Report
COL759 (Cryptograpgy & Computer Security)

Vulnerabilities in standard cryptographic protocols

Sanket Sanjaypant Hire
2016CS50402

August 2020

1 SSL Protocol

SSL (Secure Sockets Layer) is an encryption-based Internet Security Protocol. It was developed to ensure **privacy**, **authentication**, and **data integrity** in Internet communications as follows:

- **Privacy**
  - In order to provide a high degree of privacy, SSL encrypts data that is transmitted across the web. This means that anyone who tries to intercept this data will only see a garbled mix of characters (encrypted data to be precise) that's nearly impossible to decrypt.
- **Authentication**
  - SSL initiates an authentication process called a handshake between two communicating devices to ensure that both devices are really who they claim to be.
- **Data Integrity**
  - SSL also digitally signs data in order to provide data integrity, verifying that the data is not tampered with before reaching its intended recipient.

2 SSL Protocol v3.0

SSL v3.0 is an obsolete and insecure protocol. Encryption in SSL v3.0 uses either the RC4 stream cipher or a block cipher in CBC mode. In this assignment, I have used the AES block cipher in CBC mode.

Basically, AES is an encryption and is meant to maintain confidentiality. However, Encryption does not maintain the integrity by itself. An attacker who can access the encrypted data can modify the bytes, thereby impacting the plaintext data (though the encryption makes the task a bit harder for the attacker, it is not as infeasible as is often assumed.)

So, to get the integrity, we need a MAC which is a short piece of information to authenticate a message; it confirms that the message came from the stated sender (it's authenticity) and has not been changed by an attacker. The MAC value protects both a message's data integrity as well as

its authenticity, by allowing verifiers (who also possess the secret key) to detect any changes to the message count. HMAC (Hash-based Message Authentication Code) is a nice MAC algorithm to use with AES-based encryption.

2.1 Encryption in SSL v3.0

Message encryption is carried out by sender as follows:

- append MAC (computed MAC of the plaintext) to plaintext and then encrypt the combined (MAC and plaintext) text using AES encryption in CBC mode.
- For using encryption of AES in CBC mode, we have to add byte padding to the (plaintext + MAC), so as the combined length is an integral multiple of the length of cipher blocks (in this case, an integral multiple of 16 bytes.)
- This encryption method is commonly known as MAC-then-pad-then-Encrypt method.

2.2 Decryption in SSL v3.0

Message decryption is carried out by receiver as follows:

- Firstly, the ciphertext is received by the receiver.
- AES decryption is carried on ciphertext after removing the padding blocks using the size of padding stored in the final byte of the received ciphertext.
- This deciphered text is then broken up into the decrypted plaintext and MAC (say extracted MAC.)
- To authenticate and verify the integrity of the message, we have to compute the MAC of the decrypted plaintext and check if it is equal to the extracted MAC. If found equal then the receiver has received the right message from the sender, otherwise the message is tampered by the middle man.

3 SSL v3.0 Vulnerability: POODLE

The POODLE attack (Padding Oracle on Downgraded Legacy Encryption) exploits a vulnerability in the SSL 3.0 protocol. The padding oracle is a situation when the attacker knows or can guess why the data that they sent to the server is rejected: whether it is because the padding was incorrect or whether the MAC was wrong.

Padding oracles are used for other attacks, too. Some protocols don't respond directly but may, for example, first check the padding and only later check the MAC. In those cases, if the attacker gets a quick response, it's a padding error, but if the response takes a bit longer, it's a MAC error.

This vulnerability lets an attacker eavesdrop on communication encrypted using SSLv3. The vulnerability is no longer present in the Transport Layer Security protocol (TLS), which is the successor to SSL (Secure Socket Layer).

The POODLE vulnerability lets the attacker eavesdrop on encrypted communication. This means that the attacker can steal confidential data that is transmitted, for example, passwords or session cookies, and then impersonate the user. This can have very serious consequences, including losing control over the web application (for example, if the attacker impersonates an admin).

The attack is not very easy because it needs to be successful in three stages:

- In the first stage, the attacker must perform a successful man-in-the-middle attack (MITM). The attacker can now listen to all communication between the client and the server as well as add to this communication (impersonate the client or the server). However, if this is a secure connection, communication is encrypted using SSL/TLS, so the attacker cannot understand what is being sent.
- In the second stage, the attacker must convince the server to use the old SSL 3.0 protocol. The attacker can do this by dropping connections – after a number of such drop-outs, the server will try an older protocol, thinking that the client cannot use a newer protocol such as TLS 1.2. This is called a protocol downgrade attack or downgrade dance.
- In the third stage, when the client and the server are communicating using SSL 3.0, the attacker can use the POODLE attack to decrypt selected parts of the communication and steal confidential information.
- To make sure that the POODLE attack succeeds, the attacker should also be able to trick the user browser into running JavaScript, for example, using social engineering.

4 Implementing the POODLE Attack

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.

- Secret Message after encrypting using AES Key and IV, is then converted into series of cipher blocks (say $C_1, C_2, ..., C_n$)
- If Secret Message and MAC are integral multiple of block length, then adding full padding block enables the attacker to exploit the last byte of the full padding block i.e. 15.
- Exploiting the plaintext byte:
  - Since the decryption process has to start with the last block of data we are encrypting the message using the AES key, we replace the last block with message block containing the last character of actual secret message.
  - Iterating through the attack, in each new decryption of message most of the times our extracted MAC do not match with the computed MAC(256 times to be precise). i.e. Once in 256 times we get correct MAC match (which lets us know that the last byte is full padding length = 15) enabling the attacker to know that last byte of secret message is decrypted to 15. This knowledge helps the attacker to find the plaintext byte using the following CBC formula:

$$P_2[15] = C_{n-1}[15] \oplus C_1[15] \oplus 15$$

  - Rotate the plaintext by one character at each successful decryption, so that we can extract all the characters of plaintext one-by-one by accessing the last byte of cipher block.

5 Impacts of POODLE Attack

The POODLE attack can be used against any system or application that supports SSL 3.0 with CBC mode ciphers. This affects most current browsers and websites, but also includes any software that either references a vulnerable SSL/TLS library (e.g. OpenSSL) or implements the SSL/TLS protocol suite itself. By exploiting this vulnerability in a likely web-based scenario, an attacker can gain access to sensitive data passed within the encrypted web session, such as passwords, cookies and other authentication tokens that can then be used to gain more complete access to a website (impersonating that user, accessing database content, etc.)

6 Measures to safeguard against the POODLE Attack

- If you support the most recent TLS version because an active MITM attacker can force browsers to downgrade their connections all the way down to SSL 3, which can then be exploited.
- To prevent downgrade attack, both the servers and clients should support TLS_FALLBACK_SCSV.
- As a user, you want to protect yourself from attacks, and the best way to do that is to disable SSL 3 in your browser.
- As a web site operator, you should disable SSL 3 on your servers as soon as possible.

7 POODLE Simulation for the Assignment

- For using AES algorithm and computing MAC using HMAC algorithm, I have used [pycryptodome] python library.
- The code is only a proof of concept for poodle and not contain the complete server-client-attacker SSL handshakes through the ports.
- short overview of the implemented code (for getting more insight visit code comments):
  - Create a secret message for sender to send.
  - Generate a common cryptographic key for sender and receiver for encryption and decryption respectively.
  - generate IV(Initialization vector) and MAC(using HMAC and common cryptographic key) for encryption from the sender side and perform AES encryption in CBC mode.
  - then create a decryption function which compares the extracted MAC with computed MAC and verifies the decipherd byte.
  - after each successful decryption, extract the deciphered byte and append to the string which we will be returning finally as a decrypted message.
  - Rotate the plaintext by one character at each successful decryption, so that we can extract all the characters of plaintext one-by-one by accessing the last byte of cipher block.

8 Running the POODLE Attack Simulation

8.1 Folder Structure

- Report.pdf
- README.md
- simulate_poodle.py

8.2 How to run the code

- Install the cryptography library
  - **pip3 install pycryptodome**
- Then simply run the command:
  - **python3 simulate_poodle.py**

  [**Note:** Since this program decodes 32nd character from the deciphered text, the poodle attack only works for the secret message greater that 32 characters.]
- Sample POODLE Attack Simulation: