**Abstract**

The Browser Reconnaissance and Exfiltration via Adaptive Compression of Hypertext (BREACH) attack is a compression side channel attack, which targets information compressed in HTTP response bodies through HTTP compression.

HTTP compression is the process of compressing the content in HTTP responses on server side before sending to the client. HTTP compression is normally performed through the Deflate algorithm.

Deflate is a data compression algorithm that is a combination of LZ77 algorithm and Huffman coding. When compressing data using this algorithm, any repeated byte sequences in the input are detected and are not repeated in the output. Instead, the repeated byte sequence is stored only once, along with pointers that point out where the same sequence is found again.

Main thing that make BREACH attack possible is adaptive dictionary compression algorithm which is deflate in most cases and ability to find the length of a HTTP message body. After compressing with deflate algorithm even when encrypted, the length of the compressed data is still visible and this is another fundamental element that make the BREACH attack possible.

What really happens in this attack is attackers inject his guesses for secrets in to HTTP response bodies. Because of adaptive dictionary compression if the guessed secret is matched to the actual secret response will be highly compressed than a response which is injected an incorrect guess. Length of HTTP response is used as a measurement of compression.

In Black Hat Europe 2013 conference a group of researchers have demonstrated practically how to launch a BREACH attack and proposed couple of countermeasures. The proposed countermeasures are;

1. Disabling HTTP compression
2. Separating secrets from user input
3. Randomizing secrets per request

Using a non-adaptive (fixed) dictionary compression algorithm to prevent BREACH attack was proposed by Alawathugoda, Stebila and Boyd in 2015. They have proved the security of Fixed-Dictionary compression against the BREACH attack. In this work our aim is to deploy the Fixed-Dictionary compression algorithm into the real world Client-Server setting.