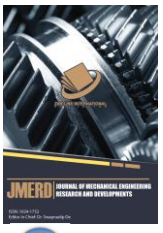
ZIBELINE INTERNATIONAL PUBLISHING

ISSN: 1024-1752 (Print)

CODEN: JERDFO

Journal of Mechanical Engineering Research & Developments (JMERD)

DOI: *http://doi.org/10.26480/jmerd.02.2019.14.17*



RESEARCH ARTICLE

NETWORK INTRUSIONS DETECTION AND PREVENTION METHOD USING A TEAM OF INTELLIGENT AGENTS

Aleksei A. Sychugov[1*], Vasiliy Yu. Meltsov[2], Alexey S. Kuvaev[3], Vyacheslav M. Grishin[4]

[1]Department of Information Security, Tula State University, 92 Lenin Ave., Tula, Russian Federation
[2,3]Department of Electronic Computing Machines, Vyatka State University, 36 Moskovskaya Str., Kirov, Russian Federation
[4]Department No. 604 "System Analysis and Management", Moscow Aviation Institute, 4 Volokolamskoe shosse, Moscow, Russian Federation
[*]Corresponding Author Email: xru2003@list.ru

ARTICLE DETAILS

Article History:

Received 01 February 2019
Accepted 11 March 2019
Available online 14 March 2019

ABSTRACT

An intrusion is defined to be a violation of the security policy of the system; intrusion detection thus refers to the mechanisms that are developed to detect violations of system security policy. Intrusion detection is based on the assumption that intrusive activities are noticeably different from normal system activities and thus detectable. Also, there is a specific set of characteristics that can serve as a sign of anomalies caused by unauthorized intrusions. The authors introduced a multi-agent system for network attack detection with the help of the detecting of computer network malfunctions, caused by an unauthorized intrusion into the network. The study has showed that the proposed system can be applied to detect anomalies in real-time mode, which is the major advantage of the proposed system.

KEYWORDS

Network attacks, rapid anomaly detection, proposed method, communication, incoming agent.

1. INTRODUCTION

Network data resource security is provided with the help of firewalls, antivirus solutions, attack detection systems, integrity monitoring systems and encryption. All of these systems are characterized either by their periodic or short-term application for solving a particular problem, or by permanent use with static settings [1]. As a result, the analysis methods used in such systems for the detection of strictly specified network intrusion types turn out to be inefficient in case of yet unknown intrusion types or modifications of the existing ones.

The object of the research is the problem of protection of network data resource that is provided with the help of firewalls, antivirus solutions, attack detection systems, integrity monitoring systems and encryption. The aim of this work is to find the methods to detect forbidden events in the network – that is to say, the anomalies caused by unauthorized intrusions. One of the major requirements for such methods is the ability to detect unspecified anomalies, including new and distributed ones.

Most of the network attacks can be conditionally divided into three types: 1) "reconnaissance" attack aimed at gaining information about the targeted computer (ping sweep, DNS zone transfer, TCP or DNS ports scanning, E-mail reconnaissance, etc.); 2) "exploit", which is a piece of software attacking software vulnerabilities (the aim is to gain control over the system, to cause its malfunction, etc.); 3) "denial-of-service" attacks by means of unauthorized overloading of CPU, exhausting hard drive storage space or bringing down the services. The attacks may originate either from a single source or from several ones [2]. Distributed attacks constitute a type of "denial-of-service" attacks, for example, flood and storm attacks, remote intrusion (NetBus, BackOrifice), local intrusion (GetAdmin), remote denial of service (Teardrop, trin00) or local denial of service.

The authors propose to solve problem of protection of network data resource with the help of multi-agent system – a number of intelligent agents ("software robots") distributed over the network, which move around it in search of relevant data, knowledge and procedures, and cooperate in solving emerging problems. The authors found that to detect network attacks the agent has to have some software components like autonomy, cooperativity, fast response, proactivity, awareness, sense of purpose and rationality.

The authors introduced a method for network attack detection with the help of the detection of computer network malfunctions, caused by an unauthorized intrusion into the network. The study has showed that the proposed method can be applied to detect anomalies in real-time mode, which is the major advantage of the method. Method implementation proved that its efficiency depends on the chosen programming language.

There is a huge number of network intrusions (like data fragmentation, ping flooding attack, smurf attack, replay attack, IP spoofing) and that number is constantly growing. Typically, the recognition of computer attacks in the dynamics of the functioning of the information system can be represented on the basis of a system analysis of the parameters space of processes in the system according to established rules and the identification of those parameters that characterize the action of the attack. In turn, a systematic description of ways and means of protecting information aimed at countering computer attacks, it is rational to implement on the basis of the theory of pattern recognition, according to which objects of computer attacks can be interpreted by recognizable images of the space of their signs [3,4].

Also, there is a specific set of characteristics that can serve as a sign of anomalies caused by unauthorized intrusions: repeat certain events, actions; unanticipated parameters in network packets; inadequate network traffic parameters; inadequate attributes of the functioning of the system; an attempt to identify and exploit vulnerabilities; suspicious network traffic (Figure 1).

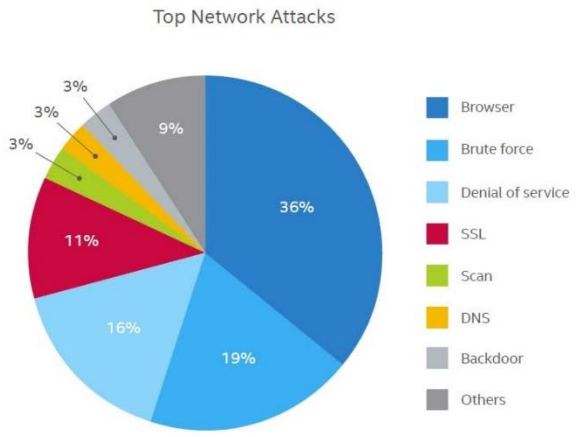


Figure 1: Top network attacks, 2016

Therefore, the main task is to provide rapid and automatic detection of these anomalies. For this purpose, the authors developed the method for network attack detection.

2. METHODOLOGY

2.1 Multi-agent system of detection of network attacks

This article investigates problem solution with the help of multi-agent system – that is to say, a number of intelligent agents ("software robots") distributed over the network, moving around it in search of relevant data, knowledge and procedures and cooperating while solving the problems [5]. An agent is an entity in the environment, providing data, capable of reflecting and interpreting the events in that environment and of executing commands. In order to detect network attacks the agent needs to have software components, which allow having the following characteristics:

1) autonomy (the ability to function with total self-control of the actions and internal state);

2) cooperativity (the ability to cooperate with other agents, exchanging messages encoded with the help of a common language);

3) fast response (the ability to perceive the state of the environment and response to the changes in due time);

4) proactivity (the ability to take up the running, that is the ability to generate tasks and act rationally);

5) awareness (possession of permanent knowledge, unchangeable during agent's activity and of operational knowledge, which can vary with time and even become false);

6) sense of purpose (the ability to plan the actions, aimed at achieving goals, that is the required states and situations);

7) rationality – the ability to act in such a way as to achieve the goals with minimal time and resources.

There are two types of problems to be solved for the purpose of formal description of mental notions: syntactic and semantic ones. Knowledge formalism (as well as the description of any other information) should have the following two aspects: its own language for formal description and its own semantic model [6]. It is proposed to apply fuzzy logic to the description of syntax. However due to the dynamic nature of agent's activity and multi-agent system, this logic should be supplemented by description methods applied to temporal aspects of knowledge, and in some cases by means of description of real-time characteristics. From the point of view of formalization semantics, symbolic structures can be interpreted with the help of corresponding functions (algorithms) and data structures.

2.2 Method for design a multi-agent system

The distributed solution for network anomaly detection by several agents can be divided into the following stages:

1) agent-manager (central agent) decomposes the original task into separate tasks, which are distributed among executive agents;

2) each executive agent solves its task, also dividing it into subtasks in some cases;

3) separate results for the selected tasks are integrated to get a general solution.

We offer the following fuzzy relational model to be used as the basis of the executive agent:

$$\mathrm{R}:\mathrm{A}\to\mathrm{B},\quad R=\begin{pmatrix} r_{11} & r_{12} & \cdots & r_{1p} \\ r_{21} & r_{22} & \cdots & r_{2p} \\ \cdots & \cdots & r_{jk} & \cdots \\ r_{m1} & r_{m2} & \cdots & r_{mp} \end{pmatrix}, \tag{1}$$

where $A=\{A_1,A_2,...,A_m\}$ is a set of linguistic terms, defined on X with membership functions $\mu_{\mathrm{A_i}}(x)\in[0,1]$ for *j=1,...,m*; $x=(x_1,x_2,...,x_m)$ is the vector of fuzzy current parameters of network operation; $\mathrm{B}=\{B_1,B_2,...,B_m\}$ is a set of linguistic terms defined on Y with membership functions $\mu_{\mathrm{B_k}}(\mathrm{y})\in[0,1]$ for *k=1,...,p*; y is a fuzzy output variable with a value ranging from 0 to 1 – this value estimates the presence of anomalies; $r_{jk}\in[0,1]$, *j=1,...,m, k=1,...,p*.

Table 1: The process of finding a cooperative solution

No.	Stage	Description
1.	Detection	Cooperative solution finding process starts at the moment when the agent detects the reasonability of such actions. For example, in case when the agent lacks information to ensure the reliability of the logical inference.
2.	Agent teaming	At this stage the agent, that has detected the feasibility of cooperation, is searching for partners. If this stage is successfully completed, a team of agents with equal responsibilities in collective actions is formed.
3.	Collective planning	At this stage the agents exchange the information (messages) in order to work out a collective plan aimed at the realization of the required goal.
4.	Collective actions	At this stage the agents are acting according to the obtained plan, interacting within the protocol.

When obtaining the term values of $A'=\{A'_1,A'_2,...,A'_m\}$ - a fuzzy set, reflecting the vector of values of different variables, including the symbolic ones, with the values of membership functions $\mu_{\mathrm{A'_i}}(x)\in[0,1]$ allows estimating the anomaly resulting from the fuzzy inference:

$$\mathrm{B}'=\mathrm{A}'\circ(\mathrm{A}\to\mathrm{B}) \tag{2}$$

The process of finding a cooperative solution can be divided into four stages (Table 1).

Detection is based upon the estimation of agent cooperation potential [7]. There is a cooperation potential in relation to the goal g of an agent i if: (1) there is a team q, such that i believes that q can collectively gain g and either (2) i cannot reach g acting separately or (3) i believes that for each action *a*, that could be performed in order to get a goal g it has another goal resulting in failure to perform action *a*. Team formation procedure is as follows: agent i (with goal g), that has a potential for cooperation with the

team q, is trying to reach such state of the team q in which it could collectively get the goal g and in which the team q is obliged to perform the actions collectively [8].

Collective planning starts if the previous stage was successful. Then there is a team of agents, obliged to perform the actions collectively. However collective operation cannot start until the team agrees upon the specific functions of each agent. Collective actions begin at the end the previous stage. In the initial state of collective actions stage there is a general plan of the team q and it (q) intends to continue working together. Normally the actions during this process are performed according to the accepted plan until the moment of its completion. However sometimes collective actions can be interrupted. The system is managed only due to the local interactions among the agents. A coordinating agent is needed to support such management.

The coordinating agent can be attached to AMP (Agent Meeting Place). AMP is the agent acting as a broker between the agents, requesting some resources belonging to other agents and the agents that can provide these resources. It is a regular agent, supplemented by auxiliary components. These auxiliary components should contain a unified description of agents available at AMP and their abilities (resources, functions, etc.) on the one hand and on the other hand they should provide a unified access to them [9]. These conditions are guaranteed by the following AMP components:

1) Basic services; these can include remote invoking of the objects, their systematization, doubling and other basic options.

2) Linked ports for receiving and sending agents to AMP with the help of relevant protocols.

3) Agent name authentication component (agent identification, "authorization").

4) Access subsystem, monitoring the functions of the incoming agent, presence of the required service at AMP and helping the agent select further route, etc.

5) Surface router, acting as an interface between the agents and AMP components, registered in this interface. This router has a limited vocabulary to satisfy agents' requests.

6) Linguistic log which is a database helping agents and AMP successfully communicate. The log registers vocabularies and languages, but instead of languages' description or terms' definitions there are links to them, i.e. the log provides the information that can possibly be understood at AMP.

7) Resource manager registers the agents at AMP and their resources and manages AMP resources.

8) Agent execution environment registered at AMP. It controls the access to agent's components, analyzes scripts and provides access to the basic features, etc.

9) Message delivery system; any local tools, resident agents of AMP, etc. can send the messages; the system logs the events and searches for the agents for specific type of events or messages.

3. RESULTS AND DISCUSSION

In this article proposed the multi-agent system for network attack detection (Figure 2) has a "knowledge-based architecture". The following requirements define the features of the proposed architecture:

1) In the real application the agent deals with unforeseen events in space (network) and time and in the presence of other agents.

2) The agent has to retain the ability to respond adequately and to make decisions.

3) The agent must have an architecture that would allow coping with uncertainty and lack of information, responding to unforeseen events using relatively simple rules [10].

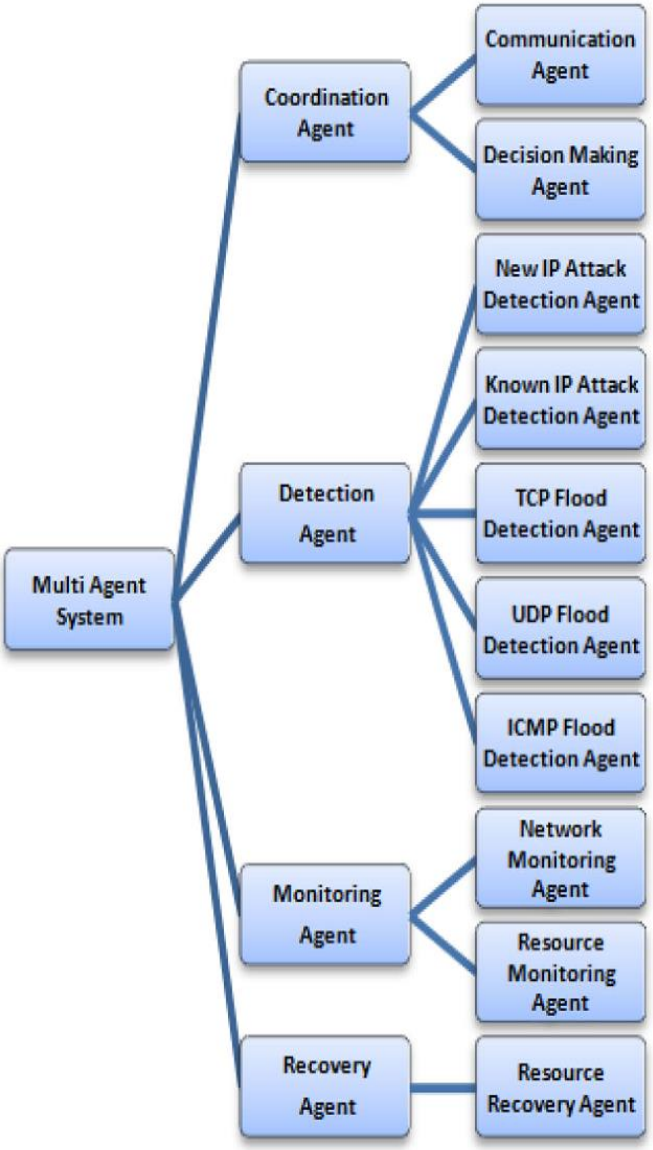


Figure 2: Multi-agent system for network attack detection

The agent's competence in such architecture is based upon the following types of knowledge:

1) knowledge of the network;

2) knowledge of agents' capabilities;

3) knowledge of other agents' capabilities;

4) knowledge of interaction with network nodes;

5) knowledge of communication with other agents (which types of communication are feasible and useful for obtaining additional information) [11-13].

The architecture includes three levels, with each one corresponding to different types of agent's capabilities (each of these levels has its own features):

1) event response level helps maintain agent's ability to respond to the events coming from upper levels, even if they were not planned;

2) planning level generates, executes and dynamically reconstructs partial plans, for example, for choosing mobile robot's route;

3) expectance and modeling level simulate the behavior of external environment entities and of the agent itself, which can be used for interpretation of the observed behavior and prediction of a possible one in future.

4. CONCLUSION

The proposed system is implemented as a combination of message exchange technology and a contextual activation of controlling rules (according to the characteristics of the subject area), acting as an intermediary examining data at different levels (perceived input and output data of different levels), adding new data at different levels and deleting some old one. In fact, subsystem rules act as a filter between the sensors of the agent and its internal levels as well as between the levels and the executive elements. Other components of the analyzed system consist of intermediate memory, communication manager and man-machine interface.

Intermediate memory serves to store user's and communication manager's current data, generated by controlling level. There are the following types of information stored in the memory: goals to be achieved, state of the tasks in process of making agreements with other agents. Experimental studies of the system have showed: 1) rather high accuracy of anomaly detection (96% on average); 2) rather wide range of detected anomalies. To estimate the proposed method, we used KDD Cup 1999 Data. – kdd.ics.uci.edu/databases/kddcup99/ kddcup99.html.

This article has introduced a method for network attack detection with the help of the detection of computer network malfunctions, caused likely by an unauthorized intrusion into the network. The study has showed that the proposed method can be applied to detect anomalies in real-time mode, which is the major advantage of the method. Method implementation proved that its efficiency depends on the chosen programming language. The choice is restricted by the requirements to support: 1) network interaction, 2) multithread processing, 3) symbolic computation, and to consider procedure calls as messages between the objects.

REFERENCES

[1] Ayatollahi, H., Khansari, M., Rabie, H.R. 2017. A push-pull network coding protocol for live peer-to-peer streaming. Computer Networks, 130, 145-155.

[2] Aslam, S., ul Islam, S., Khan, A., Ahmed, M., Akhundzada, A., Khan, M.K. 2017. Information collection centric techniques for cloud resource management: taxonomy, analysis and challenges. Journal of Network and Computer Applications, 100(15), 80-94.

[3] Ha, T., Yoon, S., Risdianto, A.C., Kim, J.W., Lim, H. 2016. Suspicious flow forwarding for multiple intrusion detection systems on software-defined networks. IEEE Network, 30(6), 22-27.

[4] Vieira, T.P.B., Tenório, D.F., Da Costa, J., Freitas, E., Del Galdo, G., De Sousa Júnior, R.T. 2017. Model order selection and Eigen similarity-based framework for detection and identification of network attacks. Journal of Network and Computer Applications, 90(15), 26-41.

[5] Russell, S., Norvig, P. 2007. Artificial intelligence: a modern approach. Willians, Moscow.

[6] Anwar, S., Inayat, Z., Zolkipli, M.F., Zain, J.M., Gani, A., Anuar, N.B., Khan, M.K., Chang, V. 2017. Cross-VM cache-based side channel attacks and proposed prevention mechanisms: a survey. Journal of Network and Computer Applications, 100(15), 259-279.

[7] Vittikh, V.A., Skobelev, P.O. 1998. Multi-agent systems for modeling of self-organization and cooperation processes. Available at: https://www.witpress.com/Secure/elibrary/papers/AI98/AI98022FU.pdf

[8] Nieto, A., Roman, R., Lopez, J. 2016. Digital witness: safeguarding digital evidence by using secure architectures in personal devices. IEEE Network, 30(6), 34-41.

[9] Choi, J. 2017. Secret key transmission for OFDM based machine type communications. Journal of Communications and Networks, 19(4), 363-370.

[10] Kamal, A.E., Imran, M., Chen, H.-H., Vasilakos, A.V. 2017. Survivability strategies for emerging wireless networks. Computer Networks, 128, 1-4.

[11] Grasel, F.D.S., Fontoura, L.A.M. 2016. Computacional study of the electronic effects in rotational barrier of the N-arilcarbamatos N-CO bond. Periodico Tche Quimica, 13(25), 7-15.

[12] Izaddoost, A., Heydari, S.S. 2017. Risk-adaptive strategic network protection in disaster scenarios. Journal of Communications and Networks, 19(5), 509-520.

[13] Cui, Y., Lai, Z., Dai, N. 2016. A first look at mobile cloud storage services: architecture, experimentation, and challenges. IEEE Network, 30(4), 16-21.