Fehlerbehebung für RADIUS Invalid Authenticator und Message-Authenticator

Inhalt

[Einführung](#)
[Authentifizierer-Header](#)
[Authentifizierung der Antwort](#)
[Wann ist ein Validierungsfehler zu erwarten?](#)
[Ausblenden von Kennwörtern](#)
[Neuübertragungen](#)
[Buchhaltung](#)
[Message-Authenticator-Attribut](#)
[Wann sollte der Message-Authenticator verwendet werden?](#)
[Wann ist ein Validierungsfehler zu erwarten?](#)
[Validieren des Message-Authenticator-Attributs](#)
[Zugehörige Informationen](#)

Einführung

In diesem Dokument werden zwei RADIUS-Sicherheitsmechanismen beschrieben:

- Authentifizierer-Header
- Message-Authenticator-Attribut

In diesem Dokument werden diese Sicherheitsmechanismen, ihre Verwendung und der Zeitpunkt beschrieben, an dem ein Validierungsfehler zu erwarten ist.

Authentifizierer-Header

Gemäß RFC 2865 ist der Authentifizierer-Header 16 Byte lang. Wenn sie in einer Access-Request verwendet wird, wird sie als Anforderungsauthentifizierer bezeichnet. Wenn sie in einer beliebigen Antwort verwendet wird, wird sie als Antwortauthentifizierer bezeichnet. Es wird verwendet für:

- Authentifizierung der Antwort
- Ausblenden von Kennwörtern

Authentifizierung der Antwort

Wenn der Server mit dem richtigen Antwortauthentifizierer antwortet, kann der Client berechnen, ob diese Antwort auf eine gültige Anfrage zurückzuführen ist.

Der Client sendet die Anforderung mit dem zufälligen Authentifizierer-Header. Anschließend berechnet der Server, der die Antwort sendet, den Response-Authentifizierer mithilfe des Anforderungspakets zusammen mit dem freigegebenen geheimen Schlüssel:

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

Der Client, der die Antwort empfängt, führt den gleichen Vorgang aus. Wenn das Ergebnis identisch ist, ist das Paket richtig.

> **Hinweis:** Der Angreifer, der den geheimen Wert kennt, kann die Antwort nur verfälschen, wenn er in der Lage ist, die Anforderung zu schnappen.

Wann ist ein Validierungsfehler zu erwarten?

Der Validierungsfehler tritt auf, wenn der Switch die Anforderung nicht mehr zwischenspeichert (z. B. aufgrund eines Timeouts). Sie können es auch erleben, wenn der gemeinsame geheime Schlüssel ungültig ist (ja - Access-Reject beinhaltet auch diesen Header). Auf diese Weise kann das Netzwerkzugriffsgerät (Network Access Device, NAD) die gemeinsame geheime Diskrepanz erkennen. In der Regel werden sie von den Clients/Servern für Authentifizierung, Autorisierung und Abrechnung (Authentication, Authorization, Accounting - AAA) als Übereinstimmung mit einem gemeinsamen Schlüssel gemeldet, aber die Details werden nicht offen gelegt.

Ausblenden von Kennwörtern

Der Authentifizierer-Header wird auch verwendet, um das Senden des User-Password-Attributs im Klartext zu vermeiden. Zuerst wird die Message Digest 5 (MD5 - secret, Authentifizierer) berechnet. Anschließend werden mehrere XOR-Operationen mit den Chunks des Kennworts ausgeführt. Diese Methode ist anfällig für Offline-Angriffe (Regenbogentabellen), da MD5 nicht mehr als sicherer Einwegalgorithmus wahrgenommen wird.

Hier ist das Python-Skript, das das Benutzerkennwort berechnet:

```
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]))
```

Neuübertragungen

Wenn eines der Attribute in der RADIUS Access-Request geändert wurde (z. B. RADIUS-ID, Benutzername usw.), sollte das neue Authentifiziererfeld generiert und alle anderen davon abhängigen Felder neu berechnet werden. Wenn es sich um eine erneute Übertragung handelt, sollte sich nichts ändern.

Buchhaltung

Die Bedeutung des Authentifizierer-Headers unterscheidet sich für eine Access-Request und eine Accounting-Request.

Bei einer Access-Request wird der Authenticator nach dem Zufallsprinzip generiert, und es wird erwartet, dass er eine Antwort mit dem ResponseAuthenticator erhält, der korrekt berechnet wurde. Dies belegt, dass die Antwort auf diese spezifische Anforderung zutrifft.

Bei einer Accounting-Request ist der Authentifizierer nicht zufällig, sondern wird berechnet (gemäß RFC 2866):

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

Auf diese Weise kann der Server die Accounting-Nachricht sofort überprüfen und das Paket verwerfen, wenn der neu berechnete Wert nicht mit dem Authenticator-Wert übereinstimmt. Die Identity Services Engine (ISE) gibt Folgendes zurück:

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

Der Grund hierfür ist in der Regel der falsche gemeinsame geheime Schlüssel.

Message-Authenticator-Attribut

Das Message-Authenticator-Attribut ist das in RFC 3579 definierte RADIUS-Attribut. Es wird für einen ähnlichen Zweck verwendet: zur Unterzeichnung und Validierung. Dieses Mal wird es jedoch nicht zur Validierung einer Antwort, sondern einer Anforderung verwendet.

Der Client, der eine Access-Request sendet (es kann auch ein Server sein, der mit einer Access-Challenge antwortet) berechnet den Hash-Based Message Authentication Code (HMAC)-MD5 aus seinem eigenen Paket und fügt dann das Message-Authenticator-Attribut als Signatur hinzu. Anschließend kann der Server überprüfen, ob er die gleiche Operation ausführt.

Die Formel ähnelt dem Authentifizierer-Header:

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

Die HMAC-MD5-Funktion verwendet zwei Argumente:

- Die Nutzlast des Pakets, das das mit Nullen gefüllte 16-Byte-Meldungsauthentifizierungsfeld enthält
- Das gemeinsame Geheimnis

Wann sollte der Message-Authenticator verwendet werden?

Der Message-Authenticator MUSS für jedes Paket verwendet werden, das die Extensible Authentication Protocol (EAP)-Nachricht (RFC 3579) enthält. Dies schließt sowohl den Client ein, der die Zugriffsanfrage sendet, als auch den Server, der mit der Access-Challenge antwortet. Wenn die Validierung fehlschlägt, sollte das Paket auf der anderen Seite stumm gelöscht werden.

Wann ist ein Validierungsfehler zu erwarten?

Validierungsfehler treten auf, wenn der gemeinsam verwendete geheime Schlüssel ungültig ist. Dann kann der AAA-Server die Anforderung nicht validieren.

Die ISE berichtet:

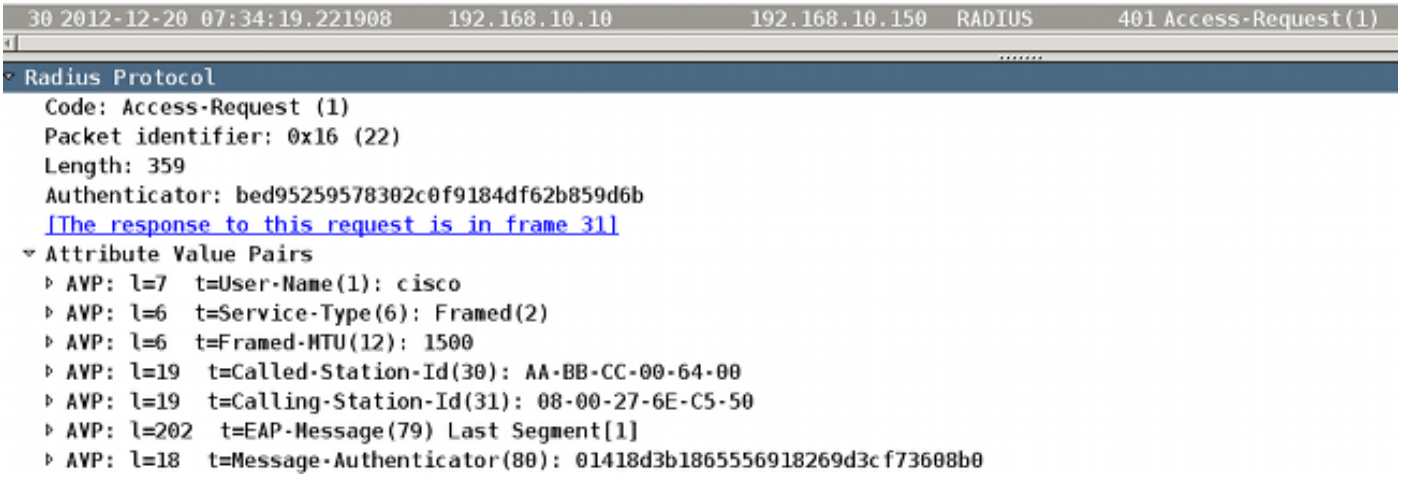
```
11036 The Message-Authenticator Radius Attribute is invalid.
```

Dies geschieht in der Regel zu einem späteren Zeitpunkt, wenn die EAP-Nachricht angehängt wird. Das erste RADIUS-Paket der 802.1x-Sitzung enthält die EAP-Nachricht nicht. Es gibt kein Meldungsauthentifiziererfeld, und es ist nicht möglich, die Anfrage zu überprüfen. In diesem Stadium kann der Client jedoch die Antwort mithilfe des Felds "Authentifizierer" validieren.

Validieren des Message-Authenticator-Attributs

Im folgenden Beispiel wird veranschaulicht, wie Sie den Wert manuell zählen, um sicherzustellen, dass er korrekt berechnet wird.

Die Paketnummer 30 (Access-Request) wurde ausgewählt. Es befindet sich in der Mitte der EAP-Sitzung, und das Paket enthält das Feld "Message-Authenticator" (Meldungsauthentifizierung). Ziel ist es, die Richtigkeit der Nachrichtenauthentifizierung zu überprüfen:

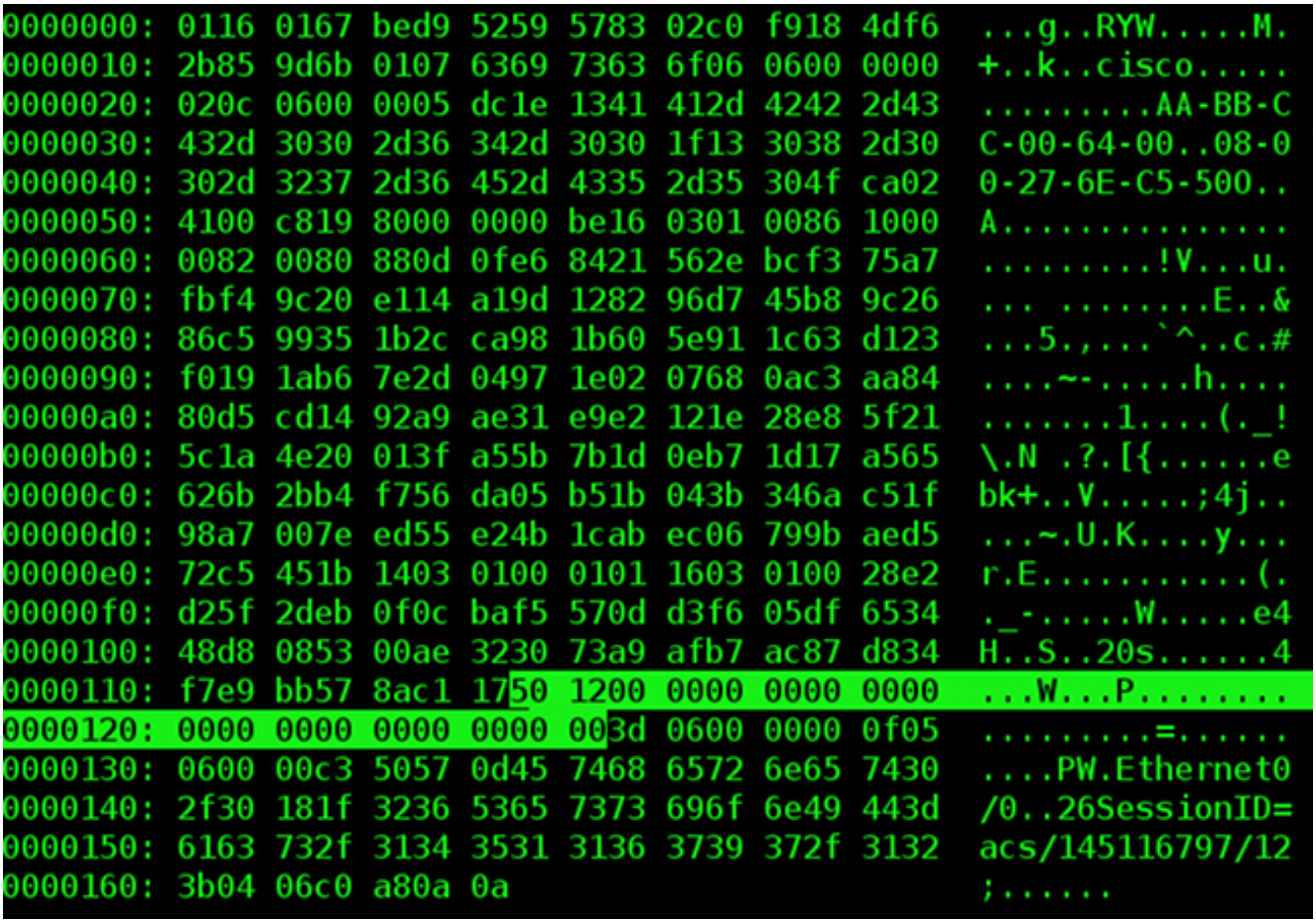


1. Klicken Sie mit der rechten Maustaste auf **Radius Protocol**, und wählen Sie **Ausgewählte Paketbytes exportieren aus**.

2. Schreiben Sie diese RADIUS-Payload in eine Datei (binäre Daten).

3. Um das Meldungsauthentifiziererfeld zu berechnen, müssen Sie dort Nullen angeben und die HMAC-MD5 berechnen.

   Wenn Sie beispielsweise einen Hex-/Binär-Editor wie vim verwenden, nachdem Sie ":%!xxd" eingegeben haben, der in den Hexadezimalmodus wechselt und 16 Byte nach "5012" (50 Hexadezimalwert ist 80 Dezimalstelle, was dem Meldungsauthentifizierer-Header entspricht, und 12 ist die Größe von 18 einschließlich des AVP-Headers:



Nach dieser Änderung ist die Nutzlast bereit. Es ist notwendig, zum Hex/Binärmodus zurückzukehren (Typ: ":%!xxd -r") und speichern Sie die Datei (":wq").

4. Verwenden Sie OpenSSL, um HMAC-MD5 zu berechnen:

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

Die HMAD-MD5-Funktion verwendet zwei Argumente: Die erste aus der Standardeingabe (stdin) ist die Nachricht selbst, die zweite die gemeinsame geheime Nachricht (in diesem Beispiel Cisco). Das Ergebnis ist genau der gleiche Wert wie der Message-Authenticator, der an das RADIUS Access-Request-Paket angeschlossen ist.

Dasselbe kann mit dem Python-Skript berechnet werden:

```
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
```

Im vorherigen Beispiel wird veranschaulicht, wie das Feld Message-Authenticator aus der Access-Request berechnet wird. Bei Access-Challenge, Access-Accept und Access-Reject ist die Logik identisch, aber es ist wichtig, daran zu denken, dass der Request Authenticator verwendet werden soll, der im vorherigen Access-Request-Paket bereitgestellt wird.

Zugehörige Informationen

- RFC 2865
- RFC 2866
- RFC 3579
- Technischer Support und Dokumentation für Cisco Systeme