Workgroup: RADEXT Working Group
Internet-Draft:
draft-dekok-radext-reverse-coa-00
Published: 19 October 2022
Intended Status: Standards Track
Expires: 22 April 2023
Authors: A. DeKok
FreeRADIUS
V. Cargatser
Cisco

Reverse CoA in RADIUS

Abstract

This document defines a "reverse change of authorization (CoA)" path for RADIUS packets. This specification allows a home server to send CoA packets in "reverse" down a RADIUS/TLS connection. Without this capability, it is impossible for a home server to send CoA packets to a NAS which is behind a firewall or NAT gateway. The reverse CoA functionality extends the available transport methods for CoA packets, but it does not change anything else about how CoA packets are handled.

About This Document

This note is to be removed before publishing as an RFC.

Status information for this document may be found at https://datatracker.ietf.org/doc/draft-dekok-radext-reverse-coa/.

Discussion of this document takes place on the RADEXT Working Group mailing list (mailto:radext@ietf.org), which is archived at https://mailarchive.ietf.org/arch/browse/radext/.

Source for this draft and an issue tracker can be found at https://github.com/freeradius/reverse-coa.git.

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 https://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 22 April 2023.

Table of Contents

1. Introduction

[RFC5176] defines the ability to change a users authorization, or disconnect the user via what are generally called "Change of Authorization" or "CoA" packets. This term refers to either of the RADIUS packet types CoA-Request or Disconnect-Request. The initial transport protocol for all RADIUS was the User Datagram Protocol (UDP).

[RFC6614] updated previous specifications to allow packets to be sent over the Transport Layer Security (TLS) protocol. Section 2.5

of that document explicitly allows all packets (including CoA) to be sent over a TLS connection:

Due to the use of one single TCP port for all packet types, it is required that a RADIUS/TLS server signal which types of packets are supported on a server to a connecting peer. See also Section 3.4 for a discussion of signaling.

These specifications assume that a RADIUS client can directly contact a RADIUS server, which is the normal "forward" path for packets between a client and server. However, it is not always possible for the RADIUS server to send CoA packets to the RADIUS client. If a RADIUS server wishes to act as a CoA client, and send CoA packets to the NAS (CoA server), the "reverse" path can be blocked by a firewall, NAT gateway, etc. That is, a RADIUS server has to be reachable by a NAS, but there is usually no requirement that the NAS is reachable from a public system. To the contrary, there is usually a requirement that the NAS is not publicly accessible.

This scenario is most evident in a roaming / federated environment such as Eduroam or OpenRoaming. It is in general impossible for a home server to signal the NAS to disconnect a user. There is no direct reverse path from the home server to the NAS, as the NAS is not publicly addressible. Even if there was a public reverse path, it would generally be unknowable, as intermediate proxies can (and do) attribute rewriting to hide NAS identies.

These limitations can result in business losses and security problems, such as the inability to disconnect an online user when their account has been terminated.

As the reverse path is usally blocked, it means that it is in general possible only to send CoA packets to a NAS when the NAS and RADIUS server share the same private network (private IP space or IPSec). Even though [RFC8559] defines CoA proxying, that specification does not address the issue of NAS reachability.

This specification solves that problem. The solution is to simply allow CoA packets to go in "reverse" down an existing RADIUS/TLS connection. That is, when a NAS connects to a RADIUS server it normally sends request packets (Access-Request, etc.) and expects to receive response packets (Access-Accept, etc.). This specification extends RADIUS/TLS by permitting a RADIUS server to re-use an existing TLS connection to send CoA packets to the NAS, and permitting the NAS to send CoA response packets to the RADIUS server over that same connection.

We note that while this document specifically mentions RADIUS/TLS, it should be possible to use the same mechanisms on RADIUS/DTLS [RFC7360]. However at the time of writing this specification, no implementations exist for "reverse CoA" over RADIUS/DTLS. As such, when we refer to "TLS" here, or "RADIUS/TLS", we implicitly include RADIUS/DTLS in that description.

We also note that while this same mechanism could theoretically be used for RADIUS/UDP and RADIUS/TCP, there is no value in defining "reverse CoA" for those transports. Therefore for practial purposes, "reverse CoA" means RADIUS/TLS and RADIUS/DTLS.

There are additional considerations for proxies. While [RFC8559] describes CoA proxying, there are still issues which need to be addressed for the "reverse CoA" use-case. This specification describes how a proxy can implement "reverse CoA" proxying, including signalling necessary to negotiate this functionality.

2. Terminology

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.

*CoA

Change of Authorization packets. For brevity, when this document refers to "CoA" packets, it means either or both of CoA-Request and Disconnect-Request packets.

*ACK

Change of Authorization "positive acknowlegement" packets. For brevity, when this document refers to "ACK" packets, it means either or both of CoA-ACK and Disconnect-ACK packets.

*NAK

Change of Authorization "negative acknowlegement" packets. For brevity, when this document refers to "ACK" packets, it means either or both of CoA-NAK and Disconnect-NAK packets.

*RADIUS/TLS

RADIUS over the Transport Layer Security protocol [RFC6614]

*RADIUS/DTLS

RADIUS over the Datagram Transport Layer Security protocol [RFC7360]

*TLS

Either RADIUS/TLS or RADIUS/DTLS.

*reverse CoA

CoA, ACK, or NAK packets sent over a RADIUS/TLS or RADIUS/DTLS connection which was made from a RADIUS client to a RADIUS server.

3. Concepts

The reverse CoA functionality is based on two additions to RADIUS. The first addition is a configuration and signalling, to indicate that a RADIUS client is capable of accepting reverse CoA packets. The second addition is an extension to the "reverse" routing table for CoA packets which was first described in Section 2.1 of [RFC8559].

4. Capability Configuration and Signalling

In order for a RADIUS server to send reverse CoA packets to a client, it must first know that the client is capable of accepting these packets.

This functionality can be enabled in one of two ways. The first is a simple static configuration between client and server, where both are configured to allow reverse CoA. The second method is via per-connection signalling between client and server.

The server manages this functionality with two boolean flags, one per-client, and one per-connection. The per-client flag can be statically configured, and if not present MUST be treated as having a "false" value. The per-connection flag MUST be initialized from the per-client flag, and then can be dynamically negotiated after that.

4.1. Configuration Flag

Clients and servers implementing reverse CoA SHOULD have a configuration flag which indicates that the other party supports the reverse CoA functionality. That is, the client has a per-server flag enabling (or not) reverse CoA functionality. The server has a similar per-client flag.

The flag can be used where the parties are known to each other. The flag can also be used in conjunction with dynamic discovery ([RFC7585]), so long as the server associates the flag with the client identity and not with any particular IP address. That is, the flag can be associated with any method of identifying a particular client such as TLS-PSK identity, information in a client certificate, etc.

For the client, the flag controls whether or not it will accept reverse CoA packets from the server, and whether the client will do dynamic signalling of the reverse CoA functionality.

Separately, each side also needs to have a per-connection flag, which indicates whether or not this connection supports reverse CoA. The per-connection flag is initialized from the static flag, and is then dynamically updated after that.

4.2. Dynamic Signalling

The reverse CoA functionality can be signalled on a per-connection basis by the client sending a Status-Server packet when it first opens a connection to a server. This packet contains a Capability attribute (see below), with value "Reverse-CoA". The existence of this attribute in a Status-Server packet indicates that the client

supports reverse CoA over this connection. The Status-Server packet MUST be the first packet sent when the connection is opened, in order to perform per-connection signalling. A server which does not implement reverse CoA simply ignores this attribute, as per [RFC2865] Section 5.

A server implementing reverse CoA does not need to signal the NAS in any response, to indicate that it is supports reverse CoA. If the server never sends reverse CoA packets, then such signalling is unnecessary. If the server does send reverse CoA packets, then the packets themselves serve as sufficiant signalling.

The NAS may send additional Status-Server packets down the same connection, as per [RFC3539]. These packets do not need to contain the Capability attribute, so it can generally be omitted. That is, there is no need to signal the addition or removal of reverse CoA functionality during the lifetime of one connection. If a client decides that it no longer wants to support reverse CoA on a particular connection, it can simply tear down the connection, and open a new one which does not negotiate the reverse CoA functionality.

RADIUS client implementations which support reverse CoA MUST always signal that functionality in a Status-Server packet on any new connection. There is little reason to save a few octets, and having explicit signalling can help with implementations, deployment, and debugging.

The combination of static configuration and dynamic configuration means that it is possible for client and server to both agree on whether or not a particular connection supports reverse CoA.

5. Reverse Routing

The "reverse" routing table for CoA packets was first described in Section 2.1 of [RFC8559]. We extend that table here.

In our extension, the table does not map realms to home servers. Instead, it maps keys to connections. The keys will be defined in more detail below. For now, we say that keys can be derived from a RADIUS client to server connection, and from the contents of a CoA packet which needs to be routed.

When the server recieves a TLS connection from a client, it derives a key for that connection, and associates the connection with that key. A server MUST support associating one particular key value with multiple connections. A server MUST support associating multiple keys for one connection. That is, the "key to connection" mapping is N to M. It is not one-to-one, or 1-N, or M-1.

When the server recieves a CoA packet, it derives a key from that packet, and determines if there is a connection or connections which maps to that key. Where there is no available connection, the server MUST return a NAK packet that contains an Error-Cause Attribute having value 502 ("Request Not Routable").

As with normal proxying, a particular packet can sometimes have the choice more than one connection which can be used to reach a destination. In that case, issues of load-balancing, fail-over, etc. are implementation-defined, and are not discussed here. The server simply chooses one connection, and sends the reverse CoA packet down that connection.

The server then waits for a reply, doing retransmission if necessary. For all issues other than the connection being used, reverse CoA packets are handled as defined in [RFC5176] and in [RFC8559].

That is, when the NAS and server are known to each other, [RFC5176] is followed when sending CoA packets to the NAS. The difference is that instead of originating connections to the NAS, the server simply re-uses inbound TLS connections from the NAS. The NAS is identified by attributes such as NAS-Identifier, NAS-IP-Address, and NAS-IPv6-Address.

When a server is proxying to another server, [RFC8559] is following when proxying CoA packets. The "next hop" is identified either by Operator-Name for proxy-to-proxy connections. When the CoA packet reaches a visited network, that network identifies the NAS by examining the Operator-NAS-Identifier attribute.

5.1. Retransmits

Retransmissions of reverse CoA packets are handled identically to normal CoA packets. That is, the reverse CoA functionality extends the available transport methods for CoA packets, it does not change anything else about how CoA packets are handled.

6. Implementation Status

FreeRADIUS supports CoA proxying using Vendor-Specific attributes. It also permits RADIUS clients to send Status-Server packets over a RADIUS/TLS connection which contain Operator-Name. This information is used to determne which realms are accessible via reverse CoA over which RADIUS/TLS connection.

Cisco supports reverse CoA as of Cisco IOS XE Bengaluru 17.6.1 via Vendor-Specific attributes. https://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst9300/software/release/17-6/configuration_guide/sec/b_176_sec_9300_cg/configuring_radsec.pdf

Aruba documentation states that "Instant supports dynamic CoA (RFC 3576) over RadSec and the RADIUS server uses an existing TLS connection opened by the Instant AP to send the request." https://www.arubanetworks.com/techdocs/Instant_83_WebHelp/Content/Instant_UG/Authentication/ConfiguringRadSec.htm

7. Privacy Considerations

This document does not change or add any privacy considerations over previous RADIUS specifications.

8. Security Considerations

This document increases network security by removing the requirement for non-standard "reverse" paths for CoA-Request and Disconnect-Request packets.

9. IANA Considerations

TBD - new RADIUS attribute - Capability

User Operator Namespace Identifier namespace.

+,Realm Add,(this document) -,Realm Delete,(this document)

10. Acknowledgements

Thanks to Heikki Vatiainen for testing a preliminary implementation in Radiator, and for verifying interoperability with NAS equipment.

11. Changelog

12. References

12.1. Normative References

[BCP14] 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>.

[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>.

[RFC2865] Rigney, C., Willens, S., Rubens, A., and W. Simpson, "Remote Authentication Dial In User Service (RADIUS)", RFC 2865, DOI 10.17487/RFC2865, June 2000, <https://www.rfc-editor.org/info/rfc2865>.

[RFC3539] Aboba, B. and J. Wood, "Authentication, Authorization and Accounting (AAA) Transport Profile", RFC 3539, DOI 10.17487/RFC3539, June 2003, <https://www.rfc-editor.org/info/rfc3539>.

[RFC7585] Winter, S. and M. McCauley, "Dynamic Peer Discovery for RADIUS/TLS and RADIUS/DTLS Based on the Network Access Identifier (NAI)", RFC 7585, DOI 10.17487/RFC7585, October 2015, <https://www.rfc-editor.org/info/rfc7585>.

[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>.

[RFC8559] DeKok, A. and J. Korhonen, "Dynamic Authorization Proxying in the Remote Authentication Dial-In User Service (RADIUS) Protocol", RFC 8559, DOI 10.17487/RFC8559, April 2019, <https://www.rfc-editor.org/info/rfc8559>.

12.2. Informative References

[RFC5176] Chiba, M., Dommety, G., Eklund, M., Mitton, D., and B. Aboba, "Dynamic Authorization Extensions to Remote Authentication Dial In User Service (RADIUS)", RFC 5176, DOI 10.17487/RFC5176, January 2008, <https://www.rfc-editor.org/info/rfc5176>.

[RFC6614] Winter, S., McCauley, M., Venaas, S., and K. Wierenga, "Transport Layer Security (TLS) Encryption for RADIUS", RFC 6614, DOI 10.17487/RFC6614, May 2012, <https://www.rfc-editor.org/info/rfc6614>.

[RFC7360] DeKok, A., "Datagram Transport Layer Security (DTLS) as a Transport Layer for RADIUS", RFC 7360, DOI 10.17487/RFC7360, September 2014, <https://www.rfc-editor.org/info/rfc7360>.

Authors' Addresses

Alan DeKok
FreeRADIUS

Email: aland@freeradius.org

Vadim Cargatser
Cisco

Email: vcargats@cisco.com