Finesse BOSHの実装の理解とトラブルシューティング

内容

はじめに

このドキュメントでは、BOSHを使用するFinesse接続の背後にあるアーキテクチャと、BOSH接続の問題を診断する方法について説明します。

前提条件

要件

次の項目に関する知識があることが推奨されます。

- Cisco Finesse
- Unified Contact Center Enterprise（UCCE）
- Unified Contact Center Express（UCCx）
- Webブラウザ開発者ツール
- Windowsおよび/またはMacの管理

使用するコンポーネント

このドキュメントの情報は、次のソフトウェアとハードウェアのバージョンに基づいています。

- Cisco Finesse 9.0(1) - 11.6(1)
- UCCX 10.0(1) - 11.6(2)

このドキュメントの情報は、特定のラボ環境にあるデバイスに基づいて作成されました。このドキュメントで使用するすべてのデバイスは、クリアな（デフォルト）設定で作業を開始しています。本稼働中のネットワークでは、各コマンドによって起こる可能性がある影響を十分確認してください。

背景説明

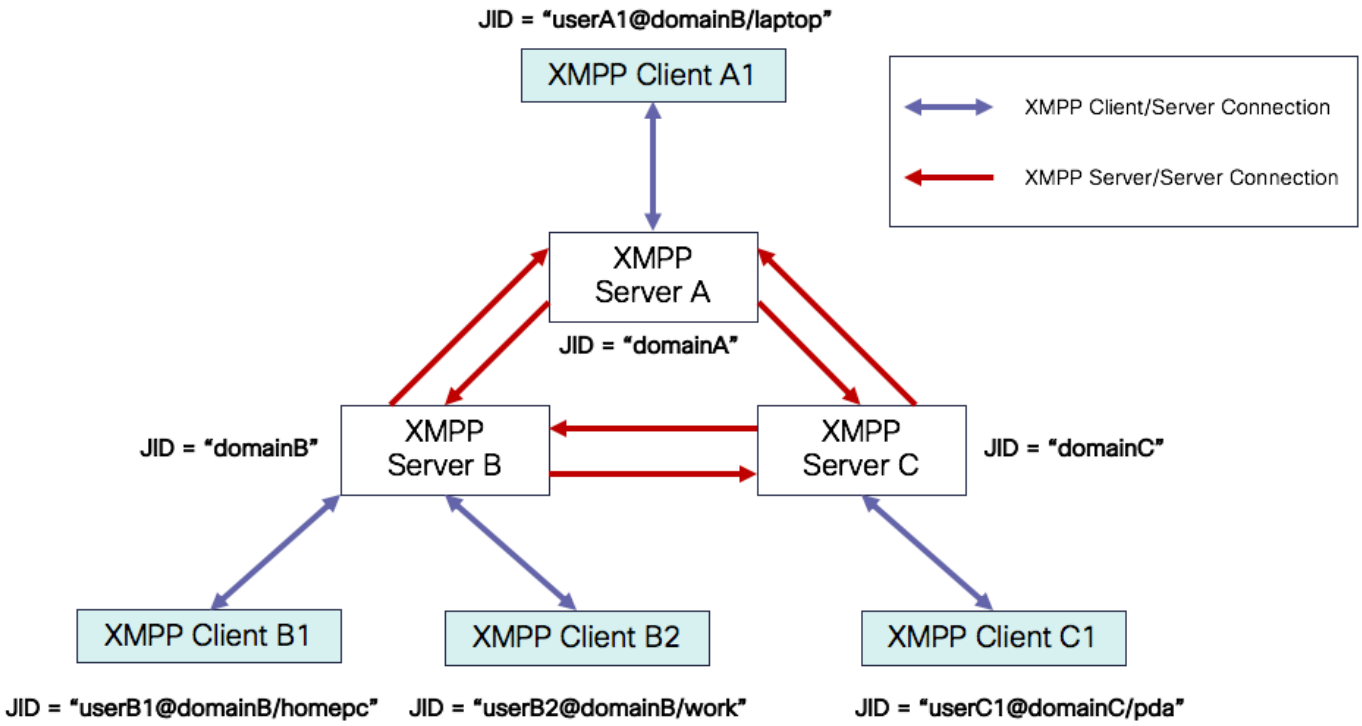
同期HTTPで双方向ストリームを使用する接続は、BOSHと呼ばれます。

Finesse BOSHの実装について

XMPPについて

Extensible Messaging and Presence Protocol(XMPP)（Jabberとも呼ばれる）は、クライアントサーバモデルのステートフルプロトコルです。XMPPを使用すると、構造化されたeXtensible Markup Language(XML)データの小片を、あるエンティティから別のエンティティに迅速に配信できます。XMPP/Jabberは、インスタントメッセージング(IM)およびプレゼンスアプリケーションで広く使用されています。

すべてのXMPPエンティティは、Jabber ID(JID)で識別されます。



JIDアドレッシング方式：user@domain/resource

ユーザ	XMPPサーバ上のクライアントユーザ名または会議室の名前
domain	XMPPサーバの完全修飾ドメイン名(FQDN)
リソース	ユーザの特定のエンティティ/エンドポイント（ラップトップ、スマートフォンなど）、セッション識別子、またはpubsubノード名の識別子

注:3つのJIDコンポーネントすべてが、すべての場合に使用されるわけではありません。通常、サーバはドメインによって、会議室はuser@domainによって、クライアントはuser@domain/resourceによって定義されます。

XMPPメッセージはスタンザと呼ばれます。XMPPには、次の3つのコアスタンザがあります。

1. <message>：一方向、1受信者

2. <presence>：一方向、複数に公開

3. <iq>: info/query – 要求/応答

すべてのスタンザはtoアドレスとfromアドレスを持ち、ほとんどのスタンザはtype、id、およびxml:langattributesも持っています。

スタンザ属性	目的

から	宛先JID
変更前	ソースJID
種類	メッセージの目的
[id]	<iq>スタンザの応答と要求をリンクするために使用される一意識別子です
xml：言語	スタンザ内の人間が読めるXMLのデフォルト言語を定義します

XMPPメッセージの例

```
<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>
```

FinesseによるXMPPの実装

WebアプリケーションをXMPPと連動させる必要がある場合は、複数の問題が発生します。ブラウザはTransmission Control Protocol（TCP；伝送制御プロトコル）を介したXMPPをネイティブでサポートしていないため、すべてのXMPPトラフィックは、ブラウザ内で実行されるプログラムで処理する必要があります。WebサーバとブラウザはHyperText Transfer Protocol(HTTP)メッセージを介して通信するため、Finesseおよびその他のWebアプリケーションはXMPPメッセージをHTTPメッセージの内部にラップします。

このアプローチの最初の難点は、HTTPがステートレスプロトコルであることです。 これは、各HTTP要求が他の要求と関連していないことを意味します。ただし、この問題は、クッキー/ポストデータの使用など、適切な方法で対処できます。

2つ目の問題は、HTTPの単方向の動作です。クライアントだけが要求を送信し、サーバは応答できるだけです。サーバがデータをプッシュできないため、HTTP経由でXMPPを実装するのは不自然です。

この問題は、XMPPがTCPにバインドされている元のXMPPコア仕様(RFC 6120)には存在しません。ただし、HTTPにバインドされたXMPPの問題に対処する場合、たとえばJavascriptがHTTP要求を送信できるため、2つの解決策が考えられます。両方とも、HTTPとXMPPの間にブリッジが必要です。

提案するソリューションは次のとおりです。

1.ポーリング（レガシープロトコル）:XEP-0025で定義されている新しいデータを要求するHTTP要求が繰り返される：Jabber HTTPポーリング

2.ロングポーリングはBOSHとも呼ばれます。XEP-0124: HTTPバインディングで定義され、XEP-0206: XMPP Over BOSHで拡張された頻繁なポーリングを使用せずに、複数の同期HTTP要求/応答ペアを効率的に使用して、2つのエンティティ間の長期双方向TCP接続のセマンティクスをエミュレートする転送プロトコルです

FinesseはBOSHを実装しています。これは、サーバ負荷の観点から見て非常に効率的であり、トラフィックの観点からも同様です。BOSHを使用する理由は、要求があるとすぐにサーバが応答する必要がないという事実を隠すことです。応答は、サーバがクライアントのデータを持つまで指定された時間まで遅延され、応答として送信されます。クライアントは、応答を受け取るとすぐに新しい要求を行います。

Finesseデスクトップクライアント（Webアプリケーション）は、30秒ごとにTCPポート7443経由で古いBOSH接続を確立します。30秒後、Finesse Notification Serviceからのアップデートがない場合、通知サービスは200 OKと（ほぼ）空の応答本文を含むHTTP応答を送信します。エージェントの存在やダイアログ（コール）イベントなどの更新が通知サービスにある場合、データはFinesse Webクライアントにただちに送信されます。

Finesse XMPP要求/応答の例

この例では、BOSH接続をセットアップするために、FinesseクライアントとFinesseサーバの間で共有される最初のXMPPメッセージ要求応答を示します。

```
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" authi
```

まとめ

1. Finesse Webクライアントに、TCPポート7443経由でFinesseサーバへの古いHTTP接続(http-bind)が設定されています。これはBOSHロングポーリングと呼ばれます。

2. Finesse Notification Serviceは、エージェントの状態やコールなどに関する更新情報をポストするプレゼンスサービスです。

3. 通知サービスに更新がある場合、HTTP応答の本文にXMPPメッセージとして状態更新を含むhttp-bind要求に応答します。

4. http-bind要求を受信して30秒後に状態更新がない場合、通知サービスは状態更新なしで応答し、Finesse Webクライアントが別のhttp-bind要求を送信できるようにします。これは、Finesse Webクライアントが引き続きNotification Serviceに接続できること、エージェントがブラウザを閉じたりコンピュータをスリープ状態にしたりしなかったことを通知サービス

が認識するための手段となります。

Finesse XMPPメッセージとXMPPノードについて

Finesseは、XMPP仕様XEP-0060: Publish-Subscribeも実装しています。この仕様の目的は、XMPPサーバ（通知サービス）がXMPPノード（トピック）に公開された情報を取得し、ノードにサブスクライブされたエンティティにXMPPイベントを送信できるようにすることです。Finesseの場合、コンピュータテレフォニーインテグレーション(CTI)サーバはCTIメッセージをFinesse Webサービスに送信して、エージェントやコンタクトサービスキュー(CSQ)の作成などの設定の更新やコールに関する情報をFinesseに通知します。この情報は、Finesse WebサービスがFinesse Notificationサービスに発行するXMPPメッセージに変換されます。次に、Finesse Notificationサービスは、特定のXMPPノードにサブスクライブされているエージェントにXMPP over BOSHメッセージを送信します。

『Finesse Webサービス開発者ガイド』で定義されているFinesse APIオブジェクトの一部はXMPPノードです。エージェントおよびスーパーバイザのFinesse Webクライアントは、これらのXMPPノードの一部のイベント更新にサブスクライブして、リアルタイムイベント（コールイベント、状態イベントなど）に関する最新情報を取得できます。次の表に、pubsubが有効になっているXMPPノードを示します。

Finesse APIオブジェクト	目的	購読
/finesse/api/User/<ログインID>	エージェントの状態およびチームマッピングを表示します	エージェントとスーパーバイザ
/finesse/api/User/<ログインID>/ダイアログ	エージェントが処理したコールを表示します。	エージェントとスーパーバイザ
/finesse/api/User/<ログインID>/ClientLog	Send Error Reportボタンからクライアントログをキャプチャするために使用されます	エージェントとスーパーバイザ
/finesse/api/User/<LoginID>/Queue/<queueID>	キュー統計データを表示します（有効になっている場合）	エージェントとスーパーバイザ
/finesse/api/Team/<TeamID>/Users	特定のチームに属するエージェントを状態情報を含めて表示します。	スーパーバイザ
/finesse/api/SystemInfo	Finesseサーバの状態を表示します。	エージェン

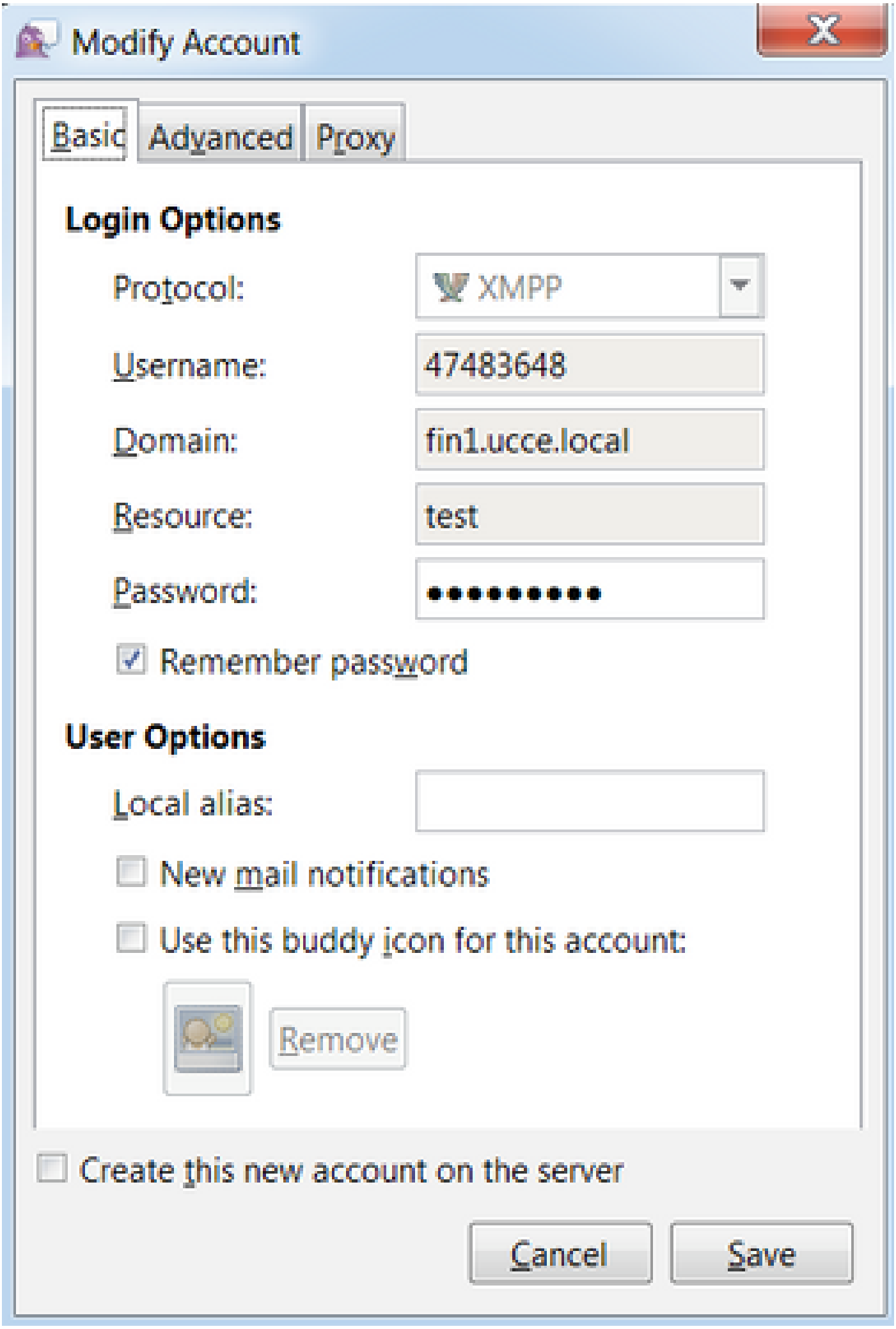
	フェールオーバーが必要かどうかを判断するために使用される	トとスーパーバイザ

例1:Pidginを使用したFinesse XMPPノードの表示

ステップ 1：XMPPクライアントPidginをダウンロードしてインストールします。

ステップ 2：Accounts > Modify > Basicの順に選択し、Login Optionsを設定します。

- プロトコル：XMPP
- ユーザ名：任意のエージェントのログインID
- ドメイン：FinesseサーバのFQDN
- リソース：プレースホルダー – 任意の値を使用できます (例：test)
- パスワード：エージェントパスワード
- Remember passwordチェックボックスにチェックマークを入れます


Modify Account
Basic
Advanced
Proxy
Login Options
Protocol:
XMPP
Username:
47483648
Domain:
fin1.ucce.local
Resource:
test
Password:
Remember password
User Options
Local alias:
New mail notifications
Use this buddy icon for this account:
Remove
Create this new account on the server
Cancel
Save

```
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
```

Webサービスログ

これらのログは/desktop/logs/webservicesフォルダにあり、Desktop-webservices.YYYY-MM-DDTHH-MM-SS.sss.logという名前が付けられています。 Finesseクライアントが指定された時間内にFinesseサーバにhttp-bindメッセージを送信しない場合、ログにはエージェントプレゼンスが使用不能になり、60秒後にプレゼンス駆動ログアウトが発生することが示されます。

```
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-UNSUBSCR
0000001044: XX.XX.XX.XXX:  Jun 17 2017 00:16:04.631 +0530: %CCBU_Smack Listener Processor (1)-6-AGENT_P
0000001051: XX.XX.XX.XXX:  Jun 17 2017 00:16:35.384 +0530: %CCBU_pool-8-thread-1-6-AGENT_PRESENCE_MONIT
0000001060: XX.XX.XX.XXX:: Jun 17 2017 00:17:04.632 +0530: %CCBU_CoreImpl-worker12-6-PRESENCE DRIVEN LO
0000001061: XX.XX.XX.XXX:: Jun 17 2017 00:17:04.633 +0530: %CCBU_CoreImpl-worker12-6-MESSAGE_TO_CTI_SER
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_M
skillGroupNumber=-1, skillGroupPriority=0, agentState=1 (LOGOUT), eventReasonCode=255, numFltSkillGroup
duration=null, nextAgentState=null, fltSkillGroupNumberList=[], fltSkillGroupIDList=[], fltSkillGroupPr
msgID=30, timeTracker={"id":"AgentStateEvent","CTI_MSG_RECEIVED":1497638824642,"CTI_MSG_DISPATCH":14976
Decoded Message to Finesse from backend cti server
```

BOSH接続解除の一般的な原因

BOSH接続はWebクライアントによって設定され、Finesseサーバはエージェントのプレゼンスが利用できるかどうかを判断します。接続を開始する責任はクライアントにあるため、これらの問題は、ほとんどの場合、ブラウザ、エージェントコンピュータ、またはネットワークに関連するクライアント側の問題です。

問題：エージェントが異なる時間に接続解除する（クライアント側の問題）

推奨される対処法

次の問題を確認してください。

1. ネットワークの問題：

- ファイアウォールのルールとログを確認する：TCPポート7443をブロックまたは抑制してはならない
- Fiddler®やWireshark®などのHTTP Webトラフィックスニファを使用して、ブラウザがTCPポート7443経由でhttp-bind要求を送信し、応答を受信することを確認します
- エージェントコンピュータとFinesseサーバの間のすべてのネットワークデバイス/インターフェイスで、過度の遅延やパケットのドロップがないかどうかを確認します

    - tracerouteは、パスの判別と遅延の判別に役立ちます
        - Microsoft® Windows® PCの場合：tracert {Finesse Server IP | Finesseサーバ FQDN}
        - Macの場合®: traceroute {Finesse Sever IP | FinesseサーバFQDN}
        - Cisco IOS®ソフトウェアでは、インターフェイスの統計情報を確認できます。show interfaces
            - 「入力キュードロップと出力キュードロップのトラブルシューティング」を参照してください。

- テストエージェントのFinesseクライアントログを収集します。 クライアントログは、次の3つの方法で収集できます。
    1. ブラウザのWebコンソールログ
        - Firefox Webコンソール
        - Microsoft Edge Webコンソール
        - Chrome Webコンソール
    2. Finesseページのエラーレポートの送信ボタンを押して、Finesseサーバログを収集します。ログは/desktop/logs/clientlogsにあります。
    3. https://<Finesse-FQDN>/desktop/locallogを使用してログインし、問題発生後にログを収集します。

クライアントは1分ごとにFinesseサーバに接続し、ドリフトとネットワーク遅延を計算します。

```
<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): <RT

2019-01-11T12:24:14.586 -05:00: : fin1.ucce.local: Jan 11 2019 11:24:14.577 -0600: Header : Client: 201
```

ログ収集の問題が発生した場合は、「Cisco Finesseデスクトップ持続ロギング問題のトラブルシューティング」を参照してください。

2.サポートされていないブラウザまたはバージョン：

互換性マトリクスに従って、サポートされているブラウザ/バージョンと設定を使用します。

UCCE互換性マトリックス

UCCX互換性マトリックス

3. 他のタブ/ウィンドウの内容/処理によるブラウザ停止状態：

エージェントワークフローをチェックして、次の点を確認します。

- 一般に、音楽/ビデオストリーミング、WebSocket接続、カスタムCustomer Relationship Management(CRM)Webクライアントなど、他のリアルタイムアプリケーションを常に実行するその他のタブまたはウィンドウが表示されます。
- 非常に多数のタブまたはウィンドウを開いている
- ブラウザのキャッシュを無効にしている

- ブラウザを長時間実行し続け、営業日の終了時にブラウザを閉じないでください

4.コンピュータのスリープ状態：

エージェントがFinesseからログアウトする前にコンピュータをスリープ状態にするかどうか、またはコンピュータのスリープ設定タイマーが非常に低いかどうかを確認します。

5.クライアントコンピュータのCPU高使用率またはメモリ高使用率の問題：

- Microsoft Windows Remote Desktop Services、Citrix® XenApp®、Citrix XenDesktop®などの共有環境でエージェントブラウザを実行する場合は、ブラウザを同時に実行するユーザ数によってブラウザのパフォーマンスが異なるかどうかを確認します
    - ユーザ数に基づいて適切なメモリおよびCPUリソースが設定されていることを確認します

- コンピュータリソースの使用率の問題を確認します。
    - Windows：
        - CPU時間の%、使用可能なメモリのMB数、および使用中のメモリの%を2秒ごとに確認するWindows PowerShell Get-Counterコマンド： Get-Counter -Counter "\Processor(_Total)\% Processor Time","\Memory\Available MBytes","\Memory\% Committed Bytes In Use" -SampleInterval 2 -Continuous
        - PowerShellを使用してWindowsのパフォーマンスカウンタを表示する代わりに、Windows Performance Monitorを使用できます
        - Task Managerを使用すると、稼働中のCPUおよびメモリの統計情報をグローバルに、プロセスごとに表示できます
    - MAC：
        - CPUとメモリのライブ合計をチェックするTerminal Topコマンド：top
            - プロセスを確認し、CPU使用率で並べ替え：上位CPU数
            - プロセスを確認し、メモリ使用率でソート：top -o MEM
        - Activity Monitorを使用すると、稼働中のCPUおよびメモリの統計情報をグローバルに、プロセスごとに表示できます

6. 予期しない、問題のあるアクティビティをバックグラウンドで実行しているサードパーティガジェット：

すべてのサードパーティガジェットを削除して、Finesseデスクトップの動作をテストします。

7.サーバまたはクライアントのNTPの問題：

- Finesseパブリッシャサーバのutils ntp statusをチェックして、NTPサーバストラタムが4以下であることを確認します
- クライアントログで、ドリフトとネットワーク遅延を確認します

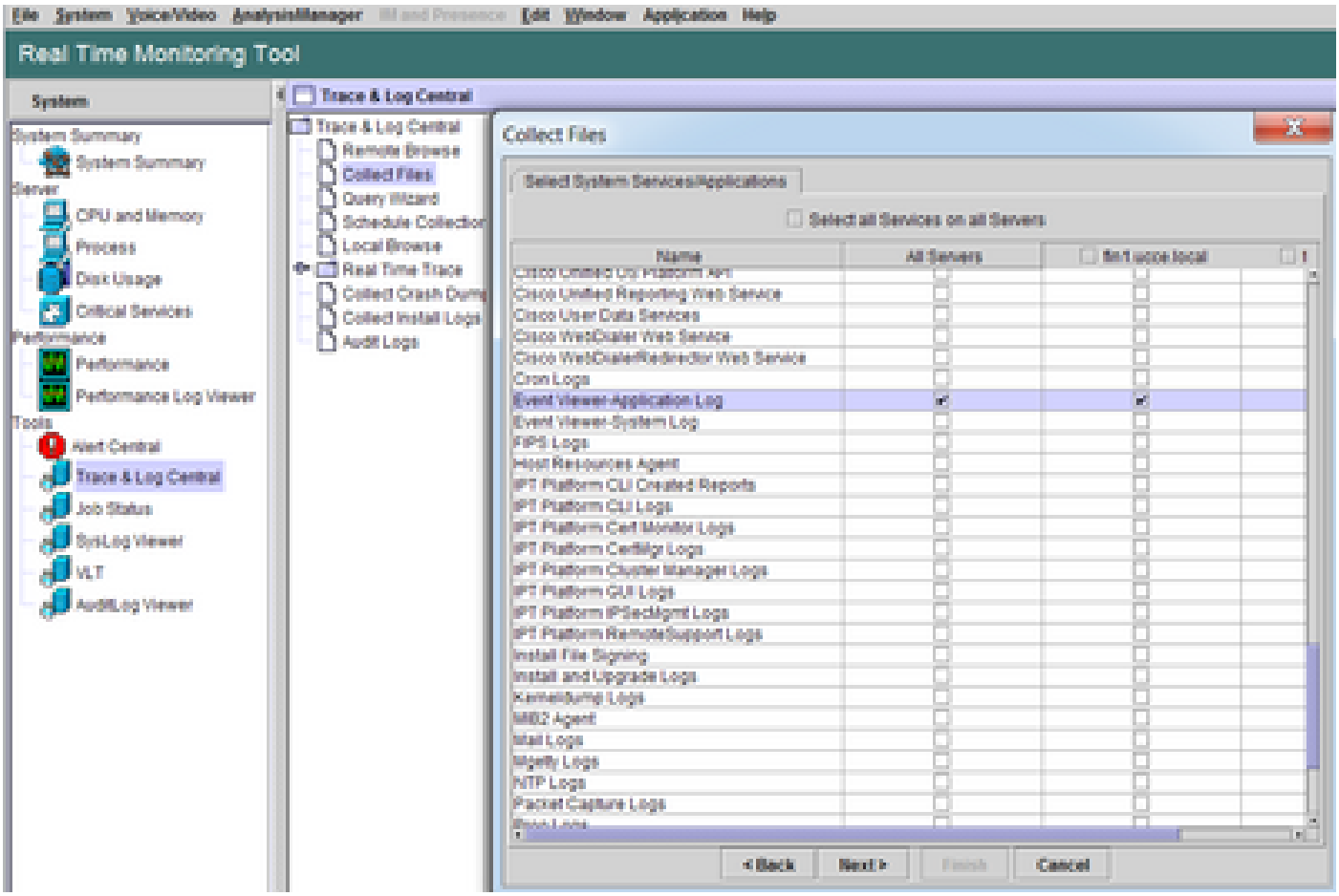
問題：すべてのエージェントが同時に接続解除する（サーバ側の問題）

推奨される対処法

次の問題を確認してください。

1. Cisco Unified Communications Manager CTIManagerサービスが切断されます。 UCCXのすべてのCTIManagerプロバイダーがシャットダウンまたはクラッシュしている場合、UCCXエージェントには赤いバナーエラーが表示されます。これが発生した場合、UCCEエージェントには赤いバナーが表示されませんが、コールはエージェントに正しくルーティングされません。

- CTIプロバイダーとして使用されているCUCMサーバでCisco CTIManagerサービスが開始されているかどうかを確認します
- Cisco CTIManagerサービスがクラッシュしたかどうかを確認するには、イベントビューア：アプリケーションがRTMTにログインして、Cisco CTIManagerサービスがクラッシュしたかどうかを調べます
    - RTMTでイベントビューアログを収集するには、System > Tools > Trace and Log Central > Collect Files > Select System Services/Applications > Event Viewer-Application Logの順に選択します。



- CLIでイベントビューアアプリケーションログを収集するには、次のコマンドを実行します。file get activelog /syslog/CiscoSyslog* abstime hh:mm:MM/DD/YY hh:mm:MM/DD/YY
- CLIでコアダンプを表示するには、utils core active listを使用します。

注：コアダンプファイル名は、core.<ProcessID>.<SignalNumber>.<ProcessName>.<EpochTime>の形式を使用します。
例： core.24587.6.CTIManager.1533441238
したがって、クラッシュの時間はエポック時間から決定できます。

---

2. Finesse/UCCX Notification Serviceが停止またはクラッシュしました。

- イベントビューアのアプリケーションログでNotification Serviceエラーを確認するか、サービスが停止したかどうかを確認します
- Notification serviceが稼働中かどうかを確認します：utils service list
- 通知サービスがシャットダウンした時刻を確認します：ファイル検索activelog /desktop/logs/openfire "Openfire stopped"
- 通知サービスが開始された時刻を確認します：ファイル検索activelog /desktop/logs/openfire "HTTPバインドサービスが開始されました"
- クラッシュによって発生した通知サービスのメモリダンプを確認します（ファイル一覧：activelog /desktop/logs/openfire/*.hprof）。
- NotificationサービスがTCPポート7443でトラフィックをリッスンしているかどうかを確認します。show open ports regexp 7443.*LISTEN
- 次の不具合が適用可能かどうかを確認します(これらの不具合は、エージェントがログインするときにログインに失敗し、ログイン済みのエージェントには赤色のバナーFinesse disconnectメッセージが表示されます)。
  - Cisco Bug ID CSCva72280:無効なXML文字のためのFinesse TomcatおよびOpenfireのクラッシュ
  - Cisco Bug ID CSCva72325:UCCX：無効なXML文字に対するFinesse TomcatおよびOpenfireのクラッシュ

クラッシュが疑われる場合は、Cisco Finesse Tomcat and Notification Serviceを再起動します。これは、ネットワークがダウンしている場合にのみ推奨されます。それ以外の場合は、Finesseサーバから切断エージェントが再起動されます。

UCCEの手順：

- utils service stop Cisco Finesse Tomcat
- utils service stop Cisco Finesse Notification Service
- utils service start Cisco Finesse Tomcat（Cisco Finesse Tomcatの起動）
- utils service start Cisco Finesse Notification Service(utils service start Cisco Finesse Notification Service)

UCCXの手順：

- utils service stop Cisco Finesse Tomcat
- utils service stop Cisco Unified CCX Notification Service
- utils service start Cisco Finesse Tomcat（Cisco Finesse Tomcatの起動）
- utils service start Cisco Unified CCX Notification Service

バイオリンを使う

Fiddlerの設定は、必要な手順を理解し、Fiddlerの動作の仕組みを理解していなければ、やや困難な作業になる可能性があります。Fiddlerは、Finesseクライアント（Webブラウザ）とFinesseサーバの間に位置するman-in-the-middle Webプロキシです。FinesseクライアントとFinesseサーバの間の接続が保護されているため、保護されたメッセージを表示するためにFiddlerの設定が複雑になります。

よくみられるバイオリン問題

FiddlerはFinesseクライアントとFinesseサーバの間に位置するため、Fiddlerアプリケーションは、証明書を必要とするすべてのFinesse TCPポートに署名付き証明書を作成する必要があります。

Cisco Finesse Tomcatサービス証明書

1. Finesseパブリッシャサーバ TCP 8445（UCCEの場合は443）
2. Finesseサブスクライバサーバ TCP 8445（UCCEの場合は443）

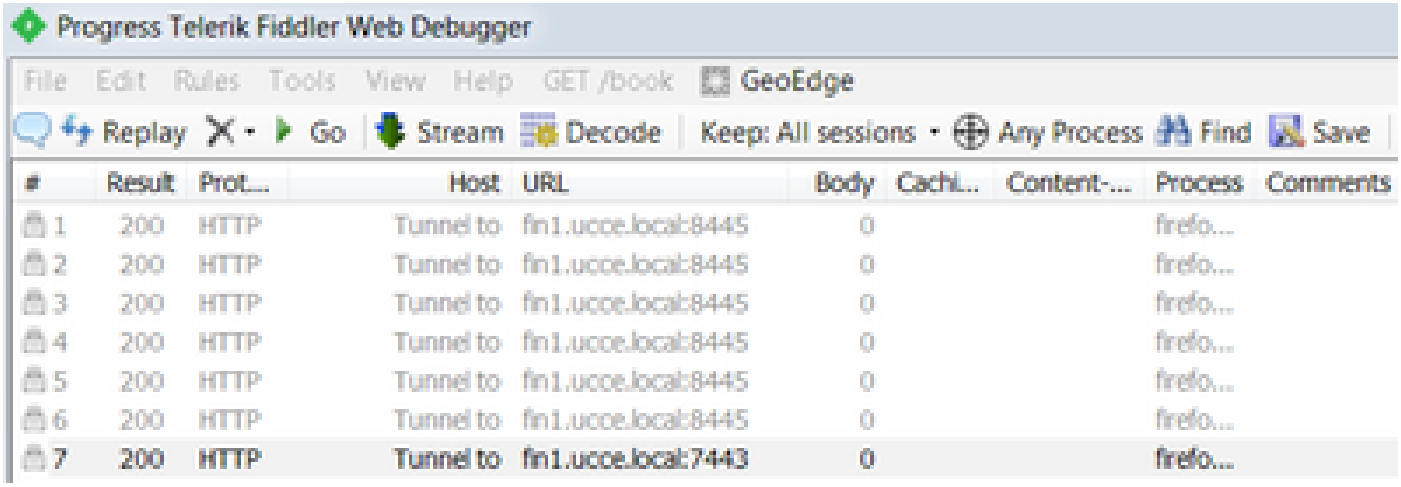
Cisco Finesse(Unified CCX)Notification Service証明書

1. Finesseパブリッシャサーバ TCP 7443
2. Finesseサブスクライバサーバ TCP 7443

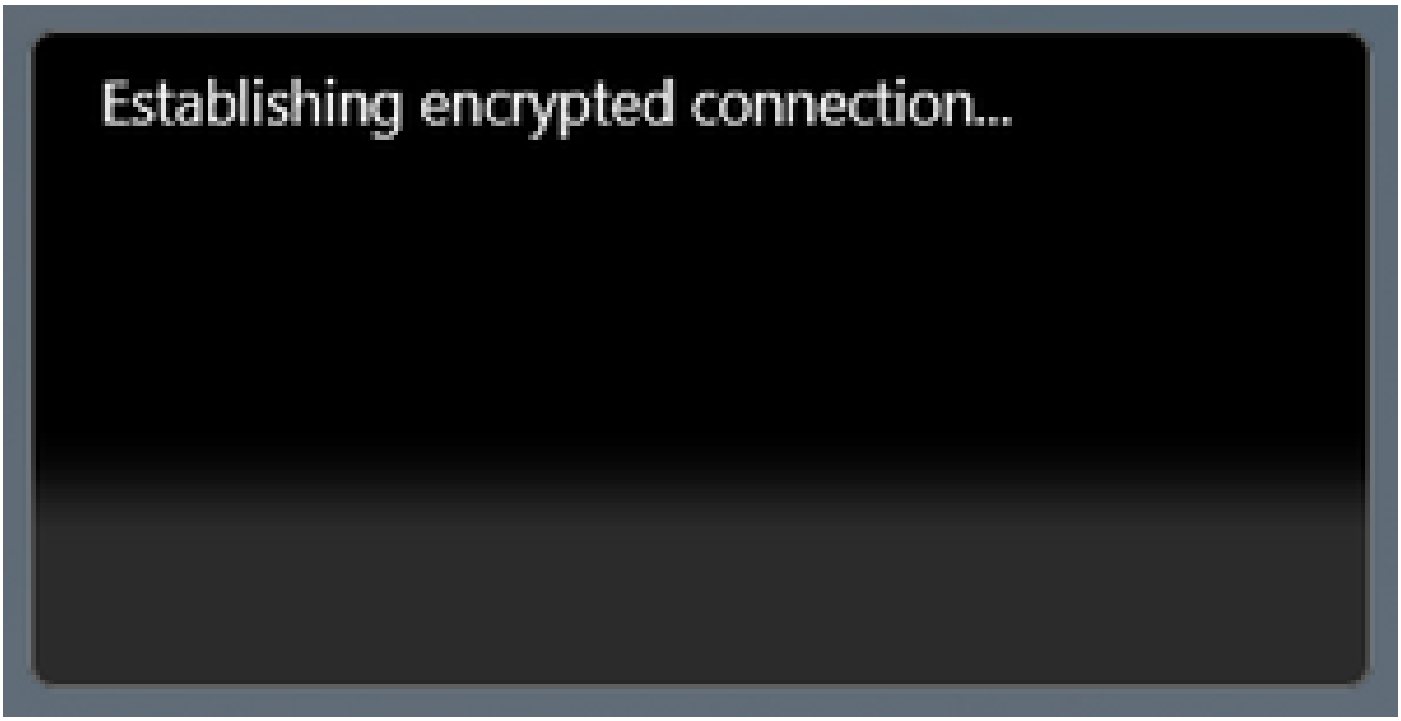
Finesseサーバの代わりにFiddlerが証明書を動的に生成するには、HTTPS復号化を有効にする必要があります。これはデフォルトでは有効になっていません。

HTTPS復号化が設定されていない場合、通知サービスへの最初のトンネル接続は表示されますが、http-bindトラフィックは表示されません。Fiddlerは次の情報のみを表示します。

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



次に、Fiddlerによって署名されたFinesse証明書は、クライアントによって信頼される必要があります。これらの証明書が信頼されていない場合、Finesseログインの「暗号化接続の確立...」段階を超えて移行することはできません。



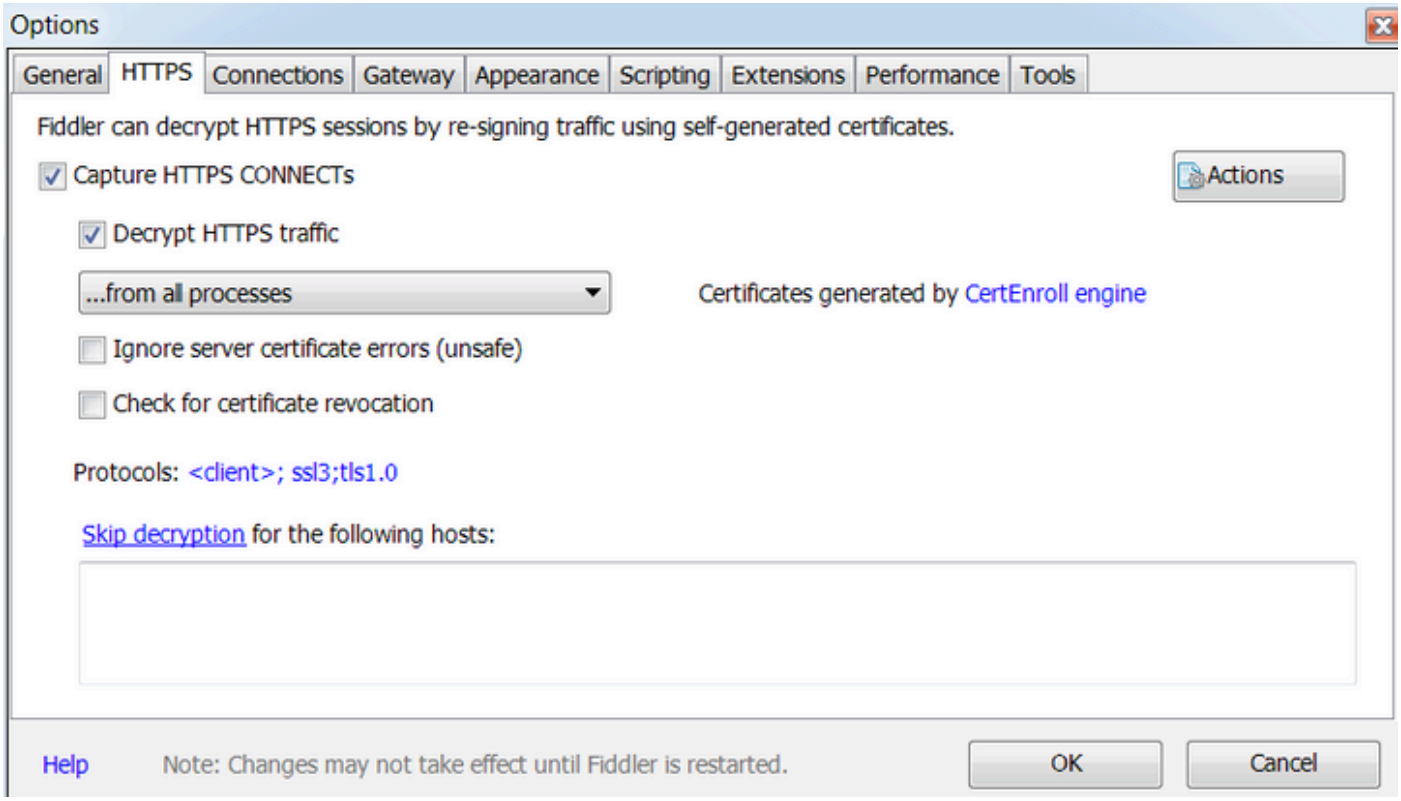
場合によっては、ログインから証明書の例外を受け入れることはできず、証明書はブラウザによって手動で信頼される必要があります。

設定手順の例

⚠ 注意：提供されている設定例は、ラボ環境のWindows 7 x64上の.NET 4.5およびMozilla Firefox 64.0.2（32ビット）用のFiddler v5.0.20182.28034用です。これらの手順は、Fiddlerのすべてのバージョン、すべてのブラウザ、またはすべてのコンピュータのオペレーティングシステムに一般化することはできません。 稼働中のネットワークで作業を行う場合は、どのような設定についても、その潜在的な影響について確実に理解しておく必要があります。詳細については、Fiddlerの公式ドキュメントを参照してください。

ステップ 1：Fiddlerのダウンロード

ステップ 2：HTTPS復号化を有効にします。Tools > Options > HTTPSの順に移動し、Decrypt HTTPS trafficチェックボックスにチェックマークを入れます。

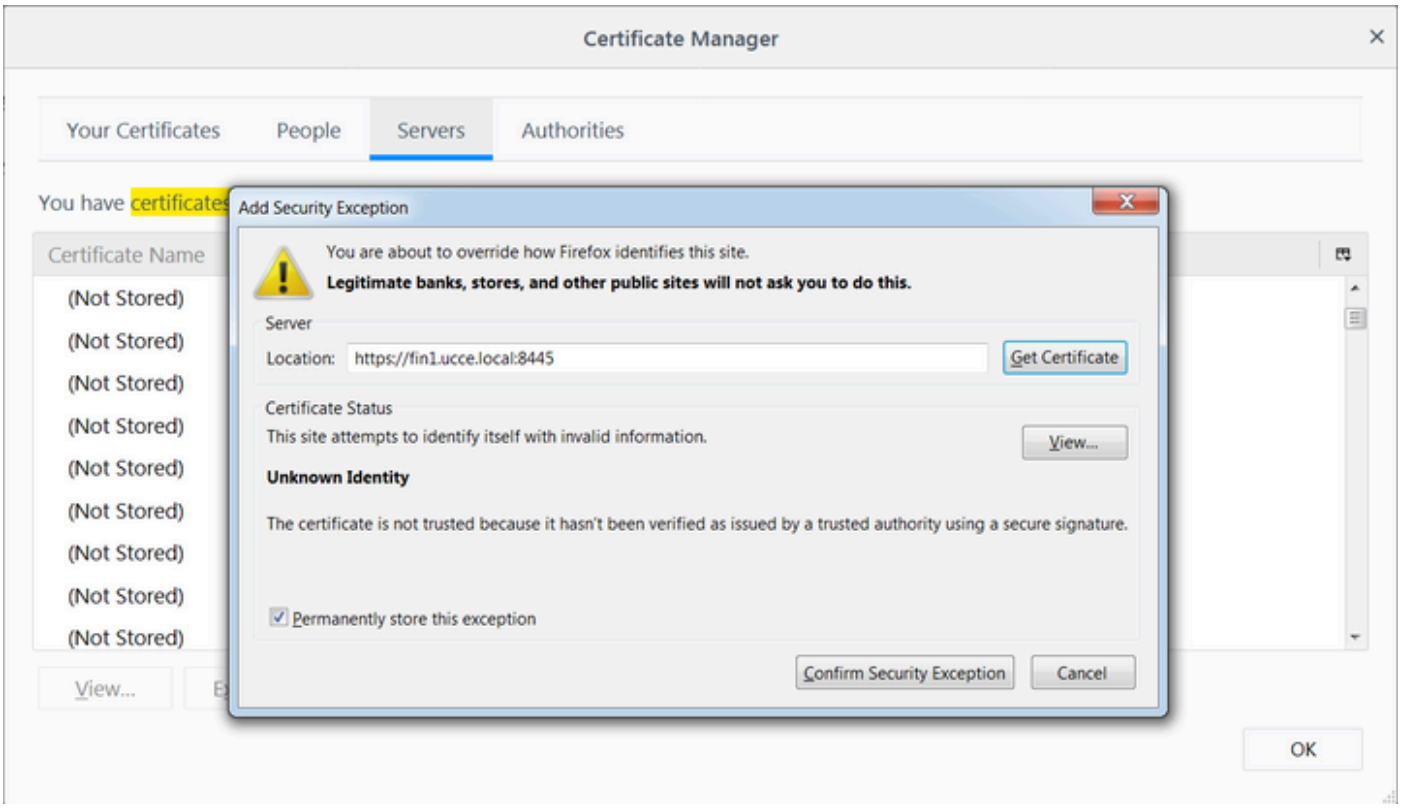


ステップ 3：警告メッセージボックスが開き、Fiddlerルート証明書を信頼するかどうかを尋ねられます。 [Yes] を選択します。

ステップ 4：警告メッセージボックスが開き、「You are about to install a certificate from a certification authority (CA) claiming to represent: DO_NOT_TRUST_FiddlerRoot...」というメッセージが表示されます。Do you want to install this certificate?」というメッセージが表示されます。[Yes] を選択します。

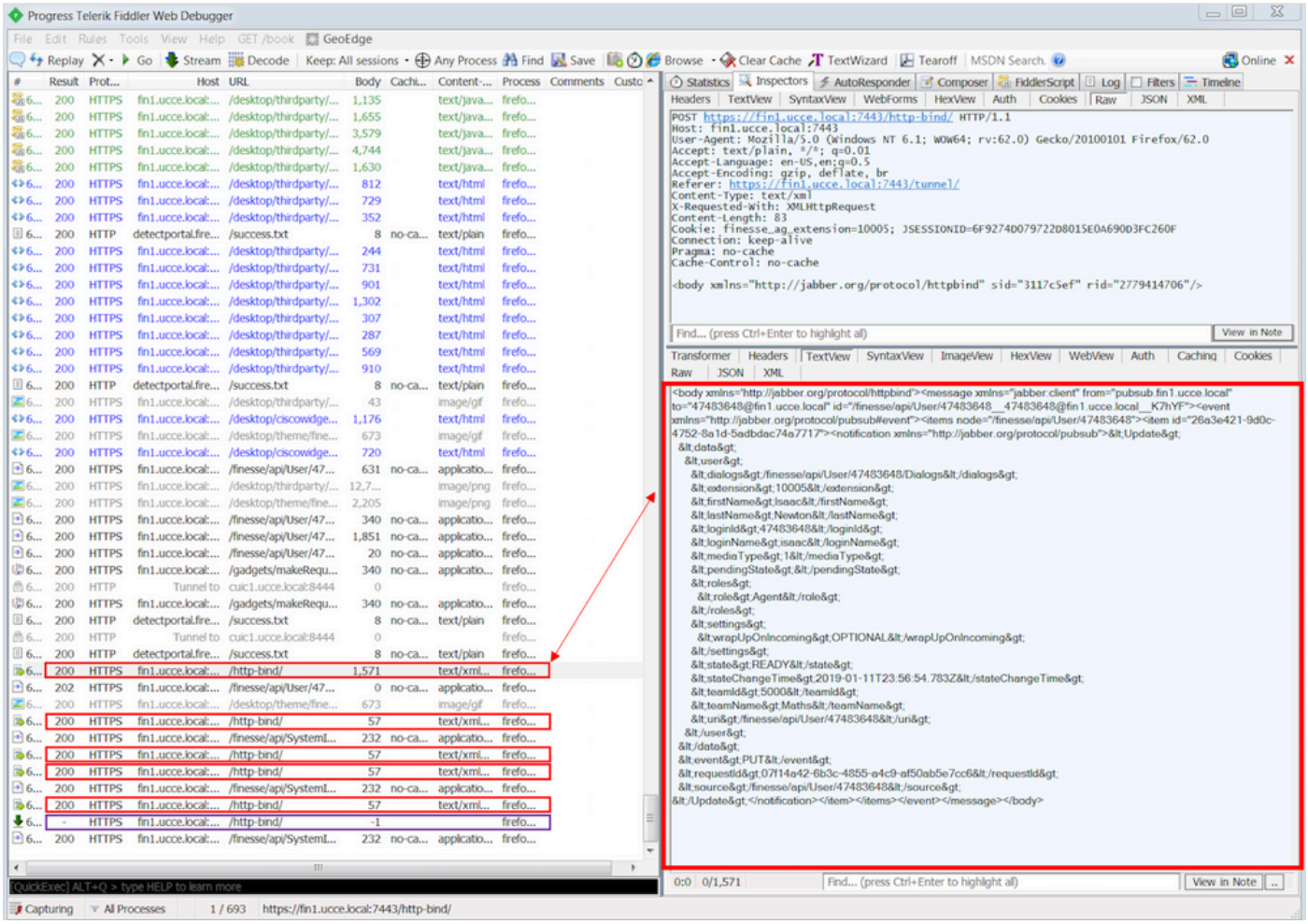
ステップ 5：Finesseパブリッシャおよびサブスクライバ証明書をコンピュータまたはブラウザの証明書信頼ストアに手動で追加します。ポート8445、7443、および（UCCEのみ）443を確認します。たとえば、Firefoxでは、Finesseオペレーティングシステム(OS)管理ページから証明書をダウンロードせずに簡単に実行できます。

オプション>オプションの検索（検索）>証明書>サーバ>例外の追加>ロケーション>両方のFinesseサーバの関連ポートにhttps://<Finesse server>:portと入力します。



手順 6：Finesseにログインし、FiddlerをFinesseクライアントからFinesseサーバに送信されるhttp-bindメッセージを確認します。

この例では、最初の5つのメッセージに、Finesseサーバが応答したhttp-bindメッセージが表示されています。最初のメッセージには、メッセージ本文に返された1571バイトのデータが含まれています。本文には、エージェントイベントに関するXMPP更新が含まれています。最後のhttp-bindメッセージはFinesseクライアントから送信されていますが、Finesseサーバから応答がありません。これは、HTTPの結果がヌル(-)で、応答本文のバイト数がヌル(-1)の場合に判別できます。



データの詳細ビュー：

XMPPメッセージの応答の本文：

<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="26a3e421-9d0c-4752-8a1d-5adbdac74a7717"><notification xmlns="http://jabber.org/protocol/pubsub"><Update>
<data>
<user>
<dialogs>/finesse/api/User/47483648/Dialogs</dialogs>
<extension>10005</extension>
<firstName>Isaac</firstName>
<lastName>Newton</lastName>
<loginId>47483648</loginId>
<loginName>isaac</loginName>
<mediaType>1</mediaType>
<pendingState></pendingState>
<roles>
<role>Agent</role>
</roles>
<settings>
<wrapUpOnIncoming>OPTIONAL</wrapUpOnIncoming>
</settings>
<state>READY</state>
<stateChangeTime>2019-01-11T23:56:54.783Z</stateChangeTime>
<teamId>5000</teamId>
<teamName>Maths</teamName>
<uri>/finesse/api/User/47483648</uri>
</user>
</data>
<event>PUT</event>
<requestId>07f14a42-6b3c-4855-a4c9-af50ab5e7cc6</requestId>
<source>/finesse/api/User/47483648</source>
</Update></notification></item></items></event></message></body>

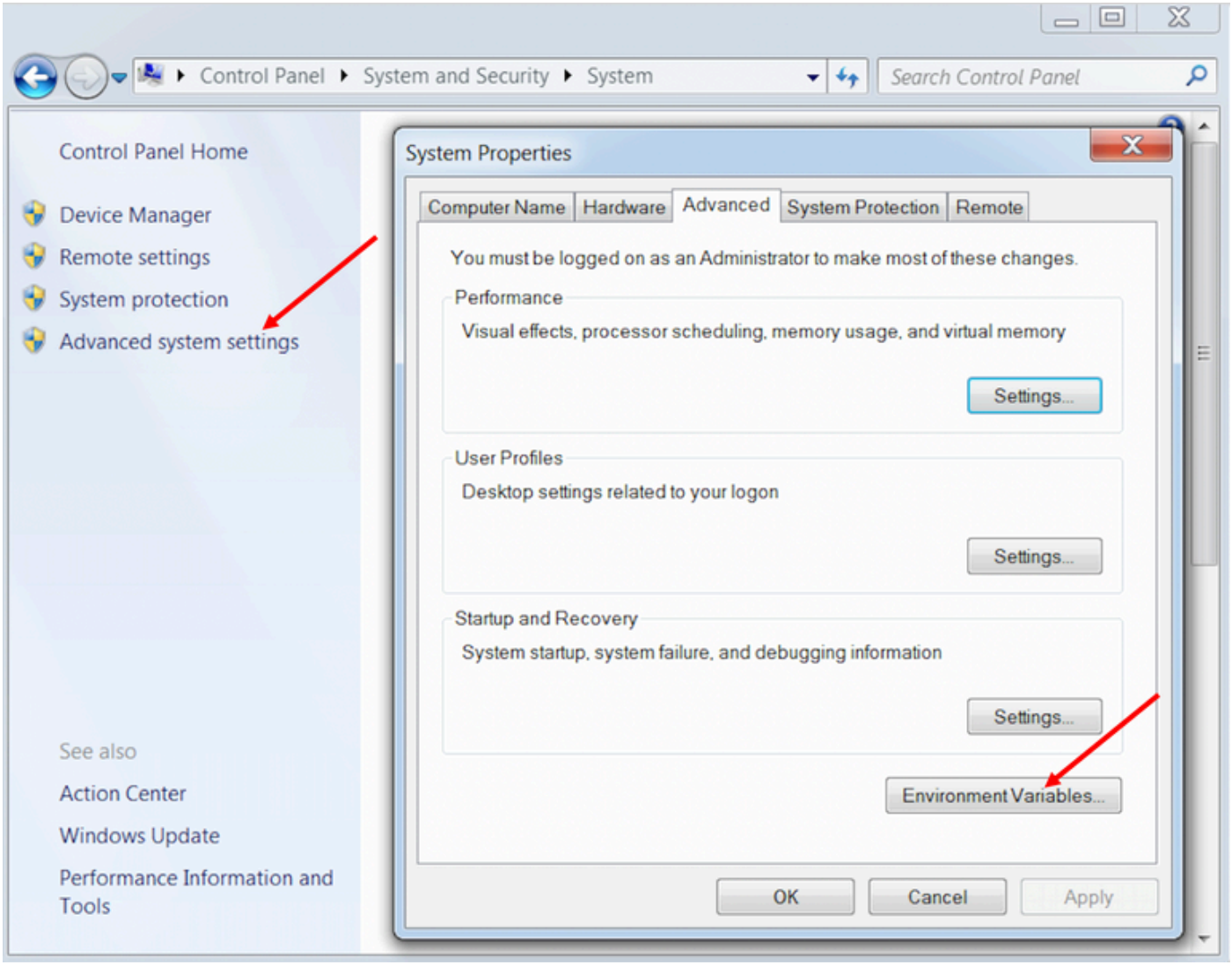
Wiresharkを使用する

Wiresharkは、HTTPSトラフィックのスニッフィングと復号化に使用できる、一般的に使用されているパケットスニフィングツールです。HTTPSトラフィックは、Transport Layer Security(TLS)を介して保護されるHTTPトラフィックです。 TLSは、ホスト間の整合性、認証、および機密性を提供します。Webアプリケーションで一般的に使用されますが、トランスポート層プロトコルとしてTCPを使用する任意のプロトコルで使用できます。Secure Sockets Layer(SSL)は以前のバージョンのTLSプロトコルで、安全ではないため使用されません。これらの名前は互換的に使用されることが多く、SSLまたはTLSトラフィックに使用されるWiresharkフィルタはsslです。

⚠ 注意：提供されている設定例は、ラボ環境のWindows7 x64上のWireshark 2.6.6(v2.6.6-0-gdf942cd8)およびMozilla Firefox 64.0.2（32ビット）用です。これらの手順は、Fiddlerのすべてのバージョン、すべてのブラウザ、またはすべてのコンピュータのオペレーティングシステムに一般化することはできません。 稼働中のネットワークで作業を行う場合は、どのような設定についても、その潜在的な影響について確実に理解しておく必要があります。詳細については、Wireshark SSLの公式ドキュメントを参照してください。Wireshark 1.6以降が必要です。

✎ 注：このメソッドは、FirefoxおよびChromeでのみ機能します。このメソッドは、Microsoft Edgeでは機能しません。

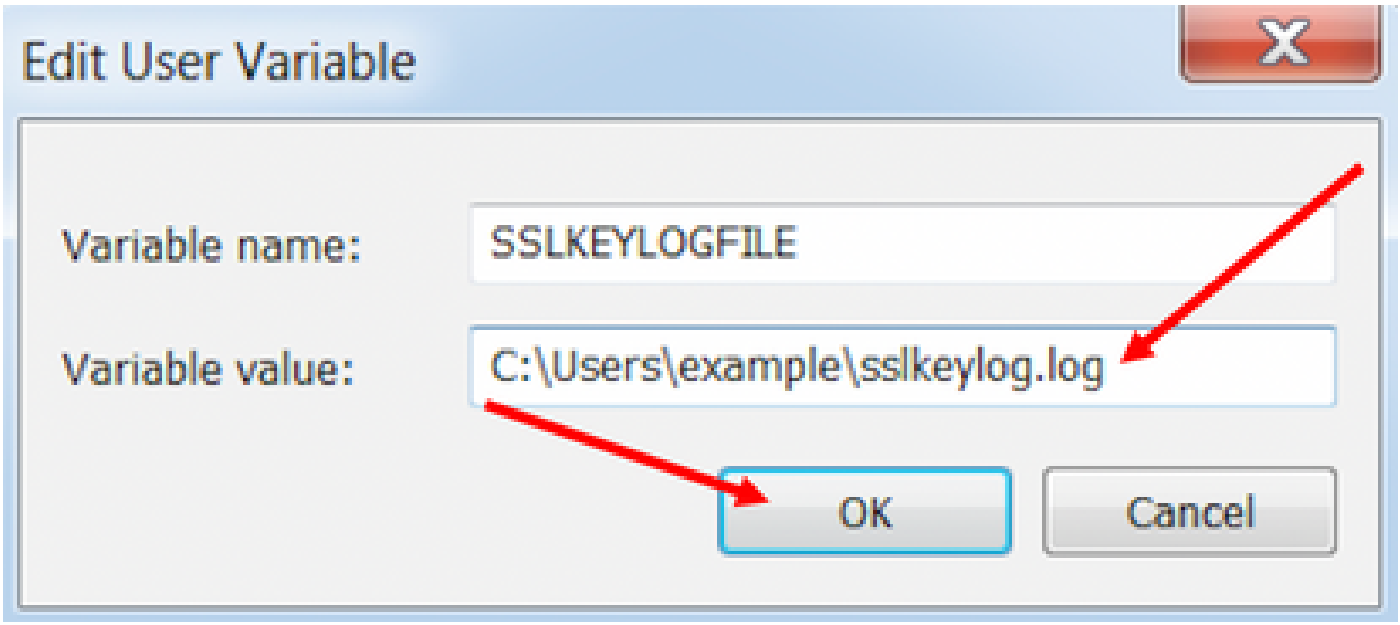
ステップ 1： エージェントのWindows PCで、Control Panel > System and Security > System > Advanced system settings Environmental Variables...の順に選択します。



ステップ 2：User variables for user <username> > New...の順に移動します。

SSLKEYLOGFILEという名前の変数を作成します。

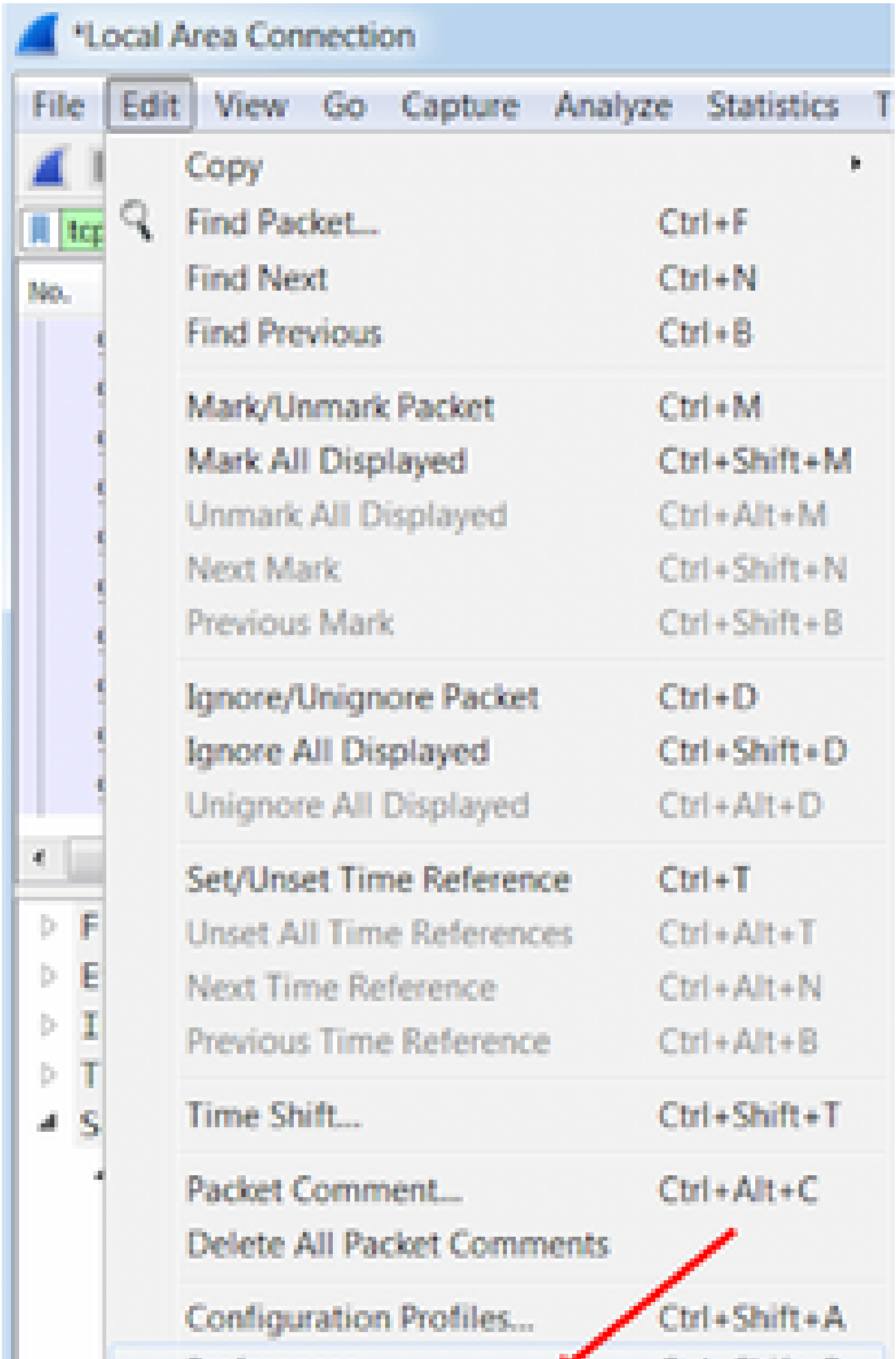
SSLプリマスターシークレットをプライベートディレクトリに保存するファイルを作成します：SSLKEYLOGFILE=</path/to/private/directory/with/logfile>



注：ユーザ変数の代わりにシステム変数を作成するか、プライベート以外のディレクトリにファイルを保存します。ただし、システム上のすべてのユーザがプリマスターシークレットにアクセスできるため、セキュリティが低下します。

ステップ 3：FirefoxまたはChromeが開いている場合は、アプリケーションを閉じます。ファイルを再度開くと、SSLKEYLOGFILEへの書き込みを開始できます。

ステップ 4：Wiresharkで、Edit > Preferences...の順に移動します。


*Local Area Connection
File
Edit
View
Go
Capture
Analyze
Statistics
Copy
Find Packet... Ctrl+F
Find Next Ctrl+N
Find Previous Ctrl+B
Mark/Unmark Packet Ctrl+M
Mark All Displayed Ctrl+Shift+M
Unmark All Displayed Ctrl+Alt+M
Next Mark Ctrl+Shift+N
Previous Mark Ctrl+Shift+B
Ignore/Unignore Packet Ctrl+D
Ignore All Displayed Ctrl+Shift+D
Unignore All Displayed Ctrl+Alt+D
Set/Unset Time Reference Ctrl+T
Unset All Time References Ctrl+Alt+T
Next Time Reference Ctrl+Alt+N
Previous Time Reference Ctrl+Alt+B
Time Shift... Ctrl+Shift+T
Packet Comment... Ctrl+Alt+C
Delete All Packet Comments
Configuration Profiles... Ctrl+Shift+A

翻訳について
シスコは世界中のユーザにそれぞれの言語でサポート コンテンツを提供するために、機械と人による翻訳を組み合わせて、本ドキュメントを翻訳しています。ただし、最高度の機械翻訳であっても、専門家による翻訳のような正確性は確保されません。シスコは、これら翻訳の正確性について法的責任を負いません。原典である英語版（リンクからアクセス可能）もあわせて参照することを推奨します。