**SSL/TLS Project Specifications**

**Title:** A Comprehensive Study of the BREACH Attack Against HTTPS

**Group members**: Esam Alzahrani, Justin Nonaka, and Thai Thruong

**Introduction:**

Secure Sockets Layer (SSL) and its successor, Transport Layer Security (TLS), are application layer protocols that provide message security by encrypting transport layer data. As mentioned how SSL provides security to internet communications, so it can be used with any protocol. Basically, SSL can be operates with any protocol to form secure communications. One well-known protocol is HTTP which operates on application layer, the top layer, to offer data communication over Internet. The secure version of HTTP is referred as HTTP secure, it is underlying SSL to form secure data communication. HTTP preforms compression before sending out data from web server to reduce used bandwidth and transmission time accordingly with reduced size after compression. Any method can be used to perform compression; however, the most common methods are gzip and deflate. Also, there is a list contains compression algorithms that are supported by HTTP.

Recently, BREACH is announced as a serious threat over TLS which exploits HTTP compression to launch an attack against SSL. Regardless a cryptography or hash algorithms used in TLS, BREACH always threatens HTTPS which is underlying TLS. As mentioned before HTTP is mandatory for web servers to reduce bandwidth and transmission time. Briefly, BREACH is able to extract sensitive information in no time depending on number of transmitted bits. Beyond that, many attacks discovered to exploit SSL such as CRIME, TIME, BEAST, and LUCKY13. These attacks exploit different weaknesses in SSL/TLS protocol which will be discussed in details in our paper. However, our research will focus on BREACH attack.

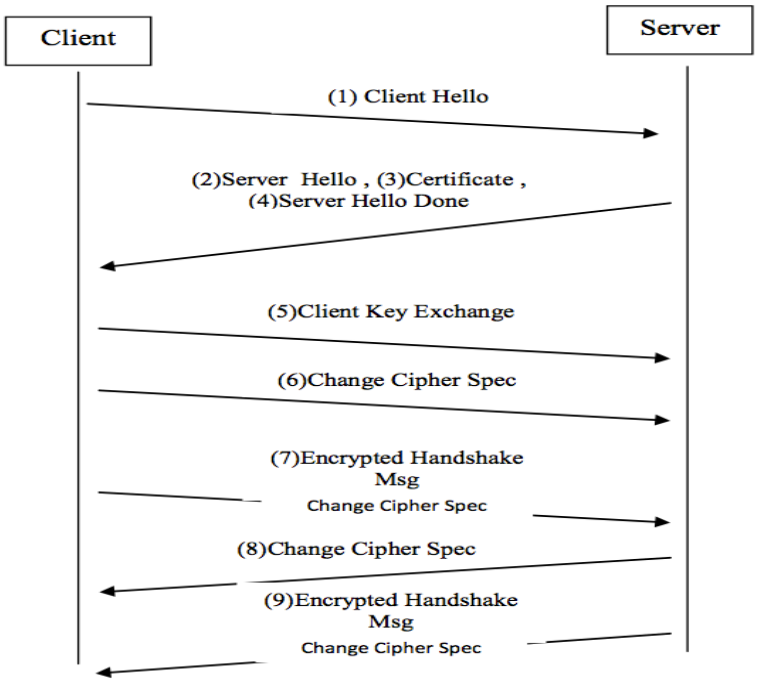


Fig 1: SSL message low diagram

**Motivations:**

HTTPS is a commonly used internet protocol and is relied upon to provide security for email messages, online banking, and other services that involve sensitive data transfers across computer networks. As the BREACH attack is the most recently disclosed vulnerability in secure internet communication we will present detailed analysis of the BREACH attack and possible defensive measures that server administrators may take. Given our limited resources, if it is possible we also hope to provide a demonstration of the breach attack against a client web browser.

**Alternatives:**

Perform a comprehensive study of previous attacks such as LUCKY13, BEAST, TIME, and CRIME.
Finding open source code or tools for mentioned attacks, then, it can be used as a base to develop BREACH attack tool.
Understand BREACH in details in order to implement it.
Understand Java Applet and OpenSSL library for exploit demonstration purposes.

**Questions:**

What are the available tools that perform attacks against SSL/TLS?
How do we detect Huffman encoding?
How GZIP and DEFLATE algorithms work?
What is the level of complexity in accomplishing BREACH attack?
Since the creators of BREACH built it with JAVA applet, is it possible to use other programming language to build this attack?

**Table of Contents:**

4. Implementation or experiment
    - Stating all information about programming language, development environment, SSL or TLS version, and web resources (server and clients).
    - Graphics to explain all implementation phases.
    - The final product and how it works.
5. Results
    - Testing results.
    - Demo if implementation was successfully built.
    - Detailed discussion of our results and BREACH creators' results.
    - The countermeasures.

6. Conclusion
    - How BREACH is a risk at all SSL versions.
    - Possible techniques to mitigate BREACH impact

7. References
    - References list for all sources information of our research
    - IEEE citation style

**Time Schedule:**

**Oct. 15-17**:
- Complete all research in background information and related study.
- Submit the results with the progress report to professor.

**Nov. 05-07:**
- Detail techniques about BREACH attack and steps to rebuild an BREACH attack
- Develop flow charts or pseudo code.

**Nov. 19-21:**
- Setup a BREACH attack and demonstrate the attack in real time (First target)
- Obstacles and difficulties while developing and setting up a BREACH attack (reasons for not successfully launch a BREACH attack)
- Alternative Solution: A comprehensive study how BREACH attack was implemented and launched by original creators.

**Tuesday December 3, 2013** – Oral Conference Style Presentation

**Saturday December 7, 2013** – Written Report IEEE Style

**References:**

[1] XMLHttpRequest, available at https://developer.mozilla.org/en-US/docs/Web/API/XMLHttpRequest
[2] N AlFardan, K Paterson, "Lucky Thirteen: Breaking the TLS and DTLS Record Protocols",Feb 2013 available at http://www.isg.rhul.ac.uk/tls/TLStiming.pdf

[3] N. AlFardan, D. Bernstein, K. Paterson, B. Poettering, J. Schuldt, "On the Security of RC4 in TLS and WPA", Jul 2013, available at http://www.isg.rhul.ac.uk/tls/RC4biases.pdf
[4] Radziszowski, "CRIME Attack on SSL/TSL", May 2013, available at http://www.cs.rit.edu/~sxj4236/crypto2_paper2.pdf
[5] I. Ristic, "Defending against the BREACH Attack", available at https://community.qualys.com/blogs/securitylabs/2013/08/07/defending-against-the-breach-attack.
[6] J. Kelsey, "Compression and Information Leakage of Plaintext", Feb 2002, available at http://www.iacr.org/cryptodb/archive/2002/FSE/3091/3091.pdf.
[7] Sarkar, Pratik, Fitzgerald, Shawn, “Attacks on SSL A Comprehensive Study of BEAST, CRIME, TIME, BREACH, LUCKY13 & RC4 Biases”
[8] J. Rizzo, and T. Duong, "Here Come The Ninjas", Ekoparty Security Conference.