AGENT-BASED INTRUSION DETECTION SYSTEM FOR INTEGRATION

Jianping Zeng, Donghui Guo*
Department of Physics, Xiamen University, Xiamen, Fujian, P.R. China

Keywords: Intrusion Detection, Agent-based, KQML, IDMEF.

Abstract: More and more application services are provided and distributed over the Internet for public access. However, the security of distributed application severs is becoming a serious problem due to many possible attacks, such as deny of service, illegal intrusion, etc. Because of weakness of the firewall systems in ensuring security, intrusion detection system (IDS) becomes popular. Now, many kinds of IDS systems are available for serving in the Internet distributed system, but these systems mainly concentrate on network-based and host-based detection. It is inconvenience to integrate these systems to distributed application servers for application-based intrusion detection. An agent-based IDS that can be smoothly integrated into applications of enterprise information systems is proposed in this paper. We will introduce its system architecture, agent structure, integration mechanism, and etc. In such an IDS system, there are three kinds of agents, i.e. client agent, server agent and communication agent. This paper is also to explain how to integrate agents with access control model for getting better security performance. By introducing standard protocol such as KQML, IDMEF into the design of agent, our agent-based IDS shows much more flexible for built in different kinds of software application system.

1 INTRODUCTION

Many applications, such as e-business, remote education and Internet-based design, are becoming popular. As you can see, it becomes a trend to deploy software application systems in the Internet to make full use of its all kinds of advantages. However, because the Internet is an open society so that anyone can access the resource on it, then the application may confront with all kinds of attacks or intrusions, such as Deny of Service (DoS), port scan, illegal intrusion by hacking user information, etc.

Of all these security events, illegal intrusion is a more serious issue. But standard security deployments such as firewalls are limited in their effectiveness because of simple access control mode and the evolving sophistication of intrusion methods. Once an attacker has breached the firewall, he can roam at will through the network (Loshin, P., 2001). This makes intrusion detection very important and necessary. Traditionally, there have been two main classes of IDSs: host-based and network-based systems. A host-based IDS monitors the detailed activity of a particular host, while network-based IDS monitors networks of computers and other devices such as, routers, gateways, and primarily uses data sniffing from network traffic. Network and host-based IDSs, can be further classified based on two methods of detection (B. Mukherjee, T. L. Heberlein, and K. N. Levitt, 1994): anomaly detection and misuse detection.

However, Masquerading attack is a typical intrusion and it can be a more serious threat to the security of computer systems and the computational infrastructure (Roy A. Maxion, and Tahlia N. Townsend, 2004). By this kind of attacks, an assailant attempts to impersonate a legitimate user after gaining access to this legitimate user's account. So, the assailant can understand the exact meanings of the information he gets, while by other kinds of attacks, he may just get some segment of data or encrypted data. A well-known instance of masquerader activity is about a FBI mole (Roy A. Maxion, and Tahlia N. Townsend, 2004). Since masquerading attack happens exactly at application layer, we call detection methods applied to it as application-based intrusion detection. And application-based detection is more difficult to achieve for the following reasons:

(1) Data of user action are difficult to be obtained compare to network-based and host-based

* Correspondence: Prof. Donghui Guo,

Zeng J. and Guo D. (2005).
AGENT-BASED INTRUSION DETECTION SYSTEM FOR INTEGRATION.
In *Proceedings of the Seventh International Conference on Enterprise Information Systems*, pages 176-181
DOI: 10.5220/0002516401760181

detection, because this kind of data can only be got from application, while the independency of application makes an application-based user action more difficult to acquire.

(2) Application-based detection can affect the performance of corresponding application due to the extra work that should be done on data generating and analysis.

(3) A masquerader may happen to have similar behavioral patterns as the legitimate user of an account to which he or she is currently logged, therefore escaping detection and successfully causing damage under the cover of seemingly normal behaviour (Coull, S.; Branch, J.; Szymanski, B.; Breimer, E., 2003).

Fortunately, There have been several attempts to tackle the problem of detecting masqueraders. Several masquerade-detection algorithms, such as Sequence-Match, IPAM, Bayes 1-Step Markov, etc are presented by Schonlau and his colleagues (M. Schonlau, W. DuMouchel, W.-H. Ju, A. F. Karr, M. Theus, and Y. Vardi, 2001). However, researches on data acquiring and overall system structure have seldom to mention. So, in this paper, we mainly focus on these topics.

An agent-based intrusion detection system is proposed. Being different from other agent-based IDSs, this system can be integrated with enterprise information system very well. The system mainly consists of three kinds of agents: client agent, server agent and communication agent. The system architecture, agent structure, integration mechanism, etc, are mainly discussed. And we explain how to integrate agents with access control model to achieve better security performance.

The paper is structured as follows. Section 2 describes related work and discusses previous efforts to utilize agent techniques for intrusion detection. Section 3 introduces our system that can be integrated with actual application and performs application-oriented intrusion detection. Section 4 discusses the overall system performance and design issues. Finally, Section 5 briefly concludes.

2 RELATED WORK

Agent technology is a very active field of distributed artificial intelligence (DAI) research in the recent years. A software agent can be defined as (J. M. Bradshaw,1997):

It's a software entity that functions continuously and autonomously in a particular environment, and is able to carry out activities in a flexible and intelligent manner that is responsive to changes in the environment. Ideally, an agent that functions continuously would be able to learn from its experience.

Although agent may not improve the techniques for intrusion detection directly, it can change the means of applying detection techniques, which will lead to high efficiency and validity of intrusion detection. Recently, there has been an accretion of approaches for building agent-based IDSs for Internet applications. In such a system, agents with differing capabilities can interact with each other to perform data acquiring, analysis, reporting. Several related agent-based IDSs can be listed as follows:

(1) Autonomous Agents For Intrusion Detection (AAFID) (Balasubramaniyan, J.S.; Garcia-Fernandez, J.O.; Isacoff, D.; Spafford, E.; Zamboni, D., 1998)

The AAFID system consists of three essential components: agents, transceivers and monitors. Agents are used as the lowest-level element for data collection. All agents in a host report their findings to a single transceiver. However, the communication between agents or transceivers is based on SNMP, System V IPC, this may lead to extra work when developing a new agent.

(2) Multi-agent based intrusion detection system (Hegazy, I.M.; Al-Arif, T.; Fayed, Z.T.; Faheem, H.M, 2003)

The system employs sniffing agent, analysis agent, decision agent and report agent to detect three kinds of attack: the Denial of Service attack, the ping swept attack and the secure coded document theft. It can be treated as a network-based detection system.

(3) An intelligent agent security intrusion system (Pikoulas, J.; Buchanan, W.; Mannion, M.; Triantafyllopoulos, K, 2002)

The system is an application-based detection one. It utilizes the Bayesian multivariate statistical model to predict user action. In such a system, a user agent resides in a user workstation, while a core agent resides on the server. User agent monitors and analyzes the user action according user profile file downloaded from server via core agent. The file contains rules that describe the legal past behavior of the user and the statistical predictions. Therefore, the file must be kept secure by itself, however, it's difficult to ensure this in this system.

Although these IDSs may be effective in detecting some kinds of intrusion, but how to integrate them with all kinds of application in enterprise information system remains a difficult problem.

3 AGENT-BASED INTRUSION DETECTION SYSTEM FOR INTEGRATION

3.1 System Architecture

We propose an architecture (which we call AIDSI for 'Agent-based Intrusion Detection System for Integration') for building IDSs that can be distributed over Internet-based applications. In this system, we use Markov model to predict the probability of user action and then make a fuzzy decision whether the user is illegal one. The system consists of three types of agents: client agent, communication agent and server agent. The relationship among them is shown in figure 1. All of the configuration and the information necessary for prediction and decision are stored in the knowledge base, and the base is located at the server side.

Client agents are installed on client workstation, and responsible for collecting extra user information and then sending to server agents with the help of communication agents. Also, client agents will check the newest version of client agents stored on server and decide whether to download and update for themselves.

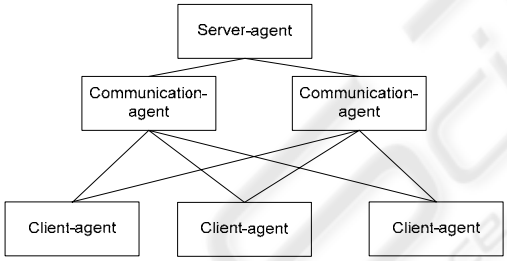

Figure 1: Relationship among agents

Server agents are running in the server process. They process the message sent from client agents, write and read from knowledge base, which stores user information and prediction model presentation. However, most important of all, server agents can inference whether the current user is a legal one or not according to the prediction model.

Communication agents monitor the client agent's sending request and check whether the client agent is a legal one. After the received message is parsed, the useful message is forwarded to the server agents. Also, the communication agents can accept other message that generated from access control module or user authentication module.

Because the system can run well in client process and work well with other security services, such as access control, and exchange message in process-level with those services. Thus the response time will greatly decrease and a more security mechanism will be achieved.

In the following section we describe each component in more detail.

3.2 Client Agent

Application-based intrusion detection system should set up a proper model for normal user, and Markov model is used in this paper. However, in order to achieve an excellent prediction accuracy, only knowledge about user's action on the application is not enough. So, the main function of client agent is to collect extra user authentication information. However, this raises many privacy concerns (Millett, L.I.; Holden, S.H., 2003), which is a hot topic recently. In AIDSI, the client agent gets extra information and sends to the server. This process needn't to be disturbed by user or system administrators, so the user privacy can be kept. Extra information about client's machine, such as Operating System, network card, etc. is used to improve the decision accuracy.

The structure of client agent can be described in figure 2. Layer 1 collects extra information by calling operating system API when the user login into the application. Layer 2 translates the message into special format defined by Knowledge Query and Manipulation Language (KQML), which is a relatively mature agent communication language (Liu Yong, Xu Congfu, Chen Weidong, Pan Yunhe, 2004.). Layer 3 simply transmits the KQML message to remote server.

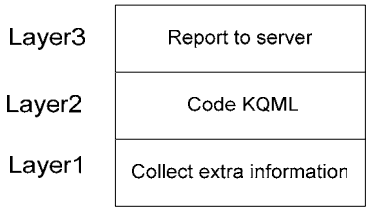

Figure 2: Structure of client agent

However, the client agent should identify itself for the server agent that it is a legal agent after connecting to the server. This is especially important in such an IDS, because abnormal user may corrupt the client agent and report wrong message to server. To achieve this goal, a distinguish ID code is assigned to each client agent. But when the user login to server in different workstation, he has to

install the client agent on every workstations. To avoid this kind of unconvenience, the client agent is designed to dynamically download from server after the user logins to server in his own application software.

Another important feature of the client agent is reliability in communication. Each client agent is connected to a default communication agent. However, when the communication is busy or fails to response, the client agent may select from local setting for communication agent lists and then tries to connect to another communication agent.

From above, we can see that the client agent do not exchange more with user application, so it can be well integrated with many kinds of user application.

3.3 Communication Agent

Firstly, the communication agent performs an ordinal check to see whether the client agent is a legal one. The checking process describes as follow:

(1) the communication agent gets the client agent ID code;

(2) the communication agent checks if the ID is contained in list of knowledge base; if so, then it allows client to connect.

(3) however, if the ID is not in list, the communication agent may give a message to the user authentication module of user application to notify the module to try another user session.

Secondly, it monitors the data sent from the instance of access control model.

In application-based detection, user action can be acquired from access control model and the user authentication module. Here we take access control model as an example.

Because access control model usually embeds in application system, IDSs can't get the data flow from the instance of the access control model. Although an access control model usually produces log file which records details about user action, but this log file can't be used to perform real-time detection. So, we define a standard event data acquiring protocol based on Intrusion Detection Message Exchange Format (IDMEF). IDMEF (D. Curry, H. Debar, M. Huang, 2000) is a XML-based language to describe intrusion events. An example message about user's access is as follow:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE IDMEF-Message PUBLIC
 "-//IETF//DTD IDMEF v0.1//EN""idmef-message.dtd">
<IDMEF-Message version="0.1">
 <Alert alertid="101" impact="attempted-user">
  <Time offset="0">
   <date>2004/10/09</date>
    <time>10:08:07</time>
  </Time>
  <Analyzer ident="800">
   <Node category="wfw">
    <name>myserver.eda</name>
   </Node>
  </Analyzer>
  <Source>
   <Node>
    <Address category="ipv4-addr">
     <address>192.168.1.90</address>
    </Address>
   </Node>
  </Source>
 </Alert>
</IDMEF-Message>
```

However, in order to keep the expansibility of AIDSI, all kinds of IDMEF message should be also encapsulated into KQML package by assigning IDMEF as the content part of KQML package.

3.4 Server Agent

The server agent has several main tasks to be performed, described as follows:

(1) set up a suitable model for each user

In order to improve the overall effectiveness, the AIDSI defines different models for each user. The model is express by the following Markov:

$MC = (X, A, \Psi)$, where X is a discrete stochastic variable, A is the probability transfer matrix and Ψ is the initial distribution of X.

$$A = \begin{bmatrix} p_{11} & p_{12} & & p_{1m} \\ p_{21} & & & \\ & & & \\ p_{m1} & & & p_{mm} \end{bmatrix},$$

where P_{ij} stands for the probability from status i to status j.

So, setting up the model actually should performs the calculation the values of P_{ij}.

The model is stored at the knowledge base, which can be a small relational database.

(2) acquire the user action at a certain time

After the communication agent gets the KQML package about user action from the instance of

access control model, the server agent can simply perform a standard action on KQML package to acquire the user action in real-time.

(3) predict the probability of user action and make decision

This is calculated by n-order Markov by the following equation:

$$V(t+1) = a_1 \times H(t) \times A(1) + a_2 \times H(t-1) \times A(2) + \cdots + a_n \times H(t-n+1) \times A(n)$$

where, $\sum_{i=1}^{n} a_i = 1$,

$H(t)$ stands for user status at time t

Then, the predict result can be got from V(t+1) by checking the maximum value.

However, the predict accuracy may be limited. AIDSI makes full use of extra information collected by the client agents. And this extra information reflects the user application environment, so a more high accuracy can be achieved.

The following explains how to get the result.

Suppose V(t+1) = (0.54 0.43 0.03 0) and the user's current action is on object 1.

Because the probability difference between object 1 (0.54) and object 2 (0.43) are small, we may not make such decision that the user is a legal or an illegal. So, extra information is used. For example, AIDSI can check whether the address of network card is in the lists of history frequent used address. If not, the user may be illegal.

(4) maintenance the user model

In order to keep the user model in a correct state, each variable in probability transfer matrix should be updated every time. And this task is performed by the server agent.

4 DISCUSSION AND DESIGN ISSUES

(1) ability for integration

The AIDSI is an agent-based intrusion detection system, which is easy to be integrated.

The client agent just acts as an extra information collector, and it need not exchange data with client of user application. In system implementation, the client agent can be written in Java language and a complied language, such as C. Layer 1 of client agent is written in C to integrate with operation system API, while layer 2 and layer 3 are developed by Java to reach the goal of dynamically download to update itself.

The communication agent and server agent are written in Java. And they can be deployed at different computers. The instance of access control model should send information about user action to communication agent, so some code can be written as DLL and export API for user application. By doing so, the agent and access control module is loosely coupled.

(2) inference ability

In AIDSI, the inference ability of agent is important for accuracy detecting application-oriented intrusion. Because the user's action in application forms a time series, Markov model is suitable for such kind of analysis. And extra user information is used to make a final decision when the predict result of Markov model is in confuse state.

Because the calculation in the inference process is much more expensive, so, multi-thread program technology must be introduced when developing server agent to ensure the overall performance of user application.

(3) independency

Independency is especially a main requirement in AIDSI for actual user application may vary from each other in the software environment, and this is achieved by introducing standard protocol, such as KQML, IDMEF. These protocols are application-oriented, so they are suitable for AIDSI.

5 CONCLUSION

Application-oriented intrusion is a kind of more serious intrusion since masqueraders can get the meaning of data in a visual mode. AIDSI, which is different from other IDS, is proposed to deal with such intrusion. Agent technology is introduced to get extra user information to improve the predict accuracy without influence user's privacy. Also standard protocols, such as KQML, IDMEF, are used in the design of the agent communication mechanism, so AIDSI is more flexible and can be integrated with enterprise information systems well.

ACKNOWLODGEMENT

This paper is supported by the funding of Fujian province of China NSFC Project No. A0410007 and Corporation Start-up Project. We would like to thank the anonymous reviewers for their comments,

which is helpful to improve the quality of the paper.

REFERENCES

Loshin, P., 2001. Intrusion detection. Computer World. http://www.computerworld.com/hardwaretopics/hardware/story/0,10801,59611,00.html.

Roy A. Maxion, and Tahlia N. Townsend, 2004. Masquerade Detection Augmented With Error Analysis. IEEE TRANSACTIONS ON RELIABILITY, VOL. 53, NO. 1, MARCH 2004

B. Mukherjee, T. L. Heberlein, and K. N. Levitt. 1994. Network intrusion detection. IEEE Network, 8(3):26–41, May/June 1994.

Coull, S.; Branch, J.; Szymanski, B.; Breimer, E.; 2003. Intrusion detection: a bioinformatics approach. Proceedings on Computer Security Applications Conference, 19th Annual, 2003,Pages:24 – 33

M. Schonlau, W. DuMouchel, W.-H. Ju, A. F. Karr, M. Theus, and Y. Vardi, 2001. "Computer intrusion: Detecting masquerades," Statistical Science, vol. 16, no. 1, pp. 58–74, Feb. 2001.

J. M. Bradshaw.1997. An introduction to software agents. In J. M. Bradshaw, editor, Software Agents, chapter 1. AAAI Press/The MIT Press, 1997.

Balasubramaniyan, J.S.; Garcia-Fernandez, J.O.; Isacoff, D.; Spafford, E.; Zamboni, D.; 1998. An architecture for intrusion detection using autonomous agents. Proceedings on Computer Security Applications Conference, 14th Annual , 7-11 Dec. 1998 Pages:13 – 24

Hegazy, I.M.; Al-Arif, T.; Fayed, Z.T.; Faheem, H.M.; 2003. A multi-agent based system for intrusion detection. Potentials, IEEE , Volume: 22 , Issue: 4 , Oct.-Nov. 2003 Pages:28 – 31

Pikoulas, J.; Buchanan, W.; Mannion, M.; Triantafyllopoulos, K.; 2002. An intelligent agent security intrusion system. Proceedings on IEEE International Conference and Workshop on the Engineering of Computer-Based Systems, Ninth Annual, 8-11 April 2002 Pages:94 – 99

Liu Yong; Xu Congfu; Chen Weidong; Pan Yunhe; KQML realization algorithms for agent communication. Fifth World Congress on Intelligent Control and Automation, 2004. WCICA 2004.Volume: 3 , June 15-19, 2004 Pages:2376 - 2379

D. Curry, H. Debar, M. Huang. 2000. IDMEF Data Model and XML DTD http://www.oasis-open.org/cover/IDMEF-provisional-draft-ietf-idwg-idmef-xml-02.html , DEC 5,2000

Millett, L.I.; Holden, S.H.; 2003. Authentication and its privacy effects. Internet Computing, IEEE , Volume: 7 , Issue: 6 , Nov.-Dec. 2003 Pages:54 - 58