2007－CSEC－37 (5)
2007／5／25

Distributed Solution Against Distributed Denial of Service (DDoS)

Ghanmi Elyes Yoshihito Oyama Eiji Okamoto
{elyes,oyama,okamoto}@cipher.risk.tsukuba.ac.jp
Laboratory of Cryptography and Information Security
Graduate School of Systems and information Engineering
University of Tsukuba

Abstract **- Distributed denial of service (DDoS) attacks has became a world wide threat and a major security problem since the second half of the 90s. The main task of defense systems is to detect and stop the attack in a short time but also recognize the legitimate traffic from the malicious one to allow users to access to the target during the attack. Unfortunately, there is no single deployment schema which allows to meet all those requirements. The detection of the attack is more accurate close to the victim whereas the distinction of the legitimate traffic is more accurate close to the source. Additionally, source or victim based solution alone can be overwhelmed by the traffic. From this point we can affirm that a distributed defense solution deployed close to the victim and the source seemed to be the most efficient one against DDoS. In this paper, we discuss the different mechanisms of the implementation of this solution and its advantages in comparison to related works.**

I. INTRODUCTION

Distributed denial of service (DDoS) attacks has become one of the major threats to the cyber security. It consists on sending a high volume of malicious but legitimate-like packets from a large number of infected attack sites. The aim of those attacks is to overwhelm the victim and render it incapable to treat the legitimate traffic by consuming its key resources. It may also, through the reflectors and zombies [12], create a congestion in the internet core which can be in the way of users using the encumbered routers.

Another traffic type called a "flash crowd" is experienced when many legitimate users start to access one particular site at the same time. For example, the vote for the host of the 2008 Olympics games triggered such an event. In a flash crowd, a link is inundated with many more requests than it can handle, and either the server runs out of resources (e.g., TCP buffer space), or there is not even enough bandwidth in the link for TCP connections to get out of slow-start and back off uniformly [13]. It can also be considered as an attack.

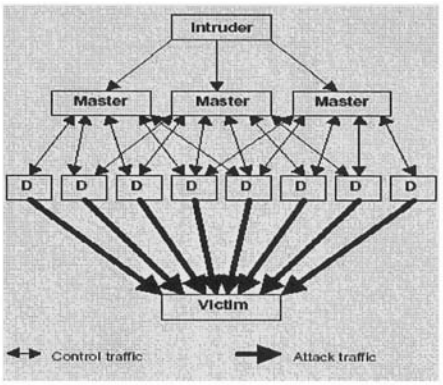

Figure 1 : Principle of a DDoS attack

The figure 1 shows a simplified attack scenario made by one attackers controlling three masters. The masters are using eight zombie machines (Devices D). If we consider that those zombies machines are simple users, we notify the toughness of the separation between the legitimate and the attack traffic and the traceback of the attacker.
A DDoS attack system requires coordination of different systems: masters, zombies, and the victim. To generate a flood of network traffic to the victim's site, the attacker issues commands to "master" computers, which in turn each send commands to a troop of zombie computers. One hacker can get 10,000 zombie machines together and aim them at one or more Web sites. Furthermore, DDoS attacks are very hard to defend against because the attacker usually falsify his IP source address which makes the separation between the genuine and the legitimate traffic almost impossible to realize. In addition, the use of zombies machines and reflectors, render the trace-back of the attackers out of our skills.
The ideal defense system is to secure all the machines connected to Internet, which is unrealistic. Most of the defending related work systems were deployed either on the source or the victim's node. This deployment strategy renders the solution less efficient.
In this paper we propose a distribute solution, composed by two agents which can be deployed in each edge node of any Internet service provider deployed on the source or victims level.
The principle of the solution is to continuously monitor the traffic and the vital resources of the victim. If an attack is detected, it warns the other node and try to collaborate together to stop the attack, and allow legitimate users to access to the server in the same time, which constitutes the main requirements for a defense system against DDoS attacks.
The reminder of this paper is organized as follows: Section II reviews the previous work in defending against DDoS attacks. In section III, we explain the motivation of our proposal. The section IV of this paper describes the proposed system, specifically its monitoring, detection and Section V shows the development of the solution on a Linux router. Section VI investigates security issues and reviews the future work and section VII concludes the paper.

II. RELATED WORK

Many previous research and commercial products tried to tackle the DDoS problem. Among them, there is the source or victim based defense solution with deterministic or probabilistic way of computing statistical values. None of them is really efficient. Savage et al [14] propose to let routers mark packets probabilistically, so that the victim can collect the marked packets and reconstruct the attack path. Song et al [15] proposed a probabilistic marking packets solution to reduce the false positive rate for reconstructing the attack path. Another enhanced scheme of probabilistic packet marking has been proposed to reduce the computational overhead [16]. Bellovin [17] proposed also a similar probabilistic packet marking solution for ICMP "traceback" in which routers generate ICMP packets to the destination with a low probability. For a significant traffic flow, the destination can gradually reconstruct the route that was taken by the packets in the flow. It has later been extended by Wu et al. [19]
But in this section, we are going to focus on the solutions which presents a cooperation and distributed aspect between its components. For more details about the other solutions please refer to the paper posted in [2].
Local aggregate-based congestion control (Local ACC) [3] provides a solution deployed on a single router to detect and rate limit the DDoS traffic or other traffic spikes (Slashdot Effect [4]). The principle of the Local ACC, is to detect early signs of congestion on the routers and rate limit each link which it has a high bandwidth.
An extension to the local ACC was Pushback [5]. It adds the communication and coordination between the nodes. If a congested node cannot impose a rate-limit itself, it asks its upstream neighbors to rate-limit the link. This link can apply the request or send a request by itself. With this communication, Pushback may inflict significant damage on the network's traffic following the same path.
Secure Overlay Service (SOS) [6], [7] consists on protecting victims by hiding their location in a large peer-to-peer overlay network. In that case they are not reachable for DDoS attacks. However, the most chief inconvenience is that the customers have to be part of these networks and be aware and cooperative with it. It may not be suitable in the case of a popular service. This was modified with WebSOS [8] by introducing Turing test to SOS, but this approach will only work for human users accessing to the service.
Active Security System (ASSYST) [9] consist on the deployment of its defense nodes on the edge routers. Actually, they are the equivalent of Node Controllers of our own solution but with a different behavior.
In [10] a collaborative defense system is proposed in which nodes act like gateways detecting and dropping attack traffic passing

through. The gateways are communicating together within the network of the source and the victim domain. A similar solution is COSSACK [11]. It forms a big cooperative group of nodes deployed at source and victims level and which communicates together to detect and drop the attack. But in these three last solution, since the system is not installed in a network, so this one doesn't participate to the defense system and is not covered.

III. MOTIVATION FOR A DISTRIBUTED SOLUTION

In defense systems against DDoS, three requirements are necessary: (1) detect quickly the attack, (2) respond on the attack to reduce it and (3)guarantee the legitimate traffic during the attack.

In DDoS defense solutions, the deployment point is,as we saw in the previous proposed solutions, very important. To detect the attack in an efficient way, the defense system have better to be as close as possible to the victim. Although, to distinguish between the legitimate and the malicious traffic, we have to deploy the solution close to the source (Ideally in the edge node of the ISP of the source), but we cannot do both placed just in one side of the network. Most of the solutions proposed in the past, were founded on either one or the other of these two scenarios, In addition, in a source or target based solution, it's very difficult to secure all the victims or all the sources. Furthermore, the whole traffic can be concentrated to pass through few nodes which can constitutes some congestion avoidance. In the distributed solution, even if the solution is partially deployed, the attack can be detected and stopped and the traffic can be shared by the defense nodes. That's why, it seems logic to propose a solution trying to combine both of them for more suddenness and efficiency treating the attacks.

IV. SOLUTION OVERVIEW

A. System Architecture

Our solution is deployed on a distributed way on the network. Each node, which can be an edge or host router, can host the solution.The nodes implicated in the defense network exchange information about the attack to collaborate to decrease and stop the attack traffic. The fact that every router or gateway in the internet contains the defense solution is unfeasible for evident reasons. That's why our solution was designed to be perfectly efficient in a partial deployment. However, it becomes more valuable as more nodes are added to the defense network, to monitor more users and protect more servers.

The distributed solution consists on two agents separated by their functionalities:

- The System Monitor (SM) monitor the router's resources and spread an alert to the rest of the network if it detects an attack.
- The Node Controller (NC) distinguish between malicious and legitimate packets, drop the attack's traffic and stamp the legitimate one.

Each node can embody one agent or both of them, depending on its resources and the authorization within the peer network. However, it's better to place a monitoring system in the edge node of a public server and a node controller in the edge node of an Internet Service Provider (ISP).

Through the figure 2, we will explain what happens during a DDoS attack and the behavior of our system during this attack.

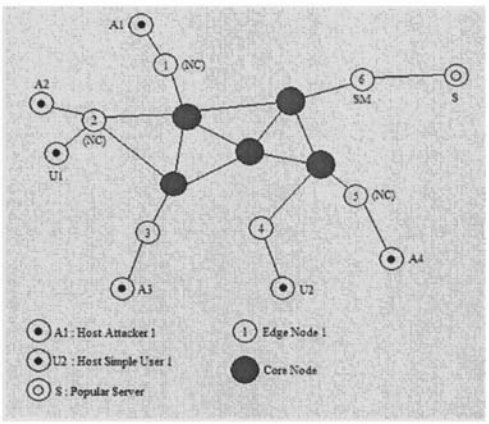

Figure 2 : Simplified deployment of the solution

We consider a DDoS attack on a popular web server S, perpetuated by a simplified number of attackers A through core nodes (in Grey). Simple users U are also represented to show the processing of the legitimate traffic. Our distributed solution is hosted as follows :

- In the edge nodes (1), (2) and (5) hosted by the Internet service Provider, the node controller (NC) functionality is installed.
- In the edge nodes (3) and (4), any defense solution is installed to represent the partial deployment of the solution.
- In the edge node (6), the system monitor (SM) is installed close to the popular server (S).

In fact, it's really difficult to handle the core nodes because they are usually privates and belongs to governments and private organisms.
Let's suppose now that the attackers A1, A2, A3 and A4 starts a massive distributed attack on the server S. They are falsifying and forging their IP source address.
Since the system monitor in the node (6) detects some abnormal behaviors in its critical resources (processor, memory, traffic,...etc), it concludes that an attack is occurring and spread an alarm to all the node controllers in its list. This alarm contains some information about the attack and a public key to help the NC recognizing the attack packets and stamp the legitimate ones. When the NC receive the alert, it starts to monitor the traffic and compute some statistic values to determinate which interfaces are responsible of the attacks, drop their originated packets in destination to the server S and stamp the others with the public key received from the SM.
In another way, after spreading the alarm, the SM, just accept the stamped packets with a valid session key. With this procedure, only legitimate clients from an ISP where the NC is installed will be accepted during the attack. This may cheer other ISP to deploy the solution to guarantee the access to the server to their customers during DDoS attacks. Belonging to the figure 2, the attackers A1, A2 and A4 will be dropped by the node controllers (1), (2) and (5). However, the attackers A3 will pass through the core nodes and reach the SM (6) (There is no CN on its edge router). But,since it won't be stamped,it will be dropped by the SM (6).
However, requests from the user U2 will be dropped by the SM because they are not stamped. This is unfortunate, but U2 can easily amend the situation by deploying a classifier node.

This example illustrates two major claims: (1) Attack traffic is controlled and the victim can resume its normal operation, and (2) legitimate traffic from networks protected by the NC continues to be served during the attack (User U1), while legitimate traffic from unprotected networks is dropped and doesn't even reach the server (User U2).It's also worth to note that this has been achieved with deployment of only few defense nodes. Naturally, if more nodes are deployed, then the scalability and effectiveness of the system is improved, but even with sparse deployment our system can provide significant benefits to its users.

B. *System Monitoring*

The SM plays an important role in our system. It monitors several system and network resources to be able to detect an attack. In fact it monitors the behavior of each peer with whom the victim or the target communicates, looking for signs of communication difficulties, such as a reduction in the number of response packets or longer inter-arrival times. In the same time, it periodically compares the observed values of the traffic statistics for each peer against a predefined model of normal traffic. If the comparison reveals the possibility of a DDoS attack, it spread the alarm. During an important event (exclusive information on a web server), it may happens that the server is overwhelmed by a legitimate traffic (flash crowd) [13]. The SM will send the alert even when it's not an attack. In those case, rather than run the risk of rendering the server's service unavailable, we have better to consider it as an attack allowing some customers and dropping others.
The system is handling several kind resources : rate of the CPU and the memory in use, number of connection request to the server and number of the packets received in a second.

Those parameters should be fixed by the network administrator, who is the most valuable actor to know his system specifications,rules and capabilities. As an example, and according to the system requirements, the administrator can set the rate at 80% of a normal use or 50% for a critical system.

When one of those rates is exceeded, the SM consult a list of the NC registered and send them an ICMP forged packet. It contains some important information to the CN, like the IP address of the server, the attack description and the key to stamp the packet. This key (16 bits) is randomly generated and is just used during the attack. In the same time, the system will start to drop the non-stamped packets.

C. *The Node Controller*

In the other side, when the NC gets the alert packet, it starts to compute the packets in direction to the server (Destination IP address = Server IP address) for each interface. In fact, it computes the rate between the outgoing packets to the server S and the whole outgoing packets. If this rate is high, so probably one or more attacker is passing through this interface. The NC will drop all the packets from this interface in direction to the victim S.
Others legitimate customers will suffer from this action, but it will be for few hours and for just one destination. In addition, it's absolutely necessary to rate limit the traffic in according to the node's resources. The other packets from the other interfaces, will be stamped. The most accurate field in the TCP/IP architecture to be used for stamping the packets is the fragmentation field (16 bits). The next figure shows the IP Header on 32 bits with the Fragmentation field in bold.

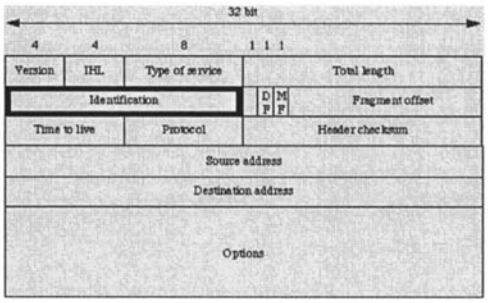

Figure3 : IP Header

As discussed in [12] this field is used for assembly of fragmented packets, but those packets represent a very small portion of the internet's traffic. Since we are going to stamp packets only during attacks, the utilization of the field will not disturb the normal traffic.

V. IMPLEMENTATION

After the design part of the project which uses the UML-RUP model, we decided to implement our solution. For that, we decided to work on a Unix environment and use C++ language to develop those two agents. Our choice was made knowing that the combination Unix/C++ is very efficient for network implementation and that we can come down in the lowest levels to be able to handle the packet and its fields.
In association to C++, we choose the gcc/makefile compiler and libpcap for the packet handling. Libpcap is a system-independent interface for user-level packet capture. It provides a portable framework for low-level network monitoring. Applications include network statistics collection, security monitoring, network debugging, etc. libpcap may be used by a program to capture packets travelling over a network and, in newer versions, to transmit packets on a network at the link layer, as well as to get a list of network interfaces that can be used with libpcap. Libpcap is the packet capture and filtering engine of many open source and commercial network tools, including protocol analyzers, network monitors, network intrusion detection systems,packet sniffers,traffic generators and network testers.
Two application are implemented : the System monitoring and the Node controller. Each application has its own functionalities and its own specification, but they are sharing some functionalities (sniffing, forging packets..etc).

VI. SECURITY CONSIDERATIONS AND FUTURE WORK

The most interesting security is the fact that the attacker can sniff a communication between the NC and the SM, get the key and use it to forge his own packets to look like a legitimate ones. Usually an attack lasts for just few hours, the attackers will hardly have the opportunity to detect the defense network and sniff the packets.
The key is randomly renewed also in each session, to do not allow the attacker to use it, if he did spoof it. Furthermore, in our future work, we are going to secure the communication between the two agents using a cryptic session.
In another part, when the attack is very intense, the SM may have to handle a large quantity of packets. As long as it has to check only the head of the packet (the fragmentation field), it will not consume its resources as much as if he would store and check the whole packet.
Also, the UDP packets cannot be stamped in the same way of the IPv4 ones. It can constitute a big problem to the defense network since their time of computing is quite big. In our solution, we choose to drop them during the attack.

An attacker could also perform a denial-of-service attack on the source network, preventing the response packets from reaching the server. In that case the edge router should be completely protected as a part of a private and functional network. Since this system is under implementation using C++ language under Linux systems, we will deploy the solution on a test network and compute some effective values to demonstrate the efficiency of this system.

VIII. CONCLUSION

We have presented a distributed solution Against as a novel approach to defend against Distributed Denial of Service (DDoS). We have demonstrated that the biggest threats in the cyber world can be handled with a distributed cooperative solution. Furthermore, we have shown the effectiveness of building a distributed solution to detect and stop the attack serving the legitimate users in the same time. Victim networks have protection from the attack and the source nodes have the warranty that their legitimate host will pass through. For that we have designed a efficient, practical and realistic solution which can be deployed in any network.

ACKNOWLEDGEMENT

We would like to thank the Laboratory of Cryptography and Information Security of the University of Tsukuba for making available their resources.

REFERENCES

[1] J. Mirkovic, G. Prier, and P. Reiher, "Attacking DDoS at the Source," in Proceedings of the ICNP 2002, November 2002.
[2] Jelena Mirkovic, D-WARD: Source-End Defense Against Distributed Denial-of-Service Attacks, Ph.D. thesis, University of California Los Angeles, 2003.

[3] R. Mahajan, S. Bellovin, S. Floyd, V. Paxson, and S. Shenker, "Controlling high bandwidth aggregates in the network," ACM Computer Communications Review, vol. 32, no.3,July,2002.
[4] S Adler, The Slashdot Effect: An Analysis of Three Internet-Publications.
[5] J. Ioannidis and S. M. Bellovin, "Pushback: Router-Based Defense Against DDoS Attacks," in Proceedings of NDSS,February,2002.
[6] A. D. Keromytis, V. Misra, and D. Rubenstein, "SOS: Secure Overlay Services," in Proceedings of SIGCOMM 2002,2002.
[7] A. D. Keromytis, V. Misra, and D. Rubenstein, "Using Overlays to Improve Network Security ," in Proceedings of SPIE ITCom Conference on Scalability and Traffic Control in IP Networks II, July 2002.
[8] Debra L. Cook, William G. Morein, Angelos D. Keromytis, Vishal Misra, and Daniel Rubenstein., "Websos: Protecting web servers from ddos attacks," in In the Proceedings of the 11th IEEE International Conference on Networks (ICON).,2003.
[9] R Canonico, D Cotroneo, L Peluso, S P Romano, and G Ventre, "Programming routers to improve network security," in Proceedings of the OPENSIG 2001 Workshop Next Generation Network Programming, September 2001.
[10] D Xuan, R Bettati, and W Zhao, "A gateway-based defense system for distributed dos attacks in high-speed networks," in Proceedings of 2001 IEEE Workshop on Information Assurance and Security, June 2001.
[11] C Papadopoulos, R Lindell, J Mehringer, A Hussain, and R Govindan, "Cossack: Coordinated suppression of simultaneous attacks," in Proceedings of DISCEX III, April 2003.
[12] S. Savage, D. Wetherall, A. Karlin, and T. Anderson, "Practical Network Support for IP Traceback," in Proceedings of ACM SIGCOMM 2000, August 2000.
[13] John Ioannidis and Steven M. Bellovin. Pushback: Router-based defense against ddos attacks. AT&T Labs Research,February,2001.
[14] Stefan Savage, David Wetherall, Anna Karlin, and Tom Anderson. Practical network support for ip traceback. In Proceedings of the 2000 ACM SIGCOMM Conference, August,2000.
[15] Dawn X. Song and Adrian Perrig. Advanced and authenticated marking schemes for ip traceback. In Proceedings of IEEE INFOCOM 2001, 2001. http://paris.cs.berkeley.edu/perrig/projects/iptraceback/tr-iptrace.ps.gz.
[16] Tao Peng, Christopher Leckie, and Kotagiri Ramamohanarao. Adjusted probabilistic packet marking for ip traceback. In Proceedings of Networking 2002, Pisa, Italy,May,2002.
[17] S. Bellovin. The icmp traceback message. Internet Draft,IETF, March 2000. draft-bellovin-itrace-05.txt (work in progress). http://www.research.att.com/~smb.
[18] S. Felix Wu, Lixia Zhang, Dan Massey, and Allison Mankin. Intension-Driven ICMP Trace-Back. Interner Draft,IETF, February 2001. draft-wuitrace-intension-00.txt.
[19] S. Felix Wu, Lixia Zhang, Dan Massey, and Allison Mankin. Intension-Driven ICMP Trace-Back. Interner Draft,IETF, February 2001. draft-wuitrace-intension-00.txt(work in progress).