A SYNCHRONIZED DISTRIBUTED DENIAL OF SERVICE PREVENTION SYSTEM

Metasebia Kassa[1] and Mulugata Libsie[2]

[1] ZTE University Ethiopian Branch, Ethiopia
metasebia.kassa@zte.com.cn; met.2000@yahoo.com
[2] Colleges of Natural Sciences,
Department of Computer Science, Addis Ababa University, Ethiopia
mulugeta.libsie@aau.edu.et

ABSTRACT

DDoS attack is a distributed source but coordinated Internet security threat that attackers either degrade or disrupt a shared service to legitimate users. It uses various methods to inflict damages on limited resources. It can be broadly classified as: flood and semantic (logic) attacks. DDoS attacking mechanisms vary from time to time and simple but powerful attacking tools are freely available on the Internet. There have been many trials on defending victims from DDoS attacks. However, many of the previous attack prevention systems lack effective handling of various attacking mechanisms and protecting legitimate users from collateral damages during detection and protection.

In this paper, we proposed a distributed but synchronized DDoS defense architecture by using multiple agents, which are autonomous systems that perform their assigned mission in other networks on behalf of the victim. The major assignments of defense agents are IP spoofing verification, high traffic rate limitation, anomaly packet detection, and attack source detection. These tasks are distributed through four agents that are deployed on different domain networks. The proposed solution was tested through simulation with sample attack scenarios on the model Internet topology. The experiments showed encouraging results. A more comprehensive attack protection and legitimate users prevention from collateral damages makes this system more effective than other previous works.

KEYWORDS

Denial of Service, Distributed Attack, Synchronized defense, Attack Prevention, Agent

1. INTRODUCTION

A Denial of Service (DoS) is any attempt by an attacker to prevent legitimate users from using the desired resources. It includes attempts to [19]: flood a network by overflowed packets thereby preventing legitimate network traffic, disrupt connections between two machines thereby preventing accessing a service, prevent a particular individual from accessing a service by denying his/her access rights, and disrupt service to a specific system.

Distributed Denial of Service (DDoS) attack is a large-scale, coordinated attack on the availability of services of a victim system or network resource, launched indirectly through many compromised computers on the Internet [17].

The number of DDoS attacks has been alarmingly increasing for the last few years. These attacks are carried out by organized criminals targeting financial institutions, e-commerce, gambling

Sundarapandian et al. (Eds): ICAITA, SAI, SEAS, CDKP, CMCA, CS & IT 08,
pp. 09–23, 2012. © CS & IT-CSCP 2012 DOI : 10.5121/csit.2012.2502

sites, etc. For instance, large organizations like Microsoft, Amazon, eBay, CNN, and Yahoo have experienced DDoS attacks in the year 2000 [2, 14, 20].

The losses caused by DDoS attacks are tremendous, especially to E-Commerce sites. Unacceptable download times often caused by DDoS attacks are estimated to have caused losses of up to $4.35 billion in the US. E-Commerce sales and worldwide businesses experienced about 3.3% of unplanned downtime in 1999, translating to $1.6 trillion in lost revenue [5]. According to a survey done by FBI [21] collected from 251 organizations, DoS attacks were the second most expensive computer crime, with a cost of more than 65 million dollars in the year 2003. In another survey, DoS attacks had caused losses for almost 3 billion dollars and have been in the top five attacks indexed by economic loses of the previous 3 years [15, 18]. According to a study by Arbor Networks [22], in 2007, the largest DDoS bandwidth attack was recorded as 40 Gigabits per second on Internet Service Providers (ISPs) but this size nearly doubled in 2008 from the previous year.

DDoS attacks range from small to large scale versions launched from thousands of bots, affecting not only the target victim, but also the infrastructure of the service provider [3]. In October 2002, an attacker flooded the root Domain Name Service (DNS) servers with traffic in an effort to deny the Internet of the DNS name lookup service which would have paralyzed the majority of Internet applications. Only five out of thirteen root servers were able to withstand the attack. Thus, accessing Internet services was degraded in the globe till the problem was solved [15, 22]. As some studies showed [14], regardless of the diligence, effort, and resources spent securing against intrusion, Internet connected systems face a consistent and real threat from DDoS attacks because of different reasons.

2. DENIAL OF SERVICE ATTACKS

DoS attacks cause either disruption or degradation on victims' shared resources, preventing legitimate users from using their access right on those resources. DoS attacks may target a specific component of a computer, the entire computer system, certain networking infrastructure, or even the entire Internet infrastructure. Attacks can be either by exploiting the natural weakness of a system, which is known as Logic attack or overloading the victim with high volume of traffic, which is called Flood attack [11].

A distributed form of DoS attack, called DDoS attack, is generated by many compromised machines to coordinately attack a victim [19]. DDoS attacks consist of at least four core elements: attack source, control master(s), agents, and victim. Attack source or control masters may use valid or spoofed IP addresses during the attack based on the attacking strategies they follow. By spoofing, the actual attacker can hide its identity and reduce the chance of being claimed by the victim.

DDoS attacks can be performed by using powerful attacking tools such as Trin00, TFN, TFN2K, Stacheldracht, and Shaft, which are easily available on the Internet. Ease of availability of various powerful DDoS attacking tools and variant natures of DDoS attacking strategies makes DDoS attack defense a challenging problem [4, 10].

3. RELATED WORK

Based on their deployment location, we present related works as: Source Network, Intermediate Network, Victim Network, or Cooperative Defense solutions.

Mirkovi´c et al. in [6] proposed a source side DDoS defense system which is a self-regulating reverse-feedback system. It consists of observation and throttling components that can be part of the source router itself, or can belong to a separate unit that interacts with the source router to obtain traffic statistics and install rate-limiting rules. The observation component monitors all packets passing through the source router and gathers statistics on two-way communication between the police address set and the rest of the Internet. This monitoring can be performed by sniffing the traffic at the source router interfaces. Periodically, statistics are compared to models of normal traffic and results are passed to the throttling component which adjusts and transfers the new rate limit rules into the source router. The imposed rate limits modify associated traffic flows and thus affect future observations, closing the feedback loop.

Shu and Dasgupta [7] proposed a router based DDoS prevention system, called denying Denial-of-Service Attacks. Accordingly, the routers are modified to provide encryption, digital signatures, and authentication, enabling the tracing of a packet back to its origin and thus stopping further traffic at the closest intelligent router point. Every group of collaborating routers is called a "hardened network". The hardened routers should be implemented at the border and access point of an Autonomous System. When arriving at the first hardened router, the packet's payload is encrypted together with one byte of its IP address and the last hardened router before the host will decrypt it. This way the packet can be traced back to the first hardened router, and an attack can be stopped at that point.

Ioannidis and Bellovin [9] proposed a network-based solution, Pushback, which tries to solve the problem of DDoS attacks from within the network using the congestion level between different routers. When a link's congestion level reaches a certain threshold, the sending router starts dropping packets and tries to identify illegitimate traffic by counting the number of times packets are dropped for a certain destination IP address, since the attacker constantly changes the source IP address. The router then sends a pushback message to the routers connecting it to other congested links, asking them to limit the traffic arriving to this destination.

Jin et al. [8] proposed Hop-count filtering (HCF), which is a victim based solution relying on the fact that the number of hops between source and destination is indirectly indicated by the time-to-live (TTL) field in an IP packet. Linking the source IP with the statistical number of hops to reach the destination is used as a reference to assess the authenticity of the claimed IP source. A hop-count based filtering scheme that detects and discards spoofed IP packets to conserve system resources is used. Their proposed scheme inspects the hop-count of each incoming packet to validate the legitimacy of the packet. Using moderate amount of storage, HCF constructs an accurate IP to Hop Count mapping table via IP address aggregation and hop-count clustering. A pollution-proof mechanism initializes and updates entries in the mapping table. By default, HCF stays in alert state, monitoring abnormal IP to Hop Count mapping behaviors without discarding any packet. Once spoofed DDoS traffic is detected, HCF switches to action state and discards most of the spoofed packets.

Because of the nature of DDoS attacks, it is not sufficient to prevent DDoS attacks with single point defense strategy [12]. Thus, current works focus on cooperative defense strategies. The following works show how distributed and coordinated defense techniques are intended to work for DDoS attacks prevention.

Cooperative Defense against DDoS Attacks which was proposed by Zhang and Parashar [13] is a global DDoS defense infrastructure built as an overlay network on top of the Internet. The scheme consists of two stages. In the first stage, each defense node detects traffic anomalies locally using a variety of existing intrusion detection system tools. According to its local defense

policy, each local defense node sets a rate limit to the traffic identified as attack traffic. In the second stage, gossip based communication mechanism is used to share information among the defense nodes that is expected to enhance the accuracy of the defense mechanism. A gossip-based scheme is used to get global information about DDoS attacks by information sharing. The proposed system continuously monitors the network. When an attack begins, individual defense nodes drop attack traffic identified according to the local information and mitigate load to the target victim. However, as local detection has a high false alarm rate, legitimate traffic will also be dropped. By correlating the attack information of each individual node, the scheme can get more information about the network attack and thus can defend against DDoS attacks more effectively.

In [1, 23] another collaborative DDoS defence system is proposed by Oikonomou et al. and Xuan et al. in which routers act as gateways, detecting DDoS attacks locally and identifying and dropping packets from misbehaving flows. Gateways are installed and communicate only within the source and the victim domains, thus providing cooperative defense of a limited scope.
Similarly, Papadopoulos et al. [24] proposed a multicast group of defense nodes which are deployed at source and victim networks. Each defense node can autonomously detect the attack and issue an attack alert to the group. Sources involved in the attack cooperate with the victim to suppress it. Since intermediate networks do not participate in the defense, the solutions proposed in [23] and [24] cannot control attack traffic from networks that do not deploy the proposed defense.

From the review we derived that, there should be at least these three critical defense functionalities that any DDoS defense approach need to achieve: accurate attack detection, rate limiting of traffic to free critical resources, and traffic differentiation to separate the legitimate from the attack traffic to minimize collateral damage. Thus any defense effectiveness should be evaluated based on these functionalities.

Besides, based on the deployment location, the defense mechanisms will have the following advantages and limitations.

Defense mechanisms deployed near the victim can identify DDoS attacks easily and accurately because it can view the aggregate attack traffic, but tracing back the source of the attack is hard and the amount of traffic to analyze is large. Moreover, large amount of flood may not give a chance to detect and react to the attack.

Intermediate network solutions, deployed at the core network infrastructure, have the advantage to rate limit large floods that would overwhelm the victim's access links as early as possible before aggregating and congesting the victim. However, accurate detection of illegitimate traffic and the willingness of service providers for deployment are somehow challenging.

A source-end solution, deployed at the edges of the Internet, has the advantage of easily tracing back to attack sources since the amount of traffic and address diversity to analyze is minimal. However, it suffers from the deployment issue as it is a pure source-end solution and it doesn't handle legitimate congestion problems.

4. PROPOSED SOLUTION

To counteract the diverse nature of DDoS attacks and protecting legitimate traffic from collateral damage, it is not sufficient to use simple message exchange among nodes about the attacks and reacting with a single point of defense location [13]. There must be a distributed defense system for effective handling of the problem. Thus, we designed a distributed and coordinated multi-

agent DDoS attack with the central coordinator defense that realizes the strengths of various deployment locations. These agents will monitor the network and will trace the incoming and outgoing packets which are addressed to a specified destination and will have the ability to read and take appropriate actions on them based on a given mission within their working boundary.
Here, agents are autonomous systems that work on behalf of targeted victims within the given and others' domain networks. They are algorithm implementers and information exchangers. The algorithms they implement include: IP spoofing verifier, packet filter (classification and packet marking), rate limiter, anomaly detector, and attack source tracer.

Figure 1 presents a complete flowchart of the packet screening process of the proposed DDoS attack defense mechanism.

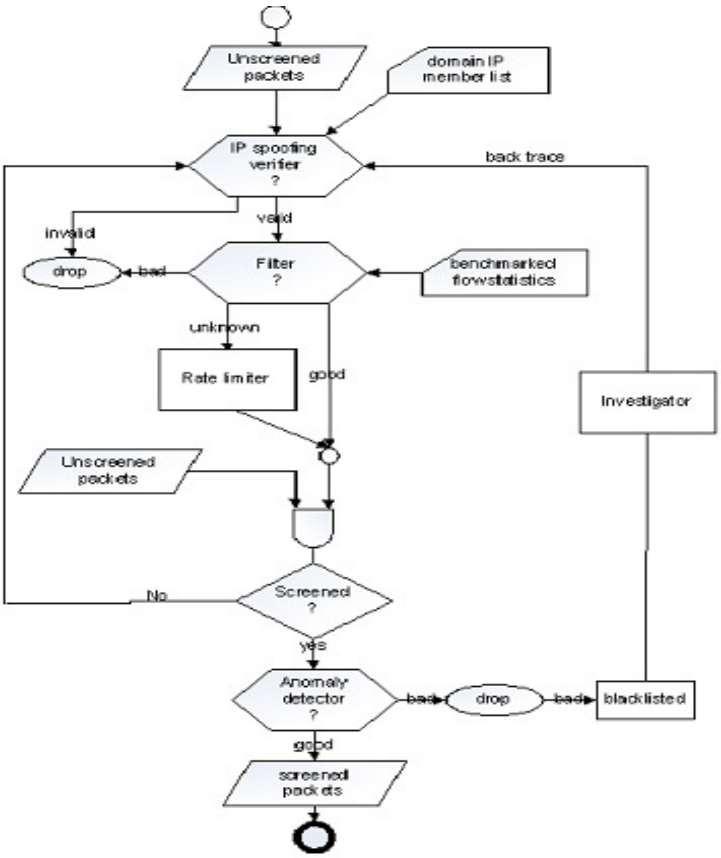

Figure 1: Flow of packet screening processes by an agent

As it is shown in Figure 1, the entire screening process can be performed by an independent agent. However, to minimize the work load and to maximize the effectiveness of the problem solving capacity of an agent, the processes among various agents that will be deployed at different locations are distributed. Each component is described in the sequel.

IP spoofing verifier: checks the validity of the source IP address of packets before they come-out through source stub borders or come-in to destination stub borders. The verifier at stub borders ensures that a packet leaving its domain has a source address from inside the domain, and a packet entering has one from outside. It checks packets' IP address field values for source address spoofing. It also identifies attack source identity during an actual attack source investigation process. A source side verifier evaluates each outgoing packet's source IP address

based on pre-established active domain members address list, which may be stored in a hash table or a database.

However, as figure 2 shows destination side verifier evaluates source address spoofing by using other methods such as hop-count filter (HCF). HCF is a light weight victim side IP spoofing validation algorithm. It is computed based on the packet time-to-live (TTL) value. TTL value is primarily used for determining a life time of a packet staying in transit. TTL value will be reduced by one when that packet passes through a hop. Thus, indirectly, this value can indicate how much route a packet travels to reach its destiny by deducing initial TTL value from final TTL value i.e., hop-count (HC = initial TTL – final TTL). Ideally, packets that are generated from one source will have same hop-count value unless they are spoofed.

Packet filter: focuses on classifying traffic type as TCP, ICMP, and UDP flows, evaluates their behaviour, and marking them as “good”, ”unknown”, or “bad” according to their behaviour. Then, forwards “unknown” packets to the next component for rate limiting and “good” packets for service priority. However, “bad” packets will be dropped and recorded in a blacklist table. Figure 3 shows the process of flow classification and counting independent flow rates of incoming and outgoing traffic.

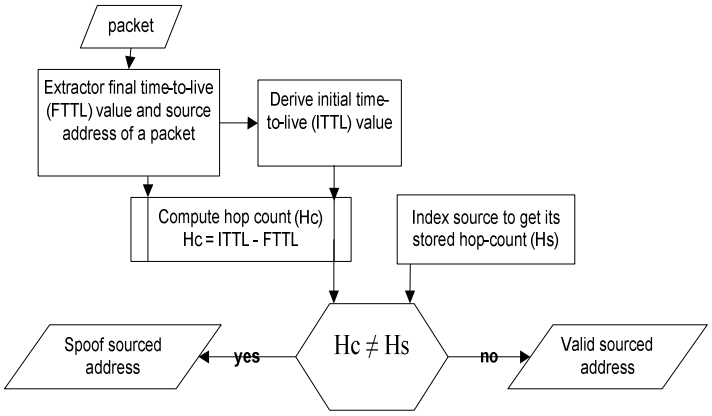

Figure 2: Source IP spoof inspection algorithm

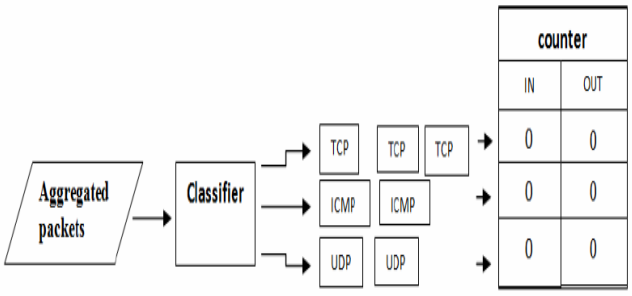

Figure 3: Packet classifier and packets counter processes

Traffic behaviour is evaluated based on to-fro (out-in) ratio for TCP packets, request-reply ratio for ICMP packets, and flow rate (number of packets per time interval) for UDP packets, and packet size (min-max range) for ICMP ping packet communication.

To detect TCP-based attacks, we adopt the concept of disproportionate TCP packet rates to and from peers. This idea is first proposed in [16]. TCP is a reliable communication protocol that

guarantees the delivery of packets by exchanging acknowledgment between peers. According to [16], for normal TCP-based communication, the number of packets sent to and received from a host should be constant and the ratio is suggested as not more than 3. Thus, non-proportional communication is good indication of DDOS attacks. During attack, there is large number of packets sent but a few or no reply from the other side will be received by the sources since the attacker usually forges its source address to redirect the reply of the receiver, or repeatedly resets the connection to make a destination machine busy with unnecessary processes. This assumption also works for ICMP request/reply communication.

UDP and ICMP packets are mainly used by bandwidth consumption attacks. As these traffic types generally utilize small amounts of bandwidth, a sudden change in the transferred ICMP or UDP bytes per second or individual packet's payload size are good indications of attacks.

During attack detection, the current sampled flow ratio or flow rate value will be compared with a benchmarked threshold value in a given time interval. Then, the flow is classified as "bad" (attack) flow if its packet ratio or rate is above the threshold; "unknown" if its rate or ratio is suspicious range; otherwise, it is considered a "good" flow. Figure 4 shows the process of evaluating flow behaviour.

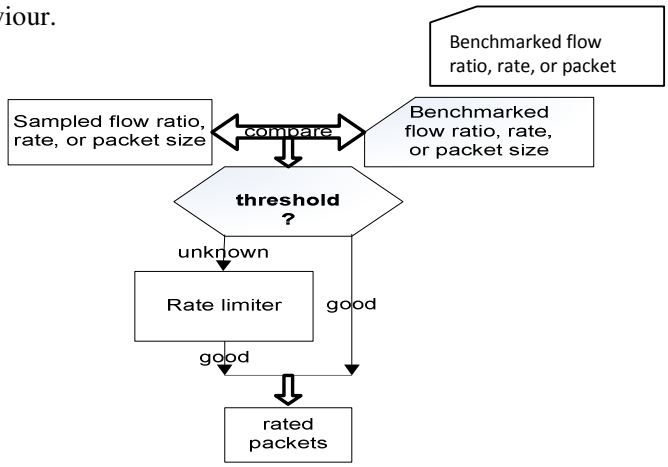

Figure 4: Packet filtering and rate limitation process

After classifying flows, there will be a process of packet marking according to their behaviour with a unique signature that can be interpreted by subsequent components. This signature can be behaviour in the option field of the packet header.

"Unknown" flow is determined by an agent based on the local knowledge of that agent about that flow. So, "unknown" flow can either be "bad" or "good" when it is detected by other agents. In addition, an agent can attach the degree of confidence value about "unknown" flow based on its information. This confidence value is further used as an input by the rate limiter to assign the degree of rate limit on that particular flow. The degree of confidence about a flow can range between 0 and 1, where 0 means certainly bad flow and 1 is certainly good flow. As the fractional value increases, the certainty of a flow being good will increase. As a result, a good flow could get its maximum bandwidth while a bad flow could be penalized by the rate limiter based on the confidence level. Therefore, unknown flow is attached with additional information such as flow identity (fid), degree of confidence I, and its destination (d) before it is forwarded to the rate limiter.

Rate limiter: focuses on protecting victims from over flooded packets during communication. It works based on the aggressiveness of flows with respect to the serving capacity of a given node (host or router) in a given time frame. Traffic will be rate-limited if its behaviour is assumed as congestion causing to the next destiny. Traffic behaviour is evaluated based on flow rate (packets

per second) for UDP packets, to-fro ratio for TCP packets, request-replay ratio for ICMP packets, packet size (min-max range) for ICMP ping packet communication.
The rate limiter receives an "unknown" flow with unique flow id (fid), in-flow rate (rate-in), confidence value I, and forwarded destination address (d). Then it sends limited rate flow by computing rate-out according to the confidence value.

Anomaly detector: focuses on protecting victims from semantic (logic) attacks before causing confusion and freezing normal service. In semantic attack, the attacker generates invalid packet content that causes an application to freeze or crash. Thus, the anomaly detector works based on the predefined signature for known attack types and learning through time from the requests for new types of attacks. The signature can be set according to local network security policy. These signatures can be stored in a database, similar to virus definition in an antivirus engine.

Attack source tracer: focuses on identifying the origin and stopping further attack originating from that source. It includes investigating detailed attacking strategies and collecting and analyzing post attack reports from remote agents.

Generally, these algorithms are used as a means of knowledge of agents to perform the entire packet screening process during DDoS attacks.

To realize the above algorithms, we identified four defense agents that can accomplish their assigned mission based on a given situation by implementing one or more of the previously discussed algorithms. These agents include Source Agent (S-Agent), Intermediate Agent (I-Agent), Destination Agent (D-Agent), and Master Agent (Magent). Though their structure and components look similar, their functionality defers and is determined by their location and the information they have about the network traffic.

As it is shown in Figure 5, each agent has four core components that communicate with each other to accomplish its assigned tasks. These components include sensor, sampler, detector, and filter. However, some particular agents like M-Agent may have other components such as investigator. An agent is designed to detect and react for various DDoS attacks by changing its assigned mission according to its working location and a given situation. The components of an agent are described in the sequel.

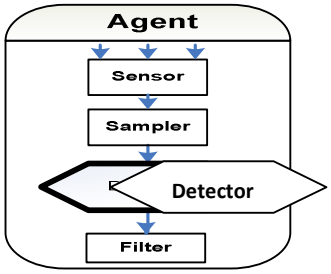

Figure 5: Structure and components of a basic defense agent

Sensor: is an initial information processing component that gathers statistical data by sniffing the traffic while it passes through a given node interface. Moreover, it calculates the amount of traffic (bits per second, number of packets per second, ratio of two-way communication flow in a certain period, etc.), determines the addresses of hosts that make the largest traffic, and

determines the validity of a source address by implementing different IP spoofing verifier methods based on the location it is deployed.
Sampler: is a secondary information processing component that works in two modes: learning and normal mode. In learning mode, it processes the network packets and constructs a benchmark of normal flows of a given network. Then, in normal mode, it analyses and compares current traffic with the benchmarked normal traffic. It picks the addresses of hosts that do not correspond to the benchmark and sends them to the detector.

Detector: its general goal is to make a decision about the beginning of attack on the basis of sensor and sampler results. A detector sends the list of attack addresses received from the sensor and the sampler to the filter.

Filter: categorizes flows as good, unknown, or bad on the basis of detector results. It takes appropriate actions such as dropping accurately detected bad packets, marking unknown (suspicious) packets for rate limitation or further detection, and prioritizing good packets.

Investigator: is a special component of an M-Agent, which is not shown in the basic agent structure of Figure 5. Its general goal is to trace the attack source and attack slaves. After receiving a message from the detector, it examines the obtained IP addresses for the presence of attack agents.

We divided the overall DDoS defense requirements and assigned them to each of our specialized agents as follows:

S-Agent: is a source end autonomous system that has the ability to read and modify each packet which passes through border routers of source stub network.

I-Agent: is an intermediate (core) node traffic monitoring system which has the ability to measure the impact of aggregate traffic which are generated by different source stubs towards a specified destination.

D-Agent: is a destination stub system which monitors and reacts on flows that are destined to victim networks.

M-Agent: is a master agent that coordinates the overall defense activities and manages investigation of actual attack source during back trace. It is the place where logical (semantic) attacks will be detected by tracing each packet's content & analyzing packet's intention according to local services. This agent may perform the tasks of an S-Agent and an I-Agent if packets are originated from the local network and not observed by previous agents. It also prioritizes packets based on their behaviour, keeps records of each agent's activities, and communicates with others for further tracing of attack source and attack slaves.

These agents are expected to work as a team with a common goal of defending the victim from various DDoS attacks. Messaging is the means of communication among agents. It can be through one-way communication where only either of two parties sends the message or two-way communication where two parties exchange with each other. Thus, during packet screening, agents communicate with each other about their status using secured messaging. Figure 6 shows the coordination and communication of the four agents during DDoS attack defense.

As it is shown in Figure 6, packets will be verified for not forging their original source address by using S-Agents at the border router of source domain network. Note that, as there are

thousands of source stub domain networks in the Internet, it is not possible to fully deploy S-Agents. Thus, the second layer I-Agents will receive screened packets from sources which have deployed S-Agents or unscreened packets from sources which have not deployed S-Agents, then do their job and forward them to their next destination. At the destination stub domain network, a D-Agent will receive screened packets from an intermediate node which uses I-Agents or unscreened packets from local domain network, then perform their mission and forward them to their final destination.

Finally, an M-Agent at the destination network does the final screening process and delivers normal packets to access the available services.

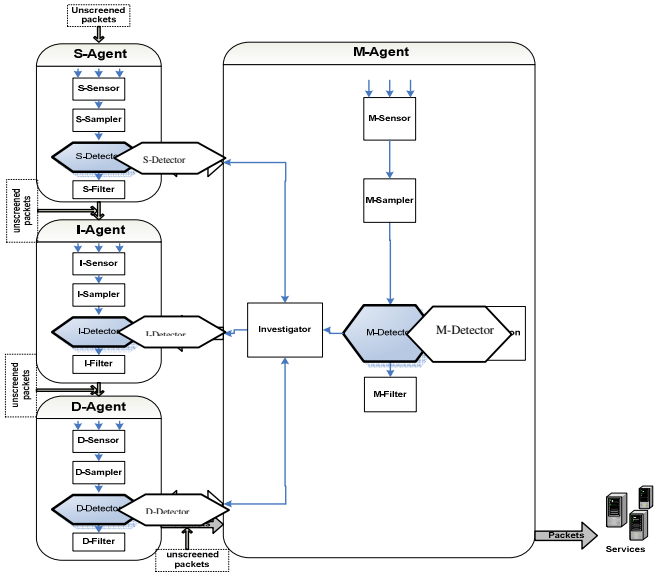

Figure 6: Coordination and communication of Multi-agent DDoS defense system

5. EXPERIMENTATION OF THE PROPOSED SOLUTION

To evaluate the performance of the proposed solution, we conducted a simulation experiment by established sample data sets and attacking scenarios. Our simulation experiment setup and scenarios specify the following four important features that are assumed as influential variables for the defense's effectiveness. These are network topology, legitimate traffic, event scheduling, and attack traffic. We use these variables to establish realizable simulation setup according to actual DDoS attack behaviours and some theoretical background of Internet services.

The goals of this simulation experiments are: illustrating the effectiveness of the proposed solution, evaluating the performance of legitimate users during attacks in the presence and absence of the defense system, and measuring collateral damages of the defense system during attack detection and prevention.

Scenario 1: Evaluating the performance of legitimate traffic (a) in the absence of both attack & defense mechanisms, (b) in the presence of flood attack but in the absence of defense system, and (c) in the presence of both flood attack & defense mechanisms.

In this first set of experiments, we switched 5% of the potential traffic generating nodes (which are around 90 of the 1800) from legitimate to attack users. They are randomly distributed throughout the different source stub domains. These nodes are attached with TCP traffic (agent) by a modified UDP agent such that it sends packets marked as TCP but it does not respond to

congestion. The victim side uses Ns2 FullTcp receiver agents to generate acknowledgments as it would be on real machines.

The bottleneck link is assumed to be 50Mbps, has a delay of 0.5s and a queue of size 1000 packets. All other input links that join have bandwidth of 50 Mbps and a delay of 0.5s. Each attack node sends out 100 Kbps modified UDP traffic with an average 0.22 second interval between packets to the victim. A good user makes request with traffic rates chosen randomly and uniformly in the range [1Kbps, 15Kbps]. If a request arrives at the server successfully, the server will return the result within a certain delay of processing time. The attack starts at 30 seconds after the start of legitimate traffic and ends at 90 seconds. Finally, we measured the packet rate of a selected client at the FTP server and evaluate its performance during the simulation period, which is a total of 120 seconds. Figure 7 shows the result of the simulation experiment. The X axis represents time intervals in seconds and the Y axis represents the rate of packets in bytes received at the server of a selected legitimate client in different situations.

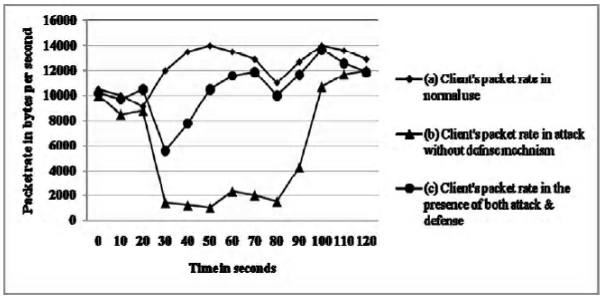

Figure 7: Simulation experiment result of scenario-1

Result Analysis: as it is shown in Figure 7a, more than 14000 bytes successfully arrived at the FTP server in a normal use of the network. However, in the presence of attack and in the absence of defense (Figure 7b), packet rate dramatically dropped down starting from 30 seconds till 90 seconds. The graph shows some progress after 90 seconds since the attack traffic lasts at 90 seconds in the simulation time. Finally, the legitimate packet rate shows significant improvement in the presence of both attacks and defense mechanism (Figure 7c).

Scenario 2: Evaluating the IP spoofing verification performance of full deployment of S-Agents and D-Agents.

In these set of experiments, we performed test runs in the simulated topology by deliberately modifying the original source address of packets, which are generated from 900 (50% of the 1800) source nodes. These nodes are randomly selected from 60 source stub domains. Each of them generates spoofed packet flows with normal rate in the range [1Kbps, 15Kbps] to the victim during the simulation time. We used a random IP spoofing mechanism to assign a forged IP to generated packets. We set queue monitor objects at edge nodes of source and victim nodes to monitor and analyze the packets departure and arrival rate from selected compromised and non-compromised source nodes. We deployed S-agents at edge nodes of source stubs, which have the address list of members' nodes in an array variable. A D-Agent is deployed at the edge node of the victim server, which has the maximum HC threshold (in our case, HC= 3 since it passes only 3 nodes from source to destination). In this experiment, we separately evaluated the performance of S-Agents and D-Agents under full and partial deployment conditions. Figure 8 presents the simulation result of scenario 2.

Result Analysis: Figure 8 shows the effect of full deployment of S-Agents and D-Agents. Figure 8a shows the successful arrival rate of packets at the destination node which originated from selected non-compromised source nodes. It indicates that the IP spoof verification process of

defense agents will not affect the performance of legitimate users. Most of the packets generated from selected legitimate users successfully arrived at the destination node.

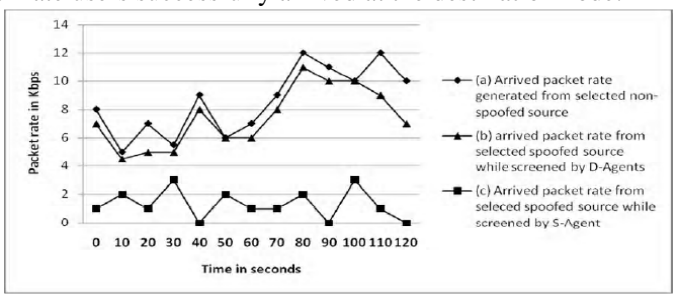

Figure 8: Simulation results scenario-2

Figure 8b shows the successful arrival rate of spoofed packets which are generated from selected compromised nodes and filtered by D-Agents. This result indicates that our D-Agent IP spoof verification mechanism is not detecting the spoofed source packets in the simulation. This is because we set a fixed maximum HC threshold value (i.e., HC=3), and all legitimate and compromised source nodes are located on the same distance, which is 3 nodes away from the target in the simulated network topology. Thus, HC value is 3 for all traffic generating nodes in this simulated network. As a result, D-Agents will not differentiate between valid and spoofed source packets. Note that D-Agents verify forged source packets based on HC difference.

Figure 8c shows arrived packet rates generated by selected compromised nodes and screened by S-Agents. The result shows that our S-Agents successfully screened and dropped spoofed source packets at source stub edge before they forwarded them to the destination. Some of the successfully arrived packets might be the result of random address generation mechanism, since there is a possibility of some of the randomly generated addresses being subset of the domain member's address list which is stored in an array variable.

Scenario 3: Evaluating the performance of defense agents on protecting the victim from overflowing.

To test this scenario, we constructed a simplified transit-stub network topology with 100 traffic generating (source) nodes and each source node is directly connected to a corresponding edge node where the traffic is marked according to the parameters that will be specified. The edge nodes are connected to a single core node, and then through another edge node, to a destination node. Each traffic generating node sends legitimate TCP requests but in different rates. Our aim is evaluating the performance of defense agents on congestion causing traffic. The bandwidth and delay of links are assigned based on the random distribution of the proposed ranges as shown in Table 1. The bottleneck link between a destination edge node and a destination node is assigned 50 Mbps. The average traffic transfer rate is 10 Kbps and has a Pareto distribution with shape parameter 1.25. Traffic sent arrives at the bottleneck link according to a Poisson process. Several sessions can simultaneously be activated by the same source node. Queue is built-up at the bottleneck link which has 100 packets size. We chose three parameters for three priority levels (red, yellow, and green). These parameters are chosen and assigned only for simulation purpose. In actual implementation, these are assumed to be automatically determined based on agents' sampler and sensor data. For each category of flow (red, yellow, and green), the average queue size is monitored (this is done using the standard exponential averaging with parameter wq = 0.01). Packets of a given color start to be dropped when the average number of queued packets exceeds the minimum weight (minw); we choose minw = 15 packets; this drop probability increases linearly with the average queue size until it reaches the maximum value maxw=45, where the drop probability is taken to be maxp=0.5. When this value exceeds, the drop

probability is 1. The differentiation is then done by using the RIO-D approach, in which the rejection probability of each type of color depends on the average number of packets of that type. Thus to have green packets dropped less than yellow, and yellow packets dropped less than red ones, we properly chose three committed information rate (CIR) values which are used to determine the fraction of packets that will be marked green, yellow, or red. The simulation takes 120 seconds. Table 1 shows the simulation result.

Table 1: Performance of defense agents on Congestion causing traffic

Packets Statistics at source edge nodes and destination node				
Code point	Sent by sources	Received by destination	Dropped by link overflow	Dropped by rate limiter
All	119678	69058	259	50361
10	25757	25757	0	0
11	57905	43301	259	14345
12	36016	0	0	36016

Key:

Code point 10 – is for green packets (good rated packets),
Code point 11 - is for yellow (unknown rated) packets, and
Code point 12 – is for red (over rated) packets.

Result Analysis: As it is shown in Table 1, the defense agent filters and totally drops high rated traffic; it rate limits unknown traffic flow, and forwards good rated traffic.

6. CONCLUSION AND FUTURE WORK

DDoS attack is a serious security issue of the Internet community since it is becoming a major cause of economic loss for many countries. It is the malicious act of attackers to disrupt or degrade network resources of legitimate users.

There have been many attempts of protecting DDoS attacks through various ways and still many scholars spend much of their time on proposing comprehensive defense systems. However, many of these attempts lack some of the critical features (functionalities) of DDoS defense such as accurate attack detection, collateral damage prevention, or comprehensive attack protection. In this paper, we proposed a distributed but synchronized DDoS prevention architecture using multiple defense agents which work in others' networks on behalf of the victim. These agents are given various missions to autonomously perform their tasks within their working boundaries and communicate with each other about their actions during defense.

We evaluated the performance of our system by implementing the functionalities of the defense schemes as queue objects in Ns-2 simulation tool. We performed intensive simulation experiments by defining sample attacking scenarios on predefined experimentation setup. We developed four scenarios which allow us to evaluate the effectiveness of the proposed solution with regards to protecting legitimate users from collateral damage, accurate congestion causing packets detection, spoof sourced packets verification functionalities and partial & full defense deployment. The experiments showed encouraging results.

As future work, we will focus on the following issues:

1. In this work, defense agents were deployed on selected fixed place (border nodes of different domain networks). In the future, defense agents shall be mobile and self determinant for efficient working locations to maximize their effectiveness.
2. Securing the defense agents from being compromised by attackers during communication shall be the other consideration since attackers can possibly target our own defense agents for their malicious actions.
3. During implementation and experimentation, due to time limitation, we didn't give much attention on anomaly detection and attack source tracing functionalities; we were focusing on IP spoof validation and high traffic rate limitation. Thus, we will deal in detail on these issues to make this work more comprehensive and effective.
4. Finally, we have to deal with some legal and technical issues and limitations as we send our remote agents to others' network to work on behalf of victim a network.

REFERENCES

[1] G. Oikonomou, J. Mirkovic, P. Reiher, and M. Robinson. 2006. A Framework for Collaborative DDoS Defense. In Proceedings of ACSAC, December 2006.

[2] Cruce Schneider, 2005. Attack Trends: Beyond the Numbers. Counterpane Internet Security, Inc. January-October, 2005

[3] Juniper Network. 2006. Combating Bots and Mitigating DDoS Attacks (Solution brief). Juniper Networks, Inc.

[4] Kevin J. Houle, CERT/CC, George M. Weaver, CERT/CC, In collaboration with: Neil Long, Rob Thomas. 2001. Trends in Denial of Service Attack Technology. Carnegie Mellon University, 2001.

[5] Suhail Mohiuddin, shlomo Hershkop, Rahul Bhan, & Sal Stolfo. 2002. Defending Against large scale Denial of Service attacks. In proceedings of the 2002 IEEE workshop on information Assurance & Security, west point, NY.

[6] J. Mirkovic. 2003. D-WARD: source-end defense against distributed denial-of-service attacks. PhD thesis, UCLA.

[7] Z. Shu and P. Dasgupta. 2003. Denying Denial-of-Service Attacks: A Router Based Solution. In International Conference on Internet Computing, pages 301–307.

[8] Jin, G., Wang, H., and Shin, K. G. 2003. Hop-count filtering: an effective defense against spoofed DDoS traffic. In Proceedings of the 10th ACM conference on Computer and communication security, Washington D.C.

[9] J. Ioannidis and S. M. Bellovin. 2002. Implementing Pushback: Router Based Defense against DDoS Attacks. In ISOC Symposium on Network and Distributed System Security, February 2002.

[10] Distributed Denial of Service Attack Tools. 2002. Internet Security Systems (ISS), Atlanta, GA.

[11] Jelena Mirkovic, Peter Reiher.2004. Taxonomy of DDoS Attack and DDoS Defense Mechanisms. 2004. DOI= http://www.eecis.udel.edu/~sunshine/publications/ccr.pdf

[12] Shibiao Lin, Tzi-cker Chiueh. A Survey on Solutions to Distributed Denial of Service Attacks. DOI= http://www.ecsl.cs.sunysb.edu/tr/TR201.pdf.

[13] Guangsen Zhang and Manish Parashar. 2006. Cooperative Defense against DDoS Attacks. Journal of Research and Practice in Information Technology, Vol. 38, No. 1, February 2006.

[14] Robert Richardson. 2009. Computer Crime and Security Survey.DOI= http://www.gocsi.com/forms/fbi/csi_fbi_survey.jhtml. Visited on January 2009.

[15] D. McGuire and B. Krebs. Attack on internet called largest ever. Oct. 2002. DOI= http://www.washingtonpost.com/wpdyn/articles/A828-2002Oct22.html. Visited on January 14, 2009.

[16] T. M. and Poleto, M. 2001. Multops: a data-structure for bandwidth attack detection. In Proceedings of 10th Usenix Security Symposium, Washington, D.C., USA, 23–28.

[17] WatchGuard Technologies,Inc,. Distributed Denial of Service. WatchGuard Designing peace of Mind, Seattle,WA., February 2000. DOI= www.watchguard com

[18] Abraham Yaar, Adrian Perrig, Dawn Song. Pi: A Path Identification Mechanism to Defend against DoS Attacks. Carnegie Mellon University

[19] Felix Lau, Stuart H. Rubin, Michael H. Smith, Lj ilj ana Traj koviC. 2000. Distributed Denial of Service Attacks. Proceedings in IEEE. June 2000.

[20] Williams, M.. EBay, Amazon, Buy.com hit by attacks. 2000. http://www.nwfusion.com/news/2000/0209attack.html. Visited on August 2008

[21] CSI/FBI. Cyber Attacks Continue, but Financial Losses are Down. 2003, DOI= http://www.gocsi.com/press/20030528.jhtml?_requestid=335314. Visited on August 2008.

[22] Robert Vamosi. 2008. Study: DDoS attacks threaten ISP infrastructure. Visited: December 04, 2008 10:20 AM PST http://news.cnet.com/8300-1009_3-83.html Posted: November 11, 2008

[23] Xuan, D., Bettati, R. and Zhao, W. 2001. A gateway-based defense system for distributed dos attacks in high-speed networks. In Proceedings of 2001 IEEE Workshop on Information Assurance and Security.

[24] Padopoulos, C., Lindell, R., Mehringer, J., Hussain, A. and Govindan, R. 2003. Cossack: Coordinated suppression of simultaneous attacks. In DARPA Information Survivability Conference and Exposition, Washington, DC, 1: 2–13.

Authors

Metasebia Kassa received his B.Sc. in Information System and M. Sc. in Computer Science from Addis Ababa University. He is currently working as senior technical Engineer (Technical Trainer) at ZTE (H.K.) Limited Ethiopian Branch (ZTE University). He has also been working as part-time Lecturer in the Department of Computer Science of Addis Ababa University. Metasebia has a broad research interest on Future Internet, Cloud Computing, Mobile Computing, and Security related topics.

Dr. Mulugeta Libsie received his B. Sc. in Statistics and Mathematics from Addis Ababa University, an M. Sc. in Computer Science from the University of Essex in England and a PhD in Computer Science from the University of Klagenfurt in Austria. Dr. Mulugeta is the author and co-author of several journal papers, conference proceedings and textbooks. He is currently an Assistant Professor in Computer Science at Addis Ababa University and Chairperson of the Doctoral Program Research Committee (DPRC). His research interest is in the area of distributed information systems.