# Chapter 3

# A Light Weight DDoS Attack Detection Mechanism

In this chapter, I present a low cost yet robust DDoS detection method to detect different types of DDoS attacks. The method attempts to detect DDoS attack by monitoring the deviation of the count of unique source IPs and the count of source IPs whose transmission rate is higher than a given threshold value. Unlike other similar existing methods, the proposed method does not need to maintain a list of source IPs which makes the detection method faster. Another advantage of the proposed method is the ability to detect DDoS attacks performed by small size botnet. A non-parametric change point modeling technique is used to identify the changes in the observed parameters in near real time.

## 3.1 Introduction

In a distributed denial of service (DDoS) attack, one or more group of compromised users send legitimate traffic to the victim to degrade or even shut down the service of the victim. The goal of such a DDoS attack typically varies from creating simple inconvenience to the user of a website to incur major financial losses to the on-line service providers. A DDoS attack usually generates a huge volume of TCP/IP packets from a large number of sources. These attack packets are generally indistinguishable from that of normal traffic packets. Thus, when all these attack packets merge at the victim site, they occupy most of the victim's network bandwidth and forces the victim to degrade its service or at worst shut down its services temporarily. One of the key challenge for DDoS attack defenders is to detect the attack as near real time as possible so that the victim gets enough time to take appropriate mitigation steps such as traffic diverting and resource allocation.

One obvious way taken by most researchers[69][19][71] is to monitor the volume of traffic that are received by the victim site. However, such methods are not robust against the bursty nature of Internet traffic. In case of the bursty nature of internet traffic such methods often identify it as an attack. On the other hand, such bursty traffic may actually be attack traffic, and a delayed decision may turn out to be very risky for the victim. Another important feature which can be monitored to detect a DDoS attack is source IP of the incoming packets. Since a DDoS attack is highly distributed, the number of source IPs involved in the attack is much larger than the number of source IPs under normal condition. Peng et al [72] used the arrival rate of new source IP addresses in the traffic during each observation period. However, this approach needs to maintain a database of trustworthy source IP addresses, which might itself be vulnerable to attack.

In this chapter, I present an effective detection scheme called Violating Source IP Count (VSC), which monitors the number of unique source IP addresses during each observation period. Our assumption is that during an attack, the number of source IP will increase abruptly, and by detecting this change we can detect the attack. Also, if the attacker attempts to launch a DDoS attack with less number of sources, the rate of transmission from each source must be high to achieve a bandwidth attack. We monitor the count of sources which transmits above a threshold. Under a less distributive attack the mean value of this count deviates significantly, which indicates the presence of an attack. To detect changes in our observed features we adopted the non-parametric CUSUM[18] approach and applied it by following the idea of Wang et al[5]. To evaluate our detection mechanism we perform several experiments on different network traces and are presented in section 3.4. The experimental results indicate that VSC has a short detection time and a high accuracy rate. The complexity of this method is very low both in terms of space and time which makes the proposed method deployable in a distributed manner in the first mile as well as in the intermediate routers to detect the attack in the beginning stage itself.Our contribution in this work is a simple and fast approach to detect DDoS attacks by monitoring the deviation of (i)the count of the unique source IPs from its mean value and (ii)the count of the sources transmitting at a high rate.

The rest of the chapter is organized as follows. Section 3.2 presents the proposed method along with the necessary theory. Section 3.3 presents the results of performing our method on various network traces. Finally, conclusions are drawn in section 3.4.

## 3.2 Violating Source IP Count(VSC)- a Light Weight DDoS Detection Mechanism

For a DDOS attack, the attacker's main goal is to overwhelm the server by sending illegitimate network traffic, using different protocol. One common characteristic of DDoS attack is that the volume of traffic (number of packets) during an attack is very high. To generate high volume of traffic the attacker either has to use a large botnet consisting of lot of compromised machines or the attacker may send traffic from a small botnet but at very high speed. We are making the assumption that under normal condition the deviation of the number of unique source IP addresses from its mean value is bounded by an upper bound. And also during normal condition the deviation of the number of source IP addresses sending packets above a threshold, referred to as violating source IP, from its mean value is also bounded by an upper bound. Based on these two assumptions we present a detection mechanism called violating Source IP count (VSC) to detect a distributed denial of service (DDoS). The key features of VSC are highlighted below.

1. Researchers have already used source IP addresses as detection feature such as in Peng et al[72]. However, we use the count of unique source IP addresses, in an observation period, as detection feature. This approach does not require to maintain a database of trustworthy IP addresses for its operation. Thus the memory requirement and speed of this algorithm is comparatively better, which is a key goal for a detection system. This feature makes VSC very suitable to be used in a distributed manner.

2. A DDoS attack may either use a small size bot sending traffic at a high speed or a large size bot consisting of many zombies. VSC monitors changes in both the number of source IP addresses and count of sources transmitting at high speed. Thus VSC traps the attacker from both the directions, hence reducing the scope of the attacker.

### 3.2.1 Overview of Violating Source IP Count

VSC attempts to detect the presence of attack by monitoring two features of the traffic, namely the number of unique source IP addresses and the number of violating source IP addresses. VSC collects the incoming packets during every observation period, say $\delta T$ and inserts the packets into a binary search tree based on their source IP address. Each node in the tree has a count field that specifies the number of packets from the source IP represented by the source IP field of the node, as illustrated in figure 3.1

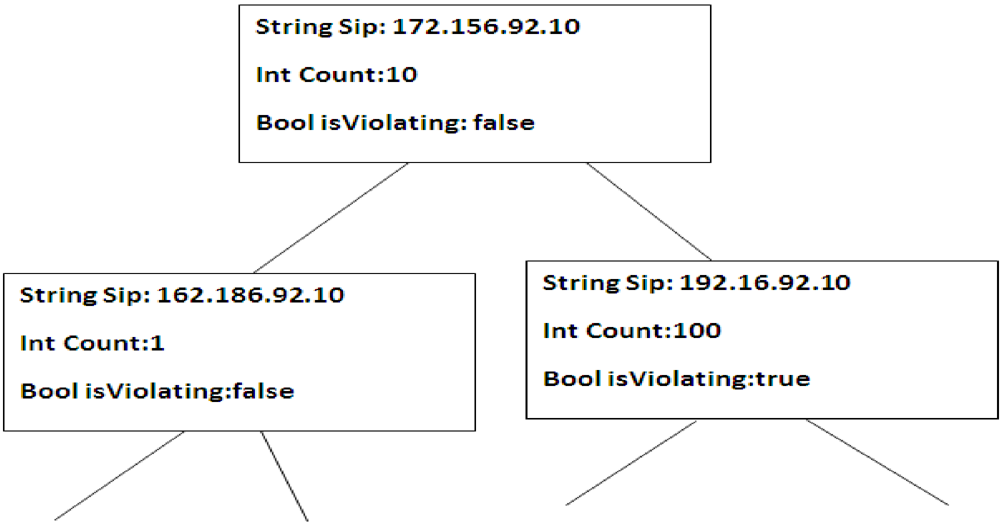


Figure 3.1: The binary search tree used in the VSC

If the number of packets for a source IP address is greater than a certain threshold, that IP address is marked as violating IP. This information is used by VSC to detect DDoS attack that uses a small size botnet to carry out the attack. Also, at the end of each observation period the number of nodes in the binary search tree gives the number of unique source IP addresses during the observation period. Thus at the end of each observation period the BST (Binary Search Tree) gives us a) the number of violating source IP addresses $V_i$, and b) the number of unique source IP addresses $X_i$ in the current observation period $t_i$.

#### 3.3.1.1 System Architecture

Figure 3.2 provides an overview of VSC mechanism. The VSC mechanism consists of three basic components, viz,

detection engine, decision engine, and response engine. The detection engine processes the incoming traffic to detect any attack. The task of the decision engine

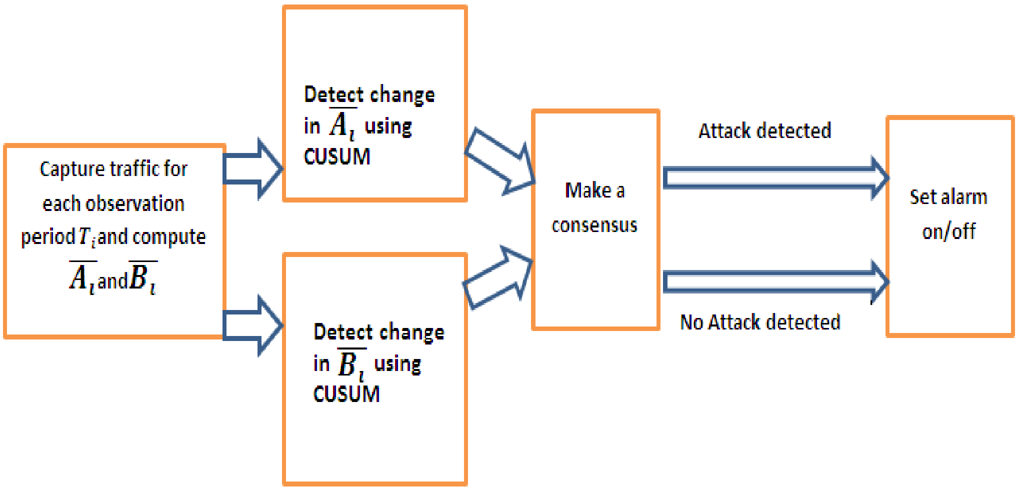

Capture traffic for each observation period $T_i$ and compute $\overline{A_i}$ and $\overline{B_i}$
Detect change in $\overline{A_i}$ using CUSUM
Detect change in $\overline{B_i}$ using CUSUM
Make a consensus
Attack detected
No Attack detected
Set alarm on/off

Figure 3.2: Architecture of VSC

is to combine the results from the detection engine and to reach a consensus about the occurrence of an attack. The response engine in turn sets an alarm on or off based on the output of the decision engine.

#### 3.3.1.2 Placement of the Detection Mechanism

The VSC can be deployed in different locations in a network as shown in Figure 3.3

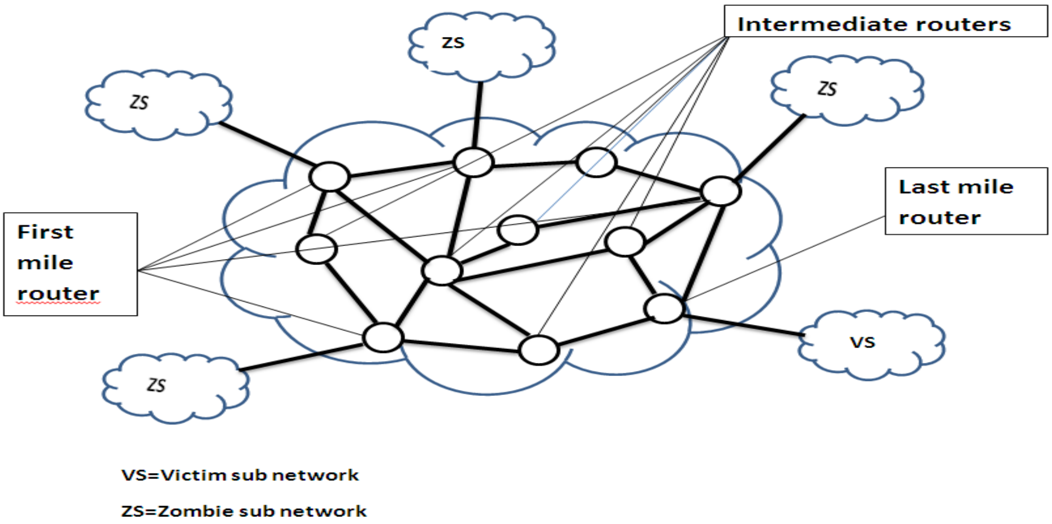


Figure 3.3: Different possible placement locations of VSC

If it is deployed in the first mile router, the detection rate will depend on the size of the bot. If a small bot is used then the number of packets from one IP must be high, which our detection engine can detect by estimating the change in the count of violating source IP count. However, in case of a large bot, the detection rate might decline. If deployed in intermediate routers, the detection rate increases as it moves towards the victim site. One can thus deploy VSC at different intermediate routers and detect the occurrence of an attack in a distributed manner. Chen et al [81] describe such an approach in their work. Another point of deployment is at the victim site, i.e, the last mile outer. Since all the attack traffic aggregate in the last mile router, deviation of $\overline{A}_n$ and/or $\overline{B}_n$ can easily be detected, if there is any. Among these different detection points the intermediate routers and the last mile router carries the greatest interest from the point of view of the victim. In this paper we present the experimental results performed on the last mile router. Detection of the attack by placing VSC at different intermediate routers in a distributed manner is out of thee scope of this paper.

## 3.2.2 Theory Behind VSC

As mentioned above, VSC monitors the number of unique source IP address $X_n$, $where\ n = 0, 1, 2, 3..$ and number of violating source IP addresses $V_n$, $where\ n = 0, 1, 2, 3..$ in each observation period $t_n$ to detect the occurrence of an attack in the network. Under the normal condition, the deviation of $X_n$ and $V_n$ from its mean value is less, however, under an attack these parameters deviate from their mean largely. Our detection engine thus monitors and detects (if any) such a significant change in these two parameters and confirms as an attack based on some threshold value. The following section describes the approach we use to detect such change in the above mentioned framework.

### 3.3.2.1 The Non-parametric CUSUM Algorithm:

Let $X_n$, $where\ n = 0, 1, 2, 3..$ and $V_n$, $where\ n = 0, 1, 2, 3..$ be the number of unique source IP addresses and the number of violating source IP address in an observation period $t_n$. Since $X_n$ and $V_n$ are highly dependent on different attributes of the network (such as size, time of the day, etc) from which they are collected, we first normalize $X_n$ and $V_n$ by the average value of $X_n$ and $V_n$ respectively. Let $\overline{X}_n$ and $\overline{V}_n$ represent the mean value of $X_n$ and $V_n$. Then $\overline{X}_n$ and $\overline{V}_n$ can be computed as follows $\overline{X}_n$=$\alpha$*$\overline{X}_n - 1$+(1-$\alpha$)*$X_n$ , $\overline{V}_n$=$\alpha$* $\overline{V}_n - 1$+(1-$\alpha$)*$V_n$ Where $\alpha$ is the memory factor and lies between 0 and 1.

Thus from $X_n$ and $V_n$ we define $A_n$=$X_n/\overline{X}_n$ $B_n$=$V_n/\overline{V}_n$ Since $A_n$ and $B_n$ are normalized values, no longer they are dependent on the current network characteristics.

We use the concept of sequential change point detection [72] in our detection algorithm. The goal of the change point detection mechanism is to detect the presence of a change of the mean value in a observed time series data. In our algorithm, it detects change in $A_n$ and $B_n$. However, accurate estimation of $A_n$ and $B_n$ are challenging task, hence we use non-parametric CUSUM method [74] in our detection algorithm. Non-parametric CUSUM is not model specific and hence suitable for our purpose. The basic idea of using non-parametric CUSUM to detect abrupt change in a time series data is based on the model presented in Peng et.al [72]. The details of non-parametric CUSUM can be found in [74]. Here we demonstrate how to apply non-parametric CUSUM on $A_n$ to detect change. Similar approach is taken in case of $B_n$.

Under normal condition, the mean of $A_n$ denoted by $c$ ($i.e,\ c = E(A_n)$) is near by 1. We chose a parameter which is the upper bound of $c$. From $A_n$ we derive another random sequence $\overline{A}_n$ such that $\overline{A}_n = A_n - c$. This transformation will make the mean value of $\overline{A}_n$ negative under normal condition, which is a basic assumption of non-parametric CUSUM algorithm [18]. Now consider $h$ as the lower bound of the amplitude of increase in the mean value of $\overline{A}_n$ during an attack and $h \gg c$. As presented in wang et al [5] the nonparametric CUSUM algorithm can be written as

$$Y_n = S_n - \min\ S_k, 1 \leq k \leq n$$

Where $S_k = \sum_{i=1}^{k} \overline{A}_i$, $S_0 = 0$ at the beginning and $Y_n$ is the accumulated positive values of $\overline{A}_i$. Thus if $Y_n$ is very large it is a clear indication of the deviation of the observed value of the random sequence from its mean value. We use a threshold value $N$ which is compared against $Y_n$ at the end of each observation period. If $Y_n$ exceeds $N$ an attack is detected. Thus we can now formally define the detection function as

$$D_N(Y_n) = \begin{cases} 0 & \text{if } Y_n \leq N \\ 1 & \text{otherwise.} \end{cases}$$

where 1́ndicates an attack and 0́detects normal traffic.

#### 3.3.2.2 Parameter Specification

Two key measures of greatest interest for a DDoS attack detection system are given below.

1. False alarm rate, i.e, the number of normal instances reported as attack over a specific period of time.
2. Detection time, i.e, time duration between the starting of an attack and the detection of the attack.

However, both these design goals are mutually conflicting, as expected! To achieve one other one often has to compromise up to extent. in practice (1,4,30 p) CUSUM is considered as optimal in terms of both false alarm rate and detection time. As presented in Brodsky et al[18]

$$\tau_N = \inf n : D_n(.) = 1$$

$$\rho_N = \frac{(\tau_N - m)^+}{N}$$

where $\tau_N = detection\ time\ \rho_N = normalized\ detection\ time\ after\ a\ change\ occurs.$ $Inf$ represents $infimum$, $n$ is the time when the attack started, $N$ is user defined threshold value

$\rho_N$ and $h$ can be related by the following equation

$$\rho_N \rightarrow \gamma = \frac{1}{h - |c - a|}$$

Where $h - |c - a|$ gives the mean of $\overline{A_n}$, after an attack begins. As mentioned in [5] the above equation gives an upper bound of the actual detection time. Thus to achieve our design goals we have to choose optimal values for the parameters $a$ and $N$. It is clear from Equation 1 and Equation 2 that once we are given $a$,h and *a detection time period* we can calculate $N$ accordingly. The parameter $a$ is used to offset $A_n$ to be $\overline{A_n}$, so that $\overline{A_n}$ has a negative mean under normal condition. If $a$ is chosen to be very high the likelihood of getting positive values in the sequence $\overline{A_n}$ is less. In turn the accumulated value i.e, $Y_n$ might not reach the threshold.

The parameter $N$ specifies the threshold for $Y_n$. If $N$ is chosen to be very high, false alarm rate will be low at a cost of high detection time. On the other hand, a small value of $N$ may increase the false alarm rate.

As mentioned earlier CUSUM algorithm needs $a$, $h$ and *detection time interval* to be specified and calculates $N$ by using Euations 1 and 2. Here $a$ is the upper bound estimation of the mean of $\overline{A_n}$. From the definition of $\overline{A_n}$ can safely assume $a$ as 1.1. The parameter $h$ specifies the amplitude of the minimum increase of the mean value of $\overline{A_n}$ under an attack. By following the same principle as in wang et al[5] we set $h = 2 * a$. We used *detection interval* as 3 sec. Assuming $c = 1$, from Equation 1 and 2 we get $N = 6.3$.

## 3.3 Performance Evaluation

In this section I mention the experimental evaluation of the proposed model under different attack conditions.

### 3.3.1 VSC Under Normal Condition

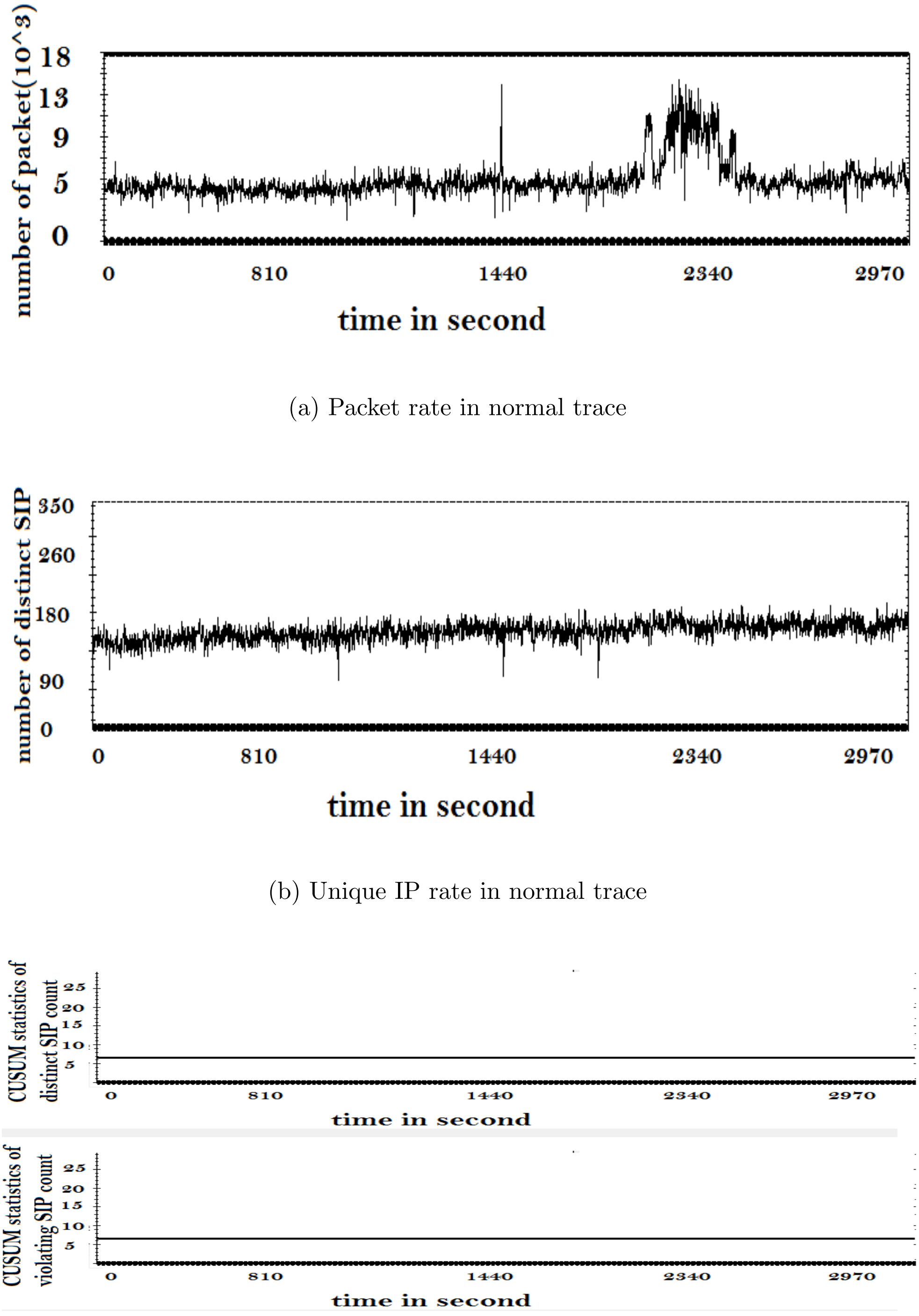
To perform our experiments AUK-VIII is used as normal packet trace.The result of VSC when applied on AUK-VIII is shown in figure 3.4. We can see that both of our test statistics are much below their threshold value.

### 3.3.2 Detection of Low Rate DDoS Attack

Our detection mechanism detects attack based on the deviation of the number of unique source IP addresses from its mean value, rather than the volume of the traffic. Thus our detection mechanism can detect low rate DDoS attack involving large number of sources. To demonstrate this we embedded a simulated spoofed DDoS attack of duration 1 minute into the normal traffic. The attack was performed at a rate of 250 packets/sec, which is much lesser than the usual traffic of the network. Hence, such an attack can easily escape a volume based detection mechanism. Figure 3.4(a) shows that during the attack period the traffic volume does not change significantly. However, as shown in figure 3.4(c), the proposed method detects the attack within a few seconds of the starting of the attack.

### 3.3.3 Detection of Randomly Spoofed Source IP Attack

we used the DARPA dataset in our experiments to show the effectiveness of VSC in detecting DDoS attack which uses a randomly spoofed source IP addresses Figure 3.6 demonstrate the result of this experiment. From Figure 3.6(c), we can see that

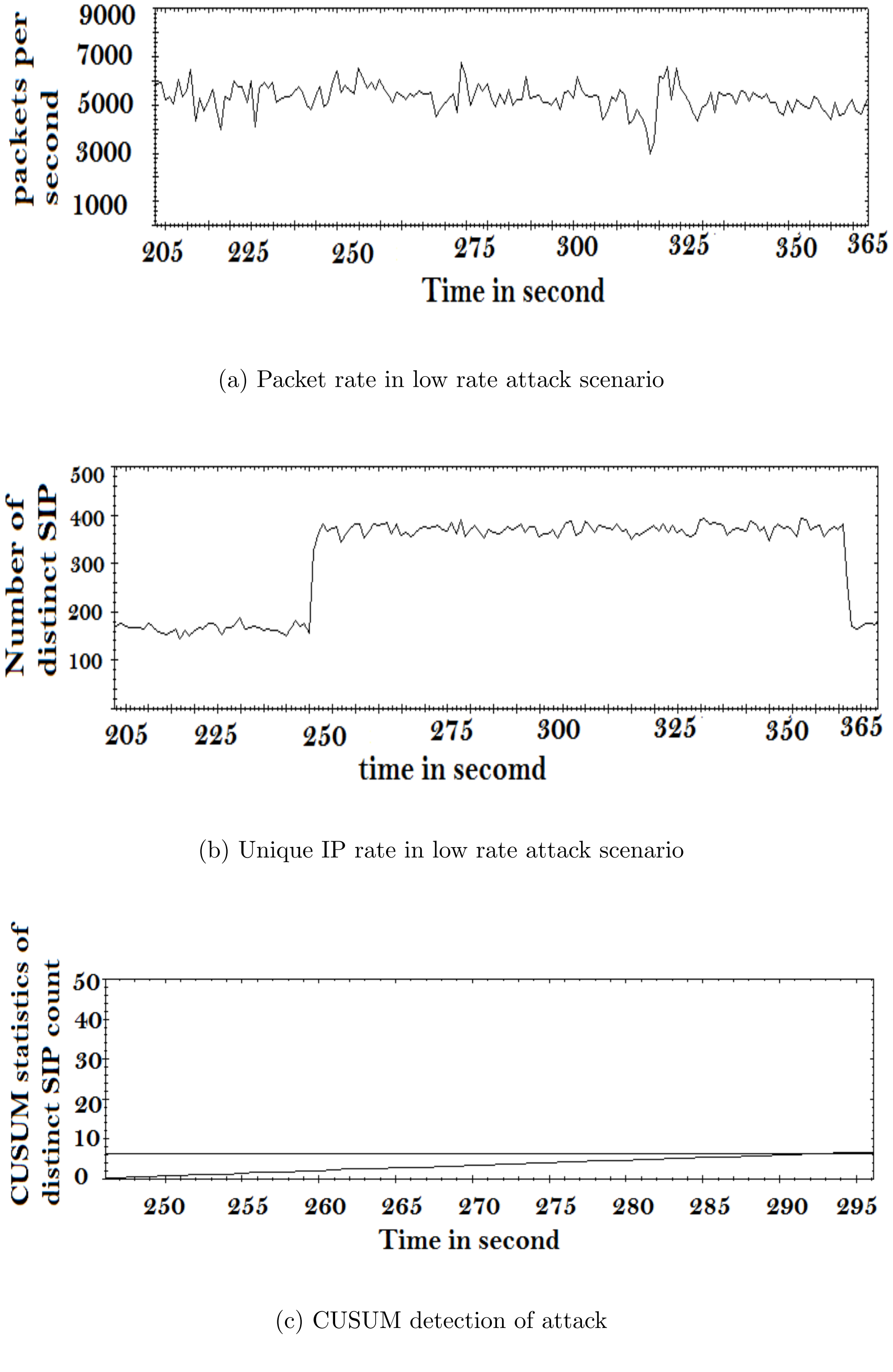


(a) Packet rate in normal trace

(b) Unique IP rate in normal trace

(c) CUSUM statistics in normal trace

Figure 3.4: Normal traffic characteristics

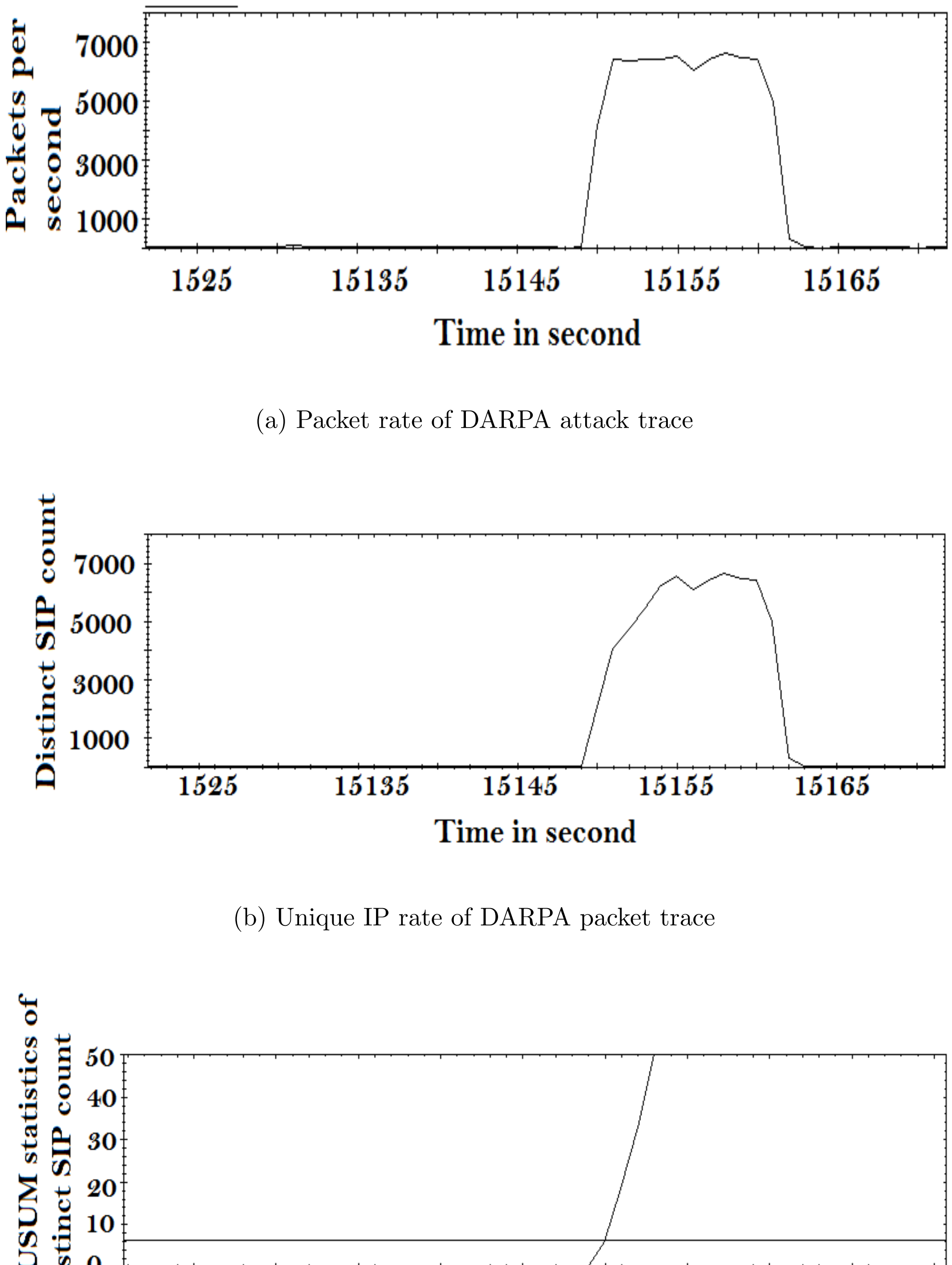


(a) Packet rate in low rate attack scenario

(b) Unique IP rate in low rate attack scenario

(c) CUSUM detection of attack

Figure 3.5: Result of VSC on low rate attack scenario



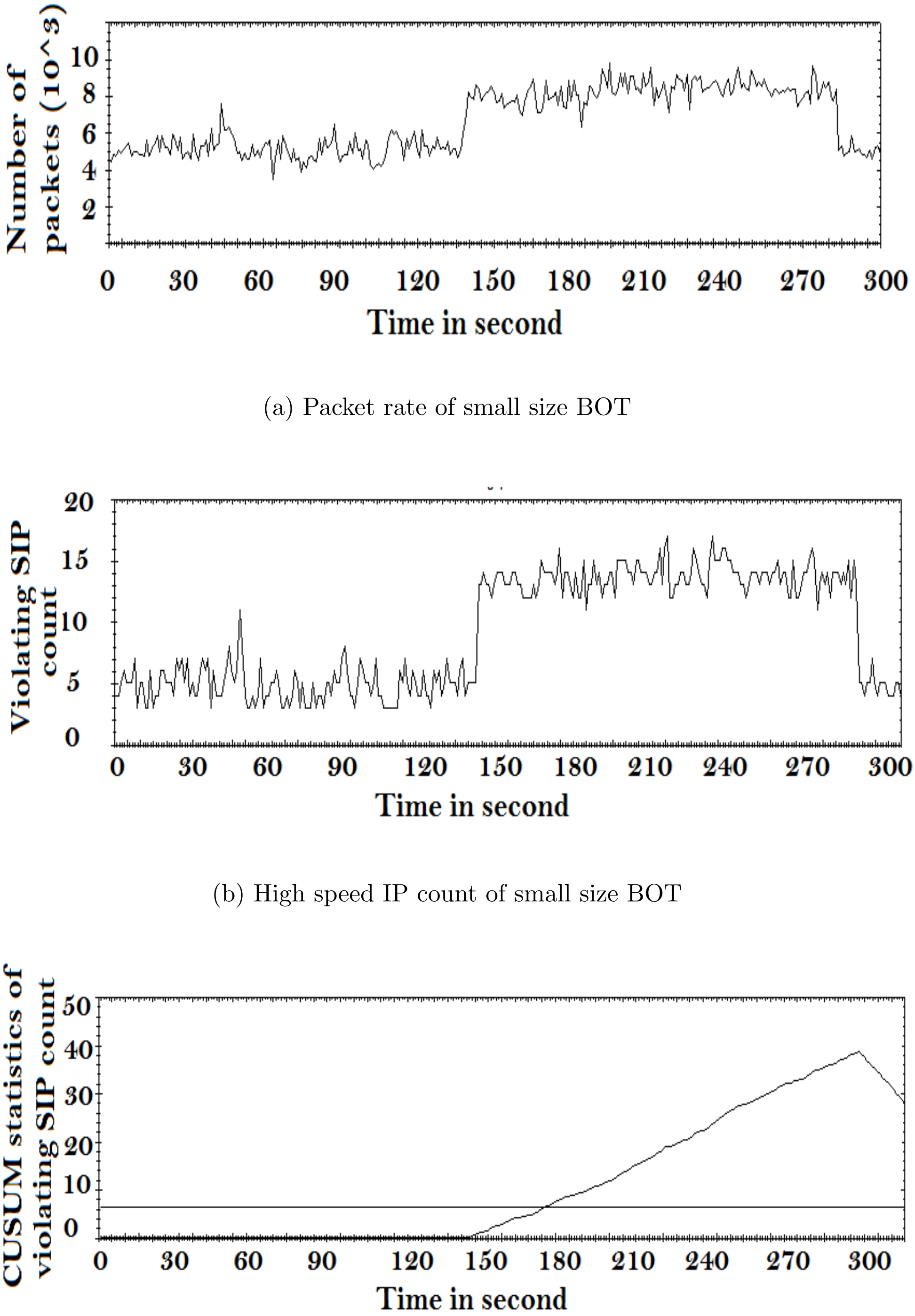
(a) Packet rate of DARPA attack trace

(b) Unique IP rate of DARPA packet trace

(c) CUSUM detection in DARPA packet trace

Figure 3.6: Result of VSC on DARPA dataset

the attack present in the DARPA dataset is easily detectable by VSC in less than 3 seconds.

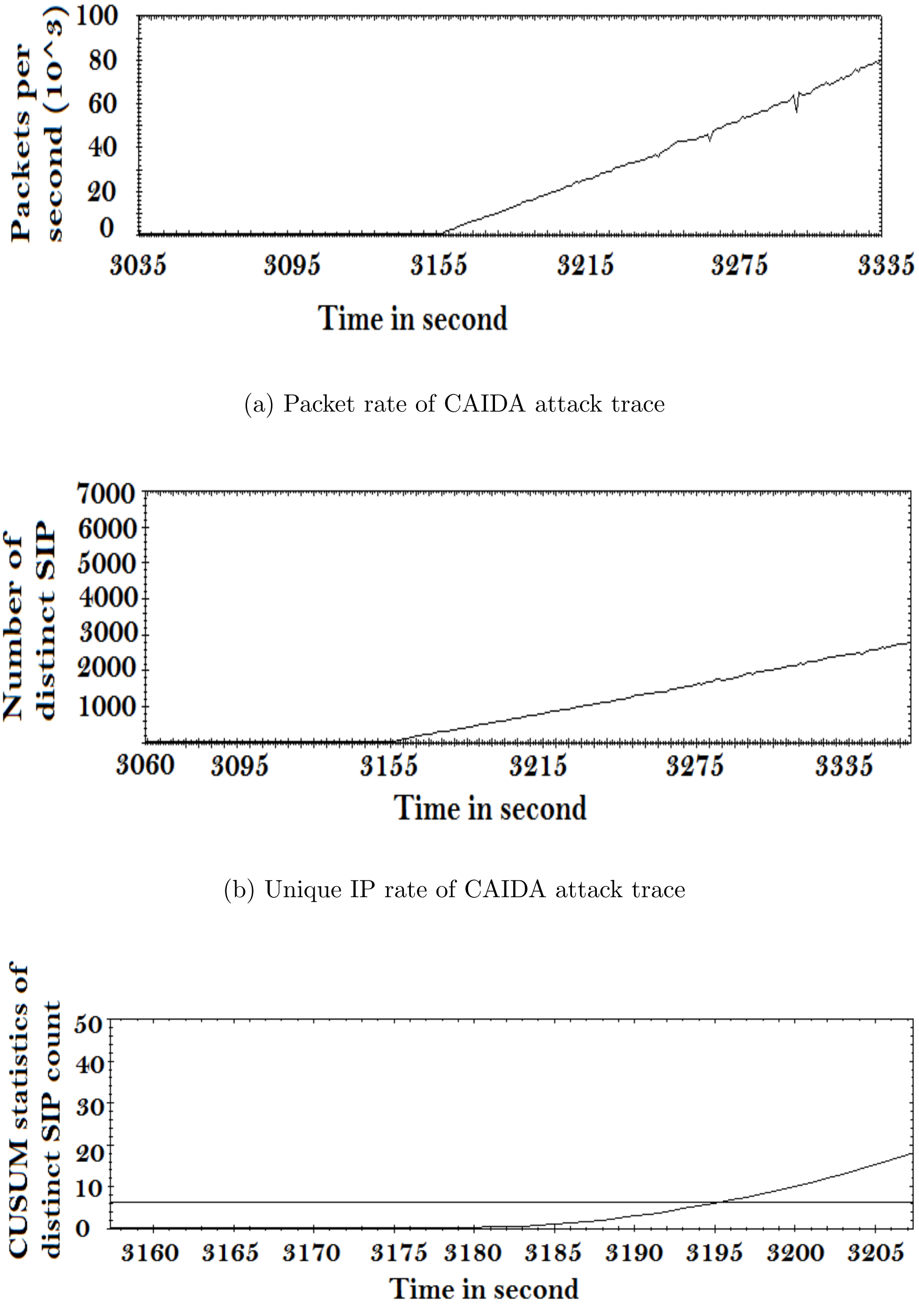


(a) Packet rate of small size BOT

(b) High speed IP count of small size BOT

(c) CUSUM detection on small size BOT

Figure 3.7: Result of VSC on small size BOT



(a) Packet rate of CAIDA attack trace

(b) Unique IP rate of CAIDA attack trace

(c) CUSUM detection on CAIDA attack trace

Figure 3.8: Result of VSC on CAIDA attack trace

### 3.3.4 Detection of DDoS Attack From a Small Size Botnet

The attacker may use a small size botnet (consisting a small number of sources) to carryout the attack. However, in that case the speed of the individual source need to be high to end up in an effective DDoS attack. We used a simulated attack consisting of 7 sources, performing an attack at 2000 packets/sec, for a duration of 10 minutes. We used 150 packets/sec as the threshold to mark an IP as violating IP. From Figure 3.7(c) it is clear that VSC can detect such an attack in less than 15 seconds, by detecting a change in the violating source IP count.

### 3.3.5 Detection of DDoS Attack in CAIDA-2007 Dataset

CAIDA 2007 is an widely used and benchmark DDoS network trace.We applied VSC on this dataset in our experiment. The result of VSC is shown in figure 3.8. The length of CAIDA is around 1 hour. The first 30 minutes contains traffic at a low rate, from a small number of sources. However, at the beginning of the second 30 half, there is an abrupt change in both number of unique source IP address as well as the traffic volume as shown in Figure 3.8(b) and 3.8(a). VSC detects this change within 8 seconds.

## 3.4 Discussion

In this chapter, I present a robust and low cost method to detect DDoS attack. Our method detects DDoS attack by monitoring the deviation of the count of unique SIPs and the count of SIPs whose transmission rate is higher than a threshold value. I demonstrated the near real time detection capability of the detection mechanism under different attack scenarios. Although VSC can detect the presence of DDoS attack almost immediately due to its low computational cost, VSC can not discriminate the attack packets from normal packets. In the next chapter, I present a DDoS defense solution which is capable of both detecting and mitigating (by discriminating attack packets from normal packets) a DDoS attack in near real time.