

Current Intrusion Detection Techniques in Information Technology - A Detailed Analysis

G. Mohammed Nazer

*Ph.D. Research Scholar, PRIST University
Asst. Prof./Computer Applications, IFET College of Engineering, Villupuram
E-mail: kgmohammednazer@gmail.com*

A. Arul Lawrence Selvakumar

*Director/Computer Applications, Adhiparasakthi Engineering College, Melmaruvathur
E-mail: aarul72@hotmail.com*

Abstract

It is a known fact that computer and network systems have certain design flaws which leads to security hazards. Intruders can exploit the security flaws and break the computer systems, and is very expensive and sometimes nearly impossible to fix all the design and programming errors. This suggests that prevention-based approaches are no more reliable and hence intrusion detection is needed as a last line of defense. This paper presents a nomenclature of intrusion detection systems that is used to do a survey and identify a number of research prototypes. This classification consists of detection principles and the operational aspects of intrusion detection system. These classifications can be used efficiently leading towards a number of future research works in the field of intrusion detection.

Keywords: Security, Intrusion Detection, Taxonomy, misuse and anomaly detection.

1. Introduction

From the period of seminal work done by Denning in 1981, so many intrusion-detection prototypes have been emerged. Intrusion detection prototypes have been emerged in order to protect the information system from security flaws. Undeniably, a taxonomy of security flaws by Landerwehr et al shows that the information system suffer from security vulnerabilities regardless of their purpose, manufacturer and origin. It is very difficult to build and maintain network systems that are not at risk to attacks.

In the light of this, the main motivation for taking an in-depth approach to the different kinds of detectors that have been employed is that it is natural to assume that different intrusion detection principles will behave differently under various circumstances. A detailed look at such intrusion detection principles is thus in order, giving us a base for the study of how the operational effectiveness is affected by the various factors. These factors are the intrusion detector, the intrusion that we wish to detect and the environment in which we wish to detect. This paper introduces a classification of intrusion detections at a time when so many commercial tools are increasingly becoming available together with the survey of important research intrusion detection systems to date. Several surveys have already been published (James and Harrell, 1996; Mansour Esmaili and Josef, 1995; Jeremy

Frank, 1994; GunarLiepins, 1989; Teresa F. Lurt, 1988; Noelle McAuliffe, 1990), but the growth of the intrusion detection has been such that many new projects have appeared in the mean time. Hence we shall present an updated and organized classification of the intrusion detection systems.

This paper is organized as follows. Section 2 describes the general architecture of intrusion detection system, Section 3 describes the proposed taxonomy that we use, Section 4 describes a summary of existing intrusion detection tools and prototypes and Section 5 presents the reusability issue of intrusion detection systems and their components.

2. General Architecture of Intrusion Detection System

An intrusion detection monitors dynamically the system actions in a given environment and decides whether these actions resembles an attack. An intrusion detection system at its primitive level is a detector that processes information coming from the system that is to be protected.

Figure1: A simple intrusion detection system

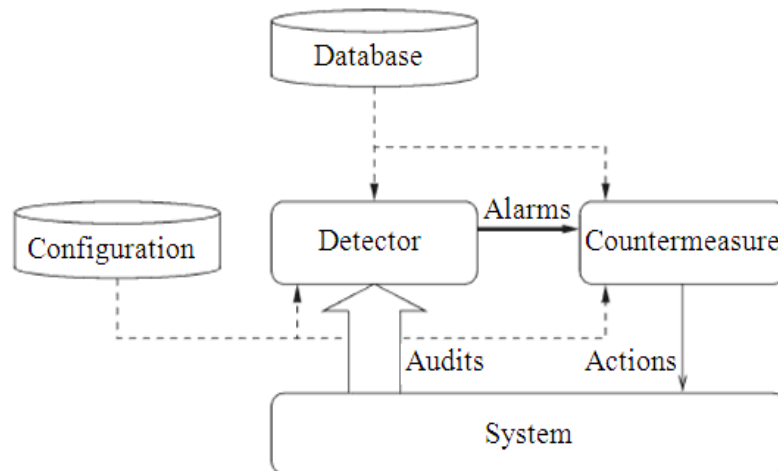


Fig.1 represents a simple intrusion detection system and uses three kinds of information namely **long term information** related to the technique used to detect intrusions (knowledge based attacks), **configuration information** about the current state of the system and **audit information** describing the events occurring on the system. The role of the detector is to eliminate unnecessary information from the audit trail and present a synthetic view of the security related actions taken by the users. A decision is then made to evaluate the probability that these actions can be considered as symptoms of an intrusion.

The following five measures to evaluate the efficiency of an intrusion detection have been highlighted.

- **Accuracy** – Inaccuracy occurs when an intrusion detection system flags as anomalous or intrusive a legitimate action in the environment.
- **Performance** – The performance of an intrusion detection system is the rate at which audit events are processed. If the performance of the intrusion detection is poor, then real-time detection is not possible.
- **Completeness** – Incompleteness occurs when the intrusion detection system fails to detect an attack. This measure is very difficult to evaluate because it is impossible to have a global knowledge about the attacks or abuses of privileges.
- **Fault Tolerance** – An intrusion detection system should itself be resistant to attacks, especially denial of service, and should be designed with this goal in mind. This is very

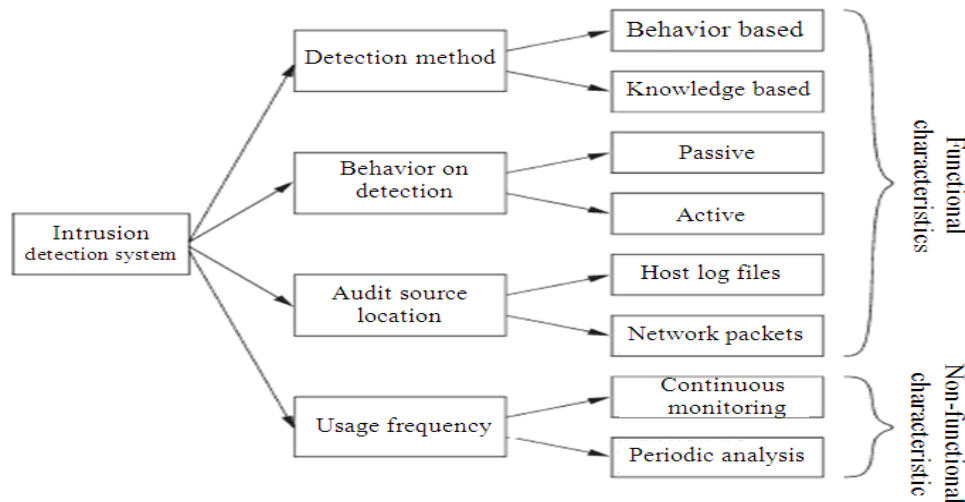
important because most of the intrusion detection systems run on top of commercially available operating systems or hardware, which are known to be vulnerable to attacks.

- **Timeliness** – An intrusion detection system has to perform and propagate its analysis as quickly as possible to enable security procedures. This implies more than the measure of performance, because it not only encompasses the intrinsic processing speed of the intrusion detection system, but also the time required to propagate the same and to react to it.

3. Intrusion Detection System – Taxonomy Elements

There are a number of concepts we use to classify the intrusion detection systems, presented in Fig. 2.

Figure 2: Characteristics of intrusion-detection systems



The detection method describes the characteristics of the analyzer. When the intrusion detection system uses information about normal behaviour, then we classify it as behaviour based. When the intrusion detection system uses the information about the attacks, then we classify it as knowledge based.

Behaviour on detection describes about the response of the intrusion detection system to attacks. When it actively react to the attacks, then it is said to be active and when it merely raises an alarm then it is said to be passive.

The audit source location differentiate among intrusion detection systems based on the kind of input information (audit trials, system logs or network packets) they analyze.

Usage frequency is an orthogonal concept. Some intrusion detection systems have real-time continuous monitoring capabilities, whereas others must be run periodically.

The three items are classified as “functional characteristics” since they refer to the internal operations of the intrusion detection engine. The fourth characteristic distinguishes RTID (Real Time Intrusion Detection) from scanners used for security assessment and this has been classified as “non-functional characteristics”.

There are two complementary trends in intrusion detection:

1. The search for evidence of attacks based on the knowledge collected from known attacks and is referred to as misuse detection or detection by appearance.
2. The search for deviations from the model of unusual behaviour based on the observations of a system during a normal state and is referred to as anomaly detection or detection by behaviour.

3.1. Misuse Detection (Knowledge-Based Intrusion Detection)

These apply the knowledge accumulated/collected about specific attacks and system vulnerabilities. The intrusion detection system contains information about these vulnerabilities and looks for attempts to exploit them. If such an attempt is detected, an alarm is raised. Therefore the accuracy of knowledge-based intrusion detection system is considered as good. But their completeness requires the knowledge of attacks be updated regularly.

Advantages of knowledge-based approaches are that they have the potential for very low false alarming rates, and that the contextual analysis proposed by the intrusion detection is detailed, which makes it easier for the security officer to take necessary corrective actions.

Disadvantages of knowledge-based approaches are the difficulty in gathering the information about known attacks and keeping them abreast with new vulnerabilities and environments. Moreover it is a more time consuming task. This approach also has to face the generalization issues such as operating system, version, platform, application, etc. Also, detection of insider attacks involving an abuse of privileges is deemed to be more difficult because no vulnerability is actually exploited by the attacker.

3.1.1. Expert Systems

These are used primarily by knowledge-based intrusion detection systems. The expert system contains set of rules that describe the attacks. Audit events are then translated in to facts carrying their semantic signification in the expert system and the inference engine draws conclusions using these rules and facts. This method increases the abstraction level of the audit data by attaching a semantic to it.

Rule-based languages are a natural tool for modeling the knowledge that experts have collected about the attacks. This approach allows a semantic browsing of the audit trial in search of evidence of attempts to exploit known vulnerabilities. They are also used for identifying proper utilisation of security policy. Knowledge about an attacker's behaviour includes their goals and the required actions for the goals. This rule-based language has the limitations of knowledge engineering (completeness issue) and processing speed (performance issue).

3.1.2. Signature Analysis

This analysis uses the same knowledge-based approach as expert systems but the acquired knowledge is transformed into information that can be found in the audit trial. This method decreases semantic level of the attack description. This technique uses a very efficient implementation and is therefore used as commercial products.

3.1.3. Petri Nets

IDIOT is a system that was developed by COAST (now called as CERIAS – Centre for Education and Research in Information Assurance and Security). The basic principle of IDIOT is to employ colored Petri-nets (CPN) for signature-based intrusion detection. The benefits of CPNs are their generality, simplicity and graphical representation. Many system administrators are assisted in writing their own signatures of attacks and integrating them in IDIOT. The generality of CPNs paves ways for integrating very complex signatures so easily. However, the matching process of a complex signature with the audit information is very difficult.

3.2. Anomaly Detection (Behavior-Based Intrusion Detection)

Anomaly detection techniques assume that an intrusion can be detected by observing a deviation from normal behaviour of the system. The model of normal behaviour is extracted from reference information collected by various means. The intrusion detection system then compares this model with the current activity and if a deviation is identified, then an alarm is raised. Therefore the intrusion detection is complete but its accuracy is a difficult issue.

Advantages of behaviour-based approaches are that they can detect attempts to exploit new and unforeseen vulnerabilities. They also help in detecting “abuse of privileges” types of attacks that do not actually involve exploiting any security vulnerability. Disadvantage of this approach is the high false rate of alarm because the entire scope of the behaviour of an information system may not be covered during the learning phase. Also, behaviour can be changed over time, creating the need for periodic on-line retraining of the behaviour profile, resulting either in the unavailability of the intrusion detection system or in additional false alarms.

3.2.1. Statistics

This is the widely used tool to build behaviour-based intrusion detection system. The user behaviour or the system behaviour is measured by a number of variables sampled over time such as login and logout time of each session, the resource duration and the amount of processor-memory-disk resources consumed during that session. The original model keeps averages of all these variables and detects whether thresholds are exceeded based on the standard deviation of the variable. This model is too simple to represent the data faithfully. Even after comparing the variables of individual users with aggregates group statistics does not yield much improvement. Therefore, a more complex model has been developed (Harold and Alfonso, 1991; Harold and Teresa, 1993), which compares profiles of long-term and short-term user or system activities. These profiles are periodically updated as the behaviour of user activities and this model is now used in a number of intrusion-detection tools and prototypes.

3.2.2. Expert Systems

Expert systems have also been used for behaviour-based intrusion detection. This approach is useful for policy based usage profiles, but is less efficient than the statistical approach for processing large amount of audit information.

3.2.3. User Intention Identification

User identification is a technique developed during SECURENET project (Paul et al, 1994). This technique models the normal behaviour of the users by the set of high-level tasks they have to perform. These tasks are then refined into actions, which in turn are related to the observed audit events of the system. Whenever an action that does not fit the task pattern, an alarm is raised.

3.2.4. Computer Immunology

This approach has been described by Forrest et al (1997). It builds a model of normal behaviour of the UNIX network services, rather than that of the users. It consists of short sequences of system calls made by the processes. Attacks that exploit flaws in the code are likely to take unusual execution paths. This tool collects a set of reference audits, which represent the appropriate behaviour of the service, and extracts a reference table containing all the known “good” sequences of system calls. These patterns are then used for live monitoring to check whether the sequences generated are listed in the table; if not, the intrusion-detection system generates an alarm. This technique has a very low false alarming rate if the reference table is exhaustively enough. One disadvantage in this technique is that it does not protect the configuration errors in a service.

3.3. Active Versus Passive Intrusion Detection

Passive systems respond by notifying the proper authority, and they do not themselves try to mitigate the damage done, or actively seek to harm the attacker. Active systems are further divided into two classes:

1. Those that exercise control over the attacked system, i.e. they modify the state of the attacked system to thwart or mitigate the effect of the attack. Such control can take the

form of terminating the network connections, increasing the security logging, killing the errant processes, etc.

2. Those that exercise control over the attacking system, i.e. they in turn attack in an attempt to remove the attacker's platform of operation.

Of the systems surveyed, one severs the network connections in response to suspected attacks, and one block suspect system calls, terminating the process if this option fails. This mode of defense is generally difficult to field since it opens up the system to obvious denial of service attacks.

3.4. Host-Based Versus Network-Based Intrusion Detection

When the intrusion detection tools were designed, the target environment was a mainframe computer and all the users were local to the host. This simplified greatly the intrusion-detection task as interaction from outside was rare. This tool analysed the audit information provided by the mainframe, either locally or on a separate machine and reported as security suspicious events. As the focus shifted from mainframe environment to distributed network environment, several prototypes of intrusion detection systems emerged.

The first research in this area was to get host-based intrusion detection systems to communicate (Jagannathan, 1993). In a distributed network environment, users hop from one machine to another, possibly changing their identities during their moves and launching their attacks on several systems. Therefore, the local intrusion-detection system on the workstation has to exchange the information with its peers. This exchange of information takes place at several levels either by exchanging the raw audit trail over the network or by issuing alarms that comes from a local analysis. Both solutions incur costs; transferring audits has potentially huge impact on network bandwidth, whereas processing them locally affects the performance of the workstations.

With the widespread use of the Internet, the intrusion-systems have become focused on attacks to the network itself. Network attacks such as DNS spoofing, TCP hijacking, port scanning, ping of death, etc. cannot be detected by examining the host audit trail. Therefore specific tools have been developed that sniff network packets in real time, searching for these network attacks. In addition, a number of classical attacks against servers can also be detected by parsing the payload of the packet and looking for suspicious commands.

Hybrid approaches have also been developed that use both network-based and host-based intrusion-detection tools in a multi-host environment DIDS (Stevan et al., 1991) uses Haystack (Stephan, 1988) running on each host to detect local attacks and NSM (Todd et al., 1990) to monitor the network.

3.5. Sources of Information

There are two sources of information:

1. Host-based information source
2. Network-based information source

3.5.1. Host-Based Information Sources

These are the only way to collect information about the activities of the users of a given machine. They are also vulnerable to alterations in the case of a successful attack. This creates an important real-time constraint on host-based intrusion-detection systems, which have to process the audit trail and generate alarms before an attacker taking over the machine can threaten the audit trail.

3.5.1.1. System Sources

There are plenty of system commands from every operating system to obtain a snap shot of information on the processes currently active on the computer. Commands such as ps, pstat, vmstat and

getrlimit from UNIX environment provide information about events because these commands refer the kernel memory directly.

3.5.1.2. Accounting

It is an oldest information source in collecting system behaviour. It gives the information about consumption of shared resources such as processor time, memory, disk or network usage and which application is currently working by the users. In the UNIX, accounting is a universal source of information. The accounting record format is the same all on UNIX workstations, compressed information to gain disk space, and the overhead incurred by the recording process is very small, whereas it is well integrated in the modern operating systems and is very easy to setup and to exploit them. However, accounting information has its own drawbacks such as

- Lack of parameterization
- Lack of precise time stamp
- Lack of precise command identification
- Absence of system daemon activity
- Delay in obtaining information

Owing to these drawbacks, it is not used for knowledge-based intrusion detection and rarely for behaviour-based intrusion detection.

3.5.1.3. Syslog

Syslog is an audit service provided by the UNIX application, which receives a text string from the applications, prefixes it with a time stamp and the name of the system on which the application runs and then it archives it either locally or remotely. Syslog is not known for its security, as Syslog daemons on several UNIX operating systems have been the subject of CERT commands Cert Coordination Center, 1995) showing the exploitation of buffer overflows in the Syslog daemon to execute arbitrary code. It is very easy to use which has made many application developers to use it as their audit trail. A number of applications and network services use it, such as login, sendmail, nfs, http and this also includes security-related tools such as sudo, klaxon, TCP wrappers. Although Syslog is a light weight audit source that does not generate large amount of audit data per machine, a large network can generate a large number of messages, very few of which are security-related. Swatch (Stephen and Todd Atkins, 1993) reduces the burden of the system administrator by correcting messages and highlighting security-related messages.

3.5.2. Network-Based Information Sources

SNMP information – The Simple Network Management Protocol (SNMP) Management Information Base (MIB) is a repository of information used in the network management. It provides configuration information such as routing tables, network address, names etc, and performance or accounting data. This part of the section describes the experiments performed in the SECURENET project (Paul et al., 1994) to use SNMP v1 common MIB for Ethernet and TCP/IP.

The exploration on the SECURENET project was about whether the counters maintained in this MIB are usable as an input information for an anomaly detection system, by examining the counters at the interface level, because this was the only place that we can identify the difference between information sent over the wire and the operating system transmitted information via the loop-back interface. The prototype collected increments on the number of bytes transmitted and received at every five minutes. The outcome of the standard deviation analysis was larger than the average for almost all sets collected during daytime activity and no correlation was observed between two interfaces. This study shows that SNMP MIBs are a potentially interesting candidate as an audit source for intrusion-detection systems. The demise of SNMP v2 owing to a lack of consensus on the security features has certainly dampened its interest to the intrusion-detection community. However, with the rise of SNMP v3, new projects are taking advantage of its features for intrusion-detection tools (Franck et al., 1997).

3.5.2.1. Network Packets

Network packets is an efficient means for gather information about the events that occur on the network. This is consistent with the trend of moving from a centralized to a distributed network environment Filtering the network packets before they enter the server is the most efficient way to monitor the server. Moreover it is very much consistent with the occurrence of denial-of-service attacks.

Most of these denial-of-service attacks originate from the network and must be detected at the network level, as a host-based intrusion-detection system that does not have the capability to acquire this type of audit information. There is an inherent duality in network sniffers, which is also apparent in the firewall with its differences between application-level gateways and filtering routers (Steven and Cheswick, 1994). If the analysis is carried out in pattern matching, signature analysis then the intrusion-detection system can perform its analysis quickly, but does not take into account session information, which could span several network packets. If the intrusion-detection system acts as an application gateway and analyzes each packet with respect to the application or protocol being followed, then the analysis is more thorough and expensive. This approach addresses several problems:

- Detection of network-specific attacks. There are a number of network attacks, particularly denial-of-service that cannot be detected in a timely fashion by searching for audit information on the host, but only by analyzing network traffic.
- Impact of auditing on the host performance. Information is collected entirely on a separate machine, with no knowledge of the rest of the network. Therefore, installation of such tools is facilitated because, both in terms of configuration and performance, they do not impact the entire environment.
- Heterogeneous audit trial formats.
- Certain tools analyze the payload of the packet, which allows the detection of attacks against hosts by signature analysis.

Following are the drawbacks of network packets:

- It is more difficult to identify the person when an intrusion is identified.
- With switched networks, it is not obvious where the sniffer should best be placed. Some tools are located on switches, others at gateways between the protected system and the outside the world. The former yields better audit information but is also more expensive.
- Encryption makes it impossible to analyze the payload of the packets and therefore to hide a considerable amount of important information on these tools.
- Systematic scanning at the firewall is difficult because it might create bottlenecks.
- These tools are inherently vulnerable to denial-of-service attacks if they rely on a commercial operating system to acquire network information.

Research is also being conducted in this area. After IDDES and NIDES, SRI is now developing a prototype called Emerald (Phillip, 1998) to deal with the analysis of network traffic.

3.6. Continuous Versus Period Intrusion Detection

A dynamic intrusion-detection tool performs a continuous, real-time analysis by acquiring information about the actions taken on the environment immediately after they happen. A static intrusion-detection tool periodically takes a snapshot of the environment and analyzes this snapshot, looking for vulnerable software and configuration errors. These static tools assess the security level of the current configuration of the environment. Examples of these tools are COPS (Dan Farmer, 1993; Daniel and Spaffors, 1990) and Tiger (David et al., 1993) for host environments and Satan (Dan and Venema, 1993) and Ballista (Secure Networks, Inc., 1997) (now called CyberCop Scanner (Network Associates Inc., a1998) since the buyout of Secure Networks by Network Associates Inc.) for networks. In the same category are virus detectors, which scan the disks looking for patterns identifying known viruses.

These checks include verifying the version of the applications installed to ensure that the latest security patches have been applied, checking for weak passwords, verifying the contents of special files in user's home directories.

First of all, security patches are not necessarily available on legacy systems, which cannot be upgraded without losing their operational requirements. Then, running these security assessment tools is often a lengthy process, particularly in a networked environment where every system has to be checked individually. Therefore, the security exposure between two consecutive runs might be significant, approximately a day or so.

These tools, as well as others specifically developed for that purpose (e.g. Tripwire (Gene and Eugene, 1994) or ATP (David and Cotrozzi, 1993)) can be used to detect the traces of an intrusion. Such traces can be the replacement of a given application by an older, vulnerable one, which would be signaled by COPS or Tiger to the system administrator as a potential intrusion. Tripwire (Gene and Eugene, 1994) extends this principle by computing the signature of a large set of system files and comparing it with a database of reference signatures kept in safe place, therefore rendering the change-detection process systematic. Dynamic intrusion-detection tools monitor the actions that take place on the system. Monitoring takes place either in real time or in batch, reviewing audit files or network packets accumulated over a given period of time., Dynamic monitoring implies real-time analysis and allows a constant assessment of the security of the system.

4. Intrusion Detection System Tools

The below table (Table 1) presents a selection of intrusion-detection tools that we encountered and shows a taxonomy of their components. The selection merely illustrates the notions described in this paper and implies no judgment of the quality of the tool, product or method on our part. Also, the number of tools and prototypes being developed throughout the world is such that an exhaustive list is difficult to compile and we shall continue to add entries to this table.

This table contains more host-based intrusion-detection systems. However, this is not the trend in intrusion detection, which is towards network information and protection of the infrastructure. There are more network-based intrusion-detection products (Internet Security Systems, 1997; WheelGroup, 0000) commercially available today than host-based ones (Haystack, 1997; Network Associates Inc., b1998), as well as recent research projects still under development. The main reason for this is probably the ease of installing a network-based tool, the performance degradation experienced by systems when an audit is started. This table also shows that, even though many techniques have been explored for intrusion detection, most commercial products available today implement one and only one technique and that the majority of the recent ones use signatures for two reasons:

- The knowledge-based approach is easier to implement than the behaviour-based one. In fact, the cost in terms of false alarms of the behaviour-based techniques has hitherto made them inappropriate for commercial intrusion detection.
- Speed is essential in processing the audits and preempts the expressiveness of the technique. Therefore, signatures are used instead of rules.

Table 1: Panorama of intrusion detection system

IDS origin	IDS Name	Time frame	Ref	Knowledge based IDS				Behaviour based IDS				HB	NB	
				E S	SA	PN	STA	STATE	BS	NN	UII			
Univ.	ASAX	1990-1997	Teresa, 1993	X									X	
AT and T	Computer Watch	1987-1990	Todd et al., 1990							X			X	
USAF	Hystack	1987-1990	Stephen Smaha,1988					X					X	

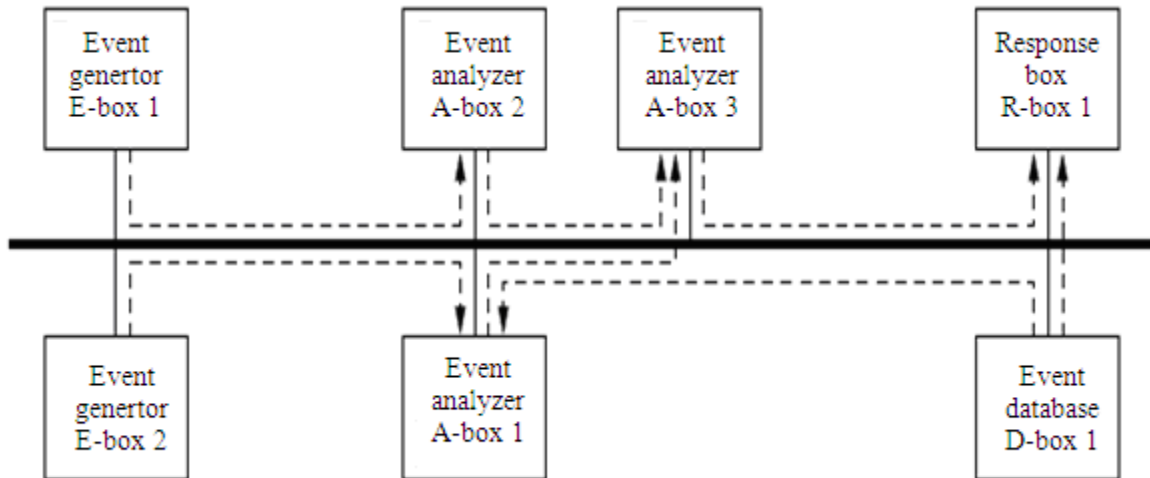
Table 1: Panorama of intrusion detection system - continued

CS Telecom	DIDS	1989-1995	Steven R. Snapp, 1991	X				X				X	X
	Hyperview	1990-1995	Daniel and Spaffors, 1990	X				X				X	
SRI	IDES	1983-1992	Haystack, 1997	X				X				X	
	NIDES	1992-1995	Paul et al., 1994	X				X				X	X
purduniv	Emerald	1996-	Phillip A. Porras, 1998		X			X					X
	IDIOT	1992-1997	Sandeep Kumar, 1994			X							X
UC davis	NSM	1989-1995	Network Associates Inc., 1998b		X			X					X
	Gr IDS	1995-	Marcus J. Ranum, 1997						X				
LANL	W and S	1987-1990	H.S. Vaccaro, 1989					X				X	
	Nadir	1990-	Steven et al., 1991	X				X				X	
Gis/wheel group	NetRange	1995-	Dan and Venema, 1993	X								X	X
	Real secure	1995-	Stephen, 1988		X							X	X
Securest consortium	Secure net	1992-1996	Paul Spirakis, 1994	X				X	X	X	X		
	Stalker	1995-	Network Associates Inc., 1998a		X						X		
Associates inc	WebStalker Cyber-Cop Server	1997-	Network Associates Inc., 1998a					X					X
	Star	1991-1992	Phillip Porras, 1992				X				X		
Stanford univ	Ustat	1992-1993	Secure Networks, Inc., 1997			X					X		
	Swatch	1992-1993	Noelle, 1990	X								X	
MNC and NCSU	JiNao	1995-	WheelGroup, 0000	X				X					X

Abbreviations used: **ES** Expert System; **SA**; Signature Analysis; **PN**: Petri Net; **STA**: State Transition Analysis; **STAT**: Statistics; **NN**: Neural Network; **UII**: User Intent; **HB**: Host Based and **NB**: Network Based

5. The Reusability Issue

The capability to reuse existing components in an environment different from the original one is the greatest challenges faced by the intrusion-detection products and prototypes. This is due to incompatible audit and alarm formats. A working group has been created under the auspices of the Defence Advanced Research Projects Agency (DARPA) to develop a common intrusion-detection framework (CIDF) (Stuart et al., 1998). This work aims primarily at coordinating the many projects funded by DARPA that are concerned with intrusion detection, and ensuring that the tools developed are able to interoperate. The CIDS description of an intrusion-detection system (Fig. 3) is mode detailed and defines all the possible roles of components that can comprise an intrusion-detection system. The interfaces of each of these component roles are then defined, so that any CIDF-compliant box can be integrated into a larger tool.

Figure 3: CIDF description of an intrusion detection system

The CIDF group is currently in the process of joining the Internet Engineering Task Force to make their work a standard in the Internet world. The Fig.3 does not include the system being monitored. Obviously, the boxes run on hardware of some kind, most likely the system that produces the events in the case of the event box, or on either the monitored system or a specific hardware in the case of other boxes.

CIDF defines four kinds of components for an intrusion-detection system and very specific roles for each of them. All these components deal with gidos (generalized intrusion-detection objects), which are represented via a standard common format. Gido streams are represented as dashed arrows in Fig. 3. Gidos carry information that is moved around in the intrusion detection system. From a semantics point of view, gidos currently represent either audit events that occurred in the system or an analysis of those audit events (henceforth referred to as alarms).

- Event boxes (E-boxes) generate audit events that are processed by the intrusion-detection system. E-boxes typically run on the system that generates the audit events, where they collect the audit events and make them available to other components of the intrusion-detection system. E-boxes produce audit events but do not consume them. Their task is to sample the particular environment for which they are specialized, and to turn occurrence in that environment into CIDF gidos for use by other components. Fig. 3 shows two event-generator boxes delivering audit events to two analyzers.
- Analysis boxes (A-boxes) process (similar to detector component) events from the E-boxes to create alarms. Analyzers take in gidos and analyze their significance (policy violations, anomalies, intrusions). Their conclusions are turned out as alarms. In the figure 3, two of the three A-boxes receive audits from E-boxes, whereas the third one aggregates information and passes it to the countermeasures.
- Database boxes (D-boxes) store events for later retrieval. D-boxes are gidos archivers. They receive events sent by E-boxes or A-boxes, store them for long-term keeping and provide a retrieval and query service. Configuration and database are private to each A-box and must be maintained independently. In Fig. 3, the D-box provides gidos to one of the analyzers and to the countermeasures.
- Response boxes (R-boxes) (sometimes called countermeasure boxes) apply countermeasures to the system according to the alarms generated. They are the active arm of the intrusion-detection system; they enforce the decisions made in response to attacks. In Fig. 3, an R-box takes its input from the third A-box.

CIDF is work-in-progress. The most important contribution of CIDF is to define interfaces by which the different kinds of boxes can communicate, thus introducing the reusability of components in

intrusion detection. It is a fact that as of today, a large number of research prototypes and products have been developed, but these heterogeneous developments do not allow the reusability of techniques or tools in different environments. Currently, the CIDF effort is giving birth to an IETF working group chartered to create standards in the intrusion-detection area. The current draft charter being discussed states that “the purpose of the Intrusion Detection Working Group is to define data formats and exchange procedures for sharing information of interest to intrusion-detection systems and their management infrastructure”.

6. Summary and Concluding Remarks

Both the research community and commercial companies pay more interest in the Intrusion Detection area. Research prototypes continue to be created and commercial products based on early research are now available. In this paper, we have given an overview of the current state-of-the-art of intrusion detection, based on a proposed taxonomy illustrated with examples of past and current projects. This classification highlights the properties of intrusion-detection systems and covers the past and current developments adequately in this area. Information sources for these tools are currently either a C2 audit trail, Syslog or network packets.

Whereas system sources were widely used in the early stages of research, the current focus of research prototypes as well as products is to protect the infrastructure, rather than the end-user station and this paradigm has introduced the usage of network sniffers that analyze packets. There are still quite a number of research issues concerning the efficiency of network and host audit sources, the formatting and existence of a common audit trail format. There are also a number of open research issues concerning the analysis of the audit trail. Signature analysis is clearly in the commercial domain now, but it has been shown to be insufficient to detect all attacks. Therefore work is still in progress to experiment with new approaches to both knowledge-based and behaviour-based intrusion detection.

References

- [1] Steven M. Bellovin and William R. Cheswick, Network firewalls, *IEEE Communications Magazine*, 32(9) (1994) 50-57.
- [2] James Cannady and Jay Harrell, A comparative analysis of current intrusion detection technologies, Proc. 4th Technology for Information Security Conference (TISC'96) (Houston, TX, May 1996)
- [3] CERT Coordination Center, Syslog vulnerability – A workaround for sendmail. Available by anonymous ftp from ftp.cert.org (October 1995).
- [4] Mansour Esmaili, Rei Safavi-Naini and Josef Pieprzyk, Computer intrusion detection: A comparative survey, Technical Report 95-07, *Center for Computer Security Research*, University of Wollongong, NSW 2522, Australia (May 1995).
- [5] Dan Farmer, Cops overview, Available from <http://www.trouble.org/cops/overview.html> (May 1993).
- [6] Dan Farmer and Wietse Venema, Improving the security of your site by breaking into it, available at <http://www.trouble.org/security/admin-guide-to-cracking.html>, 1993. (Internet white paper).
- [7] Daniel Farmer and Eugene Spaffors, The cops security checker system, Proc. Summer USENIX Conference (Anaheim, CA, June 1990) 165-170.
- [8] Stephanie Forrest, Steven A. Hofmeyr and Anil Somayaji, Computer immunology, *Communications of the ACM* 40(10) (October 1997) 88-96.
- [9] Jeremy Frank, Artificial intelligence and intrusion detection: Current and future directions, Proc. 17th National Computer Security Conference (Baltimore, MD, October 1994).

- [10] Stephen E. Hansen and E. Todd Atkins, Automated system monitoring and notification with swatch, *Proc. 7th Systems Administration Conference (LISA'93)* (Monterey, CA, November 1993).
- [11] Haystack Labs, Inc. Sytalker, Available from the company's website at <http://www.haystak.com/stalk.html> (1997).
- [12] L.Todd Heberlein, Gihan V.Dias, Karl N. Levitt, Biswanath Mukherjee, Jeff Wood and David Wolber, A network security monitor, *Proc. Symposium on Research in Security and Privacy*, Oakland CA, May 1990, 296-304, (*IEEE Computer Society Press, Los Alamitos, CA*).
- [13] Internet Security Systems, Inc. RealSecure, Internet <http://www.iss.net/prod/rsds.html> (1997).
- [14] R. Jagannathan, Teresa Lunt, Debra Anderson, Chris Dodd, Fred Gilham, Caveh Jalali, Hal Javitz, Peter Neumann, Ann Tamaru and Alfonso Valdes, System design document: Next-generation intrusion detection expert system (NIDES), Technical Report A007/A008/A009/A011/A012/A014, SRI International, 333 Ravenswood Avenue, Menlo Park, CA 94025 (March 1993).
- [15] Harold Javitz and Alfonso Valdes, The SRI IDES statistical anomaly detector, *Proc. IEEE Symposium on Research in Security and Privacy* (May 1991) 316-326.
- [16] Harold S.Javitz, Alfonso Valdez, Teresa F. Lunt, Ann Tamaru, Marby Tyson and John Lowrance, Next generation intrusion detection expert system (NIDES). 1. Statistical algorithms rationale. 2. Rationale for proposed resolver, Technical Report A016 Rationales, SRI International, 333 Ravenwood Avenue, Menlo Park, CA (March 1993).
- [17] Y. Franck Jou, Fengmin Gong, Chandru Sargor, Shyhtsun Felix Wu and W. Rance Cleaveland, Architecture design of a scalable intrusion detection system for the emerging network infrastructure, *Technical Report CDRL A005, MCNC Information Technologies Division, Research Triangle Park, N.C.27709* (April 1997).
- [18] Gene H. Kim and Eugene H. Spafford, The design and implementation of tripwire: A file system integrity checker, in: Jacques Stern, ed., *2nd ACM Conference on Computer and Communications Security* (ACM Press, COAST, Purdue, November 1994) 18-29.
- [19] GunarLiepins and H.S. Vaccaro, Anomaly detection: Purpose and framework, *Proc. 12th National Computer Security Conference (October 1989)* 495-504.
- [20] Teresa F. Lunt, Automated audit trail analysis and intrusion detection: A survey, *Proc. 11th National Computer Security Conference* (Baltimore, MD, October 1988).
- [21] Teresa F. Lunt, A survey of intrusion detection techniques, *Computers & Security*, **12**(4) (June 1993) 405-418.
- [22] Noelle McAuliffe, Dawn Wolcott, Lorryne Schafer, Nancy Kelem, Brian Hubbard and Theresa Haley, Is your computer being misused? A survey of current intrusion detection system technology, *Proc. 6th Annual Computer Security Applications Conference* (Tuscon, AZ, December 1990) 260-72.
- [23] Network Associates Inc., Cybercop scanner, Available from the company's website at <http://www.nai.com/products/security/ballista/default.asp> (1998).
- [24] Network Associates Inc., Cybercop server, Available from the company's website at <http://www.nai.com/products/secueity/cybercopsvr/index.asp> (1998).
- [25] Phiilip A. Porras and Alfonso Valdes, Live traffic analysis of tcp/ip gateways, *Proc. ISOC Symposium on Network and Distributed System Security (NDSS'98)* (San Diego, CA, March 1998) (Internet Society).
- [26] David R. Safford, Douglas Lee Schales and David K. Hess, The tamu security package: An ongoing response to internet intruders in an academic environment, *Proc. 4th USENIX Security Symposium* (Santa Clara, CA, October 1993) 91-118.
- [27] Secure Networks, Inc. Ballista security auditing system, Available from the company's website at <http://www.securenetworks.com/ballista/ballista.html> (1997).

- [28] Stephen Smaha, Haystalk: An intrusion detection system, 4th Aerospace Computer Security Applications Conference (October 1988) 37-44.
- [29] Steven R. Snapp, James Brentano, Gihan V. Dias, Terrance L. Goan, L. Todd Heberlein, Che lin Ho, Karl N. Levitt, Biswanath Mukherjee, Stephen E. Smaha, Tim Grance, Daniel M. Teal and Doug Mansur, DIDS (distributed intrusion detection system) – motivation, architecture and an early prototype, Proc. 14th National Computer Security Conference (Washington, DC, October 1991) 167-176.
- [30] Paul Spirakis, Sokratis Katsikas, Dimitris Gritzalis, Francois Allegre, John Darzentas, Claude Gigante, Dimitris Karagiannis, P. Kess, heiki Putkonen and Thomas Spyrou, SECURENET: A network-oriented intelligent intrusion prevention and detection system, *Network Security Journal* 1(1) (1994).
- [31] Stuart Staniford-Chen, Brian Tung, Phil Porras, Cliff Kahn, Dan Schnackenberg, Rich Feiertag and Maureen Stillman, The common intrusion detection framework-data formats, Internet drade draft-ietf-cidf-data-formats-00.txt (March 1998).
- [32] David Vincenzetti and Massimo Cotozzi, Atp-anti tampering program, Proc. 4th USENIX Security Symposium (Santa Clara, CA, October 1993) 79-9.
- [33] WheelGroup Corporation, Brochure of the netranger intrusion detection system, available from the company's website at http://www.wheelgroup.com/netranger/netranger_broch.html.