Detection and Honeypot Based Redirection to Counter DDoS Attacks in ISP Domain

Anjali Sardana, Krishan Kumar, R. C. Joshi
Indian Institute of Technology, Roorkee
{anjlsdec, kksaldec , joshifcc}@iitr.ernet.in

Abstract

The inherent vulnerabilities in TCP/IP architecture give dearth of opportunities to DDoS attackers. The array of schemes proposed for detection of these attacks in real time is either targeted towards low rate attacks or high bandwidth attacks. Presence of low rate attacks leads to graceful degradation of QoS in the network thus making them further undetectable. In this paper, we propose a scheme that uses three lines of defense. The first line of defense is towards detecting the presence of low rate as well as high bandwidth attacks based on entropy variations in small time windows. The second line of defense identifies and tags attack flows in real time. The last line of defense is redirecting the attack flows to honeypot server that responds in contained manner to the attack flows, thus providing deterrence and maintaining QoS at ISP level. We validate the effectiveness of the approach with simulation in ns-2 on a Linux platform.

1. Introduction

Denial-of-Service (DoS) is an intentional attempt by attacker to compromise availability of a service to legitimate users [1]. Distributed Denial-of-Service attacks (DDoS) degrade or completely disrupt services to legitimate users by eating up communication, computational and memory resources of the target through sheer volume of packets. DDoS attacks are amplified form of DOS attacks where attackers direct hundreds or even thousands of compromised “zombie” hosts against a single target [2]. As per survey conducted by FBI/CSI in 2004 [3], these attacks are second most dreadful attacks in terms of revenue losses after information thefts.

The two most basic types of DDoS attacks are bandwidth attacks and application attacks. Bandwidth attacks consume resources such as network bandwidth by overwhelming it with a high volume of packets. Application attacks use the expected behavior of protocols such as TCP and HTTP to the attacker’s advantage by tying up computational resources and preventing them from processing transactions or requests.

In many cases when a sustained high bandwidth attack reaches servers it is not possible to contain the attack at border gateway as the offending packets have already consumed the finite bandwidth available on the connection to the ISP. In this case, having a good relationship and clear communication channels with ISP are essential. High bandwidth attacks will have an impact on the ISP’s network. Since they are closer to the source of the attack they are in a better position to filter the offending traffic. However low rate attacks are critical component and remain undetected until the network functionality becomes unstable thus targeting QoS.

Current DoS research focuses on three tracks: (1) mitigation techniques: which aim to mitigate the effect of DoS attacks assuming that we can not accurately distinguish illegitimate packets; (2) classification-based techniques: which move the classification process away from the server, either to achieve simpler classification criteria or to replicate the classification process, making it less vulnerable to attacks; and (3) attack-tracking techniques: which try to identify DoS attack sources, to stop them and discourage attackers.

The proposed scheme works at ISP level and serves on three lines of defense to protect a public domain server. Firstly, sample entropy variations at a point of presence (POP) identify the presence of attack [4]. Secondly the flows are tagged as attacks in subsequent time windows. Lastly, we propose a honeypot based redirection that can retain a connection with the attacker along with providing deterrence from public domain server, thus giving a desirable QoS even in presence of attack. Connection retention is to obtain information about the attackers by logging their actions.

This paper will focus on the simulation modeling and analysis of proposed scheme. The organization of the paper is the following. First, the related work is presented in Section 2. Section 3 describes the approach for detection, characterization and redirection. Simulation experiments are explained in

0-7695-2876-7/07 $25.00 © 2007 IEEE
DOI 10.1109/IAS.2007.23

IEEE computer society

Section 4. Section 5 shows the results and analysis. Last, the conclusion and the future work are presented in Section 6.

2. Related Work

Existing DDoS solutions are classified into four broad categories in [2]: Prevention, Detection and characterization, Traceback, and Tolerance and mitigation. Prevention is a mechanism which stops the attacks before they are actually launched. The approaches to stop IP spoofing [5], filtering malicious IP addresses based on experience [6], and repairing security holes by patches [7] falls under this category. The process of identifying that a network or server is under attack after launch of the attack is called detection. Detection can be passive, proactive, and on-line. Characterization means differentiating attack packets from legitimate packets. Signature based [8] techniques can detect attacks launched using known DDoS attack tools, however to detect novel attacks anomaly based techniques [9-11] are currently in practice. Traceback aims to locate the actual attack sources regardless of the spoofed source IP addresses used by attackers. Traceback solutions includes controlled flooding [12], overlay network [13], ICMP messages [14], and IP packet marking [15] [16] based techniques. Finally, Tolerance and mitigation aims to eliminate or curtail the effects of an attack and try to maximize the quality of services under attack.

A commonly used detection approach is either signature-based [8] or anomaly-based [9-11]. By contrast, an anomaly based detection system observes the normal network behavior and watches for any divergence from the normal profile. Due to the diversity of user behaviors and the emergence of new network applications, it is difficult to obtain a general and robust model for describing the normal traffic behaviors. As a result, legitimate traffic can be classified as attack traffic (false positive) and attacker traffic is classified as legitimate (false negative). To minimize the false positive/negative rate, a larger number of parameters are used to provide more accurate normal profiles. However, with the increase of the number of parameters, the computational overhead to detect attack increases. Though schemes in [9-11], have been successful in isolating large traffic changes (such as bandwidth flooding attacks), but slow rate, isotropic attacks can not be detected and characterized because these attacks do not cause detectable disruptions in traffic volume. In [17] a generic framework has been proposed to mitigate DoS attacks. We demonstrate the utility of a more sophisticated treatment of DDoS anomalies, as events that alter the distribution of traffic features. For low as well as high rate attacks, traffic distributions have appreciable deviation from normal to provide signs of DDoS attack. The characterization of attack traffic is done by first choosing suspicious flows based on volume and then monitoring acks sent by the server for these flows.

These schemes focus only on detection of attack whereas the real culprits i.e. attack sources are not trapped. The ideal solution for DDoS defense aims at mitigating attack at source. As characterization of attack sources is very difficult to achieve ideally so a feasible solution is to redirect traffic from suspected attack sources that can work in efficient manner to tolerate the attack. We also aim to sustain the services of our protected server under high bandwidth DDoS attacks using tolerance based scheme , however in our work we have complemented tolerance and mitigation by dynamically rate limiting incoming traffic at edges based on share of traffic per edge router, and per flow.

3. Approach

Figure 1 shows the procedural flowchart of the approach.

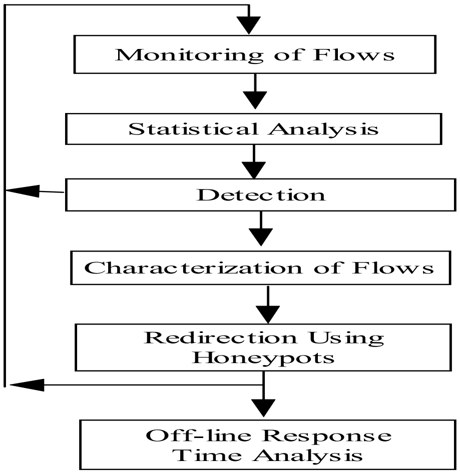
Monitoring of Flows
Statistical Analysis
Detection
Characterization of Flows
Redirection Using Honeypots
Off-line Response Time Analysis

Figure 1: The procedural flowchart of the approach

3.1. Detection

Detecting DDoS attacks involves first knowing normal behavior of our system and then to find deviations from that behavior. The normal profile or behavior is obtained using entropy $H(X)$ as a parameter to measure traffic feature distributions. The packets for each flow are monitored in a time window. As in IPv4 packets, there is no flow ID header information, so we designate different flow IDs to a unique 4-tuple SourceIP, SourcePort, DestinationIP, DestinationPort encountered in incoming packet. The traffic destined to our server and not the complete traffic is monitored at link through which POP is connected to the server to backbone. POP collects information in a time window and calculates sample entropy $H(X)$.

A metric that captures the degree of dispersal or concentration of a distribution is sample entropy. We start with frequency distribution $X = \{n_i, i = 1,......, N\}$ that feature i occurs n_i times in the sample. Then the sample entropy $H(X)$ is

$$H(X) = -\sum_{i=1}^{N}(p_i) \times \log_2(p_i) \quad (1)$$

Where $p_i = n_i / S$ and $S = \sum_{i=1}^{N} \mathrm{n_i}$

The value of sample entropy lies in the range $0 - \log_2 N$. The metric takes on the value 0 when the distribution is maximally concentrated, *i.e.,* all observations are the same. Sample entropy takes on the value $\log_2 N$ when the distribution is maximally dispersed, *i.e.* $n_1 = n_2 = ...n_n$. We have done time series analysis of traffic at POP P_s connected to protected server. Consider a random process $\{X(t), t = j\Delta, j \in N\}$ running at P_s, where Δ a constant time interval is called time window, N is the set of positive integers, and for each $t, X(t)$ is a random variable. Here $X(t)$ represents the number of packet arrivals for a flow in $t-\Delta, t$. The entropy can be computed as above.

It is found in our simulation without attack that Entropy $H(X)$ value varies within very narrow limits after slow start phase is over. This variation becomes narrower if we increase Δ i.e. monitoring period. We take average of $H(X)$ and designate that as normal entropy $H_n(X)$. The basic idea is to remove small scale perturbations by averaging over slightly longer-intervals of time. However it is also desirable that the window duration should not exceed a limit. By this way, normal profile of traffic in terms of entropy is obtained by our approach. To detect the attack, the entropy $H_c(X)$ is calculated in shorter time window Δ. We assume that the system is under attack at time t_a, which means that all attacking sources start emitting continuously, whenever there is appreciable deviation from $H_n(X)$, attack is said to be detected. The network is in normal state for time $t < t_a$ and turns into attacked state in time t_a. Let t_d denote our estimate on t_a. At time t_d following event triggers

$$(H_c(X) > (H_n(X) + a \times d)) \cup$$
$$(H_c(X) < (H_n(X) - a \times d)) \quad (2)$$

Here $a \in I$ where I is set of integers and d is deviation threshold. Tolerance factor a is a design parameter and d is absolute maximum deviation in Entropy $H(X)$ from average value $H_n(X)$ while profiling for network without attack.

3.2. Characterization

In detection phase if $H_c(X)$ is more than normal $H_n(X)$ then suspected malicious flows tend to have lower frequency values of packet arrivals and the attack is termed as low rate degradation attack. While if $H_c(X)$ is less than normal $H_n(X)$ then suspected malicious flows have high values of number of packet arrivals and the attack is coined as high rate. Once the attack is launched at POP P_s, we have aggregate of attack flows and normal flows. Let F represent set of active flows. Then

$$F = F_n \cup F_a (F_n \cap F_a = \phi) \quad (3)$$

Where F_n represent actual normal flows and F_a is set of actual attack flows. Our main task in this module is to find $F_a^* = \{f_1, f_2, ...f_m\} \subset F$

the set of m malicious flows. Ideally,

$$(F_a^* \cap F_a = F_a) AND (F_a^* \cap F_n = \phi)$$

Now the main problem is to find m :-

- as for low rate attacks, m number of least measured packet arrival flows constitute F_a^*.
- and for high rate attacks, m number of highest measured packet arrival flows form F_a^*.

To answer these questions if we can find Φ_a, the expected total attack traffic then from following equation, we can find m and F_a^*.

$$\sum_{j=1}^{m} X_i^j (t_d + \Delta) \le \Phi_a \quad (4)$$

Where i is designated flow, j varying from 1 to m for least or highest measured packet arrivals, and $X(t_d + \Delta)$ represent packet arrivals for flow i in next time window after attack is detected. The expected value Φ_a, is calculated as $\Phi_a = \Phi_{td} - \Phi_n$, where Φ_{td} the total traffic is received in $\{t_d - \Delta, t_d\}$

and Φ_n is averaged total traffic till $t_d - \Delta$ from the time bottleneck link utilization is 1.

The set F_a^* got through this process is further pruned by omitting flows from F which have been active at time $t_d - \Delta$ since we assume that all attack flows start at the same time.

3.3. Redirection using Honeypots

The flow is tagged as attack or legitimate flow in the above step. Instead of just dropping the attack flows enroute or resetting sessions they are actively redirected from hostile sources to a honeypot. Honeypot server responds to attack flows in exactly same manner as would the actual server to legitimate clients. Production server and a honeypot both are with the same IP. Since the connection with suspected flows is retained, the flows that we tag as suspicious can be treated again as normal flow if the entropy measures come in the limit of legitimate flows. Those flows can be transferred back to actual server in subsequent time window. This reduces false negatives. However if persistently in many time windows the flow remain suspicious connection remains directed at honeypot server. Also one can potentially gain more information about the attacker.

4. Simulation Experiments

Simulation is performed using ns2 [18] network simulator.

4.1. Topology

GT-ITM [19] topology generator is used to create our simulation topology. We have represented transit domain routers as POPs of the ISP and stub domains as customer domains attached to POPs as shown in fig. 2. There are four ISP domains with two peers each i.e. two other ISP domains are directly attached at POPs. We have represented transit domain routers as POPs of the ISP and stub domains as customer domains attached to POPs as shown in fig. 2. There are four ISP domains with two peers each i.e. two other ISP domains are directly attached at POPs.

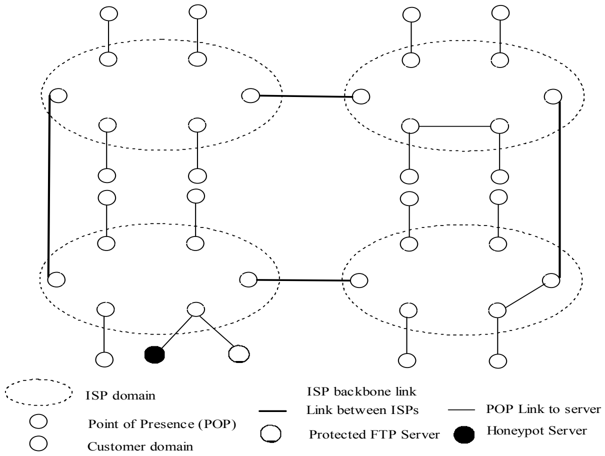

Figure 2. A short scale simulation topology

4.1. Basic parameters of simulation

Table 1 provides the basic parameters set for simulation.

Table 1. Basic Parameters of Simulation

S.No.	Parameter	Value
1.	Simulation Time	60 seconds
2..	Number of legal sources	100 per ISP domain Total 4*100=400
3.	No. of attackers	1-25 per ISP domain. Total= 1-100
4.	Access bandwidth for legitimate customers	1Mbps
5.	Bottleneck Bandwidth	310Mbps
6.	Mean attacker rate	0.1-1.0Mbps (low rate) 2.7-3.7Mbps (high rate)
7..	Attack duration	20-50 seconds

4.2. Attack detection parameters

Table 2 Attack Detection Parameter

S.No.	Parameter	Value
1.	Window Size	.2 seconds
2.	Tolerance factor a for entropy deviation	3-10

Simulations are carried at different values of tolerance factor a for different attack strengths.

5. Results and Discussion

Various aspects are discussed in this section:

5.1. Threshold setting

We conducted simulation experiments for finding out threshold for entropy under normal condition as per simulation parameters given in previous section. The normal range of entropy by using frequency distribution of number of packets per flow ID (SourceIP, SourcePort, DestinationIP, and DestinationPort) in time windows of 0.2 seconds is shown in fig. 3. Simulation is also carried by taking longer window of 1.0 second. Deviations are still lesser as expected however average is almost same.

It is found that once the utilization of bottleneck link is 100%. Entropy value also lies in small range as depicted in fig. 2.

Maximum absolute deviation from average (d):- 0.03393

The server as per its capacity planning and normal profile of legitimate clients in terms of request bytes normally have an estimate of maximum number of clients to be served at any instant of time. On these bases though we have bottleneck of 310Mbps but still on the higher side for better link utilization we assume to serve up to 400 legitimate clients with maximum 1Mbps (average 0.8Mbps) request bytes per client.

Though our work is simulation based, but on actual network for profiling purpose this kind of experiments can be conducted to find $H_n(X)$ and d.

5.2. Detection of attack

As soon as any event in "2" triggers, attack is said to have occurred. Figure 2 shows entropy profile when our network is put under low rate attack. In this case attack is launched with 100 attackers with mean rate 0.3Mbps per attacker. Clearly in first time window after attack is launched at 20 seconds, there is jump in entropy value. The positive jump and persistent high value as compared to normal reflects that it is a low rate attack and the flows which are causing this anomaly have comparatively lesser frequency than already existing ones. In case of high rate attacks, entropy value tends to be lower than normal. In our simulation using total attack strength of 300Mbps with 100 attackers, the Entropy variation is reflected in figure 3.

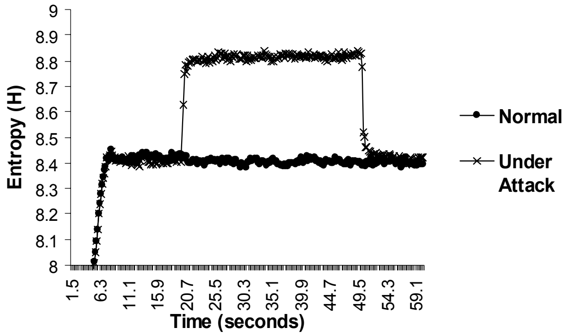

Figure 2: Entropy for low rate DDoS attack

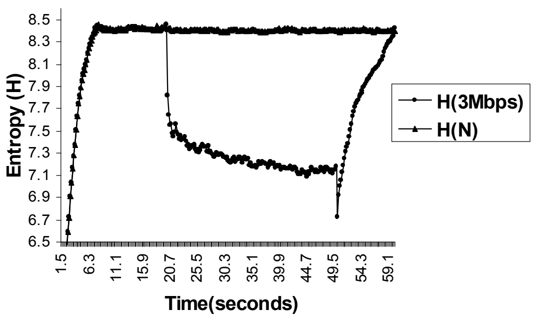

Figure 3: Entropy for high rate DDoS attack

However initially it can rise but with proper adjustment of window and start time, the same can also be lumped. In this case, the flows which have comparatively higher share of packets are reasons of anomaly. Similar trends exist for high rate attacks at different attack strengths with variation only in deviation from normal value.

5.3. Variation in response time due to honeypot redirection

In figure 4, the average response time increases with increase in attack load for a fixed client load. There steep rise in average response time after initial stability is because of the fact that as the aggregate load (client load and attack load) become greater than bottleneck bandwidth of 310 Mbps, the legitimate packets are dropped at POP. Repeated attempt for file transfer from legitimate clients leads to a sharp increase in average response time. The figure 4 also shows an edge over previous case due to presence of Honeypot server in same IP as FTP server. Although the average response time increases with increase in attack load, the absolute average response time is less.

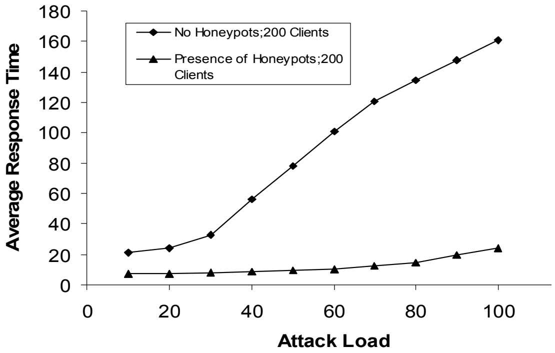

Figure 4. Comparison of variation in average response time with attack load for a given client load (200 Clients) with and without honeypots.

This is due to the fact that suspected attack flows are directed towards Honeypot server and FTP server remains free of attack, thus giving average response time that maintains QoS.

6. Conclusions and future work

Our detection scheme is able to identify low rate and high rate DDoS attacks reliably in ISP domain. The appreciable change in entropy indicates that traffic flow distributions are an effective way of detecting DDoS attacks. The increased entropy value signs low rate attack whereas dip in level of entropy marks high rate attack. The characterization of flows is done in subsequent time windows giving a real time solution. The flows tagged as suspicious are redirected to honeypots. The proposed scheme greatly reduces the response time for legitimate users in the presence of an attack. Hence the scheme has the potential to defend against DDoS attacks along with maintaining a desirable QOS. Ns-2 has been used as a testbed for validation of the scheme.

The characterization of flows is volume based wherein sophisticated attacker may be able to emulate the legitimate traffic. To overcome this, signal processing based techniques can be used in conjunction with the proposed scheme. Secondly, the overhead of state monitoring pose a DoS attack in itself. Future focus is to distribute these overheads amongst multiple ingress POPs of ISP and validate the scheme using goodput and NPSR as evaluation parameters.

References

[1] R.K.C. Chang, “Defending against Flooding-Based Distributed Denial-of-Service Attacks: A Tutorial,” *IEEE Communication Magazine,* 2002.

[2] J. Mirkovic, and P. Reiher, “A Taxonomy of DDoS Attack and DDoS defense Mechanisms,” *ACM SIGCOMM Computer Communications Review*, Volume 34, Number April 2004.

[3] Computer Crime Research Center. “2004 CSI/FBI Computer Crime and Security Survey,” Available at: http://www.crime- research.org/news/11.06.2004/423/.

[4] K. Kumar, R. C. Joshi, and K. Singh, “Detecting Low Rate Degrading and High Bandwidth Disruptive DDoS Attacks in ISP Domain,“ *In Proceedings of ISCF-2006*, pp. 83-88, December, 2006.

[5] P. Ferguson, D. Senie, “Network ingress filtering: Defeating denial of ser-vice attacks which employ IP source address spoofing,” RFC 2267, the Internet Engineering Task Force (IETF), 1998.

[6] T. Peng, C. Leckie, and K. Ramamohanarao,“Protection from Distributed Denial of Service attack using history-based IP filtering,” *In Proceedings of ICC 2003*, USA, 2003.

[7] X. Geng, and A.B. Whinston, “Defeating Distributed Denial of Service attacks,” *IEEE IT Professional* ,pp 36–42, 2002.

[8] M. Roesch, “Snort—Lightweight Intrusion Detection for Networks,” *In Proceedings of LISA ’99*, 1999.

[9] T. M. Gil, and M. Poletto, “Multops: a data-structure for bandwidth attack detection," *In Proceedings of the 10th USENIX Security Symposium*, 2001.

[10] R. B. Blazek, H. Kim, B. Rozovskii, and A. Tartakovsky, “A novel approach to detection of denial-of-service attacks via adaptive sequential and batch sequential change-point detection methods," *In Proceedings of IEEE Systems, Man and Cybernetics Information Assurance Workshop*, 2001.

[11] J. Mirkovic, G. Prier, and P. Reiher, “Attacking DDoS at the source,” *In Proceedings of ICNP 2002, Paris, France*, pp. 312–321, 2002.

[12] H. Burch, and B. Cheswick, “Tracing anonymous packets to their approximate source,” In Proceedings of 2000 USENIX LISA Conference, pp.319–327, 2000.

[13] R. Stone, “CenterTrack: An IP overlay network for tracking DoS floods,” In Proceedings of 2000 USENIX Security Symposium, pp. 199–212, July 2000.

[14] S. Bellovin, “The ICMP traceback message*,” IETF Internet Draft,* 2000.

[15] D. Dean, M. Franklin, and A. Stubblefield, “An algebraic approach to IP traceback," ACM Transactions on Information and System Security 5(2), 119-137, 2002.

[16] U.K. Tupakula, and V. Varadharajan, “A practical method to counteract Denial of Service Attacks, “ Proceedings of the 26th Australian Computer Conference in Research and Practice in Information Technology, ACM International Conference Proceeding Series, pp. 204–275, 2003.

[17] A. Sardana , B. Gandhi and R. C. Joshi, “A Novel Framework for Characterization, Source Identification and Mitigation of DoS Attacks, ” In Proceedings of ISCF-2006, pp. 99-108, December, 2006.

[18] NS Documentation Available: http://www.isi.edu/nsnam/ns

[19] GT-ITM Traffic Generator Documentation and tool http://www.cc.gatech.edu/fac/EllenLegura/graphs.html