The Study of Recent Security Attacks on Secure Socket Layer (SSL)

S. R. Ponde

Abstract – **Netscape Communication developed Secure Socket Layer Protocol (SSL later renamed to Transport Layer Security (TLS)) in 1994 for application independent secure communications over the Internet. SSL/TLS can be used for ensuring data confidentiality, integrity and authenticity during transmission of data. Since 1994, lots of vulnerabilities have been discovered in the Transport Layer Security protocol. This paper focuses on analysis of recent attacks on SSL/TLS.**

Key words – **SSL, TLS, BEAST, CRIME, Lucky 13, RC4**

I. INTRODUCTION

SSL and TLS:

As the people are using internet for many of business transactions it is important, it should be done securely without ant intervention from fraudulent. To avoid this insecurity and events of attacks or frauds, technology implemented as Secure Sockets Layer (SSL) and Transport Layer Security (TSL) protocols. It can be applied to web application and the requirements necessary to create a secure link between a server and a client machine.

Secure Socket Layer Protocol

SSL is the secure communications protocol of choice for a large part of the Internet community. There are many applications of SSL in existence, since it is capable of securing any transmission over TCP. Secure HTTP, or HTTPS, is a familiar application of SSL in e-commerce or password transactions. (Viega, 10)

According to the Internet Draft of the SSL Protocol, the point of the protocol "is to provide privacy and reliability between two communicating applications." (Freier, 3.)

The protocol release further explains that three points combine to provide connection security. These points are:

- Privacy - connection through encryption
- Identity authentication – identification through certificates, and
- Reliability –dependable maintenance of a secure connection through message integrity checking.

The current version of SSL is version 3.0, released by Netscape in 1999.

Transport Layer Security Protocol

The Internet Engineering Task Force (IETF) established the Transport Layer Security (TLS) Working Group in 1996 to come up with a standardized version of SSL and Privacy Communication Technology (PCT). The standardized protocol [1], imaginatively named TLS version 1.0, is very similar to SSL 3.0. So much so that TLS 1.0 is sometimes referred to as SSL 3.1.

The Transport Layer Security (TLS) protocol was released in January 1999 to create a standard for private communications. The protocol "allows client/server applications to communicate in a way that is designed to prevent eavesdropping, tampering or message forgery." (Dierks, 1)

According to the protocol's creators, the goals of the TLS protocol are cryptographic security, interoperability, extensibility, and relative efficiency. (Dierks, 4)

Following are the attacks found on SSL/TSL:

a) Browser Exploit Against SSL/TLS (BEAST) attack
b) Compression Ratio Info-leak Made Easy (CRIME)
c) Lucky 13 attack
d) RC4 attack

Above are the recent attacks on SSL/TSL, and discussed in this paper.

II. BEAST

The BEAST (Browser Exploit Against SSL/TLS) was developed by researchers Thai Duong and Juliano Rizzo [2] and can be carried out on TLS v1.0.TLS v 1.2 is not vulnerable to a BEAST attack. The CVE for a BEAST attack is CVE-2011-3389 [3].

Previously introduced attacks targeting SSL/TLS protocols were primarily based on falsified server certificates and man-in-the-middle-type traffic hijacking. As a departure from this, BEAST concentrates on cracking the encryption of the protocol and hijacking the session cookie.

In order to carry out successful BEAST attacks, the attacker must be able to insert his or her own program code

Associate Professor, Sinhgad Institute of Business Administration and Computer Application, Lonavala
sachinponde@gmail.com

to be run in the victim's browser. The attacker uses it to generate large amounts of known traffic that can be utilized in cracking the encryption. In addition to this, the attacker must be able to listen to the network traffic of the attack target.

Log in to any https page, after your authentication you can see your authenticated page and, if you look carefully at the URL, you can see the session ID. A session ID is a random number or combination of numbers and string that maintains the state of the page; it is assigned by the website server to the client browser. The Session ID can be found either in the cookie or in the URL of the web browser. Usually, all the session IDs will be encrypted to prevent hijacking of the session.

BEAST attack can be divided into three steps for simplicity.

Step 1: An attacker sends a malicious JavaScript to run on your machine (this can be sent via CSRF, Social engineering, A Drive-by download, the returned page can contain a JavaScript, etc.).This malicious script runs on the victim's machine and can capture the entire header info and the encrypted cookie that is assigned from the web server (running TLS 1.0) and can then send the information to any website.

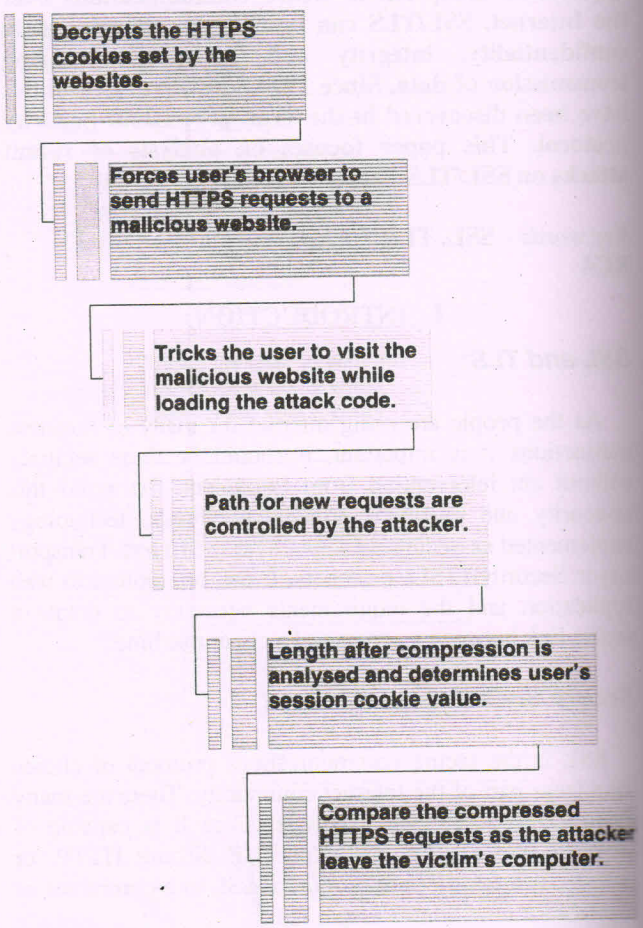
Step 2: SSL/TLS can encrypt data with two kinds of ciphers: block ciphers, such as AES and DES, and stream ciphers like RC4.TLS v1.0 gives precedence to the block cipher rather than stream ciphers. This is where vulnerability exists. There are two identical plain text messages then, after encryption, the same cipher text so the pattern in plaintext is reflected in the cipher text. This is bad. In order to prevent this, use cipher block chaining (CBC mode chaining). In CBC, if require to encrypt block A, first apply XOR with A-1. If it is the first block we cannot XOR with A-1 data so here take the initialization vector.

Step 3: The attacker compares the encrypted session details and the unencrypted data sent by the script to find the initialization vector. Once this information collected, and then decrypts the future cookies sent from the web server.

The attack was fixed in version 1.1 of the Transport Layer Security (TLS) protocol, but a lot of servers continue to support older and vulnerable protocols, like SSL 3.0, for backward compatibility reasons. Such servers are vulnerable to so-called SSL downgrade attacks in which they can be tricked to use vulnerable versions of SSL/TLS even when the targeted clients support secure versions

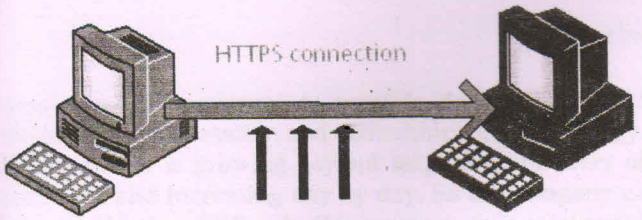
III CRIME

CRIME (Compression Ratio Info-leak Made Easy/Compression Ratio Info-leak Mass Exploitation) is a new attack that was developed by two security researchers, Juliano Rizzo and Thai Duong. It decrypts the session cookies from the hypertext transfer protocol secure (HTTPS) connections by means of brute force. The so-called CRIME attack [4] induces a vulnerable web browser get into a cookie authentication, created when a user starts a HTTPS session with a website.



The obtained cookie can be used by hackers to log in to the victim's account on the site. The cookie is obtained by tricking the browser into sending encrypted compressed requests to secure websites and exploiting the information negligently leaked in the process. Some extra data that has been tweaked by malicious JavaScript code is also embedded along with the cookies within each request. The differences in size of the compressed messages are measured to determine the cookie's contents, character by character. This is possible because TLS/SSL and SPDY use a compression algorithm called DEFLATE [5], which works by eliminating duplicate strings. Compression is a

mechanism to transmit or store the same amount of data in fewer bits. The main compression method used in TLS to compress data is DEFLATE.



Hijacking the session by decrypting the session cookies

CRIME works against TLS/SSL Compression and SPDY (a special HTTP-like protocol developed by Google, and used cautiously around the web).

IV. LUCKY 13

The new flaw has been found last year in February 2013 called as 'Lucky 13'. The TLS MAC calculation includes 13 bytes of header information (5 bytes of TLS header plus 8 bytes of TLS sequence number) which is part of attack. The process will theoretically allow man-in-the-middle attacks against SSL protected communications

Nadhem AlFardan and Kenny Paterson of the Information Security Group at Royal Holloway, University of London, announced a new TLS/DTLS attack called Lucky Thirteen. The attack allows a man-in-the-middle attacker to recover plaintext from a TLS connection when CBC-mode (cipher-block chaining) encryption is used [6].

The attack exploits a problem with the TLS specification and not a bug in specific implementations. This is not a problem with certification authorities or issued SSL/TLS certificates.

Lucky Thirteen uses a known timing attack previously believed to be impractical. There is a subtle timing bug in the way that TLS data decryption works when using the (standard) CBC-mode cipher suite. Given the right set of circumstances, an attacker can use this to decrypt sensitive information, such as passwords and cookies.

The attacks apply to all implementations that conform to TLS version 1.1 or 1.2, or DTLS version 1.0 or 1.1. They also apply to implementations of SSL 3.0 and TLS 1.0 that have countermeasures designed to defeat a previous padding oracle attack discovered several years ago. All TLS and DTLS cipher suites that include CBC-mode encryption are potentially vulnerable.

It uses very small differences between the amount of time it takes to decrypt a correctly padded TLS record and the time it takes to reject an incorrectly padded record (kind of a checksum). An attacker can modify encrypted records, send them to the server and discover whether the decryption is correctly padded or not based on the amount of time it takes to respond. With this knowledge, an attacker can interactively decrypt a record bit by bit, testing the padding over and over.

It only works in datagram TLS (DTLS) because regular TLS terminates a session after one incorrectly padded message. It also only works over LAN where you can get really precise timing.

V. RC4 Attack

Ron Rivest of RSA Security designed RC4 (Rivest Cipher 4) in 1987. Because of its simplicity and speed, its become most widely used stream cipher. It is used in common protocols such as Wired Equivalent Privacy (WEP), a security algorithm for wireless networks, and the Secure Sockets Layer (SSL) and Transport Layer Security (TLS) protocols of HTTPS. In fact, around 50% of all TLS traffic is currently protected using the RC4 algorithm. However, weaknesses have been found in the algorithm over the years indicating that, RC4's life is coming to the end.

A new type of attack against the Transport Layer Security and Secure Sockets Layer protocols has been demonstrated by security researches during the 20th International Workshop on Fast Software Encryption.

The flaw is possible – in theory - because of a weakness in the RC4 algorithm that is widely used in SSL/TLS certificates, among others, such as WEP (Wired Equivalent Privacy), WPA (Wi-Fi Protected Access), TKIP (Temporal Key Integrity Protocol), and Microsoft XBOX.

RC4 is a symmetric stream cipher with an arbitrary key size generated by a pseudo-random number generator. Since there is no randomization in computerland, the password determines the generator's initialization value.

The attack envisioned by Nadhem AlFardan, Dan Bernstein, Kenny Paterson, Bertram Poettering and Jacob Schuldt relies on "statistical flaws in the keystream generated by the RC4 algorithm, which become apparent in TLS ciphertexts when the same plaintext is repeatedly encrypted at a fixed location across many TLS sessions." [7]

To successfully decrypt traffic, attackers need to receive tens of millions of different encryptions of the same message, but they are confident they can automate the process by renegotiating the connection with the server, which would send the same data (such as a login cookie) encrypted with a different key.

The attack is strictly a partial proof of concept. Even though it has not been rendered functional yet, this theoretical approach is another failure start for TLS, a protocol that has been affected by other attacks, such as BEAST, CRIME, and Lucky 13.

This is a really clever attack on the RC4 encryption algorithm as used in TLS. A new attack against TLS that allows an attacker to recover a limited amount of plaintext from a TLS connection, when RC4 encryption is used. The attacks arise from statistical flaws in the keystream generated by the RC4 algorithm which become apparent in TLS ciphertexts when the same plaintext is repeatedly encrypted at a fixed location across many TLS sessions.

As the attack is carried out in multiple sessions, we must require a target plaintext to be repeatedly sent in the plaintext stream in multiple TLS sessions.

Only first 256 bytes of the plaintext stream targets currently by the attack in the session.

Since the first 36 bytes of plaintext are formed from an unpredictable finished message when SHA-1 is the selected hashing algorithm in the TLS Record Protocol, these first 36 bytes cannot be recovered. This means that the attack can recover 220 bytes of TLS-encrypted plaintext.

VI. REFERENCES

[1] T. Dierks and C. Allen. The TLS protocol, version 1.0, January 1999. RFC-2246.

[2] http://vnhacker.blogspot.com/2011/09/beast.html

[3] http://www.cvedetails.com/cve/CVE-2011-3389

[4] https://www.imperialviolet.org/2012/09/21/crime.html

[5] https://tools.ietf.org/html/rfc1950

[6] http://eprint.iacr.org/2004/111.pdf

[7] http://www.isg.rhul.ac.uk/tls/RC4biases.pdf

[8]http://www.educatedguesswork.org/2011/09/security_impact_of_the_rizzodu.html

[9]https://www.ssllabs.com/downloads/SSL_TLS_Deployment_Best_Practices_1.0.pdf

[10]http://thehackernews.com/2012/09/crime-new-ssltls-attack-for-hijacking.html

[11]Freier, Alan O. and Philip Karlton and Paul C. Kocher. "The SSL Protocol Version 3.0." November 1999. (14 March 2003)

[12]Dierks, T. & C. Allen. "The TLS Protocol Version 1.0." January 1999. (16 February 2003)

[13]Viega, John, Matt Messier and Pravir Chandra. Network Security with OpenSSL. Sebastopol: O'Reilly & Associates, Inc. 2002.

AUTHOR

Mr. S. R. Ponde was born in 1972. He received M.C.A. degree, in Government Engineering College, Aurangabad from Dr. B. A. M. University, Aurangabad. He joined as a Sr. Lecturer for MCA course in Sinhgad Institute of Business Administration and Computer Application, Lonavala in 2007. Presently he is working as a Associate Professor. He has published books for MCA and BCA courses.