

# 自定義Expressway SSL密碼配置

## 目錄

---

### [簡介](#)

#### [必要條件](#)

[需求](#)

[採用元件](#)

#### [背景資訊](#)

[檢查密碼字串](#)

[使用資料包捕獲檢查TLS握手中的加密協商](#)

#### [設定](#)

[停用特定密碼](#)

[使用通用演算法停用一組密碼](#)

#### [驗證](#)

[檢查密碼字串允許的密碼清單](#)

[透過協商已停用的密碼測試TLS連線](#)

[使用停用的密碼檢查TLSHandshake的資料包捕獲](#)

#### [相關資訊](#)

---

## 簡介

本文檔介紹在Expressway上自定義預配置密碼字串的步驟。

## 必要條件

### 需求

思科建議您瞭解以下主題：

- Cisco Expressway或Cisco VCS。
- TLS協定。

### 採用元件

本文中的資訊係根據以下軟體和硬體版本：

- Cisco Expressway X15.0.2版。

本文中的資訊是根據特定實驗室環境內的裝置所建立。文中使用到的所有裝置皆從已清除（預設）的組態來啟動。如果您的網路運作中，請確保您瞭解任何指令可能造成的影響。

## 背景資訊

預設Expressway配置包含預配置的密碼字串，基於相容性原因，這些字串支援某些在某些企業安全策略下可能被視為較弱的密碼。可以自定義密碼字串，以便對其進行微調，使其符合每個環境的特定策略。

在Expressway中，可以為以下每種協定配置獨立的密碼字串：

- HTTPS
- LDAP
- 反向 Proxy
- SIP
- SMTP
- TMS調配
- UC伺服器發現
- XMPP

密碼字串遵循[OpenSSL Ciphers Manpage](#)中介紹的OpenSSL格式。當前的Expressway版本X15.0.2隨附預設字串EEDH : EDH : HIGH : -

AES256+SHA : ! MEDIUM : ! LOW : ! 3DES : ! MD5 : ! PSK : ! eNULL : ! aNULL : ! aDH，為所有協定平均預配置。在Web管理頁面的維護 > 安全 > 密碼下，您可以修改分配到每個協定的密碼字串，以使用通用演算法增加或刪除特定密碼或密碼組。

## 檢查密碼字串

透過使用openssl ciphers -V 「<cipher string>」命令，您可以輸出包含特定字串允許的所有密碼的清單，這對於檢視密碼非常有用。此示例顯示檢查預設Expressway密碼字串時的輸出：

```
<#root>
```

```
~ #
```

```
openssl ciphers -V "EEDH: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
~ #

```

## 使用資料包捕獲檢查TLS握手中的加密協商

透過在資料包捕獲中捕獲TLS協商，您可以使用Wireshark檢查加密協商的詳細資訊。

TLS握手過程包括由客戶端裝置傳送的ClientHello資料包，根據為連線協定配置的密碼字串提供其支援的密碼清單。伺服器會檢視清單，將它與其自己的允許密碼清單（由其自己的密碼字串決定）進行比較，並選擇兩個系統都支援的密碼，以用於加密的作業階段。然後，它會以指示所選密碼的ServerHello資料包進行響應。TLS 1.2和1.3握手對話方塊之間有著重要的區別，但是密碼協商機制在兩個版本中都使用相同的原則。

以下是Web瀏覽器與埠443上的Expressway之間的TLS 1.3密碼協商示例（如Wireshark所示）：

| 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                         |

Wireshark中的TLS握手示例

首先，瀏覽器傳送一個包含其支援的密碼清單的ClientHello資料包：

eth0\_diagnostic\_logging\_tcpdump00\_exp-c1\_2024-07-15\_03\_54\_39.pcap

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

tcp.stream eq 7

| 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, EC  |
| 271 | 2024-07-14 21:54:39.347496 | 10.15.1.7 | 443      | 10.15.1.2   | 26105    | TCP      | 66     | 443 → 26105 [SYN, AC  |
| 272 | 2024-07-14 21:54:39.347736 | 10.15.1.2 | 26105    | 10.15.1.7   | 443      | TCP      | 60     | 26105 → 443 [ACK] Ser |
| 273 | 2024-07-14 21:54:39.348471 | 10.15.1.2 | 26105    | 10.15.1.7   | 443      | TCP      | 1514   | 26105 → 443 [ACK] Ser |
| 274 | 2024-07-14 21:54:39.348508 | 10.15.1.7 | 443      | 10.15.1.2   | 26105    | TCP      | 54     | 443 → 26105 [ACK] Ser |
| 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] Ser |

> 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 (0xc0ca9)
      - Cipher Suite: TLS\_ECDHE\_RSA\_WITH\_CHACHA20\_POLY1305\_SHA256 (0xc0ca8)
      - 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

Wireshark中的ClientHello資料包示例

Expressway會檢查其為HTTPS協定配置的密碼字串，並找到自身和客戶端都支援的密碼。在本示例中，選擇ECDHE-RSA-AES256-GCM-SHA384密碼。Expressway以其ServerHello資料包做出響應，其中指示所選密碼：

```
eth0_diagnostic_logging_tcpdump00_exp-c1_2024-07-15_03_54_39.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
tcp.stream eq 7
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 10.15.1.7 443 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=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
<
> 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: 98d41a8d7708e9b535baf26310bfea50fd668e69934585b95723670c44ae79f5
      Cipher Suite: TLS_AES_256_GCM_SHA384 (0x1302)
      Compression Method: null (0)
      Extensions Length: 52
```

Wireshark中的ServerHello資料包示例

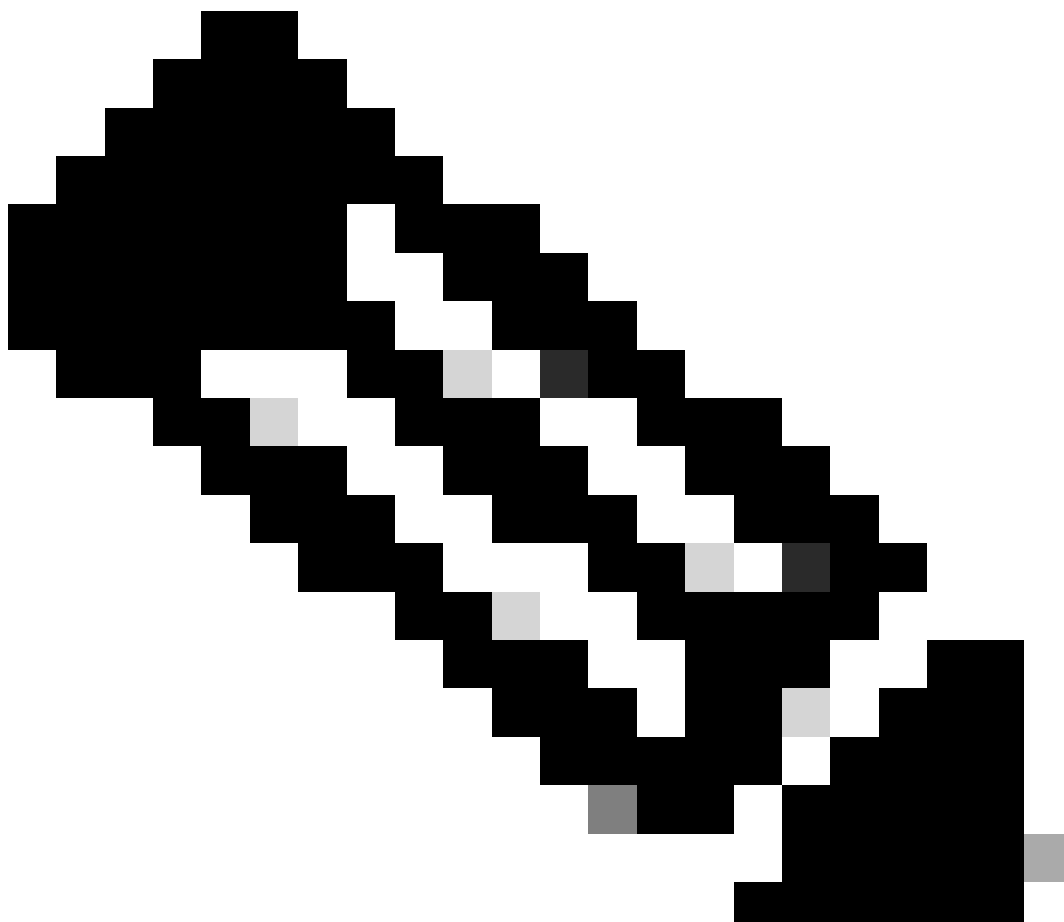
## 設定

OpenSSL密碼字符串格式包含數個特殊字元，以便對字串執行作業，例如移除特定密碼或共用共同元件的密碼群組。由於這些自定義目的通常是刪除密碼，因此這些示例中使用的字元包括：

- -字元，用於從清單中刪除密碼。部分或全部刪除的密碼可以透過稍後出現在字串中的選項再次被允許。
- !字元，也用於從清單中刪除密碼。使用它時，字串中以後出現的任何其它選項都不能再次允許刪除的密碼。
- :字元，該字元充當清單中專案之間的分隔符。

兩種方法都可用於從字串中刪除密碼，但是!是首選。有關特殊字元的完整清單，請檢視[OpenSSL Ciphers Manpage](#)。

---



注意：OpenSSL網站指出，使用！字元時，「刪除的密碼即使已明確指出，也絕不會重新出現在清單中」。這並不意味著密碼將從系統中永久刪除，而是指密碼字串的解釋範圍。

---

## 停用特定密碼

要停用特定密碼，請向預設字串增加：分隔符、！或-符號以及要停用的密碼名稱。密碼名稱必須遵循OpenSSL命名格式，在[OpenSSL Ciphers Manpage](#)中提供了此格式。例如，如果需要停用SIP連線的AES128-SHA密碼，請配置如下所示的密碼字串：

```
<#root>
```

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

```
:!AES128-SHA
```

然後，導航到Expressway Web管理頁面，導航到維護>安全>密碼，將自定義字串分配到所需協定

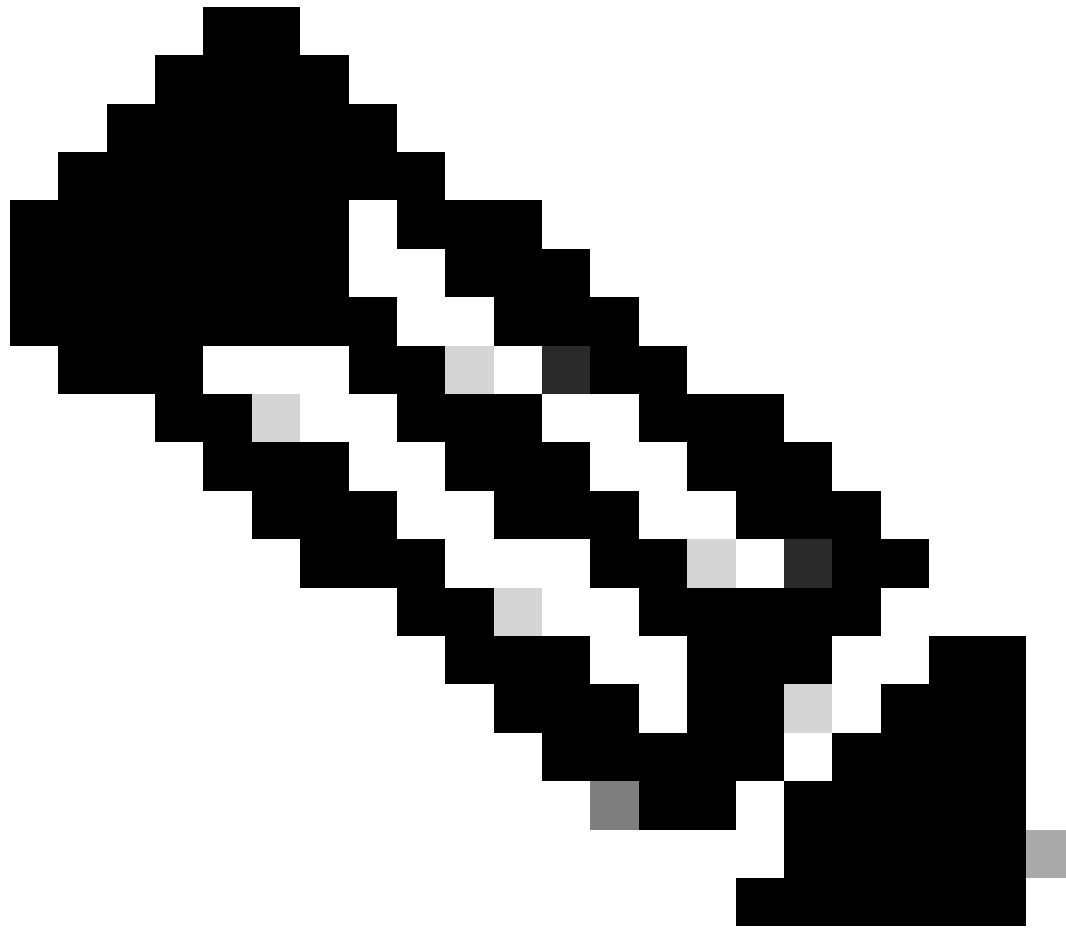
，然後按一下儲存。要應用新配置，需要重新啟動系統。在本示例中，在SIP TLS密碼下將自定義字串分配給SIP協定：

The screenshot shows the 'Ciphers' configuration page in the Expressway Web Management Portal. The page has a navigation bar with 'Status >', 'System >', 'Configuration >', 'Applications >', 'Users >', and 'Maintenance >'. The 'Configuration' section is active, and the 'Ciphers' tab is selected. The configuration table lists various TLS ciphers and their minimum versions. The 'SIP TLS ciphers' field is highlighted with a red box and contains the text '!MEDIUM:!LOW:!3DES:!MD5:!PSK:!eNULL:!aNULL:!aDH:!AES128-SHA'. A 'Save' button is also highlighted with a red box.

| Configuration Item                      | Value  |
|---|--|
| HTTPS ciphers                           | EECDH:EDH:HIGH:-AES256+SHA:!MEDIUM:!LOW:!3DES:!MD5:!PSK:!          |
| HTTPS minimum TLS version               | TLS v1.2   |
| LDAP TLS Ciphers                        | EECDH:EDH:HIGH:-AES256+SHA:!MEDIUM:!LOW:!3DES:!MD5:!PSK:!          |
| LDAP minimum TLS version                | TLS v1.2   |
| Reverse proxy TLS ciphers               | EECDH:EDH:HIGH:-AES256+SHA:!MEDIUM:!LOW:!3DES:!MD5:!PSK:!          |
| Reverse proxy minimum TLS version       | TLS v1.2   |
| <b>SIP TLS ciphers</b>                  | <b>!MEDIUM:!LOW:!3DES:!MD5:!PSK:!eNULL:!aNULL:!aDH:!AES128-SHA</b> |
| SIP minimum TLS version                 | TLS v1.2   |
| SMTP TLS Ciphers                        | EECDH:EDH:HIGH:-AES256+SHA:!MEDIUM:!LOW:!3DES:!MD5:!PSK:!          |
| SMTP minimum TLS version                | TLS v1.2   |
| TMS Provisioning Ciphers                | EECDH:EDH:HIGH:-AES256+SHA:!MEDIUM:!LOW:!3DES:!MD5:!PSK:!          |
| TMS Provisioning minimum TLS version    | TLS v1.2   |
| UC server discovery TLS ciphers         | EECDH:EDH:HIGH:-AES256+SHA:!MEDIUM:!LOW:!3DES:!MD5:!PSK:!          |
| UC server discovery minimum TLS version | TLS v1.2   |
| XMPP TLS ciphers                        | EECDH:EDH:HIGH:-AES256+SHA:!MEDIUM:!LOW:!3DES:!MD5:!PSK:!          |
| XMPP minimum TLS version                | TLS v1.2   |

**Save**

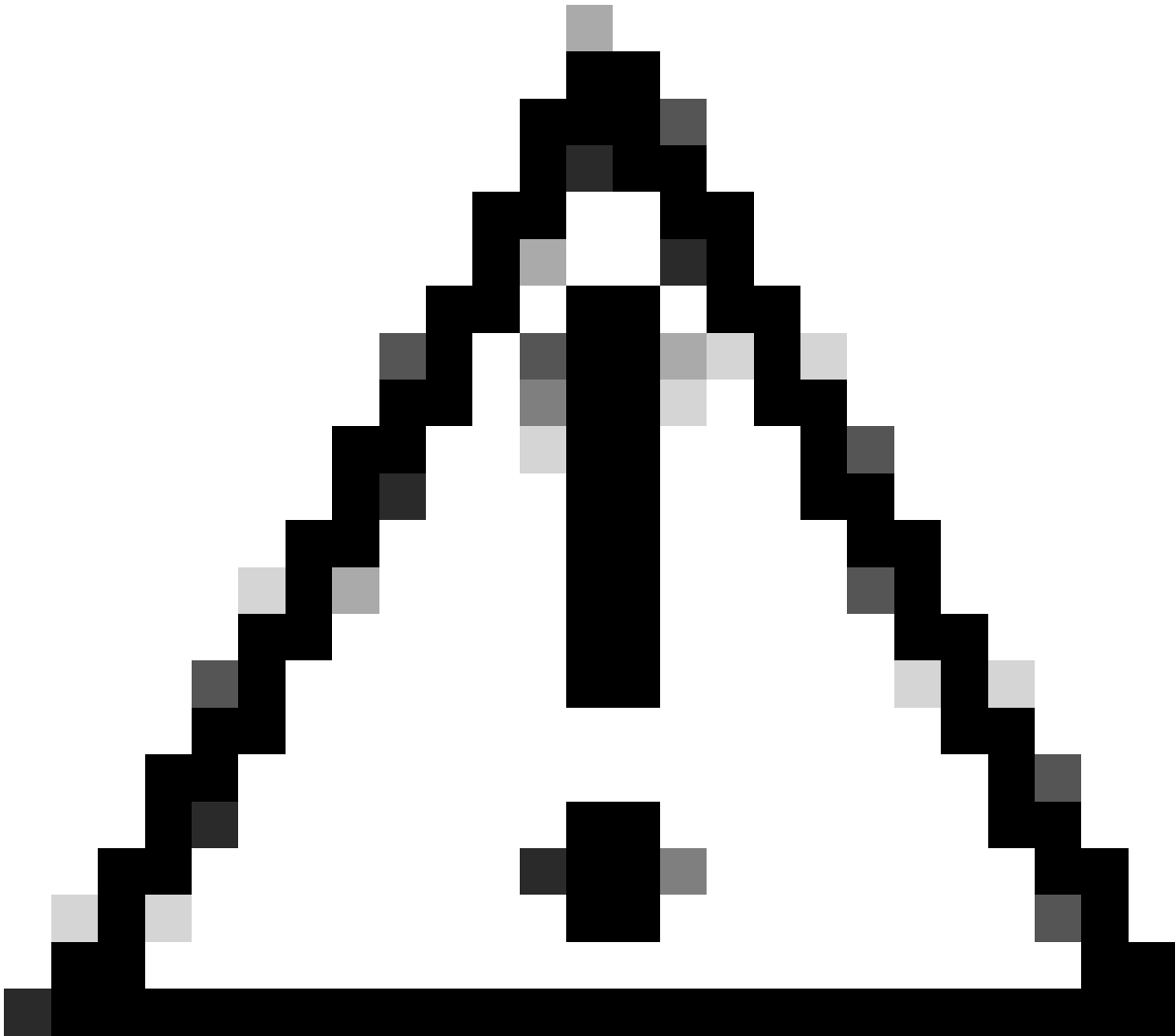
Expressway Web管理門戶上的密碼設定頁面



注意：如果是Expressway集群，請僅在主伺服器上進行更改。新配置將複製到其餘的集群成員。

---





注意：使用 [《Cisco Expressway 集群建立和維護部署指南》](#) 中提供的建議集群重新引導順序。首先重新啟動主伺服器，等待可以透過Web介面訪問它，然後根據System > Clustering下配置的清單對每台對等體執行相同的操作。

---

## 使用通用演算法停用一組密碼

要使用常用演算法停用一組密碼，請將要停用的演算法名稱、：分隔符、！或-符號以及預設字串附加到預設字串中。[OpenSSL Ciphers Manpage](#)中提供了支援的演算法名稱。例如，如果需要停用所有使用DHE演算法的密碼，請配置如下所示的密碼字串：

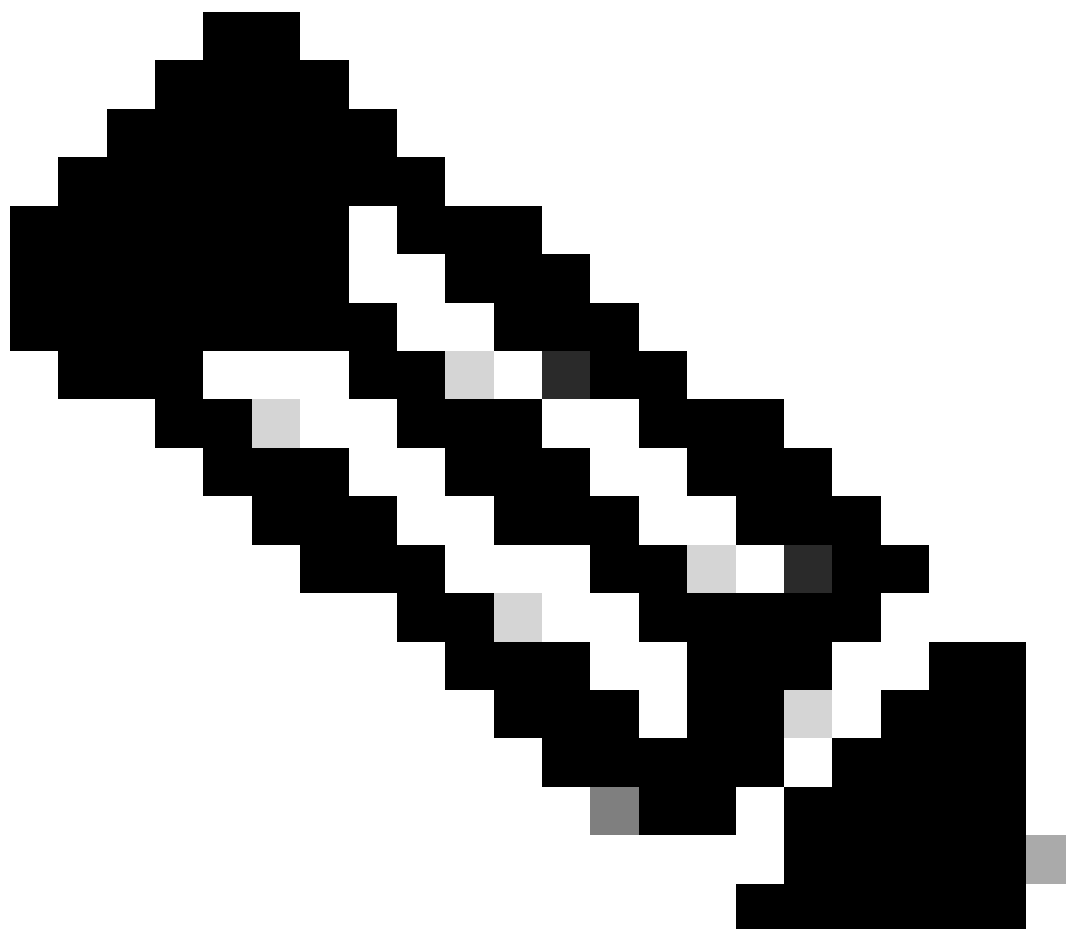
```
<#root>
```

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

```
:!DHE
```

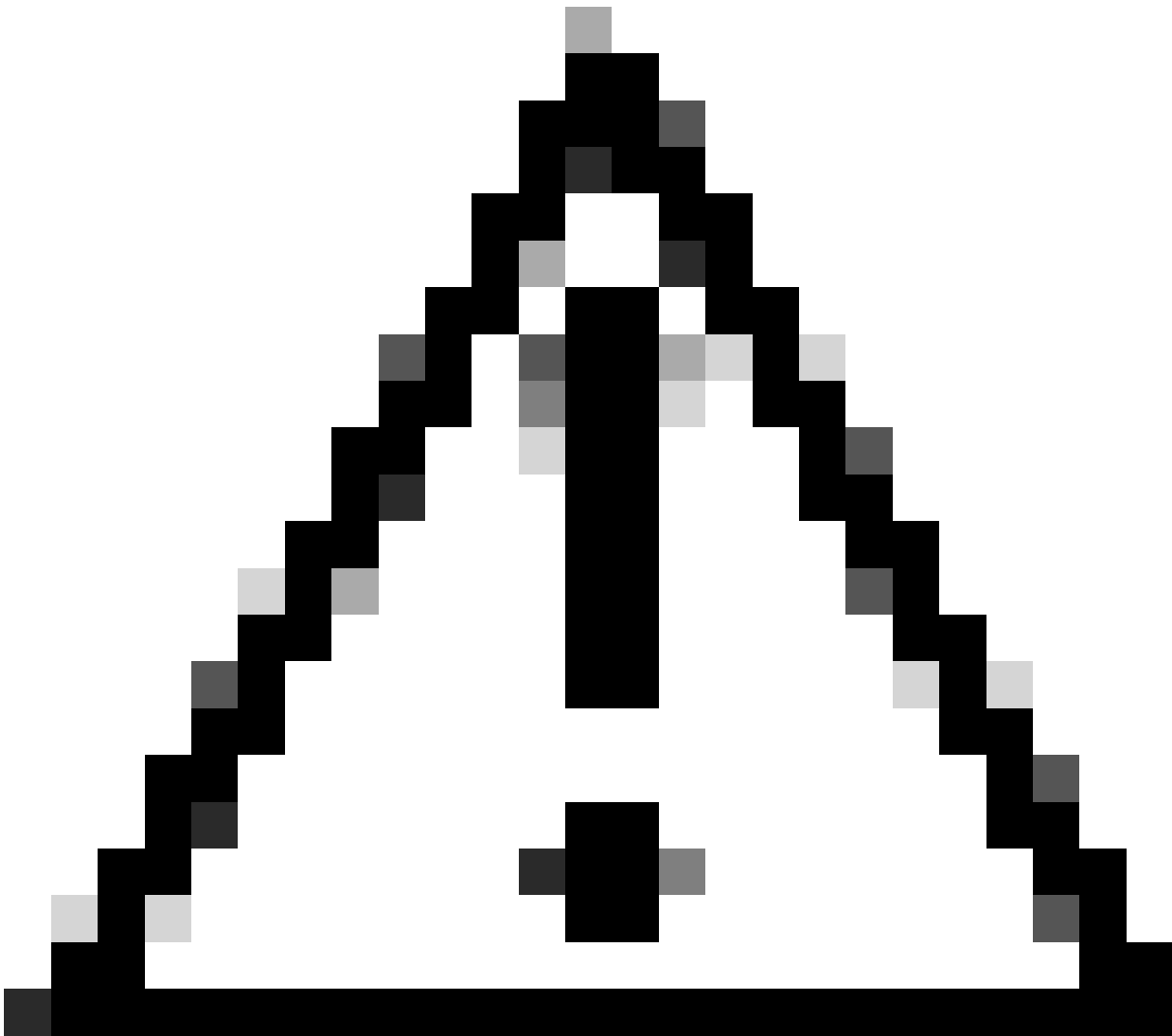
導航到Expressway Web管理頁，導航到維護>安全>密碼，將自定義字串分配到所需協定，然後按一下儲存。要應用新配置，需要重新啟動系統。

---



注意：如果是Expressway集群，請僅在主伺服器上進行更改。新配置將複製到其餘的集群成員。

---



注意：使用 [《Cisco Expressway 集群建立和維護部署指南》](#) 中提供的建議集群重新引導順序。首先重新啟動主伺服器，等待可以透過Web介面訪問它，然後根據System > Clustering下配置的清單對每台對等體執行相同的操作。

---

## 驗證

### 檢查密碼字串允許的密碼清單

您可以使用openssl ciphers -V 「<cipher string>」命令檢查自定義的加密字串。檢視輸出以確認更改後不再列出不需要的密碼。在此範例中，會檢查EECDH : EDH : HIGH : - AES256+SHA : ! MEDIUM : ! LOW : ! 3DES : ! MD5 : ! PSK : ! eNULL : ! aNULL : ! aDH : ! DH 密碼字串。命令輸出確認字串不允許使用DHE演算法的任何密碼：

```
<#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
```

```
~ #
```

## 透過協商已停用的密碼測試TLS連線

您可以使用 `openssl s_client` 命令來驗證是否已拒絕使用停用密碼的連線嘗試。使用 `-connect` 選項指定您的 Expressway 地址和埠，並使用 `-cipher` 選項指定在 TLS 握手期間由客戶端協商的單個密碼：

```
openssl s_client -connect <地址> : <埠> -cipher <密碼> -no_tls1_3
```

在本示例中，從安裝了 openssl 的 Windows PC 嘗試到 Expressway 的 TLS 連線。PC 作為客戶端，僅協商不想要的 DHE-RSA-AES256-CCM 密碼，該密碼使用 DHE 演算法：

```
<#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
---

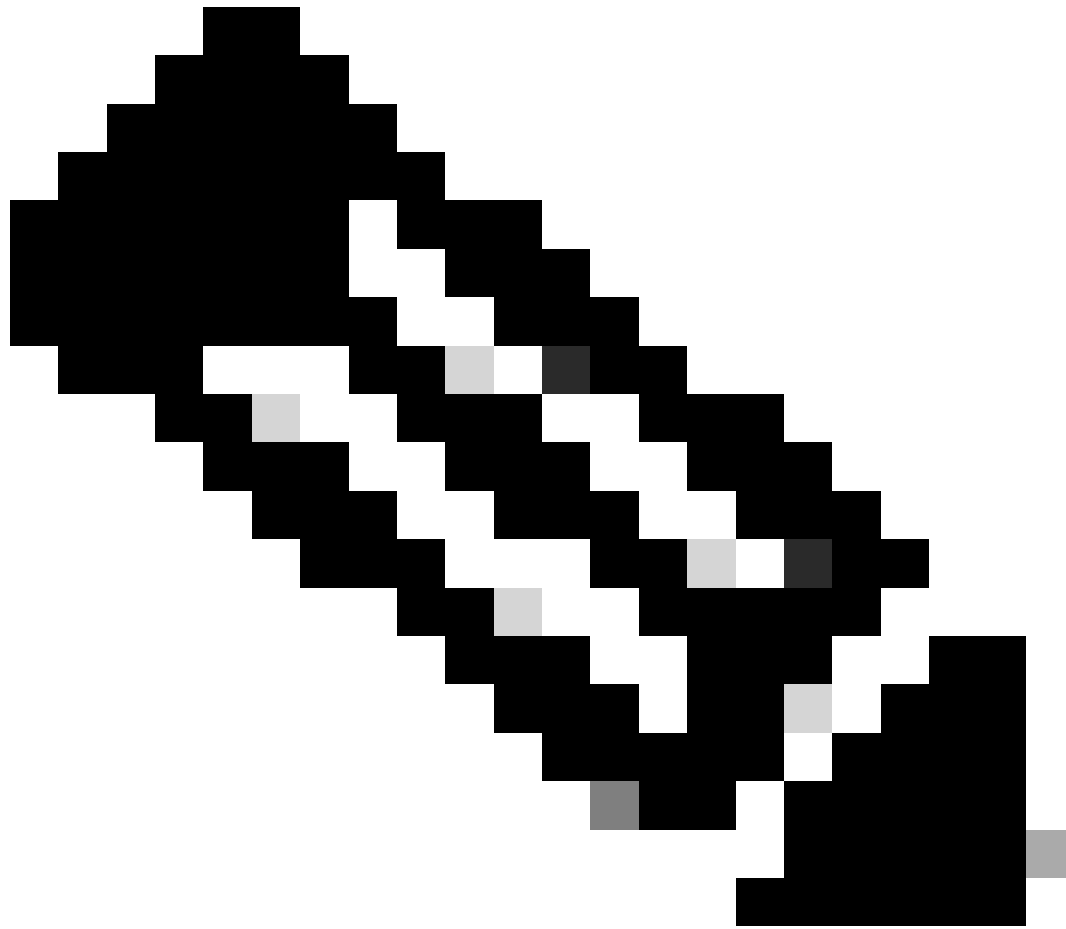
C:\Users\Administrator>
```

命令輸出顯示連線嘗試失敗並顯示「ssl/tls警報握手失敗

: ..\ssl\record\rec\_layer\_s3.c : 865 : SSL警報編號40」錯誤消息，因為Expressway配置為使用

EECDH : EDH : HIGH : -

AES256+SHA : ! MEDIUM : ! LOW : ! 3DES : ! MD5 : ! PSK : ! eNULL : ! aDH : ! DHE密碼字串來停用DHE演算法。



注意：為了使使用openssl s\_client命令的測試能夠如說明的那樣工作，需要將-no\_tls1\_3選項傳遞給命令。如果未包含，客戶端會自動在ClientHello資料包中插入TLS 1.3密碼：

---

Urgent Pointer: 0

- > [Timestamps]
- > [SEQ/ACK analysis]
- TCP payload (247 bytes)
- Transport Layer Security
  - TLV1.3 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 (0xc09f)
        - 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

帶有自動增加密碼的ClientHello資料包

如果目標Expressway支援這些密碼，則可以選擇其中一個密碼，而不是您需要測試的特定密碼。連線成功，這可以讓您相信可以使用以-cipher選項傳送至指令的已停用密碼來連線

o

## 使用停用的密碼檢查TLS握手的資料包捕獲

在使用某個停用的密碼執行連線測試時，您可以從測試裝置或Expressway收集資料包捕獲。然後，您可以使用Wireshark對其進行檢查，以進一步分析握手事件。

查詢測試裝置傳送的ClientHello。確認它只協商不需要的測試密碼，在本例中是使用DHE演算法的密碼：

The image displays a Wireshark capture of a network packet. The top pane shows a list of packets, with packet 327 highlighted in red. This packet is a Client Hello from source 10.15.1.2 to destination 10.15.1.7. The bottom pane shows the detailed structure of this packet, including the TLSv1.2 record layer, the Handshake Protocol: Client Hello, and the Cipher Suites list. The Cipher Suite TLS\_DHE\_RSA\_WITH\_AES\_256\_GCM\_SHA384 is highlighted in blue.

Wireshark中的ClientHello資料包示例

:

確認Expressway以嚴重TLS警報資料包響應，拒絕連線。在本示例中，由於Expressway不支援按其HTTPS協定配置的密碼字串使用DHE密碼，因此它會以包含故障代碼40的嚴重TLS警報資料包進行響應。



Wireshark interface showing network traffic analysis. The packet list pane displays several packets, with packet 329 highlighted in red. The packet details pane shows the structure of this packet, including TCP and TLSv1.2 layers.

| 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                         |

Packet 329 details:

- 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)

Wireshark中的TLS嚴重警報資料包

## 相關資訊

- [OpenSSL密碼個人頁面](#)
- [Cisco Expressway管理員指南\(X15.0\) -章節：管理安全性-配置最低TLS版本和密碼套件](#)

## 關於此翻譯

思科已使用電腦和人工技術翻譯本文件，讓全世界的使用者能夠以自己的語言理解支援內容。請注意，即使是最佳機器翻譯，也不如專業譯者翻譯的內容準確。Cisco Systems, Inc. 對這些翻譯的準確度概不負責，並建議一律查看原始英文文件（提供連結）。