Network Wide Protection for Cookies

Josna Joseph
Department of Computer science .
Rajagiri School of Engineering & Technology
Kochi, India
joannjos@gmail.com

Vinodh P Vijayan
Department of Computer science.
Rajagiri School of Engineering & Technology
Kochi, India
vinodhpv@rajagiritech.ac.in

***Abstract*— World Wide Web works via its underlying hypertext transport protocol. HTTP is a stateless transport protocol i.e. each time when a client gives a request to the server it will be new to the server. HTTP will not support continuity for client-server interaction. To get the continuity of the session the HTTP make use of cookies. Cookies are used for personalizing the user to the website. Cookies are carried in the HTTP headers during browser-server interaction. Cookie values are carried as plan text. Which reflect the need for the protection of cookies in the web. Even though so many methods are proposed to solve security issue of the cookies, no method completely removes the issue. In this paper first discuss about the various security issues and need for the security of cookies. Then a method is proposed to give a network wide protection for the cookies using encryption method.**

Keywords- Encryption, public-private key, Cookies, HTTP, confidentiality, integrity.

I. INTRODUCTION

Cookies are small piece of data passed between the browser and server to identify a particular user to web site. They first appeared in the Netscape navigator browser released in September of 1994. Cookies carries unique id of the user to the site or any other data that is unique to identify a user in the system. Cookies are also used to remember the username –password to a site (there will not be any need to keep on entering the username and password each time we login to a site), authentication of a client etc.

First time when a client give request to the server the cookie filed will be empty. The server will set the cookies inside the Set-Cookie filed inside the HTTP response header. The size of the cookie field is not more than 4kB. The key-value pair inside the cookie is the actual part of the cookies that carry information of users. Other than this field the cookie contains information like expiry date, domain, path, secure etc for managing the cookies inside the network. The HTTP request is shown in the Fig. 1 below.

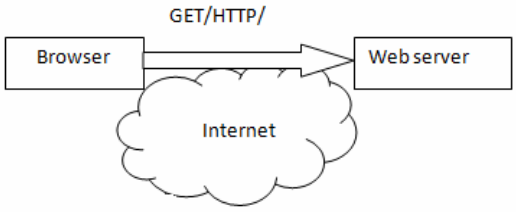

Figure 1. First http request from a client to server.

A Set-cookie filed and an example is shown below:

Setcookie: name, value, expiry date, path, domain,secure.
Eg:setcookie('UserName', 'Bob', 12-12-2010, '/', '.example', false);
Name : hold the name of the cookie.
Value: hold the actual data part of the cookie used by the server.
Expiry date: hold the expire date of the cookies(if this part is empty then the browser will delete the cookie at the end of the current browser-server session). It hold both date and time part. This field is also used to delete the cookies by setting the date/time in the past. Automatic deletion of the cookies are done when it expires.
Domain:specify to which server domain the data should be sent.
Path: restrict the passing of cookies within the domain.
Secure: specify whether the cookies should be sent through a secure path or not(if set as true then should sent the cookies only through a secure channel).
When the HTTP response reaches the web browser, it will fetch the cookies part and store it in the hard disk as a text file(eg: If you have Windows 7 or Windows Vista then the cookie folders are in these locations
C:\Users\<*username*>\AppData\Roaming\Microsoft\Windows\Cookies\
or
C:\Users\<*username*>\AppData\Roaming\Microsoft\Windows\Cookies\Low\).
The HTTP response is shown in the Fig. 2 below.

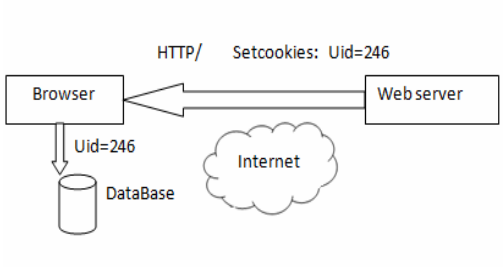

Figure 2. HTTP response with Setcookies filed.

Later when the browser give request to the same server, it check whether is there exist ant cookie belongs to the domain. If exist the it will be forwarded to the server by using the field Cookies inside the HTTP request header. There is a chance that more than one cookie exist for a particular domain. In such case the name-value pairs are forwarded together by putting a delimiter. Web browsers must support 300 cookies in total,20 cookies per domain and 4096 bytes per cookie. The resenting of cookies using the cookie field is given below in Fig. 3 below.

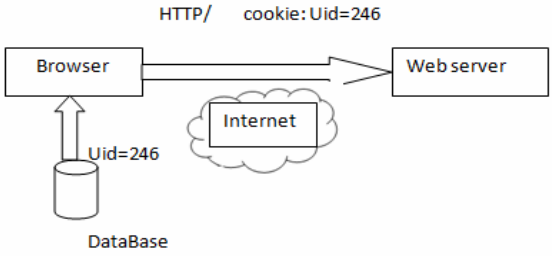

Figure 3. Resenting cookies to the server.

The browsers mainly support two types of cookies persistent and session cookies. The cookies that persist the session and retain inside the hard disk of client are called persistent cookies. As per the evaluation conducted in the paper [2].About 18.9% site using cookies used persistent cookies. Session cookies are those cookies which will get deleted just after the session. About 27.3% of sites use these cookies. 53.8% site used both persistent and session cookies combined. The graph on the usage of persistent and session cookies are shown below in Fig. 4.

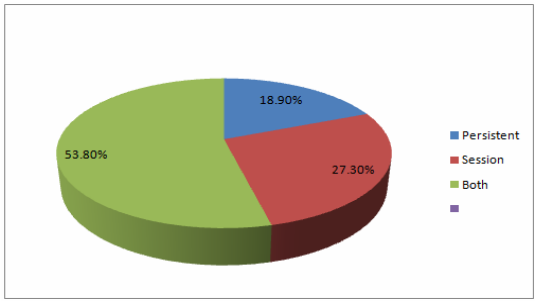

Figure 4. Usage of cookies in sites which use cookies

First party cookies: They are cookies from the site that current page belongs to.

Third party cookies: They are cookies originated at the site other than the one in which the current page belongs. For example of we visit the site www.domain1.com, the web pages on that domain may contain many feature content from another domain (eg: www.domain2.com). It can be an advertisement run by that third-party domain. This allows the third-party to set cookies on browser behalf of that. As per the analysis in paper [2] the usage of first and third party cookies are shown below in the Fig. 5.

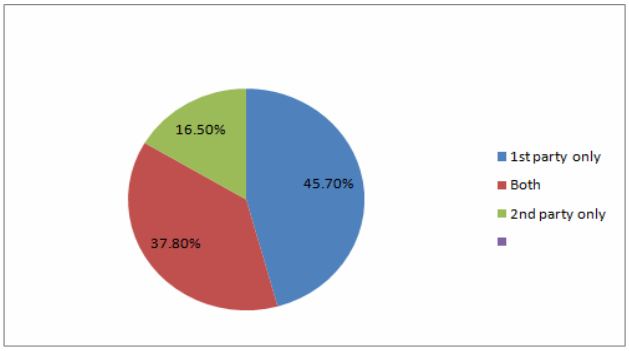

Figure 5. Usage of first and third party cookies.

II. NEED FOR SECURITY OF COOKIES

The need for the security of cookies arises because the actual users of the cookies are less aware of the existence of the cookies and what data they carry. The user also doesn't know how the values inside the cookies can be used to track them. Even though simply examining the cookie value won't reveal the actual used or what value they represented. The cookies tracked by the intruder can be resent to the server to get the information belongs to a user. Every browser provides provision for deleting the unwanted cookies which are less used by the user.

The name –value pair inside the cookie is the most sensitive information that a cookies carries. This pair is passed through the network as plain text. If an intruder is able to get thins name-value pair and able to resent it to the domain in which the cookies belongs to the intruder can access the information that the server provide to the original user, which always contain more sensitive data like username and password, user financial information etc.

The filed like domain, path etc can be used by anyone to do traffic analysis. The fields will reveal the information like in which server the users usually visit, what kind of activity the user involves. Thus the fields itself inside the cookies make it prone to traffic analysis and also confidentially of data passed between the client and server. Merchant web server cookies may hold information like user's identity and credit card number. It will make the user to use the site more friendlily but increases the risk.

Other need for security is to prevent the flow of the cookies to the server for which it's not intended for. This can be done by resetting the domain and path field of the cookies. When cookies are transmitted in plaintext it will be easy for

an attacker to do that. Among the two types of cookies the persistent cookies need more security in the end-system. Since it also persist after user session and get stored in the end system for a period of time. The power of the session cookies are they are always used to carry the identity of the user to pull out data stored in the user. Any manipulation of this ID can make the denial-of-service to the user.

The main security issue of the cookies is due to the reason that, they are sent as plain text. An intruder can easily read the value of the cookies. Also the cookies are stores as text file inside the hard disk by the browser any user having an access to the user compute can get the cookies file which when resent from their computer they will get an access to the user sensitive data.

Cross site scripting attack (XXS) attack the cookies by using malicious code. It directly attacks on the web browser database. When malicious code is executed the cookies get fetched and sent to the attacker. Which can use this cookie to impersonate the user and access data in behalf of the user. Another threat to which cookies are prone to are cookies harvesting threat. An attacker can collect all the cookies intended to a user by impersonating. Later when these cookies are sent to the domain to which the cookies belong to, to get access to data that belongs to the user. SSL (secure socket layer) in HTTPS will protect the cookies in the network but won't protect it in the end system. Also SSL layers are open to men-in-middle attacks also SSL establishment is slow.

III. RELATED WORKS

Many mechanisms can be used to secure the cookies in the network. Developments of the more secure mechanism of cookies are still under construction. A FU's secure cookie protocol was proposed which cannot provide high-level confidentiality also it is open to reply attack and volume attack [3]. In [4] Park and Sandhu proposed method to secure cookies from three point of view authentication, integrity and confidentiality. They proposed three Authentication cookies address-based, password-based and digital-signature-based. And using these cookies according to the situation. The paper [5] aims at integrity and confidentiality of the cookies. It is deals with securing web-based application, which is based on HTTP-reverse proxy. In [6] the cookies are protected against Cross Site Scripting attack (XSS), one of popular attacks which is often used to steal the cookies from a browser's database.

IV. PROPOSED SOLUTION

The proposed solution deals with the security of cookies in web. The method will make the part of the cookies needed by the server only available to server. When a client first time come into contact with the server the client will sent its public key along with the request. When the server sent back the client the cookies using HTTP request the server will encrypt the name-value pair of the cookies using the public key of the server itself. Since this value is only the required by the server , this encryption will make the cookie value to be available only to server by decrypting it using its private key. Before sending the cookies through the client , the whole cookies ie along with the other fields in the cookies are encrypted by the client public key.

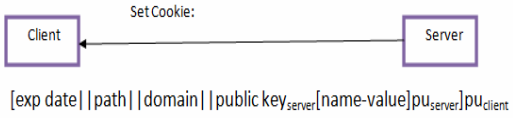

Figure 6. server to client setcookies field.

When the cookie reach the client , the client can understand the value of required field in the cookies by decrypting it, still the value is unintelligible to the client since it is encrypted using the servers public key. The fields in the cookies are kept as it is in the request which will make the cookies protected from the end- system attack.

When the client give an HTTP request to the same server the client will extract the value part from the cookies and sent it to the server along with the clients public key, which will help the server to understand the corresponding server's public-private key pair if the mechanism support different key pairs for different sessions. Using different key pair will enhance the security of cookie to a high degree.

Figure 7. Client to server cookies field.

When this HTTP request reach the server side, the server will find out the private key from its key ring and decrypt the value to get the original value which is used to retrieve client specific information in the server side.

Integrity of the cookies are handled using the message integrity code (MIC). MIC is calculated at the server side when it sent from the server. The MIC is generated using the private key of the server and the name-value pair. So that no other entity in the network can recalculate or can generate the cookie's MIC for the server. When the cookies are sent from the client to the server along with that client also sent this MIC. The server will recalculate the MIC when it get a cookie and matches with the one in the HTTP cookie filed. The matching proves that no entity had alerted the cookie value. Thus using the MIC we can ensure the integrity of the cookies. The passing of MIC between client and server is shown below.

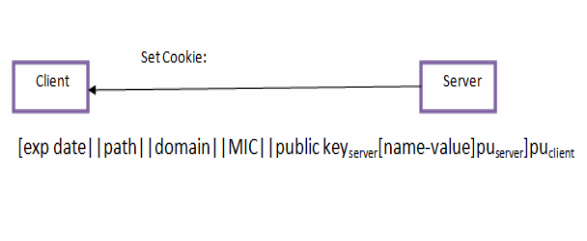

Figure 8: server to client setcookies with MIC.

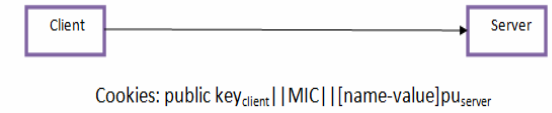

Figure 9: client to server cookies field with MIC.

Handling of private-public key pair can be done using the key ring concept used in the paper [7], which is protecting the key ring using MAC address. Instead of using MAC address other unique ids can also be used for encrypting the key ring.

V. CONCLUSION AND FUTURE WORK

This paper proposes a method to solve the confidentiality and security problem related to cookies. This technique will be useful for website that carries sensitive information about users. Also discussed about the basic cookies working, usages, various security issue related to cookies and related works. Hence the improved confidentiality and Integrity provides next level of Security in terms of cookies.

Since the different cookies used by different website requires different security levels. As a future work the decision making can be done using the secure field in the cookies. The value that can be set to this field for different degree of security is shown below. In the example given below if the secure field is 1 means the server provided both confidentiality and integrity for cookies so both key and MIC concept will be used.

TABLE I. Secure field value

Secure field value	Security level
1	Confidentiality and integrity
2	confidentiality
3	integrity
4	none

REFERENCES

[1] D. M. Kristol, " Http cookies: standards, privacy And politics", ACM Transactions on Internet Technology, Vol. 1, No. 2, November 2001, Pages 151–198.

[2] Andrew F. Tappenden and James Miller K. Elissa, "Cookies: A deployment study and the testing implication", ACM Transactions on The Web, Vol. 3, No. 3, Article 9, Publication date: June 2009.

[3] A. X. Liu1, J. M. Kovacs, Chin-Tser Huang and M. G. Gouda, "A secure cookie protocol", 2005 IEEE.

[4] J. S. Park And R.I Sandhu, "Secure cookies on web**",** july - august 2000, IEEE Internet Computing.

[5] I. Ayadi , A. Serhrouchni, G. Pujolle and N.Simoni, "Http session management :Architecture and cookies security", 2011 IEEE.

[6] R. Putthacharoen And P. Bunyatnoparat, "Protecting cookies from cross site attacks using dynamic cookies rewriting technique", Feb. 13~16, 2011 ICACT2011.

[7] H. Wu, W. Chen And Z. Ren, "Securing Cookies Using Mac Address Key Ring", 2010 Second International Conference on Networks Security, Wireless Communications and Trusted Computing.

[8] Andrew T and James M, "A Three-Tiered Testing Strategy for Cookies", International Conference on Software Testing, Verification, and Validation(2008).

[9] Chuan Y, Mengjun X and Haining W, "Automatic Cookie Usage Setting with CookiePicker", 37th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN'07).

[10] Rui G and Bosheng Z, " Cross Cookie: A Cookie Protocol for Web Mashups". International Symposium on Electronic Commerce and Security,2008 IEEE.

[11] Hondo M , Nagaratnam N and Nadalin A, "Securing Web services" , IBM SYSTEMS, JOd URNAL, VOL 41(2002).

[12] Zhang L, Han W, Zheng D and Chen K, "A Security Solution of WLAN Based on Public Key Cryptosystem", Proceedings of the 2005 11th International, Conference on Parallel and Distributed Systems(2005).