International Journal of Security and Its Applications
Vol.8, No.2 (2014), pp.183-192
http://dx.doi.org/10.14257/ijsia.2014.8.2.19

Multi-Agent Distributed Intrusion Detection System Model Based on BP Neural Network

Zhai Shuang-can, Hu Chen-jun and Zhang Wei-ming
Nanjing University of Chinese Medicine
zscdd@163.com

Abstract

On the basis of analyzing the existing intrusion detection system (IDS) based on agent, this paper proposed a multi-agent distributed IDS(DIDS) model based on BP neural network. This model adopted the modes of distributed detection and distributed response. Each Agent was independence relatively. And this model analyzed the functional design of each agent and central console. Meanwhile, to improve the performance of the system, an improved error back-propagation algorithm was designed, which could improve the detection accuracy of the system by using its good learning ability. In addition, the dynamic election algorithm and collaborative algorithm were analyzed preliminarily. Experiments proved that the system could complete the intrusion detection tasks by making full use of various resources collaboratively, and thus the detection speed and accuracy of the system could be improved.

Keywords: *BP Neural Network, DIDS, Multi-Agent*

1. Introduction

With the development of IDS technology, IDS now becomes intelligent and distributed, which tends to distributed architecture recently. For example, the thought of network layering was proposed by EMERALD [2] ,in which the whole system was divided into three layers physically, and each layer could be controlled by higher layers. DIDS [2] and GrIDS [3] were the same system like EMERALD. Actually all IDSs realized the function of LANs monitoring by the means of distributed data acquisition and hierarchical data analysis. Constructed in this way, the DIDS had the advantages of simple structure and rigorous logic. However, the DIDS also had some obvious deficiencies as follow: It might be a bottleneck that the centralized analysis component ran under heavy load which resulted in the single node invalidation; the real-time performance of the system was reduced by hierarchical analysis, and distributed collaborative network attacks were unable to be detected, *etc*.

Because of the complexity and diversity of the invasion means the network system, the neural network has greatly enriched the means of achieving intrusion detection system. Neural networks are simulated human brain processing, storage and processing of information mechanisms proposed an intelligent information processing technology, it has the ability to abstract generalization, learning and adaptive capabilities and inherent characteristics of parallel computing, making it in intrusion detection The application has obvious advantages. Agent is a class in a particular environment can sense the environment, and can run autonomously on behalf of their designers or users to achieve a series of goals to calculate entity or procedures Agent autonomy is the basic concept is different from other essential features. The so-called Multi-Agent (MAS) is defined by multiple interactions, mutual services together to accomplish a task, the ability to solve their collaboration over a single Agent. In Multi-Agent Systems, each Agent is autonomous, and real-time sensing of highly basic module unit, mutual cooperation and coordination between them, constitute a real-time highly functional entities. General Agent shall have the knowledge, goals and capabilities, and intelligent Agent with

ISSN: 1738-9976 IJSIA

mobility called mobile Agent. The purpose is to move the execution of the program data as close to the source, reducing network traffic overhead, saving bandwidth and load balancing, speeding up the task, thereby improving the processing efficiency of the distributed system. The main idea is to move the Agent handling the distribution of tasks and will be sent to the data processing code instead of $ nearby dataset collected and compared with the corresponding code to handle such a large extent reduce the amount of processing required for network communications , while increasing the parallel task processing.

To meet the needs of real-time and robustness of IDS, this paper proposed multi-agent DIDS model based on BP neural network. This model could make full use of internet resources to complete the tasks of intrusion detection collaboratively, and make real-time response. By using the powerful learning ability of BP neural network, the function of detecting suspicious intrusions could be enhanced greatly, and the detection speed would be improved effectively by multi-agent technology.

2. System Design

2.1. Main Framework

The multi-agent DIDS, with the mode of distributed detection, process and response, could detect and judge suspicious intrusions of the whole system in real time. It was a multi-agent DIDS in distributed environment. Each agent was independent, explicit and collaborative. There was an interactive relationship between global databases and local databases. When global databases or local databases finished upgrading, they would inform each other to upgrade in the form of a report, so as to complete synchronization of databases and achieve data sharing. A local database was installed in each monitored computer. All the local databases which were registered in data center could be updated by global databases or neighboring agents. The framework was shown in Figure 1[4].

Has the following advantages of multi- Agent Based Distributed Intrusion Detection Model:

(1) flexibility , scalability . Distributed architecture model and independent testing unit enables the host to detect whether an increase or increase the Agent are easy on the host. Each Agent can run as an independent entity is detected, you can put people in a distributed environment for collaboration as a detection means to detect.

(2) a variety of data sources. Since each Agent is an independent realization of different Agent can choose different data sources, such as audit data, check the system configuration, network packet capture, *etc.*, depending on the data source, that features a collection of different problem domains, data preprocessing module source encoded into different feature vectors, and then enter the corresponding smart Agent, in order to achieve accurate detection of multiple forms of intrusion.

(3) independence , portability. As each Agent is completely independent, they can develop and debug, respectively, and can be developed using different programming languages based on different platforms, simply follow the unified communication protocols and communication formats, they can communicate between collaboration, which is the distribution an important feature of the model formula.

(4) limited single point of failure. Agent -based Intrusion Detection Agent makes a single failure will only affect other parts of the Agent and the Agent and part of the collaboration with the module, the system can still work. And because the division of administrative domains, a single failure of the affected area is limited. Since independence Agent node, failure monitoring and management of network-level intrusion detection is not completely affected nodes.

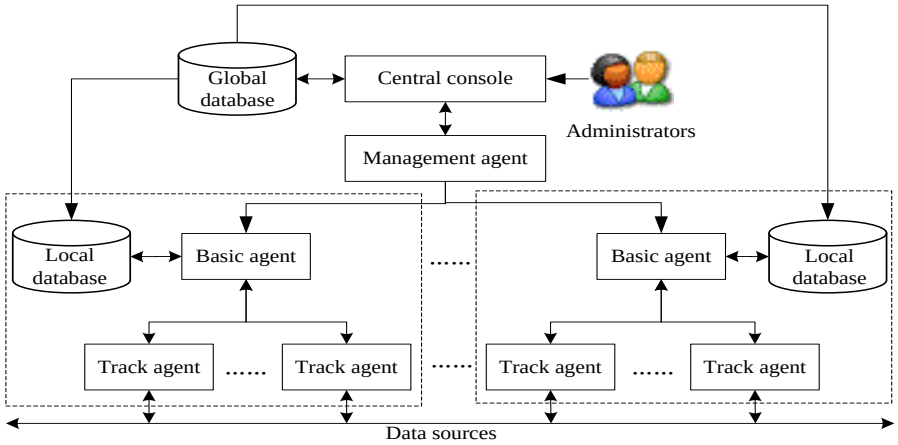

Figure 1. The Model of DIDS

2.2. Central Console

Intrusion detection could not only distinguish normal network data flows form abnormal ones, but also separate user's normal operations of computers or network devices from network attack activities. All activities on a host or network could be classified into three types: normal, abnormal and suspicious. This model was a framework of multi-agent DIDS, which was composed of two-level detections. The central console was composed of communication modules, inference modules, alarm modules, management modules and response modules, as shown in Figure 2. In order to improve the detection ability of DIDS, BP neural network was used in the inference module by quantifying local suspicious behaviors [4, 5].

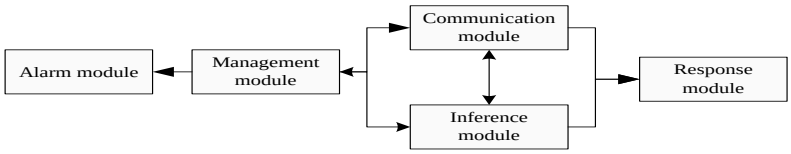

Figure 2. Architecture of a Central Console

(1) Inference modules. As the core of central console, they dealt with decisions from local agents, and formed the final results of judging suspicious behaviors.

(2) Communication modules. They were in charge of establishing, maintaining and managing communication links among different agents.

(3) Alarm modules. They were used to alert administrators whether there were network attacks or suspicious behaviors.

(4) Response modules. When the network attacks were detected, they could make responses properly against the threats according to intrusions and response rules. The response behaviors included attack tracking, target responding, source responding, evidence gathering from the host and network components, and source separating from target.

(5) Management modules. They made the whole system run normally, and sent configurations and control commands to management agents. They also displayed dynamic information for administrators through user interfaces, dealt with information interaction, received commands of administrators, and made real-time responses under strategies which were set by administrators.

The inference modules were the theoretical basis of central console. The key point of using inference modules effectively was to design relevant algorithms.

2.3. Basic Agent

Every host had only one basic agent (BA). When initializing system, all the hosts in the LAN would boot a basic agent, and then, a dynamic election of supervisor agent would occur among different agents.

BA could analyze network packets and generate tracer agent (TA), communicate with other BAs, detect collaborative intrusions in the LAN, and report suspicious results and abnormal behaviors to SA. According to the function requirement of BA, it should be composed of management modules, analysis modules, response modules, communication modules, collaboration modules and detection modules, as shown in Figure 3.

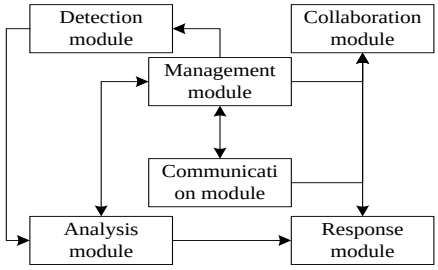

Figure 3. Architecture of a Basic Agent

(1) Management modules. As the central processing module of BA, they controlled the running conditions of other modules.

(2) Detection modules. They gathered the network data or security data, and pre-processed these data to form a data report in unified format.

(3) Analysis modules. According to the data which gathered by detection modules, they could detect and analyze network intrusions or abnormal behaviors by the step of protocol analyzing, feature extracting and pattern matching.

(4) Response modules. Under the control of management modules, they could blocking special data packets or network connections by packet filter or TCP connection reset.

(5) Communication modules. Communication tasks among SAs were managed by this module, which maintained effective communication links and reliable information transmissions.

(6) Collaboration modules. They could generate and parse mutual information among BAs.

When data flows flowed through BAs, the behaviors of data flows were classified as normal, abnormal and suspicious behavior, according to the analysis of BAs. The abnormal behaviors could be judged by pattern matching of expert knowledge base, while the suspicious behavior could not be ascertained by a single BA, which needed to collaborate with other BAs to judge. This kind of suspicious behaviors could be normal or attack behaviors after the final judgment. This structural model worked like this: Firstly, misuse and anomaly detection techniques were used to detect local information by BA. When a network attack was detected, the model would give an alarm and send a report to the center console. While, if the behavior could not be judged, a local judgment would be made and sent to the center console; secondly, the detection report was analyzed by association analysis method in the center console, and the final decision would be made combining with local judgments, then processed it accordingly. This kind of processes

synthesized local judgments of several BAs, network behaviors and host behaviors, therefore, network attacks could be fully detected [5, 6].

2.4. Track Agent and Management Agent

Track Agent (TA) which was generated by superior BA got various data in the monitored network environment widely. It could not only acquire network packets, but also system audit trails, operating system logs, system processes and register access information, etc. Different TAs collected specific network data. When raw data were got, they were preprocessed and converted into unified format and then transferred to BA for further analysis.

According to the functional requirements of BA, these modules were composed of management units, detection units and communication units.

(1)Management units. As the central processing unit of BA, they regulated and controlled the running conditions of other units.

(2)Detection units. They gathered security data, and pre-processed these data simply to form a report in unified format.

(3)communication units. They managed the communications between TA and superior BAs and maintained effective communication links and reliable information transmissions.

The whole system had only one management agent which was elected by all the local BAs dynamically. The management agent was in charge of receiving BAs' alarms in the LAN (including suspicious and abnormal behaviors), and then sending detection results to the central console. SA was composed of communication units and detection units. Communication units managed different BAs and coordinated relationships between BAs by transferring commands. The information analyzed by BA, especially the suspicious behaviors which BA could not judge, were uploaded to the central console with standard data interchange format [7].

3. Key Algorithm

3.1. Dynamic Election Algorithm

When the system was initialized or a BA found crashes of other BAs in the same network segment, an election was needed to ensure that the system worked regularly. The election aim was to choose a BA which had the lightest load as SA, meanwhile, adjusted functional units of agents by functional requirements. The load size was measured by average CPU utilization and memory utilization. The classical algorithm -Bully was used here, and specific steps were as follows [8]:

(1) An election message (election(X)) was sent to other BAs, in which the load was X.

(2) If other BAs found their loads were less than X in the same network segment, a message "OK" would be sent to the BA who launched the election, otherwise quit the election.

(3) If the BA who launched the election did not receive the message "Ok", the BA who sent the massage would be the new SA, and the election finished, otherwise quit the election and turned to (4).

(4) Other BAs ran through the sequence (1)~(3).

3.2. Collaborative Algorithm

The multi-step attacks needed to be detected by collaborations of BAs. If a detection task of multi-step attacks which was allocated by BAi was defined as B={b1,b2,⋯,bn}, then BAs would collaborate with each other in the way below[8]:

(1) It could be uncover that multi-step attacks bi∀B was sequentially composed of n atomic attacks from A (A ={a1,a2,⋯,an}) by querying the ontology base, and then, BA0 would broadcast a message to all the other BAs for subscribing detection results of network attacks.

(2) Other BAs who undertook the tasks of detecting network intrusions would reply to BA0.

(3) When a BA detected the network intrusions that BA0 subscribed, it would reply to BA0.

(4) Atomic attack messages which reported form TA or other BAs were analyzed with association analysis method. The detection model was built to analyze suspicious user s, and then speculated their intentions. If there were bad intentions, an alarm would send to the center console.

BAs were used to find out collaborative attacks in a network segment, and collaborative attacks were analyzed according to alarms from atomic attacks of TA and other BAs. When atomic attacks were gathered, BAs analyzed them with association analysis method, and found out all the possible associations which meant the sets of combinations of all the possible invaders.

3.3. Inference Algorithm based on BP Neural Network

The final judgments of suspicious behaviors depended on the inference algorithm of central console. Combining with applications of BP neural network in network intrusion detection, we would discuss the algorithm design in this section and present a practical learning algorithm which could meet the requirements of detection accuracy.

The thought of this inference algorithm was that data packets or system logs were used as data sources, and the neural network algorithm was used as detection engine. When the neural network was trained, it could recognize users who had similar features as network intrusions used in the neural network training. Then input vectors were discriminated and activation vectors were analyzed by the neural network and the importance of events was measured by weight. After that, the suspicious degree, given to measure suspicious states of special links, was used to judge whether there was a network intrusion. New network intrusions might be detected by administrators, while not detected or found their severities by the neural network. In this case, new modes would add to the neural network and could be recognized after neural network training.

3.3.1. Improved BP Neural Network Algorithm

For the differentiation of network intrusions exactly, the standard BP neural network was improved in two points:

(1) Add momentum term. The standard BP neural network adjusted weights by the gradient descent direction of errors at time t, and did not consider the gradient direction before t, which resulted in training shocks and slow convergences. In order to improving training speed, a momentum term a was added in the weight adjustment equation which was shown as follow:

$$\Delta W(t) = (\eta)\mathrm{Err}\sum\nolimits_{j} O_i + a\Delta W(t-1)$$

in the equation above,ΔW(t) was a weight adjustment variable at time t, ΔW(t-1) was also a weight adjustment variable at time t-1, a was a momentum coefficient and ranged 0 to 1. The momentum term reflected accumulated experiences before and damped the adjustment at time t. When the error surface fluctuated suddenly, we could reduce the shock trend to improve training speed.

(2) Adjust the learning rate adaptively. There was so many methods to adjust learning rate adaptively, and the common method was used like this: First, a initial learning rate was set, and if E(t) was bigger than E(t-1) after a group of adjustments of weight, the adjustment was invalid in this time and the learning rate was changed toη(t+1)=η(t-1)-η(t+1)/t, otherwise the adjustment was valid and the learning rate was changed toη(t+1)=η(t)-η(t)/t.

3.3.2. The Training Steps of Improved BP Neural Network

(1) Initialization. The weight matrix W(0) and threshold matrixΘ(0) were set to a non-zero value in random, and other parameters were also set, for example, the maximal iteration was set to N, the learning rateη to a decimal from 0 to 1, and precision E to a decimal.

(2) Training samples were supplied by sample files and expectation files, and each input sample was iterated by the input vector.

(3) The actual output and states of hidden units of neural network were computed.

(4) The training errors of output layers and hidden layers were computed separately.

(5) The training errors of current samples were saved and the iteration would not stop until the training end of this sample.

The total training errors were computed. If E(t)>E(t-1), the iteration was invalid and the learning step was decreased to re-calculate in iterative.

(7) Increase the learning step

$$\eta(t+1) = \eta(t) + \eta(t) / t \text{ 。}$$

(8) According to the total error $E = \sum_k E_k$, the weight of neural network was amended as follow.

$$\Delta W_{ji}(t) = (\eta)\mathrm{Err}_{kj} O_{ki} + a\Delta W_{ji}(t) \text{ 。}$$

(9) If errors did not meet the need, the calculation would continue in iterative, otherwise weights and thresholds were saved in corresponding files.

(10) If the number of iteration steps were over the maximum, the training failed, otherwise the training could not stop.

(11)The network training stoped.

4. Experimental Analysis

Experiments were programmed with MATLAB on Windows 2000 platform. The data of “KDDCUP data set” was used to train and test. In the experiments, 8600 data was selected from the data set. The selected data should contain common attack methods as many as possible, and each attack method should reach to a certain number of data, so as to meet the training demands of neural network.

In order to contrast the performances of the proposed models in this paper, the results of BP neural network algorithm merely were compared with those of both BP neural network algorithm and multi-agent technology after training and testing, the results of the experiment were shown in Table 1. The experiment indicated that: it could achieve better training effect by applying the neural network to the intrusion detection; the non-response rate was relatively high along with the high accuracy detected only by improved BP algorithm; the application of multi-agent technology to the DIDS can not only ensure the high accuracy but also effectively lower the non-response rate, thus meeting the system's requirements of real-time performance and accuracy.

Table 1. Experimental Results

Network attack types	Methods	Normal	DoS	U2R	Probe	Other	False response rate	Non-response rate
Training datasets	——	1056	47	15	49	——	0.038	0.1
Test dataset 1	BPNN	2344	69	37	79	11	0.0254	0.107
	BPNN+Multi-Agent							0.078
Test dataset 2	BPNN	2651	74	59	83	15	0.0312	0.113
	BPNN+Multi-Agent							0.085
Test dataset 3	BPNN	1853	59	46	70	9	0.0291	0.121
	BPNN+Multi-Agent							0.079

5. Conclusion

DIDS based on agent technology has been the important development direction in the field of intrusion detection, and has the advantages of reducing the mobile process of data, balancing the load, detecting analysis neatly, error-tolerating better, and detecting distributed intrusion effectively. On the basis of analyzing the existing IDS based on agent, this paper proposed a multi-agent DIDS model based on BP neural network. Compared with the existing IDS based on agent, this multi-agent DIDS model had two-level detection functions and enhanced abilities of distributed detection, response and processing because of local detection and global detection. It could also deal with the detection of those local suspicious behaviors which couldn't be handled better in central console and could effectively defense single point failure due to the management agent which was produced with dynamic election. Meanwhile, in order to improve the detection performance of the unknown intrusion, the detection method based on BP neural network was designed. Experiments proved that the multi-agent DIDS model based on BP neural network could effectively improve the detection accuracy and greatly reduce the workload of the central console because of the multi-agent technology, and so as to improve the detection efficiency of the system, which meant that this model had high applicability.

References

[1] W. Yan-Xin, S. R. Behera and J. Wong, "Toward the automatic generation of mobile agents for distributed intrusion detection system", Journal of System and Software, vol. 79, no. 8, **(2006)**, pp. 1-14.
[2] P. Porras, P. Neumman and E. Erald, "Event monitoring enabling response to anomalous live disturbances", Proc of the 20th National Information System Security Conference, **(1997)**, pp. 353-363.
[3] E. Spafford and Z. Bonie, "Intrusion detection using autonomous agents", Computer Networks, vol. 34, no. 4, **(2000)**, pp. 547-570.
[4] Y. Qing, W. Xiao-ping and Z. Ding-jun, "Multi-agent distributed intrusion detection system model based on evidence reasoning", Application Research of Computers, vol. 26, no. 8, **(2009)**, pp. 3063-3066.

[5] Z. Ping-hui, "Design of Intrusion Detection System Based on Neural Network", Microelectronics & Computer, vol. 26, no. 8, **(2009)**, pp. 240-242.
[6] L. Feng-chun, Z. Hao and Z. Bao-hua, "Intrusion detection system design in wireless LANs based on optimized BP algorithm", Journal of University of Science and Technology of China, vol. 40, no. 10, **(2010)**, pp. 1096-1100.
[7] M. Chang-lou and L. Yong-qing, "An Distributed Intrusion Detection System Model Based on Multi-agent", Computer & Digital Engineering, vol. 37, no. 6, **(2009)**, pp. 102-106.
[8] W. Jun, W. Chong-Jun and W. Jun, "Dynamic Hierarchical Distributed Intrusion Detection System Based on Multi-agent System", Computer Science, vol. 34, no. 2, **(2007)**, pp. 71-75.
[9] L. Ling-juan, Z. Shuang-can and G. Li-wei, "Predicting the Separation Process of Chinese Medicine Water Extraction Liquid Membrane with Support Vector Machine", Computers and Applied Chemistry, vol. 27, no. 2, **(2010)**, pp. 149-154.

Authors

Shuancan Zhai, received the B.education degree in Computer software from The PLA Information Engineering University and the M.Eng degree in Software engineering from Southeast University's Software College, China in 1988 and 2010 respectively..He is currently researching on Internet Of Things, Bioinformatics and High-throughput Sequencing Data Analysis.

Chenjun Hu, received the B.education degree in Educational Technology from Nanjing Normal University and the M.Eng degree in Computer Science from Southeast University's School of Computer Science and Engineering, China in 2000 and 2008 respectively..He is currently researching on Internet Of Things, Bioinformatics and High-throughput Sequencing Data Analysis.

Zhang Weiming, in 1989 graduated from the Changchun College of geology, and obtained a bachelor's degree, master's degree in Nanjing University of Technology in 2007. He is a lecturer in the school of information technology, Nanjing University of Chinese Medicine, teaching and research work is mainly engaged in computer network, network security, interactive web site development, data mining and other fields.