Internet Engineering Task Force (IETF)
Request for Comments: 6222
Updates: 3550
Category: Standards Track
ISSN: 2070-1721

A. Begen
Cisco
C. Perkins
University of Glasgow
D. Wing
Cisco

April 2011

Guidelines for Choosing RTP Control Protocol (RTCP) Canonical Names (CNAMEs)

Abstract

The RTP Control Protocol (RTCP) Canonical Name (CNAME) is a persistent transport-level identifier for an RTP endpoint. While the Synchronization Source (SSRC) identifier of an RTP endpoint may change if a collision is detected or when the RTP application is restarted, its RTCP CNAME is meant to stay unchanged, so that RTP endpoints can be uniquely identified and associated with their RTP media streams. For proper functionality, RTCP CNAMEs should be unique within the participants of an RTP session. However, the existing guidelines for choosing the RTCP CNAME provided in the RTP standard are insufficient to achieve this uniqueness. This memo updates those guidelines to allow endpoints to choose unique RTCP CNAMEs.

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 5741.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at http://www.rfc-editor.org/info/rfc6222.

Table of Contents

1. Introduction

In Section 6.5.1 of the RTP specification, [RFC3550], there are a number of recommendations for choosing a unique RTCP CNAME for an RTP endpoint. However, in practice, some of these methods are not guaranteed to produce a unique RTCP CNAME. This memo updates guidelines for choosing RTCP CNAMEs, superseding those presented in Section 6.5.1 of [RFC3550].

2. Requirements Notation

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 [RFC2119].

3. Deficiencies with Earlier Guidelines for Choosing an RTCP CNAME

The recommendation in [RFC3550] is to generate an RTCP CNAME of the form "user@host" for multiuser systems, or "host" if the username is not available. The "host" part is specified to be the fully qualified domain name (FQDN) of the host from which the real-time data originates. While this guidance was appropriate at the time [RFC3550] was written, FQDNs are no longer necessarily unique and can sometimes be common across several endpoints in large service provider networks. This document replaces the use of FQDN as an RTCP CNAME by alternative mechanisms.

IPv4 addresses are also suggested for use in RTCP CNAMEs in [RFC3550], where the "host" part of the RTCP CNAME is the numeric representation of the IPv4 address of the interface from which the RTP data originates. As noted in [RFC3550], the use of private network address space [RFC1918] can result in hosts having network addresses that are not globally unique. Additionally, this shared use of the same IPv4 address can also occur with public IPv4 addresses if multiple hosts are assigned the same public IPv4 address and connected to a Network Address Translation (NAT) device [RFC3022]. When multiple hosts share the same IPv4 address, whether private or public, using the IPv4 address as the RTCP CNAME leads to RTCP CNAMEs that are not necessarily unique.

It is also noted in [RFC3550] that if hosts with private addresses and no direct IP connectivity to the public Internet have their RTP packets forwarded to the public Internet through an RTP-level translator, they could end up having non-unique RTCP CNAMEs. The suggestion in [RFC3550] is that such applications provide a configuration option to allow the user to choose a unique RTCP CNAME; this technique puts the burden on the translator to translate RTCP CNAMEs from private addresses to public addresses if necessary to keep private addresses from being exposed. Experience has shown that this does not work well in practice.

4. Choosing an RTCP CNAME

It is difficult, and in some cases impossible, for a host to determine if there is a NAT between itself and its RTP peer. Furthermore, even some public IPv4 addresses can be shared by multiple hosts in the Internet. Using the numeric representation of the IPv4 address as the "host" part of the RTCP CNAME is NOT RECOMMENDED.

4.1. Persistent RTCP CNAMEs versus Per-Session RTCP CNAMEs

The RTCP CNAME can be either persistent across different RTP sessions for an RTP endpoint or unique per session, meaning that an RTP endpoint chooses a different RTCP CNAME for each RTP session.

An RTP endpoint that is emitting multiple related RTP streams that require synchronization at the other endpoint(s) MUST use the same RTCP CNAME for all streams that are to be synchronized. This requires a short-term persistent RTCP CNAME that is common across several RTP streams, and potentially across several related RTP sessions. A common example of such use occurs when lip-syncing audio and video streams in a multimedia session, where a single participant has to use the same RTCP CNAME for its audio RTP session and for its video RTP session. Another example might be to synchronize the layers of a layered audio codec, where the same RTCP CNAME has to be used for each layer.

A longer-term persistent RTCP CNAME is sometimes useful to facilitate third-party monitoring, consistent with [RFC3550]. One such use might be to allow network management tools to correlate the ongoing quality of service for a participant across multiple RTP sessions for fault diagnosis, and to understand long-term network performance statistics. An implementation that wishes to discourage this type of third-party monitoring can generate a unique RTCP CNAME for each RTP session, or group of related RTP sessions, that it joins. Such a per-session RTCP CNAME cannot be used for traffic analysis, and so provides some limited form of privacy (note that there are non-RTP means that can be used by a third party to correlate RTP sessions, so the use of per-session RTCP CNAMEs will not prevent a determined traffic analyst from monitoring such sessions).

This memo defines several different ways by which an implementation can choose an RTCP CNAME. It is possible, and legitimate, for independent implementations to make different choices of RTCP CNAME when running on the same host. This can hinder third-party monitoring, unless some external means is provided to configure a persistent choice of RTCP CNAME for those implementations.

Note that there is no backwards compatibility issue (with [RFC3550]-compatible implementations) introduced in this memo, since the RTCP CNAMEs are opaque strings to remote peers.

4.2. Requirements

RTP endpoints will choose to generate RTCP CNAMEs that are persistent or per-session. An RTP endpoint that wishes to generate a persistent RTCP CNAME MUST use one of the following two methods:

- To produce a long-term persistent RTCP CNAME, an RTP endpoint MUST generate and store a Universally Unique IDentifier (UUID) [RFC4122] for use as the "host" part of its RTCP CNAME. The UUID MUST be version 1, 2, or 4, as described in [RFC4122], with the "urn:uuid:" stripped, resulting in a 36-octet printable string representation.

- To produce a short-term persistent RTCP CNAME, an RTP endpoint MUST either (a) use the numeric representation of the layer-2 (Media Access Control (MAC)) address of the interface that is used to initiate the RTP session as the "host" part of its RTCP CNAME or (b) generate and use an identifier by following the procedure described in Section 5. In either case, the procedure is performed once per initialization of the software. After obtaining an identifier by doing (a) or (b), the least significant 48 bits are converted to the standard colon-separated hexadecimal format [RFC5342], e.g., "00:23:32:af:9b:aa", resulting in a 17-octet printable string representation.

In the two cases above, the "user@" part of the RTCP CNAME MAY be omitted on single-user systems and MAY be replaced by an opaque token on multi-user systems, to preserve privacy.

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

- 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.

It is believed that obtaining uniqueness (with a high probability) is an important property that requires careful evaluation of the method. This document provides a number of methods, at least one of which would be suitable for all deployment scenarios. This document therefore does not provide for the implementor to define and select an alternative method.

A future specification might define an alternative method for generating RTCP CNAMEs, as long as the proposed method has appropriate uniqueness and there is consistency between the methods used for multiple RTP sessions that are to be correlated. However, such a specification needs to be reviewed and approved before deployment.

The mechanisms described in this document are to be used to generate RTCP CNAMEs, and they are not to be used for generating general-purpose unique identifiers.

5. Procedure to Generate a Unique Identifier

The algorithm described below is intended to be used for locally generated unique identifiers.

1. Obtain the current time of day in 64-bit NTP format [RFC5905].

2. Obtain a modified EUI-64 identifier from the system running this algorithm [RFC4291]. If such a system does not exist, an identifier can be created from a 48-bit MAC address, as specified in [RFC4291]. If one cannot be obtained or created, a suitably unique identifier, local to the node, should be used (e.g., system serial number).

3. Concatenate the time of day with the system-specific identifier in order to create a key.

4. If generating a per-session CNAME, also concatenate the RTP endpoint’s initial SSRC, the source and destination IP addresses, and ports to the key.

5. Compute the 256-bit output of the SHA-256 digest of the key, as specified in [RFC4634].

6. Security Considerations

The security considerations of [RFC3550] apply to this memo.

6.1. Considerations on Uniqueness of RTCP CNAMEs

The recommendations given in this document for RTCP CNAME generation ensure that a set of cooperating participants in an RTP session will, with very high probability, have unique RTCP CNAMEs. However, neither [RFC3550] nor this document provides any way to ensure that participants will choose RTCP CNAMEs appropriately, and thus implementations MUST NOT rely on the uniqueness of CNAMEs for any essential security services. This is consistent with [RFC3550], which does not require that RTCP CNAMEs are unique within a session but instead says that condition SHOULD hold. As described in the Security Considerations section of [RFC3550], because each participant in a session is free to choose its own RTCP CNAME, they can do so in such a way as to impersonate another participant. That is, participants are trusted to not impersonate each other. No recommendation for generating RTCP CNAMEs can prevent this impersonation, because an attacker can neglect the stipulation. Secure RTP (SRTP) [RFC3711] keeps unauthorized entities out of an RTP session, but it does not aim to prevent impersonation attacks from unauthorized entities.

This document uses a hash function to ensure the uniqueness of RTCP CNAMEs. A cryptographic hash function is used because such functions provide the randomness properties that are needed. However, no security assumptions are made on the hash function. The hash function is not assumed to be collision resistant, preimage resistant, or second preimage resistant in an adversarial setting; as described above, an attacker attempting an impersonation attack could merely set the RTCP CNAME directly rather than attacking the hash function. Similarly, the hash function is not assumed to be a one-way function or pseudorandom in a cryptographic sense.

No confidentiality is provided on the data used as input to the RTCP CNAME generation algorithm. It might be possible for an attacker who observes an RTCP CNAME to determine the inputs that were used to generate that value.

6.2. Session Correlation Based on RTCP CNAMEs

In some environments, notably telephony, a fixed RTCP CNAME value allows separate RTP sessions to be correlated and eliminates the obfuscation provided by IPv6 privacy addresses [RFC4941] or IPv4 Network Address Port Translation (NAPT) [RFC3022]. SRTP [RFC3711] can help prevent such correlation by encrypting Secure RTCP (SRTCP),

but it should be noted that SRTP only mandates SRTCP integrity protection (not encryption). Thus, RTP applications used in such environments should consider encrypting their SRTCP or generate a per-session RTCP CNAME as discussed in Section 4.1.

7. Acknowledgments

Thanks to Marc Petit-Huguenin, who suggested using UUIDs in generating RTCP CNAMEs. Also, thanks to David McGrew for providing text for the Security Considerations section.

8. References

8.1. Normative References

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

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

[RFC4122] Leach, P., Mealling, M., and R. Salz, "A Universally Unique IDentifier (UUID) URN Namespace", RFC 4122, July 2005.

[RFC4634] Eastlake 3rd, D. and T. Hansen, "US Secure Hash Algorithms (SHA and HMAC-SHA)", RFC 4634, July 2006.

[RFC4648] Josefsson, S., "The Base16, Base32, and Base64 Data Encodings", RFC 4648, October 2006.

[RFC5905] Mills, D., Martin, J., Ed., Burbank, J., and W. Kasch, "Network Time Protocol Version 4: Protocol and Algorithms Specification", RFC 5905, June 2010.

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

[RFC5342] Eastlake 3rd, D., "IANA Considerations and IETF Protocol Usage for IEEE 802 Parameters", BCP 141, RFC 5342, September 2008.

8.2. Informative References

[RFC1918] Rekhter, Y., Moskowitz, B., Karrenberg, D., de Groot, G., and E. Lear, "Address Allocation for Private Internets", BCP 5, RFC 1918, February 1996.

[RFC3022] Srisuresh, P. and K. Egevang, "Traditional IP Network Address Translator (Traditional NAT)", RFC 3022, January 2001.

[RFC3711] Baugher, M., McGrew, D., Naslund, M., Carrara, E., and K. Norrman, "The Secure Real-time Transport Protocol (SRTP)", RFC 3711, March 2004.

[RFC4941] Narten, T., Draves, R., and S. Krishnan, "Privacy Extensions for Stateless Address Autoconfiguration in IPv6", RFC 4941, September 2007.

Authors' Addresses

Ali Begen
Cisco
181 Bay Street
Toronto, ON M5J 2T3
CANADA

EMail: abegen@cisco.com

Colin Perkins
University of Glasgow
School of Computing Science
Glasgow G12 8QQ
UK

EMail: csp@csperkins.org

Dan Wing
Cisco Systems, Inc.
170 West Tasman Dr.
San Jose, CA 95134
USA

EMail: dwing@cisco.com