Swing - A Novel Mechanism Inspired by Shim6 Address-Switch Conception to Limit the Effectiveness of DoS Attacks

Xiangbin Cheng, Jun Bi, and Xing Li
Network Research Center, Tsinghua University, Beijing, 100084, China
junbi@tsinghua.edu.cn

Abstract

Denial-of-Service (DoS) attacks play a significant role among all the network security issues today. In this paper, we present a mechanism (called Swing) to limit the effectiveness of DoS attacks. Inspired by the address-switch conception of the newly proposed shim6 protocol, Swing tries to protect servers from attacks by using a new strategy. In the mechanism, when a DoS attack is detected, the server will automatically change its address to get rid of the attack. Meanwhile, existing connections from normal clients will be kept using an address-switch protocol like shim6. A p2p network is included in the mechanism to help clients establish new connections to the server under attack situations, and side equipments are deployed near the server to monitor and reshape the network flow. This mechanism suggests a new kind of strategy to defend DoS attacks, and provides a resilient and effective solution.

1. Introduction

DoS attacks play a significant role among all the network security issues today, and the need for an effective defending mechanism is quite urgent. In a typical attack, the attacker aims to exhaust the victim's resources by sending large amounts of packets or triggering massive connection requests at a burst. So in order to limit the effectiveness of DoS attacks, a mechanism should be able to protect the server from both types of the flows above. Besides, in order to save resource and management costs, it is necessary for the mechanism be carried out only when an attack is detected. And we also think that it will be better if the users can immediately benefit from the mechanism after deploying it. This will provide greater motivation for new users and ISPs, and accelerate the deploying process of the mechanism.

In this paper, we present a mechanism called Swing to limit the effectiveness of DoS attacks. Inspired by the newly proposed shim6 protocol [10], we suggest a new type of strategy to defend DoS attacks: Dynamic Network Topology Modification with Automatic Address Switch. In our mechanism, when a DoS attack is detected, the server will automatically change its address to get away from the attack. Meanwhile, existing connections from normal clients will be kept using an address-switching mechanism like shim6. A p2p network is included in the mechanism to help clients establish new connections to the server under attack situations, and side equipments are deployed near the server to monitor and reshape the network flow. This mechanism can protect servers from DoS attacks caused by massive packets and burst requests. And each server that deploys the mechanism can benefit from the protection immediately after the deployment.

The rest of the paper is organized as follows: in Section 2, we analyze the existing strategies of DoS defense mechanisms. In Section 3, our mechanism – Swing is presented with a detailed illustration of its working process. Section 4 contains some analysis on our mechanism and discusses about future work. Section 5 concludes the paper.

2. Related Work

Defense against DoS attacks has been a very active research topic, and lots of mechanisms using different strategies have been proposed. In this section, we'll choose several of the most typical strategies, and carry out some analysis on them.

In order to defend DoS attacks, some traditional solutions focus on filtering packets with spoofed source addresses out from the internet, like ingress filtering [1] and route-based filtering [2]. Although they may help eliminate some DoS attacks, the effectiveness of these mechanisms usually requires a wide deployment, which is very hard to achieve in today's internet due to technical, financial and other reasons. Another DoS defense strategy is called "capability-based", including

978-0-7695-3106-9/08 $25.00 © 2008 IEEE
DOI 10.1109/ICN.2008.79

SIFF [3] and TVA [4] mechanisms. This strategy requires that the client get a certain token from the server before it actually sends the packets, and packets without valid tokens are dropped by routers in the network. These mechanisms work in a way that a server can control its clients' flow rates by giving out different tokens. But they require software support from many routers in the network, which makes it hard for them to be deployed. Also, strategies from the network architecture aspect have been proposed. SOS [5] and RP2P [6] build up an overlay network above current network infrastructure, and packets can only reach their destination through certain paths in the overlay. Although servers can be protected by hiding themselves from clients outside the overlay network, these mechanisms both require a constant maintenance of this overlay network, which is an extra cost. Besides, authentications need to be carried out at both the entrance and exit of the overlay network, which will increase the computing cost and affect the performance of the mechanism.

Recently, a new mechanism called "dFence" [7] has been proposed by some researchers. In this mechanism, several middle-boxes are deployed into the server's network. When an attack happens, the mechanism carries out some re-routing procedure and dynamically inserts these boxes before a server to help with DoS defense. This mechanism tries to protect the server by changing the network's topology under attack, which is a strategy similar to our mechanism. The difference between dFence and our mechanism is that we include a seamless address-switch mechanism in our strategy, so the server can change its address directly and get away from the attack in a much faster way.

To summarize, our mechanism differs from most other mechanisms in the strategy aspect. Inspired by the newly proposed shim6 protocol, we suggest a new type of strategy to defend DoS attacks: Dynamic Network Topology Modification with Automatic Address Switch. The network topology gets changed under DoS attack, and the server changes its address directly to protect itself. As for the DoS detection part, there have been many proposed methods [8] [9] to detect various kinds of DoS attacks. We can combine our mechanism as a responding part with those mechanisms, and create an integrated DoS detect and defense system.

3. Swing: A Novel Mechanism to Limit the Effectiveness of DoS Attacks

3.1 System Architecture

The architecture of Swing consists of three parts: Gateway Box (GB), Side Equipments (SE) and Backup Address Servers (BAS). GB and SE are located near the server and managed by the user that deploys Swing, while BAS is a group of servers distributed in the internet and managed by a third party. Figure 1 shows the architecture of our mechanism.

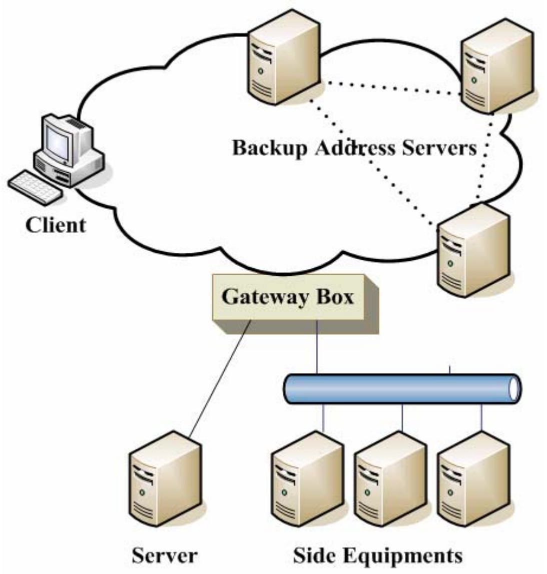

Figure 1. Swing architecture

In the figure, GB is a device transparent to network layer. Its function is to separate shim6 flows from other flows under attack situations. Shim6 flows are sent to the server directly. Other flows are guided to SE, where certain filtering and reshaping algorithms are carried out to produce an acceptable flow rate back to the server. SE can also be used as a tool to monitor and analyze the behavior of DoS attacks. BAS is designed as a way to help clients establish new connections with the server under attack, and can be used as a motivating mechanism as well. The following chapters illustrate each part of Swing in detail.

3.2 Gateway Box

Gateway Box is designed as a partition device during attack. Under normal circumstances, GB does nothing on the packets flowing through it, and its existence is completely transparent to the network layer. When an attack is detected, the server will change its address using shim6, and GB will start its function. The server will tell GB about its original address and new address, and GB separates packets into two categories according to their destination addresses. The packets containing the new address as their destination will be delivered to server directly, and those with the old address will be guided to SE for monitoring and reshaping. Meanwhile, reshaped flows from SE are also delivered to the server in consideration of those clients without shim6. The flow chart of GB during an attack is shown in Figure2.

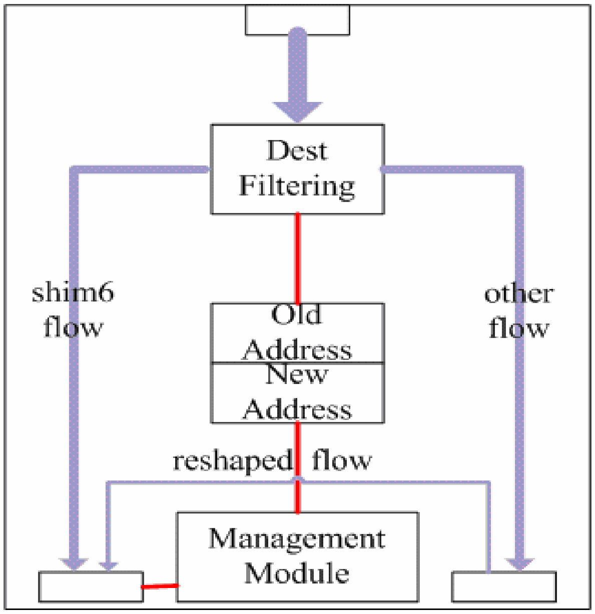

Figure 2. Flow chart of GB during attack

As indicated in Figure 2, after an attack happens, flows go through GB and get reshaped before they actually reach the server. This will lighten the server's burden, and help improve its availability.

3.3 Side Equipments

After flows have been separated at GB, packets with the server's original address will be guided to Side Equipments for further processing. At the current stage of designing, our SE is a single machine that only performs flow reshaping. However, in practice where Swing is actually put into use, SE can be either one single machine or a group of machines combined together. And its functions can be extended to network monitoring, source address validation or trapping and analyzing DoS behaviors. These extensions can make Swing a more sophisticated DoS defense system, and the further study is in the schedule of our future work.

Currently, the SE in our mechanism has the function of accepting all the packets coming in, reshaping them, and sending them back to the server at an acceptable rate. Since shim6 is a newly proposed protocol, not all clients will have shim6 support. It is important for us to consider these clients while designing our mechanism. In the current design of Swing, when SE gets the data packets during an attack, it will put them into different queues according to their source addresses, and an algorithm containing both round-robin and priority factors is carried out to control the flow rate from each queue. Packets that exceed the buffer length of its queue will be discarded. In this way, we can protect the server from bandwidth DoS attacks. The SE also puts all connection requests (like SYN) in a special queue to prevent DoS attacks like SYN-flood. The modules in SE are shown in Figure 3.

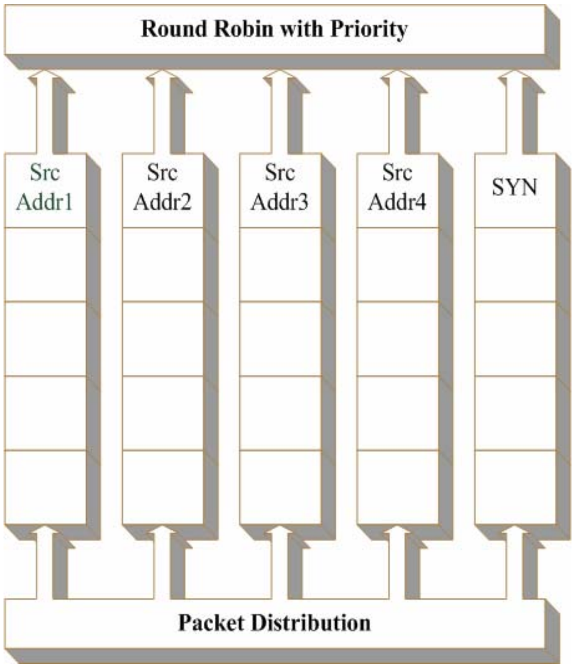

Figure 3. Modules in SE

Usually, a round-robin algorithm will seem fair with all the clients. However, we also have to consider the situation of DDoS attacks. In DDoS, packets usually come from massive different source addresses. This will create a lot of queues in SE, and normal clients will wait a longer time in the round-robin process, which leads to a fall down of their flow rates. The same problem will happen when the attackers launch an attack by sending packets with lots of randomly generated source addresses. In order to solve this problem, we combine the round-robin algorithm with priority. The server can assign a high priority number to certain addresses or networks to help them get a better flow rate under attacks. This priority can be based on the client's history of action, user type (like VIP users) or the trust level of a network. Since the priority is generated by server and directly sent to SE, DoS attackers won't be able to intercept the information, so they can't spoof packets with high prioritized source addresses. In this way, "good users" will be protected during DoS/DDoS attacks, and the network flow rate can still be successfully controlled. A low priority number can also be used to limit clients with bad behavior in the history so that normal clients will have a better flow rate.

3.4 Backup Address Servers

A high priority number will give "good clients" a better flow rate during DoS attacks. However, it would still be better if they can directly get to the server without going through SE. Based on this thought, we added BAS mechanism to Swing. These BAS are distributed in the network, and managed by a third party. A server can register in one of the BAS, and when it changes its address under an attack, it informs the BAS of the new address. In this way, BAS will always have the information about the server's current address, and authorized clients can get this information from them and set up connections with the server directly.

To balance the system load, there will be a group of servers instead of a single BAS. These servers form a p2p network, and information updates are spread among them. After the server registers at one BAS, all the other BAS will get the information. Meanwhile, the server tells its trusted clients (like VIP users) about the list of BAS and some information for authentication. The client chooses a BAS from the list, authenticates itself and gets the server's current address.

3.5 Working Process

After introducing each part of the Swing mechanism, we can take a look at its working process and the things that happen during a DoS attack. When an attack is detected, the server will first generate a new IPv6 address for itself. Since the space of IPv6 address is quite large, this procedure will not have any problem. Then, the server changes its address and tells the clients with shim6 support about the address switch so existing connections can be kept. After that, the server sends some control information to GB and SE to start them. It also informs BAS about the address change if necessary. Then GB and SE will work in the ways described above and protect the server from the attack. Meanwhile, clients of different types will be affected in different ways:

1. Connected clients with shim6 support. Existing connections will be kept using shim6. The client will automatically change the destination address field in its packets to the server's new address. So the attack won't affect this type of clients.

2. Connected clients without shim6 support. Packets from these clients after the attack will be guided to SE, where they enter different queues and wait to be sent by the round-robin algorithm. The client's flow rate will be affected to some extent by the attack, and available flow rate to the server depends on the number of queues in SE and the priority of the client.

3. "Good clients" with BAS information. If there is already a connection, their packets will be handled with a high priority in the SE. So even if the attack is launched by sending packets with lots of randomly generated addresses, or a DDoS attack happens, this type of clients will still get a relatively high flow rate. If they want to set up new connections, they can contact with BAS, get the server's current address, and connect with the server directly.

4. "Good clients" without BAS information. If there is already a connection, their packets will be treated in the same way as above. However, if they want to set up new connections, their requests have to go through SE and wait in the SYN queue. But their priority will help the requests get a better place in the queue, and reach the server in a shorter time.

5. Normal clients. Both data packets and connection requests from these clients have to get in the queue and wait to be sent. However, since they don't have the priority as the good clients, if DDoS happens, or the attack contains packets with lots of randomly generated addresses, the number of the queues will become very large, and the packets will have to wait a longer time in the round-robin, which causes them to be affected more by the attack. But the round-robin algorithm ensures that each queue can have its share in a cycle, so the availability of the server will still be protected for these clients.

6. Clients with bad history. This type of clients will be seriously affected by the attack. The low priority gives them very low flow rates, and their requests to establish new connections will wait a much longer time before they can get through. However, since these clients are often related to bad behaviors in the past (like a botnet node controlled by attackers to carry out DoS/DDoS attacks), we believe it's only fair that they be less concerned during the attack period.

Currently, Swing only works in IPv6 networks with shim6 support. However, the strategy we suggested in this paper can be also applied to IPv4 networks if corresponding address-switch protocols come up. By making some simple adjustments in the implementation to handle different address types, Swing will be able to work in both IPv4 and IPv6 network environments.

4. Analysis and Future work

4.1 Analysis

Section 3 gives a detailed illustration of Swing, and we will carry out some analysis on it in this Section. From the user's point of view, there are some major issues to concern when a new mechanism is proposed:

effectiveness, security of the mechanism system itself, extra costs brought by the new mechanism, and reason to deploy it. Swing has a good performance in all the issues mentioned above, and we will analyze them in this chapter.

1. Effectiveness

By using the methods mentioned in Section 3, Swing can protect the server from those DoS attacks caused by massive packets and burst requests. The round-robin algorithm reshapes flow rates into an acceptable range, thus protects the server from the former attack. New connection requests are gathered into a queue and join the round-robin, which protects the server from the latter attack. The priority mechanism also helps protect clients from DDoS attacks and DoS attacks by packets with randomly generated source addresses. So, the Swing mechanism can limit the effectiveness of DoS attacks effectively.

2. Robustness

Since the system itself may also become the target of an attack, it is important to study the security of the system itself. As described in Section 3, Swing consists of three parts: GB, SE and BAS. GB is a middle box completely transparent to network layer. Under normal circumstances, it does nothing. During an attack, its function is just to separate packets into two groups according to their destination addresses. This process only requires one lookup, and the size of the lookup table is only 2. By analyzing this processing procedure, we can see that the processing cost is very low, so we don't need to worry about the performance of GB. In the current stage, SE may be a vulnerable factor in the system since it's only a single machine that carries out flow reshaping functions. However, in practice where Swing is actually deployed and put into use, SE can consist of several machines combined together. This will greatly improve its ability to handle massive amounts of packets, and the risk of SE becoming a new bottleneck can be reduced to a minimum. As for BAS, since only a small group of clients know about the BAS list, the possibility of them being attacked is much smaller compared to those well-known servers. Besides, BAS is only a backup way for clients to establish connections with the server under attack situations. Under normal circumstances, these BAS don't have much importance. This makes them less likely to become DoS attack targets. And even if they have been attacked, clients are still able to establish connections through the round-robin process. So BAS also won't become a security problem in Swing.

3. Extra Cost

In Swing, GB is designed as a device transparent to network layer. SE is directly connected to GB, and only gets in the middle between the server and its network during an attack. So, no extra cost is needed in routing or firewall policies when Swing is deployed. The server will have to generate a new address and send some information to GB, SE and BAS on the detection of an attack, but no further actions are needed after the initiation. The same conclusion can be made on those costs brought by shim6 protocol. In a nutshell, deploying Swing won't bring much extra cost to the user's original network.

4. Deployment

In our mechanism, GB and SE are located near the server. As soon as a server deploys Swing, it will benefit from the protection immediately. This will provide a good motivation for the deployment of Swing. Besides, the server's ability to assign priority numbers to certain clients makes it possible for server to provide differentiated service qualities, which may also bring some commercial benefits.

4.2 Future work

In order to test its performance and other aspects, we will carry out some experiments and collect test results in our future work. In the current design, we use the most common round robin algorithm to reshape the network flows. In order to verify its performance and find the proper parameters or even a better algorithm, different algorithm implementations should be carried out to collect information. Besides, as mentioned in Section 3, many extensions can be made on SE to provide functions like network monitoring, source address validation or trapping and analyzing DoS behaviors. We will also study the combination of Swing with other mechanisms in the future, and help create a comprehensive DoS detect and defense system.

5. Conclusion

In this paper, we present a mechanism called Swing to limit the effectiveness of DoS attacks. It is based on a new strategy that the network topology gets changed under DoS attack, and the server changes its address directly to protect itself. This mechanism can protect servers from those DoS attacks caused by massive packets and burst requests. And each server that carries out the mechanism can benefit from the protection immediately after the deployment. We illustrate our mechanism in detail, and carry out some analysis. In the future, experiments will be performed to help improve the algorithm, and new functions will be added to the mechanism to create a comprehensive DoS detect and defense system.

References

[1] P.Ferguson, D.Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks Which Employ IP Source Address Spoofing", RFC 2267, January 1998.

[2] K.Park and H.Lee, “On the Effectiveness of Route-Based Packet Filtering for Distributed DoS Attack Prevention in Power-Law Internets” Proc. ACM SIGCOMM’2001, August 2001.

[3] A.Yaar, A.Perrig, and D.Song, “SIFF: A stateless Internet flow filter to mitigate DDoS flooding attacks.” Proc. IEEE S&P, 2004.

[4] X.Yang, D.Wetherall and T.Anderson, “A DoS-limiting network architecture.” Proc. SIGCOMM, 2005.

[5] A.Keromytis, V.Misra and D.Rubenstein, “SOS: Secure Overlay Services.” Proc. ACM SIGCOMM, August 2002.

[6] Shigang Chen, Randy Chow, “A New Perspective in Defending against DDoS.” Proc. IEEE FTDCS’04, 2004.

[7] A.Mahimkar, J.Dange, V.Shmatikov, et al, “dFence: Transparent Network-based Denial of Service Mitigation.” 4th USENIX NSDI, Cambridge, MA, Apr 2007.

[8] T.Gil and M.Poletto, “MULTOPS: A data-structure for bandwidth attack detection.” Proc. USENIX Security, 2001.

[9] H.Wang, D.Zhang and K.Shin, “Detecting SYN flooding attacks.” Proc. INFOCOM, 2002.

[10] E.Nordmark, “Level 3 multihoming shim protocol” draft –ietf-shim6-proto-08.txt, Apr 2007.