Entender o compartilhamento de recursos entre origens (CORS) do Finesse

Contents

Introdução

Este documento descreve o Compartilhamento de recursos entre origens completamente para que, durante a solução de problemas, os processos subjacentes sejam completamente entendidos.

Pré-requisitos

Requisitos

A Cisco recomenda que você tenha conhecimento destes tópicos:

- Cisco Unified Contact Center Enterprise (UCCE) versão 12.6.X
- Cisco Packaged Contact Center Enterprise (PCCE) versão 12.6.X
- Cisco Finesse versão 12.6.X
- Cisco Unified Intelligence Center (CUIC) versão 12.6.X

Componentes Utilizados

As informações neste documento são baseadas nestas versões de software e hardware:

- UCCE versão 12.6.2
- Finesse Versão 12.6.2
- CUIC versão 12.6.2

As informações neste documento foram criadas a partir de dispositivos em um ambiente de laboratório específico. Todos os dispositivos utilizados neste documento foram iniciados com uma

configuração (padrão) inicial. Se a rede estiver ativa, certifique-se de que você entenda o impacto potencial de qualquer comando.

Informações de Apoio

O que é CORS

O Compartilhamento de Recursos entre Origens (CORS) é uma forma dos servidores controlarem quais sites (domínios, protocolos e portas) têm permissão para acessar seus recursos. Embora os navegadores normalmente bloqueiem solicitações de diferentes origens (a política de mesma origem), o CORS dá aos servidores o poder de relaxar seletivamente essa restrição. Essencialmente, os servidores usam cabeçalhos HTTP especiais para informar ao navegador quais origens são permitidas, que tipos de solicitações são permitidas (como GET, POST e assim por diante) e quais cabeçalhos personalizados podem ser incluídos. Isso permite que os servidores decidam quem pode acessar suas APIs e como, variando de totalmente aberto a acesso estritamente limitado. O CORS funciona fazendo com que o navegador e o servidor se comuniquem por meio desses cabeçalhos HTTP para gerenciar solicitações entre origens.

O CORS usa cabeçalhos HTTP para habilitar solicitações entre origens controladas. O navegador e o servidor se comunicam por meio desses cabeçalhos, com o servidor especificando origens, métodos e cabeçalhos permitidos. Se os cabeçalhos de resposta do servidor estiverem ausentes ou forem inválidos, o navegador bloqueará a resposta, impondo a política de mesma origem. Para determinadas solicitações, o navegador envia primeiro uma solicitação de comprovação ao servidor para garantir que ele aceite a solicitação de origem cruzada real.

Os navegadores usam solicitações de comprovação para verificar se um servidor permite uma solicitação entre origens antes de enviar a solicitação real. Essas solicitações de comprovação incluem detalhes como o método HTTP e cabeçalhos personalizados. Os servidores habilitados para CORS podem então responder, permitindo ou negando a solicitação real. Se um servidor não estiver configurado para CORS, ele não responderá corretamente à comprovação e o navegador bloqueará a solicitação real, protegendo o servidor contra o acesso indesejado entre origens.

O compartilhamento de recursos entre origens (CORS) é crucial para a segurança e a funcionalidade da Web. Ele permite acesso controlado a recursos de diferentes origens (domínios, protocolos, portas), o que é necessário porque os navegadores aplicam uma política de mesma origem que normalmente bloqueia tal acesso.

Ciclo de vida de um CORS

Uma solicitação CORS consiste em dois lados: o cliente que faz a solicitação e o servidor que recebe a solicitação. No lado do cliente, o desenvolvedor grava o código JavaScript para enviar a solicitação ao servidor. O servidor responde à solicitação definindo cabeçalhos específicos de CORS para indicar que a solicitação entre origens é permitida. Sem a participação do cliente e do servidor, a solicitação CORS falha.

Os principais participantes de uma solicitação CORS são o cliente, o navegador e o servidor. O

cliente deseja alguns dados do servidor, como uma resposta JSON API ou o conteúdo de uma página da Web. O navegador atua como o intermediário confiável para verificar se o cliente pode acessar os dados do servidor.

Cliente:

O cliente é um trecho de código JavaScript em execução em um site e é responsável por iniciar a solicitação CORS

Note: O Finesse é um aplicativo da Web. Ele é instalado em um servidor, e os agentes acessam-no simplesmente usando seus navegadores da Web, eliminando a necessidade de instalações no lado do cliente ou manutenção de plug-ins ou outros softwares. Como demonstrado no exemplo do CORS em ação com o Cisco Finesse, essa arquitetura oferece suporte a recursos como relatórios de dados dinâmicos. Neste contexto, o código JavaScript do gadget de dados ao vivo do Cisco Finesse atua como o cliente, enquanto o Cisco CUIC serve como o servidor dentro do ciclo de vida do CORS. Essencialmente, o cliente Finesse baseado em navegador interage com o servidor CUIC para recuperar dados dinâmicos.

Cliente versus usuário:

Às vezes, as palavras cliente e usuário são usadas de forma intercambiável, mas são diferentes no contexto de CORS. Um usuário é uma pessoa que visita um site ou usuário do Finesse (agente ou supervisor) acessando o Finesse neste contexto, enquanto um cliente é o código real atendido por esse site. Vários usuários podem visitar o mesmo site e receber o mesmo código de cliente JavaScript.

Navegador:

O navegador, também conhecido como agente de usuário, hospeda o código do lado do cliente. Ele desempenha um papel crucial no CORS, adicionando informações extras às solicitações de saída, permitindo que o servidor identifique o cliente. Além disso, o navegador interpreta a resposta do servidor, determinando se deve entregar os dados ao cliente ou retornar um erro. Essas ações no navegador são essenciais para manter a segurança fornecida pela política de mesma origem. Sem a aplicação das regras CORS pelo navegador, os clientes podem fazer solicitações não autorizadas, comprometendo esse mecanismo de segurança vital.

Servidor:

O servidor é o destino da solicitação CORS e é o CUIC para o exemplo de gadget de dados ao vivo com o Cisco Finesse. O servidor armazena os dados que o cliente deseja e tem a última palavra sobre se a solicitação CORS é permitida ou não.

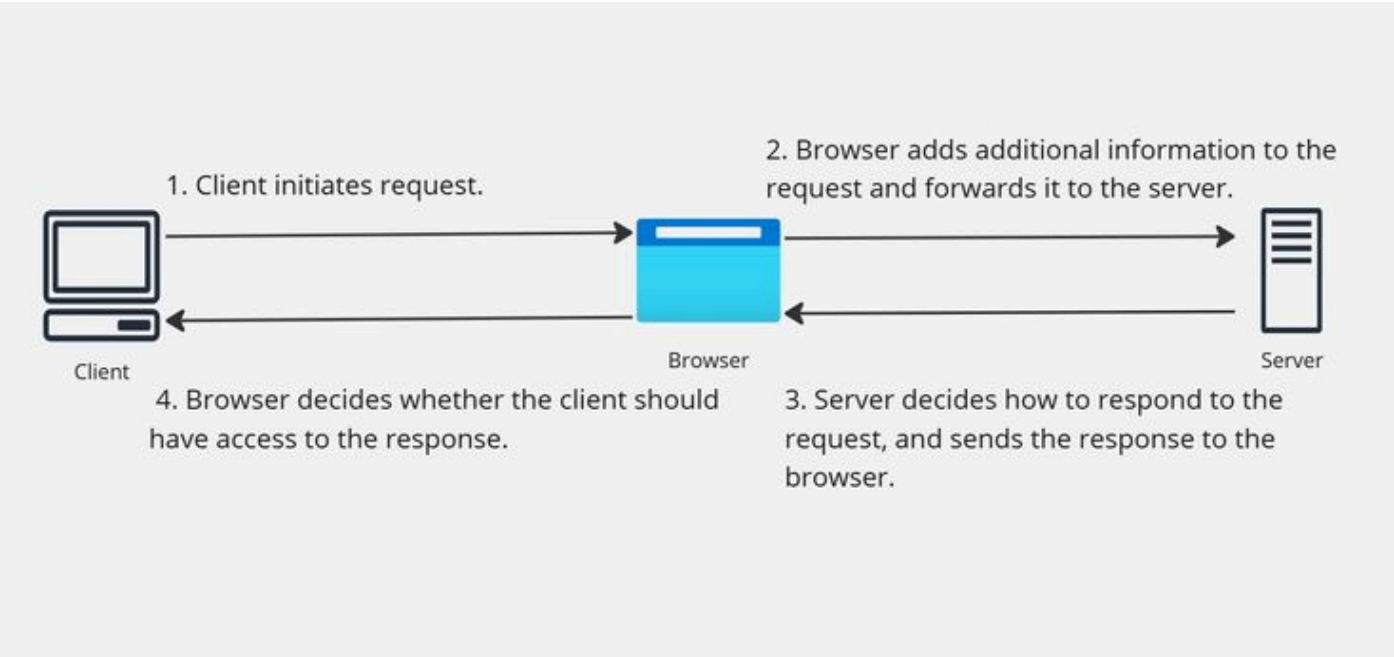
Agora que você sabe quem está envolvido em uma solicitação CORS, vamos ver como todos eles trabalham juntos. As imagens subsequentes ilustram o ciclo de vida do CORS de alto nível:

1. O cliente inicia a solicitação.

2. O navegador adiciona informações adicionais à solicitação e as encaminha ao servidor.

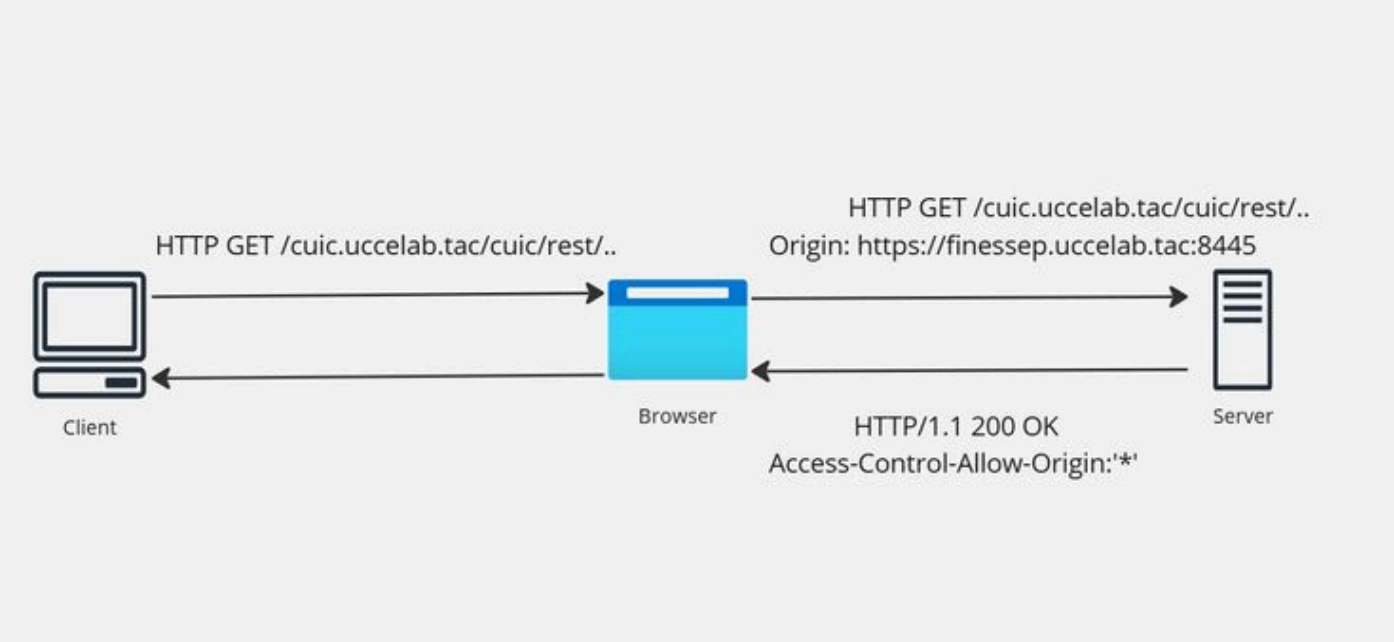
3. O servidor decide como responder à solicitação e envia a resposta ao navegador.

4. O navegador decide se o cliente deve ter acesso à resposta e passa a resposta para o cliente ou retorna um erro.

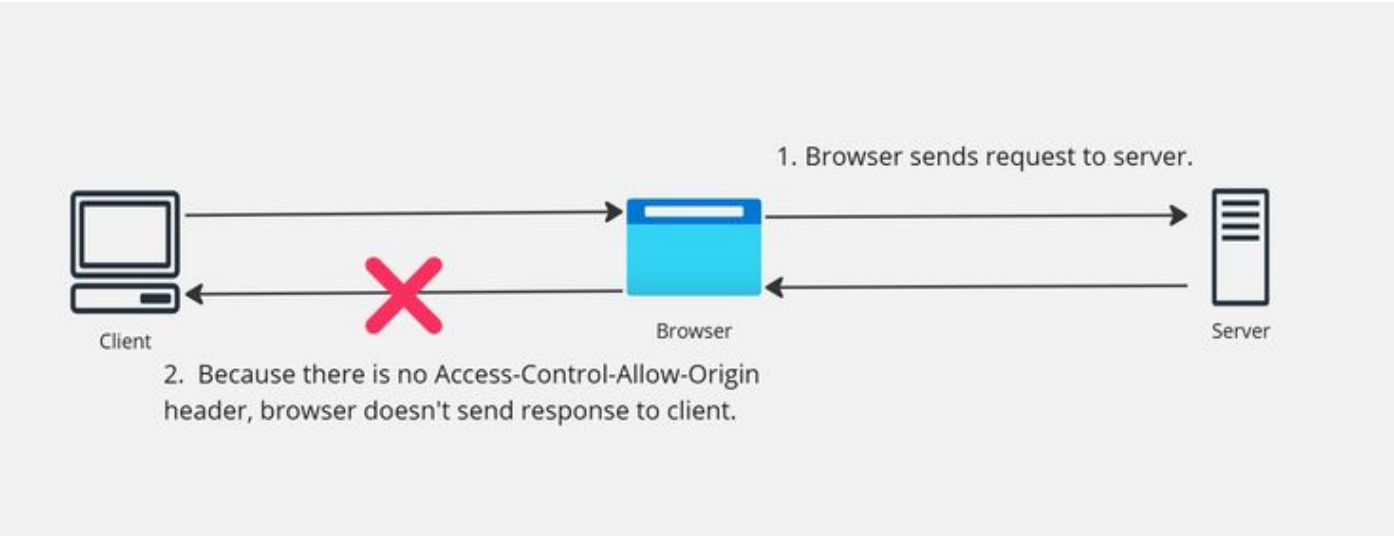


Antes de enviar uma solicitação entre origens, o navegador adiciona automaticamente um cabeçalho de origem à solicitação HTTP. Esse cabeçalho, que o cliente não pode modificar, é uma parte crucial do CORS e serve para identificar a origem do cliente (ou seja, o domínio, o protocolo e a porta da qual o recurso do cliente foi carregado). Essa medida de segurança impede que os clientes representem outras origens. O cabeçalho de origem é fundamental para o CORS, pois é como o cliente informa ao servidor de onde ele vem.

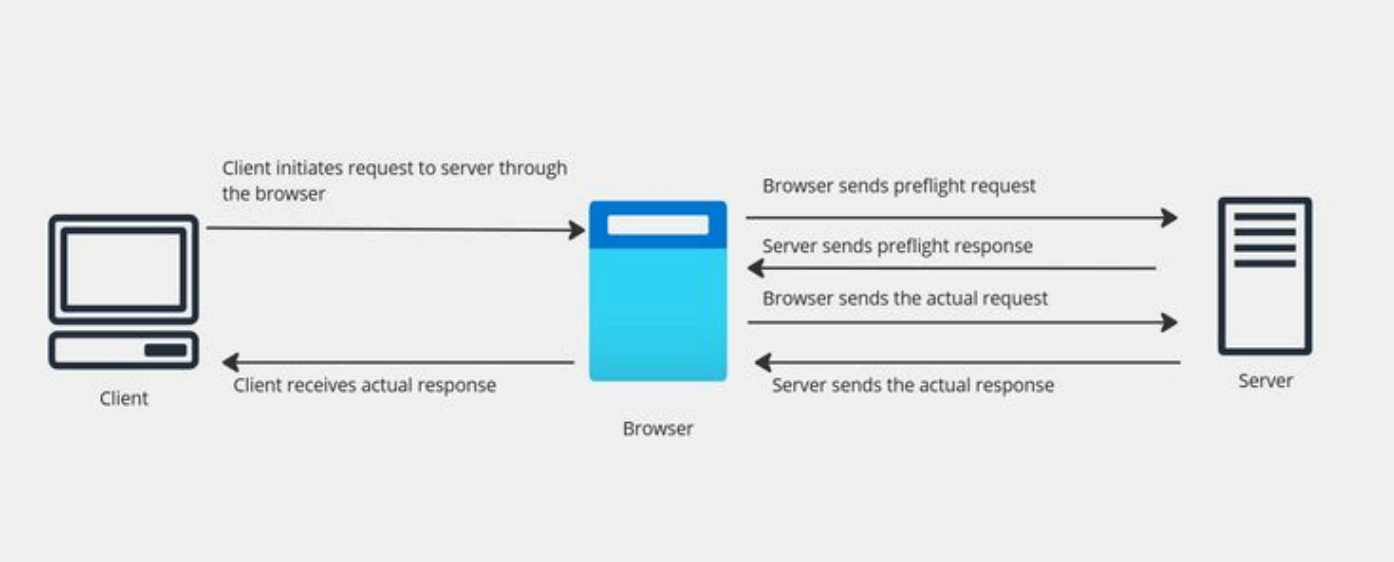
Em uma interação de Compartilhamento de Recursos entre Origens (CORS), a origem do cliente é identificada pelo cabeçalho de Origem na solicitação inicial. Em seguida, o servidor usa o cabeçalho Access-Control-Allow-Origin em sua resposta para indicar se o cliente tem permissão para acessar o recurso solicitado. Este cabeçalho de resposta é crucial; na ausência desta, a solicitação do CORS falha. O cabeçalho Access-Control-Allow-Origin pode conter um curinga (*), permitindo acesso de qualquer origem, ou uma origem específica, concedendo acesso somente a esse cliente específico. Enquanto a imagem mostra Access-Control-Allow-Origin: *, implicando que o CUIC permite todas as origens, o CUIC normalmente envia esse cabeçalho com uma origem específica em cenários reais.



Quando um navegador rejeita uma solicitação CORS, isso significa que o cliente não recebe informações sobre a resposta do servidor. O cliente sabe apenas que ocorreu um erro, mas não tem detalhes sobre o problema específico. Isso pode tornar a depuração de erros CORS desafiadora, pois é difícil distinguir uma falha CORS de outros tipos de erros. Embora a solicitação inicial seja enviada ao servidor, se a resposta do servidor não tiver um cabeçalho Access-Control-Allow-Origin válido, o navegador bloqueia a resposta e aciona um erro no lado do cliente, impedindo que o cliente veja a resposta detalhada do servidor.



Esta imagem explica todo o processo CORS, com um foco particular na etapa de comprovação, que é essencial para lidar com tipos específicos de solicitações de origem cruzada.

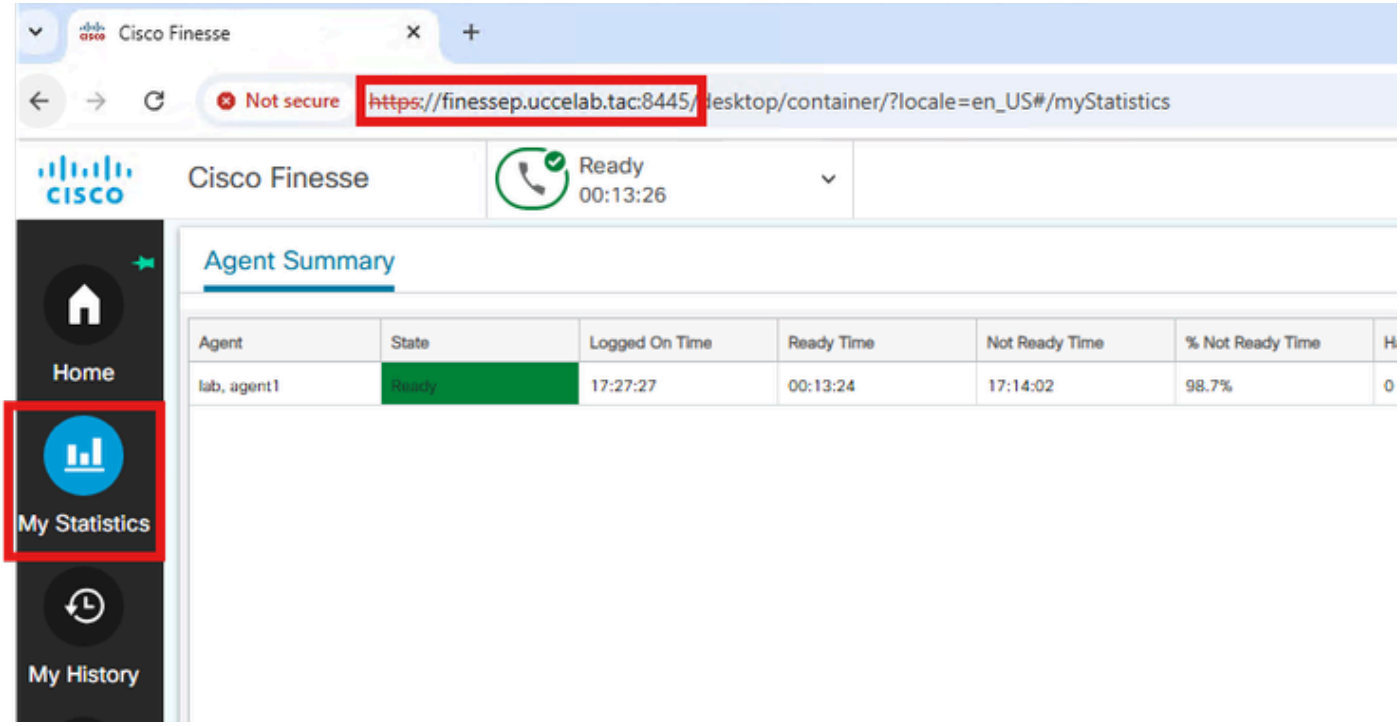


CORS em ação com o Cisco Finesse

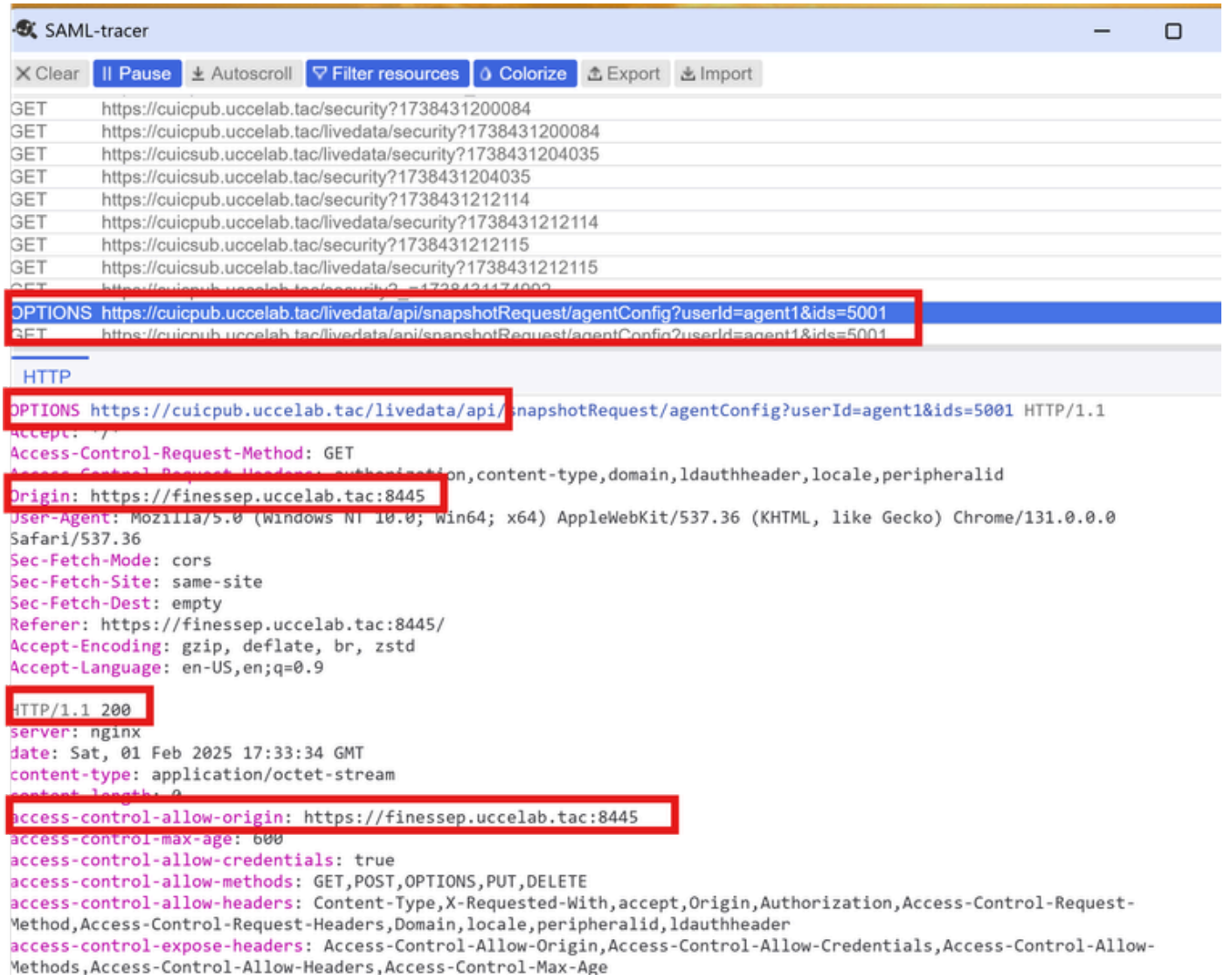
Exemplo ilustrativo: Analisando o comportamento do CORS com o gadget de dados dinâmicos

Esta seção descreve o uso típico do Compartilhamento de Recursos entre Origens (CORS) com o Cisco Finesse em centrais de contato. Os agentes e supervisores geralmente usam o Cisco Finesse para acessar relatórios de dados em tempo real (como ilustrado na imagem de exemplo).

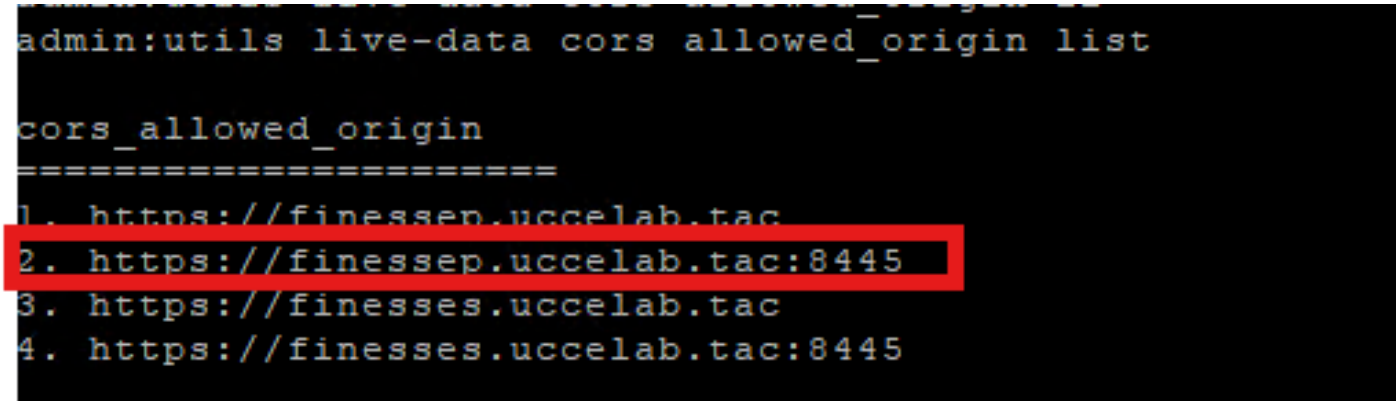
Quando um agente ou supervisor clica em um gadget de relatório, sua ação inicia uma solicitação de recuperação de dados. Essa solicitação é enviada do código JavaScript do aplicativo Finesse (atuando como cliente) para o servidor CUIC/Live Data usando um método GET. Como demonstrado na imagem do rastreador SAML, o navegador envia primeiro uma solicitação de comprovação ao servidor, o ciclo de vida do CORS descrito anteriormente.

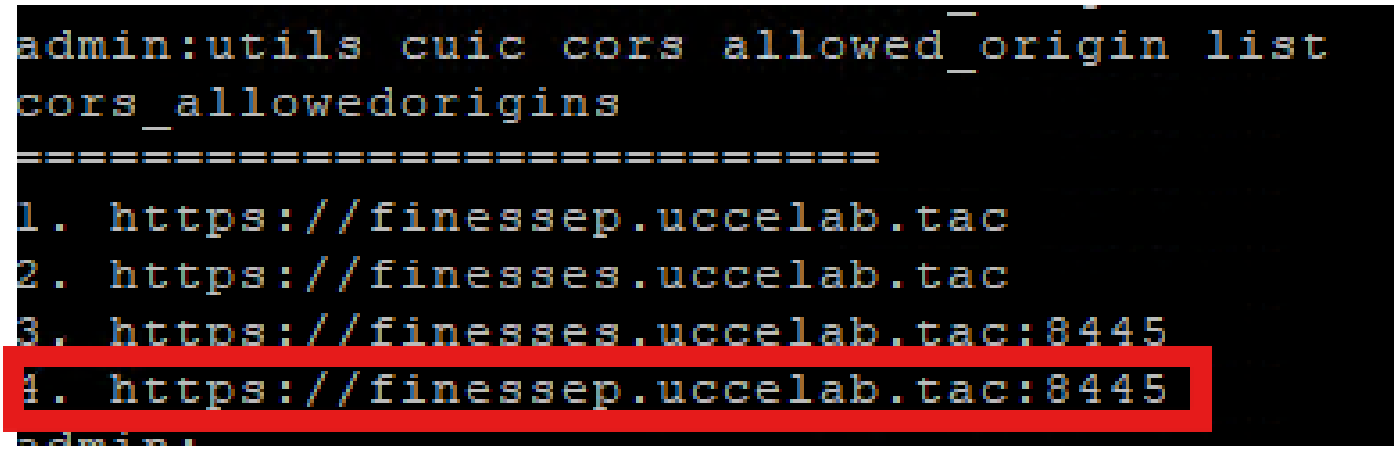


Uma solicitação HTTP OPTIONS (a solicitação de comprovação) é enviada ao servidor CUIC/Live Data. Essa solicitação especifica a origem como o nome de domínio totalmente qualificado (FQDN) do servidor Finesse, incluindo a porta 8445. Esse é o mesmo endereço e porta que os agentes usam para acessar o aplicativo Cisco Finesse.



Os comandos da interface de linha de comando (CLI) no servidor CUIC/Live Data controlam quais origens têm permissão para acessar seus recursos de dados dinâmicos. Se a origem do servidor Finesse (seu FQDN e porta) estiver configurada nessas configurações, os agentes poderão exibir os detalhes do gadget de dados dinâmicos no Finesse.





Ferramenta TAC para teste de conexão CORS

Configurações incorretas de CORS no lado do servidor podem, às vezes, causar problemas com gadgets de dados ao vivo ou de terceiros no Cisco Finesse. Este artigo fornece um link para um gadget de Verificação rápida do CORS, uma ferramenta de solução de problemas é projetada para ajudar a diagnosticar problemas de Compartilhamento de recursos entre origens que afetam gadgets Finesse, incluindo exibições de dados ao vivo e outras integrações de terceiros.

Tecnicamente, esse gadget funciona enviando solicitações de comprovação do cliente Cisco Finesse para um recurso de destino especificado. Essa funcionalidade de verificação rápida ajuda a identificar e resolver rapidamente problemas relacionados ao CORS, acelerando o processo de solução de problemas.

Para implantar o gadget Verificação rápida do CORS 12.6-v1.0 do Contact Center na área de trabalho do Finesse:

1. Baixe arquivos de gadgets da pasta Contact Center CORS Quick Check 12.6-v1.0.2.

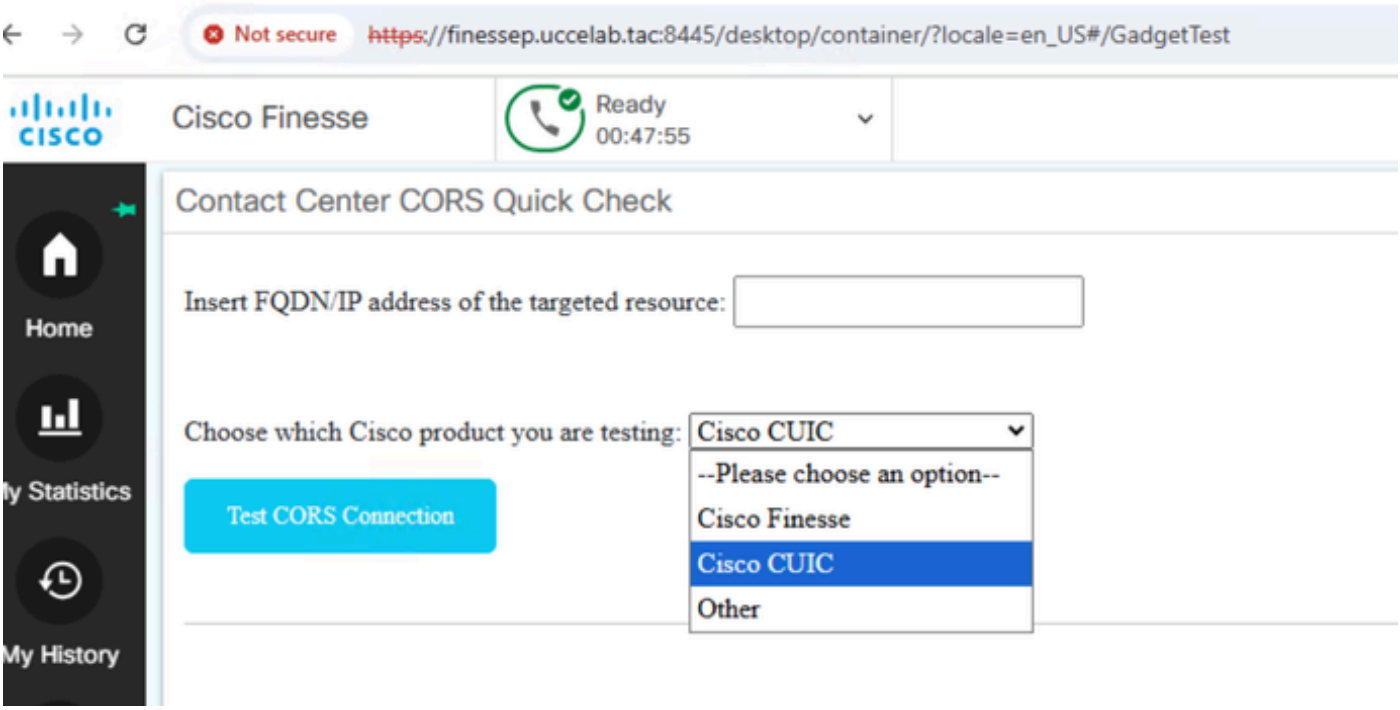
2. Copie o conteúdo da pasta Contact Center CORS Quick Check 12.6-v1.0 para o diretório 3rdpartygadget na instalação do Finesse.

3. Adicione o gadget à função de usuário desejada (Agente, Supervisor e assim por diante) no layout da área de trabalho do Finesse. O exemplo XML fornecido demonstra a configuração correta para adicionar esse gadget.

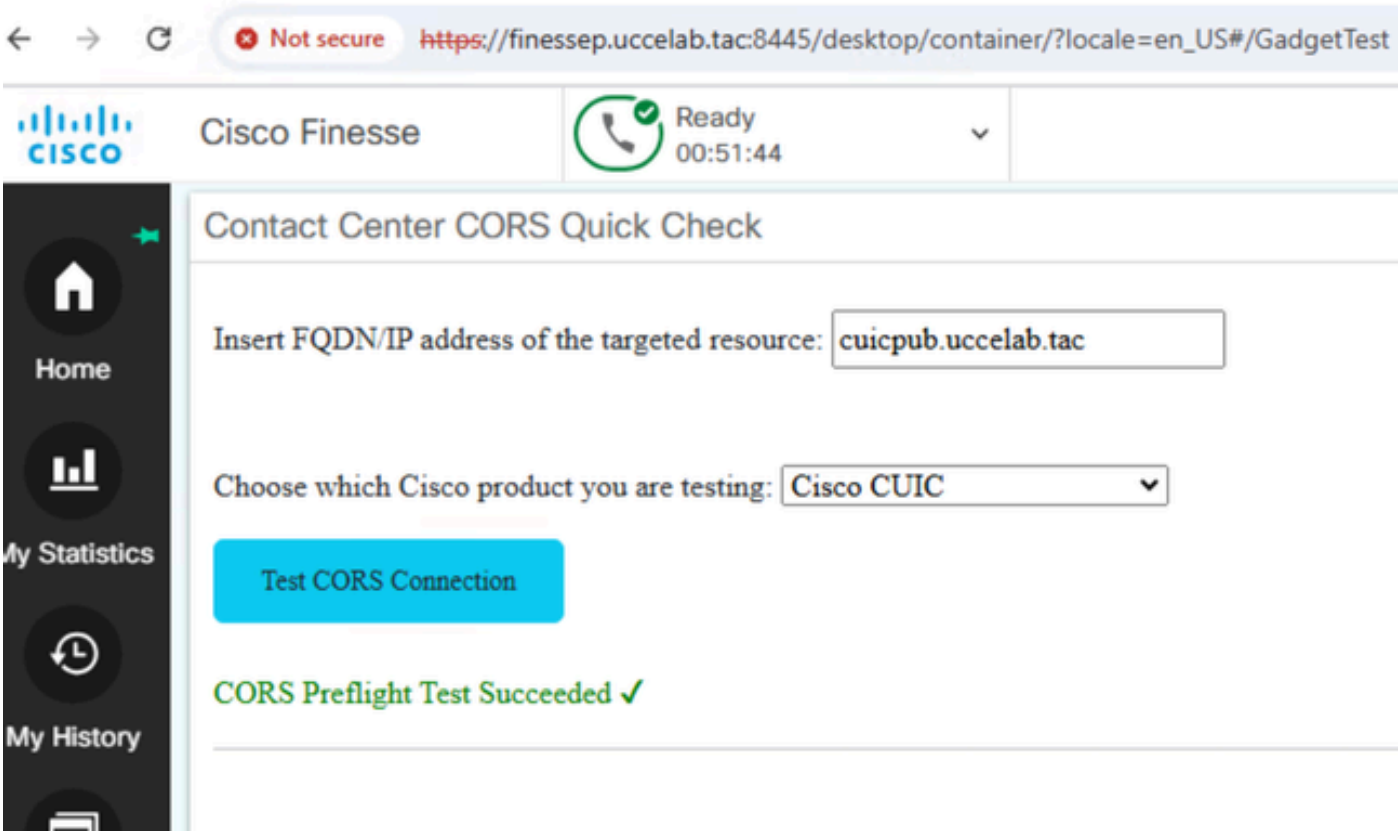
```
<gadget>/3rdpartygadget/files/TestCORSgadget.xml</gadget>
```

Consulte o capítulo Third Party Gadgets no Finesse Developer Guidee o capítulo Manage Third-Party Gadgets no Finesse Administration Guidepara obter mais informações sobre como carregar gadgets de terceiros e adicioná-los à área de trabalho.

Quando os arquivos de gadget são carregados e o serviço Cisco Finesse Tomcat é reiniciado, o gadget fica disponível e exibe a interface gráfica do usuário (GUI).



Você pode selecionar CUIC na lista suspensa superior na parte superior. Insira o nome de domínio totalmente qualificado (FQDN) do servidor CUIC no campo fornecido. Um teste bem-sucedido será realizado conforme mostrado aqui.



Um teste bem-sucedido significa que o servidor CUIC está configurado corretamente para o compartilhamento de recursos entre origens (CORS) com o servidor Finesse. Os logs do rastreador SAML do navegador mostram que uma solicitação HTTP OPTIONS (a comprovação CORS) foi enviada ao servidor CUIC. Essa solicitação incluiu o endereço do servidor Finesse no

cabeçalho Origin. O servidor CUIC respondeu com uma mensagem HTTP de 200 OK e, o mais importante, o cabeçalho Access-Control-Allow-Origin na resposta também continha o endereço do servidor Finesse. Isso confirma que o servidor CUIC está configurado para permitir solicitações da origem do servidor Finesse, verificando se o CORS está configurado corretamente.

<#root>

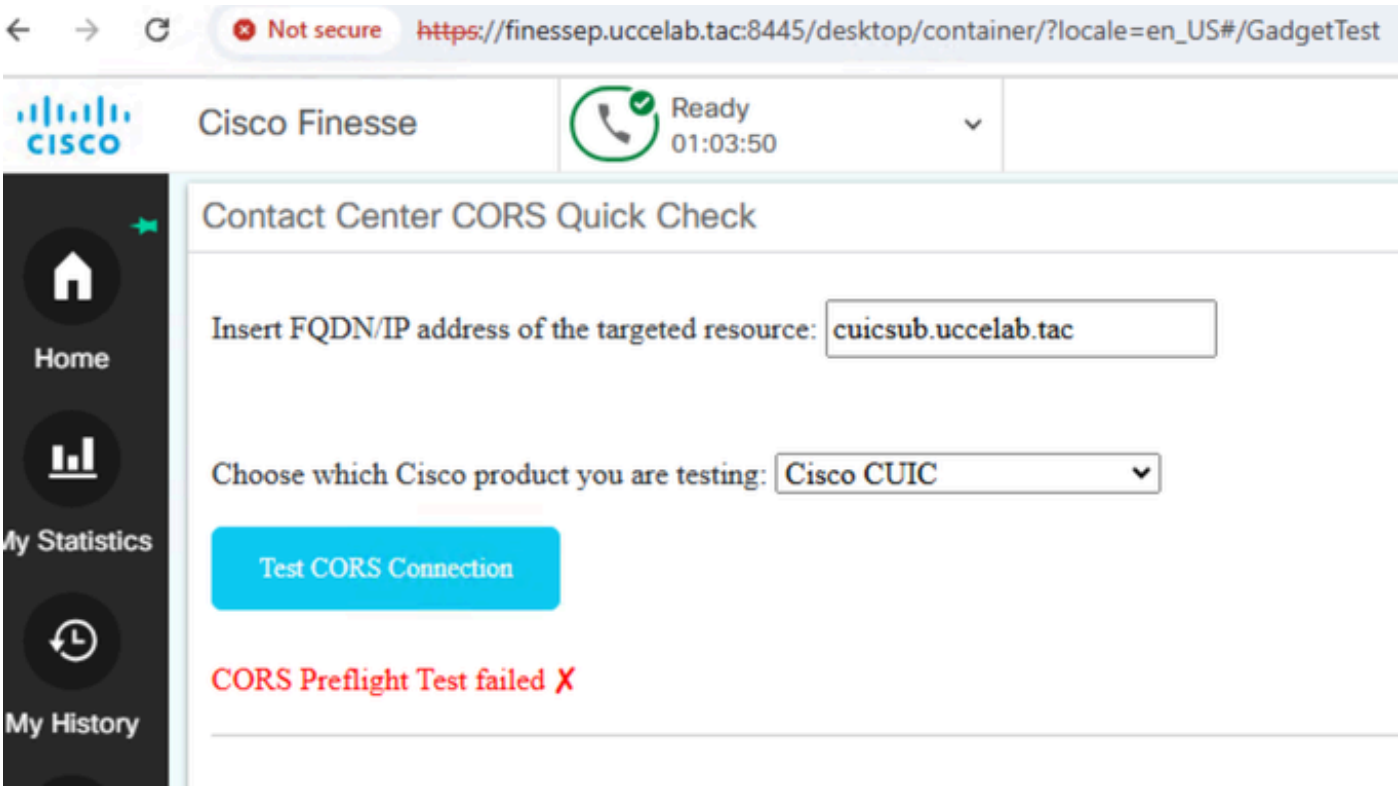
```
OPTIONS https://cuicpub.uccelab.tac/cuic/ HTTP/1.1

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
```

Neste cenário, a ferramenta demonstra uma configuração inoperante. Ao contrário do exemplo anterior, o servidor Finesse não está configurado como um assinante no servidor CUIC. Em vez disso, ele é configurado somente no editor CUIC. Como resultado, a solicitação de comprovação

do CORS falha e o servidor CUIC responde com um erro HTTP 403 (proibido).



<#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
```

Como você pode ver na saída da interface de linha de comando (CLI) do assinante do CUIC, o Cisco Finesse não está listado. Isso indica que o Finesse não está configurado atualmente como um assinante neste servidor CUIC.

<#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
```

Sobre esta tradução
A Cisco traduziu este documento com a ajuda de tecnologias de tradução automática e humana para oferecer conteúdo de suporte aos seus usuários no seu próprio idioma, independentemente da localização.

Observe que mesmo a melhor tradução automática não será tão precisa quanto as realizadas por um tradutor profissional.

A Cisco Systems, Inc. não se responsabiliza pela precisão destas traduções e recomenda que o documento original em inglês (link fornecido) seja sempre consultado.