Entender e solucionar problemas da implementação do Finesse BOSH

Contents

Introdução

Este documento descreve a arquitetura por trás das conexões Finesse que usam BOSH e como os problemas de conexão BOSH podem ser diagnosticados.

Pré-requisitos

Requisitos

A Cisco recomenda que você tenha conhecimento destes tópicos:

- Cisco Finesse

- Unified Contact Center Enterprise (UCCE)
- Unified Contact Center Express (UCCX)
- Ferramentas para desenvolvedores de navegadores da Web
- Administração de Windows e/ou Mac

Componentes Utilizados

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

- Cisco Finesse 9.0(1) - 11.6(1)
- UCCX 10.0(1) - 11.6(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

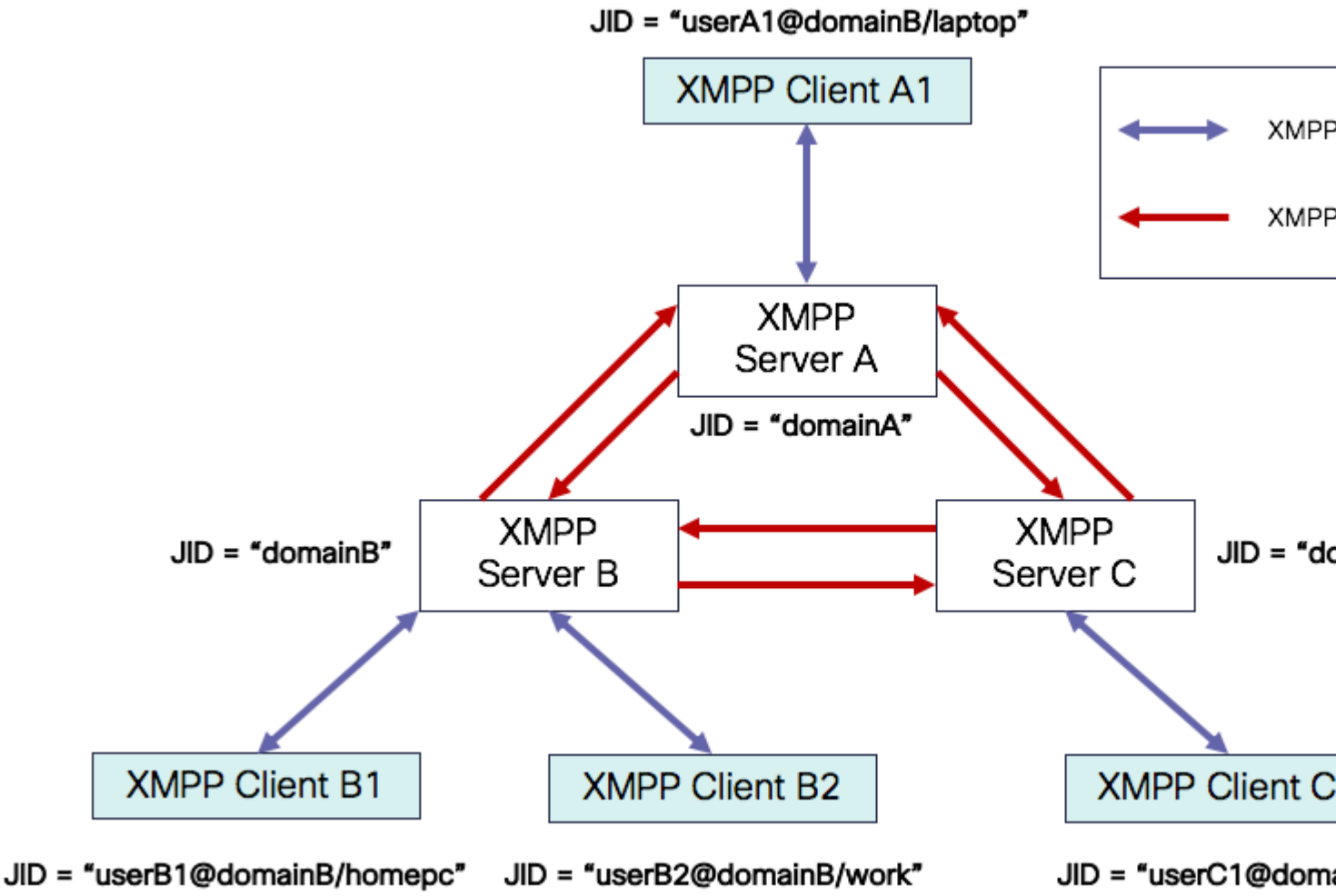
As conexões que usam fluxos bidirecionais sobre HTTP síncrono são chamadas de BOSH.

Entender a implementação do Finesse BOSH

Entender XMPP

O Extensible Messaging and Presence Protocol (XMPP) (também conhecido como Jabber) é um protocolo stateful em um modelo cliente-servidor. O XMPP permite a entrega rápida de pequenos pedaços de dados estruturados XML (eXtensible Markup Language) de uma entidade para outra. O XMPP/Jabber é amplamente usado em mensagens instantâneas (IM) e aplicativos de presença.

Todas as entidades XMPP são identificadas por sua ID Jabber (JID).



Esquema de endereçamento JID: user@domain/resource

usuário	nome de usuário do cliente no servidor XMPP ou nome da sala de conferência
domínio	Nome de domínio totalmente qualificado (FQDN) do servidor XMPP
recurso	identificador da entidade/endpoint específico do usuário (por exemplo, laptop, smartphone etc.), um identificador de sessão ou o nome do nó pubsub

---

> **Observação**: os três componentes JID não são usados em todos os casos. Normalmente, um servidor seria apenas definido pelo domínio, uma sala de conferência definida por user@domain e um cliente por user@domain/resource.

As mensagens XMPP são chamadas de estrofes. Há três estrofes centrais no XMPP:

1. <mensagem>: uma direção, um destinatário

2. <presença>: uma direção, publicar para muitos

3. <iq>: info/query - request/response

Todas as estrofes têm endereços de origem e destino e a maioria das estrofes também tem atributos type, id e xml:lang.

Atributo Stanza	**Propósito**
para	JID de destino
de	JID de origem
tipo	finalidade da mensagem
id	identificador exclusivo usado para vincular uma solicitação a uma resposta para estrofes <iq>
xml:lang	define o idioma padrão para qualquer XML legível por humanos na estrofe

Exemplo de mensagem XMPP

```
<message to='person1@example' from='person2@example' type='chat'>
  <subject> Team meeting </subject>
  <body>Hey, when is our meeting today? </body>
  <thread>A4567423</thread>
</message>
```

Implementação XMPP com Finesse

Se uma aplicação Web precisar trabalhar com XMPP, vários problemas surgirão. Os navegadores não suportam XMPP sobre Transmission Control Protocol (TCP) nativamente, portanto todo o tráfego XMPP deve ser manipulado por um programa que seja executado dentro do navegador. Servidores Web e navegadores se comunicam através de mensagens do protocolo HTTP, de modo que o Finesse e outros aplicativos Web embrulham mensagens XMPP dentro de mensagens HTTP.

A primeira dificuldade com essa abordagem é que o HTTP é um protocolo stateless. Isso significa que cada solicitação HTTP não está relacionada a nenhuma outra solicitação. No entanto, esse problema pode ser resolvido por meios de aplicação, por exemplo, através do uso de cookies/dados de postagem.

A segunda dificuldade é o comportamento unidirecional do HTTP. Somente o cliente envia solicitações e o servidor só pode responder. A incapacidade do servidor de enviar dados torna não natural a implementação de XMPP sobre HTTP.

Esse problema não existe na especificação XMPP Core original (RFC 6120), onde o XMPP está vinculado ao TCP. No entanto, se você quiser resolver o problema com XMPP vinculado ao HTTP, por exemplo,

porque o Javascript pode enviar solicitações HTTP, há duas soluções possíveis. Ambos exigem uma ponte entre HTTP e XMPP.

As soluções propostas são:

1. Pesquisa (protocolo legado): solicitações HTTP repetidas solicitando novos dados definidos em XEP-0025: Pesquisa HTTP Jabber

2. O polling longo também é conhecido como BOSH: protocolo de transporte que emula a semântica de uma conexão TCP bidirecional de longa duração entre duas entidades usando eficientemente vários pares de solicitação/resposta HTTP síncronos sem exigir o uso de polling frequente definido em XEP-0124: Associação HTTP e estendido por XEP-0206: XMPP Over BOSH

O Finesse implementa o BOSH, pois ele é bastante eficiente do ponto de vista da carga do servidor e do tráfego. A razão para usar BOSH é para encobrir o fato de que o servidor não precisa responder assim que há uma solicitação. A resposta é atrasada até um tempo especificado até que o servidor tenha dados para o cliente e, em seguida, é enviada como uma resposta. Assim que o cliente obtém a resposta, ele faz uma nova solicitação e assim por diante.

O cliente desktop Finesse (aplicação Web) estabelece uma conexão BOSH obsoleta sobre a porta TCP 7443 a cada 30 segundos. Após 30 segundos, se não houver atualizações do Finesse Notification Service, o serviço de Notificação enviará uma resposta HTTP com um corpo de resposta 200 OK e um corpo de resposta quase vazio. Se o serviço de notificação tiver uma atualização sobre a presença de um agente ou um evento de diálogo (chamada), por exemplo, os dados serão enviados imediatamente ao cliente Web Finesse.

Exemplo de solicitação/resposta XMPP Finesse

Este exemplo mostra a primeira resposta de solicitação de mensagem XMPP compartilhada entre o cliente Finesse e o servidor Finesse para configurar a conexão BOSH.

```
Finesse client request:
<body xmlns="http://jabber.org/protocol/httpbind" xml:lang="en-US" xmlns:xmpp="urn:xmpp:xbosh" hold="1"

Finesse server response:
<body xmlns="http://jabber.org/protocol/httpbind" xmlns:stream="http://etherx.jabber.org/streams" authid
```

Para resumir:

1. O cliente Web Finesse tem uma conexão HTTP obsoleta (http-bind) configurada para o servidor Finesse através da porta TCP 7443. Isso é conhecido como uma votação longa BOSH.

2. O Finesse Notification Service é um serviço de presença que publica atualizações sobre o estado de um agente, chamada, etc.

3. Se o serviço de Notificação tiver uma atualização, ele responderá à solicitação http-bind com a atualização de estado como uma mensagem XMPP no corpo da resposta HTTP.

4. Se não houver atualizações de estado 30 segundos após o recebimento da solicitação http-bind, o Serviço de Notificação responderá sem nenhuma atualização de estado para permitir que o cliente Web Finesse envie outra solicitação http-bind. Isso serve como uma forma do serviço de Notificação saber que o cliente Web Finesse ainda pode se conectar ao serviço de Notificação e que o agente não fechou o navegador ou colocou o computador em suspensão, e assim por diante.

Entender mensagens XMPP Finesse e nós XMPP

A Finesse também implementa a especificação XMPP XEP-0060: Publish-Subscribe. A finalidade desta especificação é permitir que o servidor XMPP (serviço de Notificação) obtenha informações publicadas para nós XMPP (tópicos) e envie eventos XMPP para entidades inscritas no nó. No caso do Finesse, o servidor de Integração entre Telefonia e Computador (CTI - Computer Telephony Integration) envia mensagens de CTI ao serviço da Web Finesse para informar ao Finesse sobre atualizações de configuração, como, mas não limitado a, criação de agentes ou de filas de serviço de contato (CSQ - Contact Service Queue) ou informações sobre uma chamada. Essas informações são convertidas em uma mensagem XMPP que o serviço Web Finesse publica no serviço de notificação Finesse. O serviço Finesse Notification envia mensagens XMPP sobre BOSH para agentes inscritos em determinados nós XMPP.

Alguns dos objetos da API do Finesse definidos no Finesse Web Services Developer Guide são nós XMPP. Os clientes Web Finesse do agente e do supervisor podem assinar atualizações de eventos para alguns desses nós XMPP para ter informações atualizadas sobre eventos em tempo real (como eventos de chamada, eventos de estado, etc.). Esta tabela mostra os nós XMPP que estão ativados para pubsub.

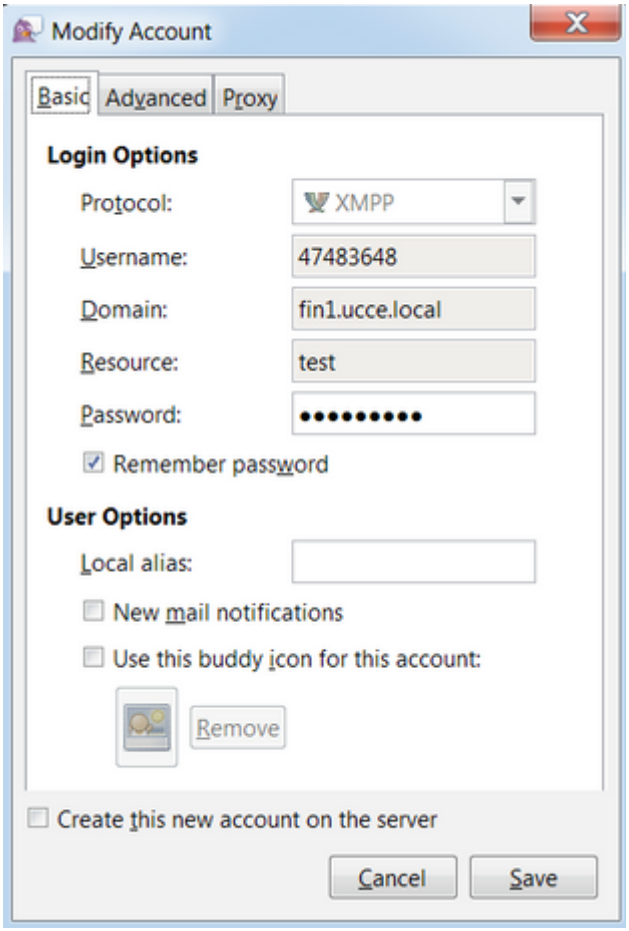
Objeto de API Finesse	**Propósito**	**Assinatura**
/finesse/api/User/<LoginID>	Mostra o estado e o mapeamento da equipe do agente	Agentes e supervisores
/finesse/api/User/<LoginID>/Dialogs	Mostra as chamadas tratadas pelo agente	Agentes e supervisores
/finesse/api/User/<LoginID>/ClientLog	Usado para capturar logs de cliente do botão **Enviar relatório de erros**	Agentes e supervisores
/finesse/api/User/<LoginID>/Queue/<queueID>	Mostra dados de estatísticas da fila (se habilitado)	Agentes e supervisores
/finesse/api/Team/<TeamID>/Users	Mostra os agentes que pertencem a uma determinada equipe, incluindo informações de estado	Supervisores
/finesse/api/SystemInfo	Mostra o estado do servidor Finesse. Usado para determinar se o failover é necessário	Agentes e supervisores

Exemplo 1: Usar Pidgin para exibir nós XMPP do Finesse

Etapa 1. Baixe e instale o Pidgin do cliente XMPP.

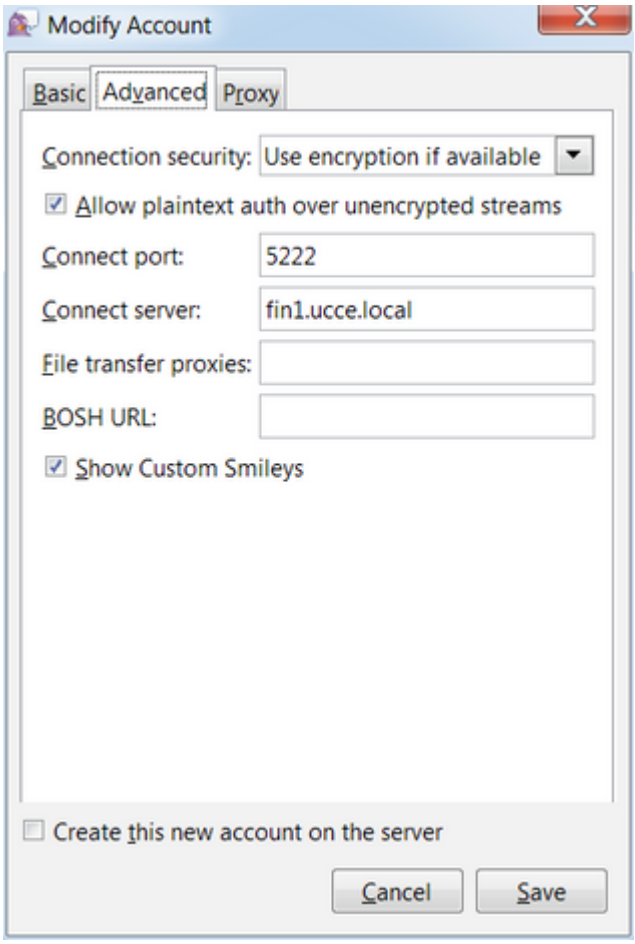
Etapa 2. Navegue até **Contas > Modificar > Básico** e configure as **Opções de login**:

- Protocolo: XMPP
- Nome de usuário: LoginID para qualquer agente
- Domínio: FQDN do servidor Finesse
- Recurso: Espaço reservado - qualquer valor pode ser usado, por exemplo, teste
- Senha: Senha do agente
- Marque a caixa de seleção **Lembrar senha**.



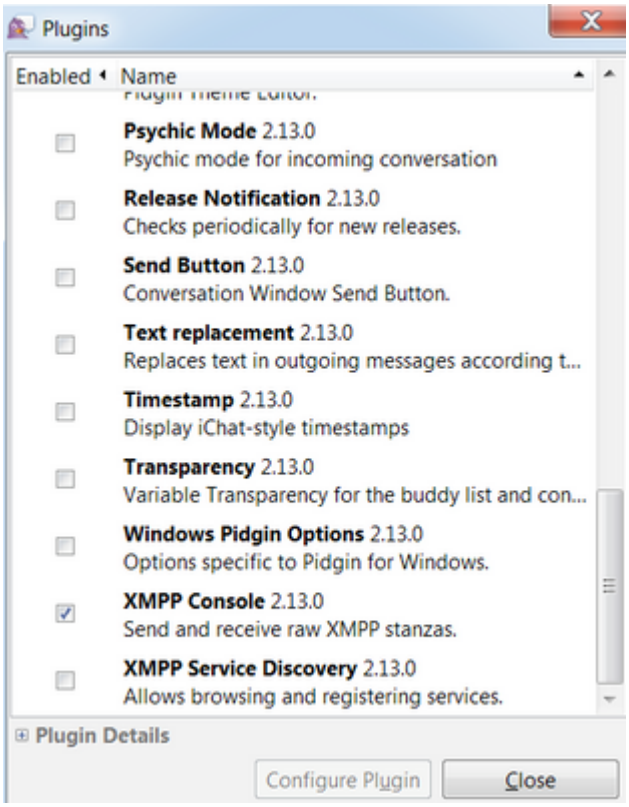
Etapa 3. Navegue até **Contas > Modificar > Avançado** e configure:

- Segurança da conexão: usar criptografia, se disponível
- Marque a caixa de seleção **Permitir autenticação de texto sem formatação de outros fluxos não criptografados**.
- Porta de conexão: 5222. Use a porta padrão 5222. Essa porta é necessária para clientes XMPP externos. Os clientes de desktop Finesse usam 7443. Não use a porta 7443.
- Servidor de conexão: FQDN do servidor Finesse

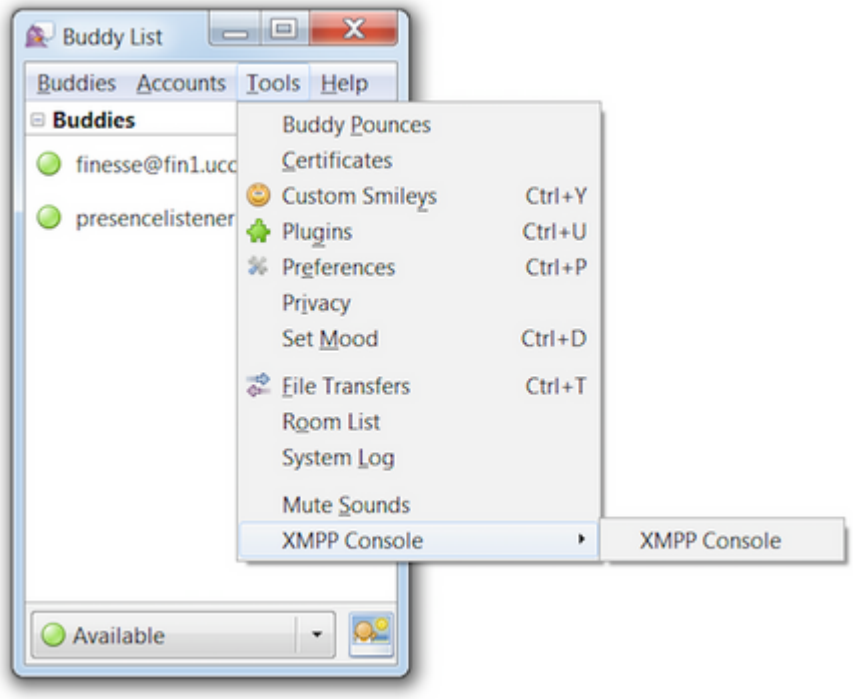


Observação: a porta 5222 é usada somente porque os clientes Web Finesse podem usar a porta 7443 para se conectar ao serviço de notificação.

Etapa 4. Navegue até **Ferramentas > Plug-ins** e ative o Console XMPP.

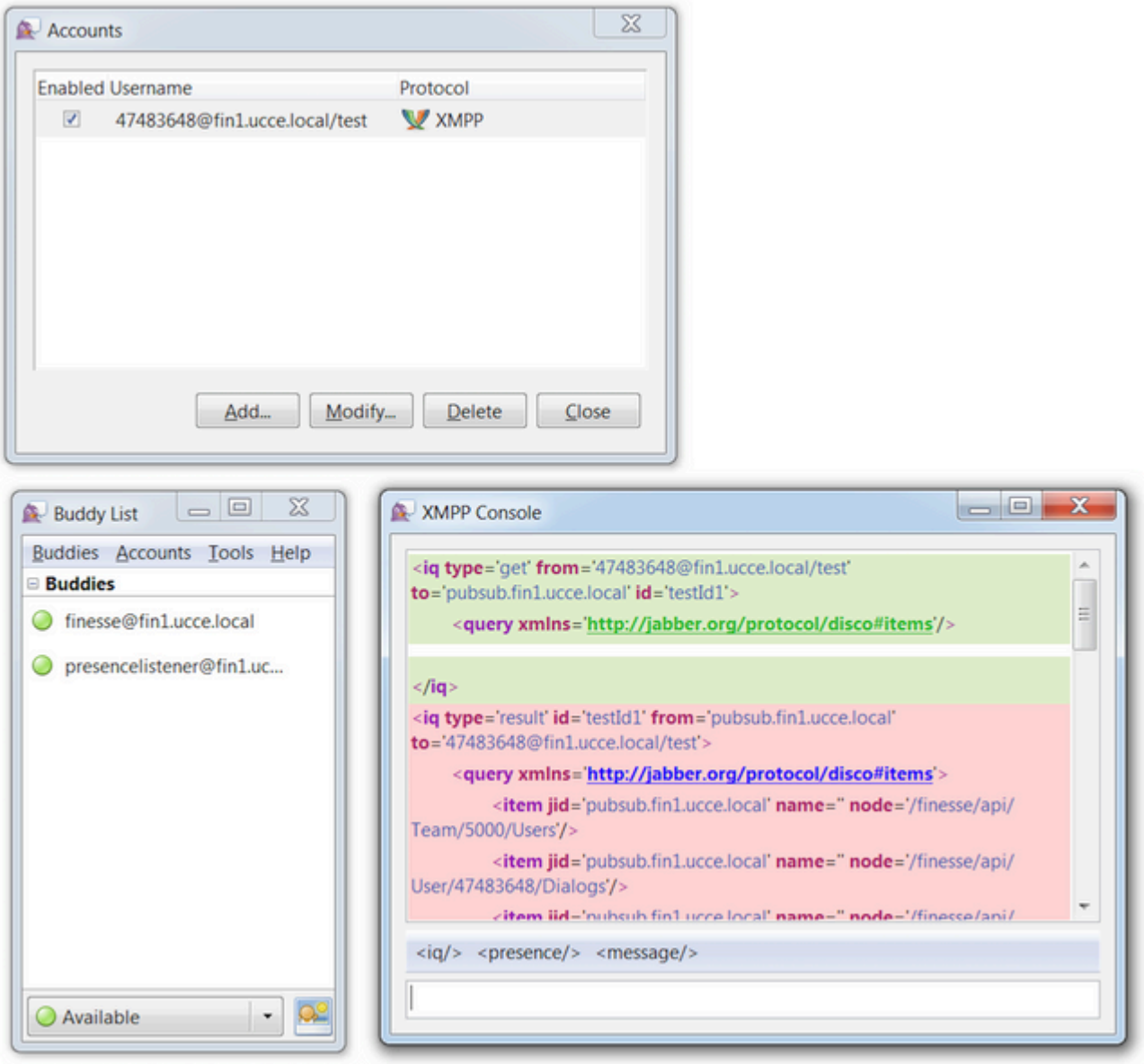


Etapa 5. Navegue até **Ferramentas > Console XMPP > Console XMPP** para abrir o Console XMPP.



Etapa 6. Execute esta mensagem **<iq>** para ver todos os nós XMPP existentes.

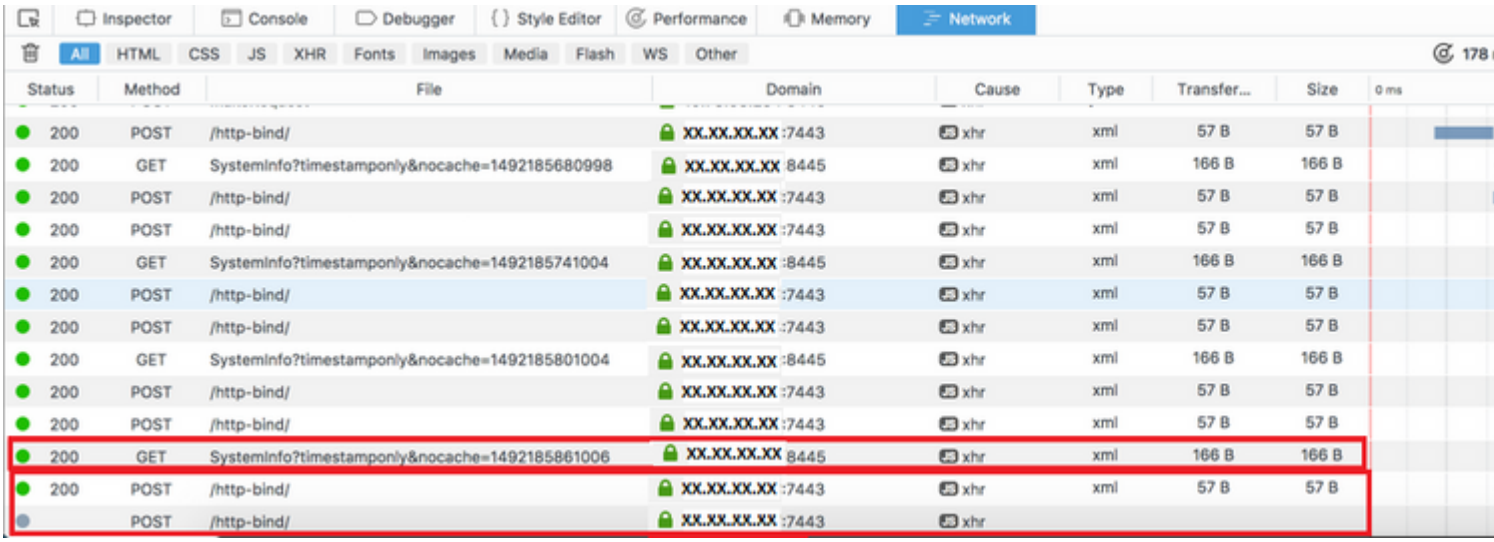
Por exemplo:



Em um ambiente de laboratório com dois agentes e duas filas do Contact Service configuradas, essa saída está contida na resposta do Finesse:

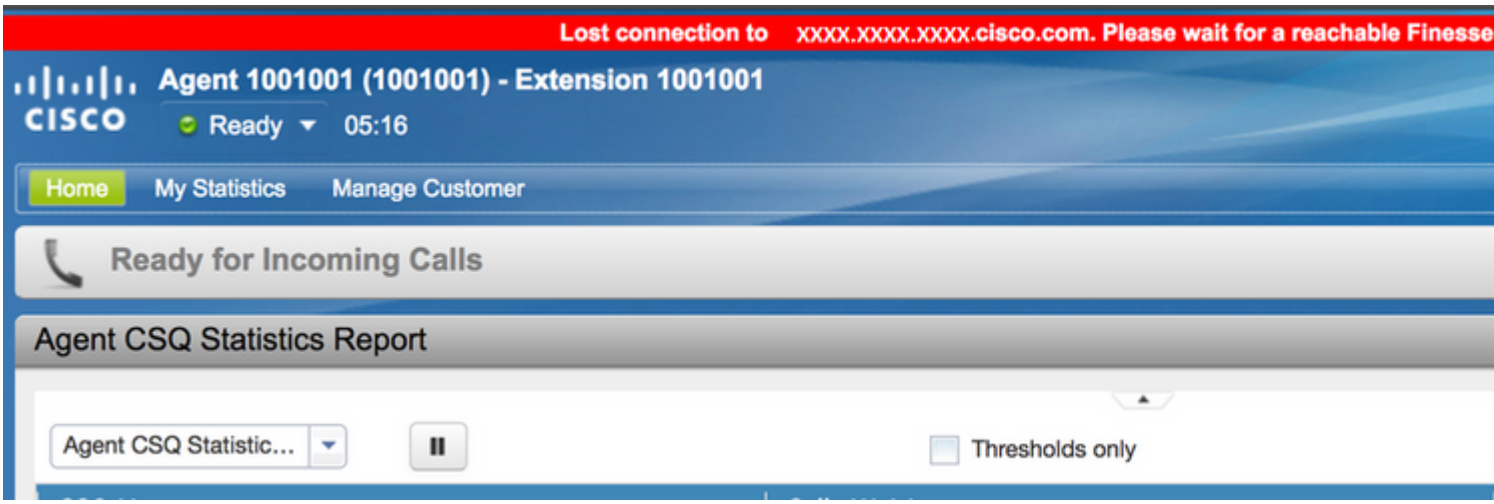
Exemplo 2: Usar a guia Rede das ferramentas de desenvolvedor do navegador para exibir mensagens HTTP

Cada navegador tem um conjunto de ferramentas para desenvolvedores. A aba Rede das ferramentas do desenvolvedor mostra as mensagens HTTP enviadas e recebidas pelo cliente Web Finesse (navegador). Por exemplo, esta imagem mostra como o cliente Web Finesse envia uma solicitação SystemInfo que verifica o status do Finesse Tomcat a cada minuto como uma verificação de failover. Além disso, as mensagens http-bind da conexão BOSH também são exibidas. O servidor Finesse envia de volta uma resposta em 30 segundos se não houver atualizações para publicar nos nós XMPP em que o cliente Web está inscrito.



Solucionar problemas da mensagem de erro de desconexão do BOSH

Quando uma desconexão BOSH ocorre, o erro Perdeu a conexão com o {Finesse Server FQDN}. Aguarde até que um Servidor Finesse acessível seja encontrado... será exibido em um banner vermelho na parte superior da área de trabalho do Finesse.



Esta mensagem é exibida porque, neste momento, nenhum evento de assinatura XMPP pode ser recebido do serviço de notificação do Cisco Finesse. Portanto, as informações de estado e os detalhes da chamada não podem ser exibidos na área de trabalho do agente.

Para o UCCX, 60 segundos depois que o navegador se desconecta, o agente é colocado em um estado Logoff. O agente pode estar no estado Pronto ou Não pronto para o logoff.

Para o UCCE, o Finesse leva até 120 segundos para detectar quando um agente fecha o navegador ou o navegador trava e o Finesse espera 60 segundos antes de enviar uma solicitação de logoff forçado ao servidor CTI, o que faz com que o servidor CTI coloque o agente em um estado Não pronto. Nessas condições, o Finesse pode levar até 180 segundos para desconectar o agente. Ao contrário do UCCX, o agente passa para um estado Não pronto em vez do estado Logoff.

Observação: a desconexão da CTI não está pronta vs. O comportamento do estado de logoff no UCCE é controlado pelo parâmetro PG /LOAD. De acordo com as Notas de versão do Unified

---

Contact Center Enterprise & Hosted Versão 10.0(1), o parâmetro /LOAD foi preterido a partir do UCCE 10.0.

Para obter mais informações sobre o comportamento do UCCE Finesse Desktop, consulte a seção Comportamento do Desktop do capítulo Mecanismos de failover do Cisco Finesse no Guia de administração do Cisco Finesse.

Observação: os valores do temporizador podem ser alterados no futuro de acordo com o requisito do produto.

Análise de log

Os registros do serviço Finesse e UCCX Notification podem ser coletados via RTMT ou via CLI:

file get ativelog /desktop recurs compress

Logs do serviço de notificação de depuração

Observação: defina logs de nível de depuração somente enquanto reproduz um problema. Desative as depurações depois que o problema tiver sido reproduzido.

Observação: o Finesse 9.0(1) não tem log de nível de depuração. O registro de nível de depuração foi introduzido no Finesse 9.1(1). O processo para ativar o registro é diferente no 9.1(1) em comparação com o Finesse 10.0(1) - 11.6(1). Para esse processo, consulte o guia Finesse Administration and Serviceability.

Ative os logs de depuração do serviço de notificação do Unified Contact Center Express (UCCX), conforme mostrado:

<#root>

```
admin:
utils uccx notification-service log enable


WARNING! Enabling Cisco Unified CCX Notification Service logging can affect system performance
and should be disabled when logging is not required.

Do you want to proceed (yes/no)? yes

Cisco Unified CCX Notification Service logging enabled successfully.

NOTE: Logging can be disabled automatically if Cisco Unified CCX Notification Service is restarted.
```

Ative os logs de depuração do Notification Service do Unified Contact Center Enterprise (UCCE) (Finesse independente), conforme mostrado:

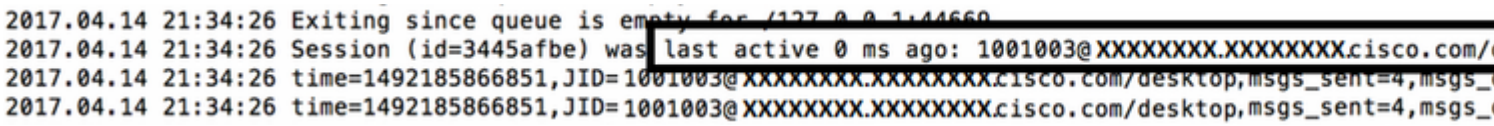
<#root>

```
admin:

utils finesse notification logging enable


Checking that the Cisco Finesse Notification Service is started...
The Cisco Finesse Notification Service is started.

Cisco Finesse Notification Service logging is now enabled.

WARNING! Cisco Finesse Notification Service logging can affect system performance
and should be disabled when logging is not required.

Note: Logging can be disabled automatically if you restart the Cisco Finesse Notification Service
```

Esses logs estão na pasta /desktop/logs/openfire e são nomeados como debug.log.

Como mostrado na imagem, o debug.log do Serviço de Notificação (Openfire) mostra a associação http com desktop junto com o endereço IP e a porta do PC do agente.

Como mostrado na imagem, o último 0 ms ativo mostra que a sessão ainda está ativa.



O fechamento da sessão ociosa pelo Openfire indica que o logoff do agente pode ser acionado em 60 segundos, quando o Finesse pode enviar um logoff forçado com um código de razão de 255 para o servidor CTI. O comportamento real da área de trabalho sob essas condições depende da configuração de Logout na desconexão do agente (LOAD) no UCCE. No UCCX, esse é sempre o comportamento.

Se o cliente Finesse não enviar mensagens http-bind para o servidor Finesse, os logs podem mostrar o tempo de atividade da sessão e mostrar o fechamento da sessão.

```
2017.06.17 00:14:34 Session (id=f382a015) was last active 0 ms ago: 1001003@xxxxx.xxxx.xxx.cisco.com/des
2017.06.17 00:15:04 Session (id=f382a015) was last active 13230 ms ago: 1001003@xxxxx.xxxx.xxx.cisco.com
2017.06.17 00:15:34 Session (id=f382a015) was last active 43230 ms ago: 1001003@xxxxx.xxxx.xxx.cisco.com
2017.06.17 00:16:04 Session (id=f382a015) was last active 63231 ms ago: 1001003@xxxxx.xxxx.xxx.cisco.com

2017.06.17 00:17:04 Unable to route packet. No session is available so store offline. <message from="pub
```

Logs do Serviço de Notificação de Informações

Esses logs estão na pasta /desktop/logs/openfire e são nomeados info.log. Se o cliente Finesse não enviar mensagens http-bind para o servidor Finesse, os logs poderão mostrar a sessão como inativa.

```
2017.06.17 00:16:04 Closing idle session (id=f382a015): 1001003@xxxxx.xxxx.xxx. cisco.com/desktop
after inactivity for more than threshold value of 60
2017.06.17 00:16:04 A session is closed for 1001003@xxxxx.xxxx.xxx. cisco.com/desktop
```

Logs de serviços da Web

Esses logs estão na pasta /desktop/logs/webservices e são nomeados Desktop-webservices.YYY-MM-DDTHH-MM-SS.sss.log. Se o cliente Finesse não enviar mensagens http-bind para o servidor Finesse dentro do período de tempo especificado, os logs poderão mostrar a presença do agente como indisponível e 60 segundos depois, um logout orientado por presença poderá ocorrer.

```
0000001043: XX.XX.XX.XXX: Jun 17 2017 00:16:04.630 +0530: %CCBU_Smack Listener Processor (1)-6-PRESENCE_
0000000417: XX.XX.XX.XXX: Jun 17 2017 00:16:04.631 +0530: %CCBU_Smack Listener Processor (1)-6-UNSUBSCRI
0000001044: XX.XX.XX.XXX:  Jun 17 2017 00:16:04.631 +0530: %CCBU_Smack Listener Processor (1)-6-AGENT_PR
0000001051: XX.XX.XX.XXX:  Jun 17 2017 00:16:35.384 +0530: %CCBU_pool-8-thread-1-6-AGENT_PRESENCE_MONITO
0000001060: XX.XX.XX.XXX:: Jun 17 2017 00:17:04.632 +0530: %CCBU_CoreImpl-worker12-6-PRESENCE DRIVEN LOG
0000001061: XX.XX.XX.XXX:: Jun 17 2017 00:17:04.633 +0530: %CCBU_CoreImpl-worker12-6-MESSAGE_TO_CTI_SERV
1, workmode : 0, reason code: 255, forceflag :1, agentcapacity: 1, agentext: 1001003, agentid: 1001003,
0000001066:  XX.XX.XX.XXX:: Jun 17 2017 00:17:04.643 +0530: %CCBU_CTIMessageEventExecutor-0-6-DECODED_ME
skillGroupNumber=-1, skillGroupPriority=0, agentState=1 (LOGOUT), eventReasonCode=255, numFltSkillGroups
duration=null, nextAgentState=null, fltSkillGroupNumberList=[], fltSkillGroupIDList=[], fltSkillGroupPri
msgID=30, timeTracker={"id":"AgentStateEvent","CTI_MSG_RECEIVED":1497638824642,"CTI_MSG_DISPATCH":149763
Decoded Message to Finesse from backend cti server
```

Razões comuns para desconexão BOSH

As conexões BOSH são configuradas pelo cliente Web e o servidor Finesse determina se a presença do agente está indisponível. Esses problemas são quase sempre problemas do lado do cliente relacionados ao navegador, ao computador do agente ou à rede, já que o ônus da inicialização da conexão é do cliente.

Problema - Os agentes se desconectam em momentos diferentes (problema no lado do cliente)

Ações recomendadas

Verifique estes problemas:

1. Problema de rede:

- Revisar regras e registros do firewall â€” A porta TCP 7443 não deve ser bloqueada nem acelerada
- Use um farejador de tráfego da Web HTTP como Fiddler® ou Wireshark® para confirmar se o navegador envia solicitações de ligação http pela porta TCP 7443 e recebe respostas

- Verificar se há retardo excessivo ou quedas de pacotes em todos os dispositivos/interfaces de rede entre o computador do agente e o servidor Finesse
    - O traceroute pode ser útil para determinar o caminho e os atrasos
        - Em um computador Microsoft® Windows®: tracert {Finesse Server IP | FQDN do Finesse Server}
        - Em um Mac®: traceroute {Finesse Server IP | FQDN do Finesse Server}
        - No software Cisco IOS®, as estatísticas da interface podem ser verificadas: show interfaces
            - Consulte Troubleshooting de Quedas de Filas de Entrada e de Saída

- Coletar logs do Finesse Client para um agente de teste. Os logs do cliente podem ser coletados de três maneiras:
    1. Logs do console Web do navegador
        - Console Web Firefox
        - Console da Web do Microsoft Edge
        - Console Web do Chrome
    2. Pressione o botão Send Error Report na página Finesse e colete os logs do servidor Finesse. Os logs estão localizados em /desktop/logs/clientlogs.
    3. Faça login via https://<Finesse-FQDN>/desktop/locallog e colete os logs depois que o problema ocorrer.

A cada minuto, o cliente se conecta ao servidor Finesse para calcular o desvio e a latência da rede:

```
<PC date-time with GMT offset>: : <Finesse FQDN>: <Finesse server date-time with offset>:
Header : Client: <date-time>, Server: <date-time>, Drift: <drift> ms, Network Latency (round trip): <RTT

2019-01-11T12:24:14.586 -05:00: : fin1.ucce.local: Jan 11 2019 11:24:14.577 -0600: Header : Client: 2019
```

No caso de qualquer problema de coleta de registros, consulte Troubleshooting Cisco Finesse Desktop Persistent Logging Problem

2. Navegador e/ou versão sem suporte:

Usar navegador/versão e configurações com suporte de acordo com as matrizes de compatibilidade:

Matriz de compatibilidade UCCE

Matriz de compatibilidade do UCCX

3. Condição de travamento do navegador devido ao conteúdo/processamento de outra guia/janela:

Verifique o fluxo de trabalho do agente para ver se ele:

- Geralmente, há outras guias ou janelas ativas que executam constantemente outros aplicativos em tempo real, como transmissão de música/vídeo, conexões WebSocket, clientes Web personalizados de Gerenciamento de Relacionamento com o Cliente (CRM) etc.
- Tem um número muito grande de abas ou janelas abertas
- Desabilitou o cache do navegador
- Mantêm o navegador em execução por muito tempo e não o fecham no final do dia de trabalho

4. Computador colocado em latência:

Verifique se o agente coloca o computador em suspensão antes de fazer logoff do Finesse ou se o

temporizador de configuração de suspensão do computador está muito baixo.

5. Problema de CPU alta ou memória alta no computador cliente:

- Se o navegador do agente for executado em um ambiente compartilhado, como o Microsoft Windows Remote Desktop Services, Citrix® XenApp®, o Citrix XenDesktop®, determine se o desempenho do navegador depende do número de usuários que executam o navegador ao mesmo tempo
  - Verifique se a memória e os recursos de CPU apropriados estão configurados com base no número de usuários

- Verifique os problemas de utilização de recursos do computador:
  - Windows:
    - Comando Get-Counter do Windows PowerShell que verifica % de tempo da CPU, megabytes de memória disponível e % de memória em uso a cada 2 segundos: Get-Counter -Counter "\Processor(_Total)\% Processor Time","\Memory\Available MBytes","\Memory\% Committed Bytes In Use" -SampleInterval 2 -Continuous
    - Como alternativa ao uso do PowerShell para exibir os contadores de desempenho do Windows, o Monitor de Desempenho do Windows pode ser usado
    - O Gerenciador de Tarefas pode ser usado para exibir estatísticas da CPU e da memória ao vivo globalmente e processo por processo
  - Mac:
    - Comando Terminal Top que verifica o total de CPU e memória ao vivo: top
      - Verificar processos e classificar por utilização da CPU: superior -o CPU
      - Verificar processos e classificar por utilização de memória: top -o MEM
    - O Monitor de Atividade pode ser usado para exibir estatísticas da CPU e da memória ao vivo globalmente e processo por processo

6. gadgets de terceiros executando atividade problemática e inesperada em segundo plano:

Teste o comportamento da área de trabalho do Finesse com todos os gadgets de terceiros removidos.

7. Problema de NTP no servidor ou cliente:

- Verifique **utils ntp status** no servidor do editor Finesse para garantir que a camada do servidor NTP seja 4 ou inferior
- Nos registros do cliente, verifique o desvio e a latência da rede

Problema - Todos os agentes se desconectam ao mesmo tempo (problema no lado do servidor)

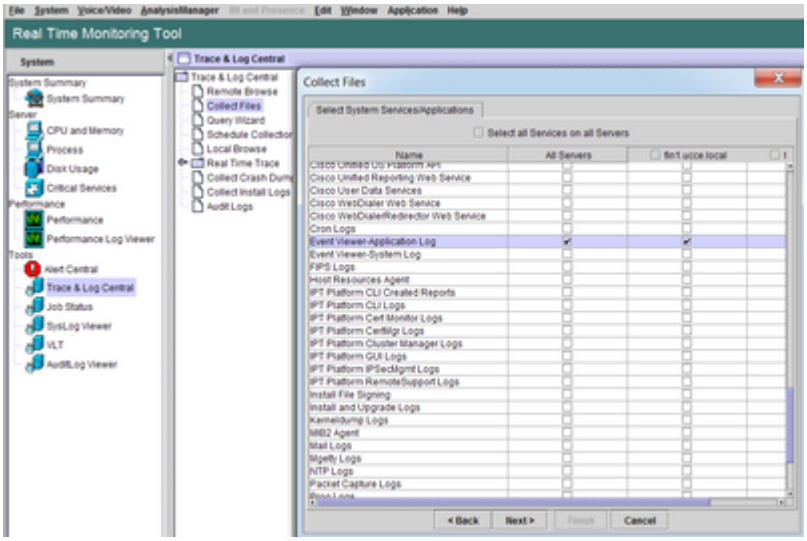
Ações recomendadas

Verifique estes problemas:

1. Desconexão do serviço Cisco Unified Communications Manager CTIManager. Se todos os provedores do CTIManager para UCCX estiverem desligados ou travarem, os agentes do UCCX verão o erro de banner vermelho. Os agentes UCCE não veem o banner vermelho se isso acontecer, mas as chamadas não são roteadas corretamente para os agentes.

- Verifique se o serviço Cisco CTIManager foi iniciado nos servidores CUCM usados como provedores CTI
- Verifique se o serviço Cisco CTIManager travou via Visualizador de Eventos - O aplicativo registra em RTMT para ver se o serviço Cisco CTIManager travou
  - Para coletar logs do visualizador de eventos no RTMT, navegue para **Sistema > Ferramentas**

> Central de Rastreamento e Log > Coletar Arquivos > Selecionar Serviços/Aplicativos do Sistema > Visualizador de Eventos - Log do Aplicativo.



- Para coletar os logs do Visualizador de Eventos-Aplicativo no CLI: file get ativelog /syslog/CiscoSyslog* abstime hh:mm:MM/DD/YY hh:mm:MM/DD/YY
- Para visualizar dumps principais na CLI: lista de utils core ative

---

Observação: os nomes de arquivo de dumps principais usam o formato: core.<ProcessID>.<SignalNumber>.<ProcessName>.<EpochTime>.
Exemplo: core.24587.6.CTIManager.1533441238
Assim, o tempo do acidente pode ser determinado a partir do tempo de época.

---

2. O Serviço de Notificação Finesse/UCCX parou ou travou:

- Verifique se há erros no serviço de notificação nos logs do aplicativo-visualizador de eventos ou se o serviço foi interrompido
- Verifique se o serviço de Notificação está ativo: lista de serviços utils
- Verifique as horas em que o serviço de Notificação foi desligado: file search ativelog /desktop/logs/openfire "Openfire stops"
- Verifique as horas em que o serviço de Notificação foi iniciado: file search ativelog /desktop/logs/openfire "HTTP bind service started"
- Verifique os despejos de memória do serviço de Notificação que resultaram de um travamento: file list ativelog /desktop/logs/openfire/*.hprof
- Verifique se o serviço de Notificação está escutando o tráfego na porta TCP 7443: show open ports regexp 7443.*LISTEN
- Verifique se esses defeitos são aplicáveis (esses defeitos causariam uma falha de logon para os agentes que estão conectados e, para os agentes já conectados, esses agentes verão a mensagem de desconexão Finesse do banner vermelho):
  - ID de bug Cisco CSCva7280 - Finesse Tomcat e Openfire Crash para caracteres XML inválidos
  - ID de bug da Cisco CSCva72325 - UCCX: Finesse Tomcat e Openfire Crash para caracteres XML inválidos

Reinicie o Cisco Finesse Tomcat e o serviço de notificação se houver suspeita de travamento. Isso só é recomendado em uma situação de inatividade da rede, caso contrário, essas reinicializações desconectam os agentes do servidor Finesse.

Etapas para UCCE:

- interrupção do serviço utils Cisco Finesse Tomcat
- serviço utils parar Serviço de Notificação Cisco Finesse
- início do serviço utils Cisco Finesse Tomcat
- início do serviço utils Serviço de notificação Cisco Finesse

Etapas para o UCCX:

- interrupção do serviço utils Cisco Finesse Tomcat
- o serviço utils interrompe o serviço de notificação do Cisco Unified CCX
- início do serviço utils Cisco Finesse Tomcat
- início do serviço utils Serviço de notificação do Cisco Unified CCX

Usar alimentador

Configurar o Fiddler pode ser uma tarefa um tanto desafiadora sem entender as etapas necessárias e entender como o Fiddler funciona. O Fiddler é um proxy da Web com funções humanas que fica entre o cliente Finesse (navegador da Web) e o servidor Finesse. Devido às conexões seguras entre o cliente Finesse e o servidor Finesse, isso adiciona uma camada de complexidade à configuração do Fiddler para exibir mensagens seguras.

Problema comum de Fiddler

Como o Fiddler está entre o cliente Finesse e o servidor Finesse, o aplicativo Fiddler precisa criar certificados assinados para todas as portas TCP Finesse que exigem certificados:

Certificados de serviço do Cisco Finesse Tomcat

1. TCP 8445 do servidor editor Finesse (e/ou 443 para UCCE)
2. Servidor de assinante Finesse TCP 8445 (e/ou 443 para UCCE)

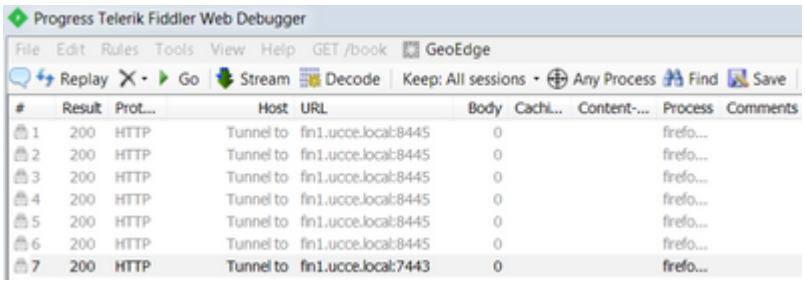
Certificados do serviço de notificação Cisco Finesse (Unified CCX)

1. TCP 7443 do servidor do editor Finesse
2. TCP 7443 de servidor de assinante Finesse

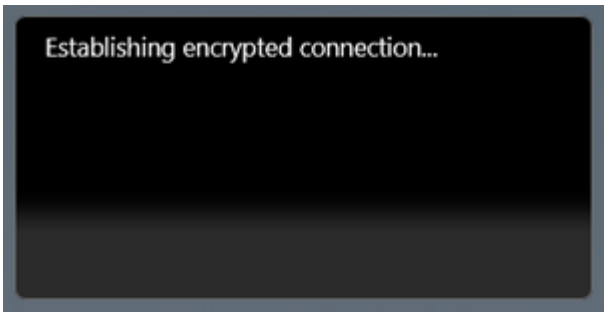
A descriptografia HTTPS deve ser habilitada para o Fiddler gerar certificados dinamicamente em nome do servidor Finesse. Isso não é habilitado por padrão.

Se a descriptografia HTTPS não estiver configurada, a conexão de túnel inicial com o serviço de Notificação será vista, mas o tráfego http-bind não. O folheador mostra apenas:

```
Tunnel to <Finesse server FQDN>:7443
```



Em seguida, os certificados Finesse assinados por Fiddler devem ser confiáveis pelo cliente. Se esses certificados não forem confiáveis, ultrapassar o estágio Estabelecendo conexão criptografada... do login do Finesse não será possível.



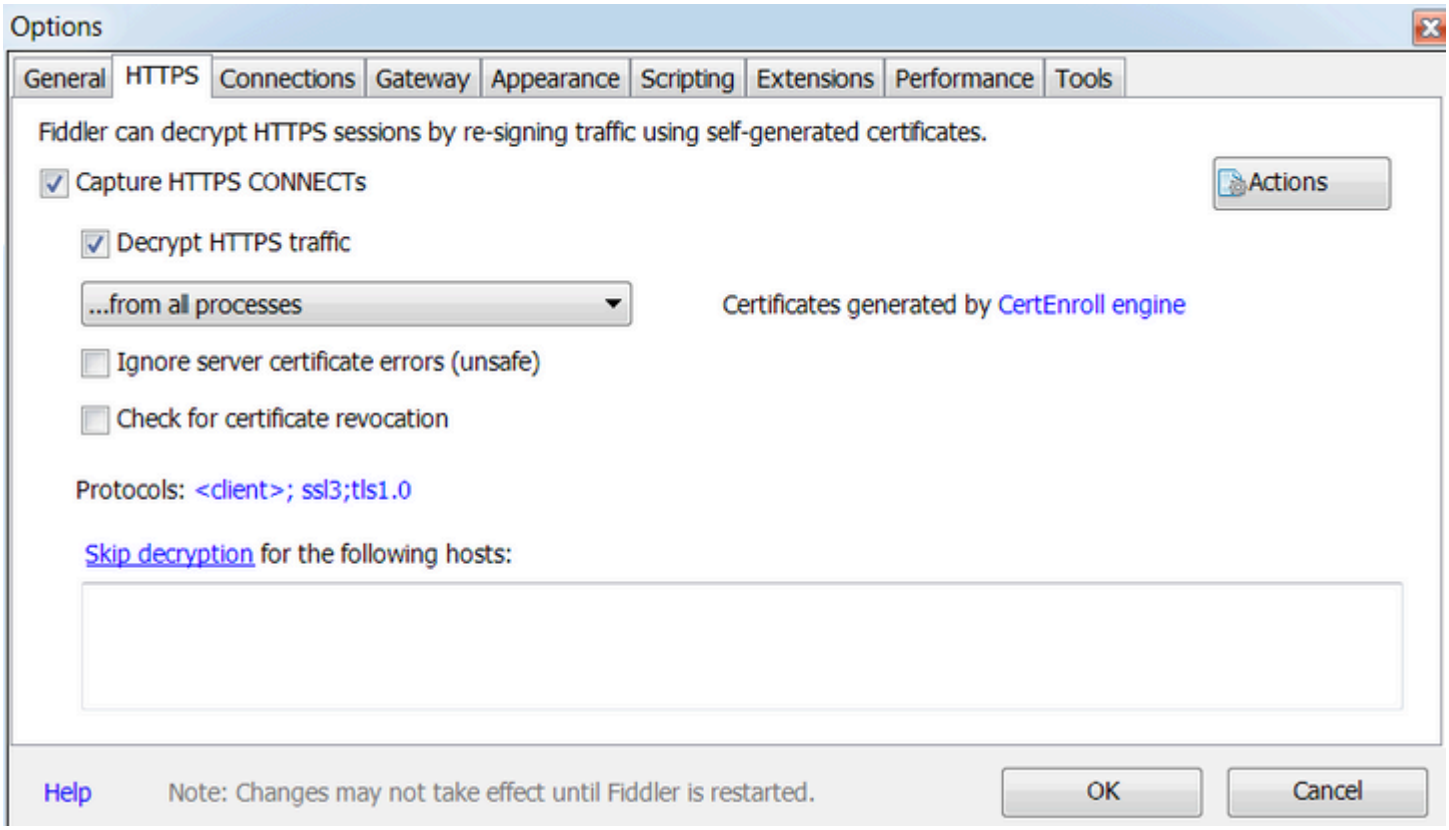
Em alguns casos, aceitar as exceções de certificado do login não funciona e os certificados precisam ser confiáveis manualmente pelo navegador.

Exemplo de etapas de configuração

> **Cuidado**: a configuração de exemplo fornecida é para Fiddler v5.0.20182.28034 para .NET 4.5 e Mozilla Firefox 64.0.2 (32 bits) no Windows 7 x64 em um ambiente de laboratório. Esses procedimentos não podem ser generalizados para todas as versões do Fiddler, todos os navegadores ou todos os sistemas operacionais do computador. Se sua rede estiver ativa, certifique-se de que você compreende o impacto potencial de qualquer configuração. Consulte a documentação oficial do Fiddler para obter mais informações.

Etapa 1. Baixar o alimentador

Etapa 2. Habilite a descriptografia HTTPS. Navegue para **Ferramentas > Opções > HTTPS** e marque a caixa de seleção **Descriptografar tráfego HTTPS**.

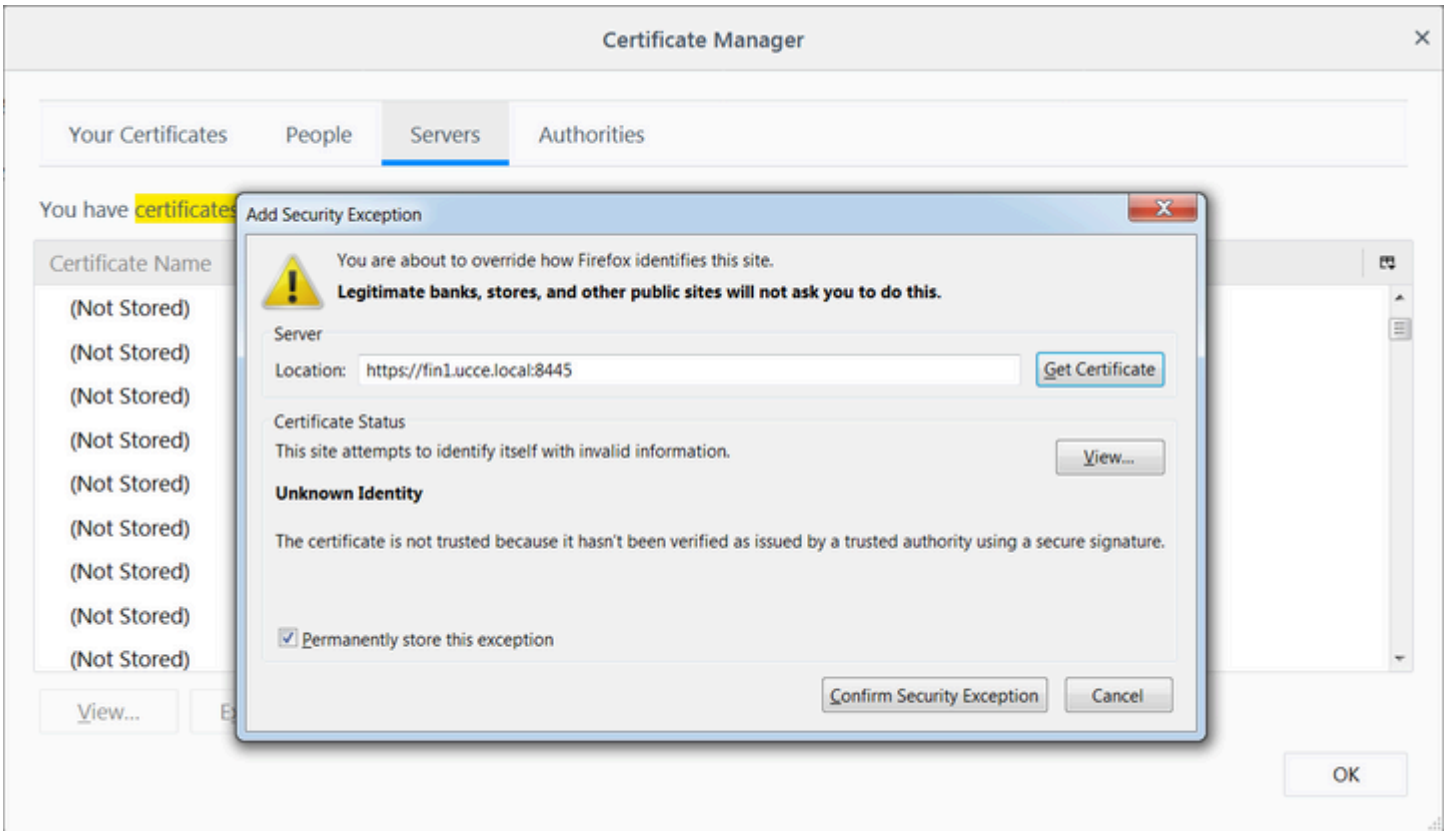


Etapa 3. Uma caixa de mensagem de aviso é aberta solicitando que o certificado raiz do alimentador seja confiável. Selecione **Sim**.

Etapa 4. Uma caixa de mensagem de aviso é aberta com a mensagem "Você está prestes a instalar um certificado de uma autoridade de certificação (CA) alegando representar: DO_NOT_TRUST_FiddlerRoot... Deseja instalar este certificado?". Selecione **Sim**.

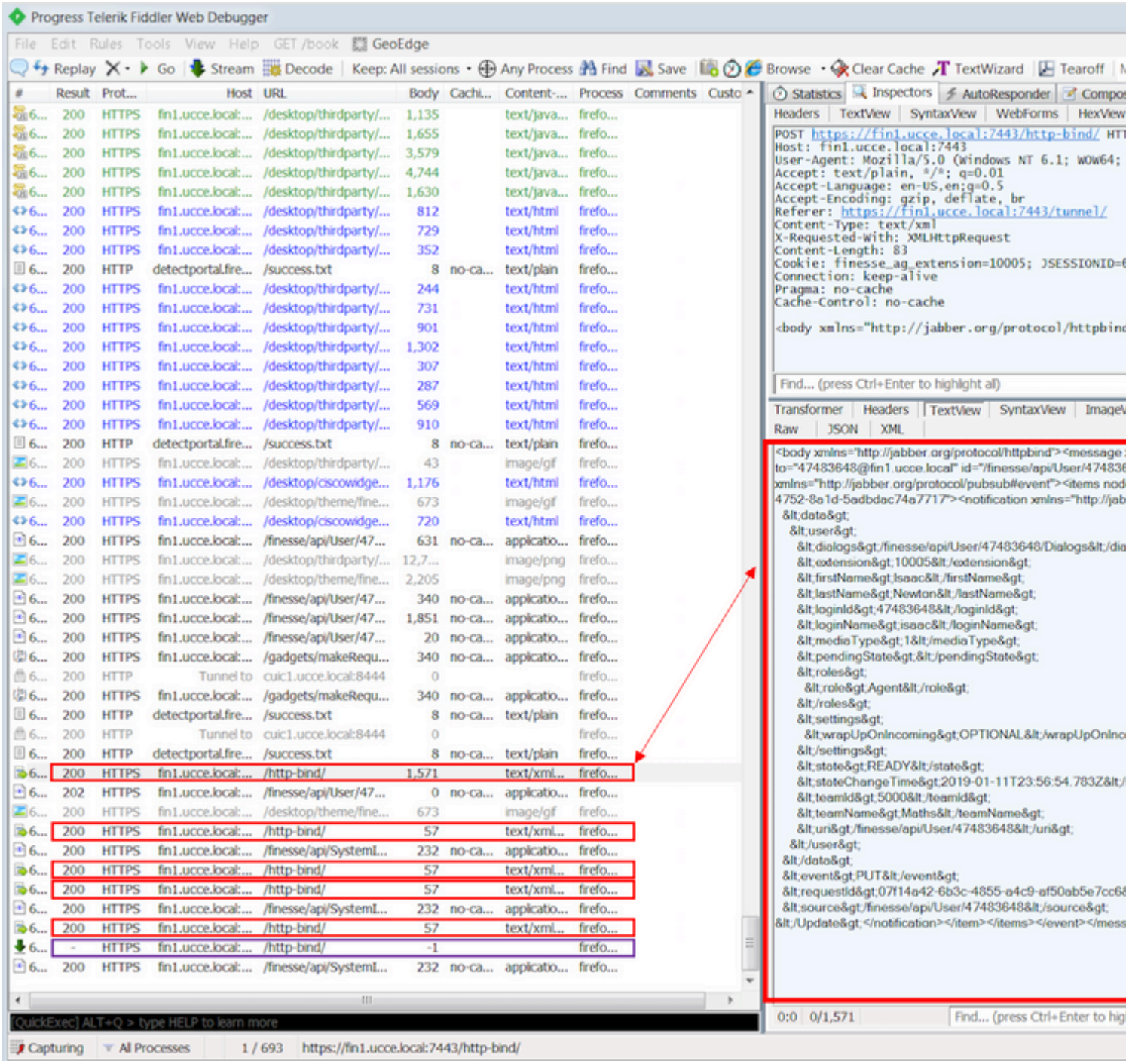
Etapa 5. Adicione manualmente os certificados de editor e assinante do Finesse ao armazenamento confiável de certificados do computador ou navegador. Assegure as portas 8445, 7443 e (somente para UCCE) 443. Por exemplo, no Firefox, isso pode ser feito simplesmente sem fazer o download de certificados da página Finesse Operating System Administration:

Options > Find in Options (search) **> Certificates > Servers > Add Exception > Location >** Insira https://<Finesse server>:port para as portas relevantes para ambos os servidores Finesse.

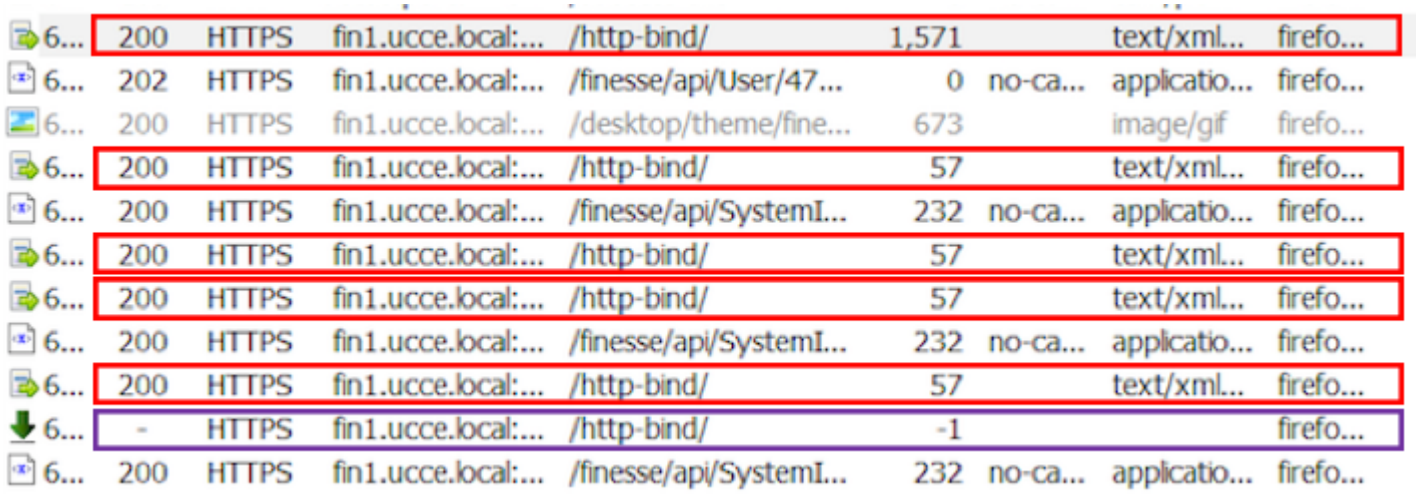


Etapa 6. Faça login no Finesse e veja as mensagens http-bind deixarem o cliente Finesse para o Finesse Server via Fiddler.

No exemplo fornecido, as 5 primeiras mensagens mostram mensagens http-bind que foram respondidas pelo servidor Finesse. A primeira mensagem contém 1571 bytes de dados retornados no corpo da mensagem. O corpo contém uma atualização XMPP referente a um evento de agente. A mensagem http-bind final foi enviada pelo cliente Finesse, mas não obteve uma resposta do servidor Finesse. Isso pode ser determinado quando você vê que o resultado HTTP é nulo (-) e o número de bytes no corpo da resposta é nulo (-1).



Visão mais detalhada dos dados:



6...	200	HTTPS	fin1.ucce.local:...	/http-bind/	1,571		text/xml...	firefo...
6...	202	HTTPS	fin1.ucce.local:...	/finesse/api/User/47...	0	no-ca...	applicatio...	firefo...
6...	200	HTTPS	fin1.ucce.local:...	/desktop/theme/fine...	673		image/gif	firefo...
6...	200	HTTPS	fin1.ucce.local:...	/http-bind/	57		text/xml...	firefo...
6...	200	HTTPS	fin1.ucce.local:...	/finesse/api/SystemI...	232	no-ca...	applicatio...	firefo...
6...	200	HTTPS	fin1.ucce.local:...	/http-bind/	57		text/xml...	firefo...
6...	200	HTTPS	fin1.ucce.local:...	/http-bind/	57		text/xml...	firefo...
6...	200	HTTPS	fin1.ucce.local:...	/finesse/api/SystemI...	232	no-ca...	applicatio...	firefo...
6...	200	HTTPS	fin1.ucce.local:...	/http-bind/	57		text/xml...	firefo...
6...	-	HTTPS	fin1.ucce.local:...	/http-bind/	-1			firefo...
6...	200	HTTPS	fin1.ucce.local:...	/finesse/api/SystemI...	232	no-ca...	applicatio...	firefo...

Corpo da resposta para a mensagem XMPP:


```
<body xmlns='http://jabber.org/protocol/httpbind'><message xmlns="jabber:client" from="pubsub.fin1.ucce.local"
to="47483648@fin1.ucce.local" id="/finesse/api/User/47483648__47483648@fin1.ucce.local__K7hYF"><event
xmlns="http://jabber.org/protocol/pubsub#event"><items node="/finesse/api/User/47483648"><item id="26a3e421-
4752-8a1d-5adbdac74a7717"><notification xmlns="http://jabber.org/protocol/pubsub">&lt;Update&gt;
 &lt;data&gt;
  &lt;user&gt;
   &lt;dialogs&gt;/finesse/api/User/47483648/Dialogs&lt;/dialogs&gt;
   &lt;extension&gt;10005&lt;/extension&gt;
   &lt;firstName&gt;Isaac&lt;/firstName&gt;
   &lt;lastName&gt;Newton&lt;/lastName&gt;
   &lt;loginId&gt;47483648&lt;/loginId&gt;
   &lt;loginName&gt;isaac&lt;/loginName&gt;
   &lt;mediaType&gt;1&lt;/mediaType&gt;
   &lt;pendingState&gt;&lt;/pendingState&gt;
   &lt;roles&gt;
    &lt;role&gt;Agent&lt;/role&gt;
   &lt;/roles&gt;
   &lt;settings&gt;
    &lt;wrapUpOnIncoming&gt;OPTIONAL&lt;/wrapUpOnIncoming&gt;
   &lt;/settings&gt;
   &lt;state&gt;READY&lt;/state&gt;
   &lt;stateChangeTime&gt;2019-01-11T23:56:54.783Z&lt;/stateChangeTime&gt;
   &lt;teamId&gt;5000&lt;/teamId&gt;
   &lt;teamName&gt;Maths&lt;/teamName&gt;
   &lt;uri&gt;/finesse/api/User/47483648&lt;/uri&gt;
  &lt;/user&gt;
 &lt;/data&gt;
 &lt;event&gt;PUT&lt;/event&gt;
 &lt;requestId&gt;07f14a42-6b3c-4855-a4c9-af50ab5e7cc6&lt;/requestId&gt;
 &lt;source&gt;/finesse/api/User/47483648&lt;/source&gt;
&lt;/Update&gt;</notification></item></items></event></message></body>
```


Usar o Wireshark

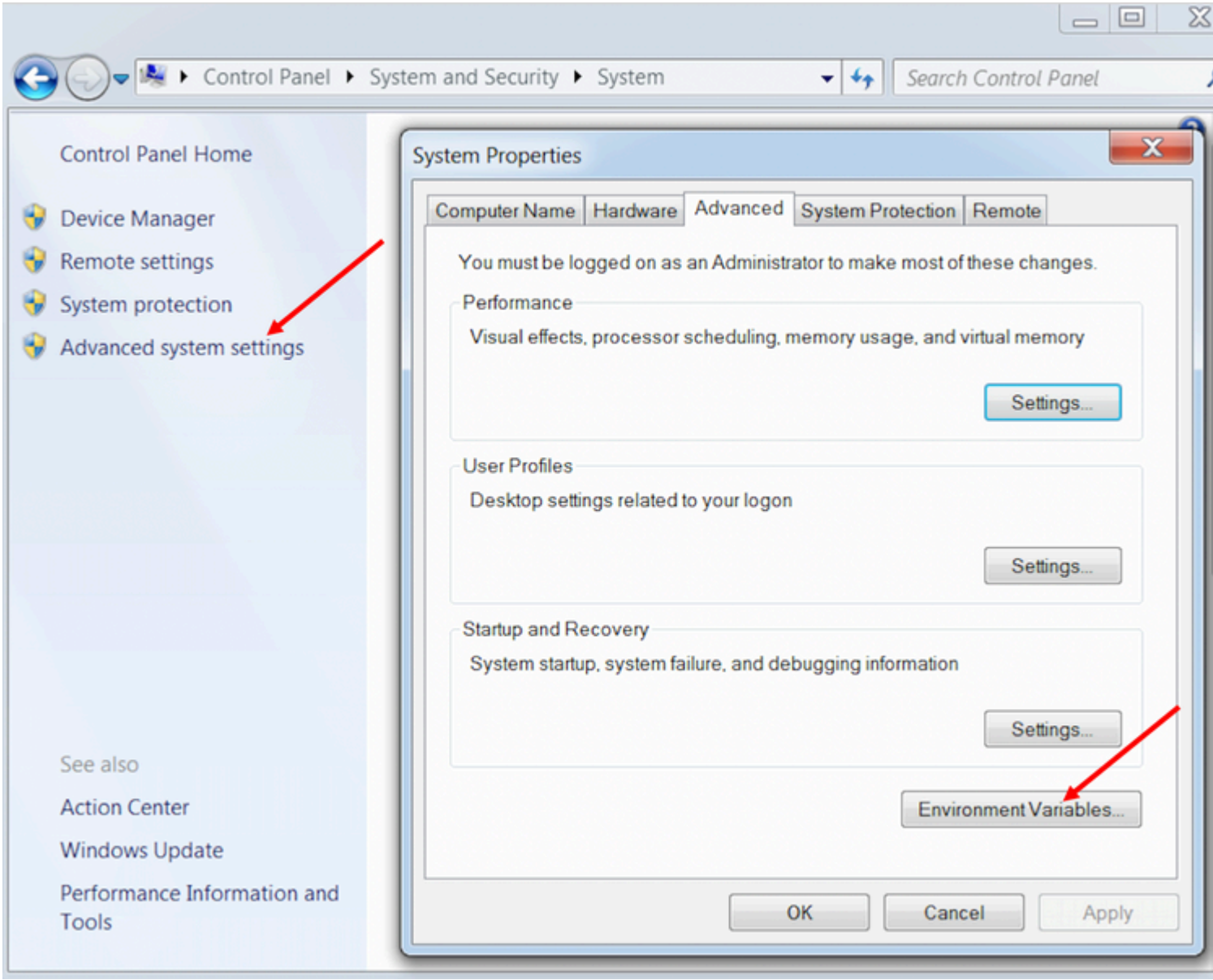
O Wireshark é uma ferramenta de detecção de pacotes comumente usada que pode ser usada para detectar e decodificar o tráfego HTTPS. O tráfego HTTPS é o tráfego HTTP protegido por Transport Layer Security (TLS). O TLS fornece integridade, autenticação e confidencialidade entre dois hosts. Ele é usado

comumente em aplicações Web, mas pode ser usado com qualquer protocolo que use TCP como o protocolo da camada de transporte. A SSL (Secure Sockets Layer) é a versão anterior do protocolo TLS, que não é mais usada por ser insegura. Esses nomes são frequentemente usados como sinônimos, e o filtro do Wireshark usado para tráfego SSL ou TLS é ssl.

> **Cuidado**: a configuração de exemplo fornecida é para o Wireshark 2.6.6 (v2.6.6-0-gdf942cd8) e Mozilla Firefox 64.0.2 (32 bits) no Windows7 x64 em um ambiente de laboratório. Esses procedimentos não podem ser generalizados para todas as versões do Fiddler, todos os navegadores ou todos os sistemas operacionais do computador. Se sua rede estiver ativa, certifique-se de que você compreende o impacto potencial de qualquer configuração. Consulte a documentação oficial do Wireshark SSL para obter mais informações. O Wireshark 1.6 ou superior é necessário.

> **Observação**: esse método só pode funcionar para Firefox e Chrome. Este método não funciona para o Microsoft Edge.

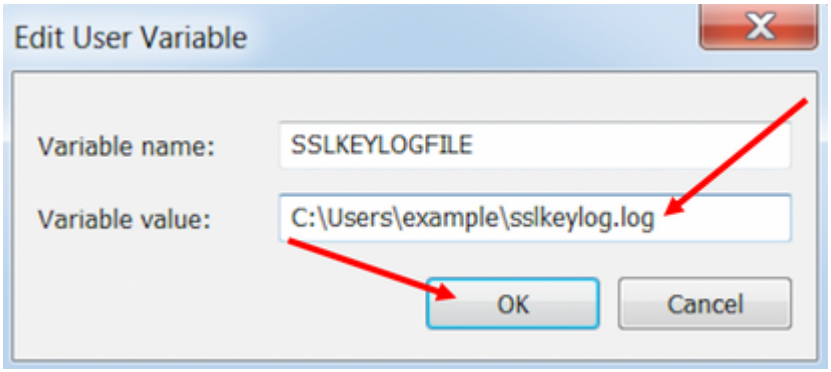
Etapa 1. No PC com Windows do agente, navegue para **Painel de Controle > Sistema e Segurança > Sistema > Configurações avançadas do sistema Variáveis de ambiente...**



Etapa 2. Navegue até **Variáveis de usuário <username> > Novo...**

Crie uma variável chamada **SSLKEYLOGFILE.**

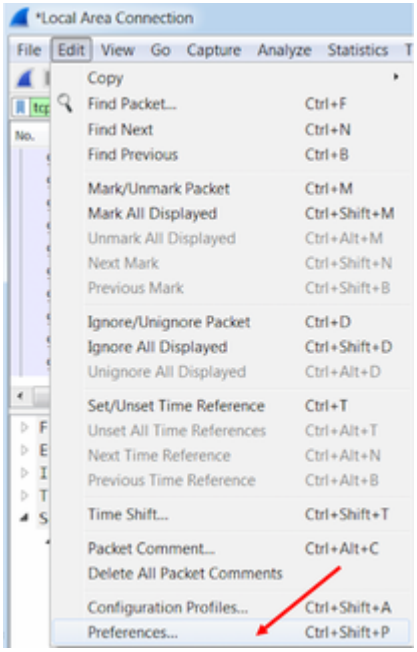
Crie um arquivo para armazenar o segredo do pré-mestre SSL em um diretório particular: SSLKEYLOGFILE=</path/to/private/diretory/with/logfile>



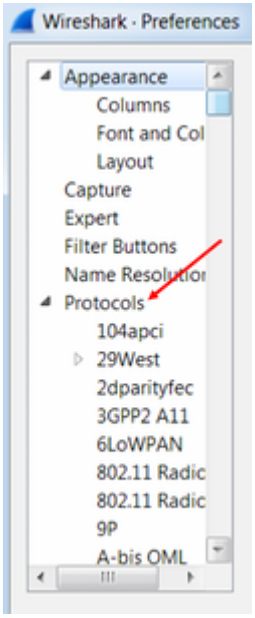
> **Observação**: crie uma variável de sistema em vez de uma variável de usuário e/ou armazene o arquivo em um diretório não privado, mas todos os usuários do sistema poderão acessar o segredo do pré-mestre, que é menos seguro.

Etapa 3. Se o Firefox ou o Chrome estiverem abertos, feche os aplicativos. Após serem reabertos, eles podem começar a gravar no SSLKEYLOGFILE.

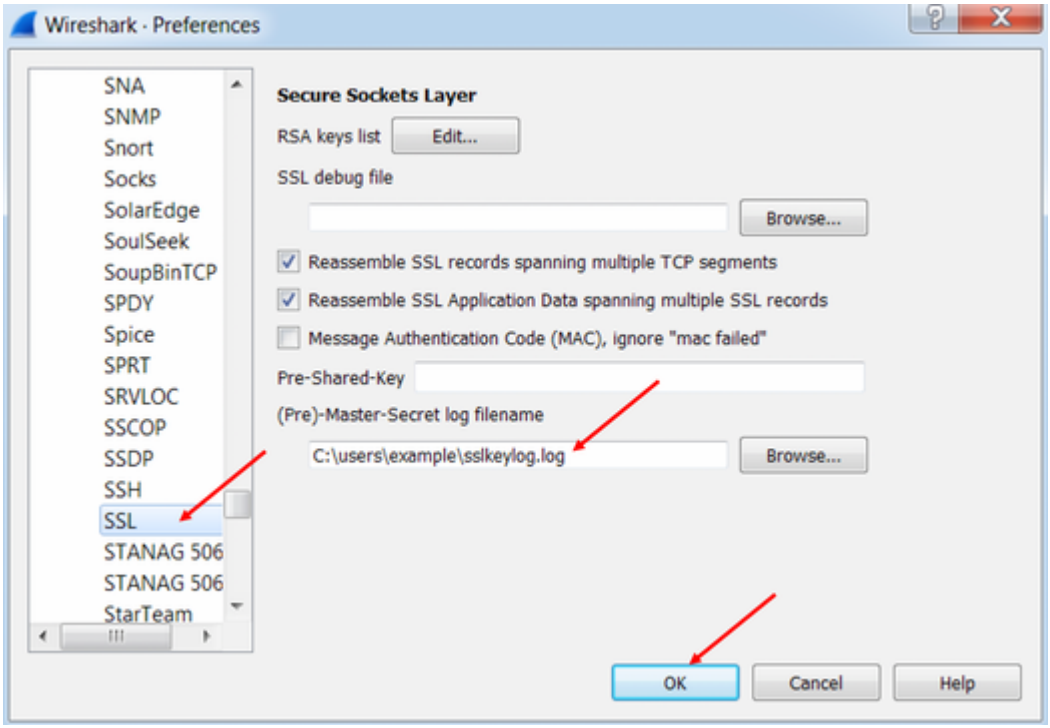
Etapa 4. No Wireshark, navegue para **Editar > Preferências...**



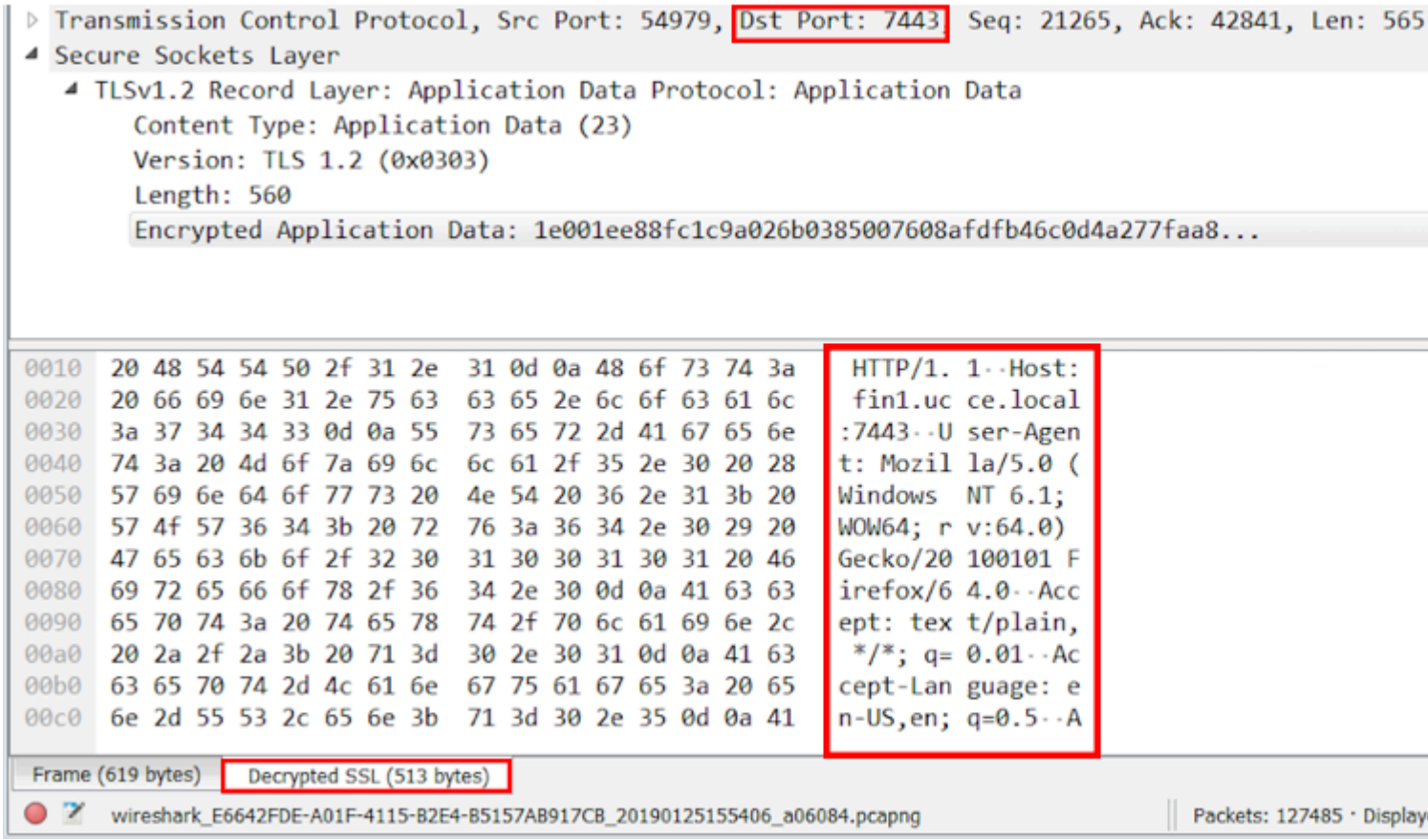
Navegue até **Protocolos > SSL**.



Etapa 5. Digite o local do nome do arquivo de log do segredo do pré-mestre configurado na Etapa 2.



Etapa 6. Use o filtro do Wireshark **tcp.port==7443 && ssl**, a comunicação HTTP segura entre o cliente Finesse e o servidor Finesse (Serviço de Notificação) é vista descriptografada.



Defeitos relacionados

- ID de bug Cisco CSCva7280 - Tomcat Finesse e travamento Openfire para caracteres XML inválidos
- ID de bug Cisco CSCva72325 - UCCX: Finesse Tomcat e Openfire Crash para caracteres XML inválidos

Informações Relacionadas

- Especificações XMPP
- XEP-0124: BOSH
- XEP-0060: Publicar-assinar
- Console Web Firefox
- Console da Web do Microsoft Edge
- Console Web do Chrome
- Windows PowerShell
- Monitor de Desempenho do Windows
- Troubleshooting de Quedas de Fila de Entrada e Quedas de Fila de Saída
- Gerenciador de Tarefas do Windows
- Terminal Mac
- Monitor de Atividade Mac
- Download do Fiddler
- Configuração do Fidler
- Download do Wireshark
- Descriptografia SSL do Wireshark
- Suporte Técnico e Documentação - Cisco Systems

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.