Standardized Parameterization of Intrusion Detection Systems

Björn-C. Bösch

***Abstract*— Efficiency of Intrusion Detection Systems (IDS) depends on their configuration and coverage of services. The coverage depends on used IDS. In the case of usage in multiple systems, operations might become complex because IDS configurations are still vendor-specific. This paper shows aspects and frame conditions for a multi-vendor IDS implementation under one central administration and notification entity.**

Subsequent it will be briefly discussed, why current management protocols are not adequate to manage IDS. A short paragraph describes the parameterization methodology to separate baseline configuration and parameterization for individual integrations. Separation of baseline configuration and parameterization for an individual integration is illustrated on basis of Snort. Analog to the Intrusion Detection Message Exchange Format a parameterization format for standardized parameterization was designed. Based on this designed green field approach the format structure was evaluated.

Usability and functionality of this approach was demonstrated by integration in the network based IDS Snort, the host based IDS Samhain and OSSec under one parameterization web front-end. This approach provides administrators one consistently administration front-end for all implemented and operated IDS. One central administration entity manages the complete IDS solution independent from IDS vendors. Updates and parameter modifications could be done from this central point. The security level is improved, because there are no longer constraints to allow connections from analyzers to the Internet or the central operations LAN for notifications or to update it.

Managers are independent from the rest of the IDS. IDS of different vendors and different analyzing levels could be managed with one administration interface.

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

I. Introduction

Since the Internet became public, CERT/CC has reported increasingly incidents per year [1]. A cause of this trend is the increase of vulnerabilities since 1999 [2]. In 2004 exploits were available within an average of less than five days [18]. In this context attacks become more and more complex with decrease of attackers knowledge. This trend supports complex hacking attacks as depicted in fig. 1.

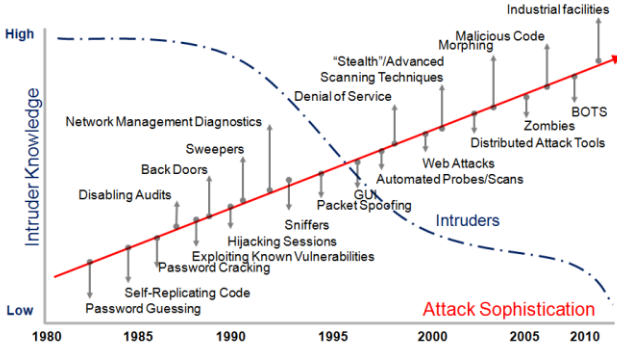
Intrusion Detection Systems (IDS) protect critical infrastructure and services against malicious actions.

Detailed knowledge of application and communication are 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 IT composites, different IDS are required to protect and monitor various computer systems and services at all levels, top to bottom.

IDS will be classified on its scope (Host based IDS or Network IDS) and detection technique (signature based IDS or anomaly based IDS). Anomaly detection defines a baseline. An intrusion will be raised when the tolerance from the baseline is exceeded. Signature based IDS compares activity against known vulnerabilities. An intrusion will be detected when activity matches a signature. Signature based IDS are actual state of the art [24].

It is very important to maintain and report the efficiency of IDS. A comprehensive reporting demonstrates the efficiency of the IDS complete and in particular. So fully interacting IDS enables this kind of reporting. To achieve this, a single point of administration and operations with a defined and processable structure is needed.

Administration access, administration files and configuration syntax of each IDS vendor are different. Individual system accounts with privileged rights are often needed to maintain IDS. Administration skills have to cover every IDS (full coverage and detection). Alternatively, a limited amount of systems and services will be protected by an IDS (reduced coverage). A third approach is to select an IDS which is able to protect all applications, operating systems and communication protocols, with its strength and weaknesses (reduced detection). All solutions with more than one IDS require additional manager with individual front-end designs including platforms for this application and



Sources: *Carnegie Mellon University, 2002 and Idaho National Laboratory, 2005*

Figure 1. Attack Sophistication vs. Intruder Technical Knowledge [3]

Manuscript received Apr 15, 2012.

Björn-C. Bösch:, *System Software and Distributed Systems Group, Faculty II - Department of Computing Science, Carl-von-Ossietzky-University Oldenburg. Oldenburg, Germany,*

administration skills for it. Operators and administrators have to be trained on each implemented IDS. Every additional IDS requires additional staff or the configuration will be more difficult, with every different IDS configuration handling, for the existing staff.

Not only training expenses have impact on IDS expenses, dedicate systems management systems are needed for each IDS. So every IDS requires on top of sensors and analyzers individual management systems to operate mid-sized and complex IDS architectures.

Today, each IDS provides its individual software maintenance solution with (automated) update communication from the 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.

IDS are used to improve the security level and reduce incidence rate and / or impact of identified risks. The selection of IDS are influenced by the frame condition that the total costs (in this case by one or more IDS) are lower than the total financial impact of the weighted risks of the counteracting measures. Therefore IDS implementations are often a compromise of integration approaches above, to have a high coverage with acceptable expenses.

A standardized communication between a common human interface for IDS and the rest of the IDS itself is necessary, to integrate IDS independent of the manufacturer and makes IDS operations more comfortable. This improves concomitant operations usability and reduces operating costs.

A widely used solution to manage and operate networking systems is the Simple Network Management Protocol (SNMP). SNMP [4] uses the User Datagram Protocol (UDP) [6] without an existing flow-control. For every IDS vendor a Management Information Base (MIB) is necessary on site of the manager to define possible values and their interpretation.

The current version 3 of SNMP supports basic cryptography and authentication [5]. The used Data Encryption Standard (DES) is vulnerable by cryptanalytic methods [7]. The Advanced Encryption Standard (AES) was propagated in November 2001 by the US-American National Institute of Standards and Technology (NIST) as new standard [8], but is not used for version 3 of SNMP. This approach requires individual add-ons on site of the manager. Furthermore the confidentiality of IDS parameterization data and the control of the data connection were not adequate protectable. So SNMP is not adequate to manage IDS secure.

Current IDS are isolated solutions. Today there is only a particular combination or interaction between IDS available. This work is focused on the fundamental question: Is it possible to separate the manager completely from the rest of heterogeneous IDS composites 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. Subsequent the methodology of parameterization was pointed out. Section III gives an overview about the parameterization format. The integration in three different free open source IDS is described in section IV. Section V presents the integration results and concludes this work.

II. Approach

This section discusses aspects of multi-vendor IDS protection and shows which entities of a heterogeneous IDS are able to share, based on the IETF IDS model. The subsequent paragraphs illustrate how the manager could be separated and become independent to the rest of the IDS. The parameterization methodology was described on a high level.

A. *Current IDS Architectures*

Current multi-vendor IDS architectures do not interact with each other. They are independent coexistent.

Based on the IETF IDS model the architecture was analyzed. Analyzers and sensors are vendor-specific entities. The manager is the only entity that could be shared with other IDS. In a multi-vendor IDS architecture the manager’s functionality could be partial shared by a notification umbrella system with IDMEF. This requires additional budgets for hardware, software, maintenance and operating costs, i.e. annual budgets for professional training. Recurrent costs (like professional trainings) could be reduced by one central and consistent administration interface.

Based on an architectural analysis it might be possible to share the manager functionality of an IDS completely. Therefore 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) [15] was created on top of the Blocks Extensible Exchange Protocol (BEEP) [12]. The BEEP framework provides confidentiality, integrity and authentication for the communication, end to end. A streamtype option with the valid values "alert", "heartbeat" or "config" is already provided by IDXP. The value "alert" is used by IDMEF. The other two values are still available for new usage. This work uses IDXP as communication framework with the "config" value in the streamtype option.

The manager will be separated 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 relations within the IETF IDS model have to be modified. Fig. 2 illustrates the modified IDS model of the IETF. 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.

B. *Parameterization Methodology*

IDS have their individual structure, syntax and semantic for management and operations. Sharing of configuration files or references between IDS is not possible and an interaction is not in place today. On the other hand, all IDS compare activity against a reference database. References

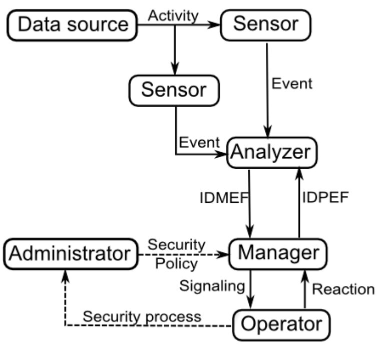


Figure 2. IDS model with standardized communication with the IDS manager

consist of a baseline part and customizing part. The baseline part describes the event itself (intrusion activity / vulnerability or baseline). Baseline parts will be customized for 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 the threshold and a 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 baseline part of a rule is vendor-specific. So this is out of scope for parameterization. The customizing part of a rule will be mapped into the standardized parameterization format. Analog to IDMEF the standardized parameterization format is named "Intrusion Detection Parameterization Exchange Format" (IDPEF).

III. IDPEF Overview

The IDPEF is described in this section on a high level.

Based on general requirements [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. IDPEF use the Extensible Markup Language (XML) [19], because:

- XML is free from licenses, license-fees and has no royalties,
- XML is human readable,
- XML is easy to edit,
- XML is easy to process with web technologies, e.g. Java Script, and
- XML is also used for IDMEF.

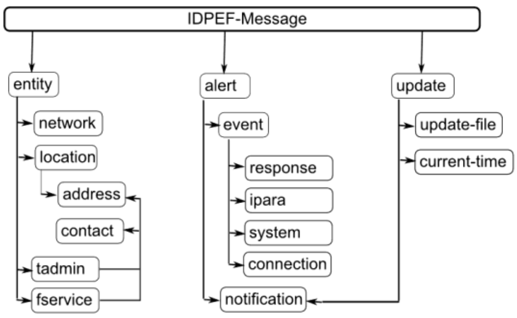


Figure 3. Node structure of an IDPEF-Message

As illustrated in fig. 3, IDPEF is split in three core sections with a root node named "IDPEF-Message". The node section "entity" includes parameter to operate the sensor (e.g. NTP-server, IP-addresses, etc.) and service information like location, field service contact, etc. Updates were scheduled and transferred within the node section "update". These two sections could be designed without any restrictions or frame conditions.

Within the section "alert", parameter of every event and response were defined. Each IDPEF parameter has to be mapped bi-unique to the corresponding parameter of each single IDS.

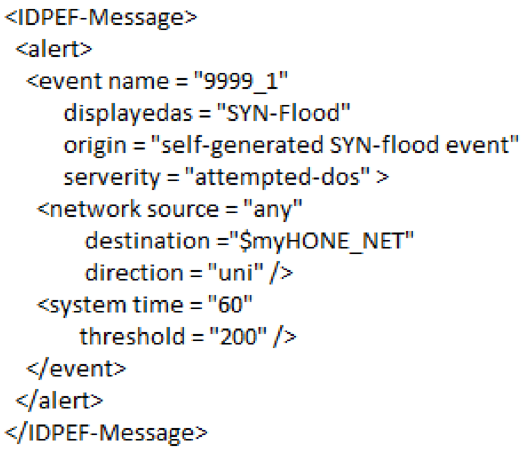
In each single event node, common attributes for the event will be defined. This are displayed name, additional information for this event, severity, priority, impact and which security value was affected in case of a cause. An option to enable or disable is added to the event and every attribute in the child nodes of the event node.

Based on the parameterization methodology example in section II B., a SYN-flood parameterization is defined as depicted in fig. 4. Under the IDPEF-Message node the section alert contains all parameters for all events. Each event node contains one event. The 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 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 [17].

IV. IDPEF Integration

The Integration of IDPEF in the three open Source IDS Snort, OSSec and Samhain is illustrated in this section.

Based on the theoretical first green field approach IDPEF and the communication proceedings were defined in [17]. Each attribute was named and underpinned with a rationale. Subsequent the attributes of IDPEF were mapped to the open



```
<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>
```

Figure 4. Example of IDPEF-SYN-Flood-Message without defined namespace

source IDS Snort [20], OSSec [22], Samhain [21] and Bro [23]. 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 [20], the Host based IDS Samhain [21] and OSSec [22] to evaluate the common applicability of this format. The implementations do not modify IDS executables. Only existing and editable IDS and systems configuration files are processed and modified. The implementation in Bro [23] will be following.

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

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

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

Snort's IDPEF communication module maps "preprocessor", "variable", "output" and "config" parameters as well as rules into event nodes of IDPEF. Dynamic loaded libraries were categorized as baseline parameters and not mapped into IDPEF.

Customizing parameters were selected and mapped into IDPEF for each Snort rule. As schematically illustrated in fig. 5 the parameters are mixed within the rule. Parameters of the rule head are mapped into IDPEF. "msg", "reference", "classtype", "priority", "logto", "resp", "react" and "act" were mapped into IDPEF. The options "rev" and "sid" were used as unique rule identifier.

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

V. Conclusion

The implementations demonstrate that IDS of different vendors and analyzing levels are able to operate under one central independent manager. A smart format is requisite to parameterize IDS analyzers. Only one central administration entity is necessary to operate, manage, maintain and administrate a heterogeneous IDS composite. The complete administration is based on the standardized format IDMEF for notifications and our free format for parameterization and software maintenance (IDPEF).

Current configurations of IDS are very individual in syntax, structure and file structure. Snorts preprocessor configuration has its own configuration line, which depends additional on the version. Baseline and customizing parameters are mixed in all contemplated IDS. As result, parameters vary in naming, separation and structure of values and / or value ranges. A separation of baseline and customizing configuration is helpful to apply IDPEF as common customizing file for IDS.

Customizing of IDS attributes are due to a small amount of parameters and values. All customizing parameters of every analyzer are able to map in IDPEF bi-unique. Baseline configuration and references depend on the internal processing and are not able to standardize with external modifications only.

All connections are initialized from the manager to analyzer entities. All updates (parameter and software) could be controlled, downloaded and distributed to each single IDS entity by 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 a security device. A connection from an IDS analyzer entity to a system outside the administrative IDS LAN is not necessary.

Selection criteria of IDS managers are independent from the criteria of the rest of the IDS. It is not longer a constraint to operate the IDS with vendor-specific managers or to operate more than one manager entity. The selection of manager software is now independent from the selection of IDS analyzers. Based on these results managers and analyzers of an IDS could be developed independently. Specialized system management manufacturers are able to enrich their products with common IDS management. This evolution supports to focus sourcing and analyzing entities more on data sourcing and analyzing. Competing IDS management products will be providing more comfort, usability and reporting features. Central managers are able to

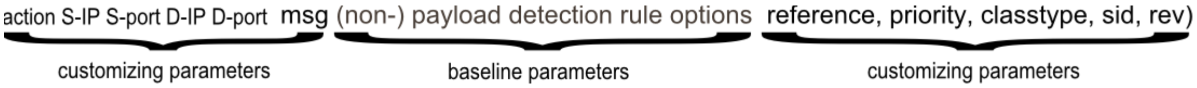

action S-IP S-port D-IP D-port msg
(non-) payload detection rule options
reference, priority, classtype, sid, rev)
customizing parameters
baseline parameters
customizing parameters

Figure 5. Mapping of snort rule to baseline and customizing parameters

provide consistency checks for a cascade analyzer environment, bulk parameter changes or comfortable update scheduling.

A central IDS management enables to control each single analyzer including download and distribution of software. No access from the IDS LAN to networks with lower security level (e.g. the Internet) or higher security level like the central systems management network is required. This improves the security level of the administrative IDS network.

The current IDS model of the IETF has no closed loop control within the IDS. The security policy will not be applied from the administrator directly to manager, analyzer and sensor. For a standardized parameterization the communication for the security policy was modified from the administrator to the manager. The standardized parameterization will be distributed from the manager via the analyzers to each single sensor of the IDS. Fig. 2 illustrates this modified communication based on the IETF IDS model.

The modifications of the communication in the IDS model effectuate control loops in entity interactions of the IDS. Fig. 6 illustrates that there are two control loops outside the technical IDS entities. The first control loop is between analyzer and operator, where the manager signalizes the operator the events and the operator carries out the reaction to the event by using the manager. The second control loop outside the IDS starts also on the IDS manager event signalization to the operator. Now, the operator starts the security process to the administrator. The administrator adjusts the security policy on the manager. The next control loop was originated by separation of the manager and analyzer by the standardized formats IDMEF and IDPEF. Additional analyzing functionality could be integrated in the manager to adjust other analyzers automatically. Between analyzer and sensor is an additional control loop where the analyzer applies the security policy to the sensors and gets the events back. This three control loops within the IDS improve interaction between the IDS entities.

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 consistently administration front-end. These findings enable new and independent evolution streams for IDS analyzers as well as managers. As a consequence new business cases for IDS manufacturers to provide enriched IDS management systems will be enabled.

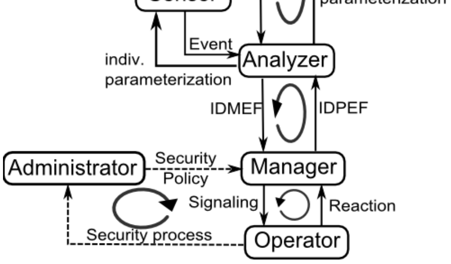


Figure 6. IDS model with standardized communication with the IDS manager

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

[4] 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).

[5] 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).

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

[7] 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).

[8] 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).

[9] Kahn, C., Porras, P., Stanifort-Chen, S., Tung, B.: "A Common Intrusion Detection Framework", 1998, available online at http://gost.isi.edu/cidf/ (last visit: 2007-06-13).

[10] J. Myers, RfC 2222: "Simple Authentication and Security Layer (SASL)", 1997, available online at http://www.ietf.org/rfc/rfc2222.txt, last visit 01. September 2007.

[11] T. Dierks and C. Allen, "RfC 2246: The Transport Security Layer", 1999, available online at http://www.ietf.org/rfc/rfc2246.txt, last visit 01. September 2007.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] B.-C. Bösch: "Ein einheitliches Austauschformat zum Parametrisieren verschiedener IDS", in UpTimes of German UNIX User Group – Frühjahresfachgespräche 2012, pages 51 - 59, March 2012.

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

[18] 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.

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

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

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

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

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

[24] 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.

Björn-C. Bösch has joined in 2007 the *System Software and Distributed Systems Group at the* department of computing science of the Carl-von-Ossietzky-University Oldenburg as external scientific researcher. His focus is to standardize the communication between IDS of different vendors to integrate them into a meta-IDS.