Journal of Artificial Intelligence
ISSN: 2229-3965 & E-ISSN: 2229-3973, Volume 3, Issue 2, 2012, pp.-85-89.
Available online at http://www.bioinfo.in/contents.php?id=71

NEURAL NETWORK - APPROACH TO NETWORK SURVEILLANCE

AMBARE V.B.[1], NARKHEDE V.P.[2] AND PUND M.A.[3]

[1]Department of I.T., PRMIT & R, Badnera, MS, India.
[2]Department of CSE, BNCOE, Pusad, MS, India.
[3]MCA Department, PRMIT& R, Badnera, MS, India.
*Corresponding Author: Email-

Received: February 21, 2012; Accepted: March 15, 2012

Abstract- In today's world, Information is one of the most valuable asset. Intrusion detection is a significant focus of research in the security of computer systems and networks. As the network of computers expands both in number of hosts connected and number of services provided, security has become a key issue for the technology developers. This work presents a prototype of a intrusion detection system for networks. There is often the need to update an installed Intrusion Detection System (IDS) due to new attack methods or upgraded computing environments. Since many current IDSs are constructed by manual encoding of expert knowledge, changes to IDSs are expensive and slow. To detect intrusions The process of learning the behavior of a given program by using machine-learning techniques (based on system-call audit data) is effective. Rule learning and hidden Markov models (HMMs) are some of the kinds of representative methods for intrusion detection. Among them, neural networks are known for good performance in learning system-call sequences. In order to apply this knowledge to real-world problems successfully, it is important to determine the structures and weights of these call sequences. However, finding the appropriate structures requires very long time periods because there are no suitable analytical solutions. In this paper, an efficient and scalable technique for computer network security is presented i.e. a novel intrusion-detection technique based on Adaptive Resonance Theory neural networks for network pattern classification, and a fuzzy logic controller for decision/action resolution. One advantage of using NNs is that it takes less time to obtain superior neural networks than when using conventional approaches. This is because they discover the structures and weights of the neural networks simultaneously.

Citation: Ambare V.B., Narkhede V.P. and Pund M.A. (2012) Neural Network - Approach to Network Surveillance. Journal of Artificial Intelligence, ISSN: 2229-3965 & E-ISSN: 2229-3973, Volume 3, Issue 2, pp.-85-89.

Introduction

IDS (Intrusion detection system)is the term for a mechanism which quietly listen to network traffic in order to detect abnormal or suspicious activity, thereby reducing the risk of intrusion. IDS is the process defense system, which detect hostile activities in a network.

Attacks on computer infrastructures are a serious problem. Over the past twelve years, the growing number of computer security incidents on the Internet has reflected the growth of the Internet itself. Because most deployed computer systems are vulnerable to attack, intrusion detection is a rapidly developing field. Intrusion detection is an important technology business sector as well as an active area of research (Allen et al., 2000)[5]. There are many reasons why a computer system behaves in an undesired way. For a problem to be categorized as a security problem it must in some ways involve the fact or possibility that a human being does something that is not permissible. It is normally the person or organization who owns the system and/or the information who decides what is allowed and what is not. Wrongdoors can be categorized as insiders or outsiders. Insiders are persons related to the owner organization who try to misuse or extend their privileges. Outsiders are attackers who are unrelated to the owner organization who try to gain entry to systems (Cheswick, 1992). Within the community of security officers and researchers, insiders threat

is considered much more dangerous than the threat from outsiders, but the media have conveyed the opposite picture to the general public.

The security of a computer system is compromised when an intrusion takes place. An intrusion can be defined as any set of actions that attempt to compromise the integrity, confidentiality or availability of a resource (Heady et al., 1990). There are prevention techniques, such as user authentication (e.g. using passwords or biometrics), avoiding programming errors, and information protection (e.g., encryption) have been used to protect computer systems as a first line of defense. These techniques alone is not sufficient because as systems become ever more complex, there are always exploitable weaknesses in the systems due to design and programming errors, or various "socially engineered" penetration techniques. The policies that balance convenience versus strict control of a system and information access also make it impossible for an operational system to be completely secure [8].

Neural networks have been actively applied to IDSs. Especially, in the 1999 Defense Advanced Research Projects Agency (DARPA) intrusion detection evaluation (IDEVAL), the detection technique based on neural networks showed superior performance to the other techniques in detecting hostbased attacks[6]. However, profiling normal behaviors requires much time due to the huge amount of audit data and computationally intensive learning algorithms. Moreover, to apply neural networks to real-world problems successfully, it is very important to determine the topology of the networks and the number of hidden nodes in the given problem, because performance hinges upon the structure of the neural networks.

The basic artificial model

To capture the essence of biological neural systems, an artificial neuron is defined as follows:

It receives a number of inputs (either from original data, or from the output of other neurons in the neural network). Each input comes via a connection that has a strength (or weight); these weights correspond to synaptic efficiencycy in a biological neuron. Each neuron also has a single threshold value. The weighted sum of the inputs is formed, and the threshold subtracted, to compose the activation of the neuron (also known as the post-synaptic potential, or PSP, of the neuron)[4].Then the activation signal is passed through an activation function (also known as a transfer function) to produce the output of the neuron.

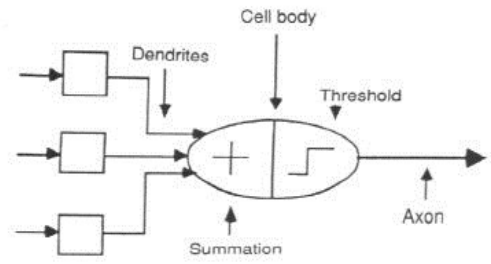

Fig.1- Artificial Neuron Model

If the step activation function is used (i.e., the neuron's output is 0 if the input is less than zero, and 1 if the input is greater than or equal to 0) then the neuron acts just like the biological neuron described earlier (subtracting the threshold from the weighted sum and comparing with zero is equivalent to comparing the weighted sum to the threshold). Actually, the step function is rarely used in artificial neural networks. Also weights can be negative, which implies that the synapse has an inhibitory rather than excitatory effect on the neuron: inhibitory neurons are found in the brain[8].

Various inputs to the network are represented by the mathematical symbol, x (n). Each of these inputs is multiplied by a connection weight. These weights are represented by w (n). In the simplest case, these products are simply summed, fed through a transfer function to generate a result, and then output. This process lends itself to physical implementation on a large scale in a small package. This electronic implementation is still possible with other network structures, which utilize different summing functions as well as different transfer functions.

Learning Process

One of the most important aspects of Neural Network is the learning process.

The learning process of a Neural Network can be viewed as reshaping a sheet of metal, which represents the output (range) of the function being mapped. The training set (domain) acts as energy required to bend the sheet of metal such that it passes through predefined points. However, the metal, by its nature, will resist such reshaping. So the network will attempt to find a low energy configuration (i.e. a flat/non-wrinkled shape) that satisfies the constraints (training data). Learning can be done in supervised or unsupervised manner.

In supervised training, both the inputs and the outputs are provided. The network then processes the inputs and compares its resulting outputs against the desired outputs. Errors are then calculated, causing the system to adjust the weights which control the network. This process occurs over and over as the weights are continually tweaked.

In unsupervised training, the network is provided with inputs but not with desired outputs. The system itself must then decide what features it will use to group the input data. This is often referred to as self-organization or adaptation.Once a network has been structured for a particular application, that network is ready to be trained. To start this process the initial weights are chosen randomly. Then, the training, or learning, begins[9].

The vast bulk of networks utilize supervised training. Unsupervised training is used to perform some initial characterization on inputs. However, in the full-blown sense of being truly self-learning, it is still just a shining promise that is not fully understood, does not completely work, and thus is relegated to the lab.

Unsupervised, or Adaptive Training

In unsupervised training, the network is provided with inputsbut not with desired outputs. The system itself must then decide what features it will use to group the input data. This is often referred to as self-organization or adaptation.

At the present time, unsupervised learning is not well understood. This adaptation to the environment is the promise, which would enable science fiction types of robots to continually learn on their own as they encounter new situations and new environments. Life is filled with situations where exact training sets do not exist. Some of these situations involve military action where new combat

techniques and new weapons might be encountered. Because of this unexpected aspect to life and the human desire to be prepared, there continues to be research into, and hope for, this field. Yet, at the present time, the vast bulk of neural network work is in systems with supervised learning. Supervised learning is achieving results.

Ai and Agent based IDS

1. Reader Agents

These agents monitor networked computers by executing commands, looking for deviations in the learned normal behavior. For each network host, there is an associated monitoring agent for each of the four levels: packet level, process level, system level, and user level. At the packet level, an agent detects changes in the numbers and sizes of packets for different protocols. At the process level, a different agent detects unusual process memory allocation, priority, CPU usage, etc. At the system level, an agent looks at overall system memory, CPU, and I/O usage. At the user level, an agent scans the file for login failures, attempts to gain root access, etc. All of these agents report anomalous behavior to a Decision/Action Agent (D/A Agent) for further processing. Figure 2 illustrates the process that Monitoring Agents follow to learn the behavior of monitored parameters using ART.

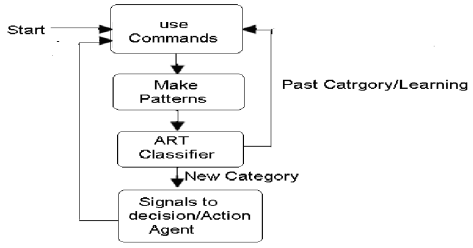

Fig. 2- Reader Agent process

2. Communicator Agents

These agents pass messages between agents. The Aglets Software includes these Messenger Agents as a primary feature.

3. Decision/Action Agents

These agents make decisions as to whether an action should be taken on behalf of the system administrator based on the information from the Monitoring Agents. Each of them has a fuzzy logic controller component in order to determine the severity of the anomaly and the age of such previous incidents to determine whether the D/A Agent further activates one or more Response Agents-- Helper Agents and Killer Agents (Figure3).

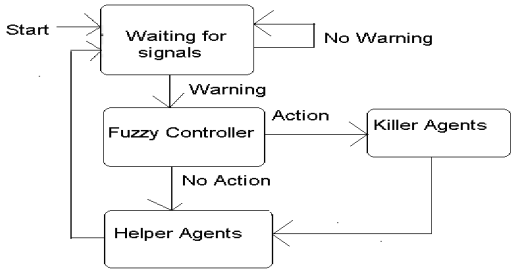

Fig. 3- Decision/Action agent process

4. Ready to help Agents

These agents provide status information to the system administrator's Graphical User Interface (GUI). They are activated by the Decision/Action Agent when a warning is received from the Monitoring Agents or when a Killer Agent has been dispatched.

5. Killer Agents

These agents terminate processes that are responsible for intrusive behavior on the network. The Decision/Action Agent dispatches a Killer Agent when the Threat Level determined by its Fuzzy Logic Controller is Medium- High or High. Once the process is terminated, the Killer Agent reports the action to the GUI using a Helper Agent.

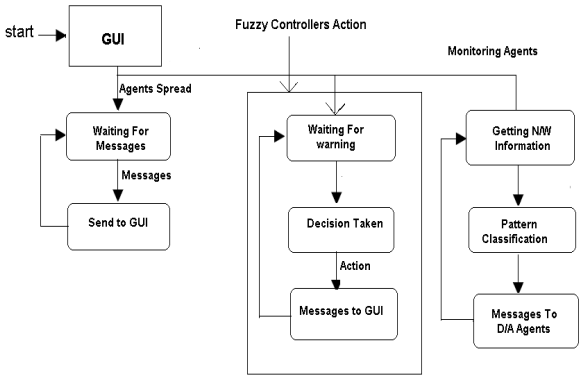

Fig. 4- Overall architecture of agent processes

The overall sequence of steps is shown in Figure 4. Once the agents are dispatched to the desired host on the network, the Graphical User Interface waits for messages from the agents to update its display. The Decision/Action Agents wait for warning signals from the Monitoring Agents and make decisions based on the information regarding violations. Any actions are relayed to the GUI for display[5]. The Monitoring Agents begin immediately sensing network status and classifying them into distinct categories using the ART neural network. After the training is completed, these agents report patterns that do not fit into known categories as anomalies to the Decision/Action Agent.

Decision support components

Some Monitoring Agents store network, system, or user behavior patterns to serve as a knowledge base or model of known "normal" behavior. When learning of "normal" behavior has ceased, the agent compares current network, system, or user behavior with its knowledge base of "normal" patterns. Any patterns that do not closely match previously seen patterns are considered anomalous and are reported to other agents for possible action against the user or process

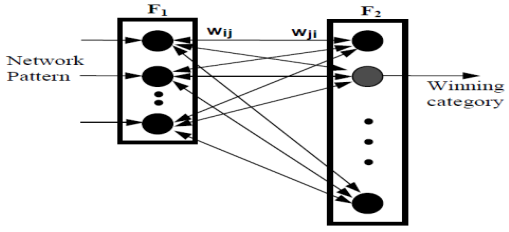

Fig. 5- ART Neural Network

ART Neural Network Classifier

The Adaptive Resonance Theory (ART) neural network classifier (developed by Grossberg) was chosen due to its ability to group presented patterns into categories without human supervision. ART is one type of an unsupervised neural network that uses competitive learning (Figure 5).

A pattern that does not closely match any of the known categories either causes the network to add a new category during the learning phase (Figure 6) or identifies the pattern as anomalous during the testing phase . ART networks self-organize stable recognition categories in response to arbitrary sequences of analog (gray-scale, continuous-valued) input patterns. ART networks encode new input patterns, in part, by changing the weights, or long-term memory (LTM) traces, of a bottom-up adaptive filter. This filter is contained in pathways leading from a feature representation field (F1) to a category representation field (F2) whose nodes undergo cooperative and competitive interactions. In anART network there is a second, top-down adaptive filter that leads to the crucial property of code self-stabilization[8].

Such top-down adaptive signals play the role of learned expectations in an ART system. They enable the network to carry out attentional priming, pattern matching, and self-adjusting parallel search .In order to cope with arbitrary sequences of analog input patterns, ART architectures embody solutions to a number of design principles, such as stability-plasticity tradeoff, the search-direct access tradeoff, and the matchreset tradeoff. A parallel search scheme updates itself adaptively as the learning process unfolds, and realizes a form of real-time hypothesis discovery, testing, learning, and recognition. After learning self-stabilizes, the search process is automatically disengaged. Thereafter input patterns directly access their recognition codes without any search. Thus, recognition time for familiar inputs does not increase with the complexity of the learned code. A novel input pattern can directly access a category if it shares invariant properties with the set of familiar exemplars of that category. The architecture's global design enables it to learn effectively despite the high degree of nonlinearity of such mechanisms.

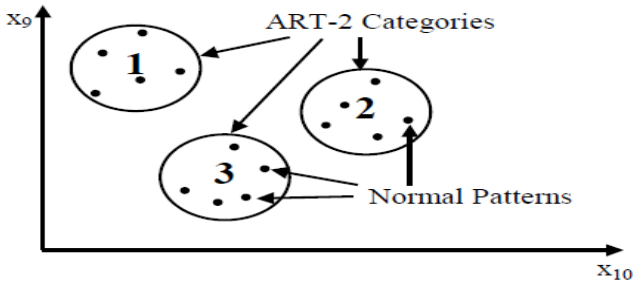

Fig. 6- ART-2 Categories created in ten dimensional hyperspace

Fuzzy Controller-

A fuzzy controller was developed for the Decision/Action Agents that must make decisions and possibly take action based on anomalous behavior. These agents receive reports of security incidents, including the severity of the event, from the Monitoring Agents and the cumulative totals are adjusted for age. Together, these incident characteristics are fed to a fuzzy controller, and a decision is made to take some action, such as terminate a Process, or do nothing, based on the controller output.

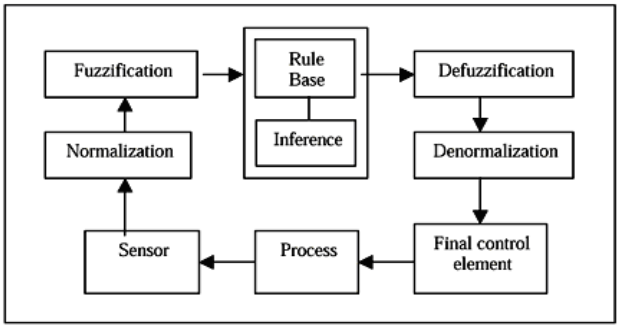

Fig. 7- Fuzzy Controller 1

There were five fuzzy sets created representing Low,Low-Medium, Medium, Medium-High, and High threat levels (Figure 8 (a)). 5^4 or 625 fuzzy rules were defined to govern the controller's decisions. For example, one such rule was "if System Threat is Low and User Threat is Low and Process Threat is Low and Packet Threat is Low, then Threat Level is Low." Based on the inputs from these four monitored levels of the network, the degree of membership to each set was calculated, and the union of the five resulting fuzzy sets was determined (Figure 8(b)) using the 2^4 or 16 active fuzzy rules[7]. A defuzzification method was applied to the set union to find the center of gravity of the set, yielding the actual Threat Value. If this value exceeded the Threat Threshold (>0.5, Medium-High or High), the controller dispatched an agent to kill the associated process, if one existed, or warned the network administrator of a high threat level. Otherwise, the controller took no action. Figures 8(a) and 8(b) show the membership functions used to implement the Fuzzy Controller.

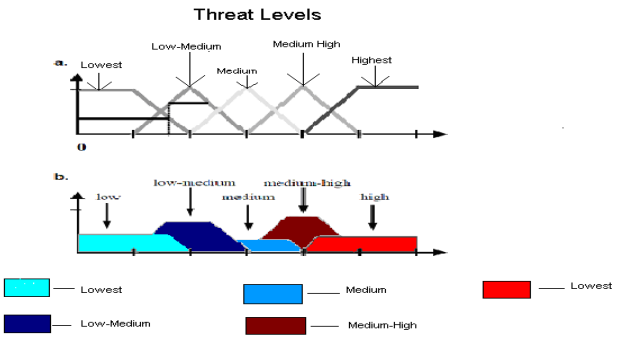

Fig. 8- a)Five fuzzy sets b) union sets

Conclusion

When a hacker attacks a system, the ideal response would be to stop his activity before he can cause any damage or gain access to any sensitive information. This would require recognition of the attack as it takes place. Recently, researchers started investigating techniques like artificial intelligence , autonomous agents and mobile agent architectures for detecting intrusion in network environment. Most existing intrusion detection systems either use packet-level information or user activities to make decisions on intrusive activities . In this paper, an agent-based intrusion detection system is described that can simultaneously monitor network activities at different levels (such as packet level, process level, system level, and user level). This system represents a novel approach to distributed intrusion detection. The system emulates

some mechanisms of the human immunity system and features distributed identification of anomalies and decentralized control of decisions and responses to those anomalies. Agents can move throughout the network observing network behavior patterns and communicating any anomalies to other agents for action. The ART neural network classifier is an ideal learning mechanism for Monitoring Agents. Observed network patterns can be classified into categories during a learning phase without loss or degradation to previously created categories. Patterns that fail to fit into known categories during the testing phase are assumed to be anomalous. A fuzzy controller takes all anomaly reports as input and determines the current Threat Level. If the threat is Medium-High or High, the Decision/Action Agent will take action to terminate the associated process.

References

[1] Kusum Bharti (2010) *International journal on computer science & engineering,* 02(05).
[2] Shihab K. (2006) *Journal of Computer Science*.
[3] William Stallings. *NETWORK SECURITY ESSENTIALS*, applications & standards 2nd Edition.
[4] Carbo J. (2005) *International journal on computer science & applications*.
[5] SANS Institute (2001) *Application of Neural Network to Intrusion Detection*..
[6] Mbaitiga Zacharie (2007) *Journal of computer science*.
[7] Jianping Zeng and Donghui Guo (2009) *International Journal of Network Security,* 8(3), 201-210, 201.
[8] Rehak M., Pechoucek M. and Celeda P. (2008) 7*th Int. Conf. on Autonomous Agents and Multiagent Systems* (AA-MAS).
[9] Padhy N.P. *Artificial Intelligence & Intelligent system*.