Finesse BOSH 구현 이해 및 트러블슈팅

목차

소개

이 문서에서는 BOSH를 사용하는 Finesse 연결의 아키텍처 및 BOSH 연결 문제를 진단하는 방법에 대해 설명합니다.

사전 요구 사항

요구 사항

다음 주제에 대한 지식을 보유하고 있으면 유용합니다.

- Cisco Finesse
- UCCE(Unified Contact Center Enterprise)
- UCCX(Unified Contact Center Express)
- 웹 브라우저 개발자 도구
- Windows 및/또는 Mac 관리

사용되는 구성 요소

이 문서의 정보는 다음 소프트웨어 및 하드웨어 버전을 기반으로 합니다.

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

이 문서의 정보는 특정 랩 환경의 디바이스를 토대로 작성되었습니다. 이 문서에 사용된 모든 디바이스는 초기화된(기본) 컨피그레이션으로 시작되었습니다. 현재 네트워크가 작동 중인 경우 모든 명령의 잠재적인 영향을 미리 숙지하시기 바랍니다.

배경 정보

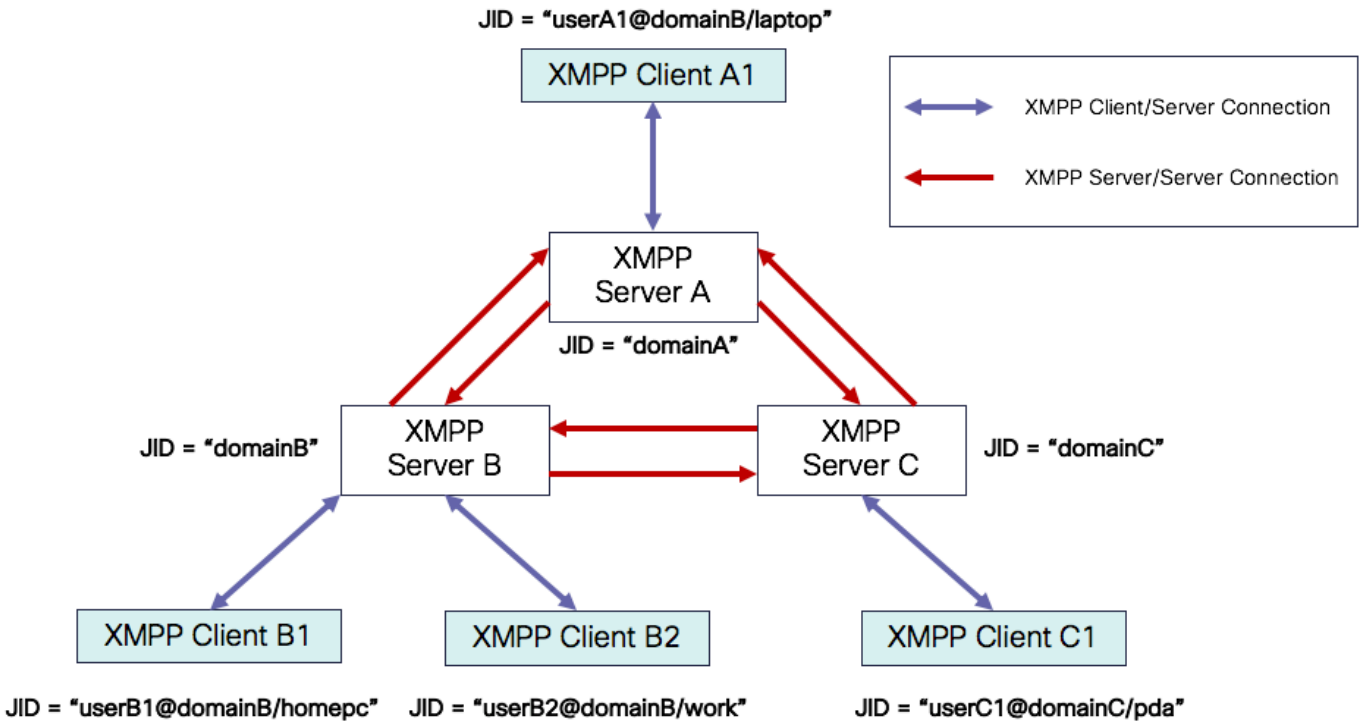
동기식 HTTP를 통한 양방향 스트림을 사용하는 연결을 BOSH라고 합니다.

Finesse BOSH 구현 이해

XMPP 이해

XMPP(Extensible Messaging and Presence Protocol)(Jabber라고도 함)는 클라이언트 서버 모델의 상태 기반 프로토콜입니다. XMPP를 사용하면 정형 eXtensible XML(Markup Language) 데이터의 작은 부분을 한 엔터티에서 다른 엔터티로 빠르게 전달할 수 있습니다. XMPP/Jabber는 IM(인스턴트 메시징) 및 프레즌스 애플리케이션에서 널리 사용됩니다.

모든 XMPP 엔터티는 해당 JID(Jabber ID)로 식별됩니다.



JID 주소 지정 체계: user@domain/resource

사용자	XMPP 서버의 클라이언트 사용자 이름 또는 회의실 이름
도메인	XMPP 서버 FQDN(정규화된 도메인 이름)
자원	사용자의 특정 엔터티/엔드포인트(예: 랩톱, 스마트폰 등)의 식별자, 세션 식별자 또는 pubsub 노드 이름

✎ 참고: 세 가지 JID 구성 요소가 모두 모든 경우에 사용되지는 않습니다. 일반적으로 서버는 도메인, 회의실은 user@domain, 클라이언트는 user@domain/resource로 정의됩니다.

XMPP 메시지를 stanzas라고 합니다. XMPP에는 3개의 코어 스탠자가 있습니다.

1. <메시지>: 한 방향, 한 수신인

2. <presence>: 한 가지 방향, 여러 곳에 게시

3. <iq>: 정보/쿼리 - 요청/응답

모든 stanza는 주소에서 보내고 받을 수 있으며 대부분의 stanza에는 type, id 및 xml:langattributes도 있습니다.

Stanza 특성	목적
수신	대상 JID
발신	소스 JID
유형	메시지의 목적
ID	<iq> stanzas에 대한 응답과 요청을 연결하는 데 사용되는 고유 식별자
xml:언어	사용자가 읽을 수 있는 모든 XML의 기본 언어를 정의합니다.

XMPP 메시지 예

```
<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>
```

Finesse를 사용한 XMPP 구현

웹 애플리케이션이 XMPP와 연동해야 하는 경우 여러 문제가 발생합니다. 브라우저에서는 기본적으로 XMPP over TCP(Transmission Control Protocol)를 지원하지 않으므로 모든 XMPP 트래픽은 브라우저 내에서 실행되는 프로그램에서 처리해야 합니다. 웹 서버와 브라우저는 HTTP(HyperText Transfer Protocol) 메시지를 통해 통신하므로 Finesse 및 기타 웹 애플리케이션은 XMPP 메시지를 HTTP 메시지 내에 래핑합니다.

이 접근 방식의 첫 번째 문제점은 HTTP가 상태 비저장 프로토콜이라는 것입니다. 이는 각 HTTP 요청이 다른 요청과 관련이 없음을 의미합니다. 그러나 이 문제는 예를 들어 쿠키/게시물 데이터를 사용하는 등 응용 가능한 방법으로 해결할 수 있습니다.

두 번째 문제는 HTTP의 단방향 동작입니다. 클라이언트만 요청을 전송하고 서버는 응답만 할 수 있습니다. 서버에서 데이터를 푸시할 수 없기 때문에 HTTP를 통해 XMPP를 구현하는 것이 부자연스럽습니다.

이 문제는 XMPP가 TCP에 바인딩되는 원래 XMPP 코어 사양(RFC 6120)에는 없습니다. 그러나 예를 들어 Javascript에서 HTTP 요청을 보낼 수 있기 때문에 HTTP에 바인딩된 XMPP로 문제를 해결하려는 경우 두 가지 해결 방법이 있습니다. 둘 다 HTTP와 XMPP 간에 브리지가 필요합니다.

제안 솔루션은 다음과 같습니다.

1. 폴링(레거시 프로토콜): XEP-0025에 정의된 새 데이터를 요청하는 반복적인 HTTP 요청: Jabber HTTP Polling

2. Long Polling을 BOSH라고도 합니다. XEP-0124: HTTP Binding and extended by XEP-0206: XMPP Over BOSH에 정의된 빈번한 폴링을 사용하지 않고 여러 동기 HTTP 요청/응답 쌍을 효율적으로 사용하여 두 엔터티 간의 긴 수명 양방향 TCP 연결의 의미를 에뮬레이트하는 전송 프로토콜입니다.

Finesse는 서버 로드 관점에서 매우 효율적이고 트래픽 와이즈이므로 BOSH를 구현합니다. BOSH를 사용하는 이유는 서버가 요청이 있는 즉시 대응할 필요가 없다는 사실을 은폐하기 위해서다. 서버에 클라이언트에 대한 데이터가 있을 때까지 응답이 지정된 시간까지 지연된 다음 응답으로서 전송됩니다. 고객이 응답을 받는 즉시 고객은 새로운 요청 등을 합니다.

Finesse 데스크톱 클라이언트(웹 애플리케이션)는 30초마다 TCP 포트 7443을 통해 오래된 BOSH 연결을 설정합니다. 30초 후에 Finesse Notification Service에서 업데이트가 없는 경우 알림 서비스는 200 OK 및 (거의) 빈 응답 본문이 포함된 HTTP 응답을 보냅니다. 알림 서비스에 에이전트 또는 대화(통화) 이벤트 존재 여부에 대한 업데이트가 있는 경우 데이터는 Finesse 웹 클라이언트로 즉시 전송됩니다.

Finesse XMPP 요청/응답 예

이 예에서는 BOSH 연결을 설정하기 위해 Finesse 클라이언트와 Finesse 서버 간에 공유되는 첫 번째 XMPP 메시지 요청 응답을 보여 줍니다.

```
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
```

요약하자면,

1. Finesse 웹 클라이언트에는 TCP 포트 7443을 통해 Finesse 서버에 대한 부실 HTTP 연결(http-bind)이 설정되어 있습니다. 이 설문조사는 BOSH의 긴 설문조사라고 합니다.

2. Finesse 알림 서비스는 상담원 상태, 통화 등에 대한 업데이트를 게시하는 프레즌스 서비스입니다.

3. 알림 서비스에 업데이트가 있는 경우 HTTP 응답 본문에 XMPP 메시지로 상태 업데이트를 사용하여 http-bind 요청에 응답합니다.

4. http-bind 요청을 받은 후 30초 후에 상태 업데이트가 없는 경우 알림 서비스는 상태 업데이트 없이 회신하여 Finesse 웹 클라이언트가 다른 http-bind 요청을 보낼 수 있도록 합니다. 이는 알림 서비스가 Finesse 웹 클라이언트가 여전히 알림 서비스에 연결할 수 있으며 에이전트가 브라우저를 닫거나 컴퓨터를 절전 모드로 전환하지 않았음을 알리는 등의 역할을 합니다.

Finesse XMPP 메시지 및 XMPP 노드 이해

Finesse는 XMPP 사양 XEP-0060: Publish-Subscribe도 구현합니다. 이 사양의 목적은 XMPP 서버(알림 서비스)가 XMPP 노드(항목)에 게시된 정보를 가져온 다음 노드에 가입된 엔터티에 XMPP 이벤트를 보낼 수 있도록 하는 것입니다. Finesse의 경우 CTI(Computer Telephony Integration) 서버는 CTI 메시지를 Finesse 웹 서비스에 전송하여 에이전트 또는 CSQ(Contact Service Queue) 생성 또는 통화 정보와 같은 컨피그레이션 업데이트에 대해 Finesse에게 알립니다. 그런 다음 이 정보는 Finesse 웹 서비스가 Finesse 알림 서비스에 게시하는 XMPP 메시지로 변환됩니다. 그런 다음 Finesse Notification 서비스는 특정 XMPP 노드에 가입된 에이전트에게 BOSH 메시지를 통해 XMPP를 전송합니다.

Finesse Web Services Developer Guide에 정의된 일부 Finesse API 객체는 XMPP 노드입니다. 에이전트 및 수퍼바이저 Finesse 웹 클라이언트는 실시간 이벤트(통화 이벤트, 상태 이벤트 등)에 대한 최신 정보를 제공하기 위해 이러한 XMPP 노드 중 일부에 대한 이벤트 업데이트를 구독할 수 있습니다. 이 표에서는 pubsub가 활성화된 XMPP 노드를 보여 줍니다.

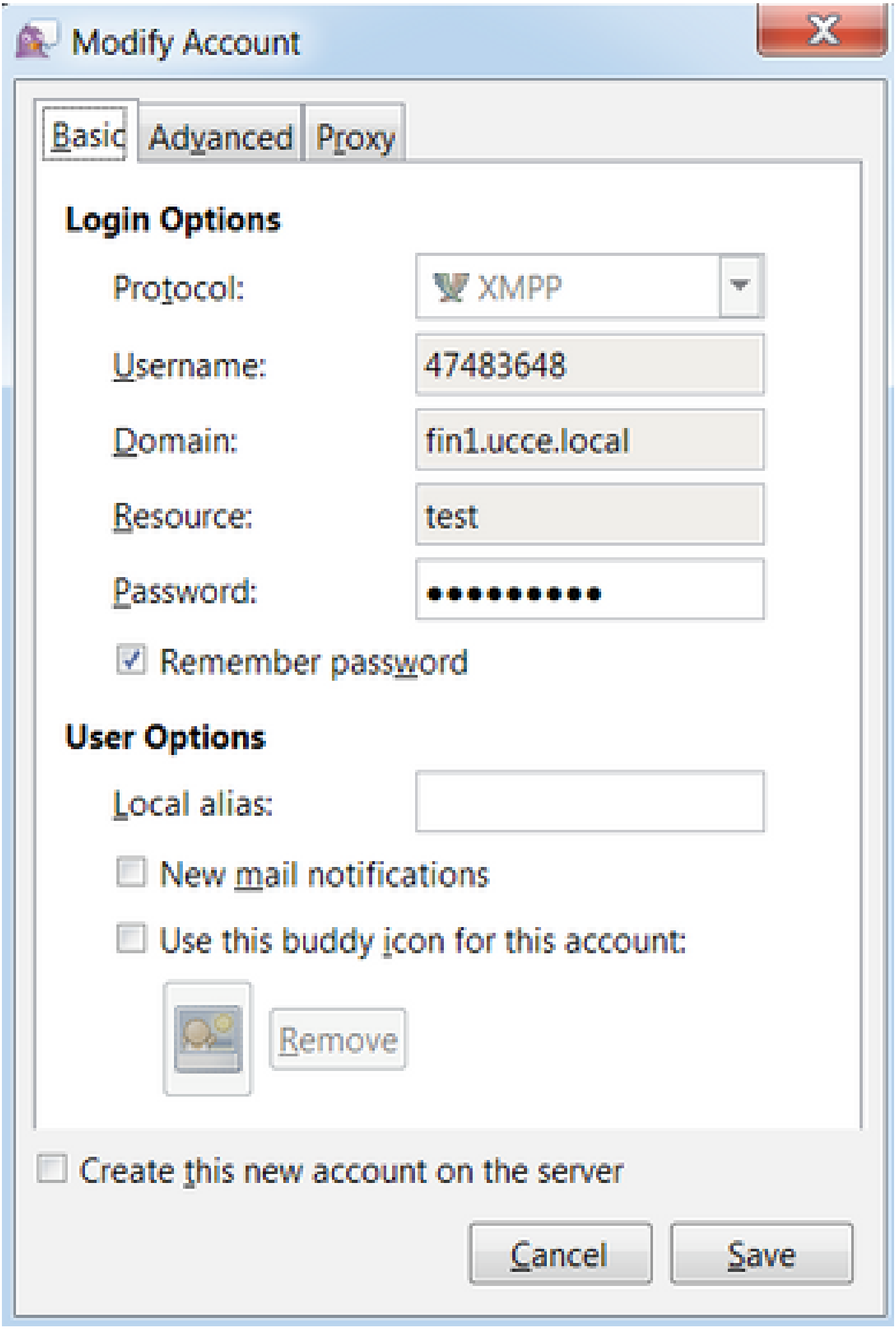
Finesse API 객체	목적	서브스크립션
/finesse/api/User/<로그인 ID>	상담원의 상태 및 팀 매핑을 표시합니다	상담원 및 수퍼바이저
/finesse/api/User/<LoginID>/대화 상자	상담원이 처리한 통화를 표시합니다	상담원 및 수퍼바이저
/finesse/api/User/<로그인 ID>/클라이언트 로그	Send Error Report(오류 보고서 보내기) 버튼에서 클라이언트 로그를 캡처하는 데 사용됩니다	상담원 및 수퍼바이저
/finesse/api/User/<LoginID>/Queue/<queueID>	대기열 통계 데이터 표시(활성화된 경우)	상담원 및 수퍼바이저
/finesse/api/Team/<팀 ID>/사용자	상태 정보를 포함하여 특정 팀에 속한 상담원을 표시합니다	수퍼바이저
/finesse/api/SystemInfo	Finesse 서버의 상태를 표시합니다. 장애 조치가 필요한지 확인하는 데 사용됩니다.	상담원 및 수퍼바이저

예 1: Pidgin을 사용하여 Finesse XMPP 노드 보기

1단계. XMPP 클라이언트 Pidgin을 다운로드하여 설치합니다.

2단계. Accounts(어카운트) > Modify(수정) > Basic(기본)으로 이동하고 로그인 옵션을 구성합니다.

- 프로토콜: XMPP
- 사용자 이름: 모든 에이전트의 LoginID
- 도메인: Finesse 서버의 FQDN
- 리소스: 자리 표시자 - 테스트 등 모든 값을 사용할 수 있습니다.
- 비밀번호: 에이전트 비밀번호
- Remember password(비밀번호 기억) 확인란을 선택합니다


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

구독 이벤트를 받을 수 없으므로 표시됩니다. 따라서 에이전트 데스크톱에 상태 정보 및 통화 세부 정보를 표시할 수 없습니다.

UCCX의 경우 브라우저 연결이 끊긴 후 60초가 지나면 에이전트가 로그아웃 상태가 됩니다. 상담원은 로그아웃이 발생할 준비 또는 준비 안 됨 상태가 될 수 있습니다.

UCCE의 경우 에이전트가 브라우저를 닫거나 브라우저가 충돌할 때 Finesse가 탐지하는 데 최대 120초가 걸리고 CTI 서버에 강제 로그아웃 요청을 보내기 전에 60초가 대기합니다. 그러면 CTI 서버가 에이전트를 준비 안 됨 상태로 전환합니다. 이러한 조건에서 Finesse는 에이전트를 로그아웃하는 데 최대 180초가 걸릴 수 있습니다. UCCX와 달리 에이전트는 Logout(로그아웃) 상태 대신 Not Ready(준비 안 됨) 상태로 전환됩니다.

참고: UCCE의 CTI 연결 해제 준비 안 됨 대 로그아웃 상태 동작은 PG /LOAD 매개 변수에 의해 제어됩니다. Unified Contact Center Enterprise 및 Hosted Release 10.0(1)의 릴리스 정보에 따르면 /LOAD 매개 변수는 UCCE 10.0부터 더 이상 사용되지 않습니다.

UCCE Finesse Desktop 동작에 대한 자세한 내용은 Cisco Finesse 관리 설명서의 Cisco Finesse 장애 조치 메커니즘 장에서 Desktop Behavior 섹션을 참조하십시오.

참고: 타이머 값은 제품 요구 사항에 따라 나중에 변경될 수 있습니다.

로그 분석

Finesse 및 UCCX 알림 서비스 로그는 RTMT 또는 CLI를 통해 수집할 수 있습니다.

파일 get activelog /desktop recurs compress

디버그 알림 서비스 로그

참고: 문제를 재현하는 동안에만 디버그 레벨 로그를 설정합니다. 문제가 재현된 후 디버그를 끕니다.

참고: Finesse 9.0(1)에는 디버그 레벨 로깅이 없습니다. 디버그 레벨 로깅이 Finesse 9.1(1)에 도입되었습니다. 로깅을 활성화하는 프로세스는 9.1(1)에서 Finesse 10.0(1) - 11.6(1)과 다릅니다. 이 프로세스에 대한 내용은 Finesse 관리 및 서비스 가용성 가이드를 참조하십시오.

다음과 같이 UCCX(Unified Contact Center Express)의 알림 서비스 디버그 로그를 활성화합니다.

<#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.
```

다음과 같이 UCCE(Unified Contact Center Enterprise)의 알림 서비스 디버그 로그(Finesse Standalone)를 활성화합니다.

<#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
```

이러한 로그는 /desktop/logs/openfire 폴더에 있으며 이름이 debug.log입니다.

이미지에 표시된 대로 알림 서비스(Openfire) debug.log는 데스크톱과 함께 에이전트 PC의 IP 주소 및 포트와 http 바인딩을 표시합니다.

```
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$SelectChannelHttpConnection@2d[illegible]a26@XXX.XXX.XXX.XX:7443<->XXX.XXX.XXX.X(:49805
2017.04.14 21:34:21 scope null||/http-bind/ @ o.e.j.s.ServletContextHandler{/http-bind,null}
2017.04.14 21:34:21 context=/http-bind||/ @ o.e.j.s.ServletContextHandler{/http-bind,null}
2017.04.14 21:34:21 sessionManager=org.eclipse.jetty.server.session.HashSessionManager@176fe4#STARTED
2017.04.14 21:34:21 session=null
2017.04.14 21:34:21 session=null
2017.04.14 21:34:21 servlet /http-bind||/ -> org.jivesoftware.openfire.http.HttpBindServlet-1643193
2017.04.14 21:34:21 chain=null
2017.04.14 21:34:21 HTTPBindLog: HTTP RECV(3445afbe):<body sid="3445afbe" rid="164053266"/>
2017.04.14 21:34:21 consumeResponse: org.jivesoftware.openfire.http.HttpSession@dd7653 status: 3 address: 1001003@nXXX.XXXX.XXX.XXX.XX.cisco.com/desktop id: 3445afbe presence:
<presence from="1001003@nXXX.XXXX.XXX.XXX.XX.cisco.com/desktop">
  <c xmlns="http://jabber.org/protocol/caps" hash="sha-1" node="http://jabber.cisco.com/caxl" ver="VNC6fNwvCxe6FJfDJIpLryVJRwM="/>
</presence> rid: 164053266
2017.04.14 21:34:21 suspended org.eclipse.jetty.server.nio.SelectChannelConnector$SelectChannelHttpConnection@2d[illegible]a26@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
```

이미지에 표시된 것처럼 마지막 활성 0ms는 세션이 여전히 활성 상태임을 나타냅니다.

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

Openfire가 유휴 세션을 닫으면 60초 내에 상담원 로그아웃이 트리거될 수 있으며 Finesse는 이유 코드 255를 사용하여 강제 로그아웃을 CTI 서버로 전송할 수 있습니다. 이러한 조건에서 데스크톱의 실제 동작은 UCCE의 LOAD(Agent Disconnect) 시 로그아웃 설정에 따라 달라집니다. UCCX에

서는 항상 이러한 동작이 수행됩니다.

Finesse 클라이언트가 http-bind 메시지를 Finesse 서버로 전송하지 않으면 로그는 세션 가동 시간을 표시하고 세션 닫기를 표시할 수 있습니다.

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

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

정보 알림 서비스 로그

이러한 로그는 /desktop/logs/openfire 폴더에 있으며 info.log라는 이름이 지정됩니다. Finesse 클라이언트가 http 바인드 메시지를 Finesse 서버로 전송하지 않을 경우, 로그는 세션이 비활성화되었음을 나타낼 수 있습니다.

```
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
```

웹 서비스 로그

이러한 로그는 /desktop/logs/webservices 폴더에 있으며 이름은 Desktop-webservices.YYYY-MM-DDTHH-MM-SS.sss.log입니다. Finesse 클라이언트가 지정된 시간 내에 http 바인딩 메시지를 Finesse 서버로 전송하지 않으면 로그는 에이전트 프레즌스를 사용할 수 없게 되고 60초 후에 프레즌스 기반 로그아웃이 발생할 수 있음을 표시할 수 있습니다.

```
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 연결이 끊긴 일반적인 이유

BOSH 연결은 웹 클라이언트에 의해 설정되며 Finesse 서버는 에이전트 프레즌스를 사용할 수 없는지 확인합니다. 이러한 문제는 거의 항상 브라우저, 에이전트 컴퓨터 또는 네트워크와 관련된 클라이언트 측 문제입니다. 연결 시작 시 발생하는 문제는 클라이언트에 있습니다.

문제 - 상담원이 서로 다른 시간에 연결 끊김(클라이언트 측 문제)

권장 작업

다음 문제를 확인합니다.

1. 네트워크 문제

- 방화벽 규칙 및 로그 검토 - TCP 포트 7443을 차단하거나 제한해서는 안 됩니다.
- 브라우저가 TCP 포트 7443을 통해 http-bind 요청을 전송하고 응답을 수신하는지 확인하려면 Fiddler® 또는 Wireshark®와 같은 HTTP 웹 트래픽 스니퍼를 사용합니다
- 에이전트 컴퓨터와 Finesse 서버 간의 모든 네트워크 디바이스/인터페이스에서 과도한 지연 또는 패킷 삭제를 확인합니다.
    - Traceroute는 경로를 확인하고 지연을 확인하는 데 유용합니다
        - Microsoft® Windows® PC: tracert {Finesse Server IP | Finesse 서버 FQDN}
        - Mac에서® traceroute {Finesse 서버 IP | Finesse 서버 FQDN}
        - Cisco IOS® 소프트웨어에서 인터페이스 통계를 확인할 수 있습니다. show interfaces
            - 입력 대기열 삭제 및 출력 대기열 삭제 트러블슈팅을 참조하십시오.
- 테스트 에이전트에 대한 Finesse 클라이언트 로그를 수집합니다. 클라이언트 로그는 다음 세 가지 방법으로 수집할 수 있습니다.
    1. 브라우저 웹 콘솔 로그
        - Firefox 웹 콘솔
        - Microsoft Edge 웹 콘솔
        - Chrome 웹 콘솔
    2. Finesse 페이지에서 Send Error Report(오류 보고서 보내기) 버튼을 누르고 Finesse 서버 로그를 수집합니다. 로그는 /desktop/logs/clientlogs에 있습니다.
    3. 문제가 발생한 후 https://<Finesse-FQDN>/desktop/locallog를 통해 로그인하고 로그를 수집합니다.

클라이언트는 1분마다 Finesse 서버에 연결하여 드리프트 및 네트워크 레이턴시를 계산합니다.

```
<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
```

로그 수집 문제가 있는 경우 Cisco Finesse Desktop 영구 로깅 문제 해결을 참조하십시오

2. 지원되지 않는 브라우저 또는 버전:

호환성 매트릭스에 따라 지원되는 브라우저/버전 및 설정을 사용합니다.

UCCE 호환성 매트릭스

UCCX 호환성 매트릭스

3. 다른 탭/창의 콘텐츠/처리로 인한 브라우저 고착 상태

상담원 워크플로에서 다음 항목이 있는지 확인합니다.

- 일반적으로 음악/비디오 스트리밍, WebSocket 연결, 사용자 지정 CRM(고객 관계 관리) 웹 클라이언트 등과 같은 다른 실시간 응용 프로그램을 지속적으로 실행하는 다른 탭이나 창을 가동합니다.
- 탭 또는 창을 매우 많이 열어 둡니다.
- 브라우저 캐싱을 비활성화했습니다.
- 오랫동안 브라우저를 실행 중이고 근무일이 끝날 때 브라우저를 닫지 않음

4. 컴퓨터 절전 모드

상담원이 Finesse에서 로그아웃하기 전에 컴퓨터를 절전 모드로 전환했는지 또는 컴퓨터 절전 모드 설정 타이머가 매우 낮은지 확인합니다.

5. 클라이언트 컴퓨터의 CPU 또는 메모리 부족 문제:

- 에이전트 브라우저가 Microsoft Windows Remote Desktop Services, Citrix® XenApp®, Citrix XenDesktop®와 같은 공유 환경에서 실행되는 경우 브라우저 성능이 브라우저를 동시에 실행하는 사용자 수에 따라 달라지는지 확인합니다
    - 올바른 메모리 및 CPU 리소스가 사용자 수에 따라 구성되었는지 확인합니다

- 컴퓨터 리소스 사용률 문제 확인:
    - 창:
        - Windows PowerShell Get-Counter 명령 - CPU 시간의 %, 사용 가능한 메모리의 MB 및 사용 중인 메모리의 %를 2초마다 확인: Get-Counter -Counter "\Processor(_Total)\% Processor Time","\Memory\Available MBytes","\Memory\% Committed Bytes In Use" -SampleInterval 2 -Continuous
        - PowerShell을 사용하여 Windows 성능 카운터를 보는 대신 Windows 성능 모니터를 사용할 수 있습니다
        - Task Manager를 사용하여 전 세계적으로 그리고 프로세스별로 라이브 CPU 및 메모리 통계를 볼 수 있습니다
    - Mac:
        - 전체 CPU와 메모리를 확인하는 Terminal Top 명령: top
            - 프로세스 확인 및 CPU 사용률별 정렬: 상위 -o CPU
            - 프로세스 확인 및 메모리 사용률별 정렬: 상위 -o MEM
        - Activity Monitor를 사용하여 전 세계적으로 그리고 프로세스별로 라이브 CPU 및 메모리 통계를 확인할 수 있습니다

6. 백그라운드에서 예기치 않은 문제 활동을 수행하는 서드파티 가젯:

모든 서드파티 가젯을 제거한 상태에서 Finesse 데스크톱 동작을 테스트합니다.

7. 서버 또는 클라이언트의 NTP 문제:

- Finesse 게시자 서버에서 utils ntp 상태를 확인하여 NTP 서버 계층이 4 이하인지 확인합니다
- 클라이언트 로그에서 드리프트 및 네트워크 레이턴시를 확인합니다

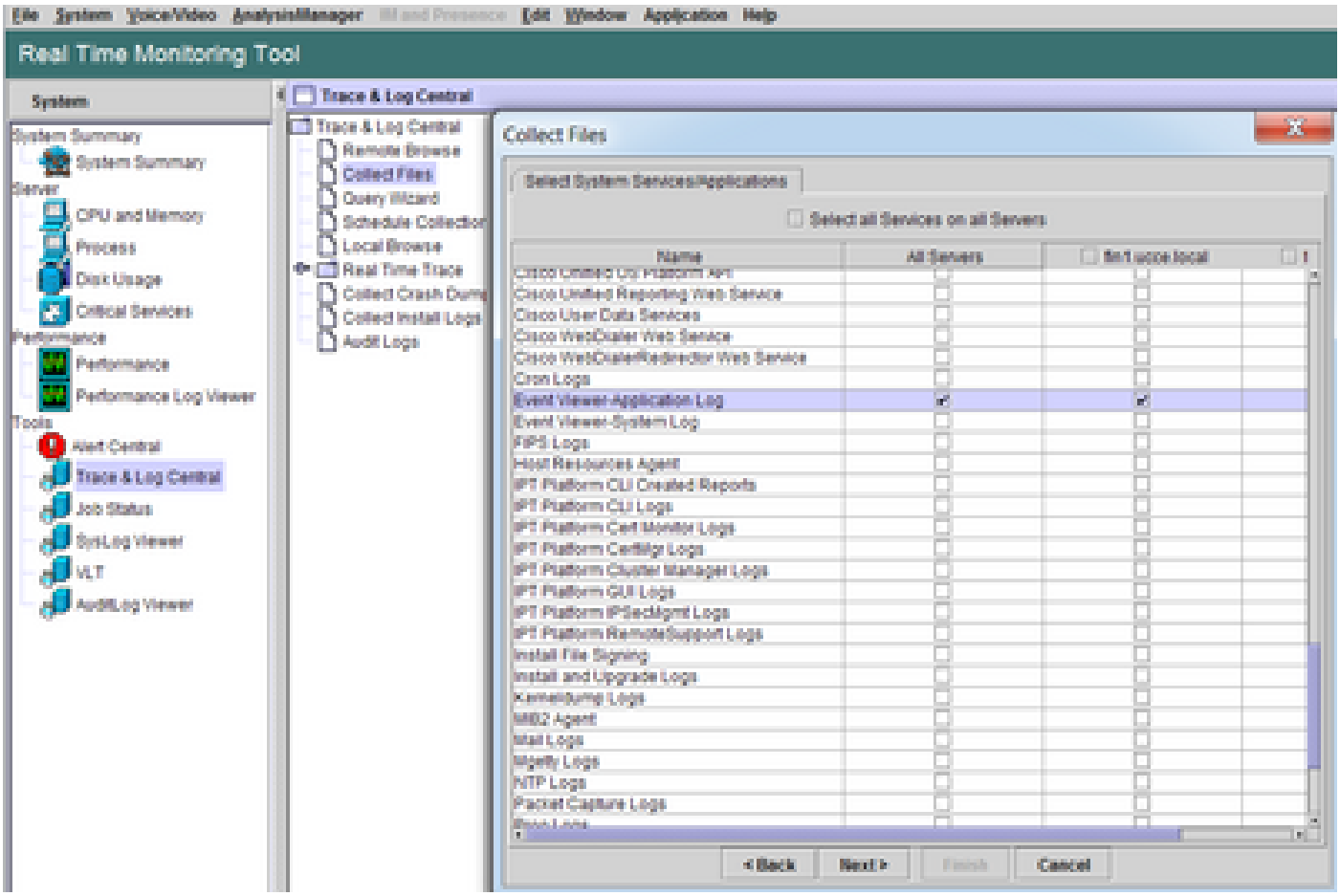
문제 - 모든 에이전트의 연결이 동시에 끊깁니다(서버 측 문제).

권장 작업

다음 문제를 확인합니다.

1. Cisco Unified Communications Manager CTIManager 서비스 연결 끊기 UCCX용 모든 CTIManager 제공자가 종료되거나 충돌할 경우 UCCX 에이전트에게 빨간색 배너 오류가 표시됩니다. UCCE 에이전트는 이 경우 빨간색 배너를 볼 수 없지만 통화를 에이전트에게 제대로 라우팅하지 못합니다.

- CTI 공급자로 사용되는 CUCM 서버에서 Cisco CTIManager 서비스가 시작되는지 확인합니다.
- Cisco CTIManager 서비스가 Event Viewer - Application logs(이벤트 뷰어 - 애플리케이션 로그)를 통해 crash했는지 확인하여 crash했는지 확인합니다.
    - RTMT에서 이벤트 뷰어 로그를 수집하려면 System > Tools > Trace and Log Central > Collect Files > Select System Services/Applications > Event Viewer-Application Log로 이동합니다.



- CLI에서 이벤트 뷰어-애플리케이션 로그를 수집하려면: file get activelog /syslog/CiscoSyslog* abstime hh:mm:MM/DD/YY hh:mm:MM/DD/YY
- CLI에서 코어 덤프를 보려면 utils core active list(유틸리티 코어 활성 목록)

참고: 코어 덤프 파일 이름은 core.<ProcessID>.<SignalNumber>.<ProcessName>.<EpochTime> 형식을 사용합니다.
예: core.24587.6.CTIManager.1533441238
따라서, 상기 충돌 시간은 상기 시대로부터 결정될 수 있다.

---

2. Finesse/UCCX Notification Service가 중지되거나 충돌했습니다.

- 이벤트 뷰어-애플리케이션 로그에서 알림 서비스 오류를 확인하거나 서비스가 중지되었는지 확인합니다
- 알림 서비스가 작동 중인지 확인: 유틸리티 서비스 목록
- 알림 서비스가 종료되는 시간 확인: 파일 검색 activelog /desktop/logs/openfire "Openfire 중지됨"
- 알림 서비스가 시작된 시간 확인: 파일 검색 activelog /desktop/logs/openfire "HTTP 바인딩 서비스 시작됨"
- 충돌로 인해 발생한 알림 서비스 메모리 덤프 확인: 파일 목록 activelog /desktop/logs/openfire/*.hprof
- 알림 서비스가 TCP 포트 7443에서 트래픽을 수신 대기하고 있는지 확인합니다. show open ports regexp 7443.*LISTEN
- 이러한 결함이 적용 가능한지 확인합니다(이러한 결함은 로그인하는 상담원의 로그인 실패를

야기하며 이미 로그인한 상담원의 경우 빨간색 배너 Finesse 연결 끊기 메시지가 표시됨).
    - Cisco 버그 ID CSCva72280 - 잘못된 XML 문자에 대한 Finesse Tomcat 및 Openfire 충돌
    - Cisco 버그 ID CSCva72325 - UCCX: 잘못된 XML 문자에 대한 Finesse Tomcat 및 Openfire 충돌

충돌이 의심되는 경우 Cisco Finesse Tomcat and Notification Service를 다시 시작합니다. 이 방법은 네트워크 다운된 경우에만 권장되며, 그렇지 않은 경우 Finesse 서버에서 에이전트 연결을 다시 시작합니다.

UCCE의 단계:

- 유틸리티 서비스 중지 Cisco Finesse Tomcat
- 유틸리티 서비스 중지 Cisco Finesse 알림 서비스
- utils service start Cisco Finesse Tomcat
- utils service start Cisco Finesse Notification Service

UCCX 단계:

- 유틸리티 서비스 중지 Cisco Finesse Tomcat
- 유틸리티 서비스 중지 Cisco Unified CCX 알림 서비스
- utils service start Cisco Finesse Tomcat
- 유틸리티 서비스 Cisco Unified CCX 알림 서비스 시작

Fiddler 사용

Fiddler를 구성하는 것은 필요한 단계를 이해하지 않고 Fiddler가 작동하는 방식을 이해하지 않고서는 다소 어려운 작업이 될 수 있습니다. Fiddler는 Finesse 클라이언트(웹 브라우저)와 Finesse 서버 사이에 있는 중간자(man-in-the-middle) 웹 프록시입니다. Finesse 클라이언트와 Finesse 서버 간의 연결이 보호되므로 보안 메시지를 보기 위해 Fiddler 컨피그레이션에 복잡성이 한층 더 심화됩니다.

일반적인 Fiddler 문제

Fiddler는 Finesse 클라이언트와 Finesse 서버 사이에 있으므로 Fiddler 애플리케이션은 인증서가 필요한 모든 Finesse TCP 포트에 대해 서명된 인증서를 생성해야 합니다.

Cisco Finesse Tomcat 서비스 인증서

1. Finesse 게시자 서버 TCP 8445(및/또는 UCCE용 443)
2. Finesse 가입자 서버 TCP 8445(및/또는 UCCE용 443)

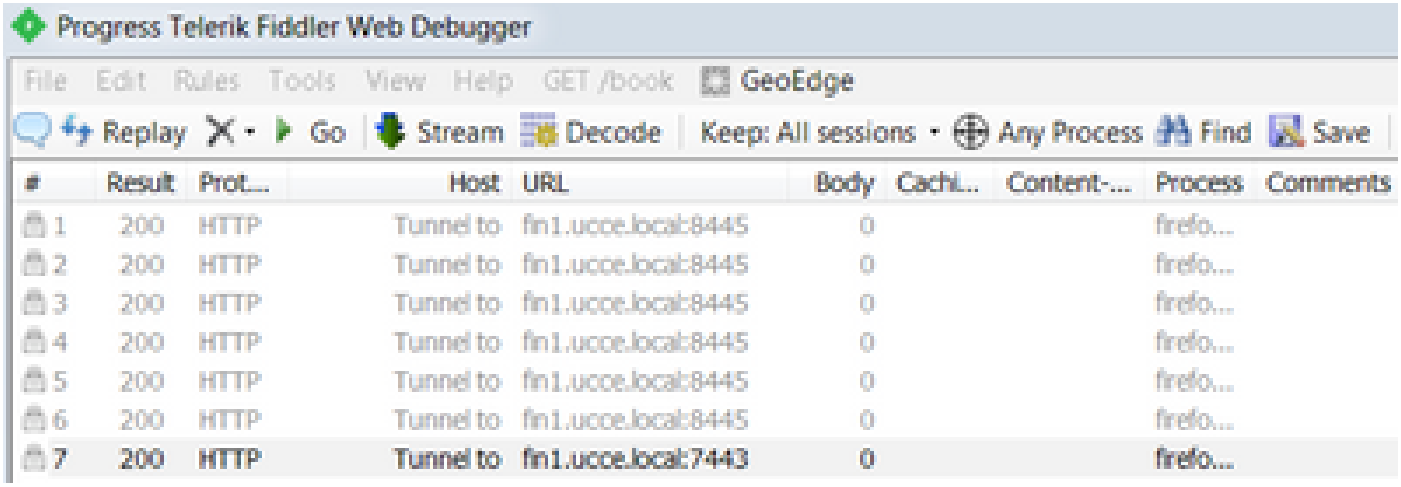
Cisco Finesse(Unified CCX) 알림 서비스 인증서

1. Finesse 게시자 서버 TCP 7443
2. Finesse 가입자 서버 TCP 7443

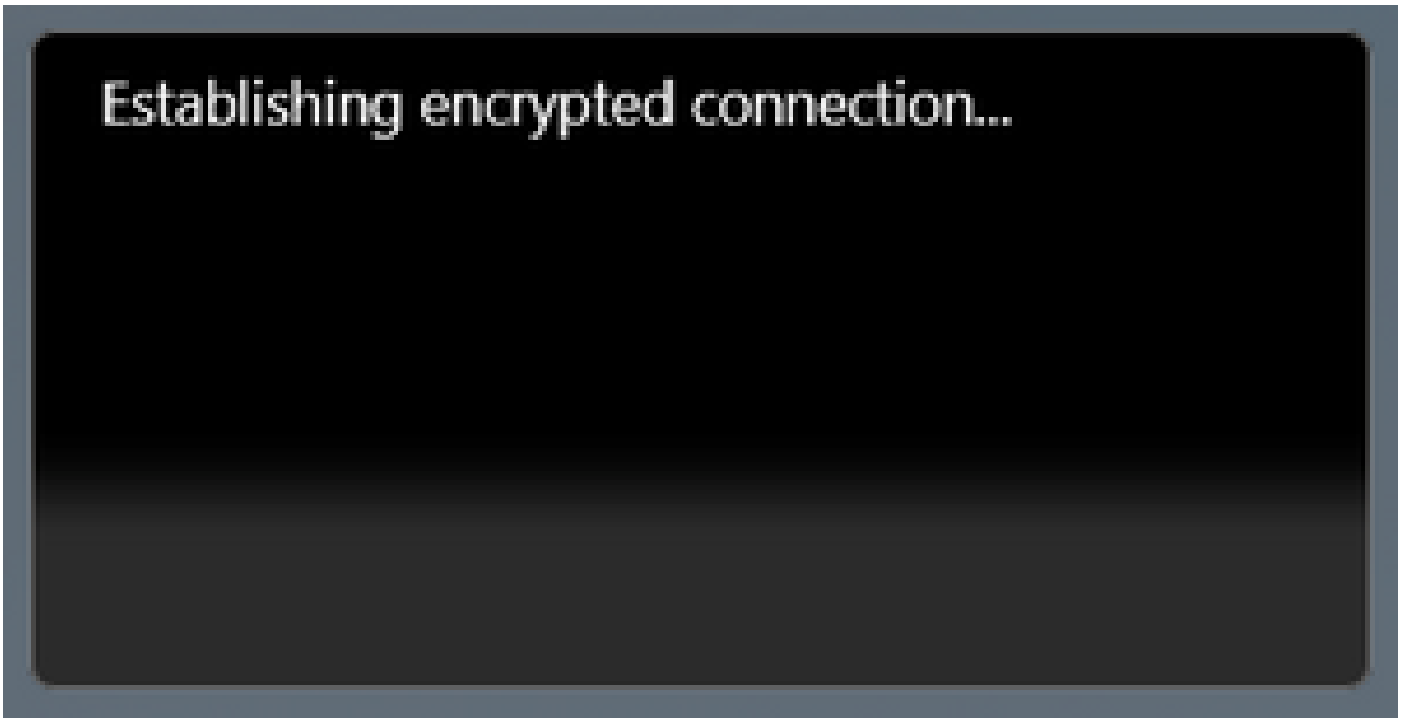
Finesse 서버를 대신하여 인증서를 동적으로 생성하려면 Fiddler에 대해 HTTPS 암호 해독을 활성화해야 합니다. 기본적으로 활성화되어 있지 않습니다.

HTTPS 암호 해독이 구성되지 않은 경우 알림 서비스에 대한 초기 터널 연결이 표시되지만 http 바인딩 트래픽은 표시되지 않습니다. Fiddler는 다음 항목만 표시합니다.

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



그러면 Fiddler가 서명한 Finesse 인증서를 클라이언트에서 신뢰해야 합니다. 이러한 인증서를 신뢰할 수 없는 경우 Finesse 로그인의 암호화된 연결 설정... 단계를 통과할 수 없습니다.



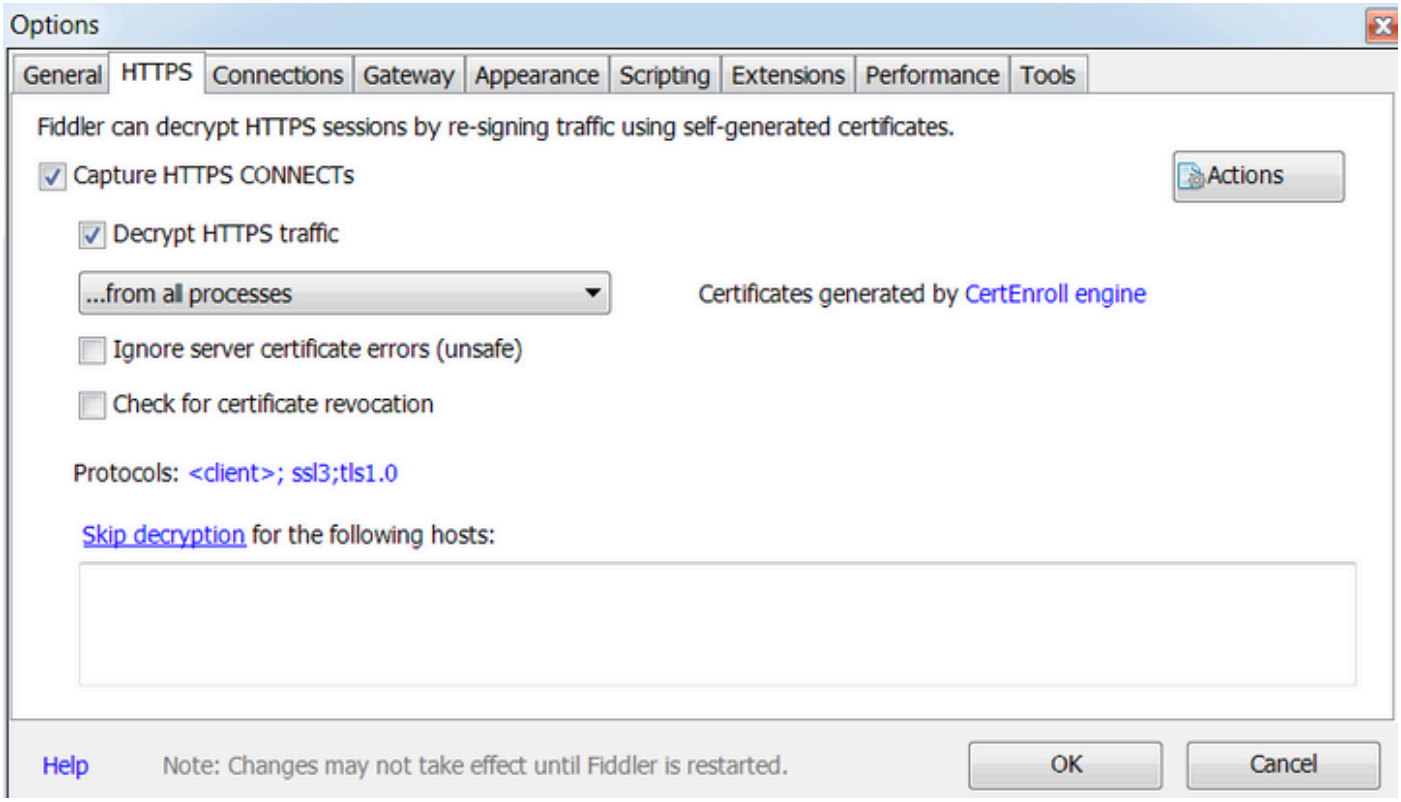
로그인의 인증서 예외를 수락하는 것이 작동하지 않는 경우도 있으며 인증서를 브라우저에서 수동으로 신뢰해야 합니다.

컨피그레이션 단계 예

⚠ 주의: 제공된 예제 컨피그레이션은 실습 환경에서 Windows 7 x64의 Fiddler v5.0.20182.28034 for .NET 4.5 and Mozilla Firefox 64.0.2(32비트)에 대해 제공됩니다. 이러한 절차는 Fiddler의 모든 버전, 모든 브라우저 또는 모든 컴퓨터 운영 체제로 일반화할 수 없습니다. 네트워크가 가동 중인 경우 모든 컨피그레이션의 잠재적 영향을 이해해야 합니다. 자세한 내용은 Fiddler 공식 문서를 참조하십시오.

1단계. Fiddler 다운로드

2단계. HTTPS 암호 해독을 활성화합니다. Tools(툴) > Options(옵션) > HTTPS로 이동하고 Decrypt HTTPS traffic(HTTPS 트래픽 해독) 확인란을 선택합니다.

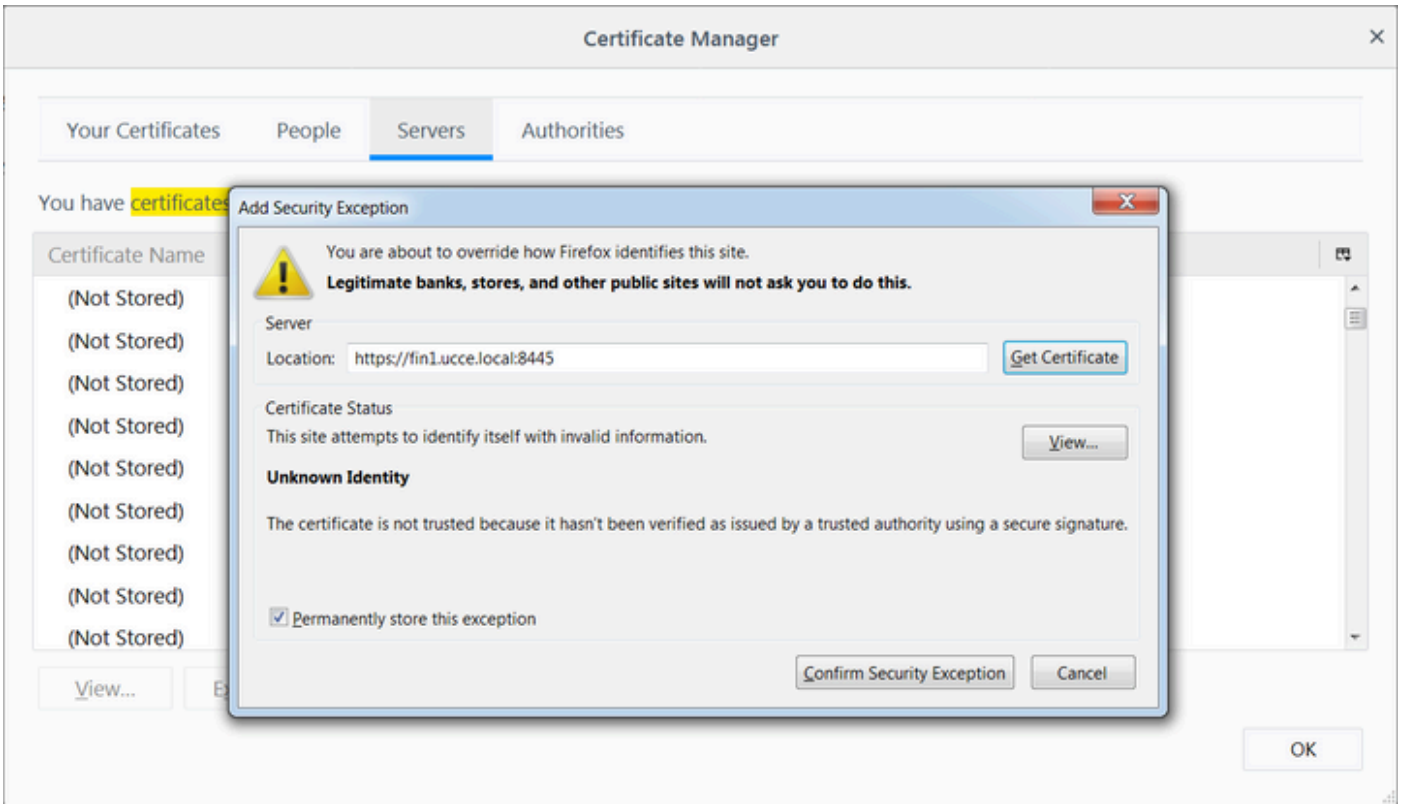


3단계. Fiddler 루트 인증서를 신뢰할지 묻는 경고 메시지 상자가 열립니다. Yes(예)를 선택합니다.

4단계. 경고 메시지 상자가 열리고 "DO_NOT_TRUST_FiddlerRoot...를 나타내는 CA(인증 기관)의 인증서를 설치하려고 합니다. 이 인증서를 설치하시겠습니까?" Yes(예)를 선택합니다.

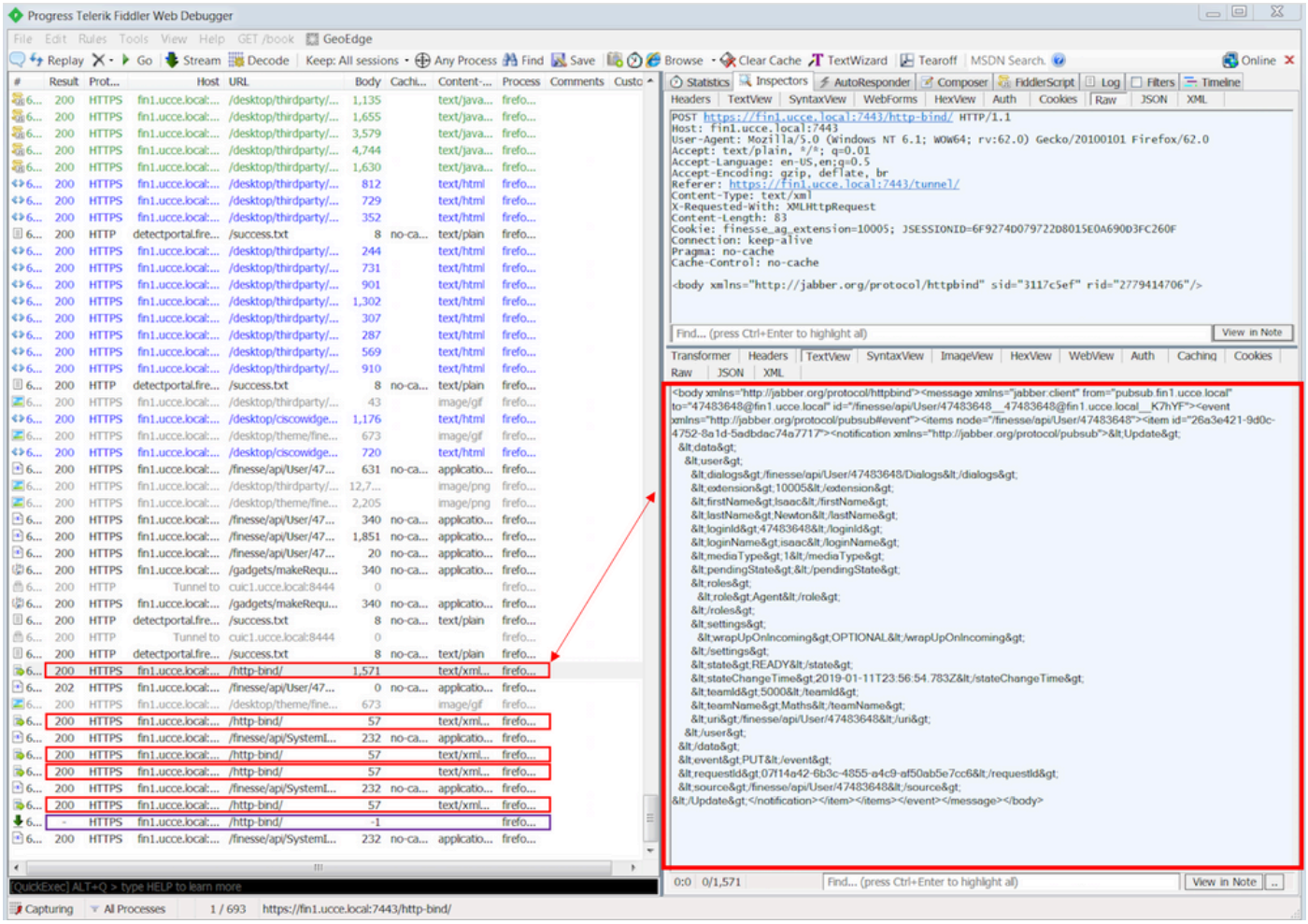
5단계. 컴퓨터 또는 브라우저 인증서 신뢰 저장소에 Finesse 게시자 및 가입자 인증서를 수동으로 추가합니다. 포트 8445, 7443 및 (UCCE만 해당) 443을 확인합니다. 예를 들어 Firefox에서는 Finesse Operating System Administration 페이지에서 인증서를 다운로드하지 않고 간단하게 이 작업을 수행할 수 있습니다.

Options(옵션) > Find in Options (search) > Certificates(인증서) > Servers(서버) > Add Exception(예외 추가) > Location(위치) > 두 Finesse 서버에 대한 관련 포트에 https://<Finesse server>:port를 입력합니다.

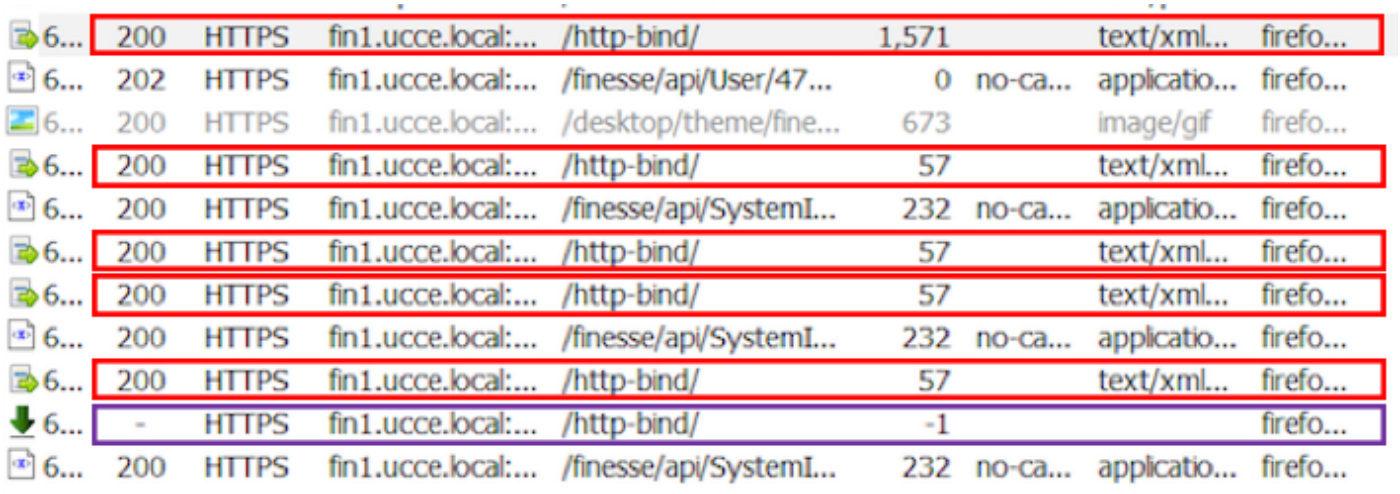


6단계. Finesse에 로그인하고 http 바인딩 메시지가 Finesse 클라이언트에서 Fiddler를 통해 Finesse 서버로 나가는지 확인합니다.

제공된 예에서 처음 5개 메시지는 Finesse 서버에서 응답한 http 바인딩 메시지를 표시합니다. 첫 번째 메시지는 메시지 본문에 반환된 1571바이트의 데이터를 포함합니다. 본문에는 에이전트 이벤트와 관련된 XMPP 업데이트가 포함되어 있습니다. 최종 http-bind 메시지는 Finesse 클라이언트에서 보냈지만 Finesse 서버에서 응답을 받지 못했습니다. 이는 HTTP 결과가 null(-)이고 응답 본문의 바이트 수가 null(-1)인 경우 확인할 수 있습니다.



더 면밀한 데이터 보기:



XMPP 메시지에 대한 응답 본문:

<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>

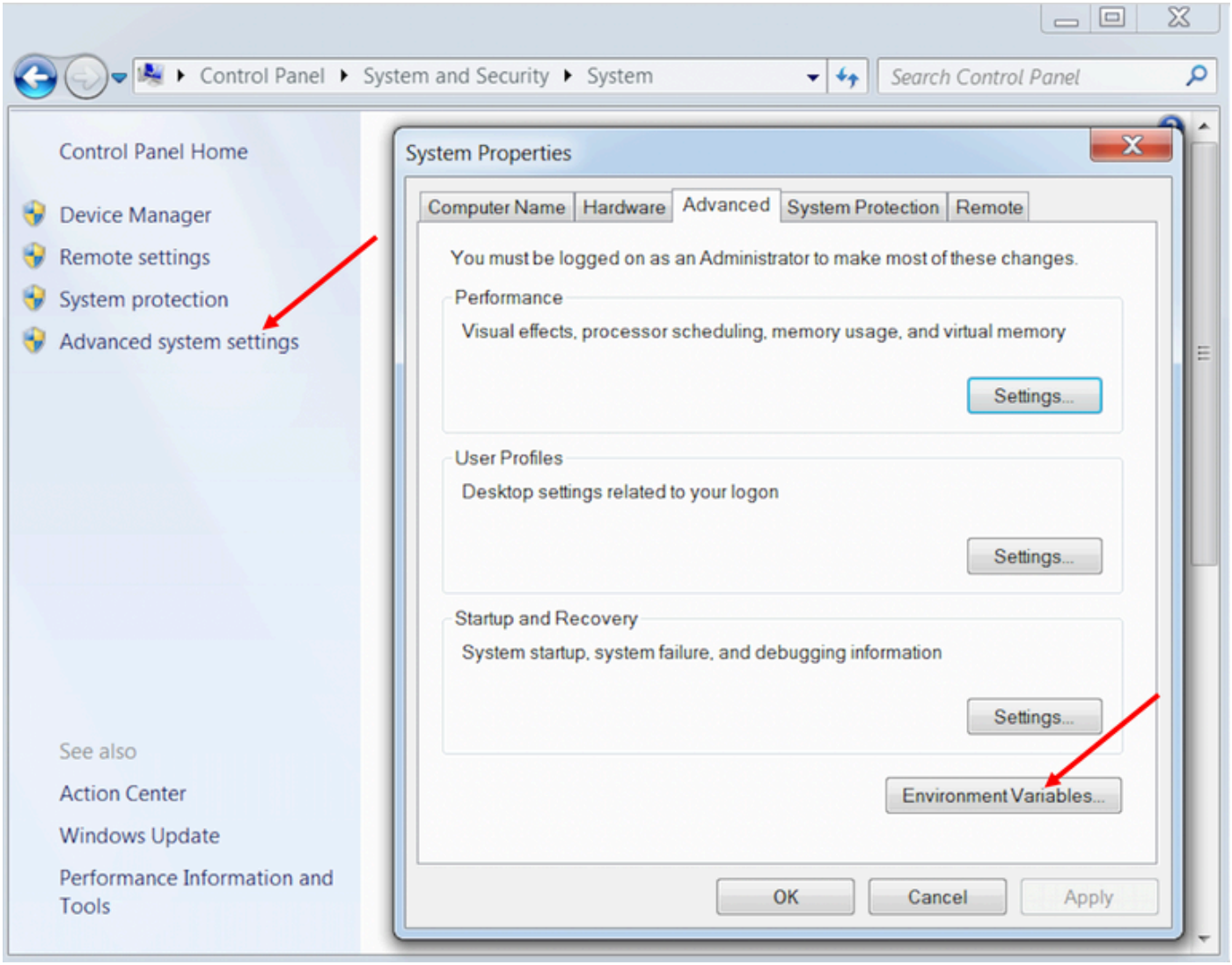
Wireshark 사용

Wireshark는 HTTPS 트래픽을 스니핑하고 디코딩하는 데 사용할 수 있는 일반적으로 사용되는 패킷 스니핑 툴입니다. HTTPS 트래픽은 TLS(Transport Layer Security)를 통해 보호되는 HTTP 트래픽입니다. TLS는 두 호스트 간의 무결성, 인증 및 기밀성을 제공합니다. 웹 애플리케이션에서 일반적으로 사용되지만 TCP를 전송 계층 프로토콜로 사용하는 모든 프로토콜과 함께 사용할 수 있습니다. SSL(Secure Sockets Layer)은 TLS 프로토콜의 이전 버전으로, 안전하지 않으므로 더 이상 사용되지 않습니다. 이러한 이름은 자주 혼용되며 SSL 또는 TLS 트래픽에 사용되는 Wireshark 필터는 ssl입니다.

⚠ 주의: 제공된 예제 컨피그레이션은 랩 환경에서 Windows7 x64의 Wireshark 2.6.6(v2.6.6-0-gdf942cd8) 및 Mozilla Firefox 64.0.2(32비트)에 대해 제공됩니다. 이러한 절차는 Fiddler의 모든 버전, 모든 브라우저 또는 모든 컴퓨터 운영 체제로 일반화할 수 없습니다. 네트워크가 가동 중인 경우 모든 컨피그레이션의 잠재적 영향을 이해해야 합니다. 자세한 내용은 공식 Wireshark SSL 문서를 참조하십시오. Wireshark 1.6 이상이 필요합니다.

✎ 참고: 이 방법은 Firefox 및 Chrome에서만 사용할 수 있습니다. 이 메서드는 Microsoft Edge에서 작동하지 않습니다.

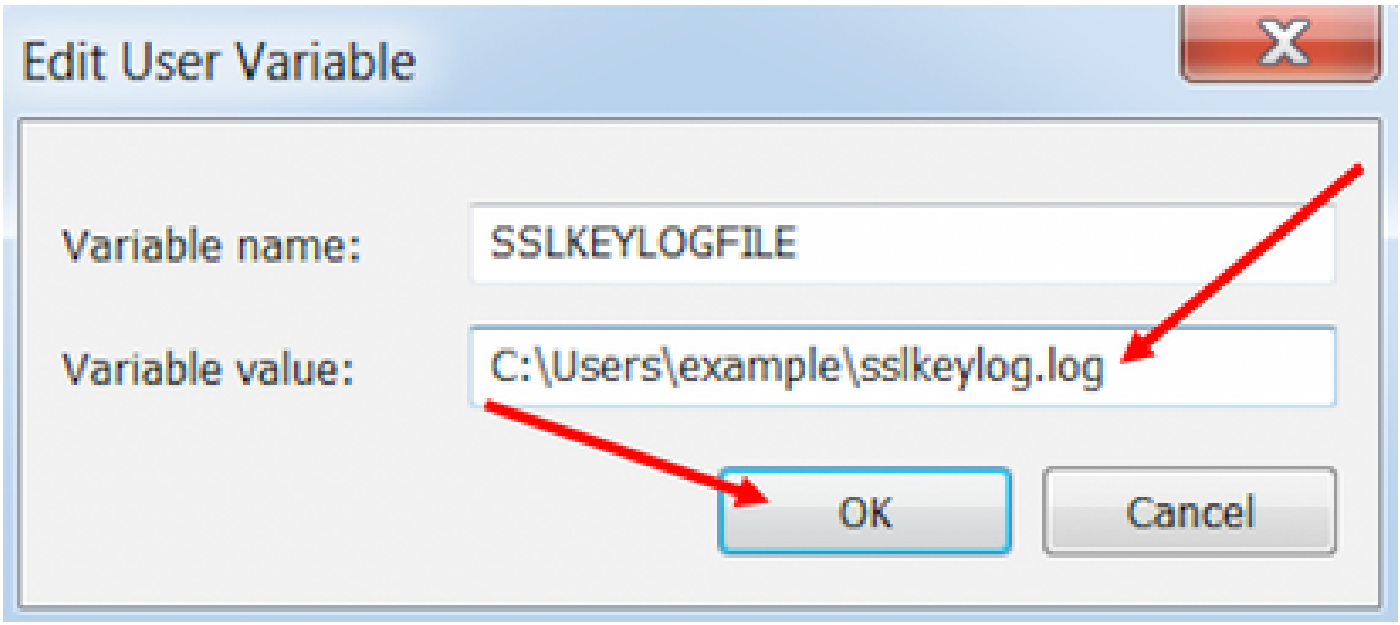
1단계. 상담원의 Windows PC에서 제어판 > 시스템 및 보안 > 시스템 > 고급 시스템 설정 환경 변수로 이동합니다.



2단계. 사용자 <username> > New...에 대한 User variables(사용자 변수)로 이동합니다.

SSLKEYLOGFILE이라는 변수를 만듭니다.

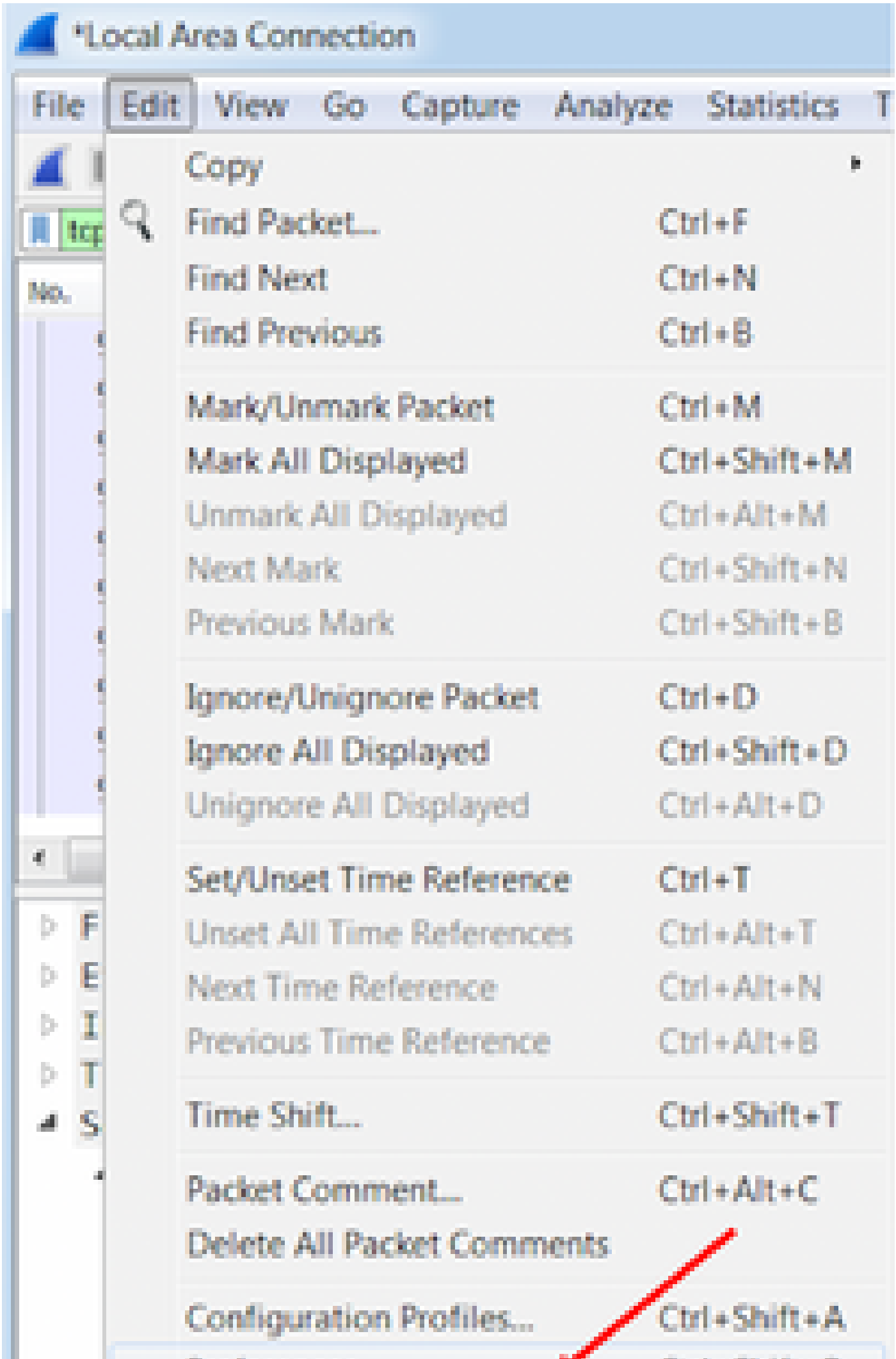
SSL 프리마스터 암호를 사설 디렉터리에 저장할 파일을 만듭니다. SSL keylogfile= </path/to/private/directory/with/logfile>



참고: 사용자 변수 대신 시스템 변수를 만들거나 파일을 비공개 디렉터리에 저장하면 시스템의 모든 사용자가 프리마스터 암호에 액세스할 수 있으므로 보안성이 떨어집니다.

3단계. Firefox 또는 Chrome이 열려 있으면 애플리케이션을 닫습니다. 다시 연 후에는 SSLKEYLOGFILE에 쓰기를 시작할 수 있습니다.

4단계. Wireshark에서 Edit(편집) > Preferences...(기본 설정...)로 이동합니다.


*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

이 번역에 관하여

Cisco는 전 세계 사용자에게 다양한 언어로 지원 콘텐츠를 제공하기 위해 기계 번역 기술과 수작업 번역을 병행하여 이 문서를 번역했습니다. 아무리 품질이 높은 기계 번역이라도 전문 번역가의 번역 결과물만큼 정확하지는 않습니다. Cisco Systems, Inc.는 이 같은 번역에 대해 어떠한 책임도 지지 않으며 항상 원본 영문 문서(링크 제공됨)를 참조할 것을 권장합니다.