Detection and Performance Evaluation of DoS/DDoS Attacks using SYN Flooding Attacks

Karthik Pai B.H.
Asst. Professor
NMAMIT
Nitte-574110

Nagesh H.R.,Ph.D.
Professor &Head
MITE
Moodabidri-574227

Abhijit Bhat
II Year M.Tech.
NMAMIT
Nitte-574110

ABSTRACT
One of the biggest concerns for security professionals today are Distributed Denial of Service (DDoS) flooding attacks. They are nothing but explicit attempts to disrupt the legitimate users' access to services. One of the more popular DDoS attack is the SYN Flood attack. The SYN flooding attacks are launched by exploiting the TCP's three-way handshake mechanism and its limitation in maintaining its half-opened connections. The proposal is to present a simple and robust mechanism that detects the SYN flooding attacks with less computational overhead. The two algorithms which would be used are an adaptive threshold algorithm and the cumulative sum (CUSUM) algorithm for change point detection. The proposal is to measure the performance in terms of the packet delivery fraction. The evaluation results are presented in NS2 simulation environment**.**

General Terms
DDoS, SYN Flood.

Keywords
CUSUM algorithm, NS2

1. INTRODUCTION
1.1 The Attacks
Distributed Denial-of-Service(DDoS) attacks have been around for so many years now . The basic aim of these attacks is to crash, hang up, or overwhelm servers with malformed packets or large volumes of traffic by using various techniques.Many DDoS flooding attacks have been launched against various organizations like Yahoo![18], SCO Group, Mastercard, Paypal, Visa etc. and many more. [17][19][21][22][23][24]. One of the more popular and a clever way of launching a DDoS attack is the SYN Flood attack which exploits the normal behaviour of a TCP connection.

1.2 SYN Flood
It has been made known that over 85% of the DoS attacks use TCP [28]. The TCP SYN flood is the most commonly-used attack.

The basis of the SYN flooding attack is in its design of the 3-way handshake that initiates a TCP connection. In the handshake, the ability of the initiator to receive packets at the IP address which it uses as the source in its initial request, or its reachability to return is verified by the third packet. Figure1 depicts the order of packets exchanged at the start of a TCP connection.

The Transmission Control Block (TCB) is a transport protocol data structure which holds all the information regarding a connection. Almost every single TCB exceeds a minimum of 280 bytes, and in some operating systems, it takes more than 1300 bytes. The TCP SYN-RECEIVED state indicates that the connection is only half open, and the validity of the request is still under consideration. The allocation of TCB is based on reception of the SYN packet, i.e., either before the connection is fully established or the initiator's return reachability has been verified.

As a result, this kind of situation leads to a clear possibility of a DoS attack where incoming SYNs result in many TCBs being cause allocated and the host's kernel memory is exhausted. To avoid this exhaustion of memory, most operating systems generally align a "backlog" parameter along with a listening socket that sets a limit on the number of TCBs simultaneously in the SYN-RECEIVED state. While this action may protect a host's available memory resource from attack, the backlog itself represents another (smaller) resource vulnerable to attack. With no space left in the backlog, it is not possible to service new connection requests until some TCBs can be reset or removed from the SYN RECEIVED state.

The aim of the TCP SYN flooding attack is to deplete the backlog of the victim server by attempting to send SYN segments enough to fill the entire backlog. The attacker uses source IP addresses in the SYNs that are non-existent or non-functional, i.e., those IP addresses that are unlikely to trigger any response that would free the TCBs from the SYN-RECEIVED state. Since the TCP attempts to be reliable, the target host keeps its TCBs blocked in SYN-RECEIVED state for a relatively long time before giving up and resetting the half opened connection. Meanwhile, services are denied to the application processes on the listener for legitimate new TCP connection initiation requests. Figure2 depicts a simplification of the series of events involved in a TCP SYN flooding attack.

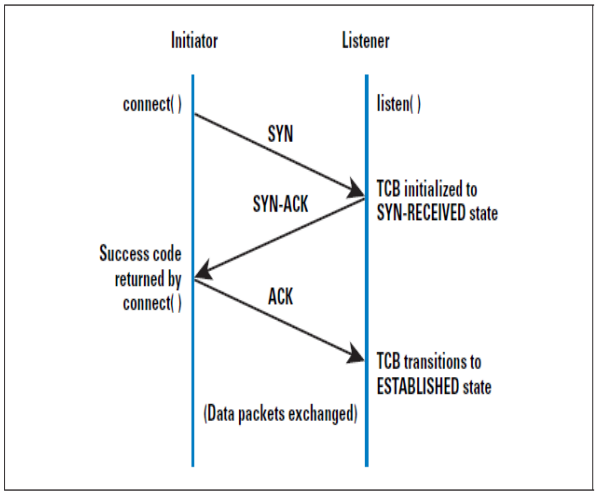

Fig 1: The 3-way TCP Handshake

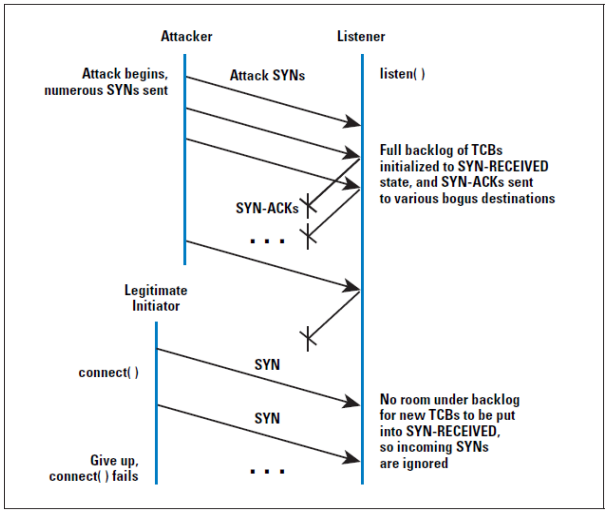

Fig 2: SYN Flood Attack

2. COUNTER MECHANISM

To counter SYN flooding attacks, several defense mechanisms have been proposed, such as Syn cache [17], Syncookies[3], SynDefender [6], Synproxying [20], and Synkill [26].All of these defense mechanisms are installed at the firewall of the victim server or inside the victim server, thereby providing no hints about the sources of the SYN flooding. They have to rely on the expensive IP traceback [2], [21], [25], [28], [29],[34] to locate the flooding sources. Because the defense line is at, or close to, the victim,the network resources are also wasted by transmitting the flooding packets.

Moreover, these defense mechanisms are stateful, i.e., states are maintained for each TCP connection or state computation is required. Such a solution makes the defense mechanism itself vulnerable to SYN flooding attacks. Recent experiments have shown that a specialized firewall, which is designed to control SYN floods, became incapable under a flood of 14,000 packets per second [8]. The stateful defense mechanisms also degrade the end-to-end TCP performance, e.g., incurring longer delays in setting up connections. In the absence of SYN flooding attacks, all the overheads introduced by the defense mechanism become superfluous. We, therefore, need a simple stateless mechanism to detect SYN flooding attacks, which is immune to the SYN flooding attacks. Also, it is preferred to detect an attack early near its source, so that one can easily trace the flooding source without resorting to expensive IP traceback.

Related Work

So far, many methods have been proposed to detect the SYN flood attacks. In [3], the authors propose a linear prediction analysis as a new paradigm for DoS SYN flood attack detection. This method is used at leaf routers and firewalls without the need of maintaining any state. They use the exponential backoff property of TCP used during timeouts. By modelling the difference of SYN and SYN-ACK packets, they show that this approach is able to detect attacks within small delays. However, given the fact that the sources of attack can be spread in different networks, there is a clear lack of analysis for the traffic near the sources. Also the detection of the source of attack in TCP-based low intensity attacks is missing.

In [4], the authors consider a non-parametric cumulative sum algorithm to measure the count of only SYN packets. They also use an EWMA(Exponential Weighted Moving Average) for getting a recent approximation of the mean rate after the change of SYN packets.

In [5] three counters algorithms for SYN flooding defense attacks are proposed. The detection scheme uses the inherent TCP valid SYN–FIN pairs behaviour to detect the various SYN flood attacks with high accuracy and short response time. The mitigation scheme, also proposed amongst the three schemes, works in highly efficient manner for the victim to detect the SYN packets during the attack. However, the drawback is that the attackers may retransmit every SYN packet more than one time to destroy the mitigation scheme. It is necessary to make it more robust and adaptive despite the schemes being stateless and requiring low computational overhead.

Moreover, there are also some other related studies such as SYN cookies, SYN filtering mechanisms [11], SYN cache, SYN proxy (firewall), SYN kill, D-SAT [12] and DiDDeM ([13] and [14]), and more related studies is in [15], [16], [17] and [18].

In the [15] and [16], the authors propose an early stage detecting method (ESDM). Here, the SYN traffic is forecasted by autoregressive integrated moving average model, and a non-parametric CUSUM algorithm is used to find the SYN flooding attacks. The ESDM achieves shorter detection time and small storage space. However, most of these exiting methods or defense mechanisms which oppose to the SYN flooding attack are effective only at the later stages, when attacking signs are obvious [16].

3. ALGORITHMS USED

3.1 Adaptive Threshold Algorithm

This is simple algorithm[2] which measures network traffic (in our case SYN packets) and compare it with previously defined threshold. This threshold is adaptively set in certain period of time and is based on the estimated mean number of SYN packets. If measured traffic exceeds a particular threshold it will be defined as anomaly and alarm will be activated.

Let us suppose that the number of SYN packet in the n-th time interval is x_n, and measured mean rate prior to n is μ_{n-1}. In this case the alarm condition is as following:

If $\mathbf{x_n >= (\alpha+1)\mu_{n-1}}$, then ALARM signalled at time n, where $\alpha > 0$ parameter indicates the percentage above mean value which is considered as a threshold. The mean μ_n can be computed over some past time window or using an exponentially weighted moving average (EWMA) of previous measurements

$$\mu_n = \beta\mu_{n-1} + (1-\beta)\, x_n$$

Where β is EWMA factor.

On the arrival of packets the algorithm checks if the packets are TCP, and if this is True it checks if TCP packets are SYN or other type of TCP. If the SYN bit is on than it will update the record. Calculating the μ and α rate than it will check the condition for xn either it is greater than (α + 1) μn-1 or not. If it is greater it will generate the alarm.

3.2 CUSUM Algorithm

Cumulative sum (CUSUM) algorithm is used in the quality control. They are well suited for checking a measuring system in operation for any departure from some target or specified values and have been widely used for detecting the small and moderate mean shifts.

In this paper, we focus on the use of non-parametric CUSUM [12] to detect TCP SYN flooding attacks. In the context of detecting SYN flooding attacks, for each SYN packet, CUSUM monitors a set of n SYN packet sample interval $\{y_1 \ldots y_n\}$ where y_n is the sum of all SYN packets in *n-th* sample interval (detection interval). Assume that the change SYN traffic $\{y_i\}$ is an independent normal distribution and μ_0 and μ_1 are the mean SYN traffic before and after the change.

A factor that needs to be addressed is the value of μ_1. i.e., the mean number of SYN packet after the change. Since this cannot be known beforehand, we approximate it with $\alpha\mu_n$, where α is the percentage parameter, which corresponds to the most probable percentage of increase of the mean number of SYN packet after a change.

Then CUSUM can be written as:

$f_n = [f_{n-1} + (\alpha\mu_{n-1}/\sigma^2)(x_n - \mu_{n-1}(\alpha\mu_{n-1}/2))]$

4. SIMULATION

We use NS2 as the simulation tool. Figure 3 shows the model for attacking simulation. In the figure each node is representing a system in the internet; we take two nodes each for source and destination nodes. Node 15 and node 19 are source nodes, node 24 and node 28 are destination nodes and node 21 represents the attacker node. The various parameter values are given below. The simulation is conducted over a period of 10 seconds.

Table 1: Parameter Values

Parameter	Value
Simulator	NS 2.34
Simulator Area	1000m x 700m
No of nodes	40
Packet size	512 bytes
Movement Model	Random
Data Rate	5 Mbps
Simulation Time	10 sec

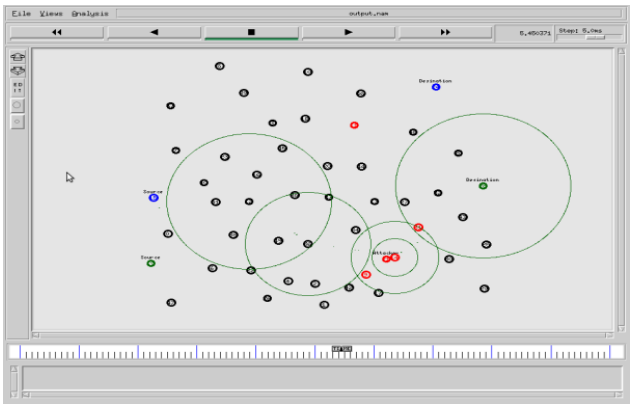

Fig 3: Screenshot of the scenario of the implement CUSUM algorithm

Simulation is started and data transfer takes place between source and destination. We then introduce some attacker nodes to implement SYN Flood attacks. The transmission collapses and packets are dropped. The attacks are then detected by the implemented CUSUM algorithm which then proceeds to eliminate the attackers. Normal data transmission is restored until the simulation stops. The simulations are conducted under three scenarios – one with 3 attacker nodes, one with 6-attacker nodes and the other with 9 attacker nodes.

5. RESULTS AND DISCUSSION

The results show that the CUSUM and Adaptive Threshold mechanisms used give a Packet Delivery ratio of 92.25% when simulations are done in the presence of 3 and 88.36% under 6 attacker nodes respectively and around 79.3% when simulated under the presence of 9 attacker nodes. The attacker nodes are introduced at the 3.1th second during which there is a huge drop in the PDF. At the 4.2th second, the implemented CUSUM algorithm detects and eliminates the attackers and normal data transmission is restored. The disadvantage of other existing mechanisms like SYN Cache, SYN Cookies, SYN Defender, SYN proxying, Synkill is that these defense mechanisms are stateful, i.e., states are maintained for each TCP connection or state computation is required. Such a solution makes the defense mechanism itself vulnerable to SYN flooding attacks. The stateful defense mechanisms also degrade the end-to-end TCP performance, e.g., incurring longer delays in setting up connections. In the absence of SYN flooding attacks, all the overheads introduced by the defense mechanism become superfluous.

The two algorithms, CUSUM and Adaptive Threshold algorithms are simple stateless mechanisms to detect SYN flooding attacks, which are also immune to the SYN flooding attacks.

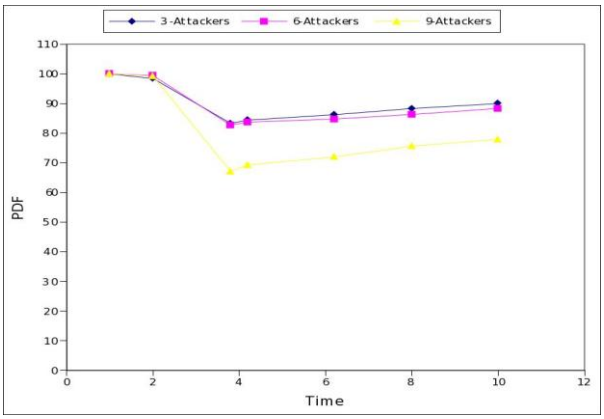

Fig 4: Comparison of Packet Delivery fraction under various attacker node scenarios

6. CONCLUSION AND FUTURE WORK

One of the vulnerabilities of TCP protocol which leads to the SYN flood attack is shown. Effective anomaly detection algorithms against SYN Flood attacks are then presented in NS2 environment and the performance of the algorithms is evaluated using Packet Delivery Fraction as the performance metric. This paper deals with the detection of SYN Flood attacks. Extension of this mechanism to prevent SYN Flood attack can be seen as a possible future work.

7. REFERENCES

[1] H. Wang, D. Zhang, and K. G. Shin, "Detecting SYN flooding attacks", in Proceedings of Annual Joint Conference of the IEEE Computer and Communications Societies(INFOCOM), volume 3, pp. 1530-1539, June 23-27, 2002.

[2] MitkoBogdanoski, Tomislav Shuminoski and Aleksandar Risteski " Analysis of the SYN Flood DoS Attack" I. J. Computer Network and Information Security, 2013, 8, 1-11 Published Online June 2013 in MECS (http://www.mecs-press.org/)DOI: 10.5815/ ijcnis.2013. 08.01.

[3] D. M. Divakaran, H. A. Murthy and T. A. Gonsalves, "Detection of SYN Flooding Attacks Using Linear Prediction Analysis", 14th IEEE International Conference on Networks, ICON 2006, pp. 218-223, Sep. 2006.

[4] V. A. Siris and P. Fotini, "Application of Anomaly Detect Algorithms for Detecting SYN Flooding Attack" *Elsevier Computer Communications*, pp. 1433-1442, 2006.

[5] S.Gavaskar, R.Surendiran and Dr.E.Ramaraj, "Three Counter Defense Mechanism for SYN Flooding Attacks", International Journal of Computer Applications, Volume 6–No.6, pp.12-15, Sep. 2010.

[6] SamanTaghaviZargar, James Joshi and David Tipper, "A Survey of Defense Mechanisms Against Distributed Denial of Service (DDoS) Flooding Attacks.

[7] C. L. Schuba, I. V. Krsul, M. G. Kuhn, E. H. Spafford, A. Sundaram and D. Zamboni, "Analysis of a Denial of Service Attack on TCP", *Proceedingsof IEEE Symposium on Security and Privacy*, May 1997.

[8] T. Nakashima and S. Oshima, "A detective method for SYN flood attacks", *First International Conference on Innovative Computing, Information and Control*, 2006.

[9] D. Nashat,X. Jiang and S. Horiguchi, "Detecting SYN Flooding Agents under Any Type of IP Spoofing", IEEE International Conference on e-Business Engineering table of contents, 2008.

[10] W. Chen and D.-Y. Yeung, "Defending Against SYN Flooding Attacks Under Different Types of IP Spoofing", ICN/ICONS/MCL '06, IEEE Computer Society, pp. 38-44, April 2006.

[11] A. Yaar, A. Perrig and D. Song, "StackPi: New Packet Marking and Filtering Mechanisms for DDoS and IP Spoofing Defense*", IEEE Journal on Selected Areas in Communications*, Volume 24, no. 10, pp. 1853-1863, October 2006.

[12] S.-W. Shin, K-Y.Kim and J.-S. Jang, "D-SAT: detecting SYN flooding attack by two-stage statistical approach", Applications and the Internet, pp.:430 – 436, 2005.

[13] J. Haggerty, T. Berry, Q. Shi and M. Merabti, "DiDDeM: a system for early detection of SYN flood attacks", GLOBECOM, 2004.

[14] J. Haggerty, Q. Shi and M. Merabti, "Early Detection and Prevention of Denial-of-Service Attacks: A Novel Mechanism With Propagated Traced-Back Attack Blocking", IEEE Journal On Selected Areas In Communications, Vol. 23, No. 10, pp. 1994-2002, October 2005.

[15] S. Qibo, W. Shangguang, Y. Danfeng and Y. Fangchun, "An Early Stage Detecting Method against SYN Flooding Attacks", *China Communication*, Vol. 4, pp. 108-116, November 2009.

[16] G. Wei, Y. Gu and Y. Ling, "An Early Stage Detecting Method against SYN Flooding Attack", International Symposium on Computer Science and its Applications, pp.263-268, 2008.

[17] P. J. Criscuolo, Distributed Denial of Service, Tribe Flood Network 2000, and Stacheldraht CIAC-2319, Department of Energy Computer Incident Advisory Capability (CIAC), UCRL-ID-136939, Rev. 1., Lawrence Livermore National Laboratory, February 14, 2000.

[18] Yahoo on Trail of Site Hackers, Wired.com, Feb. 8, 2000,[online]http://www.wired.com/news/business/0, 1367, 34221,0.html.

[19] Powerful Attack Cripples Internet, Oct. 23, 2002, [online] http://www.greenspun.com/bboard/q–and–a–fetch–msg.tcl?msgid=00A7G7

[20] Mydoom lesson: Take proactive steps to prevent DDoS attacks,Feb.6,2004,[online]http://www.computerworld. com/s/article/89932/Mydoom lesson Take proactive steps to prevent DoSattacks? taxonomyId =017.

[21] Lazy Hacker and Little Worm Set Off Cyberwar Frenzy, July 8, 2009,[online] http://www. wired.com / threatlevel /2009 /07/ mydoom/

[22] New "cyber attacks" hit S Korea, July 9, 2009, [online]

http://news.bbc.co.uk/2/hi/asia-pacific/8142282.stm

[23] Operation Payback cripples MasterCard site in revenge for WikiLeaks ban, Dec. 8, 2010, [online] http://www. guardian.co.uk / media/2010/dec/08/operation–payback–mastercard–website–wikileaks

[24] T. Kitten, *DDoS: Lessons from Phase 2 Attacks*, Jan. 14, 2013, [online] http://www.bankinfosecurity.com/ddos-attacks-lessons-from-phase-2-a-54 20/op-1.

[25] L. Garber, "Denial-of-Service Attack Rip the Internet", Computer, April 2000.

[26] Check Point Software Technologies Ltd. SynDefender: http://www.checkpoint.com/products/firewall-1.

[27] Netscreen 100 Firewall Appliance, http:// www.netscreen . com/

[28] D. Moore, G. Voelker and S. Savage, "Inferring Internet Denial of Service Activity", Proceedings of USENIX Security Symposium'2001, August 2001