瞭解Finesse的跨來源資源共用(CORS)

目錄

簡介

必要條件

- 需求
- 採用元件

背景資訊

- 什麼是CORS
- CORS的生命週期

CORS與Cisco Finesse配合使用

- 示例示例：利用即時資料小工具分析CORS行為
- 用於CORS連線測試的TAC工具

簡介

本文檔對跨來源資源共用進行了全面的說明，以便在故障排除期間能夠全面瞭解底層流程。

必要條件

需求

思科建議您瞭解以下主題：

- 思科整合客服中心企業版(UCCE)版本12.6.X
- 思科套裝客服中心企業版(PCCE)版本12.6.X
- Cisco Finesse版本12.6.X
- 思科整合情報中心(CUIC)版本12.6.X

採用元件

本文中的資訊係根據以下軟體和硬體版本：

- UCCE版本12.6.2
- Finesse版本12.6.2
- CUIC版本12.6.2

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

什麼是CORS

跨來源資源共用(CORS)是伺服器控制允許哪些網站（域、協定和埠）訪問其資源的一種方式。雖然瀏覽器通常阻止來自不同來源（相同來源策略）的請求，但CORS使伺服器能夠選擇性地放寬此限制。實質上，伺服器使用特殊的HTTP報頭來告訴瀏覽器允許哪些來源、允許哪些型別的請求（如GET、POST等），以及可以包含哪些自定義報頭。這使伺服器能夠決定誰可以訪問其API，以及如何訪問，範圍從完全開放到嚴格限制。CORS的工作方式是讓瀏覽器和伺服器通過這些HTTP報頭進行通訊來管理跨源請求。

CORS使用HTTP標頭啟用受控交叉來源請求。瀏覽器和伺服器通過這些報頭進行通訊，伺服器指定允許的來源、方法和報頭。如果伺服器的響應報頭缺失或無效，瀏覽器會阻止響應，從而實施相同來源策略。對於某些請求，瀏覽器首先向伺服器傳送印前檢查請求，以確保它接受實際的跨源請求。

瀏覽器使用印前檢查請求來檢查伺服器是否允許跨源請求，然後再傳送實際請求。這些印前檢查請求包括HTTP方法和自定義報頭等詳細資訊。然後，啟用CORS的伺服器可以響應，允許或拒絕實際請求。如果伺服器未配置為CORS，則它不會正確響應印前檢查，並且瀏覽器會阻止實際請求，從而保護伺服器免受不需要的跨源訪問。

跨源資源共用(CORS)對於Web安全和功能至關重要。它允許對來自不同來源（域、協定、埠）的資源進行受控訪問，這是必要的，因為瀏覽器會實施通常阻止此類訪問的相同來源策略。

CORS的生命週期

CORS請求包含兩端：提出請求的客戶端和接收請求的伺服器。在客戶端，開發人員編寫JavaScript代碼以將請求傳送到伺服器。伺服器通過設定特定於CORS的報頭來響應請求，以指示允許跨源請求。如果沒有客戶端和伺服器的參與，CORS請求將失敗。

CORS請求中的關鍵角色是客戶端、瀏覽器和伺服器。客戶端需要來自伺服器的某些資料，例如JSON API響應或網頁內容。瀏覽器充當可信中介來驗證客戶端是否可以從伺服器訪問資料。

用戶端:

客戶端是在網站上運行的JavaScript代碼片段，它負責啟動CORS請求

附註：Finesse是一種Web應用程式。它安裝在伺服器上，代理只需使用其Web瀏覽器對其進行訪問，無需在客戶端安裝或維護外掛或其他軟體。如使用Cisco Finesse的CORS in Action示例所示，此架構支援即時資料包告等功能。在這種情況下，Cisco Finesse live data gadget的JavaScript代碼充當客戶端，而Cisco CUIC則充當CORS生命週期內的伺服器。基本上，基於瀏覽器的Finesse客戶端與CUIC伺服器互動以檢索即時資料。

客戶端與使用者：

有時候，客戶端和使用者這兩個詞可以互換使用，但在CORS的上下文中它們不同。使用者是訪問網站的人或在此上下文中訪問Finesse的Finesse使用者（代理或主管），而客戶端是該網站提供的實際代碼。多個使用者可以訪問同一網站，並可使用同一JavaScript客戶端代碼。

瀏覽器：

瀏覽器（也稱為使用者代理）承載客戶端代碼。它通過向傳出請求新增額外資訊，使伺服器能夠識別客戶端，在CORS中扮演著關鍵角色。此外，瀏覽器解釋伺服器的響應，確定是將資料傳送到客戶機還是返回錯誤。這些瀏覽器端操作對於維護相同來源策略提供的安全至關重要。如果瀏覽器不執行CORS規則，客戶端可能會發出未經授權的請求，從而損害這一重要的安全機制。

伺服器:

伺服器是CORS請求的目標，它是Cisco Finesse的CUIC for Live data小工具示例。伺服器儲存客戶端想要的資料，並且它擁有是否允許CORS請求的最終決定權。

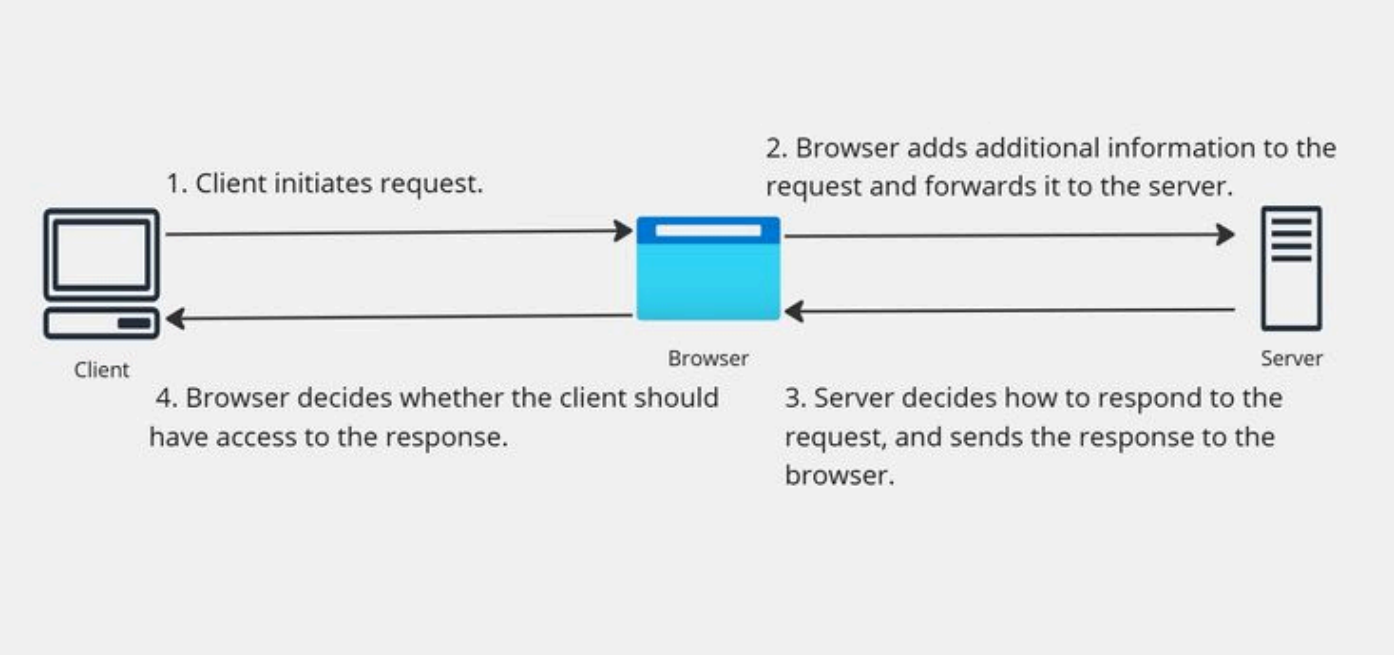
現在您已經知道誰參與了CORS請求，現在讓我們看一看他們如何協同工作。後續影象說明了高級CORS生命週期：

1.客戶端發起請求。

2.瀏覽器向請求新增其他資訊並將其轉發到伺服器。

3.伺服器決定如何響應請求，並將響應傳送到瀏覽器。

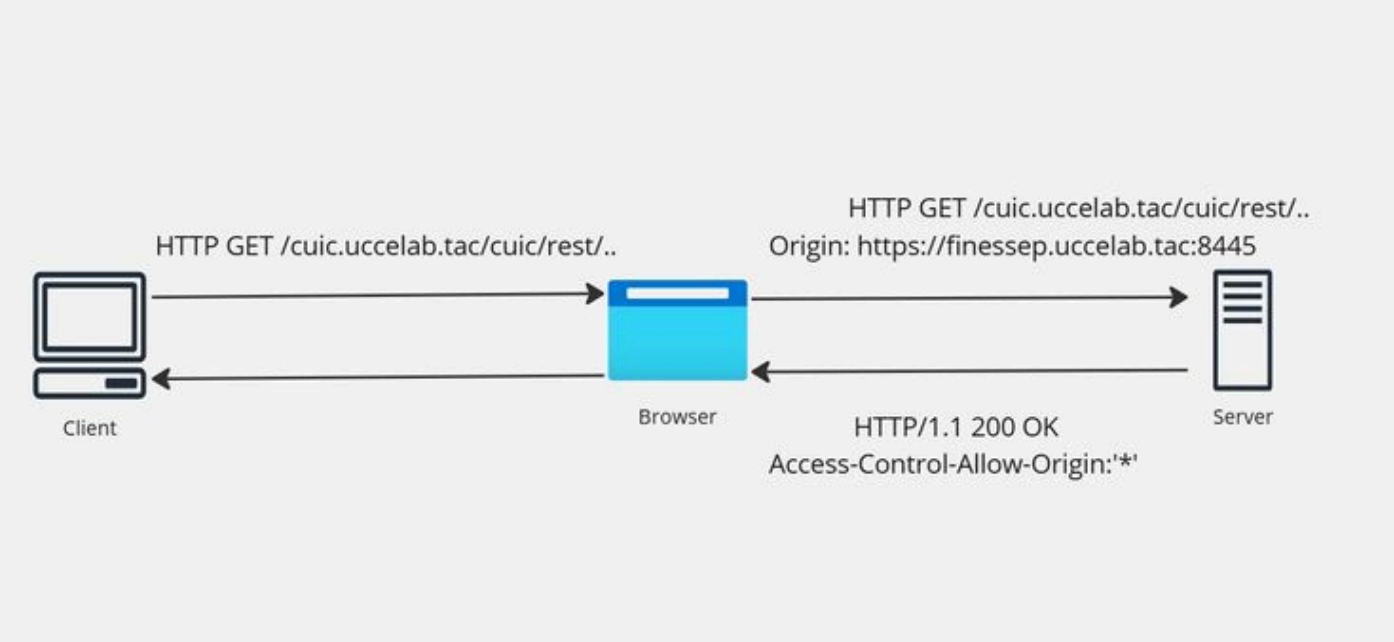
4.瀏覽器確定客戶端是否必須有權訪問響應，並將響應傳遞給客戶端或返回錯誤。



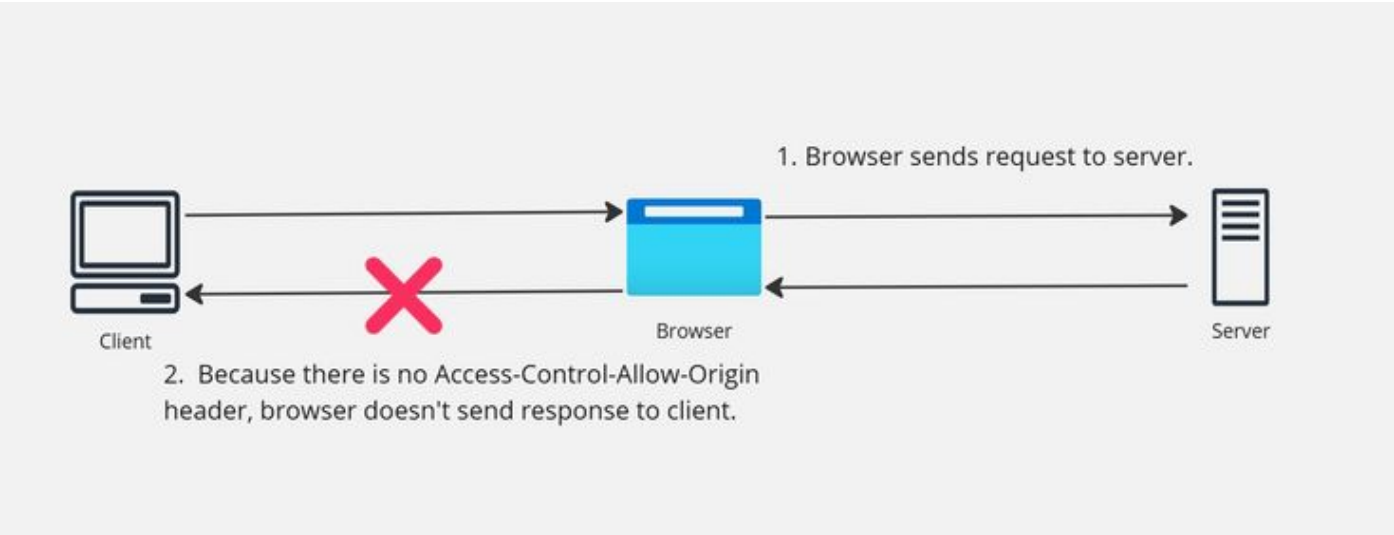
在傳送跨源請求之前，瀏覽器會自動向HTTP請求新增源頭。此報頭（客戶端無法修改）是CORS的重要組成部分，用於標識客戶端的來源（即載入客戶端資源的域、協定和埠）。 此安全措施可防止客戶端模擬其他源。原始報頭對於CORS至關重要，因為它是客戶端向伺服器通知它的來源。

在跨源資源共用(CORS)互動中，客戶端的源由初始請求中的源頭標頭標識。然後，伺服器在其響應中使用Access-Control-Allow-Origin標頭來指示是否允許客戶端訪問請求的資源。此響應報頭至關重要；如果沒有，則CORS請求失敗。Access-Control-Allow-Origin報頭可以包含萬用字元(*)（允許從任何來源訪問），也可以包含特定來源（僅授予該特定客戶端訪問許可權）。當影象顯示Access-Control-Allow-Origin時：*，暗示CUIC允許所有來源，CUIC通常在真實場景中傳送帶有特定來源的

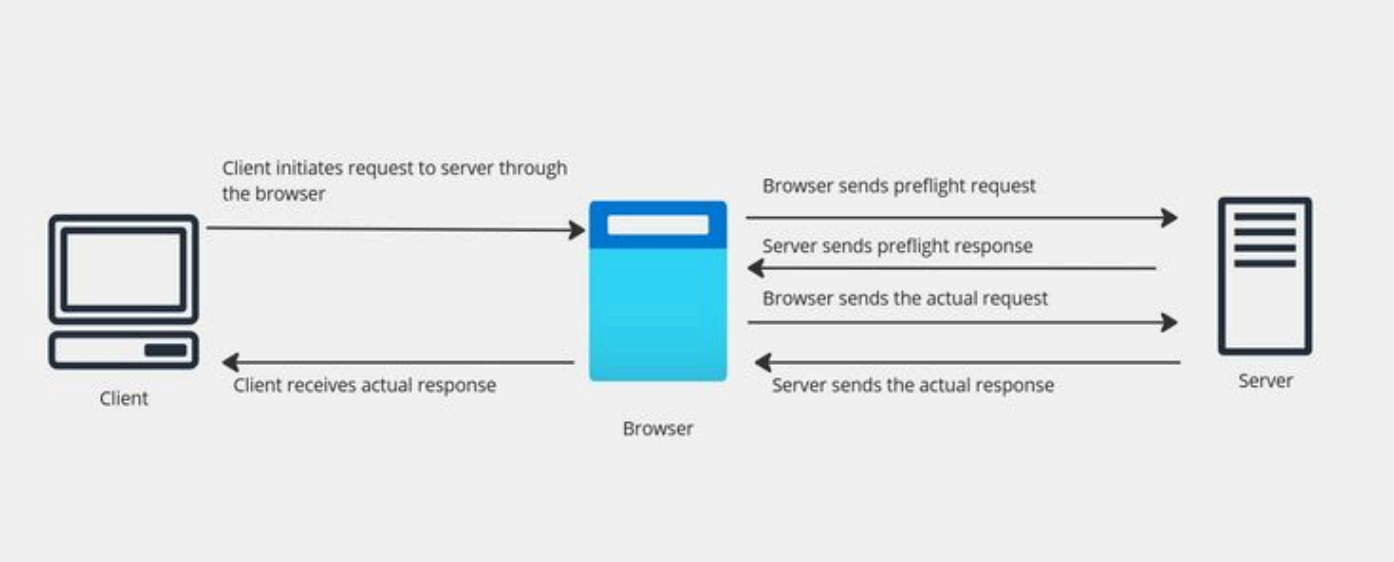
該報頭。



當瀏覽器拒絕CORS請求時，意味著客戶端不會收到有關伺服器響應的資訊。客戶端只知道發生了錯誤，但缺少有關特定問題的詳細資訊。由於很難將CORS故障與其他型別的錯誤區分開來，因此調試會使CORS錯誤變得具有挑戰性。即使將初始請求傳送到伺服器，如果伺服器的響應缺少有效的Access-Control-Allow-Origin標頭，瀏覽器也會阻止該響應並在客戶端觸發錯誤，從而阻止客戶端看到伺服器的詳細響應。



本圖說明了整個CORS流程，特別側重於飛行前階段，該階段對於處理特定型別的跨源請求至關重要。

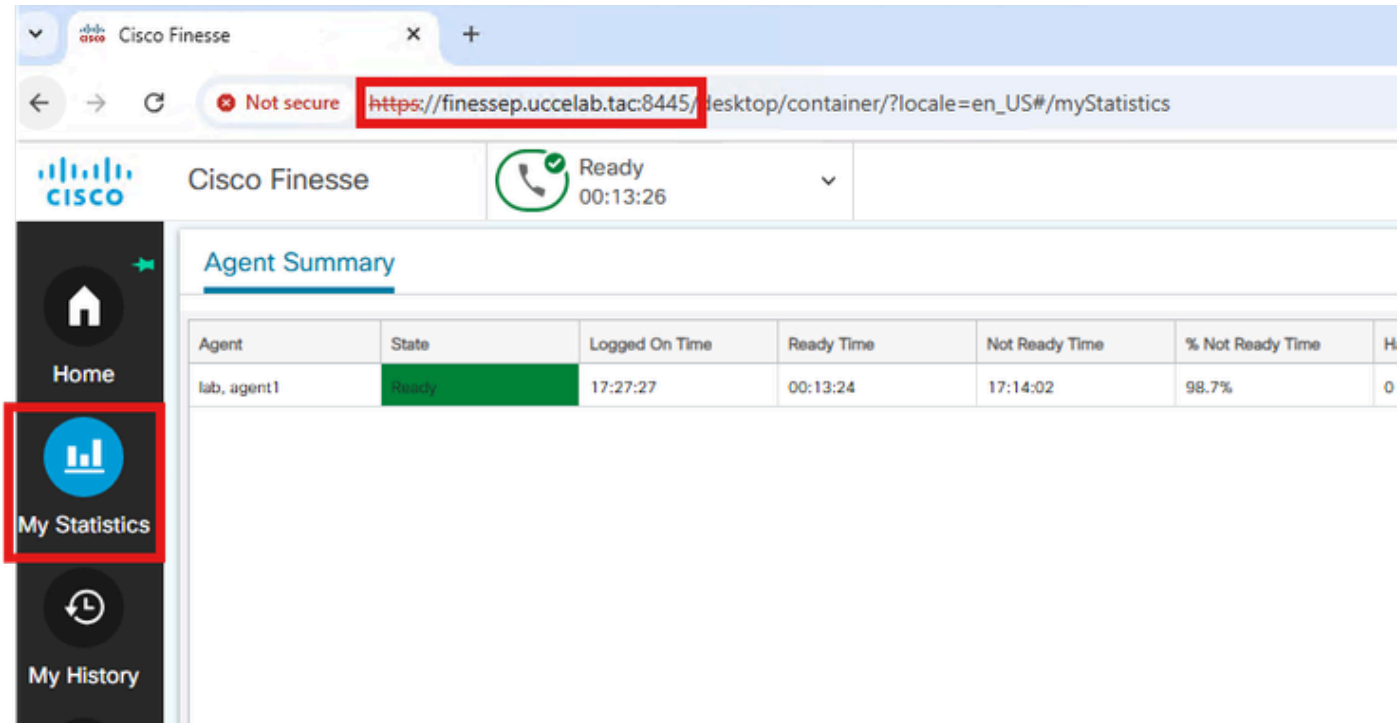


CORS與Cisco Finesse配合使用

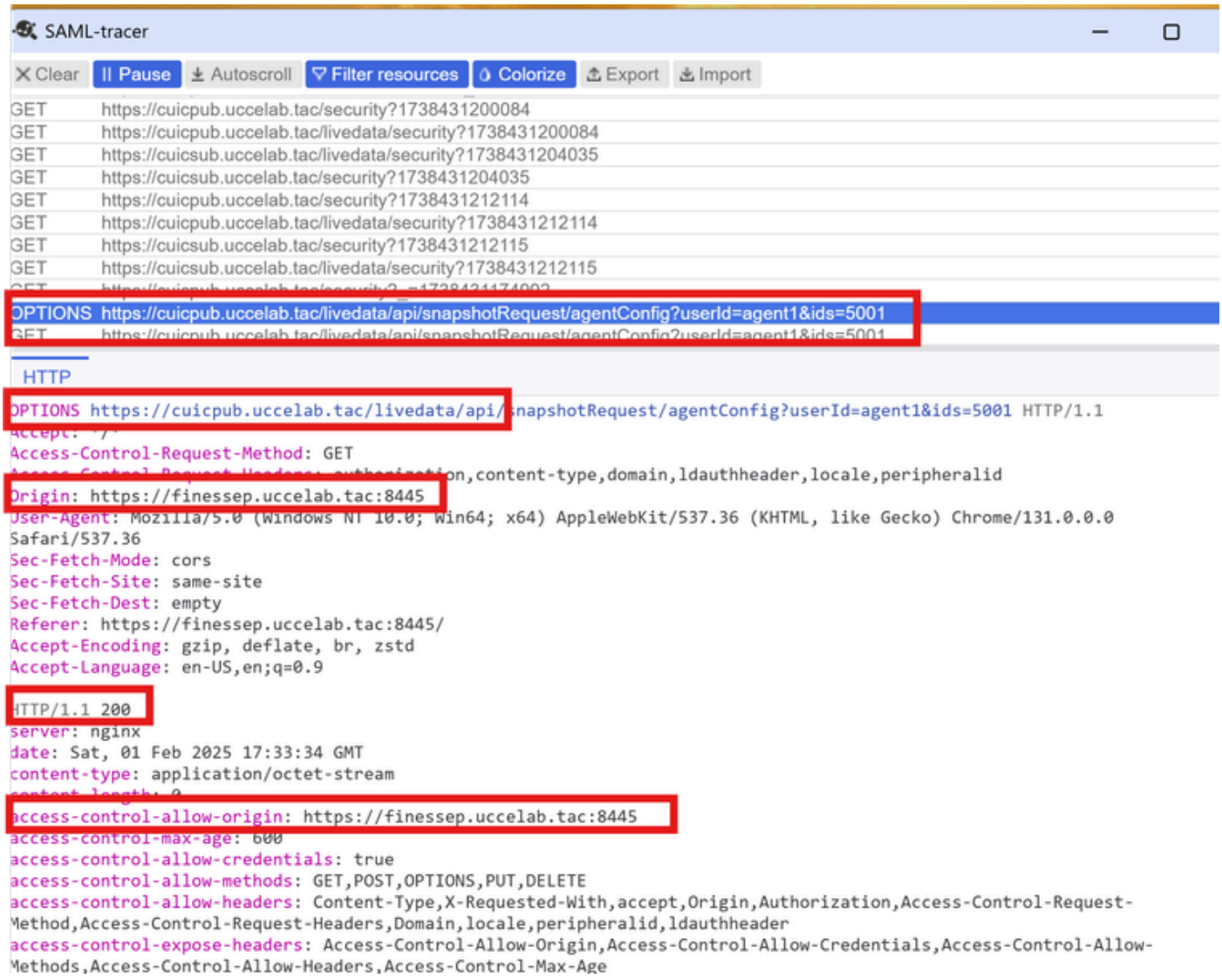
示例示例：利用即時資料小工具分析CORS行為

本節介紹在聯絡中心使用Cisco Finesse的跨來源資源共用(CORS)的典型情況。代理和主管通常使用Cisco Finesse訪問即時資料包告（如示例圖所示）。

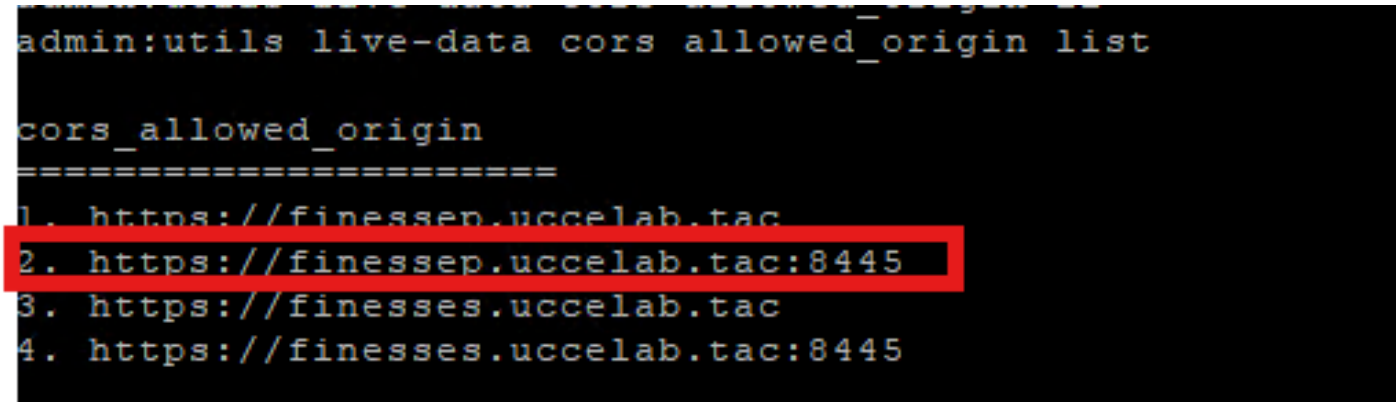
當座席或主管按一下報告小工具時，其操作將啟動資料檢索請求。此請求使用GET方法從Finesse應用程式的JavaScript代碼（充當客戶端）傳送到CUIC/Live Data伺服器。如SAML Tracer影象所示，瀏覽器首先向伺服器傳送印前檢查請求，即前面介紹的CORS生命週期。

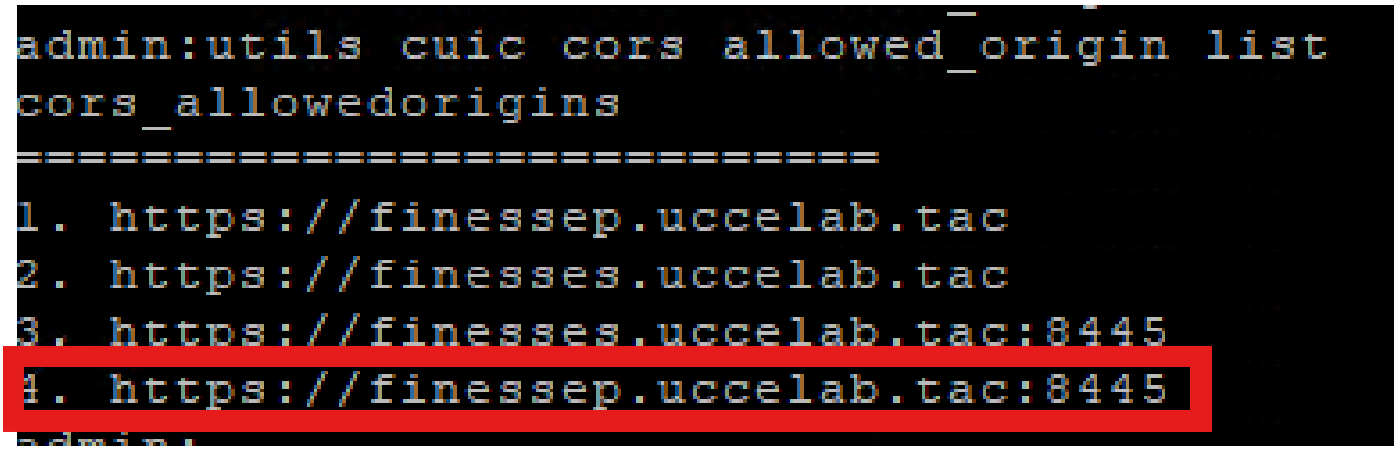


將HTTP OPTIONS請求（印前檢查請求）傳送到CUIC/Live Data伺服器。此請求將源指定為Finesse伺服器的完全限定域名(FQDN)，包括埠8445。這是代理用於訪問Cisco Finesse應用程式的地址和埠。



CUIC/Live Data伺服器上的命令列介面(CLI)命令控制允許其源訪問其即時資料資源。如果在這些設定中配置了Finesse伺服器的源（其FQDN和埠），則代理可以在Finesse中檢視即時資料小工具詳細資訊。





用於CORS連線測試的TAC工具

伺服器端的CORS配置錯誤有時會導致Cisco Finesse中的第三方或即時資料小工具出現問題。這篇文章提供了一個指向CORS快速檢查小工具的連結，該小工具是一個故障排除工具，旨在幫助診斷影響Finesse小工具的跨源資源共用問題，包括即時資料顯示和其他第三方整合。

從技術上講，此小工具的工作方式是從Cisco Finesse客戶端向指定的目標資源傳送印前檢查請求。此快速檢查功能有助於快速識別和解決與CORS相關的問題，從而加快故障排除過程。

要在Finesse案頭中部署Contact Center CORS Quick Check 12.6-v1.0小工具：

1.從Contact Center CORS快速檢查12.6-v1.0 folder.2下載小工具檔案。

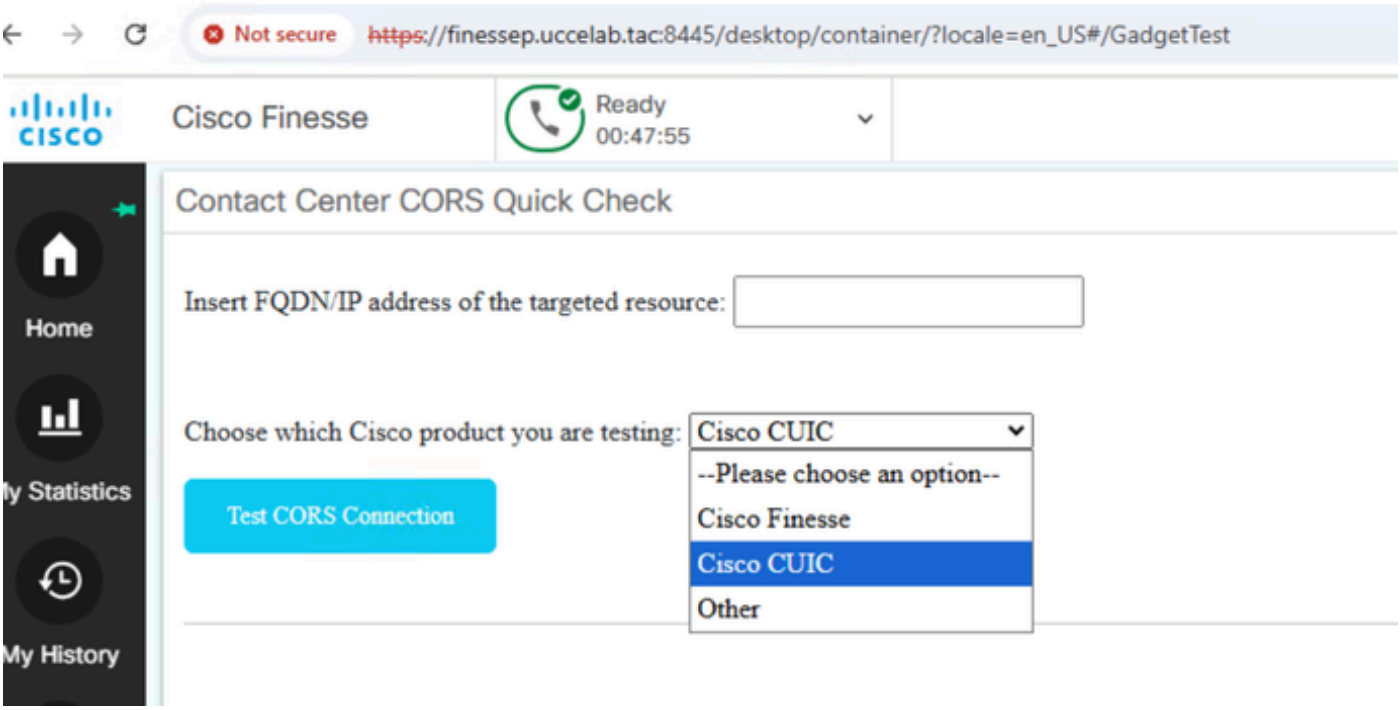
2.將Contact Center CORS Quick Check 12.6-v1.0資料夾的內容複製到Finesse安裝中的3rdpartygadget目錄中。

3.在Finesse案頭佈局中將小工具新增到所需的使用者角色（Agent、Supervisor等）。提供的示例XML演示了新增此小工具的正確配置。

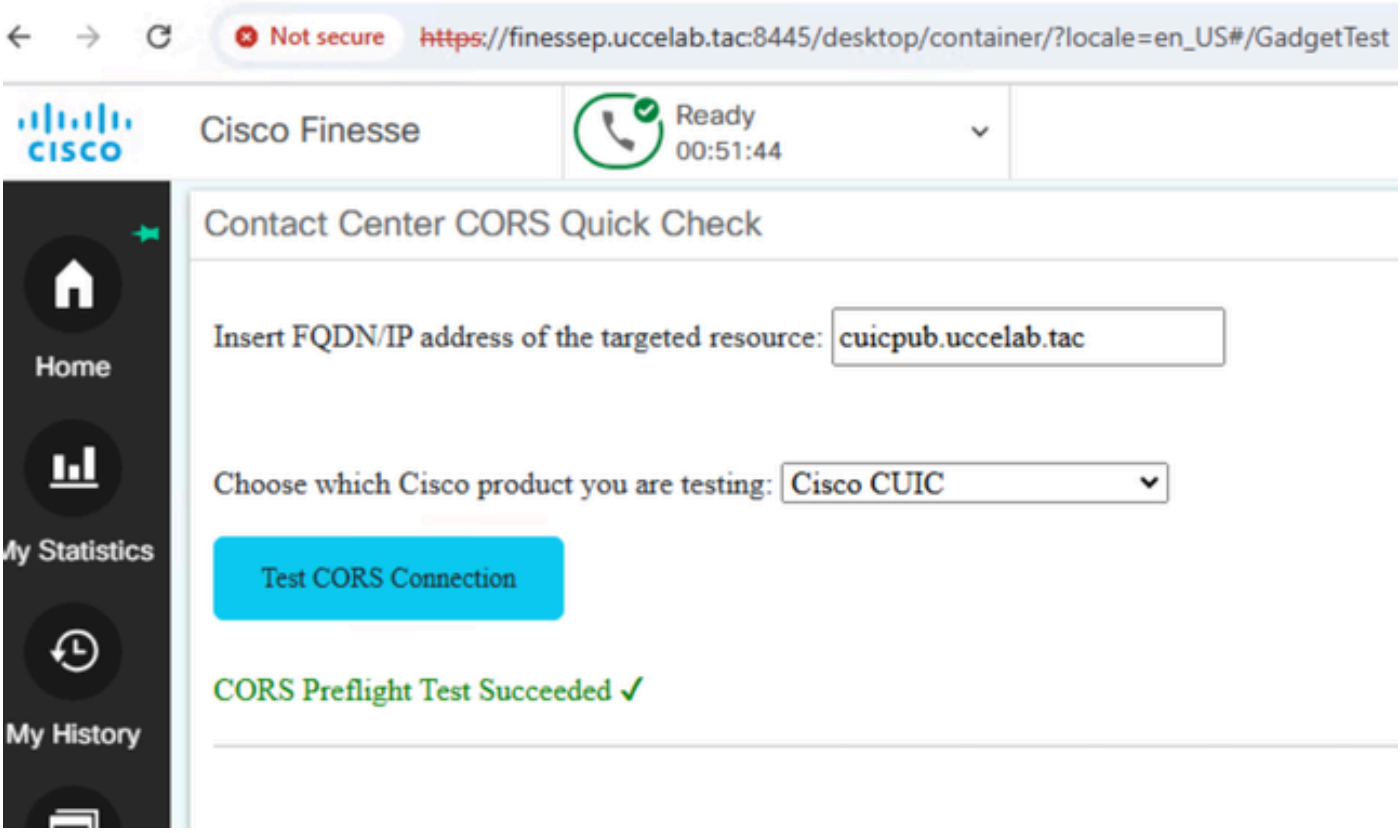
```
<gadget>/3rdpartygadget/files/TestCORSgadget.xml</gadget>
```

有關上傳第三方小工具並將其新增到案頭的詳細資訊，請參閱Finesse Developer指南中的第三方小工具一章和Finesse Administration指南中的管理第三方小工具一章。

上傳小工具檔案並重新啟動Cisco Finesse Tomcat服務後，小工具即可使用並顯示圖形使用者介面(GUI)。



您可以從頂部的下拉選單中選擇CUIC。在提供的欄位中輸入CUIC伺服器的完全限定域名(FQDN)。一個成功的測試將在這裡顯示。



成功的測試意味著CUIC伺服器已正確配置為與Finesse伺服器進行跨源資源共用(CORS)。瀏覽器的SAML Tracer日誌顯示HTTP OPTIONS請求（CORS印前檢查）已傳送到CUIC伺服器。此請求在Origin標頭中包含Finesse伺服器的地址。CUIC伺服器使用200 OK HTTP消息進行響應，重要的是，響應中的Access-Control-Allow-Origin標頭也包含Finesse伺服器的地址。這確認已將CUIC伺服器配置為允許來自Finesse伺服器源的請求，從而驗證CORS是否設定正確。

<#root>

OPTIONS https://cuicpub.uccelab.tac/cuic/ HTTP/1.1

sec-ch-ua-platform: "Windows"
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome..
sec-ch-ua: "Google Chrome";v="131", "Chromium";v="131", "Not_A Brand";v="24"
sec-ch-ua-mobile: ?0
Accept: */*

Origin: https://finessep.uccelab.tac:8445

Sec-Fetch-Site: same-site
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: https://finessep.uccelab.tac:8445/
Accept-Encoding: gzip, deflate, br, zstd
Accept-Language: en-US,en;q=0.9

<#root>

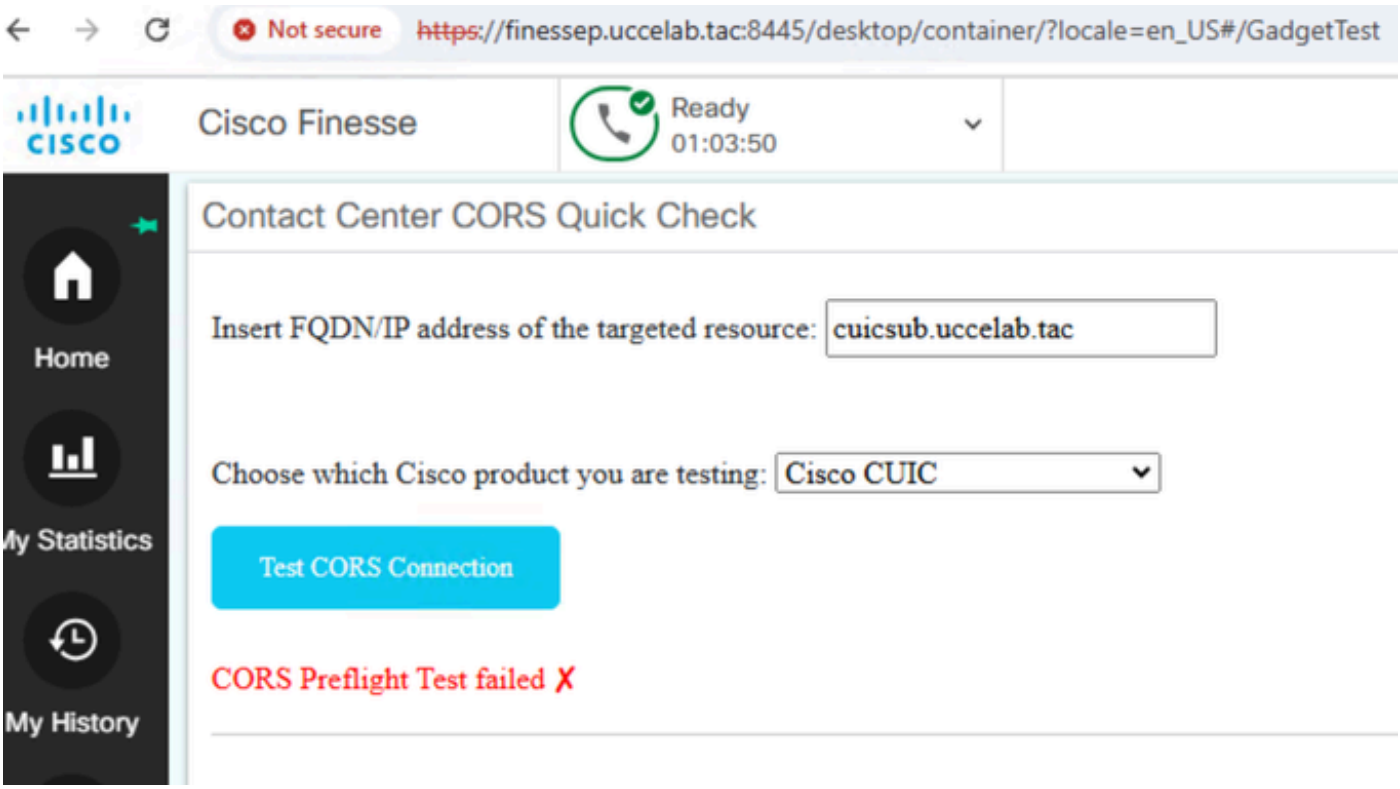
HTTP/1.1 200

server: nginx
date: Sat, 08 Feb 2025 01:27:47 GMT
content-length: 0
strict-transport-security: max-age=31536000; includeSubDomains
set-cookie: JSESSIONID=bE73993C4A7C1Fc1b33A7AaF897B8428; Path=/cuic; Secure; HttpOnly; SameSite=Strict
pragma: No-cache
cache-control: no-cache
expires: Thu, 01 Jan 1970 00:00:00 GMT
x-frame-options: SAMEORIGIN
x-xss-protection: 1; mode=block
x-content-type-options: nosniff
content-security-policy: default-src 'self' ; script-src 'self' data: 'unsafe-inline' 'unsafe-eval' ; s
vary: origin,access-control-request-method,Access-Control-Request-Headers

access-control-allow-origin: https://finessep.uccelab.tac:8445

access-control-allow-credentials: true
access-control-expose-headers: access-control-allow-origin,access-control-allow-credentials,access-cont
access-control-max-age: 600
access-control-allow-methods: DELETE,POST,GET,OPTIONS,PUT
access-control-allow-headers: referer,peripheralid,origin,access-control-request-method,locale,accept,a
allow: GET,POST,OPTIONS,PUT,DELETE

在此案例中，該工具演示了非工作配置。與先前的示例不同，Finesse伺服器未配置為CUIC伺服器上的使用者。相反，它只在CUIC發佈伺服器上配置。因此，CORS印前檢查請求失敗，CUIC伺服器以HTTP 403（已禁止）錯誤進行響應。



<#root>

OPTIONS https://cuicsub.uccelab.tac/cuic/ HTTP/1.1

Accept: */*
Access-Control-Request-Method: OPTIONS

Origin: https://finessep.uccelab.tac:8445

User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome..
Sec-Fetch-Mode: cors
Sec-Fetch-Site: same-site
Sec-Fetch-Dest: empty
Referer: https://finessep.uccelab.tac:8445/
Accept-Encoding: gzip, deflate, br, zstd
Accept-Language: en-US,en;q=0.9

<#root>

HTTP/1.1 403

server: nginx
date: Sat, 08 Feb 2025 01:54:52 GMT
content-type: text/html;charset=utf-8
content-length: 2143
strict-transport-security: max-age=31536000; includeSubDomains
set-cookie: JSESSIONID=1C7606841B83d7847486c3d18D31cEfD; Path=/cuic; Secure; HttpOnly; SameSite=Strict
pragma: No-cache
cache-control: no-cache
expires: Thu, 01 Jan 1970 00:00:00 GMT
x-frame-options: SAMEORIGIN
x-xss-protection: 1; mode=block
x-content-type-options: nosniff

從CUIC subscriber命令列介面(CLI)的輸出中可以看到，Cisco Finesse未列出。這表示Finesse當前未配置為此CUIC伺服器上的訂戶。

<#root>

```
admin:utils cuic cors allowed_origin list

cors_allowedorigins
==========================
1. https://finessep.uccelab.tac
2. https://finesses.uccelab.tac
3. https://finesses.uccelab.tac:8445
```

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）。