The 8th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications
24-26 September 2015, Warsaw, Poland

A viable SIEM approach for Android

Markus Schölzel[1] , Prof. Dr. Evren Eren[1] , Prof. Dr. Kai-Oliver Detken[2]
[1] University of Applied Sciences Dortmund, EFS 42, D-44227 Dortmund, Germany,
Email: markus.schoelzel064@stud.fh-dortmund.de, evren.eren@fh-dortmund.de
[2] DECOIT GmbH, Fahrenheitstraße 9, D-28359 Bremen, Germany,
Email: detken@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 operating systems 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 that devices can pose when already authenticated and used. Mobile devices gather and publish information. This information is incorporated into Security Information and Event Management (SIEM) software so that a threat is recognized while the device is being used.**

***Keywords* – information security; SIEM; network monitoring; IEEE 802.1X; IF-MAP, trusted network connect; TNC; event detection**

I. Introduction

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 use artificial 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. As they may become targets or be used for attacks, monitoring has to consider their frequent use by scheduling and performing short or long-time checks accordingly.

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.

II. Proposed Solution

Mobile devices need to be authenticated when connecting to a network, but they also have to stay trustworthy and prove 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 [1] has developed an open standard, IF-MAP [2] 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).

III. IF-MAP

TNC IF-MAP Binding for SOAP [2] is part of the open Trusted Network Connect (TNC) architecture developed by the Trusted Computing Group [1] (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 9), when the latest version was released.

A. 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." [1]

Trusted Network Connect is an open architecture for network protection 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 a IF-MAP as protocol, which is based on SOAP [3] and XML over HTTPS [4].

This architecture allows rejecting network access to specific devices or push already connected devices, which are not trustworthy anymore, into a quarantine network or VLAN.

B. 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 do not suffice.

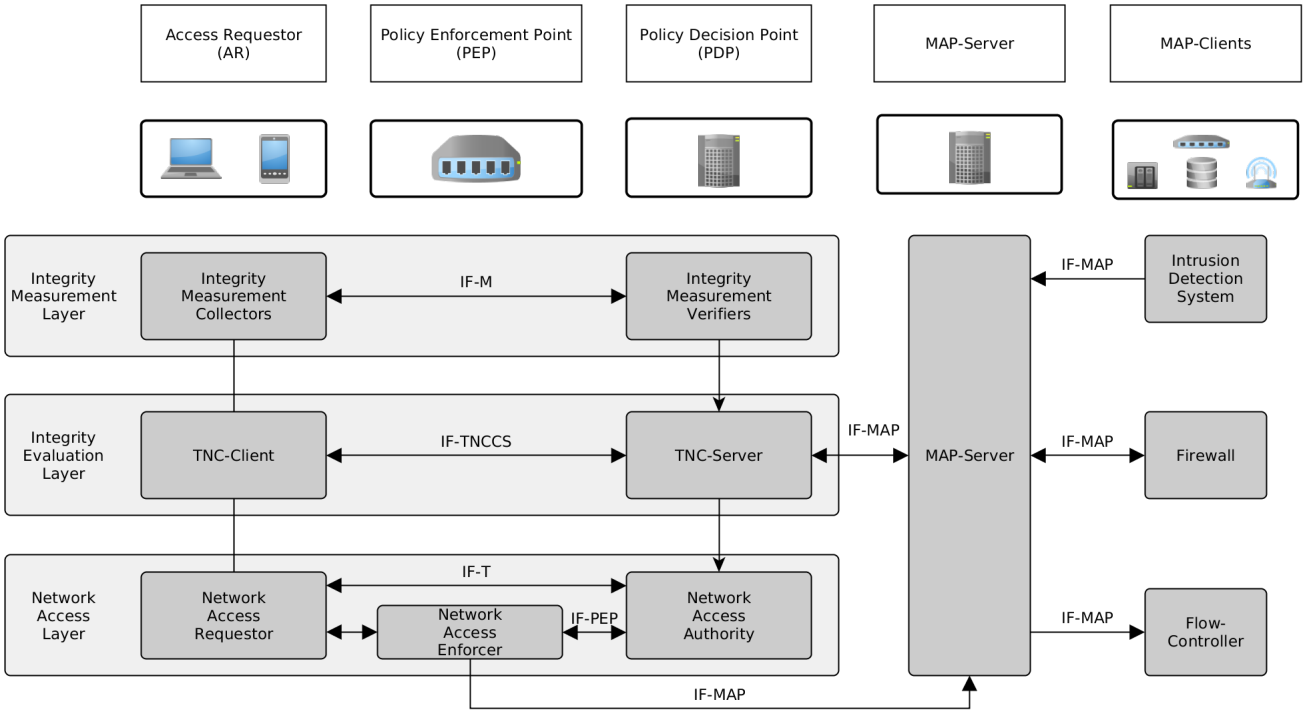

Figure 1. TNC architecture including IF-MAP, layers and 802.1x components *based on [2, 13]*

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 implement and understand IF-MAP, act as IF-MAP Client, to allow those actions. They read, modify and publish MAP Graph data. Additionally, devices gather data not only once at first authentication, but throughout the whole session. Otherwise their change of state cannot be noticed.

C. Data model

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 [5].

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 [2] defines original identifiers, while the *extended* class is used to augment the *original* class with vendor-specific or other special identifiers.

The same applies to *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, metadata and links maintained by a MAP Server.

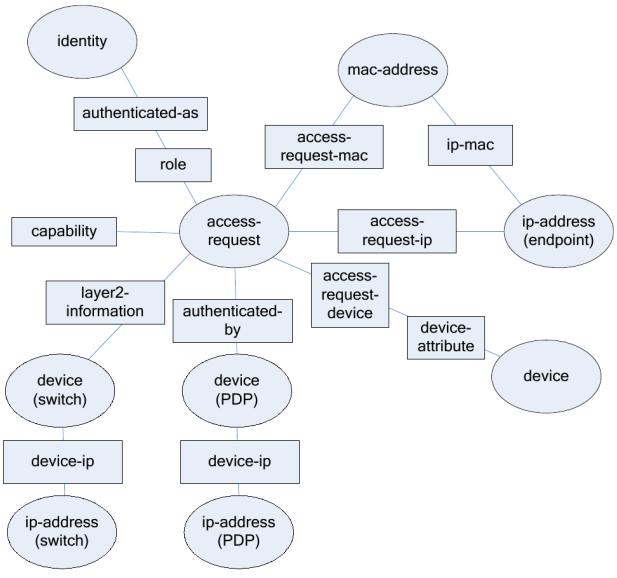

Figure 2. IF-MAP Graph with identifiers, metadata and links [2]

D. Operations

There are several operations needed to enable interaction between IF-MAP Clients and 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.

E. Authentication

While IF-MAP is carried over HTTP(S), TLS [6] is used to authenticate MAP Server and IF-MAP Clients. Therefore, it is possible to use mutual certificate-based authentication or basic authentication based on HTTP Authentication [7] with RADIUS [8] or LDAP [9].

F. 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 when the client is already inside the network, as he could gather and publish data to proof himself or another client as malicious or harmful. A manipulated client could modify, delete or steal metadata of the MAP Graph to attack other clients or violate their privacy (impersonification).

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

Android includes Security-Enhanced Linux [10] since Android 4.3[1] officially in permissive mode. Since Android 5.0[2] the enforcing mode is supported to minimize the potential damage a bug or an attack can cause.

IV. Android Data

An Android client gathers different types of information to evaluate a potential threat:

- device specific (IMEI, IMSI, ...)
- platform (build number, firmware version, ...)
- system state (cpu load, traffic, ...)
- communication (bluetooth, sms, nfc, ...)
- apps (installed, permissions)

Checking the IMEI as unique identifier against a database of known devices could be used to restrict mobile devices' network access or treat them differently. Assessment of platform data could discover out-dated or unmaintained, but still deployed, firmware versions with known vulnerabilities.

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. That introduces security risks because these apps are potentially malicious, but hardly tracked by package management software. That is why 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 SELinux as mandatory access control (MAC) there is another security policy layer to treat processes depending on their context.

SELinux supports three modes, *disabled*, *permissive* and *enforcing*, carrying out the following actions:

- disabled: No context is generated
- permissive: Logs blocked or granted operations based on the security policy
- enforcing: Security policy is enforced

Those security concepts are useful if the system is not rooted. With root access all safety precautions are obsolete as the system can be modified in many ways pretending to be a trustworthy environment and faking collected data.

So it is necessary to check for SELinux mode and root state as mobile devices rarely provide other integrity checks. TPM modules or Secure Boot could be used to detect hardware or software manipulation, but these techniques are not embedded in mobile devices.

V. Implementation

There are two projects already taking into account Android devices and IF-MAP: ESUKOM [11] and SIMU [12]. Both are focused on open source software to accomplish IT security in corporate networks considering mobile devices.

Using the DECOmap for Android client [13] as IF-MAP Client both projects support the integration of mobile devices in networks by evaluating published data using event correlation and artificial intelligence. To be able to cover mobile device specific metadata additional identifier and format extensions are designed and used.

In project ESUKOM multiple open source tools have been developed to create a SIEM system based on the correlation of metadata gathered by IF-MAP Clients:

- irond (IF-MAP Server)
- irongui (IF-MAP GUI)
- DECOIT IF-MAP-Client (IF-MAP Client for multiple components and services)
- irondhcp (IF-MAP Client for ISC DHCP)
- ifmapj (IF-MAP Library for Java)

These components are also used in the SIMU project to develop a SIEM system (based on the latest IF-MAP specifications [2] and metadata models) for the sake of simple integration, configuration and maintenance, and easy traceability of events and processes.

A detection engine analyses the gathered metadata in order to detect incidents, threats or policy violations and

[1] API 18 - "Jelly Bean", released on July 24, 2013

[2] API 21 - "Lollipop", released on November 3, 2014

initiate countermeasures by publishing specific metadata, which are retrieved by other clients to restore a secure state (e.g. by adjusting firewall rules or access policies).

VI. Issues/Improvements

IF-MAP sends its messages over SOAP/XML and HTTPS, which cause 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 components, which frequently generate many messages to keep the MAP Graph updated.

SIMU tries to tackle this problem by using CBOR [14] as data exchange format, which focuses 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.

In an environment with devices with limited resources or services generating an extensive number of events and pure IF-MAP implementations, a CBOR proxy constitutes a viable alternative to SOAP/XML, which helps to address performance problems and facilitate the usage of IF-MAP.

CBOR allows increasing 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.

VII. Monitoring without IF-MAP

Another important aspect is monitoring of mobile devices in environments without IF-MAP, as many corporate networks use monitoring systems solely based on Nagios [15] or Icinga [16]. Android is rarely considered in this context as possible threats to the monitored infrastructure.

Data collected by IF-MAP Clients can be easily transformed into information for various types of monitoring systems and statistics.

The iMonitor project [17] pursues this goal by deploying DECOmap for Android as client running on mobile devices to gather the same data (current system state) as used in IF-MAP and send them over the Nagios Service Check Acceptor (NSCA) interface [18] to Icinga.

The data are gathered and published based on events and configurable time periods using multiple event classes: *InfoEvent*, *MonitorEvent* and *AppEvent*. These events are encoded as JSON [19] strings and encapsulated in NSCA messages.

InfoEvents are created and published on first connection to give an overview on device specific data, which do not change during a session (e.g. IMEI, manufacturer, kernel, firmware and build version). An example of an InfoEvent is shown in Listing 1.

MonitorEvents are published frequently including data about changing values (e.g. traffic in and out, cpu and mem load, running processes and their properties).

To gather information about the installed apps and their permissions, AppEvents are published on first connection and after every installation or update of an app including name, version, permissions, running status, install and update time.

iMonitor collects this event data and joins it with information and events from multiple other sources like Snort, Nmap and OpenVAS to process and evaluate possible threats using methods of artificial intelligence like event correlation [20] and time series analysis.

```
{
 "timestamp": "<EVENT CREATION TIME>",
 "type": "Android",
 "ipsrc": "<IP ADDRESS>",
 "class": "info",
 "message": "Android device information
             for <IP ADDRESS>",
 "data": {
  "mac": "<MAC ADDRESS>",
  "imei": "<IMEI>",
  "imsi": "<IMSI>",
  "kernel": "<KERNEL VERSION>",
  "firmware": "<ANDROID VERSION>",
  "root": <TRUE or FALSE>,
  "selinux": "<SELINUX MODE>",
  "baseband": "<BASEBAND STRING>",
  "build": "<BUILD VERSION>",
  "brand": "<BRAND STRING>",
  "manufacturer": "<MANUFACTURER>",
  "cellnumber": <CELL NUMBER>
 }
}
```

Listing 1. InfoEvent used by *DECOmap for Android*

VIII. Conclusion and Future Work

This integration concept based on metadata, events, and the implementation of DECOmap for Android (Fig. 3) provides the research and development projects iMonitor and SIMU with the feature to monitor mobile devices and detect incidents in architectures with and without IF-MAP.

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 components.

Nagios based architectures are very common and already established. They are often easier to integrate, but

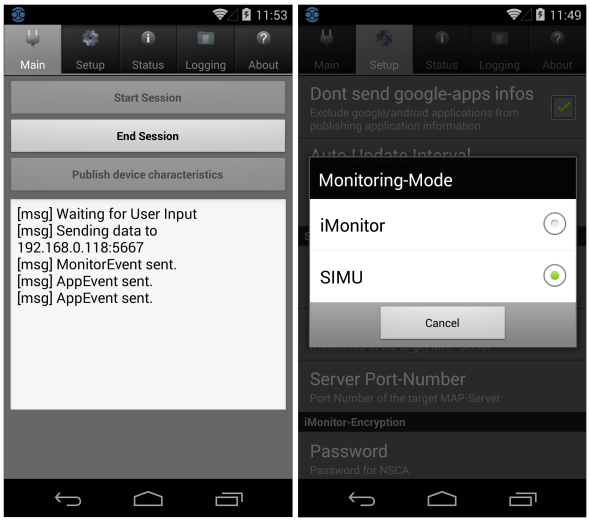

Figure 3. DECOmap for Android (Screenshot)

individual network components are barely interconnected; therefore, automation based on interaction between components is laborious.

Nagios 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 independently 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 with multiple Android versions, to collect and publish data to a 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 based on system state, behaviour and usage within the network throughout the session.

References

[1] Trusted Computing Group, Dec. 2014. [Online]. Available: https://www.trustedcomputinggroup.org

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

[3] N. Mitra and Y. Lafon, "Soap version 1.2 part 0: Primer (second edition)," World Wide Web Consortium, Apr. 2007. [Online]. Available: http://www.w3.org/TR/soap12/

[4] E. Rescorla, "HTTP Over TLS," RFC 2818 (Informational), Internet Engineering Task Force, May 2000, updated by RFCs 5785, 7230. [Online]. Available: http://www.ietf.org/rfc/rfc2818.txt

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

[6] T. Dierks and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2," RFC 5246 (Proposed Standard), Internet Engineering Task Force, Aug. 2008, updated by RFCs 5746, 5878, 6176. [Online]. Available: http://www.ietf.org/rfc/rfc5246.txt

[7] 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), Internet Engineering Task Force, Jun. 1999, updated by RFC 7235. [Online]. Available: http://www.ietf.org/rfc/rfc2617.txt

[8] C. Rigney, S. Willens, A. Rubens, and W. Simpson, "Remote Authentication Dial In User Service (RADIUS)," RFC 2865 (Draft Standard), Internet Engineering Task Force, Jun. 2000, updated by RFCs 2868, 3575, 5080, 6929. [Online]. Available: http://www.ietf.org/rfc/rfc2865.txt

[9] K. Zeilenga, "Lightweight Directory Access Protocol (LDAP): Technical Specification Road Map," RFC 4510 (Proposed Standard), Internet Engineering Task Force, Jun. 2006. [Online]. Available: http://www.ietf.org/rfc/rfc4510.txt

[10] SELinux Project, online, Feb. 2015. [Online]. Available: http://selinuxproject.org

[11] ESUKOM, "Echtzeit-Sicherheit für Unternehmensnetze durch Konsolidierung von Metadaten," online, Feb. 2015. [Online]. Available: http://www.esukom.de

[12] SIMU, "Security Information and Event Management (SIEM) für Klein- und Mittelständische Unternehmen (KMU)," online, Feb. 2015. [Online]. Available: http://simu-project.de

[13] "DECOmap for Android," online, Jun. 2015. [Online]. Available: https://github.com/decoit/Android-IF-MAP-Client

[14] C. Bormann and P. Hoffman, "Concise Binary Object Representation (CBOR)," RFC 7049 (Proposed Standard), Internet Engineering Task Force, Oct. 2013. [Online]. Available: http://www.ietf.org/rfc/rfc7049.txt

[15] Nagios Enterprises, "Nagios," Feb. 2015. [Online]. Available: http://www.nagios.org

[16] The Icinga Project, "Icinga," Feb. 2015. [Online]. Available: https://www.icinga.org

[17] iMonitor, "intelligentes IT-Monitoring durch KI-Ereignisverarbeitung," online, Feb. 2015. [Online]. Available: http://www.imonitor-project.de

[18] The Icinga Project, "Icinga Documentation: NSCA," online, Feb. 2015. [Online]. Available: http://docs.icinga.org/latest/en/nsca.html

[19] T. Bray, "The JavaScript Object Notation (JSON) Data Interchange Format," RFC 7159 (Proposed Standard), Internet Engineering Task Force, Mar. 2014. [Online]. Available: http://www.ietf.org/rfc/rfc7159.txt

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