Multi-agent system for an adaptive intrusion detection

Cheikh Mohamed[12], Hacini Salima[2]

[1] LICUS laboratory, Computer science department, Skikda University, Algeria
d.mohamedcheikh@gmail.com
[2] LIRE Laboratory, Constantine 2 University, Constantine, Algeria
salimahacini@gmail.com

Abstract. The denial-of-service attack is to make a service unusable; this can be done by overloading the network with useless information, generally leveled against application servers or web servers. The intrusion detection systems are powerful tools for the detection of attempted attacks DOS (Denial of Service). However, they suffer from a number of problems such as high rate of false positives and negatives. In this paper, we present a new mechanism for intrusion detection DOS, based on the use of adaptive agents. This self-learning mechanism guaranteed DOS attack detection with minimal false alarms.

Keywords: Network security, Behavioral approach, DOS, Intrusion Detection System, Multi-Agent Systems.

1 Introduction

Intrusion detection is a technique of detecting unauthorized access to a computer system or a computer network. An intrusion into a system is an attempt by an outsider to illegally gain access to the system [22]. The notion of intrusion detection in computer networks is a new phenomenon born, according to many, from a 1980 James Anderson's [1]. Denning [2] designed then an intrusion detection model which marked a real impetus of the field. IDSs are essential complements to the preventive security mechanisms provided for computing systems and networks. They are used in the monitoring control process for the detection of potential intrusions and infections [21].

Intrusion detection is based on two basic approaches, the behavioral approach and the misuse approach. The misuse detection, defines in a specific way, the users' actions which constitute an abuse. It uses rules defined for the detection of known intrusions [4].

The behavioral approach, used in this paper for the detection of DOS attacks, can detect unknown intrusions and therefore does not require prior knowledge about intrusions [3], [4]. In this approach, a normal pattern of activity is defined to build a profile of user activity. Any deviation from the established norm is considered as abnormal.

The behavioral approach uses multiple detection techniques [4], [5], [6], [7]. The most common is to generate a model of normal behavior through a statistical model. Despite the observed development of these behavioral techniques, the problem of high false alarms rate, especially for DOS attacks, remains posed.

The proposed approach is based on the adaptation of the normal profile of a subject to detect the DOS attack and minimize the number of false alarms (positive and negative). Agents are used in our work to ensure three main tasks:
- Learning and detection.
- The enrichment of the normal profile as and when new legitimate activities are discovered.
- The filtering used to discover illegitimate activities in the normal profile. Note that the learning phase is not assumed safe and that filtering is used, in our approach, against the circumvention of IDS.

This paper presents an adaptive model agent-based for DOS attacks detection. This adaptation induces an update of the normal profile. Section 2 of this article describes the proposed approach. Section 3 quotes some works dealing with agents-based IDSs. Finally, Section 4 concludes the paper and suggests some perspectives to improve this work.

2 Proposed detection approach

Normal behavior of a subject cannot be fully represented, as it can change over time. That's why we designed a mechanism based on agents to adapt this behavior as it evolves.

Detection based on the behavioral approach is generally performed in the learning phase and detection phase [4]. The proposed approach provides adaptation of the normal profile (on which detection is based) created during the learning phase in two specific cases:

1. The presence of activities from a legitimate user. These activities are not known a priori by the system. They don’t exist in the initial knowledge base (already created during the learning phase). The latter will be enriched in every operation of adaptation by new legitimate activities, thereby reducing the rate of false positives.

2. The presence of activities from an illegitimate user (DOS attack), following an attempt to bypass IDS. This minimizes the rate of false negatives.

So, the normal profile must adapt positively (by adding legitimate activities) or negatively (by the removal of unlawful activities). Therefore, it can detect DOS and minimize the number of false alarms. The adaptation of the knowledge base leads to a reduction in the rate of false positives and negatives in the detection.

2.1 Proposed architecture

The proposed detection system is applied to network traffic. The various parameters observed during the detection thus involve packets routed in the network. A set of nine agents is used to distribute the DOS detection task. The observed parameters were chosen based on the work of H.Güneş Kayacık et *al* [17]. They concern a traffic network and include: (serror_rate, srv_serror_rate, same_srv_rate, service, flag, dst host same srv rate, dst host serror rate, dst host srv serror rate).

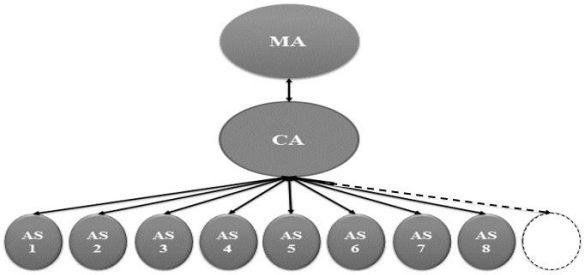

Fig.1 Overall architecture

It is characterized by three types of agents:
- Agents Sensors (ASi, i = 1, ..., n). These agents are involved in the capture of parameter values of DOS attacks, acquired from the same packet.
- The Control Agent (CA) handles the collection of agents sensors' decisions and decide on the application of adaptation.
- The Manager Agent (MA) is responsible for the overall decision making on the trigger or not of an alert, based on the decisions of ASs.

The proposed detection approach is supported by a set of assumptions:
- n: represents the number of ASs.
- (KB1, KB2, , KB9) are knowledge bases associated respectively with (AS1, AS2, ..., AS9).
- (V1, V2, , V9) represent values obtained on the same packet for the parameters (Pr1, Pr2, ..., Pr9) respectively.
- (dA1, dA2... dA9) denote the decisions of (AS1, AS2... AS9) for respectively values (V1, V2... V9).
- We denote by 1: normal packet and by 0: a confirmed DOS Attack

2.2 Intrusion Detection (DoS)

The proposed mechanism is divided into two phases: the learning phase and detection phase. The first phase is learning and the creation of KBs during a definite period of time. At the end of this phase, the system is capable of detecting DOS while allowing the system of possible adjustments to KBs.

The learning phase: It uses a statistical approach commonly used for the generation of a model of normal behavior of a subject. The task of generating the model of normal behavior (learning phase) is partitioned between ASi (i=1,2..,9). Each ASi is taking steps to model a statistical Pri on the normal profile of a packet. This modeling leads to the construction of the KBi. During this phase.

Each ASi is taking steps to model a statistical Pri on the normal profile of a packet. This modeling leads to the construction of the KBi. During this phase, each AS, builds according to the observed parameter:

1. A list of normal values in the case of a parameter not quantifiable, as number of legitimate destination port: 80, 443, etc...
2. An interval in the case of a measurable parameter, such as the length in bytes of the data source destination worm (Source Bytes) in one packet: eg [100, 250]

After the learning phase the system is ready for the detection of DOS.

The Detection phase: The detection phase brings together the various decisions of ASs to obtain a global decision that leads to the triggering or not of an alarm. Before calculating the overall decision, the decisions of ASs (equal to 0 or 1) are sent first to the CA which in turn transmits them to the MA. It performs some calculations to make an overall decision. These calculations are as follows:

After retrieving the decisions (dA1, dA2,… dA9) sent by the ASs to CA, the MA calculates the average of these decisions AD. We consider a decision range of values [0, 1] divided as follows:

[0, 0.4[(0 to 0.4), [0.4, 0.6] (0.4 to 0.6) and] 0.6, 1] (0.6 to 1) (This subdivision is just given as an example).

Where (0.4 to 0.6) represents a decision threshold to cover extreme cases (ambiguity).

MA determines to which range of values belongs AD. There are three possibilities:

AD is between 0 and 0.4 => DOS attack, or AD between 0.6 and 1 => no attack, and AD between 0.4 and 0.6 => Ambiguity.

In the third case, the number of agents who decide that a packet is normal is very close or equal to the number of agents deciding that this same packet is a DOS attack. In this case, the decision of the MA is based on the factor of confidence (FC) associated to each AS.

Two new factors are introduced Ap and An. They represent, respectively, the number of positive adaptation operations (adding in the KB) and the number of negative adaptation operations (the deletion in the KB) performed by each AS (initially An = 0 and Ap = 0). With each adaptation of ASi, its Api or Ani are incremented respectively if the adaptation is positive or negative. Each FCi (factor of confidence) is computed by the CA using the following formula:

$$FC_i = \frac{Ap_i + 2 * An_i}{\sum_{j=1}^{n} (Ap_j + 2 * An_j)} \quad (1)$$

These FCi (i=1,2,..9) give the advantage to the decision of agents that made few updates to their KBs since it is assumed that the ASs that keep their initial KBs (with little or no change) are more stable. Thus, they are better than the ASs that made many updates of their KBs.

Adaptation of the knowledge base: Two reasons induce the use of the adaptation. The first reason is that, in most cases, the learning phase is carried out in an

unreliable environment and the second reason is that the duration of the learning phase is too small in comparison with the detection system life cycle.

The adaptation of the ASs' knowledge bases is carried out during the detection phase. This adaptation is performed under the following two assumptions:

1. If the number of ASs deciding that a packet is normal is greater than or equal to SP (SP represents a Majority Threshold) and strictly less than n, an adaptation request is sent by the CA to all ASs which decide that this same packet is a DOS attack".
2. If the number of ASs deciding that a packet is a DOS attack is greater than or equal to SN (SN represents a majority threshold) and strictly less than n, an adaptation request is sent by the CA to all ASs which decide that this same packet is normal.

After the reception of the adaptation requests, each AS performs an update of its KB. This adaptation aims the KB to be enriched by new legitimate values of the normal profile or to be filtered and hence to remove some illegitimates values from KBs.

To ensure that KB created during the learning phase remains valid, ASs employ a technique to validate the addition and removal of the normal profile values of the packets. To do so, with each ASi is associated two lists of Adaptation (LPi (Positive List): Enhancements to the KB, LNi (Negative List): for delete in KBs). Both lists are initially empty. Then, whenever a Request for Adaptation (positive or negative) acquired a new value V is sent by the CA. They are two cases:

a) Positive adaptation: the ASi checks whether the number of occurrences of V in the list LPi is greater than or equal to an integer RPi associated with ASi, representing a factor validation of a Pri. The new value V must be added to KBi and removed from the LPi. By cons, if this number is less than RPi, ASi adds V to the LPi.

b) Negative adaptation: ASi checks whether the number of occurrences of V in the list LNi is greater than or equal to an integer RNi associated with ASi, representing a factor validation of a Pri. The new value V should be deleted on KBi and removed from LNi. By cons, if this number is less than RNi, ASi adds V to the LNi.

Moreover, if an AS has performed an operation of adaptation, it is imperative to send a message for CA to confirm adaptation. The adaptation is achieved by:

1. A change in a range of decision such that the interval length data in a packet. Given the decision interval [Vi, Vs] (Vi: lower value, Vs: upper value), and V a new value to add or remove:
   a) For addition:
      IF ($V \geq Vs$) THEN Vs ←V.
      IF ($V < Vi$) THEN Vi ←V.
   b) For delete:
      IF $V \geq ((Vi+Vs)/2)$ THEN Vs← (V-1) ELSE Vi←(V+1).
2. Addition of new values to a list of decisions such as the list of legitimate destination IP addresses, or removal of illegitimate values in case of negative adaptation provided that the list (KB) is not empty.

2.3 Operating principal

It is essential to assign first, to each detection parameter an AS. After this initialization, the learning phase is triggered. During this phase, the KB is generated for each AS. At the end of this phase, the detection system is put into service. At each occurrence of a packet,each ASi (according KBi and its associated parameter) acquires a value V of the associated parameter of the packet observed after the acquisition of parameter values by ASs.Each of them makes a partial decision observed on the packet (Fig. 2).

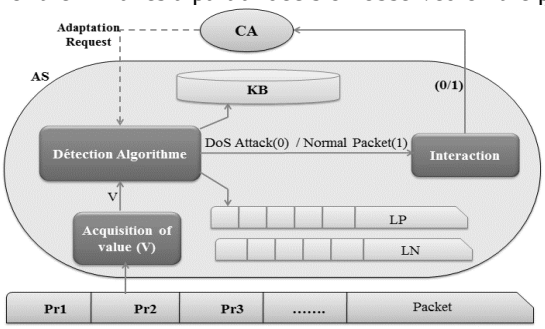

Fig.2 Architecture of AS

The second step of our detection mechanism is performed by the CA (see Fig.3).

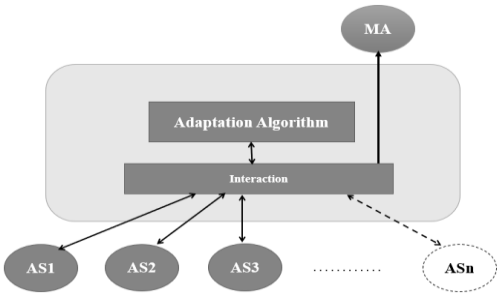

Fig.3 Architecture of CA

In parallel with the operation of adaptation, the MA (see Fig.4) decides whether the packet analyzed is normal or not.

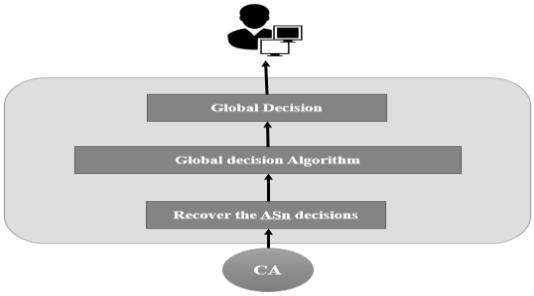

Fig.4 Architecture of MA

3 Implantation and experiments

To validate our proposition, a prototype is implemented with JADE (Java Agent Development framework). KDD 99 intrusion detection dataset is used to evaluate our proposition. It has been used to test intrusion detection systems [17].

In our experiments, we opt to detect Neptune attack which is a type of DOS attacks, and we will use the positive adaptation in ASs. So, to distinguish normal connections from Neptune attacks a set of eight agents is used to distribute the DOS detection task (see section 2.1).

First, we realized the training phase based on 5000 normal packets (taken randomly) of the 10% of KDD (see Tab.1).

	Normal	Probe	DoS	U2R	R2L
KDD 10%	97280	4107	391458	52	1124

Tab.1 The 10% KDD dataset

To accomplish this test, 2500 “Neptune” packets are randomly chosen (see Tab.2).

CategoryType (Number of samples)	
DoS	smurf (280790), neptune (107201), back (2203), teardrop (979), pod (264), land (21)
Probe	satan (1589), ipsweep (1247), portsweep (1040), nmap (231)
U2R	buffer_overflow (30), rootkit (10), loadmodule (9), perl (3)
R2L	warezclient (1020), guess_passwd (53), warezmaster (20), imap (12), ftp_write (8), multihop (7), phf (4), spy (2)

Tab.2 Category Type

The following results are obtained:

The probability of detection PDet = 1, i.e., our system has detected all Neptune attacks.

Number of Neptune packets	2500
Number of packets detected as attack	2500
Probability of detection PD	1

We also calculate the FAp (False Alarms Probability) for 2500 normal packets taken randomly.

After the test, we can notice that the system has correctly classified all normal packets and consequently the Pfa = 0.

From the experimental results, it can be concluded that our system has given very interesting results.

4 Related works

Several studies have investigated the use of agents in intrusion detection systems. We cite as an example, the project 'The computer immunology "[8] [9], JAM (Java Agent for Meta-learning) [10], and AAFID (Autonomous Agents for Intrusion Detection) [11]. Each project addresses the problem of intrusion detection in different ways.

The project 'The computer immunology "explores the design of an intrusion detection system on the basis of the idea of biological immune systems. He helped develop a sense of autonomy of agent security for computer programs by observing the normal set of system calls executed by them. The JAM project uses distributed Java agents and the concept of "data mining" to learn models of fraud and intrusive behavior. The project AFFID, meanwhile, has developed an intrusion detection system using agents for prototyping and cross-platform compatibility. The project also analyzed the agent-based approach to intrusion detection. The environment AFFID is agent-based intrusion detection, capturing and correlating data to perform intrusion detection.

Kannadiga and Zulkernine [12] presented a new intrusion detection system called DIDMA (Distributed Intrusion Detection using Mobile Agents). It uses a set of mobile agents that can move from one node to another in a network, and perform the task of aggregation and correlation of data that has a relationship with the intrusion. These data are from a different set of static software entities called agents. Chong-jun and Shi-fu [13] also presented a model for BDI agents used, and achieved a detection system based on multi-agent cooperation which effectively improves the traditional distributed intrusion detection. Vongpradhip Plaimart and [14] proposed a new architecture capable of handling the attack over the network using agent technology with the design of the network topology. Boughaci et al. [15] have established a mechanism for distributed intrusion detection based on autonomous agents and mobile. These agents are used mainly to cover information from the audit logs, to define the behavior of users, analyze data, and send alert messages in case of intrusion detection. Chung Ming-Or [20] proposed an IDS inspired by the theory of risk of the immune system.

More recently, Louati and B.Ktata [23] propose a deep learning-based multi-agent system for intrusion etection which combines the desired features of multi-agent system approach with the precision of deep learning algorithms. The performance of this model is compared against traditional machine learning approaches and other multi-agent system-based systems.

There are relatively few studies concerned with the detection-based adaptive agents [16][21][24]. For example, the work of Carb'o et al. [16] proposes an agent system based on the prediction of intrusions. There are also works that treated the detection of DOS / DDOS [18], [19].

All the presented works offer improvements in the detection and demonstrate the effectiveness of the use of agents in intrusion detection. Most of these works are based on a special mechanism that organizes and directs the agents to accomplish the detection task. However, the high false alarm rate remains a challenge for systems based specifically on the use of the behavioral approach. The intrusion detection mechanism proposed in this paper, minimizes the positive and negative false alarm rates and enhances detection by use of a self-adaptive learning.

5 Conclusion

The mechanism of intrusion detection DOS presented in this paper is based on the use of a set of agents. These agents allow the detection of DOS attack, but also the adaptation of positive and negative profile of normal behavior. This allows both minimizing the rate of false alarms and improving detection of real DOS attacks. The proposed detection system is mainly applied to the DOS attack, but it can also be used to detect other network attacks. To improve the detection rate, we consider assigning each parameter a weight representing its importance. We consider, in fact, that the importance of certain parameters is such that their faults should trigger a warning.

References

1. J. Anderson, 1980, "Computer security threat monitoring and surveillance".
2. D.E. Denning, ,1987,"An intrusion-detection model", IEEE Transactions on software engineering, SE-13: 222– 232.
3. A. Sundaram., 1996, "An Introduction to Intrusion Detection", Technical Report, Purdue University.
4. K. Boudaoud, 2000, "Détection d'intrusions : Une nouvelle approche par systèmes multi-agents", thèse de doctorat de l'école Polytechnique Fédérale de Lausanne.
5. K. Ilgun, 1992, "USTAT: A Real-time Intrusion Detection System for UNIX", Master thesis, University of California.
6. A. Mounji, 1997, "Rule-Based Distributed Intrusion Detection", Thèse de l'université de Namur, Institut d'Informatique, Belgique.
7. S. Kumar, 1995, "Classification and Detection of Computer Intrusions", Technical Report Coast-TR-95-08, Coast Laboratory, Purdue University.
8. S. Forrest, S. A. hofmeyr and A. somayaji, 1997, "Computer immunology",Communications of the ACM, 40(10) :88-96 ; Octobre.
9. C. Warrender, S. Forrest, and B. Pealmutter, 1999, "Detecting intrusions using system calls : Alternative data models ",In Proceedings of the 1999 IEEE Symposium on Security and privacy, pages 133-145. IEEE Press.
10. S. J. Stolfo, A. L. Prodromidis, S. Tselepis, W. Lee, D. Fan, and P. k. Chan, August 1997 ,"JAM : Java Agents for meta-learning over distributed databases", In Proceedings of the 3rd International Conference on knowledge discovery and Data Mining, Pages 74-81, Newport Beach, CA, USA.

11. J. Balasubramaniyan, J. O. Gracia-Fernandez, D. Isacoff, E. H. Spafford, and D. Zamboni, 1998 ,“ An architecture for intrusion detection using autonomous agents”, Technical Report COAST TR 98-05, Purdue University Departement of computer Sciences.
12. P. Kannadiga and M. Zulkernine, 2005, " DIDMA: A Distributed Intrusion Detection System Using Mobile Agents ", First ACIS International Workshop on Self-Assembling Wireless Networks (SNPD/SAWN’05) IEEE.
13. W. J. W. Chong-jun and W. J. C. Shi-fu, 2006, "Dynamic Hierarchical Distributed Intrusion Detection System Based On Multi-Agent System", 2006 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT 2006 Workshops)(WI-IATW'06) .
14. S. Vongpradhip and W. Plaimart, 2007, "Survival Architecture for Distributed Intrusion Detection System (dIDS) using Mobile Agent", Sixth IEEE International Symposium on Network Computing and Applications (NCA 2007).
15. D. Boughaci, H. drias, A. Bendib, Y. Bouznit and B. Benhamou, 2006,"A Distributed Intrusion Detection Framework based on Autonomous and Mobile Agents", International Conference on Dependability of Computer Systems (DEPCOS-RELCOMEX’06) IEEE.
16. J. Carb´o, A. Orfila and A. Ribagorda, 2003, "Adaptive agents applied to intrusion detection", international Central and Eastern European conference on multi-agent systems No3, pp. 445-453.
17. H. Güneş Kayacık, A. Nur Zincir-Heywood, Malcolm I. Heywood, "Selecting Features for Intrusion Detection: A Feature Relevance Analysis on KDD 99 Intrusion Detection Datasets ", Third Annual Conference on Privacy, Security and Trust, October 12-14, 2005.
18. J. Mirkovic and P. Reiher, , July 2005, “D-WARD: a source-end defense against flooding Denial-of-Service attacks,” IEEE Transactions Dependable and Secure Computing, vol. 2,no.3, pp. 216-232.
19. J. Mirkovic, M. Robinson, P. Reiher, and G.. Oikonomou, 2006, “A framework for collaborative DDoS defense,” Proc. of the 22nd Annual Computer Security Applications Conference, pp. 33-42.
20. C.-M. Ou, 2012, “Host-based intrusion detection systems adapted from agent-based artificial immune systems”,Neurocomputing doi:10.1016.
21. Salima Hacini, Zahia Guessoum, Mohamed Cheikh, 2013, ‘’False Alarm Reduction Using Adaptive Agent-Based Profiling’’, International Journal of Information Security and Privacy, 7(4), 53-74, (October-December 2013).
22. Kizza J.M. (2020) System Intrusion Detection and Prevention. In: Guide to Computer Network Security. Texts in Computer Science. Springer, Cham. https://doi.org/10.1007/978-3-030-38141-7_13, 2020
23. Louati, F., Ktata, F.B. A deep learning-based multi-agent system for intrusion detection. SN Appl. Sci. 2, 675 (2020). https://doi.org/10.1007/s42452-020-2414-z, 2020
24. Hacini, Salima & Guessoum, Zahia & Cheikh, Mohamed. (2018). False Alarm Reduction: A Profiling Mechanism and New Research Directions. 10.4018/978-1-5225-5583-4.ch012. 2018