

RADIUS Protocol Security and Best Practices

Abstract

Remote Authentication Dial-In User Service (RADIUS) is commonly used to provide centralized authentication, authorization, and accounting for dial-up, virtual private network, and, more recently, wireless network access. This article provides an overview of RADIUS and the Extensible Authentication Protocol (EAP) and discusses how to minimize or resolve various security issues of the RADIUS protocol using implementation and deployment best practices.

RADIUS Overview

Remote Authentication Dial-In User Service (RADIUS) is a widely deployed protocol enabling centralized authentication, authorization, and accounting for network access. Originally developed for dial-up remote access, RADIUS is now supported by virtual private network (VPN) servers, wireless access points, authenticating Ethernet switches, Digital Subscriber Line (DSL) access, and other network access types. RADIUS is described in RFC 2865, "Remote Authentication Dial-in User Service (RADIUS)," (IETF Draft Standard) and RFC 2866, "RADIUS Accounting" (Informational).

A RADIUS client (typically an access server such as a dial-up server, VPN server, or wireless access point) sends user credentials and connection parameter information in the form of a RADIUS message to a RADIUS server. The RADIUS server authenticates and authorizes the RADIUS client request, and sends back a RADIUS message response. RADIUS clients also send RADIUS accounting messages to RADIUS servers. Additionally, the RADIUS standards support the use of RADIUS proxies. A RADIUS proxy is a computer that forwards RADIUS messages between RADIUS clients, RADIUS servers, and other RADIUS proxies. RADIUS messages are never sent between the access client and the access server.

RADIUS messages are sent as User Datagram Protocol (UDP) messages. UDP port 1812 is used for RADIUS authentication messages and UDP port 1813 is used for RADIUS accounting messages. Some access servers might use UDP port 1645 for RADIUS authentication messages and UDP port 1646 for RADIUS accounting messages. Only one RADIUS message is included in the UDP payload of a RADIUS packet.

RFCs 2865 and 2866 define the following RADIUS message types:

- **Access-Request.** Sent by a RADIUS client to request authentication and authorization for a network access connection attempt.
- **Access-Accept.** Sent by a RADIUS server in response to an Access-Request message. This message informs the RADIUS client that the connection attempt is authenticated and authorized.
- **Access-Reject.** Sent by a RADIUS server in response to an Access-Request message. This message informs the RADIUS client that the connection attempt is rejected. A RADIUS server sends this message if either the credentials are not authentic or the connection attempt is not authorized.
- **Access-Challenge.** Sent Sent by a RADIUS server in response to an Access-Request message. This message is a challenge to the RADIUS client that requires a response.
- **Accounting-Request.** Sent by a RADIUS client to specify accounting information for a connection that was accepted.
- **Accounting-Response.** Sent by the RADIUS server in response to the Accounting-Request message. This message acknowledges the successful receipt and processing of the Accounting-Request message.

RADIUS Protocol Security and Best Practices

A RADIUS message consists of a RADIUS header and RADIUS attributes. Each RADIUS attribute specifies a piece of information about the connection attempt. For example, there are RADIUS attributes for the user name, the user password, the type of service requested by the user, and the IP address of the access server. RADIUS attributes are used to convey information between RADIUS clients, RADIUS proxies, and RADIUS servers. For example, the list of attributes in the Access-Request message includes information about the user credentials and the parameters of the connection attempt. In contrast, the list of attributes in the Access-Accept message includes information about the type of connection that can be made, connection constraints, and any vendor-specific attributes (VSAs).

RADIUS attributes are described in RFCs 2865, 2866, 2867, 2868, 2869, and 3162. RFCs and Internet drafts for VSAs define additional RADIUS attributes.

For Point-to-Point Protocol (PPP) authentication protocols such as Password Authentication Protocol (PAP), Challenge-Handshake Authentication Protocol (CHAP), Microsoft Challenge Handshake Authentication Protocol (MS-CHAP), and MS-CHAP version 2 (MS-CHAP v2), the results of the authentication negotiation between the access server and the access client are forwarded to the RADIUS server for verification.

To provide security for RADIUS messages, the RADIUS client and the RADIUS server are configured with a common shared secret. The shared secret is used to secure RADIUS traffic and is commonly configured as a text string on both the RADIUS client and server.

Extensible Authentication Protocol (EAP) Overview

The Extensible Authentication Protocol (EAP) was originally developed as an extension to PPP allowing for deployment of arbitrary network access authentication mechanisms. With PPP authentication protocols such as CHAP, MS-CHAP, and MS-CHAP v2, a specific authentication mechanism is chosen during the link establishment phase. During the connection authentication phase, the negotiated authentication protocol is used to validate the connection. The authentication protocol itself is a fixed series of messages sent in a specific order. With EAP, the specific authentication mechanism is not chosen during the link establishment phase of the PPP connection. Instead, each PPP peer negotiates to perform EAP during the connection authentication phase. When the connection authentication phase is reached, the peers negotiate the use of a specific EAP authentication scheme known as an EAP type.

Once the EAP type is agreed upon, EAP allows for an open-ended exchange of messages between the access client and the authenticating server (the RADIUS server) that can vary based on the parameters of the connection. The conversation consists of requests for authentication information and the responses. The length and detail of the authentication conversation is dependent upon the EAP type.

Architecturally, EAP is designed to allow authentication plug-in modules at both the access client and authenticating server ends of a connection. By installing an EAP type library file on both the access client and the authenticating server, a new EAP type can be supported. This presents vendors with the opportunity to supply a new authentication scheme at any time. EAP provides the highest flexibility to allow for more secure authentication methods.

You can use EAP to support authentication schemes such as Generic Token Card, One Time Password (OTP), MD5-Challenge, Transport Level Security (TLS) for smart card and certificate support, as well as any future authentication technologies. EAP is a critical technology component for secure connections.

RADIUS Protocol Security and Best Practices

In addition to support within PPP, EAP is also supported within the IEEE 802 link layer. IEEE 802.1X, an IEEE standard for network port authentication, defines how EAP is used for authentication by IEEE 802 devices, including IEEE 802.11b (WiFi) wireless access points and Ethernet switches. IEEE 802.1X differs from PPP in that only EAP authentication methods are supported. As a result, it is not possible to negotiate the use of PAP with IEEE 802.1X.

EAP over RADIUS

EAP over RADIUS is not an EAP type, but the passing of EAP messages of any EAP type by the remote access server to a RADIUS server for authentication. An EAP message sent between the access client and access server is formatted as the EAP-Message RADIUS attribute (RFC 2869, section 5.13), and sent in a RADIUS message between the access server and the RADIUS server. The access server becomes a pass-through device passing EAP messages between the access client and the RADIUS server. Processing of EAP messages occurs at the access client and the RADIUS server, not at the access server.

EAP over RADIUS is used in environments where RADIUS is used as the authentication provider. An advantage of using EAP over RADIUS is that EAP types do not need to be installed at each access server, only at the RADIUS server. However, the access server must support the negotiation of EAP as an authentication protocol and the passing of EAP messages to a RADIUS server.

In a typical use of EAP over RADIUS, the access server is configured to use EAP and to use RADIUS as its authentication provider. When a connection attempt is made, the access client negotiates the use of EAP with the access server. When the client sends an EAP message to the access server, the access server encapsulates the EAP message as a RADIUS message and sends it to its configured RADIUS server. The RADIUS server processes the EAP message and sends a RADIUS-formatted EAP message back to the access server. The access server then forwards the EAP message to the access client.

RADIUS Security Issues and Solutions

The following sections describe RADIUS security issues and possible attacks on RADIUS servers. These issues depend on the ability of the attacker to capture the RADIUS messages sent between the access server and the RADIUS server. This implies that the attacker has physical access to the network and is in the routing path between the access server and the RADIUS server.

RADIUS Access-Request messages sent by RADIUS clients are not authenticated. By default, there is no cryptographic verification of the incoming Access-Request message by the RADIUS server. The RADIUS server verifies that the message originated from an IP address for a configured RADIUS client, but source IP addresses for RADIUS messages can be easily spoofed.

The solution is for the RADIUS server to require the Message-Authenticator attribute in all Access-Request messages. The Message-Authenticator attribute is the Message Digest-5 (MD5) hash of the entire Access-Request message using the shared secret as the key. The access server must send Access-Request messages with the Message-Authenticator attribute and the RADIUS server must silently discard the message if the Message-Authenticator attribute is either not present or fails verification. Normally, the Message-Authenticator attribute is only required for EAP over RADIUS messages.

For example, you can configure the Routing and Remote Access service in the Windows 2000 operating system to send the Message-Authenticator attribute in every Access-Request message by selecting the **Always use digital signatures** check box for the properties of a RADIUS server. You can configure the Windows 2000 Internet Authentication Service (IAS) to require the Message-

RADIUS Protocol Security and Best Practices

Authenticator attribute in every Access-Request message by selecting the **Client must always send the signature attribute in the request** check box for the properties of a RADIUS client.

If it is not possible to use the Message-Authenticator attribute for all Access-Request messages, then use an authentication counting and lockout mechanism. An example is remote access account lockout in Windows 2000, which prevents a user from making remote access connections after a specified number of authentication attempts within a specified amount of time. For more information, see "Remote Access Account Lockout" in the "Internet Authentication Service for Windows 2000" white paper at <http://www.microsoft.com/windows2000/techinfo/howitworks/communications/remotearchive/ias.asp>.

The RADIUS shared secret can be weak due to poor configuration and limited size. In many RADIUS installations, the same shared secret is used to protect many RADIUS client-server pairs, and the RADIUS shared secret does not have sufficient randomness (information entropy) to prevent a successful offline dictionary attack. For a guess of the RADIUS shared secret, the Response Authenticator field and the contents of the Message-Authenticator attribute are easily computed. These results are compared to the values contained within a captured Access-Accept, Access-Reject or Access-Challenge message. An easily guessed RADIUS shared secret can be easily compromised.

The situation is exacerbated by RADIUS client and server implementations that limit the size of the shared secret and require that it be comprised only of characters that can be typed on a keyboard, which uses only 94 out of 256 possible ASCII characters.

The solution to this issue is:

If the shared secret must be a sequence of keyboard characters, choose shared secrets at least 22 characters long and consist of a random sequence of upper and lower case letters, numbers, and punctuation. If the shared secret can be configured as a sequence of hexadecimal digits, use at least 32 random hexadecimal digits.

RFC 2865 recommends shared secrets at least 16 characters, but for a total of 128 bits of entropy, each character must contain a full 8 bits of entropy. If the shared secret is limited to keyboard characters (as opposed to hexadecimal digits), each character has only 5.8 bits of entropy. To provide 128-bits of entropy, the RADIUS client, server, or proxy should allow the configuration of shared secrets at least 22 keyboard characters long. For example, shared secrets for Windows 2000 IAS can be up to 64 keyboard characters long.

To ensure a random shared secret, use a computer program to generate a random sequence at least 22 characters long. Use a different shared secret for each RADIUS server-RADIUS client pair.

Sensitive attributes are encrypted using the RADIUS hiding mechanism. The RADIUS hiding mechanism uses the RADIUS shared secret, the Request Authenticator, and the use of the MD5 hashing algorithm to encrypt the User-Password and other attributes such as Tunnel-Password (RFC 2868, section 3.5) and MS-CHAP-MPPE-Keys (RFC 2548, section 2.4.1). This is a well-known issue and RFC 2865 states:

"The User-Password hiding mechanism described in Section 5.2 has not been subjected to significant amounts of cryptanalysis in the published literature. Some in the IETF community are concerned that this method might not provide sufficient confidentiality protection [15] to passwords transmitted using RADIUS. Users should evaluate their threat environment and consider whether additional security mechanisms should be employed."

RADIUS Protocol Security and Best Practices

The use of a stream cipher and MD5 as a cipher primitive are part of the RADIUS specification. The only standard way to further protect the attributes that are hidden is to use Internet Protocol Security (IPsec) with Encapsulating Security Payload (ESP) and an encryption algorithm such as Triple Data Encryption Standard (3DES), to provide data confidentiality for the entire RADIUS message.

If IPsec with ESP and an encryption algorithm is not possible, RADIUS implementers and network administrators can minimize their vulnerability by doing the following:

- Require the use of the Message-Authenticator attribute (RFC 2869, section 5.14) on all Access-Request messages.
- Use cryptographically strong Request Authenticators.
- Require the use of strong user passwords.
- Use an authentication counting and lockout mechanism to prevent an online dictionary attack against a user's password.
- Use a shared secret with 128 bits of entropy.

Poor Request Authenticator values can be used to decrypt encrypted attributes. As noted in RFC 2865, a secure Request Authenticator must be temporally and globally unique. The Request Authenticator and shared secret combine to determine the key stream used to encrypt the User-Password and other attributes. It is possible for an attacker with the ability to capture traffic between the RADIUS client and server and attempt network access to create a dictionary of RADIUS Request Authenticators and the corresponding key stream used to encrypt the User-Password and other attributes. If the value of the Request Authenticator is ever repeated by an access server using the same shared secret, then the User-Password and other attributes contained within the repeated exchanges can be determined.

If the Request Authenticator is not sufficiently random, then it can be predicted and is also more likely to repeat. The Request Authenticator generator should be of cryptographic quality. If the Request Authenticator generator is not of cryptographic quality, you can use IPsec with ESP and an encryption algorithm such as 3DES to provide data confidentiality for the entire RADIUS message, as described in RFC 3162.

RADIUS Implementation and Deployment Best Practices

To address RADIUS security issues, you should abide by the following implementation and deployment best practices.

Implementation Best Practices

To address RADIUS security issues when implementing a RADIUS client, server, or proxy, use the following best practices:

To provide data confidentiality for the entire RADIUS message, implement IPsec using ESP and an encryption algorithm such as 3DES.

This is described in RFC 3162. By encrypting the entire RADIUS message with IPsec, sensitive RADIUS fields (such as the Request Authenticator field in the Access-Request message) and attributes (such as User-Password, Tunnel-Password, and the MPPE-Key attributes) are protected from viewing. An attacker must first decrypt the ESP-protected RADIUS message before they can

RADIUS Protocol Security and Best Practices

analyze the RADIUS message contents. Support for certificate-based IPsec authentication is recommended to prevent an attacker from launching online attacks against a RADIUS server. Alternately, or in conjunction with using IPsec, you should do the following:

- Allow the configuration and use of shared secrets at least 32 hexadecimal digits long or at least 22 keyboard characters long.
- Implement the use of the Message-Authenticator attribute for all Access-Request messages. For a RADIUS client, implement the use of the Message-Authenticator attribute for all Access-Request messages and allow for its configuration. For a RADIUS server or proxy, implement the required use of the Message-Authenticator attribute for all Access-Request messages and allow for its configuration.
- Implement a cryptographic-quality random generator for the Request Authenticator.

To provide additional protection for access client authentication in your RADIUS implementation, use the following best practices:

- Implement EAP and EAP types that use strong authentication methods. A good example of a strong EAP method is EAP-TLS, which requires the exchange of access client and RADIUS server certificates. All EAP messages require the Message-Authenticator attribute, which provides protection for Access-Request messages that are not protected with IPsec.
- Implement authentication methods that use mutual authentication. With mutual authentication, both ends of the connection authenticate their peer. If either authentication fails, the connection attempt is rejected. For example, EAP-TLS and MS-CHAP v2 are mutual authentication methods. With EAP-TLS, the RADIUS server validates the user certificate of the access client and the access client validates the computer certificate of the RADIUS server. With MS-CHAP v2, both the access client and the access server provide proof of the knowledge of the user account's password.
- If you implement PAP authentication, disable its use by default. For example, OTP/Token Card uses PAP to send the authentication information. If you must implement PAP, disable its use by default and implement long shared secrets and cryptographic-quality Request Authenticators. Because IEEE 802.1X does not support PAP, this issue only applies to PPP connections.
- If you implement CHAP authentication, use a strong CHAP challenge. Like the RADIUS Request Authenticators, the CHAP challenge should be random and of cryptographic quality.
- If you implement MS-CHAP authentication, do not support LAN Manager encoding of MS-CHAP challenge responses or password changes.

Deployment Best Practices

To address RADIUS security issues when deploying a RADIUS solution, use the following deployment best practices:

To provide data confidentiality for the entire RADIUS message, configure your RADIUS clients and servers to use IPsec with ESP with 3DES for all RADIUS traffic.

The configuration of IPsec ESP with 3DES for RADIUS traffic depends on the IPsec implementation. For example, if you are using Windows 2000 Routing and Remote Access service as an access server and Windows 2000 IAS as a RADIUS server in an Active Directory service domain

RADIUS Protocol Security and Best Practices

environment, you can configure the active IPsec policy for the appropriate system container with a rule that uses ESP and 3DES encryption for all traffic to and from UDP ports 1812 and 1813. For more information, see Windows 2000 Server Help.

Alternately, or in conjunction with using IPsec, you should do the following:

- Use strong shared secrets consisting of a random sequence of hexadecimal digits at least 32 digits long or a random sequence of upper and lower case letters, numbers, and punctuation at least 22 characters long. Ideally, the shared secret should be computer-generated.
- Use a different shared secret for each RADIUS client-RADIUS server pair.
- Require the use of the Message-Authenticator attribute for all Access-Request messages.
- Configure each RADIUS client to send the Message-Authenticator attribute with all Access-Request messages. Configure each RADIUS server to require each RADIUS client to send the Message-Authenticator attribute with all Access-Request messages.

Use RADIUS clients, servers, and proxies that use cryptographically strong Request Authenticators. To provide additional protection for access client authentication for your RADIUS deployment, use the following best practices:

- If PAP is not required, disable its use at the access server and on the RADIUS server.
- The only acceptable usage of PAP for secure connections is with OTP and Token Card authentication, where the password has high entropy and changes for each use. However, enabling PAP allows misconfigured access clients to negotiate PAP with their access servers and send unprotected user account passwords. A better solution is to use EAP and EAP types for OTP and Token Card authentication.
- If MS-CHAP is required, disable the use of LAN Manager encoding.
- If you are using Windows 2000 IAS, set the value of the registry key **HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\RemoteAccess\Policy\Allow LM Authentication** to 0 on the IAS server.
- Use EAP and an EAP type with a strong authentication method.
- IEEE 802.1x authentication for wireless access points requires the use of EAP, uses the Message-Authenticator attribute to protect each Access-Request message, and does not support PAP authentication.
- Use a mutual authentication method such as EAP-TLS or MS-CHAP v2.

Summary

This article provided an overview of both RADIUS and EAP and described how RADIUS security issues are addressed or minimized using implementation and deployment best practices. These practices include using strong shared secrets, the Message-Authenticator attribute, cryptographic-quality values for the Request Authenticator, different shared secrets for each RADIUS client/server pair, and IPsec to provide data confidentiality for RADIUS messages.