2009 International Symposium on Computing, Communication, and Control (ISCCC 2009)
Proc .of CSIT vol.1 (2011) © (2011) IACSIT Press, Singapore

Distributed Intrusion Detection System using Mobile Agents

Dr. Bhushan Trivedi [1], Jayant Rajput [2], Chintan Dwivedi [3] and Pinky Jobanputra [4+]

[1 2 3]MCA Dept.,GLS ICT, Ahmedabad, India

[4] CICA, Education Campus, Changa, India

Abstract. The increasing number of network security related incidents makes it necessary for organizations to actively protect their sensitive data with the installation of intrusion detection systems (IDS). Autonomous software agents, especially when equipped with mobility, promise an interesting design approach for such applications. We evaluate the implications of applying mobile agent technology to the field of intrusion detection and present a distributed intrusion detection system (IDS) based on mobile agents that considers large-scale network environment in order to monitor multiple hosts connected via a network as well as the network itself. The design and implementation of our Distributed Intrusion Detection prototype relies on Security Agents which monitor network traffic and report intrusion alerts to a central management node. Our model comprises four major components: the IDS monitor, the Agent server which distributes intelligent mobile agents called mobile IDS agents, Authentication and Utility tool. Finally, we present discussion of design and implementation issues, and directions for future work

Keywords: Intrusion, Mobile agents

1. Introduction

There are two ways to protect our network against malicious attempts. First is to build complete secure network system by applying all complicated cryptographic, authentication and authorization methods. However, this solution is not realistic. In practice, it is impossible to have completely secure system, because the user uses operating system and other applications to accomplish his/her job. Almost all applications have one or the other vulnerabilities. Second way is to detect an attack as soon as possible preferably in real-time and take appropriate action. This is essentially what an Intrusion Detection and Preventation System (IDS and IPS) does. An IDS does not usually take preventive measures when an attack is detected; it is a reactive rather than pro-active. However, IDS which uses mobile agents are new class of intrusion detection system. Mobile agents can be defined as "self-contained and identifiable computer autonomous programs, bundled with their code, data, and execution state that can move within a heterogeneous network of computer systems. They can suspend their execution on an arbitrary point and transport themselves into another computer system."[1]. Mobile agents have special characteristics which can help intrusion detection in several ways. The use of mobile code and mobile agents computing paradigms have been proposed in several researches [2, 3, 4]. The advantages include: overcoming network latency, reducing network load, executing asynchronously and autonomously, adapting dynamically, operating in heterogeneous environments, and having robust and fault-tolerant behavior. Moreover, implementation of mobile agents in languages such as JAVA provides mobile agents with system and platform independence and considerable security features [6].

The presented system in this paper addresses many problems plaguing current IDSs. First, the approach provides highly-distributed IDS with a minimum amount of traffic generated over the network. There are mobile processing units to capture and analyze relevant data asynchronously and independently from the

[+] Corresponding author.
E-mail address: (pinkyjobanputra.mca@ecchanga.ac.in please specify).

main machine. Second, mobility makes the IDS highly secure against attacks targeting the IDS itself. Third, roaming the internal network, agents are capable of detecting attacks from within the network. Fourth, agents may be programmed to have sufficient decision making intelligence. They decide how to respond to the detected attacks. Fifth, the set of signatures that the agent uses in the detection process is dynamic, meaning that it evolves in real-time depending on the feedback from the main machine. Sixth, the system is shown to be highly adaptive since population of IDS agents' increases during attack and decreases during peaceful states.

In the next section, we present a literature review of previous work in the domain of mobile agent-based intrusion detection systems. In section 3, we describe the prototype of IDS. Section 4 provides a discussion of design and implementation issues. Finally, section 5 presents a conclusion and directions for future work

2. Related Work

Historically, the intrusion detection technology dates back to 1980 [7]. It became a well-established research area following the introduction of the model in [8] and the prototypes presented in [9] and [10]. These systems were centralized. A centralized server running on a specific individual machine monitors data flow at a strategic point in the network and collects and analyzes data from the log files. Once an attacker deactivates this host, he or she is able to gain considerable access to the whole network. This limitation, we believe, is the main vulnerability of currently implemented centralized IDSs.

Distributed IDSs were introduced to overcome this susceptibility. The approach in [11] proposes architecture for a distributed intrusion detection system based on multiple independent entities called Autonomous Agent for Intrusion Detection (AAFID) framework. In a similar approach, a mobile agent-based architecture and model consists of a large number of small mobile agents that perform the tasks of monitoring, decision-making, notification and reaction to attempted intrusions [12]. New specialized agents can be added whenever a new form of attack is identified or removed dynamically from the system. Subsequent work like that portrayed in [13], [14], or [15] presents a fully distributed architecture where data collection and information analysis are performed locally without referring to the central management unit. For instance, the designed architecture in [15] comprises two components: IDS agents and a stationary secure database (SSD). IDS agents are stationary and participate in cooperative algorithms to decide if the network is being attacked. The SSD acts as a trusted database for the agents to obtain latest misuse signatures.

Another architecture for an entirely distributed IDS based on multiple independent entities working collectively is discussed in [5]. These entities are called Autonomous Agents. The approach was claimed to solve some of the problems in existing commercial IDSs associated with centralization, configurability, and scalability. Computation is performed (and thus intrusion detection) at any point where sufficient information is available. These systems use network resources inefficiently.

3. System Architecture

This section presents the architecture of our distributed IDS. We detail the inner components, and then illustrate the role of each with an example. The architecture is comprised of the following components: (1) a main IDS monitor, (2) Agent Server (3) Authentication, and (4) Utility Tool. A high level view of the architecture is given in Fig 1.

3.1. IDS Monitor

This component is the cornerstone of our distributed framework. It is responsible for monitoring network segments (subnets), and acts as central intrusion detection and processing unit. Its main capabilities are: acting as a cross-relating unit for multiple logs sent by dispatched agents, providing and updating rule sets for each of the agents, and interfacing the IDS to the system administrator. Once logs are collected, the raw data is linked to structures for analysis by the detection engine. The detection engine processes the captured packets by checking them (the header and/or the content of the packet, depending on the security level) against a set of rules. If the rules match the data in the packets, then alerts are triggered and written into the output alert files and responses are sent to both the user interface and the dispatched agents.

A major function of the IDS monitor is the collection and correlation of IDS data from distributed IDS mobile agents. The main objective here does not lie in identifying isolated intrusions; rather, it is in the linkage of events across a network, providing organizations with a heuristic analysis of combined data or even a static-state assessment of correlated intrusions from separate IDS mobile agents. This will enable us to figure out about attacks which are not possible to be judged by single IDS component running on a specific machine.

The IDS monitor acts as a secure, trusted repository for the mobile agents to obtain latest information about attacks that they should look for and to update their severity lists. Attached to the IDS monitor is a database that contains information like attack traces or signatures (rule set).

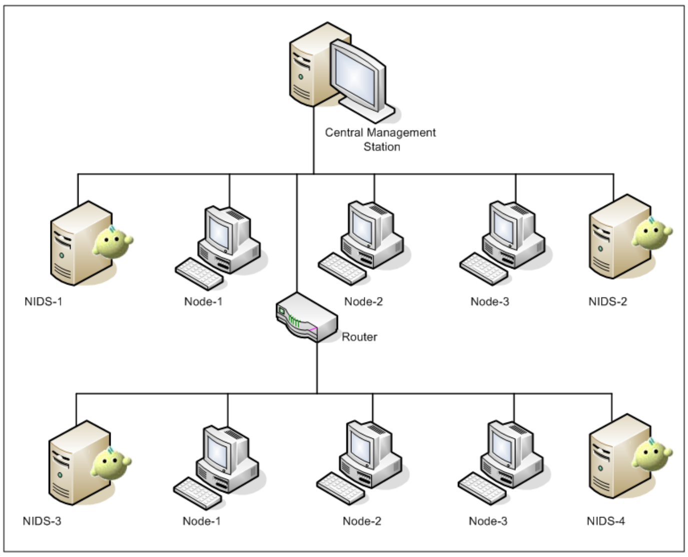

Fig. 1: System Architecture

3.2. Agent Server

This component is responsible for allowing the network administrator to monitor and control the mobility of the mobile agents. It provides a framework for execution to the mobile agent. It will help to instantiate the mobile agent and move from one machine to another machine in the network. It will also help to log the information about mobile agent arrived and created. It will display the summary of the mobile agent instantiated or arrived in the system. It will also allow the administrator to control the life of the mobile agent by activating, deactivating, cloning, disposing it.

Mobile Agent: Each subnet in the network will have a mobile IDS agent roaming among all its hosts at all times. This agent is responsible for detecting intrusions based on data gathered by sniffing on the network traffic. In general, sniffing is used for: (1) network analysis and troubleshooting, (2) performance analysis and benchmarking or, (3) eavesdropping for clear-text passwords and other interesting tactics of data. Each IDS agent is “armed” with a light detection engine that enables it to detect most well-known attacks. Once a host receives a mobile IDS agent, the latter’s first job is to create a thread and start sniffing and dumping into a log file. The log file is created in a share mode so that detection can proceed in parallel with sniffing to avoid detection latency. The agent starts the detection process on a separate thread so that sniffing and detection engine use the same file, one to read from while the other to dump into. Once an intrusion is caught, the agent checks it and responds accordingly. The agent moves to its next hop if no severe attacks take place at the current host during a certain time interval, and the cycle starts over.

How does it work? First, let us describe the initial state of the system as it is shown in figure 1. On Central Management System (CMS) Agent server is installed. The Agent Server that resides on this device has the necessary information about the directories at which log and alert files should be saved.

Scenario 1: This is the idle state of the system where no intrusions are detected. When the system is initially started, the IDS monitor sends a Startup request to the Agent Server. The message specifies the number of agents to be launched and the corresponding IP address sets that each agent is expected to visit. This implies that the IDS monitor has a registry containing all IP addresses in the local network. The Agent server, in turn, creates the agents and dispatches them into the network. Once an agent is received by the first host on its list, it starts sniffing and detecting processes. After a while, the agent informs the Agent Server on the host where it is running about its willingness to move to another host. This Agent Server copies the agent's code to the destination host via the Agent Server residing there (which will take care of running it on the new host) and then deletes the code after shutting down its process. This, in effect, moves the agent to the destination host. In this scenario, the agent population remains constant.

Scenario 2: Now, assume that an agent on its trip catches an attack that triggers an alarm. The agent checks the attack and responds accordingly. The agent sends the logged data as well as the alert file to the main machine and creates a clone for itself. The agent assigns half its visiting list to its clone and keeps the second half for it. Thus the agent population starts to increase when continuous attacks are launched against the network. The agent clones roam the assigned segments and may clone themselves when they detect new attacks till we reach the all-Snort state. The all-Snort state is reached when every host has a mobile IDS agent running all the time (equivalent to running snort on every host).

After a while where no attacks are detected, the clones start to dispose. Every clone should send a message to the main machine to request disposal. The main machine determines the parent agent of the clone and sends it a "Visiting_list_update" message to handover the child segment. After the parent sends an acknowledgment to the main machine that it's visiting list is successfully updated, the main machine sends the child agent an acknowledgment message for disposal.

3.3. Authentication

For authenticating user and the owner of the mobile agent digital signature is used.

3.4. Utility Tool

This component will provide a text editor for the user(s) of system to carry out their daily text operations like viewing XML/text files, writing rules etc.

4. Implementation

In this section, the implementation of the prototype distributed IDS is discussed.

4.1. Implementation of IDS

The prototype IDS has been implemented on top of Snort [16] and the well known mobile agent system, Aglets. Each agent carries with it a lightweight snort engine [17] that detects local intrusions in "semi-real time", while a full fledged snort engine is installed at the main station that performs in depth analysis of log files and controls the behaviors of the agents plus their routes accordingly.

4.2. Aglets

Mobile Agent based system is implemented using Aglets. Aglets was chosen as the mobile agent platform because of its availability, ease of running, reliable messaging, dynamic routing on agent itinerary, and support for mobile agents and being open source.

4.3. Light Weight Snort

Light-weight snort is an open source IDS component [18] runs against a limited rule set. It was selected in the prototype because of its lightweight, popularity, portability, support of multiple operating systems, configurability, and the availability of multiple output options (alerting to syslog, file, or win popup).

4.4. Data Storage: XML

In this system, the information about the IDS node and snort rule stored using XML.

4.5. Discussion and Results

Intrusion Detection system is implemented as per the architecture shown in Figure 1. The system is configured as follows: The Central Management System is set as the IDS monitor is running in addition to the mobile agent platform. On specific PC, snort is installed in each subnet, which is working as NIDS for that subnet. When IDS monitoring starts up will allow the administrator to dispatch an agent on all the IDS node machine. It gets information about the IDS machine from the xml file and displays the information of all the activity performed. IDS monitor provides the facility to view filter list on the basis of criteria defined by the administrator for each node in the network.

5. Conclusion and Further Work

The paper aim is in building up robust distributed IDS which covers the flaws of the other models while uses their useful features. In conclusion, The Mobile Agent based IDS could even detects attack against IDS control centers while agents roam through the network to spot intrusions. Nonetheless, weaknesses are unavoidable in a new design, and many areas discussed in this paper would benefit from further efforts to clarify the design fine points and implementation details. Further work should also look into mobile agent's intercommunication and negotiation which can help investigative mobile agents to share their knowledge. In addition intrusion pattern's knowledge sharing between IDS control centers can be considered for further studies

6. References

[1] Peter Braun, Wilhelm R. Rossak, Mobile Agents: Basic Concepts, Mobility Models, and the Tracy Toolkit, published by Morgan Kaufmann (December 22, 2004), ISBN-10: 1558608176.

[2] Andreas Fuchsberger, "Intrusion Detection Systems and Intrusion Prevention Systems"; Information Security Group, Royal Holloway, University of London, Egham, Surrey TW20 0EX, United Kingdom 2005.

[3] J. P. Anderson. Computer Security Threat Monitoring and Surveillance. Technical Report, James P. Anderson Co., April 1980.

[4] Richard A. Kemmerer and Giovanni Vigna, Intrusion detection: a brief history and overview Reliable Software Group, Computer Science Department, University of California Santa Barbara 2003.

[5] J.Balasubramainyan, J.O. Garcia-Fernandez, D.Isacoff, E.H. Spafford, D.Zamboni, An architecture of intrusion detection using autonomous agents, Department of Computer Science, Purdue University coast TR 98-05, 1998.

[6] S. Fuenfrocken. Integrating Java-based Mobile Agents into Web Servers under Security Concerns. Proceedings of the Thirty-First Hawaii International Conference on System Sciences Volume: Jan. 1998.

[7] J. Anderson. Computer Security Threat Monitoring and Surveillance. Technical report, James P Anderson Co., Fort Washington, PA, Arpil 1980.

[8] D. Denning. An intrusion-detection model. Proceedings of the IEEE Symposium on Security and Privacy, pages 118-131, 1986.

[9] D. Bauer and M. Koblentz. NIDX – an expert system for real-time network intrusion detection.

[10] R. Schoonderwoerd, O. Holland, and J. Bruten. Ant-like agents for load balancing in telecommunications networks. Proceedings of the first International Conference on Autonomous Agents, 1997.

[11] J. Balasubramaniyan, J. Garcia-Fernandez, D. Isacoff, E. Spafford, and D. Zamboni. An Infrastructure for Intrusion Detection using Autonomous Agents. COAST technical Report 98/05, June 11, 1998.

[12] M. Bernardes, E. Moreira. Implementation of an Intrusion Detection System Based on Mobile Agents. Proceedings of the International Symposium on Software Engineering for Parallel and Distributed Systems, 2000.

[13] G. White, E. Fisch, and U. Pooch. Cooperating security managers: A peer-based intrusion detection system. Network, IEEE, Volume: 10, Issue: 1, 1996.

[14] J. Barrus and N. Rowe. A distributed autonomous-agent network-intrusion detection and response system. In proceeding of the 1998 Command and Control Research and Technology Symposium, 1998.

[15] A. Smith. An Examination of an Intrusion Detection Architecture for Wireless Ad Hoc Networks, 5th National. Colloquium for Information System Security Education, May2001.

[16] Snort website: www.snort.org