Guía de Troubleshooting de Autenticación Inválida RADIUS y Message-Authenticator

Contenido

Introducción

Este documento describe dos mecanismos de seguridad RADIUS:

- Encabezado del autenticador
- Atributo Message-Authenticator

Este documento describe cuáles son estos mecanismos de seguridad, cómo se utilizan y cuándo debe esperar que se produzca un error en la validación.

Encabezado del autenticador

Según RFC 2865, el encabezado Authenticator tiene 16 bytes. Cuando se utiliza en una solicitud de acceso, se denomina autenticador de solicitud. Cuando se utiliza en cualquier tipo de respuesta, se denomina autenticador de respuesta. Se utiliza para:

- Autenticación de respuesta
- Ocultación de contraseña

Autenticación de respuesta

Si el servidor responde con el autenticador de respuesta correcto, el cliente puede calcular si esa respuesta estaba relacionada con una solicitud válida.

El cliente envía la solicitud con el encabezado de autenticador aleatorio. A continuación, el servidor que envía la respuesta calcula el autenticador de respuesta con el uso del paquete de solicitud junto con el secreto compartido:

```
ResponseAuth = MD5(Code + ID + Length + RequestAuth + Attributes + Secret)
```

El cliente que recibe la respuesta realiza la misma operación. Si el resultado es el mismo, el paquete es correcto.

> **Nota:** El atacante que conoce el valor secreto no puede falsificar la respuesta a menos que pueda rastrear la solicitud.

¿Cuándo debería esperar una falla de validación?

La falla de validación se produce si el switch ya no almacena en caché la solicitud (por ejemplo, debido al tiempo de espera). También puede experimentarlo cuando el secreto compartido no es válido (sí; Access-Reject también incluye este encabezado). De esta manera, el dispositivo de acceso a la red (NAD) puede detectar la discordancia de secreto compartido. Generalmente, los clientes/servidores de autenticación, autorización y contabilidad (AAA) informan de este error como una discordancia de clave compartida, pero no revela los detalles.

Ocultación de contraseña

El encabezado Authenticator también se utiliza para evitar el envío del atributo User-Password en texto sin formato. Primero se calcula el resumen de mensaje 5 (MD5 - secret, authenticator). Luego se ejecutan varias operaciones XOR con los trozos de la contraseña. Este método es susceptible de ataques fuera de línea (tablas de arco iris) porque MD5 ya no se percibe como un algoritmo unidireccional fuerte.

Esta es la secuencia de comandos de Python que calcula la contraseña de usuario:

```
def Encrypt_Pass(password, authenticator, secret):
       m = md5()
       m.update(secret+authenticator)
       return "".join(chr(ord(x) ^ ord(y)) for x, y in zip(password.ljust
(16,'\0')[:16], m.digest()[:16]))
```

Retransmisiones

Si alguno de los atributos de la solicitud de acceso RADIUS ha cambiado (como el ID de RADIUS, el nombre de usuario, etc.), se debe generar el nuevo campo Authenticator y se deben volver a calcular todos los demás campos que dependen de él. Si se trata de una retransmisión, nada debería cambiar.

Contabilidad

El significado del encabezado Authenticator es diferente para una solicitud de acceso y una

solicitud de contabilidad.

Para una solicitud de acceso, Authenticator se genera aleatoriamente y se espera que reciba una respuesta con ResponseAuthenticator calculada correctamente, lo que demuestra que la respuesta estaba relacionada con esa solicitud específica.

Para una solicitud de contabilidad, Authenticator no es aleatorio, pero se calcula (según RFC 2866):

```
RequestAuth = MD5(Code + ID + Length + 16 zero octets + Attributes + Secret)
```

De esta manera, el servidor puede verificar el mensaje de contabilización inmediatamente y descartar el paquete si el valor recalculado no coincide con el valor Authenticator. Identity Services Engine (ISE) devuelve:

```
11038 RADIUS Accounting-Request header contains invalid Authenticator field
```

La razón típica de esto es la clave secreta compartida incorrecta.

Atributo Message-Authenticator

El atributo Message-Authenticator es el atributo RADIUS definido en RFC 3579. Se utiliza para fines similares: para firmar y validar. Pero esta vez, no se utiliza para validar una respuesta sino una solicitud.

El cliente que envía una solicitud de acceso (también puede ser un servidor que responde con un desafío de acceso) calcula el código de autenticación de mensajes basado en hash (HMAC)-MD5 desde su propio paquete y, a continuación, agrega el atributo Message-Authenticator como una firma. A continuación, el servidor puede verificar que realiza la misma operación.

La fórmula es similar al Encabezado del autenticador:

```
Message-Authenticator = HMAC-MD5 (Type, Identifier, Length, Request Authenticator,
Attributes)
```

La función HMAC-MD5 toma dos argumentos:

- La carga útil del paquete, que incluye el campo Message-Authenticator de 16 bytes rellenado con ceros
- El secreto compartido

¿Cuándo se debe utilizar Message-Authenticator?

Message-Authenticator DEBE utilizarse para cada paquete, que incluye el mensaje EAP (protocolo de autenticación extensible) (RFC 3579). Esto incluye tanto el cliente que envía la solicitud de acceso como el servidor que responde con el Desafío de acceso. El otro lado debería descartar silenciosamente el paquete si falla la validación.

¿Cuándo debería esperar una falla de validación?

La validación fallará cuando el secreto compartido no sea válido. A continuación, el servidor AAA no puede validar la solicitud.

ISE informa:

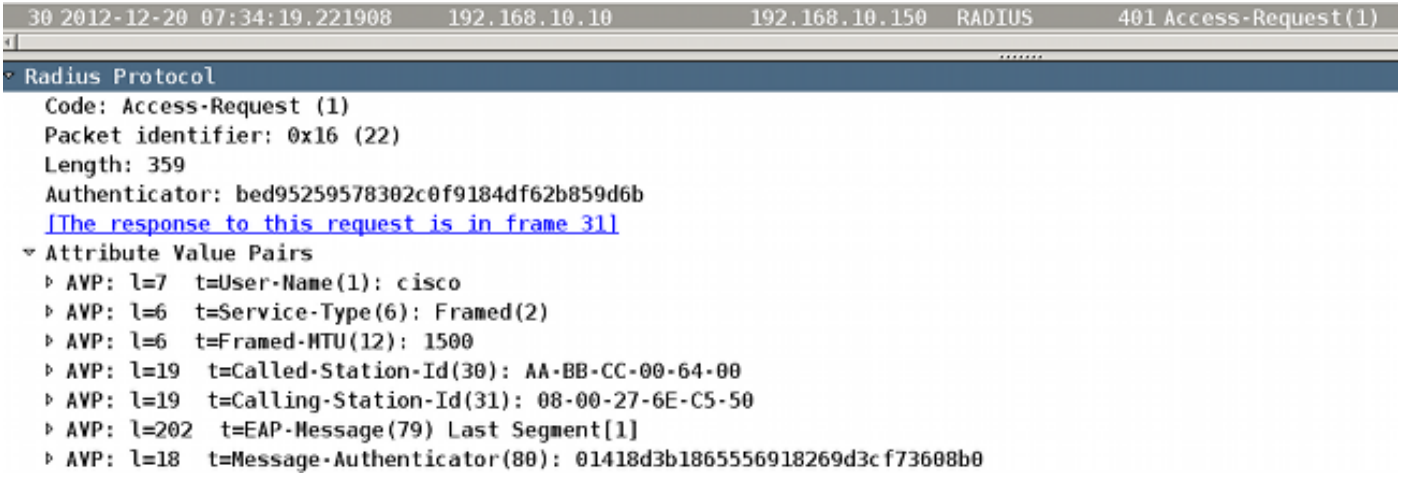
```
11036 The Message-Authenticator Radius Attribute is invalid.
```

Esto suele ocurrir en la etapa posterior cuando se adjunta el mensaje EAP. El primer paquete RADIUS de la sesión 802.1x no incluye el mensaje EAP; no hay campo Message-Authenticator y no es posible verificar la solicitud, pero en esa etapa, el cliente puede validar la respuesta con el uso del campo Authenticator.

Validar el atributo Message-Authenticator

Este es un ejemplo para ilustrar cómo se cuenta manualmente el valor para asegurarse de que se calcula correctamente.

Se ha elegido el número de paquete 30 (Access-Request). Se encuentra en medio de la sesión EAP, y el paquete incluye el campo Message-Authenticator . El objetivo es verificar que Message-Authenticator es correcto:

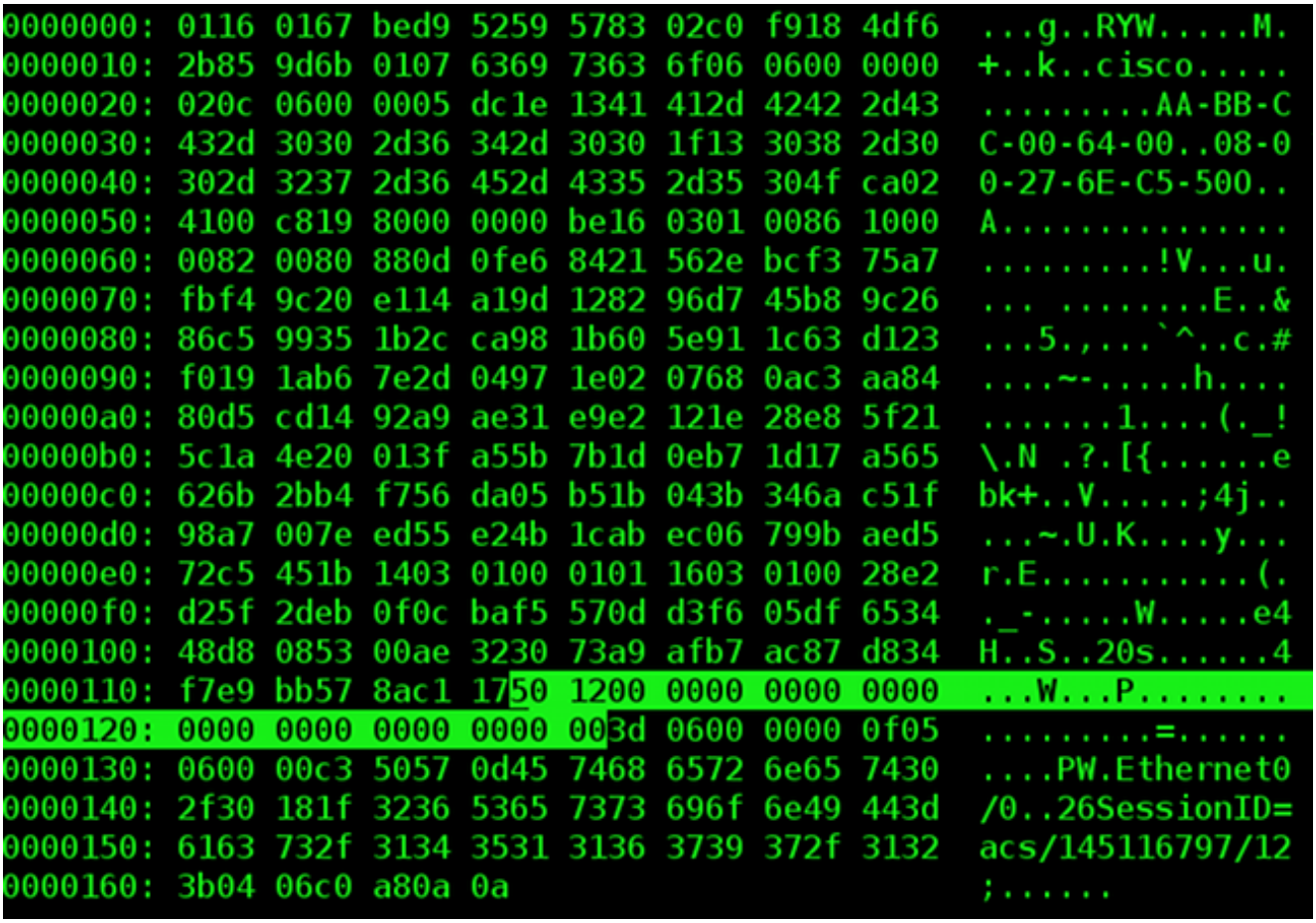


1. Haga clic con el botón derecho del ratón en **Radius Protocol** y elija **Export selected packet bytes**.

2. Escriba esa carga útil RADIUS en un archivo (datos binarios).

3. Para calcular el campo Message-Authenticator, debe poner ceros allí y calcular el HMAC-MD5.

   Por ejemplo, cuando utiliza el editor hexadecimal/binario, como vim, después de escribir ":%!xxd", que cambia al modo hexadecimal y reduce a cero 16 bytes a partir de "5012" (50hex es 80 en dec, que es el tipo Message-Authenticator, y 12 es el tamaño que es 18, incluyendo el encabezado Attribute Value Pairs (AVP):



Después de esa modificación, la carga útil está lista. Es necesario volver al modo hexadecimal/binario (tipo: ":%!xxd -r") y guarde el archivo (":wq").

4. Utilice OpenSSL para calcular HMAC-MD5:

```
pluton # cat packet30-clear-msgauth.bin | openssl dgst -md5 -hmac 'cisco'
(stdin)= 01418d3b1865556918269d3cf73608b0
```

La función HMAD-MD5 toma dos argumentos: el primero de la entrada estándar (stdin) es el mensaje en sí y el segundo es el secreto compartido (Cisco en este ejemplo). El resultado es exactamente el mismo valor que Message-Authenticator asociado al paquete RADIUS Access-Request.

Lo mismo se puede calcular con el uso de la secuencia de comandos Python:

```
pluton # cat hmac.py
#!/usr/bin/env python

import base64
import hmac
import hashlib

f = open('packet30-clear-msgauth.bin', 'rb')
try:
   body = f.read()
finally:
   f.close()

digest = hmac.new('cisco', body, hashlib.md5)
```

```
d=digest.hexdigest()
print d

pluton # python hmac.py
01418d3b1865556918269d3cf73608b0
```

En el ejemplo anterior se muestra cómo calcular el campo Message-Authenticator desde Access-Request. Para Access-Challenge, Access-Accept y Access-Reject, la lógica es exactamente la misma, pero es importante recordar que se debe utilizar Request Authenticator, que se proporciona en el paquete Access-Request anterior.

Información Relacionada

- RFC 2865
- RFC 2866
- RFC 3579
- Soporte Técnico y Documentación - Cisco Systems