International Journal of Application or Innovation in Engineering & Management (IJAIEM)
Web Site: www.ijaiem.org Email: editor@ijaiem.org
Volume 3, Issue 3, March 2014 ISSN 2319 - 4847

Parallel Ranking Assist against Distributed Reflection Denial of Service Attack

Sumitha.J.S[1], D.Devibala[2]

[1]M.Tech Scholar, Computer Science &Engineering Department, SRM University, Chennai
[2]Assistant Professor, Computer Science &Engineering Department, SRM University, Chennai

Abstract

Distributed Reflection Denial of Service is the recent iteration in the series of Denial of Service attacks. It works similar to Distributed Denial of Service, in that it uses many sources to attack one victim and the attacker hides behind the zombies. In this paper, we concentrate on assisting the nodes of network during the DRDoS attack, by using detection algorithm to detect the attack whenever a suspicious flow is noticed and then by proper analysis of the network we can find the attack free path which can be used by the nodes in the network. We use Rank Correlation based Detection algorithm which helps to find whether the network is experiencing a channel failure or is under attack. Once the attack is detected, the attack path and source are multicast to all nodes, so that the nodes in the network can avoid any traffic from them, thus reducing the effect of DRDoS attack for a specified period of time.

Keywords: Distributed Reflection Denial-of-Service (DRDoS) attack, Distributed Denial of Service attack, Rank Correlation based Detection, matching algorithm.

1. INTRODUCTION

DRDoS is the next generation of Distributed Denial of Service (DDoS), which uses an ingenious variation on the traditional SYN attack to actually trick innocent servers and core infrastructure routers into unknowingly executing a DDoS attack. DRDoS uses legitimate hosts called "reflectors" to flood the victim by making slaves spoof the victim's address. A reflector may be any IP host that will respond to other request messages, like SYN, SYN/ACK, ICMP request, DNS queries and so on.

The procedure of DRDoS attack is briefed in Figure 1. It works as follows: An attacker first controls some zombies and locates a large number of reflectors. Then it sends attack commands to zombies. When received attack commands, the zombies send request packets with victim's address to the reflectors. That is, zombies send Request with Source: victim and Destination: Reflector. And reflector, based on the forged source addresses in those Request packets will send Response with Source:Reflector and Destination:Victim. At last, victim is flooded by the numerous unsolicited response packets from the reflectors adding up to significant bandwidth, enough to congest the victim's Internet connectivity. With bandwidth maxed out, legitimate clients are not able to connect with the victim.

There have been some packet-level defense methods. Filtering all incoming response packets, which is of low cost, will result in no general access to the remote server [3]. Inspecting packet content and tracking protocol status maybe helpful, but need a lot of computation which is also vulnerable to attacks [6, 7]. Along with more protocols being exploited to launch DRDoS, countermeasures should treat different protocols specifically the list should be updated manually. These problems give rise to need for some protocol independent methods to help detecting most kinds of DRDoS. This paper concentrates on solving this problem. The basic traffic pattern near the victim under DRDoS is studied, and proposes a general detection method: the Rank Correlation based Detection [1], which is protocol independent and its computation cost is not affected by network throughput. In this algorithm, once an attack alarm raises, upstream routers will sample and test rank correlation coefficient of suspicious flows and use the range of correlation value for further detection. Correlation has been successfully used in DDoS detection, e.g., correlation coefficient has been successfully employed to discriminate DDoS attacks from flash crowds [8]. The rest of the paper is organized as follows, section 2 provides an overview of existing countermeasures against DRDoS attacks, Section 3 gives a detailed definition of the attack detection steps and algorithms used in proposed system and finally conclusions are drawn in Section 4.

There are many detection mechanisms for DDoS attack, but they cannot be applied to DRDoS because, compared to the typical DDoS attack, the DRDoS attack is more dangerous, for the following reasons. First, the DRDoS attack traffic is further diluted by the reflectors, which makes the attack traffic even more distributed. Second, the distributed reflector denial of service (DRDoS) attack has the ability to amplify the attack traffic, which makes the attack even more potent. There are detection mechanisms for Detecting DRDoS attacks like, using a simple response packet confirmation mechanism, monitoring the volume of traffic that is received by the victim, source IP address monitoring and so on.

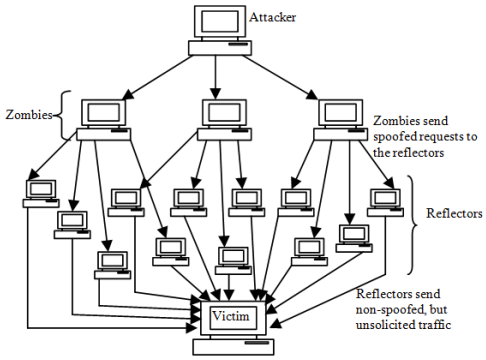

Fig. 1: DRDoS Attack

2. PROBLEM ANALYSIS

Simple response packet confirmation mechanism [6] monitors only limited pairs of requests and responses, and confirms the validity of the received response packets based on the request–response relationship. Therefore, the proposed method does not need complicated state management such as the stateful inspection method, and thus the detection mechanism becomes simple and is of low cost.

Next method is monitoring the volume of traffic that is received by the victim [7]. A major drawback of these approaches is that they do not provide a way to differentiate flash crowds from DDoS attacks. As the internet traffic is very dynamic, a sudden increase of in traffic may be mistaken as an attack. If we delay our response in order to ensure that the traffic increase is not just a transient burst, then we risk allowing the victim to be overwhelmed by a real attack. Moreover, some persistent increases in traffic may not be attacks, but actually "flash crowd" events, where a large number of legitimate users access the same website simultaneously.

A better approach is to monitor the number of new source IP addresses [4], rather than the local traffic volume. It has been observed that during bandwidth attacks, most source IP addresses are new to the victim, whereas most source IP addresses in a flash crowd appeared at the victim before. It was proposed to monitor the number of new IP addresses in a given time period in order to detect bandwidth attacks. It was demonstrated as a sensitive variable for detecting bandwidth attacks than monitoring the total volume of incoming traffic.

All these methods use packet-level defense mechanism. This paper concentrates on some rule independent mechanism to detect the attack. The Rank Correlation based Detection [1] makes use of Spearman's rank correlation coefficient for the algorithm.

3. SYSTEM DESCRIPTION

In this paper we focus in detecting and avoiding the DRDoS attack using a method which is protocol independent and has less computation cost. Figure 2 gives an outline on how the proposed system works. We propose to have a shared monitoring of network by all hosts. Each node has a monitor, which helps to maintain an event list which consists of the source, destination, time frames and role. Seeing the event list, monitor can easily find the suspicious flow. Once a suspicious flow is found, the rank correlation based detection algorithm is used to detect the DRDoS attack. Once the attack is detected, the list is used to find attack path and the packets are not entertained from that path for a specified time period. After the time period is over, the network is restored and again if there is any suspicious flow RCD algorithm is used to detect the attack. If there is no attack, the packet transfer can occur in normal manner.

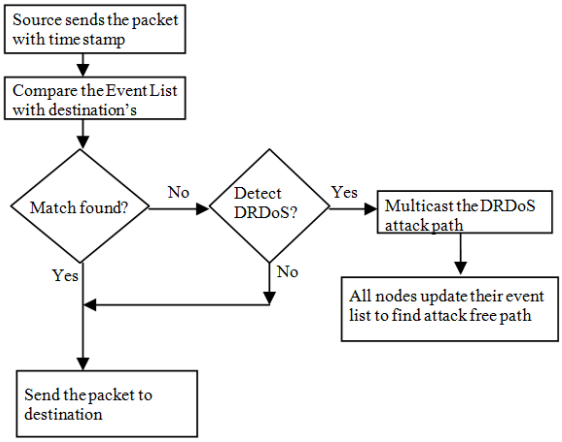

Fig. 2 : Flowchart illustrating the system process

3.3 Shared Monitoring of Network

The basic idea is to set up a monitor at each node in the network to produce relevant information about the network state and to share them among all the nodes. A monitor can be thought of as an instance of the ethereal network packet sniffer. It captures the traffic and displays the detailed information on it. For each captured packet monitor displays a complete view of packet headers and payload and add some general statistics as the timestamp, frame number and length in bytes. Information is stored in the form of list of events. Events are the single transmitted packet or the times in which the channel is idle, which can be inferred from the timestamp of the packets and the packet transmission times.

The combination of different list of events leads to the better understanding of what happened in the network, in particular in distinguishing the jamming attacks and channel failures, where packets are sent by one peer and never received by the other peer. Both the channel failure and a jamming attack make the FCS check of the packet fail, thus the packet in transit will be incorrectly received and dropped, incrementing the "dropped frames" counter in the device driver at the receiver. The difference between the two cases is the amount of incorrectly received frames at the receiver. Suppose if the receiving station is under jamming network, where the packets which pass through the jamming area get scrambled. The monitor placed at the sender's side will see the number of frames sent on the channel and the monitor at the receiver end won't see anything received correctly, and will keep on increasing the incorrectly received frames counter. The sender will retry the transmission a number of times and all these retransmissions will be dropped as well, incrementing the counter.

Jamming and channel failure have the same basic signature (which is packets transmitted and never received), but differentiate on their position in the event list. A few packets disappearing here and there are index of channel failures, while a sequence of disappearing packets is considered as jamming.

3.4 Rank Correlation based Detection

3.2.1 Correlation Coefficient

The responses from the reflectors are found to have inherent relations: linear relation, as they are stimulated by same attacking flow. The Rank Correlation based Detection algorithm [1] is based on this observation. The Pearson's correlation coefficient is suited for linear relationship, but due to its sensitivity to outliers caused by traffic burst, the linearity may not be obvious. The Spearman's rank correlation coefficient (Spearman's rho) is more suitable for detection, where a raw value is converted to a ranked value and then Pearson's correlation is applied.

In Spearman's correlation coefficient, for two random variables X and Y of ranked values, the expected values are μ_X and μ_Y , and standard deviations are σ_X and σ_Y . The coefficient $r_{X,Y}$ is their covariance normalized by the standard deviation:

$$r_{X,Y} = \frac{\operatorname{cov}(X,Y)}{\sigma_X \sigma_Y} = \frac{E((X-\mu_X)(Y-\mu_Y))}{\sigma_X \sigma_Y} \qquad (1)$$

Where E is the expected value, and *cov* is the covariance which could also be represented using E, then it has:

$$r_{X,Y} = \frac{E(XY)-E(X)E(Y)}{\sqrt{E(X^2)-E^2(X)}\sqrt{E(Y^2)-E^2(Y)}} \qquad (2)$$

The value range of $r_{X,Y}$ is [-1,1], closer to 1 represents stronger positive linear relationship while closer to -1 represents stronger negative linear relationship, whereas 0 means no linear relationship.

3.2.2 Rank Correlation based Detection Algorithm

For this algorithm, we suppose that the packets pass though one router to reach the victim. The packet flow is sampled per unit time T. Fig.3 shows two suspicious flows *fa* and *fb*, their respective set of source reflectors are *Ra* and *Rb*, where the set of uninvolved reflectors are *Ro* when a suspicious flow is alerted. In this algorithm, once an alert appears, routers in the path will sample flows for sufficient time.

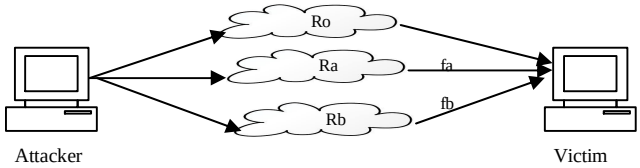

Fig. 3 Attacking Scenario

Ideally, for two pure attacking flows *fa* and *fb*, correlation coefficient *ra,b* will be close to 1. The traffic in the background will not allow satisfying this assumption for Internet, but, correlation between two suspicious flows is strong compared to others. When a DRSoS attack happens, we use two threshold values $\delta 1$ and $\delta 2$ to judge about the suspicious flow, whether they are reflection flows or not. *Ra,b* = 1 means that both are reflection flows.

$$Ra,b = \begin{cases} 0, \text{ for } \delta 1 \le r_{a,b} \le \delta 2 \\ 1, \text{ for } r_{a,b} < \delta 1 \text{ or } r_{a,b} > \delta 2 \end{cases} \quad (3)$$

The threshold values are determined based on different factors like network scenarios, attack contexts etc. So, the values differ for different networks and are derived statistically after proper analysis.

Steps in Rank Correlation based Detection Algorithm

1. Suspicious flow is located.
2. For each suspicious flow, sample the number of packets for a short time period.
3. The coefficient $r_{X,Y}$ is calculated for the suspicious flow pairs according to (2)
4. Compare coefficients for suspicious flows and make decision by (3).
5. If reflection flow is confirmed, then discard these flows on the routers.

The computation cost of Rank Correlation based Detection Algorithm is not affected by the network throughput because of only taking packet count into consideration. The result could also be used to pick out and discard malicious flows.

3.5 Matching Algorithm

The basic algorithm to match two lists of events is as follows: we start from the first list and for every event (packet or channel idle) we try to find a matching event on the second list that is, given a packet we look for it on the second list. As we don't have cheaters into play for now, what we find is that for every packet on the first list we find it on the second one if the network worked fine, else we find a channel idle event if some problem (jamming or malfunctioning) happened. Continuing the example above, we'd have transmitted packets on the first event list and channel idle (together with a high number of dropped packets) on the second one. We can find unmatched events on the second list at the end (for example if the first node was jammed), so we merge the two lists and run the rank correlation algorithm again.

Since all nodes participate in the detection process, we extend it in order to match multiple lists. The idea is to merge one list at a time with the result of the previous merge. In other words, we merge lists 1 and 2, and then we match the result with list 3, until we processed every list. We obtain in this way an aggregated list of all events which happened in the network in a given time frame. We have to notice here that a node might not overhear the traffic of every other node because of range. We supposed that each node has relevant information to offer, but this is not always true.

The key feature here is that the monitoring system is distributed. A single station alone cannot tell if it is experiencing an attack or just a temporary network failure, and cooperation among all nodes is required for the nodes to understand what is going on. The event lists are shared among all nodes in the network.

All nodes send their evidences to every other node in the network. Every node executes the rank correlation algorithm to generate the aggregated event list to have a clear view of what happened in the network in the given time frame.

3.6 Multicast the DRDoS attack to the neighboring nodes

Now that the DRDOS attack is detected, the address of the DRDOS attack path is sent to the entire network by multicasting. Neighbor nodes receive the IP address of the DRDOS attack path and store it in the event lists to prevent future attacks from that node in the network. The multicasting of the DRDOS attack address is done by source.

3.7 Sending data to the destination

The data send process is done by splitting the chosen text file into packets for transmission. The data send process is invoked after the source finds a DRDOS attack free path. In the case of jamming/network malfunction, the source waits till the network is restored, starts the training process to find the DRDoS attack and if any detected, selects a path free from DRDoS attack. The source sends the data directly to the destination through the 'safe' path. Destination receives the data in the form of packets and checks for anomalies to detect any loss of data in the data due to DRDOS attack.

4. Conclusion and Future Works

DRDoS attacks are a growing problem. The main question is "How do we know if we are under attack"? The option we have covered has its pros and cons. The Solution concentrates on detecting DRDoS independent of specific protocols using the Rank Correlation based Detection algorithm. We also suggest some methods to reduce the disadvantages of DRDoS by identifying the path causing the attack and avoiding the path to send packets for a specified time period. There are a lot of interesting works in the future, including:

1) Extend the experiment against real DRDoS in the Internet.
2) The algorithms can be used in more complicated network scenario which uses many routers.
3) Include tracing methods to find the attacker for better avoidance of the attack.

REFERENCES

[1] Wei Wei, Feng Chen, Yingjie Xia, and Guang Jin "A Rank Correlation Based Detection against Distributed Reflection DoS Attacks", IEEE Communications Letters, Vol. 17, no. 1, January 2013.

[2] Lei Zhang, Shui Yu, Di Wu and Paul Watters "A Survey on Latest Botnet Attack and Defense", 2011 International Joint Conference of IEEE TrustCom-11/IEEE ICESS-11/FCST-11.

[3] Vern Paxson, "An Analysis of Using Reflectors for Distributed Denial-of-Service Attacks", Computer Communication Review 31(3), July 2001.

[4] "Detecting Distributed Denial of Service Attacks Using Source IP Address Monitoring", Tao Peng, Christopher Leckie, Kotagiri Ramamohanarao, In Proceedings of the Third International IFIP-TC6 Networking Conference(2002).

[5] Yonghui Li, Yulong Wang, Fangchun Yang, Sen Su , "Traceback DRDoS Attacks", Journal of Information & Computational Science 8: 1 (2011) 94–111

[6] T. Hiroshi, O. Kohei, and Y. Atsunori, "Detecting DRDoS attacks by a simple response packet confirmation mechanism," Computer Commun., vol. 31, no. 14, pp. 3299–3306, 2008.

[7] T. Vogt, "Application-level reflection attacks." Available: http://www.lemuria.org/security/application-drdos.html.

[8] S. Yu, W. Zhou, W. Jia, S. Guo, Y. Xiang, and F. Tang, "Discriminating DDoS attacks from flash crowds using flow correlation coefficient," IEEE Trans. *Parallel Distrib. Syst.*, vol. 23, no. 6, pp. 1073–1080, 2012.