computing@computingonline.net
www.computingonline.net

Print ISSN 1727-6209
On-line ISSN 2312-5381
International Journal of Computing

MONITORING ANDROID DEVICES BY USING EVENTS AND METADATA

Markus Schölzel [1], Evren Eren [2], Kai-Oliver Detken [3], Leonid Schwenke [3]

[1] University of Applied Sciences Dortmund, EFS 42, D-44227 Dortmund, Germany,
markus.schoelzel064@stud.fh-dortmund.de, www.inf.fh-dortmund.de
[2] City University of Applied Sciences, Flughafenallee 10 (ZIMT), D-28199 Bremen, Germany,
evren.eren@hs-bremen.de, www.hs-bremen.de
[3] DECOIT GmbH, Fahrenheitstraße 9, D-28359 Bremen, Germany,
detken/schwenke@decoit.de, http://www.decoit.de

Abstract: Mobile devices such as smartphones and tablet PCs are increasingly used for business purposes. However, the trustworthiness of the operating system and apps is controversial. They can constitute a threat to corporate networks and infrastructures, if they are not audited or monitored. The concept of port-based authentication using IEEE 802.1X restricts access and may provide statistical data about users entering or leaving a network, but it does not consider the threat devices can pose if they have already been authenticated and used. Security information and event management (SIEM) software has to incorporate information about mobile devices during their usage. Those devices have to gather and publish information to make this possible. This can be achieved by using a client on the mobile device, which is proposed here. It collects metadata including information about device specific data, platform or system state, which is sent via multiple supported protocols to a central SIEM component, where the data is analyzed in assessment procedures for threat analysis by using artificial intelligence and rule-sets.

Keywords: information security, SIEM, network monitoring, IEEE 802.1X, IF-MAP, trusted network connect, TNC, event detection.

1. INTRODUCTION

Wired and wireless communication networks grow together and service access is becoming more and more ubiquitous, multimodal and standardized solutions are necessary. However, mobile devices and systems pose specific requirements and because of the diversity of network access technologies, the increasing number of services, mobile devices are more vulnerable with respect to IT-security. Therefore, monitoring of mobile devices is a need to increase IT security for enterprise networks.

Monitoring of network devices is vital to infrastructure management. By deploying relevant security information and event management techniques it is possible to monitor networks and respond immediately to malicious events.

These software systems heavily rely on sensors which report incidents, and intelligence to evaluate their importance and implications. However, sensors are typically static like servers, switches, and desktop computers - compared to mobile devices.

Most of the time mobile devices are not permanently connected to the corporate network and only sporadically used. Thus, monitoring has to consider this fact and schedule and perform short or long-time checks accordingly, as they may become targets or used for attacks.

Inspection and monitoring of mobile devices is only feasible if they are properly registered and report data about their status, incidents and generally work as sensors.

Security information and event management (SIEM) systems are seen as an important security component of company networks and IT infrastructures. These systems allow to consolidate and to evaluate messages and alerts of individual components of an IT system. At the same time messages of specialized security systems (firewall-logs, VPN gateways etc.) can be taken into account. However, practice showed that these SIEM systems are extremely complex and only operable with large personnel effort. Often SIEM systems are installed but neglected in continuing operation.

This paper introduces a viable approach for mobile device monitoring and an implementation for Android-based devices by presenting which data can be used and to which extent it is useful to collect data, extending the approach described in [1].

2. PROPOSED SOLUTION

Mobile devices need to be authenticated when connecting to a network, but they also have to stay trustworthy and with integrity during their session.

This can be achieved by collecting their status reporting frequently. These data should comprise traffic statistics, load and usage, information about software and apps including permissions and should be analyzed by monitoring systems to grant or deny network access.

The Trusted Computing Group [2] has developed an open standard, IF-MAP [3], to allow devices to log on and stay connected only, if they are trustworthy. This kind of trust means purity and integrity (free of malicious software). Additionally, IF-MAP can be used for sharing arbitrary metadata across arbitrary entities. Its intended purpose was to enable network devices to share security sensitive information with the goal to integrate arbitrary tools (like NAC solutions, Firewalls, IDS, etc.), thus easing their configuration and extending their functionality.

By implementing and using an app to collect and publish data about mobile devices (chapter 5), SIEM systems are able to perform threat analysis and detect malicious behavior or incidents, which is essential to evaluate the state of a mobile device and for the SIEM system to be able to work as intended.

To work with different kinds of SIEM solutions a format-independent data structure is needed, which can also be extended to be usable for different mobile device platforms and has to be transferable to a well-defined data format for IF-MAP or other protocols (chapter 8).

3. DEFINITION OF SIEM

The acronyms SEM, SIM, and SIEM are often used in the same context, although correctly the term SIEM is a combination of the other two. The first area provides long-term storage, analysis and reporting of log data and is known as *security information management (SIM)*. The second area deals with real-time monitoring, correlation of events, notifications and console views and is commonly known as *security event management (SEM)* [23]. Both areas can be combined differently to set-up a SIEM system.

SIEM technology provides in detail real-time analysis of security alerts, which have been generated by network hardware and applications. SIEM can be used as software, appliances or managed services, and is also used to log security data and generate reports for compliance purposes. The objective of SIEM is to help companies to respond faster to attacks and organize numerous of log data.

The term Security Information and Event Management (SIEM) has been published by Mark Nicolett and Amrit Williams of Gartner in 2005 [24] and describes the product capabilities of gathering, analyzing and presenting information from network and security devices. Further features are identity and access management applications, vulnerability management and policy compliance tools, operating system, database and application logs, and external threat data.

A key focus is to monitor and help manage user and service privileges, directory services and other system configuration changes, as well as providing log auditing and review and incident response.

A complete SIEM system consists of different modules (e.g. event correlation, anomaly detection, identity mapping). These modules are responsible for the “intelligence” of a SIEM system, determining the complexity of events to be detected by the system [22].

4. IF-MAP SPECIFICATION

TNC IF-MAP Binding for SOAP [3] is part of the open *Trusted Network Communications (TNC)*, formerly Trusted Network Connect, architecture developed by the Trusted Computing Group [2] (TCG), which is an international industry standards group.

The first specifications of the standard were published April 28, 2008 (Version 1.0 Revision 25), and continuously updated and improved until March 26, 2014 (Version 2.2 Revision 10), when the latest version was released.

Similar but proprietary and/or vendor-dependent NAC standards are Microsoft’s Network Access Protection (NAP, [27]), which allows interoperation with TNC-compliant technology, and Cisco’s NAC Appliance [28], which has reached its end-of-life.

4.1 BACKGROUND

The *Trusted Computing Group (TCG)* is a not-for-profit organization formed to develop, define and promote open, vendor-neutral, global industry standards, supportive of a hardware-based root-of-trust, for interoperable trusted computing platforms. [2]

Trusted Network Communications (TNC) is the TCG approach for Network Access Control (NAC) solutions. TNC is the reference architecture for NAC that defines the necessary entities and the interfaces through which they are communicating in an interoperable way. IF-MAP is an optional part of the TNC framework.

The following specifications are worked out by the TNC Work Group, their uses can be found in Fig. 1:

a. TNC Architecture for interoperability
b. Integrity Measurement Collector Interface (IF-IMC)
c. Integrity Measurement Verifier Interface (IF-IMV)
d. Trusted Network Communication Client-Server Interface (IF-TNCCS)
e. Vendor-Specific IMC/IMV Message Interface (IF-M)
f. Network Authorization Transport Interface (IF-T)
g. Policy Enforcement Point Interface (IF-PEP)
h. Metadata Access Point Interface (IF-MAP)
i. Clientless Endpoint Support Profile (CESP)
j. Federated TNC

Several main vendors adapted the TNC specification in their network devices, such as:

a. ArcSight (HP)
b. Aruba Networks
c. Cisco Systems
d. Extreme Networks
e. Fujitsu
f. IBM
g. Pulse Secure
h. Juniper Networks
i. McAfee
j. Microsoft
k. Nortel
l. Symantec

TNC was originally a network access control standard with a goal of multi-vendor endpoint policy enforcement. But in 2009 TCG announced expanded specifications which extended the specifications to systems outside of the enterprise network. Additional uses for TNC which have been reported include industrial control system (ICS), SCADA security, [25] and physical security.

In common TNC based on established standards such as IEEE 802.1X, using multiple components including an Access Requestor (AR), Policy Decision Point (PDP) and Policy Enforcement Point (PEP), but also extending this standard by specifying a MAP Server and IF-MAP Clients to collect and evaluate further information for the TNC platform.

As shown in Fig. 1 the MAP Server collects data from different IF-MAP Clients using IF-MAP as protocol, which is based on SOAP [4] and XML over HTTPS [5].

The base specification defines the two roles – client and server – and three different operations: publish, search and subscribe to distribute and access information. In addition, the basic data model is defined, consisting of identifiers (entities) and metadata, which can be attached either to the identifiers directly or connect two identifiers as a kind of relationship, called link. Thereby an undirected information graph originates. Both metadata and identifiers provide specified instruments to use them for arbitrary domains [22].

In summary, this TNC architecture allows rejecting network access to specific devices or push already connected devices, if they are not trustworthy anymore, into a quarantine network or VLAN.

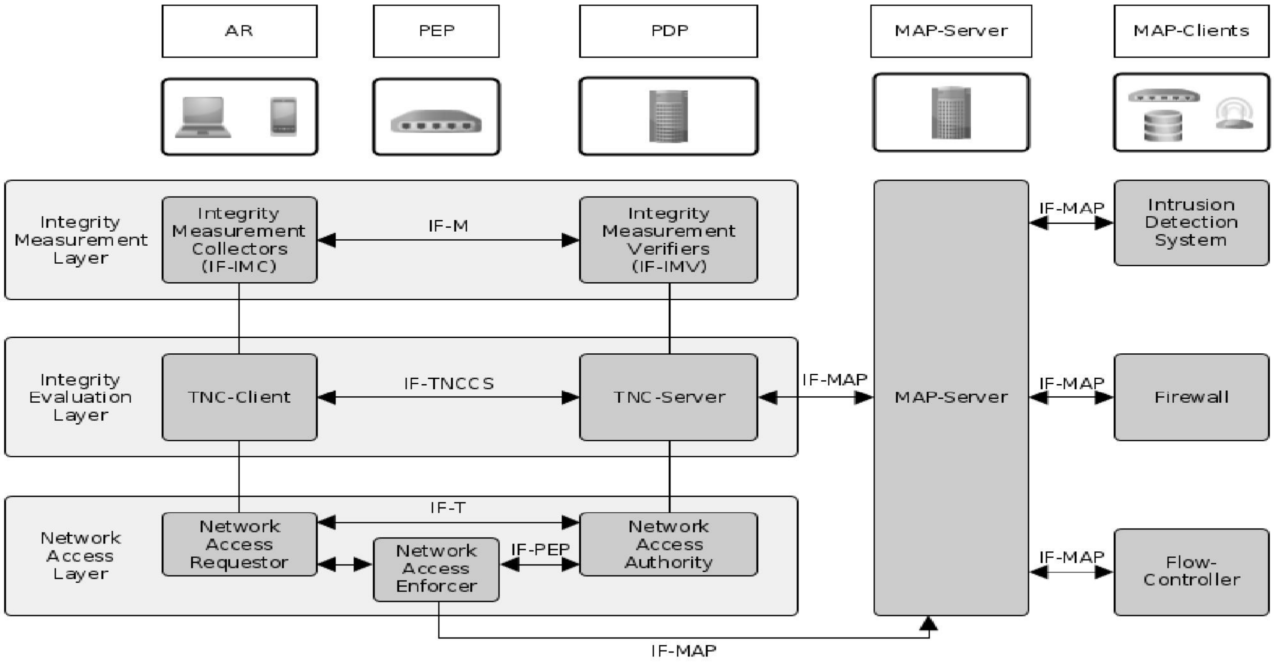


Fig. 1 – TNC architecture including IF-MAP and IEEE 802.1X components (based on [3])

4.2 APPLICATION

IF-MAP can be used to control access to networks and restrict access for untrusted devices, if traditional approaches as firewalls or IEEE 802.1X will not suffice.

For this purpose the standard focuses on specific metadata gathered on the mobile device and published to the MAP Server. These data can include integrity or authentication information, which is evaluated to grant access to other services or revoke certain permissions.

Different network components need to implement and understand IF-MAP, act as IF-MAP Client, to allow those actions. They have to read, modify and publish MAP Graph data. Moreover the devices also need to gather and observe data while being connected to network and not only once, on their first authentication, otherwise their change of state cannot be noticed.

Therefore the design of new applications often requires the definition of additional and domain-specific metadata.

As conclusion IF-MAP can provide the following benefits [22]:

a. Integration of existing security systems by a standardized, interoperable network interface
b. Avoidance of isolated data silos within a network infrastructure
c. Extended functionality of existing security tools (e.g. automatic responses on detected intrusions, identity-based configuration of packet filters)
d. Vendor independence implementation

4.3 DATA MODEL

Metadata plays an important role for securing network applications. The TCG already established specifications providing large amounts of standardized metadata and identifiers useful for network security.

The MAP Server receives data from IF-MAP Clients and maintains them in an undirected, labeled graph (MAP Graph) with links as edges and identifiers as nodes, additional metadata can be attached to them. The associated data types are represented as XML documents [6].

An *identifier* is a globally unique value within a space of values divided in categories to describe diverse objects in a network. There two classes of identifiers: *Original Identifiers* and *Extended Identifiers*. The IF-MAP document [3] defines original identifiers, while the *extended* class is used to augment the *original* class with vendor-specific or other special identifiers.

Same goes with *metadata*, which can be attached to links or identifiers to annotate them with additional information: *Standard Metadata* and *Vendor-specific Metadata*; the latter option extends the first option.

Links are unnamed, bi-directional bindings to represent a relationship between two identifiers. Links must be annotated with metadata to exist and show the relationship between the connected identifiers.

Fig. 2 shows an example graph with identifiers, meta-data and links maintained by a MAP Server.

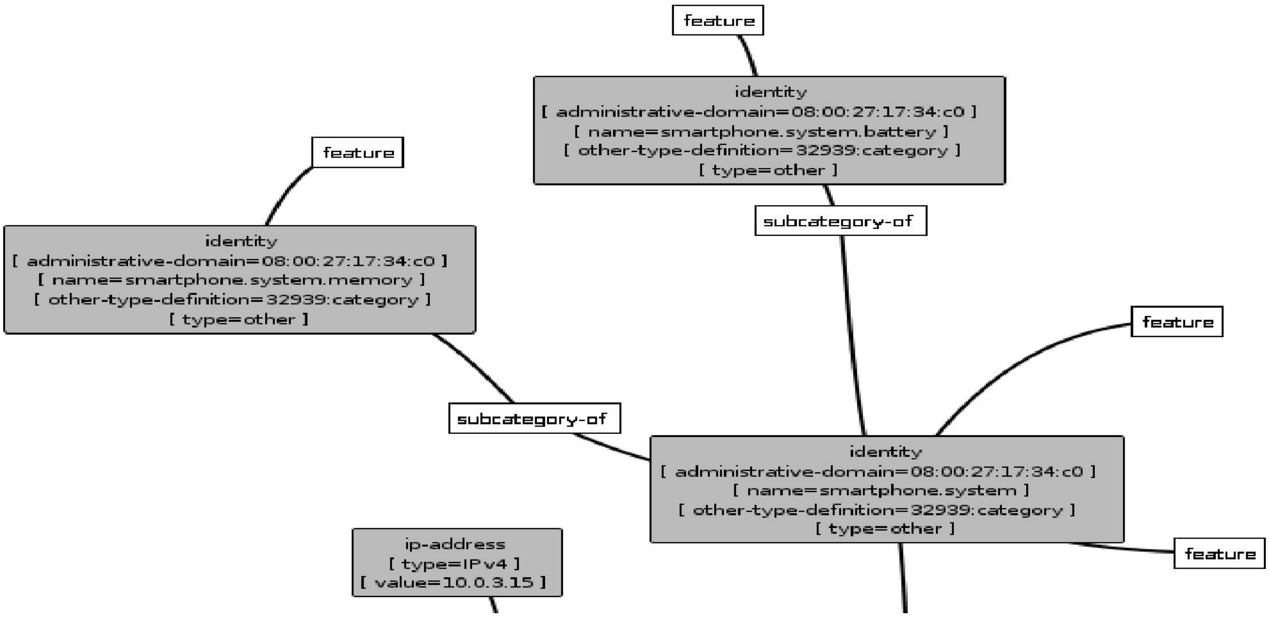

feature
identity
[administrative-domain=08:00:27:17:34:c0]
[name=smartphone.system.battery]
[other-type-definition=32939:category]
[type=other]
feature
subcategory-of
identity
[administrative-domain=08:00:27:17:34:c0]
[name=smartphone.system.memory]
[other-type-definition=32939:category]
[type=other]
feature
subcategory-of
identity
[administrative-domain=08:00:27:17:34:c0]
[name=smartphone.system]
[other-type-definition=32939:category]
[type=other]
ip-address
[type=IPv4]
[value=10.0.3.15]
feature

Fig. 2 – Part of Android metadata in a MAP graph (rendered by irongui [11])

4.4 OPERATIONS

There are several operations needed to interact with the MAP Graph, add/remove/modify/search data or subscribe to changes: publish adds, modifies, removes data and requests the MAP Server to notify subscribers about changes; search explores the MAP Graph; subscribe adds a subscription to specific identifier changes; poll requests MAP Graph updates over an asynchronous channel from the MAP Server.

4.5 AUTHENTICATION

As IF-MAP is carried over HTTP(S), TLS [7] is used to authenticate MAP Server and IF-MAP Clients. Therefore it is possible to use a mutual certificate-based authentication or a basic authentication based on HTTP Authentication [8] with RADIUS [9] or LDAP [10].

4.6 PRIVACY AND SECURITY

The IF-MAP specification even considers attacks and their countermeasures. Most of them (replay, flooding or man-in-the-middle attacks) can be prevented using TLS, but it is easy to cause harm if the client is already inside the network, especially if the client gathers and publishes data to proof himself trustworthy or another client as malicious and harmful. A manipulated client could modify, delete or steal metadata of the MAP Graph to attack other clients or violate their privacy (personification).

Hence, it is important to not only authenticate the user, but also the client software and platform the software is running on.

5. ANDROID DATA

With the purpose of data collection on a variety of Android devices from different manufacturers, an app is used as client to gather information independently from specific models. The basic metadata collected by the client is summarized in Table 1. Extended data acquisition with additional information depends on the device model and manufacturer.

Table 1. Meta data for Android devices

Component	Gathered data
Device specific data	IMEI, IMSI, MAC
Platform	Build number, firmware version, kernel version
Security	SELinux mode and root state
System state	Traffic in/out, load
Communication interfaces	State of bluetooth/NFC and other interfaces
Apps	Installed apps with versions and granted permissions

These metadata contain unique identifiers (e.g. IMEI) to recognize devices by checking them against a database of known devices, which could be used to restrict network access or treat them differently. An assessment of platform data could discover out-dated or unmaintained, but still deployed, firmware versions with known vulnerabilities.

The patch level of the device and information regarding vulnerabilities for the Android version can be determined by inspecting platform-specific information. Hence, considerable amount of knowledge is needed and kept in the background since there are various manufacturers with specific adaptations, versions and back-ported changes, which have to be documented or detected through testing. Also, it should be noted that Android devices often represent an immutable platform, whereby software versions are settled. Decision solely based on the version and patch level could lead to exclusions of whole model generations which are only some few months older.

Another important aspect in mobile device assessment are installed apps, as Android devices allow installation of apps from unknown sources with a variety of different permissions. This introduces security risks as these apps are potentially malicious, but hardly tracked by package management software. So it is necessary to monitor installed apps and their permissions as they could steal confidential data or breach security.

To reduce the impact of individual destructive apps, every app runs in a sandbox and can only access its own data. Malicious apps need to break out of their sandbox to affect other apps or the system, but with *Security-Enhanced Linux (SELinux)* [12], a kernel security module providing an additional mandatory access control (MAC) mechanism, there is another security policy layer restricting system and user processes depending on their context. Android includes SELinux since Android 4.3 (API 18) officially in permissive mode, and since Android 5.0 (API 21) the enforcing mode is supported to minimize the potential damage a bug or an attack can cause by enforcing policies that confine access to resources.

SELinux supports three modes: *disabled*, *permissive* and *enforcing*.

- disabled: no contexts generated
- permissive: logging blocked or granted operations based on the security policy
- enforcing: security policy is enforced

These concepts are useful if the system is not rooted. With root access many safety precautions are obsolete as the system can be tampered with and modified in many ways to pretend to be a trustworthy environment and fake collected data.

The analysis of this information is essential in order to evaluate a mobile device holistically and to issue an extended or restricted set of permissions for a device based on its properties.

In chapter 9 another approach developed by Google is described, to verify the tampering state of Android devices using Google Play Services.

6. IMPLEMENTATION

There are already two projects taking Android devices and IF-MAP into account: ESUKOM [13] and SIMU [14]. Both accomplish IT security in corporate networks while focusing on open source software. The projects even take mobile devices into consideration, because mobile devices can provide a big security issue to a network.

6.1 IF-MAP BASED SIEM PROJECTS

SIMU and iMonitor [15] are development projects in the SIEM domain which consider Android mobile devices as important elements in networks in monitored environments.

SIMU is based on technology developed in the ESUKOM-project and other numerous projects to accomplish a SIEM software application with trusted components and services using IF-MAP. Tangible results of SIMU are the development of open-source software components based on IF-MAP in open-source and commercial products for network environments. These components include among others:

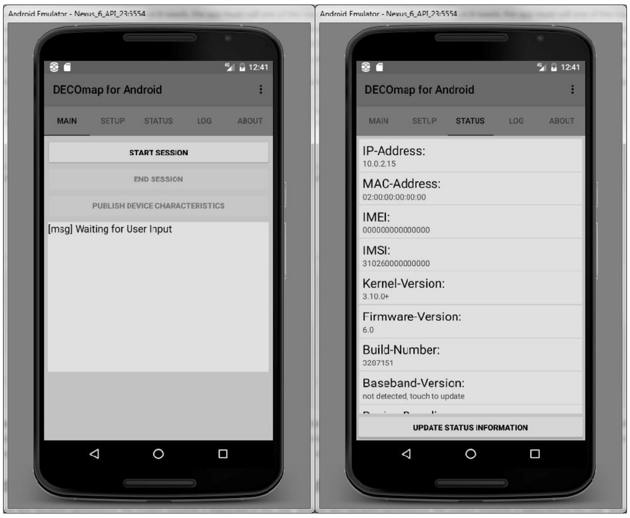
- irond (IF-MAP Server)
- irongpm (Graph-Pattern-Matching Engine)
- DECOmap IF-MAP Client (IF-MAP client based on Java to publish log information from multiple components and services)
- ifmapj (IF-MAP library for Java)

Using these components allows the implementation of a monitored network infrastructure using the latest IF-MAP specifications [3] with multiple network components and services. The data format used by DECOmap for Android is compatible with the ESUKOM/SIMU components and allows the monitoring of Android mobile devices in IF-MAP environments.

The goal of the project iMonitor is to provide a SIEM software, based on Icinga, for environments, where IF-MAP is not applicable. Icinga [20] and Nagios [19] are common tools for network monitoring, but do not consider Android mobile devices nor SIEM aspects with their specific properties and behavior in networks. By deploying DECOmap for Android as client and gathering data from mobile devices over *NSCA (Nagios Service Check Acceptor)* [26], iMonitor is able to perform its assessment procedures and threat analysis using artificial intelligence and rule-sets [16].

6.2 DECOMAP FOR ANDROID CLIENT

Fig. 3 shows the *DECOmap for Android* client [17] which makes it possible to integrate mobile Android devices into the IF-MAP structure. To include the security factor of mobile devices, the named projects integrate the DECOmap for Android Client into their concept. The client collects data from mobile Android devices to open up the possibility of analyzing them for possible security issues by evaluating published data using event correlation and artificial intelligence.


Android Emulator - Nexus_6_API_23:5554
12:41
DECOmap for Android
MAIN
SETUP
STATUS
LOG
ABOUT
START SESSION
END SESSION
PUBLISH DEVICE CHARACTERISTICS
Android Emulator - Nexus_6_API_23:5554
12:41
DECOmap for Android
MAIN
SETUP
STATUS
LOG
ABOUT
UPDATE STATUS INFORMATION

[msg] Waiting for User Input

IP-Address:
10.0.2.15

MAC-Address:
02:00:00:00:00:00

IMEI:
000000000000000

IMSI:
310260000000000

Kernel-Version:
3.10.0+

Firmware-Version:
6.0

Build-Number:
3287151

Baseband-Version:
not detected, touch to update

Fig. 3 – Screen examples of the App DECOmap for Android

To support both projects, the DECOmap for Android client can send two different kinds of metadata structures. Each project including mobile devices in its own way, that is why they are expecting different data formats. While SIMU depends on the official IF-MAP standard and only sends out the devices name and IP address, ESUKOM defines its own metadata, identifier and format extensions, to support deeper analysis of the device itself. Therefore it is necessary for the client to collect the in chapter 5 mentioned data. Within the status-tab of the client, it's possible to view the state of the collected data.

Besides the monitoring over IF-MAP based environments, it is possible for the client to run in iMonitor mode. This mode enables the possibility to communicate over NSCA with an iMonitor server. Chapter 8 provides more information about the monitoring project iMonitor [15] and its procedure of handling data. The named processes are implemented in the DECOmap for Android client and can be used as alternative to IF-MAP.

For a better illustration of the processes and structure Fig. 4 shows a communication flow chart of an example scenario using IF-MAP. At first the user starts the application. After setting up the configuration to the desired functionality, the user can start the connection to the data server. Depending on the configuration the user can now manually or automatically initialize the data publication. In this case the data is published automatically. After a configured time, a process starts to collect the latest required data from the device and parses them into an IF-MAP event. At last the event is send to the IF-MAP server through the established communication channel. After a given delay, the publication process is repeated. On the server side the data is analyzed and correlation takes place. If a security problem occurs, the server is able to take countermeasures and disconnect the client from the network or push him into a quarantine network or VLAN.

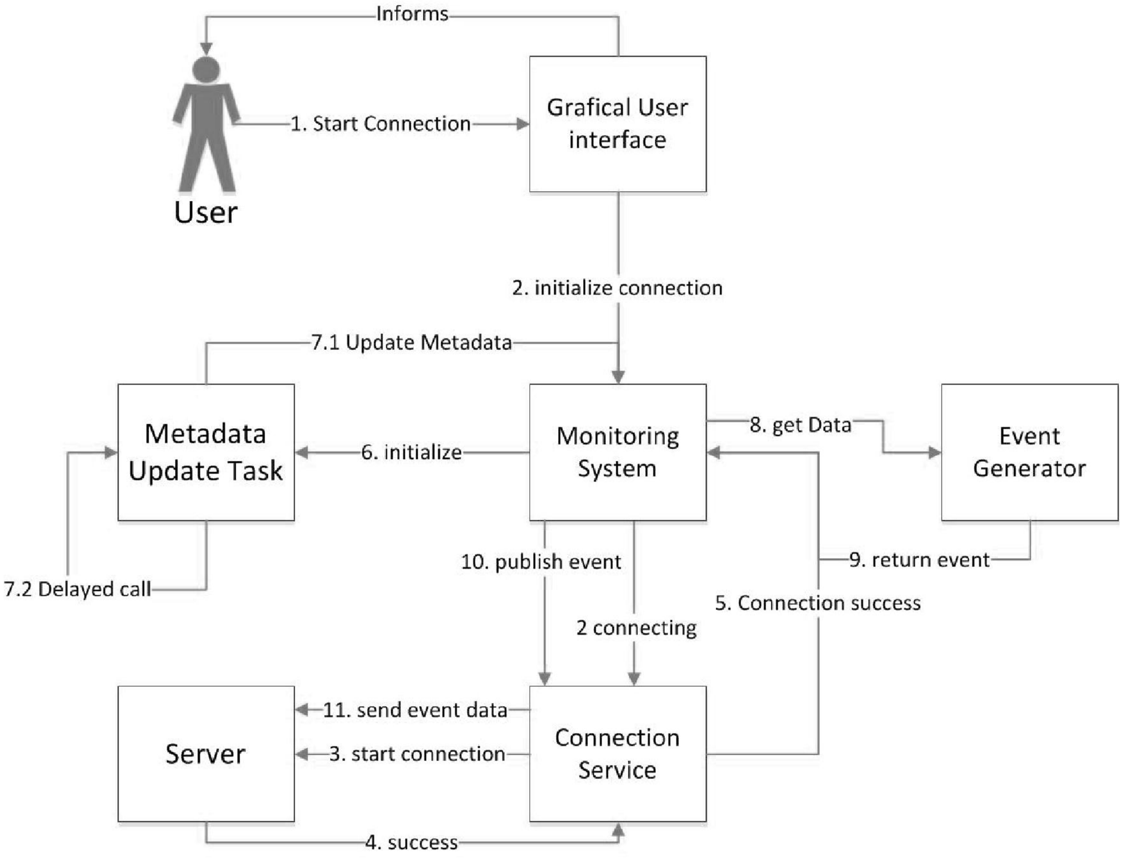


Fig. 4 – Communication flow chart for DECOmap for Android client

To summarize the functionality of the DECOmap for Android client: Internal data is regularly collected from the mobile device. Collected data is parsed afterwards into the desired publishing format. Finally, the parsed data is send to the server, where it is being analyzed for security issues.

With *ironcontrol* [29] there is another Android client for IF-MAP structures available, which pursues a different goal. By allowing the execution of IF-MAP operations (section 4.4) metadata can be published, searched and retrieved on mobile devices. So ironcontrol supports IF-MAP on Android, but only in the sense of adding, viewing and changing data. There is no collection of data on the mobile device, which is needed to allow an assessment.

7. ISSUES/IMPROVEMENTS

IF-MAP sends its messages over SOAP/XML and HTTPS, which causes issues in environments with bandwidth or resource limitations. This is particularly a problem for the low-end Android devices, where the needed processing power causes high battery drain in the background, and also for components, which have to keep the MAP Graph updated by frequently generating many messages.

SIMU tries to tackle this problem by using *CBOR (Concise Binary Object Representation*) [18] as data exchange format, which is focusing on multiple design goals including compactness of encoders and decoders, compact message sizes, and extensibility while being applicable on low-end devices and to high-volume applications.

A CBOR proxy, between devices with limited resources or services generating an extensive number of events and pure IF-MAP implementations, constitutes a viable alternative to SOAP/XML, which can help to address performance problems and facilitate the usage of IF-MAP. That is important for small bandwidth scenarios with mobile devices especially.

It allows increasing the number and variety of devices usable as IF-MAP Clients while staying compatible with the IF-MAP structure already in use by mapping IF-MAP commands and parameters to numerals, short identifiers and associative arrays. This solves protocol-based issues and allows connecting a numerous variety of clients, which are too low-end or too high-volume. Solving those issues is a necessary task as the number of IF-MAP Clients or IF-MAP-enabled data sources is essential to make IF-MAP useful in corporate networks.

8. MONITORING WITHOUT IF-MAP

IF-MAP is not always a viable solution for existing network environments. Thus, it is essential to consider specific use cases.

Nagios [19] and Icinga [20] are widespread monitoring tools which rely on sensors for event detection and data acquisition with respect to system state, performance, and threat analysis. NSCA is used as communication protocol between sensors and the NSCA server instance used by Nagios/Icinga to collect data.

DECOmap for Android supports NSCA as sensor for mobile devices in Nagios/Icinga environments. Similar to the IF-MAP approach data is gathered on devices and published to a central NSCA service instance.

iMonitor uses Icinga as monitoring component and defines three event message classes for mobile devices: InfoEvent, MonitorEvent and AppEvent. These event messages are triggered at configurable time periods and on event detection. Listing 1 depicts an example of an AppEvent generated by DECOmap for Android formatted as JSON string. JSON strings are used in iMonitor to encode messages which are encapsulated in NSCA messages.

InfoEvents contain device specific data, which do not change during a network session. They are published at initial network access and can be used to recognize a specific device solely based on IMEI or a class of devices based on the combination of manufacturer, kernel, firmware and build version.

MonitorEvents contain data about frequently changing properties (traffic going in and out, load and running processes with their properties). These events are published at configurable time periods to determine the state of the monitored to device.

```
{
 "timestamp": "2015-02-17 13:31:22",
 "type": "Android",
 "ipsrc": "10.0.3.15",
 "class": "apps",
 "message": "Android application information for 10.0.3.15",
 "data": {
  "name": "com.android.inputmethod.latin",
  "label": "Android Keyboard (AOSP)",
  "version": "4.4.4-eng.buildbot.20141001.101335",
  "running": true,
  "installTime": 1412151689000,
  "updateTime": 1412151689000,
  "permissions": [
        "android.permission.ACCESS_NETWORK_STATE",
        "android.permission.DOWNLOAD_WITHOUT_
                          NOTIFICATION",
        ...
  ]
 }
}
```

Listing 1 – AppEvent generated by DECOmap for Android

AppEvents are published at initial network access and triggered on app installation changes: if an app gets updated or installed, multiple AppEvents are generated and published to update saved record data and to allow an evaluation of the system based on the new apps and their provided properties: name, permissions, version, install and update time, and running status (e.g. in *background/foreground* or only *perceptible* to some degree by the user).

Received events and messages from multiple clients and components should not be regarded as single events but considered holistically in order to support artificial intelligence allowing the assessment of devices and network components and security evaluation.

9. OTHER APPROACHES AND FUTURE WORK

Assessment of mobile devices is not solely restricted to network environments, but also quite important in multiple other contexts. Google features SafetyNet attestation [21] to allow verification of a devices current tampering state and its compatibility with the defined security policy of a specific app. In this context (non-)tampered means policy compliance, if the environment is safe to run an app on the device. It does not check for known bugs or vulnerabilities in the system, but for the root state, the presence of specific apps and changes to files owned by the system (not user). This check is often used by payment apps, but could also be integrated in DECOmap for Android as additional flag for trust. The disadvantage, beside the absence of vulnerability checking, is the dependency on Google Play Services, which are pre-installed on all official Android devices, but remain proprietary. Which files and apps are checked, as well as the changing "tampering criteria", are not publicly documented.

Another approach to evaluate a device is to check for modification based on the state of the bootloader. Many manufactures implement hardware or software locks to confine the access to the devices hardware and software protect the devices from tampering and destruction. An API needs to be developed which allows abstraction from manufacturer- and model-specific implementations to be useful for clients similar to DECOmap for Android, allowing a simple but reliable integration.

Trusted Platform Module (TPM) chips from TCG or secure boot could also be added as properties for network environments with or without IF-MAP to verify the integrity of a device, but these chips are not yet widely deployed in mobile devices.

10. CONCLUSION

The key feature of a SIEM system is to correlate sensor data efficiently and usefully to find out which event is an anomaly and which is not. Therefore it is necessary to analyze a data basis of the same format. IF-MAP can handle this with its extensible metadata definition.

Therefore this integration concept based on metadata, events, and the implementation of DECOmap for Android. Both research and development projects iMonitor and SIMU provide the feature to monitor mobile devices and detect incidents in architectures without and with IF-MAP, because IF-MAP is not always applicable.

The TNC architecture using IF-MAP as protocol for communication is a concept with the goal of collecting and evaluating metadata across multiple clients to create a trusted infrastructure. This approach may be less common, but offers a wide variety of opportunities and a flexible way to correlate information and interact between individual network hardware components and applications from different manufacturers.

Nagios/Icinga based architectures are very common and already established. They are often easier to integrate, but individual network components are barely interconnected, therefore automatic configuration changes based on detected abnormal behavior from clients or interaction between hardware components and applications is laborious.

Nagios/Icinga and IF-MAP based approaches both have their use cases, so network administrators have to consider the different approaches, but regardless of the used systems and their different techniques mobile devices have to be taken into account, therefore Android devices need to gather and share data with the systems independent from specific techniques.

Thus, it is more important to have Android clients as sensors than to focus on specific protocols or technologies, although building a trusted network based on the IF-MAP standard is a viable solution.

These presented concepts to integrate Android in SIEM environments could be adopted in other projects and solutions as mobile devices need to be taken into account regardless of the used SIEM systems.

For this purpose, well-maintained apps and APIs are necessary, which work reliable on multiple Android versions independent from specific manufacturers and hardware components allowing comprehensive data collection and communication with monitoring system using multiple protocols.

Then again this information has to be evaluated and incorporated for monitoring by means of SIEM systems with or without IF-MAP support to allow an assessment not limited to credential verification on login, but holistically, based on system state, behavior and usage, within the network throughout the session.

11. REFERENCES

[1] M. Schölzel, E. Eren and K.-O. Detken, "A viable SIEM approach for Android," in *Proceedings of the IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)*, Warsaw, Poland, 2015, pp. 803-807.

[2] Trusted Computing Group, 2016, [Online]. Available: https://www.trustedcomputing group.org

[3] TCG Trusted Network Connect, *TNC IF-MAP Binding for SOAP 2.2 r10*, 2014, [Online]. Available: http://www.trustedcomputinggroup.org/wp-content/uploads/TNC_IFMAP_v2_2r10.pdf

[4] N. Nitra and Y. Lafon, *SOAP version 1.2 part 0: Primer (second edition)*, 2007, [Online]. Available: http://www.w3.org/TR/soap12/

[5] E. Rescorla, *HTTP Over TLS (RFC 2818)*, 2000, [Online]. Available: http://www.ietf.org/rfc/rfc2818.txt

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

[7] T. Dierks and E. Rescorla, *The Transport Layer Security (TLS) Protocol Version 1.2 (RFC*

5246), 2008, [Online]. Available: http://www.ietf.org/rfc/rfc5246.txt

[8] 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)*, 1999, [Online]. Available: http://www.ietf.org/rfc/rfc2617.txt

[9] C. Rigney, S. Willens, A. Rubens and W. Simpson, *Remote Authentication Dial In User Service (RFC 2865)*, 2000, [Online]. Available: http://www.ietf.org/rfc/rfc2865.txt

[10] K. Zeilenga, *Lightweight Directory Access Protocol (LDAP): Technical Specification Roadp Map (RFC 4510)*, 2006, [Online]. Available: http://www.ietf.org/rfc/rfc4510.txt

[11] Trust@FHH, *irongui*, 2015 [Online]. Available: https://github.com/trustathsh/irongui

[12] SELinux Project, 2016, [Online]. Available: https://selinuxproject.org

[13] ESUKOM, *Echtzeit-Sicherheit für Unternehmensnetze durch Konsolidierung von Metadaten*, 2016, [Online]. Available: http://www.esukom.de

[14] SIMU, *Security Information and Event Management (SIEM) für Klein- und Mittelständische Unternehmen (KMU)*, 2016, [Online]. Available: http://simu-project.de

[15] iMonitor, *intelligentes IT-Monitoring durch KI-Ereignisverarbeitung*, 2016, [Online].

[16] C. Elfers, *Event Correlation Using Conditional Exponential Models with Tolerant Pattern Matching Applied to Incident Detection*, Shaker Verlag GmbH, Aachen, 2014, 279 p.

[17] *DECOmap for Android*, 2015, [Online]. Available: https://github.com/decoit/Android-IF-MAP-Client

[18] C. Bormann and P. Hoffman, *Concise Binary Object Representation (RFC 7049)*, 2013, [Online]. Available: http://www.ietf.org/rfc/rfc7049.txt

[19] Nagios Enterprises, *Nagios*, 2016, [Online]. Available: https://www.nagios.org

[20] The Icinga Project, *Icinga*, 2016, [Online]. Available: https://www.icinga.org

[21] Google Developers, *SafetyNet - Google APIs for Android*, 2016, [Online]. Available: https://developers.google.com/android/reference/com/google/android/gms/safetynet/SafetyNet

[22] K.-O. Detken, D. Scheuermann, B. Hellmann, "Using Extensible Metadata Definitions to Create a Vendor-Independent SIEM System," in *Advanced in Swarm and Computational Intelligence*, Proceedings Part II, Editors: Y. Tan, Y. Shi, F. Buarque, A. Gelbukh, S. Das, A. Engelbrecht, ISBN 978-3-319-20471-0, publishing house Springer, Beijing, China, 2015, pp. 439-453.

[23] A. Jamil, *The difference between SEM, SIM and SIEM*, 2010, [Online]. Available: http://www.gmdit.com/NewsView.aspx?ID=9IfB2Axzeew=

[24] A. Williams, *The Future of SIEM – The market will begin to diverge*, 2007, [Online]. Available: https://techbuddha.wordpress.com/2007/01/01/the-future-of-siem-—-the-market-will-begin-to-diverge/

[25] S. Howard, *Securing SCADA and Control Networks*, 2010, [Online]. Available: http://www.automation.com/automation-news/article/securing-scada-and-control-networks

[26] The Icinga Project, *Nagios Service Check Acceptor (NSCA)*, [Online]. Available: http://docs.icinga.org/latest/en/nsca.html

[27] Microsoft, *Network Access Protection (NAP)*, [Online]. Available: https://technet.microsoft.com/en-us/library/dd125338(v=ws.10).aspx

[28] Cisco, *NAC Appliance (Clean Access)*, [Online]. Available: http://www.cisco.com/go/nac

[29] Trust@FHH, *ironcontrol*, 2015 [Online]. Available: https://github.com/trustathsh/ironcontrol-for-Android

Markus Schölzel *wrote his thesis in cooperation with DECOIT GmbH working on publicly funded projects and graduated with a Bachelor of Science (B.Sc.) from the University of Applied Sciences in Dortmund in 2015. Since 09/2015 he is working towards his master's degree in Computer Science.*

Prof. Dr.-Ing. Evren Eren, *graduated from the University of Bremen as an Electronics Engineer in 1988 and started at Krupp Atlas Elektronik, working within the marine division as a Software Engineer. In 1992 he changed to the Bremen Institute for Industrial Technology and Applied Work Science (BIBA), where he worked as research scientist in EU funded projects. 1998 he obtained his PhD degree and moved to DETECON as Senior Consultant. Between 09/1999 and 08/2016 he was professor at the University of Applied Sciences in Dortmund. Since 09/2016 he is professor at the City University of Bremen. His working and research areas encompass IT-security and communication networks.*

Prof. Dr.-Ing. Kai-Oliver Detken, *graduated from the University of Bremen as an Electronics Engineer in 1993. After study he worked from 1993 till 1997 at the institute BIBA in Bremen as research scientist in EU-funded projects. In 1998 he changed to the company OptiNet GmbH to manage industrial projects in his professional areas. In 2001 he founded his own company DECOIT GmbH and work as docent for Computer Science at the University of Applied Science in Bremen simultaneously. In 2003 he obtained his PhD degree and got from the University of Applied Science in Bremen the title professor in 2008. His working and research areas includes networks, Internet technologies, Voice over IP, and IT security.*

Leonid Schwenke *works as software engineer at DECOIT GmbH and took part on different research projects in the field of mobile computing, IT security, and trusted computing. He graduated in 2016 with a Bachelor of Science (B.Sc.) from the University of Bremen. His thesis was written in the subject area of cognition, where he researched a self-correcting system on a mobile computing basis. Currently he works on his master's degree in Computer Science with a focus on Artificial Intelligence, Cognition and Robotic.*