DNSOP Working Group
Internet-Draft
Updates: 7873 (if approved)
Intended status: Standards Track
Expires: September 12, 2019

O. Sury
Internet Systems Consortium
W. Toorop
NLnet Labs

March 11, 2019

Algorithms for Domain Name System (DNS) Cookies construction
draft-sury-toorop-dns-cookies-algorithms-00

Abstract

[RFC7873] left the construction of Server Cookies to the discretion of the DNS Server (implementer) which has resulted in a gallimaufry of different implementations. As a result, DNS Cookies are impractical to deploy on multi-vendor anycast networks, because the Server Cookie constructed by one implementation cannot be validated by another.

This document provides precise directions for creating Server Cookies to address this issue. Furthermore, [FNV] is obsoleted as a suitable Hash function for calculating DNS Cookies. [SipHash-2.4] is introduced as a new REQUIRED Hash function for calculating DNS Cookies.

This document updates [RFC7873]

Status of This Memo

Table of Contents

1. Introduction

In [RFC7873] in Section 6 it is "RECOMMENDED for simplicity that the Same Server Secret be used by each DNS server in a set of anycast servers." However, how precisely a Server Cookie is calculated from this Server Secret, is left to the implementation.

This guidance has let to DNS Cookie implementations, calculating the Server Cookie in different ways. This causes problems with anycast deployments with DNS Software from multiple vendors, because even when all DNS Software would share the same secret, as RECOMMENDED in Section 6. of [RFC7873], they all produce different Server Cookies based on that secret and (at least) the Client Cookie and Client IP Address.

1.1. Contents of this document

In Section Section 2 instructions for constructing a Client Cookie are given

In Section Section 3 instructions for constructing a Server Cookie are given

In Section Section 4 the different hash functions usable for DNS Cookie construction are listed. [FNV] and HMAC-SHA-256-64 [RFC6234] are obsoleted and AES [RFC5649] and [SipHash-2.4] are introduced as a REQUIRED hash function for DNS Cookie implementations.

1.2. Definitions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "*NOT RECOMMENDED*", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

2. Constructing a Client Cookie

The Client Cookie is a nonce and should be treated as such. For simplicity, it can be calculated from Client IP Address, Server IP Address and a secret known only to the Client. The Client Cookie SHOULD have at least 64-bits of entropy. If a secure pseudorandom function (like SipHash24) is used there's no need to change Client secret periodically and change the Client secret only if it has been compromised.

It's recommended but not required that a pseudorandom function is used to construct the Client Cookie:

```
Client-Cookie = MAC_Algorithm(
    Client IP Address | Server IP Address, Client Secret )
```

where "|" indicates concatenation.

3. Constructing a Server Cookie

The Server Cookie is effectively message authentication code (MAC) and should be treated as such.

The Server Cookie is not required to be changed periodically if a secure pseudorandom function is used.

The 128-bit Server Cookie consists of Sub-Fields: a 1 octet Version Sub-Field, a 1 octet Cookie Algorithm Sub-Field, a 2 octet Reserved Sub-Field, a 4 octet Timestamp Sub-Field and a 8 octet Hash Sub-Field.

3.1. The Version Sub-Field

The Version Sub-Field prescribes the structure and Hash calculation formula. This document defines Version 1 to be the structure and way to calculate the Hash Sub-Field as defined in this Section.

3.2. The Cookie algo Sub-Field

The Cookie Algo value defines what algorithm function to use for calculating the Hash Sub-Field as described in Section 3.5. The values are described in Section 4.

3.3. The Reserved Sub-Field

The value of the Reserved Sub-Field is reserved for future versions of Server Side Cookie construction. Even though the value has no specific meaning in this Version, note that it *is* used in determining the Hash value as described in Section 3.5.

3.4. The Timestamp Sub-Field

The Timestamp value prevents Replay Attacks and MUST be checked by the server to be within a defined period of time. The DNS Server SHOULD allow Cookies within 1 hour period in the past and 5 minutes into the future to allow operation of low volume clients and certain time skew between the DNS servers in the anycast.

The DNS Server SHOULD generate new Server Cookie at least if the received Server Cookie from the Client is older than half an hour.

3.5. The Hash Sub-Field

It's important that all the DNS servers use the same algorithm for computing the Server Cookie. This document defines the Version 1 of the Server Side algorithm to be:

```
  Hash = Cookie_Algorithm(
      Client Cookie | Version | Cookie Algo | Reserved | TimeStamp,
      Server Secret )
```

4. Cookie Algorithms

Implementation recommendations for Cookie Algorithms [DNSCOOKIE-IANA]:

Number	Mnemonics	Client Cookie	Server Cookie
1	FNV	MUST NOT	MUST NOT
2	HMAC-SHA-256-64	MUST NOT	MUST NOT
3	AES	MAY	MAY
4	SIPHASH24	MUST	MUST

[FNV] is a Non-Cryptographic Hash Algorithm and this document obsoletes the usage of FNV in DNS Cookies.

HMAC-SHA-256-64 is an HMAC-SHA-256 [RFC6234] algorithm reduced to 64-bit. This particular algorithm was implemented in BIND, but it was never the default algorithm and the computational costs makes it unsuitable to be used in DNS Cookies. Therefore this document obsoletes the usage of HMAC-SHA-256 algorithm in the DNS Cookies.

The AES algorithm [RFC5649] has been the default DNS Cookies algorithm in BIND until version x.y.z, and other implementations MAY implement AES algorithm as implemented in BIND for backwards compatibility. However it's recommended that new implementations implement only a pseudorandom functions for DNS Cookies, in this document that would be SipHash24.

[SipHash-2.4] is a pseudorandom function suitable as message authentication code, and this document REQUIRES compliant DNS Server to use SipHash24 as a mandatory and default algorithm for DNS Cookies to ensure interoperability between the DNS Implementations.

5. IANA Considerations

IANA is requested to create and maintain a sub-registry (the "DNS Cookie Algorithm" registry) of the "Domain Name System (DNS) Parameters" registry. The initial values for this registry are described in Section 4.

6. References

6.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC5649] Housley, R. and M. Dworkin, "Advanced Encryption Standard (AES) Key Wrap with Padding Algorithm", RFC 5649, DOI 10.17487/RFC5649, September 2009, <https://www.rfc-editor.org/info/rfc5649>.

[RFC6234] Eastlake 3rd, D. and T. Hansen, "US Secure Hash Algorithms (SHA and SHA-based HMAC and HKDF)", RFC 6234, DOI 10.17487/RFC6234, May 2011, <https://www.rfc-editor.org/info/rfc6234>.

[RFC7873] Eastlake 3rd, D. and M. Andrews, "Domain Name System (DNS) Cookies", RFC 7873, DOI 10.17487/RFC7873, May 2016, <https://www.rfc-editor.org/info/rfc7873>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

6.2. Informative References

[FNV] Fowler, G., Noll, L., Vo, K., Eastlake, D., and T. Hansen, "The FNV Non-Cryptographic Hash Algorithm", <https://datatracker.ietf.org/doc/draft-eastlake-fnv>.

[SipHash-2.4] Aumasson, J. and D. Bernstein, "SipHash: a fast short-input PRF", 2012, <https://131002.net/siphash/>.

Authors' Addresses

Ondrej Sury
Internet Systems Consortium
CZ

Email: ondrej@isc.org

Willem Toorop
NLnet Labs
Science Park 400
Amsterdam 1098 XH
Netherlands

Email: willem@nlnetlabs.nl