ISSN 22773061

# Mitigating the Distributed Denial of Service (DDoS) Attacks in Campus Local Area Network(CLAN)

Angshu Maan Sen, K. Anand Kumar Singha

Computer Centre, Dorgakona, Assam University Silchar

angshumaan_sen@yahoo.com

Computer Centre, Dorgakona, Assam University Silchar

k.anandsingha@gmail.com

## ABSTRACT

The Campus Local Area Network (CLAN) of academic institutions interconnect computers ranging from one hundred to about twenty five hundred and these computers are located in academic building(s), hostel building(s), faculty quarter(s), students amenities centre, etc all around the campus. The students, faculty and the supporting staff members use the network primarily for internet usage at both personal and professional levels and secondarily for usage of the available services and resources. Various web based services viz: Web Services, Mail Services, DNS, and FTP services are generally made available in the campus LAN. Apart from these services various intranet based services are also made available for the users of the LAN.

Campus LAN users from the hostels change very frequently and also sometime become targets (we call as soft targets) to the attackers or zombie because of either inadequate knowledge to protect their own computer/ laptop, which is also a legitimate node of the campus LAN; or their enthusiastic nature of experimentation. The interconnectivity of these legitimates nodes of the campus LAN and that of the attackers in the World Wide Web, make the computers connected in the LAN (nodes) an easy target for malicious users who attempt to exhaust the resources by launching Distributed Denial-of-Service (DDoS) attacks. In this paper we present a technique to mitigate the distributed denial of service attacks in campus wide LAN by limiting the bandwidth of the affected computers (soft targets) of the virtual LAN from a unified threat management (UTM) firewall. The technique is supported with help of bandwidth utilization report of the campus LAN with and without implementation of bandwidth limiting rule; obtained from the UTM network traffic analyzer. The graphical analyzer report on the utilization of the bandwidth with transmitting and receiving bits of the campus LAN after implementation of our bandwidth limiting rule is also given.

## Indexing terms/Keywords

DDoS, CLAN, Firewall, VLAN, Syn Flood, UTM.

## Academic Discipline And Sub-Disciplines

Computer Science and Engineering, Electronics and Communication Engineering, Information Technology.

## SUBJECT CLASSIFICATION

Network Security.

## TYPE (METHOD/APPROACH)

Case Study Analysis; Experimental.

Council for Innovative Research

Peer Review Research Publishing System

Journal: INTERNATIONAL JOURNAL OF COMPUTERS & TECHNOLOGY

Vol 10, No 6

editor@cirworld.com
www.cirworld.com, member.cirworld.com

## INTRODUCTION

A campus wide local area network (LAN) is a computer network that spans in an academic campus connecting the academic departments located with in a relatively small area. Most of the campus wide LANs are confined to a group of buildings interconnected with each other through either optical fibre cable (OFC) using Fibre Distributed Data Interface (FDDI) Technology or unshielded twisted pair (UTP) cable located within 100 metres distance or inside the Department using layer-2 manageable switches. It connects workstations and personal computers called as nodes (individual computer) to various servers available in a LAN and are also connected to internet through a layer-3 switch via Firewall (optional) for access to the internet. Each node has its own central processing unit with which it executes programs; but it is able to access data and devices and users share resources like files, printers, drives etc or other applications Users can also use the LAN to communicate with each other, by sending email or engage in chat sessions, playing games, sharing resources etc. [1]. LANs are capable of transmitting data at very fast rates, as they are interconnected through OFC or UTP and also because the data has a short distance to cover. A large campus wide LAN can accommodate many thousands of computers (nodes) by dividing into logical groups with different Default Gateway in a subnet and creating Virtual LANs (VLAN) with Spanning Tree Protocol (STP) to avoids the broadcast storms of L2 Switch.. Sometime wireless LAN facility for a specific area, conference room or smart class room is created for users who can get access to resources available in a campus wide LAN as well as get access to the internet.

VLANs support logical grouping of network nodes to reduce broadcast traffic and allow more control in implementing security policies. VLANs are implemented in the campus wide LAN to enhance security and traffic control; to ease network adds, moves, and changes; to contain broadcasts. It helps to enhance manageability of switched LANs [2]. With the migration from shared to switched LANs the term VLAN has become a common term not only within standards committees and engineering departments, but in many network management centers and campuses [ 3]. Generally these VLANs are implemented in a campus wide LAN by creating and writing rules in a network address table of layer-3 switch. Network managers of the campus wide LAN define VLANs based on the following characteristics: physical ports, protocol type, MAC address, and IP subnets. Since the primary objective of implementing VLANs is to enhance network manageability during the network planning and design stages, centralized VLAN management is an important requirement. When VLANs can be defined remotely, and managed from a central location network managers can more easily design their networks based on business objectives such as improved service to users, while also continuously monitoring VLAN performance and adjusting VLAN policies and definitions.

However, the interconnectivity among computers in the campus LAN and that of the World Wide Web, renders the computers connected in the LAN (nodes) an easy target for malicious users who attempt to exhaust their resources and launch Distributed Denial-of-Service (DDoS) attacks through SYN flooding from the user end to the entire campus wide local area network users. In a **SYN flood** an attacker sends a succession of SYN requests to a target's system in an attempt to consume enough server resources to make the system unresponsive to legitimate traffic. A SYN flood attack works by not responding to the server with the expected ACK code. The malicious client can either simply not send the expected ACK, or by spoofing the source IP address in the SYN, causing the server to send the SYN-ACK to a falsified IP address - which will not send an ACK because it "knows" that it never sent a SYN [4] .

The server will wait for the acknowledgement for some time, as simple network congestion could also be the cause of the missing ACK, but in an attack increasingly large numbers of half-open connections will bind resources on the server until no new connections can be made, resulting in a denial of service to legitimate traffic. Some systems may also malfunction badly or even crash if other operating system functions are starved of resources in this way.

In this paper we present a technique to mitigate the distributed denial of service attacks in campus wide LAN by limiting the bandwidth of the affected VLAN through a unified threat management firewall. We also present here the report from the network traffic analyzer on the utilization of the bandwidth during the DDoS attack with the SYN flooding with transmitting and receiving bits and the report after implementation of our bandwidth limiting rule.

## HISTORY OF DDOS ATTACKS

A Denial of Service attack is a malicious attempt by a single person or a group of people to cause the victim, site, or node to deny service to its customers. In a denial-of-service attack a legitimate users of a service is prevented from using that service. There are two general forms of DoS attacks: those that crash services and those that flood services. The most common types of DoS attack are:

- Consumption of computational resources, such as bandwidth, disk space, or CPU time.
- Disruption of configuration information, such as routing information.
- Disruption of state information, such as unsolicited resetting of TCP sessions.
- Disruption of physical network components, such as preventing the access to the servers.

When the malicious attempt is derived from a single host of the network, it constitutes a Denial of Service attack. On the other hand, it is also possible that a lot of malicious hosts coordinate to flood the victim with an abundance of attack packets also termed as SYN flooding, so that the attack takes place simultaneously from multiple points. This type of attack is called a Distributed Denial of Service, or simply DDoS attack [5]. In DoS attacks IP address of the attacker is forged through spoofing so that the location of the attacker cannot easily be identified and to also to prevent filtering of the packets based on the source address.

## DDoS Attack Description

DoS attacks attempt to usurp the available resources of the victim's campus wide network. These resources can be network bandwidth, computing power, or operating system data structures. To launch a DDoS attack, the attacker first install the malicious code into one vulnerable machine (may be one workstation, laptop or even a file server) which is either running no antivirus software or out-of-date antivirus software, or those that have not been updated to the latest signature of the antivius software. The affected machine then first build a network of computers, also called as attack network comprising in a single VLAN or in multiple VLAN. The computers in the attack network is used to produce the volume of traffic needed to deny services to computer users. To create this attack network, attackers discover vulnerable sites or hosts on the network.

The next step for the intruder is to install new programs (known as attack tools) on the compromised hosts of the attack network. The hosts that are running these attack tools are known as zombies, and they can carry out any attack under the control of the attacker. Many zombies together form what we call an army [6 ].

For initial identification of the vulnerable hosts [7 ] [8 ] [ 9] the attackers use scanning techniques, such as Random scanning, Hit-list scanning, Topological scanning, Local subnet scanning, Permutation scanning.

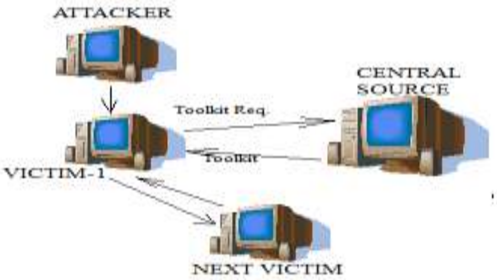
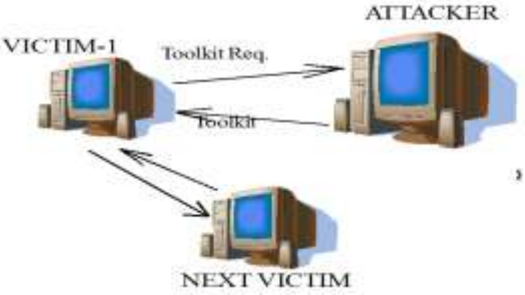
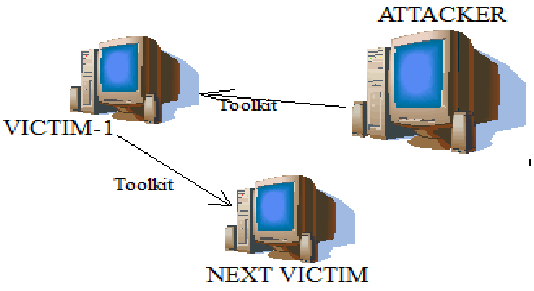
- **Random scanning**: In this an attacker probes IP addresses randomly from the IP address space and checks their vulnerability. When it finds one it installs the malicious code into the machine and the process is continued.
- **Hit-list scanning**: In this an attacker already starts with a pre-collected list of a large number of potentially vulnerable machines. In their effort to create their army, they begin scanning down the list in order to find more vulnerable machines.
- **Topological scanning**: In this an attacker uses information contained on the victim machine (an already-compromised host) in order to find new targets. New targets are found by looking for URLs of the unaffected machines in the disk of the victim machine that it wants to infect. Then it renders these URLs targets and checks their vulnerability.
- **Local subnet scanning**: This type of scanning acts behind a firewall in an area (the soft targets) that is considered to be infected by the malicious scanning program. The compromised host looks for targets in its own campus wide local area network. It uses the information that is hidden in the private IP addresses generally used to configure any campus wide LAN of an academic institution. More specifically, a single copy of the scanning program is running behind a firewall and tries to break into all vulnerable machines that would otherwise be protected by the firewall. This mechanism can be used in conjunction with other scanning mechanisms: for example, a compromised host can start its scans with local subnet scanning, looking for vulnerable machines in its local network. As soon as it has probed all local machines, it can continue the probing process by switching to another scanning mechanism in order to scan off-local network machines.
- **Permutation scanning:** In this technique all machines share a common pseudorandom permutation list of IP addresses constructed using any block cipher of 32 bits with a preselected key [8].

A campus virtual community named Myclub2.com [11] is introduced through a case study which establish, labors its design goals and value degree settings of network behavior, analyzes user's initial motivation and induced motivation. The work also elaborates the setting mode and the function system of incentive mechanism. Ramamoorthi et al introduces an anomaly detection mechanism to detect DDoS attacks using Enhanced Support Vector Machine (ESVM) with string kernels [12]. In this work normal user access behavior attributes is used as training samples for ESVM, which produces the model file. Application and Network layer DDoS attacks are also classified in the work. with ESVM. Zhijun et al describes two typical types of DDoS, Flood DDoS (FDDoS) and Low-rate DDoS (LDDoS) attacks [13]. Through experimental results it shows that FDDoS sends a large amount of traffic to the victim which is easy to be detected wher as LDDoS organizes a small quantity of traffic to the victim but it is difficult to detect. A scheme to counter application layer DDoS attack and to schedule the flash crowd during DDoS attacks is introduced [14]. In this scheme, an Access Matrix is defined to capture the access patterns of the legitimate clients and the normal flash crowd.

## DDoS Attack Propogation

There are three groups of mechanisms for propagating malicious code and building attack networks [10]. These are Central source propagation, Back-chaining propagation, Autonomous propagation. The description along with their graphical representation is presented in table 1 in the following page.

**Table 1: DDoS Attack Propagation Type**

| Type | Description | Graphical representation |
|---|---|---|
| **Central source propagation**: | After the discovery of the vulnerable victim, toolkit request is sent and then attack toolkit is transferred from a central source to the newly made victim. After the toolkit is transferred, an automatic installation of the attack tools takes place on this victim, controlled by a scripting mechanism. |  ATTACKER<br>CENTRAL SOURCE<br>Toolkit Req.<br>Toolkit<br>VICTIM-1<br>NEXT VICTIM |
| **Back-chaining propagation** | The attack toolkit is transferred to the victim from the attacker. More specifically, the attack tools that are installed on the attacker include special methods for accepting a connection from the victim and sending a file to it that contains the attack tools. This back-channel file copy can be supported by simple port listeners that copy file contents. |  ATTACKER<br>VICTIM-1<br>Toolkit Req.<br>Toolkit<br>NEXT VICTIM |
| **Autonomous propagation** | The attacker transfers the attack toolkit to the victim at the exact moment that it breaks into that system. This mechanism differs from the previously mentioned mechanisms in that the attack tools are planted into the compromised host by the attackers themselves and not by an external file source |  |

## DDoS Attack Methodology

A perpetrator in DDoS attack attempts either Internet Control Message Protocol (ICMP) flood or SYN flood in a campus LAN. In ICMP flooding Smurf attack, Ping flood, and Ping of death are tried. A smurf attack relies on misconfigured network devices that allow packets to be sent to all computer hosts on a particular network via the broadcast address of the network, rather than a specific machine. The network then serves as a smurf amplifier. In such an attack, the perpetrators will send large numbers of IP packets with the source address faked to appear to be the address of the victim. The network's bandwidth is quickly used up, preventing legitimate packets from getting through to their destination. Ping flood is based on sending the victim an overwhelming number of ping packets, usually using the "ping" command. It is very simple to launch, the primary requirement being access to greater bandwidth than the victim. Ping of death is based on sending the victim a malformed ping packet, which might lead to a system crash.

A SYN flood occurs when a host sends a flood of TCP/SYN packets, often with a forged sender address. Each of these packets is handled like a connection request, causing the server to spawn a half-open connection, by sending back a TCP/SYN-ACK packet (Acknowledge), and waiting for a packet in response from the sender address (response to the ACK Packet). However, because the sender address is forged, the response never comes. These half-open connections saturate the number of available connections the server is able to make, keeping it from responding to legitimate requests until after the attack ends. The SYn flooding methodology is explained graphically in the figure. It is clear from the Fig 1 that the perpetrator send series of SYN request and receives Ack without responding to the Ack-Req. and thus half open connections at the server side is created and thus exhaust the all the ports that the server can open for providing service. This prevents a legitimate user to get the service form the server.

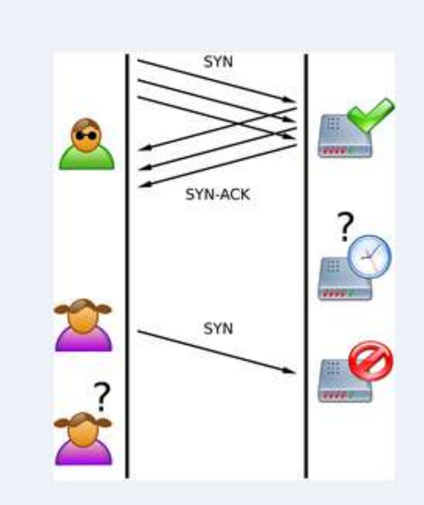


**Fig 1: SYN Flooding in Campus LAN**

.

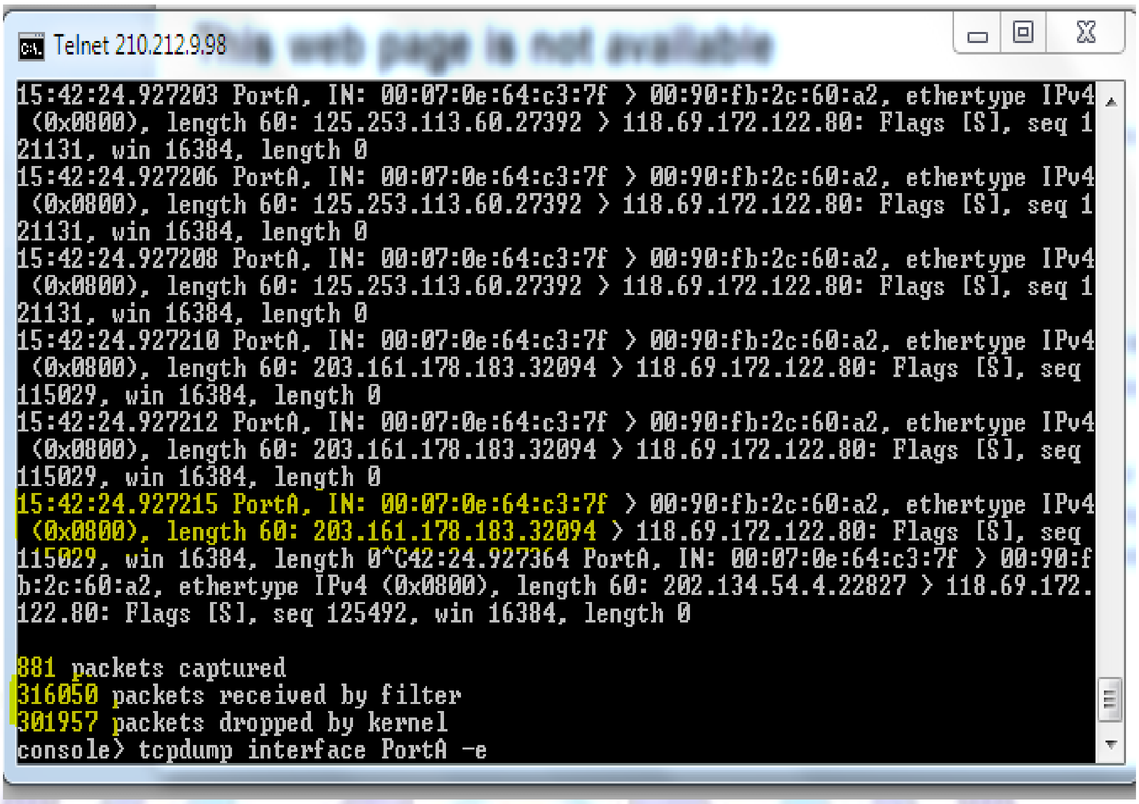
## DDoS Limiting Methodology

The DDoS attack in a campus LAN is confirmed when there is severe flip flop between the availability and non-availability of internet and intranet connectivity. Generally as the campus LAN is heavily used by it's users, a support maintenanece team is deployed to provide support and administer the campus LAN. In case of non-availability or any flip flop as mentioned above, the team is informed and a call is registered regarding the unavailability of the services. The DDoS attack in a campus LAN can be confirmed through the mechanism as given below:

- Information from the users regarding non availability of services
- Availability of PING Reply status to any URL although non-availability of the concerning web pages through any web browser.
- Report of bulk consumption of the available bandwidth from the Firewall/UTM.
- Report of changing of Dynamic global IP from Firewall/UTM.
- Report of Increase in SYN /ICMP/UDP Flood and the drop of traffic due to this.

In support of the above we present our university case study report with snapshot in fig 2, 3 and 4. In the Denial of Service (DoS) settings of the Cyberoam firewall we apply destination based "SYNC"/"UDP" flooding to check for the staus of flooding.

During the troubleshooting session, we remove one by one optical Fibre Cable connected from the Core Switch to manageable switch at various buildings in the campus LAN physically and check the status of flooding through firewall. Each manageable switch is configured with different VLAN (private IPs) having different Broadcast Addresses. Adopting this process we observe that there was maximum flooding from the Pcs connected to one particular VLAN of the Network. The snapshots given at Fig 2 reports about this attack which is taken from the remote on the accessible global/WAN IP attached to the firewall (firewall support team from remote destination). This snapshot clearly shows that the inbound traffic is generated from the global/Public IPs only which is not possible /true in case of a campus LAN as all users of the campus LAN are configured with private IP addresses. The snapshot further elaborates the "Packets Received by the Firewall /filter" and the "Packets dropped by the Kernel of the Firewall". It can be seen from the figure that 881 packets have been captured whereas 316050 packets have been received by the filter and 301957 packets have been dropped by the kernel indicating flooding of the campus LAN confirming the DDoS attack of the LAN. The generated report is taken form Port A which is configured with the IP for internet usage.

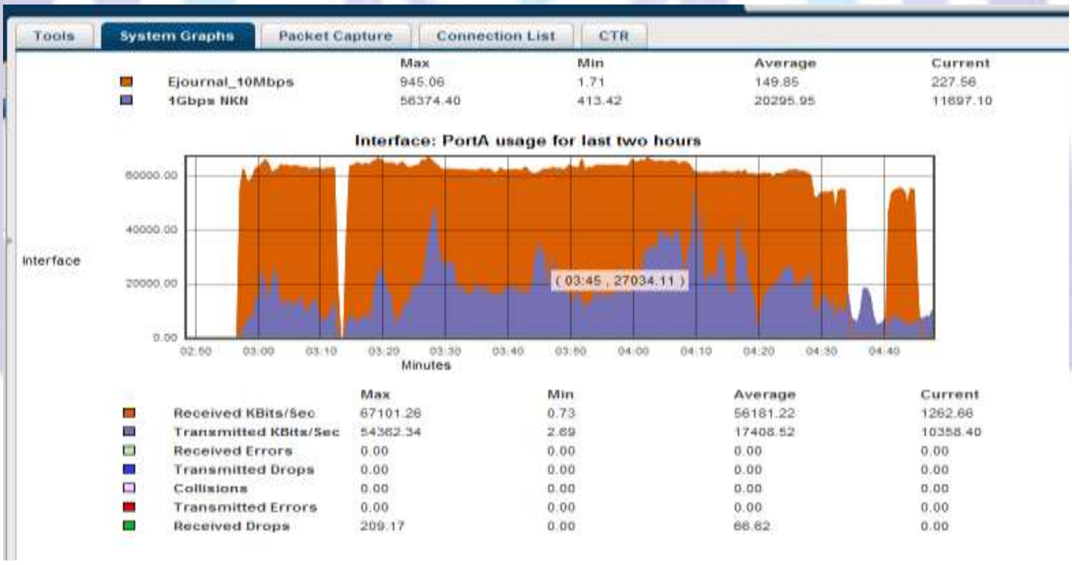


```
15:42:24.927203 PortA, IN: 00:07:0e:64:c3:7f > 00:90:fb:2c:60:a2, ethertype IPv4
 (0x0800), length 60: 125.253.113.60.27392 > 118.69.172.122.80: Flags [S], seq 1
21131, win 16384, length 0
15:42:24.927206 PortA, IN: 00:07:0e:64:c3:7f > 00:90:fb:2c:60:a2, ethertype IPv4
 (0x0800), length 60: 125.253.113.60.27392 > 118.69.172.122.80: Flags [S], seq 1
21131, win 16384, length 0
15:42:24.927208 PortA, IN: 00:07:0e:64:c3:7f > 00:90:fb:2c:60:a2, ethertype IPv4
 (0x0800), length 60: 125.253.113.60.27392 > 118.69.172.122.80: Flags [S], seq 1
21131, win 16384, length 0
15:42:24.927210 PortA, IN: 00:07:0e:64:c3:7f > 00:90:fb:2c:60:a2, ethertype IPv4
 (0x0800), length 60: 203.161.178.183.32094 > 118.69.172.122.80: Flags [S], seq
115029, win 16384, length 0
15:42:24.927212 PortA, IN: 00:07:0e:64:c3:7f > 00:90:fb:2c:60:a2, ethertype IPv4
 (0x0800), length 60: 203.161.178.183.32094 > 118.69.172.122.80: Flags [S], seq
115029, win 16384, length 0
15:42:24.927215 PortA, IN: 00:07:0e:64:c3:7f > 00:90:fb:2c:60:a2, ethertype IPv4
 (0x0800), length 60: 203.161.178.183.32094 > 118.69.172.122.80: Flags [S], seq
115029, win 16384, length 0^C42:24.927364 PortA, IN: 00:07:0e:64:c3:7f > 00:90:f
b:2c:60:a2, ethertype IPv4 (0x0800), length 60: 202.134.54.4.22827 > 118.69.172.
122.80: Flags [S], seq 125492, win 16384, length 0

881 packets captured
316050 packets received by filter
301957 packets dropped by kernel
console> tcpdump interface PortA -e
```

**Fig 2: Network Packet statistics collected from Firewall during DDoS attack.**

In Fig 3, we present the Port A bandwidth usage report taken from the firewall. The figure clearly shows the flooding of the Campus LAN with the received and the transmitted bits in Kbps causing the choke of the entire LAN bandwidth.

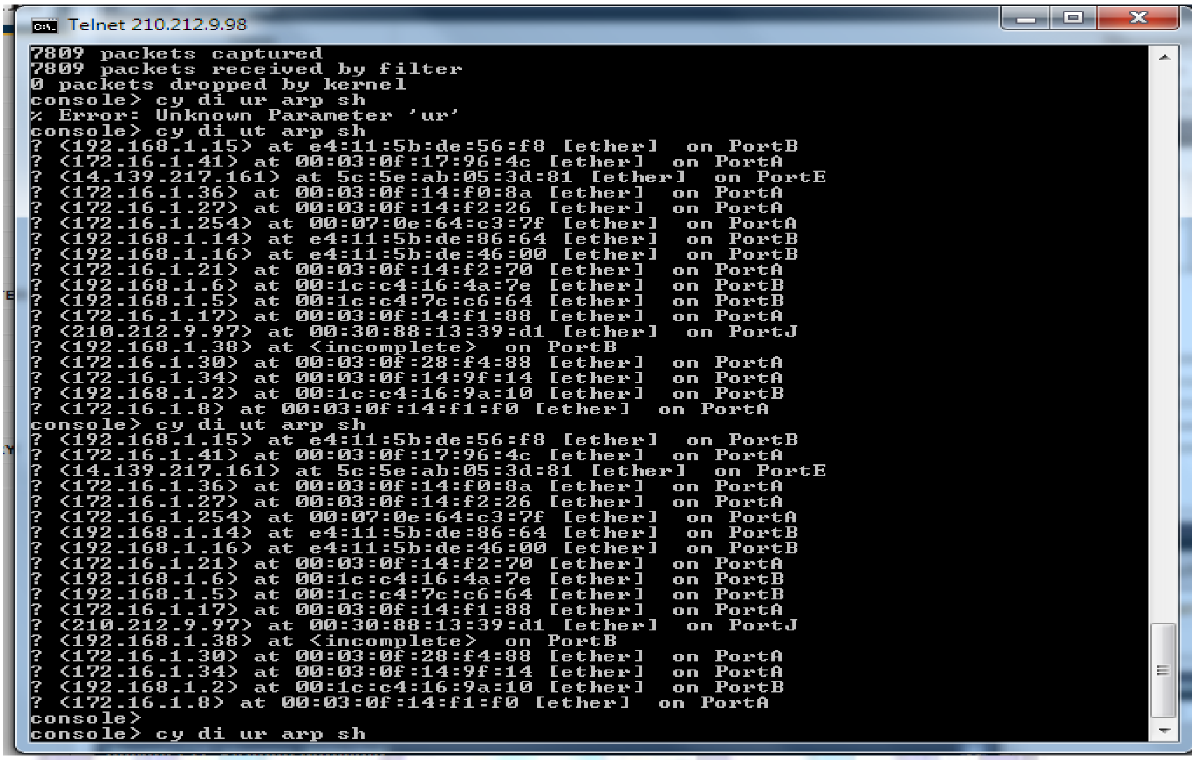


| | | Max | Min | Average | Current |
|---|---|---|---|---|---|
| 🟧 | Ejournal_10Mbps | 945.06 | 1.71 | 149.85 | 227.56 |
| 🟪 | 1Gbps NKN | 56374.40 | 413.42 | 20295.95 | 11697.10 |

| | | Max | Min | Average | Current |
|---|---|---|---|---|---|
| 🟧 | Received KBits/Sec | 67101.26 | 0.73 | 56181.22 | 1262.66 |
| 🟪 | Tranxmitted KBits/Sec | 54362.34 | 2.69 | 17408.52 | 10358.40 |
| 🟩 | Received Errors | 0.00 | 0.00 | 0.00 | 0.00 |
| 🟦 | Transmitted Drops | 0.00 | 0.00 | 0.00 | 0.00 |
| 🟪 | Collisions | 0.00 | 0.00 | 0.00 | 0.00 |
| 🟥 | Transmitted Errors | 0.00 | 0.00 | 0.00 | 0.00 |
| 🟩 | Received Drops | 209.17 | 0.00 | 66.62 | 0.00 |

**Fig 3: Report of Consumption of bandwidth at the Port configure for accessing Internet/Intranet**

Further for confirmation of the DDoS attack of the network and for the detection of the probable victims/flooders of the Campus LAN we take a diagnostics view of the Cyberoam Utilities of Address Resolution Protocol(ARP). The generated report is placed in fig 4.

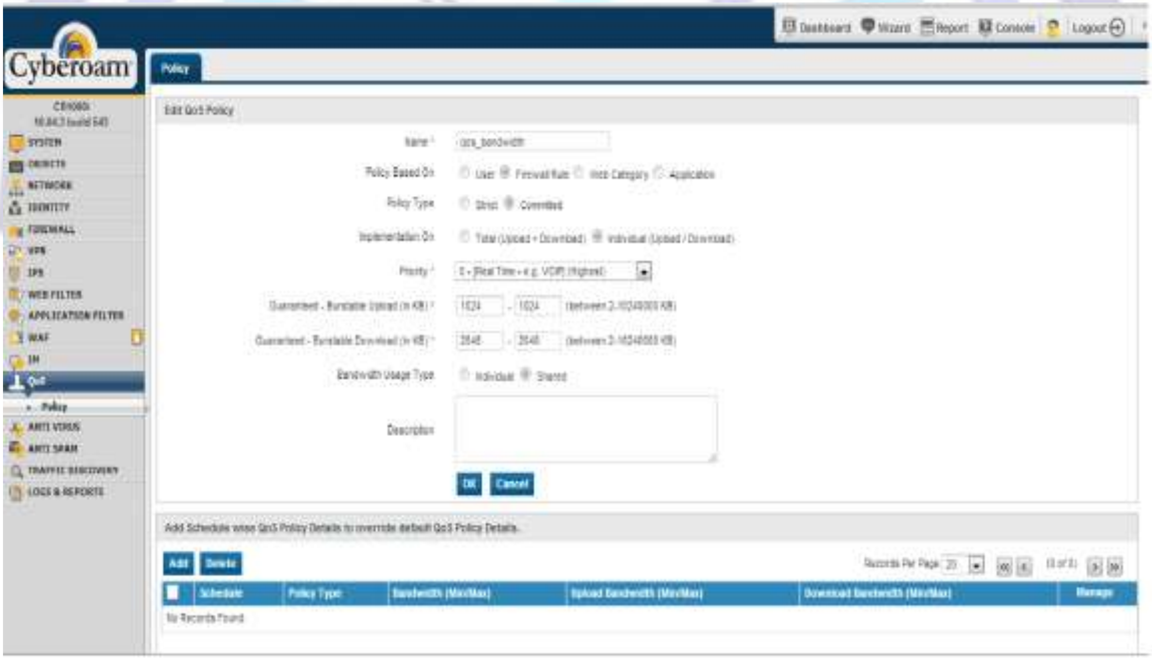


```
7809 packets captured
7809 packets received by filter
0 packets dropped by kernel
console> cy di ur arp sh
% Error: Unknown Parameter 'ur'
console> cy di ut arp sh
? <192.168.1.15> at e4:11:5b:de:56:f8 [ether]  on PortB
? <172.16.1.41> at 00:03:0f:17:96:4c [ether]  on PortA
? <14.139.217.161> at 5c:5e:ab:05:3d:81 [ether]  on PortE
? <172.16.1.36> at 00:03:0f:14:f0:8a [ether]  on PortA
? <172.16.1.27> at 00:03:0f:14:f2:26 [ether]  on PortA
? <172.16.1.254> at 00:07:0e:64:c3:7f [ether]  on PortA
? <192.168.1.14> at e4:11:5b:de:86:64 [ether]  on PortB
? <192.168.1.16> at e4:11:5b:de:46:00 [ether]  on PortB
? <172.16.1.21> at 00:03:0f:14:f2:70 [ether]  on PortA
? <192.168.1.6> at 00:1c:c4:16:4a:7e [ether]  on PortB
? <192.168.1.5> at 00:1c:c4:7c:c6:64 [ether]  on PortB
? <172.16.1.17> at 00:03:0f:14:f1:88 [ether]  on PortA
? <210.212.9.97> at 00:30:88:13:39:d1 [ether]  on PortJ
? <192.168.1.38> at <incomplete>  on PortB
? <172.16.1.30> at 00:03:0f:28:f4:88 [ether]  on PortA
? <172.16.1.34> at 00:03:0f:14:9f:14 [ether]  on PortA
? <192.168.1.2> at 00:1c:c4:16:9a:10 [ether]  on PortB
? <172.16.1.8> at 00:03:0f:14:f1:f0 [ether]  on PortA
console> cy di ut arp sh
? <192.168.1.15> at e4:11:5b:de:56:f8 [ether]  on PortB
? <172.16.1.41> at 00:03:0f:17:96:4c [ether]  on PortA
? <14.139.217.161> at 5c:5e:ab:05:3d:81 [ether]  on PortE
? <172.16.1.36> at 00:03:0f:14:f0:8a [ether]  on PortA
? <172.16.1.27> at 00:03:0f:14:f2:26 [ether]  on PortA
? <172.16.1.254> at 00:07:0e:64:c3:7f [ether]  on PortA
? <192.168.1.14> at e4:11:5b:de:86:64 [ether]  on PortB
? <192.168.1.16> at e4:11:5b:de:46:00 [ether]  on PortB
? <172.16.1.21> at 00:03:0f:14:f2:70 [ether]  on PortA
? <192.168.1.6> at 00:1c:c4:16:4a:7e [ether]  on PortB
? <192.168.1.5> at 00:1c:c4:7c:c6:64 [ether]  on PortB
? <172.16.1.17> at 00:03:0f:14:f1:88 [ether]  on PortA
? <210.212.9.97> at 00:30:88:13:39:d1 [ether]  on PortJ
? <192.168.1.38> at <incomplete>  on PortB
? <172.16.1.30> at 00:03:0f:28:f4:88 [ether]  on PortA
? <172.16.1.34> at 00:03:0f:14:9f:14 [ether]  on PortA
? <192.168.1.2> at 00:1c:c4:16:9a:10 [ether]  on PortB
? <172.16.1.8> at 00:03:0f:14:f1:f0 [ether]  on PortA
console>
console> cy di ur arp sh
```

**Fig 4: Report of ARP collected from firewall**

## Limiting DDoS

To mitigate the affect of DDoS attack we apply the bandwidth limiting methodology by creating a separate rule for the affected VLAN. The Screen Shot of the Quality of Service (QoS) Policy after application of the limiting rule is presented at fig 5. We call the limiting rule as “qos_bandwidth”, which is policy based firewall rule” defined in figure 5. The QoS policy type is Committed and implemented on Individual Upload/ Download having different priorities from ‘0’ to ‘7’; ‘0’ means the highest priority (applicable for VOIP, etc) and ‘7’ means the lowest priority (applicable for P2P, etc). We apply the bandwidth limiting rule in the upload and download to 1024 Kb and 2048 Kb so that the flooding by the affected victims of the identified VLAN does not consume the entire bandwidth of the campus LAN.



Edit QoS Policy

Name * qos_bandwidth

Policy Based On ○ User ● Firewall Rule ○ Web Category ○ Application

Policy Type ○ Strict ● Committed

Implementation On ○ Total (Upload + Download) ● Individual (Upload / Download)

Priority * 0 - [Real Time - e.g. VOIP] (Highest)

Guaranteed - Burstable Upload (in KB) * 1024 - 1024 (between 2-1024000 KB)

Guaranteed - Burstable Download (in KB) * 2048 - 2048 (between 2-1024000 KB)

Bandwidth Usage Type ○ Individual ● Shared

Description

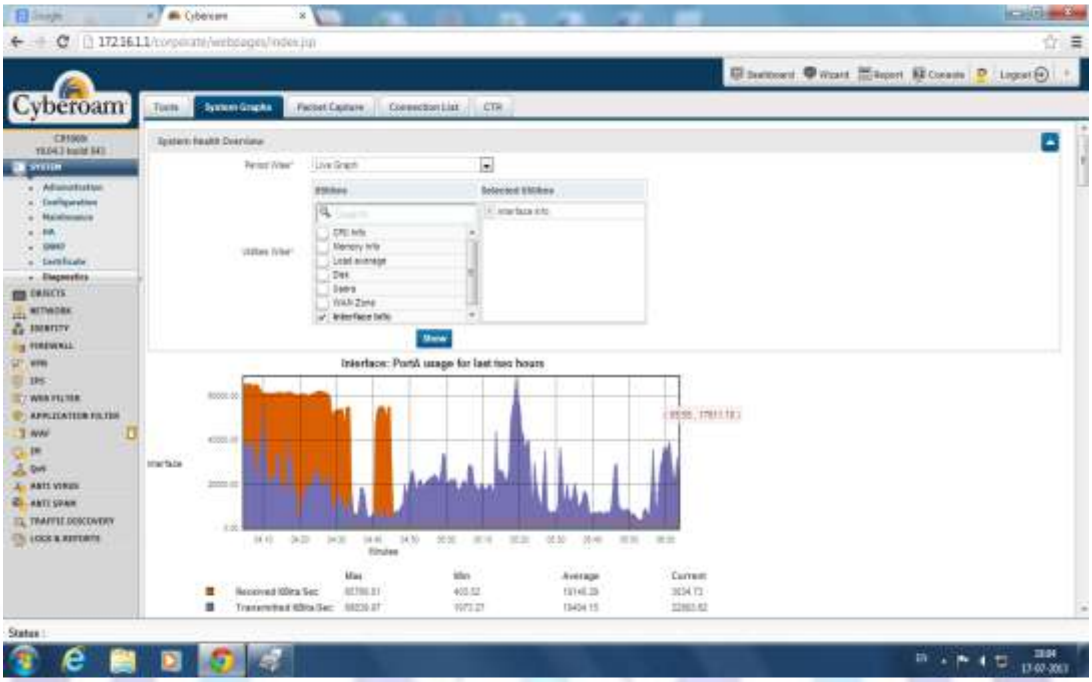
Add Schedule wise QoS Policy Details to override default QoS Policy Details.

Records Per Page 20 (1 of 1)

| □ | Schedule | Policy Type | Bandwidth (Min/Max) | Upload Bandwidth (Min/Max) | Download Bandwidth (Min/Max) | Manage |
|---|---|---|---|---|---|---|
| No Records Found | | | | | | |

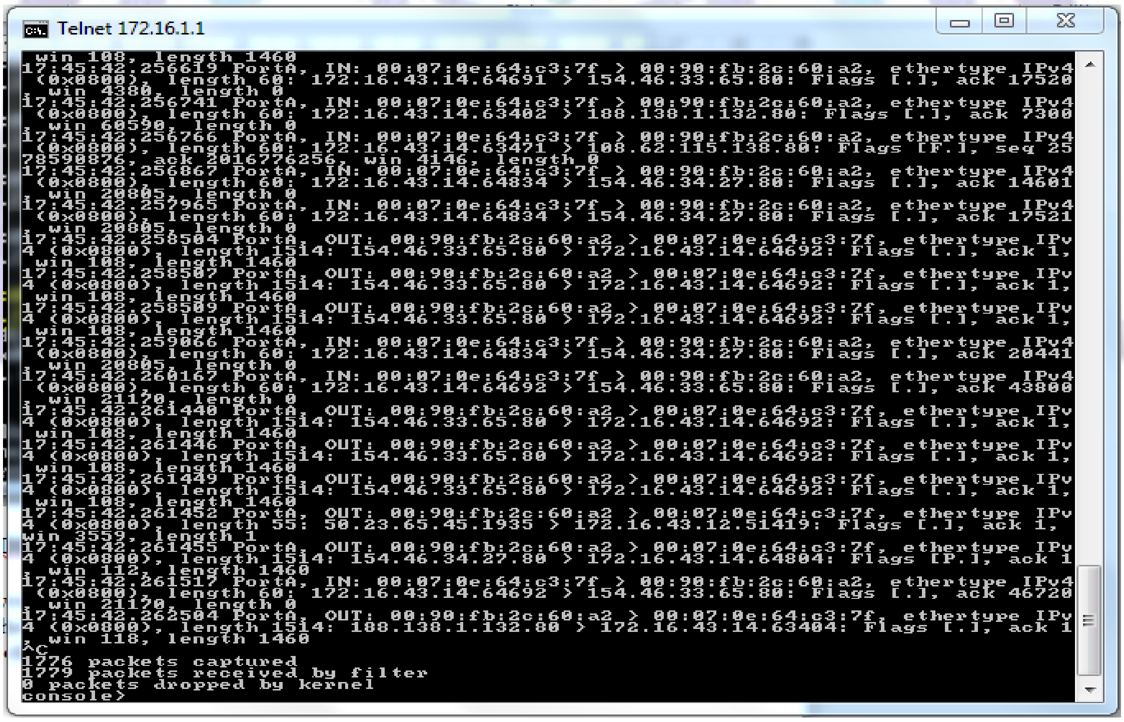
**Fig 5: Application of QOS policy on the DDoS Victims .**

In fig 6 , we present the Port A bandwidth usage report taken from the firewall after applying the rule in the firewall with QoS as “qos_bandwidth” policy. The figure clearly shows the flooding of the Campus LAN is restricted and the received and the transmitted bits which was causing the choke of LAN bandwidth is removed. The Snapshots given at Fig 7 clearly shows the inbound traffic generated from the Private IPs only which is true/ideal in case of a campus LAN as all users of the campus are configured with private IP address. The snapshot further elaborates the “Packets Received by the Firewall /filter” and the “Packets dropped by the Kernel of the Firewall”. It can be seen from the figure that zero packets have been dropped by the kernel.



| | Max | Min | Average | Current |
|---|---|---|---|---|
| Received KBits/Sec | 65786.31 | 403.52 | 19146.29 | 3634.73 |
| Transmitted KBits/Sec | 66236.37 | 1073.27 | 19404.15 | 22863.62 |

Fig 6: Limiting the DDoS attack trough QOS(Bandwidth) limiting rule from firewall.


Telnet 172.16.1.1

```
 win 108, length 1460
17:45:42.256619 PortA, IN: 00:07:0e:64:c3:7f > 00:90:fb:2c:60:a2, ethertype IPv4
 (0x0800), length 60: 172.16.43.14.64691 > 154.46.33.65.80: Flags [.], ack 17520
, win 4380, length 0
17:45:42.256741 PortA, IN: 00:07:0e:64:c3:7f > 00:90:fb:2c:60:a2, ethertype IPv4
 (0x0800), length 60: 172.16.43.14.63402 > 188.138.1.132.80: Flags [.], ack 7300
, win 60590, length 0
17:45:42.256766 PortA, IN: 00:07:0e:64:c3:7f > 00:90:fb:2c:60:a2, ethertype IPv4
 (0x0800), length 60: 172.16.43.14.63471 > 108.62.115.138.80: Flags [F.], seq 25
78590876, ack 2016776256, win 4146, length 0
17:45:42.256867 PortA, IN: 00:07:0e:64:c3:7f > 00:90:fb:2c:60:a2, ethertype IPv4
 (0x0800), length 60: 172.16.43.14.64834 > 154.46.34.27.80: Flags [.], ack 14601
, win 20805, length 0
17:45:42.257965 PortA, IN: 00:07:0e:64:c3:7f > 00:90:fb:2c:60:a2, ethertype IPv4
 (0x0800), length 60: 172.16.43.14.64834 > 154.46.34.27.80: Flags [.], ack 17521
, win 20805, length 0
17:45:42.258504 PortA, OUT: 00:90:fb:2c:60:a2 > 00:07:0e:64:c3:7f, ethertype IPv
4 (0x0800), length 1514: 154.46.33.65.80 > 172.16.43.14.64692: Flags [.], ack 1,
 win 108, length 1460
17:45:42.258507 PortA, OUT: 00:90:fb:2c:60:a2 > 00:07:0e:64:c3:7f, ethertype IPv
4 (0x0800), length 1514: 154.46.33.65.80 > 172.16.43.14.64692: Flags [.], ack 1,
 win 108, length 1460
17:45:42.258509 PortA, OUT: 00:90:fb:2c:60:a2 > 00:07:0e:64:c3:7f, ethertype IPv
4 (0x0800), length 1514: 154.46.33.65.80 > 172.16.43.14.64692: Flags [.], ack 1,
 win 108, length 1460
17:45:42.259066 PortA, IN: 00:07:0e:64:c3:7f > 00:90:fb:2c:60:a2, ethertype IPv4
 (0x0800), length 60: 172.16.43.14.64834 > 154.46.34.27.80: Flags [.], ack 20441
, win 20805, length 0
17:45:42.260167 PortA, IN: 00:07:0e:64:c3:7f > 00:90:fb:2c:60:a2, ethertype IPv4
 (0x0800), length 60: 172.16.43.14.64692 > 154.46.33.65.80: Flags [.], ack 43800
, win 21170, length 0
17:45:42.261440 PortA, OUT: 00:90:fb:2c:60:a2 > 00:07:0e:64:c3:7f, ethertype IPv
4 (0x0800), length 1514: 154.46.33.65.80 > 172.16.43.14.64692: Flags [.], ack 1,
 win 108, length 1460
17:45:42.261446 PortA, OUT: 00:90:fb:2c:60:a2 > 00:07:0e:64:c3:7f, ethertype IPv
4 (0x0800), length 1514: 154.46.33.65.80 > 172.16.43.14.64692: Flags [.], ack 1,
 win 108, length 1460
17:45:42.261449 PortA, OUT: 00:90:fb:2c:60:a2 > 00:07:0e:64:c3:7f, ethertype IPv
4 (0x0800), length 1514: 154.46.33.65.80 > 172.16.43.14.64692: Flags [.], ack 1,
 win 108, length 1460
17:45:42.261452 PortA, OUT: 00:90:fb:2c:60:a2 > 00:07:0e:64:c3:7f, ethertype IPv
4 (0x0800), length 55: 50.23.65.45.1935 > 172.16.43.12.51419: Flags [.], ack 1,
win 3559, length 1
17:45:42.261455 PortA, OUT: 00:90:fb:2c:60:a2 > 00:07:0e:64:c3:7f, ethertype IPv
4 (0x0800), length 1514: 154.46.34.27.80 > 172.16.43.14.64804: Flags [P.], ack 1
, win 112, length 1460
17:45:42.261517 PortA, IN: 00:07:0e:64:c3:7f > 00:90:fb:2c:60:a2, ethertype IPv4
 (0x0800), length 60: 172.16.43.14.64692 > 154.46.33.65.80: Flags [.], ack 46720
, win 21170, length 0
17:45:42.262504 PortA, OUT: 00:90:fb:2c:60:a2 > 00:07:0e:64:c3:7f, ethertype IPv
4 (0x0800), length 1514: 188.138.1.132.80 > 172.16.43.14.63404: Flags [.], ack 1
, win 118, length 1460
^C
1776 packets captured
1779 packets received by filter
0 packets dropped by kernel
console>
```

Fig 7: Network Packet statistics after application of the qos_bandwidth policy from Firewall.

## ACKNOWLEDGMENTS

We extend our thanks to the Cyberoam Support Team at Ahmedabad, India.

## REFERENCES

[1] http://en.wikipedia.org/wiki/Local_area_network. visited 26th July 2013 .

[2] Christensen, K. J., "Local Area Networks-evolving from shared to switched access" IBM Systems Journal v34 n3 (`95) p347-74 .

[3] Zhu, M. and Molle, M., “Design and Implementation of Application-based Secure VLAN”, Proceedings of the 29th Annual IEEE International Conference on Local Computer Networks (LCN’04), November 2004.

[4] "RFC 4987 – TCP SYN Flooding Attacks and Common Mitigations". Tools.ietf.org. visited 26th July 2013.

[5] https://en.wikipedia.org/wiki/SYN_flood visited 26th July 2013 .

[6] The Internet Protocol Journal - Volume 7, Number 4 Distributed Denial of Service Attacks Patrikakis, C. Masikos, M. and Zouraraki , O. National Technical University of Athens.

[7] Kevin Tsui, "Tutorial-Virus (Malicious Agents)," University of Calgary, October 2001.

[8] Nicholas Weaver, "Warhol Worms: The Potential for Very Fast Internet Plagues," http://www.iwar.org.uk/comsec/resources/worms/warhol-worm.htm

[9] Nicholas Weaver, U.C. Berkeley BRASS group, "Potential Strategies for High Speed Active Worms: A Worst Case Analysis," February 2002.

[10] Moore,D. and Shannon,C. "The Spread of the Code Red Worm (crv2)," July 2001, http://www.caida.org/analysis/security/codered/coderedv2_analysis.xml#animations

[11] Deng, G. and Huang, S. “Case Study on Incentive Mechanism and Its Effectiveness Of Campus Network in Virtual Community----Case of Myclub2.com”, 978-1-4244-6359-6/10/$26.00 ©2010 IEEE

[12] Ramamoorthi, A., Subbulakshmi, T. and Mercy Shalinie, S.; “Real Time Detection and Classification of DDoS Attacks using Enhanced SVM with String Kernels”, IEEE-International Conference on Recent Trends in Information Technology, ICRTIT 2011, 978-1-4577-0590-8/11/$26.00 ©2011 IEEE, MIT, Anna University, Chennai. June 3-5, 2011

[13] Zhijun WU; WANG, C. and ZENG, H.; “Research on the Comparison of Flood DDoS and Low-rate DDoS”, 978-1-61284-774-0/11/$26.00 ©2011 IEEE.

[14] Renuka Devi, S. and Yogesh, P., “An Effective Approach to Counter Application Layer DDoS Attacks”, ICCCNT'12, IEEE-20180, 26th_28th July 2012, Coimbatore, India