

瞭解Finesse BOSH實施並對其進行故障排除

目錄

[簡介](#)

[必要條件](#)

[需求](#)

[採用元件](#)

[背景資訊](#)

[瞭解Finesse BOSH實施](#)

[瞭解XMPP](#)

[XMPP消息示例](#)

[採用Finesse的XMPP實作](#)

[Finesse XMPP請求/響應示例](#)

[瞭解Finesse XMPP消息和XMPP節點](#)

[示例1：使用Pidgin檢視Finesse XMPP節點](#)

[示例2：使用瀏覽器開發者工具「網路」頁籤檢視HTTP消息](#)

[排除BOSH斷開連線錯誤消息故障](#)

[日誌分析](#)

[調試通知服務日誌](#)

[資訊通知服務日誌](#)

[Web服務日誌](#)

[BOSH斷開連線的常見原因](#)

[問題 — 代理在不同時間斷開連線 \(客戶端問題\)](#)

[建議的操作](#)

[問題 — 所有代理同時斷開連線 \(伺服器端問題\)](#)

[建議的操作](#)

[使用Fiddler](#)

[常見的Fiddler問題](#)

[示例配置步驟](#)

[使用Wireshark](#)

[相關缺陷](#)

[相關資訊](#)

簡介

本文檔介紹使用BOSH的Finesse連線背後的體系結構，以及如何診斷BOSH連線問題。

必要條件

需求

思科建議您瞭解以下主題：

- Cisco Finesse
- 整合客服中心企業版(UCCE)
- 整合客服中心Express版(UCCX)
- Web瀏覽器開發工具
- Windows和/或Mac管理

採用元件

本文中的資訊係根據以下軟體和硬體版本：

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

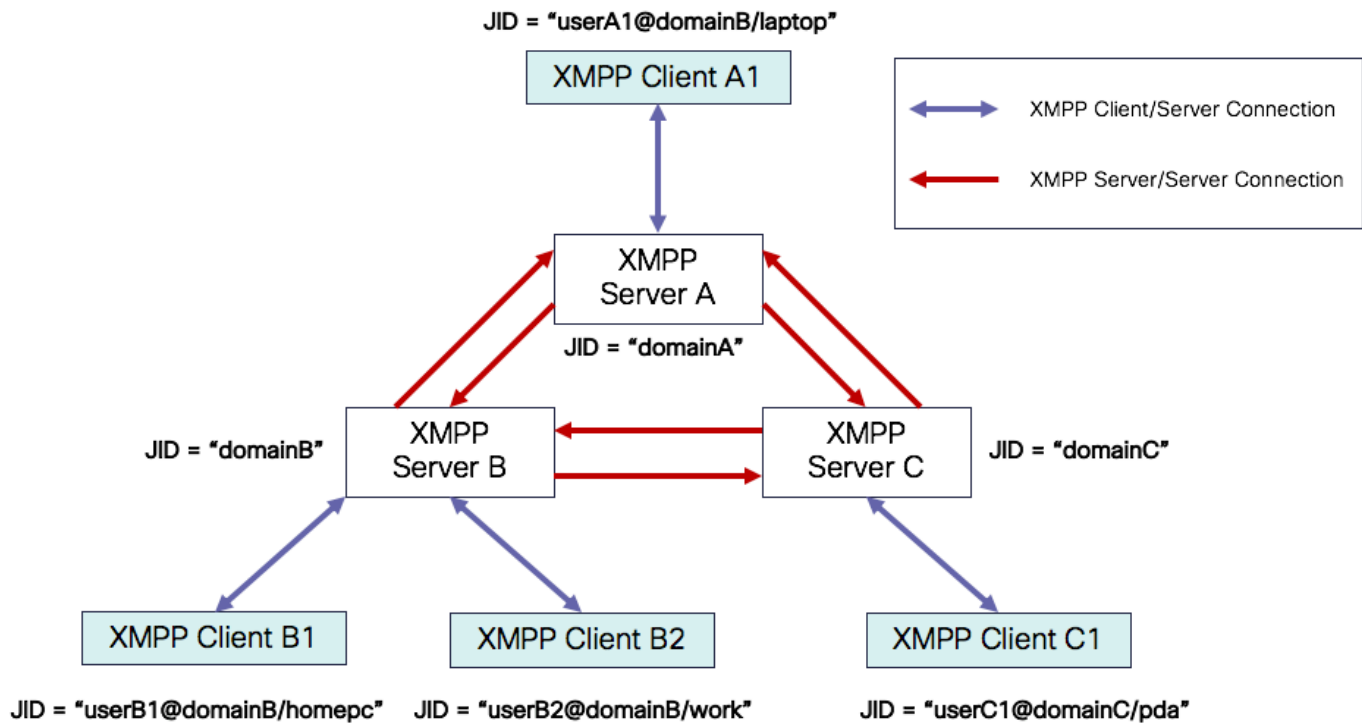
使用通過同步HTTP的雙向流的連線稱為BOSH。

瞭解Finesse BOSH實施

瞭解XMPP


可擴充訊息和狀態通訊協定(XMPP)（也稱為Jabber）是使用者端 — 伺服器模型中的有狀態通訊協定。XMPP允許將結構化的可擴展標籤語言(XML)資料小片段從一個實體快速傳送到另一個實體。XMPP/Jabber廣泛用於即時消息(IM)和線上狀態應用。

所有XMPP實體均通過其Jabber ID(JID)標識。



JID編址方案：user@domain/resource

使用者	xmpp伺服器上的客戶端使用者名稱或會議室的名稱
域	XMPP伺服器完全限定的域名(FQDN)
資源	使用者特定實體/終端的識別符號 (例如, 筆記型電腦、智慧手機等)、會話識別符號或公共節點名稱

 注意：所有三種JID元件並非在所有情況下都使用。伺服器通常僅由域定義，會議室由user@domain定義，客戶端由user@domain/resource定義。

XMPP消息稱為stanzas。XMPP有三個核心標準：

1. <message>：一個方向，一個收件人
2. <presence>：一個方向，發佈到多個
3. <iq>：資訊/查詢 — 請求/響應

所有stanzas均具有自始至終地址，並且大多數stanzas也具有type、id和xml:langattributes。

Stanza屬性	目的
----------	----

成長至	目標JID
自	源JID
類型	消息的用途
id	用於將請求與<iq> stanzas的響應連結的唯一識別符號
xml:lang	定義stanza中任何可讀的XML的預設語言

XMPP消息示例

```
<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>
```

採用Finesse的XMPP實作

如果Web應用程式需要與XMPP配合使用，則會出現多個問題。瀏覽器本身不支援透過傳輸控制通訊協定(TCP)的XMPP，因此所有XMPP流量必須透過在瀏覽器中執行的程式處理。Web伺服器 and 瀏覽器通過超文本傳輸協定(HTTP)消息進行通訊，因此Finesse和其他Web應用程式將XMPP消息包裝在HTTP消息中。

這種方法的第一個困難是HTTP是一種無狀態協定。這表示每個HTTP請求與任何其他請求都不相關。但是，此問題可以通過應用方法（例如，通過使用cookie/post資料）來解決。

第二個困難是HTTP的單向行為。只有客戶端傳送請求，伺服器只能響應。伺服器無法推送資料，因此通過HTTP實施XMPP是不自然的。

原始XMPP核心規範(RFC 6120)中不存在此問題，其中XMPP已繫結到TCP。但是，如果要解決繫結到HTTP的XMPP的問題，例如，因為Javascript可以傳送HTTP請求，則有兩種可能的解決方案。兩者都需要HTTP和XMPP之間的網橋。

推薦的解決方案包括：

1. 輪詢（傳統協定）：重複的HTTP請求，請求在XEP-0025中定義的新資料：Jabber HTTP輪詢

2. 長輪詢也稱為BOSH：傳輸協議，它通過有效使用多個同步HTTP請求/響應對，而不需要使用XEP-0124:HTTP繫結中定義並由XEP-0206:XMPP Over BOSH擴展的頻繁輪詢，來模擬兩個實體之間長壽命雙向TCP連線的語義

Finesse實現了BOSH，因為它從伺服器負載的角度和流量方面來看非常高效。使用BOSH是為了掩蓋伺服器不必在請求發出時立即作出響應這一事實。響應延遲到指定的時間，直到伺服器有客戶端的資料，然後作為響應傳送。客戶端收到響應後，就會發出新的請求，以此類推。

Finesse案頭客戶端（Web應用程式）每30秒通過TCP埠7443建立過時的BOSH連線。30秒後，如果沒有Finesse通知服務的更新，通知服務將傳送一個HTTP回覆，回覆正文為（近）200 OK。例如，如果通知服務更新了代理的存在或對話（呼叫）事件，則資料會立即傳送到Finesse Web客戶端。

Finesse XMPP請求/響應示例

此示例顯示了在Finesse客戶端和Finesse伺服器之間共用的第一個XMPP消息請求響應以設定BOSH連線。

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
```

總結一下：

1. Finesse Web客戶端通過TCP埠7443設定到Finesse伺服器的陳舊HTTP連線(http-bind)。這被稱為BOSH長調查。
2. Finesse通知服務是一種線上狀態服務，用於發佈有關座席、呼叫等狀態的更新。
3. 如果通知服務有更新，它會使用狀態更新作為HTTP響應正文中的XMPP消息來回覆http-bind請求。
4. 如果在收到http-bind請求後30秒內沒有狀態更新，則通知服務會回覆而不進行任何狀態更新，以允許Finesse Web客戶端傳送另一個http-bind請求。這樣，通知服務就知道Finesse Web客戶端仍然能夠連線到通知服務，並且代理沒有關閉其瀏覽器或將其電腦置於睡眠狀態，以此類推。

瞭解Finesse XMPP消息和XMPP節點

Finesse還實施XMPP規範XEP-0060：發佈訂閱。此規範的目的在於允許XMPP伺服器（通知服務）獲取發佈到XMPP節點的資訊（主題），然後傳送XMPP事件到訂閱該節點的實體。對於Finesse，電腦電話整合(CTI)伺服器會向Finesse Web服務傳送CTI消息，以通知Finesse有關配置更新的資訊，例如（但不限於）建立座席或聯絡服務隊列(CSQ)或呼叫資訊。然後，此資訊會轉換為XMPP消息，Finesse Web服務會將其發佈到Finesse通知服務。然後Finesse通知服務通過BOSH將XMPP消息傳送到訂用到某些XMPP節點的代理。

[Finesse Web Services Developer Guide](#)中定義的一些Finesse API對象是XMPP節點。代理和

Supervisor Finesse Web客戶端可以訂閱某些XMPP節點的事件更新，以便獲得有關即時事件(如呼叫事件、狀態事件等)的最新資訊。此表顯示了已啟用pubsub的XMPP節點。

Finesse API對象	目的	訂閱
/finesse/api/User/<LoginID>	顯示座席的狀態和組對映	代理和主管
/finesse/api/User/<LoginID>/對話方塊	顯示座席處理的呼叫	代理和主管
/finesse/api/User/<LoginID>/ClientLog	用於從Send Error Report(傳送錯誤報告)按鈕捕獲客戶端日誌	代理和主管
/finesse/api/User/<LoginID>/Queue/<queueID>	顯示隊列統計資訊 (如果已啟用)	代理和主管
/finesse/api/Team/<TeamID>/Users	顯示屬於包含狀態資訊的某一團隊的座席	主管
/finesse/api/SystemInfo	顯示Finesse伺服器的狀態。用於確定是否需要故障轉移	代理和主管

示例1：使用Pidgin檢視Finesse XMPP節點

步驟 1. 下載並安裝XMPP客戶端Pidgin。

步驟 2. 導覽至Accounts > Modify > Basic，然後設定Login Options:

- 協定：XMPP
- 使用者名稱：任何代理的LoginID
- 域：Finesse伺服器的FQDN
- 資源：佔位符 — 可以使用任何值，例如，測試
- 密碼：代理密碼
- 選中Remember password覈取方塊



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

在瀏覽器斷開連線60秒後，代理將進入「註銷」狀態。座席可以處於「就緒」或「未就緒」狀態，以便註銷發生。

對於UCCE，Finesse最多需要120秒來檢測代理關閉瀏覽器或瀏覽器崩潰的時間，並且Finesse在向CTI伺服器傳送強制註銷請求之前等待60秒，這會導致CTI伺服器將代理置於「未就緒」狀態。在這些條件下，Finesse註銷代理可能需要180秒。與UCCX不同，代理會進入「未就緒」狀態，而不是「註銷」狀態。

 註:UCCE中的CTI斷開未就緒與註銷狀態行為由PG /LOAD引數控制。根據Unified Contact Center Enterprise & Hosted Release 10.0(1)的發行說明，從UCCE 10.0開始，/LOAD引數已棄用。

有關UCCE Finesse案頭行為的詳細資訊，請參閱[Cisco Finesse管理指南](#)中Cisco Finesse故障轉移機制一章的案頭行為部分。

 註：以後可根據產品要求更改計時器值。


日誌分析

Finesse和UCCX通知服務日誌可以通過RTMT或CLI收集：

檔案get activelog /desktop recurs compress

調試通知服務日誌

 注意：僅在重現問題時設定調試級別日誌。重現問題後關閉debug。

 註:Finesse 9.0(1)沒有調試級別記錄。Finesse 9.1(1)中引入了調試級別日誌記錄。在9.1(1)中啟用日誌記錄的過程與Finesse 10.0(1)- 11.6(1)不同。有關此過程，請參閱Finesse管理和適用性指南。

啟用Unified Contact Center Express(UCCX)的通知服務調試日誌，如下所示：

```
<#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.

啟用Unified Contact Center Enterprise(UCCE)(Finesse Standalone)的通知服務調試日誌，如下所示：

```
<#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

這些日誌位於/desktop/logs/openfire資料夾中，名為debug.log。

如圖所示，通知服務(Openfire)debug.log顯示與案頭的http繫結以及代理PC的IP地址和埠。

```
xxx.xxx.xxx.xx:1:34:21 [Session-1, SSL_NULL_WITH_NULL_NULL] received 0 sent 0
2017.04.14 21:34:21 REQUEST /http-bind/ on org.eclipse.jetty.server.nio.SelectChannelConnector$SelectChannelHttpConnection@2d5a26@xxx.xxx.xxx.xx:7443<->xxx.xxx.xxx.xx:49805
2017.04.14 21:34:21 scope null[/http-bind/ @ o.e.j.s.ServletContextHandler{/http-bind,null}
2017.04.14 21:34:21 context=/http-bind[/ @ o.e.j.s.ServletContextHandler{/http-bind,null}
2017.04.14 21:34:21 sessionManager=org.eclipse.jetty.server.session.HashSessionManager@176fe4#STARTED
2017.04.14 21:34:21 session=null
2017.04.14 21:34:21 session=null
2017.04.14 21:34:21 servlet /http-bind[/ -> org.jivesoftware.openfire.http.HttpBindServlet-1643193
2017.04.14 21:34:21 chain=null
2017.04.14 21:34:21 HTTPBindLog: HTTP RECV(3445afbe): <body sid="3445afbe" rid="164053266"/>
2017.04.14 21:34:21 consumeResponse: org.jivesoftware.openfire.http.HttpSession@4d7652 status: 3 address: 1001003@xxx.xxx.xxx.xxx.cisco.com/desktop id: 3445afbe presence:
presence from="1001003@xxx.xxx.xxx.xxx.cisco.com/desktop">
< xmlns="http://jabber.org/protocol/caps" hash="sha-1" node="http://jabber.cisco.com/caxl" ver="VNC6fNwvCxe6FJfDJlryVJRwM=" />
</presence> rid: 164053266
2017.04.14 21:34:21 suspended org.eclipse.jetty.server.nio.SelectChannelConnector$SelectChannelHttpConnection@2d5a26@xxx.xxx.xxx.xx:7443<->xxx.xxx.xxx.xx:49805
2017.04.14 21:34:24 Launching thread for /127.0.0.1:44667
2017.04.14 21:34:24 Launching thread for /127.0.0.1:44656
```

如圖所示，最近活動的0 ms表明會話仍處於活動狀態。

```
2017.04.14 21:34:26 Exiting since queue is empty for /127.0.0.1:44660
2017.04.14 21:34:26 Session (id=3445afbe) was last active 0 ms ago: 1001003@xxxxxxxx.cisco.com/desktop
2017.04.14 21:34:26 time=1492185866851, JID=1001003@xxxxxxxx.cisco.com/desktop, msgs_sent=4, msgs_queue=0, msgs_drop=0, bytes_sent=3748
2017.04.14 21:34:26 time=1492185866851, JID=1001003@xxxxxxxx.cisco.com/desktop, msgs_sent=4, msgs_queue=0, msgs_drop=0, bytes_sent=3748
```

Openfire關閉空閒會話表示座席註銷可在60秒內觸發，Finesse可以將原因代碼為255的強制註銷傳送到CTI伺服器。在這些條件下，案頭的實際行為取決於UCCE中Logout on Agent Disconnect(LOAD)設定。在UCCX中，這始終是行為。

如果Finesse客戶端不向Finesse伺服器傳送http-bind消息，則日誌可以顯示會話運行時間並顯示會話關閉。

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

資訊通知服務日誌

這些日誌位於/desktop/logs/openfire資料夾中，名為info.log。如果Finesse客戶端不向Finesse伺服器傳送http-bind消息，則日誌可以顯示會話變為非活動狀態。

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

Web服務日誌

這些日誌位於/desktop/logs/webservices資料夾中，名為Desktop-webservices.YYYY-MM-DDTHH-MM-SS.sss.log。如果Finesse客戶端在指定的時間內未向Finesse伺服器傳送http-bind消息，則日誌可以顯示代理呈現變為不可用，並且60秒後，會發生呈現驅動註銷。

```
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斷開連線的常見原因

BOSH連線由Web客戶端設定，Finesse伺服器確定代理是否不可用。這些問題幾乎總是與瀏覽器、代理電腦或網路有關的客戶端問題，因為啟動連線的責任由客戶端承擔。

問題 — 代理在不同時間斷開連線 (客戶端問題)

建議的操作

檢查以下問題：

1. 網絡問題：

- 檢視防火牆規則和日誌 — 不得阻止或限制TCP埠7443
- 使用[Fiddler](#)®或[Wireshark](#)®等HTTP Web流量監聽器確認瀏覽器通過TCP埠7443傳送http-bind請求並接收響應
- 檢查代理電腦和Finesse伺服器之間的所有網路裝置/介面是否有過多的延遲或丟包
 - Traceroute可用於確定路徑並確定延遲
 - 在Microsoft® Windows® PC上：tracert {Finesse Server IP | Finesse伺服器FQDN}
 - 在Mac上®: traceroute {Finesse Server IP | Finesse伺服器FQDN}
 - 在Cisco IOS®軟體中，可以檢查介面統計資訊：show interfaces
 - 請參閱[輸入佇列捨棄和輸出佇列捨棄疑難排解](#)
- 收集測試代理的Finesse客戶端日誌。可通過三種方式收集客戶端日誌：
 1. 瀏覽器Web控制檯日誌
 - [Firefox Web控制檯](#)
 - [Microsoft邊緣Web控制檯](#)
 - [Chrome Web主控台](#)
 2. 按Finesse頁面上的[Send Error Report](#)按鈕，並收集Finesse伺服器日誌。日誌位於/desktop/logs/clientlogs中。
 3. 通過https://<Finesse-FQDN>/desktop/locallog登入，並在問題發生後收集日誌。

客戶端每分鐘都會連線到Finesse伺服器以計算漂移和網路延遲：

```
<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: 201
```

如果出現任何日誌收集問題，請參閱[排除Cisco Finesse案頭永續性日誌記錄問題](#)

2. 不支援的瀏覽器和/或版本：

根據相容性清單使用支援的瀏覽器/版本和設定：

[UCCE相容性矩陣](#)

[UCCX相容性清單](#)

3. 由於其他頁籤/視窗的內容/處理而導致瀏覽器停滯狀態：

檢查座席工作流程以檢視他們是否執行以下操作：

- 通常具有其他頁籤或視窗，它們持續運行其他即時應用程式，如音樂/影片流、WebSocket連線、自定義客戶關係管理(CRM)Web客戶端等

- 開啟大量頁籤或視窗
- 已禁用瀏覽器快取
- 已長時間保持瀏覽器運行，在工作日結束時不會關閉瀏覽器

4. 電腦進入睡眠狀態：

檢查代理是否在註銷Finesse之前讓其電腦進入睡眠狀態，或者其電腦睡眠設定計時器是否很低。

5. 客戶端電腦上的CPU或記憶體過大問題：

- 如果代理瀏覽器在共用環境中(如Microsoft Windows Remote Desktop Services、Citrix® XenApp®、Citrix XenDesktop®)運行，則確定瀏覽器效能是否取決於同時運行瀏覽器的使用者數量
 - 確保根據使用者數量配置正確的記憶體和CPU資源
- 檢查電腦資源利用率問題：
 - Windows:
 - Windows [PowerShell Get-Counter](#)命令，每2秒檢查一次CPU時間百分比、可用記憶體百萬位元組數和使用記憶體百分比：Get-Counter -Counter "\Processor(_Total)\% Processor Time", "\Memory\Available MBytes", "\Memory\% Committed Bytes In Use" -SampleInterval 2 — 連續
 - 除了使用PowerShell檢視Windows效能計數器外，還[可以使用Windows效能監視器](#)
 - [任務管理器](#)可用於全域性和逐個進程檢視即時CPU和記憶體統計資訊
 - Mac:
 - 檢查即時總CPU和記憶體的Terminal [Top命令](#)：[top](#)
 - 檢查進程並按CPU利用率排序：前 — o CPU
 - 檢查進程並按記憶體利用率排序：top -o MEM
 - [活動監控器](#)可用於全域性和逐個進程檢視即時CPU和記憶體統計資訊

6. 第三方小工具在後台執行意外的、有問題的活動：

在刪除所有第三方小工具的情況下測試Finesse案頭行為。

7. 伺服器或客戶端上的NTP問題：

- 檢查Finesse發佈伺服器上的utils ntp status，以確保NTP伺服器層數為4或更低
- 在客戶端日誌中，檢查漂移和網路延遲

問題 — 所有代理同時斷開連線 (伺服器端問題)

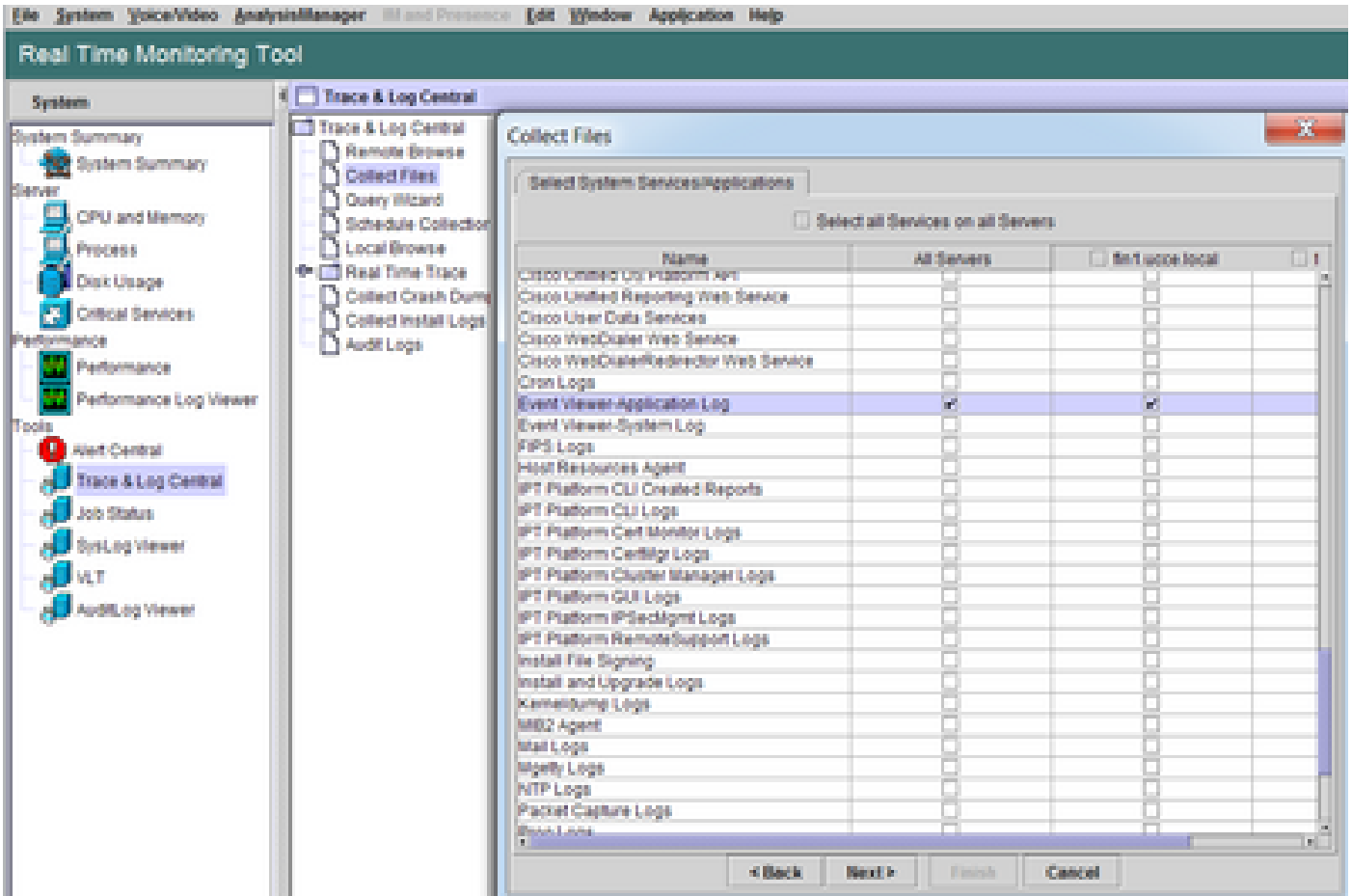
建議的操作

檢查以下問題：

1. Cisco Unified Communications Manager CTIManager服務斷開連線。如果UCCX的所有CTIManager提供程式都處於關閉或崩潰狀態，則UCCX代理會看到紅色橫幅錯誤。如果發生這種情況，UCCE代理看不到紅色標語，但呼叫無法正確路由到代理。

- 檢查在用作CTI提供程式的CUCM伺服器上是否啟動了Cisco CTIManager服務

- 檢查Cisco CTIManager服務是否通過RTMT上的事件檢視器 — 應用程式日誌崩潰，以檢視Cisco CTIManager服務是否崩潰
 - 要在RTMT上收集事件檢視器日誌，請導航到System > Tools > Trace and Log Central > Collect Files > Select System Services/Applications > Event Viewer-Application Log。



- 在CLI上收集Event Viewer-Application日誌：file get activelog /syslog/CiscoSyslog* abstime hh:mm:MM/DD/YY hh:mm:MM/DD/YY
- 在CLI上檢視核心轉儲：使用核心活動清單



註：核心轉儲檔名使用的格式為
 : core.<ProcessID>.<SignalNumber>.<ProcessName>.<EpochTime>。
 示例： core.24587.6.CTManager.1533441238
 因此，可以根據紀元時間確定碰撞的時間。

2. Finesse/UCCX通知服務已停止或崩潰：

- 檢查事件檢視器 — 應用程式日誌中是否有通知服務錯誤，或者檢查服務是否已停止
- 檢查通知服務是否為up: utils服務清單
- 檢查通知服務關閉的時間：檔案搜尋活動日誌/desktop/logs/openfire "Openfire stopped"
- 檢查通知服務的啟動時間：檔案搜尋活動velog /desktop/logs/openfire "HTTP bind service started"
- 檢查由崩潰引起的通知服務記憶體轉儲：file list activelog /desktop/logs/openfire/*.hprof

- 檢查通知服務是否正在偵聽TCP埠7443上的流量：show open ports regexp 7443。*LISTEN
- 檢查這些缺陷是否適用（這些缺陷將導致登入的座席登入失敗；對於已登入的座席，這些座席將看到紅色橫幅Finesse斷開消息）：
 - 思科錯誤ID [CSCva72280](#) - Finesse Tomcat和Openfire Crash（針對無效XML字元）
 - 思科漏洞ID [CSCva72325](#) - UCCX：針對無效XML字元的Finesse Tomcat和Openfire Crash

如果懷疑發生崩潰，請重新啟動Cisco Finesse Tomcat和通知服務。只有在網路出現故障時才會建議這樣做，否則會重新啟動斷開代理與Finesse伺服器的連線。

UCCE的步驟：

- utils service stop Cisco Finesse Tomcat
- utils service stop Cisco Finesse Notification Service
- utils service start Cisco Finesse Tomcat
- utils service start Cisco Finesse Notification Service

UCCX的步驟：

- utils service stop Cisco Finesse Tomcat
- utils服務停止Cisco Unified CCX通知服務
- utils service start Cisco Finesse Tomcat
- utils service start Cisco Unified CCX Notification Service

使用Fiddler

如果不瞭解所需的步驟並瞭解Fiddler的工作原理，配置Fiddler是一項有點挑戰性的任務。Fiddler是一個中間人Web代理，位於Finesse客戶端（Web瀏覽器）和Finesse伺服器之間。由於Finesse客戶端和Finesse伺服器之間的連線受到保護，這為Fiddler配置增加了一層複雜性，以便檢視安全消息。

常見的Fiddler問題

由於Fiddler位於Finesse客戶端和Finesse伺服器之間，因此Fiddler應用程式需要為需要證書的所有Finesse TCP埠建立簽名證書：

Cisco Finesse Tomcat服務證書

1. Finesse發佈伺服器TCP 8445（和/或443 for UCCE）
2. Finesse使用者伺服器TCP 8445（和/或UCCE的443）

Cisco Finesse(Unified CCX)通知服務證書

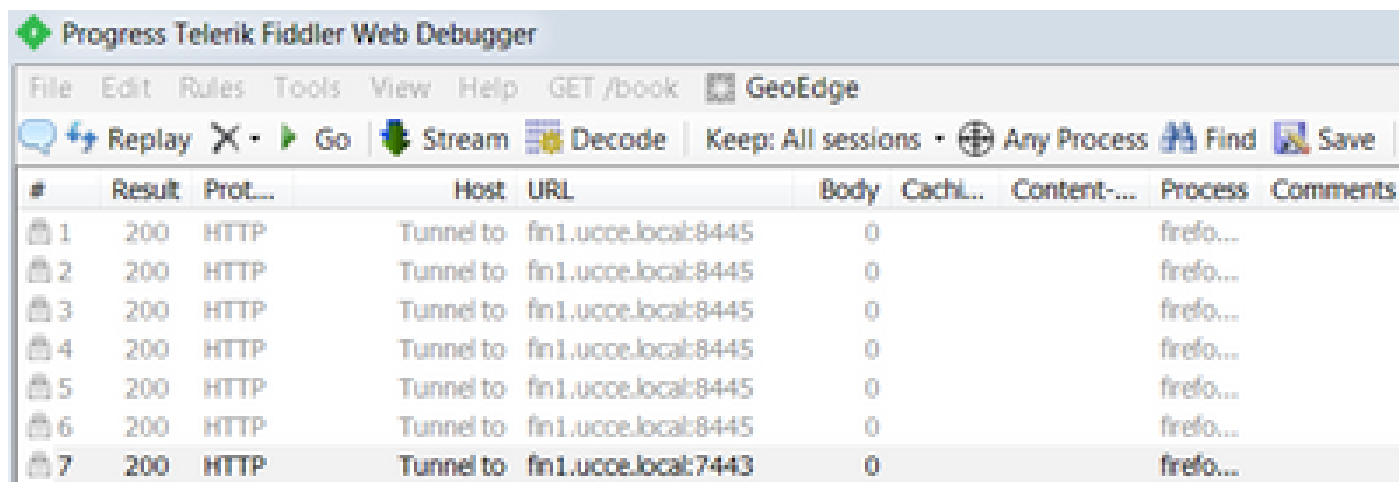
1. Finesse發佈伺服器TCP 7443
2. Finesse使用者伺服器TCP 7443

必須啟用HTTPS解密，Fiddler才能代表Finesse伺服器動態生成證書。預設情況下未啟用此功能。

如果未配置HTTPS解密，則會看到與通知服務的初始隧道連線，但不會顯示http-bind流量。

Fiddler僅顯示：

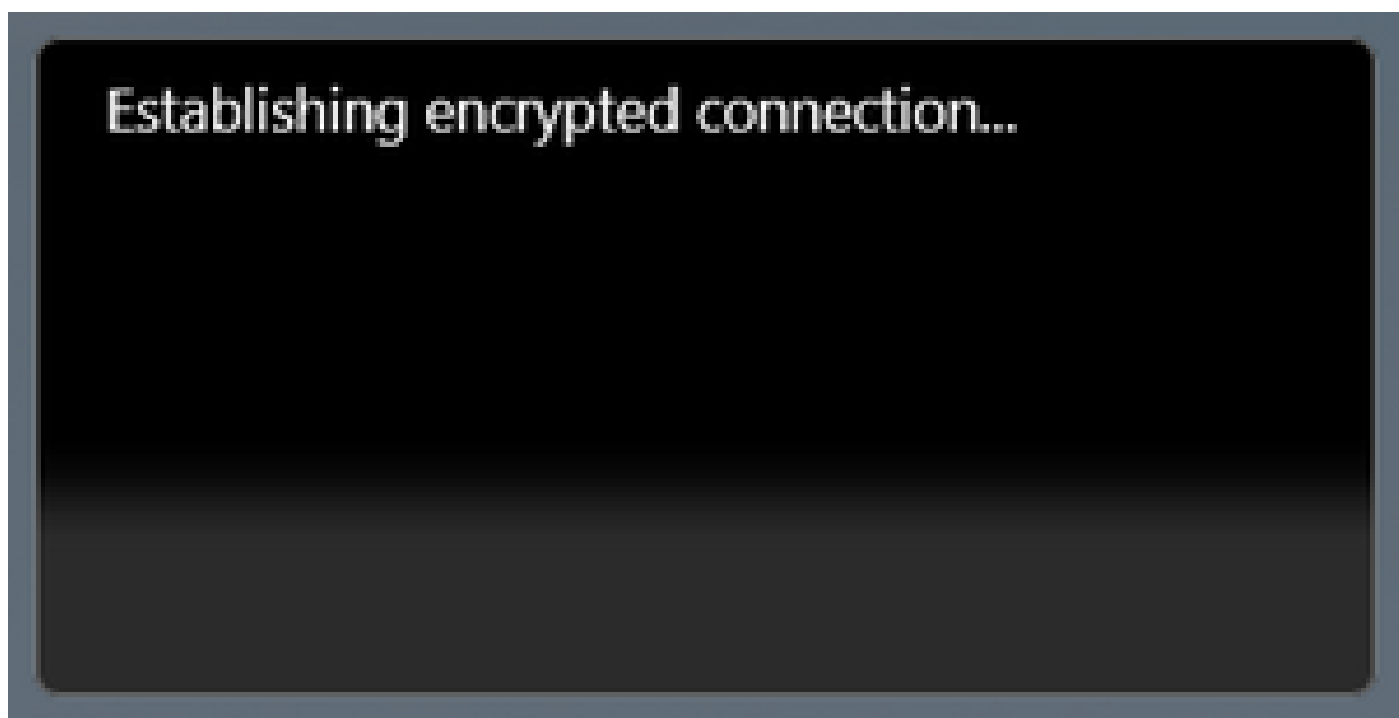
Tunnel to <Finesse server FQDN>:7443



The screenshot shows the Fiddler Web Debugger interface. The title bar reads "Progress Telerik Fiddler Web Debugger". The menu bar includes "File", "Edit", "Rules", "Tools", "View", "Help", and "GET /book". The address bar shows "GeoEdge". The toolbar contains "Replay", "Go", "Stream", "Decode", "Keep: All sessions", "Any Process", "Find", and "Save". Below the toolbar is a table with the following columns: #, Result, Prot..., Host, URL, Body, Cachi..., Content-..., Process, and Comments. The table contains seven rows of data, all with a "200" result and "HTTP" protocol. The host and URL columns show "Tunnel to" followed by the destination address.


#	Result	Prot...	Host	URL	Body	Cachi...	Content-...	Process	Comments
1	200	HTTP	Tunnel to	fin1.uoce.local:8445	0			firefo...	
2	200	HTTP	Tunnel to	fin1.uoce.local:8445	0			firefo...	
3	200	HTTP	Tunnel to	fin1.uoce.local:8445	0			firefo...	
4	200	HTTP	Tunnel to	fin1.uoce.local:8445	0			firefo...	
5	200	HTTP	Tunnel to	fin1.uoce.local:8445	0			firefo...	
6	200	HTTP	Tunnel to	fin1.uoce.local:8445	0			firefo...	
7	200	HTTP	Tunnel to	fin1.uoce.local:7443	0			firefo...	


然後，由Fiddler簽名的Finesse證書必須由客戶端信任。如果這些證書不可信，則無法通過 Establishing encrypted connection.. stage of Finesse login。



在某些情況下，從登入中接受證書例外不起作用，需要瀏覽器手動信任證書。

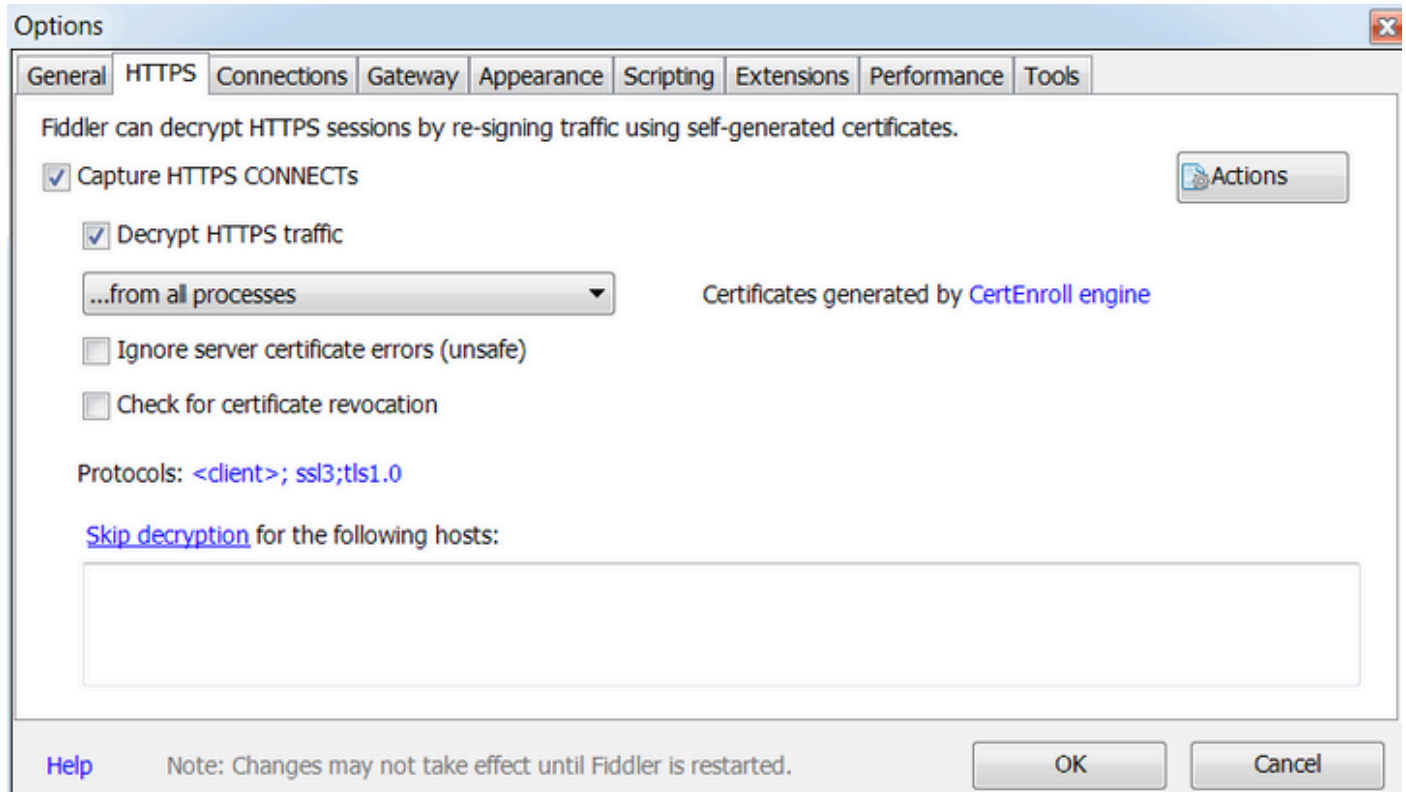
示例配置步驟

 注意：提供的示例配置用於Windows 7 x64上的Fiddler v5.0.20182.28034（用於.NET 4.5和Mozilla Firefox 64.0.2）（32位），位於實驗室環境中。這些過程不能泛化到Fiddler的所有版

 本、所有瀏覽器或所有電腦作業系統。 如果您的網路運作中，請確保您瞭解任何組態可能造成的影響。有關詳細資訊，請參閱[官方Fiddler文檔](#)。

步驟 1. 下載Fiddler

步驟 2. 啟用HTTPS解密。導覽至Tools > Options > HTTPS，然後勾選Decrypt HTTPS traffic覈取方塊。

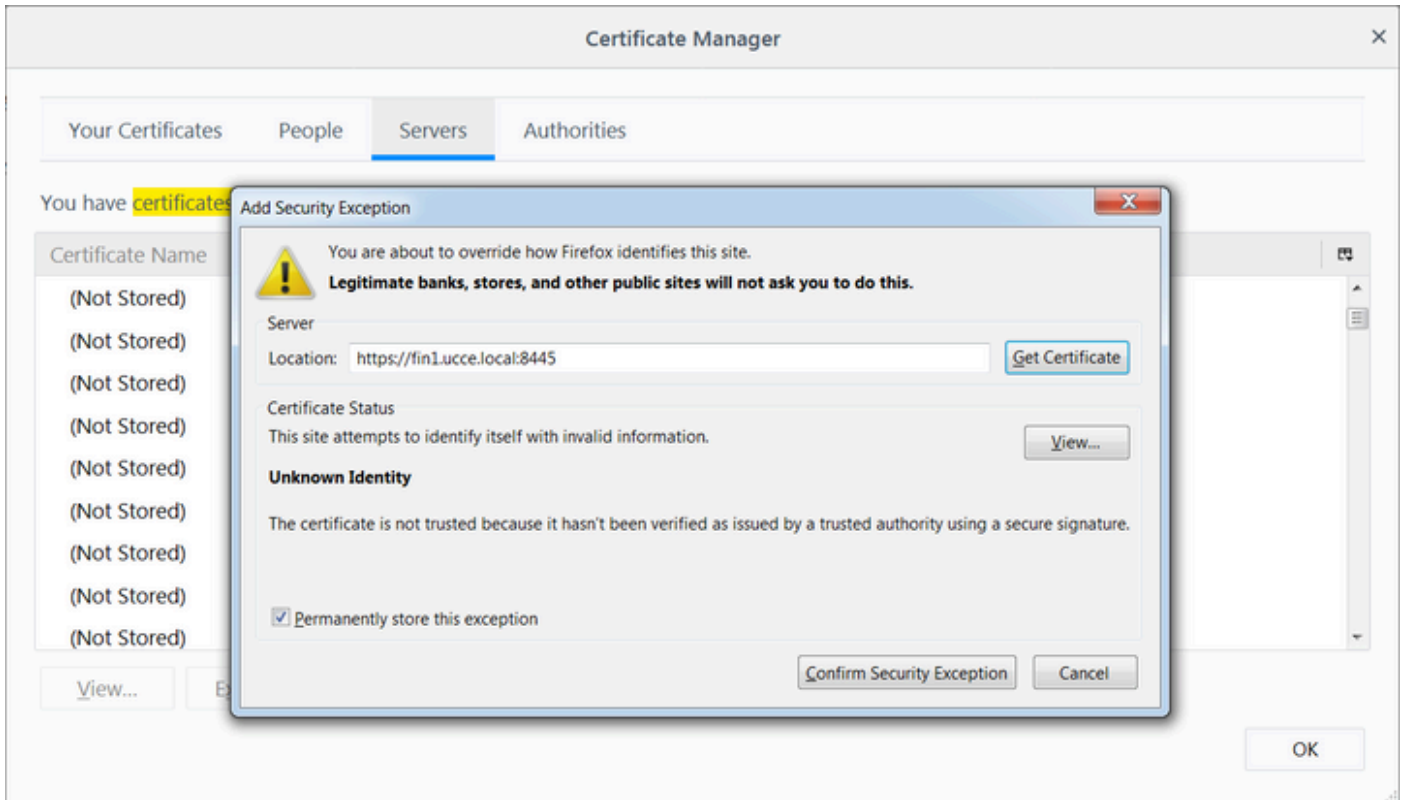


步驟 3. 將開啟一個警告消息框，要求信任Fiddler根證書。選擇Yes。

步驟 4. 此時將開啟一個警告消息框，顯示以下消息：「您將安裝來自證書頒發機構(CA)的證書，該證書宣告表示：DO_NOT_TRUST_FiddlerRoot.....是否要安裝此證書?」。選擇Yes。

步驟 5. 手動將Finesse發佈者和訂閱者證書新增到電腦或瀏覽器證書信任儲存。確保埠8445、7443和 (僅適用於UCCE) 443。例如，在Firefox上，無需從Finesse Operating System Administration頁面下載證書即可完成此操作：

選項>在選項(搜索)>Certificates > Servers > Add Exception > Location > Enter https://<Finesse server>:port為兩個Finesse伺服器的相關埠。



步驟 6. 登入到Finesse，並檢視http-bind消息，該消息通過Fiddler將Finesse客戶端傳送到Finesse伺服器。

在提供的示例中，前5條消息顯示由Finesse伺服器響應的http-bind消息。第一條消息包含消息正文中返回的1571位元組資料。正文包含有關代理事件的XMPP更新。最終http-bind消息已由Finesse客戶端傳送，但尚未從Finesse伺服器獲得響應。當您看到HTTP結果為null(-)且響應正文中的位元組數為null(-1)時，可以確定這一點。

The screenshot shows the Fiddler Web Debugger interface. The top pane displays a list of intercepted requests. The bottom pane shows the raw XML body of a selected request, which is an XMPP message. A red box highlights the message body content.

#	Result	Prot...	Host	URL	Body	Cach...	Content...	Process	Comments	Custo
6...	200	HTTPS	fin1.ucce.local:...	/desktop/thirdparty/...	1,135		text/java...	firefo...		
6...	200	HTTPS	fin1.ucce.local:...	/desktop/thirdparty/...	1,655		text/java...	firefo...		
6...	200	HTTPS	fin1.ucce.local:...	/desktop/thirdparty/...	3,579		text/java...	firefo...		
6...	200	HTTPS	fin1.ucce.local:...	/desktop/thirdparty/...	4,744		text/java...	firefo...		
6...	200	HTTPS	fin1.ucce.local:...	/desktop/thirdparty/...	1,630		text/java...	firefo...		
6...	200	HTTPS	fin1.ucce.local:...	/desktop/thirdparty/...	812		text/html	firefo...		
6...	200	HTTPS	fin1.ucce.local:...	/desktop/thirdparty/...	729		text/html	firefo...		
6...	200	HTTPS	fin1.ucce.local:...	/desktop/thirdparty/...	352		text/html	firefo...		
6...	200	HTTP	detectportal.fire...	/success.txt	8	no-ca...	text/plain	firefo...		
6...	200	HTTPS	fin1.ucce.local:...	/desktop/thirdparty/...	244		text/html	firefo...		
6...	200	HTTPS	fin1.ucce.local:...	/desktop/thirdparty/...	731		text/html	firefo...		
6...	200	HTTPS	fin1.ucce.local:...	/desktop/thirdparty/...	901		text/html	firefo...		
6...	200	HTTPS	fin1.ucce.local:...	/desktop/thirdparty/...	1,302		text/html	firefo...		
6...	200	HTTPS	fin1.ucce.local:...	/desktop/thirdparty/...	307		text/html	firefo...		
6...	200	HTTPS	fin1.ucce.local:...	/desktop/thirdparty/...	287		text/html	firefo...		
6...	200	HTTPS	fin1.ucce.local:...	/desktop/thirdparty/...	569		text/html	firefo...		
6...	200	HTTPS	fin1.ucce.local:...	/desktop/thirdparty/...	910		text/html	firefo...		
6...	200	HTTP	detectportal.fire...	/success.txt	8	no-ca...	text/plain	firefo...		
6...	200	HTTPS	fin1.ucce.local:...	/desktop/thirdparty/...	43		image/gif	firefo...		
6...	200	HTTPS	fin1.ucce.local:...	/desktop/thirdparty/...	1,176		text/html	firefo...		
6...	200	HTTPS	fin1.ucce.local:...	/desktop/theme/fine...	673		image/gif	firefo...		
6...	200	HTTPS	fin1.ucce.local:...	/desktop/cscowidge...	720		text/html	firefo...		
6...	200	HTTPS	fin1.ucce.local:...	/finesse/api/User/47...	631	no-ca...	applicato...	firefo...		
6...	200	HTTPS	fin1.ucce.local:...	/desktop/thirdparty/...	12,7...		image/png	firefo...		
6...	200	HTTPS	fin1.ucce.local:...	/desktop/theme/fine...	2,205		image/png	firefo...		
6...	200	HTTPS	fin1.ucce.local:...	/finesse/api/User/47...	340	no-ca...	applicato...	firefo...		
6...	200	HTTPS	fin1.ucce.local:...	/finesse/api/User/47...	1,851	no-ca...	applicato...	firefo...		
6...	200	HTTPS	fin1.ucce.local:...	/finesse/api/User/47...	20	no-ca...	applicato...	firefo...		
6...	200	HTTPS	fin1.ucce.local:...	/gadgets/makeRequ...	340	no-ca...	applicato...	firefo...		
6...	200	HTTP	Tunnel to	cuic1.ucce.local:8444	0		firefo...			
6...	200	HTTPS	fin1.ucce.local:...	/gadgets/makeRequ...	340	no-ca...	applicato...	firefo...		
6...	200	HTTP	detectportal.fire...	/success.txt	8	no-ca...	text/plain	firefo...		
6...	200	HTTP	Tunnel to	cuic1.ucce.local:8444	0		firefo...			
6...	200	HTTP	detectportal.fire...	/success.txt	8	no-ca...	text/plain	firefo...		
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...	applicato...	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...	applicato...	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...	applicato...	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...	applicato...	firefo...		

```

POST https://fin1.ucce.local:7443/http-bind/ HTTP/1.1
Host: fin1.ucce.local:7443
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:62.0) Gecko/20100101 Firefox/62.0
Accept: text/plain, */*; q=0.01
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Referer: https://fin1.ucce.local:7443/tunnel/
Content-Type: text/xml
X-Requested-With: XMLHttpRequest
Content-Length: 83
Cookie: finesse.ag_extension=10005; JSESSIONID=6F9274007922D8015E0A69003FC260F
Connection: keep-alive
Pragma: no-cache
Cache-Control: no-cache

<body xmlns="http://jabber.org/protocol/httpbind" sid="3117c5ef" rid="2779414706"/>
<message xmlns="http://jabber.org/protocol/httpbind" 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">&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;07114e42-6b3c-4855-a4c9-af50ab5e7cc6&lt;/requestId&gt;
&lt;source&gt;finesse/api/User/47483648&lt;/source&gt;
&lt;/Update&gt;</notification></item></items></message></body>
  
```

更近的資料檢視：


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...

XMPP消息的響應正文：

```
<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-5adbdc74a7717"><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>
```

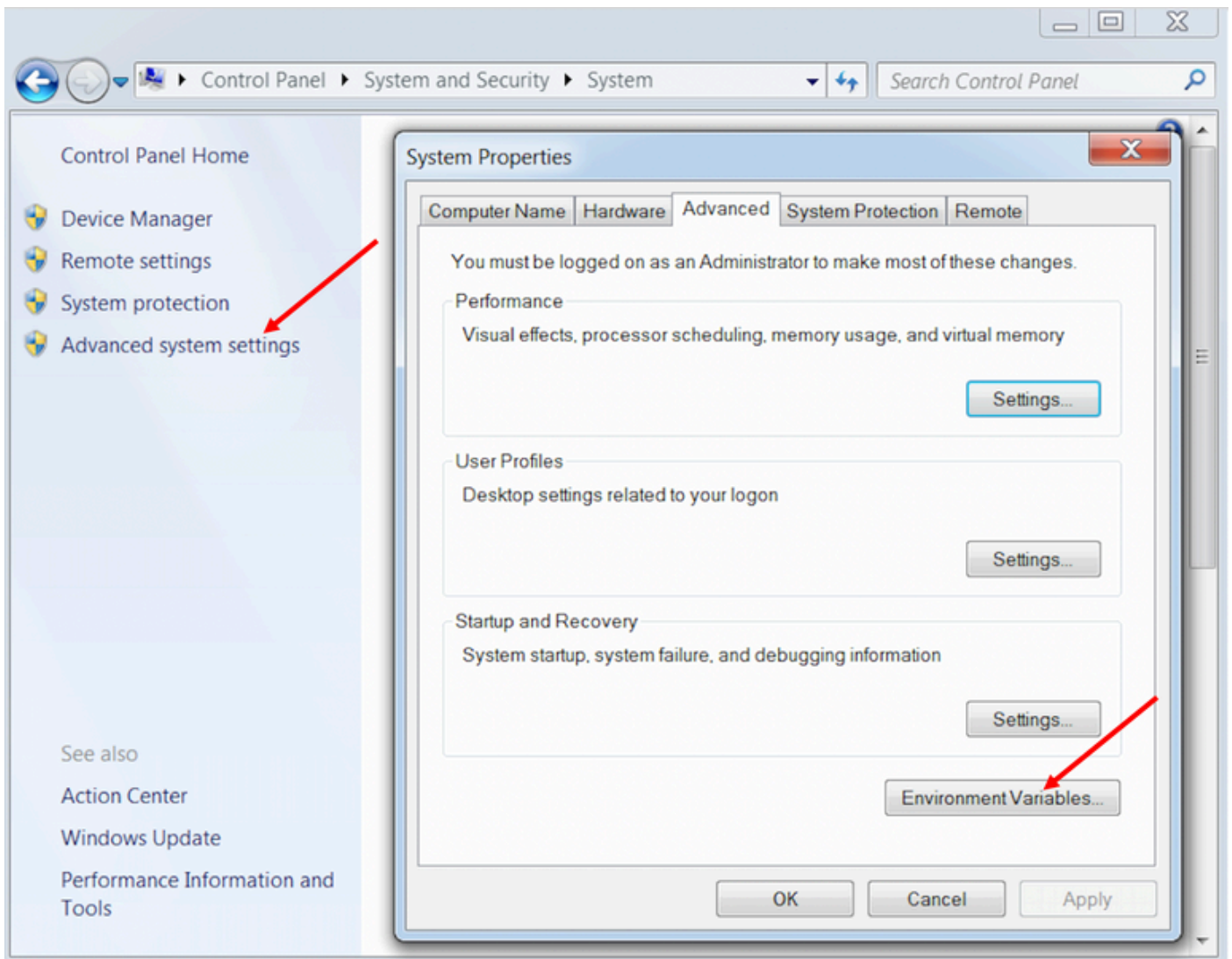
使用Wireshark

Wireshark是常用的資料包嗅探工具，可用於嗅探和解碼HTTPS流量。HTTPS流量是通過傳輸層安全(TLS)保護的HTTP流量。TLS為主機提供完整性、身份驗證和機密性。它通常用於Web應用程式，但可以與任何使用TCP作為傳輸層協定的協定一起使用。安全套接字層(SSL)是TLS協定的前一個版本，不再使用，因為它是不安全的。這些名稱通常可互換使用，用於SSL或TLS流量的Wireshark過濾器為ssl。

 注意：提供的示例配置適用於實驗室環境中Windows7 x64上的Wireshark 2.6.6(v2.6.6-0-gdf942cd8)和Mozilla Firefox 64.0.2 (32位)。這些過程不能泛化到Fiddler的所有版本、所有瀏覽器或所有電腦作業系統。如果您的網路運作中，請確保您瞭解任何組態可能造成的影響。有關詳細資訊，請參閱[官方Wireshark SSL文檔](#)。需要Wireshark 1.6或更高版本。

 註：此方法只能用於Firefox和Chrome。此方法不適用於Microsoft Edge。

步驟 1. 在代理的Windows PC上，導航到控制面板>系統和安全>系統>高級系統設定環境變數.....

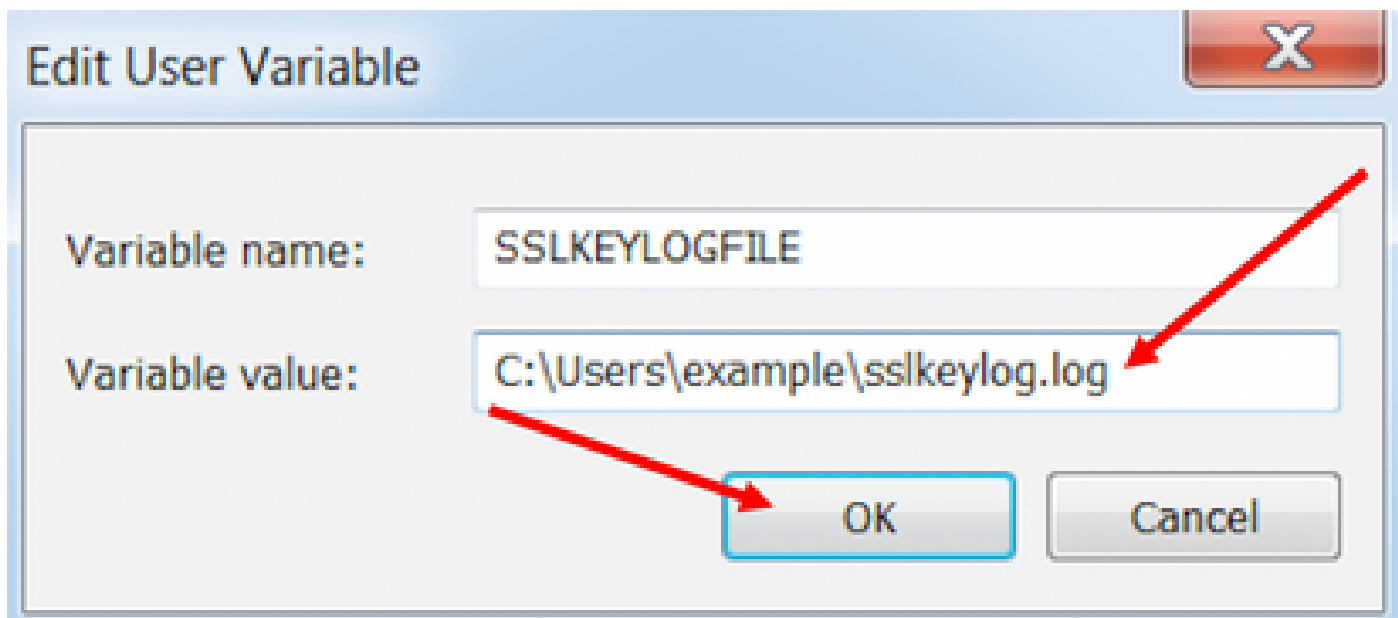



步驟 2. 導航到 User <username> > New... 的使用者變數

建立名為 SSLKEYLOGFILE 的變數。

建立一個檔案以將 SSL 預主金鑰儲存在專用目錄中

: SSLKEYLOGFILE=</path/to/private/directory/with/logfile>



 註：建立一個系統變數而不是使用者變數並/或將檔案儲存在非專用目錄中，但系統上的所有使用者都可以訪問安全性較低的premaster金鑰。

步驟 3. 如果Firefox或Chrome處於開啟狀態，請關閉應用程式。重新開啟後，它們可以開始寫入SSLKEYLOGFILE。

步驟 4. 在Wireshark上，導航到編輯>首選項.....

Local Area Connection

File Edit View Go Capture Analyze Statistics T

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

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）。