P2P-AIS: A P2P Artificial Immune Systems Architecture for Detecting DDoS Flooding Attacks

Karim Ali
David R. Cheriton
School of Computer Science
University of Waterloo
Waterloo, Ontario
karim@uwaterloo.ca

Issam Aib
David R. Cheriton
School of Computer Science
University of Waterloo
Waterloo, Ontario
iaib@uwaterloo.ca

Raouf Boutaba
David R. Cheriton
School of Computer Science
University of Waterloo
Waterloo, Ontario
rboutaba@uwaterloo.ca

***Abstract*—The Human Immune System (HIS) plays an important role in protecting the human body from various intruders ranging from naive germs to the most sophisticated viruses. It acts as an Intrusion Detection and Prevention System (IDPS) for the human body and detects anomalies that make the body deviate from its normal behavior. This inspired researchers to build Artificial Immune Systems (AISes) which imitate the behavior of the HIS and are capable of protecting hosts or networks from attacks. An Artificial Immune System (AIS) is capable of detecting novel attacks because it is trained to differentiate between the normal behavior (self) and the abnormal behavior (non-self) during a tolerization (i.e training) period. Although several AISes have been proposed, only a few make use of collaborative approaches. In this paper we propose P2P-AIS, a P2P approach for AISes in which peers exchange intrusion detection experience in order to enhance attack detection and mitigation. P2P-AIS implements Chord as a distributed hash table (DHT) protocol to organize the peers.**

***Index Terms*—intrusion detection, artificial immune systems, peer-to-peer systems, distributed denial of service.**

I. Introduction

Some anomaly-based IDSes [1], [2] rely on predicting the future behavior of the system given its history. Although these approaches are successful in capturing changes in the normal behavior of a system, they require a longer training period and, in some cases, are infeasible to apply because of the size of data sets involved [3]. This led to a new approach inspired by the HIS to enhance the efficiency of anomaly detection, AISes. The HIS can adaptively detect and defend against harmful and previously unseen invaders which perfectly suits the purpose of anomaly-based IDS. Other properties of the HIS that are desirable in anomaly-based IDSes are given by [4] stating that the HIS is:

- Distributed: components interact locally to provide global protection, so there is no central control and hence no single point of failure.
- Dynamic: individual components are continually created, destroyed, and circulated throughout the body.
- Error tolerant: the effect of any single HIS action is small, so a few mistakes in classification and response are not catastrophic.
- Adaptable: the HIS can learn to recognize and respond to new infections and retain a memory of those infections to facilitate future responses.
- Autonomous: no outside control is required.
- Robust: is a consequence of being diverse, distributed, dynamic and error tolerant.

Therefore, we believe that AISes can be more effective if their distributed aspect is implemented through a p2p infrastructure, where peers share intrusion detection information among each other. This will allow peers who encountered an attack to notify other peers of such an attack to take the necessary precautions. Consequently, the propagation of the attack to other peers in the network can be mitigated. Moreover, peers can exchange information about novel attacks with other peers that might not necessarily have encountered those attacks before.

In this paper we propose a P2P Artificial Immune System (P2P-AIS) in which peers exchange intrusion detection experience in order to enhance attack detection and mitigation. The rest of the paper is divided as follows. Section II explains some of the related work done in the field. Section III gives some necessary background information about different DDoS attacks and the parameters that can be used for their detection. In section IV we describe our p2p approach to AIS. We provide an evaluation and analysis of our P2P-AIS in section V. Section VI concludes the paper and gives some future research directions.

II. Related Work

There are several AIS approaches developed to detect intrusions that deviate from the normal behavior of a system. LISYS [4] is an example of how HIS can be applied to computer security. It uses negative selection [5] mechanisms in order to differentiate between self (normal) behavior and non-self (abnormal) behavior. LISYS was successful in identifying unauthorized TCP connections in a broadcast network. However, there are several characteristics of LISYS that limits its usage in anomaly detection. First, it only monitors TCP connections so any intrusions done using other protocols will evade the detection mechanism. In addition, it assumes that the monitored network is a broadcast network. Therefore, it will not be useful for switched networks like Local Area Networks (LANs). Moreover, LYSIS characterizes normal behavior as

978-1-4244-4624-7/09/$25.00

TCP datapaths [4] which will not be suitable to detect attacks against some systems like web servers which are expected to receive connections from multiple changing sources.

An immunity-based approach to characterize intrusions in computer networks is described in [3] where two techniques are explained in the work: Positive Characterization (PC) and Negative Characterization (NC). The results show that the NC approach is more useful for detector generation in AISes as it was able to detect 87.5% of the attacks with a maximum false alarm rate of 1% while only using 10% of the resources consumed by the PC technique.

A cooperative AIS is suggested in [6] based on a multi-agent system. The system utilizes dynamic collaboration between individual AIS agents to address the well-known false positives problem in anomaly detection. However, IDS detectors are generated in a pseudorandom way leading to a poor coverage of non-self (abnormal) space. Moreover, the system has a high false positives rate of 18%. Other cooperative approaches were suggested like multi-agent systems [7], [8] and p2p approaches [9], [10]. However, those techniques are not capable of defending the network against novel attacks as opposed to AISes.

III. Detecting DDoS Attacks

There are various types of DDoS attacks, each has its own characteristics and behavior. Therefore, the parameters that need to be monitored to detect DDoS attacks differ according to the type of attack. This section discusses, due to space limitations, IP spoofing and TCP SYN Flooding attacks and the parameters that can successfully detect them.

A. IP Spoofing

An adversary tries to manipulate the source IP address in packets used to launch a DDoS attack unless enough hosts have been compromised to be used in the attack [11]. Usually, spoofing the source IP address results in using IP addresses that do not exist in the network creating half-open connections [12]. That eventually can lead to a DoS attack if the attacker floods the victim machine with overwhelming amounts of traffic [13] without caring about receiving responses to his attack packets. Moreover, this helps in hiding the real location of the attacker. Therefore, IP spoofing is primarily used in DDoS attacks to disguise agent machines or to perpetrate reflector attacks [14], [15]. Defending against such type of attacks can be achieved at the gateway to a network by blocking packets from outside the network with a source address inside the network. This will prevent an attacker forging the IP address of an internal machine in that specific network. However, this will not prevent attackers already existing inside the network from spoofing IP addresses of other hosts in the network. Therefore, blocking packets at the gateway does not competely solve the problem. In order to successfully detect IP spoofing attacks targeting a host in a network, two parameters need to be monitored during a detection window w_i:- (1) the total number of source IPs sIP_i that the host encounters and (2) the number of new source IPs nIP_i that the host encounters.

B. TCP SYN Flooding

More than 90% of DoS attacks use TCP SYN flooding [15] leading to the disruption of many popular services over the Internet like eBay and Yahoo [16]. TCP SYN flooding exploits the nature of the three-way-handshake of TCP. Mainly, it results from opening multiple TCP connections with the victim using a spoofed IP address which forces the victim to reply back with the SYN/ACK packet but to a non-existing entity. Therefore, the connection remains half open for the whole timeout period of the TCP connection which is typically set to 75 seconds [15]. Establishing a lot of half-open connections eventually exhausts the victim resources leading to the temporary denial of future TCP requests. An important parameter that helps detect TCP SYN flooding attacks is the rate of half-open TCP connections during a certain period. In other words, in a detection window w_i (probably less than 75 seconds) we should monitor $tSYN_i$ the total number of TCP SYN packets initiating new TCP connections (i.e TCP SYN packets of fragment offset equal to zero).

IV. P2P-AIS

P2P-AIS aims at detecting DDoS attacks using an AIS approach and sharing this intrusion detection information with peers in the system so that they can take necessary precautions to defend against such attacks. We will first explain the working mechanism of the artificial immune component of the system then explain how peers will exchange the intrusion detection information.

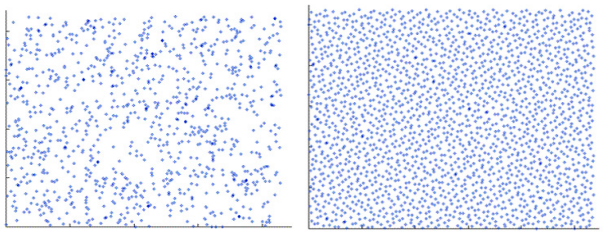
A. AIS Detectors

Detectors in P2P-AIS are the probes that check incoming traffic for anomalies. Each detector d will be represented as an array of bits of fixed length l. Detectors are randomly generated during the training phase of P2P-AIS using Quasi-Random generation algorithms [17] to help cover more non-self patterns (as opposed to random algorithms) using the same number of generated detectors. Fig. 1a shows the coverage generated by a detector population of size 100,000 using a uniform random algorithm, while Fig. 1b shows the same result using quasi-random generation. Notice that the latter provides a better distribution.

Each detector has 5 different states during its life time: *immature, mature, active, memory, and dead*. The life cycle of a detector in P2P-AIS is similar to that explained in [4].

B. Monitored Parameters

P2P-AIS needs to monitor some parameters and characteristics of the incoming network traffic to detect any malicious behavior or intrusions. Those parameters depend on the type of attack that we are interested in detecting. We will focus on the DDoS flooding attacks described in section II. Table I shows those types of attacks and the corresponding parameters that we need to monitor per detection window w_i in order to detect them successfully.



(a) (b)

Fig. 1: (a) uniform random *vs.* (b) quasi-random generation.

TABLE I: Various Flooding attacks and their detection parameters

Flooding Attack	**Monitored Parameters**
IP Spoofing	sIP_i - total number of source IP addresses in w_i
	nIP_i - the number of new source IP addresses in w_i
TCP SYN	$tSYN_i$ - the total number of half-open TCP connections in w_i
ICMP Echo	$tICMP_i$ - the total number of ICMP echo requests received in w_i
UDP	$tUDP_i$ - the rate of UDP packets in w_i

C. Mapping Parameters to Bit Strings

Having defined the parameters we need to monitor, we then map their values to bit arrays in order to be used in the representation of the detectors used for the AIS component of P2P-AIS. The number of bits that will be used to represent each parameter value in the detector will depend on the maximum values for such parameters so that the designated bits will accommodate all possible values. The maximum values will be derived from analyzing the training dataset of DARPA 1999 [18]. We are interested in the inside tcpdump files that contain the network traffic of the victim machines inside the simulated network. The first week of the data contains no attacks so it will give us a good insight of the maximum possible values of the parameters that we need to monitor. We analyzed the tcpdump files using Wireshark Network Analyzer [19]. Fig. 2 shows the distribution of various types of packets sent to a single host (hobbes.eyrie.af.mil) extracted from the data of the first day in the DARPA 1999 dataset.

D. Matching Method

Incoming traffic will be represented as feature vectors that have the same representation of the detectors as previously explained. However, those feature vectors will represent real data that needs to be monitored and not randomly generated strings as in the case for the detectors. In order to detect anomalies, feature vectors need to be matched against detectors. Several methods can be used [3], [6] to formulate the matching method between the detectors and feature vectors. We opted for using the hamming distance [4] because it involves less calculation overhead.

E. Learning Method

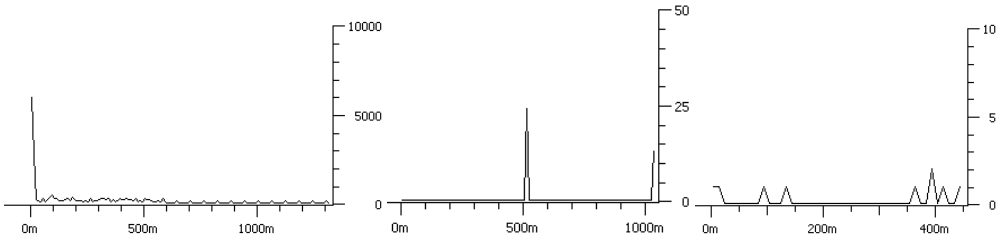
In order to make use of the previous experience in detecting anomalies, each peer in P2P-AIS maintains its memory detectors. During the tolerization period, if a memory detector matches any of the self patterns it will be discarded because this signals that there is a change in the normal behavior of the system. Therefore, there is no need to keep that memory detector anymore and a new immature detector will be generated. This helps P2P-AIS adapt to changes in normal traffic behavior, hence minimizing the possibility of raising false alarms. Maintained memory detectors will be shared among peers so that peers can benefit from the detection experience of one another.

F. P2P Overlay

P2P-AIS aims at sharing detection information among peers supporting the AIS component in order to enhance each peer detection and help in promptly mitigating propagating attacks. P2P-AIS uses Chord [20] as a p2p overlay for detection information exchange. The algorithms for node join and leave are essentially the same as in Chord, but with a difference in the exchanged peer information. Only memory detectors are shared between peers on the Chord ring. A memory detector D of size n and representation (d_1, d_2, ..., d_n) is replicated to peers of ID equal to: $(d_1, d_2, \ldots, d_n)\ mod\ 2^m$, where m ($\leq\ n$) determines how many memory detectors each peer should store and make available for all peers on the Chord ring. When m increases each peer will store and share a smaller number of memory detectors. For example, if a peer P has a memory detector D = (1,0,1,0,0,1,1,1,0,1,1,1,0,1) and m = 3, P will send D to the peer of ID = (1,0,1,0,0,1,1,1,0,1,1,1,0,1) mod 23 = (1,0,1) = 5. The peer with ID 5 will be responsible for all detectors that have the values corresponding to (x,x,x,x,x,x,x,x,x,x,1,0,1) where x = 0 or 1.

V. Evaluation

In order to evaluate P2P-AIS, we intend to use the DARPA 1999 intrusion detection dataset. Although other datasets exist, this one is more appropriate to our system as it involves 2 weeks (week 1 and week 3) with no attacks involved while the other 3 weeks have several attacks injected into the traffic. The weeks without attacks will serve as the training dataset that will be used during the tolerization period while generating the detectors. The other weeks will be used to test the viability



(a) (b) (c)

Fig. 2: (a) UDP packets received per second (b) TCP packets received per second (c) ICMP packets received per second.

of the system in detecting intrusions. Particularly, week 5 will be used to test if P2P-AIS is able to detect novel attacks or not as the dataset for that week contains 18 attacks that are not represented in the previous datasets. In order for the peers to have different experiences, each peer will be trained with the dataset of one day from the first week. This will make the detector set generated at one peer different from those generated at other peers giving the opportunity for the peers to make use of the exchanged intrusion detection information. After the tolerization period is over, the peers will be subject to the dataset injected with attacks for all the days of the second week. The data of the third week will be used the same way as that of the first week to enhance the experience of each peer and its ability to generate more useful detectors. The datasets of the fourth and fifth week will be used to measure the ability of the system to detect novel attacks.

VI. Conclusion and Future Work

The HIS inspired researchers to build Artificial Immune Systems (AISes) capable of protecting hosts or networks from novel attacks. An Artificial Immune System (AIS) is capable of detecting novel attacks because it is trained to differentiate between the normal behavior (self) and the abnormal behavior (non-self) during a tolerization (i.e training) period. In this paper, we proposed a p2p acrchitecture for AISes that can be used to defend against DDoS flooding attacks. We provided a list of network parameters (Table I) that should be monitored by P2P-AIS to defend against the set of flooding attacks we considered in section II. In P2P-AIS, peers on the Chord ring exchange intrusion detection experience in order to enhance attack detection and mitigation. We are planning to design a prototype for P2P-AIS and evaluate the system as indicated in section V. Moreover, we will examine the possibility of using the architecture of P2P-AIS to help defend against worm propagation.

References

[1] P. D'haeseleer, S. Forrest, and P. Helman, "An immunological approach to change detection: algorithms, analysis and implications," May 1996, pp. 110–119.

[2] T. Lane and C. E. Brodley, "Temporal sequence learning and data reduction for anomaly detection," *ACM Transactions on Information System Security*, vol. 2, no. 3, pp. 295–331, 1999.

[3] D. Dasgupta and F. Gonzalez, "An immunity-based technique to characterize intrusions in computer networks," *IEEE Trans. Evol. Comput.*, vol. 6, no. 3, pp. 281–291, Jun. 2002.

[4] S. A. Hofmeyr and S. A. Forrest, "Architecture for an artificial immune system," *Evolutionary Computation*, vol. 8, no. 4, pp. 443–473, 2000.

[5] F. Esponda, S. Forrest, and P. Helman, "A formal framework for positive and negative detection schemes," *IEEE Trans. Syst., Man, Cybern. B*, vol. 34, no. 1, pp. 357–373, Feb. 2004.

[6] K. Luther, R. Bye, T. Alpcan, A. Muller, and S. Albayrak, "A cooperative ais framework for intrusion detection," Jun. 2007, pp. 1409–1416.

[7] D. Dasgupta, F. Gonzalez, K. Yallapu, J. Gomez, and R. Yarramsettii, "Cids: An agent-based intrusion detection system," *Computers & Security*, vol. 24, no. 5, pp. 387–398, 2005. [Online]. Available: http://www.sciencedirect.com/science/article/B6V8G-4FSX676-1/2/36f8a8e3757b82db6b512f9d4489575d

[8] K. Juszczyszyn, N. Nguyen, G. Kolaczek, A. Grzech, A. Pieczynska, and R. ł aw Katarzyniak, "Agent-based approach for distributed intrusion detection system design," *Computational Science*, vol. 3993, pp. 224–231, may 2006.

[9] C. Duma, M. Karresand, N. Shahmehri, and G. Caronni, "A trust-aware, p2p-based overlay for intrusion detection," 2006, pp. 692–697.

[10] R. Janakiraman, M. Waldvogel, and Q. Zhang, "Indra: a peer-to-peer approach to network intrusion detection and prevention," Jun. 2003, pp. 226–231.

[11] J. Mölsä, "Mitigating denial of service attacks: a tutorial," *Journal of Computer Security*, vol. 13, no. 6, pp. 807–837, 2005.

[12] C. E. R. Team, "Cert advisory ca-96.21: Tcp syn flooding and ip spoofing attacks," CERT, Tech. Rep., 1996. [Online]. Available: http://www.cert.org/advisories/CA-1996-21.html

[13] P. Ferguson and D. Senie, "Network ingress filtering: Defeating denial of service attacks which employ ip source address spoofing," RFC 2827 (Best Current Practice), May 2000, updated by RFC 3704. [Online]. Available: http://www.ietf.org/rfc/rfc2827.txt

[14] R. Chang, "Defending against flooding-based distributed denial-of-service attacks: a tutorial," *IEEE Commun. Mag.*, vol. 40, no. 10, pp. 42–51, Oct. 2002.

[15] D. Moore, C. Shannon, D. J. Brown, G. M. Voelker, and S. Savage, "Inferring internet denial-of-service activity," *ACM Transactions on Computer Systems*, vol. 24, no. 2, pp. 115–139, 2006.

[16] H. Wang, D. Zhang, and K. G. Shin, "Detecting syn flooding attacks," vol. 3, Jun. 2002, pp. 1530–1539.

[17] H. Niederreiter, *Random number generation and quasi-Monte Carlo methods*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 1992.

[18] R. Lippmann, J. W. Haines, D. J. Fried, J. Korba, and K. Das, "The 1999 darpa off-line intrusion detection evaluation," *Computer Networks*, vol. 34, no. 4, pp. 579–595, 2000.

[19] A. Orebaugh, G. Ramirez, J. Burke, and L. Pesce, *Wireshark & Ethereal Network Protocol Analyzer Toolkit (Jay Beale's Open Source Security)*. Syngress Publishing, 2006.

[20] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan, "Chord: A scalable peer-to-peer lookup service for internet applications," in *SIGCOMM '01: Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications*. New York, NY, USA: ACM, 2001, pp. 149–160.