A Distributed Intrusion Detection System Using Cooperative Agents

HAKIMI, Z.[1*,†], BARATI, M.[1*], JAVADI, A.H.[2]

[1]Department of Computer Engineering, Qazvin Branch, Islamic Azad University, Qazvin, Iran
(phone: +98 281 3665 275; fax: +98 281 3665 279)
[2]Institute of Behavioral Neuroscience, University College London, London, UK
(phone: +44 20 7679 1029)
[*] These authors contributed to the same extend
[†] Corresponding author e-mail: hakimi@qiau.ac.ir

Abstract. Due to the extension of attacks to computer networks in the current decade, it is crucial to secure the networks in an appropriate and effective way. One solution is to have a host that supervises the situation for all the computers in the network and makes decision regarding possible attacks. This method is not effective in recently developed networks due to their large extension and high speed. We propose a new distributed intrusion detection system that is based on cooperative agents ('DIDCA'). Agents are grouped into subsets of agents that are in connection with a leader on top. Leaders of different subsets communicate together to share all the information throughout the network. We simulated our method using DARPA 1999 database and compared it with Snort. Our results showed that our method achieved a better performance.

[HAKIMI, Z., BARATI, M., JAVADI, A.H.. **A Distributed Intrusion Detection System Using Cooperative Agents**. Life Sci J 2013;10(8s): 253-257] (ISSN:1097-8135). http://www.lifesciencesite.com. 40

Keywords: distributed intrusion detection system, Snort, Java agent development environment (JADE), DARPA, agent

Introduction

Network security is an important concept that is defined by protection of valuable resources such as services and information in the network. An intrusion is a set of actions that try to affect this security and consequently damage confidentiality, integrity or availability of resources. Therefore, application of intrusion detection systems (IDS) is necessary as a defensive system to detect possible intrusion activity.

Traditional IDSs were centralized, i.e. both data collection and processing were performed by the same station. These systems were restrained by the following limitations: (i) The weakness of single point of failure, i.e. protection of the whole network relied on efficient performance of one machine.(ii) The problem of scalability, i.e. the processing and memory capacity depended on the size of network and traffic and more importantly on the processing resources of only one machine. These limitations led to the application of distributed intrusion detection systems. A distributed IDS consists of multiple IDSs over a large network. These IDSs communicate directly with each other, or with a central machine that facilitates advanced detection methods (Einwechter, 2001). Many existing distributed solutions are not fully distributed. They collect network traffic through distributed machines but only one machine analyzes the data (Sen, 2011). Although this method has several advantages over centralized IDSs, having one analyzing machine creates a bottleneck in the system. Due to growth and extent of today's large and dynamic networks it is essential to use distributed IDSs in which data processing is also carried out over multiple of machines.

In this paper we proposed a new fully (both data collection and analysis) distributed intrusion detection system based on cooperative agents (DIDCA). Primarily DIDCA follows two main objectives, increasing scalability and also detection speed. DIDCA uses a group of software agents and proposes a new architecture for grouping agents automatically. These agents work together in a novel distributed and cooperative manner.

We used MIT DARPA 1999 dataset to simulate the traffic in a network. To create the agents and the platform for our DIDCA, we used JADE implementation of Snort based agents (see Methodology section).

Next section presents related works. Methodology section describes our methodology and tools. Experimental results section reports the experimental results of the system using DARPA 1999 dataset. Finally, last section concludes and points out some future work.

Related Work

Autonomous agents for intrusion detection (AAFID) (Balasubramaniyan et al., 1998) uses a hierarchy method. It consists of three types of entities: transceiver, monitor and agent. At the lowest level, agents collect data and forward them to transceiver which is in the upper level and after performing initial processing, the data is transmitted to monitor for attack detection. The major limitation of this system is that processes are done only when the information from all the machines are gathered together in one place.

Xiao et al. (2005) and Ramachandran and Hart (2004) used a voting process to get information on attacks from other agents, when a suspicious traffic is detected by an agent that needs further consideration. Using this method, network load is kept low but delays that are imposed by recognition processes are high.

Sen (2011) presented a distributed IDS using hierarchical cooperating agents. In this method analysis is done by agents in the lowest level. When an agent needs additional data, it sends a request to upper level agent(s) asking for some specific data. There is an agent on the top level of hierarchy which forwards data and interests between different domains. This agent imposes a single point of failure to the system.

Sasikumar and Manjula (2011) proposed a distributed IDS with layered agents. It consists of net- and host-agents in the lowest layer. These agents act as a general IDS and whenever an agent detects a suspicious activity, it reports it to upper layer agents. These mobile agents visit all machines in the network to collect reports created by net- and host-agents in order to make a more consistent decision. The main drawback of this system is scalability: mobile agents must visit all the hosts which is not efficient as its speed drops with the growth of the network size.

Gunawan et al. (2011) introduced a distributed IDS using collaborative building blocks. It works based on many collaborative components. They define a set of analysis tasks and assign these tasks to a number of hosts in order to handle large amount of data in the network.

DIDMAS is a distributed network IDS using mobile agents and snort proposed by Brahmi et al. (2011). It has different cooperative agents for data collection, filtration and detection of Intrusions. DIDMAS uses a database of rules from Snort for detection of known attacks. It poses a better performance compared to Snort.

Methodology

Tools

DARPA 1999 Dataset

We evaluated DIDCA on DARPA 1999 Intrusion Detection Evaluation dataset (Lippmann et al., 2000), which is publicly available, labeled and widely used for IDS evaluation purposes (Agarwal and Mittal, 2012, García-Ruiz et al., 2007, Thomas, 2010). It is developed by Lincoln Laboratory in Massachusetts Institute of Technology (www.ll.mit.edu/mission/communications/cyber/CSTcorpora/ideval/data/). This dataset provides a labeled dataset of different type of attacks along with normal traffic in a simulated network. The simulated network generates five weeks of data. Data of the first three weeks is for training purposes for data driven learning systems, whereas data of the last two weeks represents test data. In our study we used the last 2 test weeks to evaluate our method. DARPA 1999 categorizes attacks into four groups: probing, denial of service (DoS), user-to-root (U2R) and remote to local (R2L). Probing (such as nmap, portsweep and IPsweep) is an attack that attacker scans a machine or a network of computers to determine vulnerabilities of that system(s). In a DoS attack (such as neptune and smurf), attacker attempts to make resources (computing or memory) of target system busy and unavailable to its legitimate users' access. U2R (such as xterm and casesen) is a type of attack that attacker uses a normal user account to gain access to root privileges. R2U attacks (such as guest and xlock) occur when a remote machine which can send packets to a target system, attempts to gain unauthorized access to that system and acts as a local user. In our study we focused on probing attacks from this dataset.

JADE

More specifically we used JADE 3.7 (Java Agent Development Environment) to implement our agents. JADE (jade.tilab.com/) is a middleware for the development of multi-agent distributed peer to peer applications (Bellifemine et al., 1999). It is developed in Java by Telecom Italia Lab (TILAB). JADE provides all the basic services for the distributed peer to peer applications, i.e. each agent can dynamically discover other agents, and subsequently can communicate with them by some sorts of message exchange mechanisms.

Snort

Snort 2.9.3.1 is an open source and signature based network intrusion detection and prevention system (IPS), developed by Sourcefire (www.sourcefire.com/security-technologies/open-source/snort). It has been widely used for IPS/IDS (Roesch, 1999). Snort uses a database of rules and recognizes malicious traffic by matching it with these rules. We used Sourcefire vulnerability research team

(VRT) rules, available in (www.snort.org/vrt). In the proposed IDS, we chose Snort as the signature-based IDS. Activated rules in Snort are shown in Table 1.

Table 1. Shows enabled rules in Snort.

Rule ID	Message
1-27	Sfportscan preprocessor rules
613	SCAN myscan
615	DELETED SCAN SOCKS Proxy attempt
616	SCAN ident version request
617	DELETED SCAN ssh-research-scanner
618	DELETED SCAN Squid Proxy attempt
619	SCAN cyber copos probe
620	DELETED SCAN Proxy Port 8080 attempt
621	DELETED SCAN FIN
622	SCAN ipEye SYN scan
623	DELETED SCAN NULL
624	DELETED SCAN SYN FIN
625	DELETED SCAN XMAS
626	SCAN cyber copos PA12 attempt
627	SCAN cyber copos SFU12 probe
628	DELETED SCAN nmap TCP
629	DELETED SCAN nmap fingerprint attempt
630	SCAN synscan portscan
384	ICMP-INFO PING
255	DNS zone transfer TCP
359	FTP satan scan

DIDCA

Our main goal was to detect distributed attacks to network in addition to attacks to individual hosts in the network. In our proposed method all the processes are carried out over multiple hosts. Therefore our method does not break down due to 'single point of failure'. More importantly, the resources needed for data processing for each newly added host is provided by the host itself. This way, addition of a new host does not impose any additional load on the network. Consequently it enables the network to expand flexibly. An agent is allocated in each host that is responsible for the following tasks: (a) each agent is actually a simple IDS based on Snort that is responsible for detecting attacks to its respective host. (b) Each agent is required to communicate and collaborate with (some of) other agents in the network to detect distributed attacks.

Agents are grouped into smaller sets of agents to limit their scope of communication and reduce the overall load resultant of traffic between agents. In each of these groups, an agent is assigned as the leader. Leaders communicate together in order to share the information throughout the network to detect highly distributed attacks. Grouping, selection of the leader and their method of communication is described below.

Group Formations

Our proposed method is fully compatible with conventional local area networks (LAN). When a packet is broadcasted in a LAN, it is transferred to all computers in the network. Similarly, in normal circumstances the messages broadcasted by agents are received by all agents distributed over the network. To limit this communication and create broadcast levels, we used virtual LAN (VLAN). VLANs can be implemented in network routers and switches. Therefore using this method, we will have multiple VLANs in which messages broadcasted by agents in a certain VLAN is limited to agents allocated in that VLAN (Fig. 1).

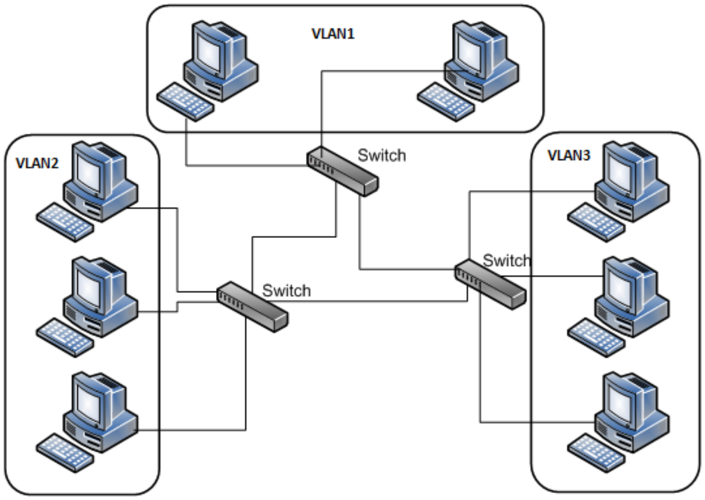

Figure 1. *A switched network topology using VLANs.*

Leader Selection

As mentioned before, we allocated one agent to each host. Subsequently these agents are grouped into VLANs. When a new agent is activated, it identifies the leader belonging to the same VLAN to be able to transfer the information to it. This is done using the algorithm explained below: The newly activated agent broadcasts a packet to identify possible existing leader in the VLAN. It will store the leader's IP address if it receives any feedback from the leader, otherwise it will assume that it is the only agent in the VLAN and therefore becomes the leader. This method is illustrated in Fig. 2.

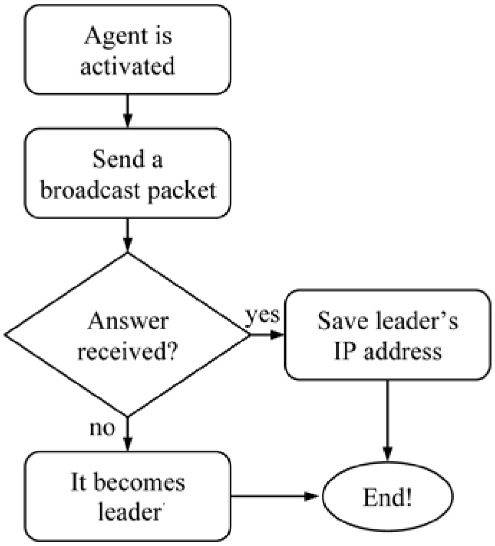

Figure 2. *Leader agent selection algorithm.*

Detection of Distributed Attacks

As mentioned above, each agent operates based on Snort on the local host. Additionally it is enabled to communicate with the leader in the respective VLAN in the event of detection of any suspicious activity. Leader is responsible to correlate these messages collected from different agents and send an alert to the network administrator whenever it detects an attack.

Experimental Results

A prototype of proposed system has been developed with JADE. Each agent is an IDS base on Snort and configured to detect malicious activities against respective hosts. In addition, each agent is programmed to do group duties in respect to previously discussed algorithms, such as: (i) identifying the group leader, (ii) Sending messages to the leader when it detects any suspicious activity, (iii) detecting distributed attacks by correlating messages collected from different agents when the agent is selected as leader.

We used the last 2 testing weeks of DARPA 1999 to evaluate our method. We focused on probe attacks which contains both centralized and decentralized attacks. They in total consisted of 37 instances of 9 different probing attack types. Fig. 3 shows the distribution of different instances. This dataset has a master identification list that contains information of all attacks. We wrote a script to compare Snort alerts with real attacks existing in the master identification list to identify hits and false-alarms. Finally we compared our results with Snort. The comparison is showed in Table 2.

Table 2. Comparison between our method ('DIDCA') and Snort.

	DIDCA	**Snort**
Hit rate	58.02%	51.14%
False alarm rate	44.80%	49.32%

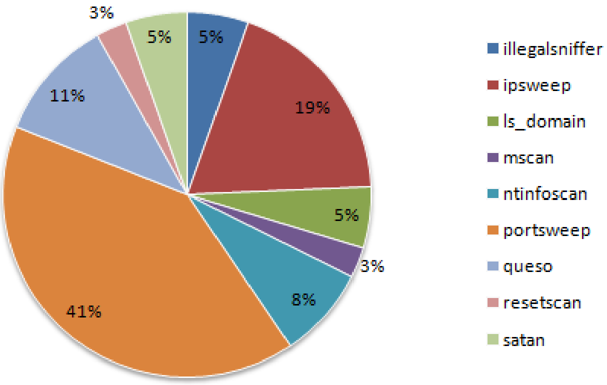

Figure 3. *Distribution of probing attack types.*

Conclusions

In this paper we presented a new distributed IDS based on cooperative agents ('DIDCA'). In DIDCA both data collection and processing are done in a decentralized and cooperative manner. We allocated software agents to each host in the network. We grouped these agents into VLANs in order to detect distributed attacks. In DIDCA we solved the problem of centralized processing in IDSs such as NSM (Heberlein et al., 1990) and DIDS (Snapp et al., 1991). In these systems a centralized machine performs all processing works and detects attacks throughout the network. A major limitation of this method is vulnerability to extension of the network. Also comparing with other distributed IDSs based on mobile agents such as in Kannadiga and Zulkernine (2005) and Sasikumar and Manjula (2011), DIDCA is more scalable due to its method of grouping, as it is not necessary to collect information regarding all the machines in the network in one machine. Finally, we compared DIDCA with a widely used IDS, Snort, by using DARPA 1999 evaluation dataset. The results showed DIDCA is more effective than Snort with a higher hit rate.

As part of future work we aim to test speed of our proposed method versus other distributed IDSs. We expect that our prototype is faster due to its method of grouping mechanism. Also we aim to add agents' authentication algorithms to verify exchanged messages between agents during their collaboration to avoid transmission of fake messages.

References

1. Agarwal, B., Mittal, N. (2012): Hybrid Approach for Detection of Anomaly Network Traffic using Data Mining Techniques, Procedia Technology, 6: 996-1003.
2. Balasubramaniyan, J. S., Garcia-Fernandez, J. O., Isacoff, D., Spafford, E., Zamboni, D. (1998): An architecture for intrusion detection using autonomous agents. Proceedings of the 14th Annual Computer Security Applications, IEEE, 13-24.
3. Bellifemine, F., Poggi, A., Rimassa, G. (1999): JADE–A FIPA-compliant agent framework, Proceedings of PAAM, London, 33.
4. Brahmi, I., Yahia, S. B., Poncelet, P. (2011): A Snort-Based Mobile Agent For A Distributed Intrusion Detection System, Proceedings of the International Conference on Security and Cryptography, Seville, Spain.
5. Einwechter, N. (2001): An introduction to distributed intrusion detection systems, Security Focus.
6. García-Ruiz, M., Vargas Martin, M., Kapralos, B. (2007): Towards multimodal interfaces for intrusion detection, Proceedings of the 122nd Convention of the Audio Engineering Society.
7. Gunawan, L. A., Vogel, M., Kraemer, F. A., Schmerl, S., Slåtten, V., Herrmann, P., König, H. (2011): Modeling a distributed intrusion detection system using collaborative building blocks, ACM SIGSOFT Software Engineering Notes, 36: 1-8.
8. Heberlein, L. T., Dias, G. V., Levitt, K. N., Mukherjee, B., Wood, J., Wolber, D. (1990): A network security monitor, Proceedings of the IEEE Symposium on Security and Privacy, IEEE, 296-304.
9. Kannadiga, P., Zulkernine, M. (2005): DIDMA: A distributed intrusion detection system using mobile agents, Proceedings of the Sixth International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, and First ACIS International Workshop on Self-Assembling Wireless Networks, IEEE, 238-245.
10. Lippmann, R., Haines, J. W., Fried, D. J., Korba, J., Das, K. (2000): The 1999 DARPA off-line intrusion detection evaluation, Computer Networks, 34: 579-595.
11. Ramachandran, G., Hart, D. (2004): A P2P intrusion detection system based on mobile agents, Proceedings of the 42nd annual Southeast regional conference, ACM, 185-190.
12. Roesch, M. (1999): Snort-lightweight intrusion detection for networks, Proceedings of the 13th USENIX conference on System administration, Seattle, Washington, 229-238.
13. Sasikumar, R., Manjula, D. (2011): A Distributed Intrusion Detection System Based on Mobile Agents with Fault Tolerance, European Journal of Scientific Research, 62: 48-55.
14. Sen, J. (2011): A Distributed Intrusion Detection System Using Cooperating Agents, arXiv preprint arXiv:1111.0382.
15. Snapp, S. R., Brentano, J., Dias, G. V., Goan, T. L., Heberlein, L. T., Ho, C. L., Levitt, K. N., Mukherjee, B., Smaha, S. E., Grance, T. (1991): DIDS (distributed intrusion detection system)-motivation, architecture, and an early prototype, Proceedings of the 14th National Computer Security Conference, 167-176.
16. Thomas, A. (2010): RAPID: Reputation based Approach for Improving Intrusion Detection Effectiveness, Sixth International Conference on Information Assurance and Security (IAS), IEEE, 118-124.
17. Xiao, K., Zheng, J., Wang, X., Xue, X. (2005): A novel peer-to-peer intrusion detection system, Sixth International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT), IEEE, 441-445.

4/2/2013