A New Mobile Agent-Based Intrusion Detection System Using Distributed Sensors

Mohamad Eid

American University of Beirut, Department of Electrical and Computer Engineering,
P.O.Box 11-0236 Beirut 1107 2020 Lebanon.

E-mail: mae33@aub.edu.lb

Abstract

This paper presents a distributed intrusion detection system (IDS), based on mobile agents, that detects intrusion from outside the network segment as well as from inside. Remote sniffers are controlled by the IDS via mobile agents, which gather intrusion detection data and send them back to the main station for analysis. The system shows a superior performance compared to central sniffing IDS techniques, and saves network resources compared to other distributed IDSs that activate too many sniffers causing bottlenecks in the network. The proposed model comprises three major components: The Network Intrusion Detection Component, the Mobile Agent Platform, and distributed sensors residing on every device in the network segment.

Key words: Mobile Agents, Intrusion Detection, Distributed Systems.

1. Introduction

Computer networks, including the world wide Internet, have grown in both size and complexity. The services they offer made them the main means to exchange data and an optimal environment for e-businesses. Unfortunately, they have also become the means to attack hosts and legitimate users. The growing importance of network security is shifting security concerns towards the network itself rather than being host-based. Security systems will soon evolve into network-based and distributed approaches to deal with heterogeneous platform technologies and support scalable solutions.

Among all security issues, intrusion is the most critical and widespread. Intrusion can be defined as an attempt to compromise, or otherwise cause harm, to a network. Intrusion detection involves the act of detecting unauthorized and malicious access one or more computers. In addition to identifying attacks, the IDS can be used to identify security vulnerabilities and weaknesses, enforce security policies, and provide further system auditing by exploiting the logs/alerts from the output component of the IDS.

Of a particular interest, mobile agents are intelligent program threads that function continuously and are able to learn, communicate and migrate themselves from host to host to gather information and perhaps perform specific tasks on behalf of a user [1]. A number of possible advantages out of using mobile code and mobile agent computing paradigms have been cited. This includes overcoming network latency, reducing network load, performing autonomous and asynchronous execution, and adapting to dynamic environments [3]. Moreover, implementation of mobile agents in languages such as JAVA provided mobile agent with system and platform independence and considerable security features, which are a necessity in intrusion detection systems [2].

The presented system in this paper addresses many issues in current IDSs. First, the approach provides a highly distributed IDS that reduces traffic in the network. There are local processing units to analyze relevant data and send summaries of alerts to the main station. Second, current IDSs such as the one described in [18] comprise many sensors distributed over the network and a centralized management station. These systems cause many bottlenecks and consume a lot of network resources. In the proposed system, mobile agents are dispatched to hosts where they activate the sensor there, process collected data, and send it to the main station, which signals the agents to either stop collecting data or continue, with possible changes to the collection frequency and context.

This paper is organized as follows: We present in the next section a literature review of previous work in the domain of mobile agent-based intrusion detection systems. Then, we describe our system and describe in details the different components, including its agent population and their interactions. We briefly discuss the advantages and drawbacks of the current state of the art. Next, we go over the partial results obtained from a prototype that we've built. Finally, we provide directions for future work.

2. Literature Review of Previous Work

Historically, the intrusion detection technology dates back to 1980 ([4]) and became a well-established research area after the introduction of the model of [5] and the prototypes presented in [6] and [7]. These systems were centralized. A single machine monitors data flow at a strategic point in the network and collects and analyzes data from the log files. Once an attacker destabilizes 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 IDSs.

Distributed IDSs were introduced to overcome this susceptibility where mobile agents are considered to play a prominent role in the implementation of such technologies. The approach in [8] proposes an architecture for a distributed intrusion detection system based on multiple independent entities called Autonomous Agent for Intrusion Detection (AAFID) framework. The proposed system allows data to be collected from multiple sources, thus combining traditional host-based and network-based IDSs. Several problems face this framework including scalability, performance, security, and user interface. Agents can be added or removed dynamically from the system, and whenever a new form of attack is identified, new specialized agents can be deployed into the system [9].

Subsequent work like [11], [12], or [10] present a fully distributed architecture where data collection and information analysis are performed locally without referring to the central management unit. For instance, [10] proposes a system imitating the functioning of natural distributed systems to achieve the efficiency found in natural systems. In this system, the detection of an intrusion triggers an alert pheromone (represented by mobile agents) that diffuses in the network searching for antibody agents. Mobile response agents (the lymphocytes) will migrate to the battlefield to initiate a defensive action.

3. System Architecture

This section presents the architecture of our distributed IDS. The architecture is made up of the following components: (1) an intrusion detection processor, (2) a mobile agent platform, and (3) distributed sensors. A high level view of the architecture is given in Figure 1.

A. Intrusion Detection Processor (IDP)

This component is the cornerstone of our distributed framework. It is responsible for monitoring network segments (subnets), and acts as a central intrusion detection and agent data processing unit. The unit is placed on a strategic node to monitor network traffic for all devices on the segment. Furthermore, it is setup to send real-time alerts that are generated using rule-sets to check for errant packets entering into the segment. It has three main capabilities: packet sensing, packet logging, and intrusion detection.

Every now and then, log files are sent to the central intrusion processing unit (via mobile agents) for packet decoding and processing. The IDP monitors agent's movement in the network and guides them towards critical locations in the network if malicious activities were detected. To guarantee proper interaction with mobile agents, the IDP should exchange data and messages with the mobile agent platform. As a network watcher, the IDP provides the following intrusion detection services:

- Monitor incoming network traffic
- Integrate correlating data sent by individual mobile agents to implement a multi-point detection, especially to deal with distributed attacks coming from within the network.
- Monitor established connections within the network at low level by scanning packets.
- Gather evidence of the attacker's behavior during the time window between the attack detection and the response.
- Look for the exploitation of known vulnerabilities in the network by checking on local intrusion signatures such as files integrity and user behavior profiles.

B. Mobile Agent Platform

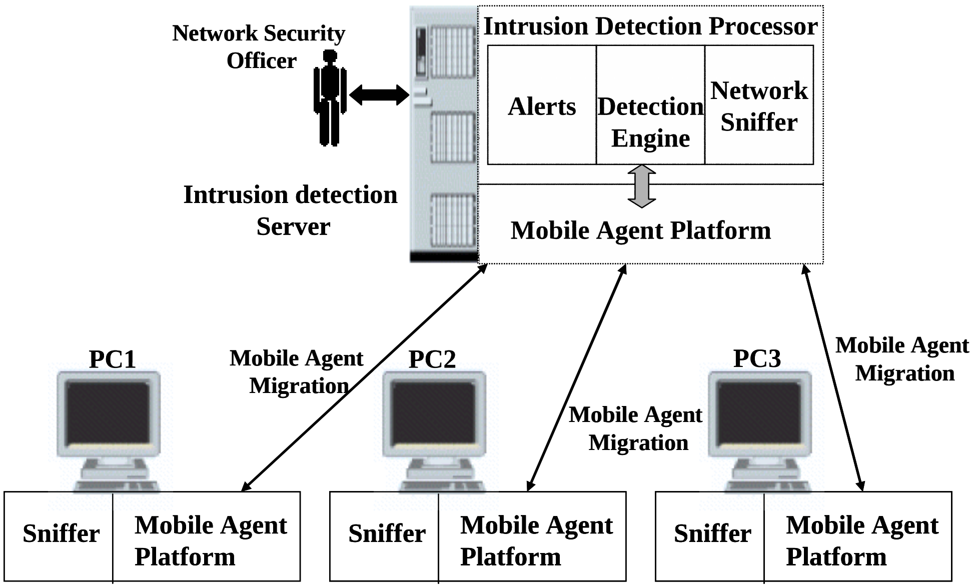

Figure 1: General Architecture for the System

A mobile agent platform (MAP) can create, interpret, execute, transfer, and terminate (kill) agents. The platform is responsible for accepting requests made by network users (in our case the IDP) and generating mobile agents plus sending them into the network to handle the tasks (in our case to start sniffing activities within the local network, stop it when necessary, and send the collected data back to the IDP for further analysis).

C. Distributed Sensors or Sniffer

A sniffer [15] is a device used to tap into networks to allow an application or hardware device to eavesdrop on network traffic. The traffic can be IP, IPX, or AppleTalk network packets. 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 tidbits of data. Depending on the IDP's instructions, the agent may run the sniffer for a predetermined period of time, collect the data, and send it in one batch to the IDP. Alternatively, it may run the sniffer and send data as it is captured to the IDP until it receives instructions to stop sniffing.

D. How does it work?

When the system is initially started, the IDP starts its own sniffer and sends a 'START' request to the MAP. 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 IDP has a registry containing all IP addresses in the local network. The MAP, in turn, creates the agents and dispatches them into the network. Now assume that an agent on its trip sends a report to the IDP that triggered an alarm. The IDP will send a 'LUDGE' message to the agent causing it to reactivate the sniffer at its current location and stay there, in an effort to gather more evidences on the current attack in order to study the behavior. The IDP will prompt the MAP to create a new agent that will takeover the agent's task. In this scenario, the number of active sniffers may increase to form an alert stage for faster reaction.

4. Implementation

A. IDS Implementation

The prototype IDS has been implemented on top of Snort [14] and a mobile agent system that was created locally. Snort is a full-fledged open-source network based IDS (NIDS) that has many capabilities such as packet sniffing, packet logging and intrusion detection [15]. Snort is a signature-based IDS that uses rule-sets to check for errant packets crossing a node in the network. A rule is a set of requirements that will trigger an alert. Snort was chosen as the NIDS because of

Table 1: Messages exchanged in the system

Message	Arguments	Description
Message Exchange between MAS and Snort		
CONNECT	IP address, Port #	Requests an HTTP connection between Snort and MAP.
START	# of agents, IP lists	Snort to MAP to create and dispatch agents when the system starts.
LOGRCVED	None	MAS tells Snort that the log file is successfully received.
PROCEED	None	Snort sends this signal if no alerts were generated out of the log file.
LODGE	None	Snort sends this signal if malicious activities were detected.
CLOSE	None	To terminate the HTTP connection between Snort and MAP.
Messages Exchanged between the local MAS and the Agent		
NXTCONNECT	IP address, Port #	Starts a connection between the agent and next host for migration.
SENDFILE	None	The agent sends this message to its MAP to copy itself to new host.
CONNECT	IP address, Port #	Requests an HTTP connection to the MAP.
SENDDATA	None	Send log file from the host where the agent resides to the main station.
SNDINFO	None	Sends information about host (host name, IP address, active directory).
PROCEED	None	Tell agent to continuously run sniffer when an intrusion is detected.
CLOSE	None	Close the client socket with the next host.
DELETE	None	Tell the MAP residing to delete the agent's directory.
LODGE	None	MAP Sends this signal if malicious activities were detected.

its availability, ease of configuration and customization.

B. Mobile Agent System (MORPHEOUS)

MORPHEOUS [16] is a prototypical mobile agent system that was developed as a final year project at the American University of Beirut. The system was chosen as the mobile agent platform because of its availability (including C# source code), ease of running, and support for mobile agents. It consists of four entities: the agent factory (AF), the listeners, the officer agents (OA), and the soldier agents (SA). The core of the agent system is the agent factory. It accepts requests made by the network users (in our case the Snort requests), generates the mobile agents and sends them to the network to handle tasks. On the AF host, many officer agents reside to keep track of the dispatched agents (Soldier Agents) over the network and the data gathered by these agents. The last element is the listener, which is a small program that will reside in each host in the network and will be responsible for accepting, running, and deleting SAs.

C. WinDump Sniffer:

WinDump [17] is the porting to the Windows platform of TcpDump that runs on all the operating systems supported by WinPcap, i.e. Windows 95, 98, ME, NT4, 2000 and XP. It was selected in the prototype because of its lightweight, popularity, support of multiple operating system, and ability to dynamically reconfigure its execution state.

D. Discussion and Results

Figure 2 presents the prototype network that we used to proof-concept our work. The network comprises a Linux server and two Windows hosts. Network credentials about the three computers are shown in the figure. The system is configured as follows: The Linux box is set as the intrusion detection processor where Snort is installed and is running in addition to the mobile agent platform. The other two PCs have WinDump installed on each as well as the mobile agent platform.

When the system starts up, Snort sends MORPHEOUS an HTTP request to start sniffing and provides it with the IP addresses of PC1 and PC2. MORPHEOUS creates an agent, assigns to it the task of starting and stopping WinDump and then dispatches it into the network. The MAP listens to Snort at a specific IP address and port

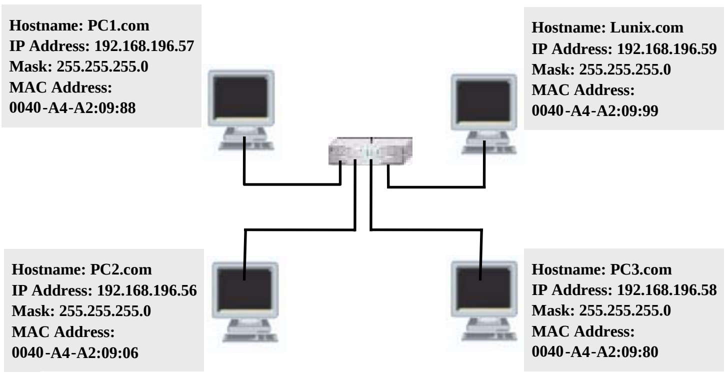

Figure 5: The Sample Network.

number. When a request is sent, the MAP checks for the type of the message (START, PROCEED, or LODGE). A summary of possible message exchanges between Snort, MAP, and the agent are detailed in Table 1. Using several experiments, the overall trip of the agent took roughly 4.42 sec (4 sec are for activating the sniffers and 0.42 sec for agent migrations, messaging between the components, and processing activities).

5. Conclusion and Future Work

Inspired from real life where policemen roam city streets looking for dangerous people and when they suspect something, they watch and follow more closely, we present an architecture for Distributed Intrusion Detection System based on mobile agents. An expansion of the distributed IDS seems to be possible using response and immunity components. Automating the response mechanisms decreases the time window an attacker has before being encountered by a human.

References

[1] Stefan Fuenfrocken. How to Integrate Mobile Agents into Web Servers. Technical Report, Department of Computer Science, Darmstadt University of Technology, Alexanderstr. 10, D 64283 Darmstadt, Germany.

[2] Stefan Fuenfrocken. Integrating Java-based Mobile Agents into Web Servers under Security Concerns. Technical Report, Department of Computer Science, Darmstadt University of Technology, Alexanderstr. 6, 64283 Darmstadt, Germany.

[3] Wayne Jansen, Peter Mell, Tom Karygiannis, Don Marks. Applying Mobile Agents to Intrusion Detection and Response. NIST Interim Report (IR) - 6416. ACM October 1999.

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

[5] D. E. Denning. An intrusion-detection model. In proceeding of the IEEE Symposium on Security and Privacy, pages 118-131, April 1986.

[6] D. S. Bauer and M. E. Koblentz. NIDX – an expert system for real-time network intrusion detection. In Proceeding of the Computer Networking Symposium, pages 98-106, Washington, DC, April 1988.

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

[8] Jai Sundar Balasubramaniyan, Jose Omar Garcia-Fernandez, David Isacoff, Eugene Spafford, Diego Zamboni. An Infrastructure for Intrusion Detection using Autonomous Agents. COAST technical Report 98/05, June 11, 1998.

[9] Richard Feiertag, Sue Rho, Lee Benzinger, Stephen Wu, Timothy Redmond, Cui Zhang, Karl Levitt, Dave Peticolas, Mark Heckman, Stuart Staniford, and Joey McAlerney. Intrusion detection inter-component adaptive negotiation. Computer Networks 34 (2000) 605-621.

[10] Serge Fenet and Salima Hassas. A distributed Intrusion Detection and Response System based on mobile autonomous agents using social insects communication paradigm. Published by Elsevier Science B. V., 2001.

[11] G. B. White, E. A. Fisch, and U. W. Pooch. Cooperating security managers: A peer-based intrusion detection system. 10(1): 20-23, 1996.

[12] 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.

[13] Sabeel Ansari, Rajeev S.G., and Chandrashekar H.S. Packet Sniffing: A Brief Introduction. IEEE, JANUARY 2003.

[14] Snort website: www.snort.org (Accessed in January 15, 2003)

[15] Martin Roesch. Snort - Lightweight Intrusion Detection for Networks. A white paper on the design features of Snort 2.0 from: www.sourcefire.com/technology/whitepapers.html (accessed in January 15, 2004).

[16] Mohamed Mohsen and Khaled Heloue. Mobile Agents System for Data Retrieval. Final Year Project Report, American University of Beirut, August 2003.

[17] The main website of Windump: www.tcpdump.org (Accessed in January 10, 2004).

[18] Rajeev Gopalakrishna, Eugene H. Spafford. A Framework for Distributed Intrusion Detection using Interest Driven Cooperating Agents. Purdue University, 2001.