Detecting and Alerting TCP –IP Packets againt TCP SYN attacks

Parasa Harika[#1],Mrs D.Raaga Vamsi[#2]
[1] *M.Tech(CSE),Gudlavalleru engineering college,Gudlavalleru.*
[2] *Assistant professor, Gudlavalleru engineering college,Gudlavalleru.*

ABSTRACT:
Transmission Control Protocol Synchronized (TCP SYN) Flood has become a problem to the network management to maintain the network server from being attacked by the malicious attackers. Possibly one of the problems in detecting SYN Flood is that hosting server nodes or firewalls cannot distinguish the SYN packets of normal TCP connections from your of SYN Flood attack. Moreover, considering that the fee of normal network traffic differentiates, we are unable to work with an explicit threshold of SYN arrival rates out to detect SYN Flood traffic. Taking into consideration time period variant of arrival traffic. We first investigate the status of the arrival rates of both normal TCP SYN packets and SYN Flood attack packets. Our new detection mechanism based on the stats of SYN arrival rates. Our proposed mechanism can detect SYN Flood traffic quickly and precisely despite time variance of the traffic. Experimental results show that the proposed detection method using the combination of packet filtering and syn flood based traffic monitoring can detect TCP SYN Flood in the network and alerts are sent to the administrator through e-mail mechanism.

I INTRODUCTION

DOS attacks are not hard to be performed by utilizing the weakness of one's network protocol and by iterating requests of service when it comes to the application. Most organizations have opened their Web sites and also other ports on TCP to maintain their sites. So, these attacks aim at directly the application of the Web server or TCP protocol to suspend their Internet services. Because we are part of a DDoS (Distributed Denial of Service) attack, the assault is coordinated across many hijacked systems using a single attacker[1]. SYN flood attacks [2][3] disturb the establishment of the TCP connection. An attacker does not respond to the SYN ACK packet direct from server participating in a huge amount of SYN packets sent to the server due to an attacker with spoofed source IP addresses. Subsequently, TCP by the server should keep the big number of the half-open state each connection and exhaust the memory resource to be followed by the cease of server function to be eventually down. This SYN flood attack has been widely observed world-wide, and occupies approximately 90% of this very DoS attack [4]. The mechanism to avoid these flooding-based DoS/DDoS attacks is generally classified into three types; source-end defense, network-path based defense and victim-end defense, utilizing network distance from attackers. The source-end defense aims to detect the presence of attack traffics at source networks, especially for the entry point of one provider, even then it is hard to distinguish attack flows in source networks. The network-path based defense really should be to identify the Internet paths traversed by attack flows regardless of the trouble to look for the path, because the source IP address of the attack packet is usually spoofed. The reason for a victim-end usually is to identify and block flooding-based DDoS packets along at the victim, but no explicit attack pattern can be used to separate attack packets from legitimate ones. SYN flooding can be one of the preferred techniques for Dos (Denial of Service) and DDos(Distributed Denial of Service) attack, which is the reason 90% among. The SYN flooding attacks exploit TCP's three-way handshake mechanism and also its limitation in maintaining half-open connections, and so are performed by the attacker submitting a stream of TCP SYN (connection request) packets in the target system, filling its connection request queue, and as a consequence denying legitimate users' access to the target system [1]. Several approaches appear to have been proposed to defend the SYN flooding attacks, such as Random Drop, SYN Cookie, Restriction of Bandwidth, Reverse path forwarding and so on. But these approaches have not had a good effect on it, and without detecting efficiently they easily cause multiple problems like paralysis of firewall, paralysis of router and denying legitimate users' access.
Distributed Denial-of-Service (DDoS) attack remains a bad problem on the Internet, since it takes advantage of the death of authenticity in the IP protocol, destination oriented routing, and stateless nature of the Internet. Among various DDoS attacks, TCP SYN flooding will be the most commonly used one and yet dominates DDoS attacks as documented in the recent NANOG report in 2011. There are two common sizes reasons for SYN flood attacks. The very first is the inherent asymmetry feature in TCP three-way handshake protocol, which lets the attacker to access substantial resources along at the server, while sparing its own resources. The other is the server cannot control the packets it receives, particularly the SYN packets can possibly reach the server without its approvement. Fig. 1 illustrates the three-way handshake protocol:

1) Litigant sends a SYN packet to some server to complete a live open request;
2) The server reserves connection resources (backlog queue) to follow the TCP state on receiving a SYN packet and replies utilizing a SYN/ACK packet in response;
3) Finally, the client sends an ACK returning to the server as a possible acknowledgement, and the connection is established when receiving this ACK upon the server side. We call the ACK packet in the whole third step as CliACK. During SYN flood attacks, an attacker generates numerous SYN requests but never sends the CliACK packets to finish up the connections. The victim server's backlog queue often is easily exhausted and most of the new incoming SYN requests are dropped. Furthermore, other system resources like network bandwidth are occupied.

II BACKGROUND AND RELATED WORK

Monitoring and classification of real traffic

According to **Ohsita,framework** deployed a traffic monitor at the gateway of Osaka University. They used an optical-splitter to split the 1000 Base-SX fiber-optic cable and recorded the headers of all of the packets transferred on this link. That is, they monitored all the packets in both the inbound and outbound directions at Osaka University. We use tcpdump [13] to read the headers of packets. Although tcpdump cannot necessarily read the headers of all packets at wire-speed, confirmed that the headers of less than 0.01% of the packets were not recorded and these losses did not affect the results of our statistical analysis.They first classified monitored packets into *flows*. We defined a series of packets which have the same (src IP, src port, dest IP, dest port, protocol) fields as a single *flow* and we classify these *flows* into the following five groups. Group N Flows that completed the 3-way handshake and were closed normally by an FIN or RST packet at the end of connections. Group Rs Flows terminated by a RST packet before a SYN/ACK packet was received from the destination host. These flows were terminated this way because the destination host was not available for the service specified in the SYN request. Group Ra Flows terminated by a RST packet before an ACK packet for the SYN/ACK packet was received. These flows were terminated this way because the SYN/ACK packets were sent to a host that was not on the Internet. Group Ts Flows containing only SYN packets. These flows are not terminated explicitly (i.e., by RST/FIN packets) but by the timeout of flows. There were three reasons that flows could be classified into this group. One was that, the destination node did not respond the SYN packet. A second was that the source address of the SYN packet was spoofed and the destination sent the SYN/ACK packet to the spoofed address. The third was that all of the SYN/ACK packets were discarded by the network (e.g., because of due to e.g., network congestion). Group Ta Flows containing only SYN and its SYN/ACK packets. Like Group Ts flows, these flows were terminated by the timeout of flows. In this case, however, it was because all the ACK packets were dropped. To identify the traffic of normal flows, we focused on the Group N flows. Hereafter, we refer these flows as *normal traffic* and to Groups Rs, Rs, Ts and Ta flows as *incomplete traffic*.

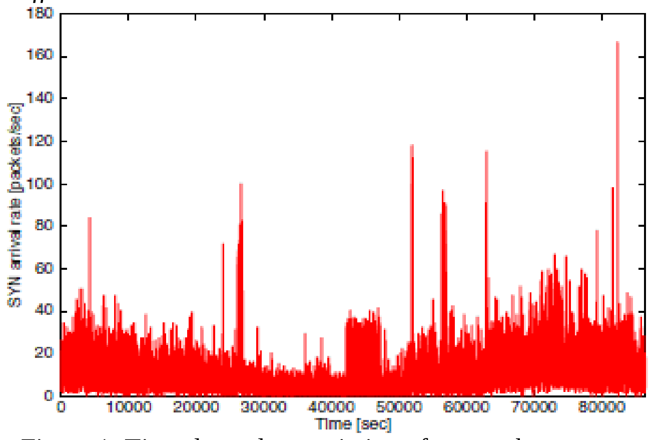


Figure 1; Time dependent variation of syn packets.

Detecting a SYN Attack

It is very simple to detect SYN attacks. The net stat command shows us how many
connections are currently in the half-open state. The half-open state is described as SYN_RECEIVED in windows and as SYN_RECV in UNIX systems.

```
#netstat –n –p TCP
tcp   0   0   10.100.0.200:21   237.177.154.8:25882
SYN_RECV
tcp   0   0   10.100.0.200:21   236.15.133.204:2577
SYN_RECV
tcp  0  0  10.100.0.200:21  127  160.6.129:51748
SYN_RECV
tcp   0   0   10.100.0.200:21   230.220.13.25:47393
SYN_RECV
tcp0   0   10.100.0.200:21   227.200.204.182:60427
SYN_RECV
tcp   0   0   10.100.0.200:21   232.115.18.38:278
SYN_RECV
tcp   0   0   10.100.0.200:21   229.116.95.96:5122
SYN_RECV
tcp00      10.100.0.200:21      236.219.139.20749162
SYN_RECV
tcp0   0   10.100.0.200:21   238.100.72.228:37899
SYN_RECV
```

```
# netstat -n -p TCP | grep SYN_RECV | grep :23 | wc
l 769
```

The other method for detecting SYN attacks is to print TCP statistics and look at the TCP
parameters which count dropped connection requests. While under attack, the values of these parameters grow rapidly. In this example we notice the value of the TcpHalfOpenDrop parameter.

```
# netstat -s -P tcp | grep tcpHalfOpenDrop
```

tcpHalfOpenDrop = 473
It is important to note that every TCP port has its own backlog queue, but only one variable of the TCP/IP stack controls the size of backlog queues for all ports.

Operating System: Windows2000

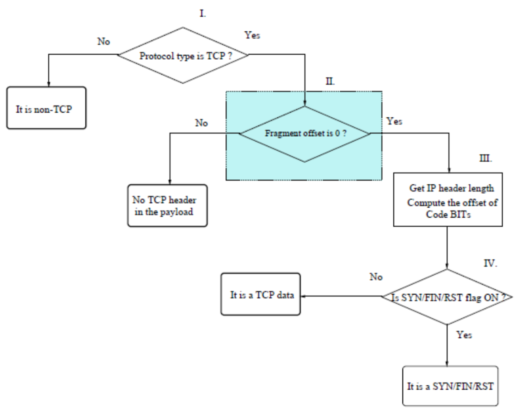
The foremost large parameter in Windows 2000 in addition to in Windows Server 2003 is SynAttackProtect. Letting this parameter allows the operating system to firmly take care of incoming affiliates effortlessly. The security can certainly be set up by introducing a SynAttackProtect DWORD regard in the next few registry key:

HKLM\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters.

Normally, when a SYN attack is detected the SynAttackProtect parameter changes the behavior of one's TCP/IP. This lets the operating system out to take care of more SYN requests. It works by disabling a few plug options, incorporating additional holds up to connection indications and changing the right time out for connection requests. Every time the value of SynAttackProtect serves out to 1.

3. PROPOSED FRAMEWORK

In SYN floods, attacker would send an instant barrage of SYN packets from IP addresses (often spoofed) that will not generate replies to the SYN/ACKs. To remain effective, attacker needs to send new barrages of bogus connection requests frequently. Almost all the SYN flooding packets will not be retransmitted. Then again, Should a legitimate client's SYN packet is lost, it'd retransmit the SYN packet many times before dropping. Our mitigation scheme utilizes the characteristic of SYN floods and client's persistence. We use three counting filters [1] to record related information: • C-1: to record the initial SYN packets of each connection;



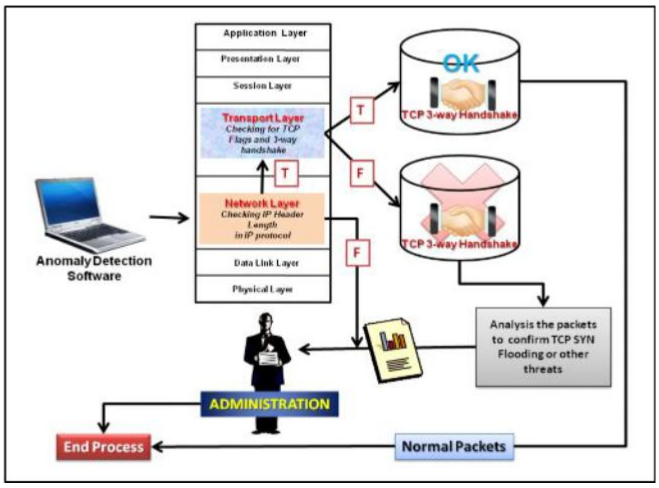
• C-2: to record the SYN packets, whose connections have completed the three-way handshake?
• C-3: to record the opposite SYN packets.

The mitigation scheme starts working once detecting SYN floods. If a SYN packet is received, its 4-tuple is extracted as an item and queried from the three Cs. The results are: 1)The item is not in any of the three Cs. This TCP connection is new, and then we drop this SYN packet and insert the item to C-1; 2)The item is in C-1. This is the second SYN packet. We pass it and move the item from C-1 to C-3; 3)The item is in C-2. We pass the packet;
4)The item is in C-3. We pass the packet with a certain probability p. We insert the item to C-3 and obtain the number, n, of this item in C-3. Let p = 1/n, then P is smaller as the increasing of n.

Spoofed steps:

for each packet:
extract the final TTL 1;
extract source IP address S;
Find Initial TTL 1;
Find Initial TTL 2 use to extract stored *Hs* from IP mapping table;
if (Hc != Hs)

the packet is spoofed;
else
the packet is legitimate;
There may be 3 (three)
kinds of network analysis behavior: · It used protocol to detect packets which can be found quite short which violate specific application layers protocol. · Rate-based detection which detects floods in traffic making use of a time-based model of normal traffic volumes especially DoS attacks.
It detects throughout the behavioral or relational changes in how individual or sets of hosts learn one other on any network.
Algorithm for Packet Filtering System:


Application Layer
Presentation Layer
Session Layer
Transport Layer Checking for TCP Flags and 3-way handshake
Network Layer Checking IP Header Length in IP protocol
Data Link Layer
Physical Layer
T
F
T
F
OK
TCP 3-way Handshake
TCP 3-way Handshake
Anomaly Detection Software
Analysis the packets to confirm TCP SYN Flooding or other threats
ADMINISTRATION
End Process
Normal Packets

For each packet arrival do

Check the IP Header=20 then

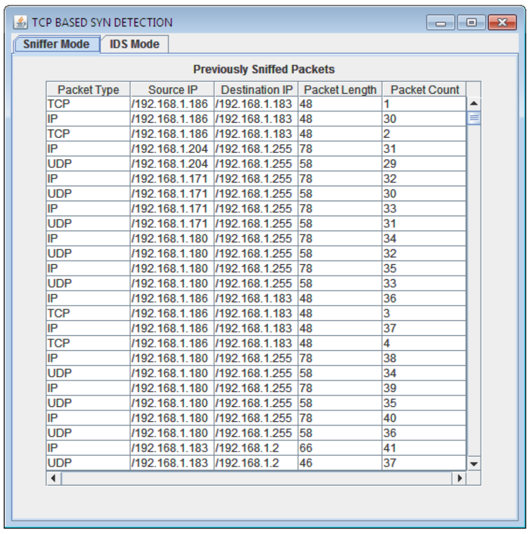
Choose the protocol=TCP or UDP or Http

Check the payload packet

```
If payload normal
Goto destination
Else distinguish the packet for analysis
If TCP SYN flood then
Report alert mail to admin
Else analysis for other threats
End if
End if
Other  protocol
Goto destination
Endif
End for
```

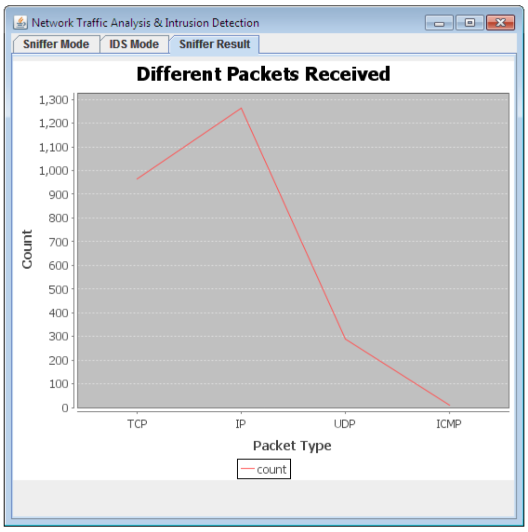
5. EXPERIMENTAL RESULTS

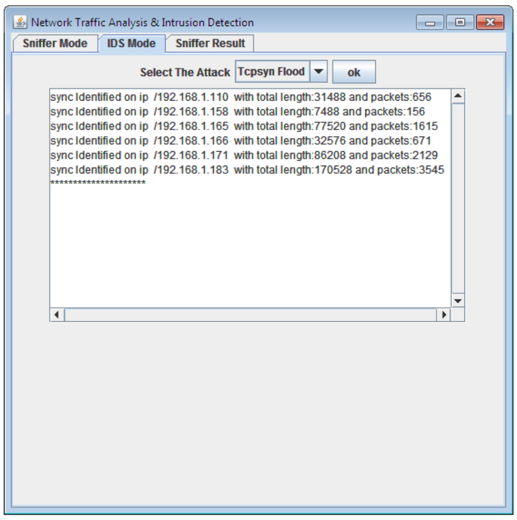
All experiments were performed with the configurations Intel(R) Core(TM)2 CPU 2.13GHz, 2 GB RAM, and the operating system platform is Microsoft Windows XP Professional (SP2).



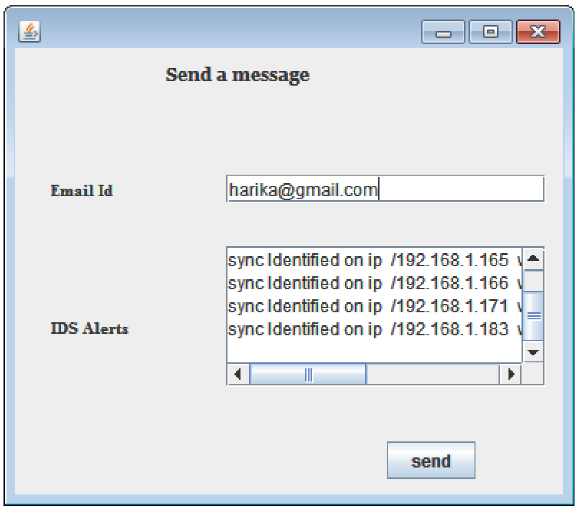
Previously Sniffed Packets

Packet Type	Source IP	Destination IP	Packet Length	Packet Count
TCP	/192.168.1.186	/192.168.1.183	48	1
IP	/192.168.1.186	/192.168.1.183	48	30
TCP	/192.168.1.186	/192.168.1.183	48	2
IP	/192.168.1.204	/192.168.1.255	78	31
UDP	/192.168.1.204	/192.168.1.255	58	29
IP	/192.168.1.171	/192.168.1.255	78	32
UDP	/192.168.1.171	/192.168.1.255	58	30
IP	/192.168.1.171	/192.168.1.255	78	33
UDP	/192.168.1.171	/192.168.1.255	58	31
IP	/192.168.1.180	/192.168.1.255	78	34
UDP	/192.168.1.180	/192.168.1.255	58	32
IP	/192.168.1.180	/192.168.1.255	78	35
UDP	/192.168.1.180	/192.168.1.255	58	33
IP	/192.168.1.186	/192.168.1.183	48	36
TCP	/192.168.1.186	/192.168.1.183	48	3
IP	/192.168.1.186	/192.168.1.183	48	37
TCP	/192.168.1.186	/192.168.1.183	48	4
IP	/192.168.1.180	/192.168.1.255	78	38
UDP	/192.168.1.180	/192.168.1.255	58	34
IP	/192.168.1.180	/192.168.1.255	78	39
UDP	/192.168.1.180	/192.168.1.255	58	35
IP	/192.168.1.180	/192.168.1.255	78	40
UDP	/192.168.1.180	/192.168.1.255	58	36
IP	/192.168.1.183	/192.168.1.2	66	41
UDP	/192.168.1.183	/192.168.1.2	46	37





```
sync Identified on ip  /192.168.1.110  with total length:31488 and packets:656
sync Identified on ip  /192.168.1.158  with total length:7488 and packets:156
sync Identified on ip  /192.168.1.165  with total length:77520 and packets:1615
sync Identified on ip  /192.168.1.166  with total length:32576 and packets:671
sync Identified on ip  /192.168.1.171  with total length:86208 and packets:2129
sync Identified on ip  /192.168.1.183  with total length:170528 and packets:3545
*********************
```



Email Id: harika@gmail.com

IDS Alerts:

```
sync Identified on ip  /192.168.1.165
sync Identified on ip  /192.168.1.166
sync Identified on ip  /192.168.1.171
sync Identified on ip  /192.168.1.183
```

6. CONCLUSION AND FUTURE WORK

Security management operations protect computer networks against denial-of-service attacks, unauthorized disclosure of information, and the modification or destruction of data. Moreover, the automated detection and immediate reporting of these events are required in order to provide the basis for a timely response for attack.This system effectively detects the Syn flood type of attacks with email alerts. The Future scope of the project include identifying more attacks and taking an action to stop the attack by closing the connection or reporting the incident for further analysis by network administrator through routers and firewalls.

REFERENCES

[1] “Information, Computer and Network Security Terms GlossaryandDictionary,”http://www.javvin.com/networksecurity/SignatureDetection.html

[2] D. Whyte, E. Kranakis, and P. Van Oorschot, “DNS-based Detection of Scanning Worms in an Enterprise Network,” Proceeding of the Network and Distributed Systems Symposium (NDSS), 2005.

[3] P. Barford, J. Kline, D. Plonka, and R. Amos, "A Signal Analysis of Network Traffic Anomalies," Proceeding of the ACM SIGCOMM Internet Measurement Workshop, Marseilles, France, November 2002.

[4] M. Basseville, and I. V. Nikiforov, Detection of Abrupt Changes: Theory and Application, Prentice Hall, 1993.

[5] Mahoney, M, and P.K. Chan, PHAD: Packet Header Anomaly Detection for Identifying Hostile Network Traffic, Florida Tech. Technical Report (2001-04).

[6] Mahoney, M., and P. K. Chan, Learning Nonstationary Models of Normal Network Traffic for Detecting Novel Attacks, in Proc. SIGKDD, pp 376-385, 2002

[7] Mahoney, M, Network Traffic Anomaly Detection Based on Packet, in Proc. Conference ACM , 2003.

[8] G.Yang, Introduction to TCP/IP Network Attacks, Iowa University Nov. 1997

[9] Denial of Service and Distributed Denial of Service Protection, white paper by 3com Corporation, 2005.

[10] Y. Ohsita, S. Ata, M. Murata, "Detecting Distributed Denial-ofService Attacks by Analyzing TCP SYN Packets Statistically", Proc. IEEE Communications Society Globecom, pp. 2043-2049 , 2004.

[11] M. Bellaice, J.C. Gregoire, "Source Detection of SYN Flooding Attacks", ESR Group, 2009.