

Detecting DDoS attacks with passive measurement based heuristics

Christos Siaterlis
csiater@netmode.ntua.gr

Basil Maglaris
maglaris@netmode.ntua.gr

National Technical University of Athens
Network Management and Optimal Design (NETMODE) Lab
Iroon Politechniou 9, Zographou, 157 80 Athens, Greece

Abstract

Network traffic anomalies such as Distributed Denial of Service attacks or the propagation of a new worm are hard to detect on non-congested ISP backbone links. The research community hasn't managed to offer reliable detection metrics that can be implemented with the current technology constraints to network administrators yet. In this work we explore and evaluate the effectiveness of several potential heuristics in detecting flooding attacks. Our observations are based on a daily network traffic analysis for a period longer than 3 months and on more than 40 experiments that were conducted with the use of common DDoS tools in the production network of an academic ISP. The data analyzed are based on different types of passive measurements that are available today to ISP's. We identify multiple effective detection metrics that could give network administrators insight to malicious activities passing through their networks.

1. Introduction

Distributed Denial of Service attacks or more accurately packet flooding attacks, in contrast to logical DoS attacks that exploit certain OS or application vulnerabilities, have received the attention of the networking research community for the past few years. Nevertheless they are still a hard open problem. Network engineers face such unpleasant events more and more often as the DDoS phenomenon has escalated in the 2000-2003 period. Some of the attacks that have reached the mass media and highlight their increasing complexity and their wicked use, are the attacks against anti-spam black-list companies like Osirusoft, against the "AlJazeera" news network and against the 13 root name servers. Worms are also an emerging threat and they are not unrelated to the DDoS problem as they are being used to conquer attack agents [2].

Network administrators expect the research community

to provide useful techniques for detecting and mitigating these problems but until now their weapons are spoofing prevention techniques (like Ingress [10] and RPF filtering [7]), custom detection methods [1] and manually employed countermeasures (firewall filtering, rate limiting [8] or via route blackholes[6]).

In the research arena, most of the state of the art DDoS detection algorithms assume that the detection infrastructure is located near a saturated link in the vicinity of the victim, where the detection is "easy". The tradeoff in this case is that detection algorithms can be simplified but local response is ineffective as the available bandwidth has already been consumed in the upstream path. To couple with this problem, techniques like "IP traceback" [16] or "IP Pushback" [12] aim to find the attack source and potentially move the countermeasures near the sources of the attack. They assume though some sort of automated large scale cooperation and their success in a diverse networked world is doubtful.

In another possible scenario, that has received much less attention, the server under attack belongs to a customer hosted in a well connected ISP that performs DDoS detection on a over-provisioned link. Such underutilized high bandwidth links are a common practice to ISP's backbone networks. In this case attack detection is much harder as link saturation is no longer the identifying anomaly signature. Additionally our sensors have to cope with high data rates which impose constraints in the detection algorithm's complexity. This way, complex processing techniques like power spectral density estimation [4] or clustering algorithms [9] are promising but not readily available. Another aspect of the importance of this case for the security management of ISP's, is that it's preferable to perform DDoS detection at a few points of the over-provisioned backbone and not necessarily on small, congested customer links. It would be economically questionable to expect customers to pay for a dedicated DDoS detection service.

Based on these constraints, we explore the effectiveness of several potential detection metrics that are based on pas-

sive measurements and manage to identify a number of heuristics that can help building reliable DDoS and worm detection mechanisms. We also propose to combine these heuristics with the use of a data fusion algorithm. Our analysis is based on DDoS experiments and a day by day traffic analysis for a period longer than three months. As we described earlier, the link that was monitored could sustain packet floods without severe congestion. This fact made the detection of traffic anomalies challenging and in the same time allowed us to conduct DDoS attacks without causing any harm to legitimate users of the network.

This paper is structured as follows: we begin in section 2 with a brief introduction of the available passive measurement techniques. In section 3, we present the topology and the traffic characteristics of our experimentation platform, an academic-network ISP. In section 4 we continue with an evaluation of several potential DDoS detection metrics. Before we conclude, we will summarize in section 5 the main results of our analysis and discuss future directions.

2. Network monitoring with passive measurements

The methods available today to network administrators to monitor their networks can be categorized in the following types:

1. **Packet capturing.** The most powerful passive measurement method is through direct packet capturing. The main problem of this method is that the monitor has to cope with very high link speeds that impose constraints on the complexity of the statistics that are kept. Specialized hardware like network processors [3] might help keeping up with the increasing data transfer rates in the future.
2. **SNMP based measurements.** This measurement type is very generic as it doesn't actually specify the content or the method of the measurement but the way it is retrieved, mainly through the use of the SNMP protocol. Nevertheless this category includes all the useful measurements that one can obtain from the network devices through the MIB's they implement. These measurements lack detail and are mainly packet or byte counters that are refreshed every few seconds.
3. **Flow based accounting.** A flow is defined as a unique set of the following 5 characteristics <protocol, source IP, source port, destination IP, destination port> and defines a higher level description of a traffic stream. This information that is kept by the routers is much smaller in size than whole packets but in the same time loses some additional information like TCP header flags. Another characteristic of this measurement type

is that it is near real-time, in a sense that a flow is exported to the monitoring station when it has expired [5]. Sampling is some times required because a router can't keep up with the high transfer rates.

3. Experimentation platform

To evaluate the effectiveness and usability of potential DDoS detection heuristics we have performed a series of experiments on an academic research network. As we argued in the introduction, DDoS detection on an over-provisioned high-bandwidth link where traffic is aggregated but stays in low utilization levels holds great interest. In practice, a single hosted network with a fast upstream link had to be monitored. The Gigabit Ethernet link between an academic ISP and a large university was a good candidate. Some information of interest is that this link keeps a sustained rate of 250Mbps with peaks higher than 400Mbps and contains a rich network traffic mix carrying both standard network services like web traffic, but also peer-to-peer application traffic, online games, as well as streaming audio and video traffic. This fact is significant because some detection algorithms might work fine in simulation or lab-testbed experiments, but their high false alarm rate when facing real traffic renders them useless. We conducted more than 40 experiments over several days during business hours and with background traffic generated from the more than 4000 hosts of the university campus. In our experiment scenario the victim was located inside the campus with a 10Mbps link whereas the attacker was outside the campus coming directly from its ISP. The attacker was connected to a 100Mbps interface to simulate the aggregation of traffic from several attacking hosts (Fig. 1). Using well known DDoS tools like Stacheldraht and TFN2K we performed a series of flooding attacks with spoofed IP's and specifically SYN-floods, UDP and ICMP attacks. The attacks used the common method of selecting source addresses from the attacker's real subnet in order to bypass any e-gress or RPF filtering

For presentation purposes we have used 2 characteristic DDoS attack scenarios for all upcoming figures. The first was a TCP SYN attack against a web server which consisted of 2 pulses with duration of 90 seconds and the second one was a UDP attack that lasted one minute. Our comments and the analysis of the performance of various detection metrics are based on our experiments and on the daily analysis of the observed traffic patterns for more than 3 months. In this period the emergence of the MsBlaster and Welchia worms, as well as DDoS attacks against Spamhaus servers [18] took place. During this period of time we had to keep track of our observations and for this purpose we developed a simple threshold-based alarm generation engine. The thresholds for each metric were static during this

3 month period. They were defined based on previous experience in order to maintain a 100% detection rate for all the attacks initiated by us. In the next section, we will present the alarm triggering rate without any quantitative measurement of the false alarm rate because the interpretation of the observed traffic patterns is subjective. Nevertheless we will still make some qualitative statements about the performance of the metrics. In any case, we have to underline the fact that the alarm rate was so small that it was feasible for a security specialist to closely observe each distinct alarm. Once again our approach shows its practical importance for network administrators.

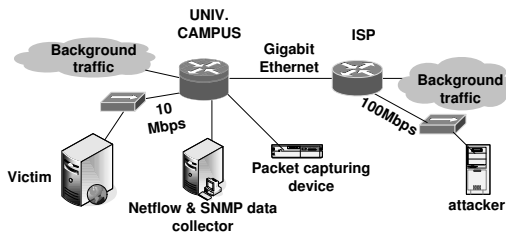


Figure 1. Experiment topology

protocol	packets/sec	Mbps
tcp	37184.8 (92.55%)	204.47 (94.32%)
ftp	1155.3 (2.87%)	8.19 (3.78%)
smtp	168.6 (0.42%)	0.94 (0.43%)
http/s	4011.6 (9.98%)	21.19 (9.78%)
nntp	362.9 (0.90%)	1.76 (0.81%)
p2p	7536.8 (18.75%)	41.53 (19.16%)
other	23911.9 (59.54%)	130.60 (60.25%)
udp	2854.1 (7.10%)	12.25 (5.64%)
dns	180.9 (0.45%)	0.19 (0.09%)
realaud	1312.1 (3.27%)	9.93 (4.58%)
other	1361.10 (3.38%)	2.11 (0.97%)
icmp	111.2 (0.28%)	0.075 (0.03%)
Avg: 216.80Mbps	Stddev:6.53M	Peak: 237.13Mbps

Figure 2. Partial analysis of typical traffic mix on the monitored link during a 30 min time interval

4. Detection Metrics Evaluation

The main result of our study is that some of the commonly used DDoS detection metrics were proved ineffective while some of the modified metrics we propose can drastically serve in the task of identifying traffic anomalies caused even by unsuccessful flooding attempts. We will present the heuristics we evaluated grouped by the measurement methodology.

4.1. Packet capturing

Our packet capturing infrastructure consists of commodity hardware (Intel P4 2.4GHz with an Intel Gig. Ethernet card) and open-source software. We have developed a custom preprocessor plugin for the popular open source

IDS Snort that produces traffic statistics based on captured packet data (libpcap format). The statistics kept were chosen to be simple so that it is feasible to run the plugin at high wire-speeds with minimum packet drops (< 0.1%). Using these tools we are able to collect data of the incoming and outgoing TCP, TCP SYN, TCP FIN, UDP, ICMP packet rates and their corresponding share of the link utilization sampled in regular time intervals. The time granularity is set to 30sec so that we can detect even short-living anomalies. All the data produced by our plugin are stored in round robin databases with the use of the RRDtool [15].

- Symmetry of TCP flows. Due to the nature of the TCP protocol we expect a loose symmetry on the incoming versus outgoing packet rates. This symmetry has already been used as a DDoS detection principle in MULTOPS [11] and D-WARD [14] which use the similar but less effective $\frac{TCP\ in\ packets/sec}{TCP\ out\ packets/sec}$ metric (Fig.7). We propose the use of the ratio

$$\frac{incoming\ SYN\ packets/sec}{outgoing\ FIN\ packets/sec}$$

Our metric is very similar to the metric proposed by [19] which lacked the incoming/outgoing discrimination. This was actually its disadvantage as it could be fooled by an attacker who sends bogus FIN packets. Our metric is resilient to such evasion techniques and in the face of normal traffic, measured on over 1 sec intervals, showed to be a fairly stable metric taking values in the area of 1. SYN attacks are clearly identifiable by this metric as Fig. 3 shows. This metric can obviously recognize only TCP SYN attacks and was very reliable without generating many false alarms. More specifically the total number of distinct alarms during 90 days of operation was only 19 (Fig.8).

- ICMP and UDP attacks are mainly bandwidth consumption attacks and as these traffic types generally utilize small amounts of bandwidth, sudden changes in the transferred ICMP or UDP bytes/sec are good indications of attacks. An improvement over this simple approach is the UDP ratio :

$$\frac{incoming\ bit/sec}{outgoing\ bit/sec}$$

The intuition behind this metric is that although there isn't a clear symmetry in the UDP traffic as in the case of TCP, there is still a fairly stable site dependant behavior (ratio value) depending on the presence of DNS, NFS, streaming servers etc (Fig. 11). Once again the DWARD project [14] uses a similar metric and more specifically UDP, ICMP $\frac{incoming\ packets/sec}{outgoing\ packets/sec}$ that has similar but inferior performance as UDP attacks are

The SYN attack experiment

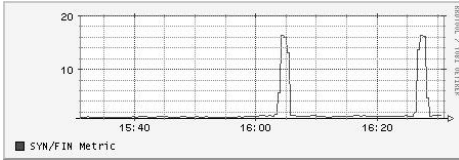


Figure 3. SYN/FIN rate is a good metric.

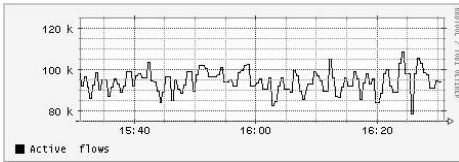


Figure 4. The 'Number of active flows' metric failed to detect the attacks.

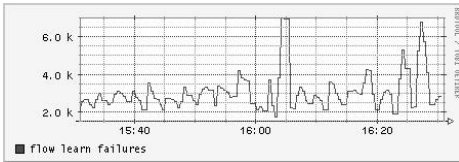


Figure 5. The 'Flow learn failure' heuristic partially detected the attacks.

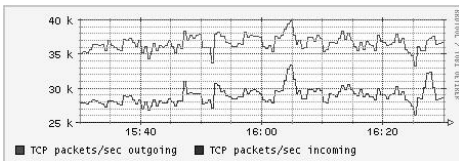


Figure 6. Just the packet rate is not a clear indication of attack.

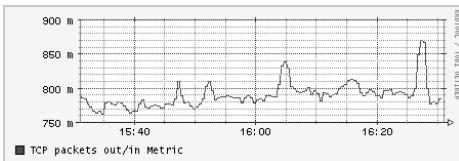


Figure 7. The in/out packet ratio is inferior to the SYN/FIN ratio.

actually aiming at the generation of high bandwidth streams. We have to note here that this metric generated some false alarms and the total number of distinct alarms was 144. These were mainly due to curious packet transfers of peer-to-peer applications but also to DDoS like DNS traffic targeted to our DNS servers that are hosting the Spamhaus blacklist.

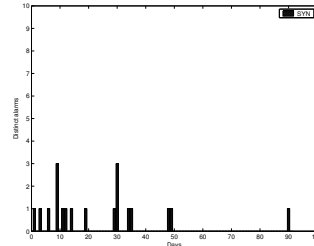


Figure 8. SYN/Fin ratio alarms during 90 days of operation.

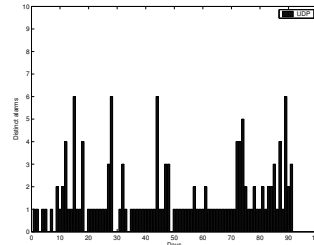


Figure 9. UDP byte ratio alarms.

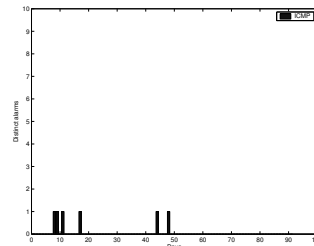


Figure 10. ICMP byte ratio alarms.

- The ICMP $\frac{\text{outgoing bit/sec}}{\text{incoming bit/sec}}$ ratio as we mentioned above serves as an indication of an ICMP flood. In the same time it can help lower the false alarm rate of UDP attack detection because most of the times during a UDP attack a reverse ICMP stream is generated (Fig. 12). In this case our metric is sensitive to two distinct phenomena namely UDP-floods and ICMP-floods. This metric was very stable generating negli-

gible false alarms (only 6 in 90 days) while keeping detection rates high (Fig. 10).

- Persistent failing connection attempts could be a potential DoS anomaly signature. We can argue that human initiated connection attempts are not going to insist if they fail. This is translated as a small rate of RST and ICMP port unreachable packets. Nevertheless this is not the case with the widely used peer-to-peer file sharing clients that often persist on accessing clients that are no longer available. In this case this metric had limited value.

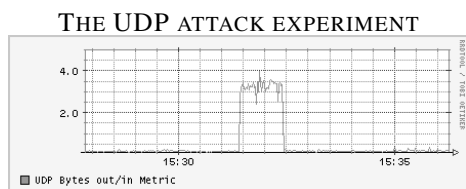


Figure 11. UDP byte ratio is a good heuristic for UDP attacks.

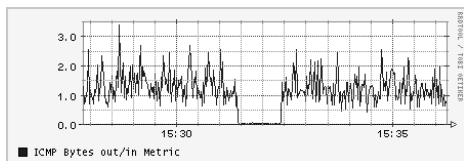


Figure 12. ICMP byte ratio can potentially help detecting a UDP attack.

4.2. SNMP based measurements

For the retrieval of information kept in the router's MIB's we developed a simple SNMP data collector in Perl. Our program uses a polling approach with a period of 30 seconds and stores the acquired data in round robin databases.

- Number of active flows. In the presence of a spoofed attack the number of active flows should rise suddenly (mentioned in [1]) but as we see in Fig. 4 this is not a reliable detection criterion.
- Flow generation rate. We discovered that the number of learning failures of a flow accounting algorithm was able to identify spoofed flooding attempts. The reason is that although the number of flows exhibits a high fluctuation when facing of normal traffic the flows are created and removed from the routers cache in a reasonable time interval. When a flooding attack occurs the amount of 'transports that are not completed'[5]

(for example with TCP FIN or RST) is large, so the entries are not removed gracefully but are filling up the cache causing flow learning failures. A sensor based on this metric cannot discriminate between flows of different protocols and this way can't separate the different attack types. Nevertheless it is a good indication of spoofed attacks.

- Site dependant and policy metrics. Examples of such metrics are: the small UDP and ICMP bit rate in a link, a small rate of fragmented packets or the CPU utilization of legacy routers without distributed processing. A network administrator, knowing by experience the network's behavior, can define a clear policy in terms of upper and lower thresholds of what constitutes normal behavior or not with the additional aid of the sensor's statistic reports. Although these thresholds cannot be very tight to avoid a high rate of false positives due to traffic burstiness, we can use anomalous looking events as hints. One commonly used metric of this type is the packet or bit rate on a link. We could expect a visible ramp-up during an attack. Unfortunately this is not always the case, especially in high bandwidth links where the volume of the aggregated attack stream is still a small percent of the total (Fig. 6).

4.3. Netflow data analysis

To get more detailed information about the flows seen by the router we configured a Netflow collector that gathers and stores the flow entries that are exported by the router. Furthermore we have developed our custom scripts to process this data and calculate our detection metrics.

- Flow length distribution. It would be reasonable to assume that the distribution of the number of packets in a flow (per protocol) would provide a good sign for a spoofed DDoS attack. We would expect a high number of flows with few packets (1-3) as a result of the randomness in the source addresses and ports. Our analysis revealed that this metric is actually sensitive to spoofed DoS attacks but also to port scans¹ and worm propagation. These are the main reasons for its rather erratic behavior(Fig. 14).
- Flow size distribution. Similarly, the distribution of the average packet size in a flow was also very sensitive to port scans producing thus false indications of attacks (Fig. 13). Nevertheless it provides a useful hint for anomalous events. A similar detection approach is adopted by the Panoptis project [13] which calculates a threshold on the average flow size in a time interval.

¹a phenomenon that although it might be considered by some administrators as malicious it is still rather frequent

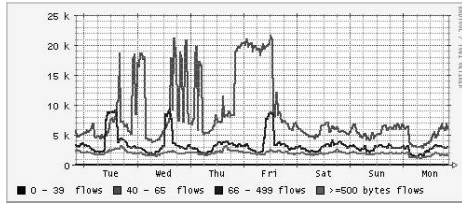


Figure 13. Number of packets distribution of TCP flows has high fluctuations.

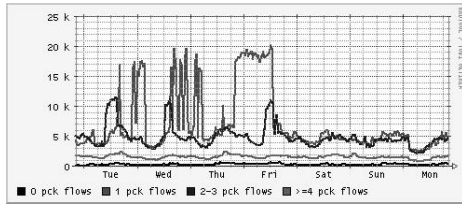


Figure 14. Average packet size distribution of TCP flows has high fluctuations.

5. Discussion

We have reviewed several detection metrics that are proposed in the DDoS literature and were able to suggest improvements while keeping a practical orientation in our research. Our detection metrics are simple and can be implemented without special hardware but are still achieving a small false alarm rate. Beyond these facts, we mentioned several less accurate detection metrics that although they are not definite indications but mere hints, there is a potential gain if we integrate them into a single higher level indication. We propose thus the use of a data fusion algorithm like "Theory of Evidence" as the mathematical framework that will integrate all the proposed heuristics[17].

6. Conclusion

We have explored and evaluated a set of potential DDoS detection metrics that are based on complementary passive measurement methods like packet capturing, SNMP data acquisition and Netflow based traffic monitoring. The evaluation of these metrics was based on experiments in an academic ISP network. If network administrators start using the detection heuristics that were shown to be effective, like the SYN/FIN packet ratio or the Flow learning failures, to detect DDoS attacks reliably we will make a necessary step for automatic response and countermeasure deployment.

References

- [1] P. Barford and D. Plonka. Characteristics of network traffic flow anomalies. In *Proceedings of the 1st ACM SIGCOMM Internet Measurement Workshop*, pages 69–74, New York, Nov. 1–2 2001. ACM Press.
- [2] CERT/CC. CA-2003-20 W32/blaster worm. <http://www.cert.org/advisories/CA-2003-20.html>.
- [3] I. Charitakis, D. Pnevmatikatos, E. Markatos, and K. Anagnostakis. S2I: a tool for automatic rule match compilation for the ixp network processor. In *Proceedings of SCOPES 2003*, Vienna, September 2003.
- [4] C.-M. Cheng, H. Kung, and K.-S. Tan. Use of spectral analysis in defense against DoS attacks. In *Proceedings of IEEE GLOBECOM, 2002*.
- [5] CISCO. Netflow. <http://www.cisco.com/go/netflow>.
- [6] CISCO. Remote triggered black hole filtering. <ftp://ftp-eng.cisco.com/cons/isp/security/>.
- [7] CISCO. Unicast reverse path forwarding enhancements for the ISP-ISP edge. <ftp://ftp-eng.cisco.com/cons/isp/security/URPF-ISP.pdf>.
- [8] CISCO. Using CAR during DoS attacks. http://www.cisco.com/warp/public/63/car_rate_limit_icmp.html.
- [9] C. Estan, S. Savage, and G. Varghese. Automatically inferring patterns of resource consumption in network traffic. In *Proceedings of the ACM SIGCOMM Conference*, Karlsruhe, Germany, August 2003.
- [10] Ferguson and Senie. RFC2827 network ingress filtering: Defeating denial of service attacks which employ IP source address spoofing, May 2000.
- [11] T. M. Gil and M. Poletto. MULTOPS: A data-structure for bandwidth attack detection. In *USENIX*, editor, *Proceedings of the 10th USENIX Security Symposium, August 13–17, 2001, Washington, DC, USA*.
- [12] J. Ioannidis and S. M. Bellovin. Implementing pushback: Router-based defense against DDoS attacks. In *Proceedings of NDSS*, San Diego, February 2002. The Internet Society.
- [13] C. Kotsokalis, D. Kalogeras, and B. Maglaris. Router-based detection of DoS and DDoS attacks, June 2001. 8th Workshop of the HP OpenView University Association.
- [14] J. Mirkovic, G. Prier, and P. Reiher. Attacking DDoS at the source. In *Proceedings of ICNP 2002*, pages 312–321, Paris, France, November 2002.
- [15] T. Oetiker. About RRDtool. <http://people.ee.ethz.ch/~oetiker/webtools/rrdtool>.
- [16] S. Savage, D. Wetherall, A. Karlin, and T. Anderson. Practical network support for IP traceback. In *Proceedings of the 2000 ACM SIGCOMM Conference*.
- [17] C. Siaterlis and B. Maglaris. Towards multisensor data fusion for DoS detection. In *Proceedings of ACM SAC'04*, Nicosia, Cyprus, 2004.
- [18] Spamhaus. Virus and DDoS attacks on spamhaus. <http://www.spamhaus.org/cyberattacks/>.
- [19] H. Wang, D. Zhang, and K. G. Shin. Detecting SYN flooding attacks. In *Proceedings of the INFOCOM 2002*, volume 3, pages 1530–1539, Piscataway, NJ, USA, June 23–27 2002. IEEE Computer Society.