# Secured MGCP Communication between Voice GW and CUCM via IPsec Based on CA Signed Certificates Configuration Example

TAC **Document ID: 118886**

Contributed by Mateusz Korab, Cisco TAC Engineer.
Apr 07, 2015

## Contents



## Introduction

This document describes how to successfully secure Media Gateway Control Protocol (MGCP) signalling between a voice gateway (GW) and CUCM (Cisco Unified Communications Manager) via Internet Protocol Security (IPsec), based on Certificate Authority (CA) signed certificates. In order to set up a secured call via MGCP, signalling and Real−time Transport Protocol (RTP) streams need to be secured separately. It seems to be well documented and quite simple to set up encrypted RTP streams, but a secure RTP stream does not include secure MGCP signalling. If the MGCP signalling is not secured, the encryption keys for the RTP stream are sent in the clear.

## Prerequisites

### Requirements

Cisco recommends that you have knowledge of these topics:

- MGCP voice gateway registered to CUCM in order to send and receive calls
- Certificate Authority Proxy Function (CAPF) service started, cluster set to mixed−mode
- Cisco IOS® image on GW supports crypto security feature
- Phones and MGCP GW configured for Secure Real−time Transport Protocol (SRTP)

## Components Used

The information in this document is based on these software and hardware versions:

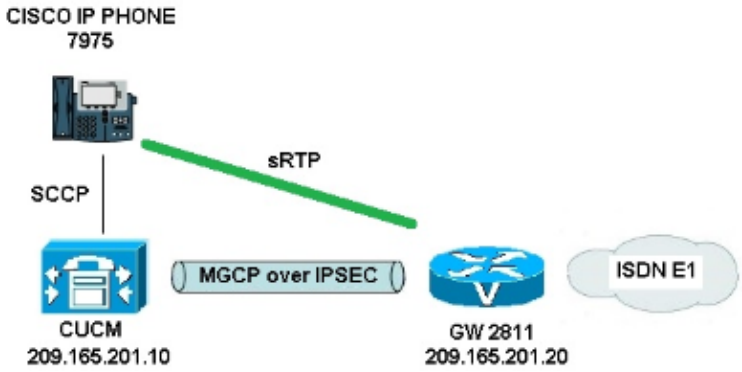
- CUCM − single node − runs GGSG (Cisco's Global Government Solutions Group) version 8.6.1.20012−14 in Federal Information Processing Standard (FIPS) mode
- 7975 phones which run SCCP75−9−3−1SR2−1S
- GW − Cisco 2811 − C2800NM−ADVENTERPRISEK9−M, Version 15.1(4)M8
- E1 ISDN voice card − VWIC2−2MFT−T1/E1 − 2−Port RJ−48 Multiflex Trunk

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, make sure that you understand the potential impact of any command.

# Configure

***Note***: Use the Command Lookup Tool (registered customers only) in order to obtain more information on the commands used in this section.

## Network Diagram




In order to successfully set up IPsec between CUCM and voice GW, complete these steps:

1. Configure the CA on the voice GW and generate a CA−signed certificate for voice GW
2. Generate a CUCM CA−signed IPsec certificate
3. Import CA, CUCM, and voice GW CA certificates on CUCM
4. Configure IPsec tunnel settings on CUCM
5. Configure the IPsec tunnel setting on the voice GW

## 1. Configure the CA on the Voice GW and Generate a CA−signed Certificate for Voice GW

As a first step, the Rivest−Shamir−Addleman (RSA) key pair needs to be generated on the voice GW (Cisco IOS CA server):

```
KRK−UC−2x2811−2#crypto key generate rsa general−keys label IOS_CA exportable
```

Enrollments completed via Simple Certificate Enrollment Protocol (SCEP) will be used, so enable the HTTP server:

```
KRK−UC−2x2811−2#ip http server
```

In order to configure the CA Server on a gateway, these steps need to be completed:

a. Set the PKI server name. It needs to be the same name as the key pair generated previously.

```
KRK-UC-2x2811-2(config)#crypto pki server IOS_CA
```

b. Specify the location where all database entries will be stored for the CA server.

```
KRK-UC-2x2811-2(cs-server)#crypto pki server IOS_CA
```

c. Configure the CA issuer name.

```
KRK-UC-2x2811-2(cs-server)#issuer-name cn=IOS
```

d. Specify a certificate revocation list (CRL) distribution point (CDP) to be used in certificates that are issued by the certificate server and enable automatic granting of certificate reenrollment requests for a Cisco IOS subordinate CA server.

```
KRK-UC-2x2811-2(cs-server)#cdp-url http://209.165.201.10/IOS_CA.crl
KRK-UC-2x2811-2(cs-server)#grant auto
```

e. Enable the CA server.

```
KRK-UC-2x2811-2(cs-server)#no shutdown
```

The next step is to create a trustpoint for the CA certificate and a local trustpoint for the router certificate with a URL enrollment that points to a local HTTP server:

```
KRK-UC-2x2811-2(config)#crypto pki trustpoint IOS_CA
KRK-UC-2x2811-2(ca-trustpoint)#revocation-check crl
KRK-UC-2x2811-2(ca-trustpoint)#rsakeypair IOS_CA

KRK-UC-2x2811-2(config)#crypto pki trustpoint local1
KRK-UC-2x2811-2(ca-trustpoint)#enrollment url http://209.165.201.10:80
KRK-UC-2x2811-2(ca-trustpoint)#serial-number none
KRK-UC-2x2811-2(ca-trustpoint)#fqdn none
KRK-UC-2x2811-2(ca-trustpoint)#ip-address none
KRK-UC-2x2811-2(ca-trustpoint)#subject-name cn=KRK-UC-2x2811-2
KRK-UC-2x2811-2(ca-trustpoint)#revocation-check none
```

In order to generate the router's certificate signed by the local CA, the trustpoint needs to be authenticated and enrolled:

```
KRK-UC-2x2811-2(config)#crypto pki authenticate local1
KRK-UC-2x2811-2(config)#crypto pki enroll local1
```

After that, the router's certificate is generated and signed by the local CA. List the certificate on the router for verification.

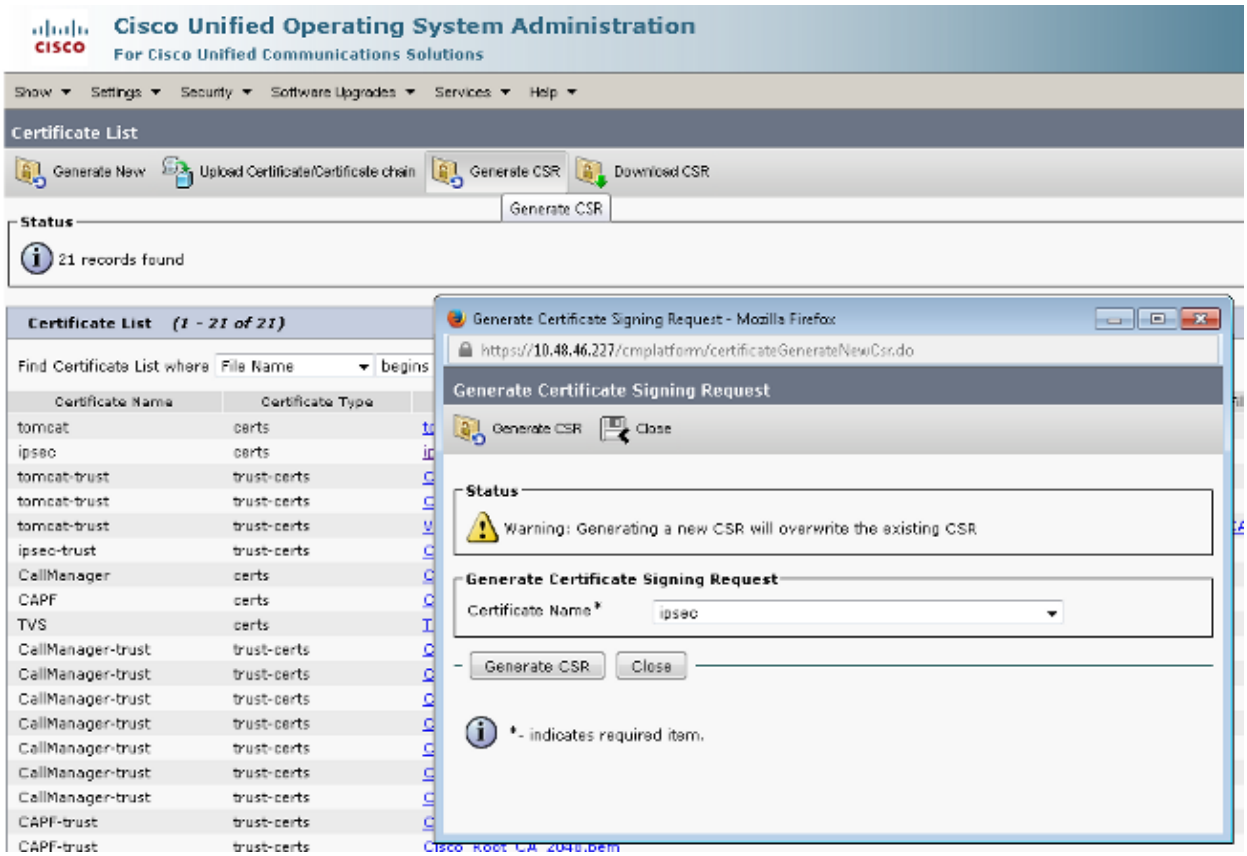
```
KRK-UC-2x2811-2#show crypto ca certificates
Certificate
  Status: Available
  Certificate Serial Number (hex): 02
  Certificate Usage: General Purpose
  Issuer:
    cn=IOS
  Subject:
    Name: KRK-UC-2x2811-2
    cn=KRK-UC-2x2811-2
  CRL Distribution Points:
    http://10.48.46.251/IOS_CA.crl
  Validity Date:
    start date: 13:05:01 CET Nov 21 2014
    end   date: 13:05:01 CET Nov 21 2015
```

```
  Associated Trustpoints: local1
  Storage: nvram:IOS#2.cer

CA Certificate
  Status: Available
  Certificate Serial Number (hex): 01
  Certificate Usage: Signature
  Issuer:
    cn=IOS
  Subject:
    cn=IOS
  Validity Date:
    start date: 12:51:12 CET Nov 21 2014
    end   date: 12:51:12 CET Nov 20 2017
  Associated Trustpoints: local1 IOS_CA
  Storage: nvram:IOS#1CA.cer
```

Two certificates should be listed. The first one is a router's (KRK−UC−2x2811−2) certificate signed by the local CA and the second one is CA certificate.

## 2. Generate a CUCM CA−signed IPsec Certificate

The CUCM for IPsec tunnel set up uses a ipsec.pem certificate. By default, this certificate is self−signed and generated when the system is installed. In order to replace it with a CA−signed certificate, first a CSR (Certificate Sign Request) for IPsec from the CUCM OS Admin Page needs to be generated. Choose ***Cisco Unified OS Administration > Security > Certificate Management > Generate CSR***.




After the CSR is generated, it needs to be downloaded from CUCM and enrolled against the CA on the GW. In order to do that, enter the ***crypto pki server IOS_CA request pkcs10 terminal base64*** command and the sign request hash needs to be pasted via terminal. The granted certificate is displayed and needs to be copied and saved as the ipsec.pem file.

```
KRK-UC-2x2811-2#crypto pki server IOS_CA request pkcs10 terminal base64
PKCS10 request in base64 or pem

% Enter Base64 encoded or PEM formatted PKCS10 enrollment request.
% End with a blank line or "quit" on a line by itself.
-----BEGIN CERTIFICATE REQUEST-----
MIIDNjCCAh4CAQAwgakxCzAJBgNVBAYTAlBMMQ4wDAYDVQQIEwVjaXNjbzEOMAwG
A1UEBxMFY2lzY28xDjAMBgNVBAoTBWNpc2NvMQ4wDAYDVQQLEwVjaXNjbzEPMA0G
A1UEAxMGQ1VDTUIxMUkwRwYDVQQFE0A1NjY2OWY5MjgzNWZmZWQ1MDg0YjI5MTU4
NjcwMDBmMGI2NjliYjdkYWZhNDNmM2QzOWFhNGQxMzM1ZTllMjUzMIIBIjANBgkq
hkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAkfHxvcov4vFmK+3+dQShW3s3SzAYBQ19
0JDBiIc4eDRmdrq0V2dkn9UpLUx9OH7V0Oe/8wmHqYwoxFZ5a6B5qRRkcO10/ub2
ul1QCw+nQ6QiZGdNhdne0NYY4r3odF4CkrtYAJA4PUSce1tWxfiJY5dw/Xhv8cVg
gVyuxctESemfMhUfvEM203NU9nod7YTEzQzuAadjNcyc4b1u91vQm5OVUNXxODov
e7/OlQNUWU3LSEr0aI9lC75x3qdRGBe8Pwnk/gWbT5B7pwuwMXTU8+UFj6+lvrQM
Rb47dw22yFmSMObvez18IVExAyFs5Oj9Aj/rNFIdUQIt+Nt+Q+f38wIDAQABoEcw
RQYJKoZIhvcNAQkOMTgwNjAnBgNVHSUEIDAeBggrBgEFBQcDAQYIKwYBBQUHAwIG
CCsGAQUFBwMFMAsGA1UdDwQEAwIDuDANBgkqhkiG9w0BAQUFAAOCAQEAQDgAR4Ol
oQ4z2yqgSsICAZ2hQA3Vztp6aOI+0PSyMfihGS//3V3tALEZL2+t0Y5elKsBea72
sieKjpSikXjNaj+SiY1aYy4siVw5EKQD3Ii4Qvl15BvuniZXvBiBQuW+SpBLbeNi
xwIgrYELrFywQZBeZOdFqnSKN9XlisXe6oU9GXux7uwgXwkCXMF/azutbiol4Fgf
qUF00GzkhtEapJA6c5RzaxG/0uDuKY+4z1eSSsXzFhBTifk3RfJA+I7Na1zQBIEJ
2IOJdiZnn0HWVr5C5eZ7VnQuNdiC/qn3uUfvNVRZo8iCDq3tRv7dr/n64jdKsHEM
lk6P8gp9993cJw==
quit
% Granted certificate:
MIIDXTCCAsagAwIBAgIBBTANBgkqhkiG9w0BAQQFADAOMQwwCgYDVQQDEwNJT1Mw
HhcNMTUwMTA4MTIwMTAwWhcNMTYwMTA4MTIwMTAwWjCBqTELMAkGA1UEBhMCUEwx
DjAMBgNVBAgTBWNpc2NvMQ4wDAYDVQQHEwVjaXNjbzEOMAwGA1UEChMFY2lzY28x
DjAMBgNVBAsTBWNpc2NvMQ8wDQYDVQQDEwZDVUNNQjExSTBHBgNVBAUTQDU2NjY5
ZjkyODM1ZmZlZDUwODRiMjkxNTg2NzAwMGYwYjY2OWJiN2RhZmE0M2YzZDM5YWE0
ZDEzMzVlOWUyNTMwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQCR8fG9
yi/i8WYr7f51BKFbezdLMBgFDX3QkMGIhzh4NGZ2urRXZ2Sf1SktTH04ftXQ57/z
CYepjCjEVnlroHmpFGRw7XT+5va6XVALD6dDpCJkZ02F2d7Q1hjiveh0XgKSu1gA
kDg9RJx7W1bF+Iljl3D9eG/xxWCBXK7Fy0RJ6Z8yFR+8QzbTc1T2eh3thMTNDO4B
p2M1zJzhvW73W9Cbk5VQ1fE4Oi97v86VA1RZTctISvRoj2ULvnHep1EYF7w/CeT+
BZtPkHunC7AxdNTz5QWPr6W+tAxFvjt3DbbIWZIw5u97PXwhUTEDIWzk6P0CP+s0
Uh1RAi34235D5/fzAgMBAAGjgaowgacwLwYDVR0fBCgwJjAkoCKgIIYeaHR0cDov
LzEwLjQ4LjQ2LjI1MS9JT1NfQ0EuY3JsMAsGA1UdDwQEAwIDuDAnBgNVHSUEIDAe
BggrBgEFBQcDAQYIKwYBBQUHAwIGCCsGAQUFBwMFMB8GA1UdIwQYMBaAFJSLP5cn
PL8bIP7VSKLtB6Z1socOMB0GA1UdDgQWBBR4m2eTSyELsdRBW4MRmbNdT2qppTAN
BgkqhkiG9w0BAQQFAAOBgQBuVJ+tVS0JqP4z9TgEeuMbVwn00CTKXz/fCuh6R/50
qq8JhERJGiR/ZHvHRLf+XawhnoE6daPAmE+WkIPtHIIhbMHCbbxG9ffdyaiNXRWy
5sI5XycF1FgYGpTFBYD9M0Lqsw+FIYaT2ZrbOGsx8h6pZoesKqm85RByIUjX4nJK
1g==
```

***Note***: In order to decode and check the content of the Base64 encoded certificate, enter the ***openssl x509 -in certificate.crt -text -noout*** command.

The granted CUCM certificate decodes to:

```
Certificate:
    Data&colon;
        Version: 3 (0x2)
        Serial Number: 5 (0x5)
        Signature Algorithm: md5WithRSAEncryption
        Issuer: CN=IOS
        Validity
            Not Before: Jan  8 12:01:00 2015 GMT
            Not After : Jan  8 12:01:00 2016 GMT
        Subject: C=PL, ST=cisco, L=cisco, O=cisco, OU=cisco,
 CN=CUCMB1/serialNumber=56669f92835ffed5084b2915867000f0b669bb7dafa43f3d39aa4d1335e9e253
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
            RSA Public Key: (2048 bit)
```

```
                Modulus (2048 bit):
                    00:91:f1:f1:bd:ca:2f:e2:f1:66:2b:ed:fe:75:04:
                    a1:5b:7b:37:4b:30:18:05:0d:7d:d0:90:c1:88:87:
                    38:78:34:66:76:ba:b4:57:67:64:9f:d5:29:2d:4c:
                    7d:38:7e:d5:d0:e7:bf:f3:09:87:a9:8c:28:c4:56:
                    79:6b:a0:79:a9:14:64:70:ed:74:fe:e6:f6:ba:5d:
                    50:0b:0f:a7:43:a4:22:64:67:4d:85:d9:de:d0:d6:
                    18:e2:bd:e8:74:5e:02:92:bb:58:00:90:38:3d:44:
                    9c:7b:5b:56:c5:f8:89:63:97:70:fd:78:6f:f1:c5:
                    60:81:5c:ae:c5:cb:44:49:e9:9f:32:15:1f:bc:43:
                    36:d3:73:54:f6:7a:1d:ed:84:c4:cd:0c:ee:01:a7:
                    63:35:cc:9c:e1:bd:6e:f7:5b:d0:9b:93:95:50:d5:
                    f1:38:3a:2f:7b:bf:ce:95:03:54:59:4d:cb:48:4a:
                    f4:68:8f:65:0b:be:71:de:a7:51:18:17:bc:3f:09:
                    e4:fe:05:9b:4f:90:7b:a7:0b:b0:31:74:d4:f3:e5:
                    05:8f:af:a5:be:b4:0c:45:be:3b:77:0d:b6:c8:59:
                    92:30:e6:ef:7b:3d:7c:21:51:31:03:21:6c:e4:e8:
                    fd:02:3f:eb:34:52:1d:51:02:2d:f8:db:7e:43:e7:
                    f7:f3
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 CRL Distribution Points:
                URI:http://10.48.46.251/IOS_CA.crl

            X509v3 Key Usage:
                Digital Signature, Key Encipherment, Data Encipherment, Key Agreement
            X509v3 Extended Key Usage:
                TLS Web Server Authentication, TLS Web Client Authentication,
 IPSec End System
            X509v3 Authority Key Identifier:
                keyid:94:8B:3F:97:27:3C:BF:1B:20:FE:D5:48:A2:ED:07:A6:75:B2:87:0E

            X509v3 Subject Key Identifier:
                78:9B:67:93:4B:21:0B:B1:D4:41:5B:83:11:99:B3:5D:4F:6A:A9:A5
    Signature Algorithm: md5WithRSAEncryption
        6e:54:9f:ad:55:2d:09:a8:fe:33:f5:38:04:7a:e3:1b:57:09:
        f4:d0:24:ca:5f:3f:df:0a:e8:7a:47:fe:74:aa:af:09:84:44:
        49:1a:24:7f:64:7b:c7:44:b7:fe:5d:ac:21:9e:81:3a:75:a3:
        c0:98:4f:96:90:83:ed:1c:82:21:6c:c1:c2:6d:bc:46:f5:f7:
        dd:c9:a8:8d:5d:15:b2:e6:c2:39:5f:27:05:d4:58:18:1a:94:
        c5:05:80:fd:33:42:ea:b3:0f:85:21:86:93:d9:9a:db:38:6b:
        31:f2:1e:a9:66:87:ac:2a:a9:bc:e5:10:72:21:48:d7:e2:72:
        4a:d6
```

## 3. Import CA, CUCM, and Voice GW CA Certificates on CUCM

The CUCM IPsec certificate is already exported to a .pem file. As a next step, the same process needs to be completed with the voice GW certificate and the CA certificate. In order to do that, they need to be first displayed on a terminal with the ***crypto pki export local1 pem terminal*** command and copied to separate .pem files.

```
KRK-UC-2x2811-2(config)#crypto pki export local1 pem terminal
% CA certificate:
-----BEGIN CERTIFICATE-----
MIIB9TCCAV6gAwIBAgIBATANBgkqhkiG9w0BAQQFADAOMQwwCgYDVQQDEwNJT1Mw
HhcNMTQxMTIxMTE1MTEyWhcNMTcxMTIwMTE1MTEyWjAOMQwwCgYDVQQDEwNJT1Mw
gZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAK6Cd2yxUywtbgBElkZUsP6eaZVv
6YfpEbFptyt6ptRdpxgjOYI3InEP3wewtmEPNeTJL8+a/W7MDUemm3t/NlWBO6T2
m9Bp6k0FNOBXMKeDfTSqOKEy7WfLASe/Pbq8M+JMpeMWz8xnMboYOb66rY8igZFz
k1tRPlIMSf5rO1tnAgMBAAGjYzBhMA8GA1UdEwEB/wQFMAMBAf8wDgYDVR0PAQH/
BAQDAgGGMB8GA1UdIwQYMBaAFJSLP5cnPL8bIP7VSKLtB6Z1socOMB0GA1UdDgQW
BBSUiz+XJzy/GyD+1Uii7QemdbKHDjANBgkqhkiG9w0BAQQFAAOBgQCUMC1SFVlS
TSS1ExbM9i2D4HOWYhCurhifqTWLxMMXj0jym24DoqZ91aDNG1VwiJ/Yv4i40t90
```

```
y65WzbapZL1S65q+d7BCLQypdrwcKkdS0dfTdKfXEsyWLhecRa8mnZckpgKBk8Ir
BfM9K+caXkfhPEPa644UzV9++OKMKhtDuQ==
-----END CERTIFICATE-----

% General Purpose Certificate:
-----BEGIN CERTIFICATE-----
MIIB2zCCAUSgAwIBAgIBAjANBgkqhkiG9w0BAQUFADAOMQwwCgYDVQQDEwNJT1Mw
HhcNMTQxMTIxMTIwNTAxWhcNMTUxMTIxMTIwNTAxWjAaMRgwFgYDVQQDEw9LUkst
VUMtMngyODExLTIwXDANBgkqhkiG9w0BAQEFAANLADBIAkEApGWIN1nAAtKLVMOj
mZVkQFgI8LrHD6zSrlaKgAJhlU+H/mnRQQ5rqitIpekDdPoowST9RxC5CJmB4spT
VWkYkwIDAQABo4GAMH4wLwYDVR0fBCgwJjAkoCKgIIYeaHR0cDovLzEwLjQ4LjQ2
LjI1MS9JT1NfQ0EuY3JsMAsGA1UdDwQEAwIFoDAfBgNVHSMEGDAWgBSUiz+XJzy/
GyD+1Uii7QemdbKHDjAdBgNVHQ4EFgQUtAWc61K5nYGgWqKAiIOLMlphfqIwDQYJ
KoZIhvcNAQEFBQADgYEAjDflH+N3yc3RykCig9B0aAIXWZPmaqLF9v9R75zc+f8x
zbSIzoVbBhnUOeuOj1hnIgHyyMjeELjTEh6uQrWUN2ElW1ypfmxk1jN5q0t+vfdR
+yepS04pFor9RoD7IWg6e/1hFDEep9hBvzrVwQHCjzeY0rVrPcLl26k5oauMwTs=
-----END CERTIFICATE-----
```

The % CA certificate decodes to:

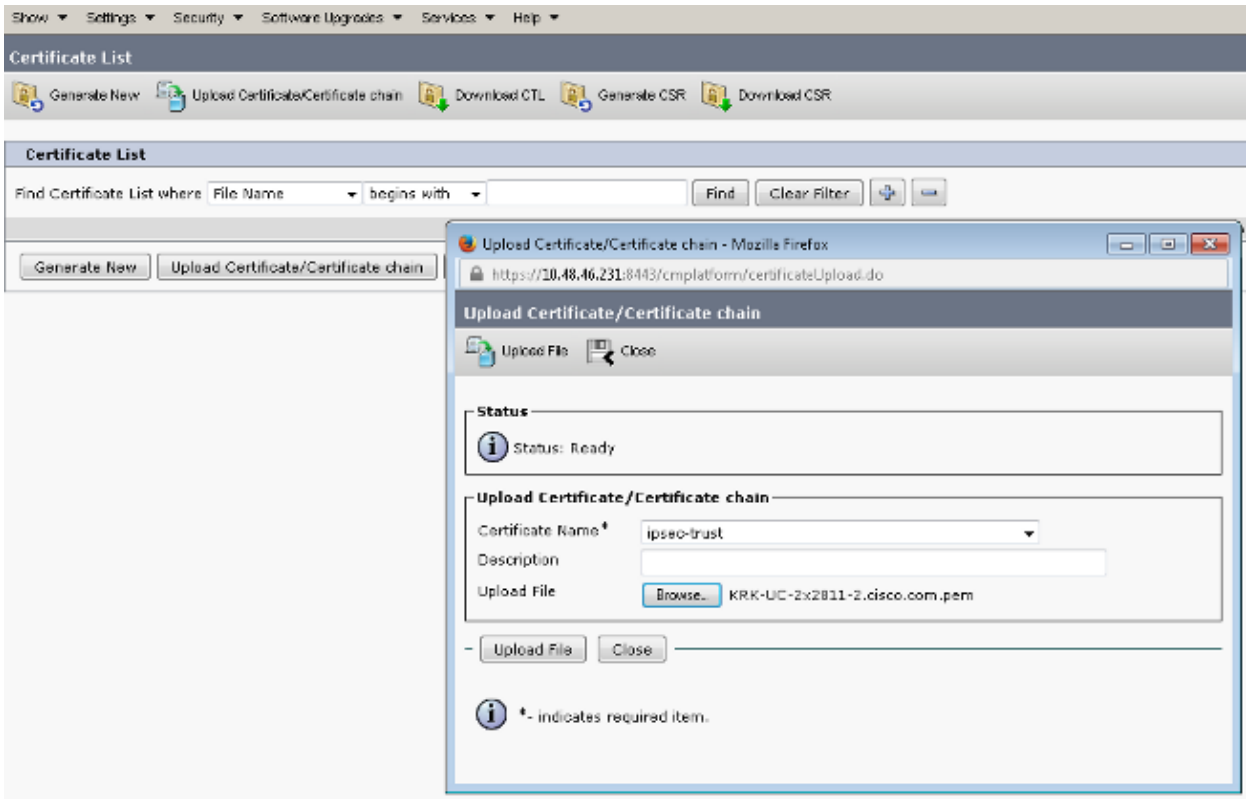
```
Certificate:
    Data&colon;
        Version: 3 (0x2)
        Serial Number: 1 (0x1)
        Signature Algorithm: md5WithRSAEncryption
        Issuer: CN=IOS
        Validity
            Not Before: Nov 21 11:51:12 2014 GMT
            Not After : Nov 20 11:51:12 2017 GMT
        Subject: CN=IOS
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
            RSA Public Key: (1024 bit)
                Modulus (1024 bit):
                    00:ae:82:77:6c:b1:53:2c:2d:6e:00:44:96:46:54:
                    b0:fe:9e:69:95:6f:e9:87:e9:11:b1:69:b7:2b:7a:
                    a6:d4:5d:a7:18:23:39:82:37:22:71:0f:df:07:b0:
                    b6:61:0f:35:e4:c9:2f:cf:9a:fd:6e:cc:0d:47:a6:
                    9b:7b:7f:36:55:81:3b:a4:f6:9b:d0:69:ea:4d:05:
                    34:e0:57:30:a7:83:7d:34:aa:38:a1:32:ed:67:cb:
                    01:27:bf:3d:ba:bc:33:e2:4c:a5:e3:16:cf:cc:67:
                    31:ba:18:39:be:ba:ad:8f:22:81:91:73:93:5b:51:
                    3e:52:0c:49:fe:6b:3b:5b:67
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Basic Constraints: critical
                CA:TRUE
            X509v3 Key Usage: critical
                Digital Signature, Certificate Sign, CRL Sign
            X509v3 Authority Key Identifier:
                keyid:94:8B:3F:97:27:3C:BF:1B:20:FE:D5:48:A2:ED:07:A6:75:B2:87:0E

            X509v3 Subject Key Identifier:
                94:8B:3F:97:27:3C:BF:1B:20:FE:D5:48:A2:ED:07:A6:75:B2:87:0E
    Signature Algorithm: md5WithRSAEncryption
        94:30:2d:52:15:59:52:4d:24:b5:13:16:cc:f6:2d:83:e0:73:
        96:62:10:ae:ae:18:9f:a9:35:8b:c4:c3:17:8f:48:f2:9b:6e:
        03:a2:a6:7d:d5:a0:cd:1b:55:70:88:9f:d8:bf:88:b8:d2:df:
        74:cb:ae:56:cd:b6:a9:64:bd:52:eb:9a:be:77:b0:42:2d:0c:
        a9:76:bc:1c:2a:47:52:d1:d7:d3:74:a7:d7:12:cc:96:2e:17:
        9c:45:af:26:9d:97:24:a6:02:81:93:c2:2b:05:f3:3d:2b:e7:
        1a:5e:47:e1:3c:43:da:eb:8e:14:cd:5f:7e:f8:e2:8c:2a:1b:
        43:b9
```

The % General Purpose Certificate decodes to:

```
Certificate:
    Data&colon;
        Version: 3 (0x2)
        Serial Number: 2 (0x2)
        Signature Algorithm: sha1WithRSAEncryption
        Issuer: CN=IOS
        Validity
            Not Before: Nov 21 12:05:01 2014 GMT
            Not After : Nov 21 12:05:01 2015 GMT
        Subject: CN=KRK-UC-2x2811-2
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
            RSA Public Key: (512 bit)
                Modulus (512 bit):
                    00:a4:65:88:37:59:c0:02:d2:8b:54:c3:a3:99:95:
                    64:40:58:08:f0:ba:c7:0f:ac:d2:ae:56:8a:80:02:
                    61:95:4f:87:fe:69:d1:41:0e:6b:aa:2b:48:a5:e9:
                    03:74:fa:28:c1:24:fd:47:10:b9:08:99:81:e2:ca:
                    53:55:69:18:93
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 CRL Distribution Points:
                URI:http://10.48.46.251/IOS_CA.crl

            X509v3 Key Usage:
                Digital Signature, Key Encipherment
            X509v3 Authority Key Identifier:
                keyid:94:8B:3F:97:27:3C:BF:1B:20:FE:D5:48:A2:ED:07:A6:75:B2:87:0E

            X509v3 Subject Key Identifier:
                B4:05:9C:EB:52:B9:9D:81:A0:5A:A2:80:88:83:8B:32:5A:61:7E:A2
    Signature Algorithm: sha1WithRSAEncryption
        8c:37:e5:1f:e3:77:c9:cd:d1:ca:40:a2:83:d0:74:68:02:17:
        59:93:e6:6a:a2:c5:f6:ff:51:ef:9c:dc:f9:ff:31:cd:b4:88:
        ce:85:5b:06:19:d4:39:eb:8e:8f:58:67:22:01:f2:c8:c8:de:
        10:b8:d3:12:1e:ae:42:b5:94:37:61:25:5b:5c:a9:7e:6c:64:
        d6:33:79:ab:4b:7e:bd:f7:51:fb:27:a9:4b:4e:29:16:8a:fd:
        46:80:fb:21:68:3a:7b:fd:61:14:31:1e:a7:d8:41:bf:3a:d5:
        c1:01:c2:8f:37:98:d2:b5:6b:3d:c2:e5:db:a9:39:a1:ab:8c:
        c1:3b
```

After they are saved as .pem files, they need to be imported to CUCM. Choose ***Cisco Unified OS Administration*** > ***Security*** > ***Certificate management*** > ***Upload Certificate/Certificate***.

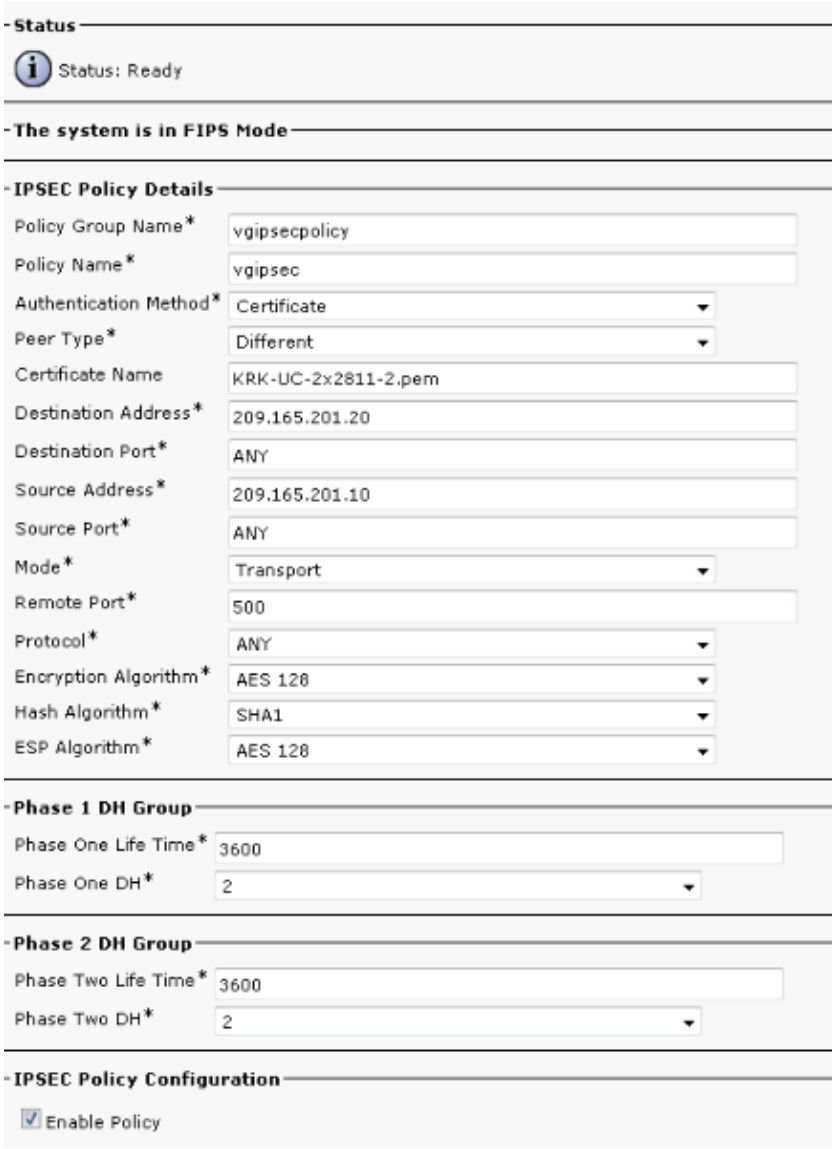
- CUCM certificate as IPsec
- Voice GW certificate as IPsec−trust
- CA certificate as IPsec−trust:




## 4. Configure IPsec Tunnel Settings on CUCM

The next step is configuration of the IPsec tunnel between CUCM and the voice GW. The IPsec tunnel configuration on CUCM is performed via the Cisco Unified OS Administration web page (https://<cucm_ip_address>/cmplatform). Choose ***Security > IPSEC Configuration > Add new IPsec policy***.

In this example, a policy called "vgipsecpolicy" was created, with authentication based on certificates. All appropriate information needs to be filled in and correspond to the configuration on the voice GW.




***Note***: The voice gateway certificate name needs to be specified in the Certificate Name field.

## 5. Configure the IPsec Tunnel Setting on the Voice GW

This example, with inline comments, presents the corresponding configuration on a voice GW.

```
crypto isakmp policy 1    (defines an IKE policy and enters the config-iskmp mode)
 encr aes                 (defines the encryption)
 group 2                  (defines 1024-bit Diffie-Hellman)
 lifetime 57600           (isakmp security association lifetime value)

crypto isakmp identity dn       (defines DN as the ISAKMP identity)
crypto isakmp keepalive 10      (enable sending dead peer detection (DPD)
keepalive messages to the peer)
crypto isakmp aggressive-mode disable  (to block all security association
                                       and ISAKMP aggressive mode requests)

crypto ipsec transform-set cm3 esp-aes esp-sha-hmac  (set of a combination of
                                                      security protocols
                                                      and algorithms that are
                                                      acceptable for use)
 mode transport
crypto ipsec df-bit clear
no crypto ipsec nat-transparency udp-encapsulation
```

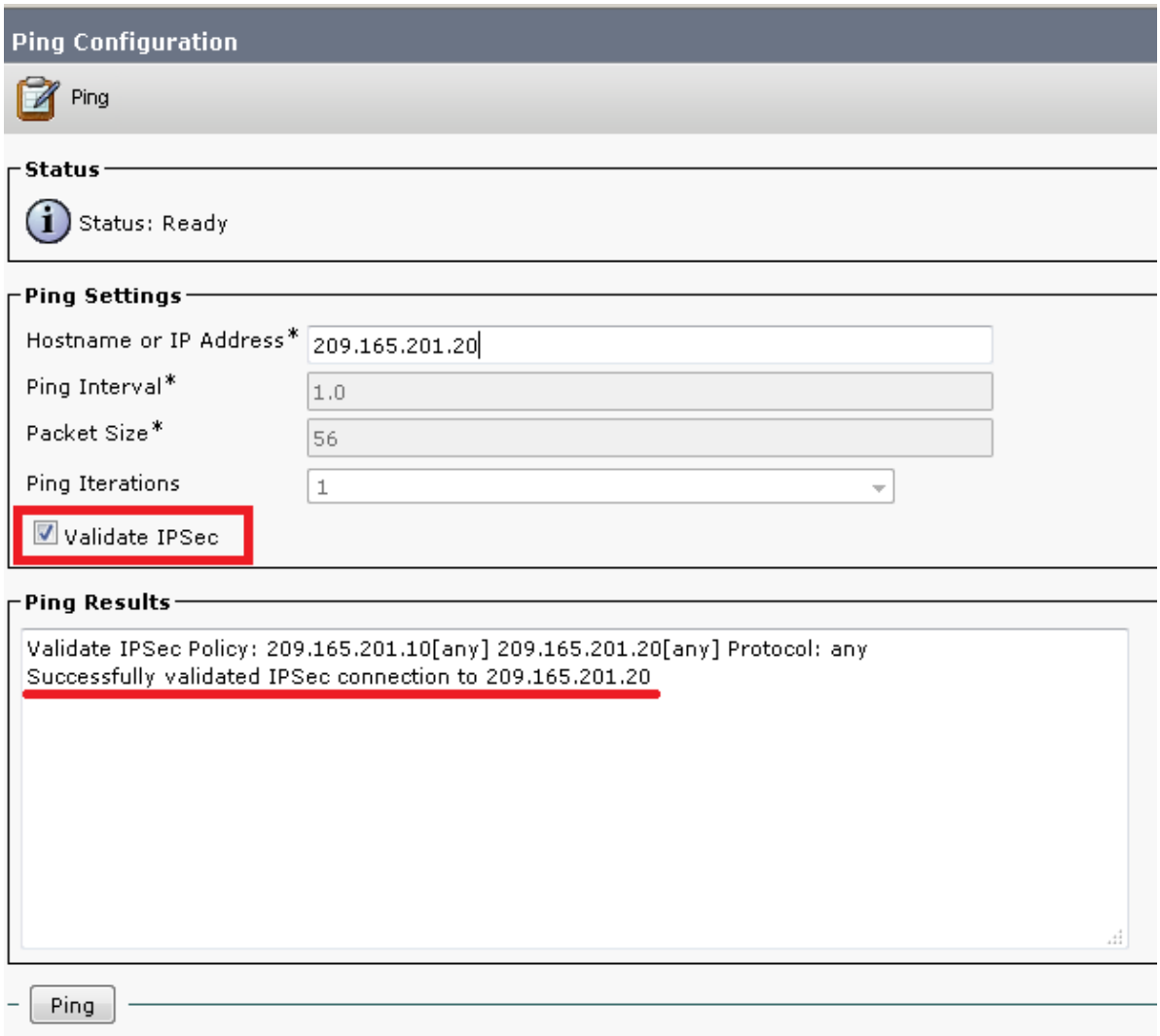
```
!
crypto map cm3 1 ipsec-isakmp     (selects data flows that need security
                                   processing, defines the policy for these flows
                                   and the crypto peer that traffic needs to go to)
 set peer 209.165.201.10
 set security-association lifetime seconds 28800
 set transform-set cm3
 match address 130

interface FastEthernet0/0
 ip address 209.165.201.20 255.255.255.224
 duplex auto
 speed auto
 crypto map cm3      (enables creypto map on the interface)

access-list 130 permit ip host 209.165.201.20 host 209.165.201.10
```

# Verify

Use this section in order to confirm that your configuration works properly.

## Verify IPsec Tunnel Status on the CUCM End

The fastest way to verify the IPsec tunnel status on CUCM is go to the OS Administration page and use the ***ping*** option under Services > Ping. Ensure the ***Validate IPSec*** check box is checked. Obviously, the IP address specified here is the IP address of the GW.




***Note***: See these Cisco bug IDs for information on the validation of the IPsec tunnel via the ping feature on CUCM:

− Cisco bug ID CSCuo53813 − Validate IPSec Ping results blank when ESP (Encapsulating Security Payload) packets are sent
− Cisco bug ID CSCud20328 − Validate IPSec Policy shows incorrect error message in FIPS mode

## Verify the IPsec Tunnel Status on the Voice Gateway End

In order to verify whether the setup runs fine or not, it needs to be confirmed that the Security Associations (SAs) for both layers (Internet Security Association and Key Management Protoco (ISAKMP) and IPsec) are created properly.

In order to check if the SA for ISAKMP is created and works correctly, enter the ***show crypto isakmp sa*** command on the GW.

```
KRK-UC-2x2811-2#show crypto isakmp sa
IPv4 Crypto ISAKMP SA
dst                  src              state          conn-id status
209.165.201.20    209.165.201.10    QM_IDLE           1539 ACTIVE

IPv6 Crypto ISAKMP SA
```

***Note***: The proper status for the SA should be ACTIVE and QM_IDLE.

The second layer is SAs for IPsec. Their status can be verified with the ***show crypto ipsec sa*** command.

```
KRK-UC-2x2811-2#show crypto ipsec sa

interface: FastEthernet0/0
    Crypto map tag: cm3, local addr 209.165.201.20

   protected vrf: (none)
   local  ident (addr/mask/prot/port): (209.165.201.20/255.255.255.255/0/0)
   remote ident (addr/mask/prot/port): (209.165.201.10/255.255.255.255/0/0)
   current_peer 209.165.201.10 port 500
     PERMIT, flags={origin_is_acl,}
    #pkts encaps: 769862, #pkts encrypt: 769862, #pkts digest: 769862
    #pkts decaps: 769154, #pkts decrypt: 769154, #pkts verify: 769154
    #pkts compressed: 0, #pkts decompressed: 0
    #pkts not compressed: 0, #pkts compr. failed: 0
    #pkts not decompressed: 0, #pkts decompress failed: 0
    #send errors 211693, #recv errors 0

     local crypto endpt.: 209.165.201.20, remote crypto endpt.: 209.165.201.10
     path mtu 1500, ip mtu 1500, ip mtu idb FastEthernet0/0
     current outbound spi: 0xA9FA5FAC(2851757996)
     PFS (Y/N): N, DH group: none

     inbound esp sas:
      spi: 0x9395627(154752551)
        transform: esp-aes esp-sha-hmac ,
        in use settings ={Transport, }
        conn id: 3287, flow_id: NETGX:1287, sibling_flags 80000006, crypto map: cm3
        sa timing: remaining key lifetime (k/sec): (4581704/22422)
        IV size: 16 bytes
        replay detection support: Y
        Status: ACTIVE

     inbound ah sas:

     inbound pcp sas:

     outbound esp sas:
      spi: 0xA9FA5FAC(2851757996)
        transform: esp-aes esp-sha-hmac ,
        in use settings ={Transport, }
        conn id: 3288, flow_id: NETGX:1288, sibling_flags 80000006, crypto map: cm3
        sa timing: remaining key lifetime (k/sec): (4581684/22422)
        IV size: 16 bytes
        replay detection support: Y
        Status: ACTIVE

     outbound ah sas:

     outbound pcp sas:
KRK-UC-2x2811-2#
```

***Note***: Inbound and outbound Security Policy Indexes (SPIs) should be created in status ACTIVE, and counters for number of packets encapsulated/decapsulated and encrypted/decrypted should grow every time any traffic via a tunnel is generated.

The last step is to confirm that the MGCP GW is in the registered state and the TFTP configuration was downloaded properly from CUCM without any failures. This can be confirmed from the output of these commands:

```
KRK-UC-2x2811-2#show ccm-manager
MGCP Domain Name: KRK-UC-2x2811-2.cisco.com
Priority        Status                  Host
```

```
============================================================
Primary         Registered              209.165.201.10
First Backup    None
Second Backup   None

Current active Call Manager:    10.48.46.231
Backhaul/Redundant link port:   2428
Failover Interval:              30 seconds
Keepalive Interval:             15 seconds
Last keepalive sent:            09:33:10 CET Mar 24 2015 (elapsed time: 00:00:01)
Last MGCP traffic time:         09:33:10 CET Mar 24 2015 (elapsed time: 00:00:01)
Last failover time:             None
Last switchback time:           None
Switchback mode:                Graceful
MGCP Fallback mode:             Not Selected
Last MGCP Fallback start time:  None
Last MGCP Fallback end time:    None
MGCP Download Tones:            Disabled
TFTP retry count to shut Ports: 2

Backhaul Link info:
    Link Protocol:      TCP
    Remote Port Number: 2428
    Remote IP Address:  209.165.201.10
    Current Link State: OPEN
    Statistics:
        Packets recvd:   0
        Recv failures:   0
        Packets xmitted: 0
        Xmit failures:   0
    PRI Ports being backhauled:
        Slot 0, VIC 1, port 0
FAX mode: disable
Configuration Error History:
KRK-UC-2x2811-2#

KRK-UC-2x2811-2#show ccm-manager config-download
Configuration Error History:
KRK-UC-2x2811-2#
```

# Troubleshoot

This section provides information you can use in order to troubleshoot your configuration.

## Troubleshoot the IPsec Tunnel on the CUCM End

On CUCM there is no Serviceability service responsible for IPsec termination and management. CUCM uses a Red Hat IPsec tools package built in to the operating system. The daemon that runs on Red Hat Linux and terminates IPsec connection is OpenSwan.

Every time the IPsec policy is enabled or disabled on CUCM (OS Administration > Security > IPSEC Configuration), the Openswan daemon is restarted. This can be observed in the Linux messages log. A restart is indicated by these lines:

```
Nov 16 13:50:17 cucmipsec daemon 3 ipsec_setup: Stopping Openswan IPsec...
Nov 16 13:50:25 cucmipsec daemon 3 ipsec_setup: ...Openswan IPsec stopped
(...)
Nov 16 13:50:26 cucmipsec daemon 3 ipsec_setup: Starting Openswan IPsec
 U2.6.21/K2.6.18-348.4.1.el5PAE...
Nov 16 13:50:32 cucmipsec daemon 3 ipsec_setup: ...Openswan IPsec started
```

Every time there is a problem with the IPsec connection on CUCM, the last entries in the messages log should be checked (enter the ***file list activelog syslog/messages**** command) in order to confirm that Openswan is up and runs. If Openswan runs and started with no errors, you can troubleshoot the IPsec setup. The daemon responsible for the set up of IPsec tunnels in Openswan is Pluto. Pluto logs are written in order to secure logs on Red Hat, and they can be gathered via the ***file get activelog syslog/secure.**** command or via ***RTMT: Security Logs***.

***Note***: More information on how to gather logs via the RTMT can be found in the RTMT documentation.

If it is difficult to determine the source of the problem based on these logs, IPsec can be verified further by the Technical Assistance Center (TAC) via root on the CUCM. After you access CUCM via root, information and logs about the IPsec status can be checked with these commands:

```
ipsec verify (used to identify the status of Pluto daemon and IPSec)
ipsec auto ––status
ipsec auto ––listall
```

There is also an option to generate a Red Hat sosreport via root. This report contains all the information required by Red Hat support in order to troubleshoot further problems on the operating system level:

```
sosreport –batch – output file will be available in /tmp folder
```

## Troubleshoot the IPsec Tunnel on the Voice Gateway End

On this site, you can troubleshoot all phases of IPsec tunnel setup after you enable these debug commands:

```
debug crypto ipsec
debug crypto isakmp
```

***Note***: Detailed steps to troubleshoot IPsec are found in IPsec Troubleshooting: Understanding and Using debug Commands.

You can troubleshoot MGCP GW problems with these debug commands:

```
debug ccm–manager config download all
debug ccm–manager backhaul events
debug ccm–manager backhaul packets
debug ccm–manager errors
debug ccm–manager events
debug mgcp packet
debug mgcp events
debug mgcp errors
debug mgcp state
debug isdn q931
```

Updated: Apr 07, 2015 Document ID: 118886