International Journal of Science and Research (IJSR)
ISSN: 2319-7064
ResearchGate Impact Factor (2018): 0.28 | SJIF (2019): 7.583

Secure Information Exchange and Performance Implications on Web Server: A Case Study of Secure Socket Layer Protocol

Simon Peter Khabusi[1], Rajni Jindal[2]

[1, 2]Delhi Technological University, Department of Computer Science & Engineering, Bawana Road, Delhi 110042, India

Abstract: ***The recent trend in internet computing and web technology has facilitated steady growth of e-commerce transactions by most global companies. The vast amounts of critical data exchanged over the internet present many security issues. Secure Socket Layer (SSL) protocol is one of the secure connection oriented protocols that enables safe and secure message transmission over the internet by ensuring data confidentiality, authentication and integrity of the communicating parties, that is web server and web client. Many researches have been undertaken on SSL protocol and its effects on web server performance. In this work, a detailed analysis of SSL protocol, its application in secure information exchange and performance implications on web server has been presented. To achieve this objective, we enabled SSL on Apache running on Windows deployment Server 2008 and captured the values of processor time, Network interface, memory available, and average disk queue length for different workloads using the Windows performance monitor and Nagios XI, the most powerful and trusted infrastructure monitoring tool on the market. The data obtained reveals that SSL protocol has detrimental effects on processor time with measurable effect on available memory bytes, whereas minimal effect on average disk queue length was observed. Conclusively, techniques such as distributed computing, SSL certificate reuse, downloading the RSA decryption function and use of Broadcom SSL 800 accelerator have been proposed to overcome the overhead on server performance.***

Keywords: SSL Protocol; Web server performance; performance counters; security threats, security services

1. Introduction

Computer Networks have grown so rapidly that security for information which span from military, Education, Health, Banking and commerce need to be guaranteed [1]. Due to the many security breaches, network security cannot go unmentioned in the current communication systems [2]. Many security mechanisms have been proposed by different authors and researchers [3-5] and among them include the Secure Socket Layer protocol which is a secure framework for data communication over internet.

Connection oriented protocols are those in which a connection is setup before information can be transferred. In Connectionless protocols, no explicit actions of connection setup are done before data transmission. Instead, packets of data are routed to their destinations basing on their header information. Connectionless protocols overcome the delay and processing overhead that results from setting up the connection. In contrast, Connection oriented protocols have the advantage of guarantee of service using the connection information. Additionally, network resources such as bandwidth can be used more efficiently by "switching" the server-client to appropriate connections during set up [23]. TCP and SSL protocols are good examples of Connection-oriented protocols.

The SSL protocol enables safe and secure message transmission by providing three basic security services that is, data integrity, confidentiality and end point authentication [1]. Designed by Netscape, SSL is capable of performing these three main functions by providing a secure communication between a client and a server [6].

Two strengths of SSL exist i.e., forty bits and one hundred twenty eight bits, which is actually the length of the generated session key in every encrypted transaction. The ease to break the encryption code correlates with the size of the key length. The 40-bit SSL sessions are supported by majority of the existing internet browsers. Netscape communicator 4.0 has the capability of encrypting messages with stronger and more reliable keys (128-bits) compared to the sessions formed using 40-bits encryption. Most entities that operate internationally using internet transactions utilize global SSL certificates program to extend strong and reliable data protection to their clients [7-8].

Research has shown that much time is spent in encrypting and/or decrypting the key that bears the private/secret key in SSL handshake. Handshaking in SSL connection initiation involves establishing capabilities for secure exchange, authentication of the server system together with exchange of the key, and authentication of client along with key exchange which are highly time consuming. [9] Notes that the handshake protocol consumes significant resources through the vast amount of information exchanged between server and client in pursuit to establish a secure connection.

Depending upon the performance aspects, SSL could cause a major impact on the windows web server performance as analyzed by [9]. Most researches have published data related to SSL impact on Windows servers subjected to a single client. However, less information is known about the impact of SSL on the servers serving multiple clients. The major drawback of the existing researches has been majorly the accuracy of the data obtained after server monitoring using the default system monitors incorporated in server machines. It is noted that SSL impacts the performance of the server

systems as indicated by the utilization of the CPU by SSL processes, rate of packet flow (per second), memory usage and link utilization. Despite SSL impact on some of these counters, it is also noted that the secure protocol does not affect the length of the disk and page fault (per second).

Analyzing SSL makes it possible to configure and experiment its impact on the server software counters hence enabling us to draw conclusions and write recommendations for future research in line with remedies to enhance the performance of the systems running SSL.

In this work therefore, the SSL protocol impact on Apache server run on a Windows machine and serving multiple clients with data of different workloads has been analyzed. We have done this in a more controlled environment using Nagios XI software.

2. Operation of SSL protocol

SSL protocol provides three security services; authentication, confidentiality and integrity [2]. On a logical perspective, it provides a secure "pipe" between the communicating parties i.e. web server and web browser. Different versions of SSL exist i.e. 2, 3 (most popular), and 3.1, are supported my most web browsers such as Google Chrome, Opera Mini, Mozilla Fire Fox, etc. URLs that use SSL protocol in their connection start with *https* instead of *http* [10]. Despite its ability to secure information exchange, its use leads to performance degradation of the web server unlike in non-secured data transmission [11]. SSL protocol also supports many more applications such as FTP and TELNET [10].

In Zona's research, it is highlighted that users normally wait for roughly 8 seconds and failure for a page to load; they go to other sites where they can attain faster response and better user experience. Such slow responses in e-commerce applications are mostly attributed to the use of SSL that consumes significant amount of processing resources [12]. Many cryptographic algorithms are used for encryption in SSL protocol, RSA being the most common [9]. The RSA falls under the asymmetric key cryptographic algorithms, because it makes use of two keys that is to say public and private used by every node in the network. A big number N is computed as a product of two random prime numbers *p* and *q* generated which is sent to the receiver in order to generate her own private key, known to herself [15]. The process is as follows;

$$N = p\,X\,q \quad (1)$$

The public key, *Kp* should not be a factor *of (p-1) (q-1).* The private key, *Ks* is obtained from the expression

$$(Kp\,X\,Ks)\ mod\ (p\text{-}1)\ (q\text{-}1) = 1. \quad (2)$$

To perform encryption on a message say M leading to ciphertext say C, proceed as follows.

$$C = M^{Kp}\ mod\ N \quad (3)$$

And to decrypt C back to the initial message, M, the computation below holds good.

$$M = C^{ks}\ mod\ N. \quad (4)$$

Note that the bit numbers used by RSA is generally 1024 though most CPUs of majority of web servers function on 32- bit and others on 64-bit. The 1024 bit data therefore is divided into 32 or 64 bit portions accordingly by the processor. Unfortunately; majority of processors lack on-chip dividers hence the division operations depend entirely on software. Therefore the server CPUs tend to slow down due to the fact that handshaking poses high computational load on it, hence resulting into poor server performance [16]. Stream ciphers and Block ciphers are also used in SSL protocol [2]. Stream ciphers perform encryption by taking 1 cipher text byte at a time unlike block ciphers which encrypt blocks of text at a time [15]. However, note that both ciphers are symmetric key ciphers in that one key is utilized for both encryption and decryption at sender and receiver sides respectively which is exchanged securely by Diffie-hell man Algorithm.

Table 1: Other Algorithms used in SSL handshake protocol

Strem Cipher		Block Cipher	
Algorithm	Key Size	Algorithm	Key Size
RC4	40	AES	128, 256
RC4	128	IDEA	128
		RC2	40
		DES	40
		DES-3	168
		Fortezza	80

SSL is conceptually considered as a separate layer within TCP/IP protocol stack lying between Application and Transport layers.

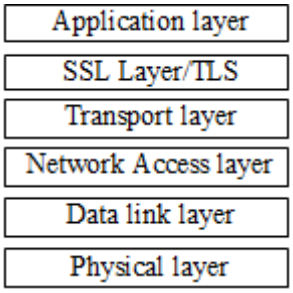

Figure 1: Position of SSL layer in TCP/IP Protocol stack

Information required to be transmitted is prepared at the sender's end and handed to SSL layer instead of transport layer directly unlike in the normal TCP communication. The encrypted application layer data is received at the SSL layer and the SSL Header denoted SH is added to the message. The resulting data from SSL (L5) is then added to the next layer, i.e., transport which also embeds its header (H4) and forwards the message to the next layer, the Network access layer and the process continues. When the receiver gets the message, the SSL layer removes the first Header (SH), and unpacks the data by decryption, and delivers the plaintext to the first layer i.e., application of the client [2] as illustrated in figure 2. The main reason for the location of the SSL after the application and before transport layers is that at the lower layers i.e. Internet and Data link layers, IP addresses and physical addresses of computers i.e. Sender, receiver and intermediate nodes which guide the packet on its transmission and delivery are added to the data hence such headers cannot be encrypted because the destination of the data could not be traced.

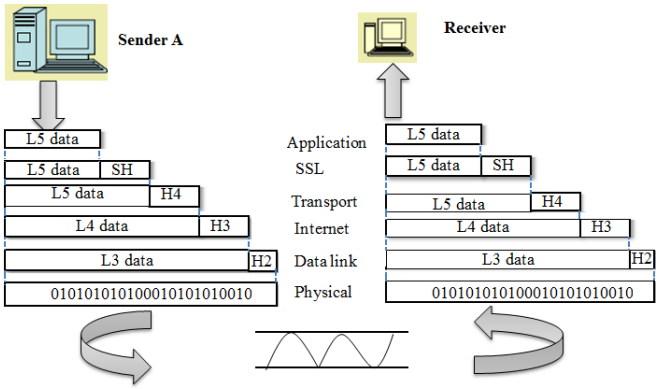

Figure 2: Cryptographic Procedure of SSL protocol

SSL protocol comprises of three sub-protocols, i.e. handshake, Record and Alert.

1) Handshake Protocol
This is the first sub-protocol in SSL enabled client-server connections that involve exchange of a series of messages consisting of type, length and content fields between the server and the client. This protocol involves; i) stablishing security capabilities**;** initiates a logical connection and establishes the security capabilities associated with that connection and consists of the *client hello* and *server hello*, ii) Server authentication and key exchange where the server is the sole sender of all messages, iii) Client authentication and key exchange which is initiated by the client and involves sending the client certificate to the server, key exchange, certification verification and Finish.

2) Record Protocol
This phase comes in after the handshake protocol phase has been completed successfully. Confidentiality and Integrity are the two security services provided by the record protocol. The input to the SSL record protocol is an application message which is fragmented into smaller blocks. Secondly, the fragmented message blocks are then optionally compressed and MAC is added. Message encryption then follows MAC addition. A header is then added to the message and handed to transport layer. At transport layer, TCP protocol processes it just like any other TCP block. The header of each block is then removed at the receiver's end and thereafter decrypted, verified, decompressed, and reassembled into application messages.

3) The Alert Protocol
An alert message is sent when any of the communicating parties detect an error. The connection is closed and end-to end transmission terminated as soon as a fatal error is detected. Under such circumstances, the parties also destroy the session identifiers, and secret keys associated with this connection before the connection is terminated. But if the severity of the error is low, the connection is not terminated; instead the server and client handle the error and continue. An alert message consists basically of two bytes; the first byte which signifies the type of error and the second byte which specifies the actual error.

Closing and Resuming SSL Connections
The client and server inform each other of the intention to close their connection prior to ending the communication. To achieve this procedure, each party sends a *Close notify* alert to the other party hence ensuring a graceful closure of the connection. A party's receipt of this alert implies that its activities must be halted immediately and its own *Close notify* alert is sent then the connection is ended from either sides. An SSL connection cannot be resumed if it ends without a *Close notify* from both communicating parties.

3. Related Work

Many researches have been undertaken in ascertaining the impact of SSL protocol on web servers.

In Mohammed A. Alnatheer [9], the server software used is the IIS 6.0. The web server is powerful, reliable, manageable, and scalable. This server infrastructure supports all 2003 windows server versions.

A single client that runs on Windows XP operating system with 2.4GHz Pentium IV processor and a server with Pentium IV processor of specification 2.79 GHz and running on windows 2003 OS were used. A Local Area Network (LAN) of speed 100Mb per second was used to connect the server and the client.

The Author used a workload of 3MB to 250MB in the experiment because it is practically difficult to monitor the effect of SSL protocol with small workloads. Memory available in Bytes, processor time (in %), Length of disk queue (average) and total bytes per second of the Network.
The captured data was analyzed using variance as a statistical analysis tool to compare the variation of the investigated system parameters while using SSL and without SSL. The Author concludes that the protocol greatly affects; the time of processing (in %) when bigger workloads e.g., 75MB or more are used, available memory (in bytes) with any workload size. Also noted that SSL protocol poses no effect on; the total bytes per second of the Network interface since the protocol has no negative effects on packet numbers sent and received from client and to the server, and the physical disk average queue length. It is also noted that the exponential growth of the time of processing is attributed to the decryption process of the ciphertext that was earlier encrypted because this process is CPU intensive and cryptographic operations which occur in the handshake protocol [9].

Performance analysis and security in e-commerce applications have been discussed in [14]. Additionally, they describe their findings on how this protocol affects the time for the server to receive a response and then respond accordingly. Their results also confirm that use of overhead secure connections rises the time of response of the client from 0.1 to 6.0 seconds averagely. Additionally, the cost of getting and forwarding secured information was about 40 % more than the total costs for handling unsecured data. Hence, though requests which transfer much data are more affected, the researchers who contrasted modules of Apache to those of CGI reached a conclusion that using modules didn't add

significant improvement on the performance of the server [14].

A discussion of the performance impacts of secure connection oriented protocols, SSL in particular has been presented by J. Almeida, Yates [11]. An analysis of , ache size , processor numbers, throughput, utilization, ratios of cache miss, control dependencies, and file access sizes, bus transactions, and network load, among other various components has been made. They draw conclusion that processor units with more core frequency improve the SSL performance; additionally, processor with high pipeline depth can improve performance of SSL operations, while issue width increase may not give any notable performance improvement most especially in circumstances where the processor usage is dominated by bulk information encryption tasks [11].

The author in SSL Accelerator [13] investigates the highest possible throughput of SSL for different combinations of encryption/decryption algorithms and varying lengths of the keys. It has also been highlighted that a public key of bits 1,280 is worth to shield from Individual attacks though a 1,536 bit key is required to shield the communicating parties from large corporation attacks. In addition, a bit key of 2,048 may be the fit for protection against attacks from bigger entities such as the government. The final results intimate that handshaking in SSL is computationally intensive and the impact of other measures of security depends mostly on the business application's specific architecture.

The authors in System Monitor [17], developed a model for internalizing and evaluating server performance. They also presented new results for possible workloads. Their work reveal that up to about 90 percent of time spent handling HTTP requests is used in the kernel for a web server saturated by client requests.
Apache server in Redhat was used by Liu *et. al* [22] to monitor the transmission of packets using Ethereal Network Monitor. They concluded that the performance indices of packet trip time of packets with SSL are weaker that those without SSL reason being that the encryption and authentication of application data need more processing time in SSL protocol.

3.1 Research Contribution

The following research gaps have been covered in this work.

1) The findings of the existing works were based on single client data. Information exchange between Windows server 2008 and multiple clients was used to actively monitor and capture data of the performance counters. This presents a more realistic scenario.
2) A more controlled environment was achieved by using Nagios XI. This guarantees the accuracy of the data obtained
3) Other key website services such as DNS IP match, DNS Resolution, HTTP, Ping and SSL Certificate Status were monitored.

4. Project Design

In this section, we present the infrastructure and network simulation used to perform our experiments. We further discuss the different tools and software packages used and their configurations.

4.1 Infrastructure

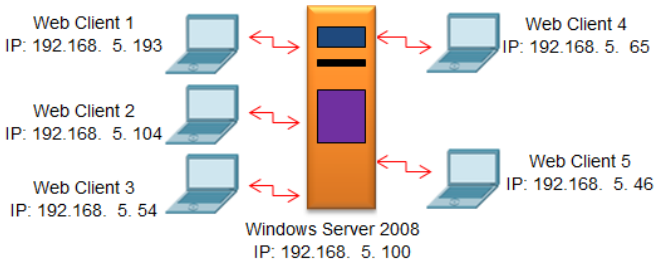

Figure 2: Network Model of the infrastructure

4.2 Network Simulation

Before carrying out the performance measures, a network was simulated in Cisco Packet tracer as shown in the figure below to ascertain the possibility of effective communication between the server and the clients.

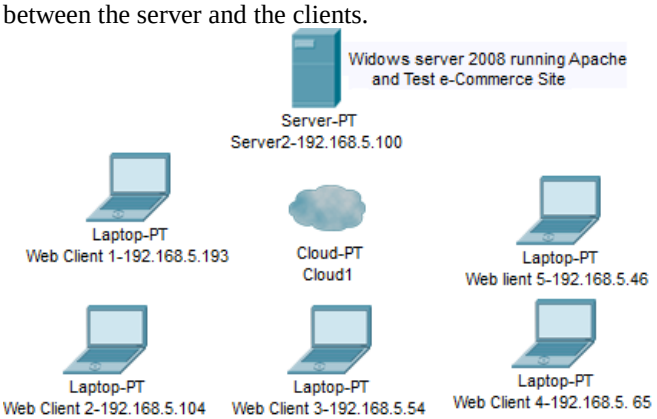

Figure 3: Network view in Cisco Packet Tracer

4.3 Windows Deployment Server 2008

Two types of WDS server exist; i.e. transport and domain based Windows Servers. Transport based WDS was designed for smaller environments without Active Directory (AD) domain. Though Transport servers require fewer infrastructures than domain-based servers, they tend to be more difficult to set up and configure.

A domain based WDS was deployed and configured on an Intel ® Core ™ i5 CPU M 569 @ 2.67GHz computer.
The WDS 2008 comes with Internet Information Services (IIS) though we used Apache as the server software due to the obvious reasons discussed below.

i) IIS requires Windows server license compared to Apache which is free.
ii) It's easier to create modules and configure them in Apache compared to IIS
iii) IIS is memory intensive as opposed to Apache which increases server hosting costs.
iv) The httpd file in Apache is easier to configure than in IIS

4.4 Monitoring Software Packages

1) System Performance Monitor
This is an in-built tool which was simply configured to capture performance parameters within the stipulated time.

2) Deployment and configuration of Nagios XI running in VMWare
The Nagios XI monitoring tool was installed in VMWare because it was designed to run in LINUX. Configurations of the parameters to be monitored were done.

Operation of Nagios XI
Operates in two modes i.e., Active monitoring and passive monitoring.

Active monitoring involves use of agent or native protocol. With agent, Nagios reaches out to the agent which gets information and responds back to Nagios Xi. After receiving the information, Nagios XI will do a couple of things;
i) Store the information for visualization
ii) Generate alerts e.g. driver is full and needs update
While using Native Protocols, Nagios XI reaches out to a network element such as a network switch using a protocol such as SNMP.

In Passive Monitoring, Nagios XI never reaches out to agents but rather the agents send information on regular basis or when something occurs and alerts Nagios. In this operation mode, information is sent to the Nagios server then Visualization and alerting can be performed.

4.5 Web Application

Sample e-commerce site was designed in Mendix software with the capability of accepting uploads and downloads of different file sizes into a backend database.

4.6 SSL Certificate generation

This involved using WAMP + SSL to open localhost with HTTPS. WAMP was downloaded and installed; openssl package was also installed using the command line interface.

WAMP was then configured to use HTTP+SSL=HTTPS
Some changes were made on files in the WAMP directory such as openssl.cnf file. Two keys were created and used to create a locally signed SSL certificate that was used in the study.

5. Experimental Description

Intel ® Core ™ i5 CPU M 569 @ 2.67GHz computer was configured and deployed Windows server 2008. Apache server, an open source, fast and secure software package distributed by Apache software Foundation was installed and configured.

A pair of keys was generated using openssl command and used to create private keys (Certificate Signing Request – CSR). The CSR can be sent to a CA to generate a SSL certificate for public usage. However, for the purposes of this experiment, we self-signed the SSL certificate using the two generated keys. We also configured five clients with Google Chrome browser for making server requests. The client was supported by Windows 7 64 bit operating system, on a Pentium IV Processor of 2.4 GHz. Both the clients and the server were connected by internet connection speed of 50 Mbps. A simple web application with a module of uploading and downloading data was hosted on the server locally and the workload of the experiment varied from 5MB to 200MB since smaller workloads cannot give us a clear overview of the SSL protocol on a web server. The processing time (% total), bytes of available memory, Network Interface (Bytes total/Sec) and average disk queue length were captured for every data size transmitted.

5.1 Performance counters and Measurement

Windows performance monitor was used to track system performance [9]. A performance counter indicates the quantity of object that can be measured in some unit, such as percent, rate per second, or peak value [9]. Performance data can be presented inform of charts, histograms and reports. The counters analyzed in this research are; *Percentage of total processor time***:** The elapsed time (in percentage) this thread used the processor to execute the instructions. *Memory Available Bytes:* This is the amount of physical memory (bytes) available for allocation to a process or for system use. *Average disk queue length:* This is the average of the disk read total and disk write total and *Network Interface in bytes total per second*. Other aspects of server and website performance monitored include: DNS IP match, DNS Resolution, HTTP, Ping, web application response speed, overload error alerts and SSL Certificate Status.

5.2 Experiment Results

In our experiment, we used the performance monitor and Nagios XI to capture the processing time, Bytes of available memory, length of the disk queue, and Network Interface (Bytes Total/Second). The data was analyzed in excel and percentage changes in parameter values obtained when not using SSL enabled connection and with SSL enabled connection were calculated and the overall patterns illustrated graphically.

a) Percentage of total processor time
The increase in workloads reflected an exponential increase in percentage of total processor time in both cases, i.e. while using SSL enabled connection and non-SSL enabled connection. However, SSL enabled connection showed a greater increase in the percentage of total processor time compared to non-SSL enabled connection.

Table 2: Percentage of Total Processor time

Workload	No SSL	SSL	% increase
5MB	0.143	0.189	32.1678
13MB	0.176	0.272	54.5455
18MB	0.197	0.291	47.7157
30MB	0.204	0.354	73.5294
68MB	0.309	0.789	155.3398

148MB	0.821	1.981	141.2911
200MB	1.001	2.105	110.2897

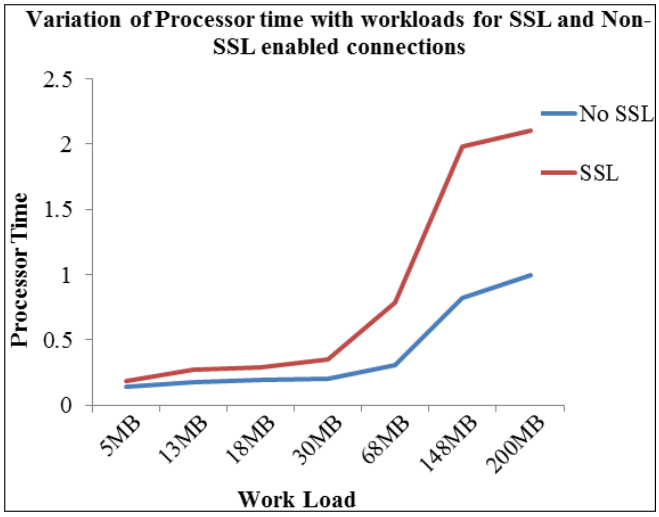

Figure 4: Line Graph showing variation of processor time with workloads for SSL and Non-SSL enabled connections

b) Memory available Bytes

The Memory available bytes data was captured for non SSL enabled connection and also for SSL enabled connection for different workloads. The Data shows that SSL decreases the memory available by about 1% or less. However, there is no direct connection of SSL impact on this counter with the workload size used as there is no defined trend in the values obtained.

Table 3: Memory Available bytes

Workload	No SSL	SSL	% Decrease
5MB	196503815	195807667	0.35
13MB	195954564	195706651	0.13
18MB	198172274	196626351	0.78
30MB	196163503	195947724	0.11
68MB	196981062	195731945	0.63
148MB	196431322	195642713	0.40
200MB	196513002	195832627	0.35

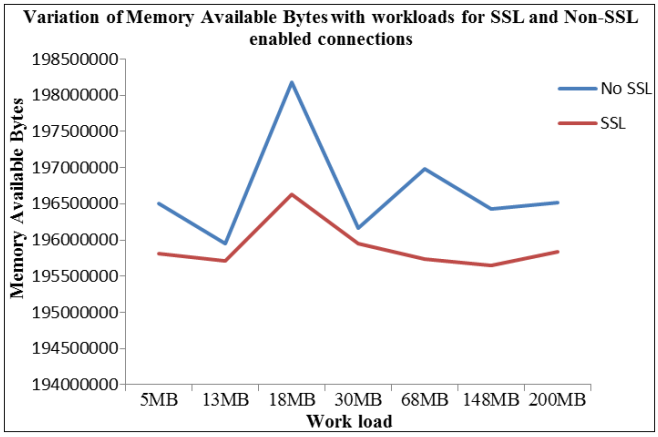

Figure 6: Line Graph showing the variation of Memory available bytes with Workloads for SSL and Non-SSL enabled connections

c) Average disk queue length

The captured data shows a very minimal impact of using SSL enabled connections in comparison to non-SSL enabled connections on the average disk queue length. There is no considerable impact of SSL on average disk queue length according to the data obtained.

Table 4: Average disk queue length comparison

Workload	No SSL	SSL	% Increase
5MB	0.0102	0.0101	-0.98
13MB	0.0105	0.0103	-1.90
18MB	0.0213	0.0218	2.35
30MB	0.0217	0.0221	1.84
68MB	0.0605	0.0608	0.49
148MB	0.0718	0.0712	-0.84
200MB	0.1950	0.1921	-1.49

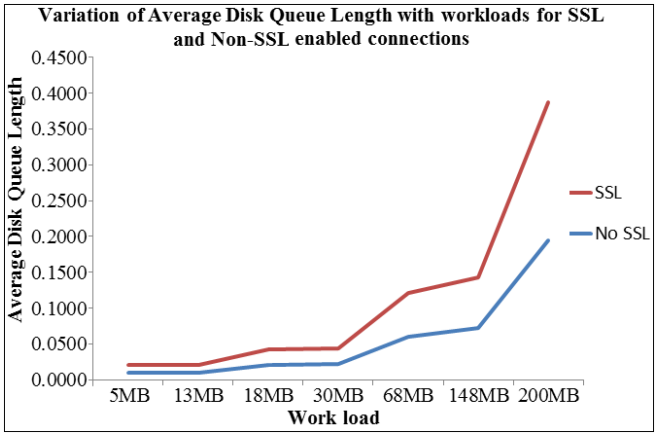

Figure 7: Line Graph showing the variation of Average Disk Queue Length with varying Workload

d) Network Interface (Bytes Total/Second)

This is defined as the rate at which bytes are sent and received on the network interface. It also includes the framing characters.

Bytes Total/sec = Bytes Received/sec + Bytes Sent/sec.

Table 5: Network Interface Bytes total/sec Comparison

Workload	No SSL	SSL	% Decreases
5MB	38757	38745	0.0310
13MB	109406	109174	0.2121
18MB	139521	139153	0.2638
30MB	269526	259071	3.8790
68MB	319565	318586	0.3064
148MB	775769	773707	0.2658
200MB	2659521	2654760	0.1790

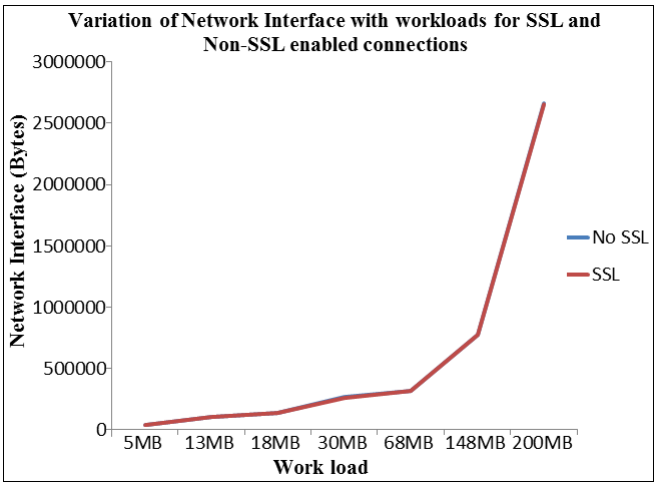

Figure 8: Line Graph showing variation of Network Interface in Bytes/sec with varying workloads for SSL enabled and Non-SSL enabled connections

e) Other Aspects Monitored

Other key website services such as DNS IP match, DNS Resolution, HTTP, Ping and SSL Certificate Status were monitored. Nagios XI provides an infrastructure that enables each and every aspect of a Windows/Linux Server or Desktop, Website, active directory, email delivery, DHCP server, FTP server, MSSQL server, etc.

Real time error alerts are also generated hence the server performance can be monitored in real time and any issues resolved within the shortest time possible.

6. Discussion and Conclusion

Conclusively, SSL greatly affects the percentage of total processor time that can be observed to grow exponentially with increasing workloads. This implies that with more that 1 million clients interacting with an e-commerce web server, the overhead can be significant. Therefore, there is need to find mechanisms of overcoming such overhead in order to provide good user experience. Such mechanisms would include; reuse of earlier SSL connection to avoid new handshake in the next connection so long as a certificate has not yet expired and is within a 24 hour period, offloading the RSA decryption function, enabling the adapter of accelerator to speed up transactions securely so that proprietary and personal information can be protected during online transactions, distributing the server functions to lower the overhead traffic on a single server and finally using the Broadcom SSL 800 accelerator to enhance performance. It is also noted that SSL enabled transactions have some impact on memory available bytes though this impact is not that significant as compared to the effect it poses on percentage of total processor time. The Average disk queue length is also less affected by SSL enabled transactions. According to the data collected by Mohammed A. Alnatheer [9] on the impact of SSL on percentage of total processor time, it can be observed that there is a correlation between the numbers of clients communicating with the server in an SSL enabled connection and the percentage of total processor time. With multiple clients engaging the server, the increase in percentage of total processor time is significant. However, to view such differences, it's necessary to use a server running on the computer with similar system specifications for clearer observation.

7. Future Work

In future work, other counters such as network usage and congestion measure, CPU performance counters such as processor time of thread and user time of thread can be incorporated in the study and analyzed using different workload sizes. Other Operating System Environments such as Linux can be used in the study with perf tool. The study can also be advanced by analyzing real life SSL applications such as e-commerce applications.

References

[1] LIU Niansheng, Y. Guohao, Wang Yu and Guo Donghui; *Security Analysis and configuration of SSL protocol*, IEEE Research Article, 2017
[2] Atul Kahate, Cryptography and Information security, 3rd Edition
[3] E. Frenzel Louis, "SSL NIC has a Knack for instantly securing transactions," Electronic Design, Vol 52, no. 27 pp. 34 December 8, 2004
[4] A. Roy Chowdhury, J.S Baras, M. Hadjitheodosiou, etc., "Security issues in hybrid networks with a satellite component,' IEEE wireless communications, vol 12 no. 6 pp. 50-61, Dec, 2005.
[5] V. Gupta, and S. Gupta, "Securing the wirelss internet,' IEEE communication Magazine, Vol 39, no. 12. Pp 68-75, December 2001.
[6] A. O Freier, P. Karlton and P.C Kocher, the SSL protocol version 3.0, http:// wp. Netscape.com/eng/SSL 3/draft 302.txt
[7] R. Kinicki *et al*., "Electronic commerce performance study," in *Proc. the Euromedia* '98, Leicester, United Kingdom, January 1998.
[8] D. Menasce, "Security Performance," *IEEE Internet Computing*, May/June 2003.
[9] Mohammed A. Alnathear; Secure Socket Layer (SSL), Impact on web server performance; Journal of advances in computer networks, Vol 2, no. 3, September 2014.
[10] Introduction to SSL. [Online]. Available: http://wp .netscape.c01l1/security/techbriefs/ssl.html
[11] J. Almeida, V. Almeida, and D. Yates, "Measuring the behavior of a world-wide web server, high performance networking VII," in Proc. the 7th Conference on High Performance Networking, White Plains, NY, April 1997
[12] C. Pfleeger and S. Pfleeger, Security in Computing, Upper Saddle River, Prentice Real, 2003.
[13] SSL Accelerator. [Online]. Available: http://www.mcdiawcb.com.sg/sonicwall/white1020pap cr/SonicWAL L SSL WP.pdf
[14] G. Paixao, W. Meira Jr., V. Almeida, D. Menasce, and A. Pereira, "Design and implementation of a tool for measuring the performance of complex e-commerce sites," in *Proc. Tools 2000 Conference*, Chicago, IL, March 2000.

[15] Geeks for Geeks [online] , RSA Algorithm in Cryptography, Date accessed: 15th August, 2019, Available on: https://www.geeksforgeeks.org/rsa-algorithm-cryptography/
[16] Performance counters reference for windows server 2003.[Online].Available: http://www.lllicrosoft.com/rcsourccs/doculTIcntation/WindowsServ/2003/all/dcployguide/
[17] Securing Application with SSL. [Online]. Available: http://publib.boulder.ibm.comlhtml/as400/v5rl/ic2931/index.htm'!info/rzai n/rzainovcrvicw.htrn
[18] Lifewire, Apache Web Server. [Online] Accessed on: 15th August, 2019 Accessible on: https://www.lifewire.com/definition-of-apache-816509
[19] Understanding SSL gigabit Ethernet accelerator adapters in power edge servers. Date accessed 14th August, 2019 [Online]. Available: http://wwwl.us.dell.com/content/topics/global.aspx/power/en/
[20] Web media SSL terms. [Online] Date Accessed: 15th August, 2019 Available: http://www.webopedia.com/TERM/S/S L.html
[21] W. Chou, "Inside SSL: accelerating secure transactions," IEEE IT Pro. September/October 2002, pp. 37-41.
[22] Polit D.F and Beck C.T.(2008). Nursing Research: Generating and Assessing Evidence for Nursing Practice, 8th Edition. Lipponcott Williams and Wilkins. NY, USA.
[23] Malathi Veeraraghavan and Mark Karol, Internetworking connectionless and connection-oriented networks, http://www.ece.virginia.edu/mv/pdf-files/bss99.pdf

Author Profile

Simon Peter Khabusi completed a Bachelor Degree of Computer Engineering from Busitema University in 2016. He is currently pursuing M.Tech in CSE at DTU, expecting to graduate in 2020.

Prof. Rajni Jindal is a Head of Department of Computer Science & Engineering at Delhi Technological University and Research supervisor. She holds the following qualifications; PhD, ME, MCA.