Android in SIEM-Umgebungen

Markus Schölzel[1] · Evren Eren[1] · Kai-Oliver Detken[2]

[1]Fachhochschule Dortmund, EFS 42, 44227 Dortmund
markus.schoelzel064@stud.fh-dortmund.de, evren.eren@fh-dortmund.de

[2]DECOIT GmbH, Fahrenheitstraße 9, 28359 Bremen
detken@decoit.de

Zusammenfassung

Die Nutzung von Smartphones und Tablets in Unternehmensnetzwerken steigt stetig, auch wenn diese Geräte nicht immer vertrauenswürdig sind. Sie können Ziel von Angriffen und Werkzeug für Attacken auf das Netzwerk sein. Zudem werden sie nur selten von Monitoring-Lösungen beachtet, da sie häufig nicht permanent, sondern nur zu bestimmten Zeiten genutzt werden und so zu temporären Fehlerquellen werden können. SIEM-Systeme müssen generell über die genutzten Geräte informiert werden, um diese sowohl in kurzfristige Überprüfungen als auch in Langzeitanalysen aufzunehmen. Portbasierte Authentisierungsstandards wie IEEE 802.1x mit zentralisierter AAA (Authentifizierung, Autorisierung und Accounting) erlauben bereits die Beschränkung des Zugangs, bieten jedoch keine umfassenden Informationen über das Verhalten der Geräte innerhalb des Netzwerks. Deshalb werden hier die Ergebnisse der Entwicklungen aus den FuE-Projekten SIMU und iMonitor vorgestellt, die Android in SIEM-Umgebungen mit und ohne IF-MAP berücksichtigen und so ein Monitoring von mobilen Endgeräten über die gesamte Nutzungsdauer im Netzwerk ermöglichen.

1 SIEM

SIEM-Systeme (Security Information and Event Management) ermöglichen als zentraler Bestandteil von Unternehmensnetzwerken ein proaktives Management von IT-Infrastrukturen. Diese Systeme setzen jedoch voraus, dass Daten erhoben und ausgewertet werden, um auf Bedrohungen und Angriffe zu reagieren. Zur Datenerhebung werden Sensoren genutzt, die bestimmte Komponenten oder Prozesse überwachen und Meldungen zu bestimmten Ereignissen erzeugen. Diese Meldungen müssen bezüglichen ihrer Wichtigkeit und Konsequenz bewertet werden.

Die Datenerhebung ist unproblematisch solange die Sensoren erreichbar und die zu überwachenden Geräte permanent mit dem Netzwerk verbunden sind. Mobile Endgeräte werden meist sporadisch genutzt, so dass sie nur spontan und kurzfristig untersucht werden können. Eine Überprüfung auf mögliche Infektionen und die Beobachtung von möglicherweise schadhaften Verhaltens kann so nicht permanent durchgeführt werden. Die Überwachung von mobilen Endgeräten setzt folglich voraus, dass diese sich ordnungsgemäß im Netzwerk anmelden, regelmäßig Daten erheben und Ereignisse melden, demnach allgemein als Sensor agieren, um ihren Zustand bewerten zu können.

Ein Großteil der in Unternehmensnetzwerken eingesetzten Monitoring-Systeme basiert in der Regel auf Nagios oder Icinga und deren Abspaltungen, wobei Android-Geräte kaum

Berücksichtigung findet. Auch andere Ansätze, wie IF-MAP innerhalb der TNC-Architektur, sind darauf angewiesen, dass Sensoren speziell für Android-Geräte entwickelt werden, um Daten über diese Geräte zu erheben und in einer Beurteilung zu berücksichtigen.

2 Monitoring-Client

Mobile Endgeräte müssen sich authentisieren, um Zugang zum Netzwerk zu erhalten. Sie müssen jedoch zusätzlich noch integer und vertrauenswürdig im Sinne eines trusted device bleiben, während sie verbunden sind, um weitere Netzwerkkomponenten und das Netz selbst nicht zu gefährden.

Da Android-Geräte permanent Bedrohungen ausgesetzt sind [Ste15] und so gezielt, auch ohne Wissen des Nutzers, eingesetzt werden können, um Angriffe auf Netzwerke durchzuführen, müssen diese Geräte besondere Beachtung in Monitoring-Systemen finden.

Durch die Entwicklung von Monitoring-Clients kann diesem Problem begegnet werden. Diese Clients werden eingesetzt, um Daten über den Zustand des Geräts zu erheben. Dazu zählen beispielsweise Traffic-Statistiken, Auslastung, Nutzung und Versionsnummern der Software, sowie Informationen über die installierten Apps mit ihren Berechtigungen. Damit kann beurteilt werden, ob das mobile Endgerät angemessene Verhaltensmuster aufweist und die Software keine bekannten Sicherheitslücken enthält. Zudem kann das Gerät kurzfristig durch weitere Sensoren im Netzwerk überprüft werden. Die erhobenen Daten werden anschließend durch das SIEM-System analysiert [Elf14, Tru15], um dem Gerät den Zugang zum Netzwerk weiterhin zu erlauben, oder, aufgrund einer Änderung des Zustands, zu beschränken.

Die Trusted Computing Group [Tru14] spezifiziert mit IF-MAP [TCG14] ein Protokoll, welches genutzt werden kann, um nur vertrauenswürdigen Geräten eine Zugangsberechtigung zum Netzwerk zu erteilen und ihnen diese auch zu entziehen, falls das Gerät seine Vertrauenswürdigkeit verliert. Gründe für den Verlust der Berechtigung können dabei beispielsweise die Installation von Schadsoftware sein oder Sicherheitslücken, die das Gerät angreifbar machen.

3 IF-MAP

TNC IF-MAP Binding for SOAP [TCG14] ist Bestandteil der Trusted Network Connect (TNC) Architektur, die durch die Trusted Computing Group [Tru14] (TCG) entwickelt wurde. Bei der TCG handelt es sich um eine internationale Organisation zur Entwicklung von offenen Industriestandards. Die Spezifikation des IF-MAP-Protokolls wurde am 28. April 2008 veröffentlicht und seit dem kontinuierlich bis zur aktuellen Version 2.2r9 vom 26. März 2014 erweitert und angepasst.

3.1 Funktion

Trusted Network Connect ist eine offene Architektur zur Netzwerkzugangskontrolle basierend auf etablierten Standards. Der IEEE 802.1X-Standard mit den Komponenten Access Requestor (AR), Policy Decision Point (PDP) und Policy Enforcement Point (PEP) wird zugrunde gelegt und um einen MAP-Server sowie IF-MAP-Clients erweitert, welche die Sammlung, Erhebung und Auswertung von Daten ermöglichen.

In Abbildung 1 ist die TNC-Architektur mit den genannten Komponenten, sowie die Kommu-

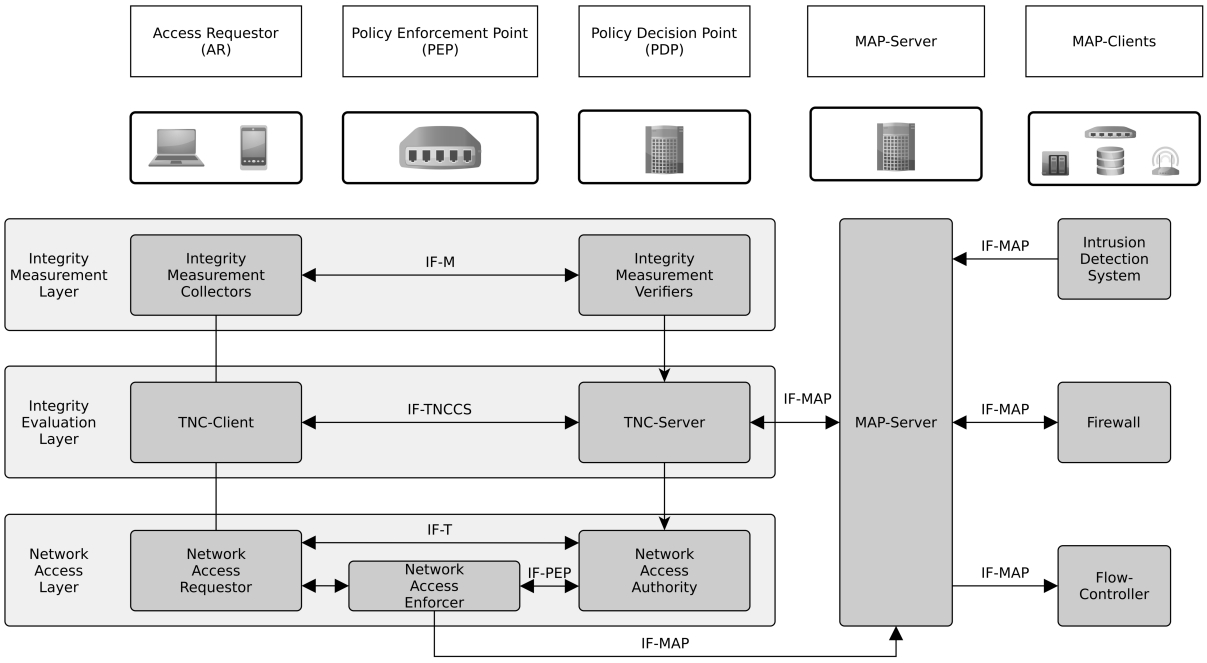

Abb. 1: TNC-Architektur mit IF-MAP, Schichten und 802.1x-Komponenten *basierend auf [TCG14, 13]*

nikationsstruktur dargestellt, welche einen Datenaustausch über das IF-MAP-Protokoll mit den MAP-Server ermöglicht.

Das IF-MAP-Protokoll basiert auf SOAP [ML07] und XML über HTTPS [Res00].

Diese Architektur ermöglicht die Beschränkung des Netzwerkzugangs auf vertrauenswürdige Geräte und sieht das Entfernen von verdächtigen Geräten aus dem Netzwerk, sowie das Verschieben in ein Quarantäne-Netz, vor.

3.2 Anwendung

IF-MAP kann zur Zugangskontrolle genutzt werden, wenn herkömmliche Ansätze, wie Firewalls oder IEEE 802.1X, nicht mehr genügen.

Das setzt voraus, dass von verschiedenen Stellen durch IF-MAP-Clients Metadaten beim MAP-Server veröffentlicht werden. Diese Daten umfassen Integritäts- oder Authentifizierensinformationen, die ausgewertet werden, um den Zugang zu Diensten zu erlauben oder bestimmte Berechtigungen zu entziehen.

Dazu müssen unterschiedliche Netzwerkkomponenten das IF-MAP-Protokoll implementieren, nutzen und als IF-MAP-Client agieren. Sie müssen dem MAP-Graphen, den der MAP-Server vorhält, Daten hinzufügen, diese auswerten, modifizieren und aktuell halten. Deshalb müssen die IF-MAP-Clients während ihrer gesamten Sitzung Metadaten erheben, um eventuelle Änderungen des Zustands publizieren zu können.

Die im MAP-Graphen gesammelten Informationen werden schließlich evaluiert, um bestimmte Geräte auszuschließen oder zu beschränken.

3.3 Datenmodell

Der MAP-Server empfängt Metadaten durch die IF-MAP-Clients und verwaltet diese in einem ungerichteten knoten-markierten Graphen (MAP-Graph), wobei die Knoten *identifier*, die Kanten *links* und *metadata* Erweiterungen zu bestimmten Knoten oder Kanten sind. Die Daten in diesem Modell werden durch das IF-MAP-XML-Schema [TCG12] definiert.

Identifier sind global eindeutige Werte aus einer Menge von Werten, welche einen einheitlichen Namensraum bieten. Dabei gibt es verschiedene identifier-Kategorien und -Typen, um Objekte im Netzwerk zu beschreiben.

Die erste Kategorie sind „Original Identifiers“, welche von jedem MAP-Server und -Client unterstützt werden müssen. Zur zweiten Kategorie gehören die „Extended Identifiers“, welche eine Erweiterung der ersten Kategorie darstellen, wodurch weitere Typen definiert werden können.

Diese Kategorien finden sich auch bei den Annotationen *metadata* wieder, welche Links und Identifiern hinzugefügt werden können: Standard Metadata und Vendor-specific Metadata, wobei letztere Kategorie die erste erweitert.

Links müssen zwingend durch Metadaten annotiert sein und repräsentieren bidirektionalen Beziehungen zwischen genau zwei Identifiern.

Diesem Modell folgend werden Daten über Geräte, Dienste und Systeme innerhalb des Netzwerks erhoben und hinterlegt, um den Zustand der verschiedenen Komponenten zu dokumentieren.

In Abbildung 2 ist ein Ausschnitt eines Map-Graphen visualisiert, welcher die hinterlegten Daten eines Android-Gerätes gemäß des ESUKOM-Datenmodells [ESU15] enthält, die durch einen Android-Client gesammelt und veröffentlicht wurden. Diese Daten können durch weitere IF-MAP-Clients abgerufen und ausgewertet werden, um den Zustand des Gerätes zu beurteilen. Diese Beurteilung kann anschließend zu Änderungen am Graphen führen.

Zur Pflege der Daten im Graphen werden in [TCG14] verschiedene Operationen definiert:

- **Publish** dient dazu Metadaten anzulegen, zu ändern und zu löschen. Ein Anlegen oder Löschen von Identifiern ist dabei nicht direkt möglich, da diese durch die annotierten Links implizit hinzugefügt oder entfernt werden. Nach einer entsprechenden Änderung können MAP-Clients über diese Anpassungen informiert werden.
- **Search** kann genutzt werden, um bestimmte Metadaten vom MAP-Server abzurufen.
- **Subscribe** erlaubt das Abonnieren von Metadaten-Änderungen, die im Zusammenhang mit bestimmten Identifiern stehen.
- **Poll** eröffnet einen zweiten Kommunikationskanal zum MAP-Server, um asynchron Änderungen am Datenmodell abzufragen.

So können die IF-MAP-Clients lesend und schreibend auf den MAP-Graphen als zentrale Informationsstelle zugreifen und sich über Änderungen benachrichtigen lassen, um Zustandsänderungen zu berücksichtigen und Bewertungen abzulegen.

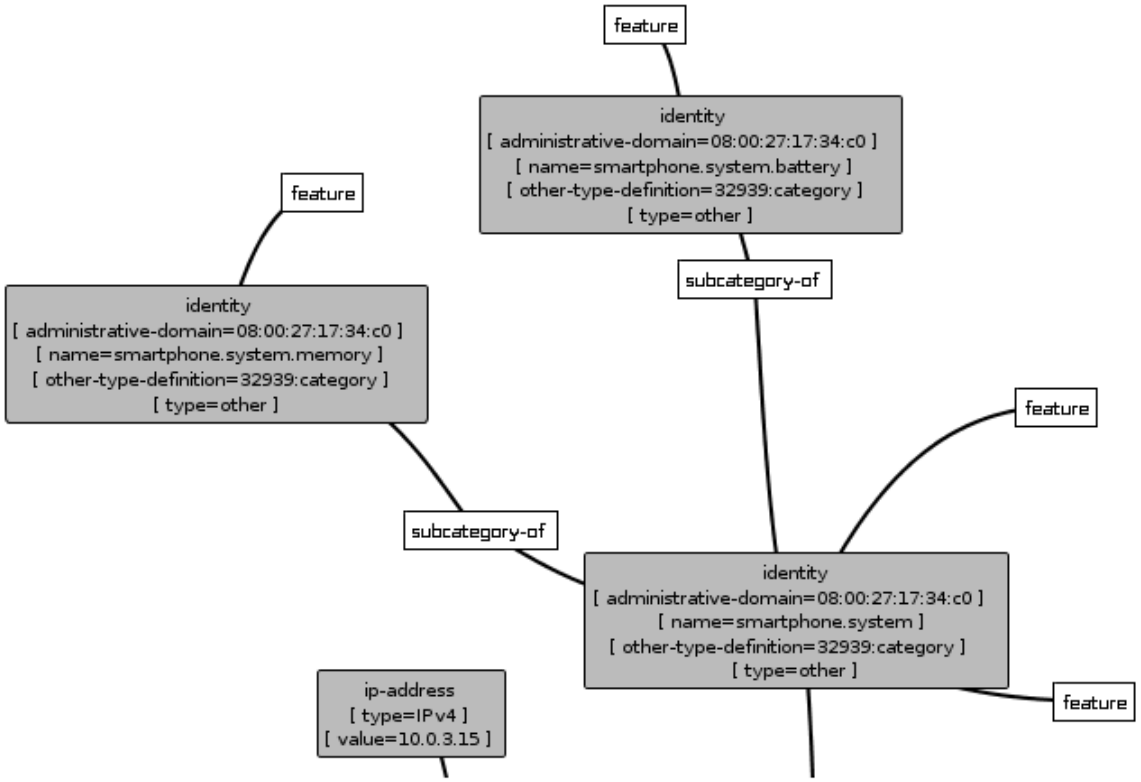

Abb. 2: Android-Daten innerhalb eines IF-MAP-Graphen (Ausschnitt, Rendering: irongui)

3.4 Privatsphäre und Sicherheit

IF-MAP nutzt zur Authentifizierung der IF-MAP-Clients und des MAP-Servers TLS über HTTP, wodurch eine gegenseitige zertifikatsbasierte Überprüfung oder eine Authentisierung mittels HTTP Authentication [FHBH+99] basierend auf RADIUS oder LDAP möglich ist.

Die Nutzung von TLS schützt zusätzlich auch vor verschiedenen Angriffen, wie Replay-, Flooding oder Man-In-the-Middle-Attacken.

Sobald der Client sich jedoch ordnungsgemäß authentisiert hat und Zugang zum Netzwerk erhalten hat, kann er potentiell Schaden anrichten.

Ein manipulierter Client könnte die im MAP-Graphen hinterlegten Daten manipulieren, löschen oder entwenden, um andere Clients anzugreifen oder die Privatsphäre der Nutzer zu verletzen, weshalb es wichtig ist, dass nicht nur Nutzer authentifiziert wird, sondern auch der IF-MAP-Client und die Plattform auf der er läuft überwacht werden.

Ein Mittel zum Schutzt vor manipulierten IF-MAP-Clients und Apps im Allgemeinen bietet Android seit der Version 4.3 (API 18, 24. Juli 2013) mit der Unterstützung von Security-Enhanced Linux [SEL15] im Permissive-Modus und seit Version 5.0 (API 21, 3. November 2014) im Enforcing-Modus. Dadurch wird eine Mandatory Access Control bereitgestellt, um den Schaden, welcher durch Softwarefehler oder Angriffe verursacht werden kann, zu minimieren.

4 Datenerhebung im IF-MAP-Client unter Android

Ein Android-IF-MAP-Client sollte verschiedene Arten von Daten erheben, um mögliche Störfalle zu erkennen:

- Gerät (IMEI, IMSI, . . .)
- Plattform (Firmware-, Kernel-Version, Buildnumber, . . .)
- Systemzustand (CPU-Load, Traffic, . . .)
- Funktechniken (Bluetooth, NFC, . . .)
- Apps (Version, Berechtigungen, . . .)

Diese Daten können helfen Geräte zu identifizieren und ihre Verhalten zu überprüfen. Dabei können bestimmte Geräte von vornherein ausgeschlossen werden, wenn sie beispielsweise eine unbekannte IMEI oder IMSI nutzen.

Einige Daten, wie der Systemzustand, können jedoch nicht ausschließlich bei der Anmeldung im Netzwerk erhoben werden, sondern müssen kontinuierlich gesammelt und weitergegeben, um sinnvoll auswertbar zu sein. Apps dagegen werden meist unregelmäßig de- und installiert, so dass diese Daten nur bei einer Veränderung der erhoben und ausgewertet werden müssen.

Dabei kann durch die Auswertung der installierten Apps, den Versionen und ihren Berechtigungen den mobilen Endgerät die Zugangsberechtigung zum Netzwerk entzogen oder eingeschränkt werden, falls bekannte Schadsoftware installiert wird oder Sicherheitslücken in bestimmten Software-Versionen bekannt werden, wodurch das Gerät angreifbar wird oder eine Bedrohung für andere Komponenten im Netzwerk darstellt. So kann auch vermieden werden, dass Apps genutzt werden, die dem Sicherheitsstandard innerhalb des Netzwerks nicht entsprechen, weil ihre Implementierung oder Kommunikationsinhalte unbekannt sind oder schützenswerte Daten auslesen.

Zusätzlich können Daten erhoben werden, ob beispielsweise bestimmte Funktechniken in besonderen Sicherheitsumgebungen aktiviert sind und somit spezielle Richtlinien missachtet werden, die eine Kommunikationsbeschränkung vorsehen.

5 Sicherheitsaspekte in Android

Die Android-Plattform führt jede App in einer eigenen Sandbox und unter einer eigenen Benutzerkennung aus, um den Zugriff auf fremde Software und Daten zu beschränken. Für einen weiterreichenden Zugriff muss Schadsoftware zunächst aus dieser Sandbox ausbrechen oder das unixoide Rechtesystem umgehen.

Mit der Nutzung von SELinux wurde eine weitere Sicherheitsschicht (Mandatory Access Control) eingeführt, die Prozesse, abhängig von ihrem Kontext, einschränkt. Dabei unterstützt SELinux drei Modi:

- Disabled: vollständig deaktiviert
- Permissive: Warnungen beim Verstoß gegen die Richtlinien
- Enforcing: Durchsetzung der Richtlinien

In der Android-Version 4.4 (API 18) wurde zunächst der Permissive-Modus eingeführt, um anschließend in Version 5.0 (API 21) die Durchsetzung der Richtlinien zu erzwingen (Enforcing).

Dabei verweigert SELinux alle Berechtigungen, welche nicht explizit in den Richtlinien erteilt werden, weshalb diese für die verschiedenen Prozesse und Dienste angelegt und gepflegt werden müssen. Die Android-Plattform liefert bereits eine Vielzahl von Regeln für verschiedene Prozesse und Dienste mit, die durch den Hersteller der Android-Software erweitert werden können, um weitere Domänen abzudecken. Eine Anpassung dieser Regeln durch den Nutzer des Gerätes ist nicht vorgesehen, weshalb dessen Berechtigungen bereits auf ein Minimum beschränkt werden.

Die beiden Konzepte Sandboxing und SELinux sind nur sinnvoll anwendbar, wenn der Nutzer keinen vollständigen Zugriff auf das System hat, da dieser ihm ermöglichen würde die Sicherheitsvorkehrungen zu umgehen.

Eine entsprechende Rechteausweitung resultiert meist aus Softwarefehlern innerhalb der Android-Plattform, weshalb die Nutzung veralteter Version häufig gefährlich ist, vor allem da Sicherheitskorrekturen teilweise nur für aktuelle Android-Versionen zur Verfügung gestellt werden.

Andere Möglichkeiten einen vollständigen Zugriff zu erlangen sind die Modifikation des eingesetzten Bootloaders oder über einen unregulierten physikalischen Zugriff.

Durch einen vollständigen Zugriff auf das System kann, neben der Modifikation des Systems, auch das Verhalten von Apps zur Laufzeit manipuliert werden. Dabei sind die Folgen von Manipulationen oft nicht auf das genutzte Gerät beschränkt, sondern können sich auch auf weitere Systeme auswirken: Ein veränderter IF-MAP-Client könnte beispielsweise falsche Daten hinterlegen oder Daten anderer IF-MAP-Clients verändern, um diese zu diskreditieren.

6 Implementierung

Die beiden Projekte ESUKOM [ESU15] und SIMU [SIM15] nutzen und entwickeln freie Software, um IF-MAP sinnvoll einsetzbar zu machen, wobei auch Android-Geräte berücksichtigt werden.

Innerhalb des ESUKOM-Projekts entstanden so verschiedene Werkzeuge:

- irond (IF-MAP Server)
- Android IF-MAP Client
- IF-MAP-Client (Events verschiedener Dienste: Snort, Radius, LDAP, . . .)
- irondhcp (IF-MAP Client for ISC DHCP)
- irongui (IF-MAP GUI)
- ifmapj (IF-MAP Library for Java)

Diese Komponenten werden im SIMU-Projekt wiederverwendet und überarbeitet, um ein SIEM-System zu entwickeln, welches die drei Ziele „leichte Integrierbarkeit“, „einfache Nachvollziehbarkeit von Ereignissen“ und „geringer Wartungsaufwand“ anstrebt. Dabei werden die aktuellen IF-MAP-Spezifikationen [TCG14] genutzt. Abbildung 3 zeigt ein Strukturszenario dieser Komponenten.

Durch die Nutzung der Komponenten kann bereits eine Infrastruktur auf Basis von IF-MAP eingesetzt werden, die verschiedene Netzwerkkomponenten einbindet, indem Daten über die verwendeten Systeme und Dienste erhoben, im MAP-Graphen abgelegt, abgerufen und ausgewertet werden.

Mit dem aktualisierten Android-Client (DECOmap for Android) ist es dabei möglich die in Abschnitt 4 genannten Daten zu erheben und per IF-MAP zur Verfügung zu stellen, um auf diesen Informationen basierend den Zustand der im Netzwerk verwendeten Android-Geräte zu bewerten und davon abhängig zu behandeln.

Zudem erlaubt die zur Verfügung gestellte ifmapj-Bibliothek eine einfache Implementierung weiterer IF-MAP-Clients, um spezielle Komponenten oder Geräte einzubinden, da eine sinnvolle Nutzung der TNC-Architektur nur möglich ist, wenn entsprechende Komponenten und Dienste eingebunden werden können.

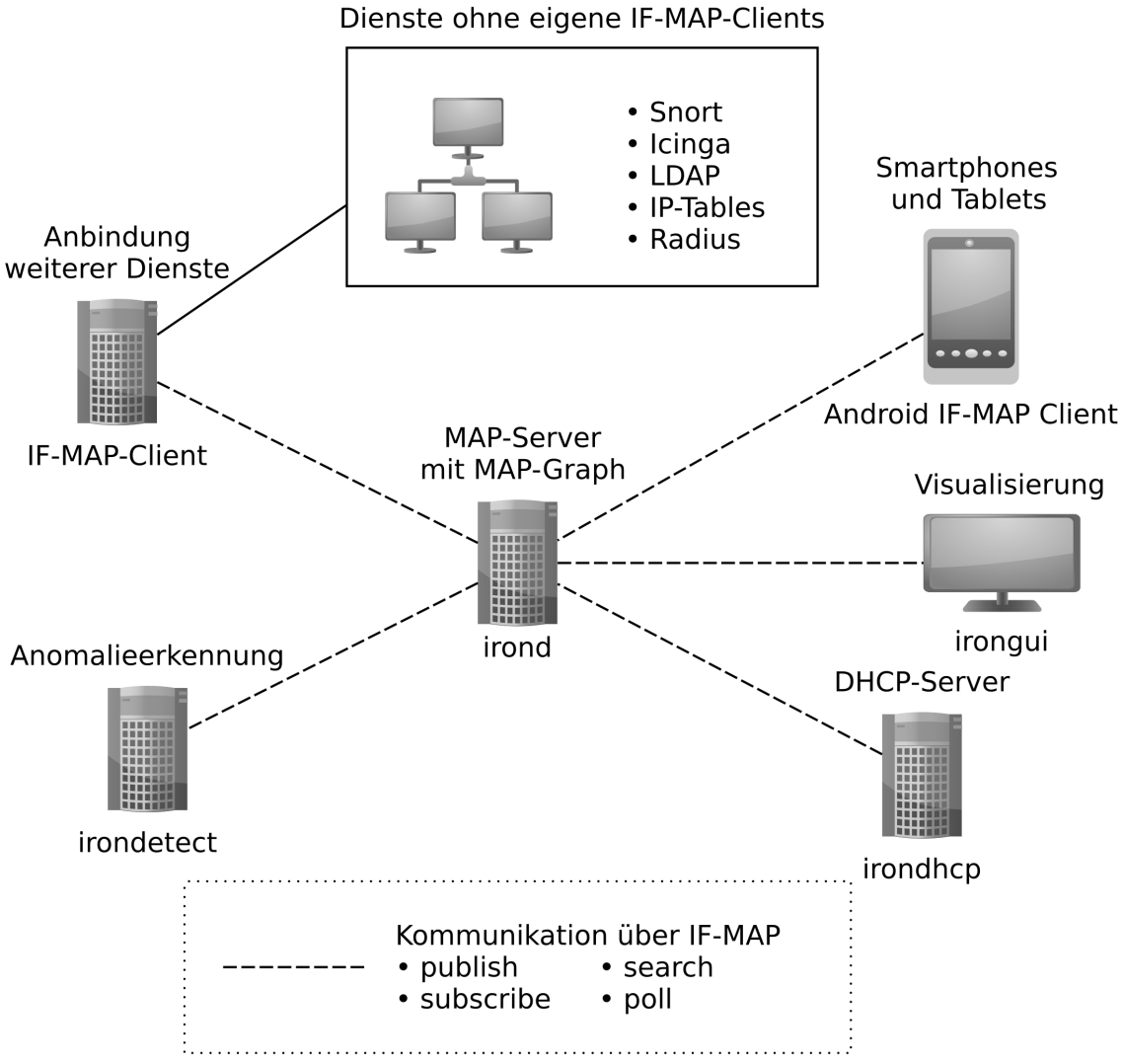

Abb. 3: IF-MAP-Szenario mit ESUKOM-Komponenten

Der DECOmap for Android-Client (siehe Abbildung 4) zeigt die erhobenen Daten und ermöglicht den einfachen Wechsel zwischen verschiedenen Monitoring-Modi (SIMU und iMonitor), zudem werden sind Protokoll-spezifische Eigenschaften zur Kommunikation in den unterschiedlichen Umgebungen einstellbar, wie Authentisierungs- oder Verschlüsselungsmethoden, da die

von IF-MAP bevorzugte zertifikatsbasierte Authentisierung nicht durch das NSCA-Protokoll unterstützt wird.

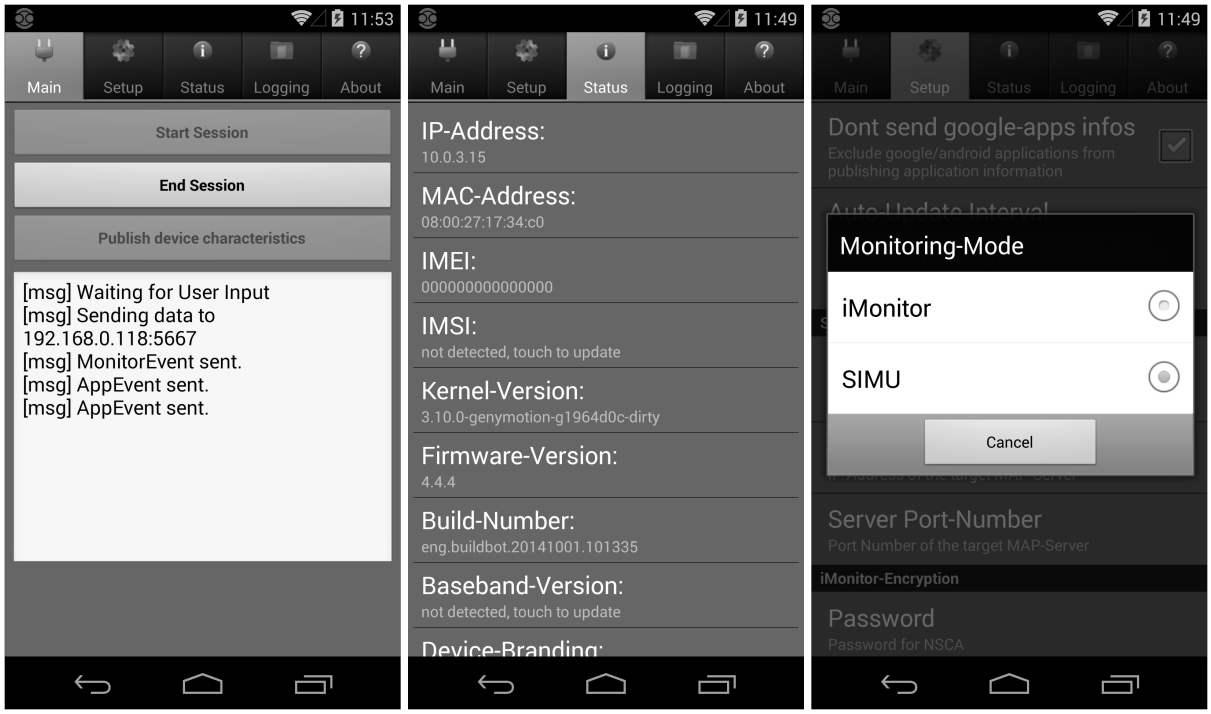

Abb. 4: DECOmap for Android (Screenshot)

7 Lösungsansatz

Ein Problem bei der Nutzung von IF-MAP ist das Nachrichtenformat mit dem die IF-MAP-Clients und der MAP-Server kommunizieren. Der Einsatz von SOAP/XML über HTTPS führt zu Schwierigkeiten bei geringer Netzwerkbandbreite und leistungsschwachen Endgeräten, wovon besonders Android-Geräte betroffen sind, wenn der Akkuverbrauch unter der Nachrichtenverarbeitung leidet.

Das SIMU-Projekt versucht diesem Problem durch den Einsatz von CBOR [BH13] zu begegnen, welches als Datenformat unter anderem geringe Code- und Nachrichtengröße sowie Erweiterbarkeit zum Ziel hat.

Ein CBOR-Proxy vermittelt zwischen den Geräten, die viele Daten generieren oder leistungsschwächer sind, und der reinen IF-MAP-Implementierung, wodurch die durch SOAP/XML verursachten Schwierigkeiten behoben werden können und der Zugang für problematische Geräte vereinfacht wird.

Um eine effiziente Zuordnung der Protokolldaten von CBOR und XML zu erlauben, werden die IF-MAP-Operationen, häufig genutzte Tags und Optionen, auf Zahlen oder Kurzidentifier und die dazugehörigen Argumente durch assoziative Arrays abgebildet, so dass eine Umwandlung in beide Richtungen ermöglicht wird.

Durch die Umsetzung dieses Konzepts kann eine Vielzahl von weiteren Geräten als IF-MAP-Client eingebunden werden, die wegen der genannten Schwierigkeiten nicht oder nur einge-

schränkt genutzt werden konnten, während die bereits eingesetzten IF-MAP-Strukturen kompatibel bleiben. Die Zahl der nutzbaren IF-MAP-Clients ist ein wesentlicher Aspekt, um IF-MAP in Unternehmensnetzwerken sinnvoll einsetzen zu können.

Anschließend müssen die im MAP-Graphen hinterlegten Daten analysiert werden, um Anomalien zu erkennen. Dazu bietet sich die Kombination verschiedener KI-Verfahren, wie Zeitreihenanalyse und Eventkorrelation, an, um mögliche Bedrohungen oder Zustandsveränderungen bei den eingesetzten Geräten zu erkennen und abhängig davon zu reagieren.

8 SIEM ohne IF-MAP

Ein weiterer wichtiger Aspekt ist das Monitoring von mobilen Endgeräten in Umgebungen ohne IF-MAP. Auch dort müssen Daten erhoben, geteilt und ausgewertet werden.

Das Icinga-basierte iMonitor-Projekt [iMo15] setzt auf einen Android-Client (DECOmap for Android, siehe Abbildung 4), welcher auch in IF-MAP-Umgebungen eingesetzt wird und die gleichen Daten über das Gerät erhebt, die auch in IF-MAP genutzt werden. Diese Daten werden in Form von Events über das NSCA-Protokoll (Nagios Service Check Acceptor) an Icinga übertragen.

Dort werden die Daten mit weiteren Informationen und Events aus verschiedenen Quellen, wie Snort oder OpenVAS, aufbereitet und evaluiert, um darauf basierend mögliche Angriffe und Gefahren zu erkennen und Handlungsempfehlungen für Gegenmaßnahmen auszusprechen.

Das Datenmodell zum Austausch der Nachrichten und Events basiert dabei auf JSON [Bra14] als Austauschformat, um eine Vielzahl von Komponenten anzubinden und auch leistungsschwache Sensoren berücksichtigen zu können. Durch den einfachen Aufbau und die große Verbreitung des Formats wird zudem eine Weiterverarbeitung der erhobenen Daten in weiteren Komponenten ermöglicht.

Listing 1 zeigt beispielhaft ein InfoEvent, welches durch den DECOmap for Android-Client generiert wurde, um Informationen über das eingesetzte Gerät weiterzugeben.

Listing 1: Durch DECOmap for Android generiertes InfoEvent

```
{
    "timestamp": "2015-02-17 13:31:21",
    "type": "Android",
    "ipsrc": "10.0.3.15",
    "class": "info",
    "message": "Android device information for 10.0.3.15",
    "data": {
        "mac": "08:00:27:17:34:c0",
        "imei": "000000000000000",
        "imsi": null,
        "kernel": "3.10.0-genymotion-g1964d0c-dirty",
        "firmware": "4.4.4",
        "root": true,
        "selinux": "Disabled",
        "baseband": "",
        "build": "eng.buildbot.20141001.101335",
        "brand": "generic",
        "manufacturer": "Genymotion",
        "cellnumber": null
    }
}
```

Neben dem dargestellten InfoEvent werden durch den Android-Client auch AppEvents und MonitoringEvents erzeugt. AppEvents enthalten Informationen über die installierten Apps mit ihren Versionen und Berechtigungen und werden beim Start der Verbindung sowie spontan bei De- und Installation von Apps generiert und übertragen.

MonitorEvents informieren in festen Intervallen über Traffic-, CPU- und Speicher-Statistiken sowie über die laufenden Prozesse mit ihren spezifischen Eigenschaften.

Der Aufbau der Event-Nachrichten umfasst dabei, wie in Listing 1 dargestellt, die obligatorischen Schlüssel timestamp, type, ipsrc, class und message zur Einordnung des Events und ein data-Objekt zum Transport der Event-spezifischen Daten, wie installierte Apps oder die Auslastung von Komponenten.

Dieses Format lässt sich sowohl für weitere Events speziell für Android-Geräte flexibel anpassen, als auch für andere Komponenten, Dienste oder mobile Endgeräte erweitern, um diese einzubinden und zu berücksichtigen.

9 Fazit und Ausblick

Android-Geräte müssen Daten erheben und diese an die Monitoring-Systeme weitergeben. Dafür bedarf es jedoch zuverlässiger APIs, welche die Daten zur Verfügung stellen, und Apps, die diese Informationen nutzbar machen. Zudem müssen diese Apps zuverlässig mit einer Vielzahl von Geräten und Firmware-Versionen zusammenarbeiten. Dabei ist das genutzte SIEM-Konzept zunächst unerheblich, weshalb Monitoring-Clients sich nicht ausschließlich auf bestimmte Umgebungen, wie IF-MAP oder Icinga konzentrieren, sondern allgemein als Sensor einsetzbar sein sollten, falls bestimmte Monitoring-Strukturen keine realisierbaren Lösungen darstellen.

Anderseits müssen entsprechende Schnittstellen zur Verarbeitung der erhobenen Daten durch das SIEM-System zur Verfügung gestellt werden, welche die speziellen Eigenschaften von mobilen Endgeräten berücksichtigen und entsprechende Konzepte zur Bewertung der erhobenen Daten entwickelt werden, um die Sicherheit und Integrität der im Netzwerk eingesetzten Geräte zu gewährleisten.

Dabei bleibt die Integritätsprüfung bei Smartphones weiterhin ein Problem, denn die in Abschnitt 5 beschriebenen Sicherheitskonzepte und die stetige Überprüfung der Geräte schützen nicht davor, dass das Gesamtsystem ausgehebelt und verändert werden kann (z. B. durch Zugang über die Hardware-Schnittstelle oder Modifikation des Bootloaders), ohne dass die eingesetzte SIEM-Lösung es mitbekommt.

Ein Integritätsschutz könnte beispielsweise mittels TPM-Chip von der TCG und Secure Boot erreicht werden, wodurch einfach festgestellt werden kann, ob ein Gerät und dessen Software verändert wurde oder nicht. Leider fehlt es momentan an solchen Chips in den mobilen Endgeräten (und dem Interesse bei den Herstellern).

Literatur

[BH13] C. Bormann and P. Hoffman. Concise Binary Object Representation (CBOR). RFC 7049 (Proposed Standard), October 2013.

[Bra14] T. Bray. The JavaScript Object Notation (JSON) Data Interchange Format. RFC 7159 (Proposed Standard), March 2014.

[Elf14] Carsten Elfers. *Event Correlation Using Conditional Exponential Models with Tolerant Pattern Matching Applied to Incident Detection (Berichte aus der Informatik)*. Shaker Verlag GmbH, Germany, 2014.

[ESU15] ESUKOM. Echtzeit-Sicherheit für Unternehmensnetze durch Konsolidierung von Metadaten. http://www.esukom.de, February 2015.

[FHBH+99] J. Franks, P. Hallam-Baker, J. Hostetler, S. Lawrence, P. Leach, A. Luotonen, and L. Stewart. HTTP Authentication: Basic and Digest Access Authentication. RFC 2617 (Draft Standard), June 1999. Updated by RFC 7235.

[iMo15] iMonitor. intelligentes IT-Monitoring durch KI-Ereignisverarbeitung. http://www.imonitor-project.de, February 2015.

[ML07] Nilo Mitra and Yves Lafon. Soap version 1.2 part 0: Primer (second edition). World Wide Web Consortium, April 2007.

[Res00] E. Rescorla. HTTP Over TLS. RFC 2818 (Informational), May 2000. Updated by RFCs 5785, 7230.

[SEL15] SELinux Project. Main page. http://selinuxproject.org, February 2015.

[SIM15] SIMU. Security Information and Event Management (SIEM) für Klein- und Mittelständische Unternehmen (KMU). http://simu-project.de, February 2015.

[Ste15] Alastair Stevenson. #v3mobility: Byod will drive android into the enterprise despite security concerns. http://www.v3.co.uk/2396457, February 2015.

[TCG12] TCG Trusted Network Connect. TNC IF-MAP Metadata for Network Security. http://www.trustedcomputinggroup.org/resources/tnc_ifmap_metadata_for_network_security, May 2012.

[TCG14] TCG Trusted Network Connect. TNC IF-MAP Binding for SOAP 2.2 r9. http://www.trustedcomputinggroup.org/files/static_page_files/FF3CB868-1A4B-B294-D093D8383D733B8A/TNC_IFMAP_v2_2r9.pdf, March 2014.

[Tru14] Trusted Computing Group. Home. https://www.trustedcomputinggroup.org, December 2014.

[Tru15] Trust@HsH (SIMU). irondetect. https://github.com/trustathsh/irondetect, March 2015.