Gaman

CUCM - External Call Control Feature

##### Document history

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Person | Version no. | Description |
| 2019-04-02 | Gaman Team | 1.0 | First version of the document |
| 2020-03-08 | Marek Słomiński | 1.1 | Troubleshooting information |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Table of Contents

[Table of Contents 1](#_Toc34551663)

[Copyright 2](#_Toc34551664)

[Use case scenario 3](#_Toc34551665)

[Scenario 1 3](#_Toc34551666)

[Scenario 2 3](#_Toc34551667)

[Proposed solution 4](#_Toc34551668)

[CUCM External Call Control Feature - Configuration 6](#_Toc34551669)

[ECC Requests & Responses 7](#_Toc34551670)

[Troubleshooting 9](#_Toc34551671)

[SSL Handshake error 9](#_Toc34551672)

[Documentation 12](#_Toc34551673)

[Contact Information 13](#_Toc34551674)

# Copyright

Copyright (C) 2020 Gaman SP z o.o

<https://gaman-gt.com>

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

# Use case scenario

Consider CUCM deployment in hotel/motel. In this deployment let’s look at two possible scenarios:

## Scenario 1

In this scenario end devices (phones, Jabbers) are typically available to people who don’t belong to your organization. As usual people can make outbound calls with the default dialing policy (emergency, local, national). Sometimes people can buy premium services that will allow them to make international calls or premium calls. At the end the question is – how to dynamically adjust those settings.

From the engineering point of view – answer is very simple. Configure different Calling Search Spaces and the sign them to the phone line once customer buys premium service. From the business point of view – people will ask what Call Manager is and how I need to set those policies.

Of course, you can build a system which will simplify the process – for example integrates via AXL and changes line configuration. But remember every upgrade requires validation of the provided service.

We think that this approach is complex therefore we looked at External Call Control Feature.

## Scenario 2

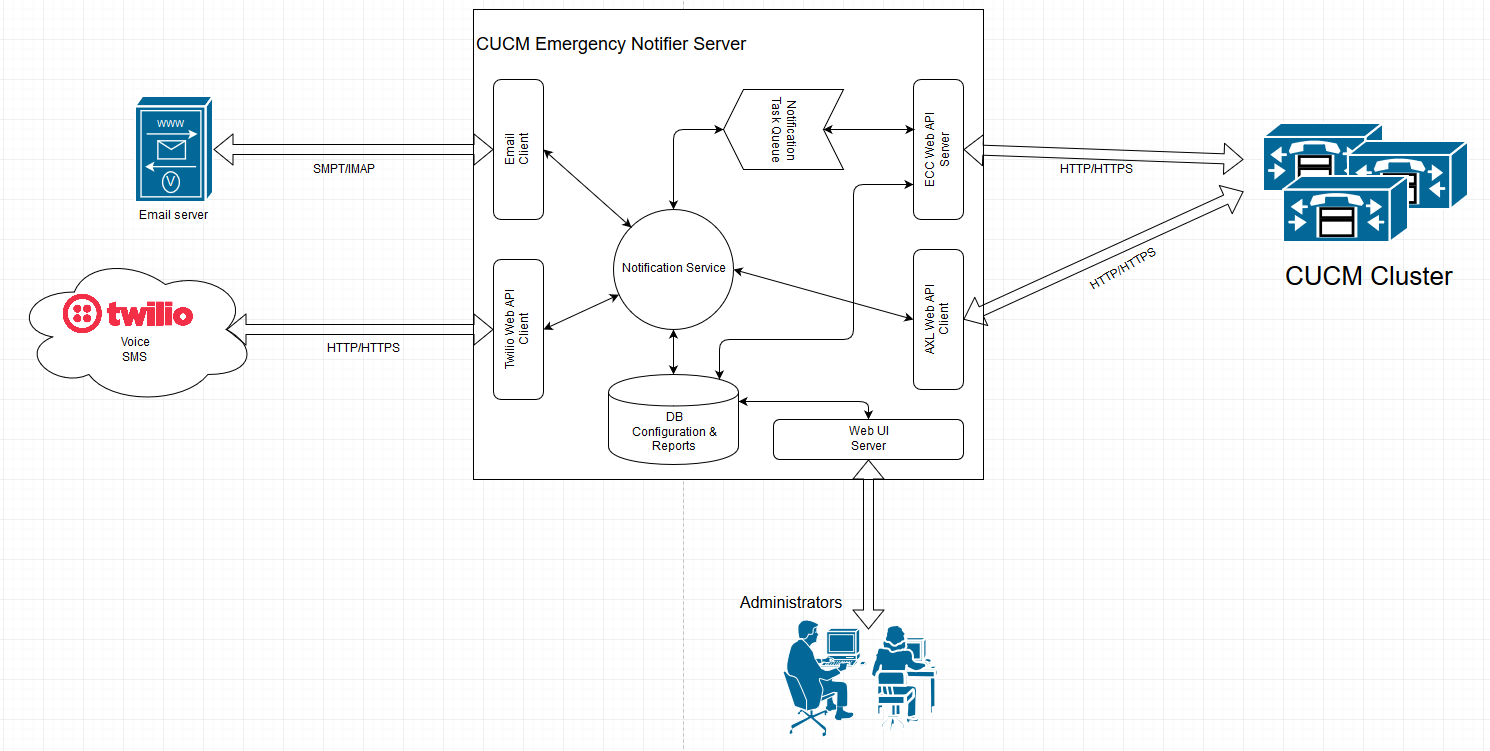
In this scenario we assume the hotel gest dials emergency number. Typical dial plan will allow to use this number and gest will be connected to Emergency Center to get the help. But in general, before service arrive it may be too late. So, the question is how we can improve the functionality of the system?

Again, we can look at the CUCM External Call Control Feature which will notify hotel room service that can react quicker that Emergency services.

In the next chapter you will find description of the solution that may well fit in both described above scenarios. At the end we decided to focus on the second scenario as it allows to save people’s lives.

# Proposed solution

Below is a diagram that illustrates how the integration will take place:



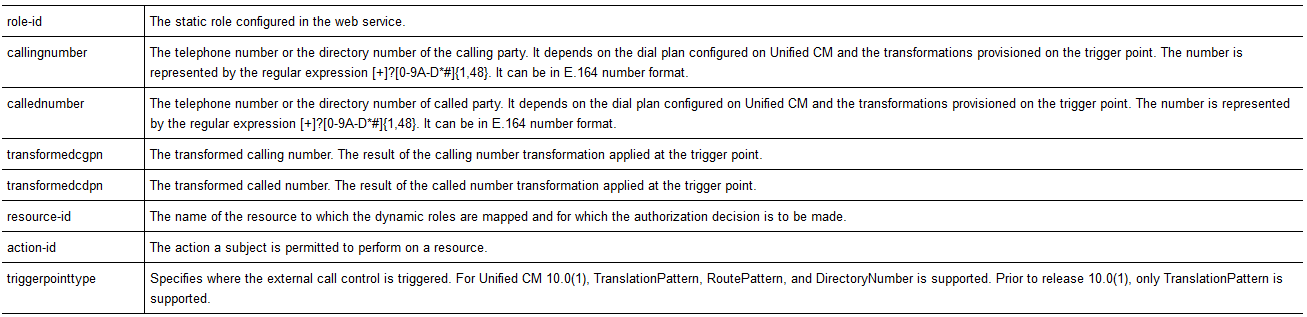
To build this integration External Call Control (ECC) system will be used. Configuration for ECC is kept on External Call Control Profile. – which allows to set:

* Primary/Backup server for external call control
* Load balancing policy
* What to do if external call control is not responding policy

ECC Profile can be assigned to multiple CUCM configuration objects. In case of the Emergency calls the best idea is to assign it to Route Patterns that route calls for Emergency Numbers. To ensure that every emergency call goes out from CUCM, allow all policy will be used in case when external call control is not responding.

ECC communication protocol is based upon XML. The full documentation of the protocol specs can be found here: <https://developer.cisco.com/site/curri/develop-and-test/documentation/latest-version/>

If an emergency route pattern is triggered, ECC makes a HTTP/HTTPS call to external call control server. In the request the following parameters are provided:



External Call Control Server gathers the data and creates a new notification task. Meantime it responds back to CUCM to allow call.

Created notification task is put in the notification queue maintained by Notifier service. Because task message doesn’t contain too much information – only the calling number – if needed Notifier Service can pull additional information (Description, Phone SEP number) can be gathered via AXL API call to CUCM.

Based on the configuration notifier service will send notification using following communication channels:

1. Email
2. SMS (managed on Twilio)
3. Call – messaged played by IVR once call is connected (managed on Twilio)

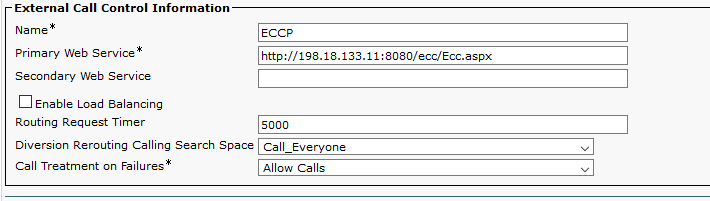
Each notification task and notification status will be written to internal system database for audit purposes.

# CUCM External Call Control Feature - Configuration

To enable External Call Control Feature on CUCM side user needs to add:

1. External Call Control Profile (ECCP) that contains information about external call control server
2. Assign ECCP to route patterns, translation patterns, etc. to trigger ECC feature.

Here is a picture taken from sample ECCP configuration:



**IMPORTANT!!!**

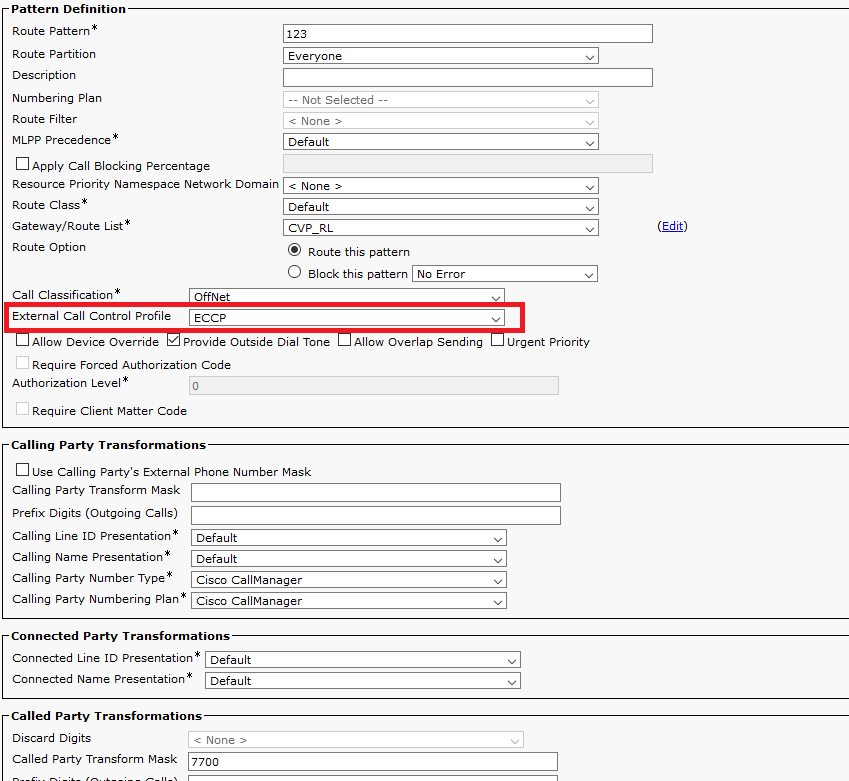
The Primary Web Service and (optional) Secondary Web Service must be configured in the following format:

**<protocol>://<ip address/host name>:<port>/<directory>**

During the test we missed the <port> and that caused failure. Here is a log snippet:

00354692.002 |02:52:34.481 |AppInfo |HttpHandler(1,39,4): -setHostPortPath <http://198.18.133.11/ecc/Ecc.aspx>  
00354692.003 |02:52:34.481 |AppInfo |HttpHandler(1,39,4): -setHostPortPath token http  
00354692.004 |02:52:34.481 |AppInfo |HttpHandler(1,39,4): -setHostPortPath token http  
00354692.005 |02:52:34.481 |AppInfo |HttpHandler(1,39,4): -setHostPortPath token [198.18.133.11](http://198.18.133.11)  
00354692.006 |02:52:34.481 |AppInfo |HttpHandler(1,39,4): -setHostPortPath token ecc  
00354692.007 |02:52:34.481 |AppInfo |HttpHandler(1,39,4): -setHostPortPath token Ecc.aspx  
00354692.008 |02:52:34.481 |AppInfo |HttpHandler(1,39,4): -setHostPortPath hScheme http, and hHost [198.18.133.11](http://198.18.133.11) , and value size 4  
***00354692.009 |02:52:34.481 |AppInfo |HttpHandler(1,39,4): -setHostPortPath PORT IS MISSING***  
00354692.010 |02:52:34.481 |AppInfo |ConnectionFailureToPDP - A connection request from Unified CM to the policy decision point failed Policy Decision Point:<http://198.18.133.11/ecc/Ecc.aspx> The cause of the connection failure:Invalid URI is specified in the External Call Control profile App ID:Cisco CallManager Cluster ID:StandAloneCluster Node ID:cucm1  
00354692.011 |02:52:34.481 |AlarmErr |AlarmClass: CallManager, AlarmName: ConnectionFailureToPDP, AlarmSeverity: Error, AlarmMessage: , AlarmDescription: A connection request from Unified CM to the policy decision point failed, AlarmParameters: PolicyDecisionPoint:<http://198.18.133.11/ecc/Ecc.aspx>, FailedToConnectReason:Invalid URI is specified in the External Call Control profile, AppID:Cisco CallManager, ClusterID:StandAloneCluster, NodeID:cucm1,

Here is a picture of a sample Route Pattern configuration. In red rectangle, place where to configure the ECCP profile:



The route pattern is 123 and it transforms called party number to 7700.

## ECC Requests & Responses

Based on the above configuration CUCM sends the following message to external call control server:

<?xml version="1.0" encoding="UTF-8"?>

<Request xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os">

<Subject SubjectCategory="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject">

<Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:subject:role-id"

DataType="http://www.w3.org/2001/XMLSchema#string" Issuer="requestor">

<AttributeValue>CISCO:UC:UCMPolicy</AttributeValue>

</Attribute>

<Attribute AttributeId="urn:Cisco:uc:1.0:callingnumber"

DataType="http://www.w3.org/2001/XMLSchema#string">

<AttributeValue>1080</AttributeValue>

</Attribute>

<Attribute AttributeId="urn:Cisco:uc:1.0:callednumber"

DataType="http://www.w3.org/2001/XMLSchema#string">

<AttributeValue>123</AttributeValue>

</Attribute>

<Attribute AttributeId="urn:Cisco:uc:1.0:transformedcgpn"

DataType="http://www.w3.org/2001/XMLSchema#string">

<AttributeValue>1080</AttributeValue>

</Attribute>

<Attribute AttributeId="urn:Cisco:uc:1.0:transformedcdpn"

DataType="http://www.w3.org/2001/XMLSchema#string">

<AttributeValue>7700</AttributeValue>

</Attribute>

</Subject>

<Resource>

<Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"

DataType="http://www.w3.org/2001/XMLSchema#anyURI">

<AttributeValue>CISCO:UC:VoiceOrVideoCall</AttributeValue>

</Attribute>

</Resource>

<Action>

<Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"

DataType="http://www.w3.org/2001/XMLSchema#anyURI">

<AttributeValue>any</AttributeValue>

</Attribute>

</Action>

<Environment>

<Attribute AttributeId="urn:Cisco:uc:1.0:triggerpointtype"

DataType="http://www.w3.org/2001/XMLSchema#string">

<AttributeValue>routepattern</AttributeValue>

</Attribute>

</Environment>

</Request>

To allow call routing external call control server must respond in the following format:

<?xml encoding="UTF-8" version="1.0"?>

<Response>

<Result>

<Decision>Permit</Decision>

<Obligations>

<Obligation FulfillOn="Permit" obligationId="continue.simple">

<AttributeAssignment AttributeId="Policy:continue.simple">

<AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">

&lt;cixml version="1.0&gt;

&lt;continue&gt;&lt;/continue&gt;

&lt;/cixml&gt;

</AttributeValue>

</AttributeAssignment>

</Obligation>

</Obligations>

</Result>

</Response>

# Troubleshooting

External Call Control Profile is a part of the Call Manager service. Because of this all messages generated by this feature can be found in Call Manager SDL service logs.

To see what’s happening in the system and check if everything works properly you will need to filter the gathered logs using the direct match (case sensitive) using following word: **Http**.

The logs should contain entries from 3 services:

* **HttpHandler** – process responsible for detecting ECCP configuration changes and validation of the data entered
* **HttpManager** – process that manages connections
* **HttpConnection** – process responsible for maintaining the connection, used when to call external control server

Cisco doesn’t provide a good documentation for the ECCP feature but based on the logs we were able to create the diagram that shows what CUCM does once the ECCP is created. The diagram was created based on the collected logs and may not exactly match the logic implemented in CUCM.

Obraz zawierający tekst

Opis wygenerowany automatycznie

## SSL Handshake error

SSL Handshake is a problem that is hard to be discovered. When you configure ECCP profile the CUCM acts as a client that tries to connect to external server. When the you provide the secure link (HTTPS) you will need to build the trust relation between CUCM and the external control server.

Th main problem with this error is a fact that CUCM don’t show this error. The only thig that is visible is the symptom of not established connection that can be identified by Connection retries. Below are the sample messages that you can find in the logs – search for application HttpConnection:

**17840553.000 |18:01:04.615 |AppInfo |HttpConnection - processRetry count=25872, threadId=907/-265147536 ACTIVE/REQ URL**[**https://gaman.local:9092/Popup/InboundCall**](https://gaman.local:9092/Popup/InboundCall)

**17840999.000 |18:02:06.456 |AppInfo |HttpConnection - processRetry count=25872, threadId=907/-265147536 ACTIVE/REQ URL**[**https://gaman.local:9092/Popup/InboundCall**](https://gaman.local:9092/Popup/InboundCall)

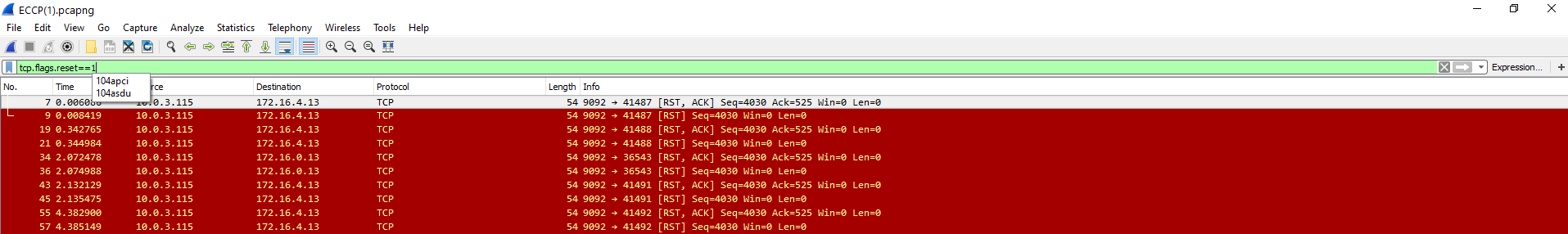
Increasing the logging level doesn’t help, no additional messages will appear. So, the question is how to troubleshoot this error?

If no messages can be found, then you will need to switch to the server side and try to find what’s happening on the network side (before application is triggered).

In our opinion the best option is to use Wireshark and collect packet dumps on the network interface that is used for communication. Once the traffic is captured the easiest way to check if the SSL handshake error appears is to check if the connection goes through the reset state. To pull the requested data, you can use the following filter:

**tcp.flags.reset==1**

When the filter is applied you should get the view like visible on the screenshot below.



On the list, search for the IP addresses that correspond to client (CUCM) and server address. If you find that connection is reset in close time periods, then the problem needs to be deeper analysed. We recommend applying filter that will only point to the communication port that is used for communication. In the test solution we used port 9092 so we used the following filter:

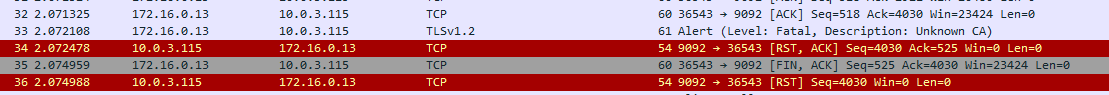
**tcp.port==9092**

The filtered results will appear on the screen. First thing to check is a start of the secure communication. This part is visible on the below screen shot:

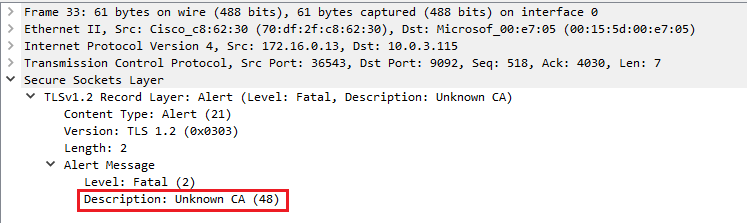


SSL communication starts with the **Client Hello** message. Please remember that the protocol visible on view must point to any secure protocol – in the screenshot – TLSv1.2.

Starting from the starting point, search for the first red message that shows connection reset. The message with reason code should be before the red message.



When you click and expand the packet details you will see the error. In the following example the message is: **Unknown CA (48)**.



The problem has been identified – the CUCM doesn’t trust the ECCP Server. So, what to do next? To solve the problem, you will need to build a trust relationship between servers – you will need upload ECCP server certificates to CUCM. The procedure is described well in the following document, section **Import the Route Server Certificate into the Trusted Store**: <https://www.cisco.com/c/en/us/td/docs/voice_ip_comm/cucm/admin/12_0_1/systemConfig/cucm_b_system-configuration-guide-1201/cucm_b_system-configuration-guide-1201_chapter_01001100.html#CUCM_TK_I3A564C4_00>

**Please note!**

You will need to import entire certificate chain (Root CA, Intermittent) to the trust store.

# Documentation

Below is a list of documents that was used to prepare this article:

* CURRI developer documentation:

<https://developer.cisco.com/site/curri/develop-and-test/documentation/latest-version/>

# Contact Information

**Email addresses**

Sales – Andrzej Skrzeczkowski: **(**[**andrzej.skrzeczkowski@gaman-gt.com**](mailto:andrzej.skrzeczkowski@gaman-gt.com)**)**

General, Sales mailbox: **(**[**info@gaman-gt.com**](mailto:info@gaman-gt.com)**)**

Support mailbox: ([**support@gaman-gt.com**](mailto:support@gaman-gt.com)**)**

To find out more about Gaman Team and our project please visit our **Website:** [**https://gaman-gt.com**](https://gaman-gt.com)

Find us in social networks:

|  |  |  |
| --- | --- | --- |
|  |  |  |
| **Linked In** | **Twitter** | **Upwork** |