ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 7, Issue 7, July 2018

Study of IDS Using Host Based Data

Anand Kiran

Research Scholar, Department of Computer Science & Engineering, Himalayan University,

Itanagar, Arunanchal Pradesh, India

ABSTRACT :Intrusion Detection System (IDS) is meant to be a software application which monitors the network or system activities and finds if any malicious operations occur. Tremendous growth and usage of internet raises concerns about how to protect and communicate the digital information in a safe manner. Nowadays, hackers use different types of attacks for getting the valuable information. Many intrusion detection techniques, methods and algorithms help to detect these attacks. This main objective of this paper is to provide a complete study about the definition of intrusion detection, host-based IDS (HIDS), Audit Logs, Limitations and solutions of Audit Logs, System Call Sequences, Limitations and solutions of system call sequences .

KEYWORDS:IDS, ICMP, TCP, UDP, IPS, KDD, HIDS .

I. INTRODUCTION

One of the main challenges when dealing with the amount of information that can be extracted directly from network packets is to create a set of features that covers most of the information space. The network features are constructed around the main abstractions of the network security domain such as packet, connection, host, and network. The idea of constructing features that will cover a reasonable part of the information space is especially hard due to the diversity of data that passes through a network link and the lack of unanimously accepted network feature classification schema in the research community. Although theoretically it is possible to design a system that can extract and use a wide range of features, due to constraints such as, large computational time, diversity of protocols and applications that exist, and amount of memory that the NIDS needs, most of the implementations make tradeoffs concentrating only on a particular set of intrusions (e.g., R2U, U2R, and DoS in Horizontal, Vertical, and Block Port Scanning for TCP and UDP; Denial of Quality of Service; Worms DDoS; TCP-SYN Flooding Attacks; TCP-SYN Flood, UDP Flood, and ICMP Flood Attacks).

Furthermore, researchers have empirically demonstrated that false correlations between the features that may be extracted by an IDS/IPS can lead to purer results concerning the accuracy and performance of the detection system. For example, if only 17 carefully selected features are used among all 41features provided in the International Knowledge Discovery and Data Mining Competition (KDD-1999), the detection rate will not change, but the speed of the detection algorithm will improve by about 50%. The lack of a widely accepted network feature classification schema also adds to the general confusion regarding the set of features that should be used for the detection purposes. Moreover, different researchers use different names for the same subsets of features, while others use the same name for completely different types. There seems to be at least two main types of features that are widely accepted in the literature. The dependent evaluation criteria uses the performance of the detection engine as a primary factor to evaluate the suitability and effectiveness of a feature or a set of features. First, it is computationally intensive, and secondly the final result is biased by the detection algorithm which tends to reward those features that perform better with that particular detection engine or mining algorithm. Conversely, the independent evaluation criteria uses the intrinsic characteristics of the data alone to evaluate the selected features. By studying the actual value of the feature while in the intrusive and normal stages, the final outcome would not be biased by any detection algorithm and thus we consider this method to be more adequate for the feature selection purposes. Furthermore, we believe that defining an independent evaluation criteriametric for evaluating the network features will allow researchers to use a deterministic way to identify those

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 7, Issue 7, July 2018

features that are highly important to the detection process. Despite the need for such a method, to our knowledge, there isn't any work that uses independent evaluation criteria for features in intrusion detection.

II. HOST-BASED IDSs

The host-based approach is one of the most common approaches in intrusion detection. An IDS using host-based data is basically a software residing on the protected host, which constantly searches for abnormalities of the data. These types of IDSs mine different sources of data such as memory contents, process behavior, system calls, file system accesses, or log files. Thus, the detection coverage of a Host-based IDS (HIDS) is physically limited to the protected host. Audit logs and system calls are the two most common types of data streams used by HIDSs. An HIDS that uses the first type of data, listens and inspects the logs that the Operating System (OS) is producing. Conversely, an HIDS that uses the latter type of data directly monitors the set of system methods that the applications invoke to accomplish their tasks (e.g., open, read, wait, write, fork, kill, close).

III. AUDIT LOGS

The OS auditing feature has been primarily adopted as a means to provide feedback to the OS developers. Later on, this functionality also proved to be a good source of information for intrusion detection. The OS logs a considerable amount of data that is not necessarily useful for intrusion detection. Thus, the HIDSs that use audit logs filter that data and monitor only a small set of selected tasks that their abusive execution might compromise the system. For instance, for Unix OS some of those programs are admintool, eject, fdformat, mount, passwd, ps, su, traceroute, whodo, to name a few. Once the set of critical applications to trace is decided, all detection techniques such as anomaly-, specification-, and signature-based can be used to discriminate between normal and abusive executions. For example, in the case of anomaly detection, techniques that involved the use of Recurrent Neural Networks, Support Vector Machines, Finite State Machines proved to successfully discriminate between the normal and abnormal executions. In these cases, the HIDS learns the normal behavior of each monitored application and signals any kind of deviation from the created profile. The signature-based techniques are also applicable to audit log information. For example, the extracted event sequences can be compared against a database of signatures to successfully pinpoint the malicious events.

IV. LIMITATIONS AND SOLUTIONS OF AUDIT LOGS

Using audit logs for security related purposes is a challenging task due to incompatibility reasons (i.e., different OSs have different formats for the log files, and save different types of data), transforming the HIDS into a platform dependent one. Furthermore, since the log functionality was not primarily designed for reporting security related events, extracting the desired data out of a log file is a difficult and time consuming task. Furthermore, in most cases the extracted data proves to be insufficient and to have a poor quality. One solution to solve this problem is to have an auditing language that formalizes the description of a security related audit log. However, since the implementation of such a solution involves multiple HIDS and OS vendors it remains very challenging in practice.Another problem is that since the system logs contain valuable information, they are also targeted by the by intruders in their attempt to either erase an attack trace or find confidential data. To solve this problem the critical audit data can be replaced with encrypted pseudonyms that can be interpreted by only the audit analyzers. In this way, if an attacker successfully manages to access the log files he will not be able to extract valuable information. Another solution is to encrypt the whole audit log who propose an encryption mechanism for private data. The proposed technique also makes it possible to use different encryption keys for different entries of the file to offer double-protection. Even though both of these techniques prevent the attackers to understand the stored data, they require extra computational time each time an entry needs to be read or written into the log.

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 7, Issue 7, July 2018

V. SYSTEM CALL SEQUENCES

Each OS provides a set of unique basic operations that any application must invoke in order to accomplish its tasks. These atomic operations are refereed to as System Calls (SC). The SC facilitate the manipulation of all resources that the OS has access to, such as the file system, memory, external devices, to name a few. Thus, any application can be indirectly monitored by studying the set of SC that it uses and the time relations between consecutive SCs. The expectation is that once an application is compromised its behavior will change in such a way that an IDS will be able to detect it by only looking at SCs. One way to use the SC for anomaly detection is to create normal profiles for each individual application by monitoring short SC sequences that it uses. The created profiles proved to differ from application to application, and be consistent throughout different runs of the same application. Furthermore, a different approach is to study the arguments of the SC invocations. This approach completely ignores the relationship between consecutive SC, the normal profiles being built based on features extracted from the arguments of each individual SC. The arguments are pictured as touples of n values. Each touple can be studied versus four main characteristics such as string length, string character distribution, structural interface, and tokens that it contains. One property of the SC arguments is that most of the times they are composed of human readable characters and their length does not usually exceed 100 characters. Moreover, the characters in a string occur with different distributions that can be learned. The structural interface of an argument can also be learned since most of them represent system paths to files that the application uses. The specification-based approach can also be successfully used when using SCs. In this case a set of rules can be defined for each individual application with respect to the allowed actions that it can perform. In other words, the HIDS detects the attack each time when the application does not respect its specifications. The allowed profiles can be built for instance by defining rules for the client-server communication, file system access, registry access, to name a few. The SC approach has the advantage of making the implementation of HIDS independent of any application design and implementation that is being monitored.

VI. LIMITATIONS AND SOLUTIONS OF SYSTEM CALL SEQUENCES

One of the disadvantages of using SC is the OS dependency, and as a consequence, the same application running on two different OSs (e.g. the same Java program) will end up having two different profiles. Moreover, due to the diversity of applications, some of the HIDSs concentrate only upon those applications considered to be critical, or susceptible to attacks, and this automatically restricts the detection scope of the HIDS. Next, a HIDS can negatively influence the performance of the host machine through the constant set of computations it needs to execute. Finally, another disadvantage of this technique that in fact applies to all host-based IDSs, is that they cannot detect network related attacks, such as worm spread, distributed denial of service attacks (DDoS), and probing. Furthermore, to gain protection of all hosts in an enterprise network, an HIDS needs to be installed, and maintained on all the hosts of the network, which can prove to be a difficult and time consuming task.

REFERENCES

[1] DARPA, Darpa intrusion detection and evaluation dataset 1999, http://www.ll.mit.edu, Website, Last accessed February 2006.
[2] K. Das, The development of stealthy attacks to evaluate intrusion detection systems, Master's thesis, MIT Department of Electrical Engineering and Computer Science, June 2000.
[3] M. Dash, K. Choi, P. Scheuermann, and H. Liu, Feature selection for clustering - a filter solution, Data Mining, 2002. ICDM 2002.Proceedings. 2002 IEEE International Conference on (2002), 115{122.
[4] M. Dash and H. Liu, Feature selection for clustering, PADKK '00: Proceedings of the 4th Pacific-Asia Conference on Knowledge Discovery and Data Mining, Current Issues and New Applications (London, UK), Springer-Verlag, 2000, pp. 110{121.
[5] M. Dash, H. Liu, and J. Yao, Dimensionality reduction of unsupervised data, Tools with Artificial Intelligence, 1997. Proceedings., Ninth IEEE International Conference on, no. 3-8, November 1997, pp. 532{539.
[6] P.A. Devijver and J. Kittler, Pattern recognition a statistical approach, Prentice Hall International, 1982.
[7] J. Doak, An evaluation of feature selection methods and their application to computer security, Tech. Report CSE-92-18, University of California at Davis, 1992.

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 7, Issue 7, July 2018

[8] P. Dokas, L. Ertoz, V. Kumar, A. Lazarevic, J. Srivastava, and P. Tan, Data mining for network intrusion detection, Proceedings of NSF Workshop on Next Generation Data Mining (Baltimore, MD), November 2002, pp. 21{30.
[9] P. Domingos, Context-sensitive feature selection for lazy learners, (1997), 227{253.
[10] Nicholas Athanasiades, Randal Abler, John Levine, Henry Owen, and George Riley, Intrusion Detection Testing and Benchmarking Methodologies, Proceedings of the First IEEE International Workshop on Information Assurance (IWIA'03) 0-7695-1886-9/03
[11] Puketza, Nicholas J.; Chung, Mandy; Olsson, Ronald A and Mukherjee, Biswanath, "A Software Platform for Testing Intrusion Detection Systems", IEEE Software, September/October, 1997, 43-51.
[12] Puketza, Nicholas J.; Zhang, Kui; Chung, Mandy; Mukherjee, Biswanath and Olsson, Ronald A., "A methodology for Testing Intrusion Detection Systems" 17thNatrional Computer Security Conference: Baltimore, MD, October, 1994.