A multiagent approach for detecting and mitigating DDoS attacks

João P. A. Pereira, Marcos A. Simplicio Jr.,Anarosa A. F. Brandão
Dept. of Computer and Digital Systems Engineering
Escola Politecnica, Universidade de São Paulo (USP), São Paulo/SP 05508–010
Email: {raijoma,mjunior}@larc.usp.br, anarosa.brandao@poli.usp.br

***Abstract*—This paper describes Arquitena, a multiagent approach for detecting and mitigating DDoS attacks in Internet Services Providers (ISP) networks. Arquitena's main property is to identify situations that characterize attack scenarios, such as a large stream of packets directed to a network service or equipment. This is accomplished by using a virtual network of agents that mirrors the actual network infrastructure, which tends to facilitate the detection of attack routes, identification of malicious traffic and protection of hypothetical victims. Together with the system's description and rationale, we describe our preliminary prototype that will be employed for its evaluation.**

***Keywords*—multiagent systems; distributed denial of service attacks; detection; mitigation; internet service providers.**

I. Introduction

Modern society is increasingly dependent on the Internet services, including the speed of the links and data availability. Internet access by companies and final users is usually done indirectly, through the infrastructure of an Internet Service Provider (ISP) that charges for this service. This dependence puts great responsibility upon the ISP, since any network unavailability can seriously affect the transactions of many of the ISP's clients (e.g., causing financial and image losses). One example of threat against network availability are the so-called Distributed Denial of Service (DDoS). Albeit diverse in nature (for a taxonomy, see [12]), such attacks usually involve a large number of entities that, by directing a large volume of traffic to a single target, compromise its capability of providing services to legitimate users.

DDoS attacks are currently among the most serious threats to the infrastructure of critical services from an ISP [1]. This is especially troubling considering that they have been gradually growing in number and volume in the last few decades [1]. In addition, among the many forms of DDoS techniques, very few can be handled by the victim alone. Aiming to tackle this issue, many detection and mitigating strategies against DDoS attacks have been conceived. This includes strategies that try to create a defense perimeter around the target [5] or rely on statistical analyzes to detect DDoS attacks [11]. A more recent technique to address this challenging scenario, however, is to employ Multiagent Systems (MAS). The advantage of using MAS is related to the difficulty of blocking DDoS attacks without using a highly distributed mechanism that covers all regions of the ISP's network [10]. In this context, using MAS becomes an interesting approach to deal with such problem, since it naturally deals with intensely dynamic environments.

The goal of this study is, thus, to employ a multiagent approach in the design of a system able to detect, mitigate and block DDoS attacks. The agents' intelligence allows them to react to changes perceived in their surrounding environment, such as a large stream of packets going to a same target (packet flooding). The resulting MAS solution consists of a self-coordinated group of agents that represent the ISP's network elements. These agents communicate among themselves and also with the underlying equipment, detecting unusual network patterns and acting accordingly. Among the characteristics aimed by Arquitena, the most fundamental are its ability to: (i) provide fast detection, (ii) identify most types of DDoS attacks, (iii) provide service continuity for legitimate traffic, (iv) achieve low false-positive and false-negative rates, and (v) impose a low operation overhead. The system runs on a simulation environment, called Delos, that is created according to the real network topology of the target ISP.

The rest of this document is organized as follows. Section II provides a brief overview of DDoS attacks. Section III presents in details the Arquitena system and the Delos simulation environment. The performance metrics to be used in Arquitena's evaluation are covered in section IV. Section V discusses the related work. Finally, section VI concludes the discussion and suggest ideas for future work.

II. Background: DDoS Attacks

A Distributed Denial of Service (DDoS) attack corresponds to an attempt to compromise the availability of some service, hindering or blocking completely its ability to handle legitimate users' requests [14, Chapter 7]. Specifically, such attacks attempt to exhaust some critical resource upon which the target service depends. For example, an attacker can use many machines to direct a large amount of spurious requests addressed to a specific target, effectively flooding the target's equipment (e.g., a server or an access router) with packets. The target equipment, unable to handle so many requests in such a short interval of time, may either collapse completely or start dropping a large portion of the received packets. Whichever the case, this affects the service provider's ability to respond to requests from legitimate users in a timely manner, thus preventing most (or all) legitimate users from accessing the service.

As shown in Figure 1, DDoS attacks are often accomplished by a large number of machines, which are recruited and controlled by some malicious entity and work in synchronization to attack a particular service or device. The participating machines typically do not belong to the attacker him/herself, but are controlled after its security is compromised (e.g., by

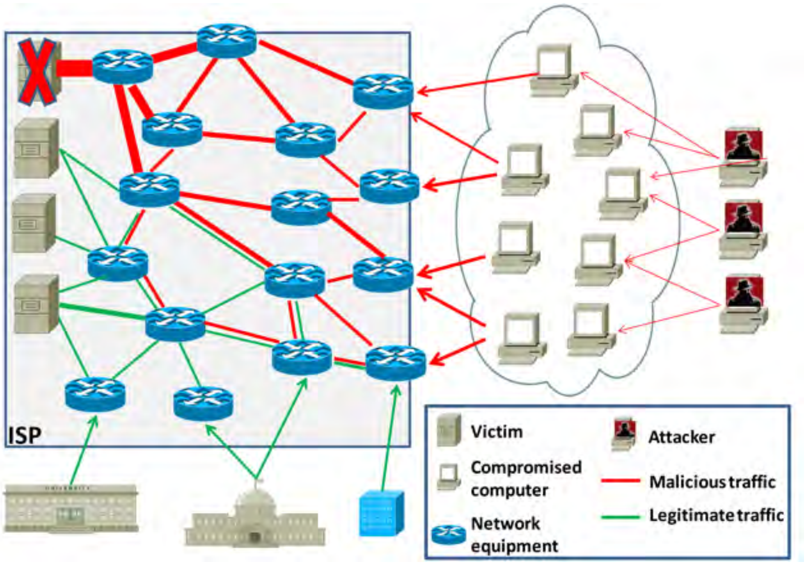


Fig. 1. A botnet-based DDoS attack. Thicker lines indicate a higher volume of traffic.

a virus or trojan horse), forming what is called a botnet [14, Chapter 6]. The higher the number of machines involved in the attack, the larger the traffic that reaches the target and, thus, the higher the chances of collapsing the service.

Preventing DDoS attacks is a challenging task, especially because it is difficult to differentiate legitimate from malicious packet flows. After all, in both cases the packets carry the IP address of the victim as their destination and they are all directed to a valid service; especially in the case of botnets, they also carry a valid IP as their source address. The main difference is, thus, in the *number* of packets in each flow, as an attacker is bound to send a huge amount of traffic to the victim in a short period of time.

Even if one is able to identify malicious packets, blocking their source while preserving legitimate traffic (e.g., by means of authentication policies, packet filtering and application of firewalls) remains a challenging issue. This happens because a malicious packet flow can only be effectively blocked if this is done near its point of origin, where the volume of traffic is still small enough to be handled with small impact over other (legitimate) packets. Otherwise, any piece of equipment trying to block the traffic would have to deal with the large amount of traffic that accumulate in the links near the intended victim, becoming itself vulnerable to a service failure due to excessive processing. Therefore, it is necessary to determine the path of malicious packet from the network's edge to the possible victim (the so-called "attack route"), applying filtering mechanisms on the relevant network equipment in a dynamic manner. Addressing this issue in a efficient manner is the main goal of this article, which proposes an intelligent orchestration mechanism that relies on holistic and updated information about the state of the network for preventing malicious packets from reaching their target.

III. A MAS for detecting, mitigating and blocking DDoS attacks: Arquitena

Arquitena is our proposal for dealing with DDoS attacks using a MAS approach. The underlying idea is to benefit from cooperation and coordination among Arquitena agents to monitor an ISP network, detect DDoS attacks, and mitigate such attacks by blocking the attackers. In order to do that, Arquitena adopts two basic types of agents, the external agent and the internal agent, whose instances run in a virtual environment (called Delos) that simulates the underlying network topology. In this section we describe the Arquitena agents, the way they interact and the environment where such interactions occur.

A. Arquitena agents

As already said, an Arquitena agent can be of one of two types: an external agent or an internal agent. The choice for their names is related to their position inside the ISP's network. The external agent has knowledge about the network topology and available resources, such as routers, firewalls, servers and load balancers, and is responsible for providing traffic information related to different pieces of network equipment to internal agents. Internal agents are responsible for monitoring network equipment while dealing with traffic information received from external agents. Such monitoring is what provides internal agents with the knowledge required for detecting and mitigating DDoS attacks. In addition, internal agents are further specialized into Minos, Adamantos and Eacos agents (see figure 2).

Minos agents are responsible for monitoring and protecting potential victims, i.e., equipment whose IP addresses are enclosed within the packets sent by attackers. Common examples are servers hosting services such as Web or DNS (Domain Name System), being connected to the wider Internet by the ISP. Adamantos agents are responsible for monitoring and protecting critical equipment at the network's core and also the network's entry points at the edge of the ISP's network. A piece of equipment is considered critical if it is usually responsible for large volumes of traffic (e.g., because it is located at a point of confluence), meaning that (1) it concentrates a lot of information about the network conditions and (2) applying filters to it can considerably reduce the traffic load at the victim. The importance of the network's entry points, on the other hand, comes from the fact that they are expected to become the final and long-living location of the filtering mechanisms applied by Arquitena. Although critical equipment at the network core might also be considered potential victims, Minos agents differ from Adamantos agents because the latter are also responsible for building attack routes and applying filtering mechanisms whenever necessary. Finally, Eacos agent behave similarly to Adamantos agents, but are responsible for less critical equipment at the network's core and, for this reason, are designed to be inactive most of the time. This "sleeping behavior" is part of an strategy to avoid communication and processing overhead while ensuring detection efficiency.

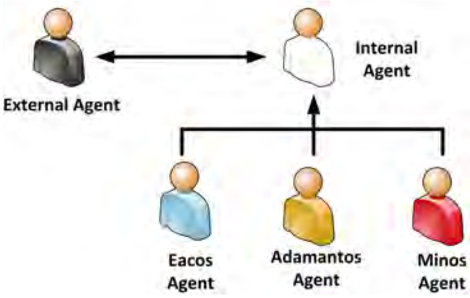


Fig. 2. Arquitena Agents

B. Interaction within Arquitena

Interaction between Arquitena agents provides coordination and learning within the system. External agents communicate with everyone in the system. They send information related to the network traffic to internal agents in order to update their knowledge about it. Therefore, internal agents learn about legitimate and malicious traffic based on information received from external agents, identifying high volumes of traffic addressed to a same target. Also, active internal agents periodically request such information from external agents, characterizing a two-way communication channel.

The language of communication employed is the Knowledge Query and Manipulation Language (KQML) [6], while the Arquitena system itself is developmed using Jason [3] and Java. The interoperability of the system with the (non-Jason) external agent can be resolved by developing an Internal Action with Java and Jason as well. This allows the creation of a wrapper for encapsulating the external agent, enabling it to communicate via KQML.

The learning algorithm adopted in Arquitena is the Random Forest [4], which operates by constructing a large number of decision trees and provides outputs generated by individual trees. This algorithm does not require data normalization, allows lots of data to be analyzed with little computational resources and enable model validation through statistical tests. Its application in the system should allow agents them to learn the average rate of packets (FPav) that nornally pass through the monitored equipment. In this manner, the Arquitena system can be considered truly intelligent, since its agents contantly learn about the traffic characteristics and this knowledge becomes mote accurate with time. The decision and learning models applied are consistent with the characteristics of each agent, which stores data and predict suitable actions.

Interaction between internal agents occur in two ways: (i) Minos agents interact with every internal agent, since it must inform the IP address of the equipment under its responsibility; and (ii) other internal agents interact with their neighbors. Since each internal agent is responsible for a specific equipment, internal agents A and B are said to be neighbors if there is a direct network link connecting the two pieces of equipment monitored by them. Such interaction provide neighbors with information related to their location and, more importantly, their current traffic flow. In addition to such information, Eacos agents must be activated whenever one of their neighbors perceive any possibility of malicious traffic flow coming from its neighborhood.

Coordination is also achieved based in the information exchanged during interaction among agents. In special, each internal agent periodically queries the external agents for traffic statistics corresponding the equipment being monitored by it. This allows internal agents to update the value of a threshold for each potential victim, accommodating natural changes on network traffic, and also to detect when the threshold is exceeded. If that happens often enough, this may indicate a DDoS attack and, thus, internal agents activates all Eacos agents in their neighborhood. Eacos agents activated in this manner may fall back to the inactive state if there is not enough traffic passing by the equipment monitored by it, or may activate other neighboring agents in a chain reaction. Eventually, Adamantos agents near attackers are triggered at the network's edge, meaning that the attack route is complete and, hence, the DDoS attack is fully detected.

Simultaneously to the above detection process, the agents also cooperatively follow specific plans aiming to prevent an excessive amount of traffic from reaching the victim during the attack, thus avoiding the disruption of that equipment's service. This is accomplished by having the agents configure (temporary) filtering mechanisms in the monitored equipments, blocking network traffic directed to the victim, something that could not be easily done manually. As these filters move toward the edge of the network where the attackers actually are, the legitimate traffic becomes less and less affected. As a final remark, we notice that the system must be manually initialized with some basic information that cannot be learned otherwise. This includes the maximum flow of packets (FPmax) that each device can support, a security parameter that depends on the specific equipment being protected. Another example is the position of the agents, the connections between them and the target equipment each of them monitor, which depend on the actual layout of the network where the Arquitena system is deployed and which are the nodes that need protection. Since this layout is not expected to change often, in principle Arquitena does not need to include mechanisms to deal with the dynamic entrance and exit of agents. Therefore, the agents are placed in fixed and permanent positions. Nonetheless, Arquitena shows some dynamism in the sense that any active internal agent may, depending on the state of the environment,

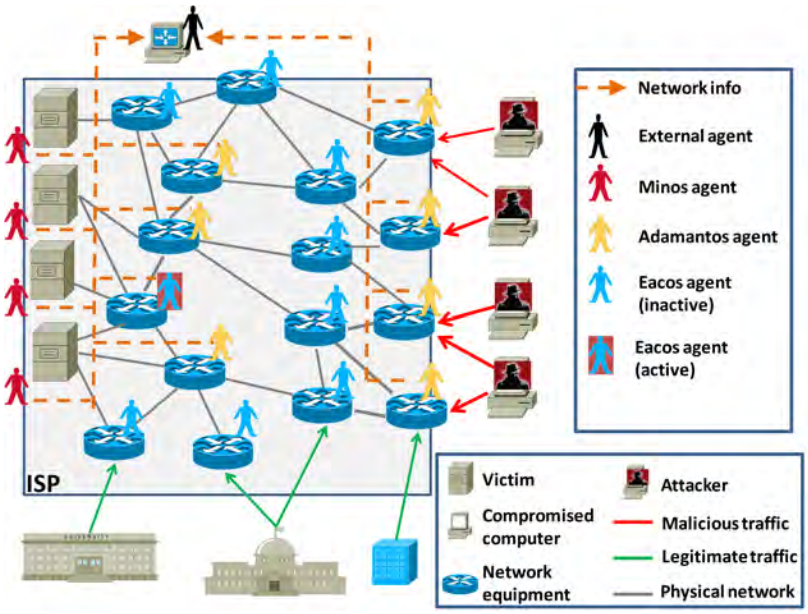


Fig. 3. Arquitena system and agents

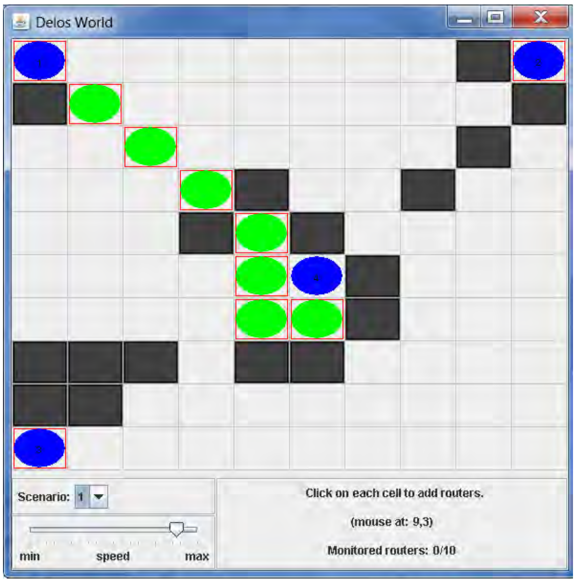
enable Eacos agents during the system's mitigation process.

C. The Delos environment

Although Arquitena could in principle be implemented in a completely distributed manner, the platform being used for implementing the system is a centralized hardware (disk, CPU and memory) where all agents are executed. This facilitates the communication between agents and, at the same time, does not require modifications on the existing network equipment being monitored.

The network itself is simulated using a simulation environment, called Delos, which handles the agents' requests and message exchanges, as well as the dynamic changes in their surroundings. Delos implements a grid that maps the ISP topology and, thus, must be configured using information from the position of the network elements and their interconnections. Aiming to facilitate interpretation by humans, Delos provides a visual interface with a color system that identifies different network elements and their current status (see Figure 4): green for inactive agents (equipment not being monitored); blue for active agents monitoring a piece of equipment under normal traffic; red for active agents monitoring equipment with anomalous traffic; gray for active agents monitoring equipment that have an active access control list applied to them due to anomalous traffic.

In its current state, Delos is still a quite simple prototype that allows for communication between agents and a few types of network equipment. Its final implementation, however, should give support to the following components, some of which are depicted in Figure 4: (i) routers (oval form); (ii) firewalls (hexagon form); (iii) servers (diamond form); (iv) load balancers (triangle form); (v) Minos, Eacos and Adamantos agents associated with these objects. Network elements that have no connections with their neighbors in the grid are separated by black squares (empty cells in a grid).


Delos World

Scenario: 1

Click on each cell to add routers.

(mouse at: 9,3)

Monitored routers: 0/10

Fig. 4. Simulation Environment: Delos

Like Arquitena, Delos is developed using Jason [3] and Java.

IV. Analysis

The main functional requirements of the Arquitena system are: (1) the ability to accurately detect a situation of DDoS attack; (2) mitigate the attack after its detection, blocking enough malicious traffic flow along the network so that the amount of malicious packets reaching the target are not enough to compromise its service; and (3) preserve as much as possible the legitimate network traffic, whether or not it is directed to the victim of the DDoS attack.

These requirements are expected to be fulfilled as a direct consequence of the sparse distribution of agents monitoring the environment in a ubiquitous manner, covering diverse choices of agents and victims. As the agents learn the traffic distribution and normal thresholds of the protected network, they can notice abnormal behavior caused by different types of DDoS attacks. When that happens, they can act accordingly, building attack routes and placing barriers for the malicious traffic on the relevant network elements. Furthermore, the fact that the system allows some agents to remain inactive enables the network administrator to balance the solution's usage of computational resources, such as CPU time and memory.

It is important to notice that, as any general purpose solution that independs on the protocols employed by the attacker, Arquitena is unable to mitigate attacks that explore specific vulnerabilities in the target system. Nonetheless, if desired, the system can used in combination with solutions for preventing speficic attacks on the service-side, such as Intrusion Detection/Prevention Systems (IDS/IPS). The advantage of using Arquitena in this scenario is that it should prevent most of the malicious packets from reaching their intended victims, reducing the overall load of the companion IDS/IPS.

The experimental evaluation of Arquitena's effectiveness is still an ongoing work. Among the metrics to be considered, we can cite: the number of packets that arrive at the victim per time interval, which indicates how well the system blocks malicious traffic and allows legitimate packets through; the time required for the system to detect an attack situation; the time required for the system to block the attack, inserting the appropriate filtering mechanisms at the network's edge; the rate of false-positives, i.e., situations in which a (large) growth of legitimate traffic is misinterpreted as an attack; and the rate of false-negatives, in which an actual attack is not detected. The benchmark scenario beign built is composed by emulated routers and injector package software. In this manner, we will be able to create controlled DDoS attacks and assess the efficiency of the Arquitena system in detecting and mitigating them.

V. Related Work

The application of MAS for the detection and mitigation of DDoS attacks is not something new in the literature. One example is the work proposed in [2], which employs multiple specific agents having a single function: to detect DDoS attacks by means of pattern recognition mechanisms. The result is a solution with a high assertiveness rate in the detection of DDoS attacks. The cost to be paid, however, is high communication overhead depending on the amount of agents involved, since those agents communicate constantly to outline the attack flow. In comparison, Arquitena tries to keep active only the essential agents for the detection of a DDoS attack, dynamically activating relevant agents when (and only when) necessary for further mitigation. This reduces the overall system's burden in terms of processing and communication. A related solution also appears in [7], which describes a MAS-based learning system focused on intrusion detection, covering, among others threats, DDoS attacks. The learning process relies on different sources of raw data, including network traffic, operating system and application activities, which are then combined. One shortcoming of this approach is that the origin of this data is both distributed and heterogeneous, which may lead to inconsistencies in different parts of the system. Arquitena tries to overcome this issue by making an external agent responsible for all data collection, providing a more homogeneous picture of the real network traffic as required by each different agent. Another important limitation of [7] when compared to Arquitena is that the former only takes advantage of the multiple agents for identifying DDoS attacks, while no mitigation mechanism is provided. The combination of detection and mitigation of DDoS attacks using MAS is explored by Juneja et al. [8] and by Singh et al. [13]. Both frameworks use the multiple agents dispersed all over the network to verify a packet's authenticity via source IP validation, a process by which the system determines whether its source corresponds to a real machine. Packets detected as forged are then blocked. However, the very fact that they rely on source IP validation for discerning legitimate traffic is probably the main limitation of this strategy, since this method is ineffective against attacks based on botnets, which are quite common nowadays [9]. The system proposed in [13] is also limited to mitigate attacks from one type of transport protocol (namely, UDP) and provides no mechanism for tracing the route taken by forged packets. Even though [8] does allow attack routes to be traced, potentially admitting more effective mitigation, its own authors admit that the number of agents required to get optimal results is something that needs further investigation [13]. Arquitena, on the other hand, was designed to be generic enough to detect attacks based on any transport-layer protocol, relying on network patterns rather than IP addresses for determining the attack routes.

VI. Conclusions

DDoS attacks have become a major threat to ISPs in the last few decades [1].

Aiming to tackle this issue, we described Arquitena, a system for detection and mitigation of DDoS attacks based on large amounts of traffic. The system is composed of multiple collaborative agents that, without manual intervention of the ISP network's administrators, detect the attack route and gradually deploys barriers to the malicious packets, pushing the defense perimeter toward the network's edge (nearer to the attackers). By covering all regions of the ISP network and constantly learning the traffic patterns in that network, Arquitena can be used as a powerful tool against DDoS attacks, ensuring the availability of the ISP network's resources to real users.

Arquitena is currently under development, and its effectiveness in preventing DDoS attacks will be evaluated both individually and in association with traditional network security solutions (e.g., Intrusion Detection/Prevention Systems). Our goal is to feed the agents with real traffic traces collected from an ISP and then simulate different attack scenarios and assess the system's behavior in each of them.

Acknowledgment

This work was in part supported by the São Paulo Research Foundation (FAPESP) under grants 2010/20620-5 and 2011/21592-8.

References

[1] Arbor. Worldwide infrastructure security report. http://www.arbornetworks.com/research/infrastructure-security-report, 2012.

[2] Z. Baig and K. Salah. Multi-agent pattern recognition mechanism for detecting distributed denial of service attacks. *IET Information Security*, 4(4):333–343, 2009.

[3] R. Bordini, J. Hubner, and M. Wooldridge. *Programming multi-agent systems in AgentSpeak using Jason*. Wiley, 1th edition, 2007.

[4] L. Breiman. Random forests. In *Machine Learning Journal, Vol. 45, Issue 1*, pages 5–32. Kluwer Academic Publishers, 2001.

[5] S. Chen and Q. Song. Perimeter-based defense against high bandwidth ddos attacks. *IEEE Trans. Parallel Distrib. Syst.*, 6(16):526–537, 2005.

[6] Tim Finin, Jay Weber, Gio Wiederhold, Mike Genesereth, Rich Fritzson, Don McKay, Stu Shapiro, Jim McGuire, Richard Pelavin, and Chris Beck. Specification of the kqml agent-communication language, 1994.

[7] V. Gorodetski, I. Kotenko, and O. Karsaev. Multi-agent technologies for computer network security: Attack simulation, intrusion detection and intrusion detection learning. *Int. J. of Computer Science Systems Science & Engineering*, 2003.

[8] D. Juneja, R. Chawla, and A. Singh. An agent-based framework to counterattack ddos attacks. *Int. J. of Wireless Networks and Communications*, 1(2):193–200, 2009.

[9] Z. Mao, Vyas Sekar, Oliver Spatscheck, Jacobus van der Merwe, and Rangarajan Vasudevan. Analyzing large ddos attacks using multiple data sources. In *Proc. of the 2006 SIGCOMM workshop on Large-scale attack defense*, LSAD '06, pages 161–168, NY, USA, 2006. ACM.

[10] J. Mirkovic, S. Dietrich, D. Dittrich, and P. Reiher. *Internet Denial of Service - Attack and defense mechanisms*. Prentice Hall, 1th edition, 2004.

[11] Y. Ohsita, S. Ata, and M. Murata. Detecting distributed denial-of-service attacks by analyzing TCP SYN packets statistically. In *IEEE GLOBECOM'04*, volume 4, pages 2043–2049, 2004.

[12] Riorey. Riorey taxonomy of ddosattacks. http://www.riorey.com/x-resources/2011/RioRey_Taxonomy_DDoS_Attacks_2.2_2011.pdf, 2011.

[13] A. Singh and D. Juneja. Agent based preventive measure for UDP flood attack in DDoS attacks. *Int. J. of Engineering Science and Technology*, 2:3405–3411, 2010.

[14] W. Stallings and L. Brown. *Computer Security: Principles and Practice (2nd ed.)*. Pearson, 2011.