Network Working Group
Internet-Draft
Updates: 6222 (if approved)
Intended status: Standards Track
Expires: January 10, 2013

E. Rescorla
RTFM, Inc.

July 09, 2012

Random algorithm for RTP CNAME generation
draft-rescorla-avtcore-random-cname-00

Abstract

RFC 6222 describes a number of mechanisms for generating a unique CNAME Unfortunately, these algorithms are rather complicated and also produce CNAMEs which in some cases are potentially linkable over multiple RTCP sessions even if a new CNAME is generated for each session. This document specifies a replacement algorithm for the algorithm in Section 5 which does not have this limitation and is also simpler to implement.

Table of Contents

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 RFC 2119 [RFC2119].

2. Introduction

[RFC6222] defines a set of algorithms for generating unique RTP CNAMES [RFC3550]. Although these algorithms attempt to provide some privacy, the CNAMEs they generate are still potentially linkable, as acknowledged in the security considerations section of RFC 6222. This document describes a simpler algorithm which produces an identifier which is compatible with RFC 6222 identifiers and is in fact indistinguishable from them without significant computational effort,

RFC 6222 Section 4.2 requires:

" An RTP endpoint that wishes to generate a per-session RTCP CNAME MUST use the following method:

o For every new RTP session, a new CNAME is generated following the procedure described in Section 5. After performing that procedure, the least significant 96 bits are used to generate an identifier (to compromise between packet size and security), which is converted to ASCII using Base64 encoding [RFC4648]. This results in a 16-octet string representation. The RTCP CNAME cannot change over the life of an RTP session [RFC3550]; hence, only the initial SSRC value chosen by the endpoint is used. The "user@" part of the RTCP CNAME is omitted when generating per-session RTCP CNAMEs."

The algorithm in Section 5 of RFC 6222 is a cryptographic hash of the following input values:

- The current time in 64-bit NTP format
- An EUI-64 or 48-bit MAC address [RFC4291].
- The initial SSRC and source and destination address/port quartets

The result of this process is a random-appearing binary value which can then be converted to a CNAME by the process described above. Unfortunately, in many settings the input values do not provide sufficient entropy, thus making it possible to determine if multiple CNAME values were generated on the same machine.

2.1. Linkability of the RFC 6222 algorithm

While the output of the RFC6222 algorithm is with high probability unique, it is not clearly unlinkable. Consider the case where we have two CNAMEs C1 and C2 and we wish to determine whether they were generated by the same endpoint. This situation might occur if multiple calls were made from some anonymous location like a domestic violence shelter. For instance, the attacker receives a call from an unknown location and then calls a number of candidate locations in an attempt to determine if they are the same. Starting with C1, the attacker exhaustively searches all the potential input values to find a set which hashes to C1. He then can simply search the nearby input space and if the result is C2, he knows that the calls involve the same endpoint.

The complexity of this attack is directly related to the entropy of the input variables. At minimum the attacker knows:

- The destination IP address and port exactly.
- The timestamp (from the RTP header) to within a few seconds. With a typical 100 ticks/second clock, this represents about 10 bits of entropy at most (and potentially more like 2-3 bits)
- The SSRC (from the RTP header).

This leaves the primary sources of entropy as the source IP address/port and the MAC/EUI-64 address. RFC 6222 is unclear on which IP address/port is to be used, but there are three main possibilities:

- A relayed address/port (known to the attacker) by looking at the RTP. [Note we are assuming that a media relay is used otherwise linkability is trivial.]
- The local IP address (most likely chosen from a very small number of local addresses in the the 10.0.x.x. or 192.168.x.x range.). As residential NATs generally assign addresses in sequence and phones are often the first item to reboot addresses 10.0.0.1, 192.168.0.1, and 192.168.1.1 are very common with the first 5 addresses in each range representing a large fraction of all devices.
- The public IP address of the peer--hard to guess but easy to determine with a scan and not really a natural choice.

Similarly, the port in use is often not chosen randomly but often from a small set of initial ports chosen by the implementation (by default Cisco devices often use 16000-16004. Thus, while in principle there are 48 bits of randomness in the IP and port, in practice they may offer no entropy (in the case where the relayed address is used as the RFC 6222 input) or only 7 bits (where the local address is used but the client is behind a residential NAT and

uses a limited port range.)

Similarly, while in principle the MAC address has 48 bits of entropy, in practice devices are easily fingerprinted and once the manufacturer is known, the MAC address is restricted to the much narrower range assigned to the manufacturer, which are again often assigned in sequence (on the order of 20-32 bits).

Thus, in order to mount the initial attack, the attacker need search somewhere between 20-30 bits (if the relayed address is known) and 70 bits. On the upper end, there is no real linkability problem, but on the lower end linkability is practical. The lower-end case is relevant to many residential and small business settings (exactly the kind operated by DV shelters) with "natural" implementations of RFC 6222.

3. Alternative Algorithm

In this document, we propose an alternative approach based on simply generating a cryptographically pseudorandom value. Implementations conformant with this specification MAY replace the algorithm in Section 5 with a random value generated using a cryptographic random number generator [RFC4096]. This value MUST be at least 96 bits but MAY be longer (see Section 4 for analysis of the length).

3.1. Comparison to RFC 6222 Algorithm

3.1.1. Ease of implementation

The biggest bottleneck to implementation of this algorithm is the availability of an appropriate cryptographically secure PRNG (CSPRNG). In any setting whcih already has a secure PRNG, this algorithm described is far simpler, and many implementations already have this capability. SIP stacks [RFC3261] are required to use cryptographically random numbers to generate To and From tags (Section 19.3). RTCWEB implementations [I-D.ietf-rtcweb-security-arch] will need to have secure PRNGs to implement ICE [RFC5245] and DTLS-SRTP [RFC5764]. And of course essentially every Web browser already supports TLS, which requires a secure PRNG.

3.1.2. Format

The output produced by this algorithm is a string of random bits. If it is of length 96 bits, it is indistinguishable from the output of the RFC6222 algorithm without significant computation (see Section 2.1).

3.1.3. Uniqueness

One concern that is often raised whenever random numbers are proposed is that of uniqueness. However, for the purposes of statistical uniqueness, the RFC6222 algorithm has equivalent properties to a PRNG, since the chance of the hashes of any two arbitrarily chosen strings colliding are the same as those of any two random strings colliding (or else this constitutes a weakness in the hash.)

3.1.4. Linkability

A basic design criterion of a good CSPRNG is that it not be possible to distinguish its output from random values. Clearly, identifying two outputs as being from the same CSPRNG would violate this requirement In order to mount the attack described in Section 2.1 would require exhaustively searching the seed space of the PRNG. Any conditions under which this was practical would represent a severe threat to the security of the CSPRNG if used in any communications security setting.

4. Security Considerations

The privacy properties of the algorithm described here are as strong or stronger than those of the RFC6222 algorithm. Because of the properties of the PRNG, there is no significant privacy/linkability difference between long and short CNAMEs. However, the requirement to generate unique CNAMEs implies a certain minimum length. A length of 96-bits allows on the order of 2^{40} CNAMEs globally before there is a large chance of collision (there is about a 50% chance of one collision after 2^{48} CNAMEs).

5. References

5.1. Normative References

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

[RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, July 2003.

[RFC4096] Malamud, C., "Policy-Mandated Labels Such as "Adv:" in Email Subject Headers Considered Ineffective At Best", RFC 4096, May 2005.

[RFC6222] Begen, A., Perkins, C., and D. Wing, "Guidelines for Choosing RTP Control Protocol (RTCP) Canonical Names (CNAMEs)", RFC 6222, April 2011.

5.2. Informative References

[I-D.ietf-rtcweb-security-arch] Rescorla, E., "RTCWEB Security Architecture", draft-ietf-rtcweb-security-arch-02 (work in progress), June 2012.

[RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, June 2002.

[RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, February 2006.

[RFC5245] Rosenberg, J., "Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols", RFC 5245, April 2010.

[RFC5764] McGrew, D. and E. Rescorla, "Datagram Transport Layer Security (DTLS) Extension to Establish Keys for the Secure Real-time Transport Protocol (SRTP)", RFC 5764, May 2010.

Author's Address

Eric Rescorla
RTFM, Inc.
2064 Edgewood Drive
Palo Alto, CA 94303
USA

Phone: +1 650 678 2350
Email: ekr@rtfm.com