

Personalizza la configurazione della crittografia SSL di Expressway

Sommario

[Introduzione](#)

[Prerequisiti](#)

[Requisiti](#)

[Componenti usati](#)

[Premesse](#)

[Controllare la stringa di crittografia](#)

[Ispezionare la negoziazione della crittografia nell'handshake TLS con un'acquisizione pacchetto](#)

[Configurazione](#)

[Disattivazione di una crittografia specifica](#)

[Disattivazione di un gruppo di cifrari tramite un algoritmo comune](#)

[Verifica](#)

[Controllare l'elenco di cifrature consentite dalla stringa di cifratura](#)

[Eseguire il test di una connessione TLS negoziando una crittografia disabilitata](#)

[Ispezionare l'acquisizione di un pacchetto di un TLSHandshake utilizzando una crittografia disabilitata](#)

[Informazioni correlate](#)

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:

No.	Time	Source	Src port	Destination	Dst port	Protocol	Length	Info
3186	2024-07-14 23:28:55.675989	10.15.1.2	29986	10.15.1.7	443	TCP	66	29986 → 443 [SYN, ECE, CWR] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM
3187	2024-07-14 23:28:55.676309	10.15.1.7	443	10.15.1.2	29986	TCP	66	443 → 29986 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM WS=128
3188	2024-07-14 23:28:55.676381	10.15.1.2	29986	10.15.1.7	443	TCP	54	29986 → 443 [ACK] Seq=1 Ack=1 Win=4204800 Len=0
3189	2024-07-14 23:28:55.679410	10.15.1.2	29986	10.15.1.7	443	TLSv1.2	248	Client Hello
3190	2024-07-14 23:28:55.679651	10.15.1.7	443	10.15.1.2	29986	TCP	60	443 → 29986 [ACK] Seq=1 Ack=195 Win=64128 Len=0
3194	2024-07-14 23:28:55.686008	10.15.1.7	443	10.15.1.2	29986	TLSv1.2	1514	Server Hello
3195	2024-07-14 23:28:55.686008	10.15.1.7	443	10.15.1.2	29986	TLSv1.2	1514	Certificate
3196	2024-07-14 23:28:55.686097	10.15.1.2	29986	10.15.1.7	443	TCP	54	29986 → 443 [ACK] Seq=195 Ack=2921 Win=4204800 Len=0
3197	2024-07-14 23:28:55.686118	10.15.1.7	443	10.15.1.2	29986	TLSv1.2	547	Server Key Exchange, Server Hello Done
3198	2024-07-14 23:28:55.696856	10.15.1.2	29986	10.15.1.7	443	TCP	54	29986 → 443 [ACK] Seq=195 Ack=3414 Win=4204288 Len=0
3199	2024-07-14 23:28:55.702443	10.15.1.2	29986	10.15.1.7	443	TLSv1.2	147	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
3200	2024-07-14 23:28:55.702991	10.15.1.7	443	10.15.1.2	29986	TLSv1.2	312	New Session Ticket, Change Cipher Spec, Encrypted Handshake Message
3207	2024-07-14 23:28:55.712838	10.15.1.2	29986	10.15.1.7	443	TCP	54	29986 → 443 [ACK] Seq=288 Ack=3672 Win=4204032 Len=0

Esempio di handshake TLS in Wireshark

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

eth0_diagnostic_logging_tcpdump00_exp-c1_2024-07-15_03_54_39.pcap

No.	Time	Source	Src port	Destination	Dst port	Protocol	Length	Info
270	2024-07-14 21:54:39.347430	10.15.1.2	26105	10.15.1.7	443	TCP	66	26105 → 443 [SYN, ECE, CWR] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM
271	2024-07-14 21:54:39.347496	10.15.1.7	443	10.15.1.2	26105	TCP	66	443 → 26105 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM WS=128
272	2024-07-14 21:54:39.347736	10.15.1.2	26105	10.15.1.7	443	TCP	60	26105 → 443 [ACK] Seq=1 Ack=1 Win=4204800 Len=0
273	2024-07-14 21:54:39.348471	10.15.1.2	26105	10.15.1.7	443	TCP	1514	26105 → 443 [ACK] Seq=1 Ack=1 Win=4204800 Len=0
274	2024-07-14 21:54:39.348508	10.15.1.7	443	10.15.1.2	26105	TCP	54	443 → 26105 [ACK] Seq=1 Ack=1 Win=4204800 Len=0
275	2024-07-14 21:54:39.348533	10.15.1.2	26105	10.15.1.7	443	TLSv1.3	724	Client Hello
276	2024-07-14 21:54:39.348544	10.15.1.7	443	10.15.1.2	26105	TCP	54	443 → 26105 [ACK] Seq=1 Ack=1 Win=4204800 Len=0

> Frame 275: 724 bytes on wire (5792 bits), 724 bytes captured (5792 bits)

> Ethernet II, Src: VMware_b3:fe:d6 (00:50:56:b3:fe:d6), Dst: VMware_b3:5c:7a (00:50:56:b3:5c:7a)

> Internet Protocol Version 4, Src: 10.15.1.2, Dst: 10.15.1.7

> Transmission Control Protocol, Src Port: 26105, Dst Port: 443, Seq: 1461, Ack: 1, Len: 670

> [2 Reassembled TCP Segments (2130 bytes): #273(1460), #275(670)]

▼ Transport Layer Security

- ▼ TLSv1.3 Record Layer: Handshake Protocol: Client Hello
 - Content Type: Handshake (22)
 - Version: TLS 1.0 (0x0301)
 - Length: 2125
 - ▼ Handshake Protocol: Client Hello
 - Handshake Type: Client Hello (1)
 - Length: 2121
 - Version: TLS 1.2 (0x0303)
 - Random: 7a61ba6edc3ff95c4b0672c7f1de5bf4542ced1f5eaa9147bef1cf2e54d83a50
 - Session ID Length: 32
 - Session ID: 98d41a8d7708e9b535baf26310bfea50fd668e69934585b95723670c44ae79f5
 - Cipher Suites Length: 32
 - ▼ Cipher Suites (16 suites)
 - Cipher Suite: Reserved (GREASE) (0xaeaa)
 - Cipher Suite: TLS_AES_128_GCM_SHA256 (0x1301)
 - Cipher Suite: TLS_AES_256_GCM_SHA384 (0x1302)
 - Cipher Suite: TLS_CHACHA20_POLY1305_SHA256 (0x1303)
 - Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 (0xc02b)
 - Cipher Suite: TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (0xc02f)
 - Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 (0xc02c)
 - Cipher Suite: TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (0xc030)
 - Cipher Suite: TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256 (0xcc9)
 - Cipher Suite: TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 (0xcc8)
 - Cipher Suite: TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (0xc013)
 - Cipher Suite: TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0xc014)
 - Cipher Suite: TLS_RSA_WITH_AES_128_GCM_SHA256 (0x009c)
 - Cipher Suite: TLS_RSA_WITH_AES_256_GCM_SHA384 (0x009d)
 - Cipher Suite: TLS_RSA_WITH_AES_128_CBC_SHA (0x002f)
 - Cipher Suite: TLS_RSA_WITH_AES_256_CBC_SHA (0x0035)

Compression Methods Length: 1

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:

The screenshot shows a Wireshark capture of a TLS handshake. The packet list pane displays the following packets:

No.	Time	Source	Src port	Destination	Dst port	Protocol	Length	Info
273	2024-07-14 21:54:39.348471	10.15.1.2	26105	10.15.1.7	443	TCP	1514	26105 → 443 [ACK] Seq=1 Ack=1 Win=4204800 Len=1460 [TCP segment of a reasse...
274	2024-07-14 21:54:39.348508	10.15.1.7	443	10.15.1.2	26105	TCP	54	443 → 26105 [ACK] Seq=1 Ack=1461 Win=64128 Len=0
275	2024-07-14 21:54:39.348533	10.15.1.2	26105	10.15.1.7	26105	TLSv1.3	724	Client Hello
276	2024-07-14 21:54:39.348544	10.15.1.7	443	10.15.1.2	26105	TCP	54	443 → 26105 [ACK] Seq=1 Ack=2131 Win=63488 Len=0
277	2024-07-14 21:54:39.349184	10.15.1.7	443	10.15.1.2	26105	TLSv1.3	314	Server Hello, Change Cipher Spec, Application Data, Application Data
278	2024-07-14 21:54:39.349635	10.15.1.2	26105	10.15.1.7	443	TLSv1.3	134	Change Cipher Spec, Application Data
279	2024-07-14 21:54:39.349976	10.15.1.7	443	10.15.1.2	26105	TLSv1.3	373	Application Data

The packet details pane for packet 277 shows the following information:

- Frame 277: 314 bytes on wire (2512 bits), 314 bytes captured (2512 bits)
- Ethernet II, Src: VMware_b3:5c:7a (00:50:56:b3:5c:7a), Dst: VMware_b3:fe:d6 (00:50:56:b3:fe:d6)
- Internet Protocol Version 4, Src: 10.15.1.7, Dst: 10.15.1.2
- Transmission Control Protocol, Src Port: 443, Dst Port: 26105, Seq: 1, Ack: 2131, Len: 260
- Transport Layer Security
 - TLSv1.3 Record Layer: Handshake Protocol: Server Hello
 - Content Type: Handshake (22)
 - Version: TLS 1.2 (0x0303)
 - Length: 128
 - Handshake Protocol: Server Hello
 - Handshake Type: Server Hello (2)
 - Length: 124
 - Version: TLS 1.2 (0x0303)
 - Random: ae5d8084b4032d2716e681a6d3052d4ea518faf7a87a8490234871ab4e603e5f
 - Session ID Length: 32
 - Session ID: 98d41a8d7708e9b535haf26310bfea50fd668e69934585b95723670c44ae79f5
 - Cipher Suite: TLS_AES_256_GCM_SHA384 (0x1302)
 - Compression Method: null (0)
 - Extensions Length: 52

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:

Status > System > Configuration > Applications > Users > Maintenance >

Ciphers

Configuration

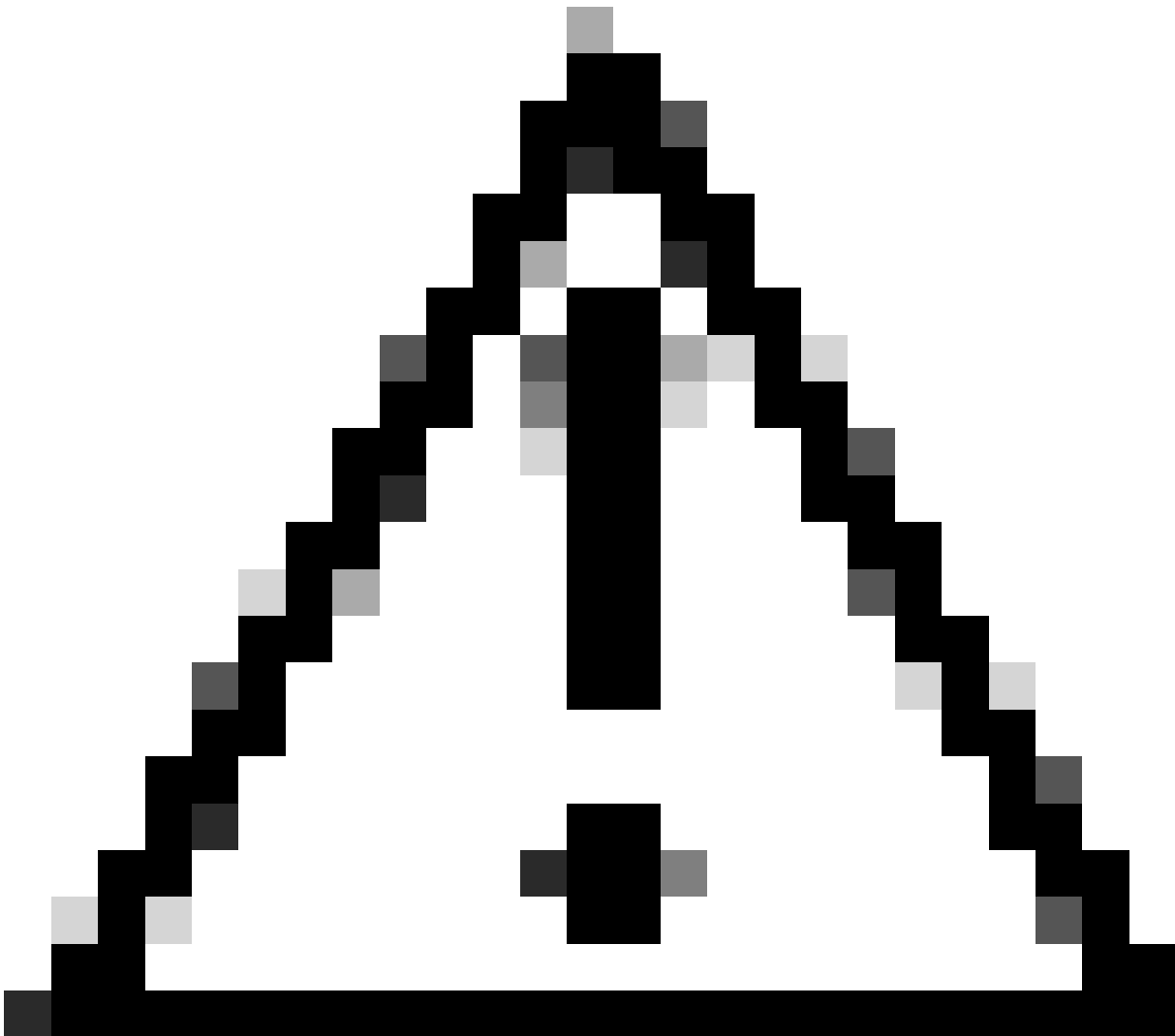
HTTPS ciphers	EECDH:EDH:HIGH:-AES256+SHA:IMEDIUM:LOW:3DES:1MD5:IPSK:!
HTTPS minimum TLS version	TLS v1.2
LDAP TLS Ciphers	EECDH:EDH:HIGH:-AES256+SHA:IMEDIUM:LOW:3DES:1MD5:IPSK:!
LDAP minimum TLS version	TLS v1.2
Reverse proxy TLS ciphers	EECDH:EDH:HIGH:-AES256+SHA:IMEDIUM:LOW:3DES:1MD5:IPSK:!
Reverse proxy minimum TLS version	TLS v1.2
SIP TLS ciphers	!IMEDIUM:LOW:3DES:1MD5:IPSK:! !eNULL:!!eNULL:!!aDH:!!AES128-SHA!
SIP minimum TLS version	TLS v1.2
SMTP TLS Ciphers	EECDH:EDH:HIGH:-AES256+SHA:IMEDIUM:LOW:3DES:1MD5:IPSK:!
SMTP minimum TLS version	TLS v1.2
TMS Provisioning Ciphers	EECDH:EDH:HIGH:-AES256+SHA:IMEDIUM:LOW:3DES:1MD5:IPSK:!
TMS Provisioning minimum TLS version	TLS v1.2
UC server discovery TLS ciphers	EECDH:EDH:HIGH:-AES256+SHA:IMEDIUM:LOW:3DES:1MD5:IPSK:!
UC server discovery minimum TLS version	TLS v1.2
XMPP TLS ciphers	EECDH:EDH:HIGH:-AES256+SHA:IMEDIUM:LOW:3DES:1MD5:IPSK:!
XMPP minimum TLS version	TLS v1.2

Save

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

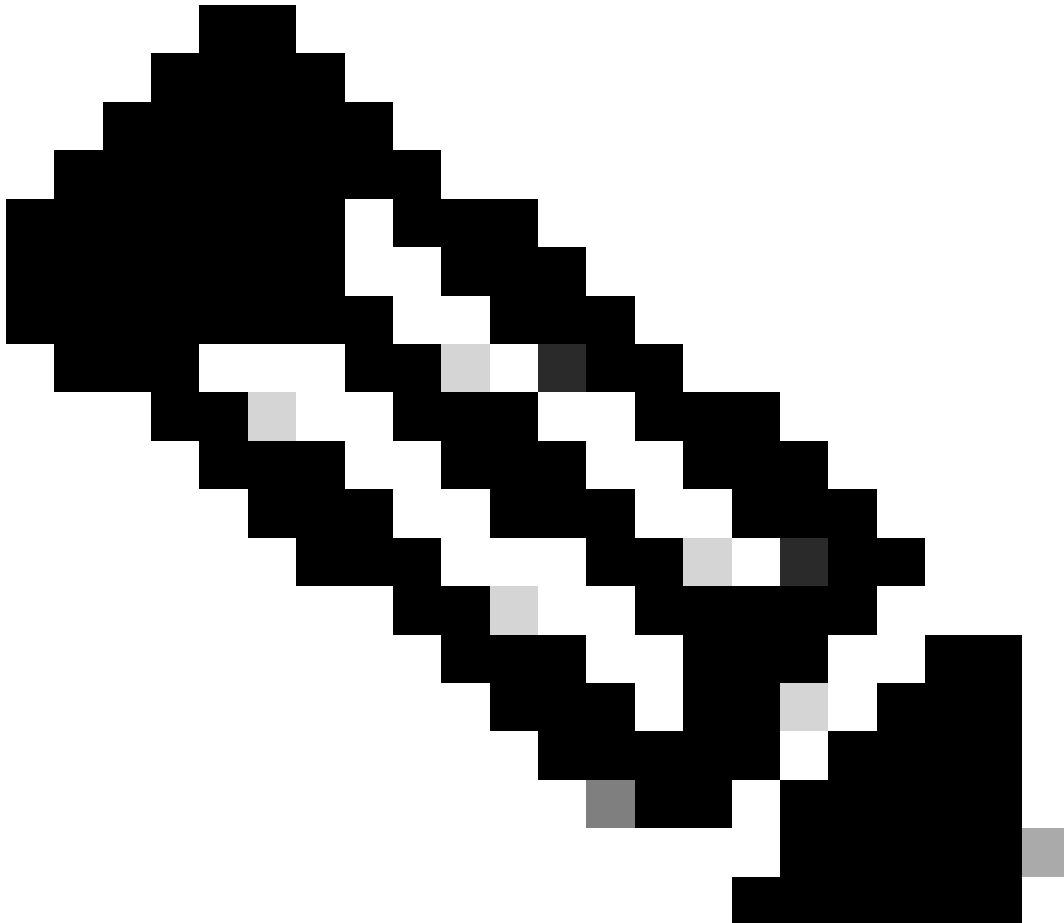
Per disabilitare un gruppo di cifrari che utilizzano un algoritmo comune, aggiungere alla stringa predefinita il separatore :, il segno ! o - e il nome dell'algoritmo da disabilitare. I nomi degli algoritmi supportati sono disponibili nella [pagina di gestione dei cifrari OpenSSL](#). Ad esempio, se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

```
<#root>
```

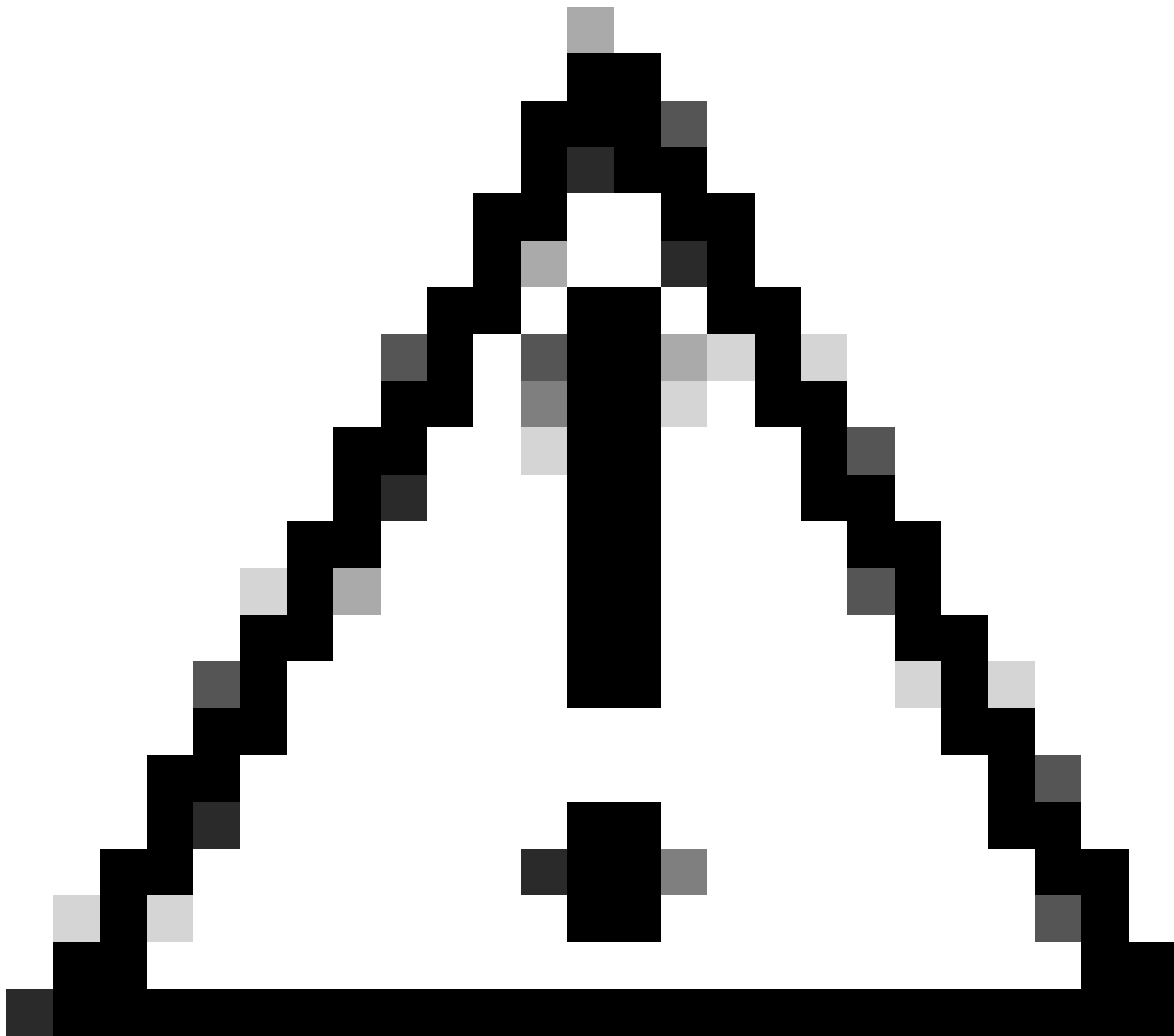
```
EECDH:EDH:HIGH:-AES256+SHA:!MEDIUM:!LOW:!3DES:!MD5:!PSK:!eNULL:!aNULL:!aDH
```

```
:!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:

Urgent Pointer: 0

- > [Timestamps]
- > [SEQ/ACK analysis]
- TCP payload (247 bytes)
- Transport Layer Security
 - TLsv1.3 Record Layer: Handshake Protocol: Client Hello
 - Content Type: Handshake (22)
 - Version: TLS 1.0 (0x0301)
 - Length: 242
 - Handshake Protocol: Client Hello
 - Handshake Type: Client Hello (1)
 - Length: 238
 - Version: TLS 1.2 (0x0303)
 - Random: 19ec4e8994cc334599cf889d4e45a812029589923c4cfcf2cef6b6fc47ec2840
 - Session ID Length: 32
 - Session ID: e0d17cb402229aa46cab70b6a637ce38d9b5a228c7b360cb43f49086ce88d5df
 - Cipher Suites Length: 10
 - Cipher Suites (5 suites)
 - Cipher Suite: TLS_AES_256_GCM_SHA384 (0x1302)
 - Cipher Suite: TLS_CHACHA20_POLY1305_SHA256 (0x1303)
 - Cipher Suite: TLS_AES_128_GCM_SHA256 (0x1301)
 - Cipher Suite: TLS_DHE_RSA_WITH_AES_256_GCM_SHA384 (0xc030)
 - Cipher Suite: TLS_EMPTY_RENEGOTIATION_INFO_SCSV (0x00ff)
 - Compression Methods Length: 1

Ciphers automatically inserted by the openssl s_client command

Cipher passed with the -cipher option

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:

The image shows a Wireshark capture of a network stream on the 'tcp.stream eq 2' filter. The packet list pane shows several packets, with packet 327 highlighted in blue. This packet is a TLSv1.2 Client Hello from source 10.15.1.2 to destination 10.15.1.7. The packet details pane shows the structure of the TLS record, including the Handshake Protocol: Client Hello and the Cipher Suites list. The Cipher Suites list contains two entries: TLS_DHE_RSA_WITH_AES_256_GCM_SHA384 (0xc09f) and TLS_EMPTY_RENEGOTIATION_INFO_SCSV (0x00ff). The first entry is highlighted in blue.

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.

Wireshark interface showing a network capture on 'Ethernet0'. The packet list pane displays several packets, with packet 329 highlighted in red. The packet details pane for packet 329 shows the following information:

No.	Time	Source	Src port	Destination	Dst port	Protocol	Length	Info
324	2024-07-14 23:00:32.459025	10.15.1.2	28872	10.15.1.7	443	TCP	66	28872 → 443 [SYN, ECE, CWR] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM
325	2024-07-14 23:00:32.459666	10.15.1.7	443	10.15.1.2	28872	TCP	66	443 → 28872 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM WS=128
326	2024-07-14 23:00:32.459760	10.15.1.2	28872	10.15.1.7	443	TCP	54	28872 → 443 [ACK] Seq=1 Ack=1 Win=4204800 Len=0
327	2024-07-14 23:00:32.460733	10.15.1.2	28872	10.15.1.7	443	TLSv1.2	172	Client Hello
328	2024-07-14 23:00:32.461070	10.15.1.7	443	10.15.1.2	28872	TCP	60	443 → 28872 [ACK] Seq=1 Ack=119 Win=64128 Len=0
329	2024-07-14 23:00:32.461855	10.15.1.7	443	10.15.1.2	28872	TLSv1.2	61	Alert (Level: Fatal, Description: Handshake Failure)
330	2024-07-14 23:00:32.461855	10.15.1.7	443	10.15.1.2	28872	TCP	60	443 → 28872 [FIN, ACK] Seq=8 Ack=119 Win=64128 Len=0

The packet details pane for packet 329 shows the following information:

- Frame 329: 61 bytes on wire (488 bits), 61 bytes captured (488 bits) on interface \Device\NPF_{122607A1-10A8-47F6-9069-936EB0CAAE1C}, id 0
- Ethernet II, Src: VMware_b3:5c:7a (00:50:56:b3:5c:7a), Dst: VMware_b3:fe:d6 (00:50:56:b3:fe:d6)
- Internet Protocol Version 4, Src: 10.15.1.7, Dst: 10.15.1.2
- Transmission Control Protocol, Src Port: 443, Dst Port: 28872, Seq: 1, Ack: 119, Len: 7
 - Source Port: 443
 - Destination Port: 28872
 - [Stream index: 2]
 - [Conversation completeness: Complete, WITH_DATA (31)]
 - [TCP Segment Len: 7]
 - Sequence Number: 1 (relative sequence number)
 - Sequence Number (raw): 3235581935
 - [Next Sequence Number: 8 (relative sequence number)]
 - Acknowledgment Number: 119 (relative ack number)
 - Acknowledgment number (raw): 810929090
 - 0101 = Header Length: 20 bytes (5)
 - Flags: 0x018 (PSH, ACK)
 - Window: 501
 - [Calculated window size: 64128]
 - [Window size scaling factor: 128]
 - Checksum: 0x163f [unverified]
 - [Checksum Status: Unverified]
 - Urgent Pointer: 0
 - [Timestamps]
 - [SEQ/ACK analysis]
 - TCP payload (7 bytes)
- Transport Layer Security
 - TLSv1.2 Record Layer: Alert (Level: Fatal, Description: Handshake Failure)
 - Content Type: Alert (21)
 - Version: TLS 1.2 (0x0303)
 - Length: 2
 - Alert Message
 - Level: Fatal (2)
 - Description: Handshake Failure (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).