A Survey of Intrusion Detection Systems

Silviu Craciunas

Department of Computer Sciences
University of Salzburg, Austria
scraciunas@cs.uni-salzburg.at

Abstract

This paper is a survey of the research in the field of intrusion detection systems. Some previous surveys in the field are [2, 24, 3, 20, 16, 5]. An extensive literature exists on the topic of intrusion detection systems that use audit data for uncovering anomalous system behavior. It is possible to identify themes that are common to many of the available techniques. The intention of the paper is to provide a survey of the literature available on this topic and more specifically on the research context of [25].

1 Introduction

Intrusion detection systems are the intrusion alarms in the computer security field. The goal is to defend the system by using an alarm that goes off whenever the system has been compromised. The intrusion can be one of a number of different types. For example, someone might steal a password and hack inside a computer for malicious purposes or users may abuse their existing privileges to gain access to other accounts or even use a program to exploit vulnerabilities in the software of the system. The detection of such intruders is an important problem for the field of system security.

In the early years of this area two major principles were set up which are still in use today regarding the detection approach : signature-based detection and anomaly-based detection. Signature-based detection uses a series of signatures to study the normal and abnormal behavior for an entity and then conclude from these signatures if an attack occurred. Anomaly-based systems define patterns of intrusions and then classify such behavior as malicious that resemble those predefined or learned patterns. The advantages of the first approach are that the rates of false positives is little and the rate of processing of the audit data is high. The problems with this approach is that it relies on a well-known database of signatures, and it is not able to detect intrusions that have not yet been made known to the intrusion detection system. Anomaly based detection offers the benefit that the user must not configure it prior to usage because it trains itself, it then also runs unattended and is able to detect even unknown attacks. The issues with the second approach rest in the fact that while it detects unknown attacks, the false alarm rates can be high.

Almost all intrusion detection systems consist of a subsystem for collecting data about the observed system also called the audit data collection agent. Then another subsystem that detects if a behavior is suspicious, the output of which is presented to the system administrator, who then can take further action, normally beginning with further investigation into the causes of the alarm.

2 Overview

One of the first works in the field was done by Anderson [1]. Anderson used audit data collected for other purposes such as system performance and analyzed

this data so that a division can be made into possible threat categories. In this paper the authors also divide attacks into four possible categories which are later used to develop intrusion models. Based on this work the first model for intrusion detection system was build by Denning [8]. Denning proposed a model of a real-time intrusion-detection system that detects several forms of computer abuse. Denning based his approach on the idea that security violations can be detected by monitoring a system's audit records for abnormal patterns of system usage. The model is a general-purpose intrusion-detection system that is independent of system or application environment. Because of this the model was often used as a base for other systems.

Two of the first projects in the area were led by Lunt at Sytek and Javitz at SRI [32] that have shown that users can be distinguished from one another through patterns of their computer usage. Thus the idea of using a normality profile gained from the normal behavior of a user to detect anomalous behavior was formed. The main idea was to create a probability function of the current usage based on previous usages and if that probability was little then this could indicate an intrusion.

Most intrusion detection system focus on network based detection. Important work are [6, 19] where the systems use statistical anomaly detection to find Remote-To-Local attacks at network services. Kruegel also proposes specific IDS for web-based attacks using the queries send by client programs to the servers and their parameters thus leading to a higher number of identified attacks. [18].

Another very influential paper is [26] where Bro, a stand-alone system for detecting network intruders in real-time is presented. Beside the normal detection mechanisms it can detect attacks on the monitor itself in terms of overload attacks, which try to stress the monitor to the point of not being able to cope with the input data, crash attacks where the monitor is forced into failure, and subterfuge attacks, where the attacker tries to fool the monitor into misinterpreting analyzed data.

Other papers in the area of network intrusion detection systems like [27] focus on precise specifications of network packet sequences to define normal and abnormal behavior. he authors of this paper define a robust specification language that enforces a strict type discipline via a combination of static and dynamic type checking. This technique is interesting because of the time specifications given. The matching time is insensitive to the number of known patterns of intrusion and the aggregation operations take constant time per packet. This can be thus viewed as a real-time detection system.

In recent years the topic of wireless networks has also been addressed. For example in [35] reasons are given to why traditional fixed network models do not work in a wireless environment and a solution is presented for a better architecture of intrusion detection systems that is distributed and cooperative. Thus, the trace analysis and detection phase are done locally on all nodes and whenever possible in cooperation with other nodes.

The direction of the network intrusion detection systems is towards distributed systems as in large networks it becomes increasingly complicated to analyze traffic. A major research problem is also the efficient detection of coordinated attacks over large networks. An example is [34] which proposes the architecture of a Coordinated Attack Response that focuses on these issues.

Another branch of research has focused on program-based detection rather than on the traditional network and host levels. That is, intrusion monitors will typically analyze network packet logs or host machine audit logs for signs of intrusion activity for specified programs. Ghosh, Schwartzbard, and Schat propose a method for program-based intrusion detection that is aimed at detecting novel attacks against systems [12].

3 Methods

For the implementation of an intrusion detection system a number of methods have been used. Most of them are statistical-based, rule-based, or model-based but more and more research is shifting towards neural networks, traps and keystroke analysis etc.

Recent research focuses on other methods for de-

tecting intrusions like Dasgupta and Gonzalez's use of a genetic classifier-based intrusion detection system, which can provide active detection and automated responses during intrusions. [7].

Gosh et al. use neural networks for detection of the misuse of programs where the results show that training neural networks through random data can be effective in detection. [11]

3.1 Statistical

This method was originally proposed by Denning [8] and extended by Lunt [23] to create usage profiles that consist of statistical data such as means, covariances and standard deviations and compare them against current usage. The statistical method was applied in the Haystack project [28]. Statistical methods can be used without profiles by describing patterns of misuse by threshold values but are largely dependent on prior knowledge about the distribution of the input data. This problem is discussed by Lankewitcz and Benard who propose non-parametric statistical methods [21]. The basic idea is that the statistical methods are used for generating rules for the normal behavior and then comparing these rules to real usage patters that might contain attacks.

3.2 Rule-based

Rule-based systems are basically if-then statements that are based on a set of rules and a control mechanism to apply those rules. This mechanism requires very large databases of rules and hence the rule-generating mechanism has to be automated as proposed by Liepins and Vaccaro [22]. Ilgun, Kemmerer and Porras [14] propose a real-time detection model that uses state diagrams. The state transition approach models penetrations of a system as a series of state transitions described as signature actions and state assertions. Although the system aims for detecting anomalies that are also detectable with known rule-based models, that use pattern matching over a sequence of audit data, the current system focuses on the effect of each individual step which in term leads to greater flexibility when detecting variations in attacks. So this could be described as the first work that uses rule-based methods to detect unknown attacks even if in just a primitive way and only for variations of existing known attacks.

3.3 Model-based

Model-based detection system were proposed by Garvey and Lunt [10, 23] and are based on modeling the activities as states connected by arcs. Using this method analysis is more efficient than in the previous techniques. Apart from that models can be verified using different variations in input data. On the down-side model based methods produce a more false negatives than signature based but can detect also unknown attacks.

4 System Calls

The focus of the research in the last years has shifted to detecting intrusions through system calls as they are the key element with which a program communicates with the underlying operating system.

The sequence of system calls produced by applications has also been the object of anomaly detection analysis. The techniques proposed so far fall into the areas of specification-based and learning-based approaches. Learning-based techniques do not rely on any a priori assumptions about the applications. Instead, profiles are built by analyzing system call invocations during normal execution.

One of the first and most influential paper in this category is [9] where the aspect is towards simplicity and practicality. The paper presents for the first time the idea of using sequences of system calls for defining the normal behavior pattern. In this way the definition of the database is compact because the sequences of system calls that are anomalous are clearly distinguishable. Furthermore it is computationally efficient and can provide the basis for an online computer immune system.

Paper [4] presents a detailed analysis of the UNIX system calls and classifies them according to their level of threat. A mechanism is then proposed to control the invocation of critical system calls. During the training phase, the system collects all distinct system

call sequences of a certain specified length. During detection, all actual system call sequences are compared to the set of legitimate ones, raising an alarm if no match is found.

This approach has been further refined in [33], where the authors study similar models and compare their effectiveness to the original technique. However, these models do not take into account system call arguments. In [17] the authors propose analyzing a bag of system calls and using learning based algorithms. *Bag of system calls* is inspired by the *bag of words* representation that has been demonstrated to be effective in text classification problems. Here, a sequence is given in the form of an ordered list with no specification of the relative order of system calls. The paper shows that the machine learning techniques on simple bag of system calls representations of system call sequences is effective and often performs better than use only subsequences for detecting intrusive behaviors of compromised processes.

The paper [29] focuses on the design of an intrusion detection system at the user level with system-call interposition. The reason behind it is that the overhead of the access at the user level is low. However, it can be seen that the access control mechanism at user level can be easily bypassed. Thus, an intrusion detection system at user level has been combined with system-call interposition. As a result, the IDS proposed in this paper can prevent attacker from bypassing the interposition mechanism.

Another method is introduced for detecting intrusions at the level of privileged processes [13]. The paper presents a method for anomaly intrusion detection at the process level which adds some simplicity to the classification model. Further it adds a novel idea by focusing the algorithms on certain privileged processes and optimizing it for false positives by looking at each process individually and then setting up according separate rules. The interesting part is that the authors use as a basis for their system the human immune system and how it works.

At implementation level [15] adds a new approach for system call extension infrastructure that doesn't require an in-kernel wrapping of the system calls but offers a user-level infrastructure for analyzing system calls. The novelty lies in providing the possibility to implement a supervisor interface that does not need to know details of the interception of system calls, architecture specific methods to access system call arguments or results or OS-specific ways to modify process data. Basically it provides a framework for improving the access control to UNIX operating systems for the purpose of building an intrusion detection system.

One of the first approaches to include system call arguments for detection and classification was published by Gaurav Tandon and Philip K. Chan [30] and later refined in [31]. These papers show that in traditional rule-based algorithm a key component is missing. By including the system call arguments in the detection and classification scheme one can reach a higher level of accuracy. Systems that only analyze system call sequences can be evaded by launching attacks that execute legitimate system call sequences. The evasion is possible because existing techniques do not take into account all available features of system calls. The main information source that is not being considered is system call arguments. In this paper the authors merge a classical rule based algorithm S-LERAD with their own argument based model A-LERAD and show that the efficiency of this approach is higher. A significant flaw of the paper is that it focuses on rule-based systems and moreover in a real life environment there is a price to pay for the increased accuracy, that of a high overhead.

In [25] the concept is taken to a new level by including system call arguments in the analysis of a system's behavior and implementing it with a learning based system, which compensates for the flaws in [30]. The novelty of the paper consists of using multiple detection models on the system calls and their arguments that allow a far greater accuracy of detection then single model methods. Second, the paper introduces a sophisticated method of combining the anomaly scores from each model into an overall aggregate score. In traditional model based techniques the aggregate score was calculated by using only a sum function over the models and comparing it to a threshold. In this way important information is lost.

The paper uses Bayesian networks to represent the models an their characteristics. Each model m_i, associated with a certain system call, assigns an anomaly score as_i to a single argument of an invocation of the system call. This anomaly score shows the probability of the occurrence of the given argument value with regards to an established profle. Based on these scores and additional information I the IDS determines if a anomalous behavior has occurred. Formally it can be expressed as :

$$C(as_1, as_2, \ldots, as_k, I) = \{normal, anomalous\}$$

A simple sum function can not deliver reliable results. Therefore the paper proposes a technique that uses Bayesian networks to perform system call classification. In this Bayesian network the root node is a variable with two states: normal and anomalous. One child node is introduced for each model (there might also be dependencies between models represented by connections). Additionally there is a confidence value represented by a node connected to the model node. The IDs takes its input from audit facilities(eg. Linux) or audit logs(eg. Solaris' BSM), monitors security-critical applications(eg. setuid)and for each program the IDS maintains data structure that characterizes the normal profile. A profile consists of a set of models for each argument and a functions that calculates the anomaly scores.

5 Trends

As noted in [2] there are some trends visible in the study of intrusion detection systems that can be recognized over the course of the years.

5.1 From host to network

The focus has shifted towards network intrusion detection systems rather than host based systems. The problem relies in the fact that with advancing and more faster network hardware it has become even more difficult to monitor the data in real time. Another issue that arises is the presence of encrypted data that cannot be verified. These difficulties may mean that another shift is yet possible back to host based intrusion detection systems.

5.2 From centralized to distributed

Because of the shift from host based to network based systems there is a necessity to focus on distributed systems rather than centralized ones. This becomes clear in the case of data collection. Because on a network of computers the host cannot collect data so it must act in a distributed way. The analysis of data is a more difficult challenge to handle in the context of distributed systems because any distributed system gains in complexity opposed to centralized software.

5.3 System calls

In the case of host-based systems the shift is more towards run-time interception of malicious attacks and therefore the focus is put on intercepting and analyzing system calls. The recent years have seen an increase in research on this subject.

6 Open issues

Several open issues remain that need to be discussed in detail.

- to what degree can a system be trusted to correctly identify attacks, i.e. how can we diminish the number of false positives
- what is the best audit data that we can analyze in order to have a high degree of efficiency
- How can we handle unknown intrusions with as few as possible(or none) slipping through the IDS
- can we continue execution after an attack occurred?
- how can we minimize the overhead of the IDS in order for it to be efficient and maybe real-time

7 Conclusion

In the area of intrusion detection systems researchers remain interested in the use of various learning and signature based algorithms to detect anomalous system behavior and on refining models for better accuracy. This paper provides a summary of the techniques that have appeared in the journals and conferences dealing with system security. It is the hope of the author that the paper will provide a good overview of the research direction in the field of intrusion detection systems.

References

[1] ANDERSON, J. P. Computer security threat monitoring and surveillance. Tech. rep., Fort Washington, PA, 1980.

[2] AXELSSON, S. Research in intrusion-detection systems: A survey. Tech. Rep. 98–17, Department of Computer Engineering, Chalmers University of Technology, Goteborg, Sweden, Dec. 1998.

[3] AXELSSON, S. Intrusion detection systems: A survey and taxonomy. Tech. Rep. 99-15, Chalmers Univ., mar 2000.

[4] BERNASCHI, M., GABRIELLI, E., AND MANCINI, L. V. Remus: a security-enhanced operating system. *ACM Trans. Inf. Syst. Secur. 5*, 1 (2002), 36–61.

[5] BLOMQVIST, D., AND SKANTZE, J. Intrusion detection : a study. Master's thesis, Uppsala University, 1995.

[6] BYKOVA, M., OSTERMANN, S., AND TJADEN, B. Detecting network intrusions via a statistical analysis of network packet characteristics, 2001.

[7] DASGUPTA, D., AND GONZALEZ, F. A. An intelligent decision support system for intrusion detection and response. In *Proc. of International Workshop on Mathematical Methods, Models and Architectures for Computer Networks Security (MMM-ACNS)* (St.Petersburg, May 21-23, 2001), Springer-Verlag.

[8] DENNING, D. E. An intrusion-detection model. *sp 00* (1986), 118.

[9] FORREST, S., HOFMEYR, S. A., SOMAYAJI, A., AND LONGSTAFF, T. A. A sense of self for Unix processes. In *Proceedinges of the 1996 IEEE Symposium on Research in Security and Privacy* (1996), IEEE Computer Society Press, pp. 120–128.

[10] GARVEY, T. D., AND LUNT, T. F. Model-based intrusion detection. In *In Proceedings of the 14th National Computer Security Conference* (1991).

[11] GHOSH, A., WANKEN, J., AND CHARRON, F. Detecting anomalous and unknown intrusions against programs. In *Proceedings of the 1998 Annual Computer Security Applications Conference (ACSAC'98), December 1998.* (1998), Los Alamitos, CA, USA : IEEE Comput. Soc, 1998, pp. 259–267.

[12] GHOSH, A. K., SCHWARTZBARD, A., AND SCHATZ, M. Learning program behavior profiles for intrusion detection. In *Proceedings 1st USENIX Workshop on Intrusion Detection and Network Monitoring* (Apr. 1999), pp. 51–62.

[13] HOFMEYR, S. A., FORREST, S., AND SOMAYAJI, A. Intrusion detection using sequences of system calls. *Journal of Computer Security 6*, 3 (1998), 151–180.

[14] ILGUN, K., KEMMERER, R. A., AND PORRAS, P. A. State transition analysis: A rule-based intrusion detection approach. *Software Engineering 21*, 3 (1995), 181–199.

[15] JAIN, K., AND SEKAR, R. User-level infrastructure for system call interposition: A platform for intrusion detection and confinement. pp. 19–34.

[16] JONES, A. K., AND SIELKEN, R. S. Computer system intrusion detection: A survey. Tech. rep., University of Virginia Computer Science Department, 1999.

[17] KANG, D.-K., FULLER, D., AND HONAVAR, V. Learning classifiers for misuse detection using a bag of system calls representation. Tech. Rep. 00000359, Computer Science Department, Iowa State University, 2005.

[18] KRUEGEL, C., AND VIGNA, G. Anomaly detection of web-based attacks, 2003.

[19] KRGEL, C., TOTH, T., AND KIRDA, E. Service specific anomaly detection for network intrusion detection.

[20] KVARNSTROM, H. A survey of commercial tools for intrusion detection. Tech. Rep. 99-8, Department of Computer Engineering, Chalmers University of Technology, Gotenborg, Sweden, Oct. 1999.

[21] LANKEWICZ, L., AND BENARD, M. Real-time anomaly detection using a nonparametric pattern recognition approach. In *In Proceedings of the Seventh Computer Security Applications Conference* (San Antonio, TX, 1991).

[22] LIEPINS, G., AND VACCARO, H. S. Anomaly detection: Purpose and framework. In *In Proceedings of the 12th National Computer Security Conference* (1989).

[23] LUNT, T. F. Using statistics to track intruders. In *Proceedings of Joint Statistical Meetings of the American Statistical Associacion* (1990).

[24] M'E, L., AND MICHEL, C. Intrusion detection: A bibliography. Tech. Rep. SSIR-2001-01, Sup'elec, Rennes, France, September 2001.

[25] MUTZ, D., VALEUR, F., VIGNA, G., AND KRUEGEL, C. Anomalous system call detection. *ACM Trans. Inf. Syst. Secur. 9*, 1 (February 2006), 61–93.

[26] PAXSON, V. Bro: a system for detecting network intruders in real-time. *Computer Networks (Amsterdam, Netherlands: 1999) 31*, 23–24 (1999), 2435–2463.

[27] SEKAR, R., GUANG, Y., VERMA, S., AND SHANBHAG, T. A high-performance network intrusion detection system. In *ACM Conference on Computer and Communications Security* (1999), pp. 8–17.

[28] SMAHA, S. E. Haystack: An intrusion detection system. In *In Fourth Aerospace Computer Security Application Conference* (1988), pp. 37,44.

[29] TABATA, T., AND SAKURAI, K. Design of intrusion detection system at user level with system-call interposing.

[30] TANDON, G., AND CHAN, P. K. Learning useful system call attributes for anomaly detection. In *FLAIRS Conference* (2005), pp. 405–411.

[31] TANDON, G., AND CHAN, P. K. On the learning of system call attributes for host-based anomaly detection. *International Journal on Artificial Intelligence Tools 15*, 6 (2006), 875–892.

[32] TERESA F. LUNT, J. V. H., AND HALME, L. R. Automated analysis of computer system audit trails. In *Proceedings of the 9th DoE Computer Security Group Conference* (1986).

[33] WARRENDER, C., FORREST, S., AND PEARLMUTTER, B. A. Detecting intrusions using system calls: Alternative data models. In *IEEE Symposium on Security and Privacy* (1999), pp. 133–145.

[34] YANG, J., NING, P., WANG, X. S., AND JAJODIA, S. CARDS: A distributed system for detecting coordinated attacks. In *SEC* (2000), pp. 171–180.

[35] ZHANG, Y., AND LEE, W. Intrusion detection in wireless ad-hoc networks. In *ACM MobiCom'2000* (2000), pp. 275–283.