Cisco Guide to Harden Cisco Unified Border Element (CUBE) Enterprise Devices

Contents

Introduction

This document will help you secure and harden your Cisco IOS and IOS-XE devices acting session border controller (SBC) running Cisco Unified Border Element (CUBE) Enterprise.

Prerequisites

Requirements

There are no specific requirements for this document.

Components Used

- CUBE Enterprise running IOS-XE 17.10.1a.

Note:

That not some features detailed in this document may not be available in older IOS-XE versions. Where possible care has been taken to document when a command or feature has been introduced or modified.

This document is not applicable to CUBE Media Proxy, CUBE Service Provider, MGCP or SCCP Gateways, Cisco SRST or ESRST Gateways, H323 Gateways, or other Analog/TDM Voice Gateways.

Background Information

This document serves as an addition to what can be found in the Cisco Guide to Harden Cisco IOS Devices. As such any duplicate items from that document will not be duplicated in this document.

Common Criteria (CC) and The Federal Information Standards (FIPS)

Cisco virtual CUBE utilizing IOS-XE 16.9+ on a CSR1000v or CAT8000v can utilize the command **cc-mode** command to enable a Common Criteria (CC) and The Federal Information Standards (FIPS) Certification enforcement on various cryptographic modules such as those found in Transport Layer Security (TLS) and . There is no equivilant command for CUBE running on Hardware Routers but later sections will provide methods to enable similar hardening manually.

Source: https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/voice/cube/ios-xe/config/ios-xe-book/m_cc_fips_compliance.html

Transport Layer Security (TLS) and Public Key Infrastructure (PKI)

This section will will discuss items around TLS and PKI which can enhance the secure provided by those protocols alongside Secure Session Initial Protocol (SIP) and Secure Real Time Protocol (SRTP) operations.

Use TCP TLS and SRTP

By default CUBE will accept inbound SIP connections via TCP, UDP, or SIP TCP-TLS. While the TCP-TLS connections will fail if nothing is configured, TCP and UDP will be accepted and processed by CUBE. For outbound connections SIP will utilize UDP connections by default unless a TCP or TCP-TLS command is present. Similarly CUBE will negotiate unsecure Real Time Protocol (RTP) sessions. Both of these protocols provide ample opportunity for an attacker to gleam data from an unencrypted SIP Session signaling or media stream. Where possible it is recommended to secure the SIP Signaling with SIP TLS and the media stream with SRTP.

Refer to the SIP TLS configuration and SRTP configuration guide:

- https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/voice/cube/ios-xe/config/ios-xe-book/m_sip_tls_support_cube.html
- https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/voice/cube/ios-xe/config/ios-xe-book/m_cc_fips_compliance.html?bookSearch=true#id_118373

Remember, security is only as strong as it's weakest link and SIP-TLS and SRTP should be enabled on all call-legs through CUBE.

The remaining sections will be add to these default configurations in an effort to provide additional security features:

Disable Non-Secure SIP Ports

Recall the previous section detailed that CUBE will accept inbound TCP and UDP for CUBE by default. Once SIP TLS is being used for all call legs it may be desirable to disable the unsecure UDP and TCP SIP Listen port 5060.

Once disabled you may use **show sip-ua status**, **show sip connections udp brief**, or **show sip connections tcp brief** to confirm CUBE is no longer listening on 5060 for inbound TCP or UDP SIP connections.

```
<#root>

Router#

show sip-ua status


SIP User Agent Status
SIP User Agent for UDP : ENABLED
SIP User Agent for TCP : ENABLED
SIP User Agent for TLS over TCP : ENABLED

Router#

show sip connections udp brief | i 5060


0 [0.0.0.0]:5060: 0

Router#

show sip connections tcp brief | i 5060


0 [0.0.0.0]:5060: 0!



!
sip-ua
 no transport udp
 no transport tcp
!



<#root>

Router#
```

```
show sip-ua status


SIP User Agent Status
SIP User Agent for UDP :
DISABLED


SIP User Agent for TCP :
DISABLED


SIP User Agent for TLS over TCP : ENABLED
Router#
show sip connections tcp brief | i 5060


Router#
show sip connections udp brief | i 5060
```

CUBE can also be configured to work alongside IOS-XE VRFs to provide further network segmentation.

By configuring VRFs and binding a VRF enabled interface to a dial-peer/tenant; CUBE will only listen for inbound connections for that IP, Port, VRF combination.

https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/voice/cube/ios-xe/config/ios-xe-book/m_voi-cube-multi-vrf.html

Enforce TLS 1.2

At the time of writing this document TLS 1.2 is the highest version of TLS supported by CUBE. TLS 1.0 is disabled in IOS-XE 16.9 but TLS 1.1 may be negotiated. To further limit the options during a TLS handshake an administrator may force the only avaialble version for CUBE Enterprise to TLS 1.2

```
!
sip-ua
 transport tcp tls v1.2
!
```

Enforce TLS Ciphers

It may be desirable to disable weaker TLS ciphers from being negotiated in a session. Starting in IOS-XE 17.3.1 an administrator can configure a TLS Profile which allows an administrator the ability to define exactly which TLS ciphers will be offered during a TLS session. In older versions of IOS-XE this was controlled using the **strict-cipher** or **ecdsa-cipher** postfix on the **crypto signaling** sip-ua command.

Note that the ciphers you select should be compatible with peer devices negotating SIP TLS with CUBE. Refer to all applicable vendor documentation to determine the best ciphers between all devices.

IOS-XE 17.3.1+

<#root>

```
Router(config)#
```

voice class tls-cipher 1

```
Router(config-class)#
```

cipher ?

```
<1-10> Set the preference order for the TLS cipher-suite (1 = Highest)
Router(config-class)#
```

cipher 1 ?

```
   DHE_RSA_AES128_GCM_SHA256      supported in TLS 1.2 & above
   DHE_RSA_AES256_GCM_SHA384      supported in TLS 1.2 & above
   DHE_RSA_WITH_AES_128_CBC_SHA   supported in TLS 1.0 & above
   DHE_RSA_WITH_AES_256_CBC_SHA   supported in TLS 1.0 & above
   ECDHE_ECDSA_AES128_GCM_SHA256  supported in TLS 1.2 & above
   ECDHE_ECDSA_AES256_GCM_SHA384  supported in TLS 1.2 & above
   ECDHE_RSA_AES128_GCM_SHA256    supported in TLS 1.2 & above
   ECDHE_RSA_AES256_GCM_SHA384    supported in TLS 1.2 & above
   RSA_WITH_AES_128_CBC_SHA       supported in TLS 1.0 & above
   RSA_WITH_AES_256_CBC_SHA       supported in TLS 1.0 & above
```

```
!
voice class tls-cipher 1
 cipher 1 ECDHE_RSA_AES128_GCM_SHA256
 cipher 2 ECDHE_RSA_AES256_GCM_SHA384
!
voice class tls-profile 1
 trustpoint TEST
 cipher 1
!
sip-ua
 crypto signaling default tls-profile 1
!
```

All Other Versions

<#root>

```
! STRICT CIPHERS
sip-ua
 crypto signaling default trustpoint TEST
```

strict-cipher

```
 ! Only Enables:
 ! TLS_RSA_WITH_AES_128_CBC_SHA
 ! TLS_DHE_RSA_WITH_AES_128_CBC_SHA1
```

```
 ! TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
 ! TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384


!
! ECDSA Ciphers
sip-ua
 crypto signaling default trustpoint TEST
```

ecdsa-cipher

```
 ! Only Enables:
 ! TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
 ! TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
!
```

Utilize large cryptographic keys

Cisco Next Generation Cryptography standards recommended 2048 for use with TLS 1.2 applications. The commands below can be used to create RSA keys for use with TLS sessions.

The label command allows an administrator to easily specify these keys on a trustpoint and the exportable command ensures that if needed, the private/public keypair can be exported with the command such as

crypto key export rsa CUBE-ENT pem terminal aes PASSWORD!123

<#root>

```
!
crypto key generate rsa general-keys modulus 2048 label CUBE-ENT exportable
!

Router#
```

show crypto key mypubkey rsa CUBE-ENT

```
% Key pair was generated at: 11:38:03 EST Mar 10 2023
Key name: CUBE-ENT
Key type: RSA KEYS
 Storage Device: private-config
 Usage: General Purpose Key
 Key is exportable. Redundancy enabled.
 Key Data:
  [..truncated..]
```

Utilize Certificate Authority (CA) Signed Certificates

Administrators should utilize CA signed certificates in lieu of self-signed certificates when creating trustpoint and identity (ID) certificate for CUBE enterprise.

CA certificates usually provide additional security mechanisms such as Certificate Revocation List (CRL) or Online Certificate Status Protocol (OCSP) URLs that can be used by devices to ensure the certificate has not been revoked. Using trusted public CA chains eases the trust relationship configuration on peer devices which may have embedded trust for well-known root CAs or already have Root CA trusts for your

enterprise domain.

Further, the CA certificates should include the CA Flag of True in Basic Constraints and CUBE's Identity Certificate should include Extended Key Usage parameter of Client Auth enabled.

Sample Root CA certtificate and an ID Cert for CUBE are shown below using:

openssl x509 -in some-cert.cer -text -noout

<#root>

Root CA Cert

Certificate:
[..truncated..]
 X509v3 extensions:

X509v3 Basic Constraints

:

critical

CA:TRUE

, pathlen:0
[..truncated..]
   X509v3

Extended Key Usage

:
     TLS Web Server Authentication, TLS Web

Client Authentication

[..truncated..]

ID Cert

Certificate:
 Data:
[..truncated..]
  Signature Algorithm:

sha256WithRSAEncryption

[..truncated..]
  Subject Public Key Info:
   Public Key Algorithm: rsaEncryption

RSA Public-Key: (2048 bit)

[..truncated..]

```
  X509v3 extensions:
   X509v3 Key Usage: critical
    Digital Signature, Key Encipherment
[..truncated..]
   X509v3
```

Extended Key Usage

```
:
    TLS Web Server Authentication,
```

TLS Web Client Authentication

```
[..truncated..]
```

Utilize strong hashes

When configuring a trustpoin for CUBE's Identity Certificate one should select strong hashing algorithms such as SHA256, SHA384, or SHA512:

<#root>

```
Router(config)#
```

crypto pki trustpoint CUBE-ENT

```
Router(ca-trustpoint)#
```

hash ?

```
 md5 use md5 hash algorithm
 sha1 use sha1 hash algorithm
```

sha256 use sha256 hash algorithm

sha384 use sha384 hash algorithm

sha512 use sha512 hash algorithm

Enable Certificate Revocation List (CRL) or Online Certificate Status Protocol (OCSP) Checks

By default IOS-XE Trustpionts will try to check the CRL listed within a certificate during the **crypto pki auth** command, later during the TLS handshakes IOS-XE will also perform another CRL fetch based on the recieved cert to confirm the certificate is still valid. The methods for CRL may be either HTTP or LDAP and connectivity to the CRL needs to be present for this to succeed. That is, DNS resolution, TCP socket and file download from the server to the IOS-XE router need to be available else the CRL check will fail. Similarly an IOS-XE Trustpoint can be configured to utilize OCSP value from an AuthorityInfoAccess (AIA) header within the certificate which performs queries an OCSP Responder via HTTP to check and perform similar checks. An administrator can override OCSP or CRL Distribution Point (CDP) within a

certificate by providing a static URL on a certificate. Further, an administrator can also configure the order in which CRL or OCSP are checked assuming both are present.

Many simply disable revocation checks with **revocation-check none** in order to simplify the process but in doing so an administrator weakens security and removes IOS-XE's mechanism to statefully check if a given certificate is still valid. Where possible, administrators should leverage OCSP or CRL to perform stateful checking of recieved certificates. For more on CRL or OCSP review the following document:

https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/sec_conn_pki/configuration/xe-17/sec-pki-xe-17-book/sec-cfg-auth-rev-cert.html

CRL Checking

<#root>

```
! Sample A: CRL from the certificate


crypto pki trustpiont ROOT-CA
 revocation-check crl
!

! Sample B: CRL Override OCSP in certificate


crypto pki certificate map CRL-OVERRIDE 1
 issuer-name eq root-ca.cisco.com
 subject-name eq root-ca.cisco.com
 alt-subject-name co cisco.com
!
crypto pki trustpoint ROOT-CA
 revocation-check crl
 match certificate CRL-OVERRIDE override cdp url http://www.cisco.com/security/pki/crl/crca2048.crl
!
```

OCSP Checking

<#root>

```
! Sample A: OCSP from the certificate


crypto pki trustpiont ROOT-CA
 revocation-check ocsp
!

! Sample B: Override OCSP in certificate


crypto pki certificate map OCSP-OVERRIDE 1
 issuer-name eq root-ca.cisco.com
 subject-name eq root-ca.cisco.com
 alt-subject-name co cisco.com
!
crypto pki trustpoint ROOT-CA
 revocation-check ocsp
 match certificate OCSP-OVERRIDE override ocsp 1 url http://ocsp-responder.cisco.com
!
```

Ordered OCSP and CRL Check

<#root>

```
! Check CRL if failure, check OCSP


crypto pki trustpoint ROOT-CA
 revocation-check crl ocsp
!
```

Enable Common Name (CN) and Subject Alternate Name (SAN) verification

CUBE can be configured to verify the certificate's CN or SAN match the hostname from the **session target dns:** command. In IOS-XE 17.8+ a TLS profile can be configured via tls profile.

IOS-XE 17.8+

<#root>

```
Router(config)#

voice class tls-profile 1


Router(config-class)#

cn-san validate ?


  bidirectional Enable CN/SAN validation for both client and server certificate
  client Enable CN/SAN validation for client certificate
  server Enable CN/SAN validation for server certificate
```

Remember that the client/server designation are in reference to the peer devices role in the TLS handshake

To further illustrate:

- **cn-san validate server:** CUBE will perform hostname validation of received peer *server* certificates for outbound TLS connections where CUBE is the client role.
- **cn-san validate client:** CUBE will perform hostname validation of received peer *client* certificates for inbound TLS connections where CUBE is the server role.
- **cn-san validate bidirection**: Enables hostname validation for both peer roles during the TLS handshake.

When using the **cn-san validate client** command (or bidirectional) you must configure a SAN to check against since the session target is check is only for outbound connections and cn-san validate server.

Client Hostname Validation:

<#root>

```
!
```

```
voice class tls-profile 1
 cn-san validate client
 cn-san 1 *.example.com
 cn-san 2 subdomain.example.com
!
```

Server Hostname Validation:

```
!
voice class tls-profile 1
 cn-san validate server
!
sip-ua
 crypto signaling default tls-profile 1
!
dail-peer voice 1 voip
 session target dns:subdomain.example.com
!
```

Prior to 17.8.1

Note: Only server hostname validation is available via this method.

<#root>

```
!
sip-ua
 crypto signaling default trustpoint TEST
```

cn-san-validate server

```
!
dail-peer voice 1 voip
 session target dns:subdomain.example.com
!
```

CUBE can also be configurued to send Server Name Indication (SNI) TLS 1.2 extension with CUBE's FQDN hostname within the TLS handshake to peer devices to facilitate their hostname validation efforts.

```
!
voice class tls-profile 1
 sni send
!
sip-ua
 crypto signaling default tls-profile 1
!
```

A note on CUBE's Mutual TLS:

- By default when CUBE is acting as a TLS server (read inbound TLS connection) it will always request a client certificate. There is no configuration to disable this behavior.
- When CUBE is acting as a TLS client and initiating an outbound TLS connection mutual TLS is up to the peer device acting as a TLS Server. In this scenario a peer device may not request a client certificate from CUBE.
- In both of these scenarios the certificate chain CUBE would send is controlled by the **trustpoint** defined in the TLS profile or on the crypto signaling command.

<#root>

```
!
sip-ua
 crypto signaling default
```

trustpoint CUBE-ENT

```
!
! OR
voice class tls-profile 1
```

trustpoint CUBE-ENT

```
!
sip-ua
 crypto signaling default tls-profile 1
!
```

Map remote TLS connections to specific trustpoints

When using **crypto signaling default** sip-ua command **ALL** inbound TLS connections are mapped to these configuration either via tls-profile or individual post-fix commands. Furthermore, all available trustpoints are checked when performing certificate validation.

It may be desirable to create specific TLS profile configurations for specific peer device based on IP address to ensure exactly the security parameters you define are applied to that TLS session. To do this use the **crypto signaling remote-addr** command to define an IPv4 or IPv6 subnet to map to a tls-profile or set of postfix commands. You may also directly map verification trustpoint via **client-vtp**) commands to lock down exactly which trustpoints are used to validate peer certificates.

The command below summarizes most items discussed up to this point:

```
!
voice class tls-cipher 1
 cipher 1 ECDHE_RSA_AES128_GCM_SHA256
 cipher 2 ECDHE_RSA_AES256_GCM_SHA384
!
voice class tls-profile 1
 trustpoint CUBE-ENT
 cn-san validate bidirectional
 cn-san 1 *.example.com
 cipher 2
 client-vtp PEER-TRUSTPOINT
```

```
 sni send
!
sip-ua
 crypto signaling remote-addr 192.168.1.0 /24 tls-profile 1
!
```

For older versions this can be done like so:

```
!
sip-ua
 crypto signaling remote-addr 192.168.1.0 /24 trustpoint CUBE-ENT cn-san-validate server client-vtp PEER
!
```

Starting in 17.8 you may also configure tls-profile and per-tenant listen ports per **voice class tenant** to provide furhter segmentation options on a given listen port.

```
!
voice class tenant 1
 tls-profile 1
 listen-port secure 5062
!
```

Enforce Strict SRTP

When enabling SRTP on CUBE Enterprise the default operation is to disallow fallback to RTP.

Where possible use SRTP on all call-legs however by default CUBE will perform RTP-SRTP as needed.

Note that CUBE does not log the SRTP keys in debugs starting in 16.11+

```
!
voice service voip
 srtp
!
! or
!
dial-peer voice 1 voip
 srtp
!
```

Trim unsecure SRTP Ciphers

By default all SRTP ciphers are sent by CUBE when creating an offer. An administrator can trim down to more secure ciphers such as the next-generation AEAD cipher suites by using the voice class srtp-crypto command in IOS-XE 16.5+.

This configuration can also change the default preference used when CUBE selects an SRTP Cipher and creates an answer to some offer with multiple options available.

Note: Some older Cisco devices or peer devices may not support AEAD ciphers. Refer to all applicable documentation when triming cipher suites.

```
<#root>

Router(config)#

voice class srtp-crypto 1


Router(config-class)#

crypto ?

  <1-4> Set the preference order for the cipher-suite (1 = Highest)


Router(config-class)#

crypto 1 ?


  AEAD_AES_128_GCM        Allow secure calls with SRTP AEAD_AES_128_GCM cipher-suite
  AEAD_AES_256_GCM        Allow secure calls with SRTP AEAD_AES_256_GCM cipher-suite
  AES_CM_128_HMAC_SHA1_32 Allow secure calls with SRTP AES_CM_128_HMAC_SHA1_32 cipher-suite
  AES_CM_128_HMAC_SHA1_80 Allow secure calls with SRTP AES_CM_128_HMAC_SHA1_80 cipher-suite
```

```
!
voice class srtp-crypto 1
 crypto 1 AEAD_AES_256_GCM
 crypto 2 AEAD_AES_128_GCM
!
voice service voip
 sip
  srtp-crypto 1
!
! or
!
voice class tenant 1
 srtp-crypto 1
!
! or
!
dial-peer voice 1 voip
 voice-class srtp-crypto 1
!
```

Disable Other Unused VoIP Protocols

If H323, MGCP, SCCP, STCAPP, CME, SRST are not being used on this gateway it is worth removing the configurations to harden CUBE.

Disable H323 and only allow SIP to SIP calls

```
!
voice service voip
 allow-connections sip to sip
 h323
  call service stop
!
```

Disable MGCP, SCCP, STCAPP, SIP and SCCP SRST.

Note: Some of these commands will delete all other configurations, ensure features are not being used before removing them completly.

<#root>

```
Router(config)#
```

no mgcp

```
Router(config)#
```

no sccp

```
Router(config)#
```

no stcapp

```
Router(config)#
```

no voice register global

```
Router(config)#
```

no telephony-service

```
Router(config)#
```

no call-manager-fallback

Call Routing and Toll Fraud

Allow Connections from Trusted IPs

By default CUBE will trust inbound connections from IPv4 and IPv6 addresses configured on dial-peer **session target** and **voice class server-group** configurations.

To add additional IP addresses utilize the **ip address trusted list** command configured via **voice service voip.**

When client/server hostname validation is configured alongside SIP TLS by way of the CN/SAN validate feature previously discussed, a successful CN/SAN validation will bypass IP address trusted list checks.

Avoid using **no ip address trusted authenticate** which will enable CUBE to accept ANY inbound connection.

```
!
voice service voip
 ip address trusted authenticate

 ip address trusted list
  ipv4 192.168.1.1
  ipv4 172.16.1.0 /24
!
```

Use **show ip address trusted list** to view the status of IP Address checking and all static and dynamic trusted list definitions derived from other configurations.

Note that the dynamic value derived from a dial-peer/server-group is removed from the trusted list when a dial-peer is shutdown or set to down down state after failing keepalive checks.

By default when an inbound call does not pass the IP Trusted list check it is silently discarded but this can be override using the **no silent-discard untrusted** voice service voip > sip command to send an erorr back to the sender. However by sending a response an attacker may use this to indicate that the device is in fact listening for SIP Traffic and ramp up their attack efforts. As such silent discard is the preferred method of handling IP Trusted List drops.

Avoid generic dial-peer routing

Using generic "catch all" destination-patterns such as **destination-pattern .T** can increase the likelihood of routing a fraudulent call through CUBE.

Administrators should configure CUBE to only route calls for known phone number ranges or SIP URIs.

See the following document for a greater explanation of CUBE Call Routing features:

https://www.cisco.com/c/en/us/support/docs/voice/ip-telephony-voice-over-ip-voip/211306-In-Depth-Explanation-of-Cisco-IOS-and-IO.html

CUBE Threat Mitigation

Malformed Packet Handling

By default CUBE will inspect SIP and RTP packets to check for for errors and drop the packet.

Rogue RTP Packets

By default IOS-XE CUBE performs source-port validation for all RTP/RTCP streams by only allowing connections negotiated via SIP SDP offer/answer signaling and cannot be disabled.

These can be monitored by checking the following command:

```
show platform hardware qfp active feature sbc global | s Total packets dropped|Dropped packets:
```

For interop with CUCM it is recommended to enable Duplex Media streaming via the Cisco CallManager Service to avoid Music on Hold being dropped when sourced from Port 4000.

RTP Port Range Hardening

By default IOS-XE uses the port range of 8000 through 48198. This can be configured to a different range such as 16384 through 32768 via the following command:

```
!
voice service voip
 rtp-port range 16384 32768
!
```

An administrator may also configure RTP port ranges per IPv4 and IPv6 Address Ranges.

This configuration also enables the VoIP application of CUBE to perform phantom packet handling more efficiently by not punting these packets to the UDP Process at the Router's CPU since the IP and Port Range are statically defined. This can help mitigate high CPU when handling a large number legitimate or ilegitimate RTP packets by bypassing the CPU punting behavior.

```
voice service voip
 media-address range 192.168.1.1 192.168.1.1
  port-range 16384 32768
 media-address range 172.16.1.1 172.16.1.1
  port-range 8000 48198
```

https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/voice/cube/ios-xe/config/ios-xe-book/m_phantom-packet-handling.html

Denial of Service (DOS) prevention

Call Admission Control featurs can be be enabled to limit calls based on Total Calls, CPU, Memory, Bandwidth. In addition Call Spikes can be detected to reject calls and prevent denial of service.

https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/voice/cube/ios-xe/config/ios-xe-book/m_voi-cube-call-admission-control.html

Address Hiding

By default CUBE will replace IP addresses in SIP headers such as, but not limited to Via, Contact, and From with it's own IP address.

This can be extended to Refer-To, Referred-By, 3xx contact header, History-Info, and Diversion headers by applying the **voice service voip** command **address-hiding**.

Additionally a new call-id is created for each call-leg mitigating IP address which may be embedded in this header value.

Where a hostname is required in place of an IP address for address hiding purposes the command **voice-class sip localhost dns:cube.cisco.com** can be configured.

Caller ID Privacy

CUBE can be configured to drop Caller ID Name values from SIP Headers with the command **clid-strip name** configured on any dial-peer.

Furthermore CUBE can interwork and understand SIP Privacy headers such as P-Preferred Identity (PPID), P-Asserted Identity (PAID), Privacy, P-Called Party Identity (PCPID), Remote-Party Identity (RPID). For more information refer to the followind document: https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/voice/cube/ios-xe/config/ios-xe-book/m_voi-paid-ppid-priv.html

SIP Digest Authentication

During SIP Registration by CUBE to a service provider or during a call signaling upstream UAS devices may return a 401 or 407 status code with an appliicable WWW-Authenticate/Proxy-Authenticate header field challenging CUBE to authenticate. During this handshake CUBE supports the MD5 algorithm for computing the Authorization header field value in a subsiquent request.

Unsupported SIP Headers or SDP

CUBE will strip unsupported SIP Headers or SDP that it does not understand. Care should be taken when using commands such as **pass-thru content sdp**, **pass-thru content unsupp**, or **pass-through headers unsupp** to ensure what data is making it through CUBE.

Removing or Modifying SIP Headers or SDP

Where additional control is required inbound or outbound SIP profiles can be configured by an administrator to flexibly modify or outright drop a sip header or SDP attribute.

Refer to the following documents on SIP Profile usage:

- https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/voice/cube/ios-xe/config/ios-xe-book/m_voi-sip-param-mod.html
- https://www.cisco.com/c/en/us/support/docs/voice/ip-telephony-voice-over-ip-voip/211306-In-Depth-Explanation-of-Cisco-IOS-and-IO.html#anc45

Other Security Features

Encrypted Passwords

CUBE requires encrypted passwords for 16.11 and later versions to encrypt SIP Registration and other IOS-XE passwords in the running configuration.

```
password encryption aes
key config-key password-encrypt cisco123
```

Access Lists

The trusted list feature operates at Layer 7 within the CUBE application. By the time the packet is dropped silently the CUBE has already started processing the packet.

It may be desirable to lock down interfaces with inbound or outbound Layer 3 or 4 access lists to drop the packet at the entry point of the router.

This ensures CPU cycles from CUBE are spent on legitimate traffic. ACLs alongside IP Trusted List and Hostname Validation provide a layered approach to CUBE security.

Zone-Based Firewall (ZBFW)

Cisco CUBE can be configured alongside IOS-XE ZBFW to provide application inspection and other security features.

Refer to the CUBE and ZBFW Guide for more information on this topic:

https://www.cisco.com/c/en/us/support/docs/unified-communications/unified-border-element/220378-configure-zone-based-firewall-zbfw-co.html