Abstract No: 253 *IT, Mathematics and Statistics*

Implementing a client-server setting to prevent the browser reconnaissance and exfiltration via adaptive compression of hypertext attacks

I. Weerasooriya, D. Jayawardhana, N. Amarasinghe, J. Alawatugoda* and R. Ragel

Department of Computer Engineering, Faculty of Engineering, University of Peradeniya, Sri Lanka
**janaka@ce.pdn.ac.lk*

The Browser Reconnaissance and Exfiltration via Adaptive Compression of Hypertext (BREACH) attack is a compression-based side-channel attack, which targets sensitive pieces of data compressed-then-encrypted in the HTTP responses. The BREACH attack was firstly demonstrated in Black Hat Europe 2013.

The HTTP compression is the process of compressing the content in the HTTP responses from the server-side, before sending them to the client. The HTTP compression is normally performed through the DEFLATE algorithm, which is a combination of the LZ77 algorithm and Huffman coding.

The main reason that makes the BREACH attack possible is that the adaptive compression-dictionary used in the DEFLATE algorithm, which enables the algorithm to develop a compression-dictionary based on the content to be compressed. After compressing with the DEFLATE algorithm, even when encrypted, the length of the compressed data is still visible. In the BREACH attack, the attacker injects his guesses of the secrets into the HTTP response bodies. Due to the adaptive compression-dictionary, if the guessed bytes match with the actual secrets, responses would be highly compressed and hence the output length differs. As the length of the responses would reveal information on how much overlap has happened, the attacker can measure how much of the attacker-injected bytes are contained in the sensitive pieces of data in the system.

The BREACH attack can be mitigated by using a non-adaptive fixed dictionary for compression, because the dictionary is independent from the inputs, and hence the attacker-injected guesses cannot affect the dictionary; the data will be compressed if they match with the dictionary entries, otherwise not. This idea was first proposed with security proofs in a reasonable model in Financial Cryptography and Data Security 2015.

In this research we implemented and deployed a non-adaptive fixed-dictionary compression algorithm into the real-world client-server setting, and facilitate a realistic mechanism to prevent the BREACH attack. Further, we verified the correctness of data recovery in the client-side.

This project is supported by the National Research Council (Grant NRC 16-020).