International Journal of Electrical and Computer Engineering (IJECE)
Vol. 3, No. 6, December 2013, pp. 732~740
ISSN: 2088-8708

A Flow-based Distributed Intrusion Detection System Using Mobile Agents

Zahra Hakimi[1], Karim Feaz[2], Morteza Barati[1]
[1]Department of Computer and Information Technology Engineering, Qazvin Branch, Islamic Azad University, Qazvin, Iran
[2]Department of Electrical Engineering, Amirkabir University of Technology, Tehran, Iran

Article Info

Article history:

Received Jul 18, 2013
Revised Oct 1, 2013
Accepted Oct 18, 2013

Keyword:

Distributed IDS
Flow-based IDS
Intrusion detection system
Nmap
NS2 simulator
Scan attack
Simpleweb traces

ABSTRACT

In recent decade, computer networks have grown in popularity. So, network security measures become highly critical to protect networks against different kind of cyber attacks. One of the security measures is using intrusion detection system (IDS). An IDS aims to detect behaviors that compromise network integrity, availability and confidentiality, by continuously capturing and analyzing events occurring in the network. A challenging problem for current IDSs is that their performance decreases in today's high speed and large scale networks. A centralize IDS cannot process such high volume of data and there is a high possibility that it discards some attacks. In this paper we propose a flow-based distributed IDS using mobile agents (MA), which performs both data capturing and data analyzing in a distributed fashion. Our distributed IDS provides a framework for deployment of a scalable and high performance IDS, which by using a grouping mechanism and help of mobile agents, effective collaboration can be established between all network members. We simulated our method in NS2. Then we compared our proposed system with a general network-based IDS and a distributed IDS. Experimental results showed its superiority using several metrics of network load, detection rate and flow loss rate.

Corresponding Author:

Zahra Hakimi,
Department of Computer and Information Technology Engineering,
Qazvin Branch, Islamic Azad University,
Qazvin, Iran.
Email: hakimi@qiau.ac.ir

1. INTRODUCTION

In the course of last decade, popularity of computer networks growed enormously. This has led to increased network attacks, which comprises networks' security. One of the ways to deal with these attacks is application of 'Intrusion Detection Systems' (IDS). An IDS inspects network traffic continuously to detect probable network threats. IDSs are categorized into two main groups: host-based IDSs and network-based IDSs. Host-based IDS resides on a host and captures the information received by that host to detect attacks to that computer. On the other hand, a network-based IDS operates on a computer network and captures all the information exchanged in network to detect network attacks. Network-based IDS, on the other hand, has a global view to network data, which is necessary to detect distributed attacks to the network. Distributed attacks target a set of computers in the network instead of only one computer. Therefore, an IDS cannot detect such attacks without having such global view.

Network-based IDSs are categorized into two groups: centralized IDS and distributed IDS. These two categories are different in their approach towards attack detection. Generally, there are two steps in attack detection: data capturing and data analysis. In centralized IDS both steps are performed in one host. This method has single point of failure problem and high memory usage especially when the rate of arrival traffic increases or network size grows. Therefore, this type of IDS cannot be effectively used for today's

large and fast networks. On the other hand, in distributed IDS each of the steps could potentially be performed in a distributed way on multiple computers. Some distributed IDSs incorporate distributed operation in only capturing step [1]. These systems consist of a set of sensors which capture data in distributed fashion and send the collected data to a central system for further analysis. Another kind of distributed IDSs have central capturing unit but perform data processing in a distributed manner [2].

Although these kinds of distributed IDSs have many advantages over centralized IDSs, the central capturing and central processing downgrades system performance in high traffic rates. Using additional hardware, some other kind of distributed systems perform both the data capturing and data processing in a distributed manner [3].

In this paper we propose a flow-based distributed IDS using mobile agents that performs both data capturing and data analyzing in a distributed fashion. We have a grouping mechanism which divides computers in the network into subsets of computers with a leader and a few members. Each member receives its own packets and constructs flow information. In the case of detection of a suspicious activity it sends suspicious flow information to the group leader. The leaders are responsible to collect suspicious flow information received by the members and perform attack detection. Also each leader can get access to other groups' information by using mobile agents, when they need additional information. This paper builds upon our previous work [4], in which a packet-based distributed IDS was developed. Here, we use flow-based analysis to increase detection speed and reduce amount of processing done by the group members and leaders. In addition, we use mobile agents to decrease network load to achieve higher performance.

The major contributions of our research work are as follows:

- Utilization of all existing resources in the network, in terms of both computing and memory. Compared to many other distributed IDSs, there is no need for any additional resources. All computers in the network contribute in attack detection. This architecture enables our system to be scalable with network size.
- Put minimum extra load on the network. As the exchanged information is in the form of flows and leaders' communication is through mobile agents.
- Ability to operate in multisite networks. We can consider multiple sites as one big site with multiple groups in which the leaders in each group can communicate with each other using mobile agents.

In order to have a measure of performance, we compared our method with a general network-based IDS and the proposed method of distributed IDS developed by Erbacher et al. [5], in terms of network load, flow loss rate and detection rate in a large scale network.

In the following we will explore some of the related studies. Section 3 gives a brief description of mobile agent and its advantages. Section 4 describes our implementation method. Section 5 presents our simulation platform. Section 6 reports the experimental results. Finally, section 7 presents our conclusions and points out some future work.

2. RELATED WORK

Many intrusion detection systems have been proposed to protect networks against different types of attacks. In the todays' large scale and high rate traffic networks, the traditional centralized IDSs cannot operate efficiently due to large amount of exchanged and processed data. Therefore distributed IDSs have been proposed. Some of them such as DiCAP [1] captures traffic data in a distributed manner. DiCAPs' main concept is to distribute input traffic over multiple capture nodes. In this method each node monitors only a subset of traffic. These nodes are organized in the form of multiple capturing clusters in which each cluster has a coordinator. Nodes decide upon the part of the traffic which they are responsible for using the rules they receive from coordinator. Additionally, the system contains a packet data analyzer. Nodes send all captured packet headers to the packet data analyzer for further processing and attack detection. The main drawback of this system is vulnerability to single point of failure due to centralized processing.

The system proposed by Erbacher et al. [5] consists of intelligent sensor objects to capture traffic and a central analyzer for attack detection. The sensors create an optimized structure for information content called feature vectors based on flow information to reduce row data size. Subsequently the data is transmitted to the central analyzer for attack detection. This will put extra load on the network specifically when the network size grows. Also the central analyzer is vulnerable to single point of failure.

In some other methods such as the proposed method by Gunawan et al. [2], data processing is performed in a distributed manner. In this system a set of analysis units are defined to increase processing speed. The output of these analysis units are forwarded to different hosts and after processing, the results are fed back to the system. Consequently the processing load is removed from a central system. The main

drawback of this system, however, is increase in network load due to extra communications in between nodes.

Some other groups of systems perform both capturing and processing in a distributed manner. NG-MON [3] is a distributed flow-based IDS that employs additional hardware to do attack detection. It defines five different phases: flow generation, flow store, flow analysis and presentation. Each phase consists of a cluster of computers to parallelize the workload in order to increase speed.

Also there are methods that incorporate a distributed IDS using agent technology. Sasikumar and Manjula [6] discuss a distributed IDS consisting of multiple layers of agents. The lowest layer is composed of host and net agents and the upper layer is composed of mobile agents. Host and net agents have got general IDS functions which process the captured data to detect possible attacks. They generate an 'event' and transmit it to the second layer, in the case they detect a suspicious activity that cannot decide whether it is intrusive or not. Mobile agents visit all hosts to collect host and net agents' generated events. They correlate related events and detect intrusions.

MAFIDS [7] is another mobile agent based system consisting of four types of agents: sniffer agent, filter agent, analyzer agent and decision agent. Sniffer agents are distributed throughout the network. They collect all the events and save them into a sniffing file. Filter agent has access to this sniffing file and preprocesses the data. For example, it sorts the packets based on the type of protocols (TCP, UDP and so on). Then analyzer agent processes the information prepared by filter agent and detect intrusions. Finally results will be transmitted to decision agent to inform admin about the intrusions.

There are some other approaches in which different type of packet sampling methods are used to cope with high rate of network traffic by discarding some certain parts of packets [8-10]. Sampling mechanisms are divided into three methods: sampling in the specific time intervals, sampling after arrival of a specific number of packets and random sampling. The best advantage of this technique is reduction of traffic rate. On the other hand, it reduces the detection rate due to abstraction of packets after sampling.

The main aim of our distributed IDS is to deal with some of fore-mentioned problems such as single point of failure and high network load. Our system is fully distributed in terms of both packet capturing and data analysis.

3. BACKGROUND

3.1. Mobile Agent

An agent is a software program which can perform a specific work without need to manual supervision [11]. In addition to work independently an agent is capable to collaborate with other agents to perform complex tasks. There exist two types of agents: static agent and mobile agent. Static agent resides on a fix location but a mobile agent is a software program which can keep the state of itself and migrate from a node in the network to another to perform the given tasks in different locations.

3.2. Advantage of using mobile agents in IDS

There are many advantages of application of mobile agents in intrusion detection systems [12]:

- **Overcoming Network Latency:** Agents acts directly on the host so can response faster than hierarchical systems with central coordinator, when an action happens on the host.
- **Reducing Network Load:** Instead of sending huge amount of data to a central station, sending MAs will reduce network load due to eliminating this data transfer.
- **Autonomous Execution:** Agents can work autonomously even if their creators are failed.
- **Platform Independence:** Since the agents run on a special agent platform, they are independent from their resident hosts' platform.
- **Dynamic Adaption:** The system can be reconfigured at run-time due to agents' dynamic behavior.
- **Static Adaption:** The code of the agents can be updated without restarting the whole system.
- **Scalability:** Distributed MA IDS architecture can reduce computational load on the system. With the increase of computing elements in a system, agent can be dispatched to new machines in the network.

4. IMPLEMENTATION

In our proposed method both data analysis and data capturing are distributed in order to use the network resources optimally. We implemented a new grouping mechanism in which each group has a leader. Each leader in collaboration with other members of the group and other leaders aims to detect network-based

and distributed attacks. In the following parts we will describe in details our grouping mechanism and the method of attack detection.

4.1. Grouping mechanism

In distributed data processing, without any external communication method, each host has access to only its local information with no global view on the network data. So without a proper orientation mechanism in between these hosts, none of the network attacks could be detected. We used a grouping mechanism using the concept of 'Virtual LAN' (VLAN) to deal with this problem. VLANs can be configured in the network switches or routers to limit broadcast level in a LAN. So in a big LAN we can have multiple VLANs and form multiple virtual networks. So each of these VLANs can be considered as a group and one can be assigned as the group leader. Each group member communicates with only its group leader. The leaders communicate with each other using mobile agents. The IP address of the leader should be configured in the members of each group and also each leader should have a list of IP addresses of other leaders. Furthermore, this grouping mechanism has the ability to operate in multisite networks. The mechanism of attack detection and method of communication and collaboration between hosts will be described in detail in the next part.

4.2. Attack detection mechanism

Our proposed system is capable of detection of different types of distributed and network-based attacks. We implemented the proposed method of TCP scan attacks detection introduced by Treurniet [13] to test our system. Treurniet presented a simplified diagram of TCP state transition diagram modeled in RFC (http://www.ietf.org/rfc/rfc793.txt). This state transition diagram is showed in Figure 1.

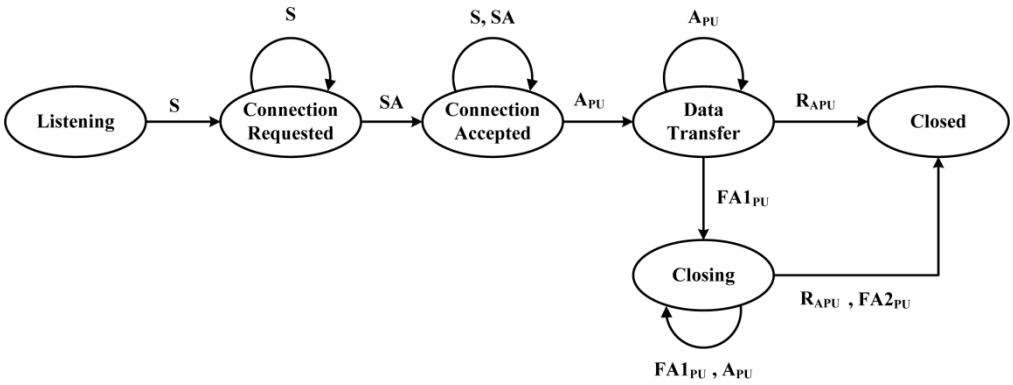

Figure 1. TCP state transition diagram [13]

In this diagram, each event will be raised with respect to different flags set in the packets of TCP connection. Table 1 lists different flags with their symbol that lead the transitions between states. A TCP connection is marked as suspicious if it does not follow this diagram states. For example, a TCP communication is suspicious if a connection in the 'Listening' state receives a TCP 'FIN' packet. For more details about attack detection see [13].

Each host captures its received data and constructs TCP connections based on TCP state transition diagram showed in Figure 2. These connections are equivalent to TCP flows. A flow is a stream of packets travelling between a particular source and a particular destination within a specific period of time. It specified by a tuple containing the source and destination IP addresses, source and destination ports and the type of protocol.

During construction of these flows the host detects suspicious flows whenever it receives a packet which does not follow TCP diagram. Then it transmits the information of that suspicious flow to the leader. The leader is responsible to collect and correlate these data to detect attacks. In this way distributed TCP scan attacks to each group can be detected by the leader.

For example, it is possible that an attacker attacks to multiple hosts in a group. Each host will send suspicious flow information to the leader after its detection. Subsequently the leader can detect the attack due to having a global view on its group data.

For detection of attacks to larger scope in the network, leaders should have a global view to all hosts in the network in addition to their group members. We used mobile agents to deal with this problem. Leaders

send mobile agents to other leaders requesting for further information about a suspicious behavior which it cannot decide on its own. Subsequently the leaders can detect attacks to multiple groups. Then they send an alarm to administrators and inform them about that attack. This way the distributed attacks to the network will be detected in a fully distributed manner and with the lowest overload on the network.

Table 1. TCP Flag Combination that Form Events in the TCP State Machine [13]

Symbol	Flag set	Event description
S	{S}	Request to open connection
SA	{SA}	Agree to open connection
A_{PU}	{A, PA, AU, PAU}	Acknowledgment of receipt
FA_{PU}	{FA,FPA, FAU, FPAU}	Request to close connection
R_{APU}	{R,RA,RP,RU,RPA,RPURAU, RPAU}	Tear down connection

5. SIMULATION PLATFORM

Experimental platform was a virtual machine with Ubuntu 10.10, Intel Core2 Quad Q9550 @ 2.83GHz and 4GB of memory. We simulated our proposed system using Network Simulator 2 (NS2) (http://www.isi.edu/nsnam/ns). NS2 is an object-oriented and discrete event simulator in network research that is widely used in different contexts [14, 15]. NS2 has a feature called Emulation. This feature helped us to inject our simulated network traffic into the simulator. The traffic we used in this simulation is described in detail in the following section. We designed two different scenarios to test our system's performance.

5.1. Simulation Traffic Specification

We have used real life recorded traffic from Simpleweb repository [16] as our normal traffic. This repository is a collection of anonymized packet headers and NetFlow data collected from various locations in the Netherlands.

We used 'Packet Headers 5' dataset (http://traces.simpleweb.org/traces/TCP-IP/location5). It is collected in a hosting-provider. At this hosting-provider, the servers are connected at 100 Mbit/s to the core network of the provider. The uplink bandwidth capacity of this hosting-provider is around 50 Mbit/s. These measurements are from December 2003 to February 2004. We have used the first five parts of these traces.

We merged these traces using mergecap tool (http://www.wireshark.org/docs/man-pages/mergecap .html) to obtain higher traffic rate. Then we replaced all of its network's IP addresses with our simulated network's IP addresses using an open source tool called tcprewrite (http://tcpreplay.synfin.net/ tcprewrite.html). The resultant dataset contained approximately 15 min of traffic with average rate of 50 Mbit/s. The resultant normal traffic's IO graph drawn by wireshark is shown in Figure 2.

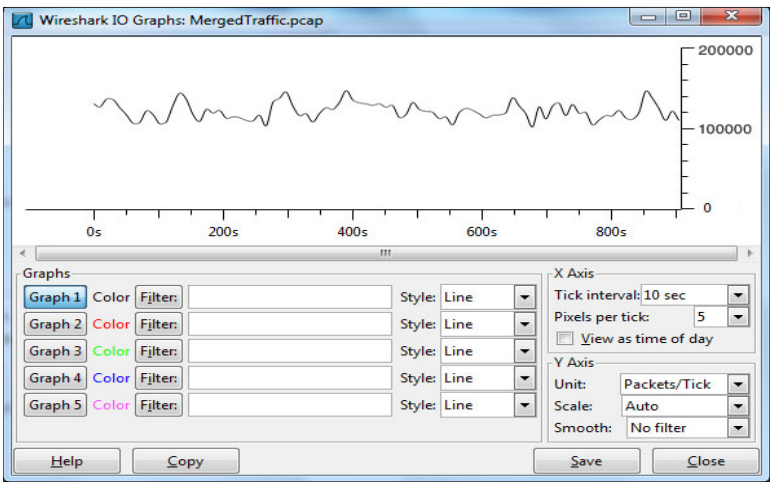

Figure 2. Normal traffic scenario from Simpleweb, X-axis denotes tick interval and Y-axis denotes packets/tick

We generated attacks dataset by using a well known tool Nmap (http://www.nmap.org) as we successfully used in our previous work [17]. Nmap is a free and open source port scanner tool that can generate different type of scan attacks. Although our system is capable of detection of many kind of known distributed attacks, but in this simulation we focused on TCP scan attacks [18]. We designed different type of TCP scan attacks in duration equal to normal traffic (15min). The generated attack traffic's IO graph draw by wireshark is shown in Figure 3. Finally we replayed this traffic along during normal traffic to simulate real world's data for input traffic to our simulated network.

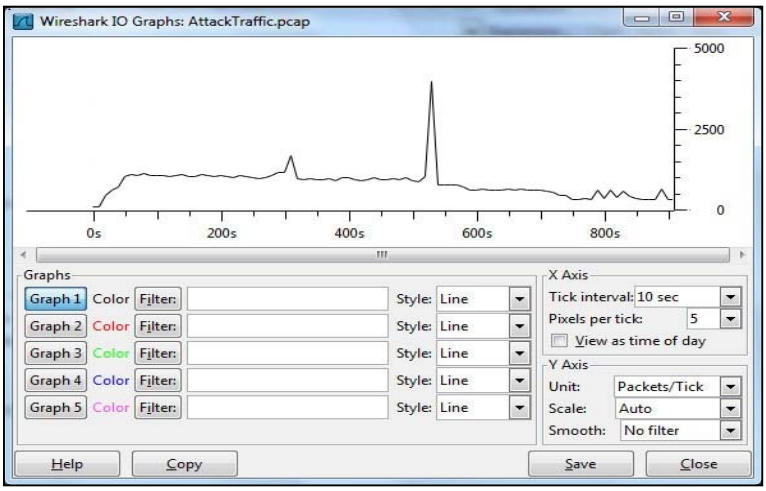

Figure 3. Attack traffic scenario from attacks generated by Nmap, X-axis denotes tick interval and Y-axis denotes packets/tick

5.2. Evaluation Scenarios

We designed two different scenarios to evaluate our systems performance in term of network load, flow loss rate and detection rate. Each scenario is described in detail in the following parts.

5.2.1. Scenario 1

In this scenario we compared our distributed IDS with the distributed IDS proposed by Erbacher et al. [5] in term of network load. We evaluated two systems with 10 different network sizes. Each network has different number of nodes (60, 120, 180, …, 600) and accordingly different number of VLANs. So we replayed our traffic and calculate total network load for each of them. The structure of simulated network is shown in Figure 4 and topology of each VLAN is shown in Figure 5.

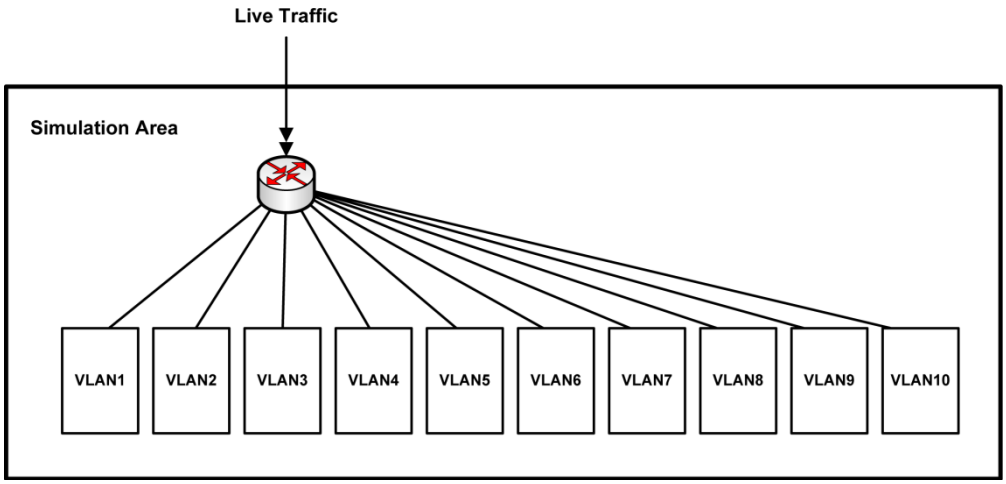

Figure 4. The structure of the simulated network

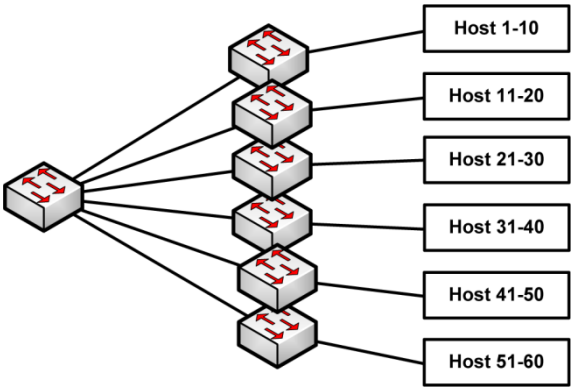

Figure 5. The topology of each VLAN

5.2.2. Scenario 2

In this scenario we aim to compare our distributed IDS with a general network-based IDS in term of flow-loss rate and detection rate.

The network we simulated in this scenario was consisted of 300 different nodes. We assigned an IP address to each node and organized these nodes in 3 different VLANs. The overall structure of this system is the same as Figure 4 and Figure 5 but with different number of VLANs and hosts.

6. EXPERIMENTAL RESULTS

Simulation result of the first scenario is shown in Figure 6. As it is obvious our proposed distributed IDS imposes very low load on the network due to its distributed nature of data capturing and processing. All of extra load is only related to bandwidth consumed by mobile agents and transmission of suspicious flow information. In Erbacher et al.'s proposed system the captured and preprocessed data by all sensors should be transmitted to a central analyzer. So the consumed network load will be grown dramatically when the network size was grown.

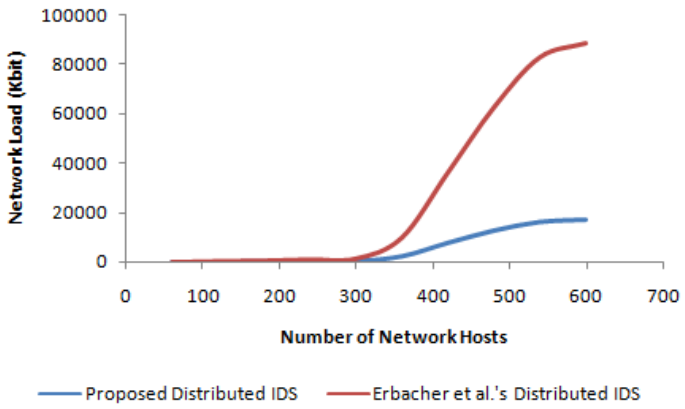

Figure 6. Network load of our system in comparision of the distributed IDS proposed by Erbacher et al.

Simulation result of the second scenario is shown in Figure 7. A network-based IDS in a high traffic rate usually cannot process all received packets and consequently drops some of the packets. This loss of packets leads to increased flow-loss rate and will reduce detection rate.

In the second scenario we also compared our systems' detection rate with the network-based IDS. The result is shown in Table 2. As it is shown, our system has the better detection rate.

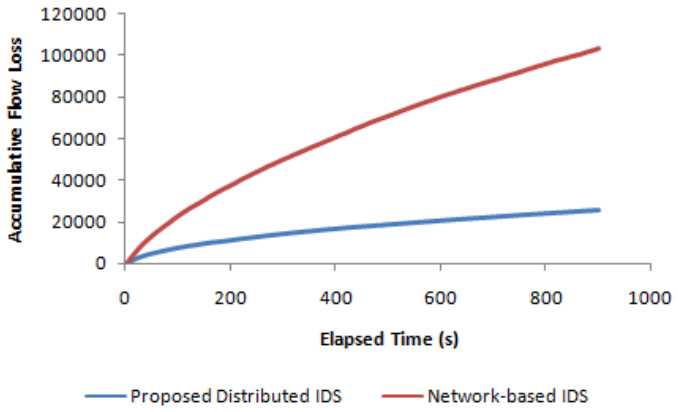

Figure 7. Flow loss rate of our system in comparision of the Network-based IDS

Table 2. Comparision of Detection Rate between Our Distributed IDS and Network-based IDS

Network-based IDS	Proposed Distributed IDS
45%	92%

7. CONCLUSION

Application of an IDS is crucial to protect networks against different types of attacks. Traditional IDSs worked centrally. Centralized IDSs suffers from single point of failure specially in high speed or large scale networks. Subsequently distributed IDSs were proposed. A distributed IDS can distribute data capturing, data analysis or both throughout the network. In this paper we proposed an efficient flow based distributed IDS which distributed both parts by optimal application of network resources (CPU and memory). We developed a new grouping mechanism in which network is divided into groups with a leader and a few members. A special communication mechanism was suggested to coordinate between group members and leader as well as between groups. We benefited from mobile agent technology in order to creating communication links between leaders to share information to make decisions about larger scope distributed attacks. We implemented our system in network simulator NS2 and compared our system with the distributed system proposed by Erbacher et al. in term of network load and with a general network based IDS in terms of flow loss rate and detection rate. The result showed that our system has a much better efficiency in high speed and large scale networks with higher detection rate of distributed attacks and with a negligible extra load on the network. As part of future work we aim to work on system's fault tolerance and test the system with other kinds of distributed attacks.

ACKNOWLEDGEMENTS

The authors would like to thank Amir Homayoun Javadi for his helpful and constructive comments.

REFERENCES

[1] C Morariu and B Stiller. "DiCAP: Distributed Packet Capturing architecture for high-speed network links". in *33rd IEEE Conference on Local Computer Networks (LCN)*, Montreal, Que. 2008: 168-175.
[2] LA Gunawan, et al. "Modeling a distributed intrusion detection system using collaborative building blocks". *ACM SIGSOFT Software Engineering Notes.* 2011; 36: 1-8.
[3] SH Han, et al. "The Architecture of NG-MON: A Passive Network Monitoring System for High-Speed IP Networks1". in *Management Technologies for E-Commerce and E-Business Applications*, ed: Springer. 2002: 16-27.
[4] Z Hakimi, et al. "An Efficient Architecture for Distributed Intrusion Detection System". in *10th International conference on Computer and Information Sciences (ISCIS)*, Iran. 2013.
[5] RF Erbacher and S Hutchinson. "Distributed Sensor Objects for Intrusion Detection Systems". in *9th International Conference on Information Technology: New Generations (ITNG)*, Las Vegas, NV. 2012: 417-424.
[6] R Sasikumar and D Manjula. "A Distributed Intrusion Detection System Based on Mobile Agents with Fault Tolerance". *European Journal of Scientific Research.* 2011; 62: 48-55.
[7] FB Ktata, et al. "Agent IDS based on Misuse Approach". *Journal of Software.* 2009; 4: 495-507.

[8] E Izkue and E Magaña. "Sampling time-dependent parameters in high-speed network monitoring". in *Proceedings of the ACM international workshop on Performance monitoring, measurement, and evaluation of heterogeneous wireless and wired networks*. 2006:13-17.
[9] H Wang, et al. "Easily-implemented adaptive packet sampling for high speed networks flow measurement". in *Computational Science–ICCS*, ed: Springer. 2006: 128-135.
[10] G He and JC Hou. "An in-depth, analytical study of sampling techniques for self-similar internet traffic". in *25th IEEE International Conference on Distributed Computing Systems (ICDCS)*. 2005: 404-413.
[11] MAdC Gatti and A Staa. "Testing & debugging multi-agent systems: a state of the art report". *Departamento de Informa tica, PUC-Rio, Rio de Janeiro*. 2006.
[12] WA Jansen, et al. *Applying mobile agents to intrusion detection and response*: US Department of Commerce, Technology Administration, National Institute of Standards and Technology. 1999.
[13] J Treurniet. "A Network Activity Classification Schema and Its Application to Scan Detection". *IEEE/ACM Transactions on Networking*. 2011; 19: 1396 - 1404.
[14] RF Malik, et al. "The New Multipoint Relays Selection in OLSR using Particle Swarm Optimization". *TELKOMNIKA Indonesian Journal of Electrical Engineering*. 2012; 10: 343-352.
[15] W Ying, et al. "A Novel Routing Protocol for VANETS". *TELKOMNIKA Indonesian Journal of Electrical Engineering*. 2013; 11: 2195-2199.
[16] RRR Barbosa, et al. "Simpleweb/university of twente traffic traces data repository". 2010.
[17] M Barati, et al. "A Novel Threshold-based Scan Detection Method Using Genetic Algorithm". in *Proceedings of the ACM international Conference on Security of Information and Networks (SIN)*, Turkey. 2013: 436-439.
[18] R Christopher. "Port Scanning Techniques and the Defense Against Them". *SANS Institute*. 2001.

BIOGRAPHIES OF AUTHORS

Zahra Hakimi was born in Karaj, Iran. She received her BSc. degree in Computer Software from Payam-e Noor University of Qazvin in September 2007. She currently is a MSc. student in IT Engineering at Azad University of Qazvin. Her research interests are network security, distributed networks and distributed traffic surveillance.
Email: hakimi@qiau.ac.ir

Karim Faez was born in Semnan, Iran. He received his BSc. degree in Electrical Engineering from Tehran Polytechnic University as the first rank in June 1973, and his MSc. and Ph.D. degrees in Computer Science from University of California at Los Angeles (UCLA) in 1977 and 1980 respectively. Professor Faez was with Iran Telecommunication Research Center (1981-1983) before Joining Amirkabir University of Technology (Tehran Polytechnic) in Iran in March 1983, where he holds the rank of professorship in the Electrical Engineering Department. He was the founder of the Computer Engineering Department of Amirkabir University in 1989 and has served as the first chairman during April 1989-Sept. 1992.
Professor Faez was the chairman of planning committee for Computer Engineering and Computer Science of Ministry of Science, Research and Technology (during 1988-1996). His research interests are Biometrics Recognition and authentication, Pattern Recognition, Image Processing, Neural Networks, Signal Processing, Farsi Handwritten Processing, Earthquake Signal Processing, Fault Tolerance System Design, Computer Networks, and Hardware Design.
Dr. Faez coauthored a book in Logic Circuits published by Amirkabir University Press. He also coauthored a chapter in the book: *Recent Advances in Simulated Evolution and Learning*, Advances in Natural Computation, Vol. 2, Aug.2004, World Scientific. He published about 300 articles in the above area. He is a member of IEEE, IEICE, and ACM, a member of Editorial Committee of Journal of Iranian Association of Electrical and Electronics Engineers, and International Journal of Communication Engineering.
Emails: kfaez@aut.ac.ir, kfaez@ieee.org, kfaez@m.ieice.org

Morteza Barati was born in Mashhad, Iran. He received his BSc. degree in Computer Software from Payam-e Noor University of Qazvin in July 2007. He currently is a MSc. student in IT Engineering at Azad University of Qazvin. His research interests are network security and artificial intelligence.
Email: m.barati@qiau.ac.ir