

# 了解Finesse BOSH实施并对其进行故障排除

## 目录

---

[简介](#)

[先决条件](#)

[要求](#)

[使用的组件](#)

[背景信息](#)

[了解Finesse BOSH实施](#)

[了解XMPP](#)

[XMPP消息示例](#)

[使用Finesse实施XMPP](#)

[Finesse XMPP请求/响应示例](#)

[了解Finesse XMPP消息和XMPP节点](#)

[示例1：使用Pidgin查看Finesse XMPP节点](#)

[示例2：使用浏览器开发者工具网络选项卡查看HTTP消息](#)

[排除BOSH断开连接错误消息故障](#)

[日志分析](#)

[调试通知服务日志](#)

[信息通知服务日志](#)

[Web服务日志](#)

[BOSH断开连接的常见原因](#)

[问题 — 座席在不同时间断开连接 \(客户端问题\)](#)

[推荐的操作](#)

[问题 — 所有代理同时断开 \(服务器端问题\)](#)

[推荐的操作](#)

[使用Fiddler](#)

[常见的Fiddler问题](#)

[配置步骤示例](#)

[使用Wireshark](#)

[相关问题](#)

[相关信息](#)

---

## 简介

本文档介绍使用BOSH的Finesse连接背后的架构，以及如何诊断BOSH连接问题。

## 先决条件

### 要求

Cisco 建议您了解以下主题：

- 思科Finesse
- 统一联系中心企业版(UCCE)
- Unified Contact Center Express (UCCX)
- Web浏览器开发工具
- Windows和/或Mac管理

## 使用的组件

本文档中的信息基于以下软件和硬件版本：

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

本文档中的信息都是基于特定实验室环境中的设备编写的。本文档中使用的所有设备最初均采用原始（默认）配置。如果您的网络处于活动状态，请确保您了解所有命令的潜在影响。

## 背景信息

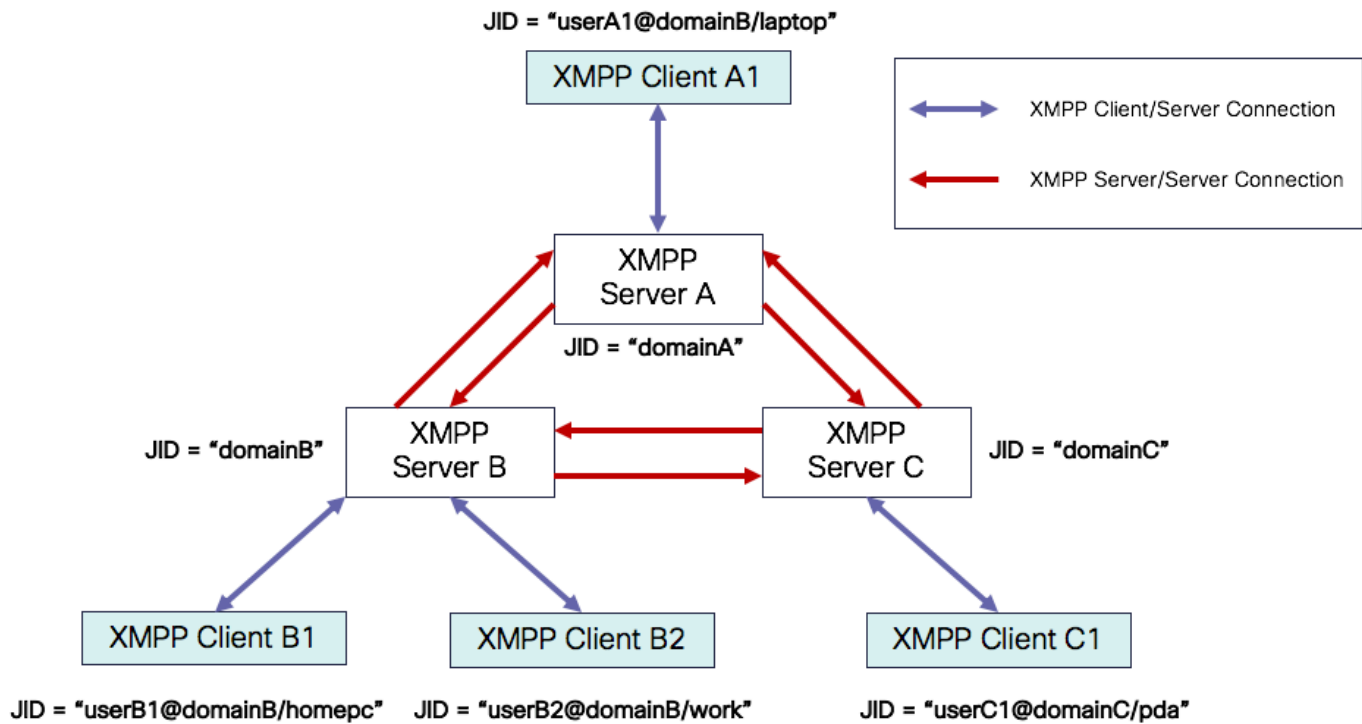
使用同步HTTP上的双向流的连接称为BOSH。

## 了解Finesse BOSH实施

### 了解XMPP


可扩展消息传送和在线状态协议(XMPP)（也称为Jabber）是客户端 — 服务器模型中的状态协议。XMPP允许将小片结构化可扩展标记语言(XML)数据从一个实体快速传输到另一个实体。XMPP/Jabber广泛用于即时消息(IM)和在线状态应用。

所有XMPP实体都通过其Jabber ID(JID)进行标识。



JID编址方案：user@domain/resource

用户	XMPP服务器上的客户端用户名或会议室的名称
域	XMPP服务器完全限定域名(FQDN)
资源	用户特定实体/终端的标识符（例如，笔记本电脑、智能手机等）、会话标识符或公共节点名称

 注意：所有三种JID组件并非在所有情况下都使用。服务器通常仅由域定义，会议室由user@domain定义，客户端由user@domain/resource定义。

XMPP消息称为标准。XMPP有三个核心标准：

1. <message>：一个方向，一个收件人
2. <presence>：一个方向，向多个用户发布
3. <iq>：信息/查询 — 请求/响应

所有stanzas均具有往返地址，并且大多数stanzas也具有type、id和xml:langattributes。

Stanza属性	目的
----------	----

到	目的JID
从	源JID
类型	消息的用途
ID	用于将请求与<iq>标准响应链接起来的唯一标识符
xml:lang	定义stanza中任何可读的XML的默认语言

## 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>
```

## 使用Finesse实施XMPP

如果Web应用程序需要与XMPP配合使用，则会出现多个问题。浏览器本身不支持基于传输控制协议(TCP)的XMPP，因此所有XMPP流量必须由在浏览器中运行的程序处理。Web服务器和浏览器通过超文本传输协议(HTTP)消息进行通信，因此Finesse和其他Web应用程序将XMPP消息包装在HTTP消息内。

此方法的第一个困难是HTTP是无状态协议。这意味着每个HTTP请求与任何其他请求都不相关。但是，这个问题可以通过应用方法（例如使用cookie/post数据）来解决。

第二个困难是HTTP的单向行为。只有客户端发送请求，服务器只能响应。服务器无法推送数据，因此通过HTTP实施XMPP是不自然的。

原始XMPP核心规范(RFC 6120)中不存在此问题，其中XMPP与TCP绑定。但是，如果您想解决绑定到HTTP的XMPP的问题，例如，由于Javascript可以发送HTTP请求，所以有两种可能的解决方案。两者都需要HTTP和XMPP之间的网桥。

推荐的解决方案包括：

1. 轮询（传统协议）：重复的HTTP请求，请求在XEP-0025中定义的新数据：Jabber HTTP轮询

2. 长轮询也称为BOSH：传输协议，它通过高效地使用多个同步HTTP请求/响应对来模拟两个实体之间长期、双向TCP连接的语义，而无需使用XEP-0124:HTTP Binding中定义并由XEP-0206:XMPP Over BOSH扩展的频繁轮询

Finesse实施BOSH是因为从服务器负载角度和流量角度来看它非常高效。使用BOSH是为了掩盖服务器不必在出现请求时立即作出响应这一事实。响应延迟到指定的时间，直到服务器有客户端的数据，然后作为响应发送。客户端收到响应后，便发出新的请求，以此类推。

Finesse桌面客户端（Web应用）每30秒通过TCP端口7443建立陈旧的BOSH连接。30秒后，如果没有来自Finesse通知服务的更新，通知服务将发送一个HTTP应答，其中包含200 OK和（几乎）空响应正文。例如，如果通知服务更新了座席或对话（呼叫）事件的状态，则数据将立即发送到Finesse Web客户端。

## Finesse XMPP请求/响应示例

此示例显示了Finesse客户端和Finesse服务器之间共享的第一个XMPP消息请求响应，用于设置BOSH连接。

Finesse client request:

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

Finesse server response:

```
<body xmlns="http://jabber.org/protocol/httpbind" xmlns:stream="http://etherx.jabber.org/streams" authi
```

综述：

1. Finesse Web客户端通过TCP端口7443设置到Finesse服务器的陈旧HTTP连接(http-bind)。这称为BOSH long poll。
2. Finesse通知服务是一种在线状态服务，用于发布有关座席、呼叫等状态的更新。
3. 如果通知服务有更新，它将使用状态更新作为HTTP响应正文中的XMPP消息来响应http-bind请求。
4. 如果在收到http-bind请求后30秒内没有状态更新，通知服务会回复而没有任何状态更新，以允许Finesse Web客户端发送另一个http-bind请求。这样，通知服务就可以知道Finesse Web客户端仍然能够连接到通知服务，并且代理未关闭其浏览器或将其计算机置于休眠状态，以此类推。

## 了解Finesse XMPP消息和XMPP节点

Finesse还实施XMPP规范XEP-0060:Publish-Subscribe。此规范的目的在于允许XMPP服务器（通知服务）获取发布到XMPP节点的信息（主题），然后发送XMPP事件到订阅该节点的实体。对于Finesse，计算机电话集成(CTI)服务器会向Finesse Web服务发送CTI消息，以告知Finesse有关配置更新的信息，例如但不限于座席或联系服务队列(CSQ)的创建或呼叫信息。然后，此信息将转换为Finesse Web服务发布到Finesse通知服务的XMPP消息。然后，Finesse通知服务通过BOSH将XMPP消息发送到订用到某些XMPP节点的代理。

[Finesse Web Services Developer Guide](#)中定义的一些Finesse API对象是XMPP节点。代理和

Supervisor Finesse Web客户端可以订用某些XMPP节点的事件更新，以获得有关实时事件（例如呼叫事件、状态事件等）的最新信息。此表显示启用了pubsub的XMPP节点。

Finesse API对象	目的	订用
/finesse/api/User/<LoginID>	显示座席的状态和组映射	座席和主管
/finesse/api/User/<LoginID>/对话框	显示座席处理的呼叫	座席和主管
/finesse/api/User/<LoginID>/ClientLog	用于从Send Error Report（发送错误报告）按钮捕获客户端日志	座席和主管
/finesse/api/User/<LoginID>/Queue/<queueID>	显示队列统计信息（如果已启用）	座席和主管
/finesse/api/Team/<TeamID>/Users	显示属于特定组的座席，包括状态信息	主管
/finesse/api/SystemInfo	显示Finesse服务器的状态。用于确定是否需要故障切换	座席和主管

示例1：使用Pidgin查看Finesse XMPP节点

步骤1:下载并安装XMPP客户端Pidgin。

第二步：导航到帐户>修改>基本并配置登录选项：

- 协议：XMPP
- 用户名：任何代理的LoginID
- 域：Finesse服务器的FQDN
- 资源：占位符 — 可以使用任何值，例如，测试
- 密码：代理密码
- 选中Remember password复选框



# 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:



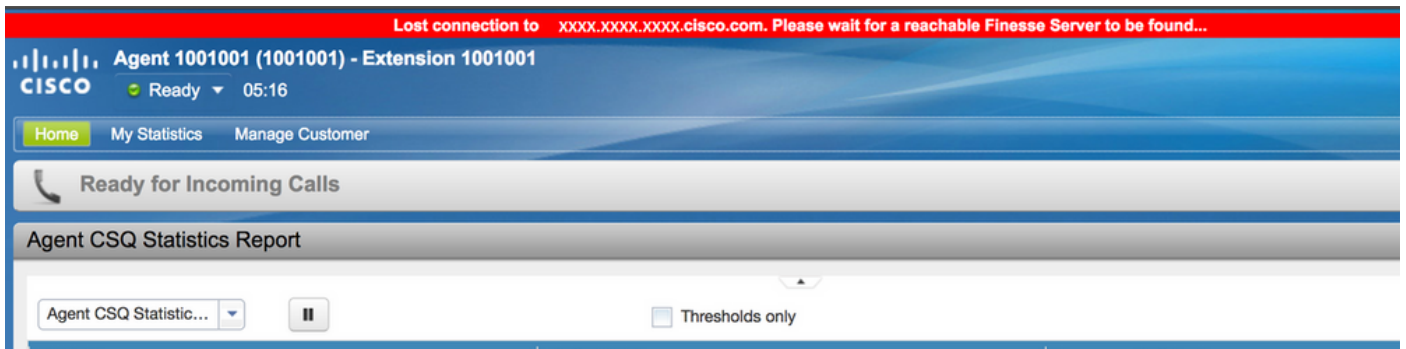
Remove

Create this new account on the server

Cancel

Save


Lost connection to {Finesse Server FQDN}。Please wait for a reachable Finesse Server to be found... ( 请等待找到可访问的Finesse服务器..... ) 显示在Finesse桌面顶部的红色横幅中。




此时会显示此消息，因为此时无法从Cisco Finesse通知服务接收XMPP订用事件。因此，状态信息和呼叫详细信息不能显示在座席桌面上。

对于UCCX，在浏览器断开连接60秒后，代理将进入“注销”状态。座席可以处于“就绪”或“未就绪”状态，以便注销发生。

对于UCCE，Finesse最多需要120秒来检测代理关闭浏览器或浏览器崩溃的时间，并且Finesse在向CTI服务器发送强制注销请求之前等待60秒，这会导致CTI服务器将代理置于“未就绪”状态。在这些情况下，Finesse最多需要180秒才能注销代理。与UCCX不同，代理将进入“未就绪”状态而不是“注销”状态。

 注:UCCE中的CTI断开未就绪与注销状态行为由PG /LOAD参数控制。根据Unified Contact Center Enterprise & Hosted版本10.0(1)的版本说明，从UCCE 10.0开始，/LOAD参数已弃用。

有关UCCE Finesse桌面行为的详细信息，请参阅[Cisco Finesse管理指南](#)中Cisco Finesse故障切换机制一章的桌面行为部分。


 注意：以后可根据产品要求更改计时器值。


## 日志分析

Finesse和UCCX通知服务日志可以通过RTMT或CLI收集：

文件get activelog /desktop recurs compress

调试通知服务日志

 注意：仅在重现问题时设置调试级别日志。重现问题后关闭调试。

 注意：Finesse 9.0(1)没有调试级别日志记录。调试级别日志记录在Finesse 9.1(1)中引入。与Finesse 10.0(1)- 11.6(1)相比，在9.1(1)中启用日志记录的流程有所不同。有关此过程，请参阅Finesse管理和适用性指南。



启用Unified Contact Center Express(UCCX)的通知服务调试日志，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.
```

启用Unified Contact Center Enterprise(UCCE) ( Finesse独立 ) 的通知服务调试日志，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
```

这些日志位于/desktop/logs/openfire文件夹中，名为debug.log。

如图所示，通知服务(Openfire)debug.log显示与桌面的http绑定以及代理PC的IP地址和端口。

```
xxx.xxx.xxx.xx:11:34:21 [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$SelectChannelHttpConnection@2d5e26@xxx.xxx.xxx.xx:7443<->xxx.xxx.xxx.xx:49805
2017.04.14 21:34:21 scope null|/http-bind/ @ o.e.j.s.ServletContextHandler{/http-bind,null}
2017.04.14 21:34:21 context=/http-bind|/ @ o.e.j.s.ServletContextHandler{/http-bind,null}
2017.04.14 21:34:21 sessionManager=org.eclipse.jetty.server.session.HashSessionManager@176fe4#STARTED
2017.04.14 21:34:21 session=null
2017.04.14 21:34:21 session=null
2017.04.14 21:34:21 servlet /http-bind|/ -> org.jivesoftware.openfire.http.HttpBindServlet-1643193
2017.04.14 21:34:21 chain=null
2017.04.14 21:34:21 HTTPBindLog: HTTP RECV(3445afbe): <body sid="3445afbe" rid="164053266"/>
2017.04.14 21:34:21 consumeResponse: org.jivesoftware.openfire.http.HttpSession@dd7653-etotus: 3 address: 1001003@xxx.xxx.xxx.xxx.cisco.com/desktop id: 3445afbe presence:
<presence from="1001003@xxx.xxx.xxx.xxx.cisco.com/desktop">
< xmlns="http://jabber.org/protocol/caps" hash="sha-1" node="http://jabber.cisco.com/caxl" ver="VNC6fNwvCxe6FjDJIPLryVJRw="/>
</presence> rid: 164053266
2017.04.14 21:34:21 suspended org.eclipse.jetty.server.nio.SelectChannelConnector$SelectChannelHttpConnection@2d5e26@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
```

如图所示，最近的0毫秒活动表明会话仍处于活动状态。

```
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@XXXXXXXXX.XXXXXXXXXX.cisco.com/desktop
2017.04.14 21:34:26 time=1492185866851,JID=1001003@XXXXXXXXX.XXXXXXXXXX.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@XXXXXXXXX.XXXXXXXXXX.cisco.com/desktop,msgs_sent=4,msgs_queue=0,msgs_drop=0,bytes_sent=3748
```

Openfire关闭空闲会话表示座席注销可以在60秒内触发，Finesse可以将原因代码为255的强制注销发送到CTI服务器。在这些条件下桌面的实际行为取决于UCCE中Logout on Agent Disconnect(LOAD)的设置。在UCCX中，这始终是行为。

如果Finesse客户端不向Finesse服务器发送http-bind消息，日志可以显示会话运行时间并显示会话关闭。

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

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

## 信息通知服务日志

这些日志位于/desktop/logs/openfire文件夹中，名为info.log。如果Finesse客户端不向Finesse服务器发送http-bind消息，日志可以显示会话变为非活动状态。

```
2017.06.17 00:16:04 Closing idle session (id=f382a015): 1001003@xxxxxx.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@xxxxxx.xxxx.xxx. cisco.com/desktop
```

## Web服务日志

这些日志位于/desktop/logs/webservices文件夹中，名为Desktop-webservices.YYYY-MM-DDTHH-MM-SS.sss.log。如果Finesse客户端在指定的时间内未向Finesse服务器发送http-bind消息，则日志可以显示代理在线状态变为不可用，并且60秒后，可能会发生在线状态驱动的注销。

```
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断开连接的常见原因

BOSH连接由Web客户端设置，Finesse服务器确定代理在线状态是否不可用。这些问题几乎总是与浏览器、代理计算机或网络相关的客户端问题，因为启动连接的责任由客户端承担。

### 问题 — 座席在不同时间断开连接（客户端问题）

#### 推荐的操作

检查以下问题：

#### 1.网络问题：

- 查看防火墙规则和日志 — TCP端口7443不得被阻止或限制
- 使用[Fiddler®](#)或[Wireshark®](#)等HTTP网络流量嗅探器确认浏览器通过TCP端口7443发送http-bind请求并接收响应
- 检查代理计算机和Finesse服务器之间的所有网络设备/接口，以查找过度延迟或丢包情况
  - Traceroute可用于确定路径并确定延迟
    - 在Microsoft® Windows® PC上：tracert {Finesse Server IP | Finesse服务器FQDN}
    - 在Mac上@traceroute {Finesse服务器IP | Finesse服务器FQDN}
    - 在Cisco IOS®软件上，可以检查接口统计信息：show interfaces
      - 请参阅[排除输入队列丢弃和输出队列丢弃故障](#)
- 收集测试代理的Finesse客户端日志。可通过三种方式收集客户端日志：
  1. 浏览器Web控制台日志
    - [Firefox Web控制台](#)
    - [Microsoft Edge Web控制台](#)
    - [Chrome Web控制台](#)
  2. 按Finesse页面上的[Send Error Report](#)按钮并收集Finesse服务器日志。日志位于/desktop/logs/clientlogs中。
  3. 通过https://<Finesse-FQDN>/desktop/locallog登录，并在问题发生后收集日志。

每分钟，客户端都会连接到Finesse服务器，以计算漂移和网络延迟：

```
<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
```

如果出现任何日志收集问题，请参阅[排除Cisco Finesse桌面持久性日志记录问题](#)

#### 2.不支持的浏览器和/或版本：

根据兼容性列表使用支持的浏览器/版本和设置：

[UCCE兼容性矩阵](#)

[UCCX兼容性列表](#)

3. 由于其他选项卡/窗口的内容/处理导致浏览器卡住的情况：

检查座席工作流程以查看他们是否执行以下操作：

- 通常具有持续运行其他实时应用程序(如音乐/视频流、WebSocket连接、自定义客户关系管理(CRM)Web客户端等的其他选项卡或窗口
- 打开大量选项卡或窗口
- 已禁用浏览器缓存
- 已长时间保持浏览器运行，在工作日结束时请勿关闭浏览器

4. 计算机进入睡眠状态：

检查代理是否在注销Finesse之前使其计算机进入睡眠状态，或其计算机睡眠设置计时器是否很低。

5. 客户端计算机上的CPU使用率高或内存不足问题：

- 如果代理浏览器在共享环境中运行，如Microsoft Windows Remote Desktop Services、Citrix® XenApp®、Citrix XenDesktop®，则确定浏览器性能是否取决于同时运行浏览器的用户数量
  - 确保根据用户数量配置正确的内存和CPU资源
- 检查计算机资源利用率问题：
  - Windows 窗口版本：
    - Windows [PowerShell Get-Counter](#)命令，每2秒检查一次CPU时间百分比、可用内存兆字节数和使用内存百分比：Get-Counter -Counter "\Processor(\_Total)\% Processor Time","\Memory\Available MBytes","\Memory\% Committed Bytes In Use" -SampleInterval 2 — 连续
    - 除了使用PowerShell查看Windows性能计数器外，还可使用[Windows性能监视器](#)
    - [任务管理器](#)可用于全局和逐个进程查看实时CPU和内存统计信息
  - MAC：
    - 检查实时CPU和内存总量的Terminal [Top命令：top](#)
      - 检查进程并按CPU利用率排序：前 — o CPU
      - 检查进程并按内存利用率排序：top -o MEM
    - [活动监控器](#)可用于全局查看实时CPU和内存统计信息，以及按进程查看的统计信息

6. 第三方小工具在后台执行意外、有问题的活动：

在删除所有第三方小工具的情况下测试Finesse桌面行为。

7. 服务器或客户端上的NTP问题：

- 检查Finesse发布服务器上的utils ntp status，以确保NTP服务器层为4或更低

- 在客户端日志中，检查漂移和网络延迟

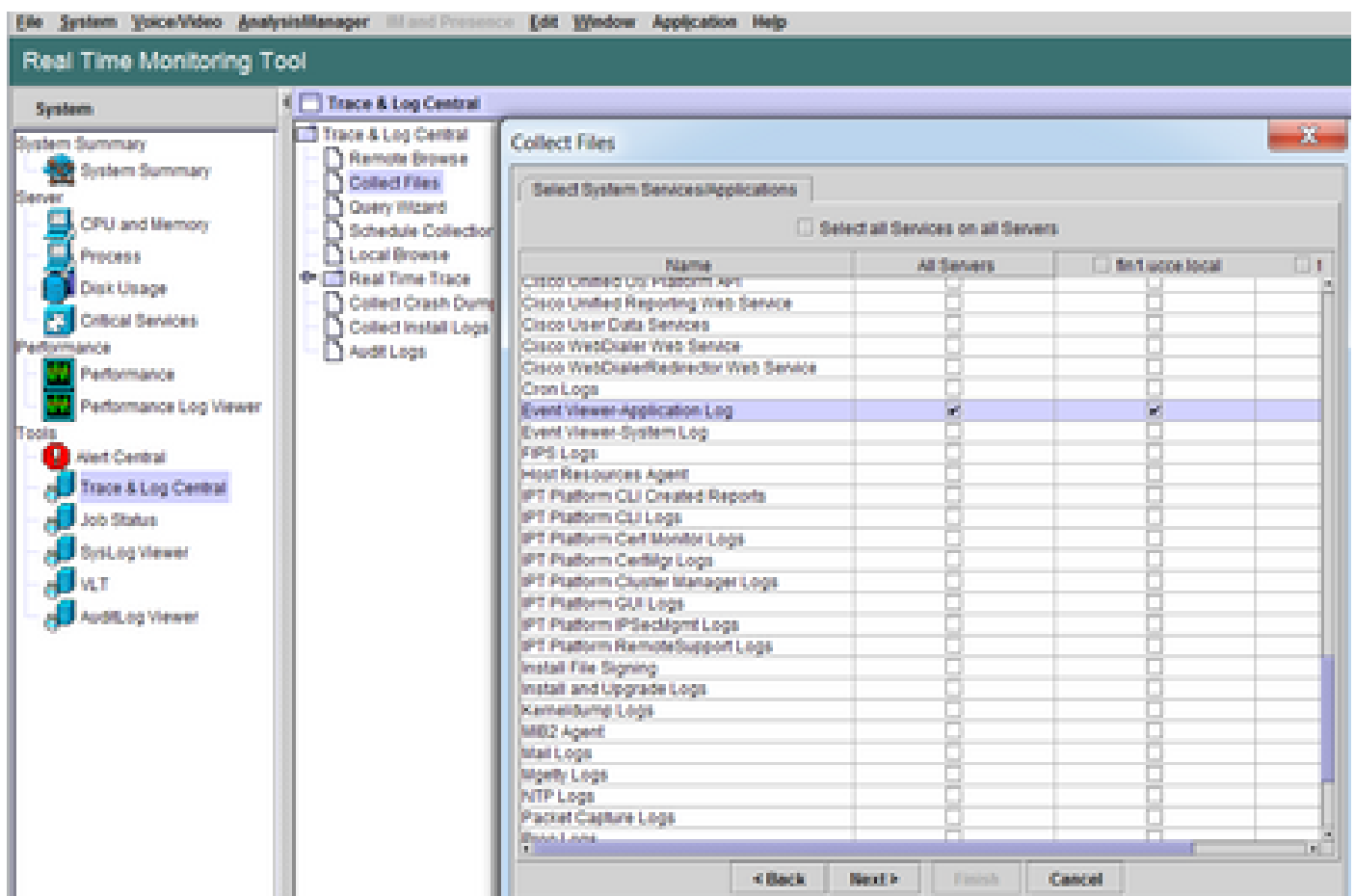
## 问题 — 所有代理同时断开（服务器端问题）

### 推荐的操作


检查以下问题：

1. Cisco Unified Communications Manager CTIManager服务断开。如果UCCX的所有CTIManager提供程序都处于关闭或崩溃状态，则UCCX代理会看到红色横幅错误。如果发生这种情况，UCCE代理不会看到红色标语，但呼叫无法正确路由到代理。


- 检查是否已在用作CTI提供程序的CUCM服务器上启动Cisco CTIanager服务
- 检查Cisco CTIManager服务是否通过RTMT上的事件查看器 — 应用程序日志崩溃，以查看Cisco CTIManager服务是否崩溃
  - 要在RTMT上收集事件查看器日志，请导航到System > Tools > Trace and Log Central > Collect Files > Select System Services/Applications > Event Viewer-Application Log。



- 在CLI上收集事件查看器 — 应用日志：file get activelog /syslog/CiscoSyslog\* abstime hh:mm:MM/DD/YY hh:mm:MM/DD/YY
- 要在CLI上查看核心转储：utils core active list

 注意：核心转储文件名使用的格式为

---

 : core.<ProcessID>.<SignalNumber>.<ProcessName>.<EpochTime>。

示例：core.24587.6.CTIManager.1533441238

因此，碰撞的时间可以从纪元时间确定。

---

## 2. Finesse/UCCX通知服务已停止或崩溃：

- 检查事件查看器 — 应用程序日志中是否存在通知服务错误，或查看服务是否已停止
- 检查通知服务是否已启动：utils服务列表
- 检查通知服务关闭的时间：file search activelog /desktop/logs/openfire "Openfire stopped"
- 检查通知服务的启动时间：文件搜索活动velog /desktop/logs/openfire "HTTP bind service started"
- 检查由崩溃引起的通知服务内存转储：file list activelog /desktop/logs/openfire/\*.hprof
- 检查通知服务是否正在侦听TCP端口7443上的流量：show open ports regexp 7443。  
\*LISTEN
- 检查这些缺陷是否适用（这些缺陷会导致登录的座席登录失败，对于已登录的座席，这些座席将看到红色横幅Finesse断开消息）：
  - Cisco Bug ID [CSCva72280](#) - Finesse Tomcat和Openfire Crash for invalid XML characters
  - Cisco Bug ID [CSCva72325](#) - UCCX:Finesse Tomcat和Openfire Crash ( XML字符无效 )

如果怀疑发生崩溃，请重新启动Cisco Finesse Tomcat和通知服务。只有在网络发生故障时才会建议这样做，否则会重新启动断开代理与Finesse服务器的连接。

### UCCE的步骤：

- 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的步骤：

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

## 使用Fiddler

如果不了解所需的步骤并了解Fiddler的工作方式，配置Fiddler可能会有些困难。Fiddler是一个中间人Web代理，位于Finesse客户端（Web浏览器）和Finesse服务器之间。由于Finesse客户端和Finesse服务器之间的连接是安全的，这为Fiddler配置增加了一层复杂性，以便查看安全的消息。

### 常见的Fiddler问题

由于Fiddler位于Finesse客户端和Finesse服务器之间，因此Fiddler应用需要为所有需要证书的

Finesse TCP端口创建签名证书：

### Cisco Finesse Tomcat服务证书

1. Finesse发布服务器TCP 8445 ( 和/或443 , 适用于UCCE )
2. Finesse用户服务器TCP 8445 ( 和/或443 , 适用于UCCE )

### Cisco Finesse(Unified CCX)通知服务证书

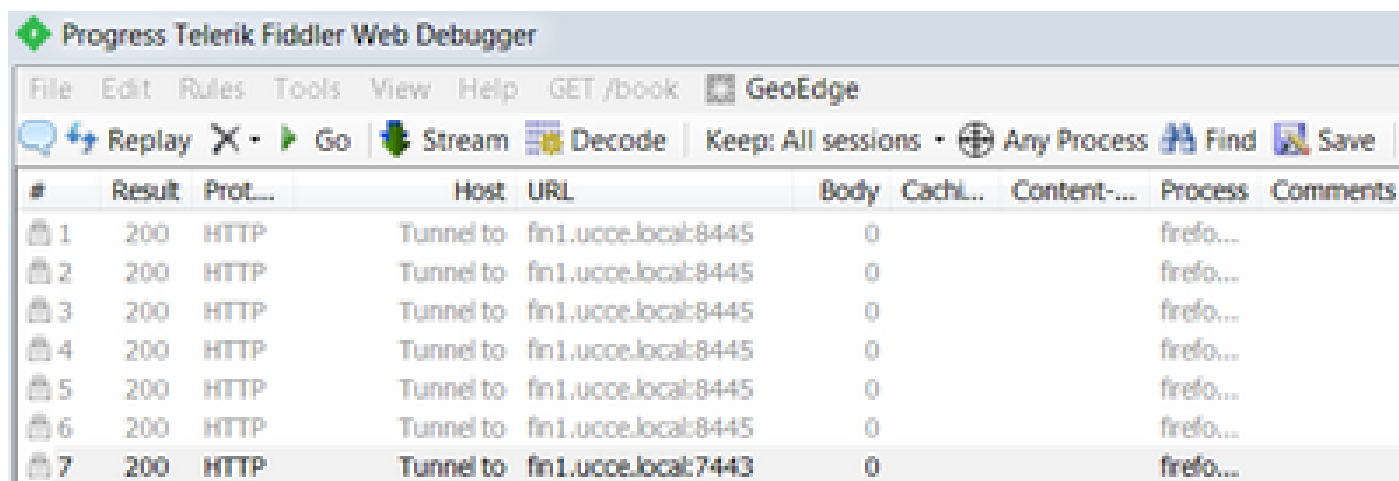
1. Finesse发布服务器TCP 7443
2. Finesse用户服务器TCP 7443

必须启用HTTPS解密，Fiddler才能代表Finesse服务器动态生成证书。默认情况下未启用此功能。

如果未配置HTTPS解密，则会看到与通知服务的初始隧道连接，但不会显示http-bind流量。

Fiddler仅显示：

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", "GET /book", and "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" column for the first six rows is "Tunnel to fin1.uccelocal:8445", and for the seventh row, it is "Tunnel to fin1.uccelocal:7443". The "Process" column for all rows is "firefox".

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

然后，客户端必须信任Fiddler签名的Finesse证书。如果这些证书不受信任，则无法通过Finesse登录的Establishing encrypted connection.. stage。




Establishing encrypted connection...

在某些情况下，从登录接受证书例外无法工作，并且需要浏览器手动信任证书。

#### 配置步骤示例

---

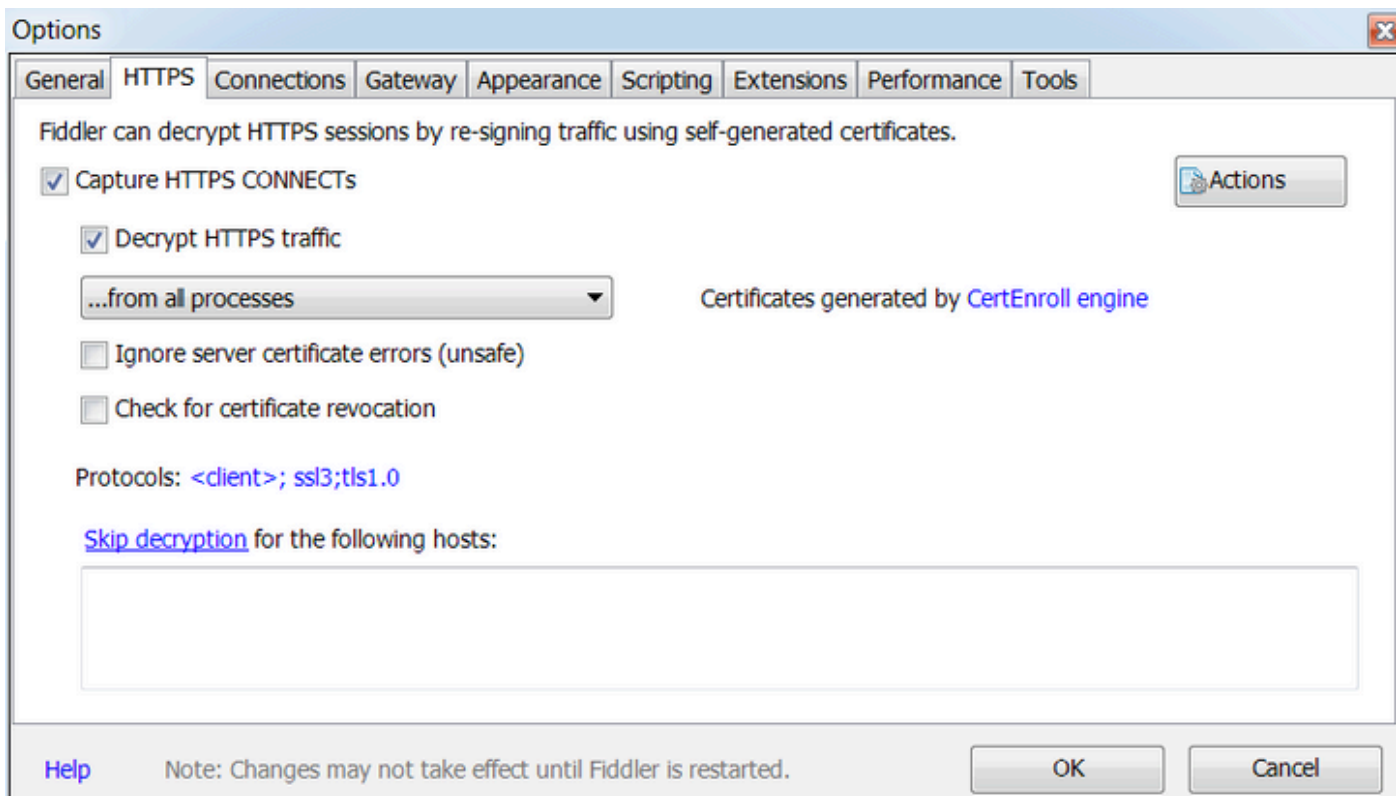
 注意：提供的示例配置适用于实验室环境中的Windows 7 x64上.NET 4.5的Fiddler v5.0.20182.28034和Mozilla Firefox 64.0.2 ( 32位 )。这些步骤不能推广到所有版本的Fiddler、所有浏览器或所有计算机操作系统。如果您的网络处于活动状态，请确保您了解任何配置的潜在影响。有关详细信息，请参阅[官方Fiddler文档](#)。

---

步骤1: 下载Fiddler

第二步：启用HTTPS解密。导航到工具>选项> HTTPS，然后选中解密HTTPS流量复选框。



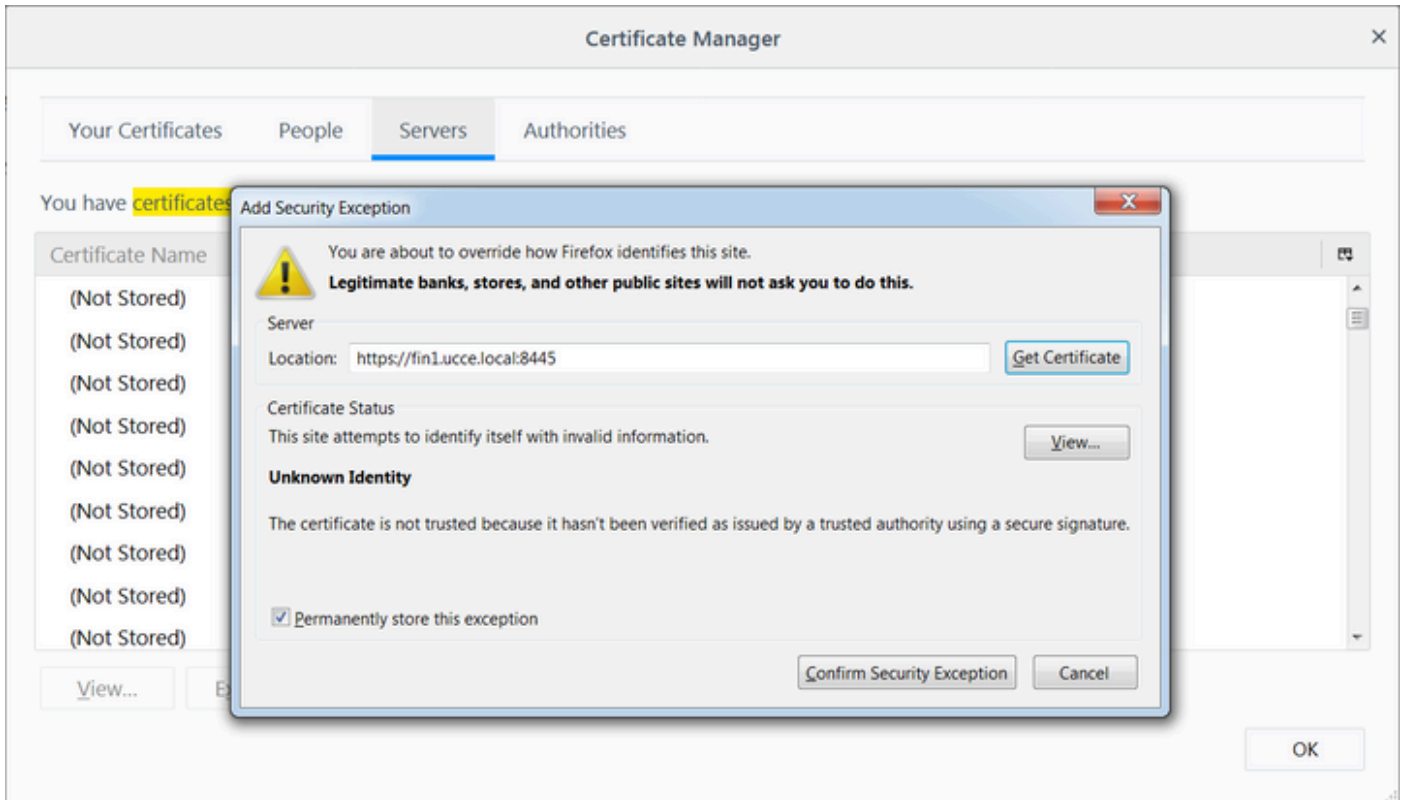


第三步：系统会打开一个警告消息框，要求信任Fiddler根证书。选择是。

第四步：系统打开一个警告消息框，显示消息“您将要安装来自证书颁发机构(CA)的证书，该证书声明表示：DO\_NOT\_TRUST\_FiddlerRoot.....是否要安装此证书？”。选择是。

第五步：手动将Finesse发布服务器和订用服务器证书添加到计算机或浏览器证书信任库。确保端口8445、7443和443（仅适用于UCCE）。例如，在Firefox上，无需从Finesse Operating System Administration页面下载证书即可完成此操作：

Options > Find in Options(search)> Certificates > Servers > Add Exception > Location > Enter https://<Finesse server>:port用于两个Finesse服务器的相关端口。



第六步：登录Finesse并查看http-bind消息通过Fiddler将Finesse客户端发送到Finesse服务器。

在提供的示例中，前5条消息显示由Finesse服务器响应的http-bind消息。第一条消息包含消息正文中返回的1571字节数据。正文包含有关代理事件的XMPP更新。最终http-bind消息已由Finesse客户端发送，但尚未从Finesse服务器获得响应。当您看到HTTP结果为null(-)且响应正文中的字节数为null(-1)时，可以确定这一点。

The screenshot shows the Fiddler Web Debugger interface. The main window displays a list of intercepted requests. A red box highlights a specific request with the following details:

#	Result	Prot...	Host	URL	Body	Cach...	Content...	Process	Comments	Custo
6...	200	HTTPS	fin1.ucce.local:...	/http-bind/	1,571		text/xml...	firefo...		

The right-hand pane shows the raw XML response for this request:

```

<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>1</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>07114e42-6b3c-4855-a4c9-af50ab5e7cc6</requestId>
                              <source>finesse/api/User/47483648</source>
                              </update>
                            </item>
                          </items>
                        </message>
                      </body>
                    
```

更近的数据视图：


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消息的响应正文：

```
<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>
```

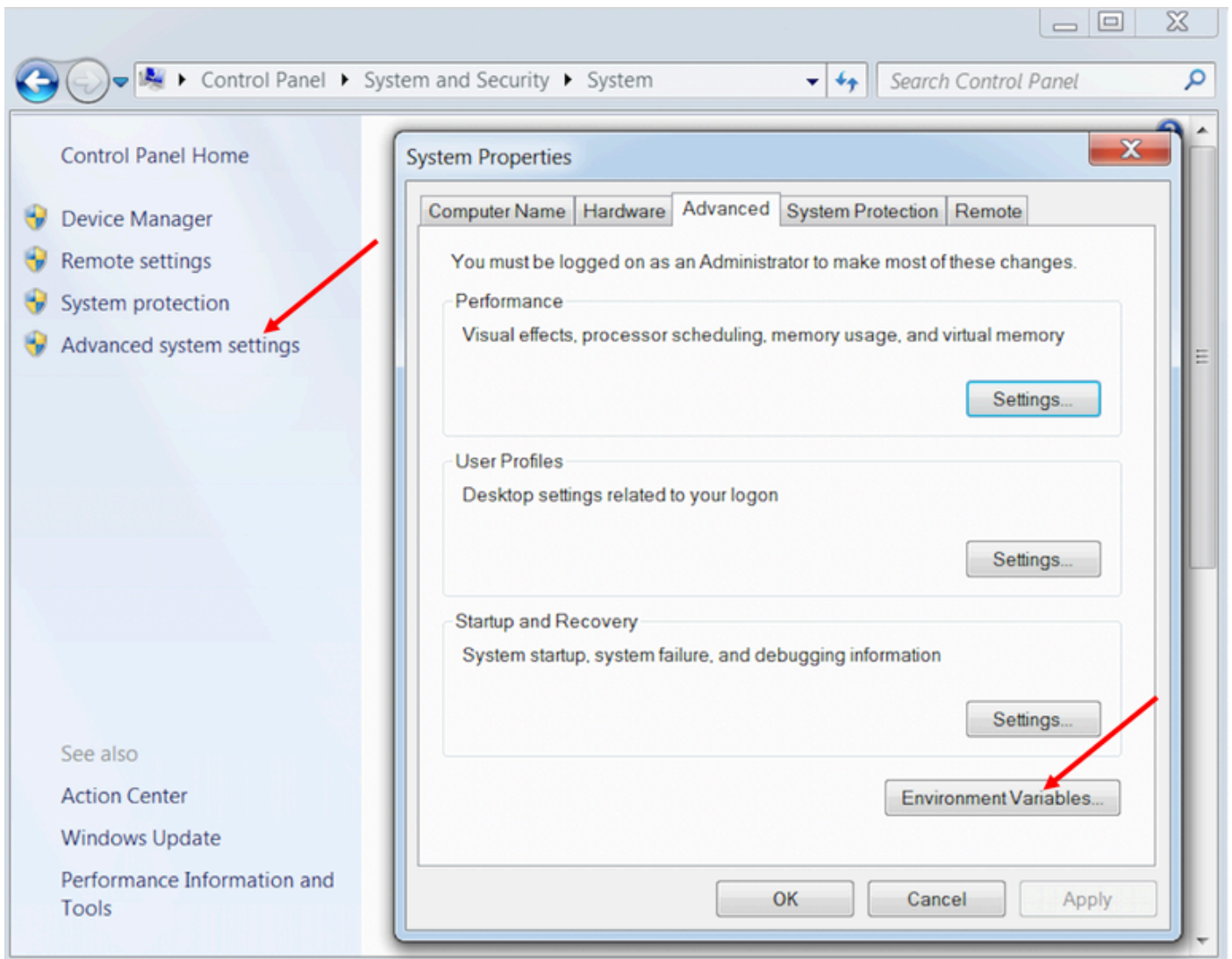
## 使用Wireshark

Wireshark是常用的数据包嗅探工具，可用于嗅探和解码HTTPS流量。HTTPS流量是通过传输层安全(TLS)保护的HTTP流量。TLS为主机提供完整性、身份验证和机密性。它通常用于Web应用程序，但可以与任何使用TCP作为传输层协议的协议配合使用。安全套接字层(SSL)是TLS协议的旧版本，由于不安全，因此不再使用。这些名称经常互换使用，用于SSL或TLS流量的Wireshark过滤器是ssl。

 注意：提供的示例配置适用于实验室环境中Windows7 x64上的Wireshark 2.6.6(v2.6.6-0-gdf942cd8)和Mozilla Firefox 64.0.2 (32位)。这些步骤不能推广到所有版本的Fiddler、所有浏览器或所有计算机操作系统。如果您的网络处于活动状态，请确保您了解任何配置的潜在影响。有关详细信息，请参阅[官方Wireshark SSL文档](#)。需要Wireshark 1.6或更高版本。

 注：此方法只能用于Firefox和Chrome。此方法对Microsoft Edge不起作用。

步骤1: 在代理的Windows PC上，导航到控制面板>系统和安全>系统>高级系统设置环境变量.....

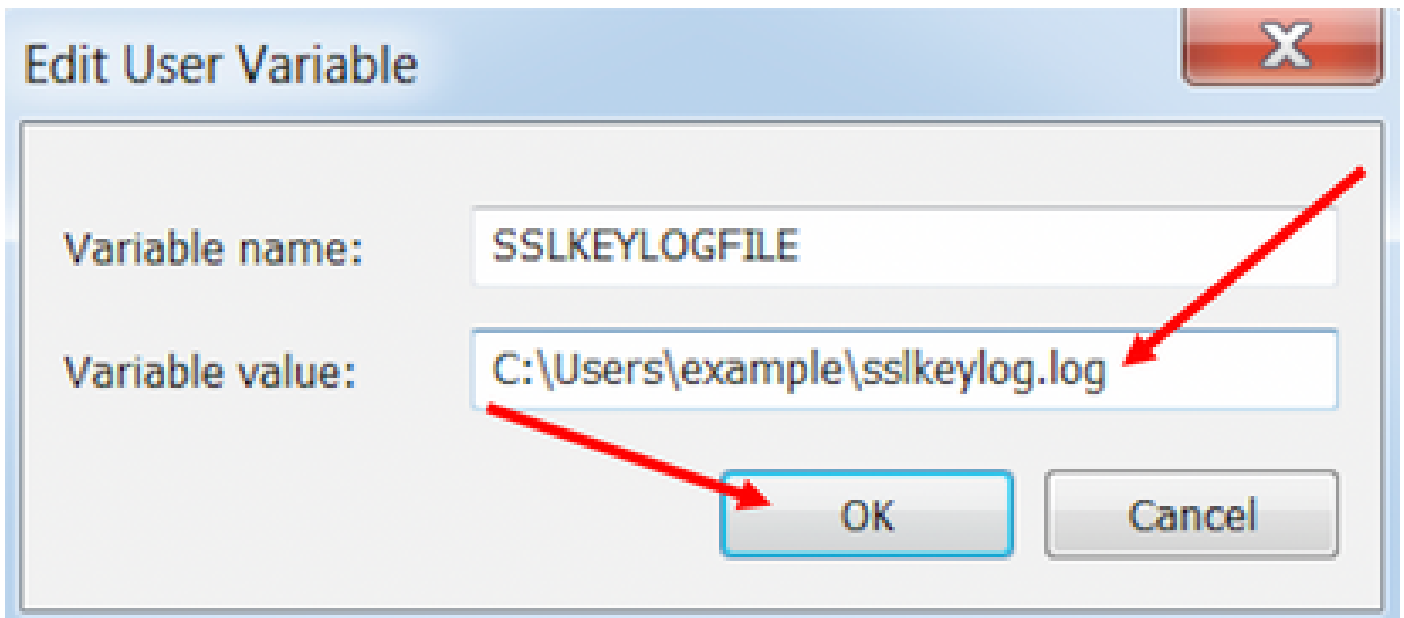



第二步：导航到User <username> > New...的用户变量。

创建名为SSLKEYLOGFILE的变量。

创建一个文件，将SSL预主密钥存储在专用目录中

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





 注：创建系统变量而不是用户变量并/或将文件存储在非专用目录中，但系统上的所有用户都可以访问安全性较低的premaster密钥。

第三步：如果Firefox或Chrome处于打开状态，请关闭应用程序。重新打开后，它们可以开始写入SSLKEYLOGFILE。

第四步：在Wireshark上，导航到编辑>首选项.....

# 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



## 关于此翻译

思科采用人工翻译与机器翻译相结合的方式将此文档翻译成不同语言，希望全球的用户都能通过各自的语言得到支持性的内容。

请注意：即使是最好的机器翻译，其准确度也不及专业翻译人员的水平。

Cisco Systems, Inc. 对于翻译的准确性不承担任何责任，并建议您总是参考英文原始文档（已提供链接）。