Write a letter

HW7 - CNS Sapienza

Pietro Spadaccino 1706250

14 December 2018

1 Introduction

Using some scanning tools, we have found some websites having weaknesses for TLS protocol. We will show them and report their vulnerabilities.

2 TLS

TLS is a protocol that enables secure comunication between two parties, usually a server and a client. It ensures privacy, by the use of symmetric encryption, identity of the parties, through public keys, and data integrity, using hashes and MACs. TLS was designed also to be compatible with changes and updates that may come in the future. As example we have the TLS handshaking phase, where the two parties agree on the cipher suite to use, comprehending the encryption algorithm, the hashing method and the key exchange protocol. As new algorithms are proposed and some old algorithms get deprecated, like DES, the protocol can be expanded without any major updates or patching.

Successful attacks on TLS, may compromise the security of the communication, both privacy and integrity, resulting in a possibility for a malicious third party to perform a man-in-the-middle attack.

3 rai.it

The grade assigned by *Qualys* is F. Even if the score is the least possible, there aren't many weaknesses except for one major one: vulnerability to POODLE attack. This attack exploits some SSL implementations, usually version 3.0. The weaknesses is indeed related to some specific implementations, not in the protocol itself, even though more than 10% servers are vulnerable to this attack, according to SSL pulse. These weak implementations fail to check the padding correctness, as specified in the RFC, and the attacker is able to disclose some data going from a client to the server. The attack will require about 256 requests to disclose a single byte of the encrypted messages, and this number scales linearly, so that after around 4096 request 16 bytes can be disclosed. A quick fix could be disabling completely SSL 3.0 on the servers and use an implementation of TLS_FALLBACK_SCSV which will handle and make impossible downgrade attacks.

The website does not support forward secrecy with some browsers. Forward secrecy enables secure conversations without being conditioned on the server's private key. What does it mean is that, if we don't use forward secrecy, a third party having access to the server and able to retrieve its private key, can decrypt and understand all past communications. In other words, by using forward secrecy, the disclosure of a long-term key does not impact on current session keys. The experiments show that there was just one browser/OS that did not support forward secrecy, and it is Internet Explorer on Windows XP, which is very outdated and should not be in use anymore. An easy fix should be endorsing the usage of ECDHE instead of plain DHE, this will enable forward secrecy in today's browsers. Older browser may not support ECDHE, and they will stick to use RSA. Since RSA does not provide forward secrecy by design, it should be used only when it is strictly necessary.

The last misconfiguration found on the server, is the support for older and weaker cipher suites. The TLS cipher suites are a list of combinations of encryption algorithm and hashing algorithm, and it is agreed between the client and the server during the handshaking phase. *Qualys* tools throws a warning for the supported protocols, because authenticated encryption with associated data (AEAD) is not supported. CBC only is supported, and it is susceptible to timing attacks. A fix could be enabling the support for AEAD cipher suites, upgrading the own TLS implementation if necessary.

4 web.uniroma2.it

The grade set by *Qualys* is F. This is due to a variety of misconfigurations that leads to weaknesses and insecurities. Like the website rai.it, it does support some cipher suites that are insecure, but the penalty inflicted by *Qualys* was greater than the one of the previous website, because here we have some suites that are remarkably obsolete and there is no reason to support them. As an example, some enabled cipher suites includes DES for encryption and MD5 for hashing, which are far from secure and even the elder clients support newer algorithms. A fix is indeed immediately disabling these suites.

In the previous cipher suites marked as insecure, there were some particular ones that can cause major problems, being vulnerable to Logjam attack. An attacker can indeed downgrade the communication and can arbitrarily read and modify data over the connection, giving the primitive for a man-in-the-middle attack. This vulnerability is a flaw of the TLS protocol and does not depend on the implementation, so the only possible fix is disabling these suites to be chosen during the handshake.

Always in the cipher suites, we find that some of them are RSA_EXPORT, possibly exposing the HTTPS communication to a man-in-the-middle attack. This attack is known as FREAK attack, and it is not a protocol weakness but it is due to implementations errors. While a quick fix can be just upgrading the TLS implementation, it should be considered, if not necessary, to disable also these cipher suites since they rely on outdated algorithms.

Another major flaw is that the website is vulnerable to OpenSSL Padding Oracle, reported in CVE-2016-2107. This vulnerability makes a malicious third party able to disclose the last byte of any block and, sometimes, also the entire message. This weakness is due to an implementation error, introduced first by the code used to patch another vulnerability CVE-2013-0169. All its details are reported in [1]. A quick fix is upgrading the TLS implementation.

References

[1] CloudFlare, *blog.cloudflare.com/yet-another-padding-oracle-in-openssl-cbc-ciphersuites/*.