SSLv3 Poodle Vulnerability

Chirag Manwani, 2018MCS2018

1 SSLv3 Introduction

Secure Sockets Layer(SSL), which is now deprecated, and its successor Transport Layer Security (TLS), are cryptographic protocols designed to provide communications security over a computer network.

Several versions of these protocols find widespread use in applications such as web browsing, email, instant messaging, and voice over IP (VoIP). SSLv3 was deprecated in 2014 after the Poodle vulnerability was found and since then TLS has been the standard to encrypt Transport Layer messages.

2 Working of SSL

The SSL protocol allows the server and client to

- Authenticate each other
- Negotiate an encryption and MAC algorithm
- Agree cryptographic keys to be used to protect payload data

Before the actual transmission of data, the Handshake protocol is followed. All the above mentioned actions are performed in the Handshake between the server and the client.

The SSL protocol establishes what is called a session. A session defines the set of cryptographic security parameters to be used. Multiple secure connections between client and server can share the same session reduces computation cost.

There are a total of four phases in the Handshake protocol, each is explained in the following subsections. The following page shows all the phases of the Handshake protocol in a diagram, which depicts the initial interaction between the client and the server.

- **Phase 1.** Establish Security Capabilities
- **Phase 2.** Server authentication and key exchange
- **Phase 3.** Client authentication and key exchange
- **Phase 4.** Finish

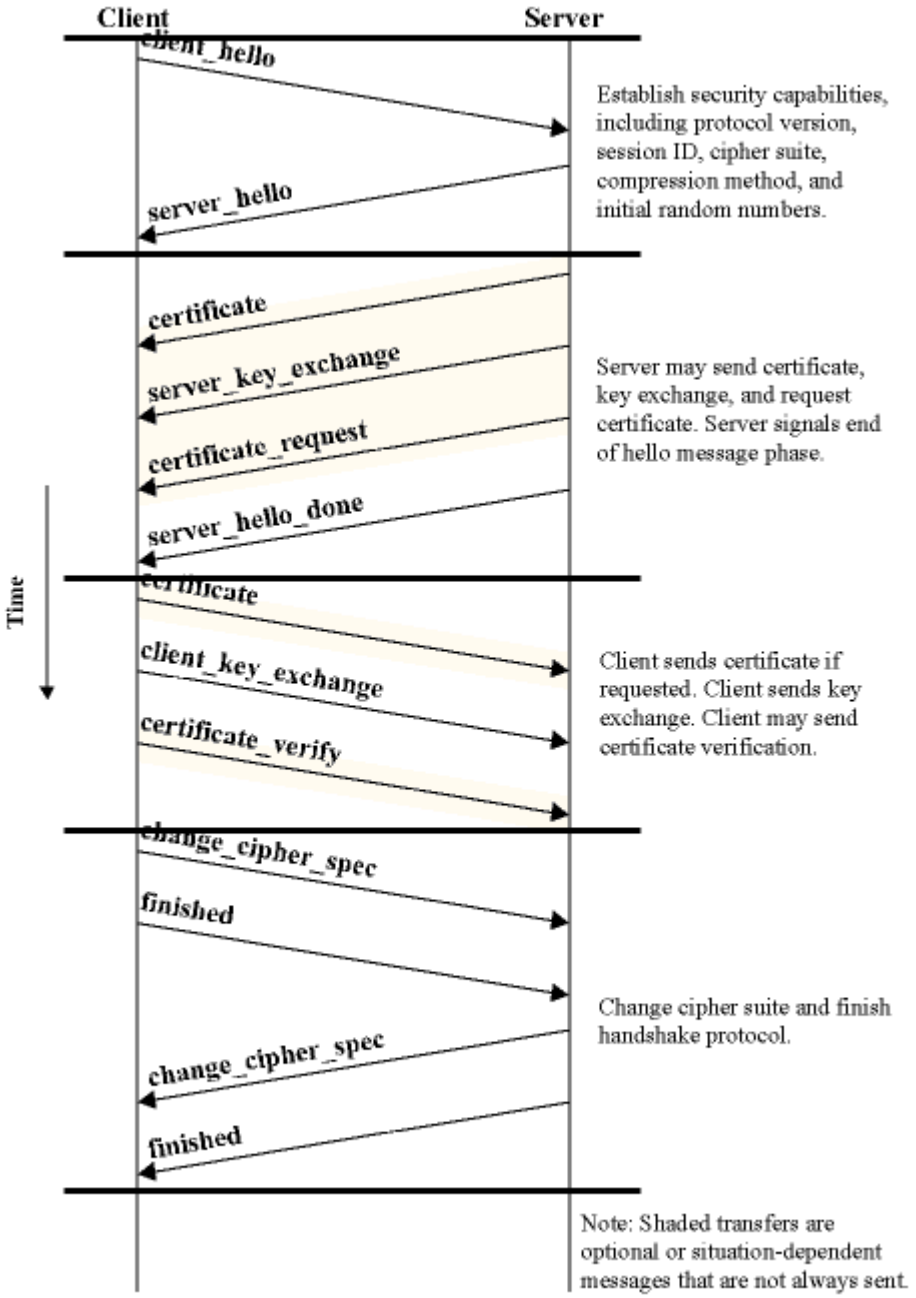

Figure 14.6 Handshake Protocol Action

Figure 1: Handshake Protocol

2.1 Phase-1

The client initiates the connection by sending a Client_hello message which contains the combinations of cryptographic algorithms supported by the client, in decreasing order of preference. The cipher algorithms supported are DES, 3DES, IDEA, etc. The MAC algorithms are based on a hash function like MD5, SHA-1 etc.
The server then replies with a Server_hello message which contains the selection by the server. The client_hello can contain a session_id to resume a previous session. Both messages have nonces (used later to generate master secret).

They also decide on the method to be used for Key exchange. The supported methods could be Diffe Hellman and its variations and RSA and its variations.

2.2 Phase-2

In phase-2 the Server sends certificate and the key exchange details to the client. These depend on the actual methods decided to perform the key exchange.

Server may send certificate_request which is usually not sent, since SSL usually used to authenticate client only. This phase ends when the Server sends the Server_hello_done message to the client.

2.3 Phase-3

In Phase-3 the client and server perform the actual key exchange along with verification of the client if required.

The client sends a Client_key_exchange message which contains the details of the key to be used and the contents depend on the actual algorithm.

2.4 Phase-4

In this phase all the details of done in the handshake protocol are confirmed by both the client and the server, and finally establish a secure session to transmit data between them.

3 The Poodle Vulnerability

The Poodle attack is a type of Man in the Middle attack, where the attacker is able to access the channel of communication being used by the server and client.

It depends on the the fact that the attacker has complete access to the messages being transmitted between the server and the client, i.e. the connection channel is not safe. The attacker can modify the message being sent from either the client side or server side, or even prevent messages from being received by the other end.

The Poodle attack begins with downgrading the security protocol to be used between the client and server from TLS to SSLv3. The most severe problem of CBC encryption in SSL 3.0 is that its block cipher padding is not deterministic, and not covered by the MAC (Message Authentication Code): thus, the integrity of padding cannot be fully verified when decrypting.

The weakness is the easiest to exploit if there's an entire block of padding, which (before encryption) consists of L-1 arbitrary bytes followed by a single byte of value L-1.

To process an incoming ciphertext record C1 ... Cn also given an initialization vector C0 (where each Ci is one block), the recipient first determines P1... Pn as

$$P_i = D_K(C_i) \oplus C_{i-1}$$

(where DK denotes blockcipher decryption using perconnection key K)
then checks and removes the padding at the end, and finally checks and removes a MAC.

If there's a full block of padding and an attacker replaces Cn by any earlier ciphertext block Ci from the same encrypted stream, the ciphertext will still be accepted if $D_K(C_i) \oplus C_{n-1}$ happens to have L-1 as its final byte, but will in all likelihood be rejected otherwise, giving rise to a padding oracle attack.

This final byte of the ith block can then be decrypted using the following-
Assume that the padding value in the last block is, 15 i.e. there is a full block of padding. The following condition holds-

$$15 = D(C_i) \oplus C_{n-1}[15]$$

We know from encryption equation that

$$C_n = E(P_n \oplus C_{n-1})$$

which gives

$$C_n = E(P_n) \oplus E(C_{n-1})$$

replace Ci by this in above equation we get.

$$15 = D(E(P_i) \oplus E(C_{i-1})) \oplus Cn - 1$$

simplifying this we get

$$15 = P_i \oplus C_{i-1} \oplus C_{n-1}$$

XOR both side by $15 \oplus P_i$, we get

$$P_i[15] = C_{i-1} \oplus C_{n-1} \oplus 15$$

This way the Poodle attach finds the last byte of the ith block.

4 Implementation steps

The following steps cover the implementation of the Poodle attack.

1. The attacker Downgrades the level of security protocol between the client and server by intercepting messages between them.
2. Once the security is downgraded to SSLv3, the attacker keeps modifying the last block of the SSL encrypted message.
    - If the server returns error which happens if the MAC does not match due to wrong padding, intercept the message and show could not connect message to the client.
    - If the server establishes a session, then the padding happened to be correct, then send a positive reply to the client.
3. Once the block C_i is known, the math explained in the previous section is used to get the last byte of this block.
4. These steps are repeated multiple times and by shifting the message the attacker can read the message byte by byte ultimately getting their hands on the actual message.