تخصيص تكوين تشفير Expressway SSL

المحتويات

المقدمة

يصف هذا المستند الخطوات اللازمة لتخصيص سلاسل التشفير التي تم تكوينها مسبقا على Expressway.

المتطلبات الأساسية

المتطلبات

توصي Cisco بأن تكون لديك معرفة بالمواضيع التالية:

- Expressway من Cisco أو Cisco VCS.
- بروتوكول TLS.

المكونات المستخدمة

تستند المعلومات الواردة في هذا المستند إلى إصدارات البرامج والمكونات المادية التالية:

- Cisco Expressway، الإصدار x15.0.2.

تم إنشاء المعلومات الواردة في هذا المستند من الأجهزة الموجودة في بيئة معملية خاصة. بدأت جميع الأجهزة المُستخدمة في هذا المستند بتكوين ممسوح (افتراضي). إذا كانت شبكتك قيد التشغيل، فتأكد من فهمك للتأثير المحتمل لأي أمر.

معلومات أساسية

يتضمن تكوين ExpressWay الافتراضي سلاسل تشفير تم تكوينها مسبقا، والتي تعمل، لأسباب التوافق، على تمكين دعم بعض التشفير الذي يمكن اعتباره ضعيفا بموجب بعض سياسات أمان المؤسسة. من الممكن تخصيص سلاسل التشفير من أجل ضبطها لتلائم السياسات المحددة لكل بيئة.

في Expressway، من الممكن تكوين سلسلة تشفير مستقلة لكل من هذه البروتوكولات:

- HTTPS
- LDAP
- الوكيل العكسي
- SIP
- SMTP
- توفير TMS
- اكتشاف خادم UC
- XMPP

تطيع سلاسل التشفير تنسيق OpenSSL الموصوف في صفحة إدارة تشفير OpenSSL. يأتي إصدار Expressway الحالي X15.0.2 مع السلسلة الافتراضية eECDH:EDH:HIGH:-AES256+SHA:!MEDIUM:!LOW:!3DES:!MD5:!PSK:!eNULL:!aNULL:!aDH تم تكوينه مسبقا لجميع البروتوكولات على حد سواء. من صفحة إدارة الويب، تحت صيانة>الأمان > التشفير، يمكنك تعديل سلسلة التشفير المعينة لكل بروتوكول، إلى إضافة أو إزالة شفرة معينة أو مجموعات من المشفرات باستخدام خوارزمية عامة.

فحص سلسلة التشفير

باستخدام الأمر OpenSSL ciphers - v "<cipher string>"، يمكنك إخراج قائمة تحتوي على جميع الشفرات التي تسمح بها سلسلة معينة، وهو ما يكون مفيدا لفحص الشفرة وضوئيا. يوضح هذا المثال الإخراج عند فحص سلسلة تشفير Expressway الافتراضية:

```
<#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
```

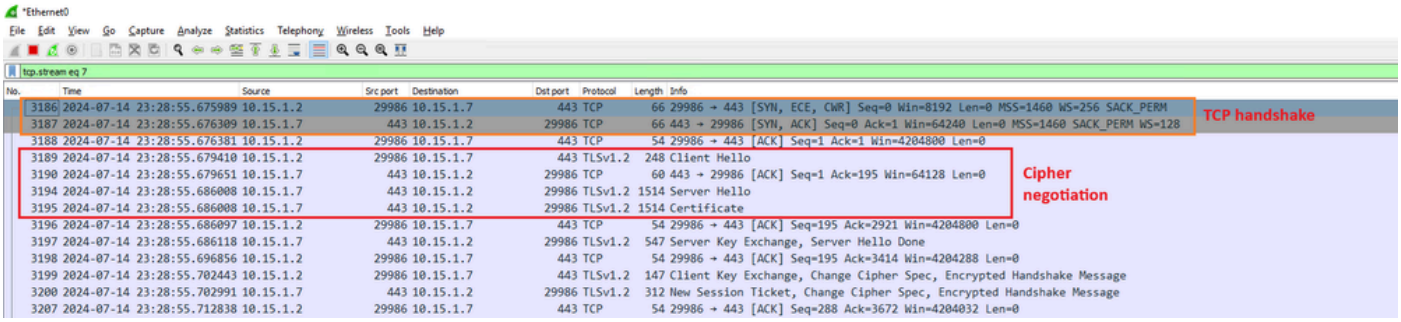
```
0xC0,0x23 - ECDHE-ECDSA-AES128-SHA256 TLSv1.2 Kx=ECDH Au=ECDSA Enc=AES(128) Mac=SHA256
0xC0,0x27 - ECDHE-RSA-AES128-SHA256 TLSv1.2 Kx=ECDH Au=RSA Enc=AES(128) Mac=SHA256
0xC0,0x09 - ECDHE-ECDSA-AES128-SHA TLSv1 Kx=ECDH Au=ECDSA Enc=AES(128) Mac=SHA1
0xC0,0x13 - ECDHE-RSA-AES128-SHA TLSv1 Kx=ECDH 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
~ #
```

فحص تفاوض التشفير في مصافحة TLS باستخدام التقاط الحزمة

من خلال التقاط تفاوض TLS في التقاط حزمة، يمكنك فحص تفاصيل تفاوض التشفير باستخدام Wireshark.

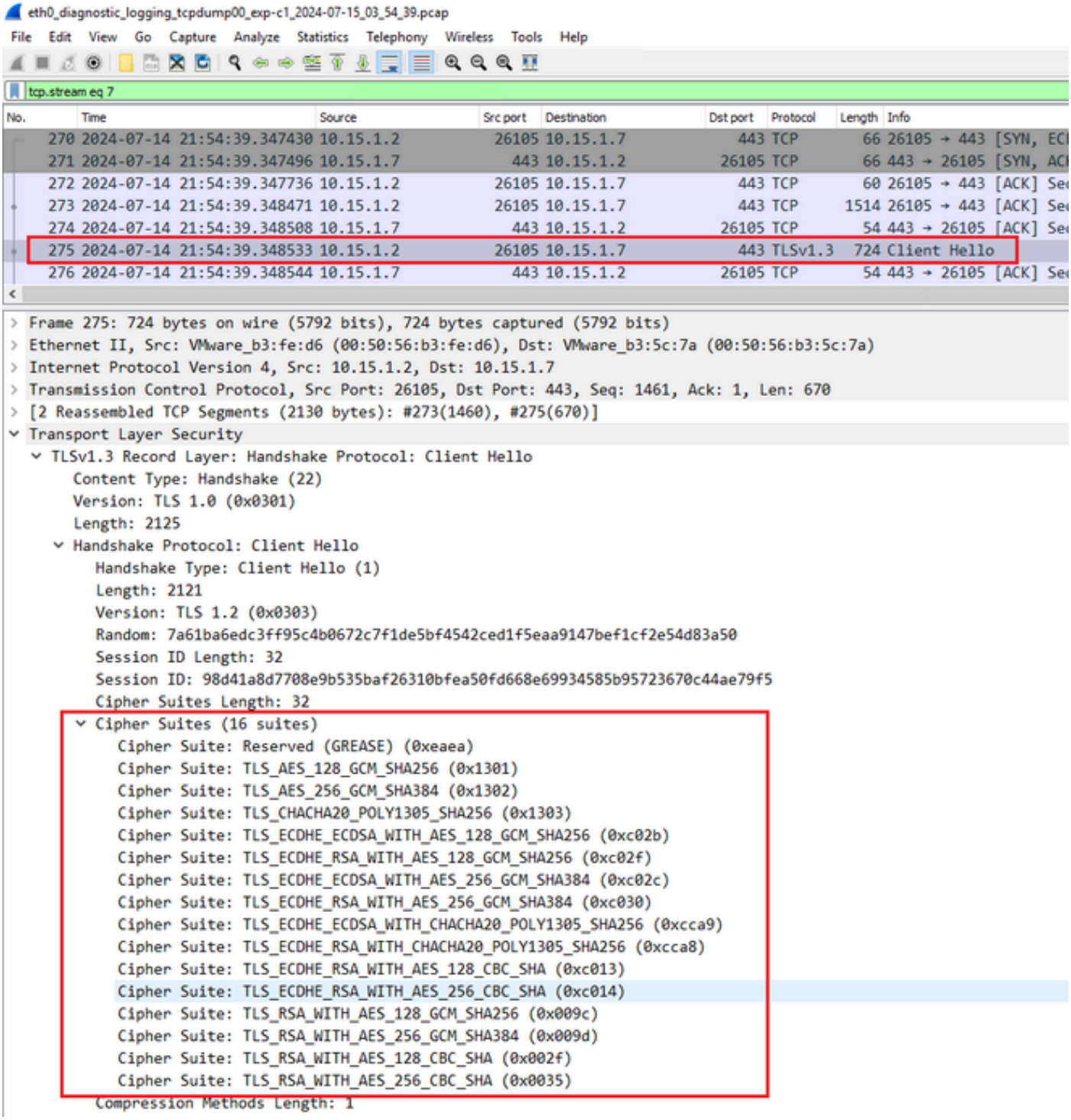
تتضمن عملية مصافحة TLS حزمة ClientHello مرسلة بواسطة جهاز العميل، مما يوفر قائمة بالمشفرات التي يدعمها وفقا لسلسلة شفرة وبروتوكول الاتصال التي تم تكوينها. يراجع الخادم القائمة، ويقارنها بقائمته الخاصة للشفرات المسموح بها (المحددة بواسطة سلسلة التشفير الخاصة به)، ويختار شفرة يدعمها كل النظامين، ليتم إستخدامها للجلسة المشفرة. ثم يستجيب مع حزمة ServerHello التي تشير إلى التشفير المختار. توجد إختلافات مهمة بين حواري TLS 1.2 و 1.3 للمصافحة، إلا أن آلية تفاوض التشفير تستخدم هذا المبدأ نفسه في كل الإصدارين.

هذا مثال على تفاوض تشفير TLS 1.3 بين مستعرض ويب و Expressway على المنفذ 443 كما هو موضح في Wireshark:



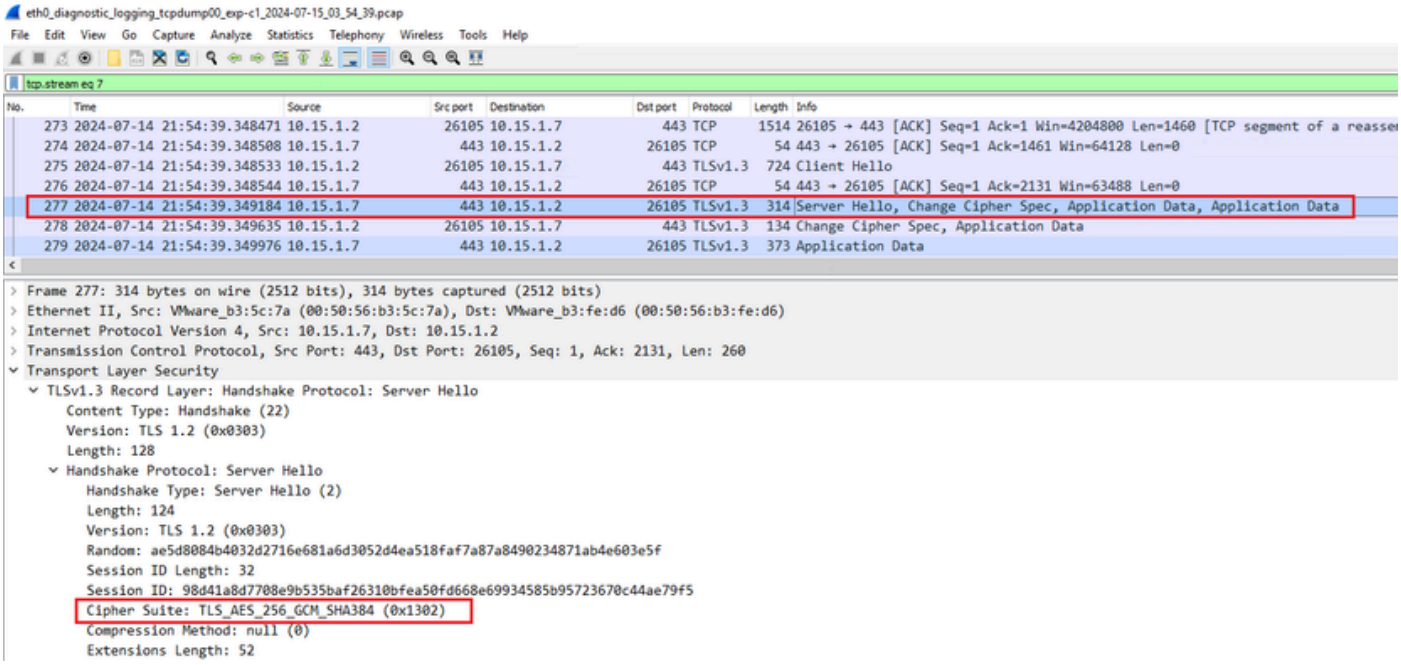
مثال على مصافحة TLS في Wireshark

أولا، يرسل المستعرض حزمة ClientHello مع قائمة من الشفرات التي يدعمها:



مثال على حزمة ClientHello في Wireshark

يتحقق Expressway من سلسلة التشفير الخاصة به التي تم تكوينها لبروتوكول HTTPS، ويبحث عن تشفير يدعم نفسه والعميل. في هذا المثال، يتم تحديد تشفير ECDHE-RSA-AES256-GCM-SHA384. يستجيب Expressway مع حزمة ServerHello الخاصة به التي تشير إلى الشفرة المحددة:



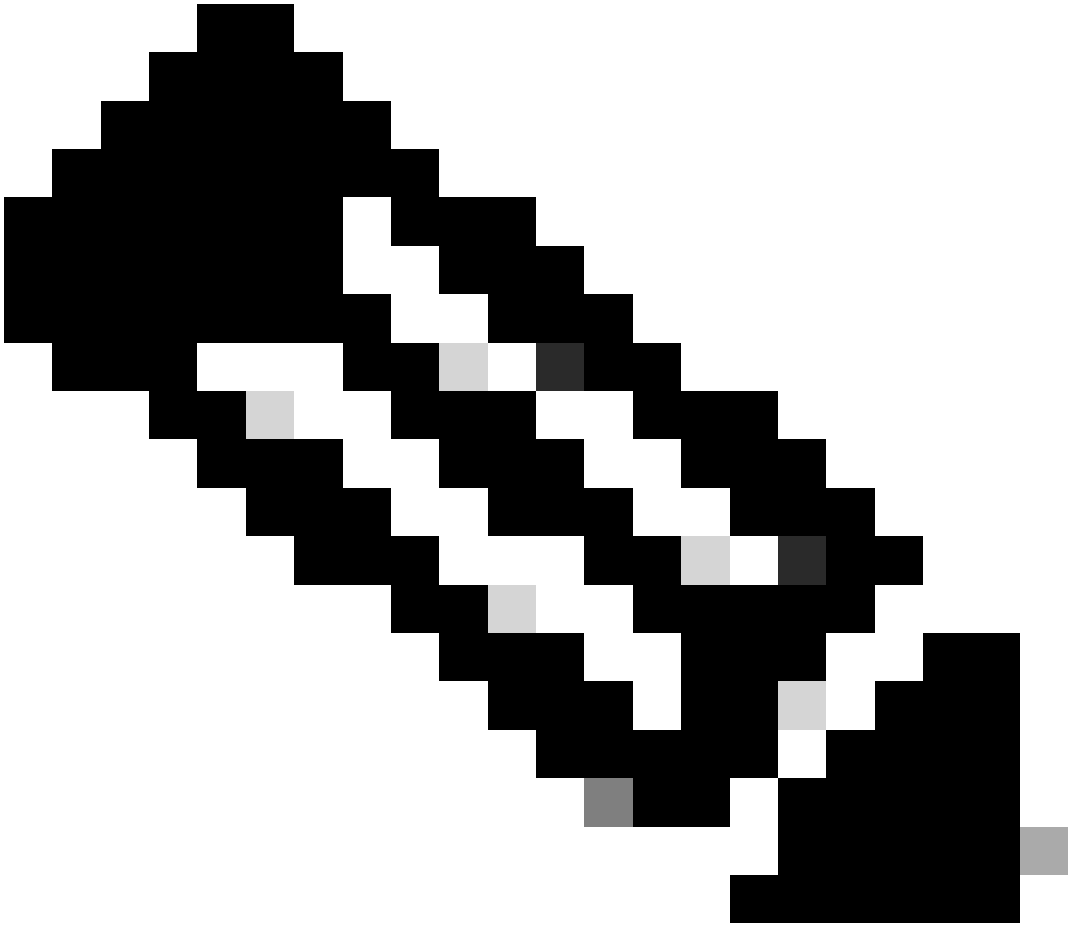
مثال على حزمة ServerHello في Wireshark

التكوين

يتضمن تنسيق سلسلة تشفير OpenSSL العديد من الأحرف الخاصة لإجراء عمليات على السلسلة مثل إزالة تشفير محدد أو مجموعة من الشفرة التي تشترك في مكون مشترك. بما أن الهدف من هذه التخصيصات هو عادة إزالة التشفير، فإن الحروف المستخدمة في هذه الأمثلة هي:

- الحرف -، المستخدم لإزالة التشفير من القائمة. يمكن السماح بعض أو كل التشفير الذي تمت إزالته مرة أخرى بواسطة الخيارات التي تظهر لاحقا في السلسلة.
- الحرف !، المستخدم أيضا لإزالة التشفير من القائمة. عند إستخدامه، لا يمكن السماح بالشفرات التي تمت إزالتها مرة أخرى بواسطة أي خيارات أخرى تظهر لاحقا في السلسلة.
- الحرف : الذي يعمل كفاصل بين العناصر في القائمة.

يمكن إستخدام كل منهما لإزالة تشفير من السلسلة، ومع ذلك، ! مفضل. للحصول على قائمة كاملة بالحروف الخاصة، راجع صفحة إدارة شفرات OpenSSL.



ملاحظة: يشير موقع OpenSSL إلى أنه عند إستخدام الحرف !، "لا يمكن إعادة ظهور المشفرات المحذوفة في القائمة حتى إذا تم ذكرها بشكل صريح". لا يعني هذا أن الشفرة محذوف نهائيا من النظام، وهو يشير إلى نطاق تفسير من الشفرة خيط.

تعطيل تشفير محدد

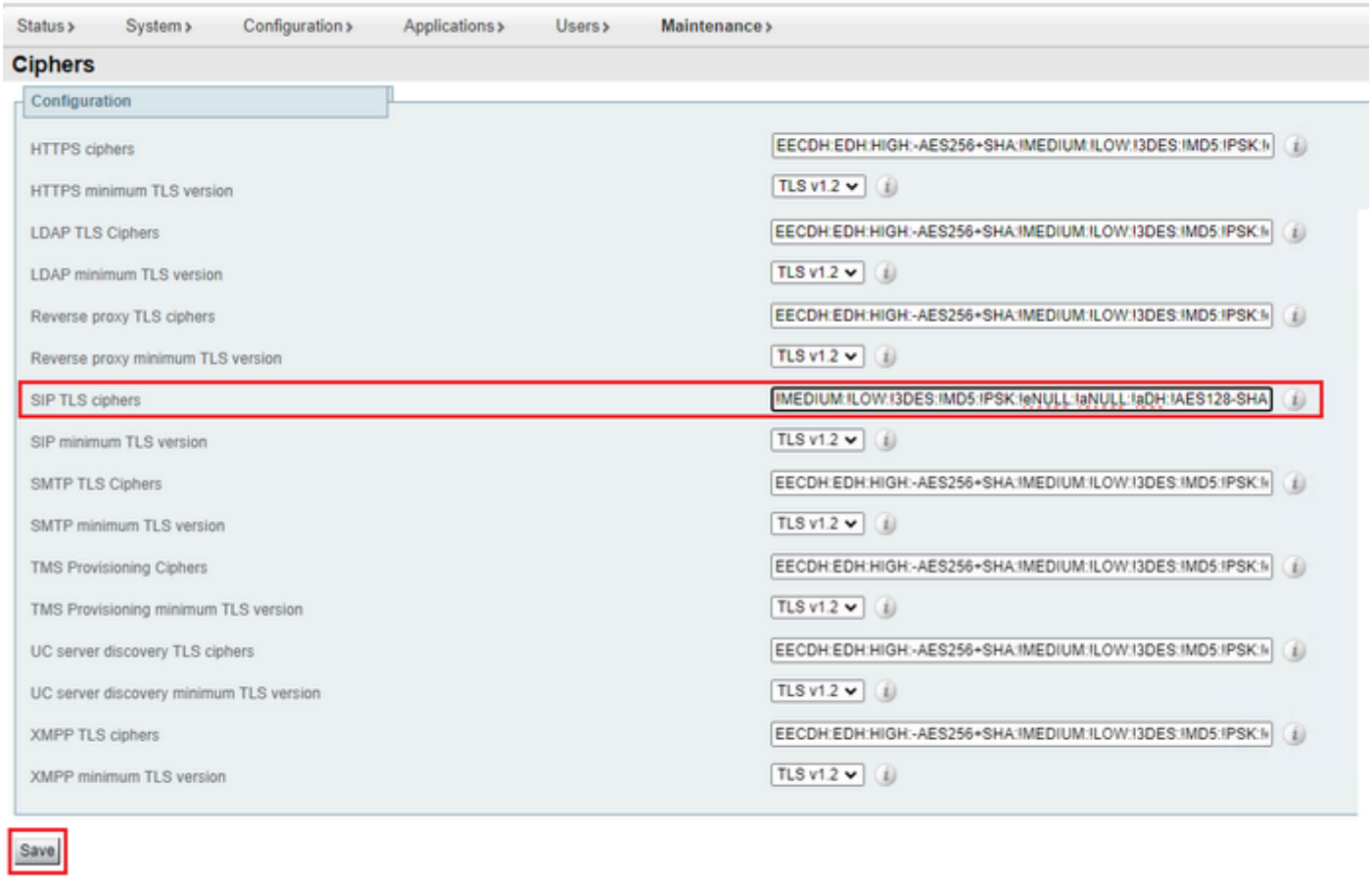
لتعطيل تشفير محدد، قم بإلحاق السلسلة الافتراضية: الفاصل، وعلامة ! وأ -، واسم التشفير التي سيتم تعطيلها. يجب أن يطيع اسم التشفير تنسيق تسمية OpenSSL، الموفر في صفحة إدارة تشفير OpenSSL. على سبيل المثال، إذا كنت بحاجة إلى تعطيل تشفير AES128-SHA لاتصالات SIP، قم بتكوين سلسلة تشفير مثل هذه:

<#root>

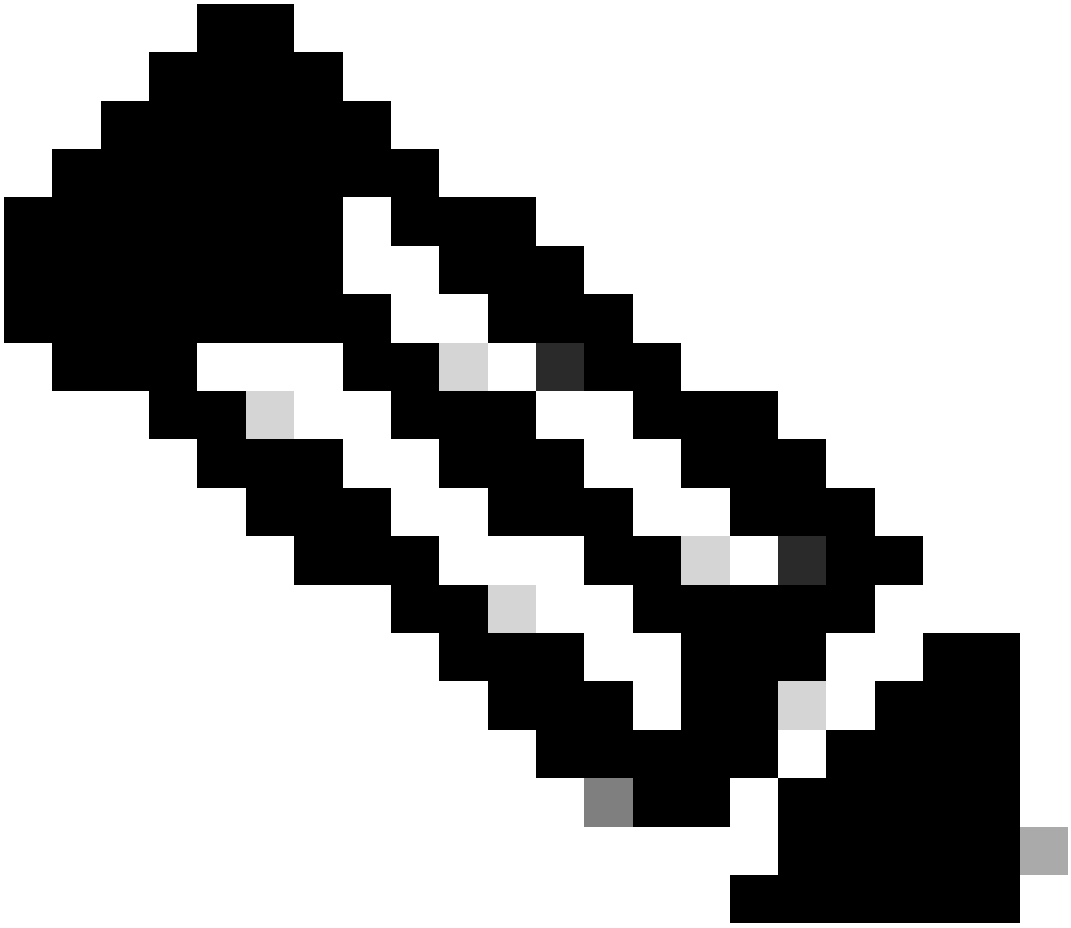
EECDH:EDH:HIGH:-AES256+SHA:!MEDIUM:!LOW:!3DES:!MD5:!PSK:!eNULL:!aNULL:!aDH

:!AES128-SHA

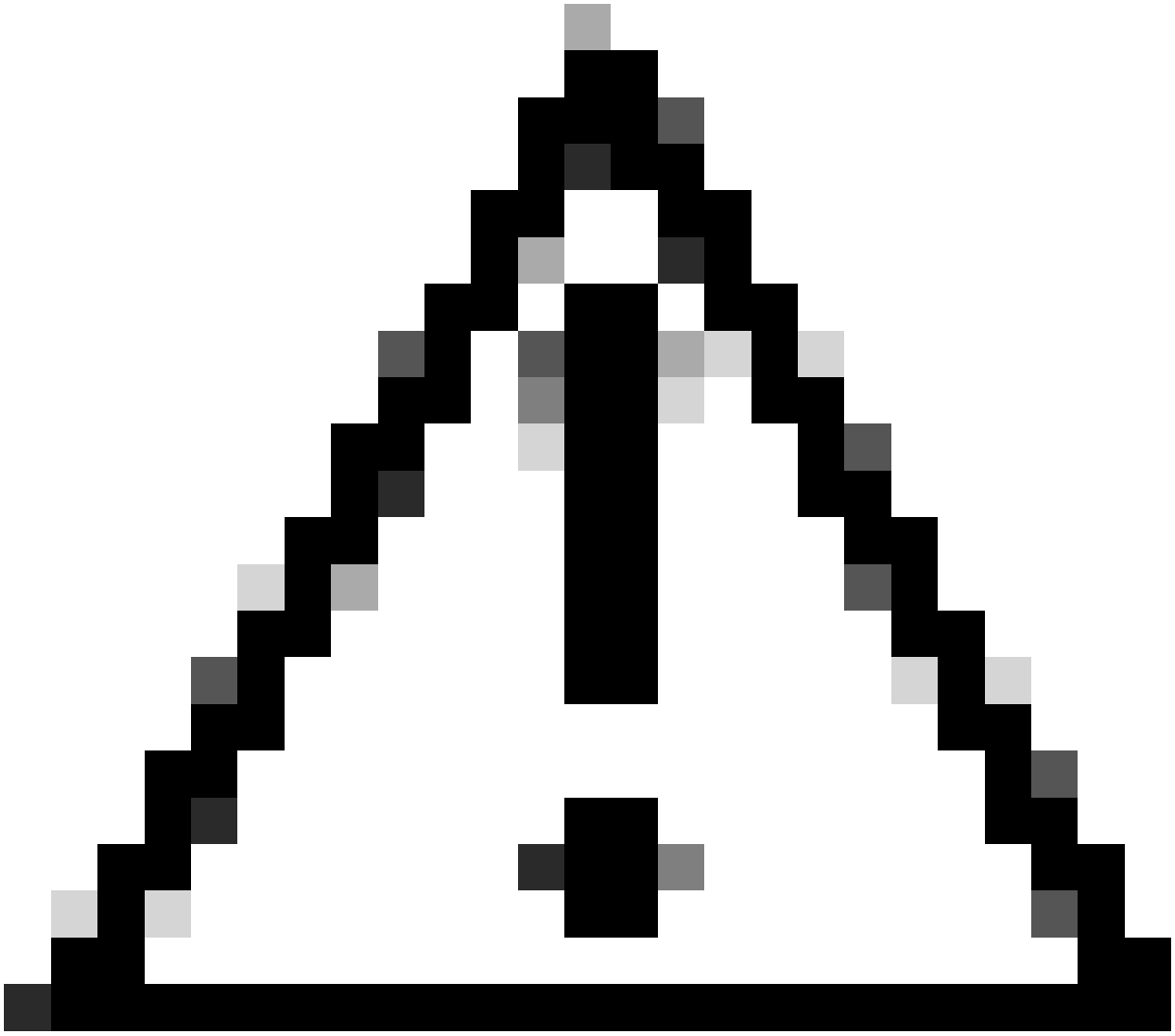
بعد ذلك، انتقل إلى صفحة إدارة الويب في Expressway، وتصفح إلى الصيانة > الأمان > التشفير، وعينت السلسلة المخصصة للبروتوكول (البروتوكولات) المطلوب، وانقر فوق حفظ. لكي يتم تطبيق التكوين الجديد، يلزم إعادة تشغيل النظام. في هذا المثال، يتم تعيين السلسلة المخصصة على بروتوكول SIP بموجب شفرات SIP TLS:



صفحة إعدادات التشفير على مدخل Expressway Web Admin



ملاحظة: في حالة مجموعة Expressway، قم بإجراء التغييرات على الخادم الأساسي فقط. يتم نسخ التكوين الجديد نسخا متماثلا إلى باقي أعضاء نظام المجموعة.



تحذير: أستخدم تسلسل إعادة تشغيل المجموعة الموصى به الوارد في دليل نشر وإنشاء مجموعة Expressway من Cisco وصياناتها. ابدأ بإعادة تشغيل الخادم الأساسي، وانتظر الوصول إليه عبر واجهة الويب، ثم قم بالمثل مع كل نظير وفقا للقائمة التي تم تكوينها ضمن النظام > التجميع.

تعطيل مجموعة من التشفير باستخدام خوارزمية عامة

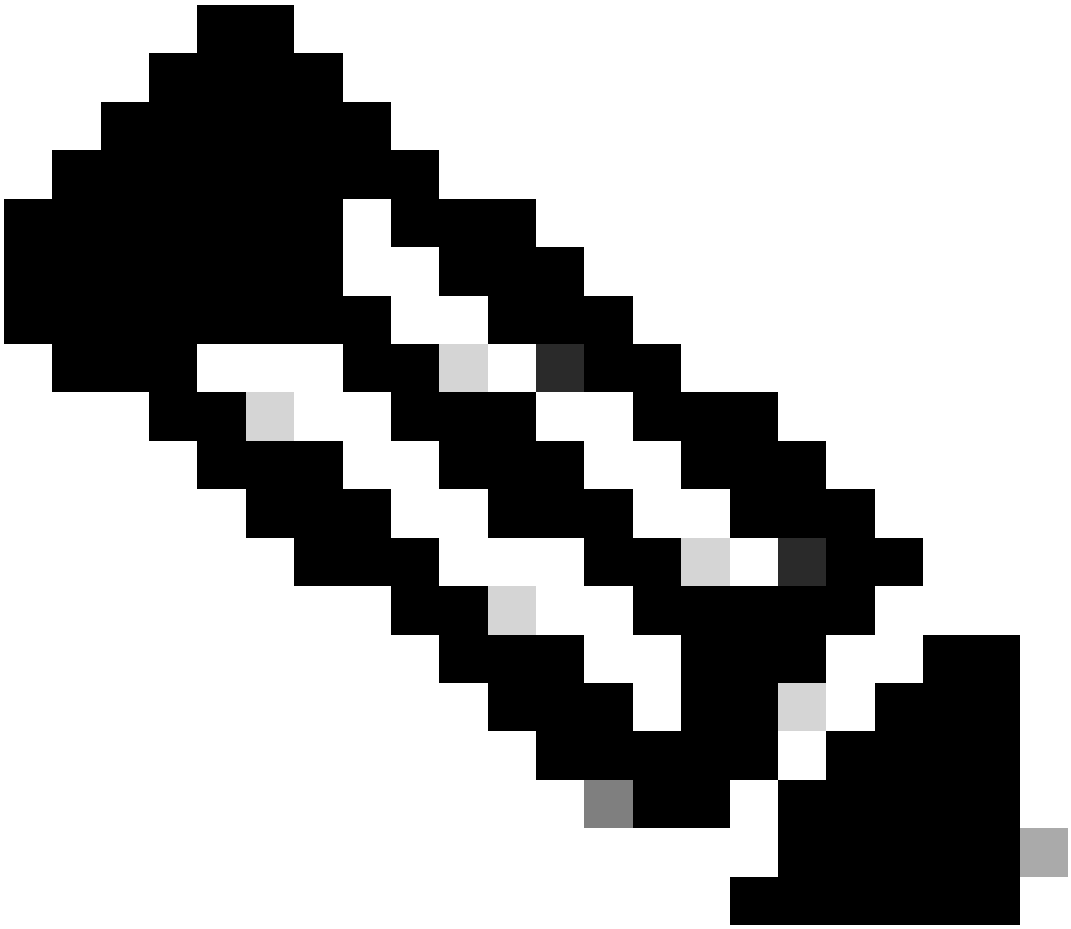
لتعطيل مجموعة من المشفرات باستخدام خوارزمية مشتركة، قم بإلحاق السلسلة الافتراضية :الفاصل، ! وأ -، واسم الخوارزمية التي سيتم تعطيلها. تتوفر أسماء الخوارزمية المدعومة في صفحة إدارة تشفير OpenSSL. على سبيل المثال، إذا كنت بحاجة إلى تعطيل جميع التشفير الذي يستخدم خوارزمية DHE، قم بتكوين سلسلة تشفير مثل هذه:

<#root>

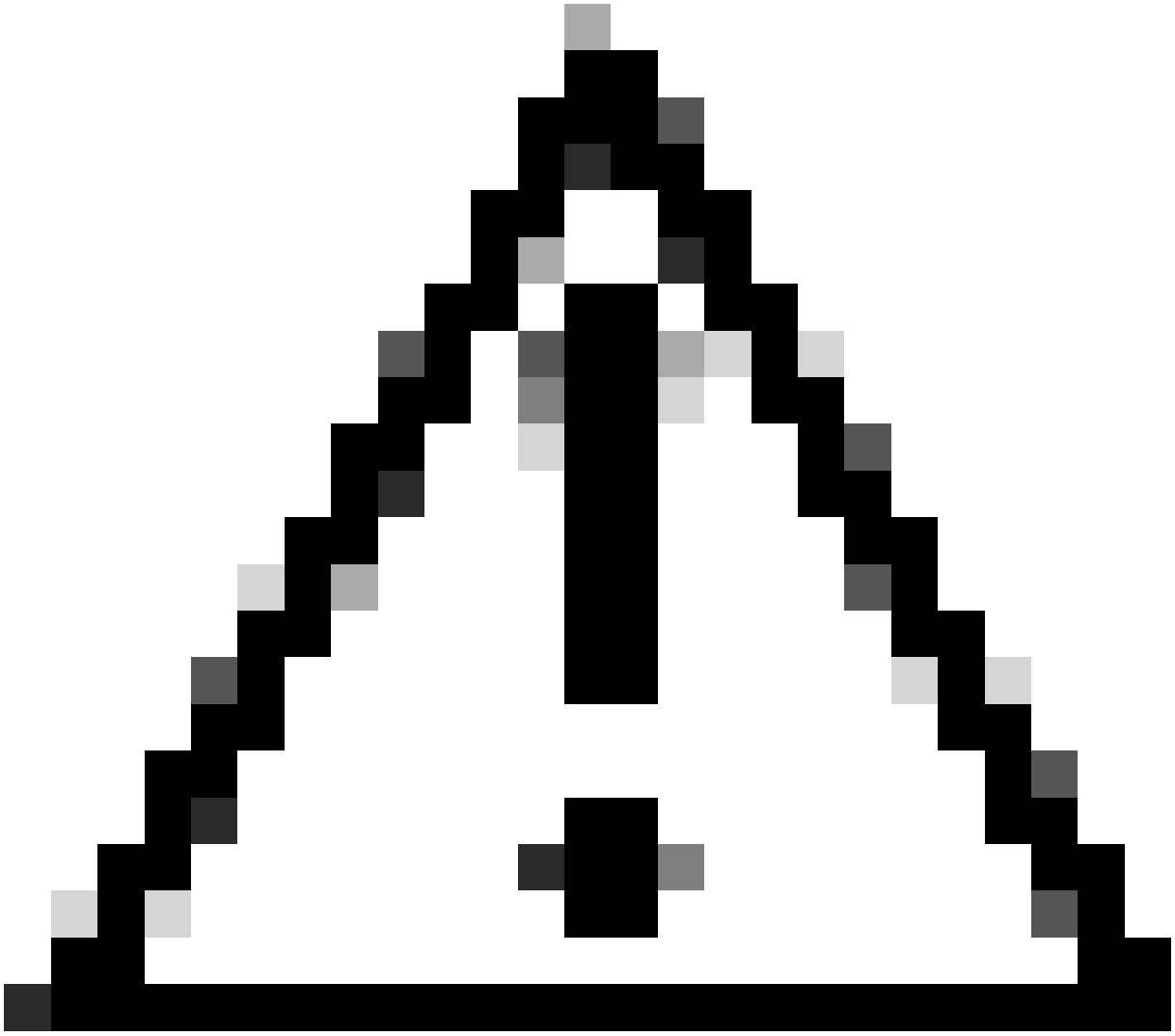
EECDH:EDH:HIGH:-AES256+SHA:!MEDIUM:!LOW:!3DES:!MD5:!PSK:!eNULL:!aNULL:!aDH

:!DHE

انتقل إلى صفحة Expressway Web Admin، وتصفح إلى الصيانة > الأمان > الشهادات، وعينت السلسلة المخصصة للبروتوكول (البروتوكولات) المطلوب، وانقر فوق حفظ. لكي يتم تطبيق التكوين الجديد، يلزم إعادة تشغيل النظام.



ملاحظة: في حالة مجموعة Expressway، قم بإجراء التغييرات على الخادم الأساسي فقط. يتم نسخ التكوين الجديد نسخا متماثلا إلى باقي أعضاء نظام المجموعة.



تحذير: أستخدم تسلسل إعادة تشغيل المجموعة الموصى به الوارد في دليل نشر وتهيئة مجموعة Expressway من Cisco. ابدأ بإعادة تشغيل الخادم الأساسي، وانتظر الوصول إليه عبر واجهة الويب، ثم قم بالمثل مع كل نظير وفقا للقائمة التي تم تكوينها ضمن النظام > التجميع.

التحقق من الصحة

فحص قائمة التشفير المسموح بها من قبل سلسلة التشفير

يمكنك فحص سلسلة التشفير المخصصة باستخدام الأمر openssl ciphers -V "<cipher string>". راجع المخرجات للتأكد من أن التشفير غير المرغوب فيه يتم إسقاطه بعد التغييرات. في هذا المثال، يتم فحص سلسلة تشفير EECDH:EDH:HIGH:-AES256+SHA:!MEDIUM:!LOW:!3DES:!MD5:!PSK:!PSKNULL:!aNULL:!aDH:!DHE. تؤكد مخرجات الأمر أن السلسلة لا تسمح بأي من التشفير الذي يستخدم خوارزمية DHE:

<#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
~ #
```

إختبار اتصال TLS بالتفاوض على تشفير معطل

يمكنك إستخدام الأمر openssl s_client للتحقق من رفض محاولة اتصال باستخدام تشفير معطل. أستخدم خيار الاتصال لتحديد عنوان Expressway والمنفذ الخاص بك، واستخدم خيار التشفير لتحديد الشفرة الفردية التي سيتم التفاوض بشأنها بواسطة العميل أثناء مصافحة TLS:

openssl s_client -connect <address>:<port> -cipher <cipher> -no_tls1_3

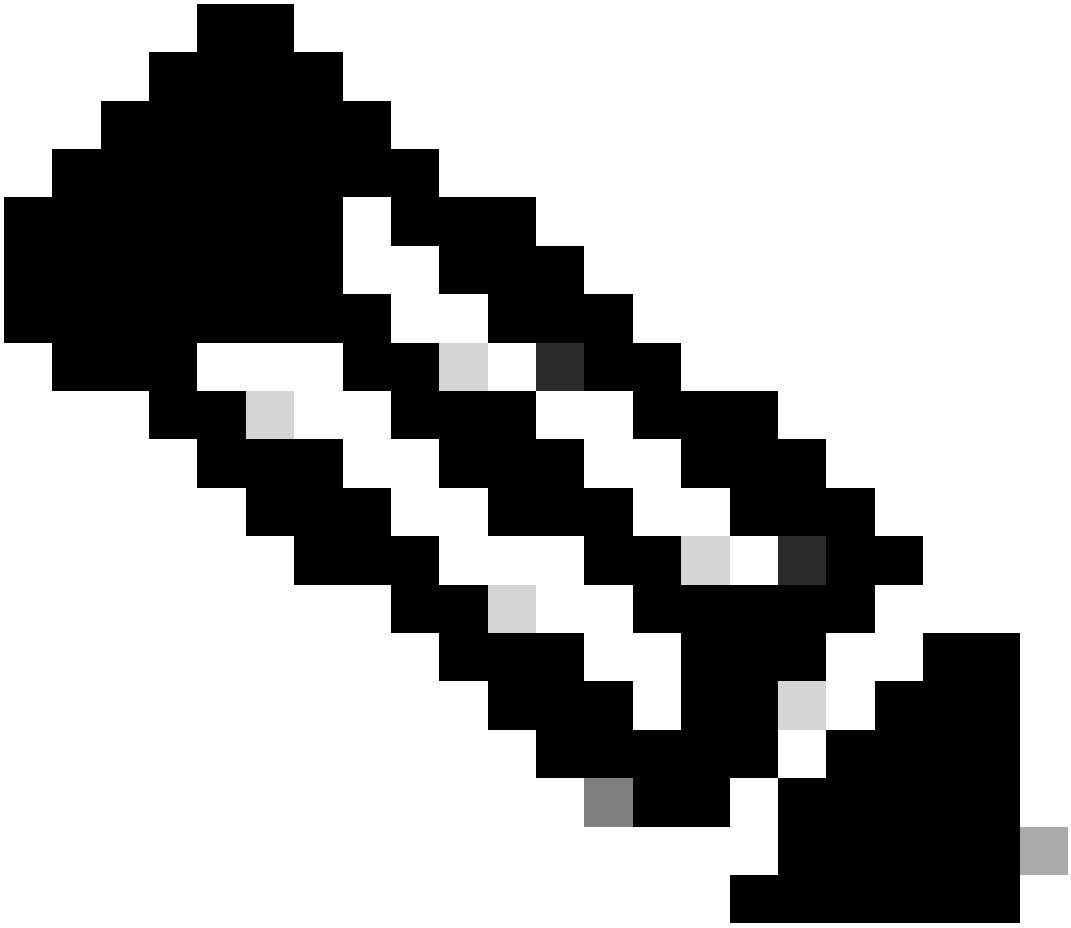
في هذا المثال، تتم محاولة اتصال TLS ب Expressway من كمبيوتر Windows مثبت عليه OpenSSL. الكمبيوتر الشخصي، باعتباره العميل، إلا بالتفاوض على تشفير DHE-RSA-AES256-CCM غير المرغوب فيه، والذي يستخدم خوارزمية DHE:

<#root>

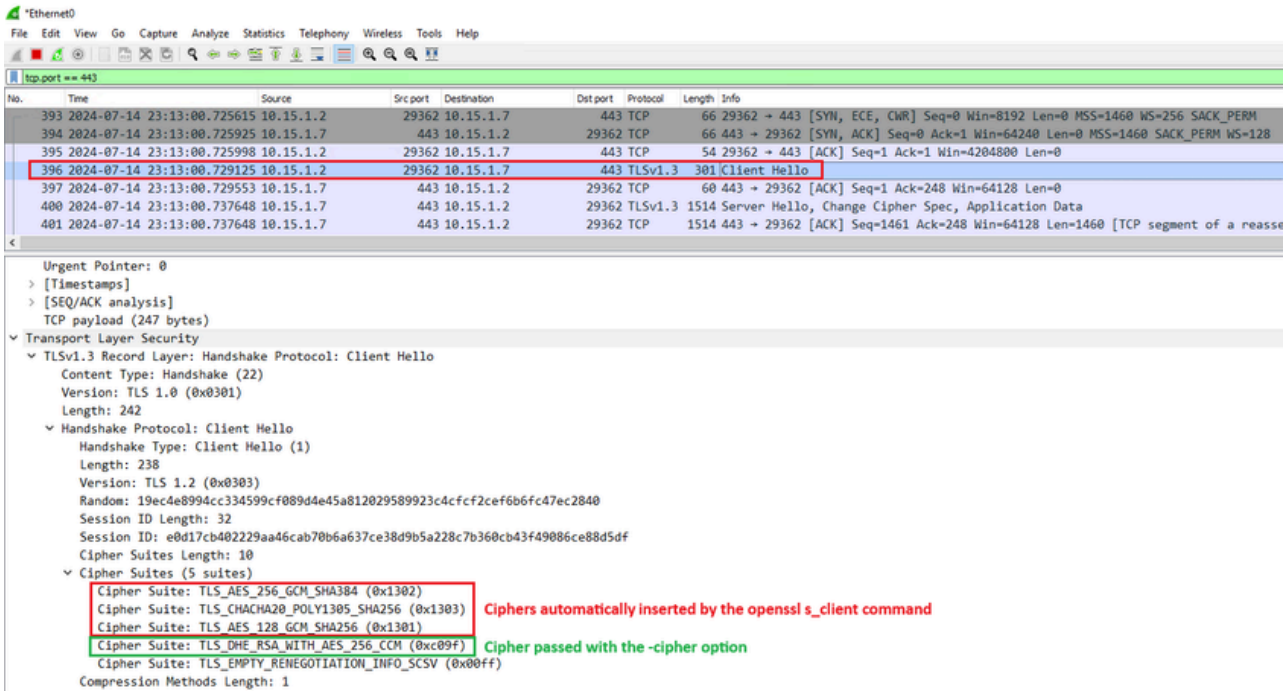
```
C:\Users\Administrator>

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>
```

يعرض إخراج الأمر محاولة الاتصال مع فشل تأكيد اتصال "SSL/TLS":..\ssl\record\rec_layer_s3.c:865:SSL رقم التنبيه 40"، نظرا لتكوين Expressway لاستخدام EECDH:EDH:HIGH:-AES256+SHA:!MEDIUM:!LOW:!3DES:!MD5:!PSK!eNULL:!aDH:!DHE تشفير سلسلة إتصالات HTTPS، الذي يعطل المشفرات التي تستخدم خوارزمية DHE.



ملاحظة: يلزم تمرير الخيار -no_tls1_3 إلى الأمر لكي تعمل الاختبارات التي تستخدم الأمر openSSL s_client كما هو موضح. إذا لم يكن متضمنا، يقوم العميل بإدراج شفرة TLS 1.3 تلقائيا في حزمة ClientHello:



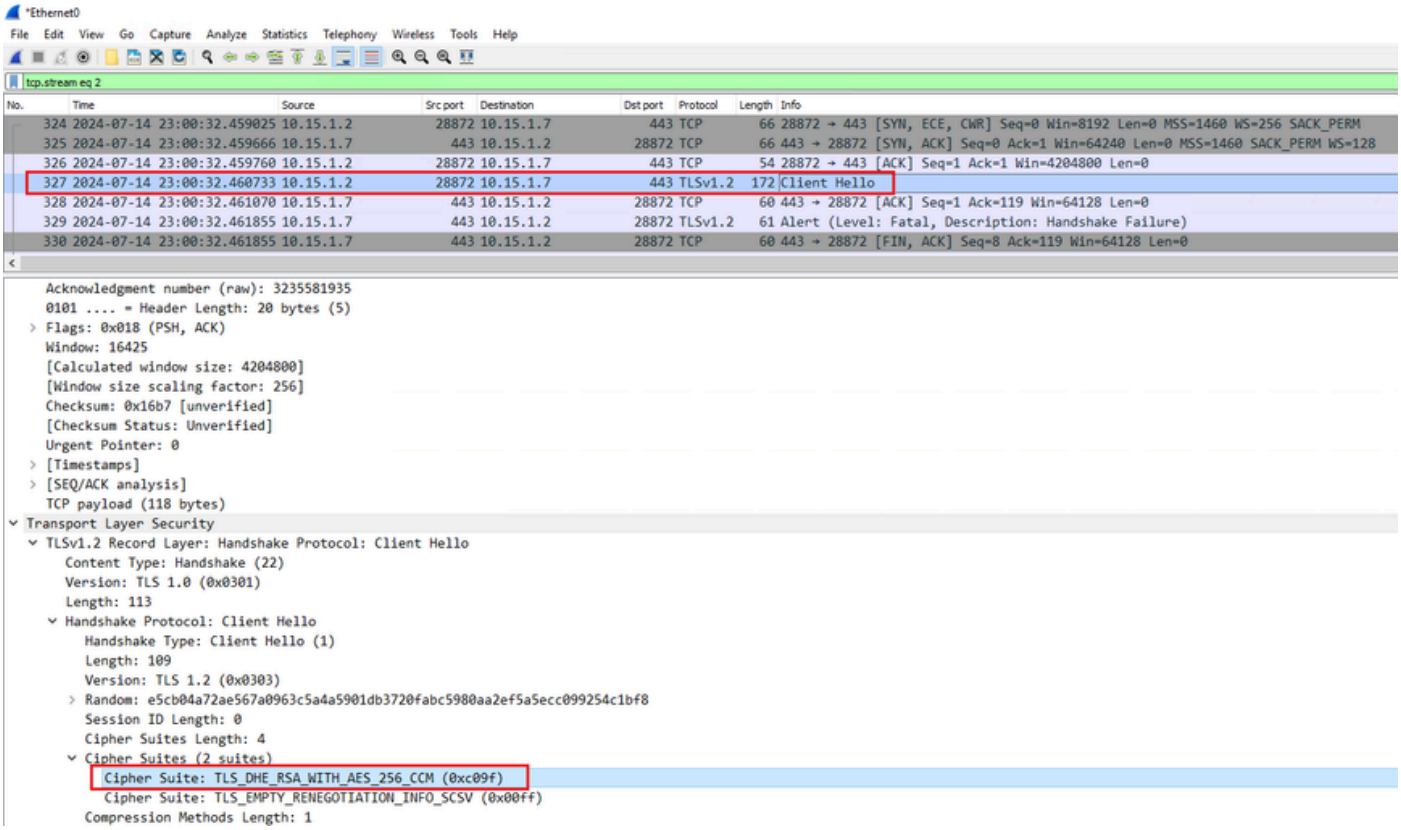
حزمة ClientHello مع شفرات تمت إضافتها تلقائيا

إذا كان الطريق السريع الهدف يدعم تلك الشفرات، يمكن إختيار واحدة منها بدلا من الشفرة المحددة التي تحتاج إلى إختبارها. الاتصال ناجح، والذي يمكن أن يقودك إلى الاعتقاد بأن الاتصال كان ممكنا باستخدام التشفير المعطل الذي تم تمريره إلى الأمر باستخدام الخيار -cipher.

فحص التقاط حزمة من مصافحة TLS باستخدام تشفير معطل

يمكنك تجميع التقاط حزمة، من جهاز الاختبار أو من Expressway، أثناء إجراء إختبار اتصال باستخدام أحد التشفير المعطل. يمكنك عندئذ فحصه باستخدام Wireshark لمزيد من تحليل أحداث المصافحة.

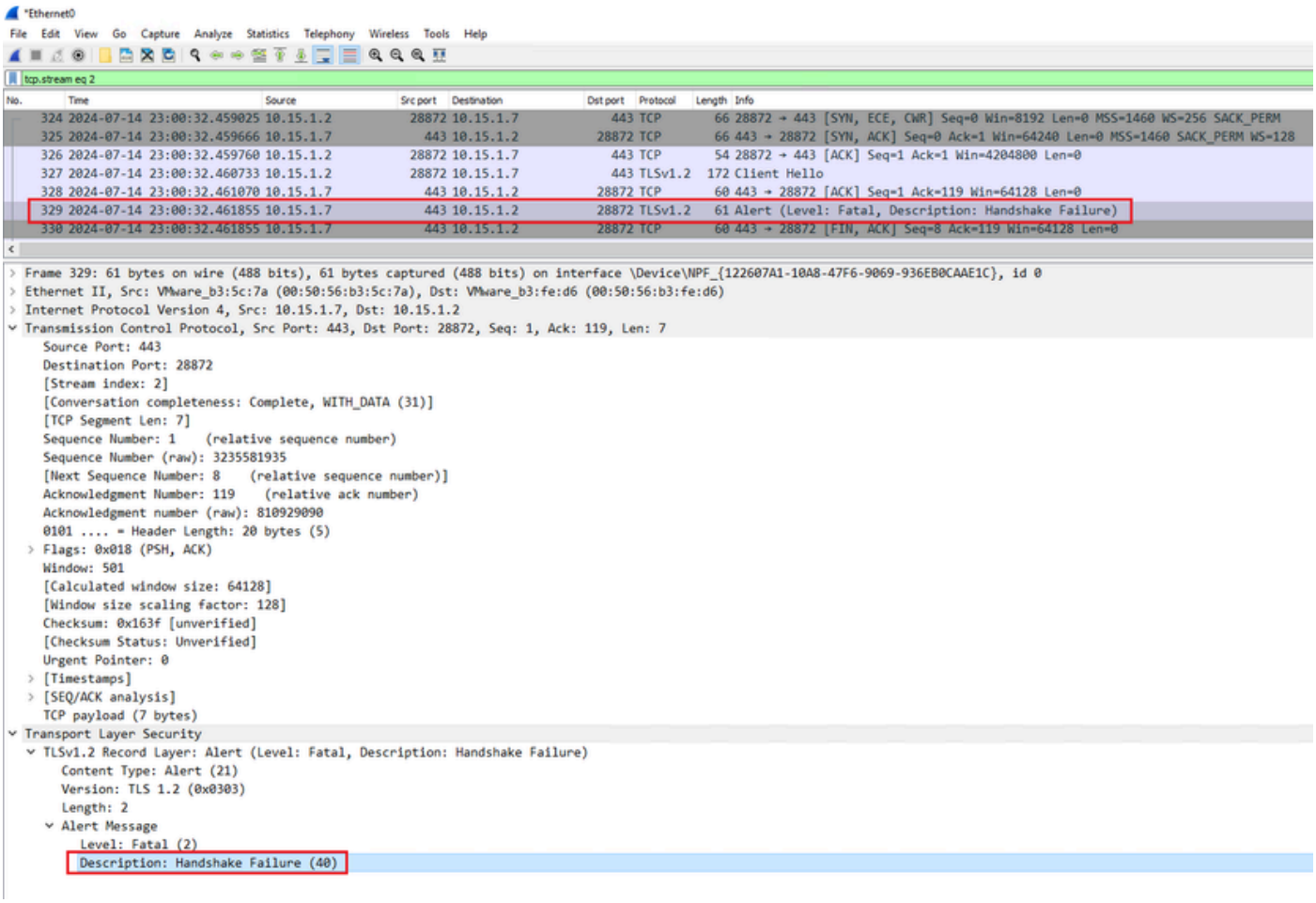
ابحث عن ClientHello الذي تم إرساله بواسطة جهاز الاختبار. تأكد من أنها تتفاوض فقط على تشفير الاختبار غير المرغوب، في هذا المثال، تشفير باستخدام خوارزمية DHE:



مثال على حزمة ClientHello في Wireshark

:

تأكيد إستجابة Expressway باستخدام حزمة تنبيه TLS خطيرة، مع رفض الاتصال. في هذا المثال، نظرا لأن Expressway ال تدعم شفرات DHE لكل سلسلة شفرة DHE تم تكوينها لبروتوكول HTTPS، فإنها تستجيب مع حزمة تنبيه TLS قاتلة تحتوي على رمز الفشل 40.



حزمة تنبيه TLS خطيرة في Wireshark

معلومات ذات صلة

- إدارة شفرات OpenSSL
- دليل مسؤول Cisco Expressway (X15.0) - الفصل: إدارة الأمان - تكوين الحد الأدنى لإصدار TLS ومجموعات التشفير

حول هذه الترجمة
ترجمت Cisco هذا المستند باستخدام مجموعة من التقنيات الآلية والبشرية لتقديم محتوى دعم للمستخدمين في جميع أنحاء العالم بلغتهم الخاصة. يُرجى ملاحظة أن أفضل ترجمة آلية لن تكون دقيقة كما هو الحال مع الترجمة الاحترافية التي يقدمها مترجم محترف. تخلي Cisco Systems مسئوليتها عن دقة هذه الترجمات وتُوصي بالرجوع دائمًا إلى المستند الإنجليزي الأصلي (الرابط متوفر).