On Compression Ratio Info-leak Mass Exploitation (CRIME) Attack and Countermeasures

Sanduni Prasadi (sanduni.prasadi06@gmail.com)[1], Jayamine Alupotha[1], Mohamed Fawzan[1], Janaka Alawatugoda[1], Roshan Ragel[1]
[1] Department of Computer Engineering, University of Peradeniya, Sri Lanka.

Abstract

Header compression is desirable for network applications, as it saves bandwidth. However, when data is compressed before being encrypted, the amount of compression leaks information about the amount of redundancy in the plaintext. This leads to the CRIME attack on web traffic protected by the SSL/TLS protocols. In order to mitigate the CRIME attack, compression is completely disabled in the TLS/SSL-layer. Although disabling compression completely mitigates the CRIME attack, it has a drastic impact on bandwidth usage.

The attack is carried out with the assumption that the attacker has the ability to view the victim's encrypted traffic. An attacker can accomplish this with a network protocol analyzer. It is also assumed that the attacker has the ability to make the victim client to send requests to the targeted web server. This can be accomplished by coercing the victim to visit an attacker-controlled site (which contains a JavaScript code that sends requests to the targeted server with attacker-injected values in request headers). The attacker will coerce the victim to send a small number of requests to guess the first byte of the secret cookie. The attacker then measures the size of the (compressed) request headers. With that information, the CRIME attack algorithm determines the correct value for the first character of the secret cookie. Since the attack relies on LZ77 loss-less data compression algorithm, the first byte of the target secret must be correctly guessed before the second byte is attempted.

Separating secret cookies from compression is presented as a proven-secure countermeasure against CRIME attack in a previous work: (1)--separates all the secret cookies from the request header. (2)--rest of the header is compressed, while the secrets are kept uncompressed. Since the secret cookie is not compressed with the attacker-injected values, the origin of the compression leakage is shut. Thus, the proposed solution completely prevents the CRIME attack and also enables header compression. This is useful in reduction of network bandwidth usage.

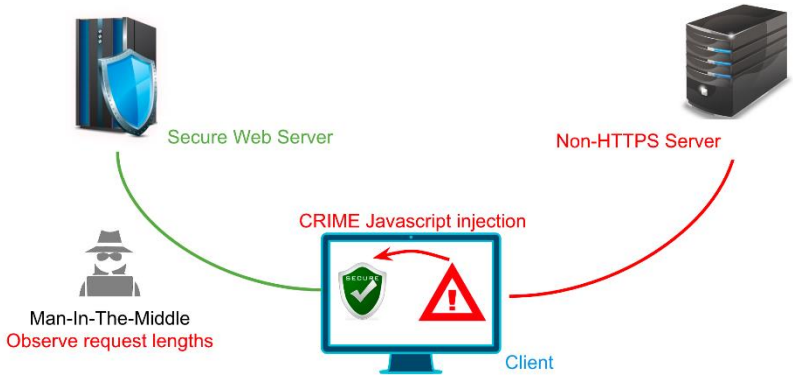

Figure 1 CRIME attack setup

In this work we create a test environment to replicate the CRIME attack and to test countermeasures.

Keywords: *CRIME attack, SSL/TLS, Security cryptography*