Release Notes

Windows Security Update Issue with CUCM SSL/TLS

Cisco Unified Communications Manager
CTI integration solution

Version: 171003

1 Problem description

We have identified an issue with some Windows security updates. After installing the security Update, all **https requests to CUCM 8.x and 9.x fail**.

When executing WebDialer SOAP request, Extension Mobility SOAP request or AXL SOAP request (on https/8443) we get the following error in the logs: **The request was aborted: Could not create SSL/TLS secure channel**.

This problem occurred in our test environment after installing Cumulative Update for Windows 10 Version 1511 for x64-based Systems (KB3163018) and Update for Windows 10 Version 1511 for x64-based Systems (KB3149135), from June 14th, 2016.

This problem occurred on a customer site after June 2016 Cumulative update for Windows 7 SP1 and Windows Server 2008 R2 SP1 KB3161608.

It should also occur with KB3161606, June 2016 update rollup for Windows RT 8.1, Windows 8.1, and Windows Server 2012 R2

These cumulative Security updates include KB3161639 https://support.microsoft.com/fr-fr/kb/3161639 which installs a new list of cipherings for Internet Explorer and Edge however they are also used in priority by .Net https API. It looks like these new cipherings are not supported by previous CUCM versions' certificate.

New priority list

```
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384_P256
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384_P384
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256_P256
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256_P384
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA_P256
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA_P384
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA_P256
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA_P384
TLS_DHE_RSA_WITH_AES_256_GCM_SHA384
TLS_DHE_RSA_WITH_AES_128_GCM_SHA256
TLS_DHE_RSA_WITH_AES_256_CBC_SHA
TLS_DHE_RSA_WITH_AES_128_CBC_SHA
TLS_RSA_WITH_AES_256_GCM_SHA384
TLS_RSA_WITH_AES_128_GCM_SHA256
TLS_RSA_WITH_AES_256_CBC_SHA256
TLS_RSA_WITH_AES_128_CBC_SHA256
TLS_RSA_WITH_AES_256_CBC_SHA
TLS_RSA_WITH_AES_128_CBC_SHA
TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384_P384
TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256_P256
TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256_P384
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384_P384
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256_P256
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256_P384
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA_P256
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA_P384
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA_P256
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA_P384
TLS_DHE_DSS_WITH_AES_256_CBC_SHA256
TLS_DHE_DSS_WITH_AES_128_CBC_SHA256
TLS_DHE_DSS_WITH_AES_256_CBC_SHA
TLS_DHE_DSS_WITH_AES_128_CBC_SHA
TLS_RSA_WITH_3DES_EDE_CBC_SHA
TLS_DHE_DSS_WITH_3DES_EDE_CBC_SHA
TLS_RSA_WITH_RC4_128_SHA
TLS_RSA_WITH_RC4_128_MD5
TLS_RSA_WITH_NULL_SHA256
TLS_RSA_WITH_NULL_SHA
SSL_CK_RC4_128_WITH_MD5
SSL_CK_DES_192_EDE3_CBC_WITH_MD5
```

2 Ongoing actions and workarounds

2.1 Cisco ticket pending

We have opened a Cisco DevNet Ticket # 876, to see if it was possible to upgrade CUCM 8.x and 9.x certificate to solve this issue.

Cisco has provided the list of cypherings supported by CUCM 9, 10, 11 par priority order. The interesting one was the one used by HTTPS.

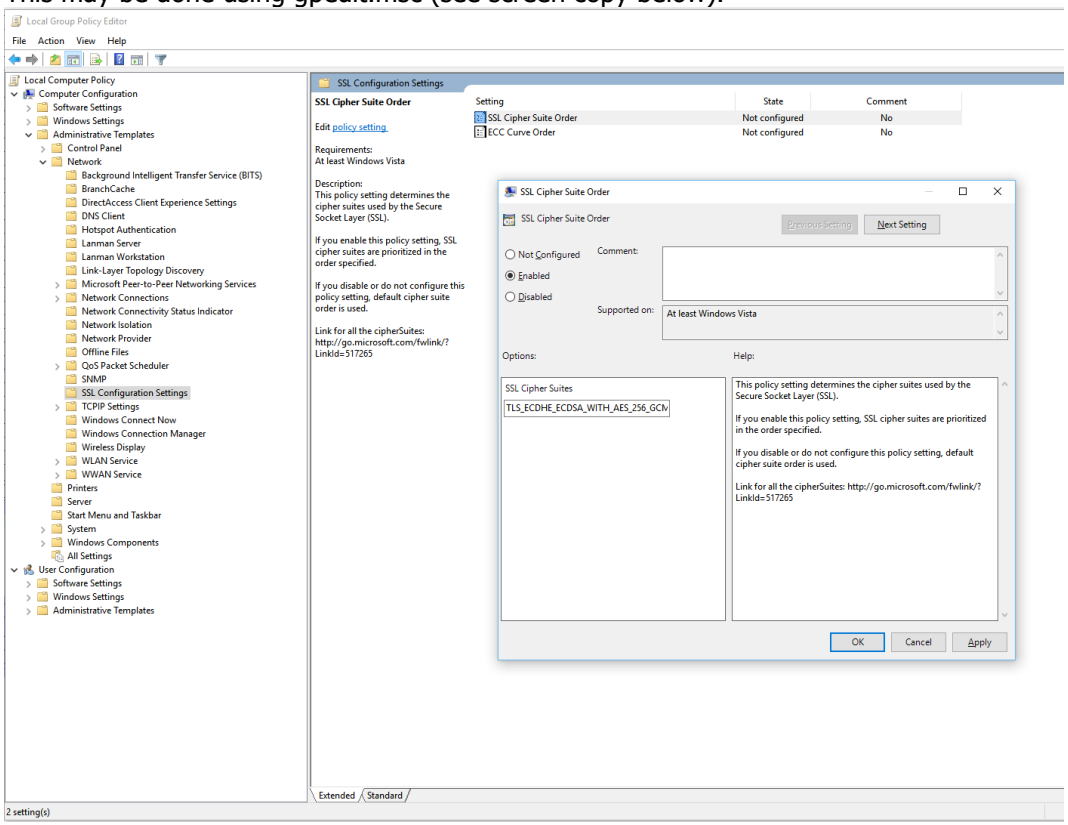
	CUCM 9.x	CUCM 10.x	CUCM 11.x
SIP TLS	RSA_WITH_NULL_SHA RSA_WITH_AES_128_CBC_SHA	RSA_WITH_NULL_SHA RSA_WITH_AES_128_CBC_SHA TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	RSA_WITH_NULL_SHA RSA_WITH_AES_128_CBC_SHA TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
SRTP	AES_CM_128_HMAC_SHA1_32	AES_CM_128_HMAC_SHA1_32 AEAD_AES_128_GCM AEAD_AES_256_GCM	AES_CM_128_HMAC_SHA1_32 AEAD_AES_128_GCM AEAD_AES_256_GCM
HTTPS	RSA_WITH_3DES_EDE_CBC_SHA DHE_RSA_WITH_3DES_EDE_CBC_SHA RSA_WITH_AES_128_CBC_SHA DHE_RSA_WITH_AES_128_CBC_SHA RSA_WITH_AES_256_CBC_SHA	RSA_WITH_3DES_EDE_CBC_SHA DHE_RSA_WITH_3DES_EDE_CBC_SHA RSA_WITH_AES_128_CBC_SHA DHE_RSA_WITH_AES_128_CBC_SHA RSA_WITH_AES_256_CBC_SHA TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	RSA_WITH_3DES_EDE_CBC_SHA DHE_RSA_WITH_3DES_EDE_CBC_SHA RSA_WITH_AES_128_CBC_SHA DHE_RSA_WITH_AES_128_CBC_SHA RSA_WITH_AES_256_CBC_SHA TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384

We do not know yet if there is a way to add more cyphering.

2.2 Patch Windows

We have found a manual solution by deleting or reordering the new cipherings added by the update.

This may be done using gpedit.msc (see screen copy below).



We have also developed a program that can be executed on the PC (with administrator's privilege). This program changes the order of one ciphering: TLS_DHE_RSA_WITH_AES_128_CBC_SHA, in the users' settings so that it is not used in priority to connect in HTTPS.

This program can be downloaded from http://telisca.com/patch/WindowsSecurityFix.zip

The application may be run from Command line, in administrator's mode. It displays the order of ciphering after execution.

It is possible to list current ciphering order by executing: **WindowsSecurityFix.exe list**

By executing **WindowsSecurityFix.exe** without parameter it moves to bottom: "TLS_DHE_RSA_WITH_AES_128_CBC_SHA". **Which solves the issue with CUCM 8 or 9.**

It is possible to revert to default Windows security patched ciphering order by executing: **WindowsSecurityFix.exe default** .

2.3 Telisca applications Workaround

We have developed workarounds in telisca applications which force the use of SSL3 instead of TLS when sending https request to CUCM 8.x or 9.x.

This Workaround has been implemented in:

- ClickNDial 2.4.2
- TSSO 2.5.4
- IPS Framework 4.7.14

We are currently implementing it in:

- IPS Phone Config/IPS Lock 4.3.2
- IPS Global Directory 4.2.5