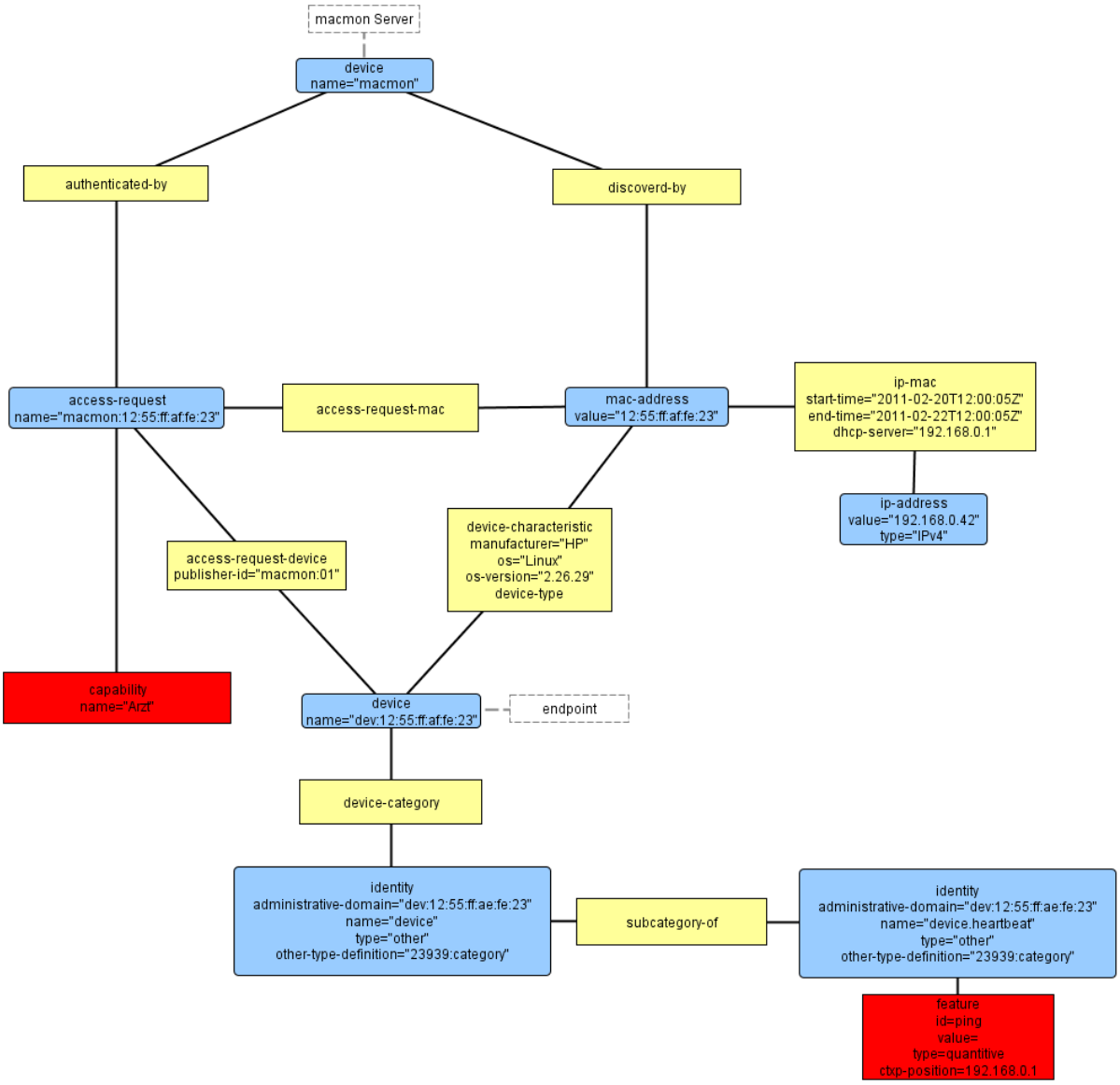


Abbildung 16: Durch iptables-IF-MAP-Client abgefragte Metadaten (2)

5.3 Anwendungsfall 3

Im dritten Anwendungsfall steht das Erkennen einer potentiell schadhaften Software, welche auf dem zugreifenden mobilen Endgerät installiert ist, im Mittelpunkt („MalApp-Detection“). Hierbei ist es für den Testfall zunächst egal, ob der Zugriff aus dem internen oder externen Netzwerk heraus erfolgt.

Die Erkennung einer potentiell schädlichen App wird durch die Detection Engine vorgenommen, welche zur Auswertung die durch den Android-IF-MAP-Client bereitgestellten Metadaten heranzieht. Dieser veröffentlicht u.a. alle auf dem Smartphone installieren Anwendungen inkl. deren Berechtigungen (im Kontext von Android sogenannte „Permissions“). Anhand des Abgleichs mit den in der jeweiligen Policy der Detection Engine festgelegten Kombination aus diesen Berechtigung mit den vorhandenen Metadaten wird dann eine Erkennung von als schädlich zu kategorisierenden Applikationen durchgeführt (Signatur-basierte Erkennungsmethode). Die entsprechende Policy-Datei ist in Abbildung 17 aufgeführt.

```
context {
}

hint {

}

anomaly {

}

signature {
        sigSuspiciousApp := "smartphone.android.app.permission.granted!1" = "android.permission.RECEIVE_BOOT_COMPLETED" and
        "smartphone.android.app.permission.granted!1" = "android.permission.CAMERA" and "smartphone.android.app.permission.granted!1" =
        "android.permission.INTERNET";

        sigInofficialApp := "smartphone.android.app.installer" != "com.android.vending";
}

condition {
        conMalApp := sigSuspiciousApp and sigInofficialApp;
}

action {
        enforcementIsolate := "enforcement.action.isolate" "./drop-client.sh" "$1" "@smartphone.device.ipaddress";
}

rule {
        anwendungsfall3 := if conMalApp do enforcementIsolate;
}
```

Abbildung 17: Policy-Defintion zur Erkennung potentiell schädlicher Apps

Die Metadaten des Android-IF-MAP-Clients welche in diesem Fall abgefragt werden sind in Abbildung 18 rot markiert (der dort abgebildete Metadatengraph wurde aus Gründen der Übersichtlichkeit auf die wesentlichen Metadaten reduziert und zeigt nur einen Teil des kompletten Graphen).

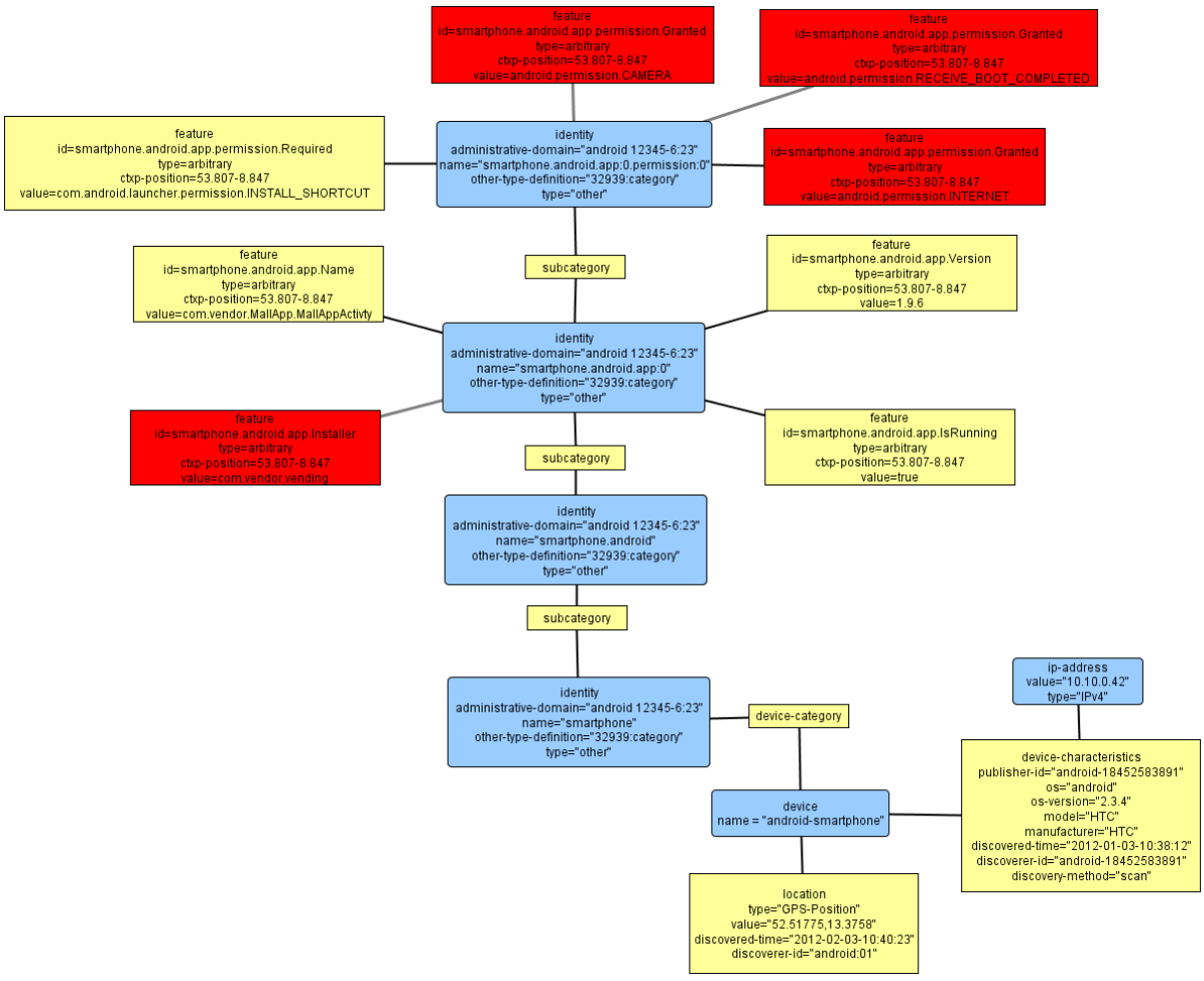


Abbildung 18: Durch die Detection Engine abgefragte Metadaten (1)

Sobald durch die Detection Engine eine potentiell schadhafte Applikation auf dem mobilen Endgerät gefunden wurde, werden entsprechende Alert-Metadaten veröffentlicht, welche eine Sperrung des Endgerätes für den Datendienst durch den iptables-IF-MAP-Client zur Folge haben.

5.4 Anwendungsfall 4

Der letzte Anwendungsfall beschäftigt sich mit der Kernanforderung der Anomalie-Erkennung. Kernpunkt dieses Anwendungsfalles ist das Erkennen von Abweichung gegenüber eines als „Normal“ definierten Verhaltens eines Endgerätes bzw. dessen Benutzers.

Als konkreter Testfall wird dabei das häufige Zugreifen auf die sensiblen Informationen des Patienten-Daten-Dienstes durch ein mobiles Endgerät simuliert. Das entsprechende Gerät wird dabei zunächst von allen beteiligten Komponenten autorisiert. Anschließend wird auf diesen Gerät innerhalb eines gewissen Zeitfensters in einer Frequenz auf diesen Dienst zugegriffen, welcher soweit von dem gemessenen Normal-Verhalten abweicht, das durch die Detection Engine eine Anomalie erkannt wird und mittels Alert-Event ein Enforcement durch den iptables-IF-MAP-Client angestoßen wird. Die Policy-Definition für diesen Anwendungsfall ist in Abbildung 19 aufgeführt.

```
context {
        ctxLastMinute := SLIDING = "00:02:00";
}

hint {
        hintPatientAccess := "patient.patient-information-access" "de.fhhannover.inform.trust.irondetectprocedures.MeanDaily" "10";
}

anomaly {
        anoPatientAccess := hintPatientAccess > 0.5 ctxLastMinute;
}

signature {

}

condition {
        conPatientAccess := anoPatientAccess;
}

action {
        iptablesIsolate := "message" "isolate" "source" "@smartphone.device.ipaddress";
        enforcementIsolate := "enforcement.action.isolate" "./drop-client.sh" "$1" "@smartphone.device.ipaddress";
}

rule {
        anwendungsfall4 := if conPatientAccess do enforcementIsolate;
}
```

Abbildung 19: Policy-Defintion zur Erkennung von Anomalien

Die Metadaten, welche zum Zwecke der Anomalie-Erkennung von der Detection Engine analysiert werden, werden durch den Patientendaten-Dienst publiziert und sind in Abbildung 20 in Rot dargestellt.

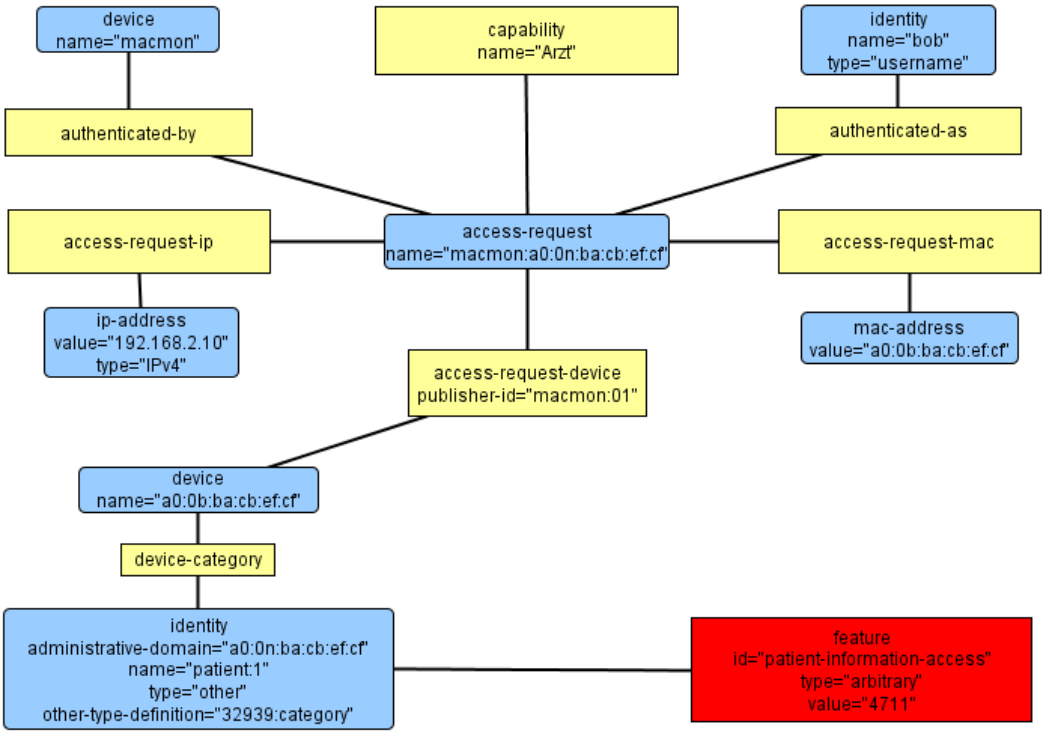


Abbildung 20: Durch die Detection Engine abgefragte Metadaten (2)

5.5 Ergebnisse der durchgeführten Tests

Die Durchführung der Testfälle verlief insgesamt positiv, allerdings gab es noch einige Punkte, die anfänglich zu Problemen und Unklarheiten geführt haben:

a. So hat sich zunächst gezeigt, dass das Aufsetzen und das Konfigurieren der Infrastruktur sowie der beteiligten Komponenten mit einen nicht unerheblichen Aufwand verbunden ist, welcher u.a. der hohen Anzahl der verwendeten Komponenten und deren teilweise nicht trivialen Konfigurationsschritte geschuldet ist. Hierbei zeigte sich auch, dass die Erstellung von Policy-Konfigurationen für die Detection-Engine aufgrund der Komplexität teilweise zu Problemen führt. An dieser Stelle wäre eine weitere Hilfestellung durch ein zusätzliches Tool hilfreich, welches eine grafische Oberfläche zum Erstellen dieser Dateien zur Verfügung stellt. Die Entwicklung einer solchen Software hätte allerdings nicht mehr in den zeitlichen Rahmen des Projektes gepasst, da diese auch mit einem nicht unerheblichen Aufwand verbunden gewesen wäre.

b. Ein weiteres Problem ergab sich im Zusammenspiel zwischen dem NCP VPN-Gateway und den iptables-Client im ersten Anwendungsfall. Wenn zum Aufbau der VPN-Verbindung der native Android-VPN-Client zum Einsatz kam, konnte der iptables-Client bei einem Zugriff auf den geschützten Patientendienst nicht die dem Gerät zugeordnete VPN-IP-Adresse auslesen. Die Abfrage der benötigten Capability-Metadaten liefert dementsprechend nicht die richtigen Daten zurück, woraufhin das mobile Endgerät nicht durch den iptables-Client freigeschaltet wurde. Wenn jedoch der NCP VPN-Client für Android zum Einsatz kam, funktionierte die Kommunikation zwischen den beiden Komponenten ohne Probleme.

Somit zeigt sich abschließend, dass mit Hilfe des Demonstrators alle definierten Testfälle erfolgreich durchgespielt werden konnten. Trotzdem gibt es einigen Stellen noch Potential für Verbesserungen, um einen insgesamt reibungsloseren Ablauf zu gewährleisten. Diese Optimierungen sind jedoch nicht mehr Bestandteil des ESUKOM-Projektes und sind im Rahmen einer prototypischen Umsetzung nicht mehr notwendig.

Durch die gemeinsame Zusammenarbeit der beteiligten Partner konnte somit alle Anforderungen an den Prototypen abgedeckt werden und der Test der festgelegten Anwendungsfälle mit einen positiven Fazit durchgeführt werden. Vor allem die Korrelation der Metadaten innerhalb des Metadatengraphen durch die Detection Engine, lief schon sehr zuverlässig und zeigt durch die beiden letzten Anwendungsfälle, welches Potential hinter der Korrelation der Metadaten zur Erkennung von Angriffen, welche ansonsten verborgen geblieben wären, steckt.

6 Zusammenfassung

Das vorliegende Dokument beschreibt die Best-Practise-Richtlinien, welche sich innerhalb der Arbeiten im Rahmen des ESUKOM-Projektes herausgestellt haben, sowie den bereitgestellten Demonstrator, welcher die definierten Anwendungsfälle des Projekts prototypisch umsetzt.

Hierzu wurde in Kapitel 2 zunächst eine einleitende, allgemeine Übersicht über das ESUKOM-Projekt gegeben, welche neben den eigentlichen Zielen des Projekts auch nochmal auf die IF-MAP-Spezifikation an sich sowie auf den Mehrwert, welcher durch diese entstehen kann, eingeht. Im darauffolgenden Kapitel 3 wurden die Problemstellungen geschildert, welche sich währen der konkreten Umsetzung mit Hilfe der IF-MAP-Spezifikation ergaben, sowie entsprechende Best-Practise-Richtlinien, welche zur Lösung der Probleme erarbeitet wurden, definiert. Diese Probleme sowie deren Lösungen wurden auch gegenüber der Trusted Computing Group (TCG) kommuniziert, so dass ESUKOM auch Einfluss auf die weitere Standardisierung nehmen konnte. Das entsprechende Dokument findet sich im Anhang („IF-MAP-Issues Version 0.4“). Innerhalb des Kapitels 4 wurde der bereitgestellte Demonstrator nochmal detailliert beschrieben, wobei auch auf die wichtigsten Konfigurationsschritte eingegangen wurde. Das Kapitel endet mit einer Schritt-für-Schritt Anleitung zum Durchspielen der einzelnen Anwendungsfälle, welche innerhalb des Demonstrators umgesetzt wurden. Das Kapitel 5 gibt eine Übersicht über die Testergebnisse, welche im Rahmen des letzten Projekts-Workshops mit dem Demonstrator erzielt wurden. Dazu wurde zunächst beschrieben, wie die Umsetzung der einzelnen Anwendungsfälle erfolgte. Abschließend wurden eine Zusammenfassung der erzielten Testergebnisse sowie ein Ausblick auf weiter mögliche Verbesserungen gegeben.

Das ESUKOM-Projekt hatte zum Ziel mit Hilfe der IF-MAP-Spezifikation die Sicherheit von Unternehmensnetzen zu steigern, gerade im Hinblick auf mangelnde Konzepte zur Sicherheit von Smartphones. Die im Projekt erarbeiteten Konzepte zeigten ein hohes Potential zur Verbesserung der IT-Sicherheit moderner Infrastrukturen. Die ersten funktionsfähigen Prototypen des Projektes können bereits erfolgreich Metadaten über IF-MAP austauschen, was auch auf einem internationalen PlugFest der Trusted Computing Group (TCG) im März 2012 in Darmstadt nachgewiesen werden konnte. Auf diese Weise ist jetzt schon eine Integration von zwei kommerziellen und fünf Open-Source-Produkten über IF-MAP erfolgreich umgesetzt worden.

Die größte Herausforderung war die Umsetzung der Detection Engine. Es ist eine offene Forschungsfrage, mit welchen Methoden welche Metadaten am geeignetsten auszuwerten sind, um bestimmte Anomalien erkennen zu können. Die Detection Engine wurde daher so ausgelegt, dass diese Konfiguration flexibel über entsprechende Policys gesteuert werden kann und in der Lage war Trainingsdaten zu sammeln. Hinzu kam, dass eine generische Schnittstelle zur Anbindung beliebiger Analyseverfahren (Correlation Methods) geschaffen wurde. Weitere Arbeiten wären notwendig, um die erarbeiteten Ansätze weiter zu entwickeln. Dies könnte durch ein Nachfolgeprojekt gemeistert werden, welches auch seitens des Konsortiums beantragt wurde. [DSBW12]

7 Anhang

7.1 Dokument „IF-MAP-Issues, Version 0.4“ vom 09.05.2012

IF-MAP Issues

Version 0.4

May 9, 2012

Abstract During the course of the ESUKOM project, the consortium ran into a number of issues when implementing the IF-MAP protocol. Although the basic adoption of the protocol was straightforward, the reasonable interaction of multiple components via IF-MAP was more complex. This document summarizes all encountered issues and highlights the approaches that were employed by the consortium in order to address them. Several members of the ESUKOM project contributed to this document. For any questions or remarks, please contact Ingo Bente (ingo.bente@fh-hannover.de).

Contents

1

1 About the ESUKOM Project

The ESUKOM project started in October 2010 with a duration of two years. It is funded by the German Federal Ministry of Education and Research. ESUKOM was and still is the first research project in Germany that completely focuses on the adoption of IF-MAP. The term ESUKOM refers to the german abbreviation of the phrase "Real-time Security for Enterprise Networks by Correlation of Metadata". The main goal of the project is to develop a network-based security system based on IF-MAP. For this purpose, a number of well-known Open Source Tools and two commercial security products are integrated by leveraging IF-MAP. At the beginning of the project, only one of the five project partners had experience with IF-MAP. For further information please visit the ESUKOM website

2 Issues

The following issues were encountered during the course of the project. All remarks refer to the IF-MAP Binding for SOAP specification version 2.0 revision 47 and the IF-MAP Metadata for Network Security specification version 1.0 revision 25.

2.1 Multiple Device Identifiers for one Endpoint

2.1.1 Problem Statement

The IF-MAP Metadata for Network Security specification includes an example where one endpoint establishes both a layer 2 and a layer 3 connection (section 9.12 Example 11). A similar use case is addressed within the ESUKOM project. The commercial NAC solution publishes metadata based upon a device's layer 2 connection. Furthermore, the commercial VPN server also publishes metadata for any layer 3 connection. Thus, a graph similar to the one depicted in figure 14 of the example was expected.

However, to link two separate `access-request` identifiers to the same `device` identifier was harder than expected. Since there are two different MAP clients involved (the VPN server and the NAC solution), both need to refer to the same `device` identifier. This introduces some sort of race condition. The first MAPC can publish its metadata straightforward, including the `access-request-device` metadata. However, the second MAPC needs to publish the `access-request-device` metadata on a link between its own `access-request` identifier and the previously used `device` identifier. Since the `name` element of the `device` identifier is generated somehwat randomly in order to fulfill certain uniqueness requirements (as specified in section 3.2.1 of the base spec), it remains an open question how the second MAPC knows the matching `device` identifier.

2.1.2 ESUKOM Approach

In order to link the two `access-request` identifiers to the same `device` identifier, the ESUKOM consortium agreed to levarage an `identity` identifier that expresses the hostname of the respective device. Both the VPN server and the NAC solution publish `authenticated-as` metadata on the link between their `access-request` and the respective `identity` identifier. The type attribute of the `identity` identifier is set to 'other', the other-type-definition attribute reflects the fact that the identity represents a hostname. Before publishing `access-request-device` metadata, the corresponding MAPC searches for other `access-request` identifiers that are linked to the same hostname `identity` identifier. If present, the MAPC can easily figure out which `device` identifier has been used by the first MAPC. Otherwise, you would end up with two separate device identifiers for the same endpoint (which might not cause any trouble but should be noted).

2.1.3 Conclusions

To link multiple `access-request` identifiers to the same `device` identifier was harder than expected when more than one MAPC was involved. The root cause for that are the uniqueness requirements for the `device` identifier's `name` element (as explained in section 3.2.1 of the base spec). This alone would not have been addressed as an issue. However, example 11 of the network security metadata spec depicts a graph where two `access-request` identifiers are linked to the same `device` identifier, even though multiple MAP clients where participating, without describing how this can be achieved. This led to some confusion whether the ESUKOM consortium may have missed some important details of the spec. However, a work around was easily implemented. Further versions of the spec might benefit from additional wording to clarify the example.

2.2 Vendor-specific Attributes for Standard Metadata

2.2.1 Problem Statement

Several use cases of the ESUKOM project involve a commercial VPN solution that manages remote access of mobile devices, including smartphones. When such a mobile device dials in, the VPN server publishes metadata to the MAPS. At this time, the VPN server knows two IP addresses that are associated to the respective mobile device:

1. the IP address that the client uses to connect to the server (ISP address),
2. the IP address the server assigns to the client to use in the VPN network (VPN address).

Thus, the VPN server publishes `access-request-ip` metadata on the links between the one `access-request` identifier and the two `ip-address` identifiers. However, the notion of the two types of IP addresses (ISP vs VPN) is not covered by the published standard metadata.

2.2.2 ESUKOM Approach

Since the notion of ISP versus VPN IP addresses is relevant for some ESUKOM use cases, approaches in order to add this information to the MAPS were discussed. One idea was to add a new attribute to the standard `access-request-ip` metadata instead of specifying a new vendor-specific metadata type. Since each standard metadata type uses the `metadataAttributes` attributeGroup which includes the `anyAttribute` element, schema validation should still work.

During the Spring PlugFest 2012 one MAPS did schema validation. Interoperability tests between the mentioned VPN server and the validating MAPS failed in the first place due to the vendor specific attribute that was added to `access-request-ip`. Although the root cause was a missing schema definition file, the question whether the approach breaks the spec or not was raised. Since section 3.9 of the spec says

> *IF-MAP standard metadata schema SHOULD NOT be extended directly with vendor-specific extensions.*

the ESUKOM consortium decided to change their approach. Instead of extending standard metadata, the new approach adds vendor-specific metadata which is published to the respective `ip-address` identifiers.

2.2.3 Conclusions

The ESUKOM consortium was confused by the purpose of the `anyAttribute` element in the `metadataAttributes` attributeGroup. It was interpreted in such a way that new attributes could be added by any vendor at any time. This notion was emphasized since the spec does no make use of the `namespace` attribute which could be used to limit the set of valid attributes to certain namespaces. If other implementers run into the same issue, the spec might benefit from additional wording like

> *Vendor-specific attributes MUST NOT be added to standard metadata types this way.*

to section 3.3.4 of the IF-MAP Binding for SOAP specification version 2.0.

2.3 Rootless Search Operations

2.3.1 Problem Statement

In order to perform a search or subscribe operation, a specific identifier must be given as root node. Queries like "give me all ip-mac metadata that is currently published" are not possible. This can turn out as a problem if a MAPC is expected to observe a wide range of or even all metadata that is published to the MAPS. One example for such a MAPC is irongui

2.3.2 ESUKOM Approach

An extension to the base protocol was necessary in order to address this issue. A new operation called `dump` was added. A MAPC that sends this operation request to the irond MAPS get a response that includes all identifiers that have metadata published to themselves or on a link that is adjacent to them. Subsequently, the MAPC can perform standard subscribe or search operations in order to obtain all published metadata. This way, the full picture of metadata objects can be visualized by irongui (given that the MAPS supports the new operation).

2.3.3 Conclusions

The dump operation was necessary in order to implement irongui. The custom identifiers introduced in IF-MAP 2.1 will allow a more sophisticated approach to achieve the same functionality without breaking the spec.

2.4 No Directed Links

2.4.1 Problem Statement

The data model of IF-MAP is based upon an undirected graph. However, under certain circumstances, the notion of directed links might be beneficial. Several standard metadata types like `authenticated-by` or `authenticated-as` already indicate some sort of direction. Since those metadata objects are published on links between identifiers of different types (`access-request` – `identity`, `access-request` – `device`), the direction can be easily derived. However, there are use cases where this might not always be possible. The current discussion about IPsec metadata addresses a similar issue (see map_sg list).

2.4.2 ESUKOM Approach

Within the ESUKOM project, custom identifiers and vendor-specific metadata were defined in order to express some sort of a hierarchical relationship within IF-MAP. The direction of the relationships can be derived based on the values of the custom identifiers. Another approach that might work is to include some sort of 'reference' that is included in the metadata which is placed on a link. The references could act as pointers to the connected identifiers, thus expressing the direction of the relationship.

2.4.3 Conclusions

The notion of directed links can be relevant for certain use cases. Although directed edges are not part of the IF-MAP data model, simple workarounds are possible in order to integrate some sort of direction on a per metadata object basis. Thus, there seems to be no need for any extensions to IF-MAP in this regard.

2.5 No Restrictions on IP or MAC Address Strings

2.5.1 Problem Statement

The formats of valid IP and MAC addresses are precisely specified. For practical implementation, these addresses are represented as strings. However, the range of valid values for these strings is strongly limited. The semantic rules for valid IP and MAC addresses are also precisely written down together with the specifications of the identifier types `ip-address` and `mac-address` in sections 3.2.4 and 3.2.5 of the IF-MAP Binding Specification. However, the XML data schemes just specify the address strings as arbitrary strings without any restrictions. The validation of the address strings, i.e. the rejection of address strings not complying with the rules for valid addresses, is completely left to the application. It might be benefitial to include IP and MAC restrictions directly within the XML schema.

2.5.2 ESUKOM Approach

The correct semantic of valid MAC addresses is quite easy to implement with restriction rules for XML strings. Just the pattern statement needs to be used to limit the strings to consist of decimal and hexadecimal digits and delimeters in the prescribed form.

Therefore, the ESUKOM project suggests to modify the definitions of the identifier types `ip-address` and `mac-address` in the following way:

```
<xsd:complexType name="IP-AddressType">
 <xsd:attribute name="administrative-domain" type="xsd:string"/>
 <xsd:attribute name="value" use="required"/>
    <xsd:simpleType>
       <xsd:restriction base="xs:string">
          <xsd:pattern value="((25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9][0-9]?|0)\.){3} (25[0-5]|2[0-4][0-9]|1[0-9][0-
9]|[1-9][0-9]?|0)"/>
          <!– IPv4 addresses–>
          <xsd:pattern value="(([1-9a-f][0-9a-f]?[0-9a-f]?[0-9a-f]?|0):){7} ([1-9a-f][0-9a-f]?[0-9a-f]?[0-9a-f]?|0)"/>
          <!– IPv6 addresses –>
       </xsd:restriction>
    </xsd:simpleType>
 </xsd:attribute>
 <xsd:attribute name="type">
    <xs:simpleType>
       <xsd:restriction base="xs:string">
          <xs:enumeration value="IPv4"/>
          <xs:enumeration value="IPv6"/>
       </xsd:restriction>
    </xsd:simpleType>
 </xsd:attribute>
</xsd:complexType>
```

```
<xsd:complexType name="MAC-AddressType">
    <xsd:attribute name="administrative-domain" type="xsd:string"/>
    <xsd:attribute name="value" use="required"/>
        <xsd:simpleType>
            <xsd:restriction base="xsd:string">
                <xsd:pattern value="([0-9a-f][0-9a-f]:){5}[0-9a-f][0-9a-f]"/>
            </xsd:restriction>
        </xsd:simpleType>
    </xsd:attribute>
</xsd:complexType>
```

2.5.3 Conclusions

The current state of the IF-MAP Binding Specification allows arbitrary strings as values for IP and MAC addresses, and the application is completely responsible for the detection of invalid values. Our approach presented in the last section helps to already recognize possible invalid values within the specified XML schemes.

The modified definition of the identifier type `mac-address` incorporates the semantic rules for valid MAC and IP addresses into the XML scheme. Any string value not representing any valid value of an IP or MAC address will be rejected as invalid by the XML scheme. Given that XML validation is done by a MAPC or a MAPS, the application only needs to check that the format of the address string complies with the specified type IPv4, IPv6 respectively.

7.2 Literaturverweise

[ESAP312] ESUKOM-Konsortium: *Bericht AP3 im Verbundprojekt ESUKOM*. 2012

[DDB11] Detken, Dunekacke, Bente: *Konsolidierung von Metadaten zur Erhöhung der Unternehmenssicherheit*. D.A.CH Security 2011: Bestandsaufnahme, Konzepte, Anwendungen und Perspektiven, Herausgeber: Peter Schartner, syssec Verlag, ISBN 978-3-00-027488-6, Oldenburg 2011

[DSBW12] Detken, Scheuermann, Bente, Westerkamp: *Automatisches Erkennen mobiler Angriffe auf die IT-Infrastruktur*. D.A.CH Security 2012: Bestandsaufnahme, Konzepte, Anwendungen und Perspektiven, Herausgeber: Peter Schartner und Jürgen Taeger, syssec Verlag, ISBN 978-3-00-039221-4, Konstanz 2012

[HEIS10] Heise Online: *27C3 – Hacker analysieren Stuxnet-Maschinencode*. Ausgabe 12/2010, Heise-Online-Verlag, Hannover 2010

[IFM-SO12] TCG Trusted Network Connect: *TNC IF-MAP Binding for SOAP*. Specification V. 2.1, rev 15, Mai 2012.

[IFM-ME12] TCG Trusted Network Connect: *TNC IF-MAP Metadata for Network Security*. Specification V.1.1, rev 8, Mai 2012.

7.3 Projektveröffentlichungen

1. Kai-Oliver Detken: Trusted Computing: Flop oder Durchbruch des TPM-Chip? NET 09/12, ISSN 0947–4765, NET Verlagsservice GmbH, Woltersdorf 2012
2. K.-O. Detken, D. Scheuermann, I. Bente, J. Westerkamp: Automatisches Erkennen mobiler Angriffe auf die IT-Infrastruktur. D.A.CH Security 2012: Bestandsaufnahme, Konzepte, Anwendungen und Perspektiven, Herausgeber: Peter Schartner, syssec Verlag, ISBN 978-3-00-039221-4, Konstanz 2012
3. Kai-Oliver Detken: Erhöhung der IT-Sicherheit im Unternehmen. Vortrag auf der CeBIT am Bremer Gemeinschaftsstand, Halle 6, 7. März, Hannover 2012
4. Kai-Oliver Detken: Unternehmensnetzwerke schützen und dennoch mobiles Potential nutzen: Mögliche Bedrohungen und Sicherheitsansätze. DuD - Datenschutz und Datensicherheit, Ausgabe 3/2012, 36. Jahrgang, ISSN 1614-0702, Springer Fachmedien Wiesbaden GmbH, Wiesbaden 2012
5. Kai-Oliver Detken: IF-MAP - Konsolidierung von Sicherheitsdaten. Loseblattwerk: Vom LAN zum Kommunikationsnetz - Netze und Protokolle, 45. Ergänzungslieferung, 02/2012, WEKA Media GmbH & Co. KG, ISBN 978-3-8245-3501-7, Kissing 2012
6. I. Bente, G. Dreo, B. Hellmann, J. Vieweg, J. von Helden: Trustworthy Anomaly Detection for Smartphones. Poster presented at the 13th International Workshop on Mobile Computing Systems and Applications (ACM HotMobile), San Diego, CA, USA, 2012.
7. Kai-Oliver Detken: Das ESUKOM-Projekt: Konsolidierung von Metadaten zur Erhöhung der Unternehmenssicherheit. Handbuch der Telekommunikation,

Deutscher Wirtschaftsdienst, 148. Ergänzungslieferung, Dezember 2011, ISBN 978-387-156-096-5, Köln 2011

8. I. Bente, J. von Helden, B. Hellmann, J. Vieweg, K.-O. Detken: ESUKOM: Smartphone Security for Enterprise Networks. Securing Electronic Business Processes, Editors: Norbert Pohlmann, Helmut Reimer, Wolfgang Schneider, Vieweg+Teubner Verlag, Springer Fachmedien Wiesbaden GmbH, 13. Information Security Solutions Europe Conference (ISSE) conference in Prague, 22.-23. November, ISBN 978-3-8348-1911-6, Wiesbaden 2011
9. K.-O. Detken, W. Dürr, J. von Helden, J. Lucius: Verbundvorhaben von ESUKOM. Präsentation des Gesamtvorhabens ESUKOM auf dem Statusseminar "IT-Sicherheitsforschung" des Deutschen Zentrums für Luft und Raumfahrt (DLR), 18.-19. Oktober, Wissenschaftszentrum Bonn, Bonn 2011
10. K.-O. Detken, D. Dunekacke, I. Bente: Konsolidierung von Metadaten zur Erhöhung der Unternehmenssicherheit. D.A.CH Security 2011: Bestandsaufnahme, Konzepte, Anwendungen und Perspektiven, Herausgeber: Peter Schartner, syssec Verlag, ISBN 978-3-00-027488-6, Oldenburg 2011
11. N. Kuntze, C. Rudolph, I. Bente, J. Vieweg, and J. von Helden: Interoperable device Identification in Smart-Grid Environments. In: Power and Energy Society General Meeting, 2011 IEEE, July 2011
12. I. Bente, G. Dreo, B. Hellmann, S. Heuser, J. Vieweg, J. von Helden, J. Westhuis: Towards permission-based attestation for the Android platform. In Proceedings of the 4th international conference on Trust and Trustworthy Computing (TRUST'11). LNCS 6740, Springer-Verlag, Berlin, Heidelberg, pp. 108-115, 2011
13. Kai-Oliver Detken, Dennis Dunekacke: Verhängnisvolle Isolierung: IT-Sicherheit - mehr als die Summe aller Einzelteile. NET 05/11, ISSN 0947–4765, NET Verlagsservice GmbH, Woltersdorf 2011
14. Ingo Bente: IF-MAP in a Nutshell. Vortrag auf dem MASSIF Members Meeting, Fraunhofer SIT, Darmstadt 2011
15. Ingo Bente: ESUKOM Research on IF-MAP. Vortrag auf dem Workshop für Security und Datacenter-Design der Universität Frankfurt, Frankfurt 2011
16. Wolfgang Dürr: Der neue NAC-Standard IF-MAP: Erhöhung der Sicherheit in Unternehmensnetzen durch Datenkonsolidierung. Fachreferat auf der CeBIT, Hannover 2011
17. Kai-Oliver Detken: Sicherheit in mobilen Welten. WEKA-Verlag, Vom LAN zum Kommunikationsnetz: Systeme und Applikationen, Ausgabe-Nr. 05/2010, November 2010, WEKA Media GmbH & Co. KG, ISBN 978-3-8245-3502-6, Kissing 2010
18. Johannes Westhuis: Integration von Trusted Computing Technologien in die Android-Plattform. Masterarbeit im Studiengang Angewandte Informatik in der Abteilung Informatik der Fakultät IV an der Fachhochschule Hannover, 19. August 2010, Hannover 2010
19. Kai-Oliver Detken, Hervais Simo Fhom, Richard Sethmann, Günther Diederich: Leveraging Trusted Network Connect for Secure Connection of Mobile Devices to Corporate Networks. Communications: Wireless in Developing Countries and Networks of the Future, IFIP World Computer Congress (WCC), Ana Pont, Guy

Pujolle, S.V. Raghavan (Eds.), ISBN-13: 978-3-642-15475-1, Springer publishing house, Brisbane, Australia 2010

20. Kai-Oliver Detken: En vogue - neue Verfahren zur Absicherung mobiler Endgeräte. NET 09/10, NET Verlagsservice GmbH, Woltersdorf 2010
21. Tobias Ruhe: Visualisierung von Informationen einer zentralen Netzwerk-Datenbank. Bachelorarbeit im Studiengang Angewandte Informatik in der Abteilung Informatik der Fakultät IV an der Fachhochschule Hannover, Juli 2010, Hannover 2010
22. Waldemar Bender: Entwicklung eines IF-MAP Clients für die Android Plattform. Bachelorarbeit im Studiengang Angewandte Informatik in der Abteilung Informatik der Fakultät IV an der Fachhochschule Hannover, Juli 2010, Hannover 2010
23. Ingo Bente, Jörg Vieweg: IRON: Intelligent Reaction on Network Events. Vortrag auf dem 4. Essener Workshop zur Netzsicherheit (EWNS), Essen 2010
24. Ingo Bente, Bastian Hellmann, Jörg Vieweg, Josef von Helden, Arne Welzel: Interoperable Remote Attestation for VPN Environments. akzeptiert für Intrust 2010 (2nd International Conference on Trusted Systems) Beijing, P.R. China 2010
25. Jennifer Richter, Nicolai Kuntze, Carsten Rudolph: Securing Digital Evidence. Paper of the Technical University of Darmstadt and Fraunhofer SIT, 15th of June, Darmstadt 2010
26. Evren Eren, Stephan Uhde, Kai-Oliver Detken: Mobile Identity Management auf Basis des SIMOIT-Projekts und der TNC@FHH Entwicklung. Konferenzbuch "Wireless Communication and Information - Radio Engineering and Multimedia Applications", 15.-16. Oktober, Herausgeber: Jürgen Sieck und Michael A. Herzog, S. 283-297, vwh Verlag Werner Hülsbusch, Berlin 2009
27. Evren Eren, Stephan Uhde, Kai-Oliver Detken: User Centric Identity Management in Mobile Scenarios: The SIMOIT Project. IEEE International Workshop on Intelligent Data Acquisition and Advanced Computing Systems (IDAACS): Technology and Applications, 21.-23. September 2009, Rende (Cosenza), Italy

7.4 Abbildungen

7.5 Glossar

.NET	Framework zur Softwareentwicklung der Firma Microsoft
ADB	Android Debug Bridge, ein vielseitiges Kommandozeilen-Tool, welches Teil des Android-SDK ist und Kommunikation mit Android-Geräten oder Emulatoren erlaubt
AES	Advanced Encryption Standard, ein symmetrisches Kryptographie-Verfahren
Android	Betriebssystem der Open Handset Alliance für mobile Endgeräte
AP	Access Point
API	Application Programming Interface. Schnittstelle von Softwarekomponenten zur Anbindung weiterer Komponenten
APK	Android Package, ein Paketformat für Android-Applikationen, Dateiendung: .apk
APP	Abkürzung für Applikation, meistens als Synonym für mobile Applikation verwendet, von Apple geprägt (App Store)
ARM	32 Bit RISC Prozessorarchitektur
ASP	Active Server Pages. Serverseitige Skriptsprache der Firma Microsoft für die Entwicklung von Webseiten
BMBF	Bundesministerium für Bildung und Forschung
Bytecode	Darstellungsformat von Instruktionen, das in nativen Maschinencode einer Zielplattform, ggf. zur Laufzeit, übersetzt oder interpretiert wird
Chain of Trust	Vertrauenskette, die beschreibt, wie das Vertrauen in die Gesamtheit einer Plattform von der Initialisierung der grundlegenden Komponenten an bis in die einzelnen Anwendungen hin aufgebaut wird
Daemon	Im Hintergrund arbeitender Systemdienst eines Betriebssystems
Dalvik VM	Auf Android eingesetzte virtuelle Maschine, in der Android Anwendungen ausgeführt werden
DHCP	Dynamic Host Configuration Protocol
DMZ	De-Militarisierte Zone

EAP	Extensible Authentication Protocol. Erweiterbares Framework zur Authentifizierung von Benutzern und Clients in Netzwerken
EAP-TNC	EAP-Protokollerweiterung für den Einsatz von TNC
ESP	Encapsulating Security Payload. Datenformat zur Beschreibung
ESUKOM	Echtzeit-Sicherheit für Unternehmensnetze durch Konsolidierung von Metadaten
FHH	Hochschule Hannover – ehemals Fachhochschule Hannover
GRE	Generic Routing Encapsulation. Protokoll, um generische Netzwerkpakete mit Hilfe von IP Paketen zu tunneln
HTTP	Hypertext Transfer Protocol
IDE	Integrated Development Environment. Entwicklungsumgebung
IDS	Intrusion Detection System
IF-IMC	Teil des TNC-Standards: Schnittstellenbeschreibung zwischen TNCC und IMC
IF-IMV	Teil des TNC-Standards: Schnittstellenbeschreibung zwischen TNCS und IMV
IF-M	Nachrichtenformat zur Kommunikation zwischen IMC und IMV
IF-MAP	Interface for Metadata Access Points, Spezifikation der TCG
IF-T	Transportprotokoll des TNC Standards
IF-TNCCS	Kommunikationsprotokoll zwischen TNCC und TNCS
IF-T-Over-TLS	IF-T Protokoll Binding für TLS
IIS	Internet Information Services. Überbegriff für die Webserver Dienste der Firma Microsoft
IKE	Internet Key Exchange. Schlüsselaustauschprotokoll des IPSec Protokollstapels
IMA	Integrity Measurement Architecture, Linux Kernel Patch zur Messung der Integrität von Softwarekomponenten zur Laufzeit von IBM
Initial Ramdisk	Üblicherweise in Form einer Image-Datei erzeugtes Dateisystems, das Dateien enthält, die zum Starten des Kernels benötigt werden
IPsec	Internet Protocol Security. Protokollstapel zur Authentifizierung und Verschlüsselung von IP-Paketen
ISAKMP/Oakley	Internet Security Association and Key Management Protocol / Oakley Key Determination Protocol. Protokollstapel für die Etablierung von Security Associations und gemeinsamem Schlüsselmaterial.
ISP	Internet Service Provider
JTSS	Java-Implementierung eines TSS von der Arbeitsgruppe IAIK der TU Graz

MAC	Mandatory Access Control. Ein Zugriffskontrollsystem, das den Zugriff von Subjekten auf Objekte mittels eines Referenzmonitors auf Ebene des Betriebssystems anhand von den Subjekten und Objekten zugewiesenen Sicherheitsattributen und von privilegierten Benutzern erstellten Regeln kontrolliert
MAP	Metadata Access Point
MD5	Message Digest Algorithm 5. Eine kryptographische Hashfunktion.
NAC	Network Access Control – Überbegriff für Systeme zur Kontrolle des Endgerätezustands vor der Erlaubnis zum Betreten eines Netzwerkes
NAT	Network Address Translation. Verfahren zur Modifikation von Netzwerkadressen in IP-Paketen. Wird üblicherweise zur Abbildung von mehreren Netzwerkadressen auf eine einzelne in IP-basierten Netzwerken verwendet
ODBC	Open Database Connectivity. Standardisierte Software-API für den Zugriff auf Datenbanken
OMA	Open Mobile Alliance, ein Zusammenschluss von Mobilfunk-Herstellern, Providern und Dienstleistern
OMA DM	OMA Device Management, ein Protokoll zur Fernwartung mobiler Geräte
OpenSSL	Open Source Implementierung des SSL/TLS-Protokolls
OTA	Over The Air, Übertragung von Daten über Mobilfunk, oft im Zusammenhang mit OTA Firmware Updates oder für Verteilung von Softwarekomponenten an Clients benutzt
PCA	Privacy-CA, eine Zertifizierungsstelle mit deren Hilfe die Vertrauenswürdigkeit eines TPM-Chips festgestellt werden kann bei gleichzeitiger Wahrung der Privatsphäre des Benutzers
PCR	Platform Configuration Register. Zustandsregister des TPM
PEP	Policy Enforcement Point
PFS	Perfect Forward Secrecy. Eigenschaft eines Protokolls zur Etablierung eines gemeinsamen Session Schlüssels, die beschreibt, dass dieser nicht kompromittiert wird, sofern einer der privaten Schlüssel der asymmetrischen Schlüsselpaare, die zur Aushandlung verwendet wurden, in Zukunft kompromittiert wird
PHP	Serverseitige Skriptsprache für die Entwicklung von Webdiensten.
PKCS	Public Key Cryptography Standards. Ein Satz von Standards zur Beschreibung von Datenstrukturen und Algorithmen, die im Rahmen von Public Key Kryptographie benötigt werden
PPTP	Point-to-Point Tunneling Protocol. Auf TCP und GRE basierendes VPN-Protokoll
PSK	Pre-Shared Key. Vorab zwischen Peers vereinbarter Schlüssel

RADIUS	Remote Authentication Dial-In User Service. Eine Software zur Authentifizierung, Autorisierung und Abrechnung von Nutzern und Diensten (AAA-System)
RSA	Asymmetrisches Kryptosystem
Security Association	Beschreibt Sicherheitsparameter, die im Rahmen einer per IPSec geschützten Verbindung ausgehandelt werden
Secure Boot	Ein Sicherheitsverfahren, das den Start eines Betriebssystemkernels unterbindet, sofern sein Zustand von einem vorher definierten Sollzustand abweicht
SHA	Secure Hash Algorithm: Satz standardisierter kryptologischer Hashfunktionen, z.B: SHA1, SHA256
SHA1	Secure Hash Algorithm 1, eine kryptographische Hashfunktion
SMS	Short Messaging Service, Nachrichtendbasierter Kommunikationsdienst in Mobilfunknetzwerken
SML	Stored Measurement Log
SNMP	Simple Network Management Protocol
SOAP	Simple Object Access Protocol
SQL	Structured Query Language. Abfragesprache für Datenbanken
SSH	Secure Shell. Ein Protokoll für den sicheren Zugriff auf entfernte Rechner
SSL	Secure Sockets Layer. Kryptographisches OSI Layer 7 Protokoll zur Absicherung von Kommunikation auf Anwendungsebene. Vorgänger von TLS
String	Zeichenkette
Switch	Netzwerkkomponente zur Kopplung von Netzwerksegmenten.
TCG	Trusted Computing Group
TCP	Transmission Control Protocol. Zustandsbehaftetes, verbindungsorientiertes OSI Layer 4 Protokoll zum Transport von Daten auf Basis des IP-Protokolls
TLS	Transport Layer Security. Kryptographisches OSI Layer 7 Protokoll zur Absicherung von Kommunikation auf Anwendungsebene. Nachfolger von SSL
TNC	Trusted Network Connect: Ein von der von der TCG entwickelter herstellerunabhängiger Standard für sicherer Netzwerkzugriff
TNCC	TNC-Client
TNCS	TNC-Server
Toolchain	Überbegriff für einen Satz von Entwicklungswerkzeugen zur Kompilierung des Quelltextes eines Produktes
TPM	Trusted Platform Module, üblicherweise hardwarebasierter Vertrauensanker, spezifiziert von der Trusted Computing Group

TSS	TCG Software Stack, ein von der Trusted Computing Group spezifiziertes Software-Interface zur Kommunikation mit dem TPM
TUN/TAP	virtuelle Netzwerk-Kerneltreiber, die auf OSI-Layer 3 bzw. 2 Netzwerkgeräte in Software simulieren
UDP	User Datagram Protocol. Zustands- und Verbindungsloses OSI Layer 4 Protokoll zur Übertragung von Daten über IP-basierte Netzwerke
URI	Uniform Resource Identifier. Eine Zeichenkette zur eindeutigen Identifizierung von Objekten, z.B URL im Internet
URL	Uniform Resource Locator. Ein auf einem URI basierender Identifier für die Adressierung von Diensten
VISA	Virtual IT-Security Architecture: BMBF-Projekt im Forschungsprogramm “KMU Innovativ”
VLAN	Virtual Local Area Network. Verfahren zur Segmentierung von Netzwerken
VPN	Virtual Private Network
VSA	Virtual Security Appliance
W3C	World Wide Web Consortium, Gremium zur Standardisierung von Techniken für das World Wide Web (WWW)
WLAN	Wireless Local Area Network
X.509	ITU.T Standard, der in einer Public Key Infrastructure zum Einsatz kommt und die verwendeten Datenstrukturen und Algorithmen beschreibt
XAUTH	Erweiterung von ISAKMP/Oakley, um externe Authentifizierungsdienste einbinden zu können
XHTML	XML-basierte Seitenbeschreibungssprache zur Entwicklung von Webseiten
XML	Extensible Markup Language. Ein von der W3C standardisiertes Protokoll für die menschen- und maschinenlesbare Darstellung von Informationen
XSD	XML Schema Definition, Empfehlung des W3C zum Definieren von Strukturen für XML-Dokumente, ebenfalls in XML beschrieben