

# Analysis of Automated Model against DDoS Attacks

Udaya Kiran Tupakula      Vijay Varadharajan

Information and Networked Systems Security Research  
Division of Information and Communication Sciences  
Macquarie University, Sydney, Australia-2109  
{udaya, vijay}@ics.mq.edu.au

*Abstract- Today Distributed Denial of Service (DDoS) attacks are causing major threat to perform online business over the Internet. Recently several schemes have been proposed on how to prevent some of these attacks, but they suffer from a range of problems, some of them being impractical and others not being effective against these attacks. Our previous work proposed an automated model that deals with the overall DDoS problem. With a new packet marking technique and agent design, our approach has many advanced features to minimize the DDoS attacks within a single ISP domain. In this paper we discuss different types of attacks that are possible on our model and propose techniques to counteract the identified attacks. We will also discuss security protocols for communication between different entities in our model.*

**Keywords:** Denial of Service, DoS, DDoS, congestion control.

## I. INTRODUCTION

Denial-of-Service (DoS) [1, 2] is an attempt by attackers to prevent access to resources by legitimate users for which they have authorization. DoS on the Internet has become a pressing issue following a series of attacks during recent times. Several papers [1, 3] have reported on the prevalence of such attacks on commercial servers and ISPs and the disruption they cause to services in today's Internet. In case of distributed denial of service (DDoS)[1,2], an attacker compromises several hosts on the Internet. Often, these are weakly secured machines and they are broken into using well-known defects in standard network service programs and common operating systems. These compromised computers are referred to as zombies. The attacker then uses these compromised computers to launch a coordinated attack on the victim machines.

Recently several schemes have been proposed on how to detect and prevent some of these types of attacks, but they suffer from a range of problems, some of them being impractical and others not being effective against these attacks. Further the proposed schemes deal only with some part of the distributed denial of service problem. For instance, [4,5] propose techniques to prevent traffic with spoofed address from crossing the network border, while some others [6,7,8,9,10,11] propose techniques to trace back the approximate source of the attacking system even in the case of spoofed source address. The work in [12] proposes a technique to automatically identify the attack at the congested router and treats the traffic by punishing only the identified aggregates that are causing congestion.

Our previous work [13] proposed an automated model that deals with the overall DDoS problem. The paper discusses techniques to identify the attack at the victim,

trace the approximate source of attack with a single packet even in case of spoofed source address, prevent the attack traffic closest to the attacking source and automatically tune to any changes in the attack traffic. This approach is based on controller-agent model and has several advantages compared to other proposed techniques.

The paper is organized as follows. Section II gives an overview of our automated model and also discusses different type of attacks that are possible on our model. Section III discusses techniques to counteract DDoS attacks on the Controller. Section IV discusses security protocols for secured communication between different entities in our model. Section V gives a general discussion of our model and Section VI concludes.

## II. AUTOMATED MODEL

Our previous work [13] considers an architecture that is divided into ISP domains and customers connected to these ISP domains. Routers are divided into internal routers and external routers. Internal routers belong to the ISP and external routers belong to the customers or another ISP. Internal routers themselves can be of two types, namely edge routers and transit routers. Edge routers are internal routers that are adjacent to one or more external routers. Transit routers are internal routers that are only adjacent to other internal routers. In Fig: 1 (R0, R3, R4, R5, R6) are edge routers and (R1, R2) are transit routers. While there is a chance for attack traffic to originate from the internal routers within the ISP's network domain, often most attacks originate outside the ISP's domain, pass through one or more internal routers of the ISP and target the victim (which is also outside the ISP's domain). Hence all the malicious traffic mostly passes through at least two edge routers of the ISP's domain. The malicious traffic passes through only one edge router of the ISP if both the victim and attacking source are connected to the same edge router. The attack traffic enters the ISP's domain at the Ingress edge and exits from the ISP's network through the Egress edge. The edge refers to adjacency between an edge router and an external router. Our work [13] focus on preventing these types of external attacks at the ingress edge router of the ISP and considers DDoS attacks with spoofed source addresses, broad attack signatures, attacking systems changing the type of attack traffic pattern. Attack signature refers to a pattern of traffic that is used to distinguish a malicious packet from normal traffic. Attack signature in this scheme refers to congestion signature in [12]. Victim can identify attack signatures with the congestion control technique discussed in [12] or by deploying some security tools like firewalls/Intrusion

detection systems that can perform real time traffic monitoring. The main aim of this approach is to identify and eliminate the malicious traffic from bad sources and provide more bandwidth to good traffic from the poor sources to the victim. This is achieved by preventing the attack traffic at nearest point to the source of attack (that is, at the Ingress edge of the ISP)

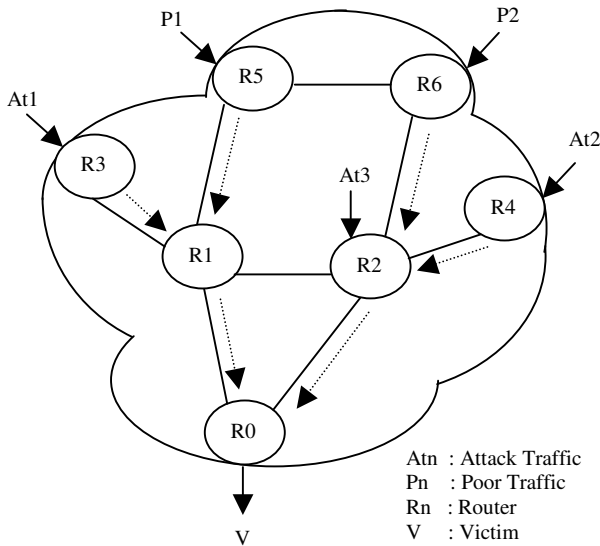


Fig 1: ISP backbone network

The proposed architecture involves a Controller-Agent model and assumes that no internal routers are compromised. In each ISP domain, there exists a controller, which is a trusted entity (within the domain) and is involved in the management of denial of service attacks. In principle, the controller can be implemented on any internal (transit or edge) router or at a dedicated host. Agents are implemented on all edge routers. The controller assigns a unique ID to each of its agents. The ID consists of a controller ID which is common for all agents and an agent ID which is unique for each agent.

During the time of an attack, the victim requests the controller in its domain to prevent the attack. A session is established between the victim and the controller after proper authentication of the victim. Depending on the number of agents present within its domain, the controller will generate and issue a unique ID (controller ID and unique agent ID) to its agents and commands the agents to mark the victims traffic with the issued ID. The controller updates the victim with the valid ID's. The agents filter the traffic that is destined to the victim and marks the packets with its unique ID (controller ID and agent's unique ID) in the fragment ID field of IP packet [15]. Packets will be marked in such a way that only the first agent that sees the traffic will mark the packet. If an agent receives a packet that is already marked then it checks the packet for a valid controller ID. Packets with valid controller ID are passed and the rest are dropped. All the fragments and packets that are marked by an attacker will be dropped in this stage. Since agents are deployed on all the edge routers, all the traffic to the victim is marked with

the controller ID and the unique ingress agent ID in the fragment ID field. As agents are implemented only on edge routers, all the traffic originating in the transit routers will be marked by the egress agent of the ISP, which is connected to the victim's network. In fig: 1 the traffic originating from router R2 is marked by the agent R0. Even if the source address of the attack traffic is spoofed, all the attack traffic originating from a particular host will have a common agent ID of the ingress agent that marked the packet. For instance, all the attack traffic At1 will have a controller ID and unique agent ID of R3 in the fragment ID field. So the victim can easily identify At1 as the attack signature for agent R3. Since the victim knows the controller ID and valid agent IDs, it can identify different attack signatures for different attacking sources based on agent ID. The attack traffic originating in the backbone (At3) will be identified as an attack signature for the egress agent (R0) that is connected to victim's network. Now the victim updates the controller with different attack signatures that are to be prevented at different agents. The controller retrieves the 32-bit IP address of the agent based on agent ID and commands that particular agent to prevent the attack traffic from reaching the victim. As attack signatures are identified based on the agent ID, only the agents through which the attack traffic is passing will receive this command. Now all the agents that receive this command will start preventing the attack traffic from reaching the victim. The traffic that is matching with the attack signature will be dropped and logged at the agent. The traffic that is not matching with the attack signature will be marked with the ID and destined to the victim. This is to enable the victim to easily track the changes in attack traffic pattern. The agents will update the controller on how much attack traffic they are receiving. Prevention will be done until the agent receives a reset signal from its controller.

The advantages associated with our approach are:

- Agents function as ordinary routers when there is no attack.
- Victim can easily identify the different attack signatures for different attacking systems.
- Once victim is authenticated, the process of identifying, preventing and tracking any changes in the attack traffic is totally automated. Once invoked, the response time to identify and prevent the attack is fast.
- Approximate source can be traced with a single packet and since all the packets are marked at ingress agent (nearest point to the attacking source); this helps to initiate legal proceedings.
- Filters are dynamically placed.
- Once attack signatures are identified, prevention of attack traffic is directly near to the attacking source (ingress agent).
- Each agent needs to check and prevent only the attack signature passing through it. The attack signature is very narrow and hence the delay experienced is very less.
- There is no need for the agents and controller to identify the attack signatures. Hence implementation costs for controller and agents is less when compared to the ACC routers [12].

- Fragmented packets destined to a potential victim are eliminated only during the times of attack.

Disadvantages of our approach are:

- The dropped fragments may include some good fragmented packets to the victim.
- Since there are only 16 bits in the fragment ID field of IPv4 packet, the proposed model is not suitable for an ISP with more than 65535 edge routers.

#### A. Attacks Possible on Automated Model

An analysis of the proposed model confirms the following attacks are possible on the proposed model. First the controller can itself be the victim of DDoS attack. Attacking sources can flood the controller with useless data or with false requests to prevent the attack. So high availability of the controller and load balancing at the controller are two important features that are to be considered while implementing the controller.

The approach requires proper authentication of victim to the controller to invoke the automated model. Since the victim network is outside the ISP domain, the communication between controller and victim is susceptible to masquerading, man-in-the-middle, repudiation of action and replay attacks. So the security protocols designed for communication between the victim and controller should provide mechanism to counteract these attacks.

The model [13] assumes that no internal routers are compromised. If there is no proper authentication of source address for the communication between controller and agents, the attacker can send false commands to the agents by spoofing his address with the 32-bit IP address of the controller. Similarly the attacker can spoof his address with the valid agent IP address and send false feedback messages to the controller. So, even if the controller and agents are located within the ISP domain and even if no internal routers are compromised, there should be proper authentication of source for all the communication between the controller and the agents.

The attacker can hack into the victim's machine and send false requests from the victim machine to the controller. We will not consider this case because if an attacker can get control of the victim machine, we believe that the attacker will create severe damage to the victim (rather than just sending a false request to the controller). So the victim should make sure that no false requests could be sent from the victim machine to the controller to invoke the automated model.

### III. COUNTERACTING DDoS ATTACKS ON THE CONTROLLER

The controller should be designed to operate manually or automatically. In case of severe DDoS attack on the victim, the victim may not be in a position to send a request to its controller. In this case the victim has to contact the controller out of band and the prevention process starts manually. In a simple scenario, the controller can be implemented on a dedicated host or any internal router. There should be a backup controller in case of crash of the primary controller.

The ISP can deploy more than one controller depending on the size of its domain.

Let us discuss some more techniques that can be used to avoid DDoS on the controller itself. The details of the controller can be made available to only pre-registered customers that wish to be protected from a DDoS attack. There can be some bandwidth specially allocated for communication between the controller and victims. The agents in the ISP can be configured to check whether the traffic directed towards the controller is originating from the respective customer domain (i.e. applying ingress filter [4] for all traffic from customers to the controller). Agents should be configured to allow traffic only from pre-registered customers to the controller. All the traffic originating from non-registered customers and all traffic from upstream ISP that is destined to the controller should be blocked at the ingress agent/router itself. If traffic to the controller from pre-registered customer is found to be malicious, then the controller can command its ingress edge agent/router (which is connected to the malicious customers network) to block all the traffic from the malicious customer domain, which is destined to the controller.

A large sized ISP can deploy multiple controllers within its domain. Multiple Controllers can be implemented with the windows 2000 advanced server or with windows 2000 datacenter server with windows clustering [16]. A cluster is a group of computers that, from victim/agent/application point of view, appear as a single computer. Windows clustering is a technology which, when implemented on 2 to 32 windows 2000 advanced servers, provide two important features: 1) High availability and 2) Automatic load balancing.

**High availability:** This feature is important for mission-critical applications. In windows clustering, if a computer in the cluster that is running critical application fails, another computer in the cluster will automatically start the application, and all the victims/agents/applications will be seamlessly directed to the computer that takes over running the application.

**Load Balancing:** This feature refers to spreading utilization across multiple computers. For example, if a controller experiences more load/utilization than a single computer can handle, it can be run on all of the computers in the cluster. Victims/agents will be seamlessly directed to the computer with the lowest utilization. So, high availability of the controller and automatic load balancing at the controller can be achieved by implementing the controller with windows 2000 advanced/datacenter server with windows clustering.

Since multiprocessing, multithreading and multitasking are possible with windows 2000 server [16], this is an added advantage when implementing controller with windows 2000 server. This enables the controller to handle attacks on multiple victims simultaneously. Further there are many advantages that can be achieved by implementing the controller with the windows 2000 advanced/datacenter server. The active directory feature in windows 2000 server can be used to simplify the authentication process of the victim to the controller. Active Directory is a directory service that stores information about various types of network objects, user accounts and computers in a hierarchical structure. With Active Directory, with a single login the users

can gain access to any service/resource that the user has permission to. In addition to active directory service, Windows 2000 server has many advanced features to supports remote authentication of users, smart card support and Internet Protocol security.

The implementation cost of controller is dependent on the size of the ISP domain. A large ISP should deploy multiple machines within the windows cluster to implement a controller and a small ISP can deploy only few machines within the windows cluster to implement a controller. However since there is need for only one controller (with multiple machines in windows cluster) in each ISP domain, it is still efficient to deploy the controller with additional cost.

#### IV. SECURITY PROTOCOLS FOR COMMUNICATION BETWEEN DIFFERENT ENTITIES

We can make certain security assumptions in our model [13] by virtue of its architecture. For example, the victim is a customer of ISP. So it can be assumed that victim has already registered with the ISP to access the services. This in turn helps to define the authentication service and protocols. Even if the model is invoked without proper authentication of the victim, since only the packet marking is done in the first stage, the victim can send a reset signal to its controller as soon as it receives marked packets. Hence prevention (the second stage in agent) process will not be invoked in this case. A complete description of all the security protocols as well as the rationale for the design choices can be found in [14]. Due to space limitations, we have decided not to include them here. In this section, we discuss simple authentication protocols for communication between the victim-controller and controller-agents.

In our model, the controller is assumed to be a trusted entity within the ISP domain. The customer who wants to be protected from a DDoS attack registers with the controller in its domain. At the time of registration, it is assumed that the controller and customers exchange their public keys. Suppose that  $(PK_C, SK_C)$  is the (public key, private key) pair of the controller and  $(PK_V, SK_V)$  is the (public key, private key) pair of the victim  $V$ . As shown in step 1 in Fig. 2, the victim that is under attack sends a request to the controller in its domain to prevent the attack. The request consists of a timestamp, a nonce (which is a random number), the identity of the controller  $C$ , and a session key  $K_{VC}$ , which is encrypted with the controller's public key  $PK_C$ . The whole request is hashed and signed with the private key of the victim  $SK_V$ . (Note that the notation  $\{x, y, z\}$  is used to indicate hashing of  $(x, y, z)$  using a suitable hash function such as SHA and then signing the hashed digest using the private key). This request is sent as a payload in the IPv4 packet with 32-bit IP address of the victim as the source address and 32-bit IP address of the controller as the destination address. Based on 32-bit source address of the received packet, the controller retrieves the public key of that particular victim from its database. Then it can verify the signed hashed digest. This authenticates the victim to the controller. Using its private key, the controller decrypts the encrypted portion to retrieve the session key. Now the controller responds by sending a new message as shown in step 2 in Fig. 2. As  $nonce_V$  is included within the signed hashed digest,  $V$  can believe that this is in response to its earlier message. As the  $K_{VC}$  is included in the signed hashed digest,  $V$  can believe that  $C$  has successfully

decrypted the session key. The successful verification of the signature authenticates the controller to the victim. The session key  $K_{VC}$  can now be used for subsequent communications between the controller and victim.

Step 1:  $V \rightarrow C$ :  $V, C, timestamp_V, nonce_V, [K_{VC}] PK_C, \{timestamp_V, nonce_V, C, K_{VC}\} SK_V$

Step 2:  $C \rightarrow V$ :  $C, V, timestamp_C, nonce_C, \{timestamp_C, nonce_C, nonce_V, V, K_{VC}\} SK_C$

Fig 2: Victim-Controller Authentication Protocols

In general, a similar protocol can be used to achieve authentication between the controller and the agents. In our model, since no internal routers are compromised, we use a symmetric key based system to achieve authentication between the controller and agents. Symmetric key authentication is more efficient from a computation perspective compared to public key based authentication. In this case, the controller shares a secret key with each of its agents. This is achieved as part of the agent installation and registration process. Suppose  $K_{CAi}$  is the secret key the controller shares with an agent  $A_i$ . If the controller wants to send a request to its agent, it encrypts the request  $R1$  with the symmetric key  $K_{CAi}$  and sends the data as a payload of IPv4 packet, with the 32-bit IP address of the controller as the source address and 32-bit IP address of the agent as the destination address. The agent is able to decrypt this message and authenticate the controller, as only the controller and the agent know the shared secret key. Similarly, the agent can send secure messages to the controller by encrypting the messages using the shared secret key. Once again we use timestamps and nonces to counteract replay attacks and use encrypted hashed digest to ensure the integrity of the messages.

#### V. DISCUSSION

In general, deployment of such solutions requires more than just technology. They often require political will and financial support. Our model can be readily implemented [14] with the existing routers with little modification and hence the financial cost is minimal for ISPs. Since this process is invoked only upon victim's request, an ISP can charge the victim to provide this service. The other advantage is that only the victim's traffic is filtered and processed separately. So the ISP need not worry about the performance degrading of its routers.

Though we have argued that our model can trace the approximate source of attack with a single packet, it should not be invoked to trace back single packets. It is obvious that potential customers have some form of security tools such as firewalls and intrusion detection systems at their end to protect their networks. For a simple DDoS attack, the victim may not even send a request to the controller to prevent the attack. The victim (customer) may request the controller only if the attack traffic increases over a threshold limit or if the attack continues for a longer period of time. We assume that the victim has a right to decide to whom s/he wants to provide access. For example, once the victim knows the

approximate network domain from which the attack traffic is originating, s/he may request the controller to completely stop all the traffic directed to the victim or to restrict the traffic to a particular bandwidth from the identified attacking domain.

Our packet marking technique helps to identify attack signatures by creating the agent ID in common for all the traffic originating from a specific Ingress agent. In many cases, apart from the agent ID, there will be more similarities within the attack packets and this helps in the identification of a narrow attack signature. Further, we believe that the victim can identify attack signatures more efficiently, since the victim can best decide what type of traffic is good or bad traffic for his/her network. Though the prevention is limited to a single ISP domain, it should be noted that there might be hundreds or thousands of routers within a medium or a large ISP. So tracing the Ingress edge of attack traffic by the process of input debugging can consume a lot of time. Our approach simplifies this process. We have only presented a basic version of our model. We are currently investigating a variety of DDoS attacks and customizing this model to suit for different attack scenarios.

We are currently extending this model to prevent DDoS attacks in multiple ISP domains. Our approach [17] involves the notion of hierarchy, where each ISP has a Controller. All these controllers will be implemented as Agents to other Controller, which we call the Master Controller. The Master Controller can be implemented at an ISP with large network or by grouping few ISP's. Whenever there is an attack, the victim can contact the Controller within its ISP's domain and the prevention of attack is done only within the ISP domain. If the attack is to be prevented in other ISP domains, the Controller in the Victim's domain will request its Master Controller to prevent the attack in other ISP domains. As prevention of attack has already started in the victim's domain, the controller in the victim's domain will also update the Master Controller with the Controller ID used to mark the packets in its domain. Now the Master Controller will assign a unique Controller ID for the Controllers in other ISP's domain and commands the Controllers to prevent the attack. An advantage of this approach is that it can be implemented using the Routing Arbiter [18] with minor modifications. There are several advantages of implementing the extended model with the Routing Arbiter. For instance, since the Routing Arbiter/ Route server does not forward any IP packets, it is itself protected from DDoS attack. Also it is possible to prevent multiple attacks on multiple victims in multiple domains, as all the policies of each ISP are stored in the Routing Arbiter Database (RADB).

## VI. CONCLUSION

In this paper, we have identified different attacks that are possible on our automated model against DDoS attacks and proposed some techniques to counteract the identified attacks. We have also discussed security protocols for communication between different entities in our automated model.

## REFERENCES

- [1] Computer emergency response team. CERT advisory CA-2000-01 denial-of-service developments. <http://www.cert.org/advisories/CA-2000-01.html>, Jan.2000.
- [2] Computer emergency response team. CERT advisory CA-1999-17 denial-of-service tools. <http://www.cert.org/advisories/CA-1999-17.html>.
- [3] David Moore, Geoffrey M. Voelker and Stefan Savage, "Inferring internet denial -of-service activity," In proceedings of the 10<sup>th</sup> USENIX Security Symposium, August 2001.
- [4] P.Ferguson and D.Senie, "Network ingress filtering: defeating denial of service attacks which employ IP source address spoofing," RFC 2267, January 1998.
- [5] SANS institute resources. Egress filtering. February 2000. <http://www.sans.org/y2k/egress.htm>.
- [6] Hal Burch and Bill Cheswick, "Tracing anonymous packets to their approximate source," In proceedings of Usenix LISA, December 2000.
- [7] Steve Bellovin, " The ICMP traceback message," <http://www.research.att.com/~smb>, 2000.
- [8] Drew Dean, Matt Franklin and Adam Stubblefield, " An algebraic approach to IP traceback," In proceedings of NDSS'01, February 2001.
- [9] Stefan Savage, David Wetherall, Anna Karlin and Tom Anderson, "Practical network support for IP traceback" In proceedings of the 2000 ACM SIGCOMM Conference, pages 295-306, August 2000.
- [10] D. Song and A.Perrig, "Advanced and authenticated marking schemes for IP traceback," Technical Report UCB/CSD-00-1107, University of California, Berkeley, June 2000.
- [11] Robert Stone: "CenterTrack: an IP overlay network for tracking DoS floods," In proceedings of the 9<sup>th</sup> Usenix Security Symposium, August 2000.
- [12] Ratul Mahajan, Steven M.Bellovin, Sally Floyd, John Ioannidis, Vern Paxson, and Scott Shenker, "Controlling high bandwidth aggregates in the network," Draft, February 2001.
- [13] Udaya Kiran Tupakula, Vijay Varadharajan, " A practical method to counteract denial of service attacks," In proceedings of the twenty -fifth Australasian computer science conference (ACSC2003).
- [14] U.K.Tupakula, V.Varadharajan, "Model and mechanisms for counteracting distributed denial of service attacks," Technical Report, Macquarie University, 2002.
- [15] J.Postel : Internet protocol. RFC 791,Sept.1981.
- [16] Alan R.Carter, "Installing, configuring & administering windows 2000 server, networking infrastructure & directory service".
- [17] U.K.Tupakula, V.Varadharajan, "Counteracting DDoS attacks in multiple ISP domains using routing arbiter architecture," In proceedings of the 11<sup>th</sup> IEEE ICON2003, September 2000.
- [18] D.Estrin, J.Postel, Y.Rekhter, "Routing arbiter architecture," <http://www.isi.edu/div7/ra/Publications>.