TCP Maintenance and Minor Extensions (tcpm)
Internet-Draft
Intended status: Standards Track
Expires: April 28, 2015

B. Briscoe
BT

October 25, 2014

The Echo Cookie TCP Option
draft-briscoe-tcpm-echo-cookie-00

Abstract

This document specifies a TCP Option called EchoCookie. It provides a single field that a TCP server can use to store opaque cookie data 'in flight' rather than in memory. As new TCP options are defined, they can require that implementations support the EchoCookie option. Then if a server's SYN queue is under pressure from a SYN flooding attack, it can ask clients to echo its connection state in their acknowledgement. This facility is similar to the classic SYN Cookie, but it provides enough space for connection state associated with TCP options. In contrast, the classic location for a SYN Cookie only provides enough space for a degraded encoding of the Maximum Segment Size (MSS) TCP option and no others.

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 28, 2015.

Table of Contents

1. Introduction

In order to initiate a connection, a TCP client sends a SYN segment to a TCP server. The server normally allocates memory to hold the required connection state then responds with a SYN/ACK segment to the address the client claims to be sending from. If a TCP server is under SYN flood attack, it can resort to including a SYN Cookie in the SYN/ACK [RFC4987] and not holding any connection state until the client follows through with an echo of the SYN Cookie. Therefore, a SYN Cookie effectively allows a TCP server to store its connection state 'in flight' for a round. Then while it is testing which client addresses correctly complete the handshake, it can protect its memory from exhaustion.

The limited size of a SYN Cookie is a known limitation. SYN Cookies are not standardised (and don't need to be), but typically the server encodes its SYN Cookie into the 16 bits of the Initial Sequence Number (ISN) [RFC0793] and the 9 least significant bits of the timestamp option [RFC7323] (if supported by the client). These fields are only large enough to hold a few common TCP options, such as a degraded record of the client's maximum segment size (MSS), the window scale option and SACK-ok. Therefore, SYN Cookies only protect a rudimentary TCP connection service--they do not protect all the facilities provided by TCP options during an attack.

These 41 bits are the only space available for SYN cookies. A server can only exploit fields that it can set to any value it chooses and that are naturally echoed by all (or at least most) TCP clients. Ideally, the server would be able to place a cookie of any reasonable size in a new generic EchoCookie TCP option on the SYN/ACK and the client would be required to echo it back in the following ACK. However, that would be of little use until most clients supported it.

A simple solution to this problem is to require that EchoCookie support must be implemented with any TCP options defined from now on. A new capability to extend the TCP option space on SYN/ACK segments, e.g. [I-D.touch-tcpm-tcp-syn-ext-opt] or [I-D.briscoe-tcpm-inner-space], could also require that the EchoCookie mechanism must be implemented with it.

1.1. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119]. These words only have such normative significance when in ALL CAPS, not when in lower case.

2. Echo Cookie TCP Option

If a TCP server's SYN queue is under pressure from a SYN flood attack, it MAY send an EchoCookie TCP option on the SYN/ACK, instead of consuming memory to hold connection state.

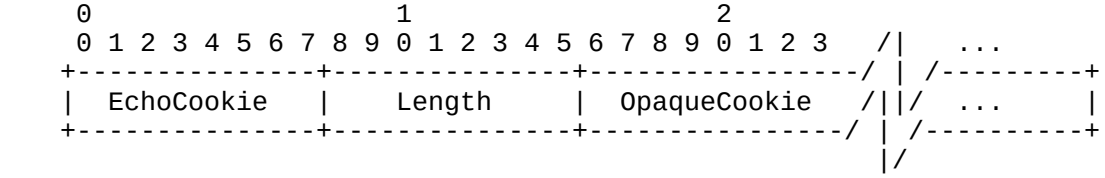

Figure 1: The EchoCookie TCP Option

The general structure of TCP options is defined in [RFC0793]. The EchoCookie TCP option is defined in Figure 1. The Option Kind is EchoCookie with value {ToDo: Value TBA}. The Length in octets can be any value greater than 1.

The OpaqueCookie field is available for the sender to fill with any amount of any type of data it wishes to store in the cookie, only constrained in size to an integer number of octets.

When a TCP receiver acknowledges a segment carrying an EchoCookie option, it MUST return an EchoCookie TCP option carrying an identical OpaqueCookie.

The mechanism a server uses to determine whether the echoed contents of the cookie are the same as the contents it sent are implementation dependent and do not need to be standardised.

The EchoCookie option with length greater than 2 is only defined on a SYN/ACK or on the ACK in response.

A client MAY send an empty EchoCookie TCP option with Length=2 on the SYN, to indicate that it supports the EchoCookie facility. This will not be necessary if support is implied by some other means (e.g. use of the Inner Space protocol [I-D.briscoe-tcpm-inner-space] implies support for EchoCookie).

If there is any TCP Payload in the SYN, it will never be necessary to include this data in a subsequent Echo Cookie. Not acknowledging the data would be sufficient to get the client to retransmit it.

If the client sends a valid TCP Fast Open (TFO) cookie [I-D.ietf-tcpm-fastopen] on the SYN of a resumed connection, there will be no need to defer establishing the connection by responding with an EchoCookie, because the client source address is already known to the server.

3. IANA Considerations

This specification requires IANA to allocate a value from the TCP Option Kind name-space against the name:

"EchoCookie"

Early implementation before the IANA allocation MUST follow [RFC6994] and use experimental option 254 and respective Experiment ID:

0xEEEE (16 bits);

{ToDo: Instead it might be prudent/possible for initial experiments to reuse Option Kinds 6 and/or 7 defined by RFC 1072 (Oct 1988) for a 4-octet Echo and Echo Reply facility that was superceded by the combined Echo and Reply facility in the Timestamp option of RFC1323 (May 1992) and formally obsoleted by RFC6247 (May 2011). Then if the experiments find that no legacy implementations recognise these options it can re-use them to avoid consuming new Option Kind values.}

{ToDo: Values TBA and register them with IANA} then migrate to the assigned option after allocation.}

4. Security Considerations

If the cookie holds state that was negotiated over a secure connection, it MUST be echoed with the same or a stronger level of security.

A SYN/ACK carrying an EchoCookie request MUST NOT exceed the size of the TCP SYN that preceded it. This ensures that the EchoCookie defence cannot amplify an attack by reflection.

A server may record a random selection of the clients to which it responds with an EchoCookie option. Then it can detect if a spoof client is mounting a reflection attack, by repeatedly asking the server to send a SYN/ACK to the same victim client that rarely or never responds. In such a case the server SHOULD limit the frequency at which it responds to such a client.

{ToDo: More?}

5. Acknowledgements

Bob Briscoe’s contribution is part-funded by the European Community under its Seventh Framework Programme through the Trilogy 2 project (ICT-317756). The views expressed here are solely those of the author.

6. References

6.1. Normative References

[I-D.ietf-tcpm-fastopen] Cheng, Y., Chu, J., Radhakrishnan, S., and A. Jain, "TCP Fast Open", draft-ietf-tcpm-fastopen-10 (work in progress), September 2014.

[RFC0793] Postel, J., "Transmission Control Protocol", STD 7, RFC 793, September 1981.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC6994] Touch, J., "Shared Use of Experimental TCP Options", RFC 6994, August 2013.

6.2. Informative References

[I-D.briscoe-tcpm-inner-space]
Briscoe, B., "Inner Space for TCP Options", draft-briscoe-tcpm-inner-space-00 (work in progress), October 2014.

[I-D.touch-tcpm-tcp-syn-ext-opt]
Touch, J. and T. Faber, "TCP SYN Extended Option Space Using an Out-of-Band Segment", draft-touch-tcpm-tcp-syn-ext-opt-01 (work in progress), September 2014.

[RFC4987] Eddy, W., "TCP SYN Flooding Attacks and Common Mitigations", RFC 4987, August 2007.

[RFC7323] Borman, D., Braden, B., Jacobson, V., and R. Scheffenegger, "TCP Extensions for High Performance", RFC 7323, September 2014.

Appendix A. Protocol Design Issues (to be Deleted before Publication)

This appendix is informative, not normative. It records outstanding issues with the protocol design that will need to be resolved before publication.

Why limit to SYN/ACK? {ToDo: Consider whether it is OK to generalise EchoCookie with Length > 2 to any segment from client or server (except the SYN, which would create a vulnerability to reflection attacks), especially the FIN, FIN/ACK etc.. It may even be possible to generalise this to cover TFO.}

Author's Address

Bob Briscoe
BT
B54/77, Adastral Park
Martlesham Heath
Ipswich IP5 3RE
UK

Phone: +44 1473 645196
Email: bob.briscoe@bt.com
URI: http://bobbriscoe.net/