Research on Distributed Intrusion Detection System Based on Mobile Agent

Zhisong. Hou
School of Information Engineering, Henan Institute of Science and Technology, Xinxiang, China, 453003
forhouor@gmail.com

Zhou. Yu, Wei. Zheng, Xiangang. Zuo
School of Information Engineering, Henan Institute of Science and Technology, Xinxiang, China, 453003
yyzhou@tom.com, karl777@163.com, zuoxg2002@163.com

***Abstract*—For the problems of traditional intrusion detection system, a distributed intrusion detection system based on mobile agent was designed. In this paper, the internal function of mobile agent was divided in architecture view, and the optimal agent migration algorithm was researched to facilitate the agent collaborative processing. The communication manner and interactive processes were applied in the design. Furthermore, the traditional Boyer-Moore (BM) algorithm was improved. The simulation results showed that the system could reduce the network load, shorten the waiting time of network. All these contributed on the improvement of the intrusion detection. Then, the dynamic adaption of the system could be implemented while false alarm rate and false negative rate would be reduced. Therefore, the system is suitable for large-scale heterogeneous network.**

***Index Terms*—intrusion, mobile agent, intrusion detection system, agent migration, BM algorithm**

I. INTRODUCTION

For the development of network technology, intrusion activities on computer system became more and more popular as the Internet users increases rapidly. Therefore, intrusion detection system, an effective method for defending intrusion, developed rapidly in recent years.

According to the increase of network size and complexity, in large heterogeneous network, traditional intrusion detection system could not complete multiple information processing. At the same time, the safety of the whole system could not be achieved as that intrusion detection system was partly failure. When network is busy, the characteristic data of intrusion activities would be loss for that the data cannot be processed by the system. Then there would be security vulnerabilities. Moreover, intrusion activity could not be found as the disadvantages of warning mechanism in current intrusion detection system. Therefore, function of intrusion activities defending would be failure in the system.

For solving the problems of traditional intrusion detection system, this design applied mobile agent in intrusion detection system. Then, a new architecture of intrusion detection system would be produced which is based on the mobile agent. Optimal BM algorithm and agent migration algorithm were developed. These improved the detecting efficiency.

II. RESEARCH ON RELATED TECHNOLOGIES

Mobile agent is the intelligent agent with mobility. It can independently and autonomous move between each node in heterogeneous network.

Mobile agent is a software entity which could represent other entities (including person or other agents). It could independently choose the option place and time or interrupt the current execution. In different circumstances, it could move to another node and return with related results. The purposes of the migration are to make the execution closer to data source, reduce network overhead, save bandwidth, balance network load, accelerate the process automatically. All these could be contribute to enhancing the efficiency of distributed system [1].

The main advantages of mobile agent include several points: network bandwidth is saved and network delay is reduced as the data processed in local node; interoperability of heterogeneous networks is achieved based on the decrease of components coupling as bottom network protocol encapsulated; dynamic adaption is implemented which depended on the current network configuration, topology architecture and flow [2].

Intrusion detection system evaluates the user or system behaviors in suspicious level, and identifies the legitimacy of the behavior based on the evaluation. Then system administrator could manage system safely and handle system attack.

Intrusion detection system includes data acquisition module, data analysis module and response module [3]. Specifically, data acquisition module collects the status and behaviors of the system, network, file data and users. The collected information would be applied for intrusion analysis. Module of data analysis assembles the collected information, and applies pattern matching, statistical analysis or integrity analysis for detection. When intrusion was detected, corresponding response would be applied for terminating intrusion or attacking, and the system would try to recover the influenced service and

doi:10.4304/jcp.7.8.1919-1926

the lost data. The architecture of the system is shown in Fig. 1.

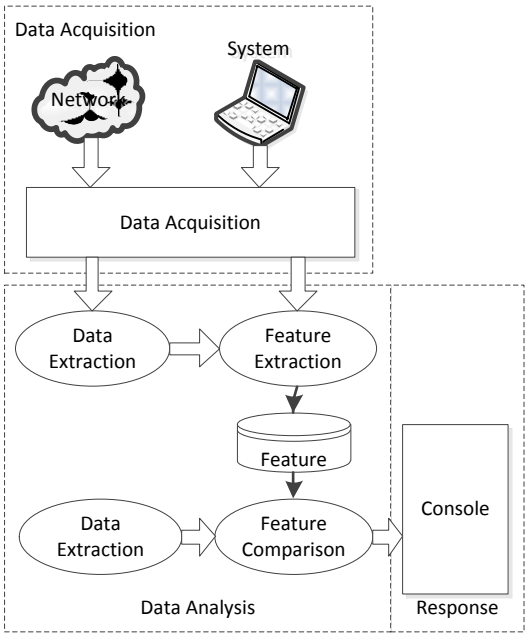

Figure 1. The structure of traditional intrusion detection system

Traditional intrusion detection system can be divided into three categories: integral structure, hierarchal structure and distributed structure. Specifically, integral structure integrates all functions, but it is weak on system prevention; hierarchal structure is a tree-structure which is composed with detector and controller. It shares information of all subsystems for intrusion detection. The disadvantage of the structure is single-point failure; in distributed architecture, intrusion detection system is divided into several modules. The modules distribute in the heterogeneous network environment, and each module receives different input information, finishes different missions. Then, they report the results to the high function units until integrated console. The distribution of the modules is matched with the characters of mobile agent [4] [5] [6].

Therefore, intrusion detection system applies technology of mobile agent. As a result, mixed structure could be implemented based on the technical advantages of mobile agent. This could compensate the defects of traditional intrusion detection system.

III. Distributed Intrusion Detection System Based on Mobile Agent

A. System Architecutre of Intrusion Detection System

The intrusion detection system based on mobile agent is composed of several mobile agents who can migrate in the whole network and be developed on mobile agent platform. The architecture of distributed intrusion detection system based on mobile agent can be shown in Fig. 2.

As can be seen from Fig. 2 that: the intrusion detection system based on mobile agent composes Management Agent, Integrated Learning Agent, Crawler Agent, Sensor,

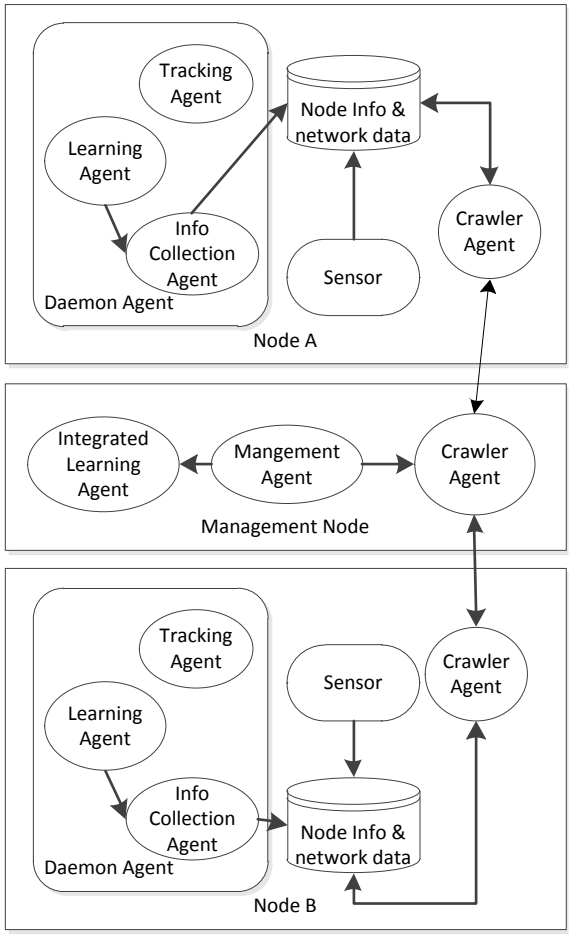

Figure 2. The architecture of intrusion detection system

Daemon Agent, Tracking Agent, Learning Agent and Info Collection Agent. The functions of them are shown below:

1) Management Agent: Management Agent is the agent which is in the highest level. There are one or more management agents in a system. In complex distributed network, several management agents are applied for network management mode of hierarchal structure. This could avoid single-point failure.

Integrated learning agent of Management Agent analyses the learning result from learning agent and responses; at the same time, management agent provides the interface between administrator and system.

Management agent records the status of all agents in the network, and completes the detection and management of the agents regularly. Management agent controls all agents. Furthermore, related agents are sent to a target system based on the network status. Management agent notices daemon agent suspend, resume or transmit agents when agent is running. At last, management agent is responsible for agent recycling as mission completed.

For hidden distributed attacks, sensor of single node could not detect intrusion, and this could lead the intrusion detection failure. Therefore, management agent has to send tracking agent to all network nodes, and

collects the information to analysis or judge whether there is an intrusion activity.

2) Daemon agent: Daemon agent is the basis of the system. Each daemon agent is processing on each node. It provides the applicable environment for operation and transmission to internal agents.

Daemon agent is controlled by management agent, and it could manage the agents on the node. Daemon agent can start, suspend or terminate agents, and transmit agents in target networks.

3) Crawler agent: crawler agent is sent to whole network by management agent for information collection. The agent is responsible for traversing the entire network. It is necessary as that premeditated network intrusion with planning, and technical preparation is with a large span and space. As a result, reducing the rate of intrusion depends on traversal whole network regularly.

4) Sensor: each node installs a sensor. Sensor monitors network information and system audit log, and then it formats the monitored data and saves the data into intrusion detection database.

In a fixed period, the saved information in database is analyzed in a certain rule. If distributed attack was found, there are two ways to deal with: firstly, sending info collection agents and learning agents to other nodes to collect and learn for detecting that if distributed attack occurred; secondly, noticing management agent, and management agent sends info collection agent and learning agent. At last, collect and send detection information to management agent for analysis.

5) Info collection: info collection agent can move and collect the information related to intrusion from database on target node.

6) Learning agent: learning agent can move in the network. It can apply data mining algorithm to extract characters of user behavior and related pattern rules.

7) Integrated learning agent: integrated learning agent is on management node. It is responsible for analysis and synthesis of the information from collection agent and the result from learning agent. Then, comprehensive safe mode and character of user could be achieved. According to the analysis, management agent could judge whether there is intrusion and send warning alarm or not.

8) Tracking agent: tracking agent is to track the path of intrusion, judge the source node of intrusion and search the node position where invaders log on the network. At the same time, tracking agent tracks the nodes where invaders reached to.

In common, a tracking agent could not finish the task of tracking intrusion path. Then, several tracking agents work together to track the intrusion path. Therefore, when several intrusion behaviors are found on the target node, daemon agent on the node sends several tracking agents which are corresponded to the intrusion behavior.

The *8* functional agents work corporately and complete intrusion detection for network system. The system is distributed on all network nodes. Each node installs tracking agent, sensor, info collection agent and learning agent. Tracking agent tracks intrusion path and judge the intrusion node; info collection agent collects information of target node, sends the information to learning agent for analysis, and extracts the pattern mode of safety and character of user behavior. Crawler agent collects information of all nodes, and sends the information to the integrated learning agent which stays on the management node. At last, management agent detects intrusion according to the analysis from integrated learning agent.

B. Mobile Agent Migration

In order to facilitate collaborative processing in intrusion detection system, it is essential to deploy an efficient computing paradigm that supports collaboration among sensors.

In the computing based on mobile agent, the data is processed locally on network node. Tracing agents which are from Daemon agent are dispatched, and expected to visit the subset of sensors to integrate local processing results [7] [8].

The computing paradigm of mobile agent supports a wide range of collaborative information processing applications which is including detection, classification, and monitoring. Although each individual sensor does not have enough information to finish detecting, a number of sensors jointly could process information for a sensing task. The mobile agent computing is to migrate to a group of sensors in a particular sequence, maximizing the information extraction while minimizing resource usage.

The measurement is made by a sensor node *s* at time *t*, $z_s(t)$ is modeled as follows

$$z_s(t) \propto \frac{L}{\left\|x(t)-x_s\right\|^2}. \tag{1}$$

where L is the information load of the host of sensor, $x(t)$ is the target location at time *t*, x_s is the location of sensor *s*, and $\left\|x(t)-x_s\right\|$ is the Euclidean distance between sensor *s* and the target at time *t*.

To quantify the contribution of individual sensors to the success of its task, the concept of information gain is introduced. From equation (1) that at any time instant, the measurement $z_s(t)$ is related to the target position $x(t)$ by $\left\|x(t)-x_s\right\|$, the distance from the target to the sensor is *s*. Thus, this distance as a good approximation is used to the gain of useful information if the mobile agent migrates to the sensor node *s*. Then, the relationship between the information gain and the target distance is modeled as a zero Gaussian used,

$$I_s(t)=\frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{\left\|\overline{x(t)}-x_s\right\|^2}{2\sigma^2}}. \tag{2}$$

σ is the standard deviation. It is a parameter that determines I_s decreases when the Euclidean distance between the node and the target increases, and $\overline{x(t)}$ is the estimated target location which is calculated from the target localization algorithm.

In the intrusion detection system, sensor on network node decides to dispatch learning agent and info collection agent. The learning agent and info collection agent traverse the network to collect useful information. Thus, the migration algorithm is the key factor to the system.

To improve migration efficiency, an optimal itinerary is chosen that consumes the least resources for fulfilling the collaborative processing task (info collection and learning). In the process, some information is needed for effective mobile planning. The information includes global information and local information. The global information could be seen as a complete route picture of the whole network. On the other hand, the local information only contains the information of the node. Sensors can exchange their local information by that an update message is periodically sent to neighboring nodes.

According to the current network, for collecting information and tracking intrusion, the sensor must determine the next hop of mobile agent. The mobile agent on the current network node, such as learning agent and info collection agent, seeks the sensor which would provide the greatest amount of information gain. The gain is the most informative sensor in its neighbor N_s. Then, the sensor returns the next hop.

Besides maximizing the information gain from the neighbor nodes, it needs to prolong the lifetime of the whole network, and to reduce the distance for the agent migration. The final decision of the next hop for the mobile agent is combined with the consideration of gain and loss. Define a cost function $C_{kj}(t)$ for mobile agent migration from node S_k to a neighbor node S_j at time t as:

$$C_{kj}(t) = a\frac{\left\|x_k - x_j\right\|^2}{d_{\max}^2} + b\frac{\left\|\overline{x(t)} - x_j\right\|^2}{dt_{\max}^2} + (1-a-b)(1-\frac{l_j(t)}{l_{\max}}) \quad . \tag{3}$$

where $\left\|x_k - x_j\right\|$ is the distance from node k to node j, $\left\|\overline{x(t)} - x_j\right\|$ is the distance from node j to the estimated target location at time t, and $dt_{\max}$ is the maximum distance from a node to the target, $l_j(t)$ is the network load of node j at time t, $l_{\max}$ is the maximum network load. It is assumed that all nodes start with the fixed network load which is an experience value. a, b are the weights used to adjust the importance of those components, and $0 \leqslant a,b \leqslant 1$.

In order to increase the lifetime of the whole network, the mobile agent should always choose the next neighbor node which has a lower network load. According to (3), the function takes all three factors into consideration with a and b controlling the weights of three factors to different mobile agents. Thus, the decision is

$$j^* = \arg\min_{j\in N_s} a\frac{\left\|x_k - x_j\right\|^2}{d_{\max}^2} + b\frac{\left\|\overline{x(t)} - x_j\right\|^2}{dt_{\max}^2} + (1-a-b)(1-\frac{l_j(t)}{l_{\max}}) \quad . \tag{4}$$

where N_s is the set of neighbor nodes of s, and node $S_j^* \in N_s$ is the next node that the agent will migrate to. At the same time, the node consumes less network load while would have more system resource.

In collaborative processing of the info collection agent and learning agent, the movement pattern of the target is an important factor that affects the final results. To reduce the migration step, the moving direction of the target should be considered.

Although the network nodes are discrete in heterogeneity network, it is assumed that the target dynamics are small, which means that the target does not change direction, network traffic and system resource abruptly. The direction and the system load are within short time interval. That can be deemed as constant. Therefore, the target changes between the time interval t to $t + 1$ is equal to $t - 1$ to t:

$$x(t+1) - x(t) = x(t) - x(t-1) \; . \tag{5}$$

By (5), the mobile agent can predict the target location at future time $t + 1$. Therefore, an updated cost function is adopted to evaluate the cost to its neighbor nodes. The neighbor node with the minimal cost function value is chosen. The new optimal decision of the next node to migrate is:

$$j^* = \arg\min_{j\in N_s} a\frac{\left\|x_k - x_j\right\|^2}{d_{\max}^2} + b\frac{\left\|\overline{x(t+1)} - x_j\right\|^2}{dt_{\max}^2} + (1-a-b)(1-\frac{l_j(t)}{l_{\max}}) \quad . \tag{6}$$

where N_s is the set of neighbor nodes of s.

Once the info collection agent and leaning agent arrive to a node, a node would be selected among unvisited nodes or living neighbor nodes. The node costs the smallest resource as calculated from (6), such that a near-optimal itinerary can be determined. If all the neighbors have been visited, the agent will migrate back to the sensor where the agent was dispatched. The agent will stop migration when the collaborative processing result has enough information to detect the intrusion.

C. Process of Intrusion Detection

Intrusion detection is completed by both network node and management node. The agent of network node collects information extracts safe pattern rules and the character of user behavior. Then it sends the information to management node; the agent of management node analyzes information of all network nodes, receives information from network node, judges and responds as analysis done.

According to the information collected ways of network agents, process of intrusion detection is composed of information collection sub-process, intrusion detection and response of management node sub-process.

The sub-process of information collection is illustrated in Fig 3.

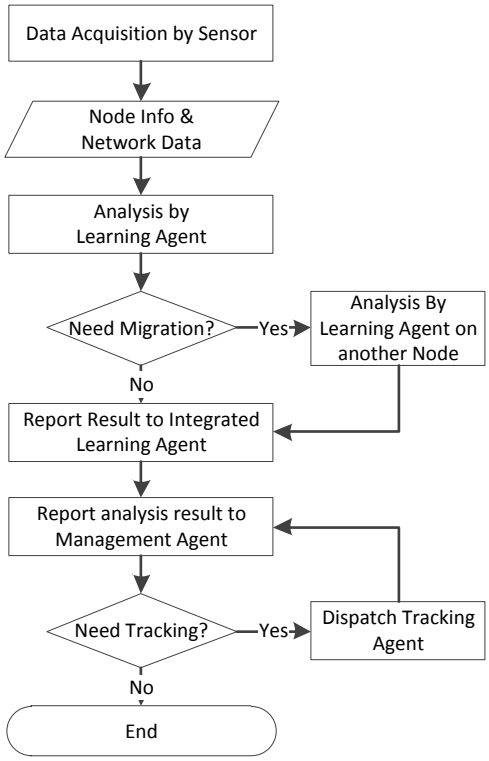

Figure 3. The process of node info collection

As seen from the Fig 3, it is necessary to collect suspected acts of intrusion for intrusion detection. The detailed steps are shown below:

1) Sensor on network node monitors network data and node data, collects information, then sends the information to info collection agent.

2) The formatted information is sent to learning agent for analysis, and then judges that if the process has to be transmitted to other node for collection information and learning. Otherwise, the process turns to step (4).

3) If the transmission is necessary, a certain node is selected and the agent is transmitted to this node. The process of collecting and analysis are finished by the selected node as well.

4) The result of learning would be reported to integrated learning agent which is on the management agent as mission completed by learning agent.

5) Management agent decides whether that tracking agent is sent according to the information from integrated agent. If intrusion behavior is confirmed, the collection of information is completed. Otherwise, tracking agent is sent.

6) Tracking agent tracks the suspected intrusion behavior until the source node, and then sends the tracking information to management agent.

The information collection sub-process only reports the information of network node. To achieve the whole network defense, it is necessary to collect the information of the whole network nodes. The process is achieved by intrusion detection and response on management node sub-process. The process of that is shown in Fig 4.

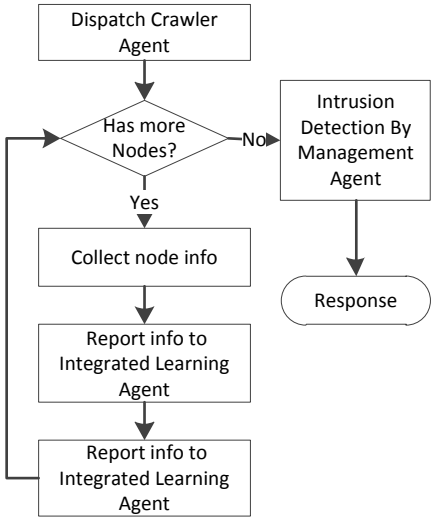

Figure 4. The process of intrusion detection

In Fig 4, intrusion detection and response process sends crawler agent in timing. The crawler agent travels the whole network nodes, collects information of target node and returns to management agent with the information. At last, the information is transmitted to integrated learning agent for analysis.

Management agent detects and judges intrusion according to the analysis of integrated learning agent. If intrusion is confirmed, the warning would be alarmed and certain protection measures would be taken to deal with the intrusion consequence, then the system turns to response process.

D. Communication among mobile Agents

There are many agents in the system. Each agent takes specific functions. Several agents need to work corporately and exchange information for intrusion detection.

There would be conflicts among agents as the autonomy of mobile agent [9]. The communication among agents is very important as that if there are several tracking agents, info collecting agents and learning agents. Moreover, the communication of integrated learning agent, info collection agent and learning agent is necessary as that management agent could receive the operational status of system.

The system sets specific public information exchange area on each node to exchange information for communication. The area is bulletin board which is used by learning agent; message board is used by tracking agent.

All agents can visit the public information exchanging area. Learning agents exchange rule information via bulletin board. Tracking agent finds the information that

if a path was tracked by other tracking agents via message board for avoiding conflict of path tracking.

There are bulletin boards and message boards installed on all management nodes, which is used for collecting messages from nodes and the tracked paths.

In the system, the path of information exchange between mobile agents is shown in Fig. 5.

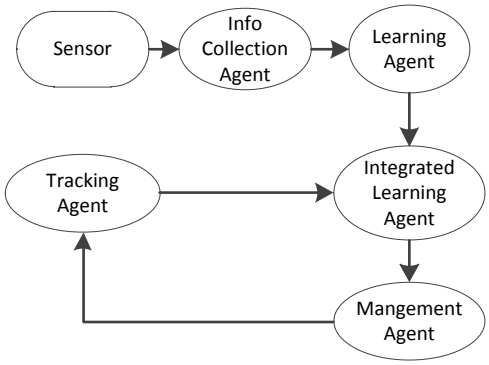

Figure 5. The path of information exchange

E. Improvement of Matching Algorithm

In the intrusion detection system, string matching algorithm in learning agent directly determines the efficiency of intrusion detection. In this system improved BM algorithm is developed.

Suppose a text string *T* of length *n*, a pattern string *P* of length *m*, both of which are an ordered collection of characters. The text *T* is defined in a finite alphabet Σ, whose size is σ. The purpose of matching is to determine the location of *P* in *T*, or *T* does not contain *P*. Let *T* and *P* start with offset 0, and be numbered sequentially from left to right. Therefore, for a given offset *s*, if every character $P[i] \in \{P[0], P[1], \ldots, P[m-1]\}$ completely matches the corresponding character $T[i+s] \in \{T[0], T[1], \ldots, T[n-1]\}$, then *P* matches *T* in offset *s*.

The basic principle of BM algorithm is: pretreat the pattern string *P* to calculate the value of the two offset function Badchar and Goodsuffix, and then match the text string *T* with the pattern string *P* by aligning the text string *T* with the pattern string *P* at their leftmost character. When they are not matched, calculate the offset by the function Badchar and Goodsuffix to find the maximum offset. Then move the text pointer to the right with the maximum offset and match them again until they are matched [10].

For a given character *a* in Σ, the offset calculated by Badchar function is:

$$Badchar[a] = \frac{m, a \notin P}{\min\{i \mid P[m-1-i] = a\}, a \in P}. \quad (7)$$

where the array space of Badchar is σ, and $0 \leqslant i < m-1$.

Goodsuffix is used to calculate offset of the text pointer to right when a suffix is matched successfully. Offset is determined by two factors:

- The position where the suffix appears again in the pattern except itself
- A suffix of the suffix mentioned above is the prefix of the pattern when the first condition is changed.

In BM algorithm, two conditions are defined as:

- Condition1(*i*, *s*): for every *k*, $1 \leqslant k < m$, $s \geq k$ or $P[k-s] = P[k]$
- Condition2(*i*, *s*): if $s < j$, then $P[i-s] \neq P[k]$

The first condition is used to calculate the offset where the suffix appears again in the pattern except itself. The second condition ensures that the calculation is the i-th position of the pattern. Therefore, if and only if both of the conditions are reached, the Goodsuffix function is:

$$Goodsuffix[i+1] = \min\{s > 0\}. \quad (8)$$

where the array space of Goodsuffix is *m* + 1, and $0 \leqslant i < m$.

Next, align the pattern string *P* with the text string *T* left-justified, and then match the string one by one from the rightmost character. If the text character *T*[*i*+*j*] does not match the pattern character *P*[*j*] in a match, the values of Badchar[*T*[*i*+*jj*]+*i*+1-*m* and Goodsuffix[j] is calculated. Take the bigger value as the offset that text pointer will move to the right and match from right to left by the same way. If the pattern is scanned from right to left, the match is done. Move the text pointer to right by the distance Goodsuffix[0], and search again until the end of the text string *T*. Finally, the position of the pattern string in text string will be located.

It is clearly that the core operator of the BM algorithm is done mainly by the function of Badchar and Goodsuffix. That means both of the functions are the key factor of the efficiency of intrusion detection. In order to improve the efficiency of the pattern matching, the BM algorithm is improved in two ways:

The first way is the improvement of the loop of pattern matching. Both the ends and the middle of the pattern with priority are compared to reduce unnecessary comparison.

The second way is to reduce the distance of the next step to move. When the pattern does not match the text window, the moving distance of the text pointer is greater than or equal to 1. Typically, when *m* is much smaller than *n*, the possibility of pattern appearance in text string is smaller. Thus, it can get the offset *m* + 1 in the matching process easily.

By this, time complexity of the BM algorithm is $O(m+\sigma)$, and the space complexity is $O(\sigma)$.

IV. System Simulation

The system is implemented on JADE (Java Agent Development Framework). JADE is a software framework fully implemented in Java language. JADE provides a stable, reliable and flexible platform for agent operation. It provides the basis platform for the system running, and it is responsible for management of the mobile Agent [11] [12].

System simulation environment consists of 10 units of PC servers running FreeBSD. Interconnection of servers uses Fast LAN.

Firstly, choose five servers, install intrusion detection system, and count the data flow of the servers with the intrusion detection system. Then, install intrusion detection systems for the other five servers, and count the data flow of the servers with the intrusion detection system again. The results are shown in Table I.

TABLE I
THE SIMULATION RESULT OF NETWORK TRAFFIC

Hosts	network traffic(packet/hour)
5	2305
10	2330

Table I shows that, as installation of intrusion detection systems, servers network traffic has not changed significantly, because data analysis and processing are done at the detected servers and intermediate nodes. This saves network bandwidth.

Performance of intrusion detection mechanism is the key to the effects of intrusion detection system, which directly affect the efficiency of the system. The ideal situation is that: when the attack packets with invasion characteristics enter the internal network, intrusion detection will not bring significant delay which can be perceived. Thus, use recognition rate of invasion and response time of intrusion detection to measure the performance of intrusion detection mechanisms.

The intrusion recognition rate is the ratio of detected intrusion attacks to the number of actual intrusion attacks. According to the characteristics and behavior of the attacks, all kinds of attack packets are sent to the internal server from external host for simulating network attacks. Using *4* kinds of typical intrusion packet, the simulation results are shown in Table II.

As can be seen from Table II, for that the intrusions have defined in the characteristics library already, system has a higher rate of intrusion recognition. In other words, as long as the attack signatures were defined, the system could identify intrusion entered in network easily.

Intrusion detection response time is the time from the attack message invading into the network to the system producing invasion. The shorter intrusion detection time provides the higher efficiency of detection system. Especially when a large number of intrusions invade into the network, if the intrusion detection response time is too long, subsequent invasions will be lost.

Calculate the average response time of *4* kinds of intrusions in Table II, and the results are shown in Table III.

As Table III shows, for different types of intrusion, there are significant differences in response time of intrusion detection mechanism. For packet attacks, intrusion detection responds in millisecond time, and detection efficiency is high. For complex behavior attacks, due to tracking users' behavior sequence, it will take much longer response time. As the complex attacks often happen at the last of a series of actions, at which point the system has responded in past several acts, so the intrusion detection system can still achieve the goal.

TABLE II
RECOGNITION RATE OF INVASION

Intrusion Name	Intrusion Action	Expected Results	Test Result	Recognition of Rate
Telnet	input wrong or non-exist username and password	100 telnet login failure events	96	96%
LAND	send 100 TCP SYN packets to the server which have the same source and destination address and port	100 LAND attacks	93	93%
TearDrop	attack the server 100 times by TearDrop tools	100 TearDrop events	95	95%
SYN Flodding	send 100 TCP SYN packets containing pseudo-address to the server	100 SYN Flodding attacks	91	91%

The simulation results show that the mechanism of the intrusion detection system can effectively reduce the network load, complete intrusion detection accurately. At last the desired goal is achieved.

TABLE III
AVERAGE RESPONSE TIME OF INTRUSION DETECTION

Intrusion Name	Average response time
Telnet	9352ms
LAND	1.13ms
TearDrop	0.52ms
SYN Flodding	1543ms

V. CONCLUSION

Applying mobile agent in intrusion detection system, the architecture of the distributed intrusion detection system based on mobile agent was implemented in the paper. Moreover, internal agents function was schemed. The information exchange method was discussed and the optimal dynamic agent migration algorithm was developed as well. Furthermore, the BM algorithm was improved and the process of intrusion detection was described. Finally, simulation of the system was practiced. The simulation result shows that: the system enhanced the ability of defending attack and intrusion detection. The single-point failure and false alarm of tradition intrusion detection system were eliminated. Adapting dynamically and executing synchronously and autonomously are achieved by mobile agent. From

system architecture, robustness and capacity of fault tolerance can be enhanced.

REFERENCES

[1] Álvaro Herreroa, Emilio Corchadoa, María A. Pellicera and Ajith Abrahamb, "MOVIH-IDS: A mobile-visualization hybrid intrusion detection system," *Hybrid Learning Machines (HAIS 2007) / Recent Developments in Natural Computation (ICNC 2007)*, vol. 72, pp. 2775–2784, August 2009. doi:10.1016/j.neucom.2008.12.033

[2] Lange D, Oshima M, "Seven Good Reasons for Mobile Agents," *Communications of the ACM*, vol. 42, pp. 88–89, March 1999. doi:10.1145/295685.298136

[3] M. Ali Aydın, A. Halim Zaim and K. Gökhan Ceylan, "A hybrid intrusion detection system design for computer network security," *Computers & Electrical Engineering*, vol. 35, pp. 517–526, May 2009. doi:10.1016/j.compeleceng.2008.12.005

[4] Ming-Yang Sua, Gwo-Jong Yub and Chun-Yuen Lin, "A real-time network intrusion detection system for large-scale attacks based on an incremental mining approach," *Computers & Security*, vol. 28, pp. 301–309, July 2009. doi:10.1016/j.cose.2008.12.001

[5] Adrian P. Lauf, Richard A. Peters and William H. Robinson, "A distributed intrusion detection system for resource-constrained devices in ad-hoc networks," *Ad Hoc Networks*, vol. 8, pp. 253–266, May 2010. doi:10.1016/j.adhoc.2009.08.002

[6] Xiaojun Tonga, Zhu Wangb and Haining Yua, "A research using hybrid RBF/Elman neural networks for intrusion detection system secure model," *Computer Physics Communications*, vol. 180, pp. 1795–1801, October 2009. doi:10.1016/j.cpc.2009.05.004

[7] Tomás E. Uribe, Steven Cheung, "Automatic analysis of firewall and network intrusion detection system configurations," *Journal of Computer Security*, vol. 15, pp. 691–715, September 2007.

[8] Abhay Nath Singh, Shiv Kumar and R. C. Joshi, "Intrusion Detection System Based on Real Time Rule Accession and Honeypot," *Communications in Computer and Information Science*, vol. 196, pp. 292–301, July 2011. doi:10.1007/978-3-642-22540-6

[9] Kuo-Hsuan Huanga, Yu-Fang Chungb, Chia-Hui Liua and Feipei Laia, el al., "Efficient migration for mobile computing in distributed networks," *Computer Standards & Interfaces*, vol. 31, pp. 40–47, January 2009.

[10] Leena Salmela, Jorma Tarhio and Petri Kalsi, "Approximate Boyer-Moore String Matching for Small Alphabets," *Algorithmica*, vol. 58, pp. 591–609, February 2009. doi:10.1007/s00453-009-9286-3

[11] Fabio Bellifeminea, Giovanni Cairea, Agostino Poggib and Giovanni Rimassac, "JADE: A software framework for developing multi-agent applications. Lessons learned," *Information and Software Technology*, vol. 50, pp. 10–21, January 2008. doi:10.1016/j.infsof.2007.10.008

[12] E. Mosqueira-Rey, Alonso-Betanzos, Guijarro-Berdinas, Alonso-Rios A4, el al, "A Snort-based agent for a JADE multi-agent intrusion detection system," *International Journal of Intelligent Information and Database Systems*, vol. 3, pp. 107–121, February 2009. doi:10.1504/IJIIDS.2009.023041

[13] Guy Helmer, Johnny S.K. Wong, Vasant Honavar and Les Miller et al, "Lightweight agents for intrusion detection," *Journal of Systems and Software*, vol. 67, pp. 109–122, August 2003. doi:10.1016/S0164-1212(02)00092-4

[14] Shmuel T. Klein, Miri Kopel Ben-Nissan, "Accelerating Boyer–Moore searches on binary texts," *Theoretical Computer Science*, vol. 410, pp. 3563–3571, September 2009. doi:10.1016/j.tcs.2009.03.019

[15] Vaibhav Gowadia, Csilla Farkas and Marco Valtorta, "PAID: A Probabilistic Agent-Based Intrusion Detection system," *Computers & Security*, vol. 24, pp. 529–545, October 2005. doi:10.1016/j.cose.2005.06.008

[16] Yu Cai, "Mobile Agent Based Network Defense System in Enterprise Network," *International Journal of Handheld Computing Research*, vol. 2, pp. 41–54, March 2011. doi:10.4018/jhcr.2011010103