BLASTRADIUS VENDOR GUIDE

VU#456537

Alan DeKok
aland@inkbridgenetworks.com
CEO, InkBridge Networks, Project Leader, FreeRADIUS
July 9, 2024

1. THE PRACTICAL IMPACT OF VU #456537

1.1. Introduction

After more than thirty years, the first attack on RADIUS[1] has been published. This white paper summarizes the attack, describes the impact of the attack, and suggests methods by which administrators and equipment vendors can protect their networks.

We refer to the original paper for technical description of the attack, including details of the cryptographic methods used. However, that description is largely intended for a cryptographic and research community. This document offers additional insight for implementors of RADIUS clients and servers.

That insight is valuable in the short term for implementors. While the RADIUS standards will eventually be updated to address these issues, that process takes time. We expect that this document will be useful during most of 2024, until the IETF issues a new standard based on the RADEXT working group document "Deprecating Insecure Practices in RADIUS"[2].

This short note gives recommendations which allow vendors and network administrators to quickly mitigate the attack, and to protect themselves from it. It also gives guidance for vendors and network administrators on how those mitigation steps may interact with legacy networking equipment.

1.1.1 Don't Panic

While this attack sounds worrying, there are a few simple steps which can be taken to mitigate it.

The most important step for network operators to take is that they should install updated software when it is provided by the vendors. Operators should also not send RADIUS/UDP packets over the Internet.

Vendors who include RADIUS clients in their products should update their software to include a Message-Authenticator in all Access-Request packets. RADIUS clients should also have a configuration flag which requires a Message-Authenticator in replies to all Access-Requests. These changes are necessary but not sufficient, to stop the attack, but also RADIUS servers also need to be updated.

RADIUS server vendors should update their software to add a Message-Authenticator attribute as the *first attribute* in all replies to Access-Request packets. There are also a few configuration flags and behavior changes needed, which are discussed in more detail below.

1.2. Terminology

The following terms are used in this document:

RADIUS - The protocol as defined in various IETF standards.

RADIUS/UDP - RADIUS over UDP as defined in RFC 2865[3].

RADIUS/TCP - RADIUS over TCP as defined in RFC 6613[4]

RADIUS/TLS - RADIUS over TLS as defined in RFC 6614[5] and updated in TLSbis[6].

RADIUS/DTLS - RADIUS over DTLS transport as defined in RFC 7360[7]

In this document, when we refer to "using TLS" or "TLS transport", we mean either TLS or DTLS transport.

1.3. Background

The RADIUS protocol was first standardized in RFC 2058 in 1997. The use-case for RADIUS is to control network access via authentication, authorization, and accounting (AAA). RADIUS is in wide-spread use, and is supported by essentially every switch, router, access point, and VPN concentrator product sold in the past twenty-five years. All of those devices are likely vulnerable to this attack. The only unaffected devices are those which have only a simple web administration interface and no RADIUS functionality, or "dumb" un-managed devices which have no administration interface at all.

The key to the attack is that in many cases, Access-Request packets have no authentication or integrity checks. An attacker can then perform a chosen prefix attack, which allows modifying the Access-Request in order to replace a valid response with one chosen by the attacker. Even though the response is authenticated and integrity checked, the chosen prefix vulnerability allows the attacker to modify the response packet, almost at will.

We note that the attack is due to a fundamental design flaw of the RADIUS protocol. It is not a flaw in any particular implementation or product. All standards compliant RADIUS clients and servers are likely vulnerable to this attack, even if they correctly implement all aspects of the RADIUS protocol.

There are a few mitigating factors which prevent the complete meltdown of the world-wide RADIUS infrastructure. Some are due to the nature of the attack, others are due to some common uses of RADIUS (e.g. WPA enterprise) being more secure than PAP / CHAP / MS-CHAP.

The first mitigating factor is that the attacker must be able to both observe and modify packets in transit. This requirement means that most equipment which is physically secure is not (or perhaps is much less) vulnerable to the attack. The second is that most public transport of RADIUS over the Internet is done via TLS, which provides complete protection against the attack. Another mitigating factor is that many common uses of RADIUS (802.1X, WPA enterprise) require the use of increased packet security via the Message-Authenticator attribute, which prevents the attack from succeeding.

The final mitigating factor is that there are only a few minor changes required to implementations of RADIUS/UDP which will stop the attack. These changes are simple to make, and allow for interoperability with legacy (i.e. non-upgraded) RADIUS systems.

This attack is the result of the security of the RADIUS protocol being neglected for a very long time. While RADIUS depends on MD5 digests for its security, MD5 has been broken since 2004. However, until now, those attacks have not been shown to be applicable to RADIUS. While the standards have long suggested protections which would have prevented the attack, those protections were not made mandatory. In addition, many vendors did not even implement the suggested protections.

The positive outcome of this attack is that those long-term failures of the RADIUS protocol will now be addressed.

1.4. The Vulnerability

The RADIUS protocol defines a “Request Authenticator” field in the packet header. RFC 2865 Section 3 describes this field as containing random data for Access-Request packets, and provides for no other way for Access-Request packets to be integrity checked.

It is therefore possible to send standards-compliant Access-Request packets which lack all integrity protection. These packets can be trivially forged or modified in transit. In this issue, an attacker modifies the Access-Request such that the response from the server is subject to a known prefix attack.

A known prefix attack on MD5 means that an attacker can cause two pieces of data “A” and “B” to have the same MD5 hash. Where MD5 is used for integrity checks as with RADIUS, an attacker can swap data “A” which is not under the attackers control for data “B” which is under the attackers control. The resulting output containing data “B” will pass all of the integrity checks for data “A”, meaning that the receiver will believe that the data has been unmodified. The receiver will then proceed to use the data which is under the attackers control, instead of the original data which was intended to be received.

1.4.1 Exploiting the issue in RADIUS

The known prefix attack requires that the RADIUS response packet be composed of the form of a known or predictable prefix, followed by an unknown or unpredictable suffix. The construction of the Response Authenticator uses the contents of the response as the known prefix, and the shared secret as the unknown suffix, as follows:

MD5(packet + secret)

where “+” denotes concatenation. We refer the reader to RFC 2865 Section 3 for a detailed technical explanation of the process by which RADIUS packets are checked for integrity.

It is important here to note that not all of the contents of the packet needs to be predictable as a known prefix. The packet can contain unknown and unpredictable attributes, so long as those attributes are at the end of the packet. The format of the attack allows the attacker to either hide those unpredictable attributes, or to pass them through unmodified.

For example, many RADIUS servers response with packets containing Message-Authenticator as the last attribute in the response. While the value of the Message-Authenticator is unknown and unpredictable, the attack can proceed with only minor changes to the attackers behavior.

1.4.2 All replies to Access-Request are vulnerable

It is important to note that while the description below uses Access-Reject, the attack is not limited to Access-Reject packets. The vulnerability is because the servers response contains a “known prefix”. It does not matter whether this prefix is Access-Accept, Access-Reject, Access-Challenge, or Protocol-Error. It does not matter if the prefix contains any number of attributes, so long as the attribute order and their contents are predictable.

Access-Accept packets are vulnerable to this attack. An attacker can log in with a known user and known password, get an Access-Accept, and change that response to obtain additional (i.e. invalid) authorization.

Access-Challenge packets are vulnerable to this attack. If the server is configured to do Multi-Factor Authentication (MFA), then the typical Access-Challenge is either empty, or else it has has predictable contents. The attacker can rewrite the Access-Challenge to Access-Accept, and bypass MFA entirely.

1.4.3 Securing One Hop is Not Enough

It is not enough to secure the communication between a RADIUS client and a RADIUS server. Many RADIUS servers support proxying packets to a remote or "home" server, which means that RADIUS proxies are also clients. While a particular RADIUS proxy may take measures to protect itself from this attack, it may also send packets to other servers which are still vulnerable.

All RADIUS security is defined "hop by hop", and provides for no "end to end" security. That is, the mitigation methods outlined in this document protect communication between only one client and server. The result is that in a multi-hop proxy chain, the existence of one vulnerable server is enough for an attack to succeed.

A similar analysis applies to partial mitigations which protect only one side of a client to server connection. That is, a proxy could take steps to protect packets it sends to home servers, and replies that it sends to clients. However, an attacker who can modify packets on one client to server path could also modify packets on two client to server paths.

The attack could then succeed if the attacker modifies packets sent to the proxy, which the proxy will forward to the home server. The attacker can then modify packets from the home server to the proxy, which would result in a successful exploitation of the vulnerability.

While have discussed proxies here, we note that RADIUS clients must also be upgraded. We recognize that this is not always possible, as operators may be using equipment where vendors no longer provide firmware updates. The mitigation methods outlined below for RADIUS servers take this limitation into account, and can protect systems even if the client cannot be upgraded.

The outcome of the above analysis leads to the inescapable conclusion that the only way to prevent this attack globally is for all RADIUS servers world-wide to be upgraded, and configured with the new mitigation methods outlined here.

1.4.4 Compatibility with existing systems must be maintained

The mitigation methods outlined below allow for continued operation of existing networks, while at the same time ensuring that those networks to be protected. That is, there is no "flag day" where all RADIUS equipment has to be upgraded all at once.

Administrators are free to update clients or servers individually, and at different times. Once administrators have verified that the mitigation methods are supported by networking equipment, they can set configuration flags to enforce the new behavior, or to protect legacy equipment. The configuration flags also prevent "down-bid" attacks where systems are downgraded to a less secure behavior.

1.4.5 Impact of the Vulnerability

Since all security of the RADIUS protocol for UDP and TCP transports is based on the shared secret, this attack is perhaps the most serious attack possible on the RADIUS protocol.

At the absolute minimum, every single RADIUS server world-wide must be upgraded to address this vulnerability. It is not sufficient to upgrade only RADIUS clients, as doing so will allow the network to remain vulnerable.

1.5. Performing the Attack

We give a high-level summary of the attack here. For technical details, please see the original paper describing the attack.

The attack requires that the attacker is able to both view, and modify RADIUS packets in transit. While this requirement limits the applicability of the attack, it is nevertheless a serious, and real vulnerability of the RADIUS protocol which needs to be fixed.

At a high level, the attack depends on injecting one or more Proxy-State attributes with special contents into an Access-Request packet. The Proxy-State attribute itself will not trigger any overflow or "out of bounds" issue with the RADIUS client or server. Instead, the contents of the attributes will allow the attacker to create an MD5 collision when the server calculates the Response Authenticator. In effect, the attacker uses the RADIUS server, and its knowledge of the shared secret, to unknowingly authenticate packets which it has not created.

The behavior of the Proxy-State attribute is extremely useful to this attack. The attribute is defined in RFC 2865 Section 5.33 as an opaque token which is sent by a RADIUS proxy, and is echoed back by RADIUS servers. That is, the contents of the attribute are never examined or interpreted by the RADIUS server. Even better, testing shows that RADIUS clients will simply ignore any unexpected Proxy-State attributes which they receive. This attribute is therefore ideally suited to an attackers purpose of injecting arbitrary data into packets, without that data affecting client or server behavior.

While it is possible to use other attributes to achieve the same effect, the use of Proxy-State is simple, and sufficient to trigger the issue.

The injected data and resulting MD5 collision allows the attacker to modify the packet contents almost at will, and the client will still accept the modified packet as being authentic. The attack allows nearly arbitrary attributes to be added to the response. Those attributes are simply part of the MD5 collision calculation, and do not increase the cost of that calculation.

Again, since the RADIUS server can be convinced to authenticate packets using a prefix chosen by the attacker, there is no need for the attacker to know the shared secret.

The attack is implemented via the following steps, which are numbered the same as in the original paper.

1. The attacker requests network access from the RADIUS client (NAS). This action triggers the NAS to send an Access-Request packet to the RADIUS server.
2. The Access-Request is observed to obtain its contents, including the Request Authenticator field. The attacker prevents this packet from reaching the server until the MD5 collision data has been calculated.. The NAS will retransmit the packet one or more times after a delay, giving the attacker time to calculate the chosen prefix.
3. Some external resources are used to calculate an MD5 collision using the Request Authenticator, and the expected contents of an Access-Reject. As Access-Reject packets are typically empty (or can be observed), the expected packet contents are known in their entirety.
4. Once an MD5 collision is found, the resulting data is placed into one or more Proxy-State attributes in the previously seen Access-Request. The attacker then sends this modified Access-Request to the RADIUS server.
5. The RADIUS server responds with an Access-Reject, and includes the Proxy-State attributes from the modified Access-Request packets.
6. The attacker discards the original Access-Reject, and uses the chosen prefix data to create a different (i.e. modified) response, such as an Access-Accept. Other authorization attributes such as VLAN assignment can also be add, modified, or deleted.
7. The NAS receives the modified Access-Accept, verifies that the Response Authenticator is correct, and gives the user access, along with the attackers desired authorization.

A sequence diagram of the attack is given on the next page.

At a conceptual level, the attack leverages the following identity:

> MD5(Access-Reject + Proxy-State + secret) == MD5(Access-Accept + attacker attributes + Proxy-State + secret)

where the attacker does not know the "secret". We refer the reader to the original paper for a full technical explanation of the cryptographic details.

The result of this attack is a near-complete compromise of the RADIUS protocol. The attacker can cause any user to be authenticated. The attacker can give almost any authorization to any user.

While the above description uses Access-Reject replies, we reiterate that the root cause of the vulnerability is in the Access-Request packets. The attack will therefore succeed even if the server replies with Access-Accept, Access-Challenge, or Protocol-Error.

It is therefore critical that all RADIUS implementations be updated immediately to address this protocol vulnerability.

1.5.1 Almost Two Decades of Ignored Mitigations

The issue of Access-Request packets lacking integrity checks was noted in 2003 in Section 4 of RFC3579, and again in 2007 in Section 2.2.2 of RFC 5080[8], which states in part about Access-Request forgery:

> To avoid this issue, server implementations may be configured to require the presence of a Message-Authenticator attribute in Access-Request packets. Requests not containing a Message-Authenticator attribute MAY then be silently discarded.
>
> Client implementations SHOULD include a Message-Authenticator attribute in every Access-Request to further help mitigate this issue.

To reiterate this point in a different manner, it has been known for two decades that this vulnerability exists. A method to prevent the attack has been recommended for 17 years. There are therefore few reasons why those recommendations have not been widely implemented. The failure to implement these recommendations has contributed significantly to the effectiveness of this attack.

To our knowledge, the only RADIUS server which implemented the recommendations of Section 2.2.2 of RFC 5080[9] was FreeRADIUS. The server was initially updated to have a configuration option which added Message-Authenticator to all proxied Access-Request packets. Then in 2012, the server was again updated to always add Message-Authenticator to all proxied Access-Request packets.

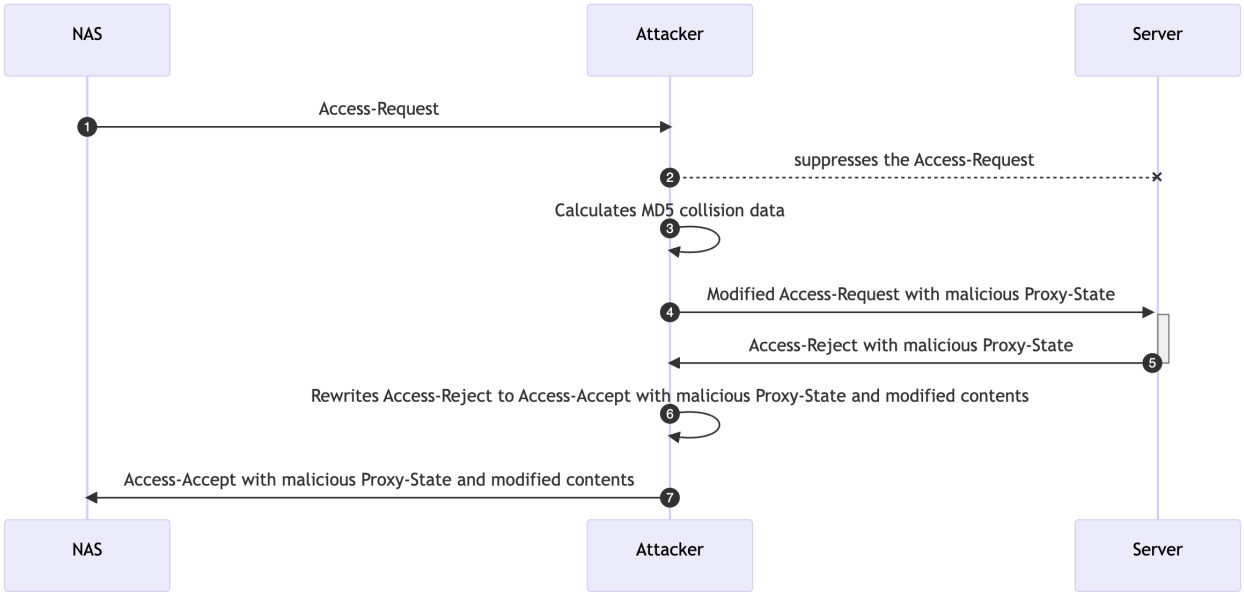

Figure1. Sequence diagram illustrating the attack.

The server also has a per-client configuration which will cause it to discard Access-Request packets which do not contain Message-Authenticator.

There have been no known interoperability issues resulting from the above behavior. This decade of experience with mitigations gives us a high degree of confidence that the mitigations outlined here will not just work, but will cause minimal issues in operational networks.

The issue of Access-Request packets lacking authentication and integrity checks was most recently discussed in another work in progress document in the IETF RADEXT working group, which states:

> This document therefore requires that RADIUS clients MUST include the Message-Authenticator in all Access-Request packets when UDP or TCP transport is used.

That recommendation was intended as a proactive measure to prevent attacks which were believed to be theoretical. The attack is now proven, and is practical for attackers who are willing to spend relatively low amounts of money, i.e. perhaps thousands to tens of thousands of dollars.

Such amounts are well within the budget of someone who has access to stolen credit card data, or nefarious corporate actors. Such amounts are minor rounding errors in the budget of nation-states.

The Access-Request forgery issue was generally believed to be at least somewhat mitigated by the requirement that Access-Request packets contain valid authentication credentials. For example, those credentials can be PAP, CHAP, MS-CHAP, or EAP. The original idea appears to be that if the user is authenticated, the rest of the packet contents can likely be trusted.

That is, the belief was that even if the Access-Request packets were modified in transit, it would not matter because the RADIUS server still had to authenticate the user, and send authorization attributes in the Access-Accept. Since the Access-Accept packets are authenticated, they could not be modified, and the RADIUS ecosystem was seen as being secure.

That underlying assumption is not true, as modified Access-Request packets can be used to attack the replies, which means that the authorization attributes are under the attacker's control. This modification of the Access-Accept packets is the vulnerability which the attack exploits.

1.6. UDP versus TCP

This attack has been demonstrated for RADIUS/UDP. While it is technically possible to implement the attack for RADIUS/TCP, the nature of TCP makes the attack more difficult, but not impossible. While the attacker must inject data into the middle of a TCP steam, data injection attacks on TCP have been known to be feasible for a long time. There was initially an attempt to address TCP packet

integrity and authentication in RFC2385, but the chosen method also uses MD5, and is therefore subject to the same chosen prefix attack outlined here. Unlike RADIUS, however, the MD5 based authentication for TCP was deprecated in 2010 in RFC 5925., and replaced with a stronger authentication method.

However the TCP stream is modified, the cost of performing the attack for TCP is only marginally higher than the cost for UDP. The attacker has to modify TCP packets and track TCP state, but has no additional cryptographic work to perform, which is the bulk of the effort.

We refer the reader to the above documents for further discussion of TCP issues. For the purposes of this document, we will concentrate on UDP, and assume that all of the attacks on RADIUS/UDP are also applicable to RADIUS/TCP, albeit with only minor changes.

1.7. The Limitations

We repeat that a successful attack requires that the attacker is able to both view, and modify RADIUS packets in transit. As such, the attack is limited to situations where the attacker either has physical access to the network equipment and cables, or can redirect the traffic to the attackers network as with DHCP spoofing, BGP hijacking, or if the attacker has administrator access to the network equipment such as a switch or router which forwards the UDP packets.

These limitations mean that the attack is at least somewhat difficult to do in practice. When coupled with the need for large amounts of CPU power to calculate the collisions, the attack can be best described as “not trivial”. i.e. there is usually no way for an unskilled attacker to run pre-packaged scripts to attack a network. The attack requires a somewhat higher level of skill.

While those requirements mitigate the severity of the attack in most situations, there are many networks which do not enforce sufficient physical security.

For example, some organizations may not place network equipment in secure locations. Attackers who have physical access to network equipment will have complete control over it, and the RADIUS attack outlined here will perhaps be the least effective of all possible attacks.

Even if that equipment is physically secured, the cables between network devices are often not physically secured. An attacker could perhaps cut the cables, and splice in new and malicious networking equipment which could implement the attack.

As with the above issue of physical access to network equipment, however, if an attacker can cut and splice networking cables, there is less utility in performing this attack. The attacker can instead simply inject any desired packet into the network, at will. The only method to prevent such as “cut and splice” attack is to ensure that all connections between networking equipment are authenticated and encrypted, such as with MACsec.

If the RADIUS client obtains its IP address via DHCP, then an attacks such as CVE-2024-3661 can potentially be performed by the attacker. The attacker could use DHCP packets and Option 121 to cause all of the RADIUS traffic from the client to be routed to a system under the attackers control.

Another limitation of this attack is that the attacker must first see a RADIUS packet in order to calculate the MD5 collision. However, as Access-Request packets are generated when a user logs in, the timing of those packets is usually under the complete control of the attacker.

The attack is also possible only for a limited time duration, as most RADIUS clients will expect to see a response within a short period of time. RFC 5080 Section 2.2.1 suggests a time limit of thirty (30) seconds, and we have found that most NAS equipment follows that recommendation. This time is sometimes extended to the need for multi-factor authentication (MFA), but it is rare to have timeouts longer than sixty (60) seconds.

In many cases, there is insufficient local CPU power available to the attacker to perform the necessary MD5 calculations. The packets must then be exfiltrated from the local network, the calculations done externally, and then the resulting data returned to the network. These necessary steps increase the cost of the attack, and the likelihood of detection.

We note that some portions of the RADIUS protocol are not affected by this attack. Specifically, the contents of obfuscated attributes such User-Password and Tunnel-Password are still believed to be secure, as no attacks have been demonstrated for those attributes. We caution the reader to not rely on this alleged security for too long. The constructs used to obfuscate those attributes are do not rely on modern cryptographic methods, and should be regarded with skepticism.

Similarly, all other RADIUS request packets such as Accounting-Request, CoA-Request, Disconnect-Request, and Status-Server contain mandated integrity checks, and cannot be modified without detection..

1.7.1 The attack will only get better with time

This attack was demonstrated using commodity hardware, and can calculate an MD5 collision in approximately five minutes. For this attack to succeed, the RADIUS client used was artificially configured to have a timeout of five minutes. While this configuration modification will not be possible on real-world NAS equipment, an attacker could simply throw more resources at the calculation, and reduce the time required significantly.

A “back of the envelope” calculation shows that the attack is possible within about ten (10) seconds, using at most a few thousand dollars worth of cloud computing power. Which means that an attacker with only modest resources could fit the attack within the necessary time window, without modifying the NAS configuration.

We believe that attacks on RADIUS/UDP are likely to appear, and the only lasting source of security is for all RADIUS systems to switch to using TLS for all traffic. That is, even traffic on a local LAN should be using TLS or DTLS.

1.8. RADIUS/UDP and the Internet

We now discuss the practice of sending RADIUS/UDP traffic over the Internet. This practice has been known to be insecure since almost the beginning of RADIUS. Such practice is essentially giving dozens, if not hundreds of unknown parties the ability to see and modify all of the RADIUS traffic.

This insecure practice is especially problematic when the NAS is in an ISPs network, and the ISP acts as a wholesaler to other ISPs. In this case, the NAS will often function as a Broadband Remote Access Server (BRAS), and have the capability to forward subscriber traffic to a tunnel server such as an L2TP Network Server (LNS). Although L2TP requires the Tunnel-Password attribute which is protected by the shared secret, other tunnel types such as IP-in-IP and GRE do not. For those protocols, the attacker can modify both the Tunnel-Type and Tunnel-Server-Endpoint attributes, and instruct the BRAS to open a tunnel. The BRAS will then establish an unauthenticated connection to a server controlled by the attacker. The attacker would then have complete access to the subscriber's internet traffic.

That is, this attack not only allows an attacker to gain unauthenticated and unauthorized access to networks, it can allow the attacker to gain complete control over the network traffic of other users on the local network.

Additionally, many modern Broadband Network Gateways (BNG)s, Wireless Lan Controllers (WLCs), and BRASs support configuring a dynamic HTTP redirect using Vendor Specific Attributes (VSA)s. These VSAs are not protected by the shared secret and could be injected into an Access-Accept by an attacker. The attacker could then setup a malicious website to launch Zero-Day/Zero-Click attacks, driving subscribers to the website using a HTTP redirect. This issue is compounded by the fact that many devices perform automatic HotSpot 1.0 style walled garden discovery. The act of simply connecting to their home WiFi connect could be enough to compromise a subscriber's equipment

In the short term, sending RADIUS/UDP traffic over the Internet is believed to be secure from this attack when the Access-Request packets include the Message-Authenticator attribute, and the RADIUS server drops all Access-Request packets which are missing a Message-Authenticator attribute. However, RADIUS/UDP packets should still never be sent over the Internet when they contain PAP, CHAP, or especially any variant of MS-CHAP. In contrast, most EAP methods such as EAP-TLS, PEAP, TTLS, SIM, AKA, and EAP-pwd, are currently secure from this attack, even if they still unnecessarily expose private user information.

This topic is discussed at length in an Internet draft in the IETF RADEXT working group. We recommend that readers refer to that document for a more in-depth discussion of RADIUS privacy and security. We expect that document to become a full RFC in late 2024.

1.9. Other Packet Codes

Other request packets such as Accounting-Request, CoA-Request, and Disconnect-Request are authenticated with the Request Authenticator field, which contains an MD5 digest of the packet plus the shared secret. That authentication / integrity check is still cryptographically vulnerable, but the conditions required to exploit it in the real world are much harder to fulfill than the attack outlined here.

Since the packets cannot be modified without detection, it is difficult for an attacker to perform a chosen prefix attack on these packets. Instead of being able to insert arbitrary data into a packet, the attacker must use normal authentication processes, which means that only a few fields (e.g. User-Name) in the packet are under the control of the attacker. In addition, those fields are often required to have a particular format, further limiting the attack surface. The attacker also cannot always control when those packets are sent, which means that the attack cannot be performed “on demand” by the attacker.

However, some RADIUS servers may echo back attacker-supplied data in an Access-Reject, such as a Reply-Message attribute which contains a copy of the User-Name. For

example, systems could include a Reply-Message saying "Sorry, *User-Name*, you are rejected". If the "User-Name" portion here is a copy of the User-Name supplied by the attacker, there may still be a vulnerability.

The vulnerability is possible with any attributes which are echoed back to the NAS in some form by the RADIUS server. The attack here uses Proxy-State simply because it is common, and its defined behavior is ideally suited to create an MD5 chosen prefix vulnerability. RADIUS server implementors and network administrators should, in general. not echo user-supplied data back to the user in an Access-Accept, Access-Reject, or Access-Challenge.

Packet such as Accounting-Request packet will also contain a unique and typically difficult to guess session identification attribute: Acct-Session-Id. Even once an attacker observes a particular value for Acct-Session-Id, the contents of any Accounting-Request packet will change depending on internal counters managed by the NAS, which are not under the attackers control.

The only remaining RADIUS packet type which is in common used is Status-Server (RFC 5997). While that packet is defined to calculate the Request Authenticator in the same manner as is done for Access-Request packets, Section 2 of RFC 5997 mandates that all Status-Server packets "MUST" contain a Message-Authenticator attribute. When implementations follow this specification, the attack is not possible.

All RADIUS response packets (Access-Accept, Accounting-Request, etc.) are authenticated with the Request Authenticator and the shared secret. So long as the corresponding request packet cannot be modified, the response packet cannot be modified either. However, for reasons outlined below, we still recommend that all Access-Accept, Access-Reject, and Access-Challenge packets contain a Message-Authenticator attribute.

We reiterate that the reliance on MD5 is still problematic. We do not claim here that other packets are secure, only that they have no yet been proven to be insecure. The only long-term solution is to move to using TLS.

1.10. The Weakest Link

As noted earlier, RADIUS security is done on a "hop by hop" basis. If the packets are passed through one or more proxies, then any one vulnerable proxy will allow the attack to take place.

If proxies must be used, every single hop in the proxy chain must be verified to follow the highest level of security, otherwise all security will be lost.

Even worse, proxies have full control over packet contents. A malicious proxy can change a reject into an accept, and can add or delete any authorization attributes it desires. While proxies are generally part of a trusted network, there is every benefit in limiting the number of participants in the RADIUS conversation.

Proxy chains should therefore be avoided where possible, and RFC 7585 dynamic discovery should be used where possible. RADIUS clients and servers should also be configured with static IP addresses, and static routes. This configuration protects them from DHCP related attacks as discussed earlier.

1.11. Vulnerable Systems

A RADIUS server is vulnerable to the attack if it does not require that all received Access-Request packets contain a Message-Authenticator attribute. This vulnerability exists for many common uses of Access-Request, including packets containing PAP, CHAP, MS-CHAP, or packets containing "Service-Type = Authorize-Only". The vulnerability is also transitive. If *any* RADIUS server in a proxy chain is vulnerable, then the attack can succeed, and the attacker can gain unauthenticated and/or unauthorized access.

Simply having the Message-Authenticator attribute present in Access-Request packets is not sufficient. The server must require that the attribute is present, and discard packets where it is missing. Similarly, the client should also require that the attribute is present, and discard packets where it is missing.

In the short term, we believe that is is most important to upgrade all RADIUS servers, as there are many fewer RADIUS servers deployed than RADIUS clients. There are also many fewer RADIUS server implementations than RADIUS client implementations. Once all of the RADIUS servers are updated as described here, systems will be more secure. However, in order to provide a robust defence in depth, all RADIUS clients must also be updated. The attack is fully mitigated only when both sides of the RADIUS conversation are updated and configured correctly.

1.12. Unaffected Systems

There are a number of systems which are not vulnerable to this attack. The most important ones are systems which only perform EAP authentication, such as with 802.1X / WPA enterprise. The EAP over RADIUS protocol is defined in RFC 3579, and Section 3.3 of that document states explicitly:

> If any packet type contains an EAP-Message attribute
> it MUST also contain a Message-Authenticator.

This requirement is enforced by all known RADIUS servers. As a result, when roaming federations such as eduroam use RADIUS/UDP, it is not possible for the attacker to forcibly authenticate users, but it may be possible for the attacker to control the authorization attributes for known and valid users.

Other roaming groups such as OpenRoaming require the use of TLS, and are not vulnerable. Other roaming providers generally use VPNs to connect disparate systems, and are also not vulnerable.

802.1X / WPA enterprise systems have an additional layer of protection, due to the use of the master session keys (MSK) which are derived from the EAP authentication. method These keys are normally carried in the MS-MPPE-Recv-Key and MS-MPPE-Send-Key attributes in the Access-Accept packet. The contents of the attributes are obfuscated via the same method used for Tunnel-Password.

While an attacker can perhaps force an Access-Accept in some situations, or strip the Message-Authenticator from packets, it is not currently possible for an attacker to see, modify, or create the correct MSK for the EAP session. As a result, when 802.1X / WPA enterprise is used, even a successful attack on the Access-Accept packet would not result in the attacker obtaining network access.

2. ACTIONS TO TAKE

2.1. What Everyone Should Do

The outcome of this issue is a number of recommended changes which everyone using RADIUS should enact. The first change is:

Do not use RADIUS/UDP.

Administrators should use RADIUS/TLS or RADIUS/DTLS, instead of RADIUS/UDP. RADIUS/UDP is acceptable only on local networks, i.e. for one hop between the NAS and initial server. After that hp, the use of RADIUS/TLS is highly preferred.

Do not use RADIUS/TCP.

Administrators should use RADIUS/TLS or RADIUS/DTLS instead of RADIUS/TCP. TCP transport is experimental, and offers no benefit over UDP or TLS.

All RADIUS traffic sent over the Internet should be secured with TLS or IPSec.

This attack is not possible if TLS transport is used. The use of TLS prevents the attack even if the attacker has physical access to the network. The use of TLS also protects users from all of the privacy issues raised in the "deprecating insecure practices" document.

It is also safe to use a VPN to connect two sites, and then use RADIUS/UDP over the VPN. However, TLS transport is still preferred for reasons outlines in the "deprecating insecure practices" document.

Physically secure all networking equipment.

An attacker who has physical access to networking equipment in effect owns and controls the equipment.

All RADIUS traffic should use a management VLAN.

There is no reason for management traffic to use the same VLAN as user traffic.

All RADIUS should be updated and configured with mitigation methods when UDP or TCP transport is used.

If it is not possible to upgrade clients, then at the minimum all RADIUS servers must be updated and configured as discussed below. No other course of action will protect systems from the attack.

The following sections describe recommended changes for RADIUS clients and servers. These behaviors and configuration flags should only be applied when RADIUS/UDP or RADIUS/TCP transport is used.

2.2 Implementation Changes

There are a number of changes required to both clients and servers in order for all possible attack vectors to be closed. Implementing only some of these mitigations means that an attacker could bypass the partial mitigations, and still perform the attack.

This section outlines the mitigation methods which protect systems from this attack, along with the motivation for those methods. A more prescriptive description of the mitigation methods is outlined in the next section.

We note that unless otherwise noted, the discussion here applies only to Access-Request packets, and to Access-Accept, Access-Reject, Access-Challenge, and Protocol-Error packets. All behavior involving other request and response packets must remain unchanged.

Similarly, the recommendations in this section only apply to UDP and TCP transport. They do not apply to TLS transport, and no changes to TLS transport are needed to protect from this attack. Clients and servers must not apply any of the new configuration flags to packets sent over TLS or DTLS transport. Clients and servers may include Message-Authenticator in all Access-Request packets and in responses to those requests which sent over TLS or DTLS transports, but this change is not recommended for those transport protocols.

We recognize that implementing this functionality may require a significant amount of effort. There is a substantial amount of work to perform in updating implementations, performing interoperability tests, changing APIs, changing user interfaces, and updating documentation. This effort cannot realistically be done in a short time frame.

There is therefore a need for an immediate and short-term action which can be implemented by RADIUS clients and servers which is both simple to do, and which is known to be safe. The recommendations in this section are known to protect implementations from the attack; to be simple to implement; and also to allow easy upgrade without breaking existing deployments.

The mitigation methods outlined here allow systems to both protect themselves from the attack, while not breaking existing networks. There is no global "flag day" required for these changes. Systems which implement these recommendations are fully compatible with legacy RADIUS implementations.

2.2.1 Clients must add Message-Authenticator to all Access-Request packets

The obvious solution to address the vulnerability is to simply mandate the suggestion of Section 2.2.2 of RFC 5080, and require that all clients (including proxies) must send Message-Authenticator as the first attribute in all Access-Request packets..

This behavior should be the default, and it should not be configurable. Disabling it would open the system up to attacks, and would prevent the other mitigation methods from working. The root cause of the attack is that Access-Request packets lack integrity checks, so the most important fix is to add integrity checks to those packets.

From a cryptographic point of view, the location of Message-Authenticator does not matter, it just needs to exist somewhere in the packet. However, as discussed below for Access-Accept etc. packets, the location of Message-Authenticator does matter. It is better to have consistent and clear messaging for addressing this attack, instead of having different recommendations for different kinds of Access-* packets

All RADIUS servers will validate the Message-Authenticator attribute correctly when that attribute is received in a packet. We are not aware of any RADIUS servers which will reject or discard Access-Request packets if they unexpectedly contain a Message-Authenticator attribute.

As discussed earlier, this behavior has been enabled in FreeRADIUS for over a decade, and there have been no interoperability problems. It is therefore safe for all clients to immediately implement this requirement.

However, many existing RADIUS clients do not send Message-Authenticator. It is also difficult to upgrade all client equipment, as the relevant vendor may have gone out of business, or may have marked equipment as “end of life” and thus will not support it. It is therefore necessary to support such systems in the interest of not breaking existing RADIUS deployments.

2.2.2 Servers must check for Message-Authenticator in all Access-Request packets

In addition to requiring that clients must include a Message-Authenticator attribute in all Access-Request packets, servers must have a per-client boolean configuration flag, which we call “require Message-Authenticator”. The default value for this flag must be “false” in order to maintain compatibility with legacy clients.

When the flag is set to “false”, RADIUS servers should follow legacy behavior for enforcing the existence of Message-Authenticator in Access-Request packets. For example, all packets containing EAP-Message must also contain a Message-Authenticator attributes. RADIUS servers must accept and validate the Message-Authenticator attribute if it is present, but otherwise do nothing if the attribute is missing.

The reason for the historical default value to be “false” is that many RADIUS clients do not send the Message-Authenticator attribute in all Access-Request packets. Defaulting to a value of "true" means that the RADIUS server would be unable to accept packets from many legacy RADIUS clients.

If this flag is “false”, then the server may be vulnerable to the attack, even if the client has been updated to always send Message-Authenticator in all Access-Requests. The attacker can simply strip the Message-Authenticator from the Access-Request, and proceed with the attack as if client had not been updated. As a result, this flag should only be set to “false” for NASes, and never for proxies.

When this flag is set to “true”, any Access-Request packets which do not contain Message-Authenticator must be silently discarded. This action protects the server from packets which have been modified in transit to remove Message-Authenticator.

Administrators can set this flag to “true” for clients which send Message-Authenticator, and leave the flag as “false” for clients which cannot be upgraded.

We note that FreeRADIUS has implemented this flag since 2008 (commit 22f82ea3db). However, the default value for the flag has historically been “false”. That default will be changed to "true" in a future release.

Section 7.2 of the paper has the following comment on this configuration option:

> If support for these old clients is not required, enabling this option would make our attacks infeasible.

Every RADIUS server implementation should therefore implement this configuration flag. Every network administrator should enable this flag for all clients which send Message-Authenticator.

While servers must validate the contents of Message-Authenticator, they must not check the location of that attribute. There is no different meaning in RADIUS if Message-Authenticator is the first, second, or last attribute in a packet. Servers must accept a RADIUS packet as valid if

it passes authentication checks, no matter the location of the Message-Authenticator attribute.

Unfortunately, there is no way for clients and servers to negotiate configuration in RADIUS/UDP or RADIUS/TCP. The server cannot determine if the packets are discarded due to an attack, or if they are discarded due to a mismatched configuration between client and server. The server should log the fact that the packet was discarded (with rate limits) in order to inform the administrator that either an attack is underway, or that there is a configuration mismatch between client and server.

As a special corner case for debugging purposes, instead of discarding the packet, servers may immediately instead send a Protocol-Error response packet. This packet must contain a Message-Authenticator attribute as the first attribute in the packet, followed by an Error-Cause attribute containing value 510 (Missing Message-Authenticator). The server must not send this response by default, as it this could cause it to respond to forged Access-Request packets. This behavior must be enabled only when specifically configured by an administrator. It must also be rate-limited.

The purpose of this Protocol-Error packet is to allow administrators to signal misconfigurations between client and server. It is intended to only be used temporarily when new client to server connections are being configured, and must be disabled permanently once the connection is verified to work.

As RADIUS clients are upgraded over time, RADIUS servers can eventually enable the “require Message-Authenticator” flag by default.

The next question is how to protect systems when clients do not send Message-Authenticator.

2.2.3 Servers must limit the use of Proxy-State by Clients

When it is not possible for a server to require Message-Authenticator in Access-Request packets, it is still possible to largely protect them from the attack. We can motivate the solution by observing that the attack requires the server to receive packets containing Proxy-State, while “real” clients (i.e. not proxies) will never send Proxy-State.

A RADIUS server can then protect itself by adding an additional per-client boolean configuration flag, which we call “limit Proxy-State”. This flag should only be examined by the server when the value for the previous “require Message-Authenticator”, flag is set to to “false”. The intention here is to permit the server to accept Access-Request packets which are missing Message-Authenticator, but also to discard the modified packets which are a vector for this attack.

When the flag is set to "false", RADIUS servers should follow legacy behavior for enforcing the existence of Message-Authenticator in Access-Request packets.

When the flag is set to "true", RADIUS servers should require that all Access-Request packets which contain a Proxy-State attribute *also* contain a Message-Authenticator attribute. This flag is motivated by the realization that NASes which do not send Message-Authenticator in Access-Request packets also never send Proxy-State. It is therefore safe to add a flag which checks for Proxy-State, because well-behaving NASes will never send it. The only time the server will see a Proxy-State from a NAS is when the attack is taking place.

As RADIUS proxies are mandated to add Proxy-State to all proxied packets, this flag should be set only when the client is a NAS which cannot be upgraded. The flag should not be set when the client is a proxy, and the “require Message-Authenticator” flag should be used instead.

The recommended behavior for this flag is to not just drop packets which contain Proxy-State, but instead to drop them only if they contain Proxy-State, but do not also contain Message-Authenticator. This recommendation allows the flag to be set even when the client is a proxy, which will presumably be an updated RADIUS server. The additional checks allow the server to be more flexible in what packets it accepts, without compromising on security.

This flag is necessary because it may not be possible to upgrade some RADIUS clients for an extended period of time, or even at all. Some products may no longer be supported, or some vendors have gone out of business. There is therefore a need for RADIUS servers to protect themselves from to this attack, while at the same time being compatible with legcy RADIUS client implementations.

The combination of these two flags is that we both obtain the positive result that the systems are protected as much as feasible, while at the same time avoiding the negative result of creating interoperability issues. The local RADIUS server will be protected from attacks on the client to server path, so long as one of the two flags is set.

These configuration flags will not protect clients (NASes or proxies) from servers which have not been upgraded or configured correctly. More behavior changes to servers and clients are required.

2.2.4 Servers must add Message-Authenticator to all replies to Access-Request packets

Servers can help protect clients by adding Message-Authenticator as the first attribute in all replies to Access-Request packets. i.e. Access-Accept, Access-Reject, Access-Challenge, and Protocol-Error. The attribute must be the first one in the packet, immediately after the 20 byte RADIUS packet header.

Adding Message-Authenticator as the first attribute means that essentially the entire packet is an unknown suffix. The attacker is therefore unable to leverage a known prefix attack, and the vulnerability is mitigated.

This behavior also protects one client to server hop, even if the server does not require Message-Authenticator in Access-Request packets, and even if the client does not examine or validate the contents of the Message-Authenticator.

We note that adding a Message-Authenticator to the end of response packets will not mitigate the attack. When the Message-Authenticator is the last attribute in a packet, the attacker can treat the Message-Authenticator as an unknown suffix, as with the shared secret. The attacker can then calculate the prefix as before, and have the RADIUS server authenticate the packet which contains the prefix. See Section 7.2 of the paper for a more complete description of this process.

The location of the Message-Authenticator attribute is critical to protect legacy clients which do not verify that attribute. Many legacy clients do not send Message-Authenticator in Access-Request packets, and therefore are highly likely to not validate it responses to those Access-Requests. Upgrading all of these clients may be difficult, or in some cases impossible. It is therefore important to have mitigation factors which protect those systems.

The requirement above to send Message-Authenticator first in response packets therefore protects those legacy clients, as the known prefix attack cannot occur, and the client will still verify the Response Authenticator for the unmodified packet.

As it is difficult to upgrade both clients and servers simultaneously, we need a method to protect clients when the server has not been updated. That is, clients cannot depend on the Message-Authenticator existing in response packets. Clients need to take additional steps to protect themselves, independent of any server updates.

2.2.4 Clients must check for Message-Authenticator in all responses to Access-Request packets

As discussed above, an attacker can remove or hide Message-Authenticator from response packet, and then perform the attack. Clients (and proxies) therefore must also have a configuration flag “require Message-Authenticator”, which mirrors the same flag for servers. When the flag is set to "false", RADIUS clients should follow legacy behavior for enforcing the existence of Message-Authenticator in response packets.

When the flag is set to “true”, the client must silently discard (as per RFC 2865 Section 1.2) any response to Access-Request packets which does not contain a Message-Authenticator attribute. This check must be done before the Response Authenticator or Message-Authenticator has been verified. No further processing of the packet should take place.

While clients must validate the contents of Message-Authenticator, they must not check the location of that attribute. There is no different meaning in RADIUS if Message-Authenticator is the first, second, or last attribute in a packet. Clients must accept a RADIUS packet as valid if it passes authentication checks, no matter the location of the Message-Authenticator attribute.

That is, if the Message-Authenticator exists anywhere in the response packet, and that attribute passes validation, then the client can trust that the response from the server has not been modified by an attacker.

When the response is discarded, the client must behave as if the response was never received. That is, any existing retransmission timers must not be modified as a result of receiving a packet which is discarded.

There is no way for clients and servers to negotiate configuration in RADIUS/UDP or RADIUS/TCP. The client cannot determine if the packets are discarded due to an attack, or if they are discarded due to a mismatched configuration between client and server. The client should log the fact that the packet was discarded (with rate limits) in order to inform the administrator that either an attack is underway, or that there is a configuration mismatch between client and server.

2.2.5 Other Client Behavior

RADIUS clients (but not proxies) must also check for the existence of the Proxy-State attribute in replies to Access-Request packets. Since a NAS / GGSN / etc. is not a RADIUS proxy, it will never sent a Proxy-State in an Access-Request,,

and therefore responses to that Access-Request should never contain a Proxy-State attribute. In addition, no standards compliant RADIUS server will respond with a Proxy-State when the Access-Request does not contain a Proxy-State attribute.

If the response to an Access-Request does contain a Proxy-State attribute, then the client can safely discard the packet, knowing that it is invalid. This behaviour should be always enabled, and should not be configurable.

This behavior will also protect the client when the new configuration flags described here are not set.

2.2.6 Responses to Status-Server are not special

While the attack works only for Access-Request packets, the response to Status-Server can also be an Access-Accept or Access-Reject. In order to simplify implementations, servers must follow the above recommendations when receiving Access-Accept or Access-Reject packets, even if the original request was Status-Server.

This requirement ensures that clients can examine responses independent of any requests. That is, the client code can do a simple verification pass of response packets prior to doing any more complex correlation of responses to request.

2.2.7 Less Preferred Flags

An alternative configuration flag with a similar effect to “limit Proxy-State” could be one called “this client is a NAS, and will never send Proxy-State”. The intention there would be to clearly separate RADIUS proxies (which should send Proxy-State), from NASes (which will never send Proxy-State). However, as noted below, validating Proxy-State is insufficient and inadequate.

Such a flag, however, depends on network topology, and not on packet integrity. That is, it works well for one NAS, but is likely to be incorrect if the NAS is replaced by a proxy. If there are multiple different pieces of NAS equipment behind a NAT gateway, flag is also likely to be incorrect.

Setting configuration flags by desired outcome is preferable to setting flags which attempt to control network topology.

2.2.8 Documentation and Logging

We also recommend that RADIUS server implementations document the behavior of these flags in detail, including how they help protect against this attack. We believe that an informed administrator is more likely to engage in secure practices.

Similarly, when either of the above flags cause a packet to be discarded, the RADIUS server should log a descriptive message (subject to rate limiting) about the problematic packet. This log is extremely valuable to administrators who wish to determine if anything is going wrong, and what to do about it.

2.3 Network Operators

The most important recommendation for network operators is that where possible, all RADIUS traffic should use TLS transport between client and server.

All other methods to mitigate the attack are less secure, and are therefore less useful. However, we recognize that not all networking equipment supports TLS transport, so we therefore give additional recommendations which operators can follow to mitigate the attack.

All networking equipment should be physically secure.

We recommend that all RADIUS traffic be sent over a management VLAN. This recommendation should be followed even if TLS transport is used. There is no reason to mix user traffic and management traffic on the same network.

Using a management network for RADIUS traffic will generally prevent anyone other than trusted administrators from performing this attack. We say “generally”, because security is limited by the least secure part of the network. If a network device has some unrelated vulnerability, then an attacker could exploit that vulnerability to gain access to the management network. The attacker would then be free to exploit this issue.

Only the use of TLS will prevent such attacks from being chained together.

We also recommend that RADIUS/UDP traffic should never be sent over the Internet. This issue is discussed in more detail in "Deprecating Insecure Practices in RADIUS".

Similarly, there are few reasons to use RADIUS/TCP. Any system which supports RADIUS/TCP likely also supports TLS, and that should be used instead.

There are additional steps which operators can take, independent of the above recommendations.

2.4 What to Avoid

We recommend not only implementing the above solutions, but also avoiding other possible solutions. The above configuration options effectively prevent the attack without affecting normal RADIUS operation. There is therefore no reason to use anything else.

Other attempted mitigation factors are discussed in the document. For example, Section 7.4 explains why decreasing timeouts simply increases the cost of the attack without preventing it. Decreasing timeouts also can negatively affect normal traffic.

Section 7.7 explains why validating Proxy-State, or looking for unexpected Proxy-State does not help. The attacker can likely just change the nature of the attack, and bypass those checks.

In short, there is no reason to perform “ad hoc” packet validation or sanity checks when it is possible to perform full packet authentication and integrity checks. There is every reason to believe that cryptographic operations designed by experts and subject to rigorous peer review are better than random guesses made by an inexperienced programmer who is coding in isolation.

2.5 Rationale and Practical Realities

This section gives a longer rationale for the above recommendations.

The reality is that there are less than ten (10) major RADIUS server implementations in wide-spread use, but there are dozens if not hundreds of vendors selling networking equipment. In addition, network equipment vendors have multiple product lines, often each with different code bases. It is therefore difficult for these vendors to quickly implement, test, and ship the fixes recommended here.

The situation is similar for network operators. There are many RADIUS clients (switches, access points, firewalls, etc.) for each RADIUS server. Where these clients are from different vendors, any firmware updates have to be procured, installed, and verified. These updates cannot always be automated.

In contrast, RADIUS servers typically run on more complex operating systems such as Linux or Windows. It is much easier there to automate software updates.

The result is that in the short term, the bulk of the effort to address these issues will fall to RADIUS servers, either for vendors to update their software, or for administrators to reconfigure their RADIUS servers.

For example, it is almost always possible for a RADIUS server proxy to be configured to add a Message-Authenticator attribute to all Access-Request packets. Even if the proxy does not add the Message-Authenticator attribute by default, RADIUS servers generally include some kind of customizable policy capabilities. It is therefore usually trivial for an administrator to manually configure a policy which adds the Message-Authenticator attribute.

Once the proxy always adds Message-Authenticator, the next server can be configured to always require it when receiving Access-Request packets, possibly also via a custom policy. That particular communication link is then no longer vulnerable to this attack.

In contrast, it is more difficult to secure the communication between a RADIUS client such as a NAS, access point or switch, and a RADIUS server. As noted earlier, the RADIUS client may simply never send the Message-Authenticator attribute in Access-Request packets. As such, it is not possible for the RADIUS server to enforce that the attribute is always present.

However, this dilemma is solved by noticing that those RADIUS clients are not proxies, and will therefore never include a Proxy-State attribute in any Access-Request packet.

Since the NAS will never send Proxy-State attribute, it is safe to set the “require message authenticator if Proxy-State is present” flag. The only packets which will trigger this behavior, then, are packets which attempt to exploit this issue. This new behavior will then never impact normal packets from the NAS, and the change will not affect normal operation of the NAS.

2.7 Interaction with the IETF

This document is a temporary white paper intended to address and clarify engineering and operational considerations around this new attack. Change control for the RADIUS protocol is still managed by the IETF in the RADEXT working group. This document does not impose a standard of any kind.

Once the vulnerability embargo has lifted, the recommendations in this document will be made part of the "Deprecating Insecure Practices in RADIUS" internet draft and any subsequent RFC. Active participants of the RADEXT working group have reviewed this document, and this document shares an author with the "Deprecating Insecure Practices in RADIUS" draft, so we believe that there is be consensus to follow the recommendations outlined herein. The new RADIUS standard will likely be published later in 2024.

2.8 Intrusion Detection Systems

Intrusion detection systems (IDS) can be updated to look for this attack.

The simplest rule which catches the attack while having some false positives is to look for Access-Request packets which contain a Proxy-State attribute. The false positives

can be significantly reduced by applying this rule only to source IP addresses which is known to a NAS, or perhaps more simply, by not applying it to source IPs which are known to be RADIUS proxies.

If the IDS is capable of doing state tracking by source IP address, the rule can be made to be auto-learning by tracking Access-Request packets from a source IP. With very few exception, these packets should either *always* contain Proxy-State (for proxies), or they should *never* contain Proxy-State (for NAS, GGSN, etc). If the system uses accounting, another reliable signal to detect the attack is where Accounting-Request packets from an IP address do not contain Proxy-State, but some Access-Request packets from the same IP address do contain Proxy-State.

Another related rule which would also have some false positives is to look for Access-Accept packets which contain a Proxy-State attribute. The false positives can be significantly reduced by applying this rule only to destination IP addresses which is known to a NAS, or perhaps more simply, by not applying it to destination IPs which are known to be RADIUS proxies.

If the IDS is capable of doing state tracking by source IP address, the rule can be made to be auto-learning by tracking Access-Accept packets from a source IP. With very few exception, these packets should either *always* contain Proxy-State (for packets sent to proxies), or they should *never* contain Proxy-State (for packets sent to a NAS, GGSN, etc). Similarly, if the system uses accounting, another reliable signal to detect the attack is where Accounting-Response packets to an IP address do not contain Proxy-State, but some Access-Accept packets to the same IP address do contain Proxy-State.

These rules should be applied inside of corporate networks, and not to traffic which is exiting the network, such as with eduroam.

2.9 Conclusions

This attack is the result of standards bodies, implementors and network operators neglecting RADIUS security for almost two decades. The standards have not mandated practices which were known to be secure, and instead simply recommended those practices. Most RADIUS products ignored even those minimal recommendations. Few administrators enabled the implemented functionality.

Perhaps the most important outcome of this attack is psychological and social: A common realization and public acknowledgement that the time of RADIUS/UDP is over, and RADIUS/TLS should now be the minimum acceptable standard.

SECURITY CHECKLISTS

We provide the following checklists as an informational guide for administrators and implementors. We do not normally recommend the use of checklists, as they are too often misunderstood and misused. In this case, in the context of the larger explanations given in this document, checklists can simplify the process of mitigating this attack.

There is no guarantee that your network will be secure when the recommendations below are followed. However, failure to follow these recommendations will likely increase the vulnerability of your network.

The recommendations below to not use MS-CHAP or MS-CHAPv2 are not a result of this issue, but are due to attacks which make those protocols “clear-text equivalent”. That is, an attacker with modest resources can see the MS-CHAP data, and obtain the clear-text password on commodity hardware with only a few milliseconds of CPU time.

In general, using MS-CHAP, MS-CHAPv2, or EAP-MS-CHAPv2 over TLS transport is secure. The TLS transport can either be RADIUS/TLS, RADIUS/DTLS, or a TLS-based EAP method such as PEAP, TTLS, EAP-FAST, or TEAP.

Vendors of RADIUS Clients

The following recommendations are for vendors of equipment which includes a RADIUS client. The recommendations below for Message-Authenticator only apply for UDP and TCP transport.

☐ Ensure that all Access-Request packets contain a Message-Authenticator attribute.

☐ Do not make this behavior configurable.

☐ Implement a per-server configuration flag which requires that all Access-accept, Access-Reject, and Access-Challenge packets coming from a server must contain a Message-Authenticator attribute.

☐ Update the documentation to recommend that this flag only be set to "false" where the RADIUS server cannot be updated, and is known to follow legacy RADIUS/UDP behavior.

☐ Implement RADIUS/TLS.

☐ Implement RADIUS/DTLS.

☐ Do not implement RADIUS/TCP. It has no better security than RADIUS/UDP.

☐ Update the documentation to recommend against using RADIUS/UDP.

☐ Update the documentation to explain that sending RADIUS/UDP across the Internet is insecure, and is likely to result in security and privacy compromises.

The following recommendations are not directly related to this issue, but are also good to follow. Please see "Deprecating Insecure Practices in RADIUS" for more information.

☐ Deprecate or remove all uses of MS-CHAP and MS-CHAPv2 over RADIUS.

☐ If MS-CHAP or MS-CHAPv2 is permitted, require the use of TLS.

☐ Deprecate or remove all uses of EAP-MS-CHAPv2 over RADIUS.

☐ If EAP-MS-CHAPv2 is permitted, require the use of TLS.

Vendors of RADIUS Servers

The following recommendations are for vendors of RADIUS servers. The recommendations below for Message-Authenticator only apply for UDP and TCP transport

☐ Implement RADIUS/TLS.

☐ Implement RADIUS/DTLS.

☐ Ensure that all proxied Access-Request packets contain a Message-Authenticator attribute.

☐ Ensure that all replies to Access-Request packets contain a Message-Authenticator attribute as the first attribute in the packet.

☐ Implement a per-client configuration flag which requires that all Access-Request packets coming from a client must contain a Message-Authenticator attribute.

☐ Update the documentation to recommend that this flag be set when the client is a RADIUS server (i.e. proxy).

☐ Update the documentation to recommend that this flag be set when the client is not a RADIUS server, but it is known to send the Message-Authenticator attribute in all Access-Request packets.

☐ Implement a per-client configuration flag which requires that all Access-Request packets containing a Proxy-State attribute must contain a Message-Authenticator attribute.

☐ Update the documentation to recommend that this flag be set when the client is not a RADIUS server, e.g. a NAS, Access Pointer / Controller, GGSN, etc.

☐ Implement logging such that packets which fail to satisfy the above requirements will generate a descriptive security incident log message.

☐ Update the documentation to recommend against using RADIUS/TCP.

☐ Update the documentation to recommend against using RADIUS/UDP.

☐ Update the documentation to explain that sending RADIUS/UDP across the Internet is insecure, and is likely to result in security and privacy compromises.

The following recommendations are not directly related to this issue, but are also good to follow. Please see "Deprecating Insecure Practices in RADIUS" for more information.

☐ Update the documentation to explain that sending MS-CHAP, MS-CHAPv2, or EAP-MSCHAPv2 over UDP or TCP is insecure and should not be used.

Network Administrators

The following recommendations are for configuring and operating RADIUS systems (clients and server).

☐ RADIUS/TLS or RADIUS/DTLS should be used everywhere, instead of RADIUS/UDP or RADIUS/TCP.

☐ Where RADIUS over (D)TLS is not possible, a VPN connection is also acceptable, but is not as preferred.

☐ RADIUS/UDP traffic should never be sent over the Internet.

☐ If systems cannot be upgraded, install a secured RADIUS proxy close to the legacy systems, in order to mitigate exposure.

The following recommendations are for configuring and operating RADIUS servers.

☐ RADIUS servers which receive packets from a NAS should be configured to set the per-client configuration flag which requires that all Access-Request packets must contain a Message-Authenticator attribute.

☐ Where the above configuration is not possible due to RADIUS client not sending the Message-Authenticator attribute, RADIUS servers should be configured to set the per-client configuration flag which requires that all Access-Request packets containing a Proxy-State attribute must contain a Message-Authenticator attribute .

☐ Where the above configuration is not available on RADIUS servers, local security policy should be added which require that all Access-Request packets containing a Proxy-State attribute must contain a Message-Authenticator attribute.

☐ Packets which fail to satisfy this local security policy should be silently discarded. That is, no packet should be sent in response.

☐ RADIUS servers should be checked to verify that they always include a Message-Authenticator attribute in all proxied Access-Request packets.

☐ Where there is no configuration flag to enable this behavior, a local security policy should be added instead.

The following recommendations help to mitigate this attack, independent of any RADIUS-specific configuration.

☐ All networking equipment should be physically secure.

☐ All network ports should be configured with 802.1X authentication.

☐ Connections between networking equipment should be secured with MACsec.

The following recommendations are not directly related to this issue, but are also good to follow. Please see "Deprecating Insecure Practices in RADIUS" for more information.

☐ RADIUS clients (NAS, switch, AP, etc.) should not be configured to use of MS-CHAP or MS-CHAPv2 over RADIUS.

☐ RADIUS servers should not permit the use of MS-CHAP or MS-CHAPv2 over RADIUS.

NOTE That Access-* replies to status-server also have to have Message-Authenticator. It's not strictly necessary to stop the attack, but it does make things easier for the recipient, and for the "require message-Authenticator" flag.

CONTACT INFORMATION

Network RADIUS SAS
26 rue Colonel Dumont
38000 Grenoble
FRANCE

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

Network RADIUS (Canada)
100 Centrepointe Drive, Suite 200
Ottawa, ON, K2G 6B1
Canada

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

Questions about this white paper can be sent to security@networkradius.com

ACKNOWLEDGEMENTS

We wish to thank Nadia Heninger (UCSD), Arran Cudbard-Bell (FreeRADIUS, Network RADIUS), Heikki Vatiainen (Radiator Software), Jouni Malinen (Hostap / wpa_supplicant), and Fabian Mauchle (radsecproxy) for valuable feedback which significantly improved this document.

ABOUT THE AUTHOR

Alan DeKok is the founder and leader of the FreeRADIUS Server project, which is the most widely used RADIUS server in the world. He has worked with RADIUS since 1997, and has contributed to, or written, many RADIUS standards. As CEO of Network RADIUS SAS, he leads an international team who have built RADIUS systems for customers across the world.

Alan can be reach at aland@networkradius.com.

REFERENCES

This document is available online at https://networkradius.com/assets/pdf/radius_and_md5_collisions.pdf

[1] Goldberg, Sharon et. al., “RADIUS/UDP Considered Harmful”, Unpublished manuscript, 2024

[2] DeKok, A., “Deprecating Insecure Practices in RADIUS”, Work in Progress, Internet-Draft, draft-ietf-radext-deprecating-radius-00, 7 November 2023, https://datatracker.ietf.org/doc/draft-ietf-radext-deprecating-radius/

[3] Rigney, C., Willens, S., Rubens, A., and W. Simpson, "Remote Authentication Dial In User Service (RADIUS)", RFC 2865, June 2000

[4] DeKok, A., "RADIUS over TCP", RFC 6613, DOI 10.17487/RFC6613, May 2012

[5] Winter, S., McCauley, M., Venaas, S., and K. Wierenga, "Transport Layer Security (TLS) Encryption for RADIUS”, RFC 6614, DOI 10.17487/RFC6614, May 2012

[6] Rieckers, J.-F., and Winter, S., “Transport Layer Security (TLS) Encryption for RADIUS”, Work in Progress, Internet-Draft, draft-rieckers-radext-rfc6614bis-02, 10 March 2023, https://datatracker.ietf.org/doc/draft-rieckers-radext-rfc6614bis/

[7] DeKok, A., "Datagram Transport Layer Security (DTLS) as a Transport Layer for RADIUS", RFC 7360, DOI 10.17487/RFC7360, September 2014

[8] Nelson, D. and DeKok, A., “Common Remote Authentication Dial In User Service (RADIUS) Implementation Issues and Suggested Fixes:, RFC5080, DOI 10.17487/RFC5080, December 2007

[9] Nelson, D. and DeKok, A., “Common Remote Authentication Dial In User Service (RADIUS) Implementation Issues and Suggested Fixes:, RFC5080, DOI 10.17487/RFC5080, December 2007