Design and Implementation of Cloud-based Application for Cloud Provider System with SSL Accelerator and Content Redirection

Boon Keong Seah
Advanced Computing Lab, MIMOS
Technology Park Malaysia, Malaysia
bk.seah@mimos.my

Abstract—**The requirement to handle large SSL connection for secure login access to applications residing in the virtual machine hosted in the cloud provider system has given rise to the need to restructure applications in the cloud utilizing the hardware load balancer or Secure Socket Layer (SSL) Accelerator. In this paper, we share our experience in designing and implementing a system in enabling SSL and non-SSL content redirection for cloud-based application. We describe the configuration and optimization rules for handling SSL transactions in the SSL Accelerator. In addition, we covered requirement by applications for sharing the HTTP session to enable seamless application access without prompting of re-login. Hence, a Single Sign-On capability in applications can be achieved.**

Keywords-SSL; security in cloud; SSO; load balancer; SSL Accelerator.

I. INTRODUCTION

With increasing demands for applications to be deployed in cloud system, there is urgent need to ensure that similar security features offered in non-cloud environment be available in the cloud system as well. SSL [1][2] has been the common protocol used to provide secure communications to the web server be it in physical servers or in the virtual environment. Nevertheless, SSL also requires high CPU computation during the SSL handshake [3][6] which will impact the performance of the system hosting it. Hence, a combination of SSL and web server will consume the server resources significantly [15]. In order to distribute the SSL processing, there are two approaches. The first approach is through utilizing a load balancer and distributes the SSL process to multiple web server with SSL enabled. The load balancing of SSL can utilize several different schemes such as round robin, SSL with session, and SSL with backend forwarding [16][17]. Another approach is through utilizing the SSL Accelerator [5] as SSL reverse proxies where the SSL handshake process is offloaded [9] to the SSL Accelerator. Nevertheless, the above two approaches enforce a HTTPS connection to the client browser including accessing to the web contents.

In our implementation approach, we have designed and developed a system for enabling applications deployed in the cloud system, where only user authentication or login will require a secure channel. A session ticket will be created in the authentication system which will redirect the user to the personalized content of the applications. Each application hosted in the non-SSL channel will only be allowed access with a valid session ticket created earlier in the user authentication system. A Single Sign-On (SSO) [4] amongst applications can be realized utilizing the session ticket and hence provide a single user authentication experience.

We detailed in Section II the motivation for implementing the HTTPS and HTTP content redirection. Section III gives an overview of the system design and the operational scenario. Section IV presents the steps for the configuration and optimization rules for handling the SSL transactions in the SSL Accelerator. The implemented system is then tested and the result of the performance is shown in Section V. Section VI presents the conclusion of this paper.

II. APPLICATION OF HTTPS AND HTTP CONTENT REDIRECTION

The system can be applied to applications hosted in the cloud where the performance and SSO of accessing the contents are of great importance. Applications such as personalized knowledge management system, media streaming, news journal content subscription and others can utilize this system to deliver contents with lower latency [15] as the contents are not accessed in the SSL protocol. In addition, the contents are protected with user authentication access rights.

In this paper, we have shown the benefits of this implementation approach in improving the performance of user access. Our implementation approach of securing the user authentication part with HTTPS whilst leaving the web content in HTTP in the cloud system is that it will enable better performance as shown in Figure 5 as compared to implementing HTTPS connection to all web contents including the user authentication as shown in Figure 4.

Major websites such as Yahoo! Mail and Facebook also have implemented this approach where only the user authentication are protected. Nevertheless the detail of such implementation approach was not published. To support high performance SSL for all contents, it involves cost and resources [15].

III. System Design

A. System Overview

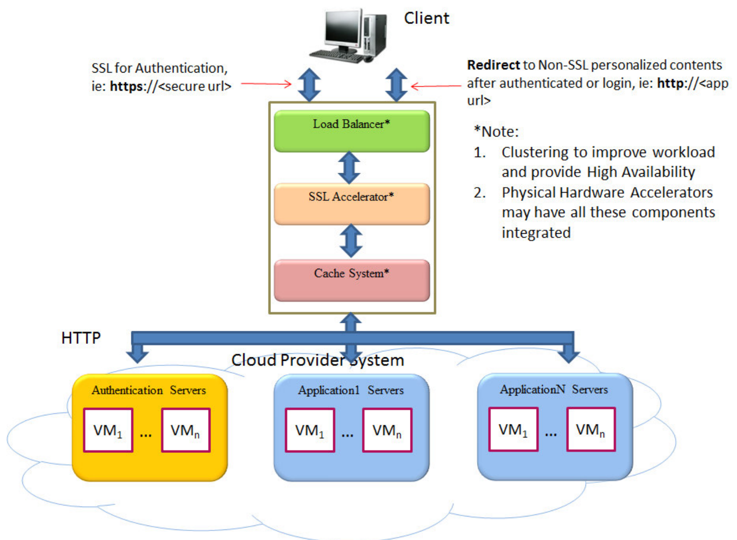

Client
SSL for Authentication, ie: https://<secure url>
Redirect to Non-SSL personalized contents after authenticated or login, ie: http://<app url>
Load Balancer*
SSL Accelerator*
Cache System*
*Note:
1. Clustering to improve workload and provide High Availability
2. Physical Hardware Accelerators may have all these components integrated
HTTP
Cloud Provider System
Authentication Servers
Application1 Servers
ApplicationN Servers
VM1 ... VMn

Figure 1. Overview of system design

Figure 1 shows the overview of the system design. In this design, we have the physical SSL Accelerator appliance [7][8] with integrated features such as load balancer, SSL hardware offloading and cache system acting as the **SSL reverse proxy**. The SSL Accelerator manages the HTTPS communications from clients and communicates in HTTP protocol to web servers hosted in the virtual machines.

In our design, the SSL Accelerator uses a combination of round robin load balancing and session persistent of client IP address to distribute the authentication and application Virtual Machine (VM) Servers load. The SSL Accelerator determines if the client IP address is previously connected; if it is, it returns the client to the same VM Server.

In our design, the architecture also takes into consideration the distribution of HTTPS amongst the SSL Accelerator. The SSL Accelerator [7][8] has the capability to scale with HTTPS load via workload distribution.

B. Operational Scenario

The system, illustrated in Figure 1, works as follows:

1) HTTPS requests from client to the personalized content are intercepted by the SSL Accelerator.

2) The SSL Accelerator establishes an SSL connection to the client.

3) The SSL Accelerator checks the client's IP address is previously connected and has the session information. If the client's IP is previously connected, SSL Accelerator establishes a HTTP connection to the same VM server.

4) The SSL Accelerator checks whether the request for static contents is cached. If the content is cached, it will be serviced by the SSL Accelerator. The details of the cached configuration are given in Section IV.

5) The SSL Accelerator will assess further optimization rules configuration such as disabling HTTP TRACE and disabling accepting weak SSL cipher connection from client. The details of the optimization rules configuration are given in Section IV.

6) Upon completion of the checking by the SSL Accelerator, the HTTPS request will be forwarded as HTTP request to the pool of back-end VM applications. The back-end forwarding methods will utilize the round robin load balancing amongst the pool of VM Servers configured.

7) VM applications checks whether there is an authenticated session.

   a) If authenticated session does not exist, then it redirects to the Authentication Server for user authentication; otherwise, it permits the client access to the application without having to re-login.

IV. Implementation Approach

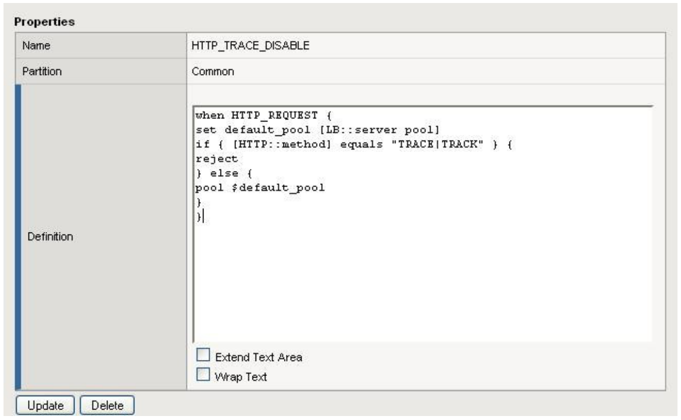
The SSL Accelerator we use to implement this system is **BIG-IP LTM F5** which is a combination of load balancer, SSL hardware offloading system and cache system. In addition, the LTM F5 has in-built IDS (Intrusion Detection System) [13] to prevent network attack such as DDoS (Distributed Denial-of-Service) and IP Spoofing. The SSL Accelerator can be appliance-based, SSL Accelerator PCI card [10] based or a combinations of SSL reverse proxy applications such as POUND [11].

The authentication and application Virtual Machine (VM) Servers are deployed in MIMOS cloud. Each of the VM is allocated with 6 Virtual CPU, 16 GB of Virtual Memory, and 20 GB of Hard Disk. The VM guest OS is based on Centos 5.4 64 bit, deployed in the KVM platform [14].

As discussed earlier in Section III, in order to further improve on security and performance, the following optimization rules can be deployed in the SSL Accelerator:

1) Disable HTTP TRACE rule

   a) The HTTP TRACE request includes all HTTP headers and cookies credentials available to the web browsers. This will enable cross-site security vulnerability. In F5, the HTTP TRACE can be disabled as shown in Figure 2.



Name	HTTP_TRACE_DISABLE
Partition	Common
Definition	(see below)

```
when HTTP_REQUEST {
set default_pool [LB::server pool]
if { [HTTP::method] equals "TRACE|TRACK" } {
reject
} else {
pool $default_pool
}
}
```

Figure 2. Disabling of HTTP TRACE

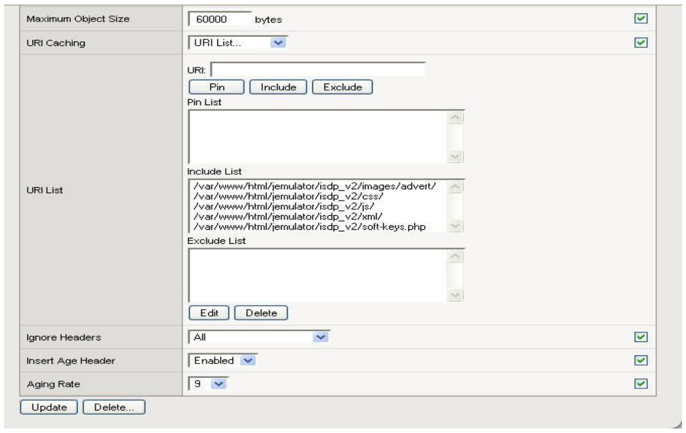
2) Disable Connection for Weak SSL Cipher

a) In the system, we also disable weak SSL cipher in the F5 for older browser as it may have SSL vulnerabilities [12]. The rule shown below will disable weak SSL Cipher connection:

- ALL:!ADH:!LOW:!EXP:!SSLv2:!NULL:HIGH:MEDIUM:RSA:RC4:

3) Cache Optimization

a) The cache optimization enables static contents caching at the SSL Accelerator which will improve the performance of the VM applications. Figure 3 illustrates the F5 configuration:



Maximum Object Size	60000 bytes
URI Caching	URI List...
URI List	Include List: /var/www/html/jemulator/isdp_v2/images/advert/ /var/www/html/jemulator/isdp_v2/css/ /var/www/html/jemulator/isdp_v2/js/ /var/www/html/jemulator/isdp_v2/xml/ /var/www/html/jemulator/isdp_v2/soft-keys.php
Ignore Headers	All
Insert Age Header	Enabled
Aging Rate	9

Figure 3. Enabling static content cache

V. PERFORMANCE

In the evaluation of the performance of the system, we have deployed six web servers running applications using the Zend Framework. For the Authentication service, we also have deployed six Apache web servers serving user authentication. We prepared the system and measured the user response time for 3000 concurrent user connections with and without both the SSL Accelerator and content switching. We use JMeter as the test benchmark tool. Figures 4 and 5, shows the respective results:

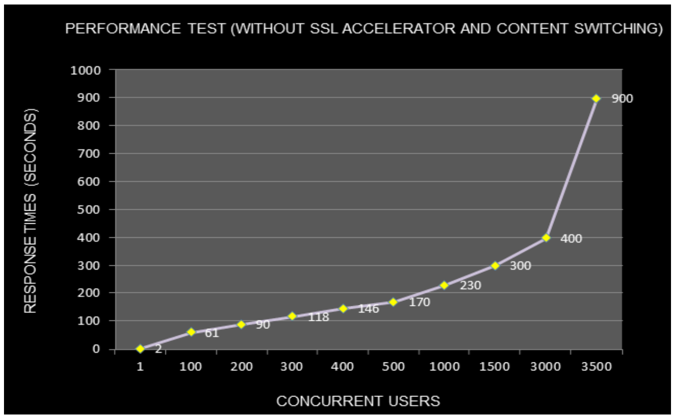


Figure 4. Concurrent user connections to the system with SSL terminating in each of the respective Apache web server.

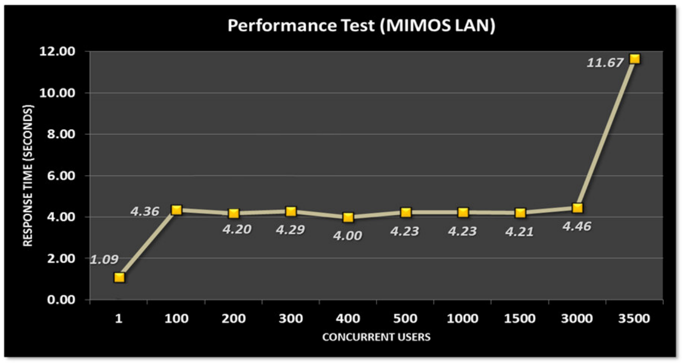


Figure 5. Concurrent user connections to the system with user authentication service using the six Apache web servers and SSL Accelerator

From Figures 4 and 5, the system we implemented shows a significant improvement of user response time. Nevertheless, in Figure 5, we noticed that the performance shows a relatively stable response time for the user connections, but degraded significantly when the concurrent user connection reaches 3000. One plausible explanation for this is that a large number of TCP connection requests are queued in the Linux network kernel buffer. This large queue will slows the network packet processing as the kernel needs to maintain the TCP connection states. In addition to the TCP connection queue size, each of the six Apache servers we used for this system implementation has a configuration of 500 maximum concurrent user connections. We did not pursue further research into modification and testing of the system implemented due to time and resource constraint.

VI. CONCLUSION AND FUTURE WORK

In this paper, we have successfully designed and implemented a content redirection utilizing SSL Accelerator as the SSL reverse proxy for virtual applications deployed in the cloud provider system. This approach enables the applications to offload the SSL process to the SSL Accelerator. In this system, the approach we have taken is to protect only the user authentication in HTTPS connection, while the personalized content is redirected to HTTP connection for successful authenticated user. In terms of performance, we have managed up to 3,000 concurrent SSL connections in less than 5 seconds. As part of future work, we intend to develop an SSL appliance to cater for SSL offloading; it will also be able to integrate to the cloud provider system directly for SSL subscription purposes. For user authentication protocol, federated SSO protocol such as SAML 2.0 [18] and OpenID [19] are currently being evaluated.

REFERENCES

[1] A. O. Frier, P. Kocher, and P. C. Kaltorn, The SSL Protocol Version 3.0 draft, March 1996.

[2] J. Viega, P. Chandra, and M. Messier, Network Security with OpenSSL, 1st ed., Oreilly Publications, 2002.

[3] R. Hatsugai, T. Saito, “Load-balancing SSL Cluster Using Session Migration”, 21st International Conference on Advanced Networking and Applications (AINA’07), Niagara Falls, USA, IEEE Press, Dec 2007, pp. 62-67.

[4] N. Chamberlin, Brief Overview of Single Sign-On Technology, Government Information Technology, 2000.

[5] S. Abbot, “On the Performance of SSL and an Evolution to Crytographic Coprocessors,” Proc. RSA Conf., San Francisco, USA, Jan 1997.

[6] G. Apostolopoulos, V. Peris, and D. Saha, “Transport Layer Security: How much does it really cost?”, IEEE INFOCOMM 1999, New York, USA, June 1999, pp. 717-725.

[7] http://www.f5.com/products/big-ip/ [retrieved: February, 2012]

[8] http://www.barracudanetworks.com/ns/products/web-site-firewall-overview.php [retrieved: February, 2012]

[9] R. Mraz, “Secure Blue: An Architecture for a Scalable, Reliable High Volume SSL Internet Server”, Proc. ACSAC 2001, Australia, Dec 2001, pp. 391-398.

[10] http://www.safenet-inc.com/products/data-protection/hardware-security-modules-hsms/ [retrieved: July, 2012]

[11] http://www.apsis.ch/pound/ [retrieved: February, 2012]

[12] A. Klein, "Attacks on the RC4 stream cipher", Designs, Codes and Cryptography, vol. 48, Springer-Verlag, 2008, pp. 269-286.

[13] http://www.f5.com/pdf/white-papers/securing-enterprise-wp.pdf [retrieved: January, 2012]

[14] A. J. Younge, R. Henschel, J. T. Brown, G. Laszewski, J. Qiu, and G. C. Fox, “Analysis of virtualization technologies for high performance computing environments”, IEEE CLOUD 2011, Washington, USA, July 2011, pp. 9-16.

[15] C. Coarfa, P. Druschel, and D. S. Wallach, "Performance analysis of TLS Web servers", ACM Trans. Comput. Syst., 2006, pp. 39-69.

[16] V. M. Suresh, D. Karthikeswaran, V. M. Sudha, and D. M. Chandraseker, “Web server load balancing using SSL back-end forwarding method”, IEEE ICAESM 2012, Tamil Nadu, India, March 2012, pp. 822-827.

[17] J. H. Kim, G. S. Choi, and R. D. Chita, "An SSL Back-End Forwarding Scheme in Cluster-Based Web Servers", IEEE Trans. Parallel Distrib. Syst., 2007, pp. 946-957.

[18] W. Baozhu, X. Bing, and S. Lianghong, “Design of web service single sign-on based on ticket and assertion”, IEEE AIMSEC 2011, Deng Feng, China , 2011, pp. 297-300.

[19] P. Urien, “OpenID Provider based on SSL Smart Cards”, Proc. CCNC 2010, Las Vegas, USA, 2010, pp. 1-2.