RADEXT Working Group
Internet-Draft
Intended status: Standards Track
Expires: 28 November 2025

A. DeKok
InkBridge
V. Cargatser
Cisco

27 May 2025

Reverse Change of Authorization (CoA) in RADIUS/TLS
draft-ietf-radext-reverse-coa-06

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-ietf-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/. Subscribe at https://www.ietf.org/mailman/listinfo/radext/.

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

Status of This Memo

This Internet-Draft will expire on 28 November 2025.

Copyright Notice

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.

This mechanism does not depend on the underlying transport protocol, or interact with it. It is therefore compatible not only with [RFC6614], and [RFC7360], but also with [I-D.ietf-radext-radiusdtls-bis] which will replace those earlier standards.

This mechanism is not needed for RADIUS/UDP, as UDP is connectionless. [RFC8559] suffices for CoA when using RADIUS/UDP. For RADIUS/TCP, while this same mechanism could theoretically be used there, RADIUS/TCP is being deprecated by [I-D.ietf-radext-deprecating-radius]. 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 those systems can implement "reverse CoA" proxying, including processing packets through both an intermediate proxy network, and at the visited network.

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 "NAK" 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.

Clients and servers implementing reverse CoA MUST 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.

The configuration flag allows administators to statically enable this functionality, based on out-of-band discussions with other administators. This process is best used in an environment where all RADIUS proxies are known (or required) to have a particular set of functionality, as with a roaming consortium.

This specification does not define a way for clients and servers to negotiate this functionality on a per-connection basis. The RADIUS protocol has little, if any, provisions for capability negotiations, and this specification is not the place to add that functionality.

Without notification, however, it is possible for clients and servers to have mismatched configurations. Where a client is configured to accept reverse CoA packets and the next hop server is not configured to send them, no packets will be sent. Where a client is configured to not accept reverse CoA packets and the next hop server is configured to send them, the client will silently discard these packets as per [RFC2865], Section 3. In both of those situations, reverse CoA packets will not flow, but there will be no other issues with this misconfiguration.

5. Reverse Routing

In normal RADIUS proxying. the forward routing table uses the User-Name attribute (via the Network Access Identifiers (NAIs) [RFC7542]) to map realms to next hop servers. For reverse CoA, [RFC8559], Section 2.1 uses the Operator-Name attribute to map operator identifiers to next hop servers.

This specification extends the [RFC8559], Section 2.1 reverse routing table to allow the next hop to be found via an open TLS connection, rather than a destination hostname or IP address. A server which needs to send reverse CoA packets to clients maintains a list of open TLS connections from clients. It also associates both a reverse CoA capability, and one or more operator identifiers with each connection.

A server MUST support associating one operator identifier with multiple connections. A server MUST support associating multiple operator identifiers with one connection. That is, the "operator identifier to connection" mapping is not one-to-one, or 1:N, or M:1, it is N:M or many-to-many.

This process occurs for all RADIUS proxies, except for the final one which sends the CoA packet to the client. That proxy forwards the reverse CoA packet to the client based on the Operator-NAS-Identifier attribute ([RFC8559], Section 3.4) and/or other NAS identification attributes such as NAS-Identifier, NAS-IP-Address, or NAS-IPv6-Address. The result is that there is a complete forwarding path from the home network back to the visited network.

5.1. Errors and Fail Over

When the server receives a reverse CoA packet, but cannot forward it, 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.

A server can also use RADIUS/UDP to send the reverse CoA packet; there is no requirement that all CoA packets use a "reversed" TLS connection.

After sending a packet, 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]. This specification permits reverse CoA packets to be sent on what would otherwise be a client to server TLS connection. It does not change the basic functionality of proxying CoA packets.

5.2. Retransmissions

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

RFC Editor: This section may be removed before publication.

This section records the status of known implementations of the protocol defined by this specification at the time of posting of this Internet-Draft, and is based on a proposal described in [RFC7942]. The description of implementations in this section is intended to assist the IETF in its decision processes in progressing drafts to RFCs. Please note that the listing of any individual implementation here does not imply endorsement by the IETF. Furthermore, no effort has been spent to verify the information presented here that was supplied by IETF contributors. This is not intended as, and must not be construed to be, a catalog of available implementations or their features. Readers are advised to note that other implementations may exist.

According to [RFC7942], "this will allow reviewers and working groups to assign due consideration to documents that have the benefit of running code, which may serve as evidence of valuable experimentation and feedback that have made the implemented protocols more mature. It is up to the individual working groups to use this information as they see fit".

6.1. FreeRADIUS

The FreeRADIUS project has implemented this specification in the v3.2.x (https://github.com/FreeRADIUS/freeradius-server/blob/v3.2.x) branch which is available on GitHub. The feature is not enabled by default, and requires a build flag WITH_COA_TUNNEL to be defined before the new functionality is included with the software.

Maturity: The implementation is at a "beta" level, but has been tested to work with other implementations.

Coverage: All of this specification is supported.

Version Compatibility: Earlier versions of this specification are not supported, but the current version is supported.

Licensing: GPLv2

Contact Information: http://freeradius.org/

Date: This information was updated May 2025.

6.2. Cisco

Cisco supports this specification as of Cisco IOS XE Bengaluru 17.6.1 via Vendor-Specific attributes. reference (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)

Maturity: The implementation is available in production.

Coverage: All of this specification is supported.

Version Compatibility: Earlier versions of this specification are not supported, but the current version is supported.

Licensing: Proprietary

Contact Information: http://cisco.com/

Date: This information was updated October 2022.

6.3. Aruba

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." reference (https://www.arubanetworks.com/techdocs/Instant_83_WebHelp/Content/Instant_UG/Authentication/ConfiguringRadSec.htm)

Maturity: The implementation is available in production.

Coverage: All of this specification is supported.

Version Compatibility: Earlier versions of this specification are not supported, but the current version is supported.

Licensing: Proprietary

Contact Information: http://hp.com/

Date: This information was updated October 2022.

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

This document requests no action from IANA.

10. Acknowledgements

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

11. Changelog

RFC Editor: This section may be removed before publication.

* 00 - taken from draft-dekok-radext-reverse-coa-01
* 01 - Bumped to avoid expiry
* 02 - Bumped to avoid expiry
* 03 - remove dynamic negotiation and cleanups
* 04 - shephards review
* 05 - tweak refs

12. References

12.1. 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/rfc/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/rfc/rfc2865>.

[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/rfc/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/rfc/rfc8559>.

12.2. Informative References

[I-D.ietf-radext-deprecating-radius] DeKok, A., "Deprecating Insecure Practices in RADIUS", Work in Progress, Internet-Draft, draft-ietf-radext-deprecating-radius-06, 25 May 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-radext-deprecating-radius-06>.

[I-D.ietf-radext-radiusdtls-bis] Rieckers, J. and S. Winter, "(Datagram) Transport Layer Security ((D)TLS) Encryption for RADIUS", Work in Progress, Internet-Draft, draft-ietf-radext-radiusdtls-bis-06, 27 May 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-radext-radiusdtls-bis-06>.

[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/rfc/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/rfc/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/rfc/rfc7360>.

[RFC7542] DeKok, A., "The Network Access Identifier", RFC 7542, DOI 10.17487/RFC7542, May 2015, <https://www.rfc-editor.org/rfc/rfc7542>.

[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/rfc/rfc7585>.

[RFC7942] Sheffer, Y. and A. Farrel, "Improving Awareness of Running Code: The Implementation Status Section", BCP 205, RFC 7942, DOI 10.17487/RFC7942, July 2016, <https://www.rfc-editor.org/rfc/rfc7942>.

Authors' Addresses

Alan DeKok
InkBridge
Email: alan.dekok@inkbridge.io

Vadim Cargatser
Cisco
Email: vcargats@cisco.com