International Journal of Engineering, Management, Humanities and Social Sciences Paradigms (IJEMHS)
Volume 30, Issue 04, Quarter 04 (Oct-Nov-Dec 2018)
ISSN (Online): 2347-601X
www.ijemhs.com

Rule-based intrusion detection system

Pravat Kumar Rautray

Aryan Institute of Engineering & Technology, Bhubaneswar

Abstract

In this paper we describe the development and testing of an agent-based intrusion detection system for Linux platform. We take a dual-approach to intrusion detection: pre-emptory and reactionary. With the pre-emptory approach, a network-based agent is implemented to monitor all packets entering the network and detect a known attack-based on a pre-defined rule. The reactionary approach is realized through a separate host-based agent to routinely check specific log files in order to detect system anomalies caused by successful attacks. Once a possible intrusion attempt has been detected by either one of the agents, it attempts to block the attack, records the attack details in a system log file, E-mails the system administrator, displays a warning through a graphical warning window. The agents operate in the background of user applications and system software without any noticeable performance effect on them. © 2002 Published by Elsevier Science B.V.

Keywordr: Intrusion detection system; Linux platform; Graphical warning interface

I. Introduction

Intrusion detection is defined as "the problem of identifying individuals who are using a computer system without authorization ('hackers') and those who have legitimate access to system but are abusing their privileges ('insider threat')" [8]. The goal of an intrusion detection system (IDS) is to identify, preferably in real-time, unauthorized use, misuse and abuse of computer systems by both system insiders and external perpetrators, and to take appropriate actions. IDSs can be classified into two categories: rule-bared and adaptive [3]. Rule-based IDSs rely on libraries and databases of known attacks represented in attack signatures [5] or production rules [7]. Adaptive systems employ more advanced techniques, including artificial intelligence, to not only recognize known attack signatures but also learn new ones [6].

This paper discusses the implementation of a rule-based IDS using autonomous agents that can be used by a distributed architecture within a heterogeneous network. The agents are processes that run on each host, monitoring that host or the entire network for intrusive activity. The agents can be configured to satisfy different system administration needs.

Most papers on IDS cover theoretical and architectural issues, with little reference to application details. The prototype implementation described in this paper gives an insight into how agents for detecting specific attacks can be implemented. The emphasis of this study is more on practical implementation of known architectures, rather than proposing new ones.

The rest of the paper is organized as follows. Some of the well-known IDSs are discussed briefly in Section 2. The implementation of our Linux-based IDS, both the pre-emptory and the reactionary agents, is presented in Section 3. We provide our conclusion in Section 4.

2. Related work

Event monitoring enabling responses to anomalous live disturbances (EMERALD) is a rule-based system developed by SRI International [11]. EMERALD adapts a distributed architecture that requires no central controller or analyzer. It is designed to monitor large distributed networks with analysis and response units called service monitors. Layers of monitors are established for performing data reduction in large network-based systems. It is being designed as a multi-platform system.

Autonomous agents for intrusion detection (AAFID) is a distributed monitoring and IDS that uses small stand alone agents to perform monitoring functions in the hosts of a network [1]. The AAFID team improves the existing models by distributing the agents over any number of hosts in a particular network thus preventing a single point of failure

commonly found in other IDSs. All agents in one host will send reports to a single transceiver resident on each host. Data reduction can also be performed by the transceivers. One or more monitors are ready to receive the results from the transceivers. In such way, the monitors have access to network-wide data, therefore they are capable to detect intrusions that involve several hosts.

Architectures such as EMERALD/AAFID remove one of the major limitations in rule-based knowledge systems, i.e. to update the IDS for new signatures.

Intrusion detection and tracking system (IDTS) is a web-based IDS, based on multi-layered manager-agent model [4]. The IDTS proposes a system that supports load balancing and performance enhancement, fine-grained data reduction, encrypted communication and optimum size analysis. The GrIDS system [14] is another distributed approach that builds a graph-representation of activity in the network-based on information reported to the graph engine by data source modules deployed at the host. GrIDS system also has scalability and configurability features. Other frameworks such as Bro [10] have focused on real-time aspects of the IDS. The common intrusion detection framework (CIDF) [2] working group is developing standard protocols and application programming interface so that the IDS system becomes modular and the components become reusable/replaceable.

3. Agent development and experiments

This section describes the prototype of agents developed. The operating system used in conducting the implementation phase of this project is Red Hat Linux (kernel version 2.2.14). Both agents are implemented using the C programming language, as they require a lower level access to the operating system and some network interfaces. The GNOME/GTK + library is used to implement the graphical warning interface. Two agents have been implemented each targeting various attack signatures. With the network-based agent, we placed our focus on detecting the sendmail denial of service (DoS) attack and identifying the popular Nmap scans. With the host-based agent, our focus is to monitor the integrity of the system log files and perform anomaly detection for events such as multiple login, su failures and login from a banned or suspicious remote host. Agents described in this section follow a generic structure that can be fit into any of the distributed architectures described in Section 2. They consist of a data collection/capture module, analyzer and a reporting module.

Network-bared autonomour intrurion detection agent (NID)

Attacking rendmail

Under Unix system, daemons like the ftp server, telnet server, and many other services normally run from the parent inetd superserver. When a connection arrives on one of its sockets, it determines the type of service requested, and forks a new process to handle the request. However, the sendmail daemon normally runs as a stand-alone daemon, usually on port 25. Any connection request to this port is handled directly by the daemon. The main server process creates a new process to handle each request. This invocation of a new sendmail process requires some resources from the process table. However, the Linux kernel sets a limit on how many processes the process table can handle. Once the process table is full, system degradation becomes noticeable. Our experiments showed that the Linux process table could be forced into reaching its limit by invoking an excessive number of sendmail processes.

4. Conclusion

This paper has demonstrated the steps required to build rule-based autonomous agents for a distributed IDS. We have described implementation of a network-based and a host-based agent for Linux. These agents are designed to run on various hosts within a heterogeneous network. The sendmail DoS attack is just one of the many attacks that could possibly be launched toward Linux systems. The network-based agent we have developed can detect a stream of packets that arrive within a short period of time, which indicates the sendmail DoS attack signature. On the other hand, the host-based agent possesses the capacity to perform

audit data collection and analysis, detecting login anomalies and system log files integrity. The accuracy of the detection model depends on how much data is available and how busy the network is at the time of the attack. The agents are also configurable which provides more options to tailor for various administrative needs.

References

[1] J.S. Balasubramaniyan, J.O. Garcia-Fernandez, D. Isacoff, E. Spaf- ford, D. Zamboni, An architecture for intrusion detection using auto- nomous agents. Technical report, Purduc University, West Lafayette, Coast Laboratory, Indianapolis, June 1998.
[2] Common intrusion detection framework (CIDF). http://seclab.cs. ucdavis.edu/cidf/, Mar. 1998.
[3] H. Debar, M. Dacier, A. Wespi, Towards a taxonomy of intrusion- detection systems, Computer Networks 31 (1999) 805-822.
[4] T. Hwang, J.W. Hong, A fault-tolerant and scalable intrusion detec- tion and tracking system for enterprise network. Proceeding of Asia- Pacific Network Operations and Management Symposium, Kyongju, Korea, Sept. 1999.
[5] T. Lane, C. Brodley, An application of machine learning to anomaly detection. 20th National Informations Systems Security Conference, MD, USA, Oct. 1997, pp. 366-377.
[6] T. Lane, C. Brodley, Temporal sequence learning and data reduction for anomaly detection, ACM Transactions on Information and SystemSecurity 2 (3) (1999) 295-331.
[7] U. Lindqvist, P. Ponnas, Detecting computer and network misuse withproduction-based expert system tool set. IEEE symposium on securityand privacy, Oakland, Canada, May 1997.
[8] B. Mukherjee, T.L. Heberlein, K.N. Levitt, Network intrusion detec-tion, IEEE Network 8 (3) (1994) 26-41.
[9] Nmap services fingerprinting program. http://www.insecure.org/nmap/.
[10] V. Paxson, Bro: a system for detecting network intruders in real-time. USENIX Security Symposium, Jan. 1998.
[11] P.P. Porras, P.G. Neumann, Emerald: event monitoring enabling responses to anomalous live disturbances. Proceedings of the 20th National Information Systems Security Conference, National Institute of Standards and Technology, 1997, pp. 353-363.
[12] J. Postel. Transmission control protocol. Internet Request for Comment RFC793, IETF, Jan. 1981.
[13] Sweeper and cleaner programs. http:/www.dsinet.org/tools/logutils/.
[14] S. Staniford-Chen, S. Cheung, R. Crawford, M. Dilger, GrIDS: a graph-based intrusion detection system. Proceedings of the 19th National Information Systems Security Conference, National Institute of Standards and Technology, 1996, pp. 361-370.
[15] W.R. Stevens, UNIX Network Programming, Prentice-Hall, New York, 1990.