A CONTROLLER AGENT MODEL TO COUNTERACT DoS ATTACKS IN MULTIPLE DOMAINS

Udaya Kiran Tupakula   Vijay Varadharajan
Information and Networked System Security Research
Division of Information and Communication Sciences
Macquarie University, Australia. *{udaya, vijay}@ics.mq.edu.au*

Abstract: In this paper we discuss techniques to prevent Distributed Denial of Service (DDoS) attacks within the ISP domain and extend the scheme to prevent the attack in multiple ISP domains. With a new packet marking technique and agent design, our model is able to identify the approximate source of attack with a single packet and has many features to minimise DDoS attacks.

Key words: Denial of Service, DoS, Packet Marking, Routing Arbiter.

1. OUR APPROACH

Our architecture involves a Controller-Agent model. In each ISP domain, we envisage that there exists a controller, which is a trusted entity (within the domain) and is involved in the management of denial of service attacks. We consider external attacks where attacks originate outside the ISP domain and target the victim, which is also outside the ISP domain. Routers are mainly classified into internal and external routers. Internal routers belong to the ISP and external routers belong to customers or other ISP’s. If internal routers are connected to one or more external routers, they are called as edge routers, otherwise they are referred to as transit routers. 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. If transit routers were known to contribute a large amount of attack traffic, then the agents can be deployed on the transit routers as well and this requires no modifications to our scheme. The controller and agents are identified with their ID's. The controller assigns an ID for itself and a unique ID 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 the controller ID and unique agent ID to each agent and commands its agents to mark the traffic to the victim. Now the controller updates the victim with the controller ID and the unique agent IDs. The agents filter the traffic that are destined to the victim and mark the traffic with controller ID and its unique agent ID in the fragment ID field. 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 at 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 ingress/first agent ID in the fragment ID field. As we have assumed that agents are deployed only on the edge routers, the traffic originating in the backbone will be marked by the egress agent of the ISP that is connected to victim's network. Since the victim knows the controller ID and valid agent IDs, it can identify different attack signatures based on agent ID. 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 agent 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. Only the traffic that is matching with the attack signature will be dropped and logged at the agent. The traffic that does not match the attack signature will be marked with the controller ID and agent ID and destined to the victim. This is to enable the victim to easily track the changes in attack traffic. 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.

2. EXTENDED MODEL

We now extend our model to prevent the attack in multiple ISP domains. Each ISP domain is to have a controller. The controller maintains the database of all the agents in its domain and the controller's in the other domains. Whenever there is DDoS attack, the prevention process will initially begin within the victim's domain. If the attack persists for a long time or if the victim requests to prevent the attack upstream, then the prevention can be performed in other ISP domains. Now the controller in the victim's domain requests to have a session with the controllers in other ISP domains. A session is established between the controllers after proper authentication between them. Now the controller in the victim's domain requests the controllers in other domains to mark the packets destined to the victim. To avoid overlapping of controller IDs, the controller in the victim's domain issues a unique controller ID for every controller. The decision is based upon the number of agents present for each controller. The controllers in each domain assign unique agent ID to its agents and update the controller in the victim's domain with the valid agent IDs. Now the controller in the victim's domain updates its agents with the valid controller IDs in other ISP domains and updates the victim with valid controller IDs and agent IDs for each controller. The controllers in other domain will command their agents to mark the traffic to the victim. The process is similar to the marking in the victim's domain except that the unused bit in the flags field of the IP packet is enabled in this case to indicate that the packets are marked in other ISP's domain. Now the traffic to the victim is marked in all domains. When the packets marked in other ISP's domain enters the victim's ISP domain, if the unused flag bit is enabled, then the agents in the victim's domain can identify that the packets are marked in the other ISP's domain. Since the controller in victim's domain has already updated its agents on the valid controller IDs, the agent still only needs to check for a valid controller ID to pass the packet.

Another way to implement our model is by introducing the notion of hierarchy. In this approach each ISP will have 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 in 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 in its domain to mark the packets to the victim. 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. The prevention process is similar to the first approach. We prefer to use the second approach, as there is already an implemented (working) architecture that suits our model (Routing Arbiter [2]) with a little modification. The Routing Arbiter project is deployed at the Network Access Point (NAP) where multiple ISP's peer with each other. The routing arbiter is developed to simplify the routing process in multiple domains by taking all the policies of the ISP's into consideration. There are many tools that can be used to enhance the functions of routing arbiter. For example, there are tools that can be used to simplify the process of routing, maintain a centralised database of all the policies of ISPs and to automatically generate low-level router configurations from the high-level policy specifications.

Our model will be implemented in a hierarchy where the routing arbiter is the master controller and the backbone routers at the network access points are the agents for the master controller. These agents are the controllers for each ISP, which have their agents within their domains. There are several advantages of implementing our 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.

3. CONCLUSION

In this paper we have proposed a Controller-Agent model to prevent DDoS attacks within the ISP domains and extended the scheme to be implemented in multiple ISP domains. A detailed description of our model and single ISP implementation can be found in [1]. In a separate paper, we will discuss the implementation of the model in more detail and also describe secure authentication between different entities in our architecture.

REFERENCES

1. U.K.Tupakula, V.Varadharajan, "Model and Mechanisms for Counteracting Distributed Denial of Service Attacks", Technical Report, Macquarie University, 2002.
2. D.Estrin, J.Postel, Y.Rekhter. "Routing Arbiter Architecture," http://www.isi.edu/div7/ra/Publications.