LIDS: A Learning Intrusion Detection System

by

Mayukh Dass

(Under the direction of Dr. Walter D. Potter)

Abstract

The detection of attacks against computer networks is becoming a harder problem to solve in the field of network security. The dexterity of the attackers, the developing technologies and the enormous growth of internet traffic have made it difficult for any existing intrusion detection system to offer a reliable service. However, a close examination of the problem shows that there usually exists a behavioral pattern in the attacks that can be learned and can be used to detect intrusions more effectively. Thus, there is a requirement for a system with learning and adapting capabilities for optimal performance.

This thesis discusses a Learning Intrusion Detection System called LIDS that includes a blackboard-based architecture with autonomous agents. It has the capability for online learning, which may result in better performance than present systems. This feature enables the system to adapt to changes in the network environment as it assimilates more network data.

Index words: Intrusion Detection, Blackboard Architecture,
Autonomous Agents, Machine Learning,
Artificial Neural Networks, Genetic Algorithm

LIDS: A Learning Intrusion Detection System

by

Mayukh Dass

B.E., Nagpur University, India, 2000

A Thesis Submitted to the Graduate Faculty
of The University of Georgia in Partial Fulfillment
of the
Requirements for the Degree

Master of Science

Athens, Georgia

2003

LIDS: A Learning Intrusion Detection System

by

Mayukh Dass

Approved:

Major Professor:	Dr. Walter D. Potter
Committee:	Dr. James Cannady Dr. Ron McClendon

Electronic Version Approved:

Maureen Grasso
Dean of the Graduate School
The University of Georgia
August 2003

DEDICATION

To my parents....

Acknowledgments

Om Shri Sai Ram.

Behind every successful work, there is a lot of devotion, hard work and sacrifice. LIDS or Learning Intrusion Detection System would not have made it to this stage without the guidance of Dr. James Cannady, Dr. Donald Nute and Dr. Don Potter. I would like to thank Dr. James Cannady for introducing me to this interesting problem of network security. His knowledge, support and guidance has a great contribution to the success of this work. I would also like to thank Dr. Donald Nute for his comments and ideas during the early design stages of developing the system.

Dr. Don Potter's contribution is more than just being my major advisor. I especially thank him for teaching me how to do research work. He is the one who gave me directions to the library and took great pains in improving my writing skills. I would also like to thank Dr. Ron McClendon for his help with the Neural Network analyzer. His comments and ideas have been very helpful during the development of the system.

I would also like to thank NED-2 project leaders for allowing me to work with them. I acquired most of my skills of building large interactive cooperative systems from my graduate assistantship work while developing NED-2 software.

I would like to thank my parents for supporting me throughout my life. Their support and guidance gave me the power to struggle and survive during my hard times.

I would also like to thank my friends Ramyaa, Abhishek Jain, Shilpa Hardas, Hajime Uchiyama, Astrid Glende, Fred Maier, Christopher Thomas, Ning So, John

Burke and Lorina Naci for their support and help during my stay in Artificial Intelligence Programs.

Profound thanks to my girlfriend, Ani Amirkhanyan for her continuous support and trust in me. Thank you Ani for the love, care and support you have given me for all these years.

Table of Contents

Chapter 1

Introduction

With the exponential growth of the Internet and networked computers, cyber crime has become one of the most important problems in the computer world. Online credit card fraud, compromised computer servers and other privacy enormities have created a cloud of distrust among online customers. According to the 2002, CSI/FBI Computer Crime and Security Survey, the total revenue loss in industry due to computer network intrusion was calculated as $455,848,000, up from $35 million reported in 2001 [1]. These numbers justify the increase in research interest in computer security.

Intrusion Detection is a problem of identifying unauthorized users in a computer system. It is also defined as the problem of protecting computer network systems from being compromised. The first published renowned literature on computer network security is dated back in 1987 [2] where Denning discussed various security concerns, presented a definition of Intrusion Detection and discussed different types of Intrusion Detection. Most of the contemporary computer security research work is based on the milestone established by Denning.

Intrusion Detection tries to identify two general categories of attacks:

- Anomaly Detection.
- Misuse Detection.

Anomaly Detection identifies activities that vary from established patterns of users, or groups of users and Misuse Detection involves a comparison of a user's activities with the known behavior of system penetration. It is difficult for present systems to detect both kinds of attacks. From previous works, we found that maintaining a profile of each authorized user or group is useful in detecting anomaly attacks in systems, but it is difficult to maintain a behavior profile for each legitimate user when the number of users increases. Behavior patterns also change with the mental state of the person and thus detecting anomaly attacks by comparing patterns with user profiles gives rise to a large number of false positive alerts. Moreover, there is no good definition for a healthy system. So flagging any behavior that is not a feature of a healthy system is not a good approach.

Intrusion Detection Systems are special software that detects network intrusions. There are two types of Intrusion Detection Systems (IDS) namely:

- Host-based Intrusion Detection System.
- Network-based Intrusion Detection System.

Though their roots are similar, their operational use is radically different. The root of all intrusion detection is based in analyzing a set of discrete, time-sequenced events for patterns of misuse. All intrusion detection sources, network or host, are sequential records that directly reflect specific actions and indirectly reflect behavior. Host-based systems examine events like what files were accessed and what applications were executed. Network-based technologies examine events such as packets of information exchanged between computers (network traffic).

The available IDSs in the market are expensive and of only limited reliability. The increasing complexity of the Internet and the maintenance cost of these systems is a setback to the performance of IDSs. This has led to worldwide research interest in developing the Next Generation Intrusion Detection Systems (NIDS), which are able to learn and adapt to the network environment for optimal performance. Some of the recent work on developing effective network security highlights new areas of research, which include artificial intelligence, data mining, statistical techniques, agent frameworks including autonomous agents , intelligent agents and mobile agents for distributed intrusion detection.

An IDS is a conglomeration of capabilities used to detect and respond to threats. The security industry supplies tools with capabilities and features that do more than detect intruders. Presently, an IDS encompasses the following capabilities:

- Event log analysis for insider threat detection.
- Network traffic analysis for perimeter threat detection.
- Security configuration management.
- File integrity checking.

The above capabilities are very generic and do not attempt to learn new attacks or new intrusion techniques. Learning new attacks is very important in a dynamic environment like the Internet as it reduces maintenance and other recurring costs. This work presents LIDS: A Learning Intrusion Detection System that not only detects network intrusions but also adapts itself to the environment. It has the capability of learning new attacks and recognizing them on their re-occurrence. Though researchers have used machine learning techniques in the past to detect intrusions like Artificial Neural Networks, no one has ever attempted to build an intelligent and learning system that will use different building blocks of machine learning like Artificial Neural Networks, Genetic Algorithms or the ID3 algorithm as different modules of the system to detect and learn intrusions. LIDS has a blackboard based architecture and performs its task of detecting intrusions with the help of eight autonomous agents. The robustness of the blackboard based architecture and its ability to maintain a common information pool has made it an integral part of LIDS. Further discussion regarding the architecture is done elaborately in Chapter 2.

This thesis contains four publications from different conferences proving the validity of the work and a concluding chapter that shows the reports generated by LIDS and future works that need to be done. Chapter 2 presents the first publication titled “LIDS: Learning Intrusion Detection System” that was published in the proceedings of the 16th International FLAIRS conference, held in St. Augustine, FL on May 12-14, 2003. This paper discusses the architecture of LIDS and presents a brief description of each autonomous agents.

Chapter 3 presents a paper titled “A Blackboard-Based Learning Intrusion Detection System: A New Approach” that will be published in the proceedings of the 16th International Conference on Industrial & Engineering Applications of Artificial Intelligence & Expert Systems to be held in Loughborough, UK on June 23-26, 2003. This paper discusses the reason to use a blackboard architecture and how it is useful while building a system that is controlled by autonomous agents. This paper also explains the importance of using autonomous agents in detecting intrusions.

Chapter 4 presents a paper titled “A GA based Intrusion Sub-classifier Filter” appeared in the digital proceedings of the 41st ACM Southeast Conference, held in Savannah, on March 7 - 8, 2003. This paper presents a detailed documentation of the ’Attack Classifying’ agent or the fifth agent of LIDS. The attack classifier is a genetic algorithm based intrusion filter that is useful in identifying different attacks by reading the system specific data. It views the problem of detecting intrusions as abductive inference and considers it similar to the “multiple-fault diagnosis” paradigm.

Chapter 5 titled “The design of a learning agent in an Multi-agent Intrusion Detection System” that gives detailed information on the structure of the learning agent. It also discusses each step taken by LIDS to learn and detect new attacks. Some of the problems faced while building this system and their solutions are provided in this chapter.

Lastly, Chapter 6 discusses some of the results obtained from the system. It also shows some initial alerts and reports generated by the system. The importance of this work and the remaining future work is mentioned in this chapter.

1.1 References

[1] Power, R. (2002), Computer Security Issues & Trends. Vol. 8, No. 1, 2002. pg 4.

[2] Denning D E (1987), An Intrusion-Detection Model, In IEEE Transaction on Software Engineering, Vol. Se-13, No. 2, February 1987, 222-232.

Chapter 2

LIDS: Learning Intrusion Detection System[1]

[1] Dass M., J. Cannady and W.D.Potter to appear in the proceedings of the 16th International Flairs Conference, St. Augustine, Florida, 12-14 May 2003 pp 12-16.

2.1 Abstract

The detection of attacks against computer networks is becoming a harder problem to solve in the field of network security. The dexterity of the attackers, the developing technologies and the enormous growth of internet traffic have made it difficult for any existing intrusion detection system to offer a reliable service. However, a close examination of the problem shows that there usually exists a behavioral pattern in the attacks that can be learned and can be used to detect intrusions more effectively. Thus, there is a requirement for a system with learning and adapting capabilities for optimal performance. This paper discusses the design of a Learning Intrusion Detection System (LIDS) that includes a blackboard-based architecture with autonomous agents. It has the capability for online learning, which may result in better performance than present systems. This feature enables the system to adapt to changes in the network environment as it assimilates more network data.

Keywords: Intrusion Detection, Blackboard Architecture, Autonomous Agents, Artificial Neural Networks

2.2 Introduction

With the rapid increase in vulnerable Internet applications and automated attack scripts, intrusions of networked systems have become an increasing problem in the field of information technology. Every year, the business industry loses a huge amount of revenue due to data manipulation caused by computer network intruders. As a result, there has been an increasing requirement to effectively protect crucial business information with a reliable, robust and flexible intrusion detection system. There are many commercially available Intrusion Detection Systems (IDS) in the market. Unfortunately they are expensive and of only limited reliability. The increasing complexity of the Internet and the maintenance cost of these systems is a setback to the performance of IDSs. This has led to worldwide research interest in developing the Next Generation Intrusion Detection Systems, which are able to learn and adapt to the network environment for optimal performance. Some of the recent work on developing effective network security highlights new areas of research, which include artificial intelligence [1], data mining [2], statistical techniques [3], agent frameworks including autonomous agents [5], intelligent agents [6] and mobile agents [7] for distributed intrusion detection [4]. However, there has been only a limited amount of research carried out in developing a Learning Network Security System that can become more intelligent while it is detecting intrusions. This paper proposes a blackboard based three-tier autonomous learning agent architecture that has learning and adaptation capability for improved performance.

2.3 Intrusions and Priori Approaches in Intrusion Detection System

As the term ”Intrusion Detection System” suggests, we are trying to develop a network security system that will detect misuse behavior in the network data stream.

These security systems collect network data from the system and audit them in order to detect intrusions. Normally IDSs are located on a centralized server, but some distributed types of IDSs can be placed on different workstations to detect intrusions. The proposed architecture is a server based or centralized IDS.

The process of Intrusion Detection can be defined as the problem of identifying individuals who are using computer network resources without authorization or attempting to prevent authorized users from accessing network resources. In an organization, intrusions can take place from the Internet or from inside the organization's computer network system. This highlights the two different types of Intrusion Detection Systems; Host Based Intrusion Detection System and Network Based Intrusion Detection System. A Host Based Intrusion Detection System can be defined as a security system that is capable of detecting inside abuses in a computer network. A Network Based Intrusion Detection System is capable of identifying abusive uses or attempts of unauthorized usage of the computer network from outside the system. This paper describes a Network Based Intrusion Detection System that will use computational intelligence techniques to detect intrusions.

There are several forms of network intrusions:

- Denial-of-service Attack - This is particularly a serious form of attack that has resulted in damages worth millions of dollars over the past few years. While a significant problem, DoS attacks are usually quite simple. They typically involve an attacker disabling or rendering inaccessible a network-based information resource.

- Guessing rlogin Attack - Here the intruder tries to guess the password that protects the computer network in order to gain access to it.

- Scanning Attacks - The intruder goes about scanning different ports of the victim's system to find some vulnerable points from where they can launch other attacks.

Most of the present approaches in detecting intrusions utilize some form of rule-based analysis. Rule-Based analysis relies on predefined rule-sets that are provided by an administrator, automatically created by the system, or both. Expert Systems are the most common form of rule-based intrusion detection approaches [3]. Rule-based systems suffer from the inability to detect attack scenarios that may occur over an extended period of time. They also lack flexibility in the rule-to-audit record representation [8]. Slight variations in the attack sequence may reduce the effectiveness of the system.

An increasing amount of research has been conducted on the usability of neural networks in accurately detecting network attacks [8], and efforts have been made to integrate a rule-based system with a neural network to develop a high performance Intrusion Detection System. Research efforts have also been made to use non-traditional AI-based techniques like Genetic Algorithms [9], Data Mining [2] and Pattern Recognition Techniques [3] to develop a high performance IDS. Nonetheless, little effort has been applied to the development of an approach that possesses the capability for continuous learning. Researchers have primarily tried to identify different innovative techniques of detecting intrusions, but they have usually overlooked the potential of a learning system that can adapt itself in the network environment and give high performance with increased experience.

2.4 Blackboard and Autonomous Agents

The blackboard architecture is considered as one of the most general and flexible knowledge system architectures for building decision-based applications. It is highly preferred over other alternatives due to its modularity, dynamic control, generality, concurrency, high design efficiency, robustness and ability in dealing with multiple knowledge sources. As a result, the blackboard-based architecture is considered to be a good solution in developing LIDS.

The proposed architecture will include the use of Autonomous Agents. For the proposed architecture, we implement software agents that perform certain security monitoring functions at a host. The agents are independently running entities whose performance is not affected by any other entities. These kinds of agents are very useful in network security because they run continuously, can resist subversion and have minimal overhead. They are also configurable, easily adaptable, scalable, dynamically reconfigurable and degrade gracefully. The proposed LIDS architecture consists of autonomous agents that are integrated in a blackboard-based architecture. Proposed Architecture. The use of blackboard techniques and autonomous agents [5] in detecting network intrusions is not a new concept [5], [10]. Dasgupta described how blackboard-based agent architecture helps in detecting intrusions [10]. Dasgupta also developed a distributed blackboard architecture that is embedded among the agents. A manager agent controls the monitoring, decision and action agents. The unidirectional flow of information in the system has a major impact on the flexibility of the system. Balasubramaniyan applied rule-based autonomous agents to detect intrusions and as a result faced the same problems as faced by rule-based Intrusion Detection Systems. Rule-based systems lack the flexibility to identify new attacks in the network data stream and must be updated frequently

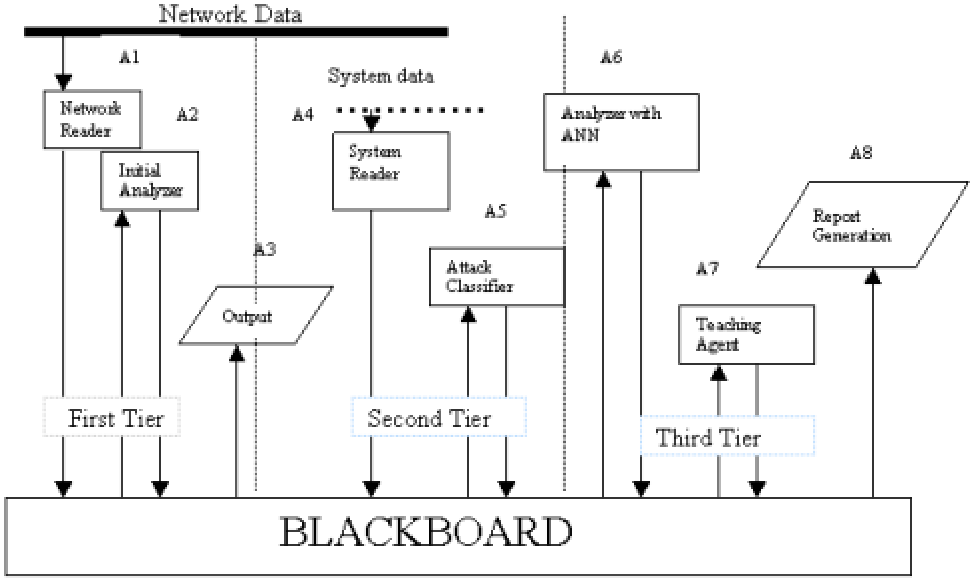


Figure 2.1: Learning Intrusion Detection System Architecture

to remain current with the evolving threat posed by network attackers. LIDS have a blackboard-based autonomous agent architecture that is designed in a multi-tier format (Figure 2.1). There are eight autonomous agents in the system that interacts with the blackboard to perform their actions. Generally, a blackboard system consists of three components; they are the action agents, a blackboard and a control mechanism that will guide these agents [11]. In the proposed system, there is no control/manager agent, but there is a control pattern embedded in each agent that guides their activities.

The first agent (A1) is called the Network Reader. It collects network data with the help of a program called tcpdump. Tcpdump is a network utility tool that records network data in a specific format. The A1 autonomous agent collects network data in groups of 1000 data packets(network activity information) and pastes them on the blackboard. The second agent (A2) is called the Initial Analyzer. It continuously observes the blackboard and whenever it finds any data that need to be analyzed, as the data posted by the first agent, it performs its action. It consists of a Rule-based classifier analyzer as a PROLOG predicate. This analyzer reports to the blackboard whether the data set is clean or not. If some trace of probable attack is found in the data set, it also suggests the type of attack. The third autonomous agent (A3) is the output agent. It helps in displaying any early alert to the security auditors or the administrator of the system, like the information posted by the analyzer agent (A2). The fourth autonomous agent (A4) is the system reader. This agent gathers system specific information of the protected system and posts it on the blackboard. These system data are very helpful in detecting the extent of damage caused by any attack. The type of information gathered includes Available Network Bandwidth, CPU Usage, Network Packets/second, Memory usage, Number of connections, Connection attempts, Protocol, Source address to destination ports ratio (variety of ports accessed) and Packet length.

There are many sub-class attacks that fall under one kind of attack. For example, a denial-of-service attack can be separated into Ping Flood Attack, a UDP Packet Storm Attack, an FTP Brute Force Attack and so on. The fifth agent (A5) is the attack classifier that identifies different sub-classes of intrusions present in the network data. This agent sends the system information from the blackboard to a micro genetic algorithm based classifier that uses the multiple-fault diagnosis concept to perform the above function and posts its result back to the blackboard. The result states which kind of attack is present and what is its probability of presence in the dataset. The ultimate purpose of an Intrusion Detection System is to identify the affected network data with some degree of confidence. This is achieved by the next autonomous agent (A6) or the main analyzer. This agent consists of a set of different kinds of Artificial Neural Networks (ANN). It looks for the different kinds of intrusions present in the dataset from the information posted on the blackboard and decides which ANN is suitable for its analysis. If no attack is present in the data set, it flags the result.

Another objective of LIDS is to learn about new attacks while actively engaged in the detection process. This is achieved by the seventh agent (A7) or the teaching agent. The initial analyzer (A2) is powered by a rule-based classifier system. This type of classifier system has a rule set in it, which is in the form of facts. As mentioned above, the analyzing agent (A2) audits the network data recorded by the network reading agent (A1) and reports whether the data has intrusion in it or it is clean data. If the analyzer (A2) finds a new network pattern and reports an intrusion alert, and in later process, it is found that the network-data is clean, the teaching agent (A7) will update the rule-set of analyzer (A2). Therefore, whenever an initial analysis has resulted in a false alarm or whenever a new type of intrusion is detected that has no supporting rules in the rule-set, we update the rules with the help of the teaching agent.

The final autonomous agent is the Report Generation agent. It generates reports for the system administrator based on the information posted on the blackboard. As mentioned earlier, one of the components of the blackboard architecture is the control mechanism. Since the proposed architecture is autonomous-agent based, there is no agent manager and hence there is a problem in implementing autonomous behavior with sequential processors. The architecture proposed here has a control pattern embedded in the agents. This pattern allows the last agent to look at the blackboard first and the first agent last in order to ensure that each agent gets a chance at least once to look at the blackboard in one process cycle.

The proposed architecture is capable of handling most of the problems faced by the present approaches. It has an online learning mechanism that updates the rules of the analyzer and uses Artificial Neural Networks to analyze the data. These features handle the problems faced by [3] where the rule updating and lack of flexibility in the rule-to-audit record representation was a problem. In most of the other approaches using autonomous agents such as [5] [10], there lacks a common data pool. A common data pool like our blackboard is very important when dealing with various analyzing techniques. Detailed information about intrusions is very important for security officials. A common data pool will also help in storing audited data for future reference. Moreover, this work is unique as this architecture represents a common platform for all the different analysis techniques. Adding a data mining technique or a genetic algorithm as an analyzer to the system will be very easy to implement.

We have completed developing Network Reader Agent (A1), Initial Analyzer Agent (A2), Attack Classifier Agent (A5) and ANN Analyzer Agent (A6). We are using readily available DSSTools to create the blackboard environment. The Attack Classifier as discussed above is a Micro Genetic Algorithm (GA). This GA along with the ANN is in the form of dlls written in C++. They are called by the main agents that are written in PROLOG.

2.5 Conclusion and Future Direction

In designing the proposed architecture we have tried to use the best AI techniques for the different knowledge-based problems. This hybrid approach should fulfill

the deficiencies of other systems and its learning capability can also make it more efficient in a dynamic network environment. The flexibility of the blackboard architecture will allow us to add more features in the future.
All of these agents are to be written in PROLOG. We will be using DSS tools [12] for developing the blackboard system. Some of these agents perform analysis of the network data with the help of Artificial Neural Networks (ANN), written as dlls in C++. We have already developed the C++ dll for the analyzers. The only part remaining is to develop the remaining agents and to integrate all of them in the system. The DSS tool that will form the blackboard environment is readily available.

2.6 References

[1] Lane, T and Brodley C E (1999), Temporal Sequence Learning and Data Reduction for Anomaly Detection, In ACM Transaction on Information and System Security, Vol. 2, No. 3, August 1999.

[2] Lee, W, Stolfo S and Mok K (2000), Adaptive Intrusion Detection: A Data Mining Approach, In Artificial Intelligence Review, Kluwer Academic Publishers, 14(6): 533 - 567, December 2000.

[3] Denning D E (1987), An Intrusion-Detection Model, In IEEE Transaction on Software Engineering, Vol. Se-13, No. 2, February 1987, 222-232.

[4] Asaka, M., Taguchi, A., and Goto, S. (1999). The implementation of IDA: An intrusion detection agent system. In Proceedings of the 11th FIRST Conference, June 1999.

[5] Balasubramaniyan, J, Fernandez J O, Isacoff D, Spafford E. and Zamboni D (1998), An Architecture for Intrusion Detection Using Autonomous Agents, In COAST Technical Report 98/5, Purdue University, June 1998.

[6] Carver, C. A., Hill, J. M. D., Sudru, J. R., and Pooch, U. W. (2000). A Methodology for Using Intelligent Agents to Provide Automated Intrusion Response. In IEEE Systems, Man and Cybernetics Information Assurance and Security Workshop, West Point, N.Y., June 2000.

[7] Asaka, M., Okazawa, S., Taguchi, A., and Goto, S. (1999). A Method of Tracing Intruders by Use of Mobile Agents. In INET' 99, June 1999.

[8] Cannady, J. (1998), Artificial Neural Network of Misuse Detection, in the Proceedings of the 1998 National Information Systems Security Conference (NISSC'98) October 5-8 1998. Arlington, VA.

[9] Ludovic, ME (1998), GASSATA, a Genetic Algorithm as an Alternative Tool for Security Trails Analysis, in the Proceedings of First International Symposium of Recent Advances in Intrusion Detection, 1998.

[10] Dasgupta D., Gonzalez F., K. Yallapu, Gomez, J., Yarramsettii, R., Dunlap, G. and Greveas, M. (2002), CIDS: An Agent-based Intrusion Detection System, In CS Technical Report No. CS-02-001., Feb, 2002.

[11] Penny, Ni. H. (1989). BlackBoard Systems. In Handbook of Artificial Intelligence, Vol IV, edited by Avron Barr, Paul R. Cohen and Edward A. Feigenbaum. Reading, MA: Addison-Wesley Publishing Company Inc.

[12] Zhu, G. (1995). DSSTOOLS : A toolkit for development of Decision Support Systems in PROLOG. M.S. thesis, AI Center, University of Georgia.

Chapter 3

A Blackboard-Based Learning Intrusion Detection System: A New Approach[1]

[1] Dass M., J. Cannady and W.D.Potter to appear in the proceedings of the 16th International Conference on Industrial & Engineering Applications of Artificial Intelligence & Expert Systems, Loughborough, UK, June 23-26, 2003

3.1 ABSTRACT

Intrusion Detection is one of the crucial real-time problems in the field of computer networking. With the changing technology and the exponential growth of Internet traffic, it is becoming difficult for any existing intrusion detection system to offer a reliable service. From earlier research, we have found that there exists a behavioral pattern in the attacks that can be learned. That is why an Artificial Neural Network is so successful in detecting network intrusions. Still, this approach is not effective in a dynamic environment where changes take place frequently. This paper proposes a blackboard-based Learning Intrusion Detection System, which is controlled by autonomous agents and has an online learning capability. This feature enables the system to adapt itself with the changing environment and to perform better than present systems.

Keywords: Network Security, Intrusion Detection, Blackboard Architecture, Autonomous Agents, Artificial Neural Network.

3.2 Introduction

Intrusion Detection has been a hard problem from the early days of computer networking. This problem has become more prominent with the rapid increase in vulnerable Internet applications and automated attack scripts. Every year, business and industry loose a huge amount of revenue due to data manipulation caused by computer network intruders. According to the 2001, CSI/FBI Computer Crime and Security Survey, more than $35 million was lost per company due to unauthorized net access costing them an average of $357,160 per incident [1]. As a result, there has been an increasing requirement to effectively protect crucial business information with a reliable, robust and flexible intrusion detection system. There are many commercially available Intrusion Detection Systems (IDS), but unfortunately they are costly and of limited reliability. These systems are rule based and are unable to maintain their performance with the increasing complexity of the Internet. This has led to worldwide research interest In effective intrusion detection techniques with artificial intelligence [2], data mining [3] and statistical techniques [4]. From the works of Denning [4], we have found that there exists a behavioral pattern in attacks. Hence, Pattern Matching approaches [4] and Artificial Neural Networks [2, 7] have been very effective in detecting intrusions.

Using a blackboard architecture in an Intrusion Detection System is not a new approach. Works of Dasgupta [5] reveal an optimistic attempt to detecting intrusions with agents in a blackboard architecture. These agents exchange information among themselves through a discrete data path. There are also some approaches with autonomous agents as in [6] where the agents interact among themselves and exchange information to detect intrusions. There has not been any attempt in building a learning system. Researchers argue that Artificial Neural Networks

(ANN) are learning systems, but they are very domain specific and cannot perform well in a dynamic environment unless they are trained dynamically. Works in [7] show effective approaches with ANN to intrusion detection. There have also been some approaches with Genetic Algorithms [8], but they failed to show the same level of performance as Artificial Neural Networks did.

This paper presents the description of a blackboard based three-tier autonomous agent architecture of a Learning Intrusion Detection System (LIDS), which is still under development. This system has a learning capability and can adapt to any computer network environment. It uses the classifying power of the ANN and the Genetic Algorithm to detect intrusions.

3.3 Intrusion Detection and Learning

Intrusion Detection can be defined as the identification of attempted or ongoing attacks on a computer system or network. Intrusion Detection can be differentiated into two categories [9]: anomaly detection and misuse detection. The former refers to the detection of abnormal behavior in the use of network services and computing resources. Misuse detection, on the other hand relies on the identification of well-defined attacks or vulnerabilities in network or computer software. Unfortunately, intrusions rarely follow an expected pattern. The increasing availability of attack tools, the rise in the number of exploitable system vulnerabilities, and the growing creativity of attackers mean that traditional intrusion detection approaches are inadequate.

Intrusion Detection Systems are also classified according to the network system area they audit. They can be Host Based or Network Based. A Host Based Intrusion Detection System can be defined as a security system that is capable of detecting inside abuses in a computer network. A Network Based Intrusion Detection System is capable of identifying abusive uses or attempts of unauthorized usage of the computer network from outside the system. Our work most closely resembles a Network Based Intrusion Detection System that uses computational intelligence techniques to dynamically detect intrusions.

Prior approaches to this problem used some form of rule-based analysis [10]. Rule-Based analysis relies on predefined rule-sets that are provided by an administrator, automatically created by the system, or both. Expert Systems are the most common form of rule-based intrusion detection approaches. Rule-based systems suffer from the inability to detect attack scenarios that may occur over an extended period of time. They also lack flexibility in the rule-to-audit record representation. Slight variations in the attack sequence may reduce the effectiveness of the system. There have been some optimistic attempts with genetic algorithms [8], data mining [3] and pattern recognition techniques [4] to develop a high performance IDS, but these techniques are still undergoing research.

Prior research [7] has demonstrated the ability to convert attack patterns into data vectors. This will allow ANN or any other pattern recognition technique to perform well. But, these approaches rely upon the training data and ultimately will fail because of the changing nature of the computer network. Our proposed architecture has another learning layer above these analysis techniques, which will regularly maintain and update the training data set of the ANN or any other machine learning technique.

Some examples of the more common types of malicious attacks in the network are:

- Denial-of-service Attack (DoS) - This is a particularly serious form of attack that has resulted in damage worth millions of dollars over the past few years. While a significant problem, DoS attacks are usually quite simple. They typically involve an attacker disabling or rendering inaccessible a network-based information resource.
- Guessing rlogin Attack - Here the intruder tries to guess the password that protects the computer network in order to gain access to it.
- Scanning Attacks - The intruder goes about scanning different ports of the victim's system to find some vulnerable points from where they can launch other attacks.

3.4 Blackboard and Proposed Architecture

The blackboard architecture is considered one of the most general and flexible knowledge system architectures for building decision-based applications. It is highly preferred over other alternatives due to its modularity, dynamic control, generality, concurrency, high design efficiency, robustness and ability in dealing with multiple knowledge sources. As a result, the blackboard-based architecture is considered a good solution in developing our proposed Intrusion Detection System. The proposed architecture will also include the use of Autonomous Agents that are software agents which perform certain security monitoring functions at a host. The agents are independently running entities whose performance is not affected by any other agents. These kinds of agents are very useful in network security because they run continuously, can resist subversion and have minimal overhead. They are also configurable, easily adaptable, scalable, dynamically reconfigurable and degrade gracefully. The proposed architecture consists of autonomous agents that are integrated in a blackboard-based architecture and placed in a tier form.

The use of blackboard techniques and autonomous agents [5] in detecting network intrusions is not a new concept. In [5] , Dasgupta described how a blackboard-based agent architecture helps in detecting intrusions. He developed a distributed blackboard architecture that is embedded among the agents. A manager agent controls the monitoring, decision and action agents. The unidirectional flow of information in the system has a major impact on the flexibility of the system. Weiss, in his work [11] suggested many approaches that can be utilized, but it all has the problem of adaptation. Our proposed system that is designed in a multi-tier format removes this inefficiency.

The proposed Learning Intrusion Detection System is shown below (Figure 3.1).

The system agents are divided into three tiers. Though they are autonomous agents, this tier bifurcation is done according to their contribution to the system. The first tier consists of all the autonomous agents required for the initial alert feature. A1 is the Network Reader. It collects network data with the help of a program called tcpdump. Tcpdump is a network utility tool that records network data in a specific format. The A1 autonomous agent collects network data in groups of 1000 data packets (network activity information) and pastes them on the blackboard. The second agent A2 is the initial analyzer. It calls a Rule-based classifier system that is written as a dll in C++. This classifier system analyzes the network data

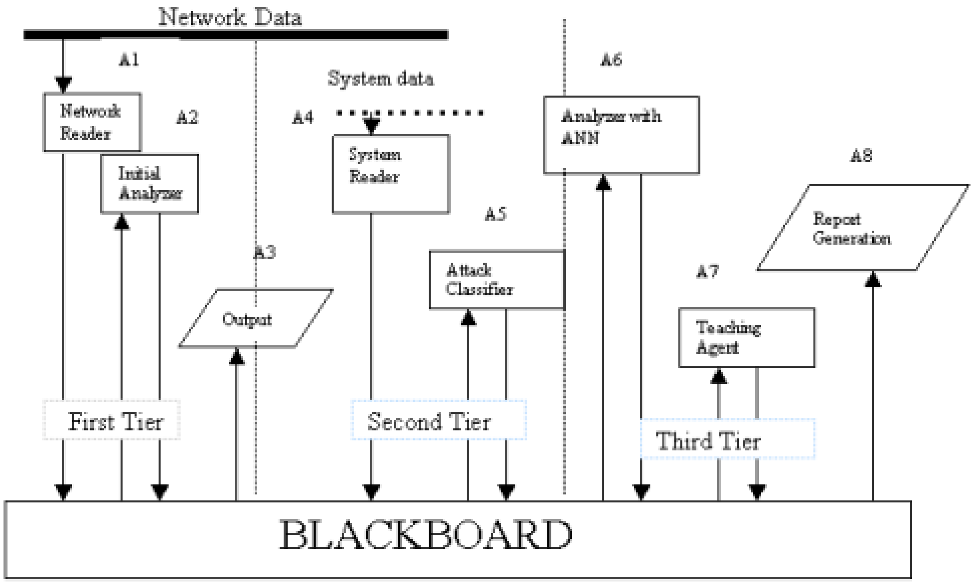


Figure 3.1: Learning Intrusion Detection System Architecture

and checks whether there is any anomaly in the network data or not. The third agent is the display agent, A3, or the output agent. This is used to report the initial analysis to the user.

The second tier consists of all the agents that analyze system specific information. It consists of the System Reader (A4) that gathers system specific information on the protected system and posts it on the blackboard. These system data are very helpful in detecting the extent of damage caused by any attack. The type of information gathered includes Available Network Bandwidth, CPU Usage, Network Packets/second, Memory usage, Number of connections, Connection attempts, Protocol, Source address to destination ports ratio (variety of ports accessed) and Packet length. There are many sub-class attacks that fall under one kind of attack.

For example, a denial-of-service attack can be separated into a Ping Flood Attack, a UDP Packet Storm Attack, an FTP Brute Force Attack and so on. The fifth agent (A5) is the attack classifier that identifies different sub-classes of intrusions present in the network data. This agent sends the system information from the blackboard to a micro genetic algorithm based classifier that uses the multiple-fault diagnosis concept to perform the above function and posts its result back to the blackboard. The result states which kind of attack is present and what is its probability of presence in the dataset.

The third tier contains the autonomous agents that give full details of the attacks. The detail contains information about the data packets that are affected. A6 analyzes the information about the attacks and decides which type of ANN will be useful in further analysis of the data. The ANNs are in the form of dlls written in C++. If the analysis finds no attack in the dataset, the agent flags the dataset as it is a false positive alarm from the initial analyzing agent. The next agent is the teaching agent. This agent updates the rule set of A2 or the Initial Analyzer so that the A2 agent is capable of adapting to the changing environment. Ultimately, the Report Generation agent displays a complete report of the analysis to the user.

In a blackboard-based architecture, there is a requirement for an agent manager to control the activity of all the participating agents, but the notion of autonomous demands all the agents to work independently. Hence, we included a control pattern in the architecture of the agent. This pattern allows the last agent to look at the blackboard first and the first agent last in order to ensure that each agent gets a chance at least once to look at the blackboard in one process cycle. We have completed developing the Network Reader Agent (A1), the Initial Analyzer Agent (A2), the Attack Classifier Agent (A5) and the ANN Analyzer Agent (A6). We are using readily available DSSTools to create the blackboard environment. The Attack Classifier as discussed above is a Micro Genetic Algorithm (GA). This GA along with the ANN is in the form of dlls written in C++.

3.5 Future work

In designing the proposed architecture we have tried to use the best AI techniques for the different knowledge-based problems. This hybrid approach should fulfill the deficiencies of other systems and the learning capability can also make it more efficient in a dynamic network environment. This paper tries to present a hybrid system using the agents-based architecture. While developing the system, vital information about intrusion detection was uncovered which needs further research. We have successfully developed an attack classifier with the Genetic Algorithm, which is capable of determining sub-classifications of attacks. This approach modelled after"multiple-fault diagnosis"is new and can be the subject of further research.

Though this system promises greater flexibility than most of the present Intrusion Detection Systems in a changing environment, we need to test it after the completion of its development. This work also helps in providing ideas in developing learning Intrusion Detection Systems.

3.6 References

[1] Raytheom (2002), The Insider Threat, White Paper of SilentRunner, Inc. published in 01/09/02.

[2] Lane, T and Brodley C E (1999), Temporal Sequence Learning and Data Reduction for Anomaly Detection, In ACM Transaction on Information and System Security, Vol. 2, No. 3, August 1999.

[3] Lee, W, Stolfo S and Mok K (2000), Adaptive Intrusion Detection: A Data Mining Approach, In Artificial Intelligence Review, Kluwer Academic Publishers, 14(6): 533 - 567, December 2000.

[4] Denning D E (1987), An Intrusion-Detection Model, In IEEE Transaction on Software Engineering, Vol. Se-13, No. 2, February 1987, 222-232.

[5] Dasgupta D., Gonzalez F., K. Yallapu, Gomez, J., Yarramsettii, R., Dunlap, G. and Greveas, M. (2002), CIDS: An Agent-based Intrusion Detection System, In CS Technical Report No. CS-02-001., Feb, 2002.

[6] Balasubramaniyan, J, Fernandez J O, Isacoff D, Spafford E. and Zamboni D (1998), An Architecture for Intrusion Detection Using Autonomous Agents, In COAST Technical Report 98/5, Purdue University, June 1998.

[7] Cannady, J. (1998), Artificial Neural Network of Misuse Detection, in the Proceedings of the 1998 National Information Systems Security Conference (NISSC'98) October 5-8 1998. Arlington, VA

[8] Ludovic, ME (1998), GASSATA, a Genetic Algorithm as an Alternative Tool for Security Trails Analysis, in the Proceedings of First International Symposium of Recent Advances in Intrusion Detection, 1998.

[9] Anderson, D., Frivold, T. and Valdes (1995), A Next-generation Intrusion Detection Expert System (NIDES): A Summary, SRI International Technical Report SRI-CSL-95-07, May, 1995.

[10] Sebring, M., Shellhouse, E., Hanna, M. and Whitehurst, R. (1988), Expert Systems in Intrusion Detection: A Case Study, In Proceedings of the 11th National Computer Security Conference.1988.

[11] Weiss S. and Kulikowski C. (1991), Computer System That Learn, Morgan Kauffman, California, 1991.

Chapter 4

A GA based Intrusion Sub-classifier Filter[1]

[1] Dass M., J. Cannady and W.D.Potter to appear in the digital proceedings of the 41st ACM Southeast Conference, Savannah, March 7 - 8, 2003.

4.1 Abstract

With the development of new technologies and the expansion of networked computer systems, sensitive data are under constant threat of attack from hackers. Intrusion is a very common threat to a network and with the increasing creativity of attackers, the development of an effective intrusion detection system (IDS) is becoming a greater challenge. A prior approach to this problem has been to develop a rule-based system, but it has proven to be unsatisfactory owing to its high maintenance cost. This has resulted in the development of Next Generation Intrusion Detection Systems, which use other Artificial Intelligence techniques such as Artificial Neural Networks, Logic Trees, Genetic Algorithms, Fuzzy Logic and Data Mining to detect intrusions. It has been shown that knowledge of the type of attack reduces the computational overhead of the IDS. Among other methods, the evolutionary search techniques such as genetic algorithms have the capacity to distinguish anomalous patterns in network traffic. This paper describes a network filter using a distributed-type micro-genetic algorithm modeled after the multiple-fault diagnosis approach to detect sub-classes of intrusion attacks. The results from our preliminary analysis indicate successful detection of sub-classes of Denial-of-Service attacks. This filter is a primary component in the intelligent intrusion detection system we are developing.

Categories and Subject Descriptors: Network Security.

General Terms: Network Security, Algorithms

Keywords: Intrusion Detection, Genetic Algorithm.

4.2 Introduction

Unauthorized attempts to gain access to sensitive information or to disrupt normal network functions are increasing day by day. The problem of detecting intrusions is becoming difficult to solve with the development of new technologies and with the exponential growth of the Internet. The goal of a reliable Intrusion Detection System (IDS) is to automatically detect violations of the security policy for a computer site by an outsider. Existing IDS like RIPPER, a rule-based expert system developed in late 1980's, which has the capability to detect anomalies in the computer network, have proven to be unsatisfactory owing to its high maintenance cost. This has resulted in the development of Next Generation Intrusion Detection Systems, which use other Artificial Intelligence techniques such as Artificial Neural Networks, Logic Trees, Genetic Algorithms, Fuzzy Logic and Data Mining to detect intrusions. From previous work in intrusion detection, it has been shown that knowledge of the type of attack reduces the computational overhead of the IDS. This paper describes a network-filter powered by an evolutionary search technique to detect sub-classes of denial-of-service attacks in the network system

4.3 Intrusion Detection Systems

4.3.1 Definition

IDS are defined as security systems that can identify attempted or ongoing attacks on a computer system or network. Developing reliable and efficient IDS that will timely and accurately detect intrusions is challenging. However, it is becoming a necessary security tool in industry. Every year, businesses lose a huge amount of revenue due to improper data manipulation caused by computer network intruders. There are two general types of intrusion detection problem, which an intrusion detection system tries to solve- anomaly detection and misuse detection [1][7].

Misuse detection involves the comparison of a user's activity with the known behaviors of attackers attempting to penetrate a system [8][9]. On the other hand, anomaly detection identifies activities that vary from the established patterns for users, or group of users [2]. Anomaly detection typically involves the creation of knowledge bases that contain the profiles of the monitored activities. In the past, researchers have used computational intelligence techniques like neural networks [2][3][4] to solve the misuse detection problem, but an effective working model is yet to emerge. Some have also used genetic algorithms [5][6] to detect intrusions. Hybrid systems, which combine one or more of these approaches may prove to be a promising solution to the problem of detecting intrusions. In the past, there have been proposals [5] for this kind of security system.

4.3.2 Complexity of the problem

Researchers have attempted a variety of possible techniques to solve this ever-growing problem. Most of these approaches are still under development. One of the most successful approaches is RIPPER, a rule-based expert system developed in late 1980's. It has the capability to detect anomalies on a computer network, but the rules must be updated regularly due to evolution of new attacks. Current systems like SNORT [21] work on the same principle. With the development of systems like NIDES (Next-generation Intrusion Detection Expert System) [12], the focus of security researchers has moved towards Artificial Intelligence [10], Data Mining [11] and Statistical Techniques [1], but little or no work has been done on defining the complexity of the network security problem.

Intrusions in a computer network do much harm to the computer system. A Denial-of-Service (DoS) attack drains the resources of a computer system. Metaphorically, we can compare a computer system with human beings. When we are suffering from diseases, we are unable to perform well. We go to a physician who diagnoses the disease and provides medicine to cure it. A computer system may be subjected to one intrusion attack or to multiple attacks. We are trying to analyze the problem of intrusion detection by using a multiple fault diagnosis approach. In this approach, we are trying to compare intrusions with diseases and different system anomalies with symptoms of diseases. After our work, we can successfully assume that if we can develop an IDS that can diagnose multiple fault occurrences in a computer system, it can reliably detect intrusion attacks. In [23], Bylander showed that the complexity of the multiple fault diagnosis problem is NP-Complete. Intrusion Detection is an intractable problem. Moreover, previous research supports the claim that the computational overhead of an IDS is directly proportional to the number of network parameters considered in the analysis. These evidently support our assumption.

4.3.3 Intrusion Filter

An intrusion filter is a very important component of an IDS. From [20], we see that it reduces the computational overhead of an intrusion detection system. If we are able to learn what different kinds of intrusion attacks are going on, we will be more efficient in detecting the attack indicators from the network signatures. For example, if we know that the system is suffering from intrusion type A and B, then we will only apply the rules of detecting intrusion A and intrusion B or a combination of both in our IDS.

Intrusion Filters are desirable because they have the following characteristics.

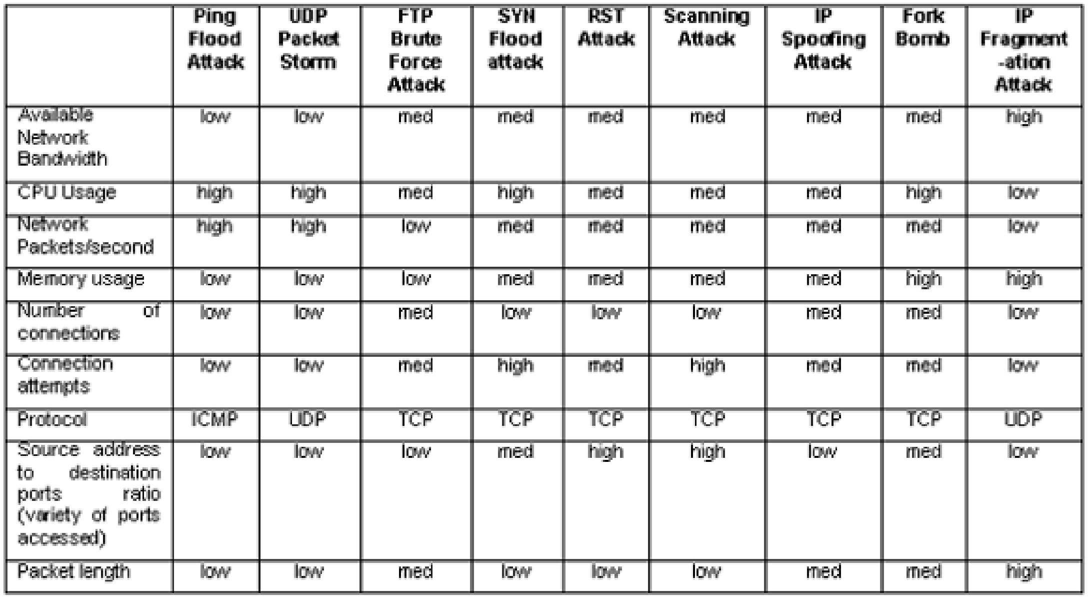
- Analysis is based on system resources - Filter analysis is dependent only on the system related data.
- Analysis is independent of the network data - The filter is completely independent from any network or network dependent data.
- It should be time efficient - The filter is fast and can be easily added to an IDS.

4.4 Genetic Algorithm

A genetic algorithm is a powerful heuristic search scheme based on the model of Darwinian evolution. Since this idea was first presented in 1975 [14], genetic algorithms have been used successfully in classification [13], and search and optimization problems. It is guided by the concept of ”survival of the fittest”. The basic idea behind a genetic algorithm is to generate solutions that converge to a global maximum (i.e. the best solution in the search space) regardless of the ”terrain” of the search space [15]. There are three basic operations, which guide this process. They are selection, crossover and mutation. In our problem, we will be using a micro-genetic algorithm where the number of individuals generated is comparatively low, but we will not be using the re-initialization process.

4.5 The Proposed Filter

Owing to the filter requirements and the problem complexity of intrusion detection, we designed the network intrusion filter based on a powerful heuristic search scheme: the Genetic Algorithm (GA). De Jong in [22] proved that the GA is useful in solving

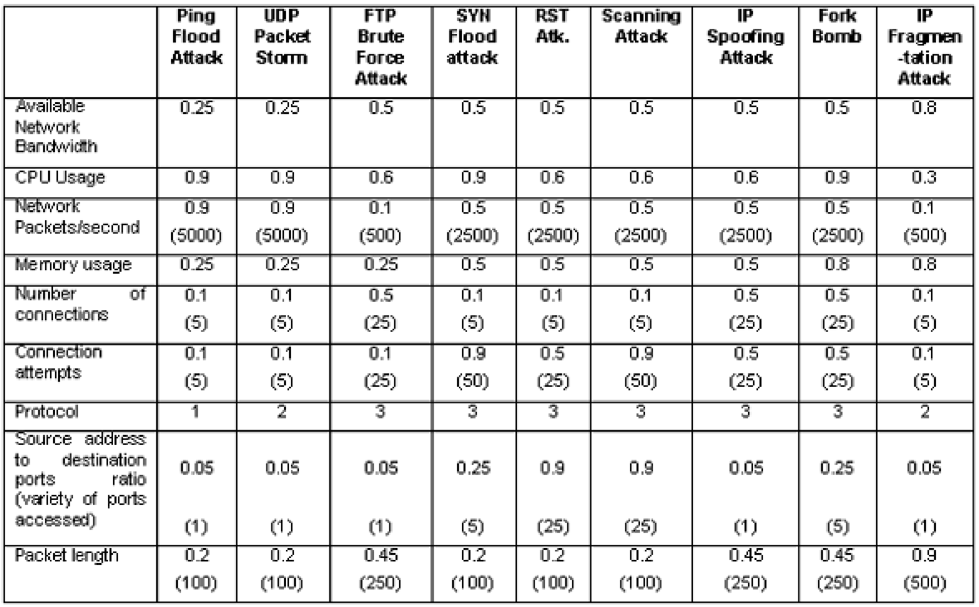


	Ping Flood Attack	UDP Packet Storm	FTP Brute Force Attack	SYN Flood attack	RST Attack	Scanning Attack	IP Spoofing Attack	Fork Bomb	IP Fragment -ation Attack
Available Network Bandwidth	low	low	med	med	med	med	med	med	high
CPU Usage	high	high	med	high	med	med	med	high	low
Network Packets/second	high	high	low	med	med	med	med	med	low
Memory usage	low	low	low	med	med	med	med	high	high
Number of connections	low	low	med	low	low	low	med	med	low
Connection attempts	low	low	med	high	med	high	med	med	low
Protocol	ICMP	UDP	TCP	TCP	TCP	TCP	TCP	TCP	UDP
Source address to destination ports ratio (variety of ports accessed)	low	low	low	med	high	high	low	med	low
Packet length	low	low	med	low	low	low	med	med	high

Figure 4.1: Causal Matrix for the filter.

NP-Complete problems. In [22], he argued that the Boolean Satisfiability problem (SAT) is a GA effective canonical problem and other NP complete problems with poor GA representations can be solved efficiently by mapping them first onto SAT problems. The GA has been successfully used in multiple fault diagnosis [15].

We use a micro-GA in order to reduce the time overhead normally associated with a GA. We have viewed the solution to our problem as abductive inference. This type of problem solving strategy has been used in [15][16]. These approaches to diagnosis follow the “reasoning from first principles” paradigm where a description of some physical system’s structure and behavior is maintained and compared to abnormal behavior [15]. The most reasonable solution to our problem will be the diagnosis or diagnoses that best explain the observed symptoms. Therefore, in



	Ping Flood Attack	UDP Packet Storm	FTP Brute Force Attack	SYN Flood attack	RST Atk.	Scanning Attack	IP Spoofing Attack	Fork Bomb	IP Fragmen -tation Attack
Available Network Bandwidth	0.25	0.25	0.5	0.5	0.5	0.5	0.5	0.5	0.8
CPU Usage	0.9	0.9	0.6	0.9	0.6	0.6	0.6	0.9	0.3
Network Packets/second	0.9 (5000)	0.9 (5000)	0.1 (500)	0.5 (2500)	0.5 (2500)	0.5 (2500)	0.5 (2500)	0.5 (2500)	0.1 (500)
Memory usage	0.25	0.25	0.25	0.5	0.5	0.5	0.5	0.8	0.8
Number of connections	0.1 (5)	0.1 (5)	0.5 (25)	0.1 (5)	0.1 (5)	0.1 (5)	0.5 (25)	0.5 (25)	0.1 (5)
Connection attempts	0.1 (5)	0.1 (5)	0.1 (25)	0.9 (50)	0.5 (25)	0.9 (50)	0.5 (25)	0.5 (25)	0.1 (5)
Protocol	1	2	3	3	3	3	3	3	2
Source address to destination ports ratio (variety of ports accessed)	0.05 (1)	0.05 (1)	0.05 (1)	0.25 (5)	0.9 (25)	0.9 (25)	0.05 (1)	0.25 (5)	0.05 (1)
Packet length	0.2 (100)	0.2 (100)	0.45 (250)	0.2 (100)	0.2 (100)	0.2 (100)	0.45 (250)	0.45 (250)	0.9 (500)

Figure 4.2: Causal Matrix used in the experiment.

order to compare the generated solution with abnormal behavior of the network, we have to have a causal matrix, which has values for each symptom corresponding to different sub-classes of Denial-of-Service attacks. The causal matrix is shown in 4.1. The values in the causal matrix are the system specific information resulting from different attacks. 4.2 shows the numerical threshold values of the system data. The data used for the experiment are from a simulated attack environment.

4.6 Denial-Of-Service Attacks

Denial-of-Service (DoS) attacks have caused extensive damage to computer networks recently. A SYN Flood attack, a type of DoS attack victimized CNN, eBay, Yahoo and Amazon on Feb.,5-11, 2000. In a DoS attack, the attackers generate an unusually large volume of requests, overwhelming the server. In these cases, the legitimate users were denied access. This type of attack can last from a few minutes to several days. Some examples of DoS attacks are the Ping Flood Attack, UDP Packet Storm, FTP Brute Force Attack, SYN Flood attack, RST Attack, Scanning Attack, IP Spoofing Attack, Fork Bombs and IP Fragmentation Attack. In this paper, we use a GA to classify these sub-classes of DoS attacks.

The following is a short description of each of these attacks.

- Ping Flood Attack - In this kind of attack, the attacker sends hundreds of thousands of ping messages (”are you there”) to the server.
- UDP Packet Storm Attack- When a connection is established between two UDP services, each of which produces output; these two services can produce a very high number of packets that can lead to a UDP Packet Storm Attack.
- FTP Brute Force Attack- In this kind of attack, the intruder tries to break the cipher by trying every possible key.
- SYN Flood Attack- It is a DoS kind of attack where the attacker creates a random source address for each packet it sends and the SYN flag set in each of these packets requests a new connection to open from the spoofed IP address. Therefore, the server responds to the spoofed IP address and then waits for confirmation that never arrives.
- RST Attack- Also known as a Reset Attack. In this kind of attack, the attacker sends RST packets with correct sequence numbers and keeps the connection open. Quite similar to a SYN Flood Attack.

- Scanning Attack- Here the attackers try to scan a network to see which patterns are on the network and in what port.
- IP Spoofing Attack- Here an outside attacker transmits packets to the server, which are supposedly coming from another inside node.
- Fork Bombs-In this type, uncontrolled creation of children by forking the main process floods the server.
- IP Fragmentation Attack-In this kind of attack, the intruder smuggles packets into the server. Here the attacker hides the TCP header in an offset IP fragment and just neglects to send the first (zero offset) packet. The server will still reassemble the attacker's packet placing the fragment with the lowest-offset at the front.

4.7 Experiment Procedure

In order to detect these DoS attacks using a GA-based filter, we need to have a causal matrix. The value for the "symptoms" is obtained from the computer system auditing programs. For the sake of our proposed filter, we have created a pseudo environment for some given values to this matrix as shown in figure 2.

In order to detect an intrusion in the network, the values of all these "symptoms" or computer system features from different filters are fed into the algorithm. The GA then generates individuals that explain this injected data. Though the problem solution is designed according to the "Multiple Fault Diagnosis" approach, the fitness function used here is along the lines of a distributed genetic algorithm (REGAL) [6] [17].If we observe the causal matrix carefully, we will find that these attacks are protocol specific. Therefore, if the input data to the GA are using the UDP protocol, the GA will consider only an IP Fragmentation Attack and UDP Packet Storm attack. The fitness function contains some fuzzy rules for classifying the input data. The fuzzy rule approach is considered here, because sometimes the input data are not an attack but an indication of one. The system must be capable of identifying these kinds of indications.

The fuzzy rules are as follows:

```
If ( the attack is present )
If ( the attack specific protocol = protocol of the entered value )
{
  If ( the causal value of the symptom >= HIGH ) . . . . . . * 0.8
  If ( the input value of causal value)
  Add 1 to the fitness . . .
If ( the causal value of the symptom < = LOW) . . . . . . *0.1
If ( the input value is < 115% of causal value)
        Add 1 to the fitness . . .
}
Else
Subtract the ratio of (difference of causal value and input value of the symptom )
                    and 10 from the fitness.
```

This kind of genetic algorithm is widely used in classifier types of problems.

As mentioned earlier, we have used a micro-GA having a very small population size. In our case, the GA is generating only 10 individuals in the population, but we have not used the reinitialization process. In a Micro-GA, the reinitialization process is important as we use a small population size, but since in our case we have a set of rules to select the best individual, it is irrelevant whether we reinitialize our population in each generation or not. Since in a real-time intrusion detection

Table 4.1: GA setup

Size of the population	10
Probability of crossover	0.6
Probability of mutation	0.025
Stable generations of convergence	30
Selection scheme	Two-player tournament selection

system, time is a crucial factor, a micro-GA is used to reduce the time overhead. The individuals are represented using binary values (i.e., 1's and 0's). The length of each individual is nine since we are trying to classify nine sub-classes of denial-of-service attacks at this time (more will be added later). We have used a two-player selection scheme in order to reduce the selection pressure. An elitism sampling mechanism and two-point crossover are used. The mutation process simply flips the bit value of the selected gene of the individual.

4.8 Experimental Results

The genetic algorithm setup for the proposed filter is shown in 4.1. It was initially written in C++ and later on transformed into a C++ dll in LIDS [20].

The results obtained from the filter are as follows:

Available Network Bandwidth:0.5

CPU :0.6

Network Packets/second :0.5 (2500)

Memory :0.5

Number of connections :0.1 (5)

Connection attempts :0.9 (50)

Protocol :3 (TCP)

Source address to destination ports (variety of ports accessed) :0.9 (25)

Packet length :0.2 (100)

This gave a result of **000111000**- That is, the input values define: SYN flood, RST attack and Scanning attack.

The next experiment was done with the following input values.

Available Network Bandwidth :0.5

CPU Usage :0.6

Network Packets/second :0.5 (2500)

Memory usage :0.5

Number of connections :0.1 (5)

Connection attempts :0.5 (25)

Protocol :3 (TCP)

Source address to destination ports (variety of ports accessed):0.05 (1)

Packet length :0.6 (300)

This gave the following result : **001000100** - It means that there is a FTP Brute Force Attack and an IP Spoofing Attack .

The convergence of the GA to a global maximum is shown in figure 4.

4.8.1 Some Discussion about the Results

We find that the GA is converging very fast. Since time is a significant factor in a real-time process and the fitness function is guided by a set of rules, fast convergence is affordable and desirable in our case. The results show that the genetic algorithm has found a good solution and can therefore efficiently work as an excellent filter to detect attacks.

4.9 Further Work

Using the genetic algorithm based filter, we can now proceed to develop the filter for all other kinds of attacks like Network Hopping, Rlogin Attacks and so on. Artificial Intelligence problem solving strategies can be categorized as ”strong” and ”weak.” This filter will allow us to use ”strong” AI strategies as it increases the assumption list. This filter has been implemented as an autonomous agent in the Learning Intrusion Detection System (LIDS) [20]. In LIDS, this filter takes the system information and detects intrusion attacks, if present. The classification accuracy of the filter is very high and it has contributed to the increased efficiency and reliability of LIDS. This work also opens further research study of using Intrusion Filters in an Intrusion Detection System (IDS).

4.10 References

[1] Denning D E (1987), An Intrusion-Detection Model, In IEEE Transaction on Software Engineering, Vol. Se-13, No. 2, February 1987, 222-232.

[2] Cannady, J.,(1998) Artificial Neural Network of Misuse Detection, in the Proceedings of the 1998 National Information Systems Security Conference (NISSC'98) October 5-8 1998. Arlington, VA

[3] Ryan, J., Lin, M., and Miikkulainen, R. (1997). Intrusion Detection with Neural Networks. AI Approaches to Fraud Detection and Risk Management: Papers from the 1997 AAAIWorkshop (Providence, Rhode Island), pp. 72-79.

[4] Debar, H., Becke, M., and Siboni, D. (1992). A Neural Network Component for an Intrusion Detection System. In Proceedings of the IEEE Computer Society Symposium on Research in Security and Privacy.

[5] Ludovic, ME, GASSATA, a Genetic Algorithm as an Alternative Tool for Security Trails Analysis, in the Proceedings of First International Symposium of Recent Advances in Intrusion Detection, 1998.

[6] Neri, F.,(2000) Comparing Local Search with Respect to Genetic Evolution to Detect Intrusions in Computer Networks. In Proceedings of the 2000 Congress on Evolutionary Computation. pp 238–243 California, USA

[7] Helman, P., Liepins, G., and Richards, W. (1992). Foundations of Intrusion Detection. In Proceedings of the Fifth Computer Security Foundations Workshop. pp.114-120 Franconia, 1992.

[8] Kumar, S. and Spafford, E. (1994) A Pattern Matching Model for Misuse Intrusion Detection. In Proceedings of the 17th National Computer Security Conference, pp 11-21. Baltimore MD, October 1994.

[9] Kumar, S. and Spafford, E. (1995) A Software Architecture to Support Misuse Intrusion Detection. Department of Computer Sciences, Purdue University; CSD-TR-95-009.

[10] Sebring, M., Shellhouse, E., Hanna, M. and Whitehurst, R. (1988) Expert Systems in Intrusion Detection: A Case Study. In Proceedings of the 11th National Computer Security Conference. Baltimore MD, October 1988.

[11] Lee, W, Stolfo S and Mok K, Adaptive Intrusion Detection: A Data Mining Approach, In Artificial Intelligence Review, Kluwer Academic Publishers, 14(6): 533 - 567, December 2000.

[12] Anderson, D., Frivold, T. and Valdes, A (May, 1995). Next-generation Intrusion Detection Expert System (NIDES): A Summary. SRI International Technical Report SRI-CSL-95-07

[13] Neri, F. (Nov 1996) Exploring the Power of Genetic Search in Learning Symbolic Classifiers. In IEEE Transaction on Pattern Analysis and Machine Intelligence., Vol 18, No. 11pp. 1135-1141.

[14] Holland, J.H. (1975) Adaptation in Natural and Artificial Systems, University of Michigan Press, Ann Arbor, 1975.

[15] Potter, W.D., Miller, J.A., Tonn, B.E., Gandham, R.V. and Lapena,C.N.,(1992) "Improving The Reliability of Heuristic Multiple Fault Diagnosis Via The Environmental Conditioning Operator," in APPLIED INTELLIGENCE: The International Journal of Artificial Intelligence, Neural Networks, and Complex Problem-Solving Technologies, 1992. Vol. 2, pp. 5-23.

[16] Davis, R.(1984), Diagnostic Reasoning Based on Structure and Behavior. In Artificial Intelligence, Vol. 24, No. 1-3, pp. 347-410, December 1984.

[17] Giordana A. and Neri F.,(1995) Search-intensive concept induction. In Evolutionary Computation, 3(4): pp 375-416.

[18] Cannady, J.,(1998) The application of Artificial Intelligence to Misuse Detection: Initial Results. In Proceedings of the First Recent Advances in Intrusion Detection(RAID) Conference, September 14-15, 1998 Louvain-la-Neuve, Belgium.

[19] Jajodia, S., Liu, P. and McCollum, C.D. (1998) Application-level isolation to cope with malicious database users. In Proceedings of the 14th Annual Computer Security Application Conference, Phoenix, AZ. pp 73-82.

[20] Dass, M., Cannady, J. and Potter, D. (2003) LIDS: Learning Intrusion Detection System. In Proceedings of 16th International FLAIRS Conference, 2003, May 12-14, 2003 St. Augustine, Florida. pp 12-16.

[21] Roesch. M.,(1999) Snort - lightweight intrusion detection for networks. In Proceedings of the 13th System Administration Conference, Seatlle, WA, November 7-12, 1999. USENIX 1999, ISBN 1-880446-25-1Lisa . pp. 229-238.

[22] DeJong, K. and Spears, W.M.(1989) Using genetic algorithms to solve NP-complete problems. In J.D. Schaffer, editor, Proceedings of the Third International Conference on Genetic Algorithms, July 1998, Madison, USA. pages 124–132.

[23] Bylander, T., Allemang, D., Tanner, M. and Josephson, J.(1991) The Computational Complexity of Abduction. In Artificial Intelligence , 49(1-3):25-60.

Chapter 5

The design of a learning agent in an Multi-agent Intrusion Detection System[1]

[1] Dass M., J. Cannady and W.D.Potter to be submitted to Network Security conference

5.1 ABSTRACT

With the increase in the attacks against computer networks and unimpressive performance of existing intrusion detection approaches, there is a need to develop a more intelligent and more reliable network security tool. As a result, we have developed a Learning Intrusion Detection System (LIDS) that is able to adapt itself to changes in the network environment and provide greater accuracy than present systems. This paper discusses the challenges in developing an adaptive intrusion detection system. A detailed discussion is provided on the design of the learning mechanism of the system and its cutting edge performance as compared to present systems.

Keywords: Network Security, Intrusion Detection, Machine Learning

5.2 Introduction

With the increase in the reported vulnerabilities of different operating systems and neglected computer security policies, there has been an exponential increase in the attacks against computer networks. According to the 2002, CSI/FBI Computer Crime and Security Survey, the total revenue loss in industry due to computer network intrusion was calculated as $455,848,000, up from $35 million reported in 2001 [1]. This information implies that there is an urgent need for a reliable, robust and flexible intrusion detection system (IDS) that will effectively protect crucial business information. There are many commercially available IDS. Most of them are rule-based systems [3][16] but owing to their unreliability, there is an increase in research for developing IDS based on Data Mining [4], Autonomous Agents [5], Artificial Neural Networks [6][7], statistical techniques [12] and Genetic Algorithms [8]. The rule-based systems are unreliable as they are not very good at avoiding false-positive alerts. Moreover, regular rule update cost is high. From the works of Dennings [2] and Cannady [9], we have found that there exists a behavioral pattern in most network attacks. Hence, building an IDS that can learn, remember and recognize attack patterns may be a solution to this increasing problem.

This paper discusses the difficulties of integrating effective machine learning into an IDS. Our discussion will be limited to the problems we faced while developing LIDS [10][11] and discuss its architecture in brief. Special emphasis will be given to the design and function of the learning agent.

5.3 Intrusion Detection

Intrusion Detection is defined as the problem of identifying unauthorized users, stealing sensitive information from a computer network. It is also an attempt to protect the system from being compromised by intruders. The first study on this exponentially increasing problem was dated 1987 [2]. From then, the computer world has seen a vast change in network security systems. With the exponential increase in the Internet and the dexterity of intruders, people are becoming more aware of the safety and the privacy of their information. Recent exposure of critical vulnerabilities in different operating systems and news about computer intrusions has created an unsecured atmosphere for online customers. As a result, industry is facing a huge amount revenue loss.

Most of the available Intrusion Detection Systems are rule based [3][16]. They are regularly updated with the changing network systems, but they are not reliable in detecting a combination of attacks or a new kind of attack. Moreover, they are expensive and bear a maintenance cost. Thus, researchers engaged themselves in developing the next generation Intrusion Detection Systems that uses different non-traditional techniques like Data Mining, Statistical Techniques, Artificial Intelligence, Agent frameworks, Genetic Algorithms and so on. Though early studies on using these techniques show a promising result, they have failed to perform well in real systems. Moreover, most of these techniques are unproductive in real-time analysis. Thus, there is a need for a reliable, robust and real-time Intrusion Detection System that can detect known attacks, can learn new intrusions and later identify them.

Intrusion Detection tries to identify two general categories of attacks, namely Anomaly Detection and Misuse Detection. Anomaly Detection identifies activities that vary from established patterns of users, or groups of users and Misuse Detection involves comparison of a user's activities with the known behavior of system penetration. It is difficult for present systems to detect both kinds of attacks. In [12], we found that maintaining a profile of each authorized user or group is useful in detecting anomaly attacks in systems, but it is difficult to maintain a behavior profile for each legitimate user when the number of users increases. Behavior patterns also change with the mental state of the person and thus detecting anomaly attacks by comparing patterns with user profiles gives rise to a large number of false positive alerts. Moreover, there is no good definition for a healthy system. So flagging any behavior that is not a feature of a healthy system is not a good approach.

Much work is done to perform Misuse Detection. Researchers have used Artificial Neural Networks, Fuzzy Logic, Genetic Algorithms and Logic Trees to detect misuse of computer systems. Still, they are not very successful when the system faces a combination of attacks. There are many attempts of using agent technology such as autonomous agents [5], intelligent agents [13] and mobile agents [14] for distributed Intrusion Detection. Still, no previous work presents a learning agent that can update the system automatically and can train intrusion analyzers automatically without the intervention of the network administrator.

5.4 LIDS: Learning Intrusion Detection System

LIDS is a blackboard-based Intrusion Detection System that learns new attack patterns and later identifies them in the computer network. LIDS uses autonomous

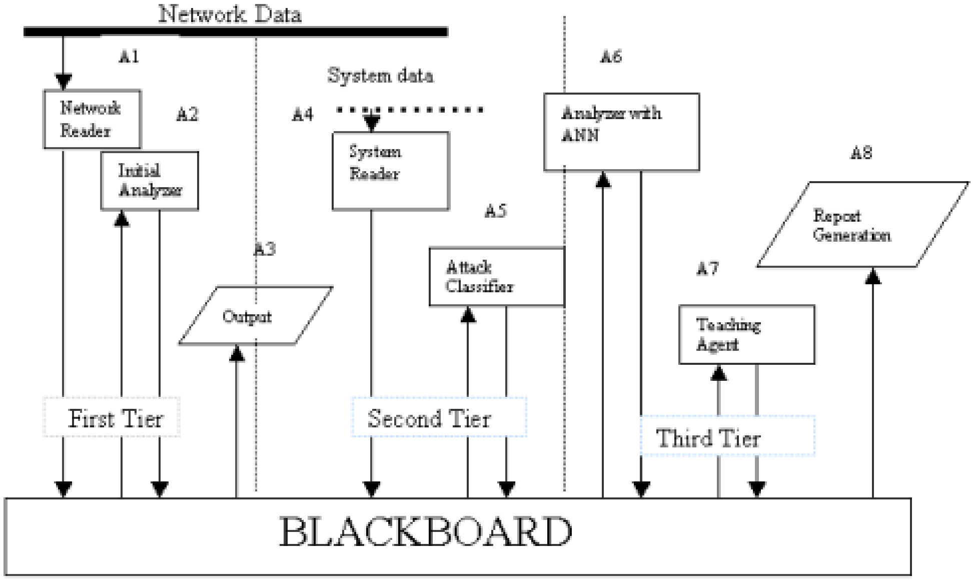


Figure 5.1: Learning Intrusion Detection System Architecture

agents to perform different functions in the system. The use of a blackboard architecture was chosen for its generality and flexibility features. Blackboard architectures are widely used in building decision-based applications. In this kind of architecture, the agents share information through a common data-sharing platform called blackboard. It is highly preferred over other alternatives due to its modularity, dynamic control, generality, concurrency, high design efficiency, robustness and ability in dealing with multiple knowledge sources. We also use autonomous software agents that are independently running entities that can resist subversion and have minimal overhead. They are also configurable, easily adaptable, scalable, dynamically reconfigurable and degrade gracefully. The architecture of LIDS is shown in Figure 5.1.

In the LIDS architecture, there are eight autonomous agents that interact with the blackboard. The eight agents are Network Reader, Initial Analyzer, Initial Alert Agent, System Data Reader, Attack Classifier, Intrusion Analyzer, Learning Agent and the Report Generation Agent. Generally, a blackboard system consists of three components; they are the action agents, a blackboard and a control mechanism that will guide these agents [10][11]. In LIDS, there is no control/manager agent, but there is a control pattern embedded in each agent that guides their activities.

The first agent also called the Network Reader, collects network data with the help of a program called tcpdump. Tcpdump is a network utility tool that records network data in a specific format. The network reader pastes the seen data stream on the blackboard. The second agent is the Initial Analyzer. This knowledge-base analyzer identifies different attacks in the system and posts alerts, if any, on the blackboard. This agent is guided by a set of rules that are automatically updated by the learning agent. The analyzer is written in the form of a PROLOG predicate. The third autonomous agent is the output agent. It helps in displaying any early alert to the security auditors or administrator of the system, like the information posted by the Initial Analyzer. The fourth autonomous agent is the System Reader. It gathers system specific information and posts it on the blackboard. These system specific data are later used by the fifth analyzer called the Attack Classifier. This is a GA classifier that is based on multiple-fault diagnosis paradigm [15] and is capable of detecting specific kinds of attacks. The information provided by the GA classifier will be used by the next autonomous agent (A6). A6 consists of an Artificial Neural Network that is used to identify the exact data stream responsible for the attack. This is called the Main analyzer. The seventh agent is the heart of LIDS, called the Teaching/Learning agent. This agent helps in updating the system with new attack information and new anomaly behaviors of the network. The final autonomous agent is the Report Generation agent that generates a final report of the attacks found or any false positives, if present.

From the above discussion, we see that LIDS uses A blackboard as a common data store. All the agents have access to it and this makes the system more flexible than a similar approach described in [5] where they have specific information flow paths. LIDS also has a control pattern embedded in the agents. This pattern allows the last agent to look at the blackboard first and the first agent last in order to ensure that each agent gets a chance at least once to look at the blackboard in one process cycle. The ultimate purpose of an Intrusion Detection System is to identify the affected network data with some degree of confidence. LIDS is capable of associating a degree of confidence with the attacks analyzed and it also handles most of the problems faced by the present approaches.

5.5 Designing the learning agent

While building LIDS, we had to answer some basic design issues and approaches to machine learning. Though these issues are common to all learning systems, they were particularly difficult in an Intrusion Detection environment. The dynamic nature of the network and the near real-time feature of the system made it hard to design the learning agent.

5.5.1 Training of the learning agent

The type of training experience is a very important design issue while building a learning system. The type of training experience available can have a significant impact on success or failure of the learner. We had an objective of building a learning Intrusion Detection System that can train itself when it sees a new kind of attack. Moreover, we are required to update the rule-set of the Initial Analyzer (A2) and train the Artificial Neural Network (ANN). Though the first task is easy, it is difficult to design the second function. The basic idea of this learning system is to reduce the number of false-positive alerts. So whenever a data stream is flagged by agent A2 or the initial analyzer and later it is found clean by the ANN, a relative rule is added into the knowledge-base of A2 to avoid future false-positives. Thus, the system starts updating its guidance rules.

From previous studies, it is found that new kinds of attacks are originated from one of the four basic attacks. These basic attacks are rlogin attack, network hopping, scanning and IP spoofing attack. A denial-of -service attack is an amalgamation of the above attacks. As a Wards Artificial Neural Network architecture is the best among other architectures for detecting the above mentioned attacks, it is also capable of identifying new attacks to some degree. LIDS saves the cost of regular updates to Intrusion Detection Systems, but it still needs to update the Main Analyzer after a drastic change in the network.

5.5.2 Identifying and representing a Target Function

This is the next important factor that guides the design of a learning system. It determines exactly what type of knowledge will be learned and how this will be used by the Initial Analyzer. In an IDS environment, we are trying to learn the attack patterns. The rules that guide the system to identify an attack are derived from the computer network data. Whenever there is a contradiction in the output of the Initial Analyzer and the Main Analyzer, the learning agent inspects the data stream, derives a rule-set and updates the knowledge base of A2 (Initial Analyzer). This process can be done in many ways but we have used an ID3 algorithm to generate the rules. The learning agent generates the new rules in the same format as other rules in A2. If a case arises where there is a contradiction between two rules, the new rule will get more priority than a previously entered rule. This preserves the integrity and the efficiency of LIDS.

5.5.3 Learning Algorithm

As mentioned earlier, we have used an ID3 algorithm to derive the rule-set. The input to the algorithm is the data stream for which the outputs of the Initial Analyzer and the Main Analyzer contradicted each other. Moreover, the learning agent posts specific information regarding the data stream that triggered the false alarm on the blackboard so that the Report Generation agent can generate detailed reports for the network administrator.

All the above agents are built in PROLOG. The analyzers like the ANN, the GA classifier, which also acts as a network filter, and the ID3 knowledge generator, are written as C++ and Visual Basic dll's. They are called by the autonomous agents while performing in a computer network. This online learning mechanism is a very strong feature of LIDS that makes it unique and different from the other known systems. This work is also the first in presenting a common platform for all the different analysis techniques.

5.6 Conclusion and Future Direction

While building LIDS, we have tried to use the best AI techniques for different knowledge base problems. This opens a new research area for developing learning IDS. We consider our work as one of the first attempts in building a self-learning system for this difficult problem. There is much more work needed to refine the learning process. The flexibility of the blackboard architecture also allows us to add more features in the future. This work may prove to be an important milestone in developing the next-generation Intrusion Detection System.

5.7 References

[1] Power, R. (2002), Computer Security Issues & Trends. Vol. 8, No. 1, 2002. pg 4.

[2] Denning D E (1987), An Intrusion-Detection Model, In IEEE Transaction on Software Engineering, Vol. Se-13, No. 2, February 1987, 222-232.

[3] White, G.B., Fisch, E.A., and Pooch, U.W. (1996). Cooperating Security Managers: A Peer-Based Intrusion Detection System. IEEE Network. January/February 1996 pp. 20-23.

[4] Lee, W, Stolfo S and Mok K (2000), Adaptive Intrusion Detection: A Data Mining Approach, In Artificial Intelligence Review, Kluwer Academic Publishers, 14(6): 533 - 567, December 2000.

[5] Balasubramaniyan, J, Fernandez J O, Isacoff D, Spafford E. and Zamboni D (1998), An Architecture for Intrusion Detection Using Autonomous Agents, In COAST Technical Report 98/5, Purdue University, June 1998.

[6] Debar, H., Becke, M., & Siboni, D. (1992). A Neural Network Component for an Intrusion Detection System. In Proceedings of the IEEE Computer Society Symposium on Research in Security and Privacy.

[7] Debar, H. & Dorizzi, B. (1992). An Application of a Recurrent Network to an Intrusion Detection System. In Proceedings of the International Joint Conference on Neural Networks. pp. (II)478-483.

[8] Ludovic, ME (1998), GASSATA, a Genetic Algorithm as an Alternative Tool for Security Trails Analysis, in the Proceedings of First International Symposium of Recent Advances in Intrusion Detection, 1998.

[9] Cannady, J. (1998), Artificial Neural Network of Misuse Detection, in the Proceedings of the 1998 National Information Systems Security Conference (NISSC'98) October 5-8 1998. Arlington, VA

[10] Dass, M., Cannady, J., Potter, D. (2003) LIDS: Learning Intrusion Detection System. In Proceedings of 16th International FLAIRS Conference, May 12-14, 2003 St. Augustine, Florida. pp 12-16.

[11] Dass, M., Cannady, J., Potter, D. (2003) A Blackboard-Based Learning Intrusion Detection System: A new approach. In Proceedings of AIE/IEA, June 23-26, 2003. Loughborough, England, UK

[12] Helman, P. and Liepins, G., (1993). Statistical foundations of audit trail analysis for the detection of computer misuse, IEEE Trans. on Software Engineering, 19(9):886-901.

[13] Carver, C. A., Hill, J. M. D., Sudru, J. R., and Pooch, U. W. (2000). A Methodology for Using Intelligent Agents to Provide Automated Intrusion Response.

In IEEE Systems, Man and Cybernetics Information Assurance and Security Workshop, West Point, N.Y., June 2000.

[14] Asaka, M., Okazawa, S., Taguchi, A., and Goto, S. (1999). A Method of Tracing Intruders by Use of Mobile Agents. In INET' 99, June 1999

[15] Dass, M., Cannady, J., Potter, D. (2003) A GA based Intrusion Sub-classifier Filter. In Digital Proceedings of ACM-SE 2003, March 7-8, 2003. Savannah.

[16] Anderson, D., Frivold, T. & Valdes, A (May, 1995). Next-generation Intrusion Detection Expert System (NIDES): A Summary. SRI International Technical Report SRI-CSL-95-07

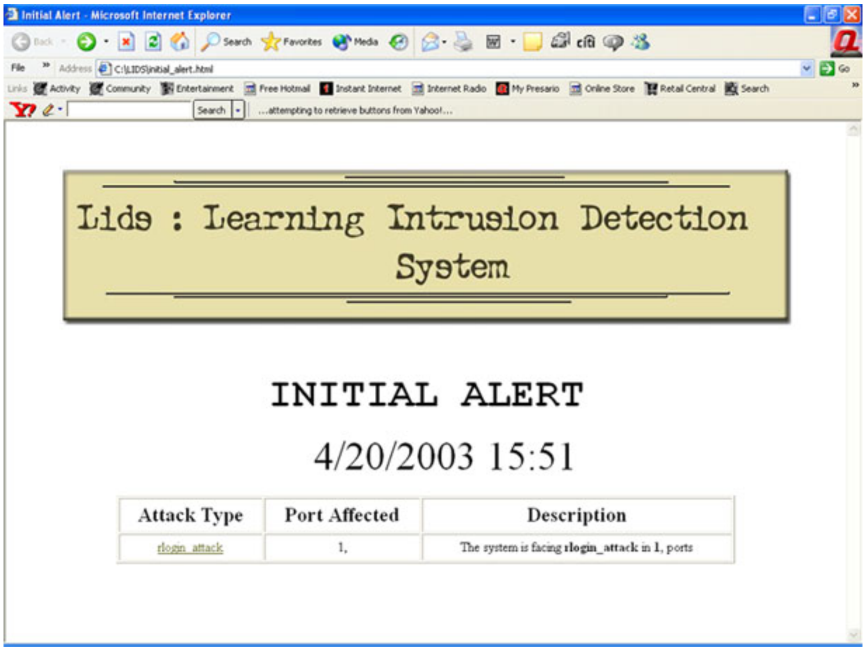
Chapter 6

Conclusion and Future Directions

6.1 Results and Reports

The preceding chapters described the architecture of LIDS and the different autonomous agents, and walked the readers through the process of detecting and learning intrusions. This chapter presents some results obtained from our system. We will also present the validity of these results in this section.

From the previous chapters, we found that LIDS is not only capable of detecting intrusions, but also able to flag new or anomalous attacks and remember them. This is how LIDS adapts itself to the environment it is in. The design philosophy of LIDS makes it very sensitive at the beginning of its learning process and as it is exposed to network data, it starts updating the knowledge base of the initial analyzer to reduce the number of false positives. Another important feature of LIDS is its capability of providing initial alerts to security personnel before it starts investigating the network data further. Figure 6.1 shows one of its generated initial alerts. These alerts are created by the third agent and are made to pop up in the default browser of the system. These alerts also have a drill-down facility in order to provide further information about the attacks observed. This helps the security personnel to have indepth knowledge about the attacks. This drilled-down information page is shown in Figure 6.2. We can also justify the initial analyzer by comparing it with the final report shown in Figure 6.3. Figure 6.3 represents the final report generated by LIDS.



Lids : Learning Intrusion Detection System

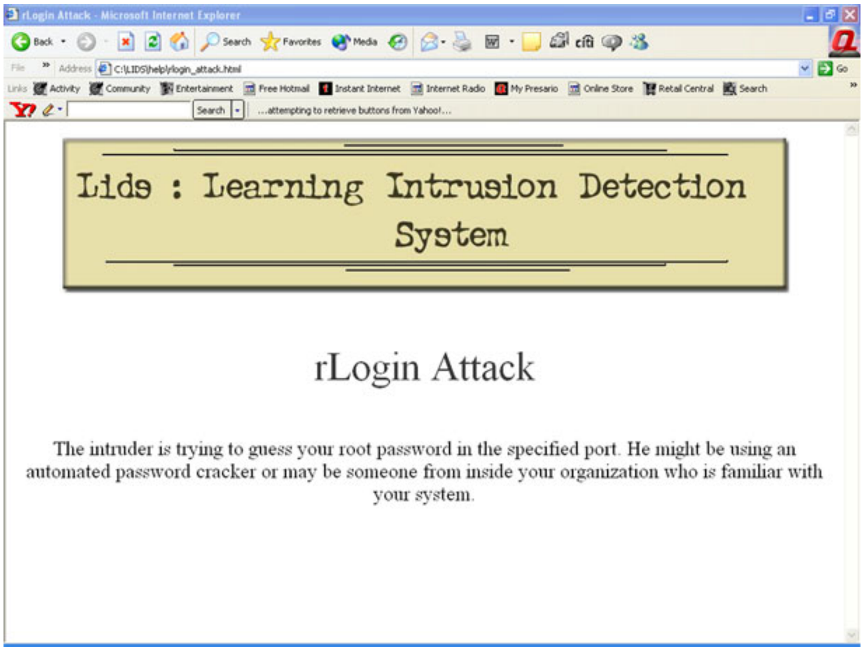
INITIAL ALERT

4/20/2003 15:51

Attack Type	Port Affected	Description
rlogin_attack	1,	The system is facing **rlogin_attack** in **1**, ports

Figure 6.1: Initial Alert Report

The report generation agent (eighth agent) gathers information provided by the ANN analyzer (sixth agent) and generates the report. The final report also generates a table showing the real network data that are identified by the artificial neural network as attacked data. It also highlights the datastream that was misclassified by the artificial neural network as attacks. This is shown in Figure 6.4

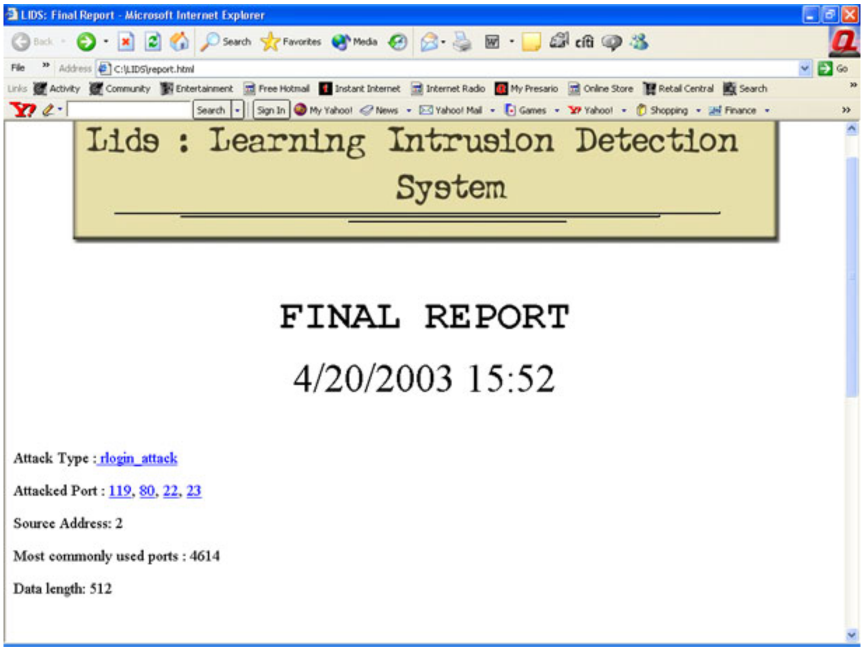


Lids : Learning Intrusion Detection System

rLogin Attack

The intruder is trying to guess your root password in the specified port. He might be using an automated password cracker or may be someone from inside your organization who is familiar with your system.

Figure 6.2: Attack Information Page



Lids : Learning Intrusion Detection System

FINAL REPORT

4/20/2003 15:52

Attack Type : rlogin_attack

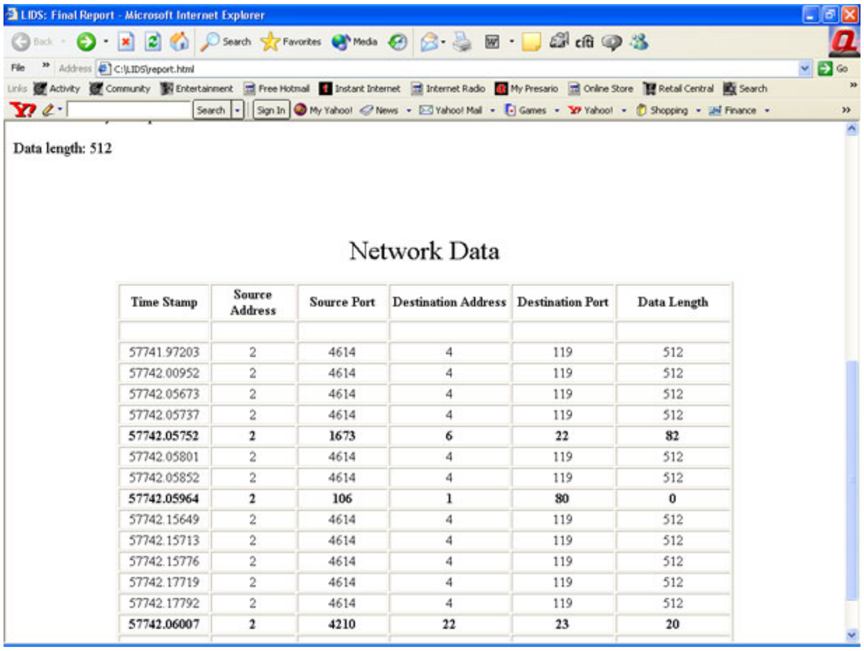
Attacked Port : 119, 80, 22, 23

Source Address: 2

Most commonly used ports : 4614

Data length: 512

Figure 6.3: Final Report



Data length: 512

Network Data

Time Stamp	Source Address	Source Port	Destination Address	Destination Port	Data Length
57741.97203	2	4614	4	119	512
57742.00952	2	4614	4	119	512
57742.05673	2	4614	4	119	512
57742.05737	2	4614	4	119	512
57742.05752	**2**	**1673**	**6**	**22**	**82**
57742.05801	2	4614	4	119	512
57742.05852	2	4614	4	119	512
57742.05964	**2**	**106**	**1**	**80**	**0**
57742.15649	2	4614	4	119	512
57742.15713	2	4614	4	119	512
57742.15776	2	4614	4	119	512
57742.17719	2	4614	4	119	512
57742.17792	2	4614	4	119	512
57742.06007	**2**	**4210**	**22**	**23**	**20**

Figure 6.4: Final Report showing Network Data

6.2 Further Work and Future Direction

While building LIDS, we have tried to use the best AI techniques for different knowledge base problems. This opens a new research area for developing learning IDS. We consider our work as one of the first attempts in building a self-learning system for this difficult problem. Further work as listed below is required to grow the interest of building a learning security tool and enhance this system to a next level of perfection.

- Refinement of the learning process - There is much more work needed to refine the learning process of LIDS. Presently, the system uses an ID3 algorithm to update the database of the initial analyzer. Further work may be done to refine this approach with a different algorithm.

- New Feature addition- The flexibility of the blackboard architecture allows us to add more features in the future. We should utilize this opportunity to add a visual feature to the system. This will help the security personnel in the industry to provide detailed reports to their authorities.

- Different analyzing techniques - As mentioned above, the flexibility of the system enables us to use different techniques in detecting network intrusions. Presently, we are using an Artificial Neural Network to detect intrusions. Later, we can try other analyzing techniques like Data Mining or Human Immunization as our main analyzer.

Intrusion Detection is a hard problem. Detecting intrusions only with the help of a bunch of rules or with a single Artificial Intelligence technique may not be a good idea. We need systems that can learn, can adapt and have intelligence to identify an attack from a false alarm. LIDS possesses all the desired features of an ideal IDS. It may be considered as an important milestone in developing the next-generation Intrusion Detection System.