

思科安全防火牆管理中心(FMC)上的代理疑難排解

目錄

[簡介](#)

- [需求](#)
- [採用元件](#)

[組態](#)

[疑難排解](#)

[驗證](#)

[已知的問題](#)

- [代理ACL限制](#)
- [代理檔案下載失敗 \(超時/傳輸不完整 \)](#)
- [代理檔案下載失敗 \(MTU問題 \)](#)

[參考資料](#)

簡介

本檔案介紹在FMC上設定代理，允許使用者透過中間伺服器連線至網際網路，增強安全性，有時改善效能。這篇文章將引導您完成在FMC上配置Proxy的步驟，並提供常見問題的故障排除提示。

需求

思科建議您瞭解以下主題：

- [思科安全防火牆管理中心\(FMC\)](#)
- [代理](#)

採用元件

本文中的資訊係根據以下軟體和硬體版本：

- FMC 7.4.x

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。

組態

在FMC GUI上配置網路http-proxy:

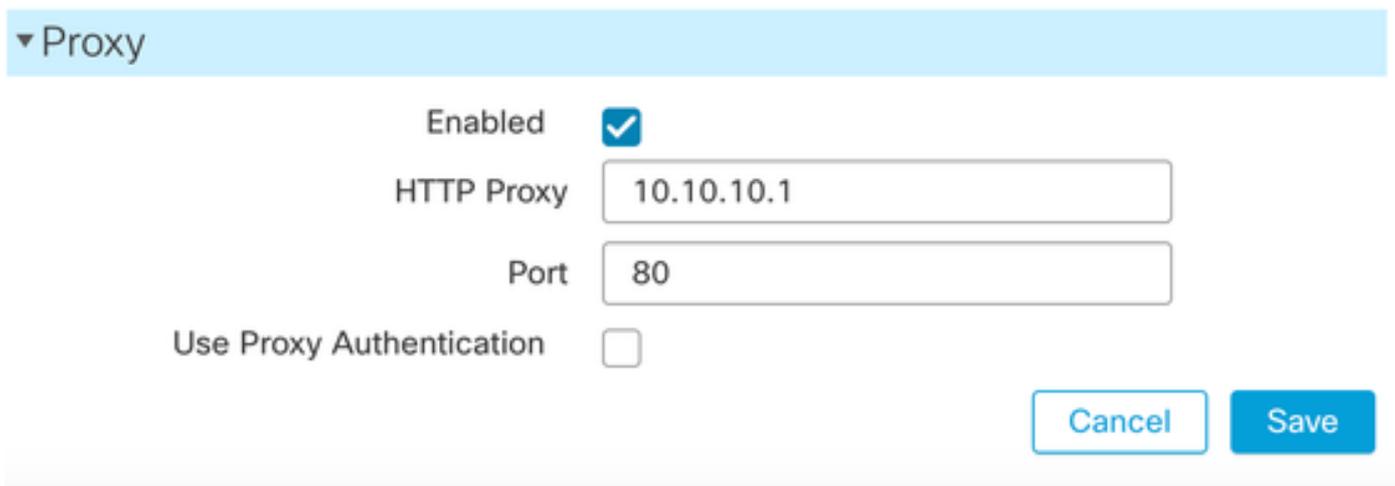
登入FMC GUI > 選擇System > Configuration，然後選擇Management Interfaces。

 附註：不支援使用NT LAN Manager(NTLM)身份驗證的代理。如果使用智慧許可，代理FQDN不能超過64個字元。

在「Proxy」區域中，配置HTTP Proxy設定。

管理中心配置為通過埠TCP/443(HTTPS)和TCP/80(HTTP)直接連線到網際網路。您可以使用可能通過HTTP摘要進行身份驗證的代理伺服器。

- 選中Enabled覈取方塊。
- 在HTTP代理欄位中，輸入代理伺服器的IP地址或完全限定域名。
- 在Portfield中，輸入埠號。
- 通過選擇使用代理身份驗證(Use Proxy Authentication)提供身份驗證憑據，然後提供用戶名和密碼。
- 按一下「Save」。



▼ Proxy

Enabled

HTTP Proxy

Port

Use Proxy Authentication

Cancel Save

 附註：對於代理密碼，您可以使用A-Z、a-z和0-9以及特殊字元。

疑難排解

訪問FMC CLI和專家模式，然後驗證iprep_proxy.conf以確保代理設定正確：

```
<#root>
admin@fmc:~$
cat /etc/sf/iprep_proxy.conf

iprep_proxy {
PROXY_HOST 10.10.10.1;
PROXY_PORT 80;
}
```

檢查活動連線以驗證活動的代理連線：

```
<#root>
admin@fmc:~$
netstat -na | grep 10.10.10.1

tcp 0 0 10.40.40.1:40220 10.10.10.1:80
ESTABLISHED
```

使用curl命令驗證請求詳細資訊和來自Proxy的響應。如果您收到回應：HTTP/1.1 200 Connection已建立，則表示FMC正在透過代理成功傳送和接收流量。

```
<#root>
admin@fmc:~$
curl -x http://10.10.10.1:80 -I https://tools.cisco.com

HTTP/1.1 200 Connection established
```

如果您已為代理配置了使用者名稱和密碼，請驗證身份驗證和代理響應：

```
curl -u proxyuser:proxypass --proxy http://proxy.example.com:80 https://example.com
```

驗證

通過代理成功建立連線

使用代理(例如curl -x http://proxy:80 -I https://tools.cisco.com)執行curl命令時，會發生一系列預期的網路互動，透過封包擷取(tcpdump)可以觀察到。這是該過程的高級概述，並提供了實際tcpdump輸出：

TCP握手啟動：

使用者端(FMC)透過傳送SYN封包來啟動與連線埠80上的代理伺服器的TCP連線。代理用SYN-ACK響應，客戶端用ACK完成握手。這將建立HTTP通訊所經過的TCP會話。

tcpdump輸出示例：

```
10:20:58.987654 IP client.54321 > proxy.80: Flags [S], seq 0, win 64240, options [mss 1460], length 0
10:20:58.987700 IP proxy.80 > client.54321: Flags [S.], seq 0, ack 1, win 65160, options [mss 1460], length 0
10:20:58.987734 IP client.54321 > proxy.80: Flags [.], ack 1, win 64240, length 0
```

HTTP CONNECT請求：

建立TCP連線後，使用者端會向代理傳送HTTP CONNECT要求，指示其建立前往目標HTTPS伺服器(tools.cisco.com:443)的通道。此請求允許客戶端通過代理協商端到端TLS會話。

示例tcpdump (已解碼HTTP)：

```
CONNECT tools.cisco.com:443 HTTP/1.1
Host: tools.cisco.com:443
User-Agent: curl/8.5.0
Proxy-Connection: Keep-Alive
```

連線建立確認：

Proxy使用HTTP/1.1 200 Connection established響應進行回覆，表示已成功建立到目標伺服器的隧道。這表示代理現在充當中繼，在客戶端和tools.cisco.com之間轉發加密流量。

示例tcpdump:

```
<#root>
```

```
HTTP/1.1
```

```
200
```

```
Connection established
```

通過隧道的HTTPS通訊：

在成功執行CONNECT響應後，客戶端會通過已建立的隧道直接與tools.cisco.com發起SSL/TLS握手。由於此流量是加密的，因此內容在tcpdump中不可見，但資料包長度和定時是可觀測的，包括TLS客戶端Hello資料包和伺服器Hello資料包。

示例tcpdump:

```
10:20:59.123456 IP client.54321 > proxy.80: Flags [P.], length 517 (Client Hello)
10:20:59.123789 IP proxy.80 > client.54321: Flags [P.], length 1514 (Server Hello)
```

處理HTTP重新導向（找到302個）：

作為HTTPS通訊的一部分，客戶端從tools.cisco.com請求資源。伺服器會使用HTTP/1.1 302 Found重新導向到另一個URL(<https://tools.cisco.com/healthcheck>)，使用者端可以按照該URL執行的指令，視要求的捲曲引數和用途而定。雖然此重定向在加密的TLS會話中發生並且不直接可見，但是它是預期的行為，並且如果TLS流量被解密，則可以觀察到。

加密的重新導向流量顯示如下：

```
10:21:00.123000 IP client.54321 > proxy.80: Flags [P.], length 517 (Encrypted Application Data)
10:21:00.123045 IP proxy.80 > client.54321: Flags [P.], length 317 (Encrypted Application Data)
```

連線拆卸：

交換完成後，客戶端和代理通過交換FIN和ACK資料包優雅地關閉TCP連線，從而確保正確的會話終止。

tcpdump輸出示例：

```
10:21:05.000111 IP client.54321 > proxy.80: Flags [F.], seq 1234, ack 5678, length 0
10:21:05.000120 IP proxy.80 > client.54321: Flags [F.], seq 5678, ack 1235, length 0
10:21:05.000125 IP client.54321 > proxy.80: Flags [F.], seq 1235, ack 5679, length 0
```

 提示：通過分析tcpdump輸出，您可以驗證通過顯式代理的HTTPS請求是否遵循預期流：TCP握手、CONNECT請求、隧道建立、TLS握手、加密通訊（包括可能的重定向）以及流暢的連線關閉。這可以確認代理和客戶端互動是否按設計方式工作，並幫助識別流中的任何問題，例如隧道或SSL協商失敗。

FMC(10.40.40.1)在埠80上與代理(10.10.10.1)成功建立TCP握手，然後是埠443上與伺服器(72.163.4.161)的HTTP CONNECT。伺服器以HTTP 200 Connection established消息回覆。TLS握手完成，並且資料流正常。最後，TCP連線正常終止(FIN)。

```

No. | Time | Source | Destination | Protocol | Length | Info
--- | --- | --- | --- | --- | --- | ---
2 2025-03-14 11:30:08.972535 10.40.40.1 10.10.10.1 TCP 60 60468 → 80 [ACK] Seq=1 Ack=26 Win=501 Len=0 TSval=99574285 TSecr=3159965226
3 2025-03-14 11:30:10.275579 10.40.40.1 10.10.10.1 TCP 95 60468 → 80 [PSH, ACK] Seq=1 Ack=26 Win=501 Len=29 TSval=995744106 TSecr=3159965226
4 2025-03-14 11:30:10.282765 10.10.10.1 10.40.40.1 TCP 66 80 → 60468 [ACK] Seq=26 Ack=30 Win=4101 Len=0 TSval=3159966536 TSecr=995744106
5 2025-03-14 11:30:12.517129 10.40.40.1 10.10.10.1 TCP 74 48716 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=995746347 TSecr=0 WS=128
6 2025-03-14 11:30:12.536846 10.10.10.1 10.40.40.1 TCP 74 80 → 48716 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1300 WS=64 SACK_PERM TSval=1921884872 TSecr=995746347
7 2025-03-14 11:30:12.536913 10.40.40.1 10.10.10.1 TCP 66 48716 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=995746367 TSecr=1921884872
8 2025-03-14 11:30:12.536989 10.40.40.1 10.10.10.1 HTTP 188 CONNECT tools.cisco.com:443 HTTP/1.1
9 2025-03-14 11:30:12.569594 10.10.10.1 10.40.40.1 TCP 66 [TCP Window Update] 80 → 48716 [ACK] Seq=1 Ack=1 Win=262528 Len=0 TSval=1921884872 TSecr=995746367
2025-03-14 11:30:12.569885 10.10.10.1 10.40.40.1 TCP 66 80 → 48716 [ACK] Seq=1 Ack=123 Win=262400 Len=0 TSval=1921884872 TSecr=995746367
2025-03-14 11:30:12.713622 10.10.10.1 10.40.40.1 HTTP 105 HTTP/1.1 200 Connection established
2025-03-14 11:30:12.713676 10.40.40.1 10.10.10.1 TCP 66 48716 → 80 [ACK] Seq=123 Ack=40 Win=64256 Len=0 TSval=995746544 TSecr=1921885012
2025-03-14 11:30:12.752166 10.40.40.1 10.10.10.1 TLSv1.2 583 Client Hello (SNI=tools.cisco.com)
2025-03-14 11:30:12.772238 10.10.10.1 10.40.40.1 TCP 66 80 → 48716 [ACK] Seq=40 Ack=640 Win=262016 Len=0 TSval=1921885092 TSecr=995746582

> Frame 8: 188 bytes on wire (1504 bits), 188 bytes captured (1504 bits) on interface 0
> Ethernet II, Src: VMware_8d:76:9d (00:50:56:8d:76:9d), Dst: Cisco_9d:b9:ff (4c:71:0d:9d:b9:ff)
> Internet Protocol Version 4, Src: 10.40.40.1, Dst: 10.10.10.1
> Transmission Control Protocol, Src Port: 48716, Dst Port: 80, Seq: 1, Ack: 1, Len: 122
> Hypertext Transfer Protocol
  > CONNECT tools.cisco.com:443 HTTP/1.1\r\n
    Request Method: CONNECT
    Request URI: tools.cisco.com:443
    Request Version: HTTP/1.1
    Host: tools.cisco.com:443\r\n
    User-Agent: curl/7.79.1\r\n
    Proxy-Connection: Keep-Alive\r\n
    \r\n
    [Response in frame: 11]
    [Full request URI: tools.cisco.com:443]
  
```

```

No. | Time | Source | Destination | Protocol | Length | Info
--- | --- | --- | --- | --- | --- | ---
3 2025-03-14 11:30:10.275579 10.40.40.1 10.10.10.1 TCP 95 60468 → 80 [PSH, ACK] Seq=1 Ack=26 Win=501 Len=29 TSval=995744106 TSecr=3159965226
4 2025-03-14 11:30:10.282765 10.10.10.1 10.40.40.1 TCP 66 80 → 60468 [ACK] Seq=26 Ack=30 Win=4101 Len=0 TSval=3159966536 TSecr=995744106
5 2025-03-14 11:30:12.517129 10.40.40.1 10.10.10.1 TCP 74 48716 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=995746347 TSecr=0 WS=128
6 2025-03-14 11:30:12.536846 10.10.10.1 10.40.40.1 TCP 74 80 → 48716 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1300 WS=64 SACK_PERM TSval=1921884872 TSecr=995746347
7 2025-03-14 11:30:12.536913 10.40.40.1 10.10.10.1 TCP 66 48716 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=995746367 TSecr=1921884872
8 2025-03-14 11:30:12.536989 10.40.40.1 10.10.10.1 HTTP 188 CONNECT tools.cisco.com:443 HTTP/1.1
9 2025-03-14 11:30:12.569594 10.10.10.1 10.40.40.1 TCP 66 [TCP Window Update] 80 → 48716 [ACK] Seq=1 Ack=1 Win=262528 Len=0 TSval=1921884872 TSecr=995746367
2025-03-14 11:30:12.569885 10.10.10.1 10.40.40.1 TCP 66 80 → 48716 [ACK] Seq=1 Ack=123 Win=262400 Len=0 TSval=1921884872 TSecr=995746367
2025-03-14 11:30:12.713622 10.10.10.1 10.40.40.1 HTTP 105 HTTP/1.1 200 Connection established
2025-03-14 11:30:12.713676 10.40.40.1 10.10.10.1 TCP 66 48716 → 80 [ACK] Seq=123 Ack=40 Win=64256 Len=0 TSval=995746544 TSecr=1921885012
2025-03-14 11:30:12.752166 10.40.40.1 10.10.10.1 TLSv1.2 583 Client Hello (SNI=tools.cisco.com)
2025-03-14 11:30:12.772238 10.10.10.1 10.40.40.1 TCP 66 80 → 48716 [ACK] Seq=40 Ack=640 Win=262016 Len=0 TSval=1921885092 TSecr=995746582

> Frame 11: 105 bytes on wire (840 bits), 105 bytes captured (840 bits) on interface 0
> Ethernet II, Src: Cisco_9d:b9:ff (4c:71:0d:9d:b9:ff), Dst: VMware_8d:76:9d (00:50:56:8d:76:9d)
> Internet Protocol Version 4, Src: 10.10.10.1, Dst: 10.40.40.1
> Transmission Control Protocol, Src Port: 80, Dst Port: 48716, Seq: 1, Ack: 123, Len: 39
> Hypertext Transfer Protocol
  > HTTP/1.1 200 Connection established\r\n
    Response Version: HTTP/1.1
    Status Code: 200
    [Status Code Description: OK]
    Response Phrase: Connection established
    \r\n
    [Request in frame: 8]
    [Time since request: 0.176633000 seconds]
    [Request URI: tools.cisco.com:443]
    [Full request URI: tools.cisco.com:443]
  
```

已知的問題

代理ACL限制

如果存在許可權問題（例如代理上的訪問清單），可以通過資料包捕獲(tcpdump)進行觀察。以下是失敗情景的簡要說明，以及tcpdump輸出範例：

TCP握手啟動：

使用者端(Firepower)在連線埠80上建立與代理的TCP連線開始。TCP握手(SYN、SYN-ACK、

ACK)成功完成，這表示代理可連線。

tcpdump輸出示例：

```
10:20:58.987654 IP client.54321 > proxy.80: Flags [S], seq 0, win 64240, options [mss 1460], length 0
10:20:58.987700 IP proxy.80 > client.54321: Flags [S.], seq 0, ack 1, win 65160, options [mss 1460], length 0
10:20:58.987734 IP client.54321 > proxy.80: Flags [.] , ack 1, win 64240, length 0
```

HTTP CONNECT請求：

連線後，使用者端會向Proxy傳送HTTP CONNECT要求，要求其建立到tools.cisco.com:443的通道。

示例tcpdump (已解碼HTTP)：

```
CONNECT tools.cisco.com:443 HTTP/1.1
Host: tools.cisco.com:443
User-Agent: curl/8.5.0
Proxy-Connection: Keep-Alive
```

來自代理的錯誤響應：

Proxy不會允許通道通過，而會拒絕要求，很可能是因為不允許此流量的存取清單(ACL)。Proxy回應一個錯誤，例如403 Forbidden或502 Bad Gateway。

顯示錯誤的tcpdump輸出示例：

```
<#root>
HTTP/1.1
403
Forbidden
Content-Type: text/html
Content-Length: 123
Connection: close
```

連線拆卸：

在傳送錯誤消息後，Proxy關閉連線，兩端交換FIN/ACK資料包。

tcpdump輸出示例：

```
10:21:05.000111 IP client.54321 > proxy.80: Flags [F.], seq 1234, ack 5678, length 0
10:21:05.000120 IP proxy.80 > client.54321: Flags [F.], seq 5678, ack 1235, length 0
10:21:05.000125 IP client.54321 > proxy.80: Flags [F.], seq 1235, ack 5679, length 0
```

 提示：在tcpdump中，您可以看到，儘管TCP連線和HTTP CONNECT請求成功，但Proxy還是拒絕了隧道設定。這通常表示Proxy有一個ACL或許可權限制來防止流量通過。

代理下載失敗（超時/傳輸不完整）

在此案例中，FMC成功連線到代理並開始下載檔案，但傳輸超時或無法完成。這通常是因為代理檢查、超時或代理的檔案大小限制。

TCP握手啟動

FMC在連線埠80上發起到代理的TCP連線，且交握成功完成。

tcpdump輸出示例：

```
10:20:58.987654 IP fmc.54321 > proxy.80: Flags [S], seq 0, win 64240, options [mss 1460], length 0
10:20:58.987700 IP proxy.80 > fmc.54321: Flags [S.], seq 0, ack 1, win 65160, options [mss 1460], length 0
10:20:58.987734 IP fmc.54321 > proxy.80: Flags [F.], seq 1, ack 1, win 64240, length 0
```

HTTP CONNECT請求

FMC向代理傳送HTTP CONNECT請求以到達外部目標。

示例tcpdump（已解碼HTTP）：

```
CONNECT tools.cisco.com:443 HTTP/1.1
Host: tools.cisco.com:443
User-Agent: FMC-Agent
Proxy-Connection: Keep-Alive
```

隧道建立和TLS握手

Proxy響應，建立HTTP/1.1 200連線，允許TLS握手開始。

tcpdump輸出示例：

```
<#root>
```

```
HTTP/1.1
```

```
200
```

```
Connection established
```

```
10:20:59.123456 IP fmc.54321 > proxy.80: Flags [P.], length 517 (Client Hello)
10:20:59.123789 IP proxy.80 > fmc.54321: Flags [P.], length 1514 (Server Hello)
```

超時或未完成下載

啟動檔案傳輸後，下載停止或不完成，導致超時。連線保持空閒。

可能的原因包括：

- 代理檢查延遲或篩選。
- 長期傳輸的代理超時。
- Proxy施加的檔案大小限制。

顯示非活動的tcpdump示例：

```
<#root>
```

```
10:21:00.456000 IP fmc.54321 > proxy.80: Flags [P.], length 1440
```

```
# FMC sending data
```

```
# No response from proxy, connection goes idle...
```

```
# After a while, FMC may close the connection or retry.
```

 提示：FMC啟動下載，但由於超時或傳輸不完整（通常由代理過濾或檔案大小限制導致）而無法完成。

代理檔案下載失敗（MTU問題）

在這種情況下，FMC連線到代理並開始下載檔案，但會話由於MTU問題而失敗。這些問題會導致封包分段或捨棄封包，尤其是對於大型檔案或SSL/TLS握手。

TCP握手啟動

FMC發起與代理的TCP握手，該過程成功。

tcpdump輸出示例：

```
10:20:58.987654 IP fmc.54321 > proxy.80: Flags [S], seq 0, win 64240, options [mss 1460], length 0
10:20:58.987700 IP proxy.80 > fmc.54321: Flags [S.], seq 0, ack 1, win 65160, options [mss 1460], length 0
10:20:58.987734 IP fmc.54321 > proxy.80: Flags [.], ack 1, win 64240, length 0
```

HTTP CONNECT請求和隧道建立

FMC會傳送HTTP CONNECT要求，且代理程式會回應，允許建立通道。

示例tcpdump (已解碼HTTP) :

```
CONNECT tools.cisco.com:443 HTTP/1.1
Host: tools.cisco.com:443
User-Agent: FMC-Agent
Proxy-Connection: Keep-Alive
```

TLS握手開始

FMC和tools.cisco.com開始協商SSL/TLS，然後交換初始資料包。

tcpdump輸出示例：

```
<#root>
HTTP/1.1
200
Connection established
10:20:59.123456 IP fmc.54321 > proxy.80: Flags [P.], length 517 (Client Hello)
10:20:59.123789 IP proxy.80 > fmc.54321: Flags [P.], length 1514 (Server Hello)
```

由於MTU而導致的封包分段或捨棄

當FMC或伺服器嘗試傳送大資料包時，MTU問題會導致資料包分段或丟包，從而導致檔案傳輸或TLS協商失敗。

當FMC和Proxy之間 (或Proxy和Internet之間) 的MTU設定錯誤或太小時，通常會發生這種情況。

顯示分段嘗試的tcpdump示例：

```
<#root>
10:21:00.456000 IP fmc.54321 > proxy.80: Flags [P.], length 1440
# Large packet
10:21:00.456123 IP proxy.80 > fmc.54321: Flags [R], seq X, win 0, length 0
# Proxy resets connection due to MTU issue
```

 提示：MTU問題會導致資料包被丟棄或分段，從而中斷TLS握手或導致檔案下載失敗。由於MTU設定不正確而發生SSL檢查或封包分段時，通常會出現這種情況。

在故障場景中，FMC獲得不帶HTTP 200的CONNECT，重新傳輸和FIN確認沒有TLS/資料交換，可能是由於MTU問題或代理/上游問題。

使用curl時，可能會遇到各種指示服務器端問題或驗證錯誤的HTTP響應代碼。以下是最常見的錯誤代碼及其含義的清單：

HTTP代碼	含義	原因
400	錯誤的請求	不正確的請求語法
401	未授權	缺少憑據或憑據不正確
403	禁止	拒絕訪問
404	未找到	找不到資源
500	內部錯誤	伺服器錯誤
502	錯誤的網關	伺服器通訊錯誤
503	服務不可用	伺服器過載或維護
504	網關超時	伺服器之間的超時

參考資料

[思科安全防火牆威脅防禦版本說明，版本7.4.x](#)

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）。