International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:1, No:5, 2007

Multisensor Agent Based Intrusion Detection

Richard A. Wasniowski

***Abstract*—** In this paper we propose a framework for multisensor intrusion detection called Fuzzy Agent-Based Intrusion Detection System. A unique feature of this model is that the agent uses data from multiple sensors and the fuzzy logic to process log files. Use of this feature reduces the overhead in a distributed intrusion detection system. We have developed an agent communication architecture that provides a prototype implementation. This paper discusses also the issues of combining intelligent agent technology with the intrusion detection domain.

***Keywords*—** Intrusion detection, fuzzy logic, agents, network security.

I. INTRODUCTION

WEB attacks are rapidly becoming one of the fundamental threats for information systems that are connected to the Internet. Malicious Internet traffic is a profound threat to both the Internet itself and to the infrastructures that use the Internet for communication. Recently published study [26] shows that malicious traffic causes an increase in the average DNS latency by 230% and an increase in the average web latency by 30%. Unfortunately, protecting networks from malicious traffic remains a problem in both the research and real life applications. When the attacks are analyzed, it is observed that most of them are similar using a reduced number of attacking techniques. It is generally agreed that classification can help designers and programmers to better understand attacks and build more secure detection and response applications. The rapid proliferation of wireless networks and mobile computing applications has changed the landscape of network security. The recent attacks on major Internet sites have shown us that no computer network is immune from intrusions. The wireless ad-hoc network is particularly vulnerable due to its features of open medium, dynamic changing topology, cooperative algorithms, lack of centralized monitoring and management point, and lack of a clear line of defense. The traditional way of protecting networks with firewalls and encryption software is no longer sufficient and effective.

Manuscript received March 31, 2005. R. A. Wasniowski is with the Computer Science Department, California State University, Carson, CA 90747.

Many intrusion detection techniques have been developed on fixed wired networks but have been turned to be inapplicable in this new environment [14]). We need to search for new architecture and mechanisms to protect wireless networks and mobile computing application. Over the past few years, intrusion detecting agents have emerged as a new software solution. Agents represent a new generation of computing systems and are one of the more recent developments in Intrusion Detection Technology. Agents are applications with predefined goals and run autonomously. They can for example, monitor an environment and issue alerts or start intervention actions based on how they are programmed. In the case of intrusion detection agents, they can serve as detectives or monitors by recognizing and retrieving data from multiple sensors for analysis and development of real-time alerts. Intelligent agents can assist users and act on their behalf. Agents can automate repetitive tasks, remember events, summarize complex data, learn, and make recommendations. Intelligent agents continuously perform two main functions, which differentiate them from other software programs: they collect data from the environment in which they operate and reason to interpret data and suggest actions. A MAS (multi agent systems) is an emerging subfield of AI that aims to provide both principles for construction of complex systems involving multiple agents and mechanisms for coordination of independent agents' behaviors. While there is no generally accepted definition of agent in AI, for the purposes of this study, we consider an agent to be a software entity. Agents can also reduce the intrusion detection workload by sifting through large amounts of data for evidence gathering. While there are multiple definitions of intelligent agents, their essential characteristic in intrusion detection is that agents are software computing entities that perform intrusion detection tasks autonomously. In order to define the characteristics of an agent further, and distinguish them from any other type of program, the following lists the attributes of typical agent systems:

- Autonomy - Being able to carry out tasks independently is the most important feature of an agent.
- Purpose - Agents perform a set of tasks on behalf of a user or other agents that are explicitly approved and programmed.
- Perception - Agents need to be able to affect is environment using some type of predefined mechanisms.
- Communications - An agent needs to be able to interact with the users and other agents.

- Intelligence - An agent needs to be able to interpret monitored events to make appropriate decisions.

Developers normally do not set out to construct an agent, but more typically they add new functionality to an existing application. Agents reason through simple to elaborate networks of rules: IF X AND Y THEN Z. To develop intelligence in agents, certain steps can be taken involving the following type of rule sequencing and construction: The user or developer provides a set of rules that describe a desired behavior: When X and Y happens, then do Z. The reasoning system is provided with interfaces to perform or initiate various desired actions; for example, it may require that an alert be sent by way of a message to a system object, by writing a file, or by other system action that a program can perform. After the reasoning system is initiated, it can wait for an event to arrive. It will extract facts from the event, and then evaluate its rules to see if the new facts cause any of them to fire. If one or more rules fire, it may cause additional action to be initiated or a record to be written or updated. The above process leads to the creation, and use of conditional rules and logic, which can be coded in a variety of ways. Unlike an expert system, an agent is embedded in its environment. It can perceive and react to it using the inputs of conditions. It can dynamically construct new rules as it works. In other words agents are capable of using sensors to monitor their environment, develop new rules, and then take actions independently.

II. Intrusion Detection Problem

Current best practices for protecting networks from malicious traffic are to deploy a security infrastructure that includes network intrusion detection systems. While those systems are useful for identifying malicious activity in a network, they generally suffer from several major drawbacks: inability to detect distributed or coordinated attacks, high false alarm rates and from producing huge amount of data that is difficult to process. False alarms and timely identification of new attacks are two of the biggest challenges to the effective use of network intrusion detection systems [15]. A potential means for addressing these shortcomings is employing multiple, distributed network intrusion detection systems. In this paper we consider the potential benefits of distributed network intrusion detection systems by addressing two open problems. The first problem is how to combine data from multiple intrusion sensors in a network. This is known as the fusion problem. The second problem is how to identify the most important data provided by multiple sensors in a network. This is known as the filtering problem.

One promising approach to addressing the intrusion problems is through the use of distributed network intrusion detection systems and information fusion. A distributed network intrusion detection systems is composed of diverse set of sensors that coordinate to identify malicious traffic. Under the DARPA dynamic database program, a number of government contractors collaborated to develop prototype multi-sensor single-stage systems. While single-stage fusion architectures are known to be optimal for several applications, there are several arguments in support of the multi-stage system. The most obvious benefit is system flexibility. In particular, a number of sensor modules may be employed in a variety of multi-stage architectures [17]. Minor extensions would enable its use in decentralized diversified settings. Sensor diversity appears in three dimensions: information, time, and space. Different sensors can measure different features of network traffic or might use different detection algorithms i.e., they can provide fundamentally different information about the network traffic which can be used to improve intrusion detection capabilities.

Since many attacks consist of multiple steps, sensor measurements can become correlated across time. Correlating sensor readings or events that occur at different times can also lead to improved intrusion detection. Sensors that monitor different sets of network addresses will have a similar affect on their measurement under such widespread attacks. In the context of intrusion detection, spatial diversity can be used to improve both the time-to-detection and false-alarm rate. Our goal for this work is to investigate how to combine data from spatially diverse distributed network intrusion detection systems in order to improve false-alarm rates in detecting attacks. We developed and implemented an intrusion detection architecture called Fuzzy Agent-Based Intrusion Detection System (FABIDS). The architecture of this system is presented below in Fig 1. Using an agent, as opposed to a search engine, has the advantage that all of these links can be viewed at any time. The advantages of this architecture is that a low volume of data must be sent over network in a distributed intrusion detection scenario This feature allows easy exploration of the trade-off between sensitivity and selectivity that affects the rate of false decisions. The distributed nature of the system and the fact that each agent is an autonomous entity increases the efficiency of the processing. We are using combination of analytical models and simulations for filtering and fusing data in order to explore the problem space (see ref [14]). Our approach is to consider probabilistic (and fuzzy) intrusion detection using thresholding, i.e., an alarm is raised if a sensor reading goes above a specified threshold. In such a system, a false alarm is defined as an alarm raised during normal conditions. We assume that there are N sensors located at different sites and an alarm is raised if the average reading of the sensors exceeds a given threshold. Since the sensors are deployed at different sites, averaging their readings should reduce the variance of the normal component of the traffic. Moreover, since the attack component of the traffic typically is correlated at different sites, averaging should not affect (sometimes increases) the variance of the attack component. Thus, the questions become what is the granularity of the data that should be passed to the analyzing site and how should the data be combined at the analyzing site? We are developing a series of analytic and simulation models to addressed this problem and assess the potential benefits of distributed sensor based intrusion detecting systems for reducing false alarms and improving timeliness of detection for different fusion and

filtering strategies. Our analysis explores the trade-offs when fusion and filtering are used together and shows that significant improvements are possible. We have initiated several projects related to developing sensor networks based Intrusion Detection Systems. As a result of these projects we have accumulated knowledge about methods to monitor and detect attacks, monitor and analyze systems logs and network packets. The main goal of those projects is to improve timeliness and accuracy in identifying attacks. Our analysis shows that combining data from distributed sensors reduces the noise caused by benign traffic while enhancing the signal caused by the attack traffic.

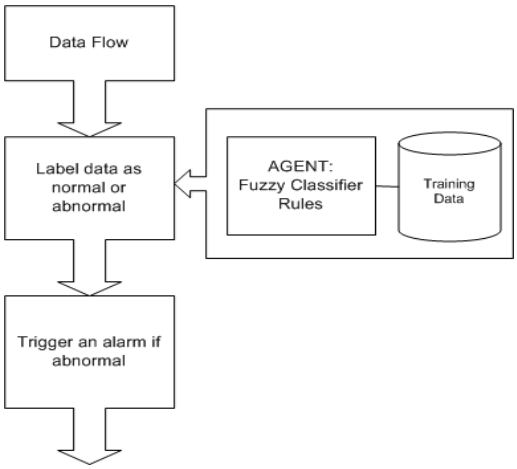

Fig 1. FABIDS Architecture

Our simulations demonstrate also that even simple summary data fusion schemes can be effective in increasing timeliness of attack identification and in reducing false alarm rates. These results indicate that distributed detection methods have promise for defending networks from malicious attacks.

III. Experiments

While conducting the research for this paper, the researcher was provided full access to the SNORT logs [13, 14]. The basic SNORT architecture is made up of three main parts, the packet decoder, detection engine and alerting system and logging system. The packet decoder can collect TCP/IP traffic at a blinding rate. Before the engine can compare any of the signatures in its database to the packets, the packet data is passed through a number of user-configurable pre-processors. These pre-processors can reassemble TCP packets into sessions, handle fragmented traffic, and even detect scans and probes. After the preprocessors have formatted the packet data to make it easier to search, the detection engine examines the contents for data that match any of the signatures in its database. If any of the signatures are matched, then the action prescribed for the signature is taken by the third part of SNORT, the alert/log system. If configured, SNORT will also capture the packet data relating to the alert and store it on the hard drive. The alert system will publish alerts to an area on the file system for examination or to a remote analysis console through standard remote log formats like syslog. To encode the descriptions of various attacks, a range of positive integers is assigned to each of the attack in the following way.
Entry point (1 bit of information):
Web server software (filters, Perl modules, etc.) or web application (HTML, server-side and client-side scripts, server components, SQL sentences, etc.)
Vulnerability (3 bits of information):
Code Injection, HTML manipulation, Overflows, Misconfiguration (default directories, sample applications, guest accounts, etc.) X if Not applicable,
Threat (3 bits of information):
Authentication, Authorization, Confidentiality, Integrity, Availability, Auditing
Action (4 bits of information):
Read, Modify, Delete, Fabricate, Impersonate, Bypass, Search, Interrupt, Probe, Unknown,
Length (1 bit of information):
Expected, Unexpected (unusually long), X - Not applicable,
HTTP element (7 bits of information):
GET/POST, HOST, COOKIE, REFERER, TRANSLATE, SEARCH, PROPFIND
Target (1 bit of information) Web application (source files, customers' data, etc.), Platform (OS command execution, system accounts, network, etc.)
Scope (1 bit of information)
Local (one user affected), Universal (all users affected), X - Not applicable
Privileges (1 bit of information):
0 - Unprivileged user, 1 - Administrator/root, X - Not applicable.

Let us consider typical common attacks directed against different types of web servers and platforms.

0, X, 1, 9, 0, 01, 1, X, 0
0, 1, 2, 0, 0, 01, 0, X, X
1, 0, 1, 3, 0, 01, 1, X, 0

In this description the web application allows SQL injection. The attacker exploits this vulnerability by executing a SQL Server extended procedure, and then adds himself to the OS users. These encoding vectors are useful in a number of ways especially in intrusion detection systems. An intrusion detection system (IDS) detects and reports attempts to break into or misuse networked computer system in real time. A traditional IDS consists of three functional components: A monitoring component, such as a packet capturer, which then

collects traffic data. An inference component, which analyzes the captured data to determine whether it corresponds to normal activity or malicious activity. And an alerting component, which generates a response when an attack has been detected. This response can be passive such as writing an entry in an event log or active such as changing configuration rules in the firewall to block the attacker's IP address. Coding web attacks into vectors could help the post processing of IDS alerts. Encoding web attacks into vectors helps the application-level firewall to decide about the action to be taken when an attack is detected. In [15, 25] real world examples of attacks against different platforms, web servers, and applications are given to illustrate how this taxonomy can be applied.

IV. CONCLUSION

As computer attacks become more and more sophisticated, the need to provide effective intrusion detection methods increases. Network-based, distributed attacks are especially difficult to detect and require coordination among different intrusion detection components or systems. We propose a solution that is more effective than current IDS's. This architecture allows local analysis and sharing of results as well as minimizing the communication costs.

We are planning to extend this work in three ways. The first is in a deeper exploration of the analytic methods for fusing data. It seems clear that our threshold-based methods can be improved. The second area of investigation is in applying our results to empirical data that we are collecting in conjunction with several intrusion detection projects. The third area is to expand our analysis to include information diversity.

REFERENCES

[1] S. Axelsson. "Intrusion Detection Systems: A Taxonomy and Survey." Technical Report No 99-15, Dept of Computer Engineering, Chalmers University of Technology, Sweden, March 2000
[2] Russell, S. J. & Norvig, P.(1995). Artificial Intelligence—A modern approach. Upper saddle River ,NJ:Prentice Hall Inc.
[3] W. Jansen, P. Mell, T. Karygiannis, and D. Marks. "Applying mobile agents to intrusion detection and response." NISTIR-6416, September 1999
[4] Young-Gyun Kim, M. Valtorta, and J. Vomlel. "A Prototypical System for Soft Evidential Update." USC CSCE TR2002-005, Department of Computer Science and Engineering, University of South Carolina, Columbia, 2002.
[5] Steffen L. Lauritzen and David J. Spiegelhalter. "Local Computations with Probabilities on Graphical Structures and their Application to Expert Systems." Journal of the Royal Statistical Society, Series B, 50 (1988), 2, pp.157-224.
[6] W. Lee and S.J. Stolfo. "Data Mining Approaches for Intrusion Detection." In Proc. of the 7th USENIX Security Symp, San Antonio, TX, 1998, pp.79-94
[7] M. Meneganti, F.S. Saviello, and R.Tagliaferri. "Fuzzy Neural Networks for Classification and Detection of Anomalies." IEEE Trans. On Neural Networks, 9/5, 1998, pp. 848-861
[8] S. Northcutt, Network Intrusion Detection: An Analyst's Handbook, New Riders, 1999
[9] J. Moy. OSPF version 2. Internet Draft, RFC-2178, July 1997
[10] Judea Pearl. Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan-Kaufmann, 1988.
[11] Studer, R., Benjamins, V. R., Fensel, D. (1998). Knowledge Engineering: Principles and Methods. Data Knowledge Engineering, 25 (1-2).
[12] Marco Valtorta, Young-Gyun Kim, and Jirí Vomlel. "Soft Evidential Update for Probabilistic Multiagent Systems." International Journal of Approximate Reasoning, 29, 1 (January 2002), pp.71-106.
[13] A. Valdes and K. Skinner. "Adaptive, Model-Based Monitoring for Cyber Attack Detection." In Proc. RAID, 2000, pp. 80-92
[14] Wasniowski RA, Agent Based Design Methodology, RAW-TR-00-12
[15] Wasniowski RA, Intrusion Detection System with Fuzzy Logic Agent, RAW-TR-01-09
[16] Wooldridge, M., and Jennings, N. (1995) "Intelligent Agents: Theory and Practice," Knowledge Engineering Review, Vol. 10, No. 2.
[17] J. Allen, A. Christie, W. Fit hen, J. McHugh, J. Pickle, and E. Stoner. State of the practice of intrusion detection technologies. Technical Report CMU/SEI-99-TR-028, Software Engineering Institute, Carnegie Mellon University, January 2000.
[18] T. Bass. Intrusion Detection Systems and Multisensor Data Fusion. Communications of the ACM, 43(4):99-105, April 2000.
[19] T. Bass, Alfredo Freyre, David Gruber, and Glenn Watt. EMail Bombs and Countermeasures: Cyber Attacks on Availability and Brand Integrity. IEEE Network, pages 10-17, March/April 1998.
[20] J. Baras, A. Cardenas, and V. Ramezani. On-line Detection of Distributed Attacks from Space-time Network Flow Patterns. In Proceedings of 24th Army Science Conference, November, 2004.
[21] K.C. Chang, R.K. Saha and Y. Bar-Shalom, On optimal track-to-track fusion. IEEE Transactions on Aerospace and Electronic Systems 33 4 (1997).
[22] H. Chen, T. Kirubarajan, Y. Bar-Shalom, Comparison of Centralized and Distributed Tracking Algorithms Using Air to Air Scenarios, in: Signal and Data Processing of Small Targets 2000, Proceedings of SPIE Vol. 4048, 2000, pp. 440–451
[23] Y. Bar-Shalom, Performance Limits of Track-to-Track Fusion versus Centralized Estimation: Theory and Application, in: Fourth ONR/GTRI Workshop on Target Tracking and Sensor Fusion, May 2001, Monterey, CA.
[24] S. Coraluppi, C. Carthel, M. Mallick, Hierarchical Multi-Hypothesis Tracking with Application to Multi-Scale Sensor Data, to appear in: Proceedings of the 2002 IEEE Aerospace Conference, March 2002, Big Sky MT, USA
[25] M. M. Mizushima, SnortMart, a Network Intrusion Detection System Data Mart, graduate senior project, CSUDH 2005.
[26] Kun-chan Lan, Alefiya Hussain, Debojyoti Dutta, Effect of Malicious Traffic on the Network, presented at PAM2003, the Passive and Active Measurement Workshop, April 6-8, 2003, La Jolla, CA, USA