Journal of Information Security, 2011, 2, 131-138
doi:10.4236/jis.2011.23013 Published Online July 2011 (http://www.SciRP.org/journal/jis)

Effectiveness of Built-in Security Protection of Microsoft's Windows Server 2003 against TCP SYN Based DDoS Attacks

Hari Krishna Vellalacheruvu, Sanjeev Kumar
Networking Security Research Lab, Department of Electrical and Computer Engineering, The University of Texas-Pan American, Edinburg, USA
E-mail: *sjk@utpa.edu*
Received December 15, 2010; *revised June* 15, 2011; *accepted July* 11, 2011

Abstract

Recent DDoS attacks against several web sites operated by SONY Playstation caused wide spread outage for several days, and loss of user account information. DDoS attacks by WikiLeaks supporters against VISA, MasterCard, and Paypal servers made headline news globally. These DDoS attack floods are known to crash, or reduce the performance of web based applications, and reduce the number of legitimate client connections/sec. TCP SYN flood is one of the most common DDoS attack, and latest operating systems have some form of protection against this attack to prevent the attack in reducing the performance of web applications, and user connections. In this paper, we evaluated the performance of the TCP-SYN attack protection provided in Microsoft's windows server 2003. It is found that the SYN attack protection provided by the server is effective in preventing attacks only at lower loads of SYN attack traffic, however this built-in protection is found to be not effective against high intensity of SYN attack traffic. Measurement results in this paper can help network operators understand the effectiveness of built-in protection mechanism that exists in millions of Windows server 2003 against one of the most popular DDoS attacks, namely the TCP SYN attack, and help enhance security of their network by additional means.

Keywords: Network Security, TCP SYN Based DDoS Attack, Prevention of Attacks

1. Introduction

When TCP/IP protocol suite was initially developed as a part of network research development by the United States Advanced Research Projects Agency (DARPA or ARPA) in 1970s [1], they were unaware of the security attacks. At that time the protocol suite designs were basically concerned with appropriate communication and the scalability of the network. There was no proper framework to defend against security attacks in the initial design of protocol suite. As time progressed TCP/IP gained more popularity than any other architecture. There has always been some hacker community who have been trying to exploit security breaches of popular TCP/IP architecture.

Whenever the hackers exploited the security breaches, the TCP/IP developer community tried to fix it by making some changes to the TCP/IP protocol suite. TCP/IP stack is still evolving to defend against security attacks. For example, recently Microsoft released a critical patch to TCP/IP on 8th September 2009 [2]. This patch corresponds to the zero window size of the TCP packet after the three-way handshake is complete and also time stamp code execution.

TCP implementation may permit the LISTEN state to be entered with either all, some, or none of the pair of IP addresses and port numbers specified by the applications. A link can become established with any user whose details are unidentified to the server ahead of time. This type of unbounded LISTEN is the target of SYN flooding attacks due to the way it is typically implemented by operating systems [3].

2. Three-Way Handshake

TCP uses three-way handshake (**Figure 1**) to establish a connection between any two nodes. The client sends a SYN request with its sequence number to the server. When a SYN is received by server for a local TCP port

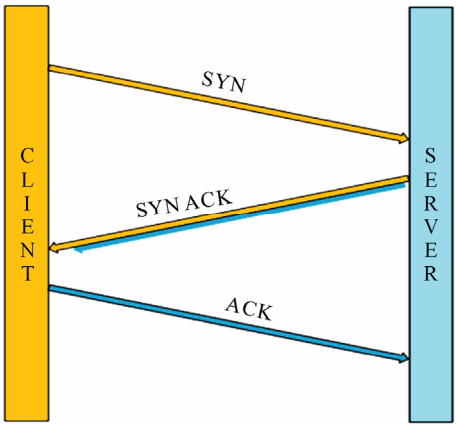

Figure 1. TCP three-way handshake.

where the connection is in the LISTEN state, then the state transitions to SYN-RECEIVED. The Transmission control block (TCB, a data structure to store all the state information for an individual connection) is initialized with information from the header fields of the received SYN segment. In second step the server responds with an ACK to received SYN and it will also sends its own sequence number (SYN) to the client. In the last step, the client responds with final ACK packet. After the last ACK is received by the server, connection state changes from SYN_RECEIVED to ESTABLISH state. The real data transfer between the client and the server is initiated after the three-way handshake is complete.

3. TCP SYN Flood Attack

Over Internet today, it is common for users to access data by using application services of a remote machine. Most of these applications like HTTP, FTP and e-mail run on top of TCP layer. The accessibility and performance of application services depend on how well the underlying Transport protocol works. By some means, if the TCP layer is made unresponsive, the person who is trying to access these services from a remote machine may think that the services are busy/unavailable. In recent years increase in online shopping and online financial transactions make unavailability of the web services, simply intolerable.

In this attack, the attacker makes the server's TCP layer unresponsive by sending a large number of open connection requests or TCP SYN packets (**Figure 2**). This is known as SYN flooding or SYN Bombing, named after specific bit in TCP header specifications. The TCP SYN flooding weakness was discovered as early as 1994 by Bill Cheswick and Steven Bellovin [3]. The SYN flooding attack was first publicized in 1996, with the release of a description and exploit tool in Phrack Magazine. By September of 1996, SYN flooding

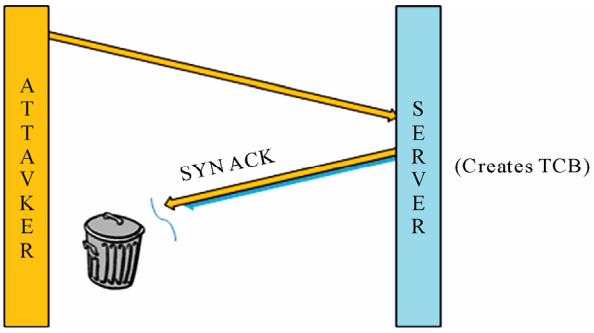

Figure 2. TCP SYN flood attack.

attacks has been observed more frequently on the internet around the world. SYN flooding was particularly serious in comparison to other known denial of service attacks at that time and even now. The community quickly developed different techniques for preventing or limiting the impact of SYN flooding attacks. Some of these techniques like SYN Cache protection and SYN Cookies protection have become important pieces of the TCP implementations in certain operating systems, although some significantly diverge from the TCP specification and none of these techniques have yet been standardized or sanctioned by the IETF process. SYN Chache is one of the most commonly used SYN flooding prevention methods, and variants of this method is implemented in many popular computer operating systems.

Suppose that an attacker directs a large number of SYN requests rapidly to the server with spoofed source IP addresses. In a traditional TCP 3-way hand shake, the server has to create a new TCB for each new connection request it received and save the incomplete state of the connection and the TCP options like window size, Maximum segment size etc. Since the TCB's are limited for each port of the server, the TCB's get filled up. In traditional TCP, the server will send several retransmissions for incomplete connections before the timeout period and eventually get deleted. Even though TCB's are going to be unallocated after certain timeout period, if the attacker manages to keep flooding the server so that no TCB's are free at any given point of time, the TCP layer becomes unresponsive to the legitimate clients.

One typical data structure used for communication is the Transmission Control Block (TCB) which is created and maintained during the lifetime of a given connection. The TCB contains the following information according to RFC 675 [4] (field sizes are notional only and may vary from one implementation to another):

16 bits: Local connection name
48 bits: Local socket
48 bits: Foreign socket
16 bits: Receive window size in octets
32 bits: Receive left window edge (next sequence num-

ber expected)

16 bits: Receive packet buffer size of TCB (may be less than window)

16 bits: Send window size in octets

32 bits: Send left window edge (earliest unacknowledged octet)

32 bits: Next packet sequence number

16 bits: Send packet buffer size of TCB (may be less than window)

8 bits: Connection state

The typical TCB size is sum of all fields which is 280 bits. For each connection standard transport layer allocates one TCB. So the total number of connections that can be supported by the server depends on the number of TCB's available in the server. A TCP synchronize (SYN) attack is a denial-of-service attack that exploits the retransmission and time-out behavior of the Synchronize-Acknowledgement (SYN-ACK) segment during the TCP three-way handshake to create a large number of half-open TCP connections. Depending on the TCP/IP protocol implementation, a large number of half-open TCP connections could do any of the following [5]:

- Use all available memory.
- Use all possible entries in the TCP Transmission Control Block (TCB), an internal table used to track TCP connections. Once the half-open connections use all the entries, further connection attempts are responded with a TCP connection reset.
- Use all available half-open connections. Once all the half-open connections are used, further connection attempts are responded with a TCP connection reset.

4. SYN Attack Protection Performance

We measured the performance of SYN attack protection in the real time traffic circumstances by sending the legitimate client connections and SYN flood attack to the web server at the same time. The legitimate/authentic clients complete the there-way handshake with the server and then send HTTP request for a web page to the server (**Figure 3**). After receiving the web page the clients close the connection with server in traditional TCP way of terminating the connection by exchanging FIN packets. On the other hand the attacker's side is made to send a flood of TCP connection requests with spoofed source IP addresses to the web server with no intention to complete the three-way hand shake with the server. The attackers IP source address are fully randomized to overcome any sort of filtering done on the server side.

We measured the number of legitimate client connections that can be established per second with the server under increasing attack loads. The attack load is incremented from low to high intensity in nonlinear fashion

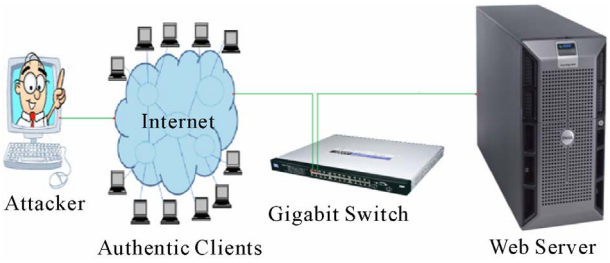

Figure 3. Experimental setup.

from 0 Mbps to 100 Mbps in all of the following experimental results to find the connection rate behavior at lower and higher intensity of attack traffic. The duration of each attack load is kept for 10 minutes (600 seconds) and the statistical readings are collected for each second. *i.e*. 600 reading for each attack load.

The server CPU utilization and Memory status of the server under different loads of SYN attack are shown in **Figures 4** and **5**. The powerful quad core CPU utilization of the server is increasing linearly as the attack load increases (nonlinear) when there is no protection. The maximum CPU utilization 41% is reached at 100 Mbps of SYN attack load. The memory consumption is just 387MB at 100Mbps attack load which is well below the 8 GB RAM installed in the server. From the graphs (**Figures 4** and **5**) it is observed that the server CPU and Memory are not consumed completely because of the SYN Attack.

The total number of TCP connections in SYN_RECEIVED state when the server is under SYN attack is shown in the **Figure 6**. Connections in SYN_RECEIVED state is also referred as half-open TCP connections means incomplete TCP connections The maximum number of half open connections supported by the server at any given instant depends on the backlog size. TCP half-open connections are increasing linearly at lower loads of SYN attack until 7 Mbps. After this point the number half open connections are falling at higher attack load. The average half open connections at each attack load shown in fig 6 is an average of 200 reading. These reading are manually logged with the help of NETSTAT command.

Netstat -n -p tcp|find/c "SYN_RECEIVED"

It is observed in **Figure 6** that the total number of half-open connections in server is unstable after 7 Mbps of SYN attack load.

Figures 7 and **8** show the average successful legitimate connections established with the web server when it is under attack, and no protection is enabled at the server. The legitimate client connections are found to decrease rapidly with increase in TCP-SYN attack load. Without any attack (as shown with 0 Mbps in the graphs), the legitimate clients connections are measured to be around

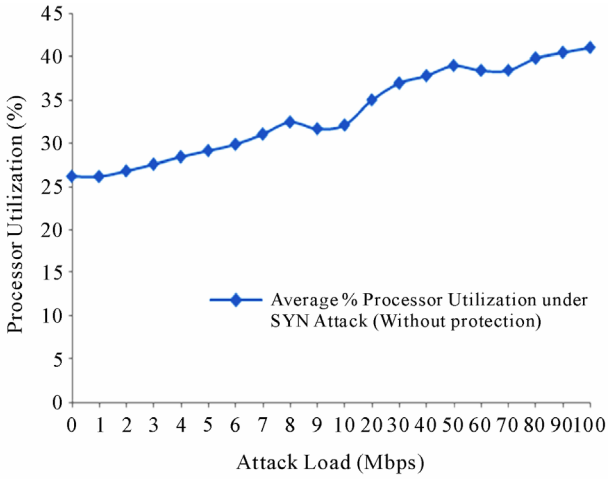

Figure 4. Server CPU utilization (without SYN attack protection) under SYN attack.

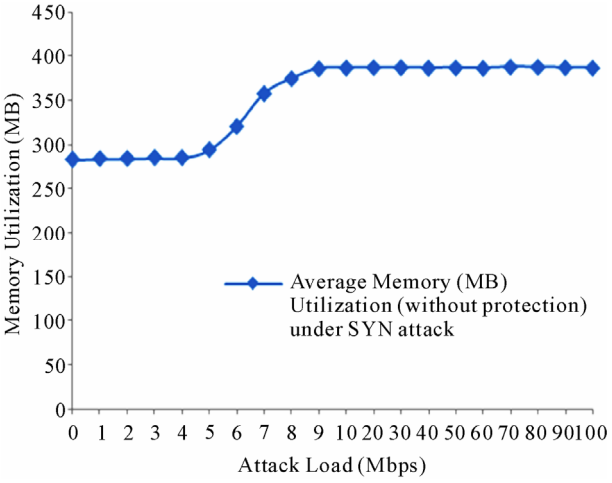

Figure 5. Memory consumption (without SYN attack protection) under SYN attack.

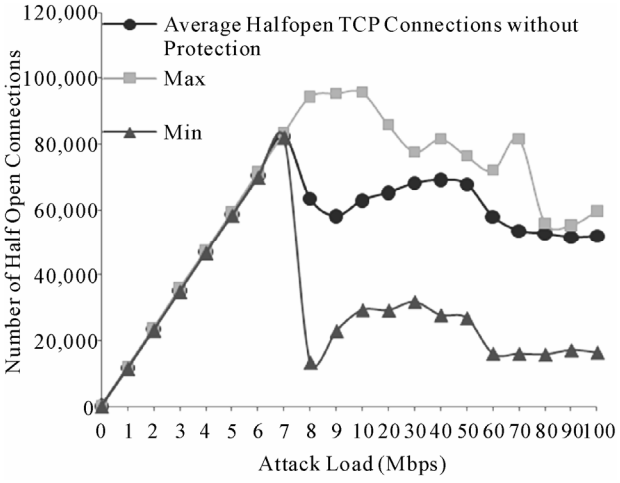

Figure 6. Server TCP connections in SYN_RECEIVED State (without SYN attack protection) under SYN Attack.

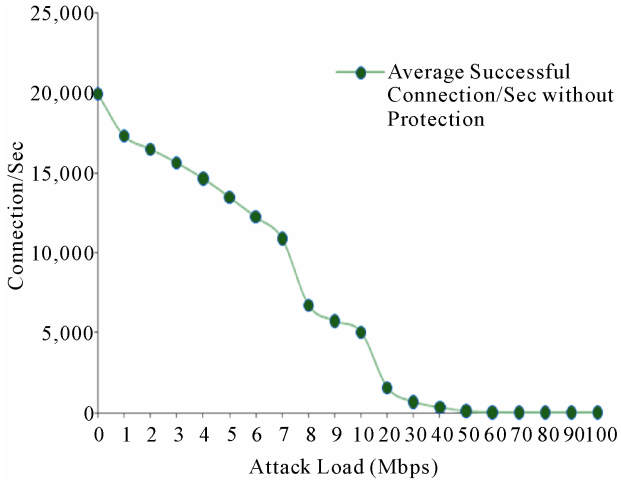

Figure 7. Successful legitimate client connections/sec vs. the attack load without SYN attack protection.

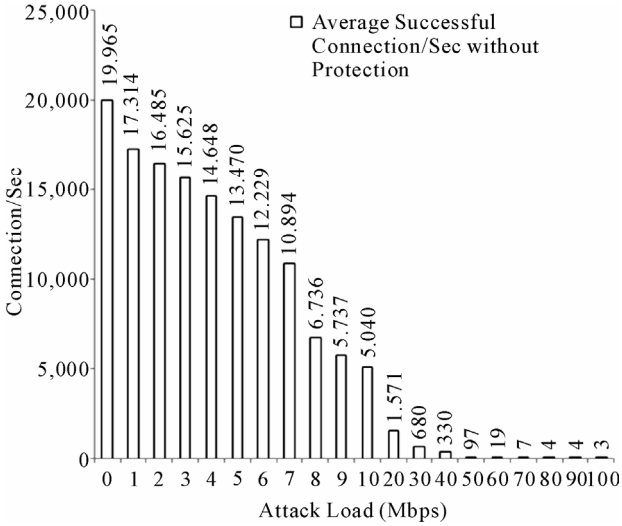

Figure 8. Successful legitimate client connections/sec without SYN Attack Protection.

20,000 connections per second (baseline value). After 60 Mbps of SYN attack load legitimate client connections/sec with the server are almost depleted well below 100 connections/sec. It is observed that around 5000 Connections per second are successful when the SYN attack load intensity is 10 Mbps.

Research community proposed different techniques to detect [6-12], Trace back [13,14] and Defend [15-21] against the TCP SYN flooding attacks. Most of the detection mechanisms proposed depend on the abnormal traffic flow statistics in the network/Internet and the prevention mechanisms depend on filtering, traffic policing and rate limiting. These mechanisms can be implemented in Internet core, firewalls, routers or end systems. When a SYN attack is detected, TCP/IP in Windows Server 2003 and Windows XP lowers the number of retransmis-

sions of the SYN-ACK segment and does not allocate memory or table entry resources for the connection until the TCP three-way handshake has been completed. Microsoft provided a feature called SYN Attack Protect in the server operating system. This feature is available in all versions of windows server 2003 but enabled by default only in some versions of windows server 2003 operating systems. The Microsoft provided definition for this protection as follows [22].

"SYN attack protection involves reducing the amount of retransmissions for the SYN-ACK's, which will reduce the time for which resources have to remain allocated. The allocation of route cache entry resources is delayed until a connection is made and the connection indication to application is delayed until the three-way hand shake is completed."

The action taken by the SYN attack protection mechanism only occurs if TcpMaxHalfOpen and TcpMaxHalfOpenRetried settings are exceeded. The three configurable threshold parameters to trigger TCP's SYN attack flooding protection feature are explained below [23].

1) TcpMaxHalfOpen specifies how many connections the server can maintain in the half-open state before TCP/IP initiates SYN flooding attack protection, by default it is 500 in windows server 2003.

2) TcpMaxHalfOpenRetried specifies how many connections the server can maintain even after a connection request has been retransmitted before TCP/IP initiates SYN flooding attack protection by default it is 400 in windows server 2003.

3) TcpMaxPortsExhausted specifies how many connection requests the server can refuse before TCP/IP initiates SYN flooding attack protection by default it is 100 in windows server 2003.

All the three entries mentioned are used only when SYN flooding protection is enabled on the server, that is, when the value of the SynAttackProtect entry is 1 and the value of the TcpMaxConnectResponseRetransmissions entry is at least 2.

The behavior of TCP/IP protocol stack in the windows server 2003 operating system heavily depends on the registry parameters. We recognized the research efforts made by Microsoft in deciding these registry key parameters for the stable response of server and its services. So we kept most of these parameters in the default state or in the state recommended by the Microsoft as mentioned above for the stable response of the server.

The next step is to enable the SYN attack protection feature in windows server 2003 and observe the server behavior under SYN attack. In the remaining part of this chapter we will observe the server ability to provide services to legitimate clients when SYN attack protection is enable and compare it with the results we had when the SYN attack protection is not active. The SYN attack protection thresholds mentioned earlier are in the default state/value for all the experiments we conducted in this paper.

The network topology created for this testing is same as shown in **Figure 3**. The CPU and Memory usage of the server under SYN attack when protection enabled is shown in the **Figures 9** and **10** respectively. The CPU utilization is nearly the same with and without protection. The memory consumed by server under SYN attack is significantly reduced when the SYN attack protection is active. Compared to the memory resources available in the server and the cost of memory today, it is not significant.

The successful legitimate client connections rate vs. attack load when the server SYN attack protection enabled is shown in the **Figures 11** and **12**. It is observed that even with protection enabled the successful connection rate is decreased as the attack load increases. The legitimate connections are unable to establish and the connection rate is less than 100 connections/sec after 80 Mbps attack load. This is an improvement over the previous scenario where the connections/sec fell below 100 at 60 Mbps without SYN protection. It is observed from **Figure 12** that the successful connection rate at 10 Mbps of attack load is around 16,000 connections/sec, which is more than two times the successful connection rate we achieved without the SYN flood attack protection. The successful connection rate is improved significantly for a given attack load but at higher attack loads after 60 Mbps, the legitimate connections are unable to be established.

Comparison of the results of these two experiments with and without TCP-SYN protection is shown in the **Figure 13**. When the TCP-SYN attack protection is used, the new client connection rate supported by the web server was improved by 226% under TCP SYN attack load of 10 Mbps.

From the results presented in this paper, it is evident that the legitimate client connection rate is improved by the use of SYN attack protection. However SYN attack protection is not effective at higher loads of SYN attack. But if we increase the number of half open connection limit on the server the successful connection rate of clients may improve [24]. A high bound for the half open connection limit can be computed from the bandwidth of the server's network and the timeout used by the servers to discard pending requests. This is kind of brute force solution that waste lots of kernel memory and slow down the server response time, but it can be effective in public servers serving large communities of clients, since such servers have extensive hardware resources. Even if you increase the half open connection limit, it is possible that at some higher load attack traffic the hash table fills up,

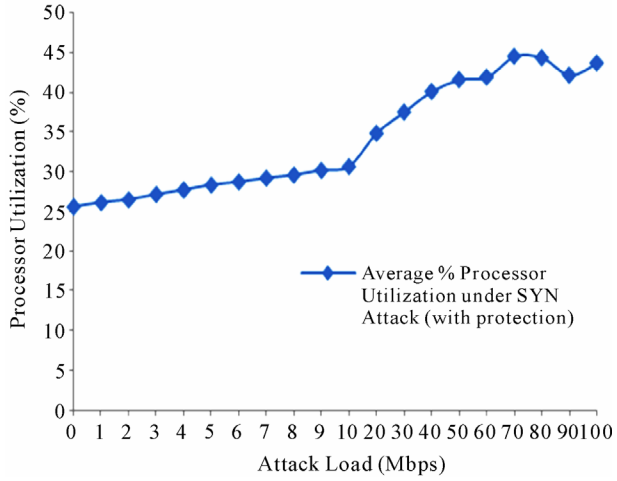

Figure 9. Server CPU utilization (with SYN attack protection) under SYN attack.

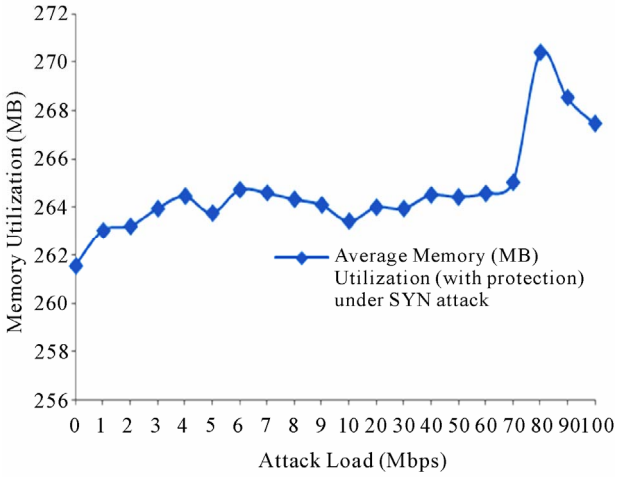

Figure 10. Memory consumption (without SYN attack protection) under SYN attack.

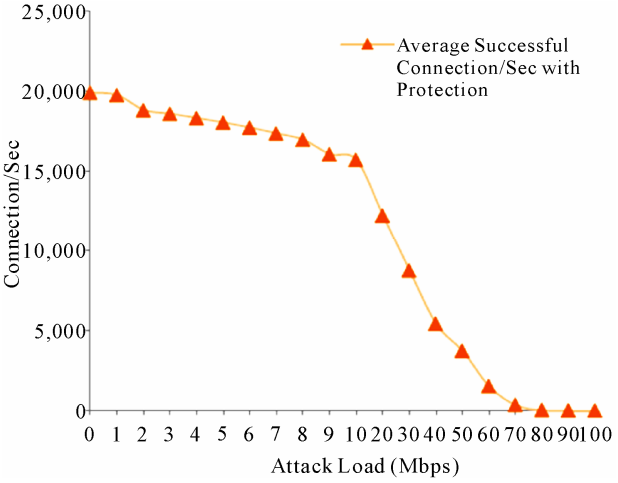

Figure 11. Successful legitimate client connections/sec vs. attack load with SYN attack protection.

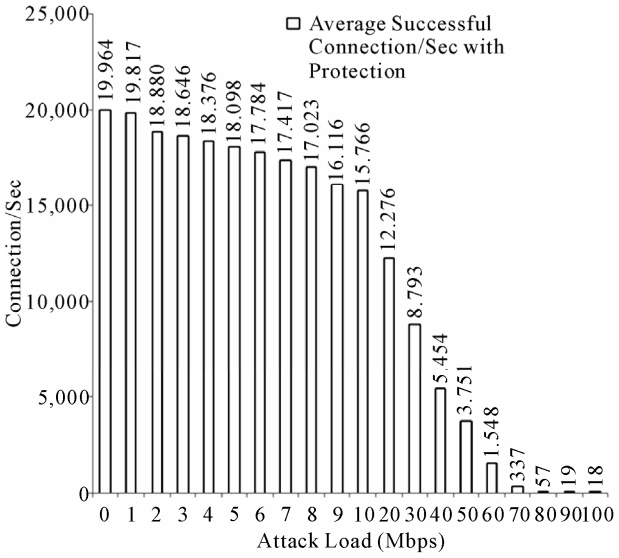

Figure 12. Successful legitimate client connections/sec with SYN attack protect (bar view).

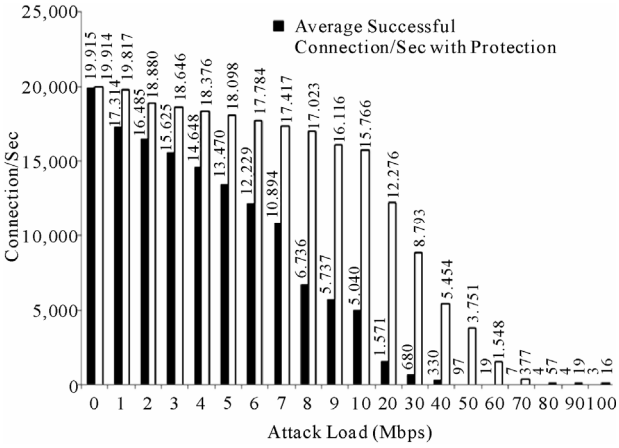

Figure 13. Comparison of successful client connections with and without TCP-SYN attack protection of the windows server.

and it could Overflow with forged connection requests.

5. Conclusions

In this paper, we evaluated the host based protection feature provided by Microsoft against TCP-SYN based DDoS attacks for its widely deployed Windows 2003 servers. It is observed that the built-in, host-based protection feature of Windows server 2003 has limited effectiveness in protecting against TCP-SYN based DDoS attacks. In the absence of any attack, Windows 2003 server was found to support around 20,000 client connections/sec, whereas when the TCP-SYN based DDoS attack traffic was increased to 50Mbps, only around 1700 client connections/sec could be established, which is a

reduction of over 90% of legitimate client connection rate. The experimental measurements show that the built-in protection provided by Microsoft for its Windows server 2003 is effective only for low intensity of the TCP-SYN based DDoS attacks, but not effective against high intensity of the DDoS attacks (exceeding 50 Mbps), and many users are not aware of this fact. This paper conveys an important message for the network managers that they must not rely only on the host-based protection mechanism that exists in the Microsoft's server 2003, and they should deploy additional security devices to effectively defend against DDoS attacks.

6. Acknowledgements

This work was supported in part by the funding from US National Science Foundation, Grant No: 0521585.

7. References

[1] Information Science Institute, "Transmission Control Protocol" RFC 793, University of Southern California, Los Angeles, September 1981. http://tools.ietf.org/html/rfc793

[2] Microsoft Corporation, "Vulnerabilities in Windows TCP/IP Could Allow Remote Code Execution (967723)," *Microsoft Security Bulletin MS*09-048-*Critical*, 8 September 2009. http://www.microsoft.com/technet/security/Bulletin/MS09-048.mspx

[3] W. M. Eddy, "TCP SYN Flooding Attacks and Common Mitigations," RFC 4987, August 2007. http://tools.ietf.org/html/rfc4987

[4] V. Cerf, Y. Dalal and C. Sunshine, "Specification of Internet Transmission Control Program," RFC 675, 1974. http://tools.ietf.org/html/rfc675#section-4.2.2

[5] Microsoft Corporation, "Transmission Control Protocol/Internet Protocol (TCP/IP)," Windows Server TechNet Library, 2003. http://technet.microsoft.com/en-us/library/cc759700(WS.10).aspx

[6] S. Shin, K. Kim and J. Jang, "D-SAT: Detecting SYN Flooding Attack by Two-Stage Statistical Approach," *The* 2005 *Symposium on Applications and the Internet*, Trento, 31 January-4 February 2005, pp. 430-436.

[7] B. Lim and M. S. Uddin, "Statistical-Based SYN-Flooding Detection Using Programmable Network Processor," 3*rd International Conference on Information Technology and Applications*, *ICITA* 2005, Vol. 2, 4-7 July 2005, pp. 465-470.

[8] R. R. Kompella, S. Singh and G. Varghese, "On Scalable Attack Detection in the Network," *Integrated Marketing Communications*, *IMC*'04, University of California, San diego, 25-27 October 2004.

[9] Y. Ohsita, S. Ata and M. Murata, "Detecting Distributed Denial-of-Service Attacks by Analyzing TCP SYN Packets Statistically," *Global Telecommunications Conference*, 2004, *GLOBECOM*'04, Vol. 4, 29 November-3 December, 2004, pp. 2043-2049.

[10] D. M. Divakaran, H. A. Murthy and T. A. Gonsalves, "Detection of SYN Flooding Attacks Using Linear Prediction Analysis," 14*th IEEE International Conference on Networks*, *ICON*'06, Vol. 1, September 2006, pp. 1-6.

[11] B. Xiao, W. Chen, Y. He and E. H.-M. Sha, "An Active Detecting Method against SYN Flooding Attack," 11*th International Conference on Parallel and Distributed Systems*, Vol. 1, 20-22 July 2005, pp. 709-715. doi:10.1109/ICPADS.2005.67

[12] S. Kumar and E. Petana, "Mitigation of TCP-SYN Attack with Microsoft's Windows XP Service Pack3 (SP2) Software," *Proceedings of the* 7*th International Conference on Networking*, Cancun, 13-18 April 2008, pp. 238-242.

[13] H. N. Wang, D. L. Zhang and K. G. Shin, "SYN-Dog: Sniffing SYN Flooding Sources," *Proceedings of the* 22*nd International Conference on Distributed Computing Systems*, Vienna, 2-5 July 2002.

[14] M. Sung and J. Xu, "IP Traceback-Based Intelligent Packet Filtering: A Novel Technique for Defending against Internet DDoS Attacks," *Proceedings of the* 10*th IEEE International Conference on Network Protocols*, Paris, 12-15 November 2002, pp. 302-311. doi:10.1109/ICNP.2002.1181417

[15] W. Chen and D. Yeung, "Defending against TCP SYN Flooding Attacks under Different Types of IP Spoofing," *Networking, International Conference on Systems and International Conference on Mobile Communications and Learning Technologies*, *ICN*/*ICONS*/*MCL* 2006, Morne, 23-29 April 2006, pp. 38-38.

[16] U. K. Tupakula, V. Varadharajan and A. K. Gajam, "Counteracting TCP SYN DDoS Attacks Using Automated Model," *Global Telecommunications Conference*, 2004, *GLOBECOM*'04, Vol. 4, 29 November-3 December 2004, pp. 2240-2244.

[17] B. AI-Dwmiri and G. Manimaran, "Intentional Dropping: A Novel Scheme for SYN Flooding Mitigation," 25*th IEEE International Conference on Computer Communications*, Barcelona, 23-29 April 2006, pp. 1-5.

[18] Q. Xiaofeng, H. Jihong and C. Ming, "A Mechanism to Defend SYN Flooding Attack Based on Network Measurement System," 2*nd International Conference on Information Technology*: *Research and Education*, *ITRE* 2004, London, 28 June-1 July 2004, pp. 208-212.

[19] H. Safa, M. Chouman, H. Artail and M. Karam, "A Collaborative Defense Mechanism against SYN Flooding Attacks in IP Networks," *Journal of Network and Computer Applications*, Vol. 31, No. 4, 2008, pp. 509-534. doi:10.1016/j.jnca.2007.12.004

[20] Y. P. Swami and H. Tschofenig, "Protecting Mobile Devices from TCP Flooding Attacks," *Proceedings of* 1*st ACM/IEEE International Workshop on Mobility in the Evolving Internet Architecture*, San Francisco, 1 Decem-

ber 2006.

[21] F. Kargl, J. Maier and M. Weber, "Protecting Web Servers from Distributed Denial of Service Attacks," ACM, May 2001.

[22] L. Jonathan, "Resisting SYN Flood Attacks with SYN Cache," *Proceedings of the BSDCon Conference on File and Storage Technologies*, February 2002. http://people.freebsd.org/~jlemon/papers/syncache.pdf

[23] Microsoft Corporation, "Microsoft Windows Server 2003 TCP/IP Implementation Details," March 2006.

[24] A. Zuquete, "Improving the Functionality of SYN Cookies," 6*th IFIP Communications and Multimedia Security Conference*, Portoroz, 26-27 September 2002.

