International Journal of Computer and Internet Security.
ISSN 0974-2247 Volume 8, Number 1 (2016), pp. 1-8

http://www.irphouse.com

Application of Netflow logs in Analysis and Detection of DDoS Attacks

Siva Balaji Yadav.C
Research Scholar
SVU College of Engineering, SV University, Tirupati-India

R. Seshadri, PhD
Director,
SVU Computer Center, SV University, Tirupati-India

Abstract

The ability to detect DDoS attacks in the Internet backbone in real-time was crucial for network operators in order to provide an infrastructure of high quality. In this work, an algorithm for the near real-time detection of massive flooding attacks as well as TCP SYN flooding attacks has been developed and implemented as an UP Frame plug-in. The algorithm has been validated successfully by replaying Net Flow v5 logs of the SWITCH backbone border routers, containing known ICMP, UDP and TCP SYN flooding attacks from the past.

Keywords: DDoS, Attack detection, TCP flooding, UPFrame.

Introduction

Overview Denial of Service (DoS) attacks are one the major annoyance of today's Internet. Their sole purpose is, as the name indicates, to prevent an Internet server from doing its work- which is to provide a service to other Internet hosts. The impact of DoS attacks can vary from annoying regular users of the affected server, for example a web site, to financial and image damage of the server's operator. Whatever reasons motivate the launch of DoS attacks, they seem to be quite popular these days [1] and the Internet technology which was not designed with security concerns provides little means to prevent them. The following section gives a quick overview of different kinds of DoS attack.

DoS attacks

There exist many different kinds of DoS attacks [2, 3], which can be divided into two categories: Attacks that exploit a network protocol to prevent the attack victim from processing other requests, and attacks that are simply consuming all the available bandwidth of the victim. For the types of flooding attacks that are dealt with in this thesis, namely UDP flood, ICMP flood and TCP SYN flood, a summary is given in the following

UDP flood

An attacker sends a large number of UDP packets to one or several ports of the victim, which eventually renders the attacked host unable to process the incoming packets.

ICMP flood

An attacker sends a large number of ICMP echo requests to the victim to consume all its available bandwidth.

TCP SYN flood

As opposed to the two attacks described above, the TCP SYN flooding attack is a protocol exploiting attack. It exploits the three-way handshake of the TCP protocol by sending many TCP SYN requests to a victim. Because the attacked host maintains half open ports to wait for the connections to be established, it eventually runs out of resources [4].

Distributed attacks

Through the use of automated tools it is possible to lead all the attacks discussed above in a distributed way. This is done by compromising poorly secured hosts, so called zombies, and installs clients of the attack tool. Then the attacker can call all the zombies to launch an attack on the same host simultaneously [5-7].

DDoS attacks are a threat to the Internet. They decrease the service quality of Internet services– therefore it is important to have means to prevent or at least mitigate them. To apply countermeasures against Denial of Service attacks, or even to only analyze them, the first crucial step is to detect such attacks. In the DDoSVax project [8], the environment to detect DDoS attacks is the collection of flow-level network data, captured by the SWITCH [9] backbone border routers. The data is exported in Cisco NetFlow v5 format toUPFrame, a framework for real-time processing of NetFlow data. The task given for this thesis was to develop an algorithm which detects (Distributed) Denial of Service attacks on the basis of analyzing NetFlow records. The algorithm was to be implemented as an UPFrame plug-in for real-time processing, which restricted the possibilities in terms of resource usage and processing time [10].

Methodology for Algorithm

Basic idea:

Distributed Denial of Service flooding attacks concentrate network traffic from many hosts, each not needing to provide a high bandwidth network link, on a single victim host. Hence the closer to the victim a DoS detection algorithm is deployed, the easier it is to detect an attack. The idea of the algorithm developed in this thesis is to take advantage of this fact when finding a way to detect DoS attacks in the provided Internet backbone traffic. Algorithms for detection of DoS attacks in the backbone developed so far are mostly based on statistical anomalies of IP packet header fields. While some statistical analyses can be applied to network traffic at flow level as well (e.g. flow length, average packet size for a certain protocol), the approach chosen here also takes advantage of having a global view of a whole autonomous system (AS) and aims at detecting DoS attacks by observing potential victims. To detect an attack, the ratio of incoming to outgoing packets of a host is considered as a very general way to describe a Denial of Service attack. The idea is that while for regular network connections traffic is always bidirectional to a certain degree, in case of a DoS attack traffic is highly asymmetric in terms of the packet ratio. Thus, the primary objective of the algorithm presented here is to find hosts with a exceedingly high in/out packet ratio.

Description

Aggregating and storing the input data:

First of all, the continuous input stream of flows has to be aggregated for discrete time intervals. Here it has to be considered that the flows exported in the NetFlow logs are not strictly ordered in chronological order according to their end time. Furthermore, flows can be spread over several intervals, so there has to be taken care of assigning all the packets of a flow to the respective intervals. Since only the start and end time of a flow is known but not the time distribution of the contained packets, a linear distribution is assumed as an approximation. The possibility of flows exceeding one interval makes it necessary to store data for several intervals, so no further measures are taken on the basis of data that might not be appropriate due to possible subsequent flows. It is also necessary to store the number of packets for both sender and receiver of a flow since the sender could be falsely considered a victim after receiving many packets if the number of packets sent before was discarded. Besides the number of incoming and outgoing packets, the algorithm keeps track of the number of sent and received bytes and the number of incoming and outgoing short TCP flows for every sender and receiver IP address of input flows. The granularity of the observed addresses can be set to single host addresses or address ranges through specifying a subnet mask in the configuration.

Observation time intervals:

The length of the flows in the input is dependent on the active timeout of the NetFlow exporting router, in our case 15 minutes. That means the algorithm has to keep data for many intervals, depending on the interval length if we want to distribute also traffic reported in long flows to our e.g. 1 minute interval statistics. However, considering that most of the flows are shorter than 2 minutes, it is not necessary to store that much data. Instead, a two-phase method is applied. In the first place data is only stored for a small number of intervals, and only hosts whose packet ratio exceeds a threshold after that short observation period are considered interesting and therefore are to be observed for the maximum flow length to await possible subsequent flows containing packets for the respective interval. For long flows to an IP address which is only observed for the short period of time, only the last intervals are stored. This proceeding delays the detection of an attack for up to the difference between the short period and the maximum flow length. A packet ratio check of all stored IP addresses is done after each interval, and the observation time of high ratio hosts is extended. If the high ratio of a host is lowered under the threshold after processing a flow sent by it, the host is no longer considered interesting and therefore put back to short time observation.

Attack detection:

If a host has still a high ratio value for an interval after waiting for the maximum flow length, it becomes an attack candidate. Before it is logged as an attack victim, however, it has to pass some more tests. To be considered a victim of a flooding attack, its packet ratio has to exceed the attack threshold; furthermore the numbers of received packets and bytes have to exceed the respective thresholds. Involving more parameters makes it possible to suppress false alarms in a flexible way. To detect victims of TCP SYN attacks, the algorithm employs a second test, since in this case an attack is not necessarily determined by the victim receiving much data. For this case, the ratio of the number of incoming to outgoing short TCP flows of a host during the interval is considered. If this ratio exceeds the respective threshold as well as the number of incoming short TCP flows, the host is logged as a TCP victim.

Validation on known attacks :

The idea described in the previous chapter first had to be validated. To see if the considered parameters show the expected behavior in case of an attack, analysis was done on Netflow data from the archive, captured during later discovered attacks. Since for these attacks the victim was known, it was possible to filter the network traffic data so traffic from and to the attack victim could be observed. For this purpose, the host analyzer tool was developed.

The host analyzer tool

The host analyzer tool reads NetFlowlog files and observes network traffic from and to a specific IP address. After each interval it outputs the number of packets and bytes from and to the host, along with the average number of bytes per incoming packets and the number of active incoming flows during the interval. Before quitting it also outputs flow length statistics as well as protocol statistics.

Analyses of known attacks :
In the following, the results of the analyses of known attacks are presented. For each different attack, the graph of the ratio of incoming to outgoing packets is shown along with the number of incoming bytes. If no incoming packets were seen, the number was set to 1 for the calculation of the ratio to avoid a division by 0.

Characterization of attacks :
The analyses of the known attacks clearly showed that the in/out packet ratio for an Internet host serves the purpose of indicating the ongoing of a Denial of Service attack. With the clarification of this issue, however, the question of the parameters that distinguish attacks from regular traffic still remained. The assumption that network connections are always "bidirectional to a certain degree"has now to be specified in numbers. Observation of the www.ethz.ch web server on a random day shows that for http traffic the average of the in/out packet ratio is 0.9 at an average of 876 incoming packets per minute (Figure 1).

Results

Tests of the DDoS Detector plug-in on known attacks:
The plug-in has been tested by replaying the NetFlow log files containing the known attacks discussed in Section 5 from the DDoSVax archive to UPFrame. The known attacks have all been successfully detected by the DDoS Detector plug-in, along with some other events of high volume traffic to hosts that have not been reported as DoS attack victims. The results of testing the plug-in on the known attacks are shown in the following subsections. The start and end of the detected attack are directly drawn into the graph of the off-line analysis. The times have to be interpreted as follows:

t1: Start of the attack
t2: Start of attack detection
t3: End of attack detection
t4: End of the attack

The parameters to define an attack were set as follows during these tests:
packet_ratio_attack_threshold=100
packets_in_threshold=30000
bytes_in_threshold=50000000
tcp_short_pkt_ratio_threshold=15
short_tcp_flows_in_threshold=1000 7.

Online analysis of the ICMP flooding attack
Start of the attack: 23:06
Start of attack detection: 23:19
End of attack detection: 00:51
End of the attack: 00:51

Total number of bytes received during the attack, reported by the DoS Detector (for each of the attack victims): 3337002468

Online analysis of the TCP to myeth attack (Figure 2)
Start of the attack: 12:51
Start of attack detection: 12:53
End of attack detection: 13:04
End of the attack: 13:29
Total number of bytes received during the attack, reported by the DoS Detector: 1655014

Online analysis of the UDP to IRC server attack (Figure 2)
Start of the attack: 23:26
Start of attack detection: 23:37
End of attack detection: 23:44
End of the attack: 23:45
Total number of bytes received during the attack, reported by the DoS Detector: 5880443550

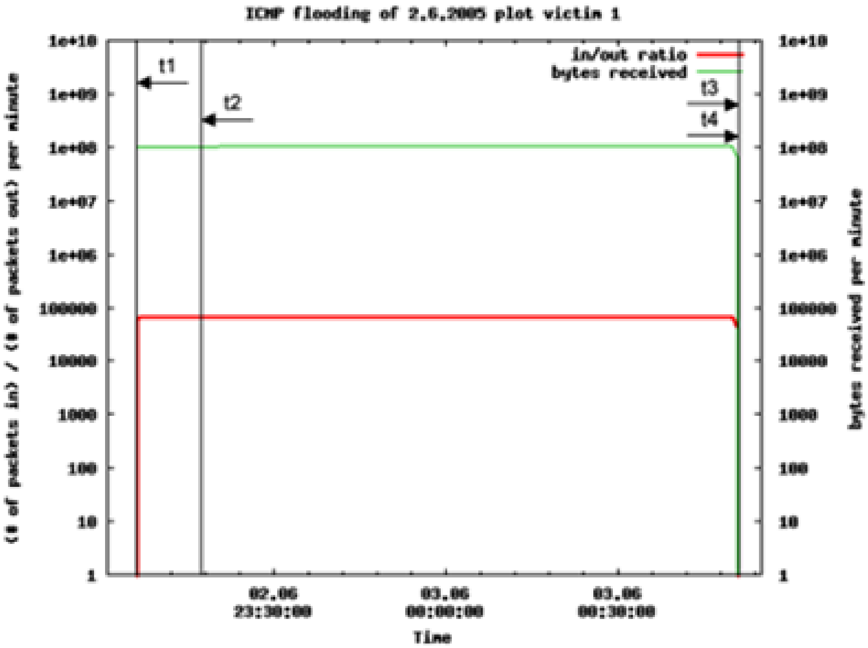

Fig 1: Online analysis of the ICMP flooding attack

Memory usage estimation:
The resource usage of the plug-in is strongly dependent on its parameters. Running the plug-in with short observation period of 3, long observation period of 17, in/out packet ratio threshold of 15 and memory usage upper bound of 500 MB, the program used between 100 and 120 MB during the tests. For an in/out packet ratio threshold of

15, the percentage of hosts with long observation time period is about 1 %. Limitation of memory can result in delay of detection of attacks or failure to detect attacks, if the victim hosts have an in/out packet ratio that is only slightly above the threshold, since these entries could be discarded in favor of entries with higher ratios. The ICMP and UDP attacks discussed in this section, however, were still detected by the plug-in when run with a memory limit of 20 MB.

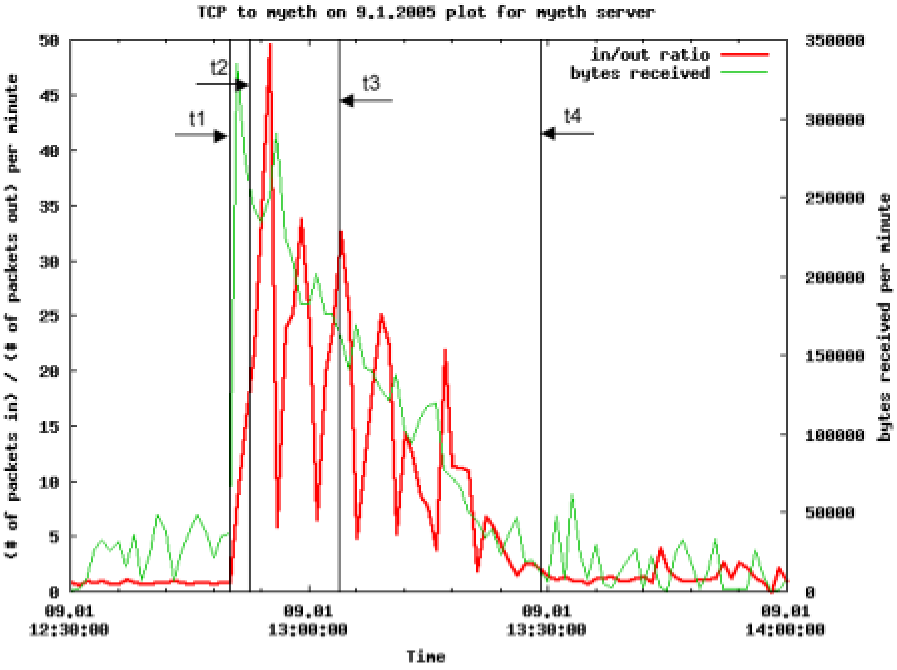

Fig 2: Online analysis of the TCP to myeth attack

Summary

The task of this diploma thesis was to design and implement an algorithm to detect DDoS attacks in the NetFlow logs of the SWITCH backbone border routers. With taking into account the analysis of existing DoS detection algorithms and extensive off-line analyses of known attacks, an algorithm was designed to meet the requirements. The algorithm has also been implemented in a real-time version as a plug-in for UPFrame, and successfully tested on archived data of known attacks from the past. The goals of this thesis were therefore fully achieved. Major challenges met during this work were the study of related work with regard to the usability of proposed techniques for the DDoSVax environment, the handling of the vast amount of NetFlowdata, that does not tolerate the slightest impreciseness, and the dealing with real-time problems arising with the design and implementation of the UPFrame plug-in.

References

[1] A. Akella, A. Bharambe, M. Reiter, and S. Seshan, Detecting DDoS attacks on ISP networks, In PODS Workshop on Management and Processing of Data Streams, 2003.

[2] J. Baras, A. Carenas, V. Ramezani, On-line detection of distributed attacks from spacetime network flow patterns, In Proc. 23rd Army Science Conf., 2002.

[3] P. Barford and D. Plonka, Characteristics of Network Traffic Flow Anomalies, In Proceedings of ACM SIGCOMM Internet Measurement Workshop, 2001.

[4] P. Barford, J. Kline, D. Plonka, A. Ron, A Signal Analysis of Network Traffic Anomalies, In Proceedings of ACM SIGCOMM Internet Measurement Workshop, 2002.

[5] R. B. Blažek, H. K., B. Rozovskii, and A. Tartakovsky, A novel approach to detection of "denial-of-service" attacks via adaptive sequential and batch-sequantial change-point detection methods, IEEE Systems, Man and Cybernetics Information Assurance Workshop, June 2001.

[6] J. B. D. Cabrera, L. Lewis, X. Qin, W. Lee, R. K. Prasanth, B. Ravichandran, R. K. Mehra, Proactive Detection of Distributed Denial of Service Attacks using MIB Traffic Variables - A Feasibility Study, In Proceedings of the 7th IFIP/IEEE International Symposium on Integrated Network Management, Seattle, WA, May 2001.

[7] A. Challita, M. El Hassan, S. Maalouf, A. Zouheiry, A Survey of DDoS Defense Mechanisms, Department of Electrical and Computer Engineering American University of Beirut, 2004.

[8] R. K. C. Chang, Defending against Flooding-Based Distributed Denial-of-Service Attacks: A Tutorial, IEEE Communications Magazine, October 2002.

[9] C.-M. Cheng, H.T. Kung, and K.-S. Tan, Use of Spectral Analysis in Defense Against DoS Attacks, In Proceedings of the IEEE GLOBECOM, Taipei, Taiwan, 2002.

[10] K. Cho, R. Kaizaki and A. Kato, Aguri: An Aggregation-based Traffic Profiler, In Proceedings of QofIS2001, 2001.