Statistical Segregation Method to Minimize the False Detections During DDoS Attacks

J. Udhayan[1] and T. Hamsapriya[2]

(Corresponding author: J. Udhayan)

Department of Mathematics and Computer Applications, PSG College of Technology
Coimbatore, India (Email: udhayangodwin@gmail.com)[1]
Department of Information Technology, PSG College of Technology, Coimbatore, India[2]

(Received Nov. 18, 2009; revised and accepted Apr. 29, 2010)

Abstract

DDoS attack aims at occupying the victim resources so as to defy the legitimate requests from reaching it. Even though the attack traffic is generated in intimidating measures, the attack traffic mostly is disguised as the genuine traffic. Hence most of the mitigation methods cannot segregate the legitimate flows from the attack flows accurately. As the result, legitimate flows have also been filtered while appeasing the DDoS flood. In this paper a statistical segregation method (SSM) has been introduced, which samples the flow in consecutive intervals and then the samples are compared against the attack state condition and sorted with the mean as the parameter, then the correlation analysis is performed to segregate attack flows from the legitimate flows. SSM is compared against various other methods and the blend of segregation methods are identified for alleviating the false detections effectively.

Keywords: Botnet, DDoS, false negative, false positive, zombie

1 Introduction

The DDoS attack is performed to deplete the resources of one or more victim servers and make their resources unavailable to their legitimate clients. It involves hurling of chunk packets from many zombies over the victim server until it goes down [5, 7]. Thus DDoS attacks can bring mission-critical systems and business operations to halt, resulting in loss of revenue opportunities, damage to the reputation etc [2, 5].

Backbone of this kind of attack is Botnet or network of zombies. Though zombies can be termed as secondary victims [16], they are not the target of DDoS attack; anyhow they are compromised by the attacker to be his accomplice without their knowledge. Hence identifying zombies will help in blocking or dropping of malicious flood caused by it. Thus identifying zombies is essential to withstand imminent DDoS attacks which may involve millions of zombies [5].

However in most of the cases, the attacker exploits the lack of authentication in the IP protocol by spoofing the source address and then the vulnerabilities in the protocols like TCP, UDP, ICMP, HTTP [9, 12, 18] to espouse the DDoS attack. However all this applications relies on the IP protocol to transmit the packets over the Internet [1, 14]. So IP level solution for DDoS maximizes the potential to confront various versions of DDoS attacks.

2 Background

Denial of Service (DoS) attacks has been around for more than a decade. In the past, such attacks have traditionally been launched from a single host or subnet, so the target system could usually detect the attack and defend it [5]. However the most recent Distributed Denial of Service (DDoS) attacks are far more destructive and harder to defend, because they are launched from many sources (zombies) simultaneously [2].

The first well-documented DDoS attack occurred in 1999, when the Trinoo attack tool was deployed from at least 227 systems [3]. It took the server of "University of Minnesota" out of service for over two days.

DDoS attacks are also becoming far more pervasive. While one might think these destructive assaults are infrequent and targets only the giant enterprises such as highly publicized assaults against Amazon, E*Trade, e-Bay, Yahoo!, and Microsoft. A recent UCSD study estimates the number of DDoS attacks exceeds 2,000 per week, and they have assaulted all sizes and types of firms.

When manipulated effectively, such attacks can severely affect the target servers and massively disrupt the business processes thus inflict millions of dollars in lost revenue.

Mindless of the amount of damage it can cause, the procedure to maneuver DDoS attack however remains simple. Attacker performs reconnaissance from network to

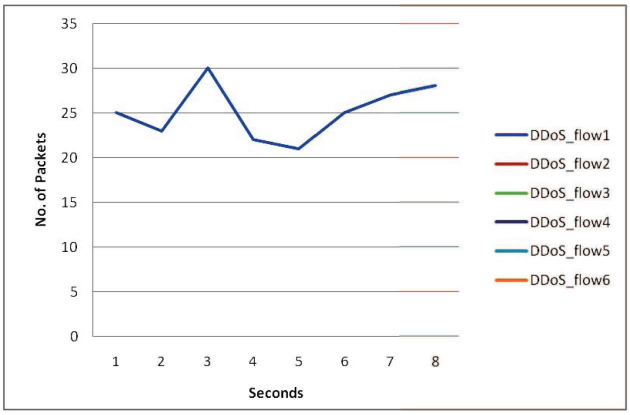

Figure 1: Similar behavior among low rate DDoS flows

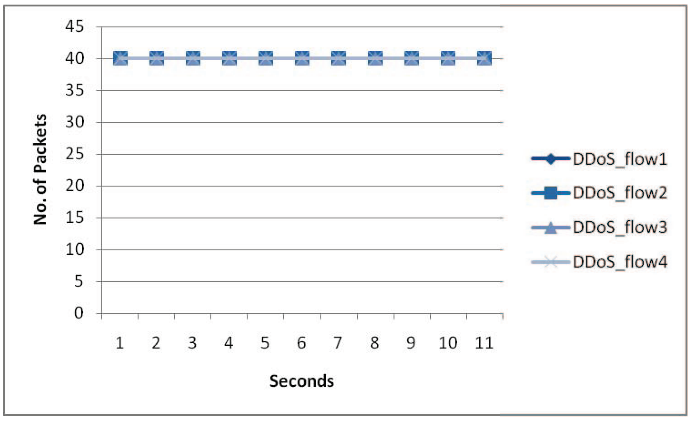

Figure 2: Constant rate or steady rate DDoS attack

network to find out the vulnerable computers. Then compromises the vulnerable computers and brought them under his control. There are lots of handy tools to do this. Once the attacker feels like that he has enough number of zombies under his belt, he can coordinate them to charge the flood against a target server in a specified time.

3 DDoS Rate Categorization

Latest DDoS agent (bot) that resides in the zombies can generate low rate, constant rate, fluctuating rate and increasing rate floods as commanded by the attacker [9]. While analyzing the real-time dataset obtained from the CAIDA, all this types of attacks have been recognized.

3.1 Low Rate Attack

This kind of mechanism masquerades the flood as a normal (legitimate) flow throughout the network, since the packets are generated to imitate the behavior of the genuine client so as to avoid the detection. The traffic never floods the bandwidth throughout the network but when it reaches the victim, because of the fact that thousands of coordinated computers involve in generating the flow which will eventually overload the victim till stalemate. This low rate flow disguise itself as the normal flow, this will annoy the detections and segregation mechanisms as well. This attack cannot be mitigated without eliminating moderate amount of genuine flows because it always maintains the rate between less than the normal rate to slightly over normal rate. While analyzing through the dataset obtained from the CAIDA a following pattern emerged as the low rate DDoS attack.

The flip-side of this approach is that the amount of zombie recruitment made for this attack is huge which is not cost-effective. However the increased usage of Internet among people from various walks of life has created more chance for the attackers.

3.2 Constant Rate Attack

The majority of earlier attacks deployed the constant rate flooding mechanism. Since the attacker commands the zombies to generate same number of packet for every interval. This therefore generates steady traffic with the rate greater than the legitimate traffic. This increased rate creates sudden packet flood to disrupt the victim's services so quickly. This therefore is the best cost-effective approach to the attacker, since he can deploy a minimal number of agents to inflict severe damage. The CAIDA dataset also exhibits few trails of constant rate attack as shown in Figure 2.

However the rate in the midway may drop a bit due to congestion or loss in the traffic because this attack is semi bandwidth attack and this doesn't aim to flood the bandwidth but at the same time it doesn't mind if it overloads the bandwidth. However this attack can be segregated from the normal flow because the rate at which the packet can be generated is always above the normal rate.

Bandwidth based attacks cause congestion throughout the network. Therefore they are used rarely because of the ubiquitous deployment of packet filters [4] which can detect and discard the flood in the network itself before reaching the source. As the result modern attackers don't prefer to use this type of attack. However this kind of attack is still at use because not all the novice attackers pay attention to the rate.

3.3 Increasing Rate Attack

The rate of this flood keeps on increasing gradually staring from the lowest possible rate, thus delays the early detection. This attack aims to cripple the victim server bit slowly than constant rate attack by taking its time. However this mechanism exhibits steady increase in the rate, this is quiet unusual with the legitimate case. So the rate based detection is substantial for detecting this kind of attacks. The following is a trace of increasing rate attack from CAIDA dataset.

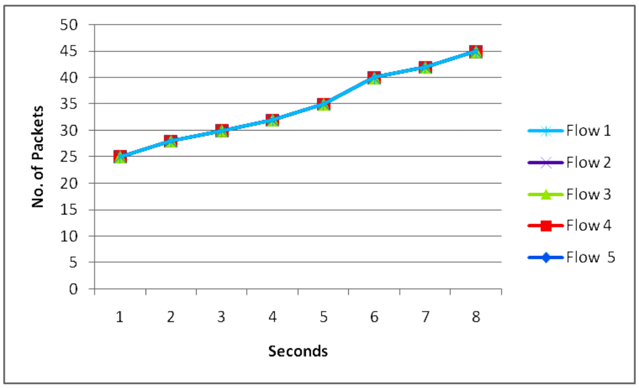

Figure 3: Increasing rate DDoS flows

3.4 Intermittent Rate Attack

This kind of attack varies the rate quiet often and breaks for every constant or varying interval so as to avoid detection. However this kind of attack deploys any of the above mentioned rates. This separation in the flood makes it less efficient because most of the times this makes the flood insufficient to clog up the server from processing the legitimate requests. No such traces were found in CAIDA dataset.

3.5 Observations

Through analyzing various CAIDA dataset, we found out that the attackers who own the small sized botnet or little number of zombies command increasing rate or constant rate flood. Nevertheless if the flow rate is not maintained, the victim server will find enough space to traceback the attacker. If the rate is increased then even after detecting the attack the victim can't handle the huge data and it will be pushed to stalemate state.

While constituting zombies the attacker not only chooses the zombies from the external network but also from the internal network, because the internal network is always offered with huge bandwidth than the transit bandwidth.

4 Pros and Cons of Various Detections

Early detections, like proactive or anomaly detection detects the attack at its initial stage [6]. Even though they detect the attack they are not good segregators of attack flows; thus end up producing false positives and false negatives. Therefore the solutions in counteracting the DDoS attack are not fully reliable.

Existing solution that looks to detect and segregate the carrier IP address (attack source) from the normal is D-Ward [10, 11, 13] and MULTOPS [14]. Those methods work based on the principle that during a DDoS attack, the TCP traffic ratio is higher than usual because the number of outgoing packets is greater than the number of incoming packets. Since it is a TCP based detection technique it is not applicable to other protocols.

D-word and its versions collects sample at various intervals, with these methods the interval needs to be varied hugely it may at times exceed the scope of defense.

Moreover the ICMP and UDP are easily exploitable to maneuver DDoS flood and as these traffic types generally utilize small amount of bandwidth, sudden change in the amount of transferred ICMP or UDP byte/sec are good indication of attacks. This change increases the packet rate to formidable rate. This increased in rate is not properly utilized by the existing segregation methods.

The problem with the widely deployed signature-based IDS is that, it cannot detect new attacks [15]. While the anomaly-based IDS [6] can catch new attack patterns, its accuracy is a concern. It may flag a new non-attack activity as intrusion, resulting in a false positive. In general, IDS is notorious for its enormous resource consumption because it requires deep packet inspection and flow state maintenance.

5 Proposed Statistical Segregation Method

Determining a threshold and behavior among the flows to distinguish legitimate traffic from attack traffic is a solution to avoid large number of false-positives which indeed remains as a challenge.

5.1 Sampling Method

Initially to detect DDoS attacks there are lot of methods available which are proactive and reactive methods [9, 16]. Among those methods one simple method to detect TCP hosted DDoS attack at the earliest is to check incoming traffic against outgoing traffic which varies massively than the normal [10]. With this method even the transit routers can detect the DDoS attack. For other DDoS attacks sudden increase in traffic, similarity in behavior among flows etc are best indicators.

If the preliminary detection of attack is positive then the sampling method is invoked. Sampling method instantaneously assigns a separate rate counter for each IP address.

Rate counter is designed to collect n number of samples, where *sample* is the set of all incoming packets per second.

However to have effective detection, the time between collecting the samples must be deterministic and less. Moreover collecting more samples always increases the accuracy but at the same time, it consumes more time and introduces processing overhead. To reduce the interval of detection to make the sampling quick and more effective, Rate counters are invoked only '3' times, each with one second interval. Therefore only three samples per source

IP address is collected. The outcome is a sample set, as follows:

$$Sampleset = \{Sample1, Sample2, Sample3\}.$$

5.2 Rate Analysis

After collecting the samples they are assigned to a statistical based segregation mechanism. The three samples are then compared, therefore 3 × 3 combinations are possible. Before comparing the samples the normal rate of the genuine client must be understood. For example normal client can generate two echo requests using ICMP [17] likewise any genuine client has an inbuilt rate limit. If that exceeds, then the flow can be a flood carrier. However outcome will be of any of the following possibilities.

Attack State.
if |normal_rate|< |*Sample1*|< |*Sample2* |<|*Sample3*|
then there may be Increasing rate DDoS attack
if |normal_rate|< |*Sample1*|= |*Sample2*|=|*Sample3*|then there may be a constant rate DDoS attack.

Inconclusive States.
All the other cases are considered as inconclusive state because the flow may be a legitimate one or may still carry DDoS attack.

Unfortunately segregation is not as easy as it seems because of the following two reasons.

1) By being lost some packets in the network the attack traffic does not satisfies the attack state condition. Which we call as disproportionate attack.

2) Legitimate traffic may appear like attack if its arrival rate is more.

Nevertheless the careful comparison using mean and standard deviation will help segregating the attack traffic from the normal. The mean, Standard Deviation (S.D) for the samples respective to every flow is calculated as follows.

$$\overline{\mu} = \overline{X} = \frac{1}{3}\sum_{i=1}^{3} X_i \tag{1}$$

where X_i is the number of packets in the i^{th} sample, $i = 1, 2, 3$.

$$\sigma = \sqrt{\frac{1}{3}\sum_{i=1}^{3}(X_i - \overline{X})^{2}} \tag{2}$$

Literally for constant rate attack Standard Deviation remains around zero and one. In case of the Increasing rate attack, the Attack state condition holds true. However in case of low rate attack and disproportionate attack neither the attack state condition nor the mean and Standard Deviation helps to segregate it, hence the following correlation analysis is appointed.

5.3 Correlation Analysis

As explained in Section 3.1, Low rate attack flow from the zombies exhibit likeness among themselves which is rarely possible with the genuine flow. This can be found when correlating the flows. Therefore covariance and the correlation are applied to segregate the genuine flow from low rate flow through analyzing for the similarity among the flow.

Covariance is a measure of how much two random variables change together. Therefore to correlate among the flows, for the flow x the flow with the closest mean slightly less or equal is considered as next variable y. The formula is expressed as

$$Cov(x, y) = \frac{\sum_{i=1}^{3}\left(Xi - \overline{X}\right)\left(Y_i - \overline{Y}\right)}{N} \tag{3}$$

where Y_i is the number of packets in the i^{th} sample ($i =$ 1, 2, 3) for flow y, where $\overline{Y}$ is the mean of samples for flow y.

Like standard deviation, when calculating covariance the range of values is infinite.

The relationship between correlation and covariance is simply represented by the following formula:

$$Correl\,(x,\ y) = \frac{Cov(x,\ y)}{\sigma_x \sigma_y} \tag{4}$$

Correlation helps to identify the similarity among the flows thus facilitate to segregate the low rate attack and the disproportionate attack.

6 Implementation Details

Most of the modern routers nowadays have Network processors integrated with packet filters [4], thus offers multiprocessing; programmability and also offers multi fields classification benefits. This kind of routers uses ABV (Aggregate Bit vector algorithm) [8], which uses bitmap intersection scheme to classify the traffic under its corresponding field, in our case the field is Source IP address.

6.1 Sampling Method Implementation

The advantage of using ABV are speed search, flexibility in specifying rules, classify the packets with less processing overhead etc.

Therefore the modern routers qualify our segregation requirement with provisions to assign and set the following rules:

Rule 1. Perform Initial detection using any of the existing detection technique.

Rule 2. Record the rate of the traffic under it's corresponding Source IP address. Adaptive packet filtering feature is more helpful to set the following rules.

Rule 3. If detection result is positive then Invoke Rate Counter.

```
Algorithm1: Packet Counter
For each (Source IP)
{
Record: the rate/sec for three consecutive odd or even Intervals as {Sample1, Sample2, Sample3}
}
```

The sampling for the entire range of incoming IP addresses is done concurrently. So the outcome, the sample sets will also emerge instantaneously.

6.2 SSM Implementation

Once the flow is sampled, the mean and standard deviation are immediately calculated. Then the flow is sorted using the Insertion sort to organize the flows in a descending order with mean as the primary key before recording into the database. Hence the Flows with greater means come in the top of the order. Though the flows are sorted using mean, additional Flow parameters including {Flow No, Sample1, Sample2, Sample3, Mean, and Standard Deviation} are also stored in the successive fields. Usually the sorting and recording into the database has to be done online to make the segregation effective, therefore the Insertion sort algorithm has been used.

The greatest of mean is considered as the attack, if it satisfies the attack state condition as given in sec 5.2. Flows with succeeding means can be segregated as attack as long as the attack state condition holds true. While traversing top-down when the attack condition fails and the inconclusive state heaps in, the correlation procedure is summoned.

The oscillation in the mean can happen due to the packet loss. This oscillation only causes little variation in the correlation results. It can be better noticed if we follow the records top-down. While traversing top-down, if the correlation of the successive record varies, if they are the Normal traffic then they can be identified with the huge variation in Mean. Therefore after applying insertion sort and attack state verification, applying the correlation will help to identify the similarity among the successive records. Thus segregation algorithm SSM could segregate the carrier IP addresses from the normal IP addresses.

While the sorting is on, the sources whichever is confirmed as the carrier IP will be blocked to mitigate the flood.

```
Algorithm: Statistical Segregation Method (SSM)

Input: A non-empty list of sorted set of numbers.
Output: Attacking IP addresses.
Procedure(Source IP, Sample1, Sample2, Sample3)
{
Calculate Mean and S.D and D.R using Eqn (1), (2), and (3)
Step1: Sort the resultant database records {Flow No,Sample1, Sample2, Sample3, Mean, and Standard Deviation} using Insertion Sort for each mean in descending order
//Step2:Pick the first column and maximum Mean
for (i = 1; j=1; i= length[A]-1; i++)
{
tempj = A[j]; tempi =A[i]; Max = tempj.FirstElement;
CompSD =tempj.LastElement; Mean = tempi.FirstElement; Sample1 =Mean.NextElement;
Sample2=Sample1.NextElement;
Sample3=Sample2.NextElement;
SD =empi.LastElement;
//Check for Increasing rate attack
//Step3: Check for increasing rate or constant rate attack
if (Mean = Max && SD =CompSD &&
normal rate< Sample1< Sample2 < Sample3or
normal rate <Sample1= Sample2=Sample3 is True)
{ Block(Source IP);}
//Attack estimation, only in case of erroneous network or congested network the following loop is applicable
elseif(Max – Packetlossrate = Mean = Max)
{ Block (Source IP);}}
else
{
Store: Flow [Flow No, Sample1, Sample2, Sample3, Mean,SD]
Call Correlation (Flow)
j=i+1; } }

Correlation (Flow)
{Flow No = i;
for ( i =0; i< n; x = i+1; y = i+2)
{
//Calculate Cov(x, y) and Correl(x, y)using (3) and (4)
if (correlation >90%)
Call Block (Source IP)
}

Block (Source IP)
{// We assume B as the blocking control
//B = 0 means do not block and B = 1 means Block the corresponding IP address
if (Source IP = active ) then
set B = 1;
else
set B = 0;
}
```

7 Experimental Results & Analysis

Generally testing the DDoS attacks in a realistic environment is extremely difficult to achieve. Therefore for this research, we used our experimental network with few nodes to mimic the large scale attack. Moreover a dataset obtained from the CAIDA has helped in adjusting the

Table 1: Flow segregation output for TFN DDoS

Source IP	Flow No	Sample			Mean()	State	Correlation (%)
		1	2	3		C.R	
X.X.1.12	1	750	750	750	750	C.R	Corr(3, 4) = 99.1%
X.X.1.18	2	750	750	750	750	I.S	
X.X.1.20	3	750	749	750	749.6	I.S	
X.X.1.22	4	749	750	748	749	I.R	
X.X.1.30	5	700	720	740	720	I.R	
X.X.1.31	6	700	719	738	719	I.R	
X.X.1.10	7	699	718	737	718	I.R	
X.X.1.15	8	695	713	730	707	I.R	
X.X.2.6	9	688	712	720	706.6	I.R	
X.X.2.8	10	544	525	500	523	I.S	
X.X.2.14	11	543	342	572	485.6	I.S	

C.R - Constant Rate, I.R - Increasing Rate, I.S - Inconclusive State, N.A - No Attack.

segregation algorithm through scrutinizing various flows with varying rate.

7.1 DDoS Scenario

The SSE lab has been used to generate various DDoS attacks and to analyze the attacks using various detection techniques. Especially the DDoS attacks were constituted using ICMP, UDP and TCP protocols, because of the space constrain we do not produce all the results but only the most relevant results.

We used JPCAP library to write a java program for recording three samples at three different intervals all the samples are saved in a vector instead of array, and the mean and S.D for those samples are calculated simultaneously. We also implemented D.R as in D-WARD using the same JPCAP library. Moreover we used Wireshark to monitor and analyze the traffic and also to load the TCP dump.

To complicate the scenario we maneuvered batch attack one batch generates attack traffic with constant rate another batch generates traffic with varying rate. Totally two batches are involved, each includes 20 computers. We confine to minimum computers because we did not wanted to down the server. Even with this number of computers the packet loss occurred. After collecting the samples by implementing Algorithm 1, the mean and standard deviation are calculated and the result is presented in Table 1 for descending order of mean by implementing Algorithm 2.

Table 1 does not produce all the same but only diverse result. Around 12 computers are classified under C.R, and the remaining 8 computers fall under the (749 to 750) mean range. However the attack State helped in determining the high mean as the attack carrier because it have confined to the C.R clause.

The mean 720 starts new set because the attack state now is I.R. For Increasing rate attack even though it suffers severe data loss the result of the samples from all the 20 computers maintained the same state I.R.

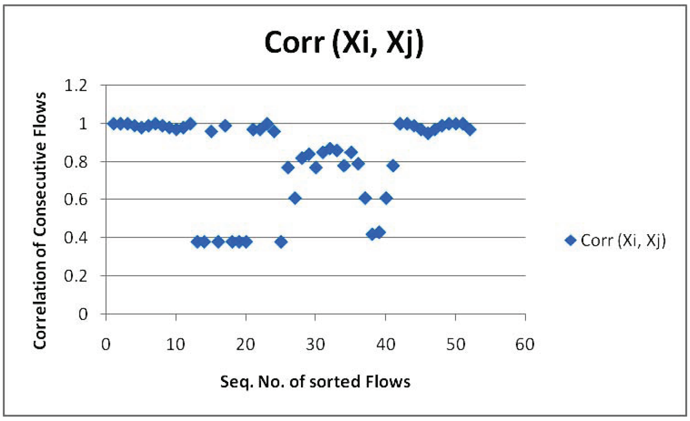

Figure 4: Flow correlation results

7.2 Low Rate Flood and Disproportionate Flood Segregation

The CAIDA dataset has been used to analyze the correlation over various types of flows.

Since the insertion sort has been used earlier to arrange the flows in descending order of mean. Effective segregation over low rate attack can be produced through correlating the successive flows as shown in Figure 4.

When the correlation results for the sorted flows are obtained even though the results remained five, however the attack flow produces very close results when correlating with each other, however when correlated the attack flow with human generated flows considerable variation is identified. Moreover correlation among the human generated flows also varied considerably.

The correlation results come within the range of 0.9 to 1 is considered as the attack. Considering the traffic which comes within this range and up in the order of the

Table 2: Important attributes in evaluating attack

No.	**Attributes**
1	IP protocol type values
2	packet size
3	server port numbers
4	source/destination IP prefixes
5	Time-to-Live(TTL) values
6	IP/TCP header length
7	TCP flag patterns
8	IP/TCP/UDP checksums

table helps to minimize the false positive.

Thus we explained the possibility of segregating the carrier IP addresses with the help of rate based statistical analysis detection technique.

8 Comparison Analysis

8.1 Existing Solution for Segregation

Generally, the attack signature of the DDoS attacks can be acquired using the network monitoring capability of the IDS. Current IDS have the capability to produce traffic statistics based on captured packet data. However most of the DDoS detection systems normally use any or all of the following attributes [13] as signature to identify the attack. The important attributes in evaluating attack are shown in Table 2.

These attributes are collected by the IDS through deep packet inspection at the cost of excessive memory and computation so these samples are always available for analysis. Consider this attributes as *Ai*.

However, computing arbitrary fingerprints might require excessive memory and computation. Several other metrics have been proposed by the research community to overcome this. One of the very successful and widely used metrics is: The ratio of TCP traffic between the two directions. Due to the nature of the TCP protocol we expect a loose symmetry on the incoming versus outgoing packet rates. This principle has been used by local detection mechanism like D-WARD.

Deciding Rate (D.R) as in D-WARD therefore is calculated for each flow using the following equation.

$$D.R = \left(\frac{No.of\ Incoming\ Packet}{No.of\ Outgoing\ Packet} \right) per\ second. \quad (5)$$

Whenever there is low amount of outgoing packets against incoming packet for a flow is evident that flow is considered as the attack flow. Where D.R is only applicable to TCP traffic, if it is applied for UDP and ICMP traffic it has no effect.

Moreover most of the DDoS attacks are performed from the botnet. Normally the Botmaster onsets the DDoS attack by giving a command to the entire botnet, e.g. [udp |IP address |Port number |Number of Packets |Option]. The zombies that are online at that time will generate the traffic as mentioned in the command. As a result this will exhibit the similarity among flow characteristics [16] like packets per flow (ppf), bytes per packet (bpp), bytes per second (bps), and packets per second (pps). For aggregated flows, characteristics include: flows per address (fpa) and flows per hour (fph).

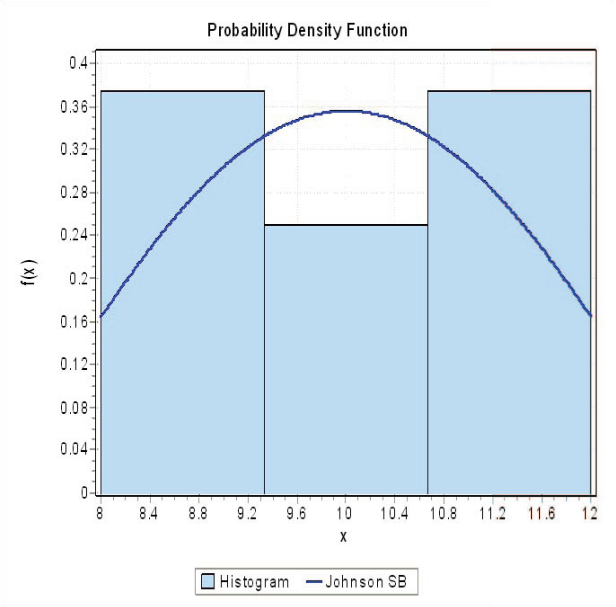

Figure 5: False positive concentration for Ai based detection on TCP DDoS flows

8.2 Analyzing SSM with The Existing Solutions

To identify the false detection for general detection mechanisms, Metrics like D.R, Ai and SSM has been applied to the dataset. The dataset contains attacks like TFN (with TCP and UDP) and Smurf attacks. The detection mechanisms generated false positives and false negatives while trying to segregate the carrier flow.

However SSM is also a flow based segregation method since it relies on the rate of the traffic, hence it is compared against the flow based metrics separately and with the other metrics separately.

The result is estimated in percentage and the result is tabulated in Table 3.

The result does not favor any single detection technique therefore we integrated the techniques. As the result the effective segregation procedures are identified for TCP, ICMP and UDP protocols as shown in Table 4.

9 Conclusion and Future Work

Only way to mitigate the imminent DDoS attack is through blocking the considerable number of zombies

Table 3: Performance of various DDoS segregation techniques

Protocol	**False Detections by applying Ai (%)/sec**		**False Detection by applying Ai& D.R (%)/sec**		**False Detection with Segregation & Ai (%)/sec**		**False Detection with with Segregation & D.R & Ai (%)/sec**	
	F.P	F.N	F.P	F.N	F.P	F.N	F.P	F.N
TCP	10.12,	3.07	6.8,	2.2	7.0	1.1	1.2	0.43
UDP	9.1	3.23	N.E	N.E	1.18	0.63	N.E	N.E
ICMP	8.1	3.2	N.E	N.E	1.83	0.52	N.E	N.E

F.P - False Positive, F.N - False Negative, N.E - No Effect.

Table 4: The effective segregation procedures are identified for TCP, ICMP and UDP protocols

Existing Segregation for TCP	**Effective Segregation**
Start Ai Correlation D.R. Analysis Action *End*	*Start* Ai Correlation D.R Analysis SSM Segregation Action *End*
Existing Segregation for UDP	**Effective Segregation**
Start Ai Correlation Rate analysis Action *End*	*Start* Ai Correlation Rate analysis SSM Segregation Action *End*
Existing Segregation for ICMP	**Effective Segregation**
Start Ai Correlation Rate analysis Action *End*	*Start* Ai Correlation Rate analysis SSM Segregation Action *End*

from the Internet, to do so a proper segregation method is essential to avoid blocking the legitimate clients. Therefore a statistical segregation method to segregate the attack traffic from the normal flow to reduce false detections is introduced in this paper. This algorithm can be integrated with the existing detection for the better use.

In future, Cross-correlation based passive analysis will be designed to investigate the many-to-one relationship against the DDoS flows which will help to detect the multiple trails of a single attacker. Moreover, an artificial neural network technique is used to automate the segregation method and to limit the manual intervention.

References

[1] A. Akella, A. Bharambe, M. Reiter, and S. Seshan, "Detecting DDoS attacks on ISP networks," *Proceedings of the Twenty-Second ACM SIGMOD/PODS Workshop on Management and Processing of Data Streams*, pp. 1-3, 2003.

[2] A. T. Bhaskar, B. N. Kamath, and S. D. Moitra, "Hybrid model for network security systems: Integrating intrusion detection system with survivability," *International Journal of Network Security*, vol. 7, no. 2, pp. 249-260, Sep. 2008.

[3] S. Bosworth, and M. E. Kabay, *Computer Security Handbook,* Wiley Publications, Edition 4, 2009.

[4] A. Habib, M. M. Hefeeda, and B. K. Bhargava, "Detecting service violations and DoS attacks," *Proceedings of the Network and Distributed System Security Symposium (NDSS)*, pp. 1-13, 2003.

[5] Y. Huang, X. Geng, and A. B. Whinston. "Defeating DDoS attacks by fixing the incentive chain," *ACM Transactions on Internet Technology*, vol. 7, no. 1, pp. 62-69, Feb. 2007.

[6] K. Hwang, H. Liu, and Y. Chen, "Cooperative Anomaly and Intrusion Detection for Alert Correlation in Networked Computing Systems," *IEEE IPDPS-2005*, 2005.

[7] R. R. Kompella, S. Singh, G. Varghese, "On scalable attack detection in the network," *IEEE/ACM Transactions on Networking*, vol. 15, no. 1, pp. 14-25, Feb 2007.

[8] K. Kumar, R. C. Joshi, and K. Singh, "An integrated approach for defending against distributed denial-of-service (DDoS) attacks," *Proceedings of IRISS*, pp. 1-6, 2006.

[9] J. Mirkovic, J. Martin, and P. Reiher, "Taxonomy of DDoS attack and DDoS defense mechanisms." *ACM SIGCOMM Computer Communication Review*, vol. 34, pp. 39-53, Apr. 2004.

[10] J. Mirkovic, and P. Reiher, "D-WARD: A source-end defense against flooding denial-of-service attacks," *IEEE Transactions Dependable and Secure Computing*, vol. 2, no. 3, pp. 216-232, Sep. 2005.

[11] J. Mirkovic, E. Arikan, S. Wei, S. Fahmy, R. Thomas, and P. Reiher, "Benchmarks for Ddos Defense Evaluation," *Proceedings of the DETER Community Workshop on Cyber Security Experimentation*, pp. 1-10, June 2006.

[12] D. Moore, C. Shannon, D. Brown, G. M. Voelker, and S. Savage, "Inferring Internet denial-of-service activity," *Proceedings of USENIX Security Symposium*, pp. 1-14, 2001.

[13] G. Oikonomou, J. Mirkovic, P. Reiher, and M. Robinson, "A framework for a collaborative DDoS defense," *Proceedings of the Annual Computer Security Applications Conference (ACSAS 22)*, pp. 33-42, Dec. 2006.

[14] T. Peng, C. Leckie, and K. Ramamohanarao, "Detecting reflector attacks by sharing beliefs," *Proceedings of the 46th IEEE Global Telecommunications Conference (GLOBECOM'03)*, pp. 1358-1362, Dec. 2003.

[15] M. Salour, and X. Su, "Dynamic two-layer signature-based IDS with unequal databases," *International Conference on Information Technology (ITNG'07)*, pp. 77-82, 2007.

[16] S. M. Specht, and Ruby B. Lee, "Distributed denial of service: Taxonomies of attacks, tools and countermeasures," *Proceedings of the 17th International Conference on Parallel and Distributed Computing Systems*, pp. 543-550, Sep. 2004.

[17] J. Udhayan, A. R. Demystifying, and R. Limiting, "ICMP hosted DoS/DDoS flooding attacks with attack productivity analysis," *IEEE International Advance Computing Conference (IACC 2009)*, pp. 558-564, Mar. 2009.

[18] H. Wang, C. Jin, and K. G. Shin, "Defense against spoofed IP traffic using hop-count filtering," *IEEE/ACM Transactions on Networking,* vol. 15, no. 1, pp. 40-53, Feb. 2007.

J. Udhayan is a Research Associate in Smart and Secure Environment (SSE), Department of Mathematics and Computer Applications in PSG College of Technology, Coimbatore, India. His research interest includes Information Security, Network security, Wireless Senor Networks, Artificial Intelligence and Cloud Computing. He has presented articles in 7 National/ International Conferences.

T. Hamsapriya received her PhD Degree from Anna University, India. She is working as a Professor in Department of Information Technology, PSG College of Technology. She has published many papers in International and National journals, She also has 15+ years of teaching experience. Her areas of interest include Parallel and Distributed Computing, Network Security and Open Source Systems.