De configuratie van het SSL-algoritme van Expressway aanpassen

Inhoud

Inleiding

Dit document beschrijft de stappen om de vooraf ingestelde algoritme koorden op Expressway aan te passen.

Voorwaarden

Vereisten

Cisco raadt kennis van de volgende onderwerpen aan:

- Cisco Express voor Cisco VCS.
- TLS-protocol.

Gebruikte componenten

De informatie in dit document is gebaseerd op de volgende software- en hardware-versies:

- Cisco Expressway versie X15.0.2.1

De informatie in dit document is gebaseerd op de apparaten in een specifieke

laboratoriumomgeving. Alle apparaten die in dit document worden beschreven, hadden een opgeschoonde (standaard)configuratie. Als uw netwerk live is, moet u zorgen dat u de potentiële impact van elke opdracht begrijpt.

Achtergrondinformatie

De standaard Expressway-configuratie omvat pregeconfigureerde algoritme strings, die om compatibiliteitsredenen ondersteuning voor bepaalde algoritmen mogelijk maken die als zwak kunnen worden beschouwd in het kader van een bepaald bedrijfsbeveiligingsbeleid. Het is mogelijk om de algoritme koorden aan te passen om hen te verfijnen om het specifieke beleid van elke omgeving te passen.

In Expressway, is het mogelijk om een onafhankelijk cijferkoord voor elk van deze protocollen te vormen:

- HTTPS
- LDAP
- Reverse proxy
- SIP
- SMTP
- TMS-provisioning
- UCS-serverdetectie
- XMPP

De algoritme koorden gehoorzamen aan het OpenSSL-formaat dat in het OpenSSL-coderingsmanpage wordt beschreven. De huidige Expressway versie X15.0.2 wordt geleverd met de standaard string EECDH:EDH:HIGH:-AES256+SHA:!MEDIUM:!LOW:!3DES:!MD5:!PSK:!NULL:!aNULL:!aDH is vooraf geconfigureerd voor alle protocollen in gelijke mate. Van de web admin pagina, onder Onderhoud > Beveiliging > Cijfers, kunt u de algoritme wijzigen die aan elk protocol is toegewezen, om specifieke algoritmen of groepen van algoritmen toe te voegen of te verwijderen met behulp van een gemeenschappelijk algoritme.

De coderingsstring inspecteren

Door de opdracht openssl-algoritmen -V "<algoritme string>" te gebruiken, kunt u een lijst met alle algoritmen uitvoeren die door een bepaalde tekenreeks zijn toegestaan, wat nuttig is voor de visuele inspectie van de algoritmen. Dit voorbeeld toont de uitvoer bij het inspecteren van de standaard expressway algoritme string:

<#root>

~ #

openssl ciphers -V "EECDH:EDH:HIGH:-AES256+SHA:!MEDIUM:!LOW:!3DES:!MD5:!PSK:!eNULL:!aNULL:!aDH"

```
0x13,0x02 - TLS_AES_256_GCM_SHA384 TLSv1.3 Kx=any Au=any Enc=AESGCM(256) Mac=AEAD
0x13,0x03 - TLS_CHACHA20_POLY1305_SHA256 TLSv1.3 Kx=any Au=any Enc=CHACHA20/POLY1305(256) Mac=AEAD
```

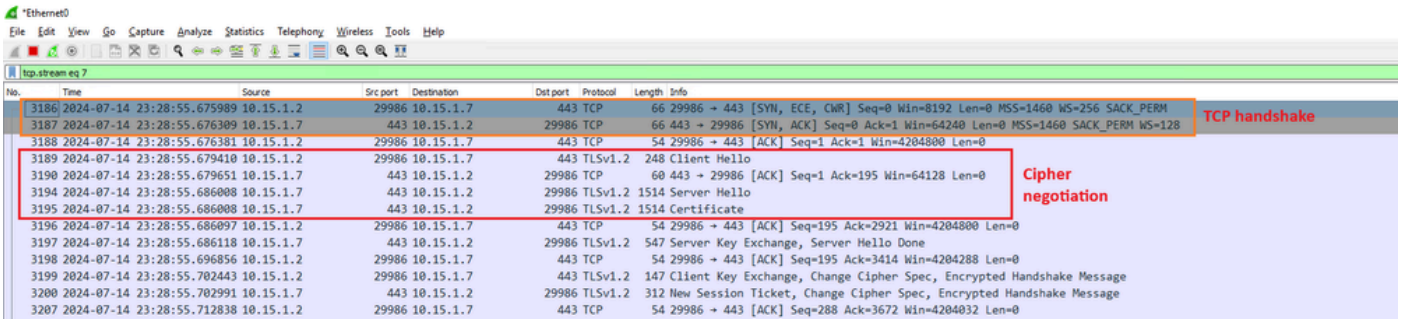
```
0x13,0x01 - TLS_AES_128_GCM_SHA256 TLSv1.3 Kx=any Au=any Enc=AESGCM(128) Mac=AEAD
0xC0,0x2C - ECDHE-ECDSA-AES256-GCM-SHA384 TLSv1.2 Kx=ECDH Au=ECDSA Enc=AESGCM(256) Mac=AEAD
0xC0,0x30 - ECDHE-RSA-AES256-GCM-SHA384 TLSv1.2 Kx=ECDH Au=RSA Enc=AESGCM(256) Mac=AEAD
0xCC,0xA9 - ECDHE-ECDSA-CHACHA20-POLY1305 TLSv1.2 Kx=ECDH Au=ECDSA Enc=CHACHA20/POLY1305(256) Mac=AEAD
0xCC,0xA8 - ECDHE-RSA-CHACHA20-POLY1305 TLSv1.2 Kx=ECDH Au=RSA Enc=CHACHA20/POLY1305(256) Mac=AEAD
0xC0,0xAD - ECDHE-ECDSA-AES256-CCM TLSv1.2 Kx=ECDH Au=ECDSA Enc=AESCCM(256) Mac=AEAD
0xC0,0x2B - ECDHE-ECDSA-AES128-GCM-SHA256 TLSv1.2 Kx=ECDH Au=ECDSA Enc=AESGCM(128) Mac=AEAD
0xC0,0x2F - ECDHE-RSA-AES128-GCM-SHA256 TLSv1.2 Kx=ECDH Au=RSA Enc=AESGCM(128) Mac=AEAD
0xC0,0xAC - ECDHE-ECDSA-AES128-CCM TLSv1.2 Kx=ECDH Au=ECDSA Enc=AESCCM(128) Mac=AEAD
0xC0,0x24 - ECDHE-ECDSA-AES256-SHA384 TLSv1.2 Kx=ECDH Au=ECDSA Enc=AES(256) Mac=SHA384
0xC0,0x28 - ECDHE-RSA-AES256-SHA384 TLSv1.2 Kx=ECDH Au=RSA Enc=AES(256) Mac=SHA384
0xC0,0x23 - ECDHE-ECDSA-AES128-SHA256 TLSv1.2 Kx=ECDH Au=ECDSA Enc=AES(128) Mac=SHA256
0xC0,0x27 - ECDHE-RSA-AES128-SHA256 TLSv1.2 Kx=ECDH Au=RSA Enc=AES(128) Mac=SHA256
0xC0,0x09 - ECDHE-ECDSA-AES128-SHA TLSv1 Kx=ECDH Au=ECDSA Enc=AES(128) Mac=SHA1
0xC0,0x13 - ECDHE-RSA-AES128-SHA TLSv1 Kx=ECDH Au=RSA Enc=AES(128) Mac=SHA1
0x00,0xA3 - DHE-DSS-AES256-GCM-SHA384 TLSv1.2 Kx=DH Au=DSS Enc=AESGCM(256) Mac=AEAD
0x00,0x9F - DHE-RSA-AES256-GCM-SHA384 TLSv1.2 Kx=DH Au=RSA Enc=AESGCM(256) Mac=AEAD
0xCC,0xAA - DHE-RSA-CHACHA20-POLY1305 TLSv1.2 Kx=DH Au=RSA Enc=CHACHA20/POLY1305(256) Mac=AEAD
0xC0,0x9F - DHE-RSA-AES256-CCM TLSv1.2 Kx=DH Au=RSA Enc=AESCCM(256) Mac=AEAD
0x00,0xA2 - DHE-DSS-AES128-GCM-SHA256 TLSv1.2 Kx=DH Au=DSS Enc=AESGCM(128) Mac=AEAD
0x00,0x9E - DHE-RSA-AES128-GCM-SHA256 TLSv1.2 Kx=DH Au=RSA Enc=AESGCM(128) Mac=AEAD
0xC0,0x9E - DHE-RSA-AES128-CCM TLSv1.2 Kx=DH Au=RSA Enc=AESCCM(128) Mac=AEAD
0x00,0x6B - DHE-RSA-AES256-SHA256 TLSv1.2 Kx=DH Au=RSA Enc=AES(256) Mac=SHA256
0x00,0x6A - DHE-DSS-AES256-SHA256 TLSv1.2 Kx=DH Au=DSS Enc=AES(256) Mac=SHA256
0x00,0x67 - DHE-RSA-AES128-SHA256 TLSv1.2 Kx=DH Au=RSA Enc=AES(128) Mac=SHA256
0x00,0x40 - DHE-DSS-AES128-SHA256 TLSv1.2 Kx=DH Au=DSS Enc=AES(128) Mac=SHA256
0x00,0x33 - DHE-RSA-AES128-SHA SSLv3 Kx=DH Au=RSA Enc=AES(128) Mac=SHA1
0x00,0x32 - DHE-DSS-AES128-SHA SSLv3 Kx=DH Au=DSS Enc=AES(128) Mac=SHA1
0x00,0x9D - AES256-GCM-SHA384 TLSv1.2 Kx=RSA Au=RSA Enc=AESGCM(256) Mac=AEAD
0xC0,0x9D - AES256-CCM TLSv1.2 Kx=RSA Au=RSA Enc=AESCCM(256) Mac=AEAD
0x00,0x9C - AES128-GCM-SHA256 TLSv1.2 Kx=RSA Au=RSA Enc=AESGCM(128) Mac=AEAD
0xC0,0x9C - AES128-CCM TLSv1.2 Kx=RSA Au=RSA Enc=AESCCM(128) Mac=AEAD
0x00,0x3D - AES256-SHA256 TLSv1.2 Kx=RSA Au=RSA Enc=AES(256) Mac=SHA256
0x00,0x3C - AES128-SHA256 TLSv1.2 Kx=RSA Au=RSA Enc=AES(128) Mac=SHA256
0x00,0x2F - AES128-SHA SSLv3 Kx=RSA Au=RSA Enc=AES(128) Mac=SHA1
~ #
```

Inspecteer de cifeeronderhandeling in de TLS-handdruk met een pakketvastlegging

Door een TLS-onderhandeling in een pakketopname vast te leggen, kunt u de details van de algoritmeonderhandeling controleren met behulp van Wireshark.

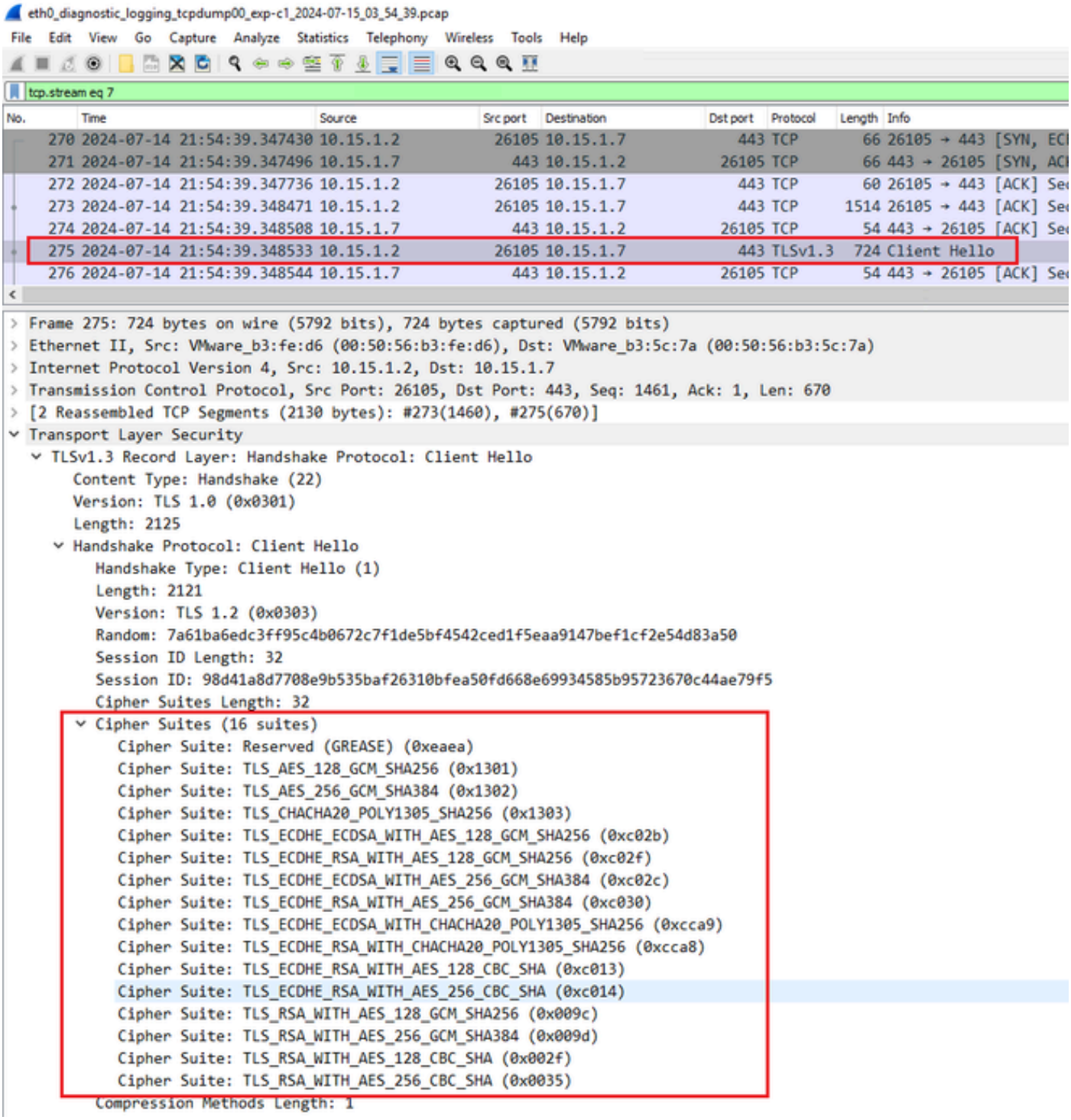
Het TLS-handshake-proces omvat een ClientHello-pakket dat door het clientapparaat is verzonden en de lijst met de door het apparaat ondersteunde algoritmen bevat volgens de ingestelde algoritme voor het verbindingsprotocol. De server bekijkt de lijst, vergelijkt het met zijn eigen lijst van toegestane algoritmen (bepaald door zijn eigen algoritme string), en kiest een algoritme dat beide systemen ondersteunen, dat gebruikt moet worden voor de versleutelde sessie. Dan antwoordt het met een ServerHello pakket dat op het gekozen algoritme wijst. Er zijn belangrijke verschillen tussen de TLS 1.2 en 1.3 handshake dialogen, maar het algoritme onderhandelingsmechanisme gebruikt dit zelfde principe in beide versies.

Dit is een voorbeeld van een TLS 1.3 algoritme onderhandeling tussen een webbrowser en Expressway op poort 443 zoals gezien in Wireshark:



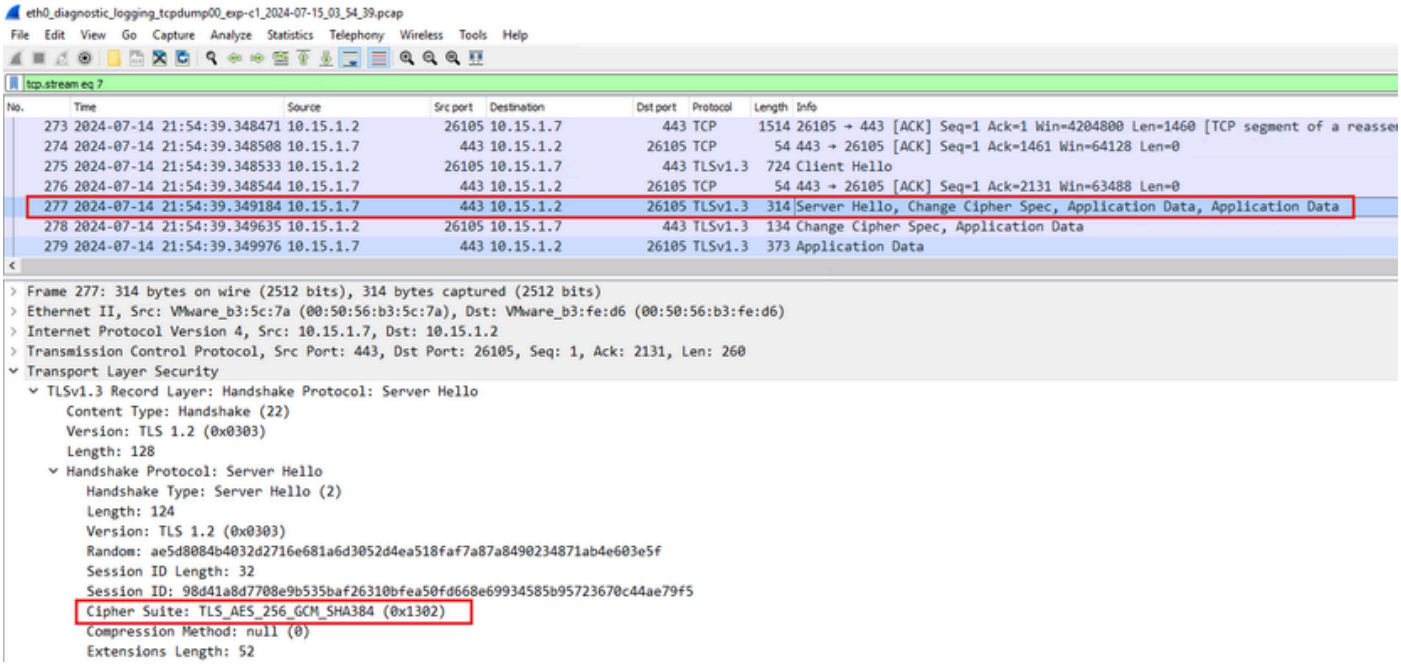
Voorbeeld van een TLS-handdruk in Wireshark

Eerst, verzendt browser een pakket ClientHello met de lijst van algoritmen het steunt:



Voorbeeld van een ClientHello-pakket in Wireshark

Expressway controleert zijn algoritme-string die geconfigureerd is voor het HTTPS-protocol en vindt een algoritme dat zowel door hem als door de client wordt ondersteund. In dit voorbeeld wordt het ECDHE-RSA-AES256-GCM-SHA384-algoritme geselecteerd. Expressway reageert met het ServerHello-pakket dat het geselecteerde algoritme aangeeft:



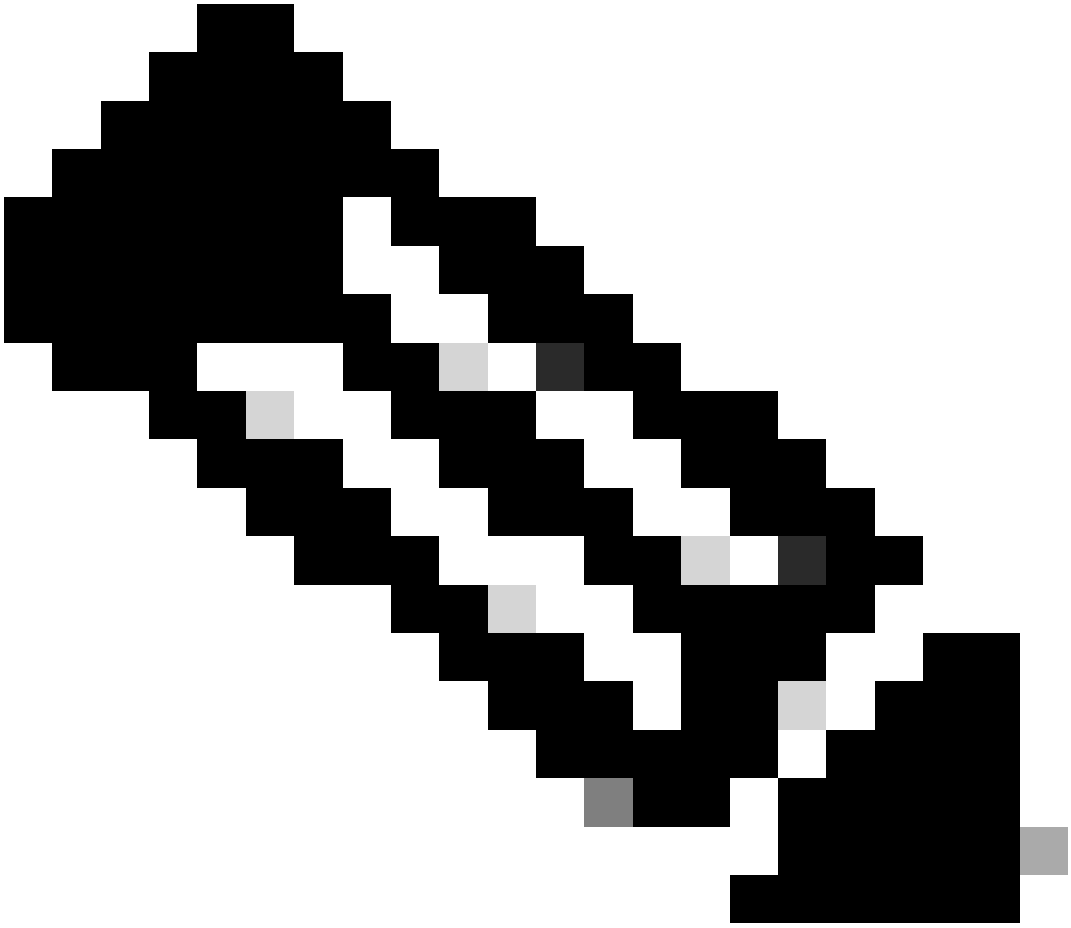
Voorbeeld van een ServerHello-pakket in Wireshark

Configureren

De OpenSSL-indeling voor algoritmen bevat verschillende speciale tekens om bewerkingen op de tekenreeks uit te voeren, zoals het verwijderen van een specifiek algoritme of een groep algoritmen die een gemeenschappelijke component delen. Aangezien het doel van deze aanpassingen meestal het verwijderen van algoritmen is, worden in deze voorbeelden de volgende tekens gebruikt:

- Het - teken, dat wordt gebruikt om algoritmen uit de lijst te verwijderen. Sommige of alle verwijderde algoritmen kunnen weer worden toegestaan door opties die later in de string verschijnen.
- Het ! teken, ook gebruikt om algoritmen uit de lijst te verwijderen. Wanneer het gebruiken van het, kunnen de verwijderde algoritmen niet opnieuw door een andere opties worden toegestaan die later in het koord verschijnen.
- Het : teken, dat fungeert als scheidingsteken tussen items in de lijst.

Beiden kunnen worden gebruikt om een algoritme van de string te verwijderen, maar ! wordt de voorkeur. Bekijk de OpenSSL-coderingsmanpage voor een volledige lijst met speciale tekens.



Opmerking: De OpenSSL-site verklaart dat bij gebruik van het !-teken "de verwijderde algoritmen nooit meer in de lijst kunnen verschijnen, zelfs als ze expliciet worden vermeld". Dit betekent niet dat de algoritmen permanent uit het systeem worden verwijderd, het verwijst naar de reikwijdte van de interpretatie van de algoritme string.

Een specifiek algoritme uitschakelen

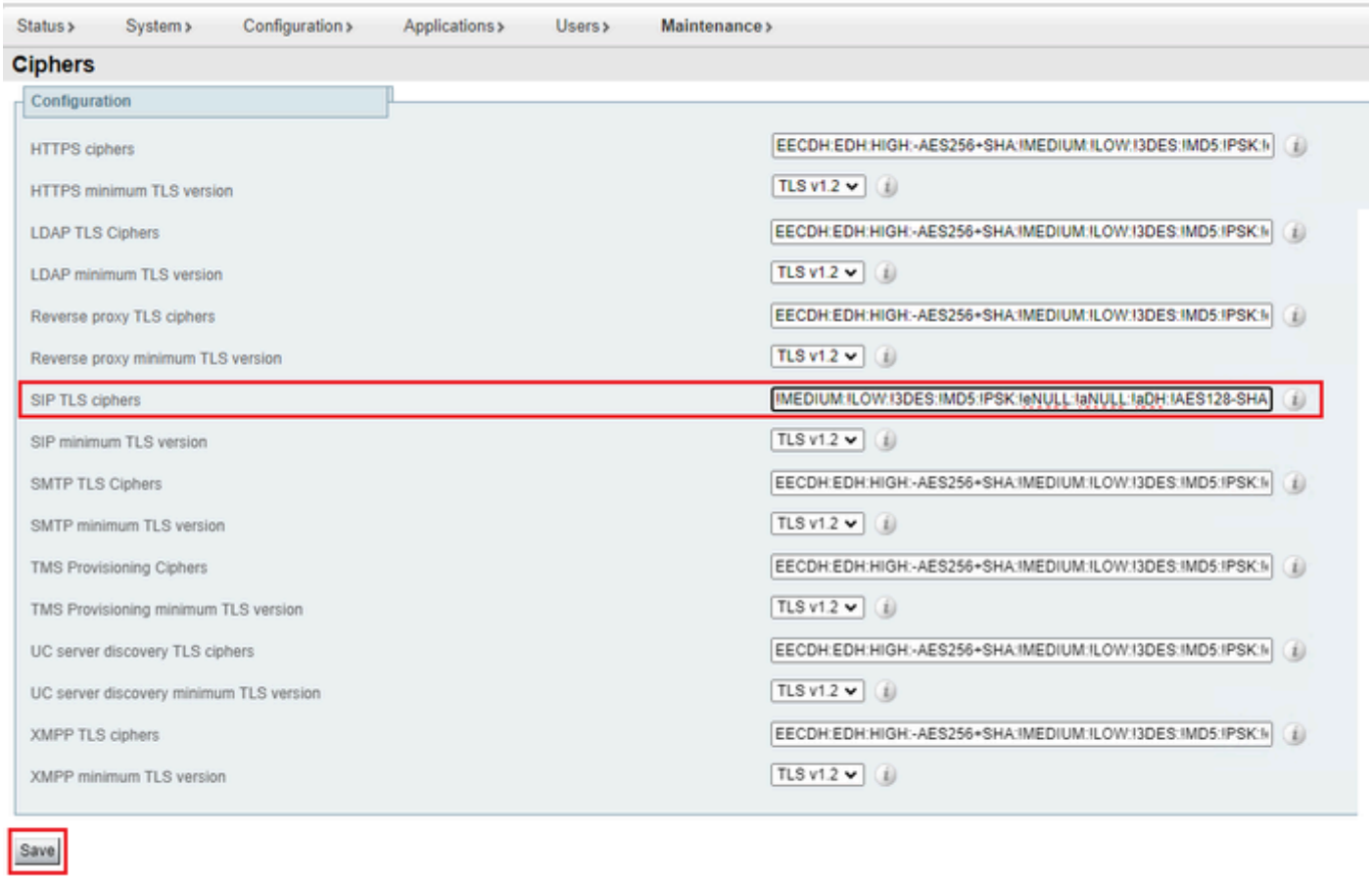
Als u een specifiek algoritme wilt uitschakelen, voegt u aan de standaardtekenreeks het : scheidingsteken, het ! of -teken en de naam van het algoritme die moet worden uitgeschakeld toe. De algoritmennaam moet voldoen aan de OpenSSL-naamgevingsindeling die beschikbaar is in het OpenSSL-coderingsmanagment. Als u bijvoorbeeld het AES128-SHA-algoritme voor SIP-verbindingen moet uitschakelen, moet u een algoritme als volgt configureren:

<#root>

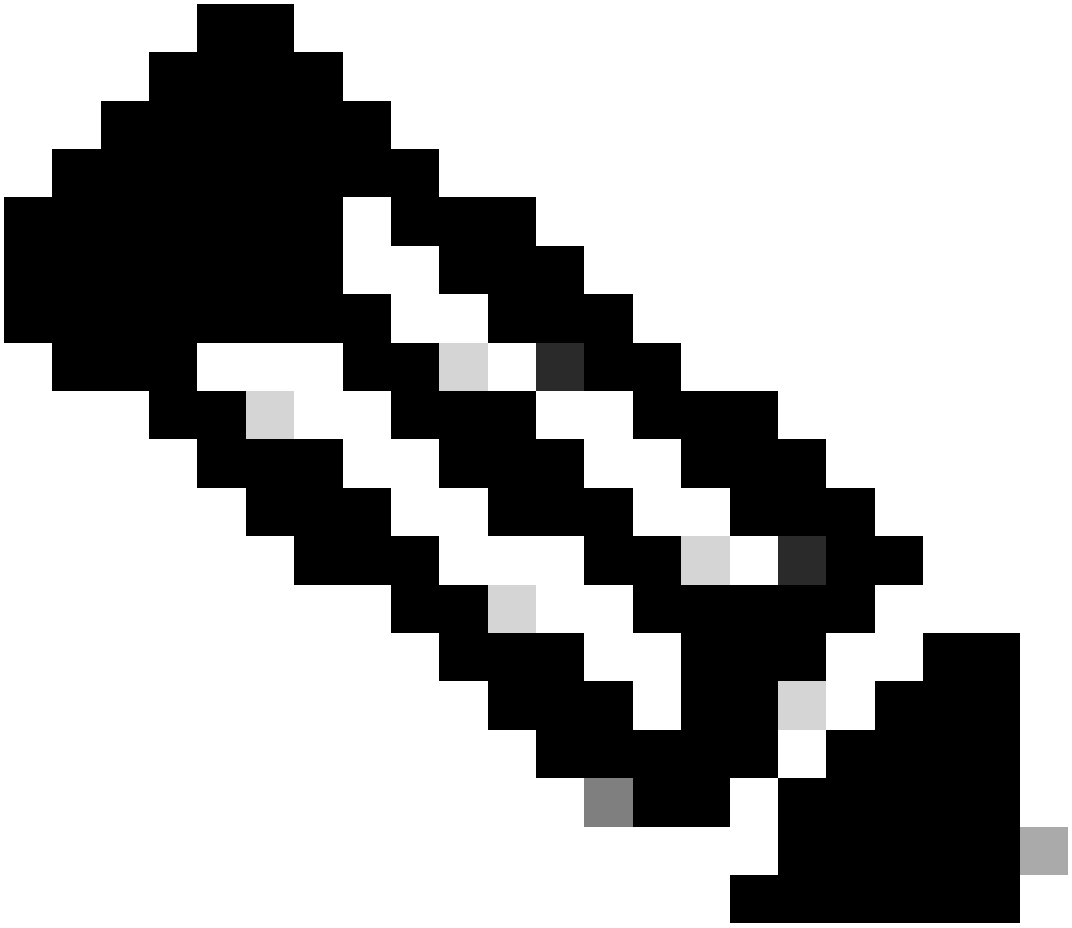
EECDH:EDH:HIGH:-AES256+SHA:!MEDIUM:!LOW:!3DES:!MD5:!PSK:!eNULL:!aNULL:!aDH

:!AES128-SHA

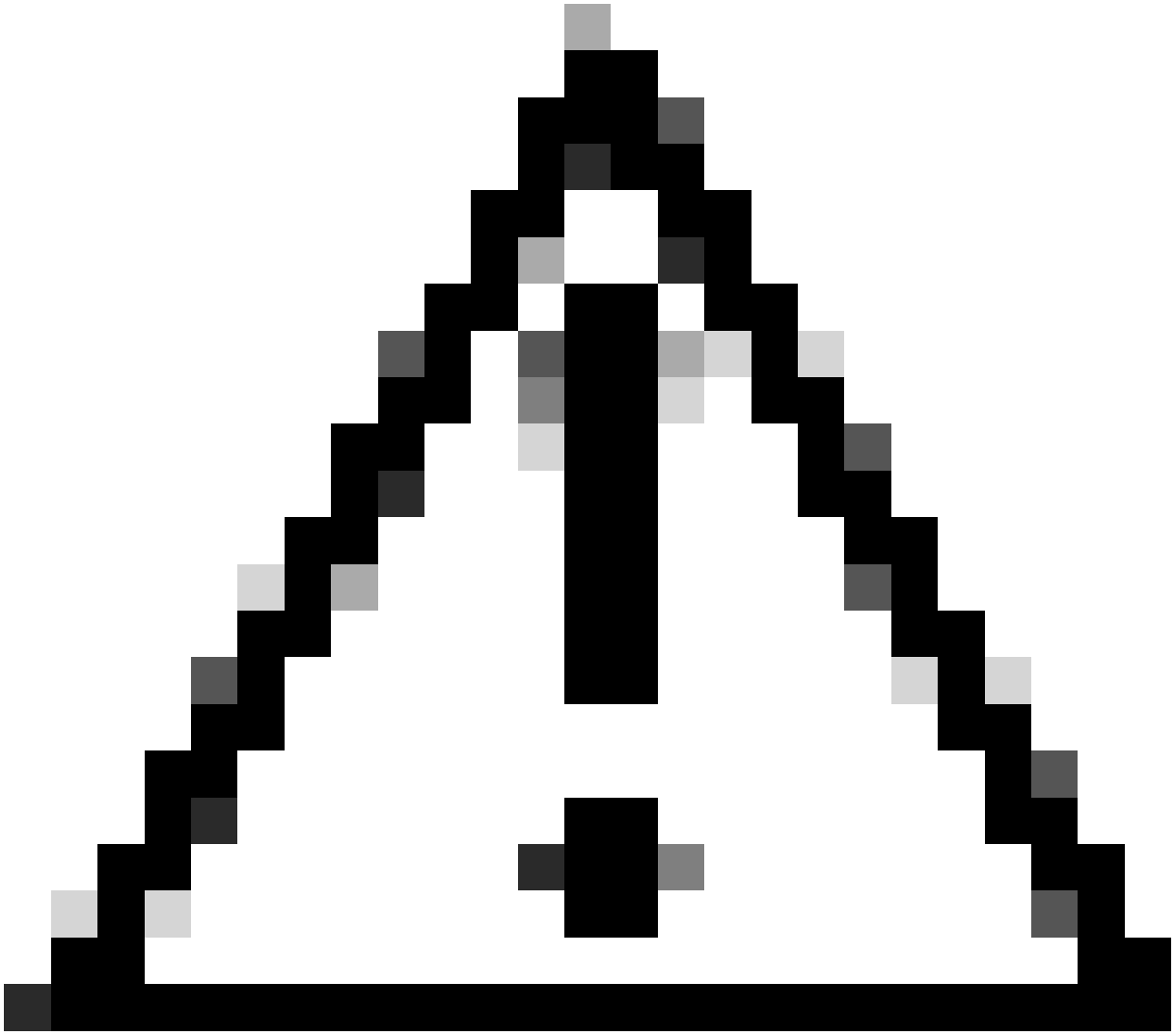
Ga vervolgens naar de pagina Expressway-webbeheer, navigeer naar Onderhoud > Beveiliging > Cijfers, wijs de aangepaste tekenreeks toe aan de vereiste protocollen en klik op Opslaan. De nieuwe configuratie kan alleen worden toegepast als het systeem opnieuw is opgestart. In dit voorbeeld wordt de aangepaste tekenreeks toegewezen aan het SIP-protocol onder SIP TLS-algoritmen:



De pagina van de Instellingen van het algoritme op het Portaal van Admin van het Web Expressway



Opmerking: in het geval van een Expressway-cluster, breng de wijzigingen alleen aan op de primaire server. De nieuwe configuratie wordt gerepliceerd naar de rest van de clusterleden.



Waarschuwing: gebruik de aanbevolen volgorde voor het opnieuw opstarten van clusters die in de implementatiehandleiding voor Cisco Expressway Cluster Creation and Maintenance wordt geboden. Begin door de primaire server opnieuw te starten, wacht tot hij via de web interface toegankelijk is en doe hetzelfde met elke peer volgens de lijst die is ingesteld onder System > Clustering.

Een groep algoritmen uitschakelen met een gemeenschappelijk algoritme

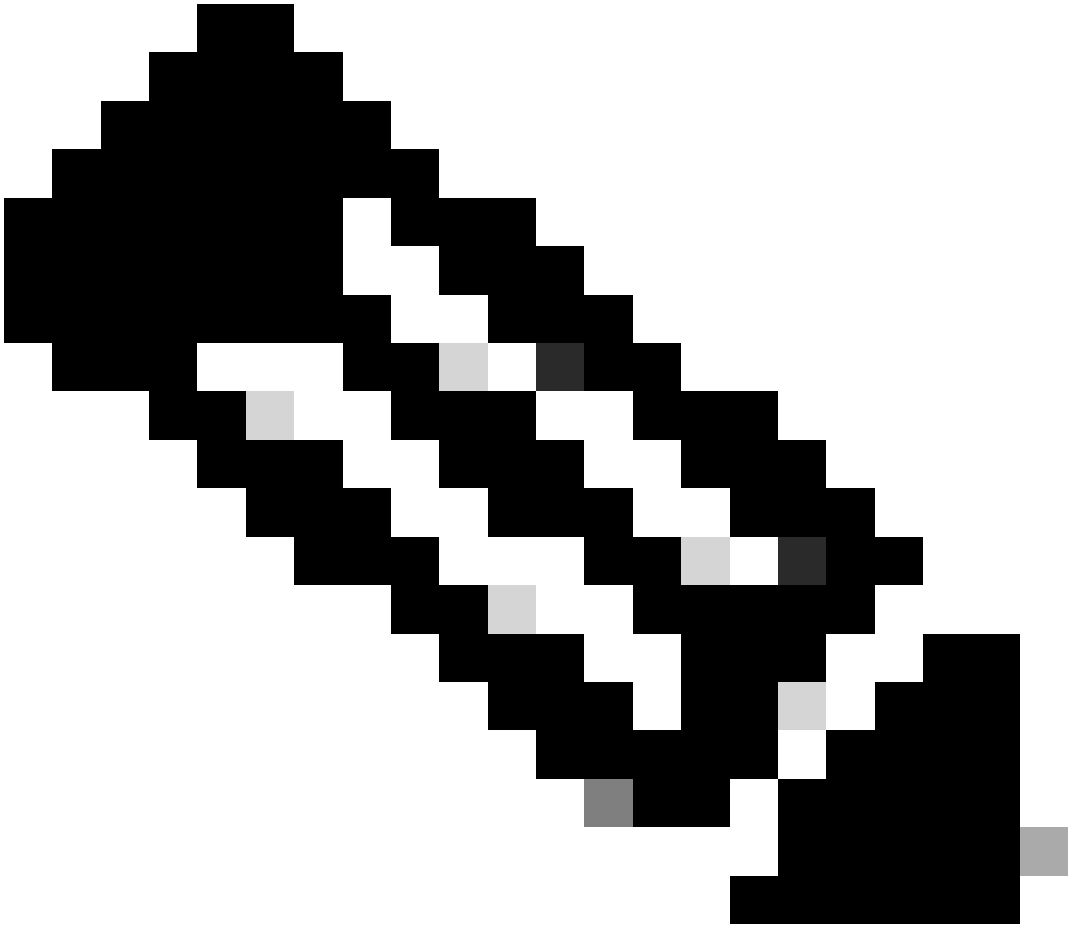
Als u een groep algoritmes wilt uitschakelen met behulp van een gemeenschappelijk algoritme, voegt u aan de standaardstring het : separator, het ! of -teken, en de algoritmenaam die moet worden uitgeschakeld toe. De ondersteunde algoritmenamen zijn beschikbaar in de OpenSSL-coderingsmanpage. Als u bijvoorbeeld alle algoritmen die DHE gebruiken moet uitschakelen, moet u een algoritme als deze configureren:

<#root>

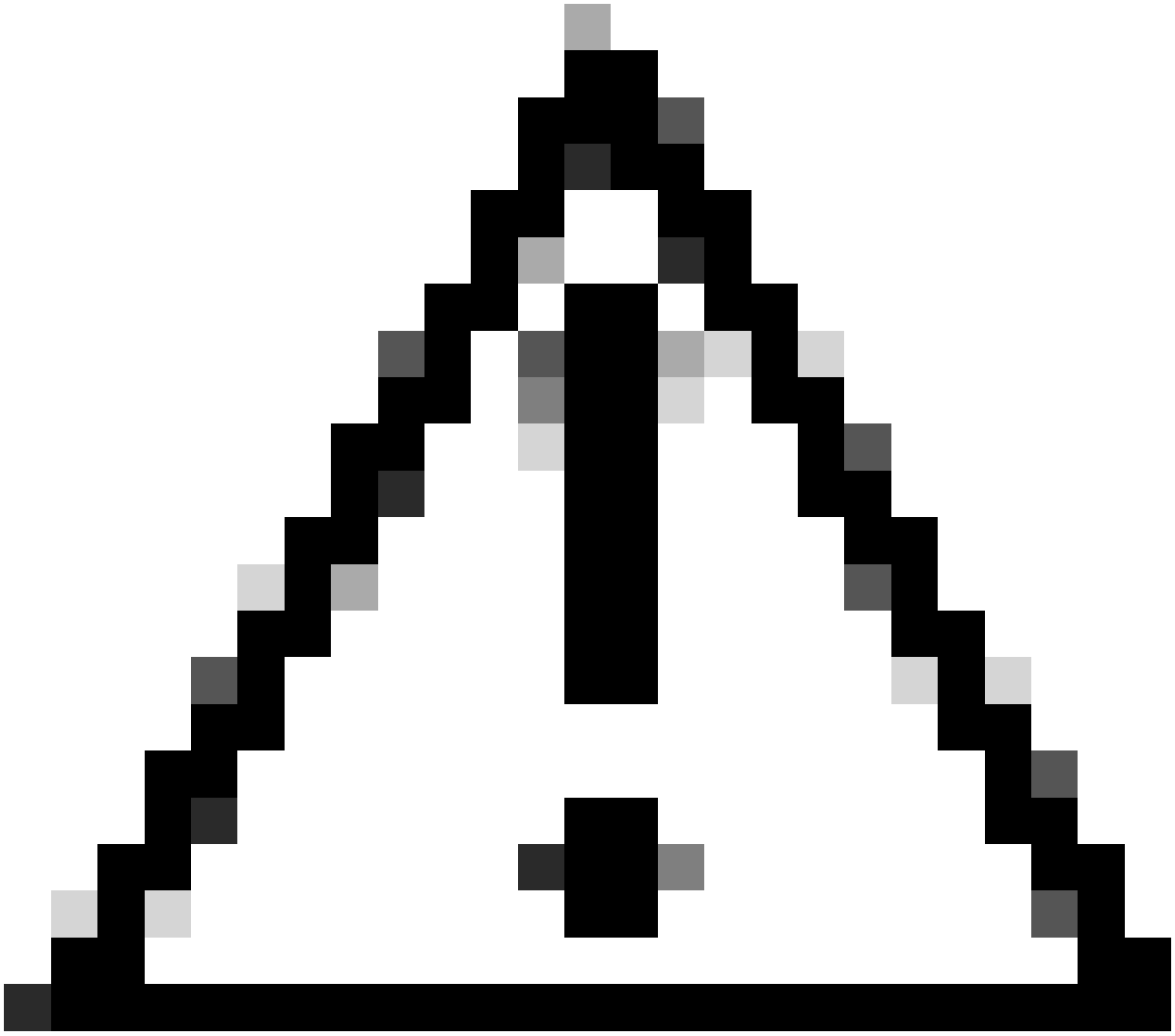
```
EECDH:EDH:HIGH:-AES256+SHA:!MEDIUM:!LOW:!3DES:!MD5:!PSK:!eNULL:!aNULL:!aDH
```

:!DHE

Navigeer naar de pagina Expressway web admin, navigeer naar Onderhoud > Beveiliging > Cijfers, wijs de aangepaste tekenreeks toe aan het/de vereiste protocol(n) en klik op Opslaan. De nieuwe configuratie kan alleen worden toegepast als het systeem opnieuw is opgestart.



Opmerking: in het geval van een Expressway-cluster, breng de wijzigingen alleen aan op de primaire server. De nieuwe configuratie wordt gerepliceerd naar de rest van de clusterleden.



Waarschuwing: gebruik de aanbevolen volgorde voor het opnieuw opstarten van clusters die in de implementatiehandleiding voor Cisco Expressway Cluster Creation and Maintenance wordt geboden. Begin door de primaire server opnieuw te starten, wacht tot hij via de web interface toegankelijk is en doe hetzelfde met elke peer volgens de lijst die is ingesteld onder System > Clustering.

Verifiëren

Inspecteer de lijst met cifers die zijn toegestaan door de reeks coderingen

U kunt de aangepaste algoritme-string controleren met behulp van de opdracht openssl -V "<algoritme string>". Bekijk de output om te bevestigen dat de ongewenste algoritmen niet meer vermeld na de veranderingen zijn. In dit voorbeeld, de EECDH:EDH:HIGH:-AES256+SHA:!MEDIUM:!LOW:!3DES:!MD5:!PSK:!NULL:!ANULL:!aDH:!DHE-coderingsstring wordt geïnspecteerd. De opdrachtoutput bevestigt dat de string geen algoritme toestaat die gebruik maakt van het DHE algoritme:

<#root>

```
    ~ # openssl ciphers -V "EECDH:EDH:HIGH:-AES256+SHA:!MEDIUM:!LOW:!3DES:!MD5:!PSK:!eNULL:!aNULL:!aDH
:!DHE
"
0x13,0x02 - TLS_AES_256_GCM_SHA384 TLSv1.3 Kx=any Au=any Enc=AESGCM(256) Mac=AEAD
0x13,0x03 - TLS_CHACHA20_POLY1305_SHA256 TLSv1.3 Kx=any Au=any Enc=CHACHA20/POLY1305(256) Mac=AEAD
0x13,0x01 - TLS_AES_128_GCM_SHA256 TLSv1.3 Kx=any Au=any Enc=AESGCM(128) Mac=AEAD
0xC0,0x2C - ECDHE-ECDSA-AES256-GCM-SHA384 TLSv1.2 Kx=ECDH Au=ECDSA Enc=AESGCM(256) Mac=AEAD
0xC0,0x30 - ECDHE-RSA-AES256-GCM-SHA384 TLSv1.2 Kx=ECDH Au=RSA Enc=AESGCM(256) Mac=AEAD
0xCC,0xA9 - ECDHE-ECDSA-CHACHA20-POLY1305 TLSv1.2 Kx=ECDH Au=ECDSA Enc=CHACHA20/POLY1305(256) Mac=AEAD
0xCC,0xA8 - ECDHE-RSA-CHACHA20-POLY1305 TLSv1.2 Kx=ECDH Au=RSA Enc=CHACHA20/POLY1305(256) Mac=AEAD
0xC0,0xAD - ECDHE-ECDSA-AES256-CCM TLSv1.2 Kx=ECDH Au=ECDSA Enc=AESCCM(256) Mac=AEAD
0xC0,0x2B - ECDHE-ECDSA-AES128-GCM-SHA256 TLSv1.2 Kx=ECDH Au=ECDSA Enc=AESGCM(128) Mac=AEAD
0xC0,0x2F - ECDHE-RSA-AES128-GCM-SHA256 TLSv1.2 Kx=ECDH Au=RSA Enc=AESGCM(128) Mac=AEAD
0xC0,0xAC - ECDHE-ECDSA-AES128-CCM TLSv1.2 Kx=ECDH Au=ECDSA Enc=AESCCM(128) Mac=AEAD
0xC0,0x24 - ECDHE-ECDSA-AES256-SHA384 TLSv1.2 Kx=ECDH Au=ECDSA Enc=AES(256) Mac=SHA384
0xC0,0x28 - ECDHE-RSA-AES256-SHA384 TLSv1.2 Kx=ECDH Au=RSA Enc=AES(256) Mac=SHA384
0xC0,0x23 - ECDHE-ECDSA-AES128-SHA256 TLSv1.2 Kx=ECDH Au=ECDSA Enc=AES(128) Mac=SHA256
0xC0,0x27 - ECDHE-RSA-AES128-SHA256 TLSv1.2 Kx=ECDH Au=RSA Enc=AES(128) Mac=SHA256
0xC0,0x09 - ECDHE-ECDSA-AES128-SHA TLSv1 Kx=ECDH Au=ECDSA Enc=AES(128) Mac=SHA1
0xC0,0x13 - ECDHE-RSA-AES128-SHA TLSv1 Kx=ECDH Au=RSA Enc=AES(128) Mac=SHA1
0x00,0x9D - AES256-GCM-SHA384 TLSv1.2 Kx=RSA Au=RSA Enc=AESGCM(256) Mac=AEAD
0xC0,0x9D - AES256-CCM TLSv1.2 Kx=RSA Au=RSA Enc=AESCCM(256) Mac=AEAD
0x00,0x9C - AES128-GCM-SHA256 TLSv1.2 Kx=RSA Au=RSA Enc=AESGCM(128) Mac=AEAD
0xC0,0x9C - AES128-CCM TLSv1.2 Kx=RSA Au=RSA Enc=AESCCM(128) Mac=AEAD
0x00,0x3D - AES256-SHA256 TLSv1.2 Kx=RSA Au=RSA Enc=AES(256) Mac=SHA256
0x00,0x3C - AES128-SHA256 TLSv1.2 Kx=RSA Au=RSA Enc=AES(128) Mac=SHA256
0x00,0x2F - AES128-SHA SSLv3 Kx=RSA Au=RSA Enc=AES(128) Mac=SHA1
~ #
```

Test een TLS-verbinding door te onderhandelen over een uitgeschakeld algoritme

U kunt de opdracht openssl s_client gebruiken om te verifiëren dat een verbindingspoging met een uitgeschakeld algoritme wordt geweigerd. Gebruik de optie -connect om uw Expressway-adres en -poort op te geven en gebruik de optie -algoritme om het enkele algoritme op te geven waarover de client tijdens de TLS-handdruk moet onderhandelen:

openssl s_client -connect <adres>:<port> -algoritme <algoritme> -no_tls1_3

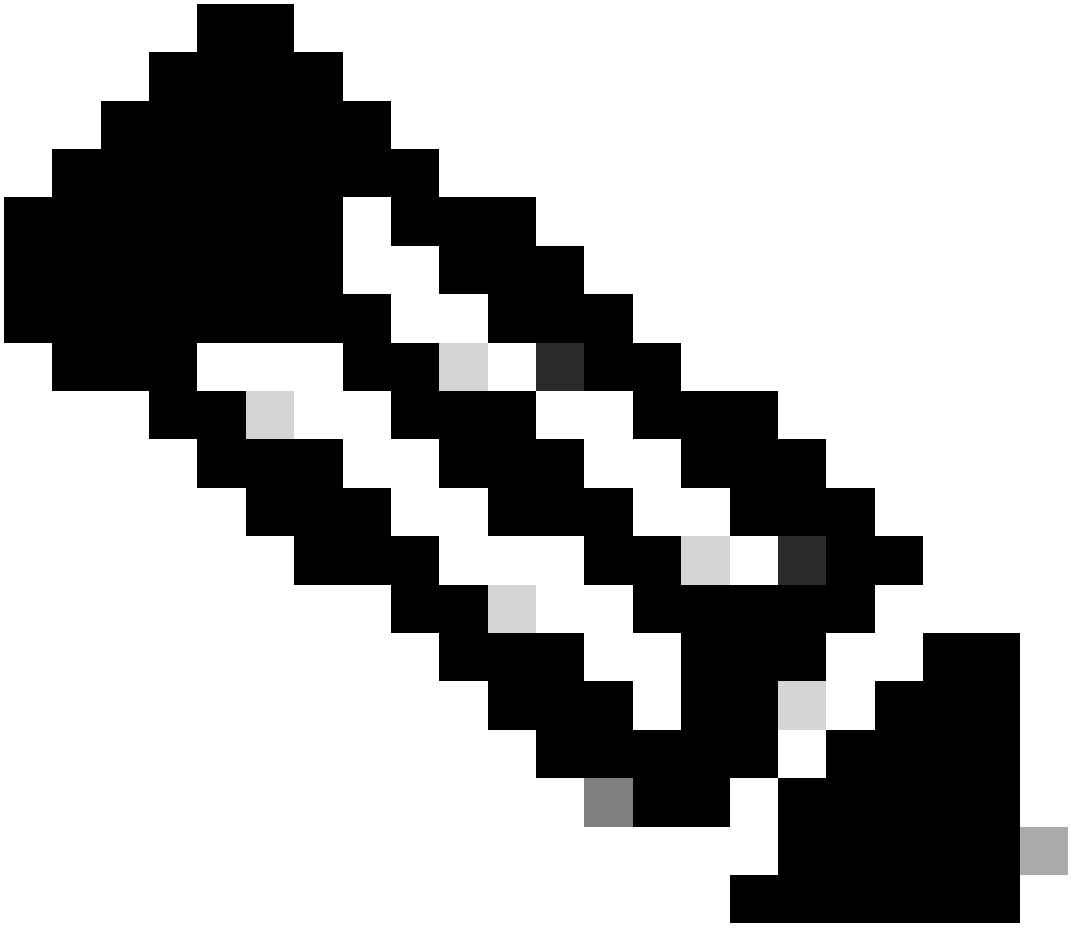
In dit voorbeeld wordt een TLS-verbinding naar Expressway geprobeerd vanaf een Windows-pc met geïnstalleerde openssl. De PC onderhandelt als client alleen over het ongewenste DHE-RSA-AES256-CCM algoritme, dat gebruik maakt van het DHE algoritme:

<#root>

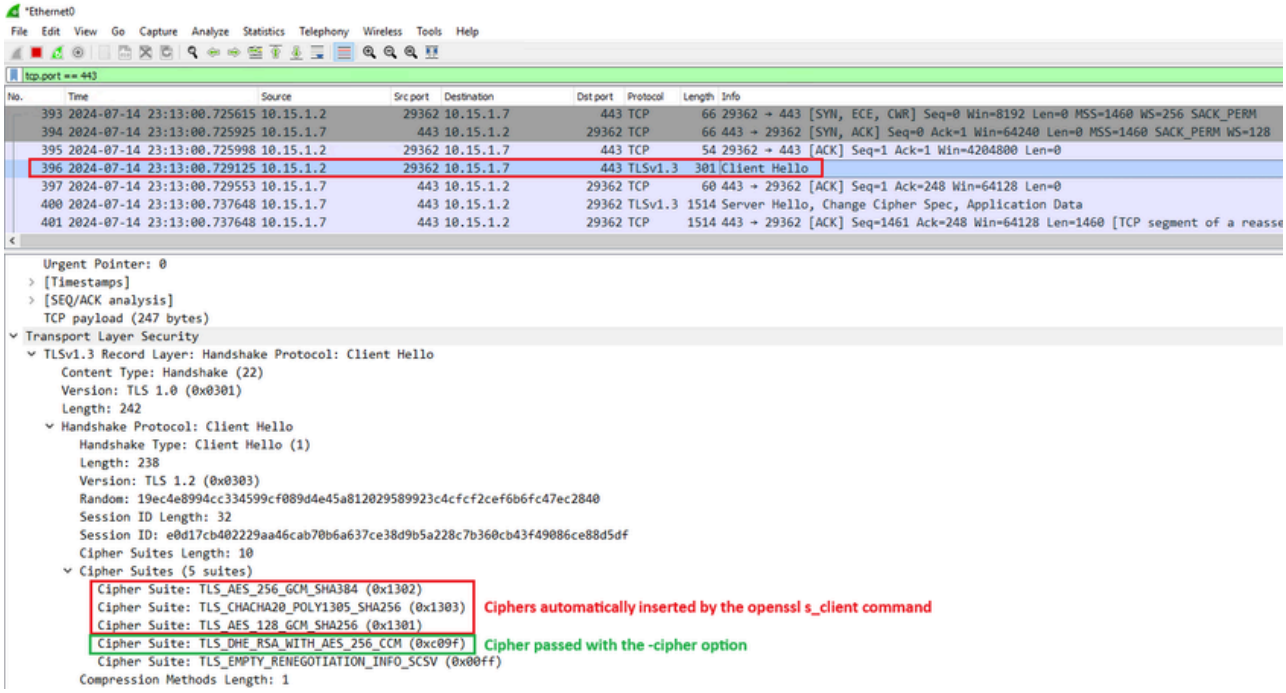
```
C:\Users\Administrator>

openssl s_client -connect exp.example.com:443 -cipher DHE-RSA-AES256-CCM -no_tls1_3


Connecting to 10.15.1.7
CONNECTED(00000154)
D0130000:error:0A000410:SSL routines:ssl3_read_bytes:
```

```
ssl/tls alert handshake failure

:..\ssl\record\rec_layer_s3.c:865:

SSL alert number 40


---
no peer certificate available
---
No client certificate CA names sent
---
SSL handshake has read 7 bytes and written 118 bytes
Verification: OK
---
New, (NONE), Cipher is (NONE)
Secure Renegotiation IS NOT supported
No ALPN negotiated
SSL-Session:
Protocol : TLSv1.2
Cipher : 0000
Session-ID:
Session-ID-ctx:
Master-Key:
PSK identity: None
PSK identity hint: None
SRP username: None
Start Time: 1721019437
Timeout : 7200 (sec)
Verify return code: 0 (ok)
Extended master secret: no
---

C:\Users\Administrator>
```

De opdrachtoutput laat zien dat de verbindingspoging mislukt met de foutmelding "ssl/tls alert handshake failure:..\ssl\record\rec_layer_s3.c:865:SSL alert number 40", omdat de Expressway is geconfigureerd om de EECDH:EDH:HIGH:-AES256+SHA:!MEDIUM:!LOW:!3DES:!MD5:!PSK:!NULL:!ANULL:!aDH: HTTDHE-algoritme voor PS-verbindingen, waardoor algoritmes worden uitgeschakeld die het DHE-algoritme gebruiken.



Opmerking: om te zorgen dat tests met de opdracht openssl s_client werken zoals uitgelegd, moet de optie -no_tls1_3 worden doorgegeven aan de opdracht. Indien niet opgenomen, voegt de client automatisch TLS 1.3-algoritmen in het ClientHello-pakket in:



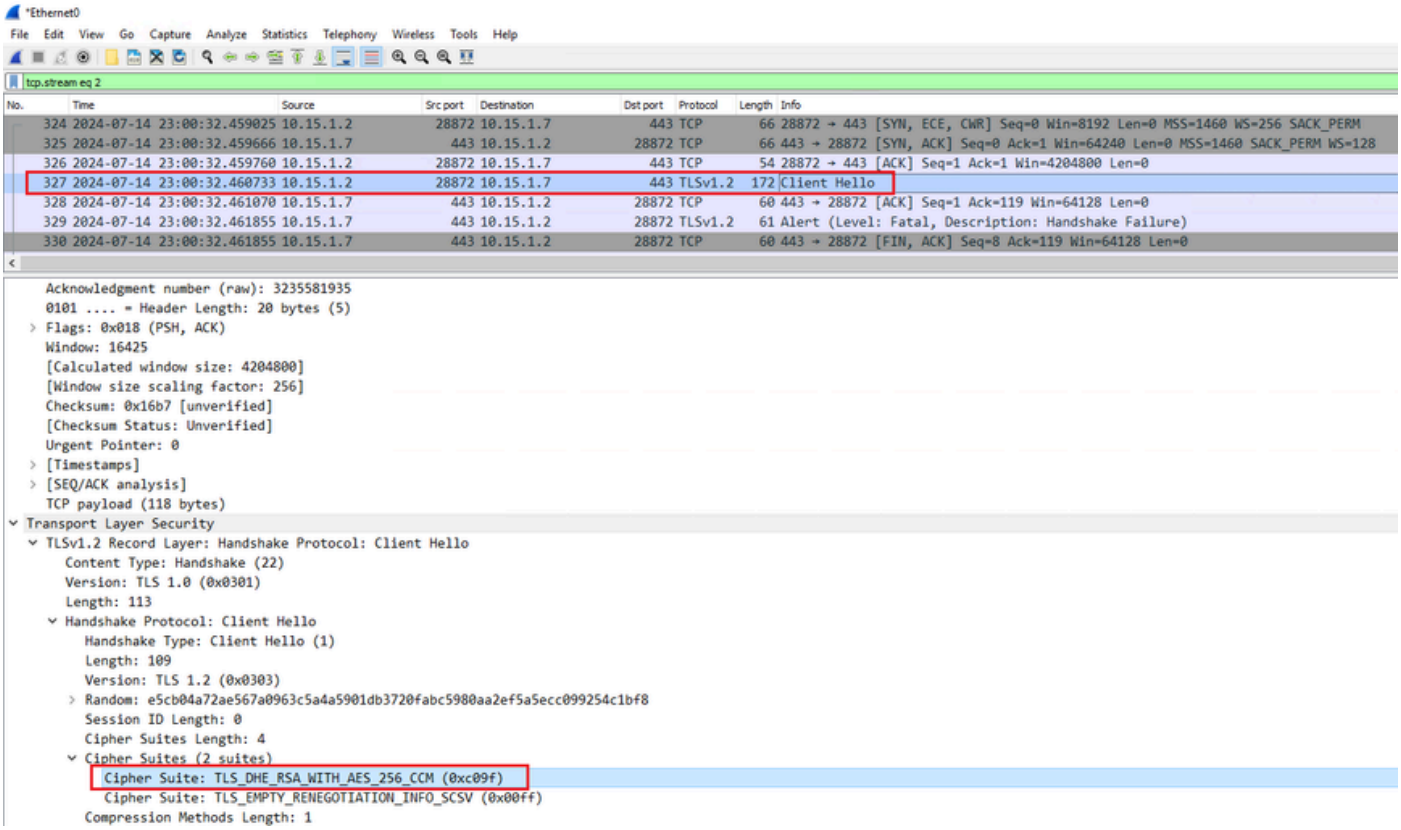
ClientHello-pakket met automatisch toegevoegde coderingen

Als de doel Expressway ondersteunt die algoritmen, kan een van hen worden gekozen in plaats van het specifieke algoritme dat u moet testen. De verbinding is succesvol, wat u ertoe kan brengen te geloven dat een verbinding mogelijk was door het gehandicapte algoritme te gebruiken dat tot het bevel met de optie - algoritme wordt overgegaan.

Inspecteer een pakketvastlegging van een TLS-handdruk met een uitgeschakeld coderingssysteem

U kunt een pakketopname, van het testapparaat of van de snelweg verzamelen terwijl u een verbindingstest uitvoert met behulp van een van de uitgeschakelde algoritmen. U kunt het dan inspecteren met Wireshark om de handshake gebeurtenissen verder te analyseren.

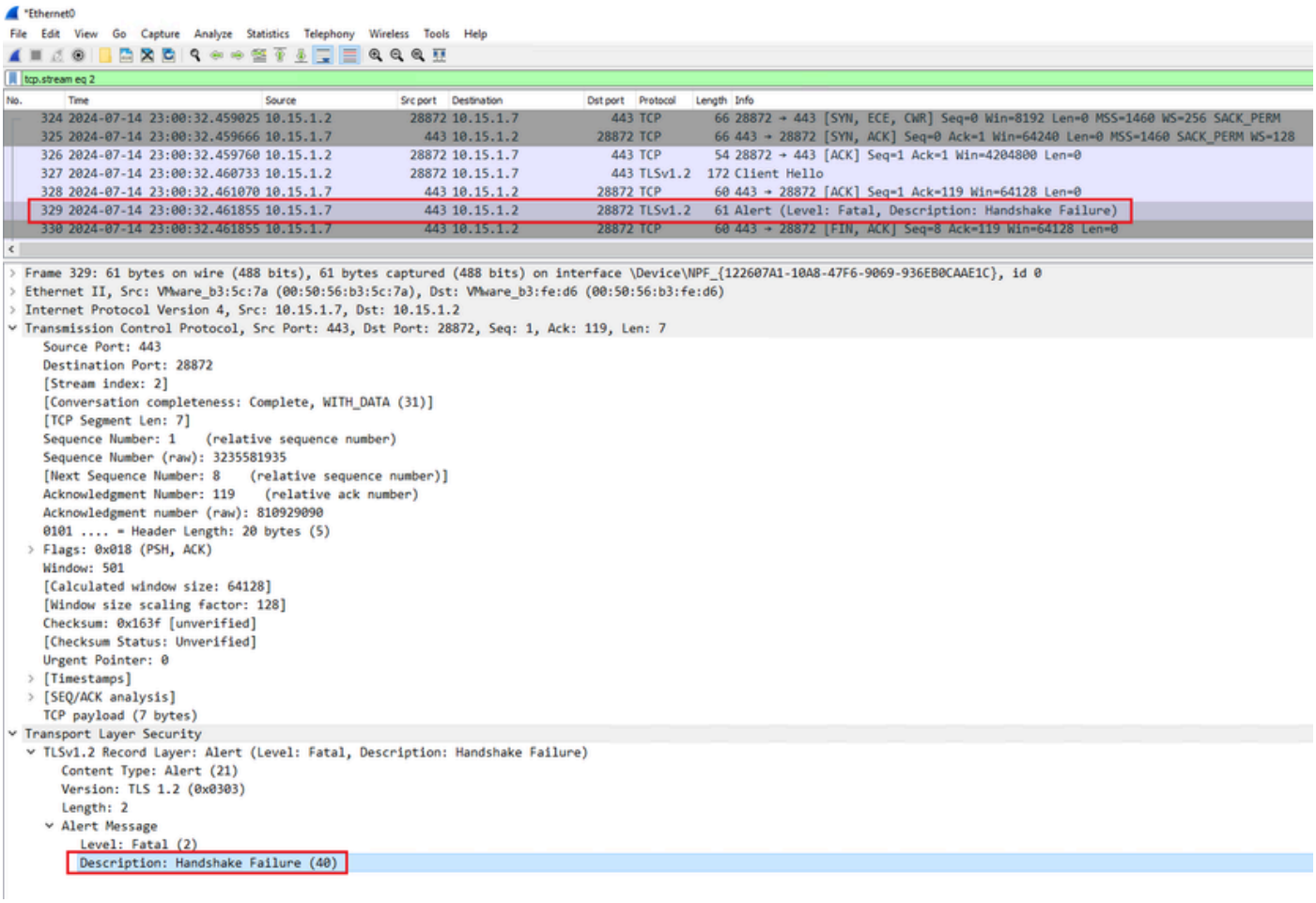
Zoek de ClientHello verzonden door het testapparaat. Bevestig dat alleen wordt onderhandeld over het ongewenste testalgoritme, in dit voorbeeld een algoritme met behulp van DHE:



Voorbeeld van een ClientHello-pakket in Wireshark

:

Bevestig dat Expressway reageert met een fataal TLS-waarschuwingspakket en de verbinding weigert. In dit voorbeeld, aangezien Expressway geen DHE-algoritmen ondersteunt per de ingestelde algoritme voor het HTTPS-protocol, reageert het met een fataal TLS-waarschuwingspakket met foutcode 40.



Een TLS-fataal waarschuwingspakket in Wireshark

Gerelateerde informatie

- OpenSSL-coderingsbeheer
- Cisco Expressway Administrator Guide (X15.0) - Hoofdstuk: Beveiligingsbeheer - Minimale TLS-versie configureren en aangepaste versies installeren

Over deze vertaling

Cisco heeft dit document vertaald via een combinatie van machine- en menselijke technologie om onze gebruikers wereldwijd ondersteuningscontent te bieden in hun eigen taal. Houd er rekening mee dat zelfs de beste machinevertaling niet net zo nauwkeurig is als die van een professionele vertaler. Cisco Systems, Inc. is niet aansprakelijk voor de nauwkeurigheid van deze vertalingen en raadt aan altijd het oorspronkelijke Engelstalige document (link) te raadplegen.