Personalizza la configurazione della crittografia SSL di Expressway

Sommario

Introduzione

In questo documento viene descritto come personalizzare le stringhe di crittografia preconfigurate in Expressway.

Prerequisiti

Requisiti

Cisco raccomanda la conoscenza dei seguenti argomenti:

- Cisco Expressway o Cisco VCS.
- Protocollo TLS.

Componenti usati

Le informazioni fornite in questo documento si basano sulle seguenti versioni software e hardware:

- Cisco Expressway versione X15.0.2

Le informazioni discusse in questo documento fanno riferimento a dispositivi usati in uno specifico ambiente di emulazione. Su tutti i dispositivi menzionati nel documento la configurazione è stata ripristinata ai valori predefiniti. Se la rete è operativa, valutare attentamente eventuali conseguenze derivanti dall'uso dei comandi.

Premesse

La configurazione predefinita di Expressway include stringhe di crittografia preconfigurate che, per motivi di compatibilità, consentono il supporto di alcune cifrature che possono essere considerate deboli in base ad alcuni criteri di sicurezza aziendali. È possibile personalizzare le stringhe di cifratura in modo da adattarle alle regole specifiche di ogni ambiente.

In Expressway è possibile configurare una stringa di crittografia indipendente per ognuno dei seguenti protocolli:

- HTTPS
- LDAP
- Proxy reverse
- SIP
- SMTP
- Provisioning TMS
- individuazione server UC
- XMPP

Le stringhe di cifratura rispettano il formato OpenSSL descritto nella pagina principale dei cifrari OpenSSL. La versione corrente di Expressway X15.0.2 viene fornita con la stringa predefinita EECDH:EDH:HIGH:-AES256+SHA:!MEDIUM:!LOW:!3DES:!MD5:!PSK:!eNULL:!aNULL:!aDH preconfigurata per tutti i protocolli allo stesso modo. Dalla pagina Web admin, in Manutenzione > Sicurezza > Cifre, è possibile modificare la stringa di cifratura assegnata a ciascun protocollo, per aggiungere o rimuovere cifrature specifiche o gruppi di cifrature utilizzando un algoritmo comune.

Controllare la stringa di crittografia

Utilizzando il comando openssl ciphers -V "<cipher string>", è possibile generare un elenco con tutte le cifrature consentite da una determinata stringa, utile per esaminare visivamente le cifrature. In questo esempio viene mostrato l'output del controllo della stringa di crittografia predefinita di Expressway:

```
<#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
```

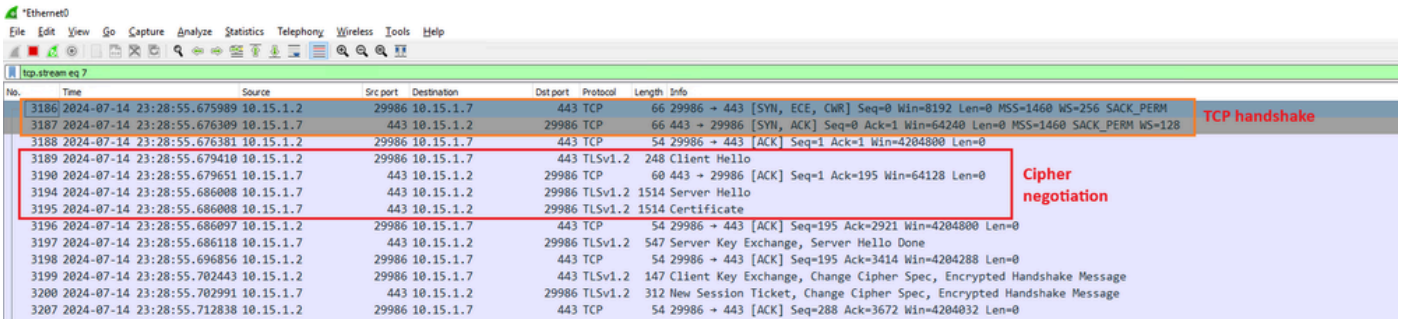
```
0xC0,0x30 - ECDHE-RSA-AES256-GCM-SHA384 TLSv1.2 Kx=ECDH Au=RSA Enc=AESGCM(256) Mac=AEAD
0xCC,0xA9 - ECDHE-ECDSA-CHACHA20-POLY1305 TLSv1.2 Kx=ECDH Au=ECDSA Enc=CHACHA20/POLY1305(256) Mac=AEAD
0xCC,0xA8 - ECDHE-RSA-CHACHA20-POLY1305 TLSv1.2 Kx=ECDH Au=RSA Enc=CHACHA20/POLY1305(256) Mac=AEAD
0xC0,0xAD - 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
~ #
```

Ispezionare la negoziazione della crittografia nell'handshake TLS con un'acquisizione pacchetto

Acquisendo una negoziazione TLS in un'acquisizione pacchetto, è possibile esaminare i dettagli della negoziazione cifratura utilizzando Wireshark.

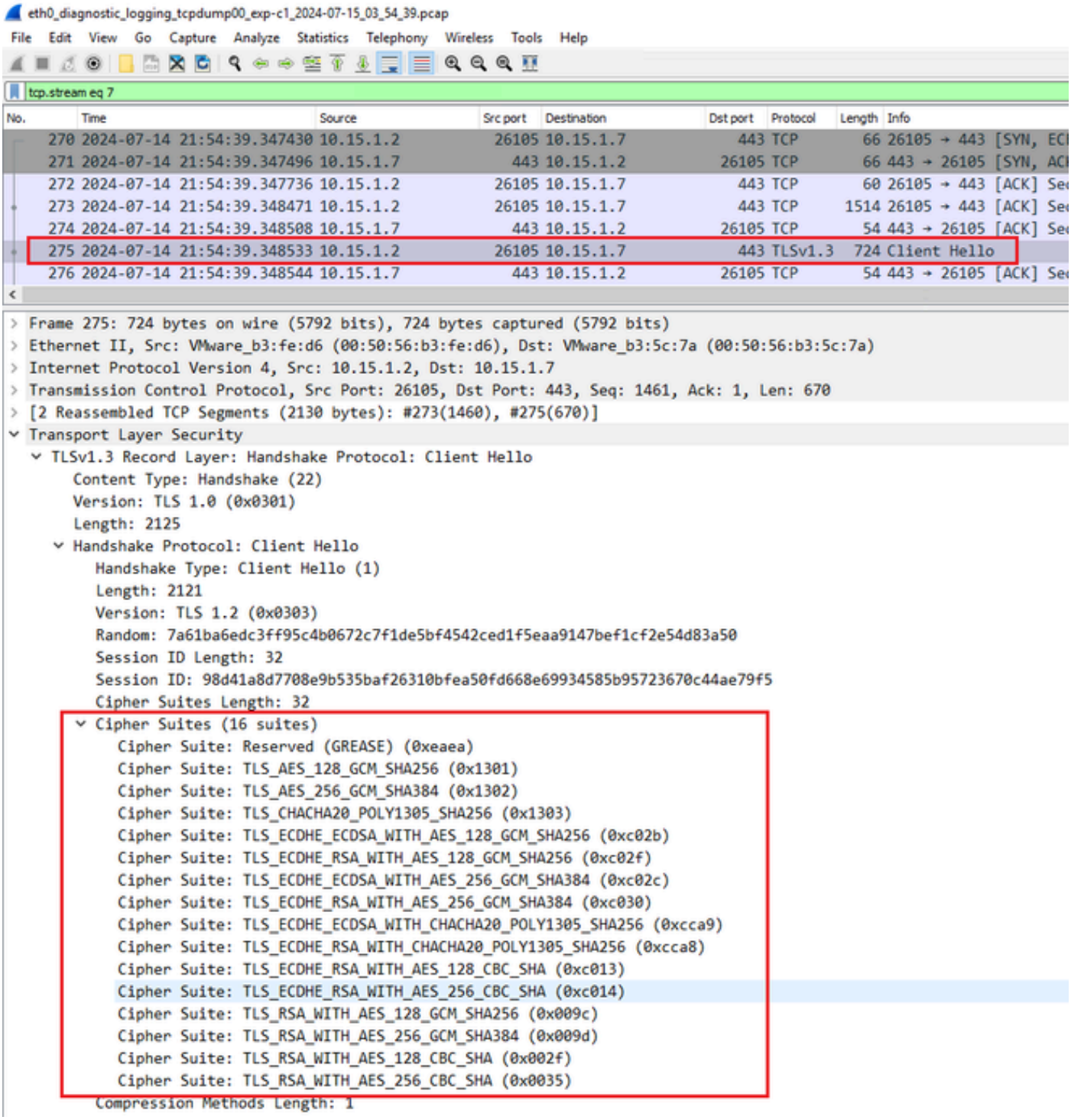
Il processo di handshake TLS include un pacchetto ClientHello inviato dal dispositivo client, che fornisce l'elenco delle cifrature supportate in base alla relativa stringa di cifratura configurata per il protocollo di connessione. Il server esamina l'elenco, lo confronta con il proprio elenco di cifrari consentiti (determinato dalla propria stringa di cifratura) e sceglie una cifratura supportata da entrambi i sistemi, da utilizzare per la sessione crittografata. Quindi risponde con un pacchetto ServerHello che indica la cifratura scelta. Esistono importanti differenze tra le finestre di dialogo di handshake TLS 1.2 e 1.3, tuttavia il meccanismo di negoziazione cifratura utilizza lo stesso principio in entrambe le versioni.

Questo è un esempio di negoziazione della cifratura TLS 1.3 tra un browser Web ed Expressway sulla porta 443, come mostrato in Wireshark:



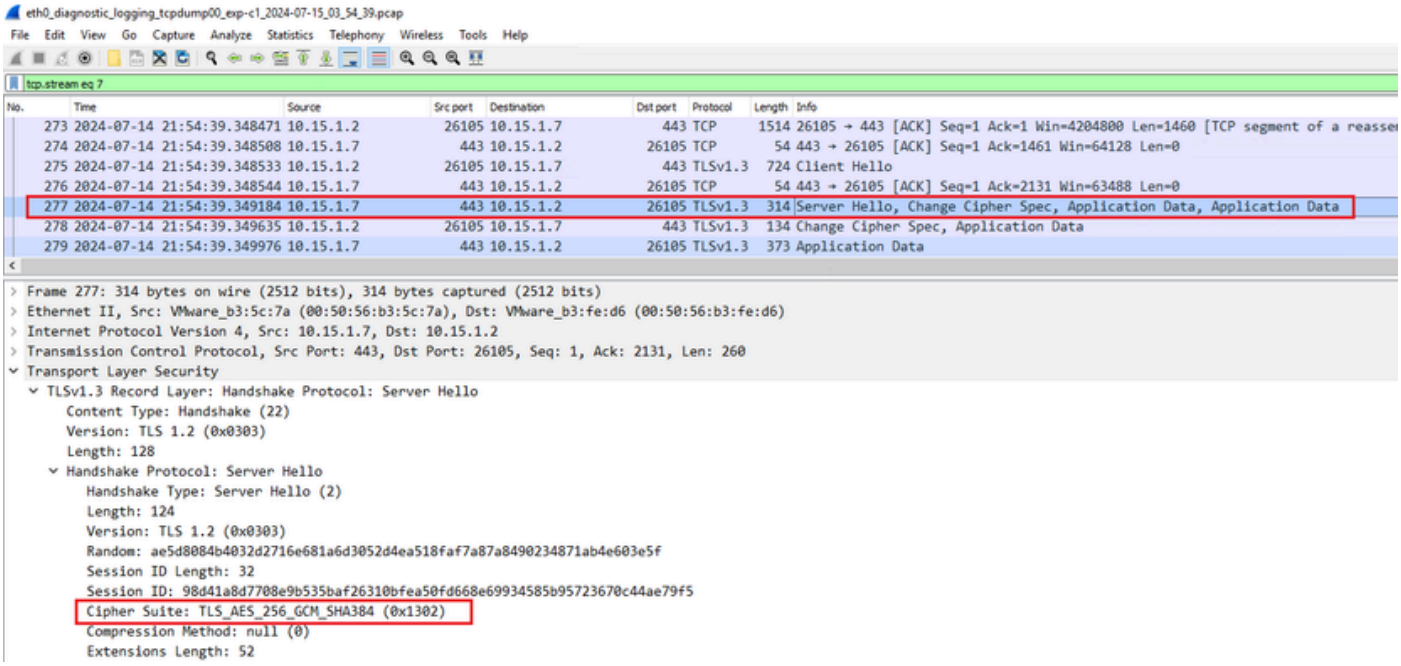
Esempio di handshake TLS in Wireshark

Innanzitutto, il browser invia un pacchetto ClientHello con l'elenco delle cifrature supportate:



Esempio di pacchetto ClientHello in Wireshark

Expressway verifica la stringa di crittografia configurata per il protocollo HTTPS e trova una crittografia supportata sia da Expressway che dal client. Nell'esempio è selezionata la cifratura ECDHE-RSA-AES256-GCM-SHA384. Expressway risponde con il pacchetto ServerHello che indica la cifratura selezionata:



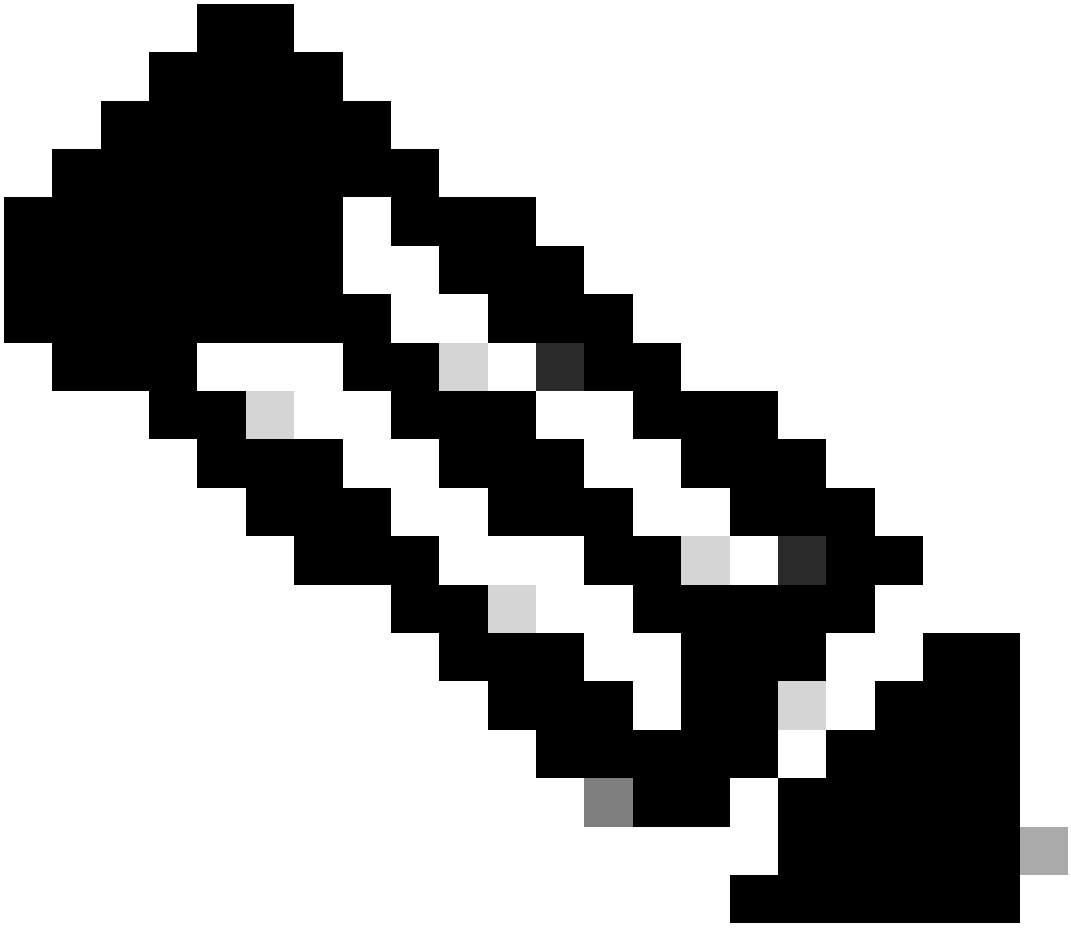
Esempio di un pacchetto ServerHello in Wireshark

Configurazione

Il formato di stringa di cifratura OpenSSL include diversi caratteri speciali per eseguire operazioni sulla stringa, ad esempio rimuovere una cifratura specifica o un gruppo di cifrature che condividono un componente comune. Poiché l'obiettivo di queste personalizzazioni è in genere la rimozione dei cifrari, i caratteri utilizzati in questi esempi sono i seguenti:

- Il carattere -, utilizzato per rimuovere i cifrari dall'elenco. Alcune o tutte le cifrature rimosse possono essere nuovamente consentite tramite le opzioni visualizzate più avanti nella stringa.
- Il carattere !, utilizzato anche per rimuovere i cifrari dall'elenco. Quando viene utilizzato, i cifrari rimossi non possono essere nuovamente consentiti da altre opzioni visualizzate più avanti nella stringa.
- Il carattere :, che funge da separatore tra le voci dell'elenco.

Entrambe possono essere utilizzate per rimuovere una cifratura dalla stringa. ! è preferibile. Per un elenco completo dei caratteri speciali, consultate la pagina di gestione dei cifrari OpenSSL.



Nota: il sito OpenSSL afferma che quando si utilizza il carattere !, "i cifrari eliminati non possono mai riapparire nell'elenco anche se sono esplicitamente indicati". Ciò non significa che i cifrari siano cancellati definitivamente dal sistema, ma si riferisce all'ambito dell'interpretazione della stringa di cifratura.

Disattivazione di una crittografia specifica

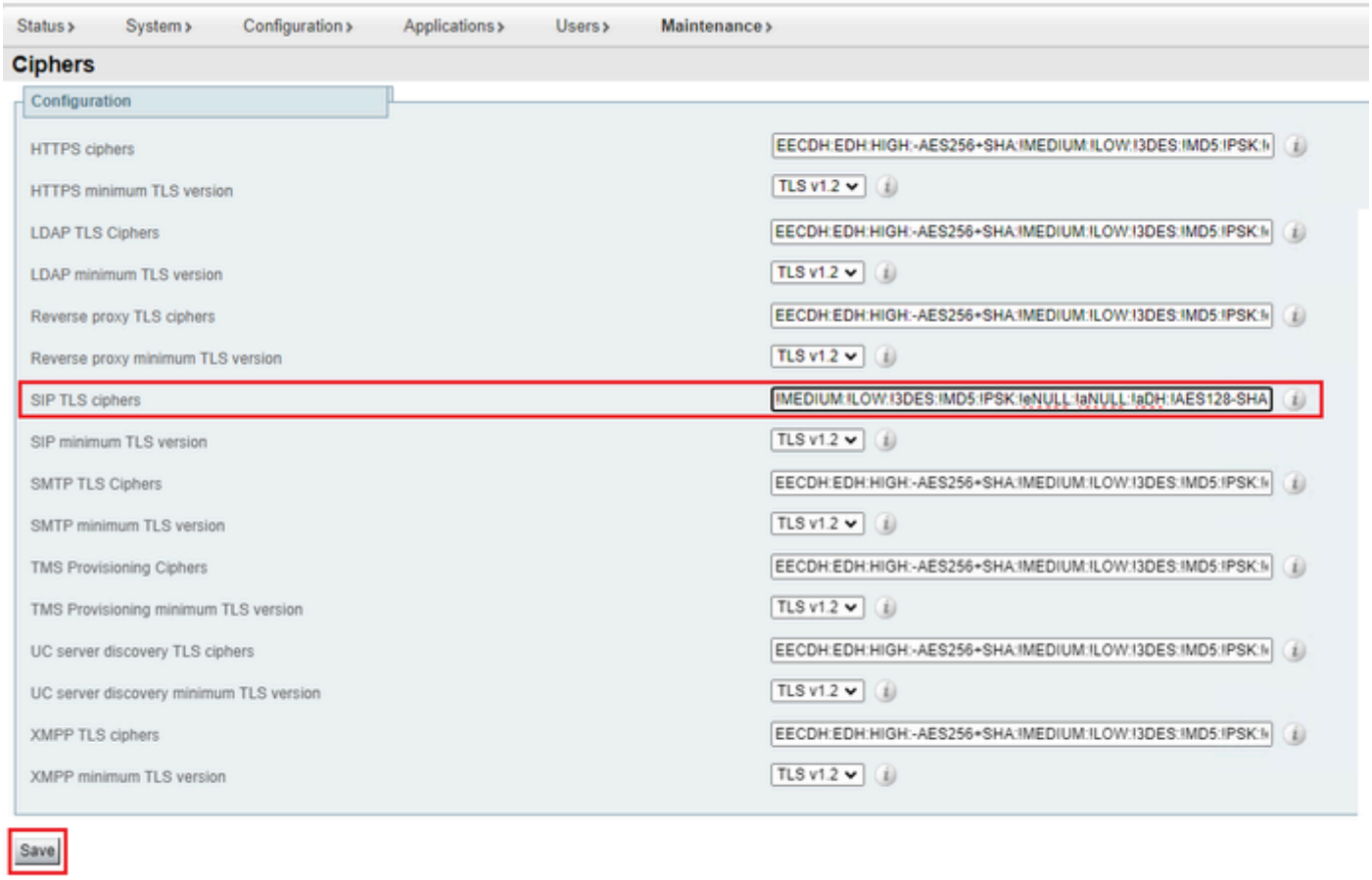
Per disabilitare una cifratura specifica, aggiungere alla stringa predefinita il separatore :, il segno ! o - e il nome della cifratura da disabilitare. Il nome della cifratura deve essere conforme al formato di denominazione OpenSSL, disponibile nella pagina di gestione dei cifrari OpenSSL. Ad esempio, se è necessario disabilitare la cifratura AES128-SHA per le connessioni SIP, configurare una stringa di cifratura come questa:

<#root>

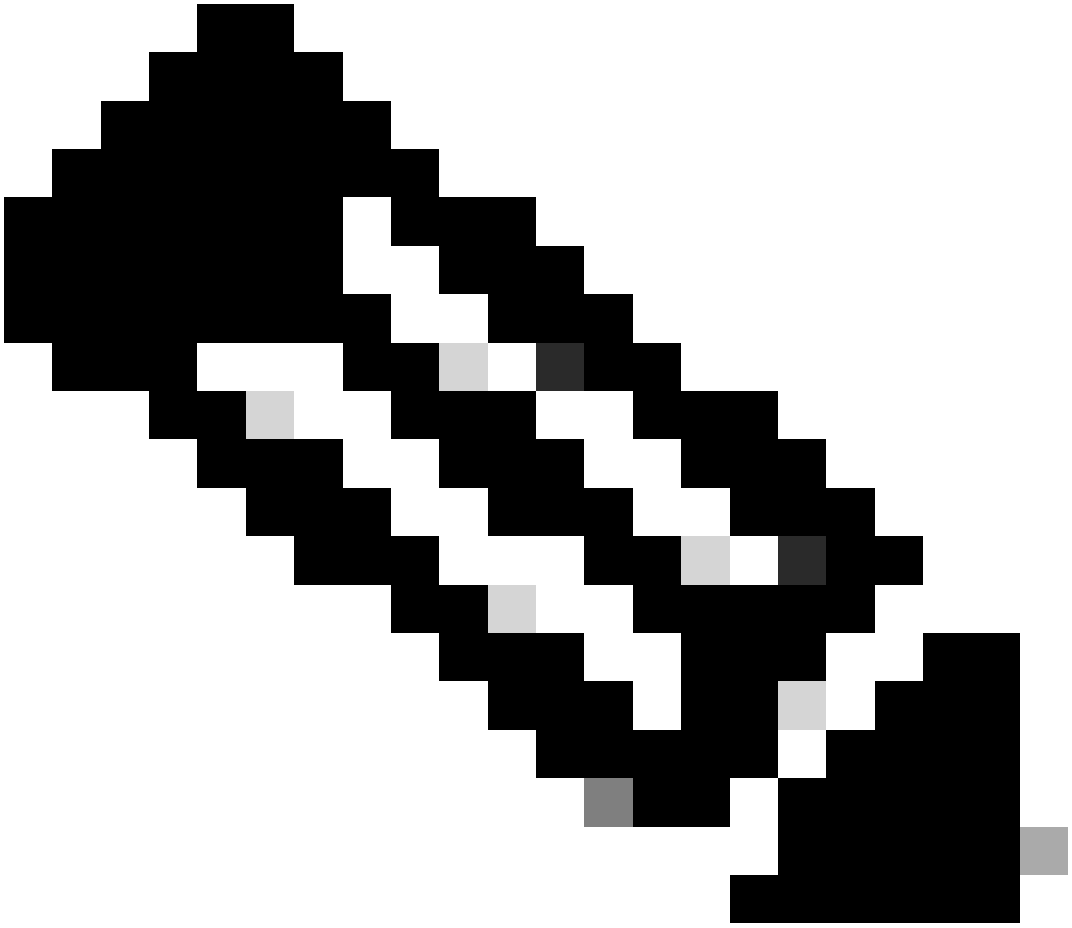
```
EECDH:EDH:HIGH:-AES256+SHA:!MEDIUM:!LOW:!3DES:!MD5:!PSK:!eNULL:!aNULL:!aDH
```

:!AES128-SHA

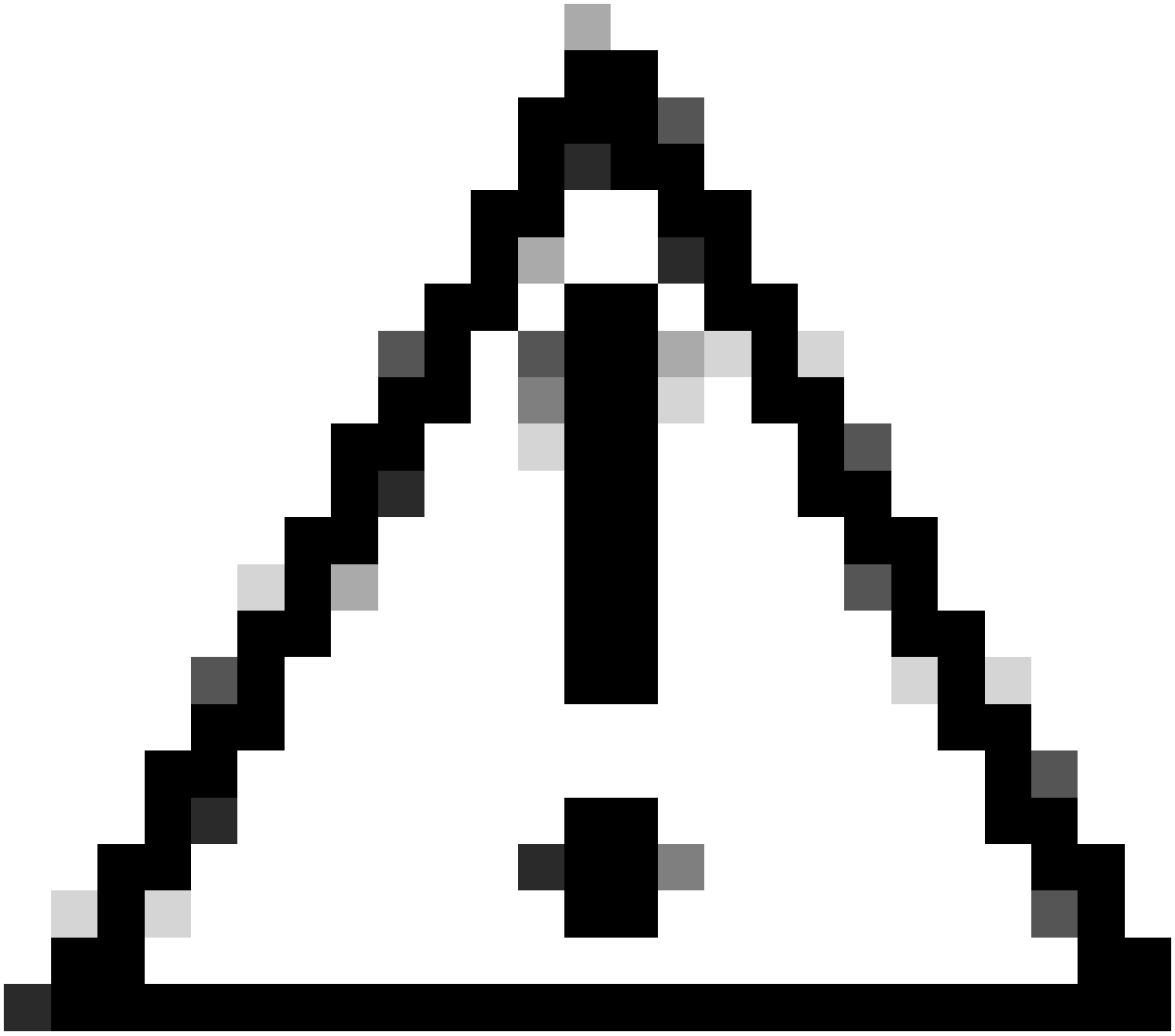
Passare quindi alla pagina Expressway web admin, passare a Manutenzione > Sicurezza > Cifre, assegnare la stringa personalizzata ai protocolli richiesti e fare clic su Salva. Per applicare la nuova configurazione, è necessario riavviare il sistema. Nell'esempio, la stringa personalizzata viene assegnata al protocollo SIP tramite i cifrari SIP TLS:



Pagina Impostazioni di crittografia sul portale Expressway Web Admin



Nota: nel caso di un cluster Expressway, apportare le modifiche solo sul server principale. La nuova configurazione viene replicata negli altri membri del cluster.



Attenzione: utilizzare la sequenza di riavvio del cluster consigliata specificata nella Guida alla creazione e alla manutenzione dei cluster Cisco Expressway. Avviare riavviando il server primario, attendere che sia accessibile tramite interfaccia Web, quindi fare lo stesso con ciascun peer in base all'elenco configurato in Sistema > Clustering.

Disattivazione di un gruppo di cifrari tramite un algoritmo comune

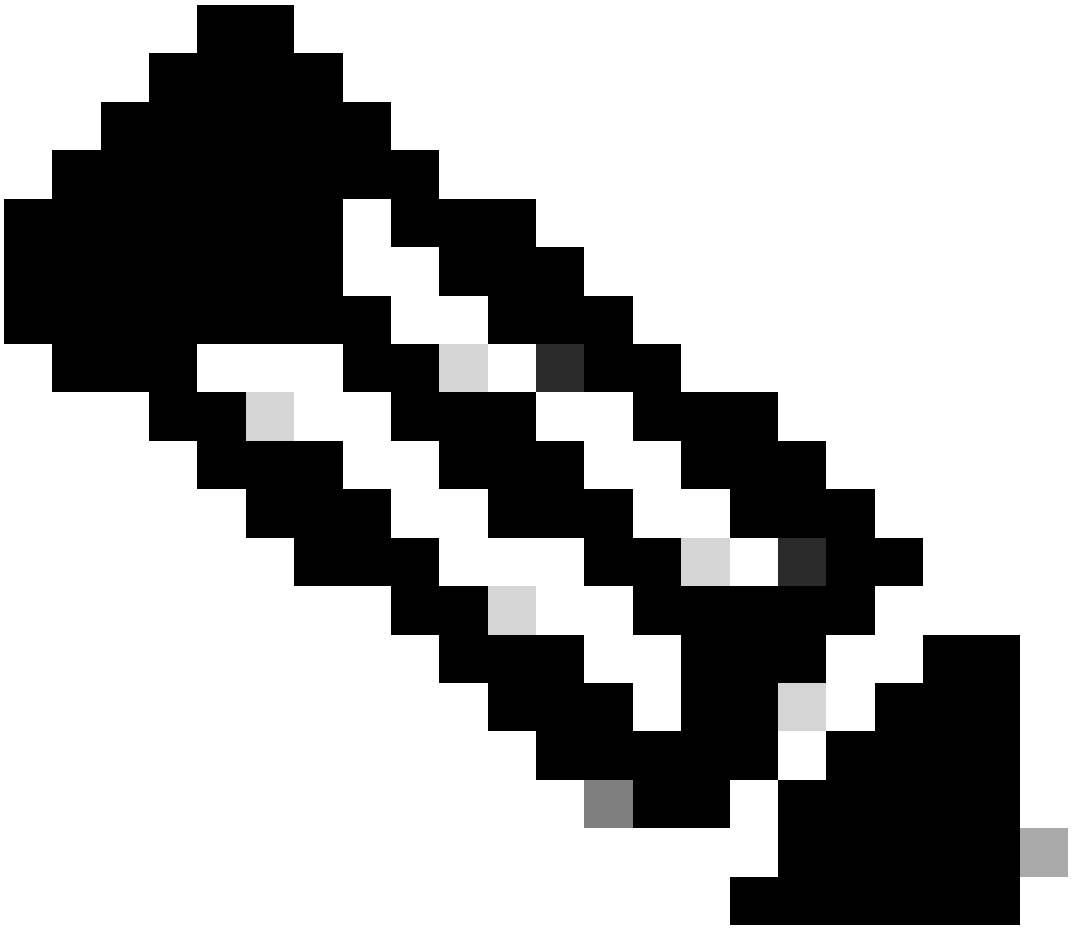
Per disabilitare un gruppo di cifrari che utilizzano un algoritmo comune, aggiungere alla stringa predefinita il separatore :, il segno ! o - e il nome dell'algoritmo da disabilitare. I nomi degli algoritmi supportati sono disponibili nella pagina di gestione dei cifrari OpenSSL. Ad esempio, se è necessario disabilitare tutti i cifrari che utilizzano l'algoritmo DHE, configurare una stringa di cifratura simile alla seguente:

<#root>

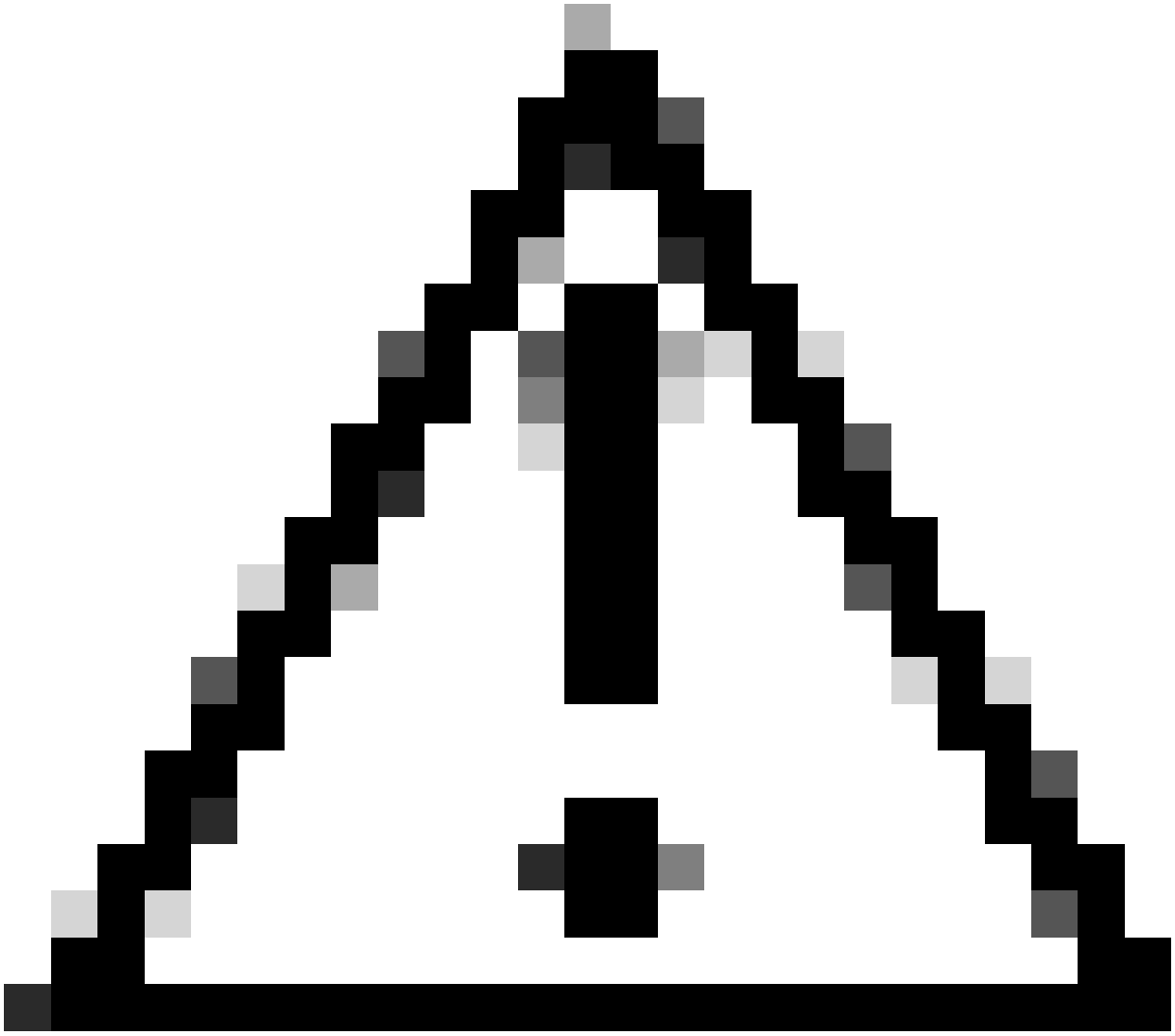
EECDH:EDH:HIGH:-AES256+SHA:!MEDIUM:!LOW:!3DES:!MD5:!PSK:!eNULL:!aNULL:!aDH

:!DHE

Passare alla pagina Expressway web admin, selezionare Manutenzione > Protezione > Cifre, assegnare la stringa personalizzata ai protocolli richiesti e fare clic su Salva. Per applicare la nuova configurazione, è necessario riavviare il sistema.



Nota: nel caso di un cluster Expressway, apportare le modifiche solo sul server principale. La nuova configurazione viene replicata negli altri membri del cluster.



Attenzione: utilizzare la sequenza di riavvio del cluster consigliata specificata nella Guida alla creazione e alla manutenzione dei cluster Cisco Expressway. Avviare riavviando il server primario, attendere che sia accessibile tramite interfaccia Web, quindi fare lo stesso con ciascun peer in base all'elenco configurato in Sistema > Clustering.

Verifica

Controllare l'elenco di cifrature consentite dalla stringa di cifratura

È possibile esaminare la stringa di cifratura personalizzata utilizzando il comando openssl ciphers -V "<cipher string>". Esaminate l'output per verificare che dopo le modifiche non siano più elencati i cifrari indesiderati. In questo esempio viene ispezionata la stringa di crittografia EECDH:EDH:HIGH:-AES256+SHA:!MEDIUM:!LOW:!3DES:!MD5:!PSK:!eNULL:!aNULL:!aDH:!DHE. L'output del comando conferma che la stringa non consente alcuna delle cifrature che utilizzano l'algoritmo 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
~ #
```

Eseguire il test di una connessione TLS negoziando una crittografia disabilitata

È possibile utilizzare il comando openssl s_client per verificare che un tentativo di connessione che utilizza una cifratura disabilitata venga rifiutato. Utilizzare l'opzione -connect per specificare l'indirizzo e la porta di Expressway e l'opzione -cipher per specificare la singola cifratura che deve essere negoziata dal client durante l'handshake TLS:

openssl s_client -connect <indirizzo>:<porta> -cipher <cipher> -no_tls1_3

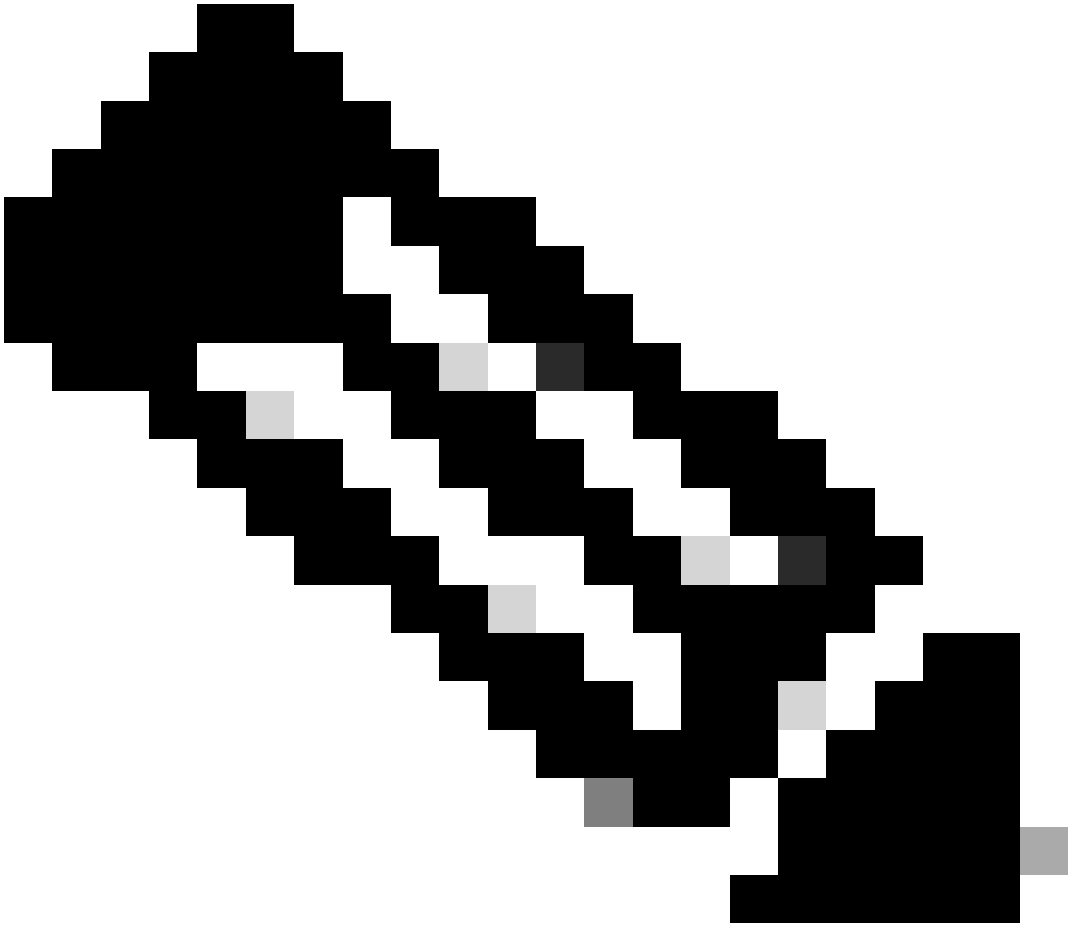
In questo esempio, viene tentata una connessione TLS verso Expressway da un PC Windows con openssl installato. Il PC, in qualità di client, negozia solo la cifratura indesiderata DHE-RSA-AES256-CCM, che utilizza l'algoritmo DHE:

<#root>

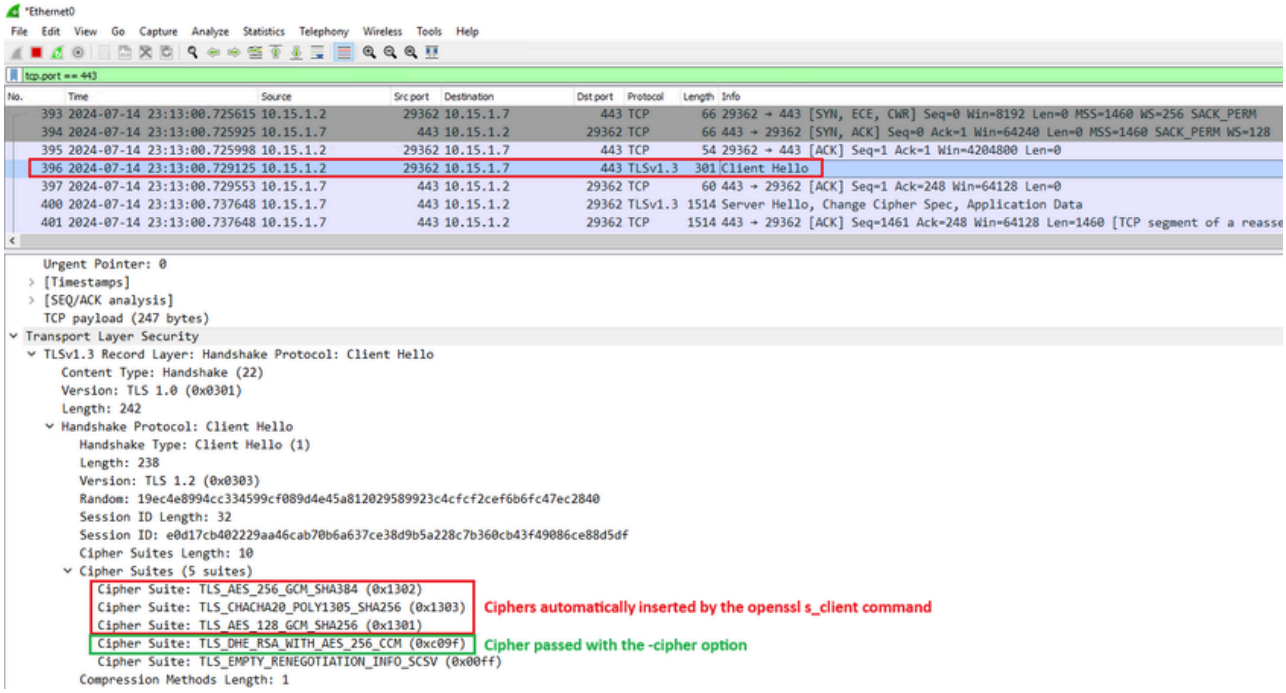
```
C:\Users\Administrator>

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>
```

L'output del comando mostra che il tentativo di connessione ha esito negativo con un messaggio di errore "ssl/tls alert handshake failure:..\ssl\record\rec_layer_s3.c:865:SSL alert number 40", perché Expressway è configurato per l'utilizzo di ECDH:EDH:HIGH:-AES256+SHA:!MEDIUM:!LOW:!3DES:!MD5:!PSK:!eNULL:!aNULL:!aDH:!DHE cipher string for HTTPS connections disabled i cifrari che utilizzano l'algoritmo DHE.



Nota: affinché i test con il comando openssl s_client funzionino come spiegato, è necessario passare al comando l'opzione -no_tls1_3. Se non è incluso, il client inserisce automaticamente le cifrature TLS 1.3 nel pacchetto ClientHello:



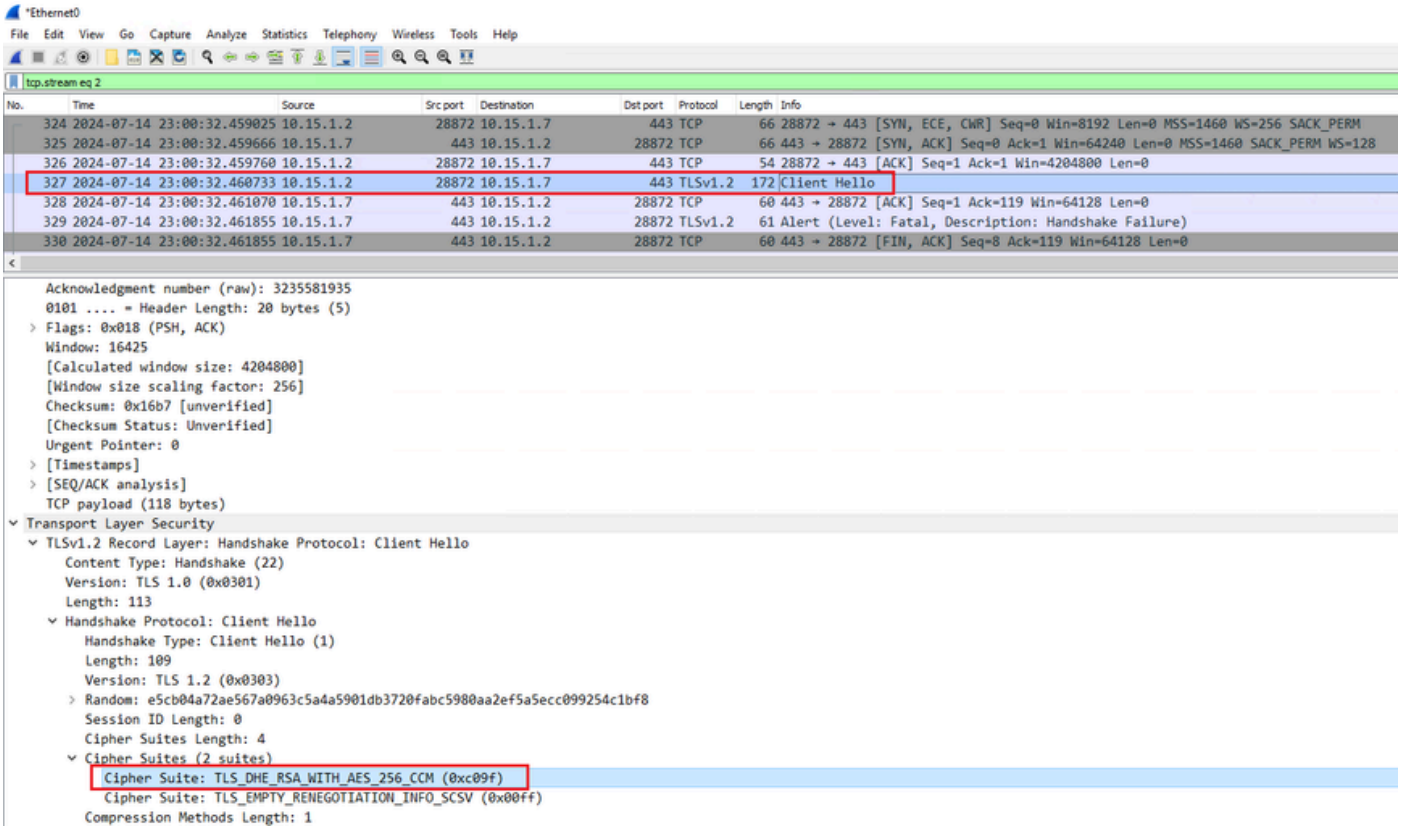
Pacchetto ClientHello con crittografia aggiunta automaticamente

Se Expressway di destinazione supporta tali cifrari, è possibile scegliere uno di essi anziché la cifratura specifica che è necessario verificare. La connessione ha esito positivo. È possibile pertanto ritenere che la connessione sia stata possibile utilizzando la cifratura disabilitata passata al comando con l'opzione -cipher.

Ispezionare l'acquisizione di un pacchetto di un handshake TLS con una crittografia disabilitata

È possibile raccogliere un'acquisizione di pacchetto dal dispositivo di test o da Expressway durante l'esecuzione di un test di connessione utilizzando una delle cifrature disabilitate. È quindi possibile ispezionarlo con Wireshark per analizzare ulteriormente gli eventi di handshake.

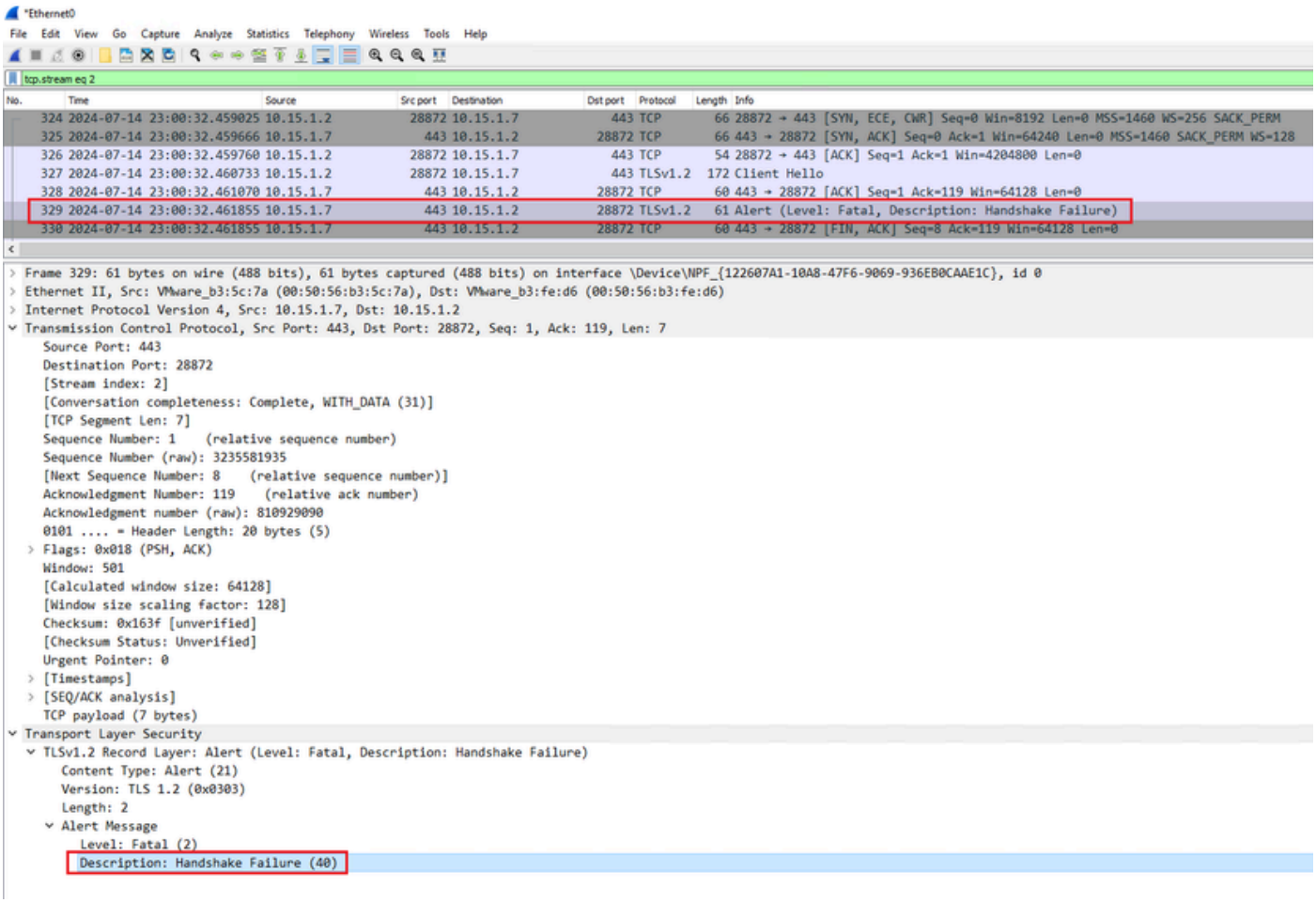
Individuare ClientHello inviato dal dispositivo di test. Confermare che venga negoziata solo la cifratura di prova indesiderata, in questo esempio una cifratura che utilizza l'algoritmo DHE:



Esempio di pacchetto ClientHello in Wireshark

:

Confermare che Expressway risponda con un pacchetto di avviso TLS irreversibile, rifiutando la connessione. In questo esempio, poiché Expressway non supporta i cifrari DHE in base alla relativa stringa di cifratura configurata per il protocollo HTTPS, risponde con un pacchetto di avviso TLS irreversibile contenente il codice di errore 40.



Un pacchetto di avviso TLS Fatal in Wireshark

Informazioni correlate

- [Manpage cifrature OpenSSL]
- [Cisco Expressway Administrator Guide (X15.0) - Capitolo: Managing Security - Configuring Minimum TLS Version and Cipher Suites]

Informazioni su questa traduzione

Cisco ha tradotto questo documento utilizzando una combinazione di tecnologie automatiche e umane per offrire ai nostri utenti in tutto il mondo contenuti di supporto nella propria lingua. Si noti che anche la migliore traduzione automatica non sarà mai accurata come quella fornita da un traduttore professionista. Cisco Systems, Inc. non si assume alcuna responsabilità per l’accuratezza di queste traduzioni e consiglia di consultare sempre il documento originale in inglese (disponibile al link fornito).