An Efficient Intrusion Detection System Design

Thomas Holz, Michael Meier, Hartmut Koenig
Brandenburg University of Technology Cottbus
Department of Computer Science
PF 10 13 44, 03013 Cottbus
Germany
Mail:{thh,mm,koenig}@informatik.tu-cottbus.de

Abstract: Intrusion detection systems have proved to be an effective instrument for protecting computer and network resources. In addition to preventive security mechanisms (e.g. authentication, encryption, or access control) they provide an automatic detection of security violations. Some systems are able to reduce arising damage by the automatic execution of intrusion response actions. For host-based systems, the most effective detection approach is audit data analysis with signature detection methods. Because of the character of audit records, these approaches are post-mortem techniques. Thus, the success of an intrusion response activity essentially depends on the time difference between the real appearance and the detection of the particular security violation. At the Brandenburg University of Technology Cottbus we are currently working on HEIDI (*High-Efficient Intrusion Detection Infrastructure*), which is a new approach for solving this intrusion detection efficiency problem. HEIDI consists of modules and mechanisms, which are aimed to maximize the detection speed in distributed environments. The main types of modules are sensors for fast local audit record capturing and preprocessing, and agents for performing the detection of local and distributed attacks. Unlike any other known system, HEIDI applies a combined signature evaluation scheme with maximal local concentration of analysis functionality. This leads to a minimal need for inter-agent network traffic and delay. Additionally, for assuring a continuous operation, HEIDI uses an adaptive mechanism to compensate temporary overload situations, e.g. audit bursts. To avoid a stop of execution, the affected agents are able to delegate their analysis functionality to other agents temporarily. By combining sensors and agents, it is possible to get tailored hierarchical intrusion detection architectures for given target environments. The HEIDI prototype implementation is currently in progress.

Keywords: intrusion detection, signature detection, audit analysis, detection efficiency

1. INTRODUCTION

The rapid progress of communication technologies brings numerous benefits to the human society, but it also increases dependencies on information systems. The growing potential of threats, that make these systems more and more vulnerable, is caused by the complexity of the technologies themselves and by the growing number of individuals, which are able to abuse the systems. Subversive insiders, hackers, and terrorists get better and better opportunities for attacks. In industrial countries, this concerns both numerous companies and the critical infrastructures, e.g. the health care system, the trade, or the military protection [1, 2].

To counter these threats, well-known preventive techniques, such as authentication of communication partners, encryption of sensitive data, and access control to resources, have to be supplemented by reactive mechanisms in many scenarios. These mechanisms aim at the detection, indication, and evaluation of security violations. Additionally, they shall allow a damage confinement. For all of these purposes, intrusion detection systems (IDSs) have proved to be appropriate instruments.

The research and development of intrusion detection technology take place since about 20 years. During this time, numerous ambitious approaches have been proposed, which led to the first commercial solutions available now [3]. Some of the main problems of using commercial systems in real-life environments are the high maintenance effort, limited effectiveness, and a poor efficiency. These systems mainly confine themselves to detecting simply structured security violations. As consequence, a big amount of sophisticated and critical attacks is not captured. Today's intrusion detection solutions are less suited for the deployment in large computer networks, especially for tight time constraints. Growing communication infrastructures (e.g. networks with switches) and increasing user requirements (e.g. privacy) raise additional problems, which are not covered by existing concepts.

At the Brandenburg University of Technology Cottbus we are investigating some special intrusion detection problems. In this paper we present the basic principles of HEIDI (*High-Efficient Intrusion Detection Infrastructure*), which is a new approach for building efficient distributed intrusion detection systems. HEIDI is based on the experiences we gathered with our older intrusion detection system AID (*Adaptive Intrusion Detection system*) [4] and other systems as well.

The paper is structured as follows: Section 2 shortly introduces the area of host-based intrusion detection and the inherent detection efficiency problem. In section 3, the basic approaches for constructing a network of distributively acting IDS components are explained. A functional overview of the HEIDI approach is given in section 4. Section 5 contains a few examples of intrusion detection architectures that can be built with HEIDI components. The conclusion in section 6 summarizes the advantages of the approach and gives an outlook on the next research steps.

2. THE INTRUSION DETECTION EFFICIENCY PROBLEM

Intrusion detection as a security function deals with the monitoring of IT systems to detect security violations. The decision which activities are to be considered as security violations in a given context is commonly determined by a security policy. Due to the large amount of monitored data, the analysis can be only efficiently processed in an automatic manner, this means by intrusion detection systems. For the security violation finding, mainly two complementary paradigms are applied: anomaly detection and misuse detection [5]. Anomaly detection aims at the capturing of unusual system or user behavior. For these purposes, profile-based algorithms are widely used. Misuse detection focuses on the identification of well-known attacks. These attacks usually are described by patterns, so called signatures. An IDS that performs such a misuse detection matches the significant data stream against the signatures contained in its database. In the case of an individual match, the IDS triggers an alarm. Concerning practical aspects, misuse detection has proven to be the more effective paradigm in comparison with anomaly detection [6].

Beside the applied detection strategy, the quality of monitored data is of considerable importance for the effectiveness of an intrusion detection solution. The origin and the information content of this data essentially determine the amount of detectable security violations. In a scenario with many different attacks to find, the analysis often requires data from several sources. In this context, the most commonly used data types are network packets and audit data. Other sources, e.g. special log files or performance data, can also provide security relevant information. According to the locations where these basic data usually exist, two main types of intrusion detection systems are distinguished: network-based and host-based.

A comparison of effectiveness reveals a difference between network-based and host-based systems. While network-based technology has proved to be robust and hence is deployed in several commercial products [3], the development of field proven host-based systems seems to be more difficult. Today’s host-based solutions mostly are able to capture simple attacks, especially by matching single step signatures [7]. A main reason for this situation is the different nature of network packets on the one hand and audit data on the other hand. Network packets are normal process data, while audit data are only generated by the audit function after a security relevant system event took place. As a consequence, network-based IDSs are able to analyze packet headers in real time. They can avoid penetrations by filtering out suspicious packets. Host-based systems almost always work in a post-mortem manner. The only way to realize a true real-time audit analyzer is a system that is integrated in the security reference monitor (SRM) of the operating system. Because such a SRM is very slow, kernel-integrated intrusion detection systems like IDA [8] are just theoretical solutions, nowadays.

The post-mortem operation mode of host-based intrusion detection systems often is unfavorable for intrusion response actions. Since the most critical attacks are carried out by special software tools, the automatic initiation of countermeasures by the IDS often takes less effect. Therefore, for many host-based intrusion detection and response scenarios it is very important to minimize the time intervals between the real appearance of security violations and their detection by the IDS. Efficient host-based intrusion detection systems have to be optimized for a high detection speed, especially if they operate in distributed target environments and if many different security violations shall be found. This is the special intrusion detection efficiency problem we consider in the following.

3. DISTRIBUTIVELY ACTING IDS COMPONENTS

Intrusion detection systems are usually applied to monitor critical servers and complete local area networks. They are often deployed in connection with other security mechanisms, e.g. firewalls, and support a superior security management. Modern distributed host-based systems consist of a set of modules for capturing, preprocessing, and analyzing the basic data, and for archiving them if required.

As far as efficiency is concerned, the primary aspect of such a system is the distribution of the time-consuming analysis functionality. Two categories of intrusion detection systems can be distinguished: systems with a centralized and a decentralized analysis. Systems with a centralized analysis are simpler to implement, but the analysis function represents a performance bottleneck and a single point of failure. Furthermore, large amounts of data have to be transferred between the monitored hosts and the central processing unit. These effects we could also observe while testing our own centralized system AID [4]. If several processing units are used synchronization is required, but load distribution and efficiency are essentially better. Additionally, the processing units can be aligned in a hierarchical manner, e.g. in a two-layered scheme where the units at the lower level perform a detection of local attacks and a unit at the upper level is responsible for finding all distributed security violations.

The most approaches apply the centralized processing paradigm [6]. Only a few systems use the distributed analysis scheme. The best known approaches are DIDS (*Distributed Intrusion Detection System*), CSM (*Cooperating Security Managers*), AAFID (*Autonomous Agents for Intrusion Detection*), and EMERALD (*Event Monitoring Enabling Responses to Anomalous Live Disturbances*).

The DIDS was the first intrusion detection system that combined local audit evaluations, network monitoring, and a central alarm correlation. The distributed analysis function was proven by this approach [9]. CSM implemented an unusual idea [10]. Here every involved host contains a security manager. When a user logs on first time on a certain host, the security manager of this host becomes responsible for recording and analyzing all subsequent actions of this user, even if he moves to another host within the monitored network. This roaming of users, however, causes an enormous amount of data between the hosts. Therefore, the CSM approach is not applicable to a fast inspection of large amounts of audit data. The AAFID concept on the other hand addresses the problem of system load caused by using intrusion detection systems [11]. The main idea of AAFID consists in the application of many small, specialized, and hierarchically grouped entities. To meet the requirements of an efficient audit analysis, however, these components (filters, agents, transceivers, and monitors) are too limited in their performance. The fourth system, EMERALD, is a military sponsored development, which aims at the application of a flexible set of complex modules, the EMERALD monitors [12]. These monitors integrate both detection and response functionality. They are designed to be interoperable with many other security functions at a very high degree. The monitors can be connected among each other within a three-layered hierarchy. An EMERALD intrusion detection system is able to protect large networks, especially in critical enterprise environments. But this approach, as well as the other three, does not aim at a high audit analysis speed. There are no documented performance data available for these systems so far.

Further problems that are related to the design of distributed intrusion detection systems are the runtime adaptability, robustness, availability, fault tolerance, and inherent security. Especially in sensitive environments, there is a fundamental operational need for the survivability of intrusion detection systems under various conditions. Every time interval in which an IDS is not running represents a threatening situation. Modern distributed intrusion detection architectures used for the protection of critical infrastructures should ensure this dynamic adaptability as extensively as possible.

4. THE HEIDI APPROACH

The objective of the HEIDI approach (*High-Efficient Intrusion Detection Infrastructure*) is to provide an infrastructure for setting up tailored intrusion detection systems to speed up the detection capability. The term "infrastructure" means that a module system is defined which can be adapted to a specific intrusion detection architecture for a given target environment and application scenario, respectively. The main characteristics of such an architecture are the placement of necessary HEIDI modules and the general specification of their interconnectivity. Further refinements of the architecture towards a real intrusion detection system can be introduced by the integration of target-specific adaptations, e.g. interfaces for capturing host-specific audit data.

Beyond the structure and placement of modules, the application of fast analysis algorithms is the second essential factor for developing efficient intrusion detection solutions. HEIDI does not consider this level, so it is open for including any appropriate analysis technique. A special efficiency-oriented signature matching algorithm can be found in [13].

4.1 HEIDI components

HEIDI distinguishes three types of components: sensors, agents, and user interfaces. Sensors collect and preprocess the basic data, in most cases audit records. The agents provide the analysis units. They can cooperate among each other. The user interfaces serve for system management and user interactions.

4.1.1 HEIDI sensors

HEIDI sensors are specialized modules to collect and to handle basic security relevant data. They aim at a very fast reading, preprocessing, and forwarding of this information. Sensors can be placed at different points of the monitored hosts, depending on the applied security policy. Different sensors at one host are coordinated by the supervising local agent. Figure 1 depicts the generic structure of a sensor.

HEIDI sensors contain permanent components (illustrated in darker grey within Figure 1), e.g. the read interface, the transformation unit for data converting, and the transfer buffer. Other components are optional (illustrated in lighter grey), e.g. the pseudonymization unit. This unit is responsible for encrypting user identifying references in the basic data, e.g. user and group IDs in audit records. This is an important technique for supporting privacy-oriented analysis.

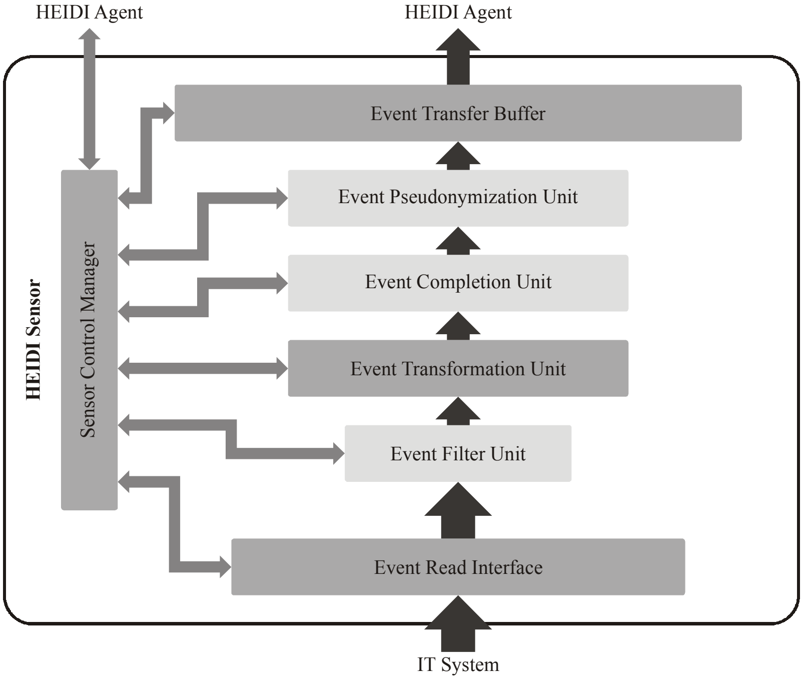

Figure 1: Structure of a HEIDI sensor

The most sensor components are connected by a data processing pipeline. They are supervised by the sensor control and configuration component. After reading the security relevant information from the local host, the data are preprocessed and forwarded to the local HEIDI agent as fast as possible.

4.1.2 HEIDI agents

After the fast capturing of basic data by the sensors, the second step to maximize the detection speed is to ensure an efficient analysis of these data. This is the task of the HEIDI agents. Beside the application of optimized analysis algorithms, they use an appropriate distribution of data. This distribution is based on a classification of the signatures into local and distributed contexts. To detect signatures with a local context, only locally preprocessed data are analyzed, while for signatures with a distributed context data from various agents are demanded.

The most efficient way to perform such an analysis in a network is to apply a combined execution scheme. Similarly to systems like DIDS, AAFID, and EMERALD, HEIDI prefers to match signatures with local context on the corresponding host. The detection of distributed attacks takes place on an appropriate central location. Unlike any other known system, HEIDI applies this hybrid concept in a stringent manner to achieve a maximal local analysis concentration and a minimal need for network traffic and delay. For this purpose, the concept includes an extended notion of signatures. HEIDI signatures are not only used for mapping complete security violation sequences. A signature can also represent a partial sequence of such an attack. This extension enables a hierarchy of agents to split the detection process for a distributed attack into a number of local sub-detections and a small amount of central combining. This principle is not applicable to all distributed attacks, but some critical security violations, e.g. several doorknob rattling variations, can be easily detected this way.

For the execution of all local and central detection processes, each involved host contains a single stationary HEIDI agent. Figure 2 shows the structure of an agent.

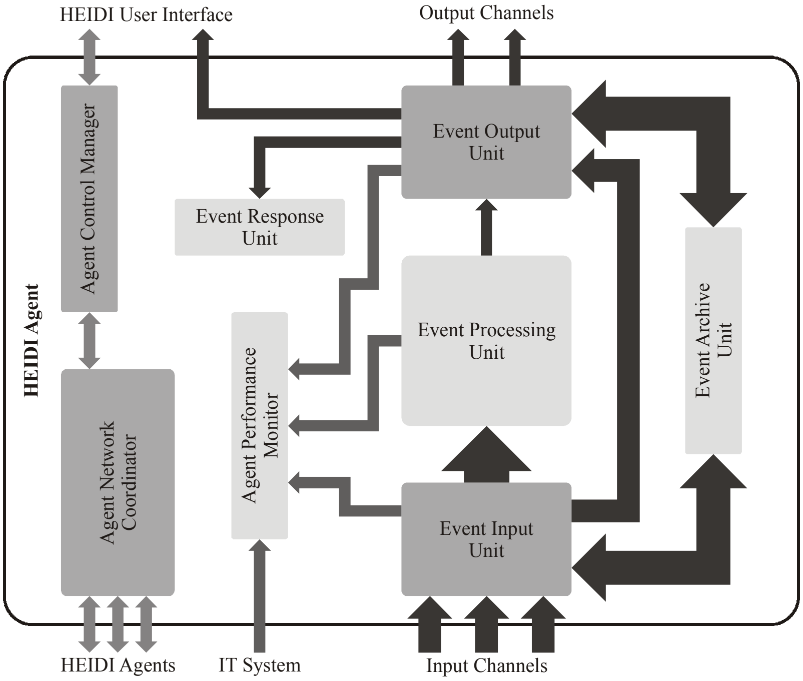

Figure 2: Structure of a HEIDI agent

A HEIDI agent receives the preprocessed information from all local sensors. It contains a central data processing pipeline, which consists of an input, a processing, and an output unit. Input and output units are responsible for receiving, synchronizing and transmitting the security relevant data. The analysis of the data takes place in the processing unit. Several analysis processes can run in parallel, e.g. a complete signature detection, a fast signature detection for particularly critical attacks, and a simple audit statistics. In addition to the detection capabilities, an agent can contain a response unit, which is able to initiate appropriate local countermeasures.

4.1.3 HEIDI user interfaces

A HEIDI user interface is a graphical application that enables a security operator to perform several tasks in the context of a given HEIDI-based IDS. The most important tasks are system configuration and the visualization of the detection results. Furthermore, a user interface can act as a link between a security management and the intrusion detection system.

Every HEIDI agent provides a single interface for the connection with a user interface. This connection can be either local or remote. Thus, a user interface can dynamically connect to a number of agents. Since several user interfaces can be attached to a HEIDI system at the same time, every interface has to read the corresponding configuration data periodically.

4.2 Handling overload situations

For assuring a continuous, robust, and efficient intrusion detection operation, HEIDI uses an adaptive mechanism to compensate temporary overload situations. In conventional systems, overload situations like an audit burst normally stop the execution of the intrusion detection system or cause a crash. To avoid this, HEIDI agents are able to delegate analyzing functionality to other agents. A destination agent receives the preprocessed data and the analysis state. The delegation functionality, the required number of destination agents, and the duration of the delegation depend on several conditions. They are calculated and negotiated dynamically. Normally, a re-delegation is carried out when the overload situation has disappeared. To estimate the load situation in the intrusion detection system, a HEIDI agent contains a monitor that evaluates the performance of both the host and some time-consuming agent components (see Figure 2).

5. HEIDI ARCHITECTURES

HEIDI sensors and agents can be combined to set up a hierarchical intrusion detection architecture for a given target environment, e.g. a single host or a local area network. Depending on the network structure and the applied security policy, special sensors and internal communication schemes can be configured. In this context, connectivity and data stream configuration are of vital importance. For every security violation to be detected, it must be determined which subset of modules is involved and where the data analysis is appropriately located. To offer a flexible and efficient setup, an agent also can act as a transceiver, that means it does not analyze, or as a delegation server. Such servers, which are HEIDI agents on demand, are required for enterprise networks with high failure safety requirements.

Figure 3 shows two different intrusion detection architectures. The left example outlines a two-layered hierarchy, the right example a three-layered. All illustrated hosts (the greater rectangles) are equipped with two sensors (smaller embedded rectangles) and the corresponding agent (greater embedded rectangle). Streams of preprocessed audit and result information are illustrated as arrows, whereas the thicknesses of the arrows indicate the transfer rates between the modules. In the left example, all agents are working on detection processes. The agent at the upper level is responsible for finding attacks with distributed context. This is the preferred HEIDI analysis scheme. In the right example, the white illustrated agents do not perform any analysis, so that their superior agent has to deal with a relatively high amount of incoming data. Such an adverse analysis distribution can exist in cases of local overload situations. The temporarily solution offered here is still better than a longer analysis suspension in most cases. In the right example, an agent at the third level serves as an overall result collector.

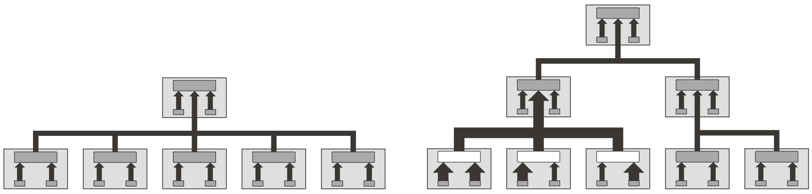

Figure 3: Two different HEIDI-based intrusion detection architectures

Figure 4 shows the expansion of the left architecture from Figure 3 by the integration of two delegation servers. The hosts on which these delegation servers run do not have sensors. In Figure 4 the upper depicted delegation server is configured to exclusively help the upper-level agent in the regular detection hierarchy. The lower server is dedicated to handle overload situations for all low-level hosts in the regular detection hierarchy. The left example shows a burst situation at two low-level hosts. In this case the lower server overtakes the data analysis partially. In the right example, there are also two hosts in an overload state. One of them is the upper-level host in the regular detection hierarchy. In this case, the processing capacities of both delegation servers are used.

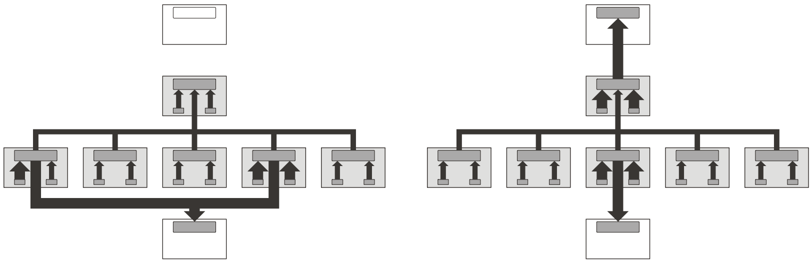

Figure 4: Scenario with two different overload situations for a single architecture

6. CONCLUSION

In the paper we have presented the basics of the intrusion detection infrastructure HEIDI. The HEIDI approach aims at a module system to set up efficient and tailored intrusion detection systems, especially for local area networks. The module system provides a set of specialized sensors for basic data capturing and flexible agents for data analysis. The analysis combines local and central signature matching processes. Furthermore, a HEIDI-based intrusion detection system is capable to react to overload situations by delegating analysis functionality among the communicating agents.

So far only a very few intrusion detection approaches or systems have the potential to practically overcome the efficiency problem of the host-based intrusion detection paradigm. It has shown that only decentralized analysis approaches like DIDS [9], AAFID [11], and EMERALD [12] are capable to meet near real-time requirements. Unfortunately, none of these systems aim at an efficient intrusion detection solution, but from an architectural point of view some aspects are comparable with HEIDI. The systems CSM [10] and EMERALD are characterized by the application of large and complex modules. This feature is similar to the HEIDI agents. In contrast to this, AAFID uses a great number of small and specialized entities. In HEIDI, the sensors play a similar role. Since HEIDI uses both complex and small modules, it is also comparable to DIDS. Depending on the different development targets, each of these systems has special module-intern structures. Regarding the module interconnection capabilities, HEIDI seems to be as potentially as EMERALD and AAFID, while DIDS and CSM are functionally limited in this context.

The implementation of the HEIDI infrastructure modules is still in progress. Currently implementations of various sensors, e.g. for capturing audit data under Sun Solaris and Microsoft Windows NT/2000/XP and for the stack-based collecting of TCP/IP packets under these operating systems are available. After finishing the implementation of the HEIDI agent, we plan to set up a first example intrusion detection system, which functionally corresponds to our system AID [4]. By comparing the two AID variants, we will evaluate the performance of the HEIDI concept. Thereafter, the efficiency and usability of the HEIDI approach will be investigated with different intrusion detection architectures.

REFERENCES

[1] Clinton Administration (ed.): The Clinton Administration's Policy on Critical Infrastructure Protection: Presidental Decision Directive 63. Washington D.C., 1998.

[2] Denning, Dorothy E.: Information Warfare and Security. Addison Wesley Longman, Inc., Reading, 1999.

[3] Meier, Michael; Holz, Thomas: Intrusion Detection Systems List and Bibliography. http://www-rnks.informatik.tu-cottbus.de/en/security/ids.html, 2002.

[4] Sobirey, Michael; Richter, Birk; Koenig, Hartmut: The Intrusion Detection System AID - Architecture, and experiences in automated audit analysis. In: Horster, Patrick (ed.): Proceedings of the IFIP TC6/TC11 International Conference on Communications and Multimedia Security at Essen, Germany, 23rd - 24th September 1996. Chapman & Hall, London, 1996, pp. 278-290.

[5] Mukherjee, Biswanath; Heberlein, L. Todd; Levitt, Karl N.: Network Intrusion Detection. In: IEEE Network 8 (1994), No. 3, pp. 26-41.

[6] Axelsson, Stefan: Research in Intrusion Detection Systems: A Survey. Goeteborg, Chalmers University of Technology, Technical Report No. 98-17, 1998.

[7] Meier, Michael; Bischof, Niels; Holz, Thomas: SHEDEL - A Simple Hierarchical Event Description Language for Specifying Attack Signatures. In: Ghonaimy, M. Adeeb (ed.); El-Hadidi, Mahmoud T. (ed.); Aslan, Heba K. (ed.): Proceedings of IFIP TC11 17th International Conference on Information Security (SEC2002), May 7-9, 2002, Cairo, Egypt. Kluwer Academic Publishers, Boston, 2002, pp. 559-571.

[8] Fischer-Huebner, Simone: IDA (Intrusion Detection and Avoidance System): Ein einbruchsentdeckendes und einbruchsvermeidendes System. Shaker, Aachen, 1993.

[9] Snapp, Steven R.; Smaha, Stephen E.; Teal, Daniel M.; Grance, Tim: The DIDS (distributed intrusion detection system) prototype. In: USENIX Association (ed.): Proceedings of the Summer 1992 USENIX Conference. USENIX Association, Berkeley, 1992, pp. 227-233.

[10] White, Gregory B.; Pooch, Udo W.: Cooperating security managers: Distributed intrusion detection systems. In: Computers & Security 15 (1996), No. 5, pp. 441-450.

[11] Spafford, Eugene H.; Zamboni, Diego: Intrusion detection using autonomous agents. In: Computer Networks 34 (2000), No. 4, pp. 547-570.

[12] Porras, Phillip A.; Neumann, Peter G.: EMERALD: Event Monitoring Enabling Responses to Anomalous Live Disturbances. In: NIST (ed.); NCSC of the NSA (ed..): Proceedings of the 20th NISSC, 1997. National Institute of Standards and Technology, Gaithersburg, 1997, pp. 353-365.

[13] Holz, Thomas; Meier, Michael; Koenig, Hartmut: High-Efficient Intrusion Detection Infrastructure. Estoril, NATO Symposium "Real Time Intrusion Detection", 2002.