Network Working Group
Internet-Draft
Intended status: BCP
Expires: April 25, 2013

B. Kihara
K. Shimizu
Lepidum Co. Ltd.

October 22, 2012

Considerations for Protocols with Compression over TLS
draft-kihara-compression-considered-harmful-01

Abstract

Transport Layer Security is essential to secret communications in the Internet, and protecting TLS is our fundamental task. This document describes a threat to TLS when compression is used and proposes possible mitigations of the threat.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on April 25, 2013.

Table of Contents

1. Introduction 3
2. Conditions of Attacks 3
3. Application to Existing Protocols 3
3.1. Attacks on HTTP 3
3.2. More Applications 4
4. Possible Mitigations 4
4.1. Abandon of Compression 4
4.2. Compressing Non-Sensitive Data Only 4
4.3. Using Static Dictionary for Compression 4
4.4. Inserting Random Paddings 4
4.5. Detecting and Blocking Attacks 5
4.6. More Mitigations 5
5. Security Considerations 5
6. IANA Considerations 5
7. Informative References 5
Authors' Addresses 6

1. Introduction

Transport Layer Security [RFC5246] is, needless to say, essential to contemporary customs in the Internet like secret communications on the web. Names, passwords, telephone numbers, credit card numbers, messages, session ids, and any other secrets are transported on TLS. Therefore, protecting TLS and underlying Public Key Infrastructure is our fundamental task. Unfortunately, TLS and PKI are not infallible, and sometimes vulnerable to new attack vectors. This document describes conditions for the CRIME attack [CRIME] which utilizes differences of compression rates to guess secret information from outside the encrypted transports and proposes possible mitigations of the threat that should be considered when someone designs protocols with compression.

2. Conditions of Attacks

"Commonly-used lossless compression algorithms leak information about the data being compressed, in the size of the compressor output." [IACR-fse-2002-3091] In other words, by nature compressing data has the risk of information leakage. Usually information from the size of compressed data is not enough to guess the secret easily; however, under some conditions the attack becomes much easier.

Observing encrypted data itself is not so useful. Though, if the attacker can inject string into the compression context where the secret is compressed, the attacker would be able to know whether the injected string matches the secret or not. In addition, difference of the size of the compressed data in multiple tries will help the attacker to perform brute-force attacks.

[[[More precise conditions are needed. For example, initalizing compression contexts on each try will be a great help.]]] [[[Note that ANY KIND OF COMPRESSION WILL REVEAL SECRETS REGARDLESS OF THE LAYER OF COMPRESSION, including TLS compression, SPDY, and HTTP gzip.]]]

3. Application to Existing Protocols

3.1. Attacks on HTTP

The original CRIME is targeted at HTTP [RFC2616]. Web browsers tend to make requests according to malicious scripts, sending secret strings automatically together with injected strings by the scripts. To make matters worse, TLS compression [RFC3749] and SPDY [I-D.mbelshe-httpbis-spdy] had been compressing all data in the same

context. Therefore, stealing session cookies by the CRIME attack was very easy. As a workaround, some web browsers disabled TLS compression and SPDY compression partially so that session cookies and bearer-token-type Authorization tokens cannot be stolen. However, compression of HTTP entities is still available and it is possible to guess some protions of HTTP entities if servers return injected strings in HTTP entities.

3.2. More Applications

[[[More applications are needed? Currently we have not found such applications.]]]

4. Possible Mitigations

4.1. Abandon of Compression

In principle, abandon of compression completely solves the problem of information leakage by compression. It is RECOMMENDED to disable compression when communications are not trivial, unless traffic increase is considerable. If data are confidentital and other mitigations are inapplicable, all kinds of compression MUST be disabled.

4.2. Compressing Non-Sensitive Data Only

The problem that this document describes is information leakage by compression. However, if transferred data are not sensitive, we do not have to take care of the problem. Therefore compressing non-sensitive data will save bandwidth without exposing sensitive data. Note that dynamically-generated contents can contain sensitive data and SHOULD NOT be compressed.

4.3. Using Static Dictionary for Compression

The CRIME attack utilizes differences of compression rate to estimate that the candidate string matches the sensitive data or not. In order to prevent such attacks, using static dictionary will be effective, especially when compressing patterned contents like HTTP headers.

4.4. Inserting Random Paddings

If it is unavoidable to compress whole data in the same context, inserting random paddings will be available to prevent disclosure of the original size of compressed data. Note that this mitigation cannot prevent attackers from guessing secrets by statistical

approaches.

[[[Requirements of padding (ranges of length, randomness, and so on) and quantitative evaluations are needed.]]]

4.5. Detecting and Blocking Attacks

To achieve attacks on compression, attackers have to make multiple traffics in order to observe differences of compresion rate. Therefore, detecting too frequent requests and blocking such requests will mitigate attacks. Note that this mitigation cannot prevent attacks completely and SHOULD be used with other mitigations.

4.6. More Mitigations

[[[There should be more mitigations.]]]

5. Security Considerations

This document focuses on security.

6. IANA Considerations

This document does not require actions by IANA.

7. Informative References

[CRIME] Rizzo, J. and T. Duong, "The CRIME Attack", , <https://docs.google.com/presentation/d/11eBmGiHbYcHR9gL5nDyZChu_-lCa2GizeuOfaLU2HOU/edit?pli=1#slide=id.g1d134dff_1_222>.

[I-D.mbelshe-httpbis-spdy] Belshe, M. and R. Peon, "SPDY Protocol", draft-mbelshe-httpbis-spdy-00 (work in progress), February 2012.

[IACR-fse-2002-3091] Kelsey, J., "Compression and Information Leakage of Plaintext", IACR fse-2002-3091, <http://www.iacr.org/cryptodb/archive/2002/FSE/3091/3091.pdf>.

[RFC2616] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999.

[RFC3749] Hollenbeck, S., "Transport Layer Security Protocol Compression Methods", RFC 3749, May 2004.

[RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, August 2008.

Authors' Addresses

Boku Kihara
Lepidum Co. Ltd.
#602, Village Sasazuka 3
1-30-3 Sasazuka
Shibuya-ku, Tokyo
JP

Email: kihara@lepidum.co.jp

Kazuki Shimizu
Lepidum Co. Ltd.