INTERNET-DRAFT D. Eastlake
Intended Status: Proposed Standard Futurewei
Updates: 7873 M. Andrews
ISC
Expires: January 20, 2020 July 21, 2019

Interoperable Domain Name System (DNS) Server Cookies
<draft-eastlake-dnsop-server-cookies-00.txt>

Abstract

DNS cookies, as specified in RFC 7873, are a lightweight DNS transaction security mechanism that provides limited protection to DNS servers and clients against a variety of denial-of-service and amplification, forgery, or cache poisoning attacks by off-path attackers. This document specifies a means of producing interoperable strong cookies so that an anycast server set including diverse implementations will interoperate with standard clients. This document updates RFC 7873.

Table of Contents

1. Introduction

DNS cookies, as specified in [RFC7873], are a lightweight DNS transaction security mechanism that provides limited protection to DNS servers and clients against a variety of denial-of-service and amplification, forgery, or cache poisoning attacks by off-path attackers. This document specifies a means of producing interoperable strong cookies so that an anycast server set including diverse implementations can be easily configured to interoperate with standard clients.

The threats considered for DNS Cookies and the properties of the DNS Security features other than DNS Cookies are discussed in [RFC7873].

In the case of an anycast set of DNS servers, it is desirable for any server in the set to be able to validate a server cookie recently provided to a client by a different server in the set. If such a server cookie is not recognized, it will typically result in one or more additional roundtrips increasing volume of traffic and final response latency.

There is no need for DNS client (resolver) Cookies to be interoperable across different implementations. Each client need only be able to recognize its own cookies.

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.

"Floor" is the function returning the integer part of its argument. That is, Floor(x) is the largest integer not greater than x.

"HMAC-SHA265-xx" is the HMAC [RFC2104] hash function using SHA-256 [RFC6234] truncated to xx bits.

"Off-path attacker", for a particular DNS client and server, is defined as an attacker who cannot observe the DNS request and response messages between that client and server.

"IP Address" is used herein as a length independent term covering both IPv4 and IPv6 addresses.

2. Changes to [RFC7873]

In its Appendices A.1 and B.1 [RFC7873] provides example "simple" algorithms for computing Client and Server Cookies, respectively. These algorithms are NOT RECOMMENDED as the cookies produces are too weak when evaluated against modern security standards.

In its Appendix B.2 [RFC7873] provides an example "more complex" server algorithm. This algorithm is replaced by the interoperable specification in Section 4 of this document, which is RECOMMENDED.

There is no reason for client cookies to be interoperable as clients need only recognize their own cookies, which are returned in the response to their queries. Thus, Appendix A.2 of [RFC7873] is unchanged.

3. Interoperable Server Cookie Configuration

The calculation of interoperable server cookies is determined by two configuration parameters: CookieMaster and SecretDuration. Methods of distributing these two parameters to the DNS servers in an anycast set are beyond the scope of this document but such methods MUST NOT reveal CookieMaster to adversaries who might try to forge server Cookies and SHOULD NOT reveal SecretDuration to such adversaries.

CookieMaster is the master secret used in deriving the interoperable server cookies. It is a byte string not longer than 255 bytes. It SHOULD NOT be shorter than 12 bytes and SHOULD have at least 96 bits of entropy [RFC4086].

SecretDuration is a parameter that determines when, on average, the server rolls over the derived secret used in calculating server cookies. It is a 16-bit quantity giving the average seconds between rollovers as an unsigned integer. Values less than 16 are Interpreted as being equal to 16. The maximum time that can be represented is 18 hours 12 minutes and 15 seconds.

4. Server Cookie Calculations

The calculations and internal variables specified in this section are logical. Any other calculations and variables yielding the same Server Cookies may be used.

Particular Server Cookies are calculated as specified in Section 4.1. How and when the derived ServerSecret, used in calculating Server Cookies, is calculated is specified in Section 4.2. And initialization considerations are covered in Section 4.3.

State variables needed by the DNS server for this purpose, in addition to the configuration quantities give in Section 3, are as follows:

ServerSecret is a 128-bit quantity periodically derived from CookieMaster as specified in Section 4.2.

Time is 64 bits and gives the time since the beginning of January 1, 1900, in seconds ignoring leap seconds, as an unsigned integer in network byte order.

K is a 32-bit unsigned integer representing the ServerSecret epoch.

4.1 Individual Server Cookie Calculation

The value of an interoperable server cookie is a 128-bit quantity structured as follows:

```
 <---   4 bytes   --->
+--------------------+
|       Nonce        |      32 bits
+--------------------+
|       Time         |      32 bits
+--------------------+
|                    |
+-      Hash        -+      64 bits
|                    |
+--------------------+
```

Nonce = a 32-bit locally generated random number [RFC4086] which SHOULD have 32 bits of entropy. The presence of the nonce field assures a very low probability that there would be duplicate cookies.

Time = the Time state variable defined above. The presence of this field makes it easy to reject old cookies (unless they are very old (close to some exact multiple of 2**32 seconds (a little over 136 years) ago and you are still using the same CookieMaster secret).

Hash = The Hash part of the Server Cookie is the hard-to-guess part. (The Nonce and the Time appear in plain text.) It is calculated in a two-stage process. First, the ServerSecret is occasionally calculated at the times and by the method specified in Section 4.2. Then the Hash is calculated using the ServerSecret as show further below.

Whenever a server needs a Server Cookie to include in a reply, it calculates the Hash field of that Server Cookie as follows:

```
Hash =
   HMAC-SHA256-64( ServerSecret,
      ( Client Cookie | Nonce | Time | Client IP Address) )
```

where "|" represents concatenation and the "Client Cookie" is as specified in [RFC7873].

With this method, a server sends a new 128-bit cookie back with every request.

4.2 ServerSecret Calculation Schedule

A different Server Secret is used for each time epoch. Epoch k covers the seconds numbered (see definition of Time above) from EpochStart(k) to EpochStart(k+1)-1.

The ServerSecret used during epoch K is

```
ServerSecret =
       HMAC-SHA256-128( "DNSCookie",
           ( EpochStart(K) | CookieMaster | EpochStart(K) ) )
```

where "|" represents concatenation and "DNSCookie" is that 9 byte ASCII [RFC20] string without any zero termination byte.

As describe in Section 7.1 of [RFC7873], it is desirable for jitter to be applied to the time of updating the key being used by the servers in an anycast group. Including pseudo random jitter, the start time of epoch K is as follows:

```
       EpochStart( K ) =
           K*SecretDuration +
           Floor((HMAC-SHA256-32("DNSjitter",
```

```
                  K | CookieMaster)%SecretDuration)*8/10)
```

where "|" is concatenation, "%" is the modulus operator (x%y is the remainder of x divided by y), and the output of HMAC-SHA256-32 is considered a 32-bit unsigned integer.

If Time < EpochStart(K+1), then ServerSecret is up to date. If Time >= EpochStart(K+1), then ServerSecret needs to be updated as specified at the beginning of this section and K is incremented. There are various strategies for keeping ServerSecret up do date. The test at the beginning of this paragraph and conditional update of ServerSecret could be done once a second. However, it would be more efficient, when that test is done, to save EpochStart(K+1) as the next time the test and conditional updates should be performed.

4.3 Initialization

Whenever the value of SecretDuration or CookieMaster or both is set of if a significant change is made to Time, the steps in this section MUST run before any Server Cookies are generated. (A change to Time is significant if the first two steps below would result in a value for K different from the current value.)

- o K is set to Floor(Time/SecretDuration).

- o If Time < EpochStart(K), then set K to K-1.

- o Calculate and set ServerSecret as specified at the beginning of Section 4.2.

5. IANA Considerations

This document requires no IANA actions.

6. Security Considerations

The minor adjustments to the Server Cookie calculation algorithm made by this document do not affect the security considerations provided in [RFC7873].

Normative References

[RFC20] - Cerf, V., "ASCII format for network interchange", STD 80, RFC 20, DOI 10.17487/RFC0020, October 1969, <https://www.rfc-editor.org/info/rfc20>.

[RFC2104] - Krawczyk, H., Bellare, M., and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication", RFC 2104, DOI 10.17487/RFC2104, February 1997, <https://www.rfc-editor.org/info/rfc2104>.

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

[RFC4086] - Eastlake 3rd, D., Schiller, J., and S. Crocker, "Randomness Requirements for Security", BCP 106, RFC 4086, DOI 10.17487/RFC4086, June 2005, <http://www.rfc-editor.org/info/rfc4086>.

[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, <http://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>.

Informative References

None.

Acknowledgements

The suggestions and contributions of the following are gratefully acknowledged:

Petr Spacek

Author's Address

Donald E. Eastlake 3rd
Futurewei Technologies
1424 Pro Shop Court
Davenport, FL 33896 USA

Telephone: +1-508-333-2270
EMail: d3e3e3@gmail.com

Mark Andrews
Internet Systems Consortium
950 Charter Street
Redwood City, CA 94063 USA

Email: marka@isc.org

Copyright, Disclaimer, and Additional IPR Provisions