Securing Video Streaming Over Internet Protocol Version 6(IPv6)

Cheng Kian Yong [*1], **Azizol Abdullah** [#2], **Mohd. Taufik Abdullah** [#3]

Faculty of Computer Science and Information Technology Universiti Putra Malaysia

Abstract
The worth of a video can never be ignored in this age of Internet. With streaming, the file can be watching as soon as it begins downloading by end user. However, many raise awareness of security threats and vulnerabilities that exit all around the Internet, thus security becomes a key problem to be handled when valuable multimedia assets are floating over the network. This paper presents secure video streaming over Internet Protocol Version 6. The paper proposed client authentication in defending Hypertext Transfer Protocol Secure (HTTPS) server against active man in the middle attack. The process of authorization is done by allowing the server access to only authorized users. The cipher suites described in this paper use Camellia in cipher block chaining (CBC) mode as a bulk cipher algorithm, keys generated through DHE_RSA, DHE_DSS, and RSA.A keyed-hash algorithm is used SHA-1 to generate the Hashed Message Authentication Code (HMAC) for every HTTPS packet. After all these encryption process, the data is finally traveling over the network media. As a result, this paper hopes to apply the best possible security on video streaming among the Internet.

Index Terms
Authorization, client authentication, cipher block chaining, DHE_DSS, DHE_RSA, HMAC, HTTPS, IPv6, man in the middle attack, RSA, SHA-1, Video Streaming.

1. BACKGROUND

In the technology driven world, the worth of a video can never be ignore because many of people prefer watching and participating on an on-line seminar than read the static text as well as business today, every on-line business today require some form of video streaming for communicating with their client or associates. These video streaming can either transmitted through Transmission Control Protocol (TCP) or User Datagram Protocol (UDP).

TCP is a connection oriented protocol, which designed to provide reliable connection over the Internet [1]. TCP use sequence number to detect lost or duplicated packets and to determine the next expected packet. When a connection time out, retransmission of packet is done automatically to ensure the user at the other end could receive all information he sent. A TCP communication begins with three-way handshake. First, the client sends a connection request to server using a synchronization (SYN) packet. Then the server responses packet with SYN and acknowledgement (ACK) flags on. After the server receives an acknowledgement from the client, connection is established. A new child process will be created at the server to handle client requests and information such as the incoming Internet Protocol (IP) address, outgoing IP address, incoming port number, outgoing port number and sequence number will be kept in TCP Control Block (TCB) [1].

However, many raise awareness of security threats and vulnerabilities that exit all around the Internet, such as active attack. Active attack define as an unauthorized change of a system, in which attacker does not merely eavesdrop, but take action involves change, delete, reroute, creation of a false stream or some modification of the data stream. Active attack can be subdivided into four categories: masquerade, replay, modification of message, and denial of service. In current login server communication, client login process starts after the connection establishment. Since the client stays anonymously while resources were allocated, it can be easily exploited and resources can be consumed.

Hence, it is important to tailor security standard during video streaming such as X.800 security standard to provide data integrity, authentication, data confidentiality, and non-repudiation. Data integrity is a mechanism for enforcing that a video is viewed under the rules specified by content owner. Authentication is the verification of the integrity to assurance that the communicating identity of the source, identity of the receiver, and the integrity of the video stream during delivery can be verified. Data confidentiality is the protection of data from unauthorized disclosure. Non-repudiation is a mechanism for controlling the number of copied of a video stream a user is allowed of a signature of serial number into the video stream for subsequent identification [15].

2. INTRODUCTION

A. Problem Statement

In this age of Internet, the worth of a video can never be ignored. Nowadays, there are many video player software example, VideoLAN media player, allow client to stream video from a server, but it require client to have the software installed on the client's computer before streaming.

Unsecured application streaming become an issue when server and client transfer valuable and confidential

Manuscript received November 5, 2012
Manuscript revised November 20, 2012

information for a variety purpose. This happened because software streaming was established through insecure communication. As a consequence, this valuable and confidential information attract the attention of people who intend to steal or misuse the information, or to disrupt the system storing or communicating it. Beside, software streaming uses UDP for transmission. Thus UDP provides an unreliable service and datagram may arrive out of order, missing without notice, or appear duplicated [14]. Another downside to UDP is that many network administrators disable their firewall to UDP traffic, and limiting the potential audience of UDP-based streams [10].

Since attacker able to steal or misuse the information through active attack, so it is important to secure data during transmission from server to client. For example, when client request to stream a video from a server, hacker can use replay attack by capture data from server then replay a fake video to client.

The securities issue for a digital video, whether it is transmitted over Internet or not, are generally the following: authentication, data confidentiality, non-repudiation, data integrity. Due to traditional cryptography are not intended for large continuous media, especially video, and are not designed for streaming media service in heterogeneous environment [18], hence it is important to tailor security standard during video streaming such as X.800 security service.

X.800 defines a security service as a service which provide by a protocol layer of communication open system[15], used to ensure that adequate security of the system or of data transfer. This paper will tailor X.800 security standard during video streaming over secure socket in Internet Protocol version 6 (IPv6).

B. Objective

The objective of this paper is:

(i) Tailor X.800 security standard during video streaming to provide confidentiality, authentication, and non-repudiation on streaming video.

(ii) To secure video streaming over secure socket layer in IPv6.

The objective of this paper is to tailor X.800 security standard to provide data integrity, authentication, data confidentiality, and non-repudiation for streaming video.

C. Scope of Study

This paper will implement web browser streaming using IPv6, in the same time, secure transmit video date over Internet for real-time viewing. Thus, any organization can arrange video seminars for a large group of attendees or select a few and provide them password to log on to the web seminar.

This paper use TCP to stream video. The most important thing use TCP allows congestion control. For example, server can support 100 TCP connections across a DSL link all going at max speed, and all 100 connections will be productive, because they all "sense" the available bandwidth. Compare with 100 different UDP applications connection, all the connection will pushing packets as fast as they can.

HTTP operates on top of the TCP, which handles all the data transfers. The main idea using TCP is to maximize the date transfer rate while ensuring overall stability and high throughput of the entire network.

3. PROPOSED CLIENT AUTHENTICATION

The main goal of this paper is to tailor X.800 security standard during video streaming to provide data confidentiality, authentication, and non-repudiation on streaming video. The proposed client authentication as shown in Fig.2, allow HTTPS server to authenticate user when client establish a secure connection in order to tailor X.800 security standard.

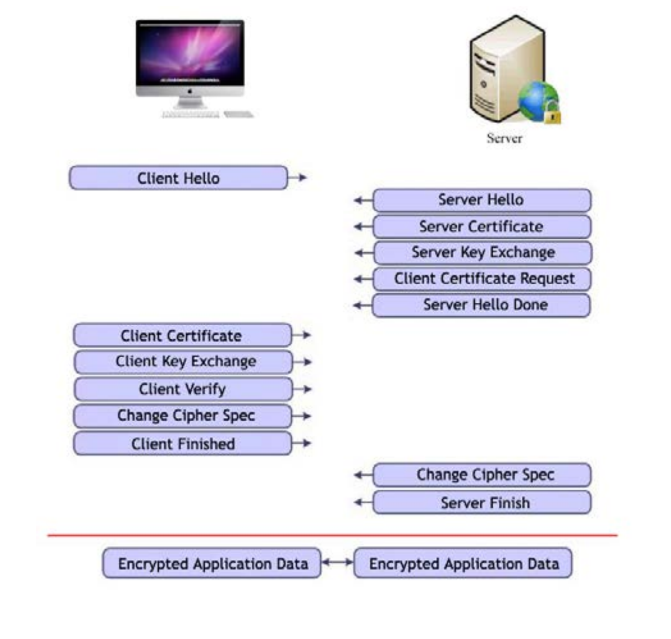


Fig.2: Proposed Client Authentication.

At the beginning of a Secure Socket Layer (SSL) session, the client sends a client “Hello” message which listed the cryptographic capabilities of the client. These cryptographic capabilities include version of SSL, cipher suites, and data compression method that supported by client. The “Hello” message also contains a 28-byte random number. Then server responds with server “Hello” message. Server “Hello” message contains the cryptographic method, session ID, the data compression method selected by the server, and another random number. The server generally chooses the strongest common cipher suite that supported for both client and server. In this paper, the server uses X.509 V3 digital certificates with SSL V3. The server application requires a digital certificate for client authentication, so its send “Client Certificate Request” message. In this message, the servers include a list of types of digital certificates supported and distinguished name of acceptable certificate authorities. After that, server sends “Hello Done” message and waits for client response. For client, once receipt the server “Hello Done” message, client will verify and respond to the validate of the server’s digital certificate. Then client sends a “Client Key Exchange” message which contains the pre-master secret, message authentication code keys, a 46-byte random number used to generation of the symmetric encryption keys, encrypted with the public key of the server and signed with the client’s private key. By verifying the signature of this message, the server can verify the ownership of client digital certificate. To sends a “Change Cipher Spec” message to make the server switch to the newly negotiated cipher suite, the client uses a series of cryptographic operations to convert the pre-master secret into a master secret. The next message send by client, “Finished” message is the first message encrypted with this cipher method and keys. SSL handshake was ends when the server responds with “Change Cipher Spec” and a “Finished” message. Finally, encrypted data can be sent [9].

4. METHODOLOGY

This paper classified the process of video streaming into two main parts: Test-bed I and Test-bed II. Test-bed I tested the test-bed using normal Hypertext Transfer Protocol (HTTP) server and Test-bed II tested the test-bed through HTTPS server.

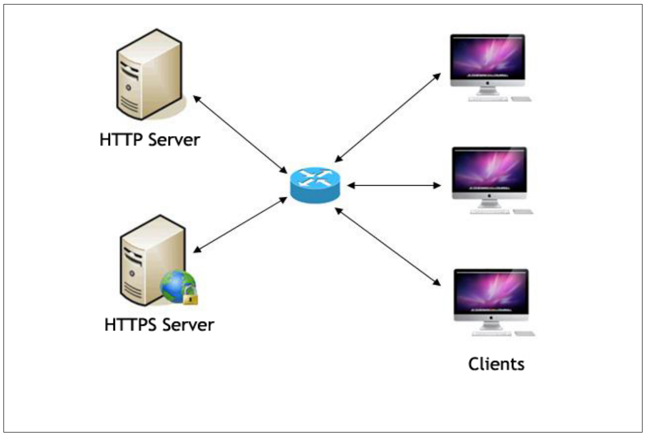


Fig.3: Test-bed Model.

D. Test-bed I

This part of the test-bed present the client establish a connection to HTTP server. After the connection was establish, client require to login to the web site in order to stream video using IPv6. To show HTTP server is unsecured, wireshark used to capture the transmission packet. After client successful login to the HTTP server, the client was allow to stream video. During the streaming, performance of HTTP server is measured.

E. Test-bed II

In this part of the test-bed, a client was establishing a secure connection to HTTPS server. After the secure connection was establish, client require to login to the secure web site to stream video using IPv6. Wireshark used to capture the transmission packet between client and server. After client successful login to the HTTPS server, the client was allow to stream video. During the streaming, performance of HTTPS server is measured.

5. MEASUREMENT

The parameter use to measure the performance between both HTTP and HTTPS server is same, which is time sequence, network interface statistic monitoring, and throughput.

F. Time Sequence

Time sequence graphs show the general activity and event that happen during the lifetime connection.

G. Bandwidth Monitoring

This paper use ifstat to monitoring network interface statistic monitoring. ifstat is a tool use to report network interface bandwidth.

H. Throughput

Throughput is a measurement of the rate at which data can be successful sent through the network.

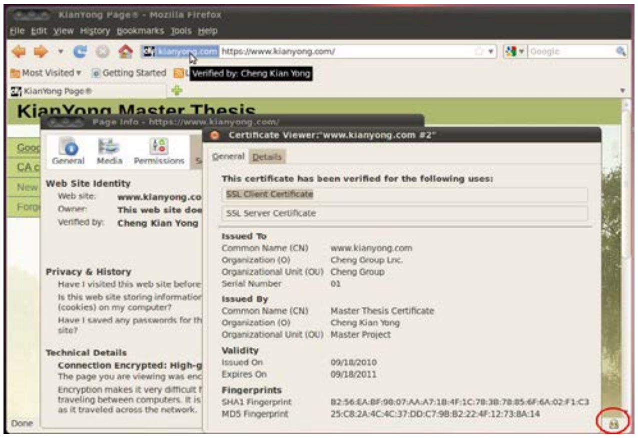
6. RESULT AND DISCUSSION

This section discusses the outcome obtained from the paper. The paper compare benefits of HTTPS and evaluate the performance between HTTP and HTTPS, and finally shows man in the middle attack and man in the middle prevention mechanism.

I. Analyzing TCP Stream for HTTPS

HTTPS is identical to HTTP because it follow the same basic protocols, both system use same URL but different port, port 80 for HTTP and port 443 for HTTPS.

Data transfer using HTTP protocol is in American Standard Code for Information Interchange (ASCII) based, however HTTPS transfer data through an encrypted protocol. When using an HTTPS connection server responds to the initial connection by offering a list of encryption methods it supports, in response, client select a connection method and the client and server exchange certificate to authenticate their identities. In order to host HTTPS connection, server must have a public key certification. Most modern browser display a “lock” icon in the status bar or possibly in the address field [13], as illustrated in Fig.4, when a secure HTTPS website is being accessed, generally user can click on the lock icon to display more information about the secure website.


KianYong Page® - Mozilla Firefox
File Edit View History Bookmarks Tools Help
kianyong.com https://www.kianyong.com/
Google
Most Visited
Getting Started
Verified by: Cheng Kian Yong
KianYong Page®
KianYong Master Thesis
Page Info - https://www.kianyong.com/
General
Media
Permissions
Done

Web Site Identity
Web site: www.kianyong.co
Owner: This web site doe
Verified by: Cheng Kian Yong

Privacy & History
Have I visited this web site before
Is this web site storing informatior (cookies) on my computer?
Have I saved any passwords for th site?

Technical Details
Connection Encrypted: High-g
The page you are viewing was enc
Encryption makes it very difficult t traveling between computers. It is as it traveled across the network.

Certificate Viewer:"www.kianyong.com #2"
General Details
This certificate has been verified for the following uses:
SSL Client Certificate
SSL Server Certificate

Issued To
Common Name (CN) www.kianyong.com
Organization (O) Cheng Group Lnc.
Organizational Unit (OU) Cheng Group
Serial Number 01

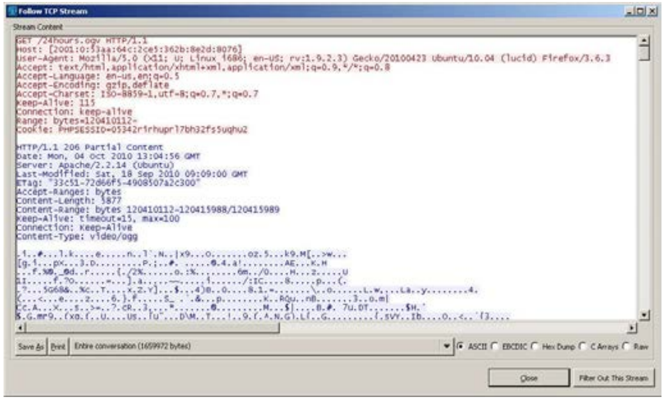
Issued By
Common Name (CN) Master Thesis Certificate
Organization (O) Cheng Kian Yong
Organizational Unit (OU) Master Project

Validity
Issued On 09/18/2010
Expires On 09/18/2011

Fingerprints
SHA1 Fingerprint B2:56:EA:BF:98:07:AA:A7:1B:4F:1C:78:38:78:85:6F:6A:02:F1:C3
MD5 Fingerprint 25:C8:2A:4C:4C:37:DD:C7:9B:B2:22:4F:12:73:8A:14

Fig.4: HTTPS connection.

Fig.5 showed TCP Stream for HTTP. The figure shows that the stream content is displayed in the same sequence as it appeared on the network in plaintext. However, HTTPS using encrypted protocol as illustrated in Fig.6, the stream content displayed was encrypted.



```
GET /24hours.ogv HTTP/1.1
Host: [2001:0:53aa:64c:2ce5:362b:8e2d:8076]
User-Agent: Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.9.2.3) Gecko/20100423 Ubuntu/10.04 (lucid) Firefox/3.6.3
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 115
Connection: keep-alive
Range: bytes=120410112-
Cookie: PHPSESSID=05342rirhupr17bh32fs5uqhu2

HTTP/1.1 206 Partial Content
Date: Mon, 04 Oct 2010 13:04:56 GMT
Server: Apache/2.2.14 (Ubuntu)
Last-Modified: Sat, 18 Sep 2010 09:09:00 GMT
ETag: "33c51-72d66f5-4908507a2c300"
Accept-Ranges: bytes
Content-Length: 5877
Content-Range: bytes 120410112-120415988/120415989
Keep-Alive: timeout=15, max=100
Connection: Keep-Alive
Content-Type: video/ogg
```

Fig.5: HTTP TCP Stream.

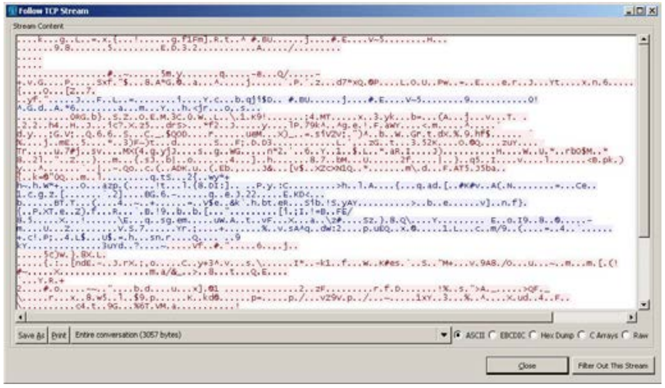
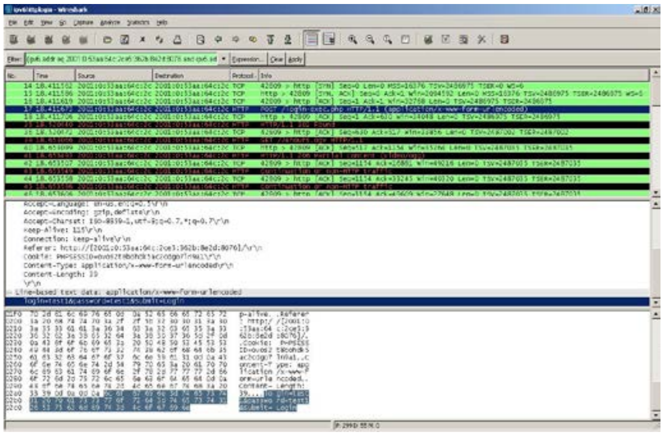


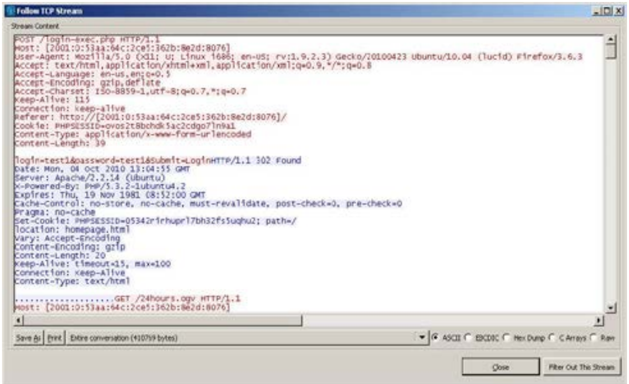
Fig.6: HTTPS TCP Stream.

Fig.7 and Fig.8 shows HTTP login packet captured from wireshark. From the packet captured, attacker was able to view user’s user name and password in plaintext.



```
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7\r\n
Keep-Alive: 115\r\n
Connection: keep-alive\r\n
Referer: http://[2001:0:53aa:64c:2ce5:362b:8e2d:8076]/\r\n
Content-Type: application/x-www-form-urlencoded\r\n
Content-Length: 39
\r\n
Line-based text data: application/x-www-form-urlencoded
login=test1&password=test1&submit=Login
```

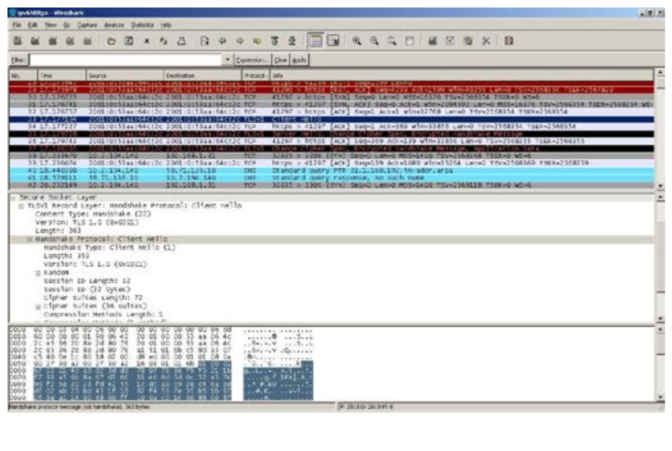
Fig.7: HTTP login packet



```
POST /login-exec.php HTTP/1.1
Host: [2001:0:53aa:64c:2ce5:362b:8e2d:8076]
User-Agent: Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.9.2.3) Gecko/20100423 Ubuntu/10.04 (lucid) Firefox/3.6.3
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 115
Connection: keep-alive
Referer: http://[2001:0:53aa:64c:2ce5:362b:8e2d:8076]/
Cookie: PHPSESSID=ovos2t8bchdk5ac2cdgo7ln9a1
Content-Type: application/x-www-form-urlencoded
Content-Length: 39

login=test1&password=test1&Submit=LoginHTTP/1.1 302 Found
Date: Mon, 04 Oct 2010 13:04:55 GMT
Server: Apache/2.2.14 (Ubuntu)
X-Powered-By: PHP/5.3.2-1ubuntu4.2
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Pragma: no-cache
Set-Cookie: PHPSESSID=05342rirhupr17bh32fs5uqhu2; path=/
location: homepage.html
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 20
Keep-Alive: timeout=15, max=100
Connection: Keep-Alive
Content-Type: text/html

....................GET /24hours.ogv HTTP/1.1
Host: [2001:0:53aa:64c:2ce5:362b:8e2d:8076]
```

Fig.8: HTTP TCP Stream login packet.

Since HTTPS server establish connection through secure connection, all the data was in cipher text thus attacker needs to decrypt the cipher suites packet to get user's user name and password as shown in Fig.9.



ipv6-https - Wireshark
File Edit View Go Capture Analyze Statistics Help
Filter: Expression... Clear Apply
No. Time Source Destination Protocol Info

```
Secure Socket Layer
  TLSv1 Record Layer: Handshake Protocol: Client Hello
    Content Type: Handshake (22)
    Version: TLS 1.0 (0x0301)
    Length: 363
    Handshake Protocol: Client Hello
      Handshake Type: Client Hello (1)
      Length: 359
      Version: TLS 1.0 (0x0301)
      Random
      Session ID Length: 32
      Session ID (32 bytes)
      Cipher Suites Length: 72
      Cipher Suites (36 suites)
      Compression Methods Length: 1
```

Fig.9: HTTPS login packet.

J. *Performance Analysis of Secure Streaming*

This paper discusses the performance of secure streaming using TCP over IPv6. The parameter use to measure the performance between both HTTP and HTTPS server is same, which is time sequence, network interface statistic monitoring, and throughput.

i) Time Sequence Graph

Time sequence graphs show the general activity and events that happen during the lifetime connection [12]. Fig.10 illustrated time sequence graph for HTTP connection. Fig.11 shows time sequence graph for HTTPS connection. Table 1 shown time sequence performance.

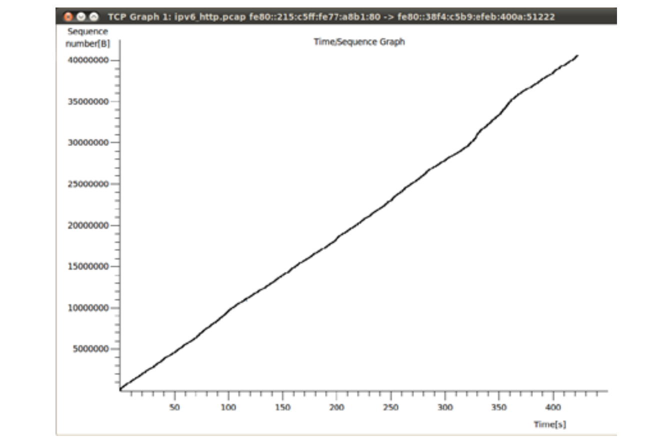


Fig.10: HTTP Time Sequence Graph.

The X-axis represents time, and Y-axis represents sequence number space. The slope of this curve gives the throughput over time. The TCP time sequence number versus time graph shows that the traffic is moving along without interruption, packet loss or long delay.

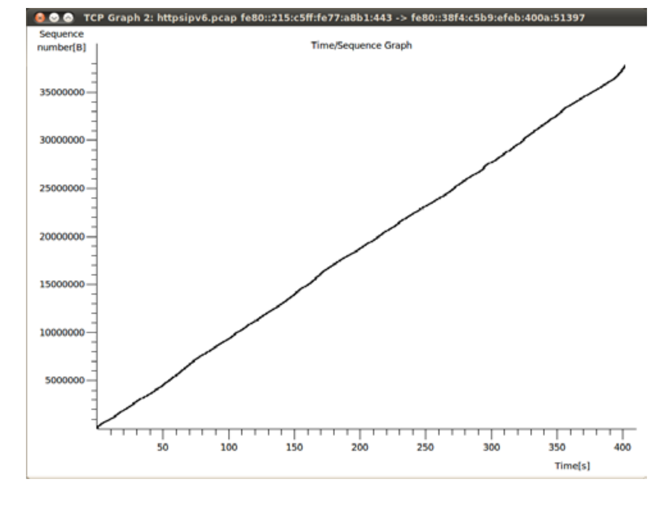


Fig.11: HTTPS Time Sequence Graph.

Table 1: Time Sequence Performance.

Time (s)	Sequence Number (B)		Different (%)
	HTTP	HTTPS	
50	4700000	4500000	4.2553
100	95000000	93000000	2.1052
150	14000000	13900000	0.7142
200	18200000	18600000	-2.1505
250	23500000	23000000	2.1277
300	27900000	27800000	0.3584
350	33400000	32800000	1.7964
400	38200000	37200000	2.6178
			Average : 1.5

From the table 1 result, average time sequence performance of HTTPS was only slightly about 1.5% slower than HTTP. The different may cause by TCP slow start which will explain detail in bandwidth monitoring. According to paper [6], 2% rate was the threshold where direct HTTP started to degrade on performance and was no longer competitive with the other approaches. [6]. Form the time sequence graph result, this paper shown that HTTPS streaming was slightly about 1.5% slower then HTTP which was accepted threshold for user to perform video streaming.

ii) Bandwidth Monitoring

ifstat used to monitoring network interface statistic. ifstat is a tool use to report network interface bandwidth [19]. Fig.12 and Fig.13 illustrated network interface statistic monitoring for HTTP and HTTPS. The Y-axis represents total data (Kbps) transfer by the network interface (eth0), and X-axis represents time in second.

This paper discusses video streaming through TCP protocol. TCP using an algorithm called slow start to maximize the date transfer rate while ensuring overall stability and high throughput of the entire network, by first send data at low data rate, and then gradually increase the rate until the destination report packet loss [10].

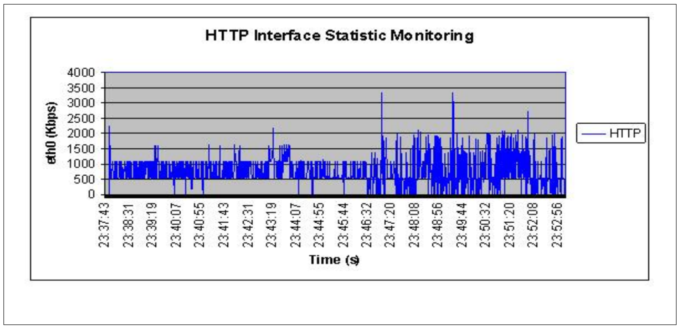


Fig.12: HTTP Interface Statistic Monitoring.

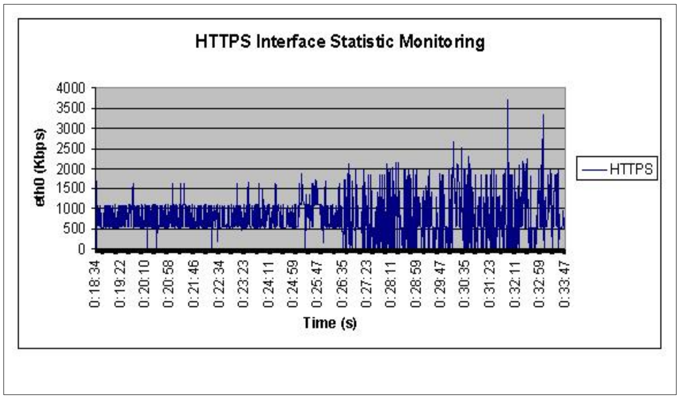


Fig.13: HTTPS Interface Statistic Monitoring.

iii) Throughput Measurement

In communication networks, throughput is the average rate of successful deliver a message over a communication channel. Fig.14 illustrated throughput graph for HTTP connection and Fig.15 shows throughput graph for HTTPS connection.

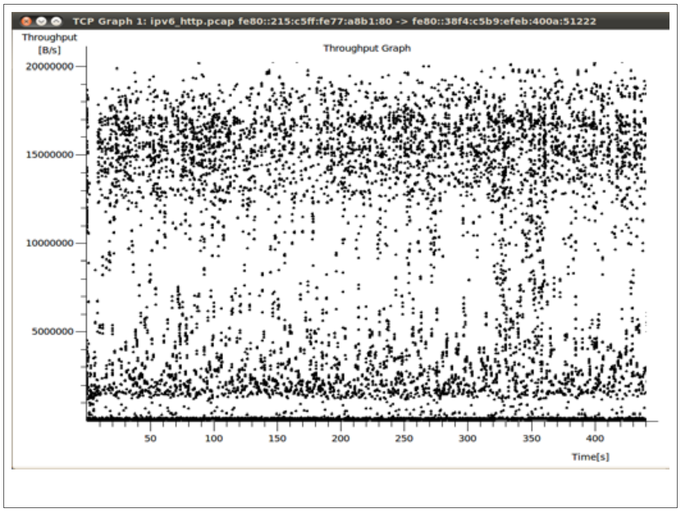

TCP Graph 1: ipv6_http.pcap fe80::215:c5ff:fe77:a8b1:80 -> fe80::38f4:c5b9:efeb:400a:51222

Fig.14: HTTP connection throughput graph.

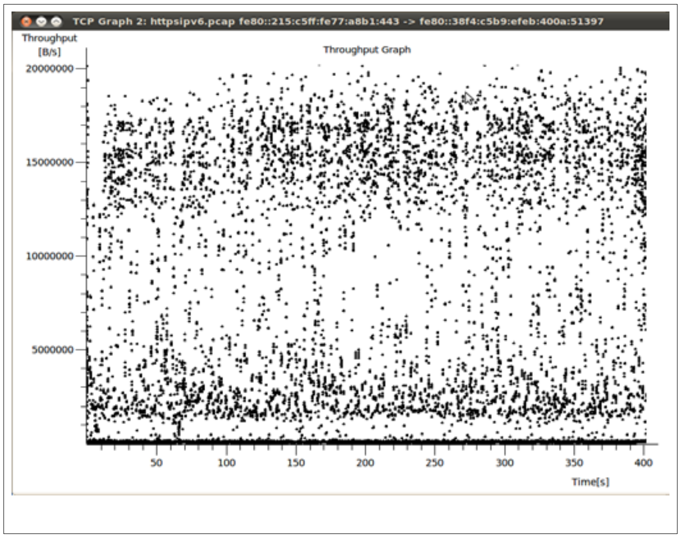

TCP Graph 2: httpsipv6.pcap fe80::215:c5ff:fe77:a8b1:443 -> fe80::38f4:c5b9:efeb:400a:51397

Fig.15: HTTPS connection throughput graph.

Table 2:Throughput Measurement.

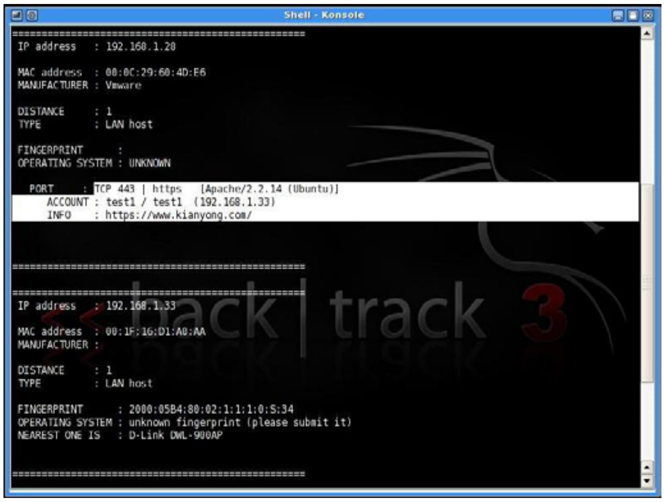
Protocol	Throughput (B/s)		
	Minimum	Maximum	Average
HTTP	0	±20000000	±10000000
HTTPS	0	±20000000	±10000000

From Table 2, the average transfer rate was about 10MB/s over the channel. This shows that the throughput between HTTP and HTTPS is almost same.

In [7], cable modems offers a maximum transmission speed of 10Mbps. From table 2 shown that the average transfer rate for HTTP and HTTPS in this paper was about 10MB/s which are accepted in threshold throughput cable modem rate.

K. Man in the middle attack

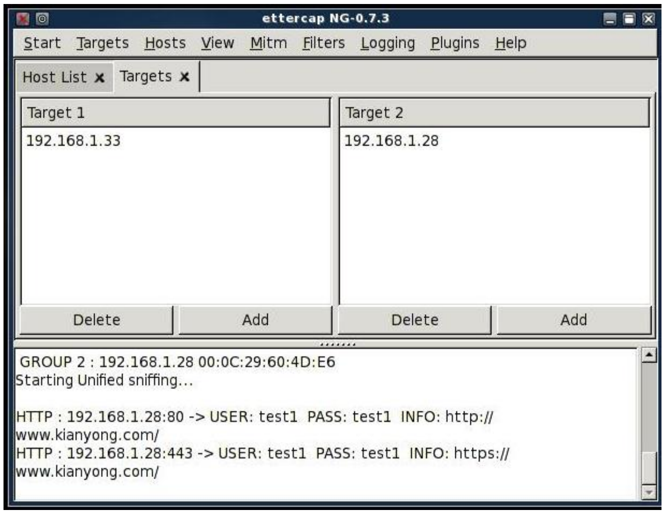
The purpose of HTTPS is to create a secure communication over top of HTTP by the use of Transport Layer Security (TLS) or SSL. SSL/TLS can consider a very effective and secure [11]. However, attacker could attempt to intercept HTTPS traffic by using a custom certificate. This would present a certificate warning message in the user’s browser and likely alert the user to the attack. Since, most of the user would ignore the warning and continue thus exposing all of their data. Fig.16 shows man in the middle attack on HTTPS using backtrack3.



```
==================================================
 IP address   : 192.168.1.28

 MAC address  : 00:0C:29:60:4D:E6
 MANUFACTURER : Vmware

 DISTANCE     : 1
 TYPE         : LAN host

 FINGERPRINT      :
 OPERATING SYSTEM : UNKNOWN

   PORT     : TCP 443 | https   [Apache/2.2.14 (Ubuntu)]
      ACCOUNT : test1 / test1  (192.168.1.33)
      INFO    : https://www.kianyong.com/


==================================================

==================================================
 IP address   : 192.168.1.33

 MAC address  : 00:1F:16:D1:A0:AA
 MANUFACTURER :

 DISTANCE     : 1
 TYPE         : LAN host

 FINGERPRINT      : 2000:05B4:80:02:1:1:1:0:S:34
 OPERATING SYSTEM : unknown fingerprint (please submit it)
 NEAREST ONE IS   : D-Link DWL-900AP


==================================================
```

Fig.16: Blacktrack3 man in the middle attack.

Fig.17 illustrated man in the middle rogue program, etthercap, used to sniff connection between client and server. From the figure, although the communication is established using HTTPS, port 443, however, if user ignore the warning and continue proceed then they will expose all of their data.



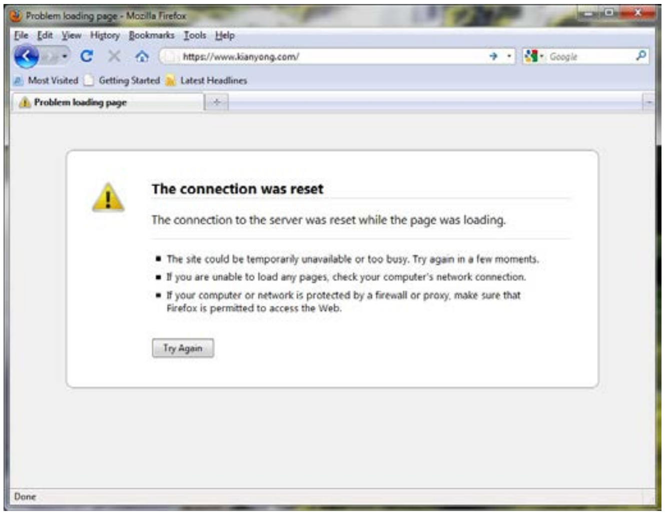
Target 1	Target 2
192.168.1.33	192.168.1.28

```
GROUP 2 : 192.168.1.28 00:0C:29:60:4D:E6
Starting Unified sniffing...

HTTP : 192.168.1.28:80 -> USER: test1  PASS: test1  INFO: http://
www.kianyong.com/
HTTP : 192.168.1.28:443 -> USER: test1  PASS: test1  INFO: https://
www.kianyong.com/
```

Fig.17: Man in the middle rogue program.

L. Client Authentication

To prevent man in the middle attack, this paper proposed client authentication to authenticating each user. When server request for client authentication, the client will sends the server a certificate and a separate piece of digitally signed data to authenticate it. Then the server use the digitally signed data to validate the public key in the certificate to authenticate the identity of the certificate claims to represent. In this case, web server request for authenticating but attacker does not have client certification, thus web server will refuse the connection from attacker. Fig.18 shows connection reset at client site when web server cancel the connection.



The connection was reset

The connection to the server was reset while the page was loading.

- The site could be temporarily unavailable or too busy. Try again in a few moments.
- If you are unable to load any pages, check your computer's network connection.
- If your computer or network is protected by a firewall or proxy, make sure that Firefox is permitted to access the Web.

Fig.18: Connection reset at client site.

7. Conclusion

This paper has included a detailed of secure video streaming performance between HTTP and HTTPS. The test-bed result shows that performance of HTTPS is only slightly slower that HTTP. A general discussion for best-known active attack such as man in the middle attack over HTTPS communication is discussed. We have used client authentication to prevent man in the middle attack over HTTPS communication which able to tailor X.800 security standard to provide data integrity, authentication, data confidentiality, and non-repudiation for secure video streaming.

References

[1] Postel, J. (ed.), "Internet Protocol - DARPA Internet Program Protocol Specification", RFC 791-RFC 793, USC/Information Sciences Institute, September 1981.

[2] Adame, A. & Kong, B. (2008, June). Performance Management an Analysis of an IPv6 Sensor on the Move Using Commercial Network Management Software. Master's Thesis, Naval Postgraduate School, Monterey, California.

[3] Cho, K., M. Luckie, and B. Huffaker, (2004). "Identifying IPv6 Network Problems in the Dual-Stack World". In Proceedings of the ACM SIGCOMM Workshop on Network Troubleshooting: Research, Theory and Operations Practice Meet Malfunctioning Reality, Portland, Oregon '04, pp 283 – 288.

[4] L. Colitti, G. D. Battista, and M. Patrignani, (2004) "IPv6-in-IPv4 tunnel discovery: methods and experimental results," IEEE Transactions on Network and Service Management, vol. 1, no.1.

[5] Csondes, T., S. Dibuz, and P. Kremer, (2000). "Experiments on IPv6 Testing", In Proceedings of the IFIP TC6/WG6.1 13th International Conference on Testing Communicating Systems: Tools and Techniques, pp 113 – 126.

[6] Steinberg, J., and Pasquale, J. (2007, June) "Improving Wireless Access of Video Over the Internet" In Proceedings of the Networking and Services, 2007. ICNS. Third International Conference '07, pp 88.

[7] George Lawton, (2005, July) "Video Streams into the Mainstream," Journal of ACM Digital Library Computing Machinery, vol. 33, no. 7, pp. 12-17.

[8] Hagen, S. (2006). IPv6 Essentials, 2nd ed. Sebastopol:O'Reilly.

[9] How SSL work: the ssl handshark. Retrieved 12 September 2010 from http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/index.jsp?topic=%2Fcom.ibm.itame2.doc_5.1%2Fss7aumst18.htm.

[10] Streaming Methods: Web Server vs. Streaming Media Server. Retrieved 14 September 2010 from http://www.microsoft.com/windows/windows media/compare/webservvstreamserv.aspx.

[11] SSL/TLS Strong Encryption: An Introduction Retrieved 24 September 2010 from http://httpd.apache.org/docs/2.0/ssl/ssl_intro.html.

[12] Time Sequence Graph. Retrieved 24 September 2010 from http://ait.web.psi.ch/services/linux/hpc/hpc_user_cookbook/tools/network/tcp/tcptrace/manual/node12_ct.html.

[13] Difference Between http & https. Retrieved 13 August 2010 from http://www.hoax-slayer.com/difference-http-https.shtml.

[14] User Datagram Protocol. Retrieved 13 August 2010 from http://www.cinqueterreliguria.net/download/streaming%20media/User%20Datagram%20Protocol.htm.

[15] International Telecommunication Union (ITU), Security Architecture for Open Systems Interconnection for CCIT Applications, Recommendation ITU-T X.800, ITU, Geneva, 1991.

[16] Ted Shorter (2005), "A "Real-Life" Man-in-the-Miiddlle Attttacckk on SSL", RSA conference 2005.

[17] Ted Shorter (2005) , "Preventing Man in the Middle Phishing Attacks with Multi-Factor Authentication", RSA conference 2005.

[18] Heather Yu, "Multimedia Encryption - Streaming Video Encryption, Preserve real time playback and decrease cost via partial encryption" Retrieved 12 September 2010 from http://encyclopedia.jrank.org/articles /pages/6816/Multimedia-Encryption.html.

[19] Ifstat. Iftstat version 1.1. http://gael.roualland.free.fr/ifstat

[20] Dapeng, W., Y. T. Hou, W. Zhu, Y. Zhang, and J. Peha (2001). "Streaming Video over the Internet: Approaches and Directions", IEEE Transactions on Circuits and Systems for Video Technology, vol. 11, No. 3, pp 282- 300.