ISSN 2047-3338

Independent and Comprehensive Intrusion Detection Management

Björn-C. Bösch

***Abstract*— Coverage of services and configuration influence the efficiency of Intrusion Detection Systems (IDS). Today, IDS have vendor-specific configurations and this limits a wide coverage of services by IDS. Operations might become complex, in case of usage of multiple systems. Efforts and frame conditions for a multi-vendor IDS implementation under one central administration and notification entity will be demonstrated. This solution provides administrators one consistent front-end for all integrated IDS. The security level will be improved by one central administration entity for the complete IDS solution independent of the respective IDS vendor. Updates and parameter modifications could be done from this supervising point. There is still no constraint to allow a connection from any analyzer to the Internet or the central operations LAN for notifications or to update itself. Managers are independent from the rest of the IDS. IDS of different vendors and analyzing levels are able to manage with one administration interface.**

***Index Terms*— IDS Management, IDXP, Intrusion Detection, Standardization and Parameterization**

I. INTRODUCTION

SINCE the Internet become public, CERT/CC has reported increasing incidents per year [1] effected on an annually rising number of vulnerabilities since 1999 [2]. In 2004 exploits were available within an average of less than five days [3]. In this context complexity of attacks increases when at the same time the needed knowledge for the corresponding attack decreases [4]. As result, the group of potential (and participating) attackers is rising.

Intrusion Detection Systems (IDS) protect critical infrastructures and services against malicious actions. Detailed knowledge of application and communication is necessary to protect services adequate. IDS are scoped on a single application (special kind of Host based IDS), a single operating system (Host based IDS) or communication protocols (Network IDS). To detect intrusions in an IT landscape, different IDS are required to protect and monitor computer systems at all levels, top to bottom.

B.-C. Bösch is with the System Software and Distributed Systems Group, Faculty II - Department of Computing Science, Carl-von-Ossietzky-University, Oldenburg, Germany, (e-mail: bjoern.carsten.boesch@uni-oldenburg.de)

IDS will be classified in their scope (Host based IDS or Network IDS) and their detection technique (signature based IDS or anomaly based IDS). Anomaly detection defines a baseline. An intrusion will be raised when the defined tolerance from the baseline exceeds. Signature based IDS compares activity against known attacks or vulnerabilities. An intrusion will be detected when activity match a signature. Signature based IDS are actual state-of-the-art [5].

Today, each IDS provides its individual software maintenance solution with (automated) update communication from the own IDS management network through the vendors via the Internet. No central entity operates as software distributor to maintain different IDS components over the IDS management network for all IDS entities. Every update communication has to be established and monitored against misuse.

Administration access, administration files and configuration syntax are individual for every IDS vendor. Dedicate system accounts with privileged rights are often needed to maintain an IDS. The configuration files and syntax are individual by every IDS vendor and additionally differs often from version to version. These facts make it very complex to select the best starting point for an IDS and the integration strategy. At the end, two opposite strategies are established:

- Focus on detection: Multiple specialized IDS protect each system, provided service or a limited amount of systems and services.

- Focus on usability: One IDS with a wide coverage of most applications, operating systems and communication protocols is integrated, with its strengths and weaknesses.

The strategy "usability" could be overlapped by the "detection" strategy, when an additional IDS covers a previously out of scope service of the first IDS integration. As result, two independent IDS will be operated. This case occurs, when a service has an own embedded IDS in the application e.g., a WiFi controller with a rough access point detector to monitor the access-point landscape.

At the end, every IDS will be operated with its own manager separated from the other IDS in independent coexistence. The administrator has to handle several front-end designs including platforms for the application and has to maintain the administration skills for every IDS. Every additional IDS

requires additional staff or operations will be more difficult with every different IDS configuration structure for the existing administration staff.

One supervising entity over different IDS enables:

- Free selection and combination of sensor and analyzer units to use the best fitting IDS for the proposed implementation focus.

- Security policy of each single analyzer could be maintained and adjust in context of security policies of other IDS.

- Provides an effective way to adjust the security policies of the whole IDS with one consistent front-end for administration over all IDS entities.

- Reduces management systems and applications over all IDS as well as operating platforms and training expenses.

- Interoperability of IDS will be enriched and IDS will be no longer operated in independent coexistence.

Current IDS are isolated solutions. Today there is only a particular combination or interaction between IDS available. Research of IDS interoperability are primarily focused on correlation of alerts and logging messages [6], [7] or [8] based on available exchange protocols [9] and formats [10].

Today there is no available work focused on parameterization of IDS. Simple Network Management Protocol (SNMP) is used to manage and operate networking systems, but SNMP [11] is hard to control, because it uses the User Datagram Protocol (UDP) [12] without an existing flow-control. On site of the manager a Management Information Base (MIB) is necessary for every IDS vendor to define and interpret possible values.

The current SNMP version 3 supports basic cryptography and authentication [13] but uses the Data Encryption Standard (DES) which is vulnerable against cryptanalytic methods [14]. The Advanced Encryption Standard (AES) was propagated in November 2001 by the US-American National Institute of Standards and Technology (NIST) as new standard [15].

The confidentiality of IDS parameterization data and the control of the data connection are not adequate protected by SNMP. Thus SNMP is not sufficient to manage IDS secure. The bottom line is, that there is no adequate existing format for complete IDS management today. This work is focused on the fundamental question: Is it possible to separate the manager completely from the rest of a heterogeneous IDS landscape with a standardized format between manager and analyzer?

The remaining paper is organized as follows: Section II analyses current IDS architectures and describes basics of the solution approach. The subsequent subsection points out the methodology of parameterization. Subsection II-C and II-D give an overview about the parameterization format and the integration in three different free open source IDS. Section III presents the integration results and concludes this work.

II. APPROACH

This section illustrates the IETF IDS model and points out, which enhancements have to be done for independent IDS management for multi-vendor IDS architectures. Parameterization methodology and structure of the standardized parameterization format are superficial described. This section concludes with the theoretical integration and a brief mapping overview of parameters.

A. Current IDS Architectures and Formats of the IETF

Current multi-vendor IDS architectures do not interact with each other. They are in independent coexistence. Based on IDMEF it is possible to integrate an additional general monitoring system as notification umbrella. This approach improves the alert management, not the daily administration of IDS in detail.

This work is based on the IETF IDS model, including architecture and entity definitions of [16]. Result of the IDS architecture analysis is, that the entities analyzer and sensor are vendor-specific. The manager is the only entity which could be shared with other IDS. In a multi-vendor IDS architecture the manager functionality could be partial shared by a notification umbrella system with IDMEF. To share the manager functionality of an IDS completely, the communication between a general manager and vendor-specific analyzers has to be standardized.

Today, IDMEF standardizes notifications to a monitoring application. As transport protocol the Intrusion Detection eXchange Protocol (IDXP) [9] is already created on top of the Blocks Extensible Exchange Protocol (BEEP) [17]. The BEEP framework provides confidentiality, integrity and authentication for the communication. A streamtype option with the valid values "alert", "heartbeat" or "config" is already provided by IDXP. The value "alert" is used by IDMEF. The value "heartbeat" is provided for synchronization of two or more analyzers, acting as one analyzer. The IDXP could be used as communication framework, but the heartbeat exchange format is not needed to standardize in an one vendor heartbeat environment.

This work uses IDXP with the "config" value in the streamtype option as communication framework to separate the manager from the rest of the IDS with a standardized communication between analyzer and manager. The communication between sensor and analyzer will be still vendor-specific. The communication in the IETF IDS model has to be modified. As visualized in fig. 1, the security policy will be applied to the manager and distributed to the analyzers and forwarded to the sensors instead of directly from the administrator to all IDS entities. Operators and administrators use the manager as single point of human interface to run the IDS.

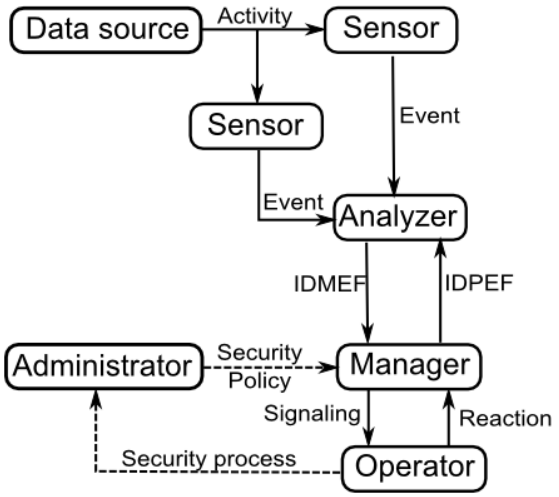

Fig. 1. Enriched IDS model of the IETF with standardized communication between analyzer and manger and the manager as single point of administration

B. Parameterization Methodology

IDS have their individual structure, syntax and semantic for management and operations. On one hand, neither an interaction nor a sharing of configuration files or references between IDS is possible and is not in place today. On the other hand, every IDS compares activity against a reference database. References consist of a baseline part and a customizing part. The baseline part describes payload and non-payload characteristics of the event itself (intrusion activity or baseline). Baseline parts are customized to the individual implementation.

For example, a SYN-flood contains in the baseline part the attack description. In this case, the TCP/IP protocol with a set SYN-flag. The customizing part defines threshold and time interval for the individual implementation of the event. As result more than 200 SYN-requests within 1 second cause a SYN-flood signalization.

The vendor-specific requirements for internal processing characterize the design of the baseline part of a rule. So this is out of scope for parameterization. The customizing part of a rule defines the environmental integration and is the focus of the standardized parameterization format. Analog to IDMEF the standardized parameterization format is named "Intrusion Detection Parameterization Exchange Format" (IDPEF).

C. IDPEF Overview

Based on documented requirements in [18] and [16], IDPEF was created on top of IDXP. The purpose of this format is to parameterize the analyzer to the individual implementation and to maintain the IDS in operations. Analogue to IDMEF, IDPEF is set up on the Extensible Markup Language (XML) [19].

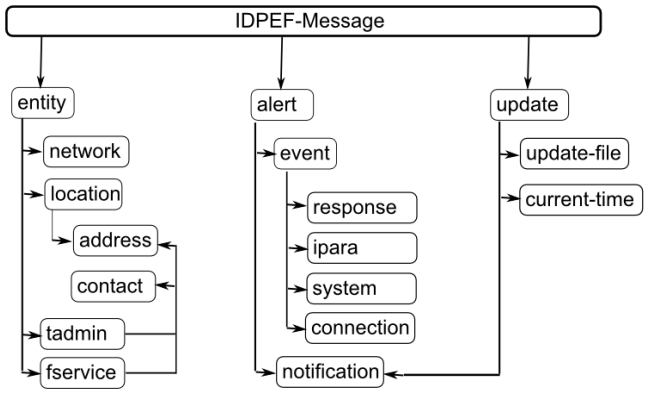

Fig. 2. XML node structure of the Intrusion Detection Parameterization Exchange Format

As illustrated in Fig. 2, IDPEF is split in three core sections with a root node named "IDPEF-Message". The node section <entity> includes parameters to operate the sensor (e.g. NTP-server, IP-addresses, etc.) and provide service information like location, field service contact, etc. Updates are scheduled and transferred within the node section <update>. These two sections were designed without any restriction or frame condition, but accordingly to the set out requirements [18].

More challenging is the section <alert>. Parameters of every event and response are defined within this section. Each IDPEF parameter has to be mapped bi-unique to the corresponding parameter of each single IDS. In each single event node, all attributes for the event are defined here. These are displayed name, additional information for this event, severity, priority, impact and which security value was affected in case of a cause. An "enable/disable" option is added to the event and every attribute in the child nodes of the <event> node. Within the child nodes responses, thresholds, intervals and if required individual parameters are defined. The other child nodes of <alert> are the <notification> node and the <response> node. The "notification" node includes general parameters like IP-address, structure of the notification, etc. to set up the notification communication. Individual responses, like execution of scripts with expected parameters are defined

```
<IDPEF-Message>
 <alert>
  <event name = "9999_1"
      displayedas = "SYN-Flood"
      origin = "self-generated SYN-flood event"
      serverity = "attempted-dos" >
   <network source = "any"
        destination ="$myHONE_NET"
        direction = "uni" />
   <system time = "60"
        threshold = "200" />
  </event>
 </alert>
</IDPEF-Message>
```

Fig. 3. XML Example IDPEF messages of a SYN-Flood attack

in the <response> node.

Based on the practical example in section 2.B, a SYN-flood parameterization is defined as depicted in fig. 3. Under the IDPEF-Message node the section <alert> contains all parameters for the events. Each <event> node contains one event. The <event> node itself includes the attributes "name" to identify the rule bi-unique and "displayedas" to define what will be displayed when the event causes. The attribute "origin" keeps more information about the background of this attack. "severity" classifies the priority of the event. Threshold and time interval are set in the child node "system" with the attributes "time" for the time interval and "quantity" for the threshold. The complete XML Schema Definition for IDPEF was defined in [20].

D. IDPEF Integrations

Based on the theoretical first green field approach IDPEF was defined [20]. Each attribute was named and underpinned with a rationale. Subsequent the attributes of IDPEF are mapped to the open source IDS Snort [21], Samhain [22], OSSec [23] and Bro [24]. Improving adjustments were carried out within this phase. Based on these theoretical mappings the software implementations were carried out.

This theoretical approach was implemented first in three open source IDS, the network IDS Snort [21] and the two host based IDS Samhain [22] and OSSec [23] , to test the common applicability of this format. The implementations do not modify IDS executables. Only existing configuration files are processed and modified. Implementations in Bro [24] will follow.

As human interface an IDPEF web front-end was created that enables IDXP based communication to a selected analyzer. Attribute values are modified over the front-end and send back as IDPEF update to the analyzer. Additional software updates including upload of update files and new signatures are scheduled within the front-end and also send to the analyzer.

On site of the analyzer individual IDXP / IDPEF communication modules are created. These modules modify configuration files of operating system and IDS software and schedules updates and their execution.

Each attribute of individual IDS configuration files were assigned as baseline parameter or customizing parameter. Baseline parameters are not transferred into or modified by IDPEF. Customizing parameters are bi-unique mapped to an IDPEF attribute.

Snort's IDPEF communication module maps "preprocessor", "variable", "output" and "config" parameters as well as rules into IDPEF. Dynamic loaded libraries were categorized as baseline parameters.

Customizing parameters are selected and mapped into IDPEF for each Snort rule. As schematically illustrated in Fig. 4 the parameters are mixed within the rule. Parameters of the rule head are mapped into IDPEF. The general rule options and post detection rule options are classified as customizing parameters. The payload detection rule options and the non-

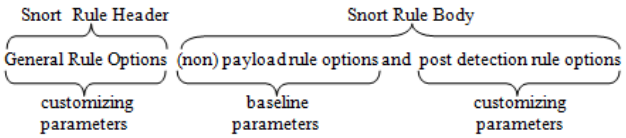

Fig. 4. XML Example IDPEF messages of a SYN-Flood attack

payload detection rule options are classified as baseline parameters, because they are part of the signatures and are not mapped in IDPEF. The options "rev" and "sid" are used as unique rule identifier.

Samhain's configuration file includes customizing sections only. Each section and its parameters are mapped bi-unique into IDPEF. Only the sections "external" and a high percentage of "Misc" were not integrated in this implementation. These sections were mostly classified as baseline configuration or contain IDMEF alternative notifications and are not mapped into IDPEF.

OSSec's configuration bases on XML structures. All nodes in the core configuration file (ossec.conf) are mapped in IDPEF. The structure of OSSec rules is split in a grouping rule without alert function and baseline-information in the <match> node. The corresponding sub rules are connected with the <if_sid> or <if_matched_sid> node with the grouping rule node. Additional baseline-information is provided in the <match> and <regex> nodes. For the proof-of-concept integration every single rule, including the remaining nodes and attributes, were mapped separately into the IDPEF. The grouping rule does not contain any customizing parameter and does not have any impact on the evaluation of the applicability of IDPEF. A more complex solution with a change of the configuration structure of OSSec is able to address the sub rule structure adequate.

III. CONCLUSION

The implementations demonstrate that one central independent manager is able to operate IDS of different vendors and analyzing levels. Only one central administration entity is necessary to operate, manage, maintain and administer a heterogeneous IDS landscape based on a small format. All connections are initialized from the central manager to the distributed analyzer entities. All updates (parameter and software) could be controlled, downloaded and distributed to each single IDS entity from one central management entity. The communication is easier to control, because there is only one communication port from the manager to all IDS entities necessary and the content could be inspected by security devices. No Connection from an IDS analyzer entity to a system outside the administrative IDS LAN is necessary.

The format is able to parameterize different analyzers. Baseline configurations have to be still initially set up. It is not possible to modify or operate baseline configurations (including rules) by external modifications only, because they are highly vendor-specific and depend on their internal software module structure and processing. Analyzing

references depend on internal processing of the analyzer. Standardization impairs the core development of analyzers.

The format requires a rudimental configuration to reach the analyzer and to apply the parameterization. The implementations show that customizing parameters are able to due to a small amount of attributes. The configurations of all three IDS mix baseline and customizing parameters. A separation of baseline and customizing configuration is helpful to apply IDPEF as common customizing file for IDS.

The standardized access and format reduces the complexity of IDS and provides administrators consistent maintenance environment for different IDS. The operator has one consistent front-end to operate all IDS. The format supports keeping analyzers up to date and the BEEP framework grants confidentiality and integrity for the communication. The communication is initialized from the manager and could be limited to a few numbers of systems. All maintenance could be scheduled, executed and monitored by one central point.

The manager is an independent entity of an IDS. Selection criteria of a manager are independent from the criteria of the rest of the IDS. It is not longer a constraint to operate IDS with vendor-specific managers or to operate more than one manager entity. Based on these results, manager and analyzers of an IDS could be developed independently. The selection of manager software is now independent from the selection of IDS analyzers. Specialized systems management manufacturers are able to enrich their products by common IDS management. This evolution supports IDS management products with more comfort, usability and reporting features. Supervising managers are able to provide consistency checks for a cascade analyzer environment, bulk parameter changes or comfortable update scheduling.

The supervising IDS management controls each single analyzer including download and distribution of software and updates. No access from the IDS LAN to other networks (i.e. the Internet or the central systems management network) is necessary. This raises the security level of the administrative IDS network.

All contemplated IDS of this work are able to use temporarily Secure Shell connections as manager application to modify the configuration files on the analyzers. As result, the entity interaction ends at the analyzer. With a standardized communication and format a vendor-independent interaction between analyzers could be established. As next step, analyzing of reported events could be integrated in the manager.

The development stream for manager should be now focused on usability, monitoring and alerting as well as additional analyzing and maintenance features. The rest of the IDS should be focused on effective performance and detection of intrusions in their development streams.

On the whole, the manager is an independent system of IDS and could be separated from the rest of the IDS. It is possible to operate different IDS with one consistent administration front-end. This finding enables new and independent evolution streams for IDS analyzer and manager.

REFERENCES

[1] CERT / CC: CERT/CC Statistics 1988-2006, 2007, available online at http://www.cert.org/stats/ (last visit: 2007-06-13).

[2] J. Havrilla: Attack Sophistication vs. Intruder Technical Knowledge in Vulnerability Discovery: Bridging the Gap Between Analysis and Engineering, 2006, available online at http://www.pghrims.org/resources/policyholder/cert-2003-04-22-pghrisk.pdf (last visit: 2011 11 26).

[3] Symantec, Threat Report for July 04 - December 04 Volume VII, 2005, available online at http://eval.veritas.com/mktginfo/enterprise/white_ papers/ent-whitepaper_symantec_internet_security_threat_report_vii. pdf, last visit 20. May 2007.

[4] Mark Baker: Security Basics, 09.03.2006, available online at http:// impact.asu.edu/cse494sp09/SecurityBasics.ppt (last visit: 2012 03 09)

[5] T. Werner, C. Fuchs, E. Gerhards-Padilla, P. Martini: "Nebula - generating syntactical network intrusion signatures", 2009, 4 th International Conference on Malicious and Unwanted Software (MALWARE), pp 31 -38.

[6] Abe Chin-Ching Lin: Enhancing interoperability of security operation center to heterogeneous intrusion detection systems, 2005, Security Technology, 2005. CCST '05. 39th Annual 2005 International Carnahan Conference on, pp. 216 - 221.

[7] A. Madani: Log management comprehensive architecture in Security Operation Center (SOC), 2011, Computational Aspects of Social Networks (CASoN), 2011 International Conference on, pp. 284 - 289.

[8] Nguyen Doan Man and Eui-Nam Huh: A Collaborative Intrusion Detection System Framework for Cloud Computing, 2012, Proceedings of the International Conference on IT Convergence and Security 2011, Lecture Notes in Electrical Engineering, pp. 91-109.

[9] B. Feinstein and G. Matthews: The Intrusion Detection Exchange Protocol (IDXP), 2007, RfC 4767, available online at http://www.ietf.org/rfc/rfc4767.txt, last visit 01. September 2007.

[10] H. Debar, D. Curry and B. Feinstein: The Intrusion Detection Message Exchange Format (IDMEF), 2007, RfC 4765, available online at http://www.ietf.org/rfc /rfc4765.txt, last visit 01. September 2007.

[11] J. Case, R. Mundy, D. Partain, B. Stewart: Introduction and Applicability Statements for Internet Standard Management Framework, Dec 2002, RfC 3410, available online at http://tools.ietf.org/html/rfc3410 (last visit: 2011 11 28).

[12] J. Postel: User datagram Protocol, Aug 1980, RfC 768, available online at http://www.ietf.org/rfc/rfc768.txt (last visit: 2011 11 28).

[13] U. Blumenthal, B. Wijnen: User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3), Dec 2002, RfC 3414, available online at http://tools.ietf.org/html/rfc3414 (last visit: 2011 11 28).

[14] M. Eichlseder: AES und DES, Nov 2007, available online at http://opt.math.tu-graz.ac.at/~aistleitner/Proseminar20072008/ Eichlseder_Ausarbeitung.pdf (last visit: 2011-12-04).

[15] NIST: Processing Standards Publication 197: Announcing the Advanced Encryption Standard (AES), Nov 2001, available online at http://csrc. nist.gov/publications/fips/fips197/fips-197.pdf (last visit: 2011 11 28).

[16] M. Wood, M. Erlinger: Intrusion Detection Message Exchange Requirements, March 2007, RfC 4766, available online at

http://www.ietf.org/rfc/rfc4766.txt, last visit 01. September 2007.

[17] M. Rose: The Blocks Extensible Exchange Protocol Core, Mar 2001, RfC 3080, available online at http://www.ietf.org/rfc/rfc3080.txt, last visit 01. September 2007.

[18] B.-C. Bösch: Intrusion Detection Parameterization Exchange Data Model, Jubilee 35. International Convention on Information and Communication Technology, Electronics and Microelectronics (μpro), Information Systems Security, May 2012.

[19] W3C: Extensible Markup Language (XML), 2011, available online at http://www.w3.org/XML/ (last visit: 2011-12-03).

[20] B.-C. Bösch: Intrusion Detection Parameterization Exchange Format, 2011, unpublished.

[21] SNORT: http://www.sort.org (last visit: 2011-12-03)

[22] Samhain: http://www. http://la-samhna.de/ (last visit: 2011-12-03)

[23] OSSec: http://www.ossec.net (last visit: 2011-12-03)

[24] Bro: http://www.bro-ids.org (last visit: 2011-12-03)

[25] B.-C. Bösch: Standardized Parameterization of IDS, International Journal of Advanced Research in Computer Engineering & Technology (IJARCET), Vol. 1 No. 3, May 2012, pp:1-5

Björn-C. Bösch studied technical computer science at the University of Applied Science in Emden In February 2000 earns his diploma degree in applied electrical engineering. Subsequent he starts his professional carrier as SYSTEMS ENGINEER at a global European based ICT provider by designing and integrating network security solutions and IP-networks. From 2005 to 2009 he was responsible for the security management in big transition projects. In 2009 he changes into business consulting with a main focus as SOLUTION ARCHITECT for security solutions and certification support. Bösch is also certified ISO 27001 AUDITOR.

In 2007 he starts as EXTERNAL SCIENTIFIC RESEARCHER at the Carl-von-Ossietzky-University in Oldenburg in the *System Software and Distributed Systems Group at the* department of computing science. His work is focused to standardize the communication between IDS of different vendors and integrates them to a meta-IDS.