Volume 4, Issue 1, January 2014 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Analysis of Security threats over TLS/SSL

Jateen Gadhiya*, Naimisha Trivedi
LDCE, Gujarat Technological University,
India

Abstract— Secure Socket Layer(SSL) also known as Transport Layer Security is de facto standard for web security. It provide confidentiality and integrity of information in transit across the public networks using their powerful cipher suites but still it contains some loopholes or flaws in its foundation. In this paper we discuss TLS standard along with various attacks found in recent years like CRIME, Beast, Lucky 13, BREACH etc. and their proposed mitigations. We analyse current state of e-commercial internet security. Also discuss security related issues and problem already exist in the system..

Keywords— Attack, Compression, Mitigations, Security, TLS.

I. INTRODUCTION

In our day to day life, we generally perform many activities over internet like online shopping, cash transfer, profiling etc. which required highly confidential data input. All those data transfer carried through wired or wireless network. So powerful security mechanism is implemented at transport layer of TCP/IP protocol stack known as Transport layer security (TLS) or secure socket layer*. Internet users normally think that they secure whenever there is HTTPS instead of HTTP in address bar of their browser, but this is not obvious scenario. Recently many attacks were discovered over TLS.

In this paper, first we understand security mechanism provided by TLS. Later on we discuss various vulnerability exist as well as attacks designed over them. Finally, examine current state of internet security.

II. TRANSPORT LAYER SECURITY

Secure Socket Layer (SSL), now known as Transport Layer Security(TLS) is firstly developed by Netscape foundation. SSL Version 1.0 was never published, but SSL version 2.0 was officially released in 1995[11].

A. Security Mechanism of TLS

TLS is trio of cryptographic services:

- Authentication
- Confidentiality
- Integrity

The Protocol consists of various cipher suites for secure communication. Authentication is achieved by asymmetric ciphers like RSA, Diffie-hellman etc. Confidentiality is obtained by performing symmetric encryption of plaintext transfer through networks. Common powerful symmetric ciphers implemented by TLS are AES, DES-3, RC4 etc. Integrity is achieved by calculating Message Authentication Code (MAC) of packets by MD5 or SHA-1. The whole flow of all processes is shown in fig. 1. It is also known as HEE.

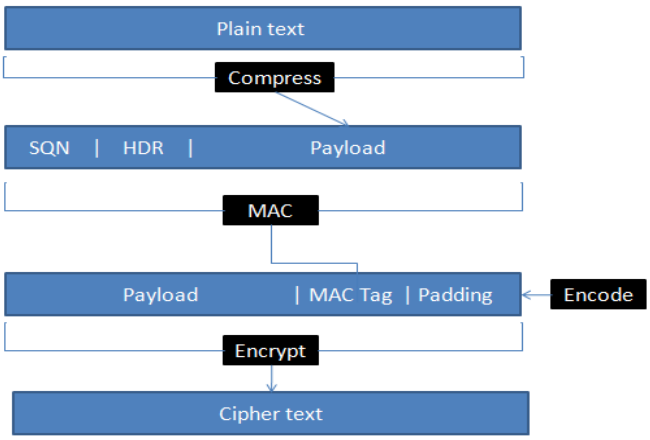

Figure 1. Security Mechanism provided by TLS

HEE stands for Hash-Encode-Encrypt. First of all, plaintext from application layer is compressed by HTTP Compression. Various algorithms for compression are Deflate and GZIP. Compressed text from application layer is input

to TLS protocol layer. At TLS level, the compressed text is become payload of data packet. MAC is calculated over that packet and appended to original data. Then after, Encoding is performed. This is optional one, only required if block cipher is implemented. After Encoding, symmetric encryption is performed over data or packet. This process is performed on both the side i.e. client and server.

B.TLS Protocol Stack

TLS/SSL is designed to make use of TCP to provide a reliable end-to-end secure service. SSL is not a single protocol but rather two layers of protocols as illustrated in Figure 2.

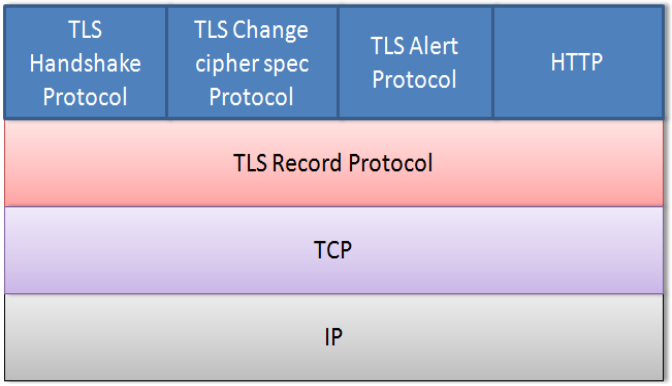

Figure 2 TLS Protocol Stack

The handshake protocol The handshake protocol having the most tedious part of the SSL protocol. It is used to initialization of a session between two parties. Within the message of this protocol, various parameters such as algorithms and keys used for data encryption are negotiated. Because of this protocol, authentication of parties to each other and negotiate appropriate parameters of the session between them is possible.

The Change Cipher Spec protocol This protocol is the simplest and easiest TLS protocol. It consists of a single message that carries the value of 1. The main aim of this message is to cause the pending session state to be established as a fixed state, which results, for example, in defining the used set of protocols. This type of message must be sent by the client to the server and vice versa. After exchange of messages, the session state is considered agreed. This message and any other TLS messages are transferred using the TLS record protocol.

The Alert Protocol The Alert Protocol is used by parties to convey session messages associated with data exchange and functioning of the protocol. Each message in the alert protocol consists of two bytes. The first byte always takes a value, "warning" (1) or "fatal" (2) , that determines the severity of the message sent.

The Record Protocol is a heart of Transport Layer Security. It provide confidentiality and Integrity by encryption and MAC generation. The full mechanism is already explained in section A.

III. SECURITY THREATS ON TLS/SSL

Since last decade, Many flaws has been identified and their respective attacks was developed against security provided by TLS/SSL. Some well known attacks are discussed below along with their analysis of impact on sensitive information.

A. BEAST

BEAST stands Browser Exploit Against SSL/TLS[3]. Researchers Thai Doung and Juliano Rizzo found a way to exploit the vulnerability and demonstrated a live attack against Paypal at the Ekoparty security conference in September of 2011. Beast is type of chosen plaintext attack and violate the same origin policy. TLS generally uses two mechanism for security, first one is Initialization Vectors(IV) and Cipher Block Chaining Mode (CBC) in their cipher suite. But, Beast exploit this mechanism. In CBC mode, IV and Plaintext are two input, but for every block in CBC, the cipher text from previous block is IV for the next block. Due to this vulnerability, attacker is able predict the input for next encryption by implementing MITM. IV generally used to make encryption nondeterministic, it add randomness to encryption flow. Requirements for successfully perform the attack are: 1) Passive Network Eavesdropping, 2)Chosen boundary Format Privilege and 3) chosen block wise plaintext injection. This attack is feasible if attacker acquire all above mentioned necessity.

Countermeasures have been already taken by IETF by their version of TLS 1.1. In TLS 1.1 and 1.2, the Galois Counter Mode(GCM) is prioritize against CBC mode of encryption, so it will patch out vulnerability due to CBC. Many other mitigations are also provided by different browser vendors by appending patch and enabling TLS version 1.1 and preferably v1.2.

B. CRIME

CRIME Stands for Compression Info-Leak Mass Exploitation[2]. It is compression side channel based attack. It reveals plaintext information by using compression information of TLS compression. The combination of plaintext injection and information leakage due to compression is lead to development of this attack. It target the Http Header and information relies in that header like secret cookies, session IDs etc. This may lead the attacker to successfully perform session hijacking on secure web connection. More the redundancy in plaintext, higher the compression ratio and smaller the data transmission. The requirement for this attack is TLS compression enabled web server and privilege for Man in the Middle(MITM) to observe and fetch web traffic. Countermeasure of the attack is already been adopted by disabling

TLS level compression by latest version of web browser like Chrome, Mozilla, IE etc. and this will not affecting performance much more. So, the current TLS server are not vulnerable to this attack.

C. BREACH

BREACH stands for Browser Reconnaissance and Ex-filtration via Adaptive Compression of Hypertext[4]. It attack on HTTP responses of web application of a compression side-channel by CRIME style mechanism. Yoel Gluck, Neal Harris, and Angelo Prado give practical demonstration of BREACH at BlackHat USA 2013. In CRIME attack model, it gives information about session cookies to the attacker who is able to measure the size of these requests and inject chosen plaintext into the user's Http request. In September 2012 when major vendors disabled TLS compression in browser and server-side, the CRIME attack was effectively mitigated. TIME resurrected the CRIME attack focusing on the time differential in the HTTP response, occurring due to the difference of size as an effect of HTTP compression of the response body. Finally, BREACH revived the CRIME attack by targeting the size of compressed HTTP responses and extracting secrets hidden in the response body.

Once again, like the CRIME attack, BREACH exploited the compression and encryption combination used to interact with users and web-servers. The working mechanism of BREACH is similar to CRIME, except CRIME targeted TLS compression, while BREACH targets HTTP compression. HTTP response compression compresses the body of responses but not header information. The algorithm used, DEFLATE[6], is comprised of two components. LZ77 replaces occurrences of three or more characters with "pointer" values to reduce space. Huffman coding replaces characters with symbols in order to optimize the description of the data to the smallest size possible. BREACH works by attacking the LZ77 compression while minimizing the effects of Huffman coding. If this isolation is not performed, too many false positives will result, reducing the effectiveness of the attack.

As BREACH focuses on the HTTP compression of the response body, it is possible to mount on all versions of SSL/TLS, and does not require TLS-layer compression[8]. The cipher suite used during the session negotiation does not affect this attack. The number of requests required are proportional to the size of secret, but in general BREACH attack can be exploited with just a few thousand requests, and under a minute. In short, the scope of this attack includes a considerable portion of the HTTP traffic in the Internet as a large portion of enterprise applications and online websites use HTTP compression to optimize bandwidth. The three main requirements for exploitation of the vulnerability to be effective are:

1. The application supports HTTP compression.
2. The response should reflect back user's input.
3. The response should have some sensitive/secret information embedded in the body.

If the user's input is not reflected, there is no possible way to mount a chosen plaintext attack and measure the size of the responses. This attack targets the secret information in the response body (e.g. CSRF tokens), not the session cookie in the request header. So this is useful only if the response of this attack contains sensitive information.

D. Lucky 13

The Lucky Thirteen attack[1] is a cryptographic timing attack against implementations of the Transport Layer Security (TLS) protocol, first reported in February 2013 by its developers Nadhem J. AlFardan and Kenneth G. Paterson of the Information Security Group at Royal Holloway, University of London. A Message Authentication Code (MAC) is used to authenticate and to provide integrity of the message. The best practice is to encrypt a message first, then apply the MAC on the resulting ciphertext. However, in TLS it is done in a different fashion. The message is added in the block, a MAC is applied to the plaintext, and then up to 255 bytes of padding are added to grow the message to a multiple of the cipher block size (8 or 16-byte). This message block is finally CBC-encrypted. CBC mode decryption takes the current encrypted block, decrypts it, and XORs in the previous ciphertext block. After decryption, the padding is first validated, and after successful validation it is removed and then integrity of data is checked against the calculated MAC. However, this method of CBC-encrypting in TLS has a problem with protecting the padding. The padding oracle attack is applicable to the implementations of SSL 3.0 and TLS 1.0. The padding is not protected by the MAC, so an attacker can tamper with the padding and perform a padding oracle attack. During decryption of the message, the padding is checked first. If there is a valid padding, only then the MAC is checked; otherwise the server throws an error stating whether that an invalid padding or MAC error has occurred. The padding oracle attack uses CBC decryption to determine the plaintext by modifying the previous ciphertext block. An attacker can modify the encrypted message based on these error messages, and after repetitive requests can eventually get the message decrypted by the server without the encryption key.

Countermeasures for lucky thirteen is move towards other mode of encryption like AES-GCM or AES-CCM instead of CBC along with RC4. This may also helpful against other attacks like BEAST. Also by careful implementation of all MAC-then-Encode-then-Encrypt(MEE) this Lucky 13 attack can be mitigated.

IV. SURVEY REPORT ON TLS SERVER

Qualys SSL Labs made non-commercial research work over SSL/TLS implementations of well known top web sites of today's internet. Some of them are facebook.com, google.com, ymail.com, youtube.com etc. As a part of their survey they come with the following results. SSL Pulse[7], a service from SSLLabs.com had surveyed above 150 thousands TLS server based on handshake parameters, cipher suites implemented, certificate chain and many more parameters. From survey they conclude that 47.9% TLS server out of total surveyed are unsecured by different means and vulnerabilities.

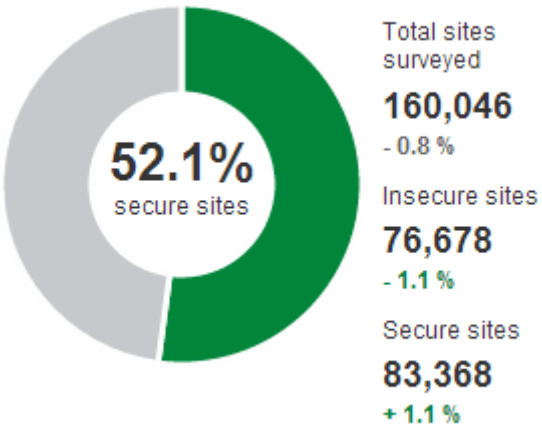

Figure 3 Summary from SSL Pulse

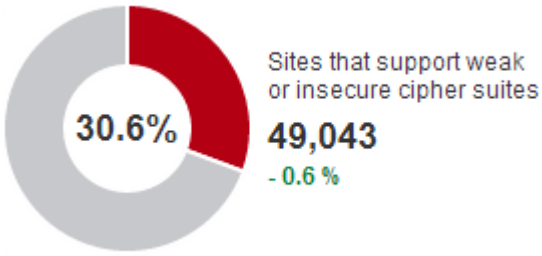

Figure 4 survey result for cipher suites

As shown in above figures, figure 3 shows summarization of total survey, from that 76,678 servers have insecure implementation of TLS. In figure 4, it shows that 30.6% TLS server support insecure cipher suites. Here, insecure means having support for ciphers less than 128 bits. They also made work over attacks that we discussed earlier.

Figure 5 BEAST attack

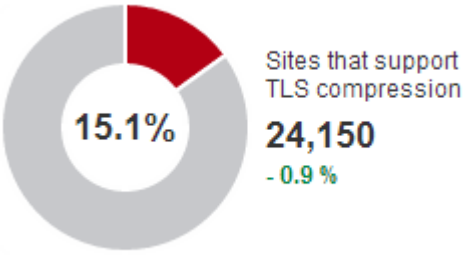

Figure 6 CRIME attack

As we discuss that earlier version TLS 1.1 are vulnerable to BEAST attack and TLS server that enabled TLS compression are vulnerable to CRIME attack. Figure 5 shows that 69.6% i.e 1,11,445 TLS server are insecure from BEAST attack and 15.1% TLS server are still vulnerable to CRIME attack, but this is only because implementers or developers generally unaware of attacks designed against TLS/SSL.

V. CONCLUSIONS

There is no any better option for TLS for online security for sensitive information, but TLS is vulnerable many loopholes and side channel leakage. Also many attacks practically performed based on that vulnerability that we discussed like BEAST, CRIME, BREACH, Lucky 13 etc. We also analyze survey report made by SSL Labs. The result from survey shows bad results for online security. So, there is need for some important changes in TLS implementation and it's foundation. Also need for research work in attacks designed. Some of the proposed changes may be eliminating certificate transmission, mandatory forward secrecy etc[12].

REFERENCES

[1] AlFardan, Nadhem J., and Kenneth G. Paterson. "Lucky thirteen: Breaking the TLS and DTLS record protocols." *IEEE Symposium on Security and Privacy*. 2013.
[2] Rizzo, Juliano, and Thai Duong. "The CRIME attack." *ekoparty Security Conference*. Vol. 8. 2012.
[3] Duong, Thai, and Juliano Rizzo. "Here come the⊕ Ninjas." *Unpublished manuscript* (2011). Cited count 9
[4] GLUCK, YOEL, NEAL HARRIS, and ANGELO ÁNGEL PRADO. "BREACH: REVIVING THE CRIME ATTACK." (2013).
[5] Rizzo, Juliano, and Thai Duong. "Practical padding oracle attacks."*Proceedings of the 4th USENIX conference on Offensive technologies, WOOT*. Vol. 10. 2010.
[6] Deutsch, L. Peter. "DEFLATE compressed data format specification version 1.3." (1996).
[7] As of October 02, 2013. "SSL Pulse: Survey of the SSL Implementation of the Most Popular Web Sites". Retrieved 2013-10-10.
[8] Kelsey, John. "Compression and information leakage of plaintext." *Fast Software Encryption*. Springer Berlin Heidelberg, 2002.
[9] Soghoian, Christopher, and Sid Stamm. "Certified lies: Detecting and defeating government interception attacks against ssl (short paper)." *Financial Cryptography and Data Security*. Springer Berlin Heidelberg, 2012. 250-259.

[10] Holz, Ralph, et al. "X. 509 forensics: Detecting and localising the SSL/TLS men-in-the-middle." *Computer Security–ESORICS 2012*. Springer Berlin Heidelberg, 2012. 217-234.
[11] Wikipedia contributors. "Transport Layer Security." *Wikipedia, The Free Encyclopedia*. Wikipedia, The Free Encyclopedia, 4 Dec. 2013. Web. 10 Dec. 2013.
[12] Corella, Francisco, and Karen Lewison. "It Is Time to Redesign Transport Layer Security." (2013).

