Agent-Based Approach for Distributed Intrusion Detection System Design*

Krzysztof Juszczyszyn, Ngoc Thanh Nguyen, Grzegorz Kolaczek, Adam Grzech, Agnieszka Pieczynska, and Radosław Katarzyniak

Institute of Information Science and Engineering, Wrocław University of Technology, Wyb. Wyspianskiego 27, 50-370 Wroclaw, Poland
krzysztof.juszczyszyn@pwr.wroc.pl

Abstract. The aim of this paper is to propose an architecture of distributed Intrusion Detection System (IDS). It is assumed that IDS system will detect and track dissemination and activity of the Internet worms. A general architecture for such a distributed multiagent system is proposed and the tasks, techniques and algorithms to be used are sketched.

1 Introduction

The aim of Intrusion Detection Systems (IDS) is to recognize and notify administrator of the system of various security events as well of incidents and anomalies that can be observed in the context of user behavior or system's and its environment states. All circumstances that indicate security policy violation must be evaluated by IDS due to its role to prevent and reduce the range of unauthorized usage of the system resources. Malicious traffic in the form of Internet worms, denial of service attacks, ports scans etc. became one of the most crucial problems for all that use Internet for communication and so it is also one of the most important challenges for intrusion detection systems. That's why a search for efficient mechanisms to protect networks from malicious traffic is in the main stream of interest of many research and operational communities. Only distributed monitoring and processing of a set of values related to some fundamental network traffic and communication invariants may create an appropriate starting point to evaluation of network security level.

Due to size and dynamics of the modern wide area network systems a distributed multiagent architecture seems to be an effective solution [6]. The system proposed in this paper falls into that class and is intended to allow fast detection and analysis of the ongoing attacks. It is assumed that security system must be able to recognise abnormal states of the network system and point out the sources of worm attack, which is not a trivial task in distributed network environment [10].

In the paper a general architecture of the multiagent system is proposed (Sec. 2), next the tasks assigned to the classes of agents in the system are defined (Sec. 3). The information integration processes are defined both on the level of monitoring agents (Sec. 4) and managing agents (Sec. 5) which infer about global anomalies and their sources.

* This work was supported by the Polish State Committee for Scientific Research under Grant No. 3 T11C 029 29 (2005-2007).

V.N. Alexandrov et al. (Eds.): ICCS 2006, Part III, LNCS 3993, pp. 224 – 231, 2006.

2 The Architecture of the Multi-agent System

The aim of this paper is to propose a framework for distributed multiagent Intrusion Detection System (IDS). It is assumed that the network system is consisted of the set of nodes. There are also two types of agents in our multiagent system: monitoring agents (MoA) and managing agent (MA). Monitoring agents (MoA) observe the nodes, process captured information and draw conclusions that are necessary to evaluate the current state of system security (Fig. 1). Each agent MoA monitors its own area of responsibility consisting of the set of nodes. It is assumed that these areas may mutually overlap [6].

It is known that in the case of worm attack there occur at least two kinds of anomalies: in observed traffic characteristics and in communication scheme which tends to be constant under normal conditions (see sec. 4.2). The system properties observed by the agent MoA fall into two basic (and physically different) categories: 1. Traffic measurement. 2. Communication pattern measurement.

In order to store and process information about the nodes under control each MoA agent uses two matrices: Traffic Matrix M_t and Communication Matrix M_c. Both have $n \times n$ size where n is the number of nodes controlled by a MoA agent. The rules for setting their values are given in the next sections.

Our architecture is intended to allow worm propagation detection and worm-based attacks on the basis of the fusion of information gathered from matrices M_t and M_c of all MoA agents.

In our approach following techniques are used: 1. Standard traffic analysis methods for traffic measurement. 2. Graph theoretic methods to identify abnormal communication patterns. 3. Consensus and data fusion methods to manage the system and generate alerts. The basic functional structure of monitoring and managing agents is shown on the Fig.2.

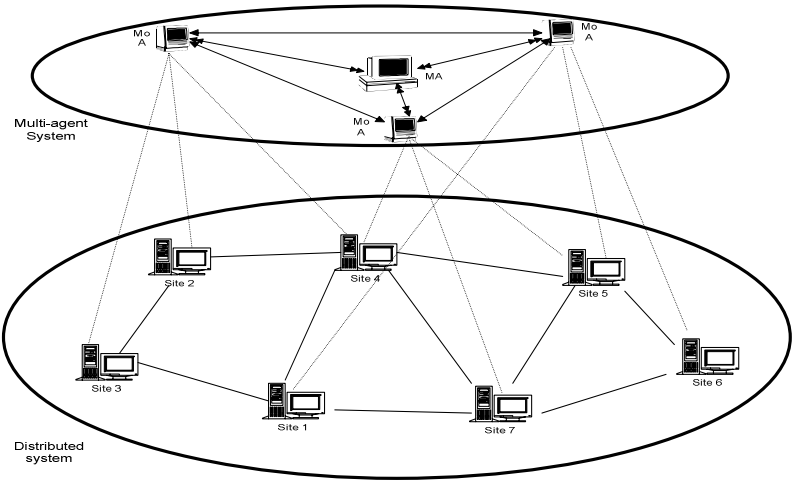

Fig. 1. The conception of the multi-agent system

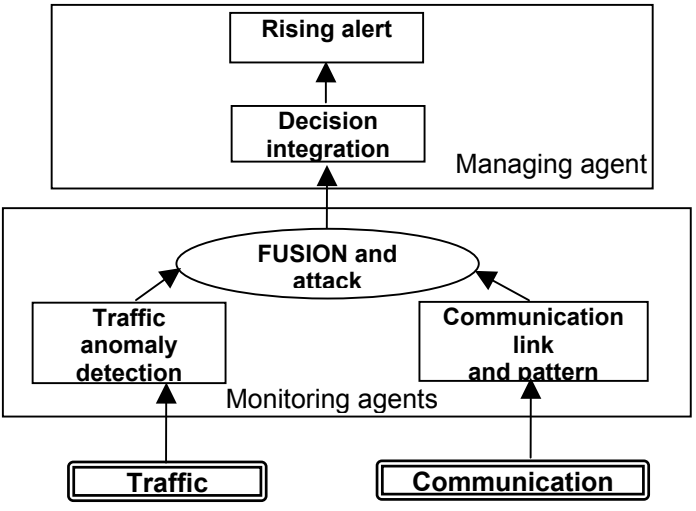

Fig. 2. The functional structure of monitoring and managing agents

3 The Main Tasks of the Intrusion Detection System

The multi-agent intrusion detection system has a number of tasks. Monitoring agents have similar tasks but managing agent(s) has different tasks. The tasks referring to work organization, knowledge processing, ontology integration have been outlined in work [6]. In this paper we present the description of the main task of the system, which is relied on detecting the resource of the attack as well as the kind of the attack. Denote by $\boldsymbol{A}$ the set of agents in the multi-agent systems; $\boldsymbol{S}$ – the set of sites of monitored system and $\boldsymbol{T}$ – the set of discrete ordered moments of time. To realize this task the system uses monitoring agents and one or more managing agents. This task is divided into following subtasks:

1. As mentioned above, a monitoring agent A_i observes the nodes in his region and analyzes gathered information in order to determine a tree $T_i^{(j)} = (S_i, R_i^{(j)})$, where $S_i \subset S$, $R_i^{(j)}$ is a binary relation in S_i such that pair $<s,s'> \in R_i^{(j)}$ for $s,s' \in S_i$ if and only if in the opinion of agent A_i the attack has came directly from site s to s'.
2. The decisions of monitoring agents are successively sent to the managing agent. The task of this agent is to integrate these decisions in order to determine the global tree representing the propagation situation of the attack in the whole system. Owing to this global tree one should get to know the resource of the attack as well as its propagation plan. For realizing this task the managing agent should have the following tools:

   - Criterion for assessment if the set of trees representing the knowledge states of monitoring agents in a time interval is enough for making a sensible decision.
   - An effective algorithm for integrating given trees and determining a global tree.

For the fist task a conception of an algorithm is presented in Section 4.3. Section 5 contains some idea for the second task. These tasks have been discussed in detail in numerous meetings by our project team consisting of the authors of work [6].

4 Making Decision Process of Monitoring Agents

4.1 Traffic Analysis

Network traffic anomalies are one of the most important sources of information for contemporary Intrusion Detection Systems. Traffic anomalies typically are related to states when network processes deviate from normal behavior. These anomalous events will disrupt the normal behavior of some measurable network data.

Anomaly detection system requires precise characteristic of network behavior and specially definition of parameters values related to the normal traffic. Selection of substantial set of variables for traffic description depends on several network specific factors such as the dynamics of the network being studied in terms of traffic volume, the type of network data available, and types of applications running on the network.

There are a few main streams that can be observed in the literature as the online modeling of network traffic, parsimonious traffic models that accurately capture fractal and multi-fractal scaling properties, such as the self-similar models introduced by Norros [9]. Lakhina et al. also have shown the application of Principal Component Analysis to separate of the high-dimensional space occupied by a set of network traffic measurements into disjoint subspaces corresponding to normal and anomalous network conditions [10].

In our approach, we only consider one source type of network traffic anomalies - malicious traffic. We assume that the Intrusion Detection System should recognize anomalies related to phenomenon like self-propagating worms, viruses, port scans and denial of service attacks. The choosing of certain traffic analysis method is now an open question and depends on the first assumptions concerning simulation and testing framework. However, a MoA will track traffic anomalies within its area of responsibility and the results of the tracking will be stored in a traffic matrix M_t which will have $n \times n$ *size* where n is a number of network nodes belonging to MoA's area of responsibility.

The values of M_t matrix are to be set according to the following rules:

$$M_t(i,j) = \begin{cases} 0 & : \text{there is no data link between nodes } i \text{ and } j \\ 1 & : \text{nodes } i \text{ and } j \text{ communicate} \\ \varepsilon & : \text{link state is unknown} \\ a_k & : \text{anomaly of type } k \text{ was detected between } i \text{ and } j \end{cases}$$

Where a_k (anomaly type) is intended to identify traffic anomaly by means of the method of detection or magnitude. This information maybe important for monitoring agent's decision making process. The M_t matrix of any given MoA will be periodically sent to managing agent AM in order to reason about global traffic disturbances caused by the worm attack.

4.2 Communication Patterns

4.2.1 Communication Patterns – State of the Art

Recent results show that network traffic show some quantitative and topological features that appear to be invariant and characteristic for given network. Moreover, general rules underlying that features are the same for almost any network of remarkable size. These distinct features concern topology of network communication, considered as origin-destination flows graph, the distribution of data volumes sent between destinations and the in/out ratio of data sent between subnets and outside world.

The general idea of detecting worm attack by observation of communication patterns is to compare patterns existing under normal state of the network to new ones which occur under attack. In both cases there exist several data flows in the network (origin-destination flows). Within a network under attack there are also scanning and attack flows which differ substantially from normal network activity [8]. Moreover, total scanning rate into the sub-network (or given set of nodes) is a function of the number of all infected nodes in the network.

Also, proportion between a number of internal and external data flows for a given subnet is constant for a long time periods and different scales (subnet sizes) or traffic types (protocols) [1]. For any network node its fan-in (percent of hosts that originate conversations with that node) and fan-out coefficients are constant both for local and wide-area traffic.

For communication patterns in entire network there are also some interesting results. In particular, it was proved that the IP graph has heavy-tailed degree distribution showing scale-free structure according to power law (probability $P(k)$ of the node having degree of k equals approximately $k^{-\gamma}$) [4]. Under worm attack the structure of communication is heavily affected and the power law distribution changes. There is also a detectible dependence between worm propagation algorithm, scale of the attack and communication pattern disturbance [5].

Similar relationships occur also on the level of given communication protocol, for example after investigation of the topology of e-mail corporate network (with e-mail addresses as nodes and e-mails as links) it was found that the resulting network exhibits a scale-free link distribution and small-world behaviour, as for known social networks [3]. This result was then used to propose an anti-spam tool [2].

4.2.2 Detectable Communication Patterns

Monitoring agents of proposed IDS system will gather information about communication within the network under control, then the existing communication patterns will be discovered (global patters are by definition visible only for monitoring agents). The system will be viewed as a graph consisting of nodes (each monitoring agent will have a set of nodes under control) and edges which appear if there exists data flow between given pair of nodes. In our approach we postulate tracking the following communication patterns:

- Node degree distribution (global pattern)
- Clustering coefficient for a given node (local, detected on global level)
- Fan-in and Fan-out ratios (local, detected on global level)

All the above patterns are (according to results listed in previous section) invariant during most time of normal system activity. But under worm attack they will change leading to alert and taking chosen countermeasures.

Each MoA agent stores data about communication in the form of M_c matrix. The values of M_c are set according to the following rules:

$$M_c(i,j) = \begin{cases} 0 & : \text{there is no communication between nodes } i \text{ and } j \\ 1 & : \text{nodes } i \text{ and } j \text{ communicate} \\ \varepsilon & : \text{no knowledge about communication between } i \text{ and } j \end{cases}$$

1. *Node degree distribution* is computed after gathering information in form of M_t matrices from MoA agents.

2. The *clustering coefficient C* is the probability that two nearest neighbours of vertex *i* are also neighbours of each other. C provides a quantitative measure for cliques in communication graph. For node *i* clustering C_i is given by:

$$C_i = \frac{2k_i}{n_i(n_i - 1)}$$

where n_i is the number of its neighbours and k_i – the number of connections between them. Thus high *C* reflects that a node belongs to a clique. There is also a power-law distribution of *C* values reflecting the fact that nodes with high degree (hubs) show low values of clustering coefficient (they connect cliques).

3. Fan-in and Fan-out ratios are computed by the managing agent:

Fan-in is the number of nodes that originate conversations with node *i*, while Fan-out is the number of hosts to which *i* initiates conversations. Experiments showed that both Fan-in and Fan-out for given node and their distribution for all nodes tend to be constant under normal conditions.

4.3 The Outline of Algorithm for Monitoring Agent Decision Making Process

The MoA agent's algorithm for decision making process is invoked periodically and uses values of M_t and M_c matrices as input data. It should be noted the MoA stores acquired values of M_t and M_c thus creating the history of system behaviour. The algorithm itself consists of the following steps:

Given: Matrices M_t and M_c (as defined in Sec. 4.1 and 4.2.2).
Result: The tree $T_i^{(j)} = (S_i, R_i^{(j)})$ (see Sec. 3).
BEGIN

1. Detect traffic anomalies (using chosen technique). Fill cells in M_t.
2. Create a list of traffic anomalies (on the basis of the values from M_t).
3. Compute the communication patterns (on the basis of the values from M_c).
4. Create a list of communication anomalies (where a communication anomaly is unexpected or rapid change of clustering, fan-in/fan-out ratio for a given node or a change in a node degree distribution).
5. If any of the anomalies' lists is not empty, perform an attack backtracking analysis which will return result in the form of attack tree $T_i^{(j)} = (S_i, R_i^{(j)})$ as defined in Sec. 3. Note, that constructing such a graph requires data fusion from

two different sources – traffic and communication anomalies detection. The joint usage of two physically different groups of measures will provide more accuracy in tracking attacks, which may be guided by different worm propagation algorithms.

END.

Note also that some fields in each M_t and M_c may be unknown (if their actual values were for some reason not observed by the MoA) which result in some uncertainty in attack investigation analysis. The level of this uncertainty will be also a part of the algorithm's result.

5 Integration of Monitoring Agents' Decisions

In this section we present an outline of the realization scheme for the second task defined for the managing agent. As mentioned, the managing agent successively obtains from monitoring agents their decisions in form of trees, which are related to particular moments of time. It should have a criterion for assessing the susceptibility of these trees to a credible global tree and an effective algorithm for its determining.

Formally, we may assume that for given time interval $[t, t']$ in the database of the managing agent there is a set of trees, that comes from one monitoring agents:

$$T_{[t,\,t']} = \{T_i^{(j)} = (S_i, R_i^{(j)}): i=1,\ldots,m \text{ and } t \le t_j \le t'\}.$$

It seems to be intuitive that the special subject of interest of the agent is such set $T_{[t,\,t']}$, where t' is the current time moment. For this set the mentioned criterion for credibility susceptibility (CS) can be formulated in the following way:

Let $G_i^{(j)} = (S_i, V_i^{(j)})$ be a non-directed graph which arises from tree $T_i^{(j)}$ in such way that it contains the same edges as tree $T_i^{(j)}$, but these edges are non-directed. Let

$$S = \bigcup_{i=1}^{n} S_i \quad \text{and} \quad V = \bigcup_{i=1}^{n} V_i \,.$$

The criterion CS can be defined as: *Non-directed graph* (S,V) *should be coherent.*

The intuition of this condition is that the knowledge of monitoring agents referring to attacked sites is in some sense complete. It means that if (in opinions of some monitoring agents) two sites are attacked then there is a relationship between them. This condition is good only if only one site is the source of investigated attack. We accept this assumption because it is the most common case in practice.

If the set $T_{[t,\,t']}$ of trees satisfies the condition included in criterion CS then the managing agent MA can use an algorithm for determining the global tree representing the attack propagation. The idea of this algorithm is based on consensus methods [7]. The sketch of this algorithm is as follows:

Given: Set $T_{[t,\,t']}$ of trees satisfying the condition included in criterion CS.
Results: Tree (S,V) representing the global attack propagation
BEGIN

1. Let $S = \bigcup_{i=1}^{n} S_i$ and $V = \varnothing$;
2. Complete V using postulates *Closure of knowledge*; *Consistency of knowledge* and *Superiority of knowledge* defined in work [7].
3. For such edges referring to which some of monitoring agents are in conflict, use a consensus method for solving.
4. If there is a lack of edges for creating a tree, ask monitoring agents MoA for completing data.

END.

6 Conclusions

In this paper a distributed, multi-agent IDS design conception has been presented. This conception takes into account many symptoms of the ongoing worm attacks. The same architecture may be as well adapted to the control network communication limited to chosen protocol or service (described regularities might be applied on different levels: IP traffic, protocols, etc.). The future works should concern the detailed design of the system and working out algorithms for its functioning.

References

1. Allmanz M. et.al. A First Look at Modern Enterprise Traffic, In Proc. Internet Measurement Conference, October 2005, 217-231.
2. Boykin O., Roychowdhury V. Personal Email Networks: An Effective Anti-Spam Tool, IEEE Computer, Vol. 38, No. 4, (2005), 61-68.
3. Ebel H., Mielsh L., Bornholdt S., Scale-free topology of e-mail networks, Physical Review E 66, 2002, 121-131.
4. Faloutsos M., Faloutsos P., Faloutsos C., On power-law relationships of the Internet topology. In Proc.ACM SIGCOMM '99 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication, August 1999, 251-262.
5. Kohler E., Liy J., Paxson V., Shenker S., Observed Structure of Addresses in IP Traffic, In Proc. SIGCOMM Internet Measurement Workshop, November 2002, 253 - 266.
6. Kolaczek G., Kuchtiak-Pieczynska A., Juszczyszyn K., Grzech A., Katarzynak R., Nguyen N.T. (2005): A Mobile Agent Approach to Intrusion Detection in Network Systems. In: Proceedings of KES 2005, Lecture Notes in Artificial Intelligence 3682 (2005) 514-519.
7. Nguyen N.T.: Consensus systems for conflict solving in distributed systems. Information Sciences 147 (1-4) (2002) 91-122
8. Nicol D., Liljenstam M., Liu J., Multiscale Modeling and Simulation of Worm Effects on the Internet Routing Infrastructure, In Proc. Performance Tools Conference, 2003, 1-10.
9. Norros I., A storage model with self-similar input, Queueing Syst. 16 (1994) 387–396,.
10. Lakhina A., Crovella M., Diot. C. Diagnosing Network-Wide Traffic Anomalies. In Proc. of ACM SIGCOMM 2004. Portland. (2004) 219 – 230.