RADIUS - UDP VS TCP

Document	2024-10-15A
Prepared by	Alan DeKok, CEO
Date	2024-10-15

InkBridge Networks

1. Introduction 3
1.1 Reasons to use UDP 3
1.2 Summary 4
2. UDP and TCP 5
2.1 UDP to TCP Proxying 5
2.2 Issues with RADIUS over TCP 5
2.3 Benefits of RADIUS over TCP 5
3. Users of RADIUS over TCP 7
3.1 EDUROAM 7
3.2 OpenRoaming 7
3.3 Private Proxies 7
3.4 Recommendations 7
4.1 Team 8

1. INTRODUCTION

In 1993, the RADIUS protocol was defined by the IETF to use the UDP protocol for data transport. This transport has proven to be sufficient for most needs, and is used world-wide in many situations.

Updated standards have allowed the use of RADIUS over TCP (RFC 6613), RADIUS over TLS (RFC 6614), and RADIUS over DTLS (RFC 7360). This white paper describes the differences between UDP and TCP as transport protocols for RADIUS.

For the purposes of this paper, we do not usually make any distinction between RADIUS over TLS and RADIUS over TCP. We are concentrating here on the behaviour of the network transport layer, and not on the security of the network.

1.1 Reasons to use UDP

RFC 2865 Section 2.4 describes why RADIUS was originally designed using UDP instead of TCP. We describe those, and other, reasons below.

Application Fail-over

In order for application fail-over to occur, implementations must save a copy of the original packet, and implement application-specific timeouts, retransmissions, and acknowledgements.

As RADIUS needs its own mechanisms to support this functionality, the guaranteed delivery and transport-layer retransmission of TCP is not as useful as they would be for another protocol such as HTTP.

Different Timing Requirements

The timing requirements for authentication are two-fold. One, a short delay for authentication is acceptable, based on human time frames. Second, the guaranteed delivery in TCP ensures that data arrives eventually, but not necessarily in a timely manner.

As such, the retransmission behaviour of TCP is unnecessarily quick. At the same time, the guaranteed delivery of TCP is unimportant, if application fail-over has already occurred.

That is, it is more useful for the RADIUS application to fail-over to a different server in human time frames (i.e. seconds), instead of TCP returning a guaranteed response minutes later.

Accounting Timeliness

Similar limitations apply to accounting packets. RFC 5080 Section 2.2.1 defines retransmission behaviour for RADIUS clients. If accounting packets have to be retransmitted, it is best to do that at the application layer (i.e. RADIUS), instead of at the transport layer (i.e. TCP). The main reason for this limitation is that accounting data changes over time.

That is, if we relied on the transport to perform retransmissions, then the RADIUS server would receive out of date information about user sessions. It is better instead for the RADIUS client to discard old accounting data, and to send retransmissions with updated accounting data.

Unconnected Sockets

The use of unconnected sockets in UDP can be a significant benefit. There is no need for the RADIUS client to track *connection* state as separate from the RADIUS server *application* state. That is, if a RADIUS server is responding to packets, then it is alive, independent of any particular connection state. If a RADIUS server is not responding to packets, then it is dead, even any underlying network connections are still open.

In some situations, TCP may indicate to the application that the connection is open, even if there are problems with the other application or an intermediate network node. Since the RADIUS client has to implement application fail-over itself anyways, the extra information obtained from connected sockets is not necessary, and is sometimes wrong.

Low Volume of Traffic

In many situations, RADIUS servers will see a low volume of traffic. For example, common ISP deployments will see tens to perhaps hundreds of packets per second. In enterprise environments, the packet rates will be even lower. Perhaps a maximum of tens of packets per second on the high end, with substantial amounts of time where there is no RADIUS traffic at all.

In those situations, TCP will be in a perpetual “slow start” scenario. This scenarios is where each packet causes TCPs network discovery and analysis to run, which negates the benefit of TCP.

Further, if the RADIUS client is sending a small number of packets to the RADIUS server, then each RADIUS packet ends up in one TCP packet. This correlation means that TCP is largely behaving as UDP, which further negates the benefit of UDP.

Intermediate Firewalls

In many networks, TCP connections pass through a stateful firewall. When the TCP connections are idle for extended periods, the firewall may discard it's knowledge of the connection. This discarding is done without informing the NAS or RADIUS server that the connection is no longer functional.

When the RADIUS client next sends a packet over that connection, the packet will be discarded by the firewall. The RADIUS client does not receive any indication that the connection is down, or that the packet was discarded. The RADIUS client must then rely on application timers to decide that the connection is down. These timers can introduce significant delays before a the existing connection is closed, and a new connection opened.

Once a new connection has been opened, the RADIUS client can retransmit the packet. However, depending on local client configuration, this delay may cause the NAS to give up on the packet.

The only work-around for this situation is to set the "connection dead" timers to be aggressively low, which can further destabilize the network.

Limitations of RADIUS

The RADIUS protocol limits itself to a maximum of 256 outstanding packets on any one connection, independent of the transport protocol. As discussed in RFC 2865 Section 2.5, any RADIUS client sending more than 256 packets is required to open a new connection.

Therefore, each TCP connection used for RADIUS data ends up carrying a small amount of traffic, entirely negating the bulk data transfer benefits of TCP.

RADIUS over TCP is being formally deprecated

The "deprecating insecure practices" document is formally deprecating RADIUS over TCP.

RADIUS over (D)TLS is preferred

For the long term, the "deprecating insecure practices" document is recommending that all systems move to using DTLS or TLS for RADIUS transport.

1.2 Summary

RADIUS has traditionally used UDP as the underlying transport protocol, for all of the reasons outlined above.

The ubiquitous and enduring nature of RADIUS over UDP demonstrates the usefulness and applicability of the protocol.

For nearly all RADIUS use-cases, TCP offers no benefit over UDP, and has many negatives.

2. UDP AND TCP

When a RADIUS system uses TCP as the transport, it creates issues related to UDP and TCP interaction. This section describes those issues

2.1 UDP to TCP Proxying

All NAS equipment implements RADIUS over TCP, along with the bulk of RADIUS proxies. Any system using TCP must therefore at some point accept RADIUS over UDP, and proxy it as RADIUS over TCP. There are, unfortunately, issues with this process.

The main issue is that TCP has congestion control, and is therefore limited in the amount of data it can send. When an application tries to send more data than a connection can handle, the application blocks, and the data is not sent. This process typically happens at the network layer, and is not necessarily visible to the application.

The result of this blocking is that the RADIUS client is unable to distinguish network issues from issues related to the RADIUS server itself. This limitation negatively affects application fail-over, as discussed earlier in this document.

2.2 Issues with RADIUS over TCP

This section describes issues when using RADIUS over TCP.

Head of Line Blocking

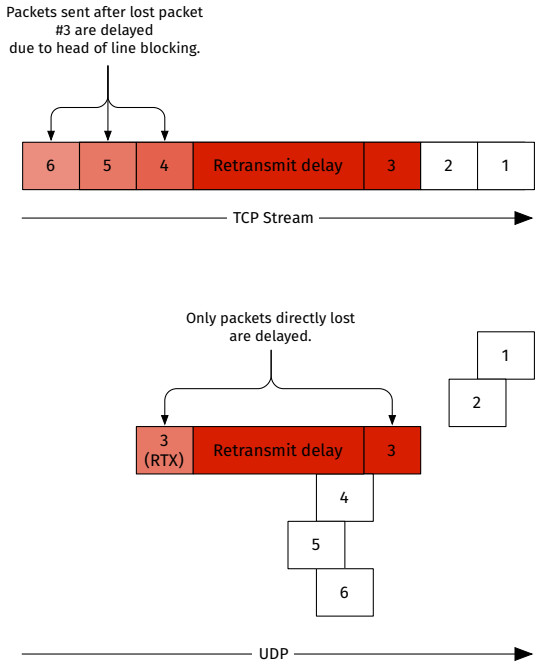
TCP has the “head of line” blocking issue, which UDP does not have. This issue is where a sent packet is lost in the network, and that loss causes subsequent packets to be delayed. In UDP, each packet is independent of others, so the loss of one packet has no immediate affect on subsequent packets.

Since each RADIUS packet is independent of every other RADIUS packet, the “in order” nature of TCP is a drawback.

Large traffic spikes

Another issue related to Head of Line Blocking is that a significant number of pending packets can build up in the outbound TCP buffers of the RADIUS client during poor network conditions. The main symptom of this situation is extreme load spikes on the receiving RADIUS server and on systems downstream of that server.

These traffic spikes are especially pronounced between RADIUS proxy servers processing high volumes of packets. These traffic spikes can cause major



disruption as proxies, databases, and even logging servers process the sudden increase in load.

While RADIUS over UDP will still have traffic spikes, this load is better under the control of the RADIUS application. With RADIUS over TCP, the operating system controls packet buffering and retransmissions. The result is a disconnect between the needs of the RADIUS application, and the operating system. This disconnect cause application and network instability.

Late retransmissions

In TCP, retransmissions are handled by the operating system, and not the application. The guaranteed delivery nature of TCP ensures that packets are sent, even after the RADIUS application determines that the packets are no longer relevant. The packets may be received by the server many seconds or even minutes after they have been sent by the client.

The data in these packets is stale or no longer relevant. These late packets therefore cause increased network load for no additional gain.

When the RADIUS client is in control of retransmissions, as with UDP, then the client can discard stale packets. No such capability is present with RADIUS over TCP.

2.3 Benefits of RADIUS over TCP

This section describes the benefits when using RADIUS over TCP. It also discusses drawbacks to using RADIUS over UDP

UDP Fragmentation

Sending large RADIUS packets, such as with EAP, can cause the packets to be fragmented at the IP/UDP layer. This fragmentation works in local networks, where the Access Point and RADIUS server are placed closely together. However, UDP fragmentation on the wider Internet is known to not work.

Therefore, it can be difficult to use RADIUS over UDP across the Internet. In contrast, the TCP transport layer will fragment application data into MTU-sized packets, which can traverse the wider Internet.

Similar issues apply when internal enterprise networks rely on multiple layers of encapsulation (e.g. VLANs and MPLS). The MTU at a switch or access point can be significantly larger than the MTU in the rest of the network. The result is that when EAP packets are sent over RADIUS, they may fit within the MTU at the edge, but will be fragmented through the rest of the network.

As UDP fragmentation is poorly supported by networking equipment, the main effect of this process is that packets appear to be randomly lost in the network. This loss leads to failed authentications, which can be difficult to debug.

Caveat

It is rare for networks to have significant differences in MTU across the corporate backbone.

RADIUS security is insufficient for sending packets across the Internet. The packets are sent largely in the clear. This means that any intermediate node can see the data. That is, they can see which users are logging in, when they log in, and where those users are logging in from.

In addition, it is a serious security risk to expose RADIUS servers to Internet traffic. Any inadvertent misconfiguration or software error could cause a security breach, including loss of all user data and passwords.

As a result, the industry best practice for the past twenty years has been to connect RADIUS servers at the network layer via IPSec, and then to send UDP traffic over the IPSec connection. This practice has worked well for every kind of RADIUS traffic, for both high and low volumes of traffic.

Where IPSec cannot be used, TLS must be used instead. The system authentication and verification can then happen at the TLS layer, before RADIUS is involved.

Cooperation with other Traffic

TCP is designed to have multiple cooperating flows. This scenario its where each flow gets an even proportion of the available network bandwidth. TCP uses congestion signalling and control in order to fairly divide bandwidth among active sessions.

In contrast, UDP has no such congestion signalling. A sender can completely overwhelm the network, resulting in congestive collapse.

Caveat

In practice, RADIUS traffic is low enough that congestive collapse isn't an issue. Where there is sufficient RADIUS traffic to fill a network, that traffic is important enough that RADIUS traffic is prioritized over all others.

That is, there is no reason to have "fair" distribution of traffic between RADIUS and subscribers/employees. The RADIUS traffic is almost always has a higher QOS priority than all other traffic.

The best practices in the RADIUS industry have been to use IPSec to connect disparate sites. The traffic prioritization can then be applied to management IPSec traffic.

Link State Monitoring

In some situations, TCP an give explicit signals that a link is down such as when TCP keep-alives are used. A RADIUS client using TCP transport can then detect the dead link, and retransmit packets earlier than if it just relied on application-layer timers.

Caveat

In practice, few NAS implement RADIUS over TCP (see above). Instead, the only users of RADIUS over TCP are proxies. The RADIUS standards suggest that these proxies do not perform retransmissions, and instead rely on retransmissions from the NAS.

The interaction of these two issues means that detecting link state on intermediate nodes has no effect on retransmission behaviour. Therefore, the "early detection" capabilities offered by TCP have little to no benefit in practice.

3. USERS OF RADIUS OVER TCP

The following examples show some use-cases for RADIUS over TCP.

3.1 EDUROAM

Eduroam is the largest user of RADIUS over TLS. The technical design of Eduroam and the rationale for using TCP are described in RFC 7593.

Notwithstanding earlier comments in this document, RADIUS over TLS is an appropriate solution for Eduroam. The traffic is mostly EAP authentication in small volumes, which means:

- UDP to TCP proxying issues are largely avoided
- Use of TCP avoids UDP fragmentation issues with respect to carrying large EAP packets in RADIUS
- delays of a few seconds are acceptable, and larger delays simply mean that the user is denied access, and must retry manually
- head of line blocking is less of an issue with EAP authentication, as EAP is an explicit request / response protocol that requires multiple round trips for one user session
- no accounting traffic means that all issues related to accounting are avoided

While the success of Eduroam is proven, it does not mean that RADIUS over TCP is appropriate for all possible situations.

3.2 OpenRoaming

The Wireless Broadband Alliance (WBA) uses RADIUS over TLS (and therefore TCP) for all interconnections. For this use-case, the connections are typically sending traffic for a large number of users, and sending regular accounting packets for those users. The result is that the issues noted above with low volumes of traffic for TCP do not apply.

3.3 Private Proxies

Some equipment vendors (e.g. WiFi providers) offer centralized hosted RADIUS solutions. In these solutions, the edge equipment that they sell to consumers contain RADIUS clients. These clients connect back to the main vendor RADIUS server over the Internet, in order to use a centralized RADIUS solution.

In this use-case, the only reliable protocol which can be used is TCP. Other network security protocols such as IPSec are often be blocked in intermediate nodes in the network.

In contrast, RADIUS over TCP can use port 443 (HTTPS), along with TLS encryption. Middleware boxes and firewalls will only see the destination port and the fact that TLS is being used. These middleware boxes will then pass the RADIUS traffic unchanged from the edge node to the central server.

This use-case is also similar to Eduroam, as it is using EAP for authentication, minimal or no accounting data, and it is passing a low volume of traffic.

3.4 Recommendations

Our recommendations are to use RADIUS over TCP only when all of the following conditions are met:

- The RADIUS traffic is largely EAP,
- The RADIUS traffic has to go over the public Internet, and IPSec cannot be used, and even then, TLS should be used instead of “raw” TCP,
- The volume of the traffic is more than a hundred packets per second.

In all other situations, we recommend using RADIUS over UDP. Inappropriate use of RADIUS over TCP and result in network and application instability.

4. Background

The CEO of InkBridge Networks is Alan DeKok, who has been actively working in the RADIUS space for over two decades. He was a technical reviewer for RFC 2865 (RADIUS) , RFC 5176, (CoA), and RFC 7593 (EduROAM), among other documents. He also many wrote a number of RADIUS standards, including many of the standards discussed in this document:

RFC 6613 - RADIUS over TCP

RFC 8044 - Data Types in RADIUS

RFC 7542 - The Network Access Identifier

RFC 7499 - Support of Fragmentation in RADIUS Packets

RFC 7360 - DTLS as a Transport Layer for RADIUS

RFC 6929 - RADIUS Protocol Extensions

RFC 6158 - RADIUS Design Guidelines

RFC 5997 - Use of Status-Server in the RADIUS Protocol

RFC 5080 - Common RADIUS Implementation Issues and Suggested Fixes

4.1 Team

The rest of the InkBridge Networks team each has between ten (10) and twenty (20) years of experience with RADIUS. Each member has designed, built, and maintained high volume RADIUS servers which handle millions of users, and thousands of packets per second. These systems typically involve proxying to disparate destinations, including destinations across the wider Intern

5. Contact Information

InkBridge Networks
26 rue Colonel Dumont
38000 Grenoble
FRANCE

T +33 4 85 88 22 67
F +33 4 56 80 95 75
W http://inkbridgenetworks.com
E sales@inkbridgenetworks.com

InkBridge Networks (Canada)
100 Centrepointe Drive, Suite 200
Ottawa, ON, K2G 6B1
Canada

T +1 613 454 5037
F +1 613 280 1542

InkBridge Networks
We authenticate the Internet