Cookie-Proxy: A Scheme to Prevent SSLStrip Attack

Sendong Zhao[1,⋆], Ding Wang[1], Sicheng Zhao[2], Wu Yang[1], and Chunguang Ma[1]

[1] College of Computer Science and Technology, Harbin Engineering University, Harbin City 150001, China
[2] School of Computer Science and Technology, Harbin Institute of Technology, Harbin City 150001, China
zhaosendong@hotmail.com

Abstract. A new Man-in-the-Middle (MitM) attack called SSLStrip poses a serious threat to the security of secure socket layer protocol. Although some researchers have presented some schemes to resist such attack, until now there is still no practical countermeasure. To withstand SSLStrip attack, in this paper we propose a scheme named Cookie-Proxy, including a secure cookie protocol and a new topology structure. The topology structure is composed of a proxy pattern and a reverse proxy pattern. Experiment results and formal security proof using SVO logic show that our scheme is effective to prevent SSLStrip attack. Besides, our scheme spends little extra time cost and little extra communication cost comparing with previous secure cookie protocols.

Keywords: Secure Cookie Protocol, MitM, Defending against SSLStrip, SSL, Proxy Pattern, SVO logic.

1 Introduction

Recently, a new Man-in-the-Middle (MitM) attack called SSLStrip [1] is introduced at the Blackhat conference by Moxie Marlinspike in 2009. It attacks secure socket layer protocol, the most widely applied security mechanism which makes secure communication established between two parties over the Internet. Most seriously, this attack exploits user's browsing habits, rather than a technical pitfall in the protocol, to strongly defeat the SSL security. At a high level, the SSLStrip allows adversaries to insert themselves in the middle of a valid SSL connection. The user believes that they have a true SSL connection established, while the adversary has the ability to view the user's Web traffic in clear-text [1].

Owing to the serious damage of SSLStrip attack, some computer science researchers have paid much attention to prevent SSLStrip attack. Nick Nikiforakis *et al.* [2] presented a method to avoid the SSLStrip attack using browser's history information. A scheme with cue information is proposed by Shin and Lopes [3], which relies on web user's active exploration. Fung and Cheung [4] put forward a

⋆ Corresponding author.

T.W. Chim and T.H. Yuen (Eds.): ICICS 2012, LNCS 7618, pp. 365–372, 2012.

defending mechanism based on JavaScript code. Nevertheless, these suggestions only indicate ways to avoid a SSLStrip attack but not how to actively stop it. The cookie is a technical means to keep HTTP connection state, and it also can be used to save user's information [5]. When users relink the same Web server, browser will read the cookie information and send it to the Web site. Thus, the cookie should have the capability to check whether the HTTP connection is secure or not. Therefore, the cookie can be utilized to defend against SSLStrip attack. Some researchers have put forward some secure cookie protocols. Fu *et al.* [6] proposed a Web authentication scheme which mainly uses secure cookie as an authentication token. It uses cookie to store the authentication token with client. However, it has the following three flaws: it does not provide a high-level confidentiality; cookie replay attacks can be easily implemented on it; it is inefficient and non-scalable to defend against volume attacks. To overcome these weaknesses, Liu *et al.* [7] proposed a secure cookie protocol by improving Fu's scheme. Pujolle *et al.* [8] put forward a secure cookie protocol which implements a reverse proxy patterns [9]. Unluckily, these secure cookie protocols are both vulnerable to SSLStrip attack.

The remainder of this paper is organized as follows: In section 2, we propose a new scheme to prevent SSLStrip attack, followed by relative experiment, performance analysis, formal security analysis and comparison with previous schemes in section 3. Conclusion and the future work are discussed in section 4.

2 The Proposed Cookie-Proxy Scheme

In this section we propose a new scheme to defend against SSLStrip named Cookie-Proxy, including a secure cookie protocol and a new topology structure using a proxy pattern and a reverse proxy pattern [9]. To the best of our knowledge, this is the first time to defend against SSLStrip using secure cookie protocol. The notations of this paper are explained in Table 1.

Table 1. Notation

Symbol	Description
sk	Server Key of SSGP
$h(\bullet)$	Hash function
$E_{key}(M)$	Encrypt M using key key
$Sig_{key}(M)$	Sign M using key key
$HMAC_{key}(M)$	Keyed-hash message authentication code of M with key key
$\|\|$	String concatenation operation

2.1 A New Topology Structure

The topology structure mainly contains a secure LAN guaranteed proxy (SLGP) and a secure server guaranteed proxy (SSGP), as shown in Fig. 1. It is clear to see that SLGP and SSGP are implemented on the gateway of client LAN and

server Ethernet, respectively. To facilitate presentation, we assume that the LAN of clients is a secure LAN, which means that there are no attacks such as ARP spoofing attack, DNS spoofing attack and so on. This assumption can guarantee the absolute security of the client LAN.

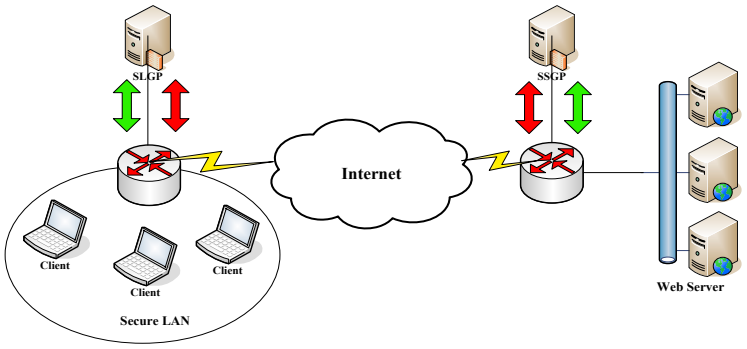

Fig. 1. The topology of our scheme

2.2 Our Secure Cookie Protocol

The secure cookie protocol shown in Table 2 is implemented between SLGP and SSGP. The cookie protocol implemented between the client and the server is still the commonly used cookie protocol [10]. The SLGP and the SSGP reconstruct commonly used cookie protocol and get the new form shown in Table 2. Recall that ks denoting the private key of SSGP for signature. SN denotes secure note of cookie, whose value is either 'Secure' or 'Unsecure'. ESN=$Sig_{ks}(h(SN \parallel CID))$. CID is a random digit for each cookie. k=$HMAC_{sk}$(username‖expires). $HMAC_k$=$HMAC_k$(username‖expires‖data‖CID‖SN).

Table 2. Our secure cookie protocol

Set-cookie Header							
username	expires	$E_k(data)$	SN	CID	ESN	$HMAC_k$	optional
Cookie Header							
username	expires	$E_k(data)$	SN			$HMAC_k$	optional

In our secure cookie protocol, there are some new attributes comparing with Liu's secure cookie protocol. SN denotes whether the HTTP connection is on SSL protocol (HTTPS) or not. If the value of SN is 'Secure', the connection is on SSL protocol and vice versa. The client browser will not upload cookie to the server as a request if the value of SN is secure and the protocol type of request URL is not HTTPS. The server will not return a login web page [10]. If the SSLStrip forges a login web page to SLGP, the SLGP will find and drop it. As a result, the client must send a request again. ESN is the signature of $h(SN \parallel CID)$ using private key ks.

2.3 The Mechanism of Cookie-Proxy Scheme

On client-side, the data from outside of the secure LAN is sent to SLGP. After processing, the data is sent to the gateway of the LAN again. On server-side, the data from inside and outside of server Ethernet is sent to SSGP. After processing, the data is sent to the gateway of the Ethernet again. The main function of SLGP is to check the integrity of essential attributes of set-cookie from server. The main function of SSGP is to reconstruct set-cookie from Web servers and check the integrity and validity of cookie from client. If the data itself is cipher-text, SLGP and SSGP will do nothing.

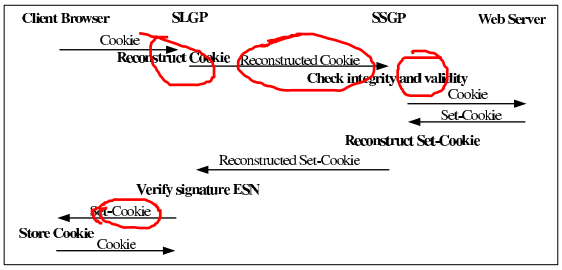

Fig. 2. The mechanism of Cookie-Proxy Scheme

According to Fig. 2, the operating mechanism of our scheme can be summarized as follows:

Step 1. Client sends request to server.
Step 2. SSGP receives cookie and checks its integrity and validity. Then, SSGP reconstructs the cookie and sends it to server.
Step 3. Server receives the request data and generates a set-cookie. Then send it to SSGP.
Step 4. SSGP reconstructs the set-cookie. Then send the modified set-cookie to client-side again.
Step 5. SLGP receives set-cookie and verifies the signature ESN. If the signature is not valid, drop the packet.
Step 6. SLGP send the set-cookie to client. The client receives the set-cookie and stores it in disk or RAM.
Step 7. Client sends request again with a new cookie.
Step 8. Go to Step 2.

In the above mechanism, SLGP should check whether set-cookie has SN or not. If not, SLGP drops this response packet. SLGP applies public key of SSGP to SN as a signature verification key verifies ESN as the message signed with the corresponding private key ks if the value of field SN is 'Unsecure'. If the signature is not valid, the response packet will be dropped.

Our scheme is composed of a secure cookie protocol, a SLGP and a SSGP. In order to achieve adequate security, all the components of our secure scheme are indispensable.

3 Experiment and Analysis

In this section, we analyze Cookie-Proxy scheme with an experiment and a formal proof, and then evaluate our scheme with other relevant schemes.

3.1 Experiment

In experiment, we utilize at least four computers and two routers to build our experimental platform, shown in Fig.1. Four computers are used as client, web server, SSLGP and SSGP respectively and two routers are deployed on client LAN and server LAN. Procedures running on client, SLGP and SSGP are based on the processes discussed in section 2. An adversary as a MitM can be of any computer in the Internet between SLGP and SSGP in Fig.1. Therefore, a computer is deployed between SLGP and SSGP to simulate the SSLStrip. The procedure running on simulated SSLStrip is in accordance with the process shown in Fig.3.

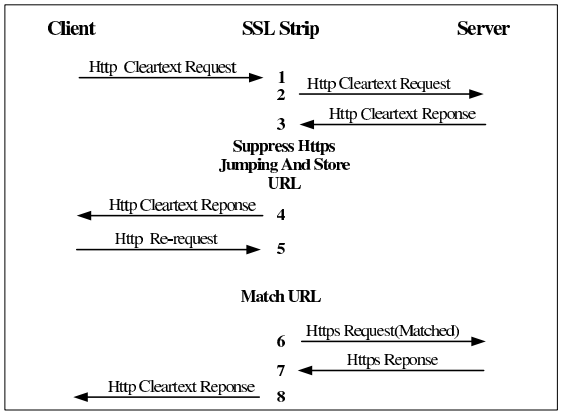

Fig. 3. The attack process of SSLStrip as MitM between client and server

To evaluate our scheme, five groups of experiment are conducted separately. On client-side, we use IE, FireFox, Oprea, Safari and Chrome as Web browser successively. The result of our scheme defending against SSLStrip attack is that our scheme can defend against SSLStrip attack effectively in all the five browsers.

3.2 Performance Analysis

Two HAMC operations and one verifying signature operation are added to our scheme, compared with traditional cookie protocols. HAMC operation is a kind of hash operation and the verifying signature operation is a modular exponentiation operation. The extra time of our scheme is composed of the consumption of two hash operations, a modular exponentiation operation and two symmetric cryptographic operations. In the process of protocol interaction, no extra

data transmission consumption is added to our scheme compared with traditional cookie protocol. Therefore, the total extra cost incurred by our scheme is $0.385(0.273 + 2 \times 0.026 + 2 \times 0.03)$ms according to [11]. Although our scheme needs some extra time, it improves the security when using HTTPS.

3.3 Security Analysis

In this section we present a formal analysis of the security properties of our new scheme, including the following five services: authentication, confidentiality, integrity, anti-replay and anti-SSLStrip attack. The SVO logic [12] analysis of our scheme is as follows:

M1: C → SLGP→ SSGP→ S: C_i
M2: S→ SSGP: username, expires, data, SN
M3: SSGP→ SLGP: username, expires,$\{\text{data}\}_k$, SN, $[h(SN)]_{ks^{-1}}$,{username, expires, data, SN$\}_k$
M4: SLGP→ C: username, expires,$\{\text{data}\}_k$, SN, {username, expires, data, SN$\}_k$
M5: Key agreement protocol
M6: C→ SLGP→ SSGP→ S: {username, expires, $\{\text{data}\}_k$, SN, {username, expires, data , SN$\}_k\}_{session-key}$

The premises of SVO logic are as follows:

A1 indicates the basic assumption, the scheme is running in unsecure environment; A2 indicates public key of each principal is public; A3 indicates private key of each principal is known by itself only; A4 S believes fresh(k); A5 S believes fresh(CID); A5 C control M1/M4/M5; A6 S believes M2/M4/M6; A7 C believes PK(S,ks)∧C received $\{\text{X}\}_{ks^{-1}}$∧C received X⊃C believes (S said X); A8 C believes (C$\xleftarrow{session-key}$ S), S believes (C$\xleftarrow{session-key}$ S); A9 SLGP believes SLGP received {username, expires, $*_1$, SN, CID, $[h(SN)]_{ks^{-1}}$,$*_2$}; A10 SSGP believes SSGP received {username, expires, $\{\text{data}\}_k$, SN, {username, expires, data , SN$\}_k$ $\}_{session-key}$; A11 C controls SV(SN|CID,ks,ESN).

The goals are as follows:

G1: S believes (C said (username, expires, data)∧S believes (username, expires, data)) By A5, A10, A8, MP
G2: S believes C says (username, expires, data) By A10, A8, A6, A4,Ax3, Nec, MP
G3: C believe S says SN By A9, A11, A5, MP

Authentication: In our scheme, the server can get the cookie from client, in which every field is the original value by using detection of SSGP. So the server can authenticate the client exactly. Obviously, the set-cookie reconstructed by SSGP is hardly to forge. G1 denotes the authentication of our scheme. Above all, the server can verify the validity of the client easily and accurately.

Confidentiality: In our scheme, SSGP encrypts data field in set-cookie. By this way, the data is invisible to client. Thus, the confidentiality is provided in our new scheme.

Integrity: If adversaries have modified the field SN of set-cookie from server or deleted this set-cookie fields, SLGP will check the integrity of domain items, and verify signature ESN. If there are any abnormalities, SLGP will drop the whole response packet. The goal G3 ensures this process.

Anti-replay: Obviously, in our scheme each cookie has its unique HMAC. If an adversary replays a cookie as a request to the server, SSGP will detect it and drop it. Also, G2 denotes the anti-repay of our scheme.

Anti-SSLStrip: The core of SSLStrip attack is to make users not recognize what the connection should be, HTTP or HTTPS. Nevertheless, our secure cookie protocol has mandatory fields: SN and ESN. The SLGP will check these fields, so the client will definitely know what the connection should be. For this reason, our scheme can defend SSLStrip attack. Also, the goal G3 ensures that MitM cannot modify the field SN.

3.4 Comparison of Relevant Cookie Protocols

In this section, we compare our scheme with the other relevant cookie protocols. Without loss of generality, the digit fields are all recommended to be 128-bit long, while the string fields are all 1024-bit long in protocols. Let T_H, T_E, T_I and T_S denote the time complexity for hash function, exponential operation, inverse operation and symmetric cryptographic operation. Typically, time complexity associated with these operations can be roughly expressed as $T_E \approx T_I > T_S \geqslant T_H$ [11].

Table 3. Comparison with relevant schemes

Item	**Ours**	**Fu *et al.* [6]**	**Liu *et al.* [7]**	**Pujolle *et al.* [8]**
Extra time cost	$T_E + 2T_H + 2T_S$	T_H	$2T_H + 2T_S$	$2T_H + 6T_S$
Extra communication cost	2560 bit	256 bit	2304 bit	2432 bit
Authentication	Yes	Yes	Yes	Yes
Confidentiality	Yes	No	Yes	No
Integrity	Yes	No	No	No
Anti-replay	Yes	No	Yes	Yes
Anti-SSLStrip	Yes	No	No	No

From Table 3, we can conclude that our scheme spends not so much extra time cost and extra communication cost, but it gets the ability of withstanding SSLStrip attack and the integrity of protocol comparing with other cookie protocols.

4 Conclusion

Our contributions in this paper are twofold. Firstly, we elaborate on the drawbacks of state-of-the-art secure cookie protocols. Secondly, we present a Cookie-Proxy scheme to defend against SSLStrip attack. The evaluation shows the

effectiveness of our scheme. However, due to the serious damage of SSLStrip attack, the cost of resisting against this attack is inevitably high. In the future, we will focus on improving the deploy-ability of our scheme and reducing security requirement of client LAN.

Acknowledgement. This paper is funded by the International Exchange Program of Harbin Engineering University for Innovation-oriented Talents Cultivation.

References

1. Callegati, F., Cerroni, W., Ramilli, M.: Man-in-the-middle attack to the https protocol. IEEE Security Privacy 7(1), 78–81 (2009)
2. Nikiforakis, N., Younan, Y., Joosen, W.: HProxy: Client-Side Detection of SSL Stripping Attacks. In: Kreibich, C., Jahnke, M. (eds.) DIMVA 2010. LNCS, vol. 6201, pp. 200–218. Springer, Heidelberg (2010)
3. Shin, D., Lopes, R.: An empirical study of visual security cues to prevent the sslstripping attack. In: Proceedings of the 27th Annual Computer Security Applications Conference, ACSAC 2011, pp. 287–296. ACM, New York (2011)
4. Fung, A.P.H., Cheung, K.W.: Sslock: sustaining the trust on entities brought by ssl. In: Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security, ASIACCS 2010, pp. 204–213. ACM, New York (2010)
5. Liu, A., Kovacs, J., Gouda, M.: A secure cookie scheme. Computer Networks 56(6), 1723–1730 (2012)
6. Fu, K., Sit, E., Smith, K., Feamster, N.: Dos and donts of client authentication on the web. In: Proceedings of the 10th Conference on USENIX Security Symposium, SSYM 2001, vol. 10, pp. 19–35. USENIX Association, Berkeley (2001)
7. Liu, A., Kovacs, J., Huang, C.T., Gouda, M.: A secure cookie protocol. In: Proceeding of 14th International Conference on Computer Communications and Networks, ICCCN 2005, pp. 333–338 (October 2005)
8. Pujolle, G., Serhrouchni, A., Ayadi, I.: Secure session management with cookies. In: Processing of 7th International Conference on Information, Communications and Signal, ICICS 2009, pp. 1–6 (December 2009)
9. Sommerlad, P.: Reverse proxy patterns. In: European Conference on Pattern Languages of Programming, EuroPLoP 2003 (2003)
10. Barth, A.: HTTP State Management Mechanism, IETF Internet-Draft (2010), `https://datatracker.ietf.org/doc/draft-ietf-httpstate-cookie/`
11. Wang, D., Ma, C., Weng, C., Jia, C.: Cryptanalysis and Improvement of a Remote User Authentication Scheme for Resource-Limited Environment. Journal of Electronics & Information Technology (in press, 2012)
12. Syverson, P., Van Oorschot, P.: A unified cryptographic protocol logic. Technical report, DTIC Document (1996)