Authenticator RADIUS y guía inválidos del Troubleshooting del Mensaje-authenticator

Contenido

Introducción

Este documento describe dos mecanismos de seguridad RADIUS:

- Encabezado del authenticator
- Atributo del Mensaje-authenticator

Este documentos abarca cuáles estos mecanismos de seguridad son, cómo se utilizan, y cuando usted debe contar con el error de la validación.

Encabezado del authenticator

Por el RFC 2865, la encabezado del authenticator es 16 bytes de largo. Cuando se utiliza en un pedido de acceso, se llama un authenticator de la petición. Cuando se utiliza en cualquier clase de respuesta, se llama un authenticator de la respuesta. Se utiliza para:

- Autenticación de la respuesta
- Ocultación de contraseñas

Autenticación de la respuesta

Si el servidor responde con el authenticator correcto de la respuesta, el cliente puede computar si esa respuesta fue relacionada con una petición válida.

El cliente envía la petición con la encabezado al azar del authenticator. Entonces, el servidor que envía la respuesta calcula el authenticator de la respuesta con el uso del paquete de pedidos 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 lo mismo, el paquete está correcto.

> Nota: El atacante que conoce el valor secreto no puede al spoof la respuesta a menos que pueda oler la petición.

¿Cuándo debe usted contar con el error de la validación?

El error de la validación ocurre si el Switch no oculta la petición más (por ejemplo, debido al descanso). Usted puede ser que también lo experimente cuando el secreto compartido es inválido (sí - el Access-Reject también incluye esta encabezado). Esta manera, el dispositivo de acceso a la red (NAD) puede detectar la discordancía del secreto compartido. Es señalada generalmente por los clientes/los servidores del Authentication, Authorization, and Accounting (AAA) como discordancía de clave compartida, pero no revela los detalles.

Ocultación de contraseñas

La encabezado del authenticator también se utiliza para evitar enviar el atributo de la Contraseña del Usuario en el sólo texto. Primero la publicación de mensaje 5 (MD5 - secreto, authenticator) se computa. Entonces varias operaciones XOP con los pedazos de la contraseña se ejecutan. Este método es susceptible para los ataques offline (tablas del arco iris) porque el MD5 no se percibe como algoritmo unidireccional fuerte más.

Aquí está el script de Python que computa la Contraseña del 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 los atributos uces de los en el pedido de acceso RADIUS han cambiado (como el RADIUS ID, username, y así sucesivamente), el nuevo campo del authenticator debe ser generado y el resto de los campos que dependen de él deben ser recalculados. Si esto es una retransmisión, nada debe cambiar.

Contabilidad

El significado de la encabezado del authenticator es diferente para un pedido de acceso y una Estadística-petición.

Para un pedido de acceso, el authenticator se genera aleatoriamente y se espera que reciba una respuesta con el ResponseAuthenticator calculado correctamente, que prueba que la respuesta fue relacionada con esa petición específica.

Para una Estadística-petición, el authenticator no es al azar, sino que se calcula (según el RFC 2866):

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

Esta manera, el servidor puede marcar el mensaje de las estadísticas inmediatamente y caer el paquete si el valor recalculado no hace juego el valor del authenticator. Las devoluciones del Identity Services Engine (ISE):

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

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

Atributo del Mensaje-authenticator

El atributo del Mensaje-authenticator es el atributo de RADIUS definido en el RFC 3579. Se utiliza para un propósito similar: para firmar y validar. Pero este vez, no se utiliza para validar una respuesta sino una petición.

El cliente que envía un pedido de acceso (él puede también ser un servidor que responde con un Acceso-desafío) computa el Hash-Based Message Authentication Code (HMAC)-MD5 de su propio paquete, y entonces agrega el atributo del Mensaje-authenticator como firma. Entonces, el servidor puede verificarla realiza la misma operación.

La fórmula parece similar a la encabezado del authenticator:

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

La función HMAC-MD5 admite dos argumentos:

- El payload del paquete, que incluye el campo del Mensaje-authenticator de 16 bytes llenó de los ceros
- El secreto compartido

¿Cuándo debe el Mensaje-authenticator ser utilizado?

El Mensaje-authenticator SE DEBE utilizar para cada paquete, que incluye el mensaje del Protocolo de Autenticación Extensible (EAP) (RFC 3579). Esto incluye al cliente que envía el pedido de acceso y el servidor que responde con el Acceso-desafío. El otro lado debe caer silenciosamente el paquete si la validación falla.

¿Cuándo debe usted contar con el error de la validación?

El error de la validación ocurrirá cuando el secreto compartido es inválido. Entonces, el servidor de AAA no puede validar la petición.

Los informes ISE:

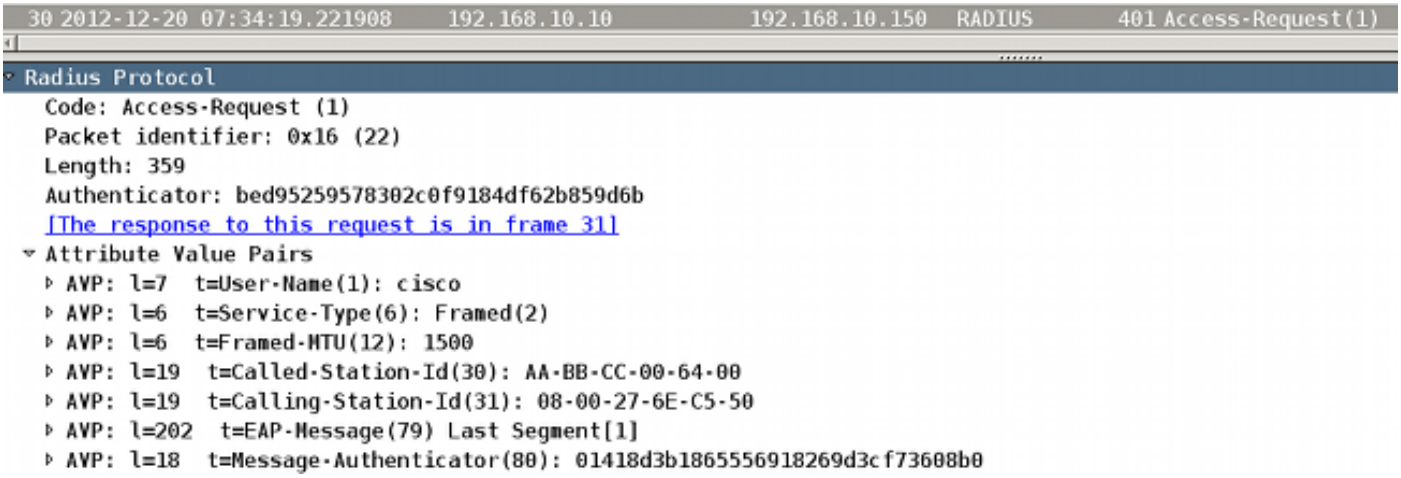
`11036 The Message-Authenticator Radius Attribute is invalid.`

Esto ocurre generalmente en la etapa posterior cuando se asocia el mensaje EAP. El primer paquete RADIUS de la sesión del 802.1x no incluye el mensaje EAP; no hay campo del Mensaje-authenticator y no es posible verificar la petición, pero en esa etapa, el cliente puede validar la respuesta con el uso del campo del authenticator.

Valide el atributo del Mensaje-authenticator

Aquí está un ejemplo para ilustrar cómo usted cuenta manualmente el valor para aseegurarse lo se computa correctamente.

Se ha elegido el paquete número 30 (pedido de acceso). Está en el medio de la sesión EAP, y el paquete incluye el campo del Mensaje-authenticator. El objetivo es verificar que el Mensaje-authenticator está correcto:

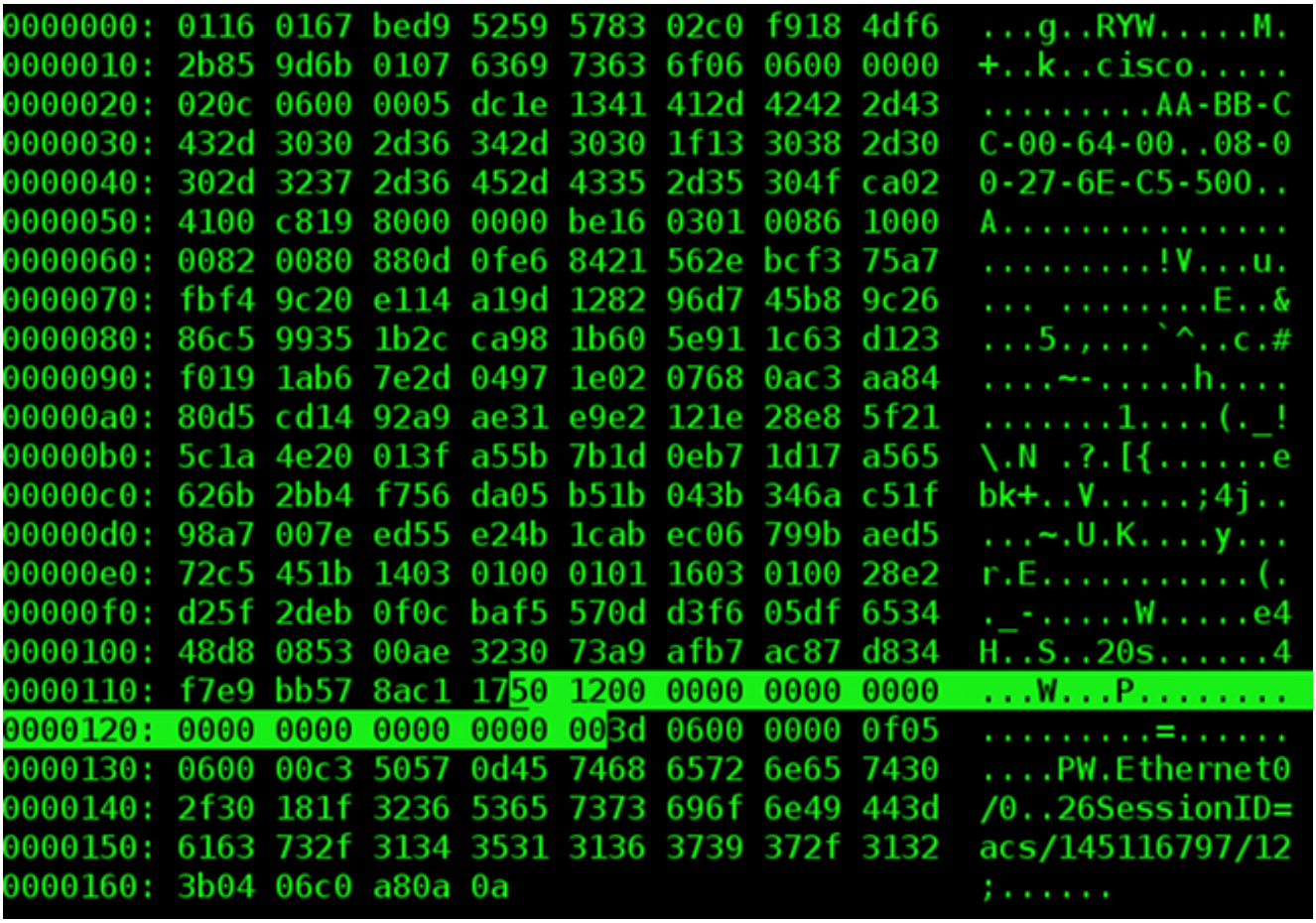


1. Haga clic con el botón derecho del ratón el **protocolo RADIUS** y elija los **bytes del paquete seleccionados exportación**.

2. Escriba ese payload RADIUS a un archivo (datos binarios).

3. Para computar el campo del Mensaje-authenticator, usted debe poner los ceros allí y computar el HMAC-MD5.

   Por ejemplo, cuando usted utiliza el maleficio/el Editor binario, tal como vim, después de que usted teclee “: ¡%! el xxd”, que conmuta para embrujar el modo y los ceros 16 bytes que comienzan después de que el "5012" (50hex es 80 en diciembre que es tipo del Mensaje-authenticator, y 12 es el tamaño que es 18 incluyendo la encabezado de los pares de valores de atributos (AVP)):



Después esa modificación, el payload está lista. Es necesario volver de nuevo al maleficio/al modo binario (tipo: “: ¡%! el xxd - r”) y salva el archivo (“: wq”).

4. Utilice el OpenSSL para computar el HMAC-MD5:

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

La función HMAD-MD5 toma dos argumentos: primer de la entrada estándar (stdin) es el mensaje sí mismo y segundo es el secreto compartido (Cisco en este ejemplo). El resultado es exactamente el mismo valor que el Mensaje-authenticator asociado al paquete access-request RADIUS.

Lo mismo se pueden computar con el uso del script de 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
```

El ejemplo anterior presenta cómo calcular el campo del Mensaje-authenticator del pedido de acceso. Para el Acceso-desafío, el access-accept, y el Access-Reject, la lógica es exactamente lo mismo, pero es importante recordar que el authenticator de la petición debe ser utilizado, que se proporciona en el paquete access-request anterior.

Información Relacionada

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