

Comprendere e risolvere i problemi relativi all'implementazione di Finesse BOSH

Sommario

[Introduzione](#)

[Prerequisiti](#)

[Requisiti](#)

[Componenti usati](#)

[Premesse](#)

[Informazioni sull'implementazione di Finesse BOSH](#)

[Informazioni su XMPP](#)

[Esempio di messaggio XMPP](#)

[Implementazione XMPP con Finesse](#)

[Esempio di richiesta/risposta Finesse XMPP](#)

[Comprendere i messaggi e i nodi XMPP Finesse](#)

[Esempio 1: utilizzare Pidgin per visualizzare i nodi Finesse XMPP](#)

[Esempio 2: utilizzare la scheda di rete Strumenti di sviluppo browser per visualizzare i messaggi HTTP](#)

[Risoluzione dei problemi relativi al messaggio di errore di disconnessione di BOSH](#)

[Analisi log](#)

[Registri del servizio di notifica di debug](#)

[Registri servizio di notifica informazioni](#)

[Registri servizi Web](#)

[Motivi comuni della disconnessione di BOSH](#)

[Problema - Gli agenti si disconnettono in momenti diversi \(problema sul lato client\)](#)

[Azioni consigliate](#)

[Problema - Tutti gli agenti si disconnettono contemporaneamente \(problema sul lato server\)](#)

[Azioni consigliate](#)

[Usa filtro](#)

[Problema comune del richiedente](#)

[Procedura di configurazione di esempio](#)

[Utilizzare Wireshark](#)

[Difetti correlati](#)

[Informazioni correlate](#)

Introduzione

Questo documento descrive l'architettura alla base delle connessioni Finesse che utilizzano BOSH e come è possibile diagnosticare i problemi di connessione BOSH.

Prerequisiti

Requisiti

Cisco raccomanda la conoscenza dei seguenti argomenti:

- Cisco Finesse

- Unified Contact Center Enterprise (UCCE)
- Unified Contact Center Express (UCCX)
- Strumenti di sviluppo per browser Web
- Amministrazione di Windows e/o Mac

Componenti usati

Le informazioni fornite in questo documento si basano sulle seguenti versioni software e hardware:

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

Le informazioni discusse in questo documento fanno riferimento a dispositivi usati in uno specifico ambiente di emulazione. Su tutti i dispositivi menzionati nel documento la configurazione è stata ripristinata ai valori predefiniti. Se la rete è operativa, valutare attentamente eventuali conseguenze derivanti dall'uso dei comandi.

Premesse

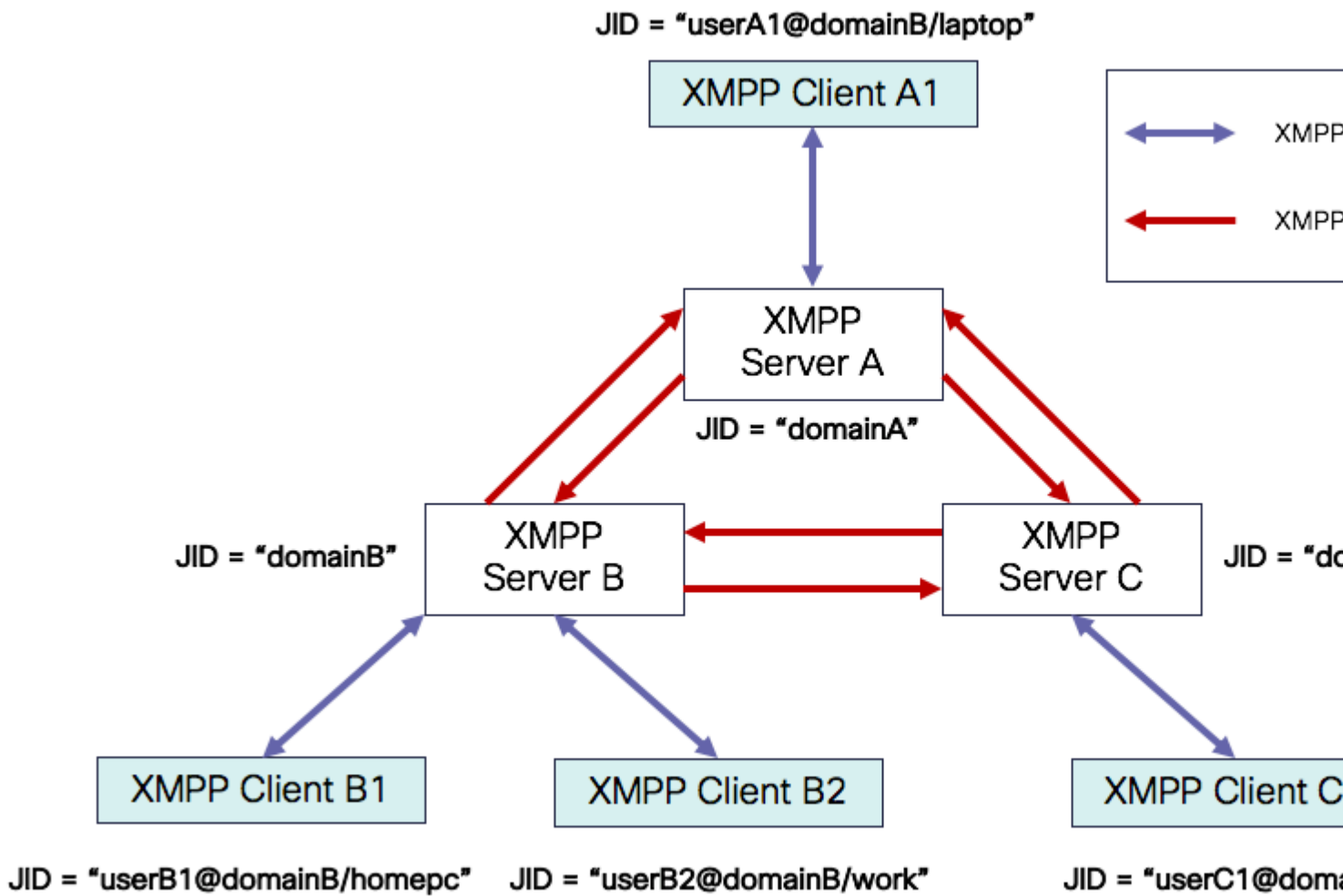
Le connessioni che utilizzano flussi bidirezionali su HTTP sincrono sono denominate BOSH.

Informazioni sull'implementazione di Finesse BOSH

Informazioni su XMPP

Il protocollo XMPP (Extensible Messaging and Presence Protocol), noto anche come Jabber, è un protocollo con conservazione dello stato in un modello client-server. XMPP consente la consegna rapida di piccole parti di dati XML (Extensible Markup Language) strutturati da un'entità all'altra. XMPP/Jabber è ampiamente utilizzato nelle applicazioni di messaggistica istantanea (IM) e di presenza.

Tutte le entità XMPP sono identificate dal relativo ID Jabber (JID).



Schema di indirizzamento JID: user@domain/resource

utente	nome utente client sul server XMPP o nome della sala riunioni
dominio	Nome di dominio completo (FQDN) server XMPP
risorsa	identificatore dell'entità/endpoint specifico dell'utente (ad esempio, laptop, smartphone e così via), identificatore di sessione o nome del nodo pubsub

Nota: i tre componenti JID non vengono utilizzati in tutti i casi. In genere, un server viene definito semplicemente dal dominio, una sala conferenze definita da user@domain e un client da user@domain/resource.

I messaggi XMPP sono chiamati stanze. In XMPP sono disponibili tre stanze principali:

1. <message>: una direzione, un destinatario
2. <presence>: una direzione, pubblica in molte
3. <iq>: info/query - richiesta/risposta

Tutte le stanze hanno gli indirizzi di destinazione e di origine e la maggior parte delle stanze ha anche gli attributi type, id e xml:langattribute.

Attributo stanza	Scopo
a	JID destinazione
da	JID di origine
tipo	scopo del messaggio
ID	identificatore univoco utilizzato per collegare una richiesta a una risposta per le stanze <iq>
xml:lang	definisce la lingua predefinita per qualsiasi file XML leggibile nella stanza

Esempio di messaggio XMPP

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

Implementazione XMPP con Finesse

Se un'applicazione Web deve funzionare con XMPP, possono verificarsi numerosi problemi. Poiché i browser non supportano XMPP su TCP (Transmission Control Protocol) in modo nativo, tutto il traffico XMPP deve essere gestito da un programma in esecuzione all'interno del browser. I server Web e i browser comunicano tramite messaggi HTTP (HyperText Transfer Protocol), pertanto Finesse e altre applicazioni Web inseriscono i messaggi XMPP all'interno dei messaggi HTTP.

La prima difficoltà con questo approccio è che HTTP è un protocollo senza stato. Ciò significa che ogni richiesta HTTP non è correlata ad altre richieste. Tuttavia, questo problema può essere affrontato con mezzi applicativi, ad esempio tramite l'utilizzo di cookie/dati di post.

La seconda difficoltà è rappresentata dal comportamento unidirezionale di HTTP. Solo il client invia le richieste e solo il server può rispondere. L'impossibilità del server di eseguire il push dei dati rende innaturale l'implementazione di XMPP su HTTP.

Questo problema non esiste nella specifica di base XMPP originale (RFC 6120), dove XMPP è associato a TCP. Tuttavia, se si desidera risolvere il problema con XMPP associato a HTTP, ad esempio, poiché

Javascript può inviare richieste HTTP, esistono due possibili soluzioni. Entrambi richiedono un bridge tra HTTP e XMPP.

Le soluzioni proposte sono:

1. Polling (protocollo legacy): richieste HTTP ripetute che richiedono nuovi dati definiti in XEP-0025: polling HTTP Jabber

2. Il polling lungo è anche noto come BOSH: protocollo di trasporto che emula la semantica di una connessione TCP bidirezionale di lunga durata tra due entità utilizzando in modo efficiente più coppie di richiesta/risposta HTTP sincrone senza richiedere l'utilizzo di polling frequenti definito in XEP-0124: Binding HTTP ed esteso da XEP-0206: XMPP Over BOSH

Finesse implementa BOSH in quanto è abbastanza efficiente dal punto di vista del carico del server e del traffico. Il motivo per utilizzare BOSH è quello di coprire il fatto che il server non deve rispondere non appena c'è una richiesta. La risposta viene posticipata fino a un tempo specificato finché il server non dispone dei dati per il client, quindi viene inviata come risposta. Non appena il client riceve la risposta, effettua una nuova richiesta e così via.

Il client desktop Finesse (applicazione Web) stabilisce una connessione BOSH non aggiornata sulla porta TCP 7443 ogni 30 secondi. Dopo 30 secondi, se non sono disponibili aggiornamenti dal servizio di notifica Finesse, il servizio di notifica invia una risposta HTTP con 200 OK e un corpo di risposta (quasi) vuoto. Se il Servizio di notifica dispone di un aggiornamento sulla presenza di un agente o di un evento di conversazione (chiamata), ad esempio, i dati vengono inviati immediatamente al client Web Finesse.

Esempio di richiesta/risposta Finesse XMPP

In questo esempio viene illustrata la prima risposta alla richiesta di messaggio XMPP condivisa tra il client Finesse e il server Finesse per impostare la connessione BOSH.

Finesse client request:

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

Finesse server response:

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

Per riepilogare:

1. Il client Web Finesse dispone di una connessione HTTP non aggiornata (http-bind) impostata sul server Finesse tramite la porta TCP 7443. Questo è noto come un sondaggio di BOSH.
2. Finesse Notification Service è un servizio di presenza che pubblica aggiornamenti relativi allo stato di un agente, di una chiamata e così via.
3. Se il servizio di notifica dispone di un aggiornamento, risponde alla richiesta http-bind con l'aggiornamento dello stato come messaggio XMPP nel corpo della risposta HTTP.
4. Se non sono disponibili aggiornamenti dello stato 30 secondi dopo la ricezione della richiesta http-bind, il servizio di notifica risponde senza aggiornamenti dello stato per consentire al client Web Finesse di inviare un'altra richiesta http-bind. In questo modo il servizio di notifica può sapere che il client Web Finesse è ancora in grado di connettersi al servizio di notifica e che l'agente non ha chiuso il browser o messo il computer in sospensione e così via.

Comprendere i messaggi e i nodi XMPP Finesse

Finesse implementa anche la specifica XMPP XEP-0060: Publish-Subscribe. Lo scopo di questa specifica è consentire al server XMPP (Servizio di notifica) di ottenere informazioni pubblicate sui nodi XMPP (argomenti) e quindi di inviare eventi XMPP alle entità sottoscritte dal nodo. Nel caso di Finesse, il server CTI (Computer Telephony Integration) invia messaggi CTI al servizio Web Finesse per comunicare a Finesse gli aggiornamenti della configurazione, quali, ma non solo, la creazione dell'agente o della coda CSQ (Contact Service Queue) o le informazioni su una chiamata. Queste informazioni vengono quindi convertite in un messaggio XMPP che il servizio Web Finesse pubblica nel servizio di notifica Finesse. Il servizio di notifica Finesse invia quindi messaggi XMPP su BOSH agli agenti che hanno sottoscritto determinati nodi XMPP.

Alcuni degli oggetti API Finesse definiti nel [manuale Finesse Web Services Developer Guide](#) sono nodi XMPP. I client Web Finesse di agenti e supervisor possono sottoscrivere aggiornamenti di eventi per alcuni di questi nodi XMPP in modo da avere informazioni aggiornate sugli eventi in tempo reale (come gli eventi chiamata, gli eventi stato e così via). Questa tabella mostra i nodi XMPP abilitati per pubsub.

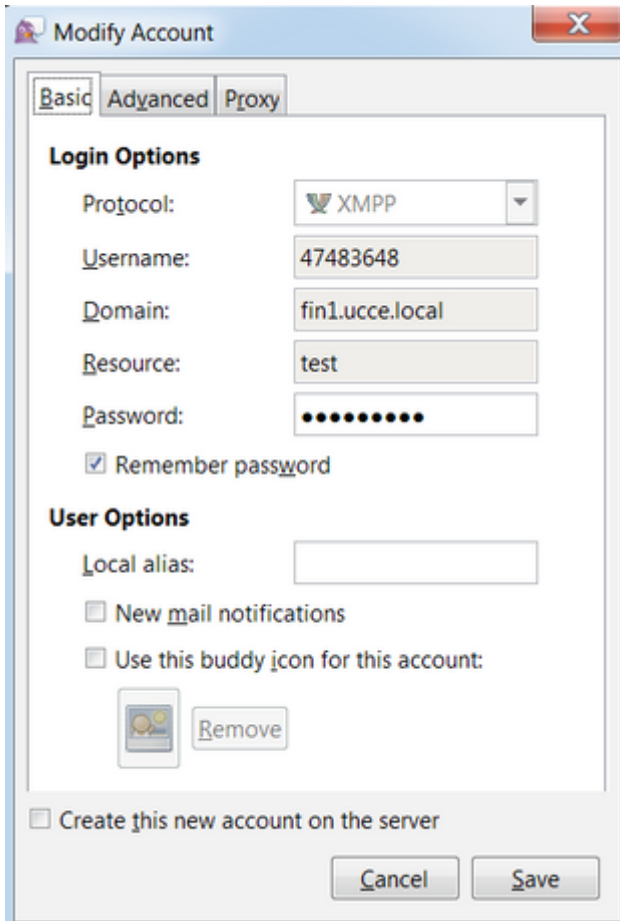
Oggetto API Finesse	Scopo	Abbonamento
/finesse/api/User/<IDLogo>	Mostra la mappatura dello stato e del team dell'agente	Agenti e supervisor
/finesse/api/User/<IDlogin>/Dialogs	Mostra le chiamate gestite dall'agente	Agenti e supervisor
/finesse/api/User/<IDlogin>/RegistroClient	Consente di acquisire i log dei client tramite il pulsante Invia segnalazione errori	Agenti e supervisor
/finesse/api/User/<IDlogin>/Queue/<IDcoda>	Mostra i dati delle statistiche di coda (se abilitati)	Agenti e supervisor
/finesse/api/Team/<IDTEAM>/Users	Mostra gli agenti che appartengono a un determinato team, incluse le informazioni sullo stato	Supervisor
/finesse/api/SystemInfo	Mostra lo stato del server Finesse. Utilizzato per determinare se il failover è necessario	Agenti e supervisor

Esempio 1: utilizzare Pidgin per visualizzare i nodi Finesse XMPP

Passaggio 1. Scaricare e installare il ping del client XMPP.

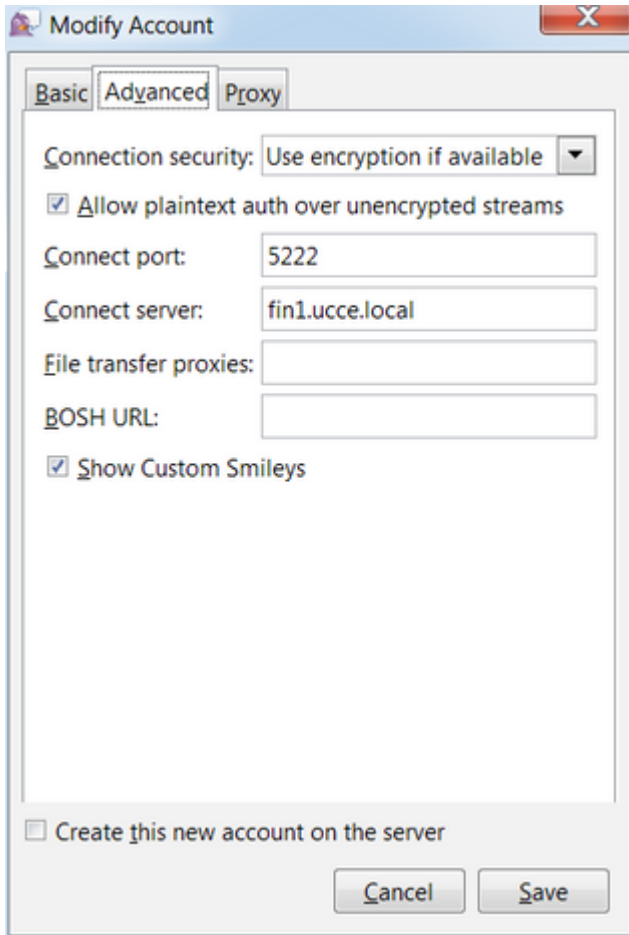
Passaggio 2. Passare a **Account > Modifica > Base** e configurare le **opzioni di accesso**:

- Protocollo: XMPP
- Nome utente: LoginID per qualsiasi agente
- Dominio: FQDN del server Finesse
- Risorsa: segnaposto - è possibile utilizzare qualsiasi valore, ad esempio test
- Password: password agente
- Selezionare la casella di controllo **Memorizza password**



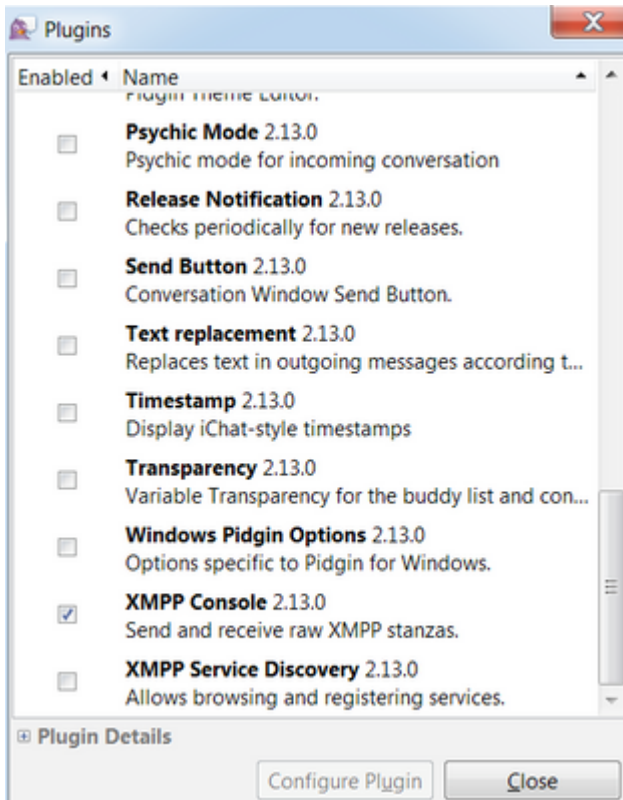
Passaggio 3. Passare a **Conti > Modifica > Avanzate** e configurare:

- Sicurezza connessione: usa crittografia se disponibile
- Selezionare la casella di controllo **Consenti autenticazione in testo normale per altri flussi non crittografati**.
- Porta di connessione: 522. Utilizzare la porta predefinita 5222. Questa porta è necessaria per i client XMPP esterni. I client desktop Finesse utilizzano 7443. Non utilizzare la porta 7443.
- Server di connessione: FQDN server Finesse

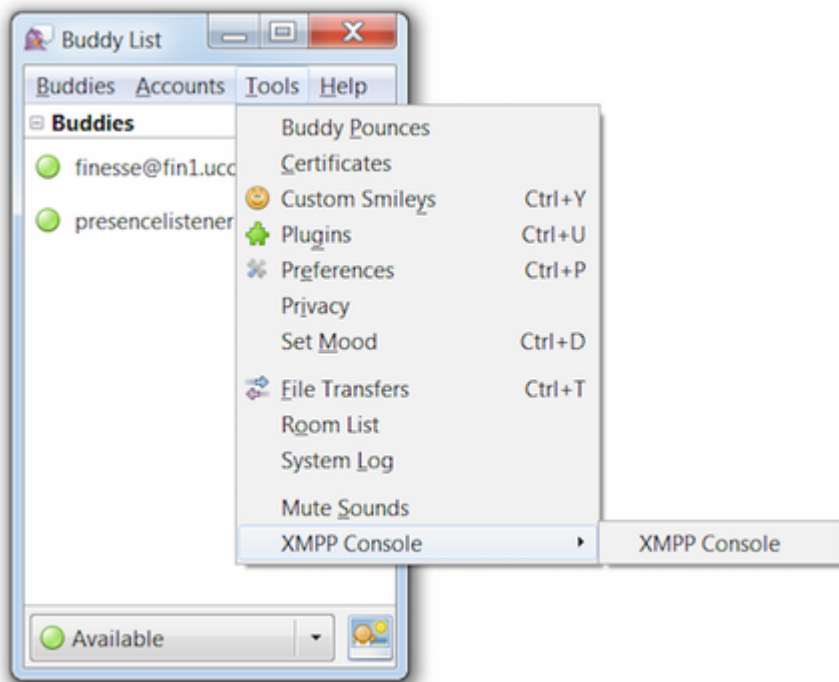


Nota: la porta 5222 viene utilizzata solo perché i client Web Finesse possono utilizzare la porta 7443 per connettersi al servizio di notifica.

Passaggio 4. Selezionare **Strumenti > Plugin** e abilitare la console XMPP.

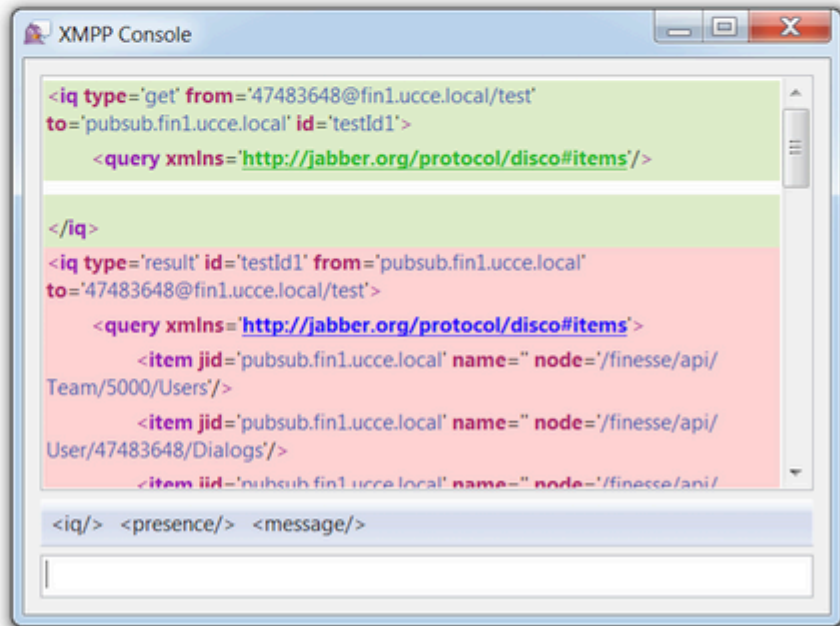
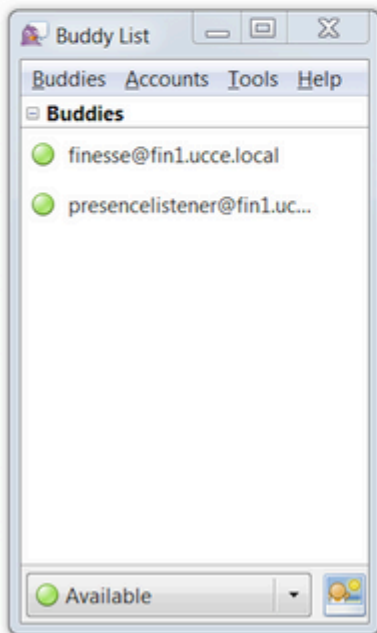
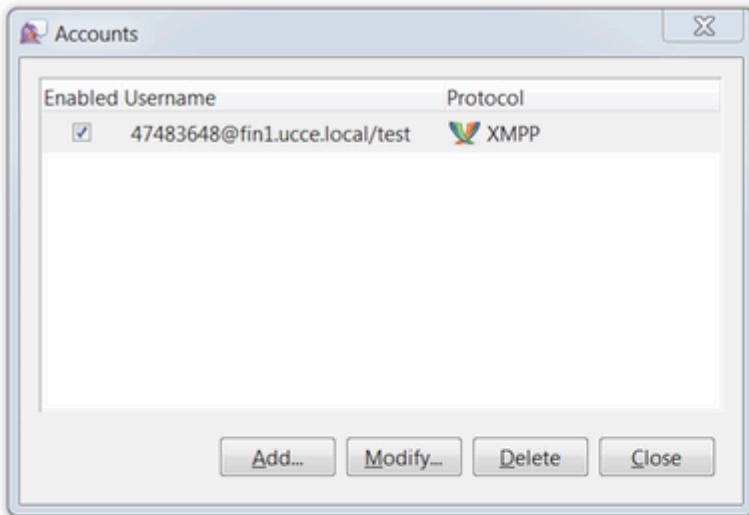


Passaggio 5. Selezionare **Strumenti** > **Console XMPP** > **Console XMPP** per aprire la console XMPP.



Passaggio 6. Eseguire questo messaggio <iq> per visualizzare tutti i nodi XMPP esistenti.

Ad esempio:



In un ambiente lab con due agenti e due CSQ configurati, questo output è contenuto nella risposta Finesse:

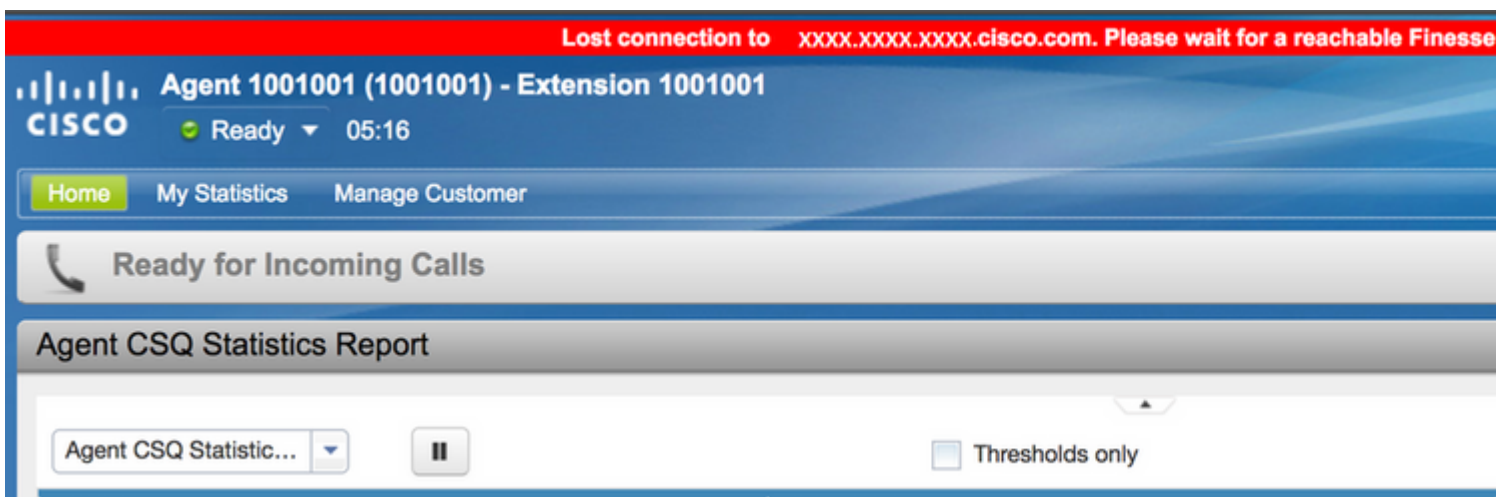
Esempio 2: utilizzare la scheda di rete Strumenti di sviluppo browser per visualizzare i messaggi HTTP

Ogni browser dispone di un insieme di strumenti di sviluppo. Nella scheda Rete degli strumenti di sviluppo vengono visualizzati i messaggi HTTP inviati e ricevuti dal client Web Finesse (browser). Ad esempio, questa immagine mostra come il client Web Finesse invia una richiesta SystemInfo che controlla ogni minuto lo stato di Finesse Tomcat come controllo di failover. Vengono inoltre visualizzati i messaggi http-bind della connessione BOSH. Il server Finesse invia una risposta entro 30 secondi se non sono presenti aggiornamenti da pubblicare sui nodi XMPP a cui è sottoscritto il client Web.

Status	Method	File	Domain	Cause	Type	Transfer...	Size	0 ms
200	POST	/http-bind/	XX.XX.XX.XX:7443	xhr	xml	57 B	57 B	
200	GET	Systeminfo?timestamponly&nocache=149218580998	XX.XX.XX.XX:8445	xhr	xml	166 B	166 B	
200	POST	/http-bind/	XX.XX.XX.XX:7443	xhr	xml	57 B	57 B	
200	POST	/http-bind/	XX.XX.XX.XX:7443	xhr	xml	57 B	57 B	
200	GET	Systeminfo?timestamponly&nocache=1492185741004	XX.XX.XX.XX:8445	xhr	xml	166 B	166 B	
200	POST	/http-bind/	XX.XX.XX.XX:7443	xhr	xml	57 B	57 B	
200	POST	/http-bind/	XX.XX.XX.XX:7443	xhr	xml	57 B	57 B	
200	GET	Systeminfo?timestamponly&nocache=1492185801004	XX.XX.XX.XX:8445	xhr	xml	166 B	166 B	
200	POST	/http-bind/	XX.XX.XX.XX:7443	xhr	xml	57 B	57 B	
200	POST	/http-bind/	XX.XX.XX.XX:7443	xhr	xml	57 B	57 B	
200	GET	Systeminfo?timestamponly&nocache=1492185861006	XX.XX.XX.XX:8445	xhr	xml	166 B	166 B	
200	POST	/http-bind/	XX.XX.XX.XX:7443	xhr	xml	57 B	57 B	
200	POST	/http-bind/	XX.XX.XX.XX:7443	xhr	xml	57 B	57 B	

Risoluzione dei problemi relativi al messaggio di errore di disconnessione di BOSH

Quando si verifica una disconnessione BOSH, l'errore ha perso la connessione a {FQDN server Finesse}. Attendere che venga individuato un Finesse Server raggiungibile... viene visualizzato in un banner rosso nella parte superiore del desktop Finesse.



Questo messaggio viene visualizzato perché al momento non è possibile ricevere eventi di sottoscrizione XMPP dal servizio di notifica Cisco Finesse. Pertanto, le informazioni sullo stato e i dettagli delle chiamate non possono essere visualizzati sul desktop dell'agente.

Per UCCX, 60 secondi dopo la disconnessione del browser, l'agente viene messo in stato di disconnessione. L'agente può trovarsi nello stato Pronto o Non pronto per consentire la disconnessione.

Per UCCE, Finesse impiega fino a 120 secondi per rilevare quando un agente chiude il browser o il browser si blocca e Finesse attende 60 secondi prima di inviare una richiesta di disconnessione forzata al server CTI, che determina il server CTI a mettere l'agente in uno stato Non pronto. In queste condizioni, Finesse può impiegare fino a 180 secondi per disconnettere l'agente. A differenza di UCCX, l'agente passa allo stato Non pronto anziché allo stato Disconnessione.

Nota: la disconnessione CTI non è pronta rispetto a Il comportamento dello stato di disconnessione in UCCE è controllato dal parametro PG /LOAD. In base alle Note di rilascio per Unified Contact

Center Enterprise & Hosted release 10.0(1), il parametro /LOAD è deprecato a partire da UCCE 10.0.

Per ulteriori informazioni sul comportamento di UCCE Finesse Desktop, fare riferimento alla sezione Desktop Behavior del capitolo Cisco Finesse Failover Mechanism nel manuale [Cisco Finesse Administration Guide](#).

Nota: i valori del timer possono cambiare in futuro in base al fabbisogno del prodotto.

Analisi log

I registri del servizio di notifica Finesse e UCCX possono essere raccolti tramite RTMT o tramite la CLI:

file get activelog /desktop recurs compress

Registri del servizio di notifica di debug

Nota: impostare i log del livello di debug solo durante la riproduzione di un problema. Dopo aver riprodotto il problema, disattivare i debug.

Nota: in Finesse 9.0(1) non è disponibile la registrazione a livello di debug. La registrazione a livello di debug è stata introdotta in Finesse 9.1(1). Il processo di abilitazione della registrazione è diverso in 9.1(1) rispetto a Finesse 10.0(1) - 11.6(1). Per questo processo, consultare la Guida all'amministrazione e alla manutenzione di Finesse.

Abilitare i registri di debug del Servizio di notifica di Unified Contact Center Express (UCCX), come mostrato:

```
<#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.
```

Abilitare i registri di debug del servizio di notifica di Unified Contact Center Enterprise (UCCE) (Finesse Standalone), come mostrato:

```
<#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

Questi registri si trovano nella cartella /desktop/logs/openfire e sono denominati debug.log.

Come mostrato nell'immagine, il file debug.log del servizio di notifica (Openfire) mostra il binding http con il desktop, l'indirizzo IP e la porta del PC dell'agente.

```
2017.04.14 21:34:21 [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$SelectChannelHttpConnection@2d5a26@XXX.XXX.XXX.XXX
2017.04.14 21:34:21 scope null|/http-bind/ @ o.e.j.s.ServletContextHandler{/http-bind,null}
2017.04.14 21:34:21 context=/http-bind|/ @ o.e.j.s.ServletContextHandler{/http-bind,null}
2017.04.14 21:34:21 sessionManager=org.eclipse.jetty.session.HashSessionManager@176fe4#STARTED
2017.04.14 21:34:21 session=null
2017.04.14 21:34:21 session=null
2017.04.14 21:34:21 servlet /http-bind|/ -> org.jivesoftware.openfire.http.HttpBindServlet-1643193
2017.04.14 21:34:21 chain=null
2017.04.14 21:34:21 HTTPBindLog: HTTP RECV(3445afbe): <body sid="3445afbe" rid="164053266"/>
2017.04.14 21:34:21 consumeResponse: org.jivesoftware.openfire.http.HttpSession@dd7653 status: 3 address: 1001003@XXX.XXX.XXX.XXX.cisco.com
<presence from="1001003@XXX.XXX.XXX.XXX.cisco.com/desktop">
  <c xmlns="http://jabber.org/protocol/caps" hash="sha-1" node="http://jabber.cisco.com/caxl" ver="VNC6fNwvCxe6FJfDJlryVJRwM="/>
</presence> rid: 164053266
2017.04.14 21:34:21 suspended org.eclipse.jetty.server.nio.SelectChannelConnector$SelectChannelHttpConnection@2d5a26@XXX.XXX.XXX.XX:7443<->
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
```

Come mostrato nell'immagine, gli ultimi 0 ms attivi indicano che la sessione è ancora attiva.

```
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.XXXXXXXXX.cisco.com/
2017.04.14 21:34:26 time=1492185866851,JID=1001003@XXXXXXXX.XXXXXXXXX.cisco.com/desktop,msgs_sent=4,msgs_
2017.04.14 21:34:26 time=1492185866851,JID=1001003@XXXXXXXX.XXXXXXXXX.cisco.com/desktop,msgs_sent=4,msgs_
```

Openfire chiudendo la sessione inattiva indica che la disconnessione dell'agente può essere attivata in 60 secondi quando Finesse può inviare una disconnessione forzata con un codice motivo di 255 al server CTI. Il comportamento effettivo del desktop in queste condizioni dipende dall'impostazione di Disconnessione agente (LOAD) in UCCE. In UCCX, questo è sempre il comportamento.

Se il client Finesse non invia messaggi di bind http al server Finesse, i log possono mostrare il tempo di attività della sessione e la chiusura della sessione.

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

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

Registri servizio di notifica informazioni

Questi registri si trovano nella cartella /desktop/logs/openfire e sono denominati info.log. Se il client Finesse non invia messaggi di bind http al server Finesse, i log possono indicare che la sessione è diventata inattiva.

```
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
```

Registri servizi Web

Questi registri si trovano nella cartella /desktop/logs/webservices e sono denominati Desktop-webservices.YYYY-MM-DDTHH-MM-SS.sss.log. Se il client Finesse non invia messaggi di bind http al server Finesse entro il periodo di tempo specificato, i log possono indicare che la presenza dell'agente non è più disponibile e 60 secondi dopo, può verificarsi una disconnessione basata sulla presenza.

```
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-UNSUBSCRIBED
0000001044: XX.XX.XX.XXX: Jun 17 2017 00:16:04.631 +0530: %CCBU_Smack Listener Processor (1)-6-AGENT_PRESENT
0000001051: XX.XX.XX.XXX: Jun 17 2017 00:16:35.384 +0530: %CCBU_pool-8-thread-1-6-AGENT_PRESENCE_MONITORING
0000001060: XX.XX.XX.XXX: Jun 17 2017 00:17:04.632 +0530: %CCBU_CoreImpl-worker12-6-PRESENCE DRIVEN LOGOUT
0000001061: XX.XX.XX.XXX: Jun 17 2017 00:17:04.633 +0530: %CCBU_CoreImpl-worker12-6-MESSAGE_TO_CTI_SERVER
1, workmode : 0, reason code: 255, forceflag :1, agentcapacity: 1, agenttext: 1001003, agentid: 1001003,
0000001066: XX.XX.XX.XXX: Jun 17 2017 00:17:04.643 +0530: %CCBU_CTIMessageEventExecutor-0-6-DECODED_MESSAGE
skillGroupNumber=-1, skillGroupPriority=0, agentState=1 (LOGOUT), eventReasonCode=255, numFltSkillGroups=0,
duration=null, nextAgentState=null, fltSkillGroupNumberList=[], fltSkillGroupIDList=[], fltSkillGroupPriorityList=[],
msgID=30, timeTracker={"id":"AgentStateEvent","CTI_MSG_RECEIVED":1497638824642,"CTI_MSG_DISPATCH":1497638824642}
Decoded Message to Finesse from backend cti server
```

Motivi comuni della disconnessione di BOSH

Le connessioni BOSH vengono impostate dal client Web e il server Finesse determina se la presenza dell'agente non è disponibile. Si tratta quasi sempre di problemi sul lato client relativi al browser, al computer agente o alla rete, in quanto l'onere di avviare la connessione spetta al client.

Problema - Gli agenti si disconnettono in momenti diversi (problema sul lato client)

Azioni consigliate

Verificare i seguenti problemi:

1. Problema di rete:

- Controllare le regole e i registri del firewall: la porta TCP 7443 non deve essere bloccata o limitata
- Utilizzare uno sniffer del traffico Web HTTP come [Fiddler®](#) o [Wireshark®](#) per confermare che il browser invia richieste di binding HTTP tramite la porta TCP 7443 e riceve risposte

- Verificare che tutti i dispositivi/interfacce di rete tra il computer agente e il server Finesse non presentino ritardi eccessivi o perdite di pacchetti
 - Il comando traceroute può essere utile per determinare il percorso e i ritardi
 - Su un PC Microsoft® Windows®: tracert {Finesse Server IP FQDN di | Finesse Server}
 - Su un Mac®: traceroute {Finesse Server IP FQDN di | Finesse Server}
 - Sul software Cisco IOS®, è possibile controllare le statistiche dell'interfaccia: show interfaces
 - Fare riferimento alla sezione [Risoluzione dei problemi relativi alle perdite delle code di input e di output](#)
- Raccogliere i log di Finesse Client per un agente di test. I log dei client possono essere raccolti in tre modi:
 1. Registri console Web browser
 - [Console Web Firefox](#)
 - [Console Web Microsoft Edge](#)
 - [Console Web Chrome](#)
 2. Fare clic sul pulsante [Invia segnalazione errori](#) nella pagina Finesse e raccogliere i log del server Finesse. I registri si trovano in /desktop/logs/clientlogs.
 3. Accedere tramite https://<Finesse-FQDN>/desktop/locallog e raccogliere i log dopo il verificarsi del problema.

Ogni minuto il client si connette al server Finesse per calcolare la deviazione e la latenza di rete:

```
<PC date-time with GMT offset>: : <Finesse FQDN>: <Finesse server date-time with offset>:
Header : Client: <date-time>, Server: <date-time>, Drift: <drift> ms, Network Latency (round trip): <RTT>
2019-01-11T12:24:14.586 -05:00: : fin1.ucce.local: Jan 11 2019 11:24:14.577 -0600: Header : Client: 2019-
```

In caso di problemi di raccolta dei log, consultare il documento sulla [risoluzione dei problemi di registrazione persistente di Cisco Finesse Desktop](#)

2. Browser e/o versione non supportati:

Usa browser/versions e impostazioni supportati in base alle matrici di compatibilità:

[Matrice di compatibilità UCCE](#)

[Matrice di compatibilità UCCX](#)

3. Condizione di blocco del browser a causa del contenuto/elaborazione di altre schede/finestre:

Controllare il flusso di lavoro dell'agente per verificare se:

- In genere sono disponibili altre schede o finestre che eseguono costantemente altre applicazioni in tempo reale, ad esempio streaming di musica/video, connessioni WebSocket, client Web CRM (Customer Relationship Management) personalizzati e così via
- Numero elevato di schede o finestre aperte
- Disattiva memorizzazione nella cache del browser
- Hanno mantenuto il browser in esecuzione per molto tempo e non chiuderlo alla fine della giornata lavorativa

4. Computer messo in sospensione:

Verificare se l'agente mette il computer in sospensione prima di disconnettersi da Finesse o se il timer di impostazione della sospensione del computer è molto basso.

5. Problema elevato di CPU o di memoria nel computer client:

- Se il browser agente viene eseguito in un ambiente condiviso, ad esempio Microsoft Windows Remote Desktop Services, Citrix® XenApp®, Citrix XenDesktop®, determinare se le prestazioni del browser dipendono dal numero di utenti che eseguono contemporaneamente il browser
 - Verificare che la memoria e le risorse CPU appropriate siano configurate in base al numero di utenti
- Verificare i problemi di utilizzo delle risorse del computer:
 - Windows:
 - Comando Windows [PowerShell Get-Counter](#) che controlla la % di tempo CPU, megabyte di memoria disponibile e la % di memoria in uso ogni 2 secondi: `Get-Counter -Counter "\Processor(_Total)\% Tempo processore", "\Memoria\MByte disponibili", "\Memoria\% Byte vincolati in uso" -SampleInterval 2 -Continuous`
 - In alternativa all'utilizzo di PowerShell per visualizzare i contatori delle prestazioni di Windows, è possibile utilizzare [Monitoraggio prestazioni di Windows](#)
 - [Task Manager](#) può essere utilizzato per visualizzare le statistiche dinamiche della CPU e della memoria a livello globale e per ogni singolo processo
 - Mac:
 - [Comando Terminal Top](#) che controlla in tempo reale la CPU e la memoria totali: `top`
 - Verifica i processi e ordina per utilizzo CPU: `inizio CPU`
 - Verifica i processi e ordina per utilizzo memoria: `inizio -o MEM`
 - [Activity Monitor](#) può essere utilizzato per visualizzare le statistiche in tempo reale della CPU e della memoria a livello globale e per ogni singolo processo

6. Gadget di terze parti che eseguono attività impreviste e problematiche in background:

Verificare il comportamento del desktop Finesse rimuovendo tutti i gadget di terze parti.

7. Problema NTP su server o client:

- Controllare **lo stato ntp** sul server di pubblicazione Finesse per verificare che lo stato del server NTP sia 4 o inferiore
- Nei log del client, verificare la deviazione e la latenza di rete

Problema - Tutti gli agenti si disconnettono contemporaneamente (problema sul lato server)

Azioni consigliate

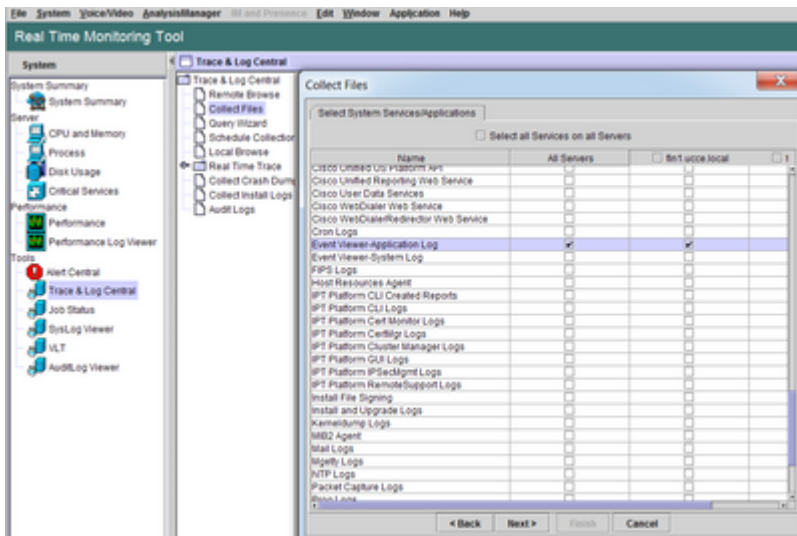
Verificare i seguenti problemi:

1. Disconnessione del servizio Cisco Unified Communications Manager CTIManager. Se tutti i provider CTIManager per UCCX vengono arrestati o arrestati in modo anomalo, gli agenti UCCX visualizzeranno il banner rosso. Gli agenti UCCE non vedono il banner rosso se ciò accade, ma le chiamate non riescono a indirizzare correttamente gli agenti.

- Verificare che il servizio Cisco CTIManager sia stato avviato sui server CUCM utilizzati come provider CTI
- Verificare se il servizio Cisco CTIManager si è arrestato in modo anomalo tramite il Visualizzatore eventi - L'applicazione accede a RTMT per verificare se si è verificato un arresto anomalo del servizio

Cisco CTIManager.

- Per raccogliere i registri del visualizzatore eventi in RTMT, selezionare **Sistema > Strumenti > Trace and Log Central > Raccogli file > Seleziona servizi/applicazioni di sistema > Visualizzatore eventi - Registro applicazioni.**



- Per raccogliere i log di Visualizzatore eventi-Applicazione sulla CLI: file get activelog /syslog/CiscoSyslog* abtime hh:mm:MM/DD/YY hh:mm:MM/DD/YY
- Per visualizzare i dump di base sulla CLI: utilizza l'elenco core active

Nota: i nomi dei file di dump di base utilizzano il formato:
core.<ProcessID>.<SignalNumber>.<ProcessName>.<EpochTime>.
Esempio: core.24587.6.CTManager.1533441238
Quindi, l'ora dell'incidente può essere determinata dall'ora dell'epoca.

2. Arresto o arresto anomalo del servizio di notifica Finesse/UCCX:

- Verificare nei registri applicazioni del Visualizzatore eventi la presenza di errori del servizio di notifica o verificare se il servizio è stato arrestato
- Verificare se il servizio di notifica è attivo: elenco servizi utilizzati
- Controllare gli orari di arresto del servizio di notifica: file search activelog /desktop/logs/openfire "Openfire stop"
- Controllare gli orari di avvio del servizio di notifica: file search activelog /desktop/logs/openfire "HTTP bind service STARTED"
- Verificare la presenza di dump della memoria del servizio di notifica causati da un arresto anomalo del sistema: file list activelog /desktop/logs/openfire/*.hprof
- Verificare che il servizio di notifica sia in ascolto del traffico sulla porta TCP 7443: show open ports regexp 7443.*LISTEN
- Verificare se questi difetti sono applicabili (tali difetti potrebbero causare errori di accesso per gli agenti che eseguono l'accesso e, per gli agenti già connessi, tali agenti vedrebbero il banner rosso Finesse disconnect message):
 - Cisco ID bug [CSCva72280](#) - Finesse Tomcat e Openfire Crash per caratteri XML non validi
 - Cisco ID bug [CSCva72325](#) - UCCX: Finesse Tomcat e Openfire Crash per caratteri XML non validi

Riavviare Cisco Finesse Tomcat e il servizio di notifica in caso di sospetto arresto anomalo. Questa operazione è consigliata solo in caso di interruzione della rete, altrimenti questi riavviano gli agenti di disconnessione dal server Finesse.

Passaggi per UCCE:

- utils service stop Cisco Finesse Tomcat
- utils service stop Cisco Finesse Notification Service
- utilità avvio servizio Cisco Finesse Tomcat
- servizio utils avvio del servizio di notifica Cisco Finesse

Passaggi per UCCX:

- utils service stop Cisco Finesse Tomcat
- utilizza il servizio stop Cisco Unified CCX Notification Service
- utilità avvio servizio Cisco Finesse Tomcat
- utilità avvio servizio Cisco Unified CCX Notification Service

Usa filtro

Configurare Fiddler può essere un'attività piuttosto impegnativa senza comprendere i passaggi necessari e il funzionamento di Fiddler. Fiddler è un proxy web man-in-the-middle che si trova tra il client Finesse (browser web) e il server Finesse. A causa delle connessioni protette tra il client Finesse e il server Finesse, ciò aggiunge un livello di complessità alla configurazione del Fiddler per visualizzare i messaggi protetti.

Problema comune del richiedente

Poiché Fiddler si trova tra il client Finesse e il server Finesse, l'applicazione Finesse deve creare certificati firmati per tutte le porte TCP Finesse che richiedono certificati:

Certificati di servizio Cisco Finesse Tomcat

1. Server Finesse Publisher TCP 8445 (e/o 443 per UCCE)
2. Finesse Subscriber Server TCP 8445 (e/o 443 per UCCE)

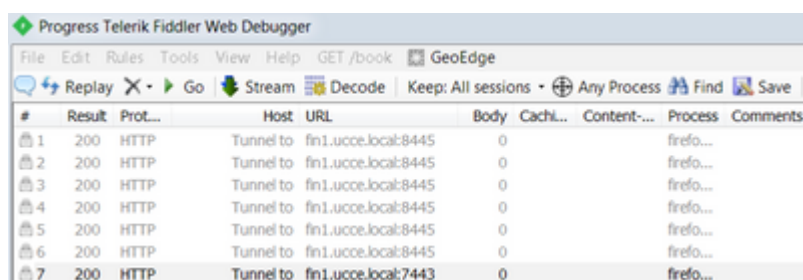
Certificati del servizio di notifica Cisco Finesse (Unified CCX)

1. Finesse publisher server TCP 7443
2. Finesse Subscriber Server TCP 7443

La decrittografia HTTPS deve essere abilitata affinché Fiddler generi dinamicamente certificati per conto del server Finesse. Questa opzione non è attivata per impostazione predefinita.

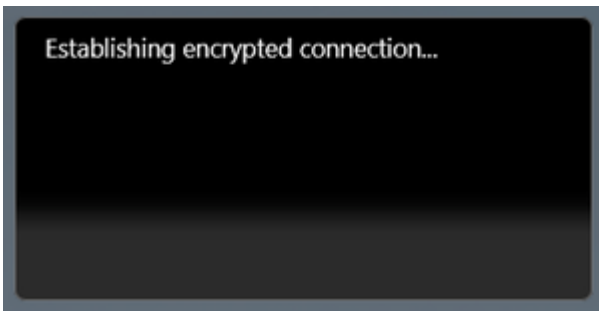
Se la decrittografia HTTPS non è configurata, viene rilevata la connessione del tunnel iniziale al servizio di notifica, ma non il traffico http-bind. Fiddler mostra solo:

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



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

I certificati Finesse firmati da Fiddler devono quindi essere considerati attendibili dal client. Se questi certificati non sono considerati attendibili, non è possibile passare oltre la fase Stabilire una connessione crittografata... di accesso Finesse.



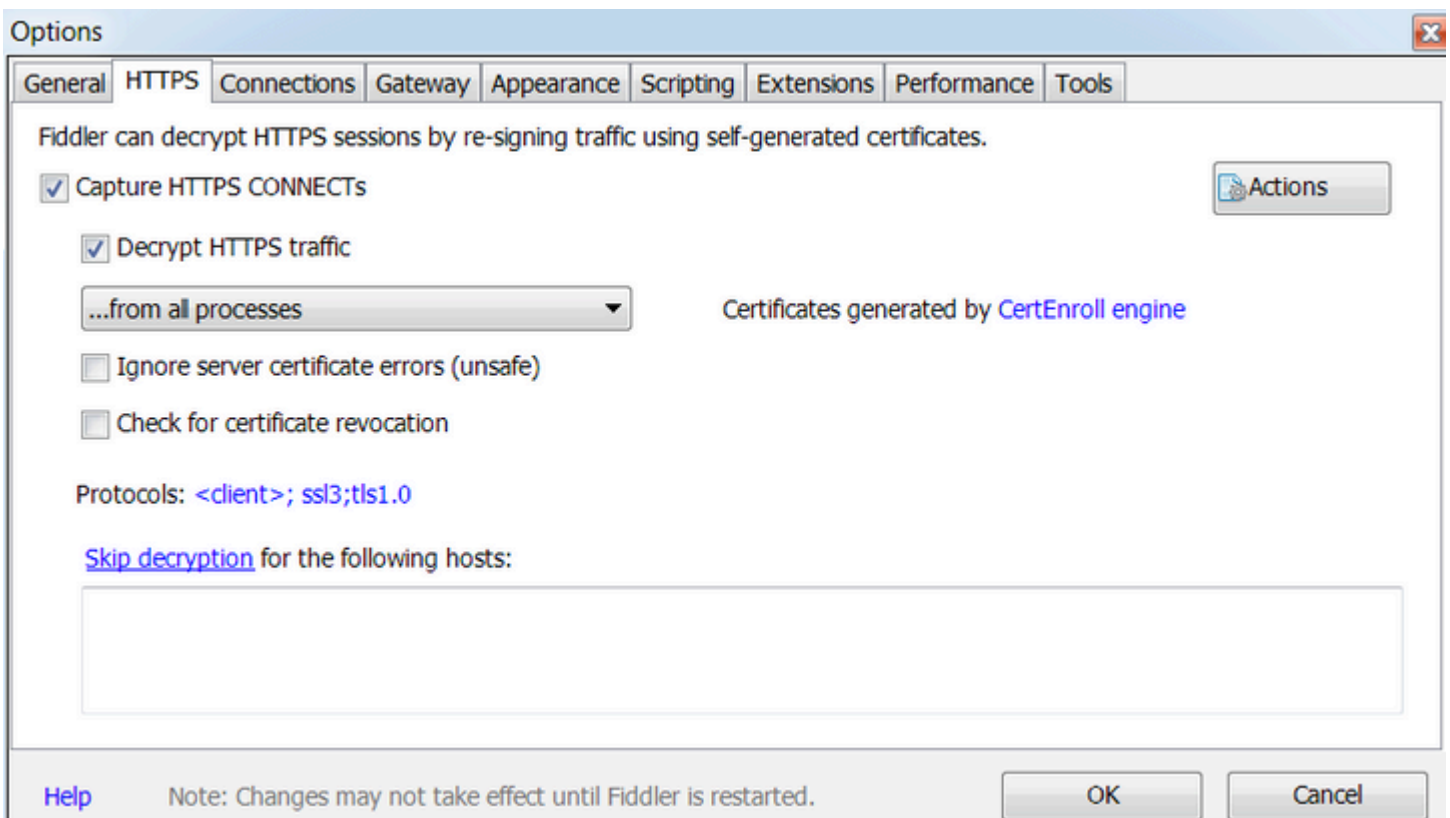
In alcuni casi, l'accettazione delle eccezioni del certificato dall'accesso non funziona e i certificati devono essere considerati attendibili dal browser manualmente.

Procedura di configurazione di esempio

Attenzione: la configurazione di esempio fornita è per Fiddler v5.0.20182.28034 per .NET 4.5 e Mozilla Firefox 64.0.2 (32 bit) su Windows 7 x64 in un ambiente lab. Queste procedure non possono essere generalizzate in tutte le versioni di Fiddler, in tutti i browser o in tutti i sistemi operativi. Se la rete è operativa, valutare attentamente eventuali conseguenze derivanti dalla configurazione. Fare riferimento alla [documentazione ufficiale del Finder](#) per ulteriori informazioni.

Passaggio 1. Scarica trovatore

Passaggio 2. Abilita decrittografia HTTPS. Selezionare **Strumenti > Opzioni > HTTPS** e selezionare la casella di controllo **Decrittografa traffico HTTPS**.

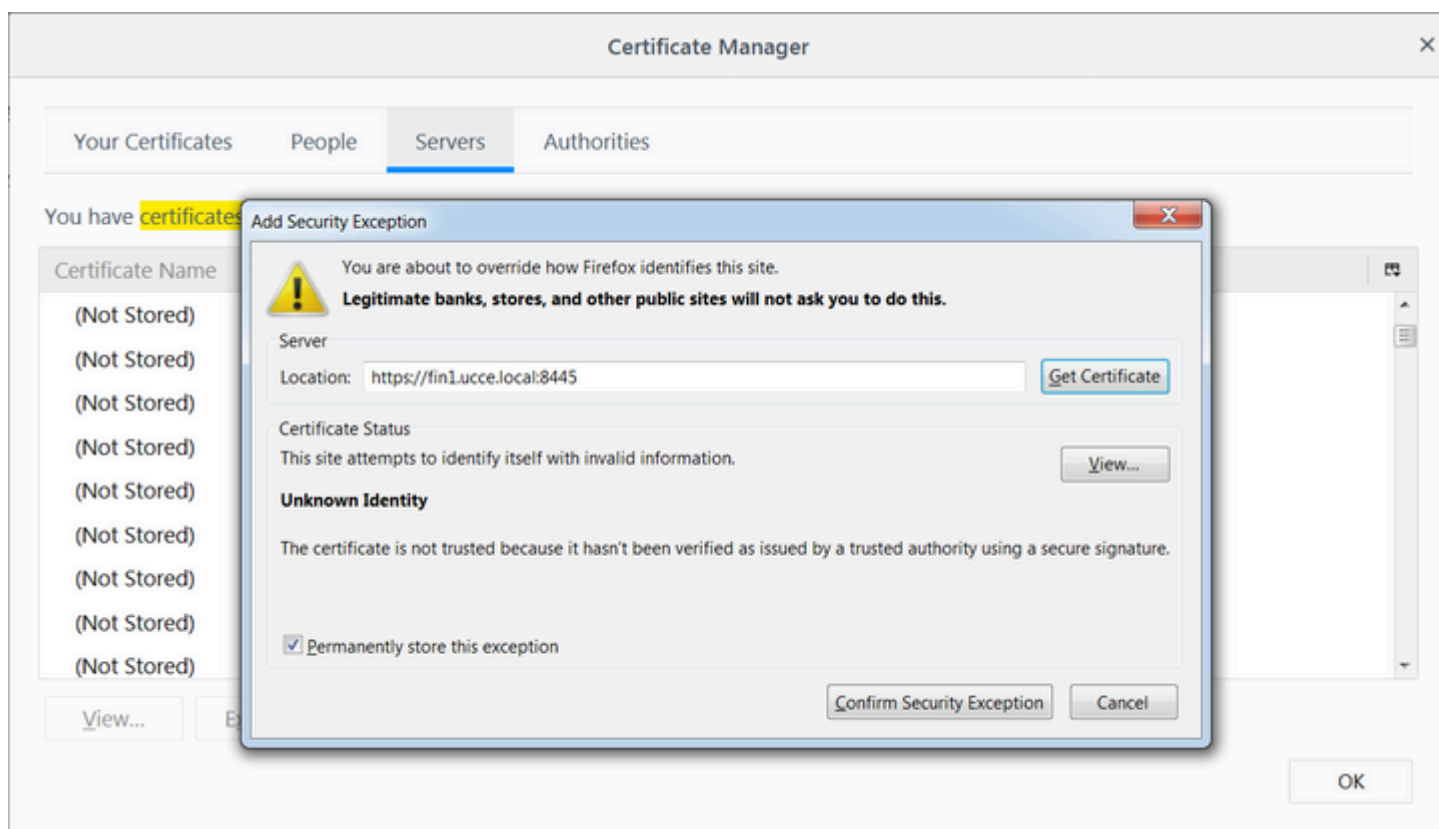


Passaggio 3. Verrà visualizzata una finestra di messaggio di avviso in cui viene richiesto di confermare l'attendibilità del certificato radice del provider di servizi di individuazione. Selezionare **Sì**.

Passaggio 4. Viene visualizzata una finestra di messaggio di avviso con il messaggio "Si sta per installare un certificato da un'Autorità di certificazione (CA) che dichiara di rappresentare: DO_NOT_TRUST_FiddlerRoot... Installare il certificato?". Selezionare **Sì**.

Passaggio 5. Aggiungere manualmente i certificati di editore e sottoscrittore Finesse all'archivio certificati del computer o del browser. Verificare le porte 8445, 7443 e (solo per UCCE) 443. Ad esempio, in Firefox, è possibile eseguire questa operazione semplicemente senza scaricare i certificati dalla pagina Amministrazione del sistema operativo Finesse:

Opzioni > Trova in Opzioni (ricerca) > Certificati > Server > Aggiungi eccezione > Posizione > Immettere https://<Server Finesse>:porta per le porte rilevanti per entrambi i server Finesse.



Passaggio 6. Accedere a Finesse e vedere i messaggi http-bind lasciare il client Finesse al server Finesse tramite Fiddler.

Nell'esempio fornito, i primi 5 messaggi mostrano i messaggi http-bind a cui il server Finesse ha risposto. Il primo messaggio contiene 1571 byte di dati restituiti nel corpo del messaggio. Il corpo contiene un aggiornamento XMPP relativo a un evento agente. Il messaggio http-bind finale è stato inviato dal client Finesse, ma non ha ricevuto risposta dal server Finesse. È possibile determinare questa condizione quando il risultato HTTP è null (-) e il numero di byte nel corpo della risposta è null (-1).

Progress Telerik Fiddler Web Debugger

File Edit Rules Tools View Help GET /book GeoEdge

Replay X Go Stream Decode Keep: All sessions Any Process Find Save Browse Clear Cache TextWizard Tearoff

#	Result	Prot...	Host	URL	Body	Cach...	Content...	Process	Comments	Custo
6...	200	HTTPS	fin1.uccelocal:...	/desktop/thirdparty/...	1,135		text/java...	firefo...		
6...	200	HTTPS	fin1.uccelocal:...	/desktop/thirdparty/...	1,655		text/java...	firefo...		
6...	200	HTTPS	fin1.uccelocal:...	/desktop/thirdparty/...	3,579		text/java...	firefo...		
6...	200	HTTPS	fin1.uccelocal:...	/desktop/thirdparty/...	4,744		text/java...	firefo...		
6...	200	HTTPS	fin1.uccelocal:...	/desktop/thirdparty/...	1,630		text/java...	firefo...		
6...	200	HTTPS	fin1.uccelocal:...	/desktop/thirdparty/...	812		text/html	firefo...		
6...	200	HTTPS	fin1.uccelocal:...	/desktop/thirdparty/...	729		text/html	firefo...		
6...	200	HTTPS	fin1.uccelocal:...	/desktop/thirdparty/...	352		text/html	firefo...		
6...	200	HTTP	detectportal.fire...	/success.txt	8	no-ca...	text/plain	firefo...		
6...	200	HTTPS	fin1.uccelocal:...	/desktop/thirdparty/...	244		text/html	firefo...		
6...	200	HTTPS	fin1.uccelocal:...	/desktop/thirdparty/...	731		text/html	firefo...		
6...	200	HTTPS	fin1.uccelocal:...	/desktop/thirdparty/...	901		text/html	firefo...		
6...	200	HTTPS	fin1.uccelocal:...	/desktop/thirdparty/...	1,302		text/html	firefo...		
6...	200	HTTPS	fin1.uccelocal:...	/desktop/thirdparty/...	307		text/html	firefo...		
6...	200	HTTPS	fin1.uccelocal:...	/desktop/thirdparty/...	287		text/html	firefo...		
6...	200	HTTPS	fin1.uccelocal:...	/desktop/thirdparty/...	569		text/html	firefo...		
6...	200	HTTPS	fin1.uccelocal:...	/desktop/thirdparty/...	910		text/html	firefo...		
6...	200	HTTP	detectportal.fire...	/success.txt	8	no-ca...	text/plain	firefo...		
6...	200	HTTPS	fin1.uccelocal:...	/desktop/thirdparty/...	43		image/gif	firefo...		
6...	200	HTTPS	fin1.uccelocal:...	/desktop/ciscowidge...	1,176		text/html	firefo...		
6...	200	HTTPS	fin1.uccelocal:...	/desktop/theme/fine...	673		image/gif	firefo...		
6...	200	HTTPS	fin1.uccelocal:...	/desktop/ciscowidge...	720		text/html	firefo...		
6...	200	HTTPS	fin1.uccelocal:...	/finesse/api/User/47...	631	no-ca...	applicato...	firefo...		
6...	200	HTTPS	fin1.uccelocal:...	/desktop/thirdparty/...	12,7...		image/png	firefo...		
6...	200	HTTPS	fin1.uccelocal:...	/desktop/theme/fine...	2,205		image/png	firefo...		
6...	200	HTTPS	fin1.uccelocal:...	/finesse/api/User/47...	340	no-ca...	applicato...	firefo...		
6...	200	HTTPS	fin1.uccelocal:...	/finesse/api/User/47...	1,851	no-ca...	applicato...	firefo...		
6...	200	HTTPS	fin1.uccelocal:...	/finesse/api/User/47...	20	no-ca...	applicato...	firefo...		
6...	200	HTTPS	fin1.uccelocal:...	/gadgets/makeRequ...	340	no-ca...	applicato...	firefo...		
6...	200	HTTP	Tunnel to	cuic1.uccelocal:8444	0			firefo...		
6...	200	HTTPS	fin1.uccelocal:...	/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.uccelocal:8444	0			firefo...		
6...	200	HTTP	detectportal.fire...	/success.txt	8	no-ca...	text/plain	firefo...		
6...	200	HTTPS	fin1.uccelocal:...	/http-bind/	1,571		text/xml...	firefo...		
6...	202	HTTPS	fin1.uccelocal:...	/finesse/api/User/47...	0	no-ca...	applicato...	firefo...		
6...	200	HTTPS	fin1.uccelocal:...	/desktop/theme/fine...	673		image/gif	firefo...		
6...	200	HTTPS	fin1.uccelocal:...	/http-bind/	57		text/xml...	firefo...		
6...	200	HTTPS	fin1.uccelocal:...	/finesse/api/SystemL...	232	no-ca...	applicato...	firefo...		
6...	200	HTTPS	fin1.uccelocal:...	/http-bind/	57		text/xml...	firefo...		
6...	200	HTTPS	fin1.uccelocal:...	/http-bind/	57		text/xml...	firefo...		
6...	200	HTTPS	fin1.uccelocal:...	/finesse/api/SystemL...	232	no-ca...	applicato...	firefo...		
6...	200	HTTPS	fin1.uccelocal:...	/http-bind/	57		text/xml...	firefo...		
6...	-	HTTPS	fin1.uccelocal:...	/http-bind/	-1			firefo...		
6...	200	HTTPS	fin1.uccelocal:...	/finesse/api/SystemL...	232	no-ca...	applicato...	firefo...		

Statistics Inspectors AutoResponder Compos

Headers TextView SyntaxView WebForms HexView

POST https://fin1.uccelocal:7443/http-bind/ HT
Host: fin1.uccelocal:7443
User-Agent: Mozilla/5.0 (windows NT 6.1; WOW64;
Accept: text/plain, */*; q=0.01
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Referer: https://fin1.uccelocal:7443/tunnel
Content-Type: text/xml
X-Requested-With: XMLHttpRequest
Content-Length: 83
Cookie: finesse_ag_extension=10005; JSESSIONID=
Connection: keep-alive
Pragma: no-cache
Cache-Control: no-cache

<body xmlns="http://jabber.org/protocol/httpbind

Find... (press Ctrl+Enter to highlight all)

Transformer Headers TextView SyntaxView ImageV

Raw JSON XML

```
<body xmlns="http://jabber.org/protocol/httpbind"><message
to="47483648@fin1.uccelocal" id="/finesse/api/User/47483648"
xmlns="http://jabber.org/protocol/pubsub#event"><items nod
4752-8a1d-5adbdc74a7717><notification xmlns="http://jab
&lt;data&gt;
&lt;dialogs&gt;/finesse/api/User/47483648/Dialogs&lt;/dia
&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;/wrapUpOnIncoming&gt;OPTIONAL&lt;/wrapUpOnInco
&lt;/settings&gt;
&lt;state&gt;READY&lt;/state&gt;
&lt;stateChangeTime&gt;2019-01-11T23:56:54.783Z&lt;/
&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-e4c9-ef50ab5e7cc6&lt;/
&lt;source&gt;/finesse/api/User/47483648&lt;/source&gt;
&lt;/Update&gt;</notification></item></items></event></mess
```

0:0 0/1,571 Find... (press Ctrl+Enter to hig

QuickExec] ALT+Q > type HELP to learn more

Capturing All Processes 1 / 693 https://fin1.uccelocal:7443/http-bind/

Visualizzazione più dettagliata dei dati:

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...
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...
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...
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...

Corpo risposta per messaggio XMPP:

```
<body xmlns='http://jabber.org/protocol/httpbind'><message xmlns="jabber:client" from="pubsub.fin1.ucce.local"
to="47483648@fin1.ucce.local" id="/finesse/api/User/47483648__47483648@fin1.ucce.local__K7hYF"><event
xmlns="http://jabber.org/protocol/pubsub#event"><items node="/finesse/api/User/47483648"><item id="26a3e421-
4752-8a1d-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>
```

Utilizzare Wireshark

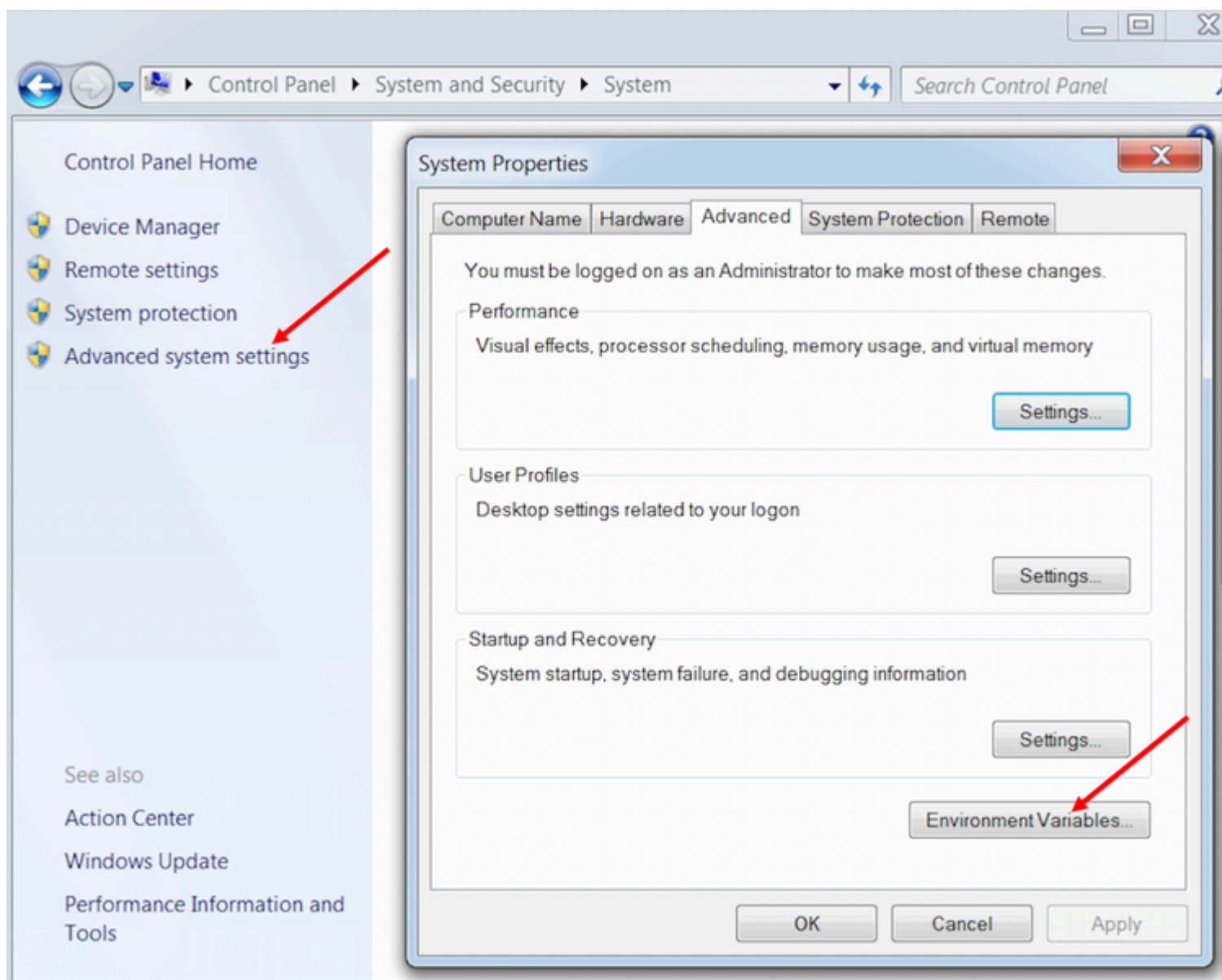
Wireshark è uno strumento comunemente usato per lo sniffing dei pacchetti che può essere usato per sniffare e decodificare il traffico HTTPS. Il traffico HTTPS è il traffico HTTP protetto tramite TLS (Transport Layer Security). TLS fornisce integrità, autenticazione e riservatezza tra due host. Viene

comunemente utilizzato nelle applicazioni Web, ma può essere utilizzato con qualsiasi protocollo che utilizzi TCP come protocollo del livello trasporto. SSL (Secure Sockets Layer) è la versione precedente del protocollo TLS, che non viene più utilizzata in quanto non protetta. Questi nomi vengono spesso utilizzati in modo intercambiabile e il filtro Wireshark utilizzato per il traffico SSL o TLS è ssl.

Attenzione: la configurazione di esempio fornita è per Wireshark 2.6.6 (v2.6.6-0-gdf942cd8) e Mozilla Firefox 64.0.2 (32 bit) su Windows7 x64 in un ambiente lab. Queste procedure non possono essere generalizzate in tutte le versioni di Fiddler, in tutti i browser o in tutti i sistemi operativi. Se la rete è operativa, valutare attentamente eventuali conseguenze derivanti dalla configurazione. Fare riferimento alla [documentazione ufficiale di Wireshark SSL](#) per ulteriori informazioni. Wireshark 1.6 o superiore.

Nota: questo metodo può funzionare solo per Firefox e Chrome. Questo metodo non funziona per Microsoft Edge.

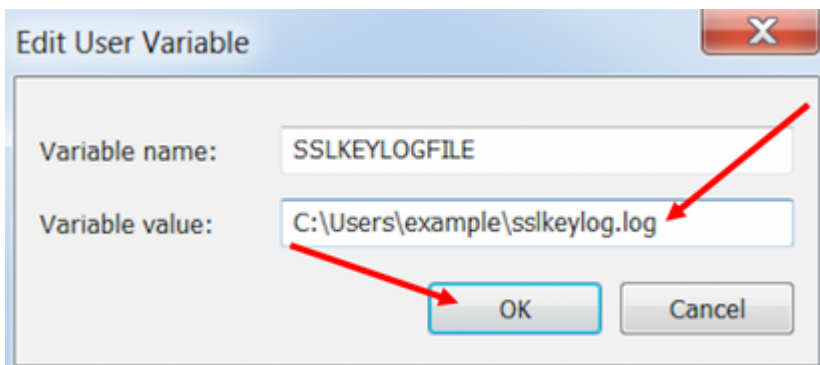
Passaggio 1. Sul PC Windows dell'agente passare a **Pannello di controllo > Sistema e sicurezza > Sistema > Impostazioni di sistema avanzate Variabili ambientali...**



Passaggio 2. Passare a **Variabili utente per utente <nomeutente> > Nuovo...**

Creare una variabile denominata **SSLKEYLOGFILE**.

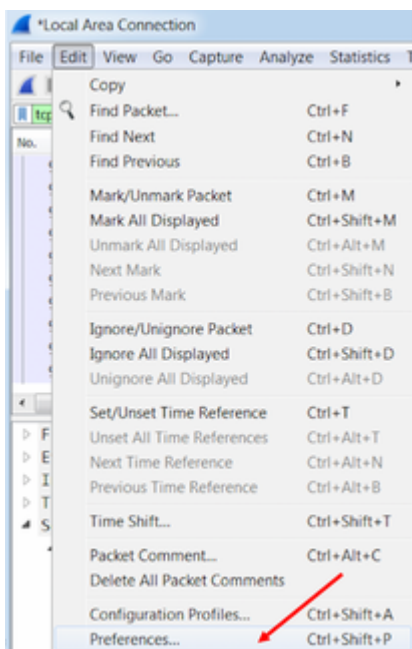
Creare un file per archiviare il segreto del premaster SSL in una directory privata:
`SSLKEYLOGFILE=</path/to/private/directory/with/logfile>`



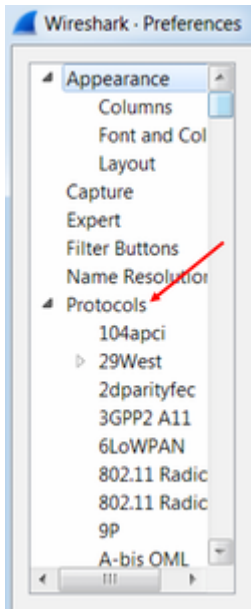
Nota: creare una variabile di sistema anziché una variabile utente e/o memorizzare il file in una directory non privata, ma tutti gli utenti del sistema possono accedere al segreto della premastro, che è meno sicuro.

Passaggio 3. Se Firefox o Chrome sono aperti, chiudere le applicazioni. Dopo la riapertura, possono iniziare a scrivere in SSLKEYLOGFILE.

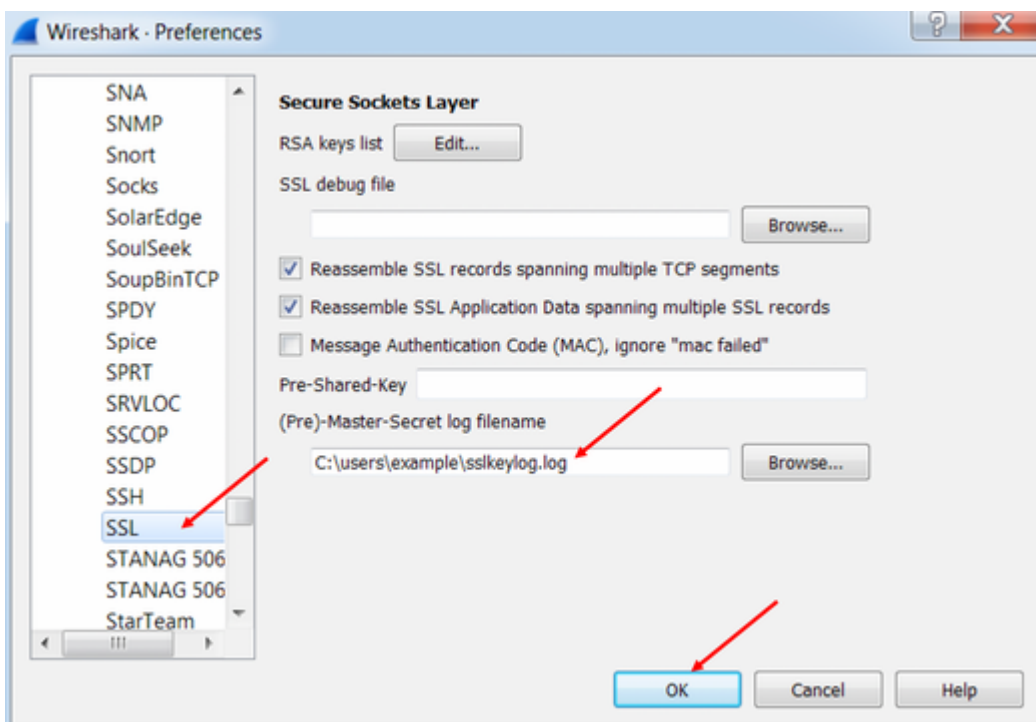
Passaggio 4. In Wireshark, selezionare **Modifica > Preferenze...**



Passare a **Protocolli > SSL**.



Passaggio 5. Immettere il percorso del nome file del registro segreto premaster configurato nel passaggio 2.



Passaggio 6. Utilizzare Wireshark filter **tcp.port==7443 && ssl**, la comunicazione HTTP protetta tra il client Finesse e il server Finesse (Servizio di notifica) risulta decrittografata.

Transmission Control Protocol, Src Port: 54979, **Dst Port: 7443** Seq: 21265, Ack: 42841, Len: 565

Secure Sockets Layer

TLSv1.2 Record Layer: Application Data Protocol: Application Data

Content Type: Application Data (23)

Version: TLS 1.2 (0x0303)

Length: 560

Encrypted Application Data: 1e001ee88fc1c9a026b0385007608afdfb46c0d4a277faa8...

0010	20 48 54 54 50 2f 31 2e 31 0d 0a 48 6f 73 74 3a	HTTP/1.1 · Host:
0020	20 66 69 6e 31 2e 75 63 63 65 2e 6c 6f 63 61 6c	fin1.uce.local
0030	3a 37 34 34 33 0d 0a 55 73 65 72 2d 41 67 65 6e	:7443 · User-Agent
0040	74 3a 20 4d 6f 7a 69 6c 6c 61 2f 35 2e 30 20 28	t: Mozilla/5.0 (
0050	57 69 6e 64 6f 77 73 20 4e 54 20 36 2e 31 3b 20	Windows NT 6.1;
0060	57 4f 57 36 34 3b 20 72 76 3a 36 34 2e 30 29 20	WOW64; rv:64.0)
0070	47 65 63 6b 6f 2f 32 30 31 30 30 31 30 31 20 46	Gecko/2010101 Firefox/6
0080	69 72 65 66 6f 78 2f 36 34 2e 30 0d 0a 41 63 63	4.0 · Accept:
0090	65 70 74 3a 20 74 65 78 74 2f 70 6c 61 69 6e 2c	text/plain,
00a0	20 2a 2f 2a 3b 20 71 3d 30 2e 30 31 0d 0a 41 63	*/*; q=0.01 · Ac
00b0	63 65 70 74 2d 4c 61 6e 67 75 61 67 65 3a 20 65	cept-Language: en-
00c0	6e 2d 55 53 2c 65 6e 3b 71 3d 30 2e 35 0d 0a 41	US,en; q=0.5 · A

Frame (619 bytes) **Decrypted SSL (513 bytes)**

wireshark_E6642FDE-A01F-4115-B2E4-85157AB917CB_20190125155406_a06084.pcapng

Packets: 127485 · Display

Difetti correlati

- Cisco ID bug [CSCva72280](#) - arresto anomalo di Finesse Tomcat e Openfire per caratteri XML non validi
- Cisco ID bug [CSCva72325](#) - UCCX: Finesse Tomcat e Openfire Crash per caratteri XML non validi

Informazioni correlate

- [Specifiche XMPP](#)
- [XEP-0124: BOSH](#)
- [XEP-0060: Pubblica-Sottoscrivi](#)
- [Console Web Firefox](#)
- [Console Web Microsoft Edge](#)
- [Console Web Chrome](#)
- [Windows PowerShell](#)
- [Monitoraggio prestazioni di Windows](#)
- [Risoluzione dei problemi relativi ai pacchetti eliminati nelle code di input e di output](#)
- [Task Manager di Windows](#)
- [Terminale Mac](#)
- [Monitoraggio attività Mac](#)
- [Download del filtro](#)
- [Configurazione del filtro](#)
- [Download di Wireshark](#)
- [Decrittografia SSL di Wireshark](#)
- [Documentazione e supporto tecnico “ Cisco Systems](#)

Informazioni su questa traduzione

Cisco ha tradotto questo documento utilizzando una combinazione di tecnologie automatiche e umane per offrire ai nostri utenti in tutto il mondo contenuti di supporto nella propria lingua. Si noti che anche la migliore traduzione automatica non sarà mai accurata come quella fornita da un traduttore professionista. Cisco Systems, Inc. non si assume alcuna responsabilità per l'accuratezza di queste traduzioni e consiglia di consultare sempre il documento originale in inglese (disponibile al link fornito).