Volume 5, Issue 7, July 2015 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

Web Application Security Model at Intranet Level Using HTTP

Dr. R. Seshadri
Prof. & Director of University Computer Centre
S.V.University, Tirupati, A.P., India

S. Venkateswarlu
Research Scholar
S.V.University, Tirupati, A.P., India

Abstract-- The online applications on web always face threats in various ways like injection attacks, cross site scripting, phishing etc. which may due to weakness at client side or at transport level or at server side intranet level. Even though some of the attacks have been controlled by applying stringent measures like using SSL or change of protocol, the problem arises at some other point in online application which is complex to configure, since existing software should be linked with the present like the customization of web applications at JVM level which by default HTTP configured. To overcome this at Intranet level at the point of Firewall/DMZ level HTTP protocol should be used at the same time it should be protected from attacks by insiders at Intranet level. So in this paper we have proposed alternative web application security design/model by using cryptographic primitives at intranet level

Keywords: Web application security, HTTPS, HTTP, JVM Configuration, Firewall-DMZ, Cryptographic primitives.

I. INTRODUCTION

The web, is a key in providing a platform for commerce, entertainment, and social interaction, is a complex delivery platform for sophisticated distributed applications with manifold security requirements. However, today's web browsers, servers, network protocols, browser extensions, and their security mechanisms were designed without analytical foundations for their future usage with additional browser features, protocols, and standards [1]–[6]. The provisions of new features to web includes complex, unclear threat models, which are unstated and unverified assumptions. As a result, new features will increase vulnerabilities and break security invariants assumed by web applications [7]–[9]. In Web applications, Security is a quality attribute that plays an important role in measuring quality and competency with other applications. The security threats include access control violations, integrity violations, sabotage, fraud, privacy violations, as well as denial of service and infrastructure attacks. All of these threats collectively have come to be known as cyberwar or cyber terrorism [18].

Internet users all over the world are using web-based systems to manage important data for them such as bank account and healthcare information. Users assume that these systems are securely designed but many web applications have severe security flaws that allow simple attacks to succeed. Many enterprises that handle private or confidential data such as credit card and debit card numbers, social security numbers and health care records have the data stored in plain text in multiple locations throughout the enterprise. Typically an enterprise will have a large number of applications that process private or sensitive data that must be adapted to handle encryption [14].

There will be applications that handle 'in-flight' data, sending and receiving data from external trading partners or other entities with a single company, and applications that handle 'static' data, or data at rest. The applications are on a variety of computers with a number of operating systems, languages and databases. The computers are on multiple networks or subnets.

The encryption and decryption of sensitive data distributed throughout the enterprise requires a large number of resources – keys and certificates – that must be managed across applications, computers and networks in a cost-effective and efficient way that does not compromise security. Additionally, user and application access to these resources must be controlled, managed and audited so that authorized access is quick and reliable, while malicious attacks are thwarted. For security HTTPS is employed but it leads to many deployment and maintenance problems.

The web security model consists of a selection of transfer protocols, web concepts, precise threat models, and two broadly applicable security goals. These design choices are informed by previous experience designing and (informally) evaluating web security mechanisms, such as preventing cross-site request forgery [10], securing browser frame communication [11], preventing DNS rebinding [12], and protecting high security web sites from network attacks [13].

The central web concepts we formalize in our model include browsers, servers, scripts, HTTP, and DNS, as well as ways they interact. For example, each script context, representing execution of JavaScript within a browser execution environment, is associated with a given "origin" and located in a browser. By making use of browser APIs, such as XMLHttpRequest, these script contexts can direct (restricted forms of) HTTP requests to various DNS names, which resolve to servers. These servers, in turn, respond to these requests and influence the browser's behavior. Browsers behavior leads to security threats like access control violations, integrity violations, sabotage, fraud, privacy violations, as well as denial of service and infrastructure attacks, so organizations uses HTTPS to ensure security at transit level. Due to this customization of application at deployment level is complex to achieve with standard protocol, so in this paper we discuss about an alternate web security model at intranet level which will be supportive at deployment/maintenance tasks.

II. BACKGROUND

Most of the Web Applications opened up for public access involve a lot of components when deployed and configured in a data center. Some data centers use a common security configuration and others use specific to application and/or product. Some of the common components include the following:

- Firewall
- DMZ
- Reverse Proxy
- Load Balancer
- Directory Server / Identity Management Server / Policy Server
- Web Server
- Application Server
- Database Server
- Utility servers such as File Server, Print Server, Cache Server etc.

All of the above are involved in both HTTP and HTTPS based setup. Various levels of security applied for web application includes:
a) Application Access: Achieved via Authentication - Application specific access management, LDAP, Active Directory – Federation Services, Site Minder etc.
b) Role / Department / Designation based Access: Achieved via Authorization to access various modules and information – by setting up roles, groups, designations etc.
c) Data access: This is either managed at the application level or on DB Server.

While HTTPS is the best way to implement Information Web Transport security, it also complicates the implementation and poses challenges to the applications – specifically when Reverse Proxy, Load Balancer and JVM level security are all involved. But at deployment level it is possible to develop a secure application level protocol at intranet level making use of cipher technology along with HTTP. The reasons we choose HTTP as the communication protocol are as follows [15]:

1) Different application level protocols have been developed for individual network services, such as FTP, ***SMTP, NNTP,*** or GOPHER. HTTP has the flexibility to be able to provide services similar to those which have been provided by these protocols.
2) Hypertext-based user-friendly graphical interface Using HTTP and the Hypertext Markup Language (HTML), distributed multimedia information systems with user-friendly graphical interfaces based on hypertext can be easily developed**.**
3) User agents and servers available on almost all platforms HTTP has now gained widespread popularity and various kinds of user agents and servers are available on almost all platforms. Even if new protocols for closed networks are developed which are superior in function or flexibility, new clients and servers have to be developed for compatibility, which is costly and an obstacle to their universal acceptance

As the Online Web applications are also used at Intranet level , networks inside the premises are usually protected using a dual home gateway and packet filter (firewall) and the Internet can only be accessed through proxies on the firewalls. So customizing an application according to business rules at firewall or at proxy level is a challenging task with HTTPS, so with help of cryptographic primitive along with HTTP we may overcome attacks like network tampering, replay attacks and middle of the man attacks at Intranet Level [16]. Problem also arises at configuration time of web application which plays a central role in the deployment and management of Web applications and infrastructures. Web applications and infrastructures are often susceptible to malicious attacks which should also be addressed. Configuration includes security configuration which will provide defense against attacks on web applications

III. PROPOSED MODEL

While HTTPS is the best way to implement Information Web Transport security, it also complicates the implementation and poses challenges to the applications – specifically when Reverse Proxy, Load Balancer and JVM level security are all involved. SSL / CA Certificates (with FQDNs and public access URLs) to be installed on each of these server components. New JVM security implementation is now posing challenges to the implementations. All of these challenges are specific to the data center and security configurations to be adjusted and at times to be compromised. In one of the scenarios, HTTPS had to be converted to HTTP at the DMZ level as JVM level HTTPS / handshake was giving a problem. Luckily the threat was minimal as the Reverse Proxy and the Web / Application Servers were cable connected in the same data center and the reason being anyone trying to tap the information transported over HTTP has to be physically present in the Data Center. Web Application deployment scenario is shown in the model is given in fig-1. The problem lies at Firewall DMZ with JVM or at proxy server or load server components require HTTP protocol, as compared with HTTPS since the Java virtual machine is configured to use a firewall proxy with the Java system properties: http.proxyHost and http.proxyPort [17]. After this configuration has been set, the URL will use the firewall proxy to contact hosts outside the firewall. so the conversion at Firewall is necessary .

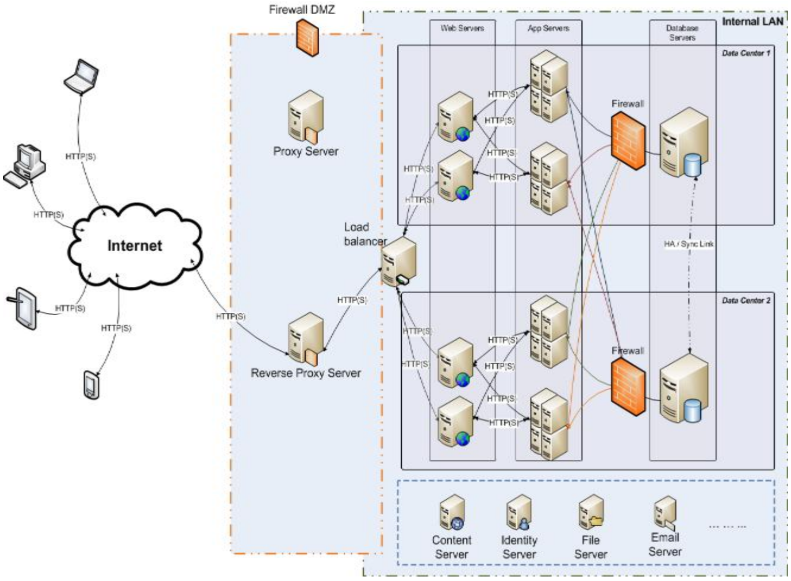

Fig-1 Web Application Deployment/architecture

At intranet level there is a possibility of tampering data if HTTP is configured and also data is prone to attacks as already discussed, so to overcome these issues a new application design equipped with cryptographic primitives. In this paper we proposing two design paradigms.

3.1 Design-1

In the proposed design-1 as shown in Figure -2,encryption is done at client side before sending data to server side where decryption is done at intranet level at data center which receives data through HTTP are essential.

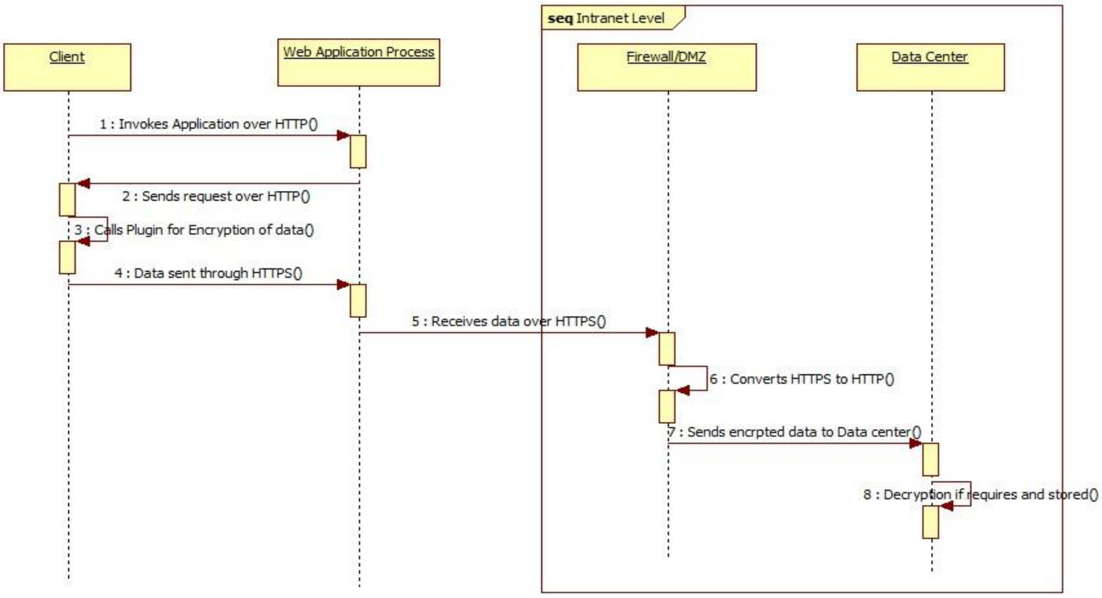
seq Intranet Level
Client
Web Application Process
Firewall/DMZ
Data Center
1 : Invokes Application over HTTP()
2 : Sends request over HTTP()
3 : Calls Plugin for Encryption of data()
4 : Data sent through HTTPS()
5 : Receives data over HTTPS()
6 : Converts HTTPS to HTTP()
7 : Sends encrpted data to Data center()
8 : Decryption if requires and stored()

Fig-2 : Application Design -1

1. Client invokes Web application URL over HTTP.
2. Sends request over HTTP to Web Application.
3. After response from Web application Client will send data.
4. Before sending data to Web application, a plugin is invoked and data is encrypted before transmitted.
5. Client uses HTTPS to transfer data to server.
6. At Server side or at corporate side intranet level, Firewall DMZ deployed at the premises will convert HTTPS to HTTP.

At Intranet Level since data is encrypted, it cannot be tampered and it is available over HTTP which is required by deployment at JVM level or at data center.

3.2 Design-2

In design-2 cryptographic primitives are used at intranet level after conversion of HTTPS to HTTP at firewall as depicted in figure-3.

1. Client sends request to Web application using HTTP.
2. After receiving response from Server over HTTP
3. Client uses transport layer security protocol HTTPS to send data securely over internet to server.
4. At Intranet level, server side or at company Firewall-DMZ.
5. Before entering into local LAN network at intranet level, data is encrypted by using secure cryptographic algorithm.
6. Data between nodes are exchanged between servers at intranet level or at deployment site over HTTP protocol.
7. At data centre level or at JVM level data or request is decrypted which reduces the problems at JVM deployment level

which requires redirection of request from another port which is not currently intended for application.

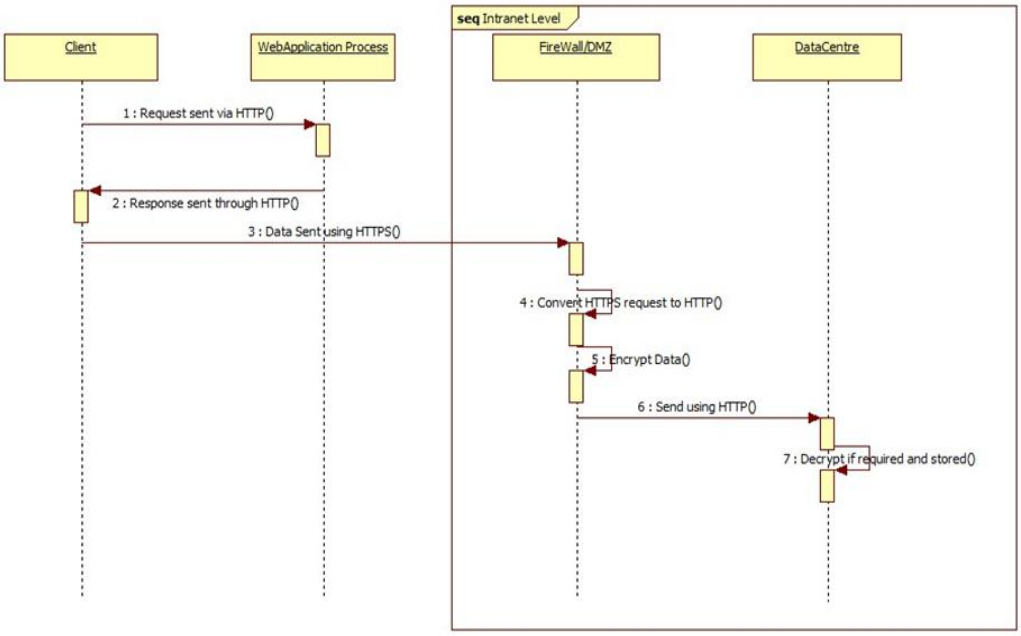
seq Intranet Level
Client
WebApplication Process
FireWall/DMZ
DataCentre
1 : Request sent via HTTP()
2 : Response sent through HTTP()
3 : Data Sent using HTTPS()
4 : Convert HTTPS request to HTTP()
5 : Encrypt Data()
6 : Send using HTTP()
7 : Decrypt if required and stored()

Fig-3 : Application Design -2

IV. SECURITY ANALYSIS

In Design-1 since data at the client side is encrypted before transmitted using HTTPS, data is secured when it passed through all the networks as well as at the intranet level. In the design-1 data is secured at transit as well as at storage at data centre if it is stored as it is received and it can be configured at JVM level with HTTP protocol conversion at intranet level without loss in security scale. In Design-2 data is encrypted after transmission through HTTPS at firewall-DMZ point, and send through HTTP and our intended purpose is served as it is secured at Intranet Level if it is attacked by insiders or any resident programs at various points at Local network and can be configured JVM level or at load balancer level before processed at data centre.

V. CONCLUSION

In this paper we have discussed about complexities arise in web applications during deployment and configuration when used with HTTPS protocol and also security implications if HTTP is used in data transfer to data centers. So we have addressed the problem at client level as well as at Firewall-DMZ at deployment side by proposing two designs by using cryptographic algorithm which we specify the algorithm in our future works.

REFERENCES

[1] E. Hammer-Lahav, "Oauth core 1.0 revision a," 2009.[Online]. Available: http:// oauth.net/ core/1.0a

[2] J. Hodges, C. Jackson, and A. Barth, "Strict transport security," 2009. [Online]. Available:http://lists.w3.org/Archives/Public/www-archive/2009Sep/att-0051/draft-hodges-strict-transport-sec- 05.plain.html

[3] A. van Kesteren, "Cross-origin resource sharing," 2009.[Online]. Available: http://www.w3.org/TR/cors/

[4] S. Stamm, "Content security policy," 2009. [Online].Available:https://wiki.mozilla.org /Security /CSP/Spec

[5] Microsoft Inc., "Xdomainrequest object," 2009. [Online]. Available:http://msdn.microsoft. com/en-s/library/cc288060%28VS.85%29.aspx

[6] A. Inc., "Cross-domain policy file specification," 2008.[Online].Available: http://www.adobe.com/devnet/articles/crossdomain policy file spec.html

[7] E. Hammer-Lahav, "Acknowledgement of the oauth security issue," 2009. [Online]. Available: http://blog.oauth.net/2009/ 04/22/acknowledgement-of-the-oauth-security-issue/

[8] T. Klose, "Confused deputy attack on cors,"2009.[Online].Available:http:/ lists.w3.org /Archives/Public/public-apps/2009AprJun /1324.html

[9] E. Nava and D. Lindsay, "Abusing internet explorer 8'sXSS filters," in BlackHat Europe, 2010. [Online]. Available:http:// p42.us/ie8xss/AbusingIE8s XSS Filters.pdf

[10] A. Barth, C. Jackson, and J. C. Mitchell, "Robust defenses for cross-site request forgery," in In Proc. of the 15th ACM Conf.on Computer and communications Security (CCS 2008).ACM, 2008, pp. 75–88.

[11] A. Barth, C. Jackson, and J. Mitchell, "Securing frame communication in browsers," *Commun. ACM*, vol. 52, no. 6,pp. 83–91, 2009.

[12] C. Jackson, A. Barth, A. Bortz, W. Shao, and D. Boneh, "Protecting browsers from dns rebinding attacks," *ACM Trans.Web*, vol. 3, no. 1, pp. 1–26, 2009.

[13] C. Jackson and A. Barth, "Forcehttps: protecting high-security web sites from network attacks," in *WWW '08: Proceeding of the 17th international conference on World Wide Web*. New York, NY, USA: ACM, 2008, pp. 525–534.

[14] nuBridges White paper, *Best Practices in Data Protection: Encryption, Key Management and Tokenization*

[15] Takahiro Kiuchi, Shigekoto Kaihara "*C-HTTP -- The Development of a Secure, Closed HTTP-based Network on the Internet*", Proceedings of SNDSS '96,IEEE

[16] Rescorla E, Schiffman A. "*The Secure Hypertext Transfer Protocol. Internet Draft, 1995* (Work in progress, available on the World Wide Web as ftp://ds.intemic.net /internet-drafts/draft-ietf-wts-shttpoo-txt)

[17] http://erights.org/elib/ distrib/vattp /DataComm Thru.htm [Available]

[18] Bhavani, chris Clifton et.al, "*Directions for Web and E-Commerce Applications Security*", Tenth IEEE International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises. WET ICE 2001

ABOUT AUTHOR

Dr. R. Seshadri working as Professor & Director, University Computer Centre, Sri Venkateswara University, Tirupati. He was completed his PhD in S.V.University in 1998 in the field of " Simulation Modeling & Compression of E.C.G. Data Signals (Data compression Techniques) Electronics & Communication Egg.". He has richest of knowledge in Research field, he is guiding 10 PhD in Fulltime as well as Part time. He has vast experience in teaching of 26 years. He published 10 national and international conferences and 8 papers published different Journals.

S. Venkateswarlu Research Scholar in SV University, Tirupati and Working as Programmer in the Computer Centre, S.V.University, Tirupati. He has 16 years of Computer programming experience.