Impact of Implementing HTTP/2 in Web Services

Nagy Ramadan Darwish
Department of Information Systems and Technology
Institute of Statistical Studies and Research

Ihab Mohamed Abdelwahab
Department of Information Systems and Technology
Institute of Statistical Studies and Research

ABSTRACT

HTTP/2 is the newest version of the HTTP1.1 protocol that was finalized in May 2015 and introduced as the IETF standard for web communication. HTTP/2 provides significant performance improvements by addressing well-known problems with HTTP/1.1 (e.g., head of line blocking and redundant headers) some of this features may have indirect impact in security. Also, HTTP/2 introduces new features like the default encryption which causes traffic hiding consequently affects a number of services (e.g., web Caching, Traffic classification).HTTP/2 may have some problems (vulnerabilities) like any new develop protocol lead to Denial of Service (DoS) attacks .The research try to figure out the pros and cons of the this new protocol version from different aspect specially security issues.

Keywords

DoS, IETF, Vulnerabilities, Caching

1. INTRODUCTION

The Hypertext Transfer Protocol version 1.0 (HTTP 1.0) was introduced in 1991 as an application-layer protocol used on the World Wide Web. HTTP 1.0 uses request–response protocol based on a client-server model where the web browser is the client that communicates with the webserver. The webserver hosts the website by the World Wide Web initiative as method to retrieve hypertext markup language (HTML) messages (Berners-Lee) [1]. This protocol has been to transfer data over LAN (Local Area Network, WAN (Wide Area Network) and the World Wide Web. Then, HTTP 1.1 was then lunched in 1999. HTTP 1.1 handled the request–response problem in HTTP 1.0 with the use of persistent connections, pipelining requests on a persistent connection and other problems which step in bringing the web forward. Today the internet services through websites became main getaway for many companies with complex sites designs with many more interconnected dependencies. The HTTP 1.1 is suffering from website's performance requirements and operational needs. The problems with HTTP 1.1 include inadequate use of transmission control protocol (TCP) connections, latency, and instances where one packet holds up the transmission of other packets known as head-of-line blocking (Stenberg,2015) [1].The problems with the existing protocol lead to the development of a new HTTP version to handle these issues. Google's SPDY protocol submitted solutions for HTTP 1.1 problems, but it was never meant to be a full replacement. The SPDY is an application layer protocol on top of TCP. The framing layer of SPDY is optimized for HTTP-like response-request streams enabling web applications that run on HTTP to run on SPDY with little or no modifications. These requests create streams in the session which are bidirectional flow of bytes across a virtual channel within a SPDY session. SPDY also introduces request prioritization.

HTTP/2 is the second release of the HTTP that built on Google's SPDY protocol, HTTP/2 was developed by the IETF's Group and published on May 2015 as RFC 7540. HTTP/2 handles the major concerns with HTTP/1.x (e.g., head of line blocking and redundant headers) furthermore, it provides new features (e.g., header compression, multiplexing and prioritization). HTTP/2 utilizes the stream of frames by enables full request and response multiplexing as shown in fig.1. This changes will make applications faster and improve users' web experience. Of course, there are several capabilities and limitations associated with such a new protocol needed to be addressed. Some of highlight features in HTTP/2 is performance.

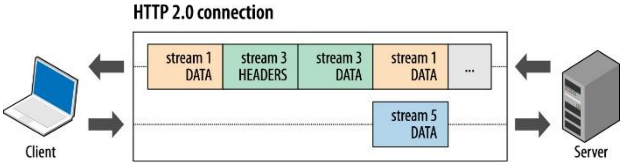


Figure (1): Schema of HTTP2 streaming, by Ilya Grigorik[6]

The following are some of the major features [4]:

- **Multiplexing and concurrency**: Several requests can be sent over the same TCP connection, and responses can be received out of order, eliminating the need for multiple connections between the client and the server and reducing head-of-line blocking. It helps with reducing SSL (Secure Sockets Layer) overhead, avoiding network congestion and improves server efficiency
- **Stream dependencies**: The client can indicate to the server which resources are more important than others.
- **Header compression**: HTTP header size is reduced.
- **Server push**: The server can send resources the client has not yet requested

Also, a new added feature to HTTP/2 is enhancing security by using TLS in internet browsers includes HTTPS encryption.

HTTPS is the secure version of HTTP over SSL/TLS. Many So far, the security investigation and tests for HTTP2 implementations still few, due to a shortage of tools that are capable of inspecting the protocol to detect or prevent attacks against web applications. It is the undisputed future of Internet connections and vulnerabilities in this protocol have the potential to cripple infrastructure [2]. The security guarantee provided by current security technology is inversely proportional to the "size" of the software layer at which the technology applies [3]. This paper explores the impact of HTTP/2 features and some of discovered vulnerabilities in web services over the new HTTP/2 protocol. Also, the paper focusses on security engineering at HTTP/2.

The remainder of the paper is organized in six sections. Section 2 presents HTTP 1.1 limitations. In section 3 The HTTP/2 Protocol is presented. Section 4 shows the related work that was introduced in other papers.. In section 5 explores HTTP/2 performance enchainment .In section 6 proposes HTTP/2 secure connection .Section 7 presents HTTP/2 Attacks. Section 8 represents the paper's conclusion and the research issues that can be focused in future work.

2. HTTP 1.1 LIMITATIONS

HTTP 1.1 was launched in 1999. HTTP 1.1 handles request–response problem in HTTP 1.0 with the use of persistent connections, pipelining requests on a persistent connection and other problems which step in bringing the web forward. HTTP 1.1 was perfect when web sites were much simpler than they are nowadays. On the other hand company web services increasing demands required a new software engineering techniques help us to build larger, more complex systems. Systems have to be built and delivered quickly; larger, complex systems are required; systems must to have new capabilities that may be previously to be impossible[5].one of this examples accessing web application (e.g. online e-commerce system) near-real-time responsiveness which HTTP 1.1 cannot achieve mainly due to the following limitations :- (1) HTTP/1.x clients need to use multiple connections to achieve concurrency and reduce latency,(2) HTTP/1.x does not compress request and response headers, causing unnecessary network traffic, (3)HTTP/1.x does not allow effective resource prioritization, resulting in poor use of the underlying TCP connection; and so on [6]. As mentioned above the demands changed and more complex systems are now required, like the web applications continued to grow in their scope and became very important in our everyday life. This reflect how the developers and users were suffering from the existing web protocol, which is the exact gap that HTTP/2 was designed to address. (4) HTTP 1.1 Initiates object transfers strictly by the client, this presents a serious problem because it hurts performance significantly in the case of loading embedded objects. The servers have to wait for an explicit request from the client which can only be sent after the client processed the parent page [3].

3. THE HTTP/2 PROTOCOL

HTTP/2 is designed to enhance the communication speed between web clients (e.g. browsers) and web servers. It is based on Google's SPDY protocol which addresses the HTTP 1.1 slow response problems through message multiplexing (i.e. multiple requests/responses in one TCP connection per origin). HTTP/2 multiplexing can technically be described as follows, HTTP/2 requests and responses are broken down to binary frames. Each frame in any flow direction can be grouped based on its stream ID, as illustrated in Fig. 2[8].

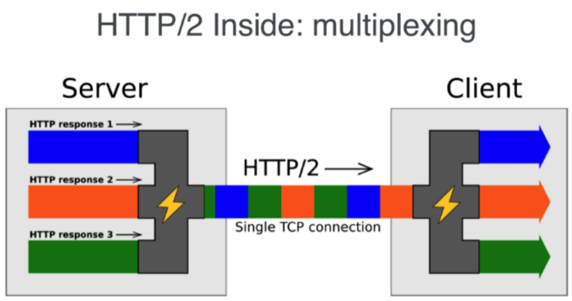


Figure (2): Multiplexing a Connection through Streams [8]

Also, this protocol optimizes bi-directional connections in which both the server and client are free to communicate with each other. Connections such as this, along with other new features of the HTTP/2 protocol, will significantly enhance web application communications by increasing communication efficiency [1]. In order to achieve multiplexing, HTTP/2 messages are broken down into independent binary frames according to their type for example, header, data, setting and control frame and the protocol allows these frames to be interleaved , prioritized within one TCP connection [7]. HTTP/2 has different mechanisms from HTTP/1.1, many of which demand more computing resources. This implies that an HTTP/1.1-enabled web server should closely monitor its resource utilization when HTTP/2 is enabled. The HTTP/2-standard states that if the host machine does not monitor resource usage, it exposes itself to a risk of a DoS attack [10]. 80% of websites supporting HTTP/2 experience in page load time reduction compared with HTTP/1.1 and the reduction grows in mobile networks.

4. RELATED WORK

During the last year several studies has been submitted to discuss HTTP1.1 performance issues, security vulnerabilities and discussed HTTP 2 new features.

- Russel et al. [1] explain the impact of using TLS over HTTP/2 in conjunction with an evaluation of web browser support. Also, the paper evaluate several architectures as a method to detect and prevent web application attacks over HTTP/2 using currently available tools.

- Stuart et al. [2] present an original research at Pacsec 2015 on the HTTP/2 protocol and its security implications. The paper focus on threats, attack vectors, and vulnerabilities found during the course of research. HTTP/2 brings with it a lot of new attack surface. More research needs to be conducted on the implications of this protocol on web security. New tools need to be developed which handle the protocol and allow penetration testers to effectively audit HTTP/2 based web sites.

- Erwin et al. [7] showed that the attack can be launched at the protocol level by sending low-rate HTTP/2 packets to a web server and demonstrate that HTTP/2 is more vulnerable to Denial-of-Service (DoS) attacks than HTTP 1.1. A variant of the DoS type of attack is to send low-rate traffic that contains resource-hungry instructions can succeed only if the victim hosts an application that consumes large-scale computing resources once activated.

- Matteo et al. [9] introduce the new feature in HTTP2 and addresses well-known problems with HTTP/1.1. The authors built a measurement platform that monitors HTTP/2 adoption and performance across the Alexa top 1 million websites on a daily basis for findings from an 11 month measurement campaign (November 2014 – October 2015). The results was 80% of websites supporting HTTP/2 experience a decrease in page load time compared with HTTP/1.1 and the decrease grows in mobile networks.

- Julien et al. [10] discuss the privacy and exposed weaknesses that may be used by a number of actors with intent to cause havoc .The lessons learned from Snowden that pushed the situation from a “no protection" default to a “maximum” protection”, the consequences of ubiquitous encryption and the affecting middle-boxes services .Also, finding an emerging techniques to balance privacy and support of middle-boxes services (cashing, prioritization, optimizations).

- Stefan et al. [11] explore new features in HTTP /2 where TLS has become the de-facto mandatory standard. Most of the modern web browsers (e.g. Chrome, Firefox, and Edge) are now supporting HTTP/2. What we can see an increase in security vulnerabilities, either because of the new protocol and/or because of new implementations from new protocol implementations. Many network forensics tools do currently not support HTTP/2.

- Erwin et al. [12] demonstrate that legitimate HTTP/2 flash crowd traffic can be launched to cause denial of service through presented a DDoS attack model .For varying investigations were conducted to analyze the behavior of a victim machine when subject to large volume, stealthy HTTP/2 traffic through the established connection streams.

- Kyriakos et al. [12] introduce a performance comparison between HTTP 1.1 and HTTP /2. The result shows that HTTP/2 provides significant performance improvements in the tail, and, for websites for which HTTP/2 does not improve median performance. Moreover the paper explore how optimizations like prioritization and push can improve performance, and how these improvements relate to page structure.

- David et al. [13] explorer that increased user concern over security and privacy on the Internet has led to widespread adoption of **HTTPS**. However, HTTPS may introduce overhead in terms of infrastructure costs, communication latency, data usage, and energy consumption. Moreover, given the opaqueness of the encrypted communication. The results show that, indeed, security does not come for free and more researches needed to enhance the cost of the “S” in HTTPS.

The above studies concern on the new protocol HTTP/2. Such studies are trying to address the security issues in HTTP/2 and highlight the different aspect of protocol implementation.

5. HTTP 2 PERFORMANCE ENCHAINMENT

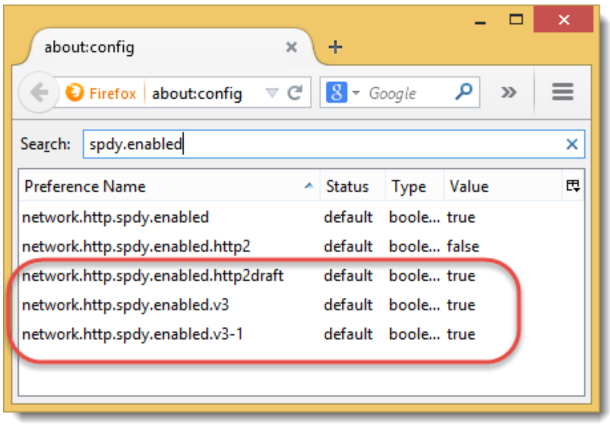
The main objective of developing HTTP/2 is to handle HTTP/1.1 problems. One of HTTP/1.1 biggest problem is performance issue and minimized latency. There is new features as we mentioned early e.g. (Multiplexing, Framing, Header compression) will resolve big part of this problem. Many studies tried to measure the performance enhancement. The following Performance Comparison held by Neumetrix Limited company (http://www.httpwatch.com) .The performance test used HttpWatch with Firefox to run a series of simple page load tests against the Google UK home page using the three protocols [14[:

- Raw HTTPS

- SPDY/3.1

- HTTP/2

The paper will explorer two types of test: Size of Request & Response Headers test and Page Load Time test. The Comparison switched between the protocols by enabling and disabling the following entries in Firefox’s about config page Fig. 3 [14]:

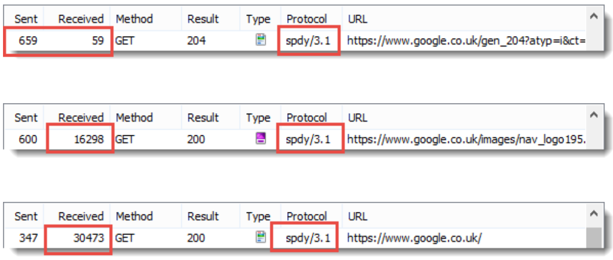


Search: spdy.enabled

Preference Name	Status	Type	Value
network.http.spdy.enabled	default	boole...	true
network.http.spdy.enabled.http2	default	boole...	false
network.http.spdy.enabled.http2draft	default	boole...	true
network.http.spdy.enabled.v3	default	boole...	true
network.http.spdy.enabled.v3-1	default	boole...	true

Figure (3): Firefox settings [14]

- Size of Request and Response Headers Test :- The result shows that HTTP/2 is the winner and it has significantly smaller header sizes due to its use of the HPACK algorithm as per fig. 4 [14]



Sent	Received	Method	Result	Type	Protocol	URL
659	59	GET	204	▤	spdy/3.1	https://www.google.co.uk/gen_204?atyp=i&ct=

Sent	Received	Method	Result	Type	Protocol	URL
600	16298	GET	200	▤	spdy/3.1	https://www.google.co.uk/images/nav_logo195.

Sent	Received	Method	Result	Type	Protocol	URL
347	30473	GET	200	▤	spdy/3.1	https://www.google.co.uk/

Figure (4): Size of Request and Response Headers Test [14]

- Page Load Time Test: The result shows that HTTP/2 is the winner and it was consistently faster than SPDY even though its response messages were often larger [14].

Also, Another life demo from Cloudfar.com to page load . This demo loads 200 image slices in both HTTP/1.1 and HTTP/2. In HTTP/1.1, the browser has to use many separate TCP connections to load the slices. The result as per the following fig. 5 , 6 [15]

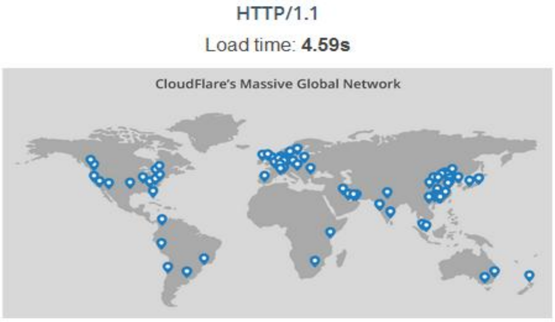


Figure (5): HTTP/1.1 load time. [15]

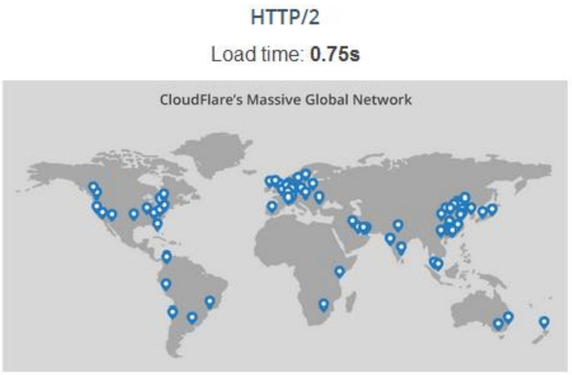
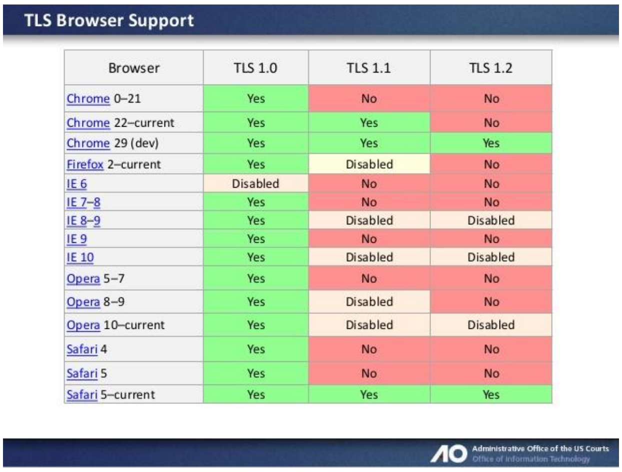


Figure (6): HTTP/2 load time. [15]

The demo shows that HTTP/2 was 6.1x faster than HTTP/1.1[15]. HTTP/2 HTTP/2 is likely to improve website performance, optimized for the modern website and handling the pervious problem like domain sharing and file concatenation.

6. HTTP/2 SECURE CONNECTION

SSL/TLS is the "S" in HTTPS. It is the HTTP encryption layer encoding messages or information from the sender and ensures that the recipient is that only authorized recipients can decrypt the message to see its contents. SSL was developed by Netscape from SSL v1 to SSL v3 till IETF developed SSL v3.1 then renamed it to TLS v1 and then TLS 1.2.TLS and predecessor SSL are often used interchangeably. TLS a more secure and efficient protocol are message authentication, key material generation and the supported cipher suites. TLS a more secure and efficient protocol are message authentication, key material generation and the supported more & new cipher suites. HTTP/2 is enhancing security by using TLS protocol, TLS is not mandatory for HTTP/2 but essential for some web browsers (e.g. Firefox and Chrome) that support HTTP/2 over TLS as per fig.7 [1].



TLS Browser Support

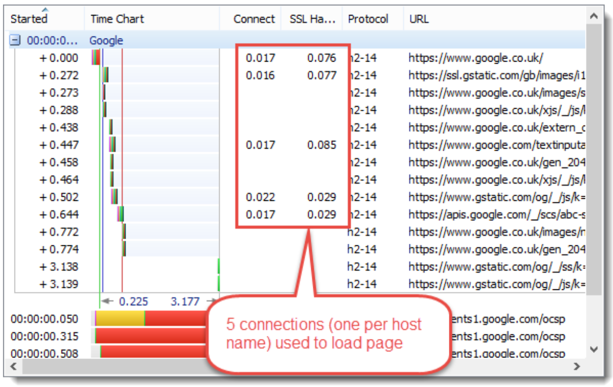
Browser	TLS 1.0	TLS 1.1	TLS 1.2
Chrome 0–21	Yes	No	No
Chrome 22–current	Yes	Yes	No
Chrome 29 (dev)	Yes	Yes	Yes
Firefox 2–current	Yes	Disabled	No
IE 6	Disabled	No	No
IE 7–8	Yes	No	No
IE 8–9	Yes	Disabled	Disabled
IE 9	Yes	No	No
IE 10	Yes	Disabled	Disabled
Opera 5–7	Yes	No	No
Opera 8–9	Yes	Disabled	No
Opera 10–current	Yes	Disabled	Disabled
Safari 4	Yes	No	No
Safari 5	Yes	No	No
Safari 5–current	Yes	Yes	Yes

Administrative Office of the US Courts
Office of Information Technology

Figure 7: Browser Supported TLS 1.2 Cipher Suites [18]

The HTTP/2 browser support HTTPS encryption. HTTPS is the secure version of HTTP over SSL/TLS. Many web services over the internet depend on HTTPS (e.g. Gmail, Facebook, and even YouTube) because it requires data confidentiality or authentication. As well as, other online system like banking payment system (e.g. VISA, MasterCard, PayPal), e-mail system and any corporate published application on the internet. Protecting those critical web systems from attacks is mandatory to keep business running and avoid losses. Per in mind that the secure encrypted traffic cannot be inspected to prevent such as attack. The protocol's use of Perfect Forward Secrecy TLS cipher suites further complicates matters by preventing inspecting technologies from capturing the keying material required to decrypt traffic for inspection PFS ensures that the successful compromise of one session will not allow all other connections between a client and server to be compromised and all non-PFS-enable cipher suites are black listed [1].

Also, HTTP/2 SSL/TLS minimize the number of connection per host than HTTP/1.1. As per the mentioned experimental parameter in the previous section .The following test will measure (Number of TCP Connects and SSL Handshakes Required during Page Load) HTTP/1.1 by increasing the maximum number of connections per host name from two to six or more during the download of a page at the cost of extra TCP connections and SSL handshakes to achieve better performance [14].On the other hand HTTP/2 support concurrency on a single TCP and SSL connection by using multiplexing to allow more than one request at a time to send and receive data on a single connection in the following fig. 8 , 9 [14]:-

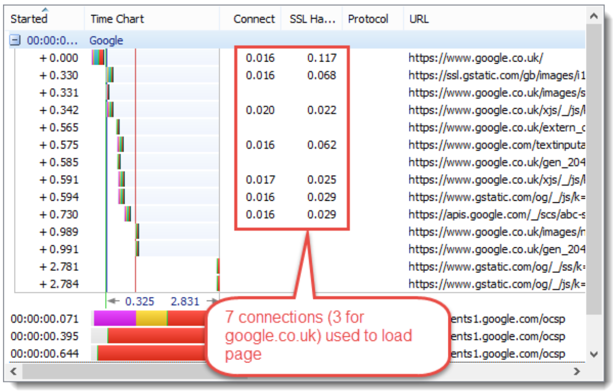


Started	Time Chart	Connect	SSL Ha...	Protocol	URL
00:00:0...	Google				
+ 0.000		0.017	0.076	h2-14	https://www.google.co.uk/
+ 0.272		0.016	0.077	h2-14	https://ssl.gstatic.com/gb/images/i1
+ 0.273				h2-14	https://www.google.co.uk/images/s
+ 0.288				h2-14	https://www.google.co.uk/xjs/_/js/l
+ 0.438				h2-14	https://www.google.co.uk/extern_c
+ 0.447		0.017	0.085	h2-14	https://www.google.com/textinputa
+ 0.458				h2-14	https://www.google.co.uk/gen_204
+ 0.464				h2-14	https://www.google.co.uk/xjs/_/js/l
+ 0.502		0.022	0.029	h2-14	https://www.gstatic.com/og/_/js/k=
+ 0.644		0.017	0.029	h2-14	https://apis.google.com/_/scs/abc-s
+ 0.772				h2-14	https://www.google.co.uk/images/r
+ 0.774				h2-14	https://www.google.co.uk/gen_204
+ 3.138				h2-14	https://www.gstatic.com/og/_/ss/k=
+ 3.139				h2-14	https://www.gstatic.com/og/_/js/k=

00:00:00.050	ents1.google.com/ocsp
00:00:00.315	ents1.google.com/ocsp
00:00:00.508	ents1.google.com/ocsp

5 connections (one per host name) used to load page

Figure (8): HTTP/2 Number of TCP Connects and SSL Handshakes)[14]



Started	Time Chart	Connect	SSL Ha...	Protocol	URL
00:00:0...	Google				
+ 0.000		0.016	0.117		https://www.google.co.uk/
+ 0.330		0.016	0.068		https://ssl.gstatic.com/gb/images/i1
+ 0.331					https://www.google.co.uk/images/s
+ 0.342		0.020	0.022		https://www.google.co.uk/xjs/_/js/l
+ 0.565					https://www.google.co.uk/extern_c
+ 0.575		0.016	0.062		https://www.google.com/textinputa
+ 0.585					https://www.google.co.uk/gen_204
+ 0.591		0.017	0.025		https://www.google.co.uk/xjs/_/js/l
+ 0.594		0.016	0.029		https://www.gstatic.com/og/_/js/k=
+ 0.730		0.016	0.029		https://apis.google.com/_/scs/abc-s
+ 0.989					https://www.google.co.uk/images/r
+ 0.991					https://www.google.co.uk/gen_204
+ 2.781					https://www.gstatic.com/og/_/ss/k=
+ 2.784					https://www.gstatic.com/og/_/js/k=

00:00:00.071	ents1.google.com/ocsp
00:00:00.395	ents1.google.com/ocsp
00:00:00.644	ents1.google.com/ocsp

7 connections (3 for google.co.uk) used to load page

Figure (9): HTTP/1.1 Number of TCP Connects and SSL Handshakes [14]

One of the major concerns for not mandating HTTP/2 SSL/TLS is the cost of web server certificate but there is an open source organization offer this certificate full free e.g. (https://letsencrypt.org) .This encourage around 5 billion website to use this server certificate as they mentioned in their web site .The anchor for any TLS-protected communication is a public-key certificate which demonstrates that the server you're actually talking to is the server you intended to talk to[16]. Sometimes deploying Server certificate is not easy for web server administrator and takes time and cost. However TSL protect user data and privacy.

7. HTTP/2 ATTACKS

Recently, researches held to analyze the new web protocol traffic especially there not enough developed tools to check the vulnerabilities, bugs or attack. HTTP/2 is designed to improve reliability and performance this enhancements have defined the protocol as being more vulnerable to distributed denial of-service (DDoS) attacks (flooding technique), or by exploiting a bug (vulnerability) in the target system's software that incapacitates the service [7]. Erwin et al. showed that the low-rate DoSs attack can be launched at the protocol level by sending low-rate HTTP/2 packets to a web server, demonstrated test Cases 1 to 5 and how the attacks can be launched consume resources as per TABLE 1 [7].

Table 1[7]. Computing Resource Consumption During Attacks

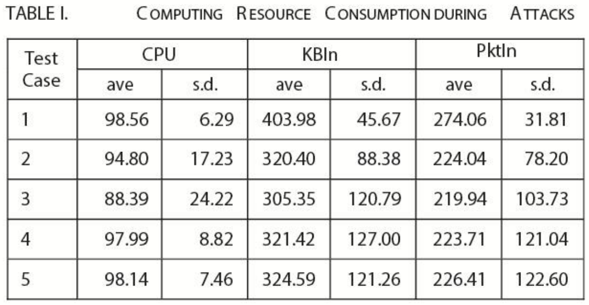


TABLE I. COMPUTING RESOURCE CONSUMPTION DURING ATTACKS

Test Case	CPU		KBIn		PktIn	
	ave	s.d.	ave	s.d.	ave	s.d.
1	98.56	6.29	403.98	45.67	274.06	31.81
2	94.80	17.23	320.40	88.38	224.04	78.20
3	88.39	24.22	305.35	120.79	219.94	103.73
4	97.99	8.82	321.42	127.00	223.71	121.04
5	98.14	7.46	324.59	121.26	226.41	122.60

In addition to the last Low rate DOS attack, a HTTP/2 recent phenomenon showed that legitimate traffic or flash crowds could have high-traffic flow characteristics as seen in DDoS attacks for example in test case1 the table 2 show the attack consumed the server memory up to 168 MB [12].

Table 2 [12] CPU and memory consumption under DoS attack

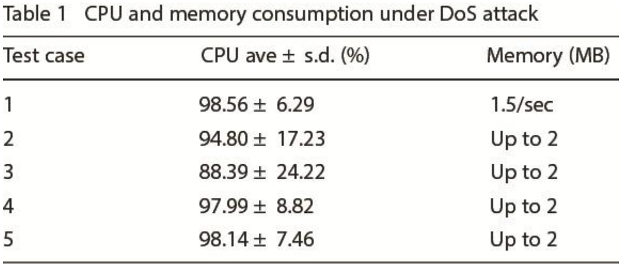


Table 1 CPU and memory consumption under DoS attack

Test case	CPU ave ± s.d. (%)	Memory (MB)
1	98.56 ± 6.29	1.5/sec
2	94.80 ± 17.23	Up to 2
3	88.39 ± 24.22	Up to 2
4	97.99 ± 8.82	Up to 2
5	98.14 ± 7.46	Up to 2

Other research submitted by Yahoo pen test team members discover the following attacks:

- HPACK Upgrades / Downgrades
- Inconsistent Multiplexing
- Malformed Frames
- Pushing arbitrary data to client
- Pushing arbitrary data to server
- Stream dependencies
- Invalid Frame States

Due to the new attack surface there is a needed for an automated code test coverage in HTTP2 implementations [2]. Also, other type of attack called cross protocol attacks since there will be implementations that will support the different versions of the HTTP protocol, both HTTP/1.x and HTTP/2. In a cross-protocol attack, an adversary causes a client to initiate a transaction in one protocol toward a server that understands a different protocol [11].As a result there is an increase in security vulnerabilities, either because of the new protocol and/or because of new implementations.

8. CONCLUSION AND FUTURE WORK

This paper is trying to highlight the impact of security function as well as other HTTP/2 functions that may lead to threats. Although there is great contribution from many parties either individual or biggest companies e.g. (Google have huge capabilities to perform a good product test prior to its lunch) but the HTTP/2 protocol not tested well before lunch as expected. As mentioned before in the recommend model "A Security Testing Framework for Scrum based Projects" Software testing is done in integrated environment to discover the bugs from a real life environment in addition to lab environment and fix such bugs before the product lunch saving cost. Also, considering a security function as an important factor in software requirement engineering phase and gives it the high weight like performance function. This will help to enhance the security, minimize bugs and attacks. Using HTTP/2 TLS will protect user data and privacy. In future, we suggest to develop Middle boxes &tools to cope with TLS protocol and protect the users form the attacks

9. REFERENCES

[1] R. Tuyl and S. Northcutt, "Practical Approach to Detecting and Preventing Web Application Attacks over HTTP/2 ", SANS Institute Reading Room site, April 6, 2016.

[2] S. Larsen and J. Villamil, "Attacking HTTP/2 Implementations", Pacsec 2015.

[3] Peter Megyesi, Zsolt Kramer and Sandor Molnar, " Comparison of web transfer protocols",

[4] "Turn-on HTTP/2 today!", [Online]. Available: https://http2.akamai.com/. [Accessed: 02- June- 2016].

[5] I.Sommerville, "Software Engineering", 10th Edition, Chapter (1), ISBN-13: 9780133943276, PP. 4-26, 2016.

[6] I. Grigorik, "High Performance Browser Networking", Chapter (12), ISBN: 978-1-4493-4476-4, PP. 214, September, 2013.

[7] E.Adi, Z. Baig, C.Lam and P. Hingston, "Low-Rate Denial-of-Service Attacks against HTTP/2 Services" The 5th International Conference on IT Convergence and Security, 17 May 2016.

[8] "7 Tips for Faster HTTP/2 Performance", 2015. [Online]. Available: https://www.nginx.com/blog/7-tips-for-faster-http2-performance/ .[Accessed: 30- June- 2016].

[9] M. Varvello, K. Schomp, D. Naylor, J .Blackburn, A. Finamore, and K. Papagiannaki , "Is The Web HTTP/2 Yet?" , Passive and Active Measurement Conference ,Volume 9631, pp 233-247, 24 March 2016 .

[10] J. Maisonneuve, V. Gurbani, T. Fossati "The security pendulum", Internet Architecture Board, August 2015.

[11] S. Winkel, C. Walker "Network Forensics and HTTP/2" , SANS Institute Reading Room site , December 27, 2015 .

[12] E.Adi, Z. Baig, C.Lam and P. Hingston , "Distributed denial-of-service attacks against HTTP/2 services" Cluster Computing , Volume 19, Issue 1, pp 79-86 , March 2016 .

[13] D.Naylor, A. Finamorey, I. Leontiadisz, Y. Grunenbergerz, M. Melliay, M. Munafòy, K. Papagiannakiz, and P. Steenkiste," The Cost of the "S" in HTTPS" , the 10th ACM International on Conference on emerging Networking Experiments and Technologies,pp 133-140,2014 .

[14] "A Simple Performance Comparison of HTTPS, SPDY and HTTP/2",2015. [Online]. Available: http://blog.httpwatch.com/2015/01/16/a-simple-performance-comparison-of-https-spdy-and-http2/comment-page-1/ . [Accessed: 30- June- 2016].

[15] "CloudFlare HTTP/2, Reload Demo "",2016. [Online]. Available: https://www.cloudflare.com/http2/ . [Accessed: 30- June- 2016].

[16] "Let's Encrypt: Delivering SSL/TLS Everywhere", 2014.]Available[Online],https://letsencrypt.org/2014/11/18/announcing-lets-encrypt.html. [Accessed: 30- June-2016].

[17] K. Zarifis, M. Holland, M .Jain,E. Katz-Bassett, R. Govindan, "Modeling HTTP/2 Speed from HTTP/1 Traces", Passive and Active Measurement Conference ,Volume 9631, pp 218-232, 24 March 2016.

[18] "SSL Cipher Suites, "2013. [Online]. "http://www.slideshare.net/tgbenson/ssl-overview-28892698/'. [Accessed: 03- August- 2016].

[19] N. Ramadan, I. Abdelwahed, "A Security Testing Framework for Scrum based Projects". International Journal of Computer Applications, March2016.