

Best Practices

HP Poly Device Security for Poly Voice Software

Contents

Version Tracking.....	5
General Notes and Reminders.....	6
Device Applicability.....	6
Device Parameters.....	6
General Security Settings.....	6
Passwords.....	6
Administrator Password.....	6
Advanced User Password.....	7
Basic User Password.....	7
Set Password Length Requirements.....	8
Lock the Basic Settings Menu.....	8
Web Server.....	8
System Web Interface Configuration.....	8
Security Banner on the System Web Interface.....	8
Brute Force Attack Prevention.....	9
Max Sessions and Idle Timeout Settings.....	10
Two-Factor Authentication.....	10
Multi-Factor Authentication for Whitelisting.....	10
Telephony APIs.....	10
Contact Directory.....	11
Minimum Ringer Volume.....	11
Hide the MAC Address.....	11
SHA256 for Digest Authentication on HTTP-based Services.....	12
SIP Signaling.....	13
SHA256/512 Digest Authentication.....	13
Enable SIP over TLS.....	13
SRTP for Audio/Video.....	13
SRTP - SHA256 Crypto Suites.....	14
Ignore Anonymously Directed Calls.....	14
Lock Signaling to the Registrar.....	14
Enable Re-Registration on Failover (RROFO).....	14

Disabling Physical Ports, Buttons, and Services	15
Disable USB Ports.....	15
Disable the auxiliary port	15
Disable the Serial Port.....	15
Disable the PC Ethernet Port.....	15
Disabling Speakerphone/Handsfree, Headset or Handset ports.....	15
Disabling Physical Buttons.....	16
Example: Disable the "Home Key" on VVX 101.....	16
Disable Do Not Disturb and Call Forwarding.....	16
Bluetooth, Wi-Fi, Mobile Phone and Headset Pairing	16
Bluetooth.....	16
Wi-Fi.....	16
Provisioning	17
Using HTTPS for Provisioning.....	17
Encrypted Configuration Files.....	17
Quick Setup.....	18
Verify the Phone's Device Certificate Common Name	18
Mutual TLS	19
Root CA Certificates for Server Authentication.....	19
Install and Store Root CA Certificates.....	19
Using Device Certificates	22
Using Custom Device Certificates	22
Installing Platform Certificates.....	23
Installing Application Certificates	25
Set Device Credentials for TLS Profiles	25
Software Update.....	26
Reset Alternate Software Upgrade Servers	26
Software Update Rollback Prevention	26
Simple Certificate Enrollment Protocol (SCEP).....	26
802.1X Access Port Security.....	28
802.1X Acronyms.....	28
Supported EAP Authentication Protocols.....	29

EAP Authentication Requirements	29
EAP-TLS.....	29
EAP-PEAPv0/EAP-MSCHAPv2 or EAP-PEAPv0/EAP-GTC.....	29
EAP-TTLS/EAP-MSCHAPv2 or EAP-TTLS/EAP-GTC.....	29
EAP-MD5	29
EAP-FAST	29
Configuring 802.1X Authentication	30
Configuring 802.1X Using a Central Provisioning Server	30
802.1X and the PC Port.....	34
Configuring 802.1X Using the System Web Interface.....	35
Configuring 802.1X Using the Local Phone User Interface.....	36
SSL/TLS Protocol Selection	36
Restricting TLS Versions.....	36
Configuring TLS Application Protocols.....	37
Poodle Security Exploit – Deprecating SSLv2 & v3.....	37
Cipher Usage.....	37
Customizing TLS Cipher Usage.....	37
Web Server – System Web Interface.....	38
FIPS Mode.....	38
Security Bulletins	39

Version Tracking

Version	Date	Changes
1.0	August 2025	Includes security capabilities in PVOS 8.3.0/9.2.0 releases and earlier

General Notes and Reminders

Device Applicability

This guide applies to the following HP products:

- Poly VVX series
- Poly CCX series
- Poly Trio series
- Poly Edge E series

Note that not all features listed in this guide are available on all devices. Refer to software release notes and other Poly documentation for details on supported features.

HP strongly recommends upgrading all devices to the latest software releases to take advantage of the latest security features and fixes for issues discovered in prior releases.

Device Parameters

When working with device parameters in configuration files, include the `device.set="1"` parameter and corresponding `device.xxxxxxx.set="1"` parameters to save the device parameter values to flash memory.

Set the `device.set="1"` parameter only once in the configuration file.

In this document, the `device.xxxxxxx` parameter and its corresponding `device.xxxxxxx.set` parameters are usually shown together. In some cases, if the `device.xxxxxxx.set` parameter is omitted, unless stated otherwise, assume that it is required to save the device configuration change.

General Security Settings

Passwords

New or factory-restored Poly phones use a default admin password (456) to access the system web interface. Poly phones prompt users to change the admin password before accessing the system web interface. Users can change the password from the start-up menu on the device or using the provisioning server.

Administrator Password

Poly phones display a warning icon on the user interface if the phone is using the default password (456). In UC Software 5.1 and later, you can change the password using these configuration settings:

```
device.auth.localAdminPassword.set="1"  
device.auth.localAdminPassword="xxxxxxxxxxxxxx"
```

Tip: When using the Cisco BroadWorks Device Management System, set the password using a **BW** tag, allowing a unique password for every device. Similar techniques can be used with other tags to create unique passwords, such as timestamps:

```
device.auth.localAdminPassword="467%BWEXTENSION-1%529"
```

Advanced User Password

Poly phones include an advanced user access feature to give users access to all administrator features except the following:

- Line configuration
- Call server configuration
- Test automation
- Network and security configuration disablement

To enable the advanced user feature, use the following parameter:

```
feature.advancedUser.enabled="1"
```

By default, the advanced user password is set to the administrator password. To set a separate password for the advanced user, use the following parameters:

```
device.auth.localAdvancedPassword.set="1"  
device.auth.localAdvancedPassword="xxxxxx"
```

To enable the system web interface access, use the following parameter:

```
feature.advancedUser.web.enabled="1"
```

To disable network and TLS security menus, use the following parameters:

```
ui.menu.advancedUser.networkConfiguration="0"  
ui.menu.advancedUser.networkConfiguration.tls="0"
```

Basic User Password

Poly phones also support a basic user password that opens a limited subset of options under the advanced settings menu. The default is 123.

To set the user password, use the following parameters:

```
device.auth.localUserPassword.set="1"  
device.auth.localUserPassword="xxxxxx"
```

Note: In PVOS 8.3.0 (Edge E series) and PVOS 9.2.0 (CCX series/Trio C60), users are unable to log in to the phone's system web interface if the basic user password is not changed from the default.

Set Password Length Requirements

To set more secure password requirements, use the following parameters to set the minimum password length:

```
sec.pwd.length.admin="<min password length>"
sec.pwd.length.user="<min password length>"
sec.pwd.length.advanced="<min password length>"
```

Lock the Basic Settings Menu

In UC Software 5.4.0 and later, Poly phones support password protecting the Basic Settings menu.

To set a password for the Basic Settings menu, use the following parameters:

```
up.basicSettingsPasswordEnabled="1"
```

Web Server

System Web Interface Configuration

In UC Software 5.3 or later, Poly phones enable the system web interface by default and set HTTPS as the default protocol. However, in Skype and Microsoft Teams base profiles, the system web interface is disabled by default.

To set the HTTP/HTTPS protocol usage, use the following parameters:

```
httpd.cfg.secureTunnelEnabled="1"
httpd.cfg.secureTunnelRequired="1"
httpd.cfg.secureTunnelPort="443"
```

To disable access to the system web interface, use the following parameter:

```
httpd.cfg.enabled="0"
```

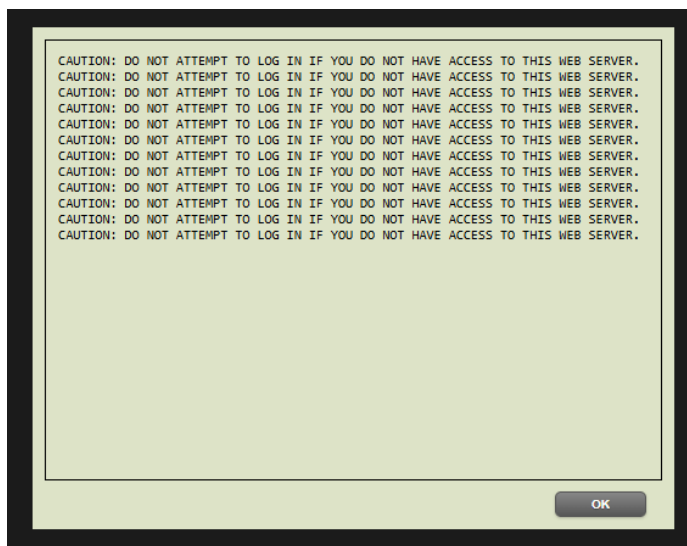
To disable the web server, use the following parameter:

```
httpd.enabled="0"
```

Note: Disabling the web server also prevents access from other services reliant on the HTTP server, such as REST or XML API access.

Security Banner on the System Web Interface

In UC Software 5.5.2 and later, you can configure a security banner with up to 2000 characters to display on the web interface.



To configure the security banner, use the following parameters:

```
feature.webSecurityBanner.enabled="1"
feature.webSecurityBanner.msg("<security banner message>")
```

Tip: You can use the HTML code "
" to enter a carriage return in the message.

Brute Force Attack Prevention

In UC Software 5.5.0 and later, Poly phones are enabled by default to prevent brute force attacks through the system web interface.

To enable brute force attack prevention, use the following parameters:

```
httpd.cfg.lockWebUI.enable="1"
httpd.cfg.lockWebUI.lockOutDuration="60"
httpd.cfg.lockWebUI.noOfInvalidAttempts="5"
httpd.cfg.lockWebUI.noOfInvalidAttemptsDuration="60"
```

Max Sessions and Idle Timeout Settings

Poly phones have preset session management rules, but you can customize these rules as needed. By default, Poly phones allow 10 concurrent sessions on the web interface and a single session to remain idle for 900 seconds (15 minutes) before it automatically ends.

To modify max session and timeout settings, use the following parameters:

```
httpd.cfg.session.maxSessionAge="<session duration>"  
httpd.cfg.session.maxSessions="<concurrent session max>"
```

Two-Factor Authentication

As an increased security option, you can enable two-factor authentication for accessing the web interface, remote packet capture, or telnet diagnostic services. When attempting to access the web interface, for example, the user will enter the admin password and then enter approval on the physical device as well.

To enable two-factor authentication, use the following parameter:

```
feature.mfa.enabled="1"
```

Multi-Factor Authentication for Whitelisting

Poly phones support multi-factor authentication (MFA) for the following whitelisting scenarios:

- If the devices are whitelisted by "SourceIp", the whitelist affects all services. For example, accepting an MFA request for telnet will also whitelist any subsequent web interface or remote packet capture attempts from the same source IP for the specified duration. Any connection attempts after the whitelist duration expires will require another MFA action via the phone.
- If the devices are whitelisted by "SourceIpAndService", connection attempts from the same source to different services are treated separately. For example, MFA authentication of a telnet connection will only whitelist subsequent telnet connections from the same source IP while the web interface or packet capture connections will require separate MFA actions via the phone.

To enable MFA for whitelisting methods, use the following parameters:

```
feature.mfa.timeout="60", 0-3600, default 60 (seconds)  
feature.mfa.whitelist.enabled="1", default 0  
feature.mfa.whitelist.duration="600", 1-3600 (seconds)  
feature.mfa.whitelist.method="<insert option>", "SourceIP" or  
"SourceIpAndService", default "SourceIp"
```

Telephony APIs

XML and REST APIs are disabled by default on Poly phones.

XML API

In UC Software 5.3 and later, Poly phones use HTTPS by default for XML APIs.

To enable XML APIs, use the following parameters:

```
apps.push.username=""  
apps.push.password=""  
apps.push.secureTunnelEnabled="1"  
apps.push.secureTunnelRequired="1"  
apps.push.secureTunnelPort="443"
```

REST API

In UC Software 5.4 and later, Poly phones use HTTPS by default for REST APIs. The password is the same as the phone's administrator password.

To enable REST APIs, use the following parameter:

```
apps.restapi.enabled="1"
```

Contact Directory

Use the following parameter to configure the local contact directory to read-only mode, which prevents users from adding, deleting, or modifying contacts from the phone's interface.

```
dir.local.readonly="1"
```

Alternatively, in UC Software 5.5 and later, you can password-protect the local contact directory to restrict edit privileges to those with a user or admin password.

To password-protect edit permissions for the contact directory, use the following parameter:

```
dir.local.passwordProtected="1"
```

Minimum Ringer Volume

In UC Software 5.5 and later, Poly phones support setting the minimum ringer volume level to define how many volume steps are accessible below the maximum level by the user. For example: Upon bootup, the volume is set to ½ the number of configured steps below the maximum (16). If the parameter is set to 8, on bootup, the ringer volume is set to 4 steps below maximum.

To set the minimum ringer volume level, use the following parameter:

```
up.ringer.minimumVolume="xx"  
Permitted values: 0-16
```

Hide the MAC Address

In UC Software 5.7 and later, Poly phones support hiding the phone's MAC address information from the phone display.

Use the following parameters to configure this feature:

```
device.mac.hide.set="1"  
device.mac.hide="1"
```

Configuring this feature:

1. Removes the MAC address from the display on the Welcome screen.

2. Removes the MAC address from the Info screen during bootup.
3. Disables the 1-4-7 multi-key combination that displays system information.
4. Locks the Status menu behind the administrator password.

SHA256 for Digest Authentication on HTTP-based Services

In PVOS 8.3.0 (Edge E) and PVOS 9.2.0 (CCX and Trio), Poly phones support SHA256 for digest authentication for various HTTP-based services.

Note: MD5 is still supported in PVOS 8.3.0 and PVOS 9.2.0 and can be disabled.

REST API

Use the following parameters to configure SHA256 for REST APIs:

```
httpd.rest.auth.digest.md5.enabled="1"  
httpd.rest.auth.digest.sha256.enabled="1"
```

XML Polling and Push API services

Use the following parameters to configure SHA256 for XML services:

```
httpd.apps.auth.digest.md5.enabled="1"  
httpd.apps.auth.digest.sha256.enabled="1"
```

TR-069

Use the following parameters to configure SHA256 for TR-069:

```
httpd.tr069.auth.digest.md5.enabled="1"  
httpd.tr069.auth.digest.sha256.enabled="1"
```

Modular Room (C60 daisy chaining)

Use the following parameters to configure SHA256 for daisy-chaining:

```
httpd.mrprov.auth.digest.md5.enabled="1"  
httpd.mrprov.auth.digest.sha256.enabled="1"
```

Telephony Event Notification services

Use the following parameters to configure SHA256 for screen capture events:

```
httpd.site.auth.digest.md5.enabled="1"  
httpd.site.auth.digest.sha256.enabled="1"
```

SIP Signaling

SHA256/512 Digest Authentication

In PVOS 8.3.0 (Edge E) and PVOS 9.2.0 (CCX/Trio), Poly phones support SHA256 and SHA512 digest authentication.

Note: MD5 is still supported and can be disabled.

Use the following parameters to configure this feature:

```
voIpProt.SIP.auth.digest.md5.enabled="1"  
voIpProt.SIP.auth.digest.sha256.enabled="1"  
voIpProt.SIP.auth.digest.sha512-256.enabled="1"
```

Enable SIP over TLS

Use the following parameters to enable this feature:

```
voIpProt.server.x.transport="TLS"  
voIpProt.SIP.outboundProxy.transport="TLS"  
reg.x.outboundProxy.transport="TLS"  
reg.x.server.y.transport="TLS"
```

Note: HP strongly recommends enabling SIP over TLS in “Over The Top” (OTT) deployment scenarios, as using unsecure and connectionless transports, like UDP, can allow SIP call generator applications like SIPVicious to send SIP messages to the phone on a known UDP SIP port and trigger the phone to perform undesired behaviors. Using secure and connection-based transport like TLS will block call generator applications from reaching the phone, even if the source TCP port the phone uses is known.

SRTP for Audio/Video

Use the following parameters to enable this feature:

```
sec.srtp.enable="1"  
sec.srtp.offer="1"  
sec.srtp.require="1"
```

Use the following registration-based parameters to configure this feature:

```
reg.x.srtp.enable="1"  
reg.x.srtp.offer="1"  
reg.x.srtp.require="1"
```

SRTP – SHA256 Crypto Suites

Use the following parameters to enable SHA256 crypto suites for SRTP offers and accept offers in SIP calls:

```
sec.srtp.offer.AES_ICM_256="1"  
sec.srtp.offer.AES_GCM_256="1"
```

Ignore Anonymously Directed Calls

Use the following parameter to ignore any inbound SIP message that doesn't specifically reference a registration acknowledgement of receipt (AOR) on the phone in the INVITE request URI:

```
voIpProt.SIP.strictUserValidation="1"
```

Lock Signaling to the Registrar

Use the following parameter to enable incoming SIP requests to be locked to the SIP registrar:

```
voIpProt.SIP.requestValidation.x.method="source"  
voIpProt.SIP.requestValidation.x.request="INVITE"  
(or any other SIP request method)
```

Enable Re-Registration on Failover (RROFO)

In addition to redundancy configuration, the RROFO feature enables the phone to lock onto and signal only with the currently registered SBC or call server thus ignoring any external malicious signaling (e.g., the SIPVicious call generator).

Note: Refer to the Engineering Advisory 66546: Configuring Optional Re-Registration on Failover Behavior on [HP Community](#) for additional details.

Use the following parameters to configure this feature on a per registration index:

```
reg.X.outboundProxy.failOver.reRegisterOn="1"  
reg.X.outboundProxy.failOver.onlySignalWithRegistered="1"  
reg.X.outboundProxy.failOver.failBack.mode="duration"  
reg.X.outboundProxy.failOver.failRegistrationOn="1"  
reg.X.outboundProxy.failOver.failBack.timeout="3600"
```

Use the following parameters to configure this feature for all registrations:

```
voIpProt.SIP.outboundProxy.failOver.reRegisterOn="1"  
voIpProt.SIP.outboundProxy.failOver.onlySignalWithRegistered="1"  
voIpProt.SIP.outboundProxy.failOver.failBack.mode="duration"  
voIpProt.SIP.outboundProxy.failOver.failRegistrationOn="1"  
voIpProt.SIP.outboundProxy.failOver.failBack.timeout="3600"
```

Disabling Physical Ports, Buttons, and Services

Disable USB Ports

In UC Software 5.2 and later, Poly phones support locking and disabling USB ports.

Use the following parameter to lock USB ports:

```
feature.usb.power.enabled="0"
```

In UC Software 5.3 and later, you can use the following parameters to lock specific USB ports:

```
feature.usbRear.power.enabled="0"  
feature.usbTop.power.enabled="0"
```

Use the following parameter to disable all USB ports:

```
feature.usb.host.enabled="0"
```

Use the following parameter to disable only USB storage devices. This allows users to use USB peripherals like headsets. This parameter is supported in UC Software 6.4.7 for VVX series, PVOS 8.0.2 for Edge E series, and PVOS 9.1.1 for CCX series and Trio C60.

```
feature.usb.host.massStorage="0"
```

Disable the auxiliary port

Use the following parameters to disable the port used to connect expansion modules:

```
device.auxPort.enable.set="1"  
device.auxPort.enable="0"
```

Disable the Serial Port

Use the following parameters to disable the serial port:

```
device.serial.enable.set="1"  
device.serial.enable="0"
```

Disable the PC Ethernet Port

Use the following parameters to disable the PC Ethernet port:

```
device.net.etherModePC.set="1"  
device.net.etherModePC="Disabled"
```

Disabling Speakerphone/Handsfree, Headset or Handset ports

Use the following parameters to disable the speakerphone, headset, and handset ports:

```
up.handsfreeMode="0"  
up.handsetModeEnabled="0"  
up.headsetModeEnabled="0"
```

Disabling Physical Buttons

The Poly phone administrator guides include key layouts for all phones in the section titled "Define the Phone Key Layout". These layouts describe the required key ID used in the following configuration example.

Example: Disable the "Home Key" on VVX 101

On VVX 101, key ID #13 maps to the "Home" button. To disable the "Home" key, use the following parameter:

```
key.13.function.prim="null"
```

Disable Do Not Disturb and Call Forwarding

Use the following parameters to disable DND and call forwarding to prevent call diversion:

```
feature.doNotDisturb.enable="0"  
feature.forward.enable="0"
```

Bluetooth, Wi-Fi, Mobile Phone and Headset Pairing

Bluetooth

Use the following parameters to disable Bluetooth radio:

```
bluetooth.radioOn="0"  
feature.bluetooth.enabled="0"
```

Wi-Fi

Poly phones support disabling Wi-Fi settings to meet requirements for high security environments.

Use the following parameters to disable Wi-Fi:

```
device.wifi.enabled="0"  
device.wifi.enabled.set="1"
```

Use the following parameters to define a specific SSID:

```
device.wifi.ssid "<SSID>"  
device.wifi.ssid.set="1"
```

Use the following parameters to configure Wi-Fi with a specific mode:

```
device.wifi.securityMode "<mode>"  
device.wifi.securityMode.set="1" Permitted values: "NULL" (default),  
"WPA-PSK", "WPA2-PSK", "WPA2-Enterprise"
```

Use the following parameters to set the passphrase or hexadecimal key for WPA2-PSK:

```
device.wifi.psk.keyType="0" (0=passphrase (default), 1=hexadecimal)  
device.wifi.psk.keyType.set="1"
```

```
device.wifi.psk.key="<enter key>"
device.wifi.psk.key.set="1"
```

Use the following parameters to set credentials for networks using WPA PSK or WPA2 PSK Enterprise:

```
device.wifi.wpa2Ent.method="<EAP setting>"
device.wifi.wpa2Ent.method.set="1"
device.wifi.wpa2Ent.user="<WPA2 username>"
device.wifi.wpa2Ent.user.set="1"
device.wifi.wpa2Ent.password="<WPA2 password>"
device.wifi.wpa2Ent.password.set="1"Permitted values for EAP Setting:
"NULL" (default), "EAP-PEAPv0/MSCHAPv2", "EAP-TLS", "EAP-TTLS-
MSCHAPv2", "EAP-PEAPv0-NONE", "EAP-TTLS-NONE", "EAP-PWD"
```

Use the following parameters to remove the Wi-Fi options from the local menu on the phone to prevent a user from seeing or modifying Wi-Fi settings:

```
feature.wifiUserSettings.enabled="0"
```

Provisioning

Using HTTPS for Provisioning

This is recommended for provisioning phones.

When setting up provisioning servers, HP recommends using a root certification authority (CA) that is included in the list of trusted root CA certificates pre-installed with the phone's software. If a custom root CA certificate (not included in the phone's software) is used to sign the provisioning server certificate, then the custom root CA certificate must be installed on the phone prior to connecting to the provisioning server.

If using client authentication as part of Mutual TLS, HP recommends using the phone's factory-install certificate, which is signed by the Polycom Root CA. If using a custom device certificate, then the custom device certificate must be installed, and additional configuration must be set to tell the phone to use the custom device certificate on the phone prior to connecting to the provisioning server.

Encrypted Configuration Files

This is not recommended.

Using an encryption SDK, you can encrypt configuration files on the provisioning server before transporting the files to the phone, which protects against malicious access to the provisioning server. Although encrypting configuration files is supported, HP does not recommend using this option due to the difficulties in managing and pre-staging the required decryption keys. If you choose to use this option, HP requires customers to contact an HP Poly Voice Solution Architect to access the Encryption SDK.

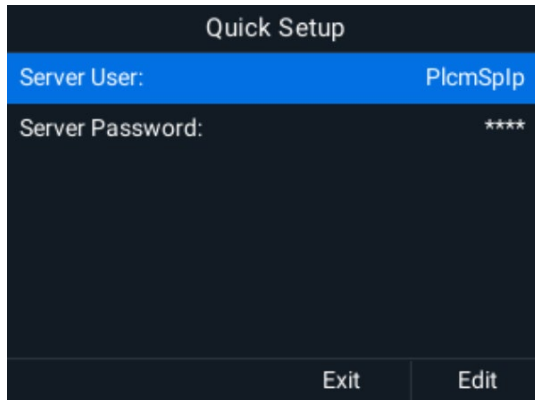
Quick Setup

The Quick Setup feature allows end users limited access to provisioning setup fields to fill in addresses and credentials.

Use the following parameters to configure this feature:

```
prov.quickSetup.enabled="1"  
prov.quickSetup.limitServerDetails="1"
```

In UC Software 5.4 and later versions, you can use the parameter `prov.quickSetup.limitServerDetails` to limit the displayed and editable fields to only a provisioning username and password.



Verify the Phone's Device Certificate Common Name

When validating a Poly phone device certificate, such as when the phone requests configuration updates, the validation of the *expiry* and *Root Issuing Authority* is insufficient as malicious attackers may have harvested fake certificate key pairs.

HP recommends limiting file names to those that match the phone's MAC address, which is used in the phone's offered certificate. This prevents access to any files created for other devices.

Example using Apache Directives:

```
<IfModule mod_ssl.c>  
  <VirtualHost _default_:443>  
    ServerName example.com  
    DocumentRoot /var/www/html  
  
    #- Verify the client CN against the request URI.  
    #- Allow only if the SSL_CLIENT_S_DN_CN (mac address) is a  
    part of the request URI  
    #- This prevents a device with HP Poly cert CN "XXXX" from  
    requesting a config for  
    #- phone "ZZZZ"  
    <Directory "/var/www/html/HP Poly/configuration/">  
      <If "%{REQUEST_URI} -strcmatch  
\"%{SSL_CLIENT_S_DN_CN}*"\">
```

```

        Require all granted
    </If>
    <Else>
Require all denied
    </Else>
</Directory>

```

Mutual TLS

Mutual TLS is a security protocol where both the client (the phone) and the server authenticate each other, using certificates, during a TLS handshake.

Root CA Certificates for Server Authentication

Poly phones are pre-installed with a list of trusted root CA certificates sourced from the most commonly encountered global CAs. This list is updated as new CAs are reported or created within the global PKI industry.

Refer to your product's Certificates Updates document for supported root certificates included on the phones at their respective support pages on [HP Support](#).

The supported root certificates are also listed in the phone's system web interface, under **Settings > Network > TLS > SSL Certificates**.

SSL Certificates					
Select Certificates Trusted root CA certificates in the CA bundle ▼					
Serial Number	Issuer	Issued To	Validity		
			From	To	
5D938D306736C8061D 1AC754846907			Oct 29 15:59:56 2008 GMT	Jan 1 00:00:00 2030 GMT	C S
5EC3B7A6437FA4E0	ACCVRAIZ1	ACCVRAIZ1	May 5 09:37:37 2011 GMT	Dec 31 09:37:37 2030 GMT	C S
570A119742C4E3CC	Actalis Authentication Root CA	Actalis Authentication Root CA	Sep 22 11:22:02 2011 GMT	Sep 22 11:22:02 2030 GMT	C S
7777062726A9B17C	AffirmTrust Commercial	AffirmTrust Commercial	Jan 29 14:06:06 2010 GMT	Dec 31 14:06:06 2030 GMT	C S
7C4504201CD4002D	AffirmTrust Networking	AffirmTrust Networking	Jan 29 14:08:24 2010	Dec 31 14:08:24 2030	C

Install and Store Root CA Certificates

You can install and store additional root CA certificates not included in the default CA bundle using the system web interface.

Type	Common Name(CA Fingerprint)	Install	
Platform CA 1	Polycom Root CA(79:E1:6B:8B:55:9A:AD:D3:1B:CD:C3:95:62:61:73:FF:57:9B)	Install	☐
Platform CA 2		Install	☐
* Auto Prov CA			☐
Application CA 1	ISRG Root X1(CA:BD:2A:79:A1:07:6A:31:F2:1D:25:36:35:CB:03:9D:43:29:A5:)	Install	☐

Supported CA Certificates

There are two types of certificates supported on Poly phones:

- **Platform certificates** - These certificates are stored on non-volatile flash memory and are available for use before the phone has connected to and received configuration from the provisioning server. Although platform certificates may be used for any secure application on the phone, they are primarily intended for use with 802.1X authentication and provisioning of configuration files. Up to two custom platform certificates are available.
- **Application certificates** - These certificates are stored in RAM and loaded after configuration has been received. As such, they are available to the wider variety of services that run after the initial network setup and configuration check is completed, such as SIP signaling, browser communications, presence, SOPI, and LDAP. Up to 7 custom application certificates are available.

Custom TLS CA Certificates

To add custom root CA certificates, set the PEM-encoded body of the certificate to the desired custom CA parameter:

Note: Keep the new lines in the certificate PEM-body strings. Include the BEGIN and END strings.

Platform CA 1 and 2 (X = 1 or 2)

```
device.set.TLS.customCaCertX.set="1"

device.sec.TLS.customCaCertX= "-----BEGIN CERTIFICATE-----
MIIFJDCCAwygAwIBAgIQRdu0k88f1rxI3/haFyBOqDANBgkqhkiG9w0BAQsFADAa
MRgwFgYDVQQDEw9Qb2x5Y29tIFJvb3QgQ0EwHhcNMMDkwMzA5MTgxMjQwWhcNNDQw
...
JZEJfBuJH6Su3TddP6cqjkh7tsgHYskxXJiq643AVycwuLOx1fgra5Yhr4d78V7S
P4vseMONTqXCDqAu4SwgeoRruqWczSYF
-----END CERTIFICATE-----"
```

Application CA 1 to 7 (where X = 1 to 7)

```
sec.TLS.customCaCert.X="-----BEGIN CERTIFICATE-----
MIIFazCCA1OgAwIBAgIRAIQz7DSQONZRGPGu2OCiwAwDQYJKoZIhvcNAQELBQAw
TzELMAkGA1UEBhMCVVMxKTAnBgNVBAoTIEludGVybmV0IFNlY3VyaXR5IFJlc2Vh
....
mRGunUHBcnWEvgJBQl9nJEiU0Zsnvgc/ubhPgXRR4Xq37Z0j4r7g1SgEEzwxA57d
emyPxgcYxn/eR44/KJ4EBs+lVDR3veyJm+kXQ99b21/+jh5Xos1AnX5iItreGCc=
-----END CERTIFICATE-----"
```

The default behavior on the Poly phones is to include all root CA certificates (pre-installed or custom) to verify the server's certificate. If only certain root CA certificates are used for verification, then you can configure different TLS applications with different TLS profiles, which control which root CA certificates are used for verification.

For Platform Profiles 1 and 2:

Parameter	Values
<code>device.sec.TLS.profile.caCertList1</code> <code>device.sec.TLS.profile.caCertList1.set="1"</code> <code>device.sec.TLS.profile.caCertList2</code> <code>device.sec.TLS.profile.caCertList2.set="1"</code>	Builtin, BuiltinAndPlatform1, BuiltinAndPlatform2, All, Platform1, Platform2, Platform3, Platform1AndPlatform2
Choose the CA certificate(s) to use for authentication:	
▪ The built-in default certificate ▪ The built-in and Custom #1 certificates ▪ The built-in and Custom #2 certificates ▪ Any certificate (built in, Custom #1 or Custom #2) ▪ Only the Custom #1 certificate ▪ Only the Custom #2 certificate ▪ Only the Custom #3 (SCEP) certificate ▪ Either the Custom #1 or Custom #2 certificate	

For Application Profiles 1 to 7 (X = 1 to 7):

Parameter	Values
<code>sec.TLS.profile.X.caCert.defaultList</code> <code>sec.TLS.profile.X.caCert.platform1</code> <code>sec.TLS.profile.X.caCert.platform2</code> <code>sec.TLS.profile.X.caCert.platform3</code> <code>sec.TLS.profile.X.caCert.application1</code> <code>sec.TLS.profile.X.caCert.application2</code> <code>sec.TLS.profile.X.caCert.application3</code> <code>sec.TLS.profile.X.caCert.application4</code> <code>sec.TLS.profile.X.caCert.application5</code> <code>sec.TLS.profile.X.caCert.application6</code> <code>sec.TLS.profile.X.caCert.application7</code>	0 or 1 (default)

Set to 1 to include that certificate to the list of root CAs to verify the server certificate.

Using Device Certificates

Poly phones are built with a unique RSA certificate/key pair, and they use a private “Polycom Root CA” certificate signing chain. Since 2009, Poly phones have been embedded with signed client certificates at the factory. Any provisioning server can verify this certificate to ensure the authenticity of the device making the query.

Below is an example of a factory-installed certificate chain on a Poly device:

```
Certificate chain
0 s:O = Polycom Inc., CN = 482567123456
  i:CN = Polycom Equipment Issuing CA 2
1 s:CN = Polycom Equipment Issuing CA 2
  i:CN = Polycom Equipment Policy CA
2 s:CN = Polycom Equipment Policy CA
  i:CN = Polycom Root CA
```

You can download the Polycom Root CA and intermediate CA certificates to use on servers to verify the certificate chain of the device certificate at <http://pki.polycom.com/pki>.

pki.polycom.com - /pki/

[\[To Parent Directory\]](#)

3/9/2009 2:52 PM	1603 Polycom Employee Issuing CA 1.crt
3/9/2009 3:06 PM	1603 Polycom Employee Issuing CA 2.crt
3/24/2009 4:31 PM	1558 Polycom Employee Policy CA.crt
3/9/2009 1:00 PM	1534 Polycom Equipment Issuing CA 1.crt
3/9/2009 1:21 PM	1534 Polycom Equipment Issuing CA 2.crt
3/9/2009 12:12 PM	1559 Polycom Equipment Policy CA.crt
3/9/2009 11:23 AM	1320 Polycom Root CA.crt

Note: Older Poly phone models may be signed by the “Polycom Equipment Issuing CA 1” intermediate certification authority. If the server requires the full certificate chain, make sure to include the “Polycom Equipment Issuing CA 1” intermediate certificate in the server’s configuration.

Using Custom Device Certificates

In addition to the factory-installed device certificates, you can also store additional device certificates on Poly phones to use for different TLS applications like 802.1X, provisioning, SIP, browser, syslog, etc.

Certificate Configuration

CA Certificates

Device Certificates

Type	Common Name(CA Fingerprint)		
Platform Credential 1	USLCRCZ893(14:88:32:FA:D5:D6:5E:2B:89:78:F9:2C:8C:71:2D:2F:A7:08:E0:)	Install	☐
Platform Credential 2		Install	☐
* Auto Prov Credential			☐
Application Credential 1	USLCRCZ893(14:88:32:FA:D5:D6:5E:2B:89:78:F9:2C:8C:71:2D:2F:A7:08:E0:)	Install	☐
Application Credential 2		Install	☐
Application Credential 3		Install	☐

Note: For applications like 802.1X, syslog, and provisioning, you can only use the device certificates set in the “Platform Credential X” and “Auto Prov Credential” slots. All other applications (like SIP, browser, presence, LDAP, etc.) can use either “Platform Credential X” or “Application Credential X” slots.

Device certificate credentials are defined in the TLS profile and assigned to the different TLS applications.

Application	Profile Name	Minimum TLS Version
* 802.1x	TLS Platform Profile 1	TLS 1.0
Syslog	TLS Platform Profile 1	TLS 1.0
Provisioning	TLS Platform Profile 1	TLS 1.0
SIP	TLS Platform Profile 1	TLS 1.2
Web Services	TLS Platform Profile 1	TLS 1.2
Presence	TLS Platform Profile 1	TLS 1.2
LDAP	TLS Platform Profile 1	TLS 1.2
SOPI	TLS Platform Profile 1	TLS 1.2
* Web Server	Web Server Profile	TLS 1.2
Exchange Services	Exchange Service Profile	TLS 1.2

For example, the default configuration for all applications uses “TLS Platform Profile 1”, which uses the “Polycom” device credentials, the factory-install device certificate.

Profile Name	Cipher Suite		Certificate	
	Type	Configuration	CA Certificate	Device Credentials
Platform Profile 1	Default	ECDHE-ECDSA-AES128-GCM-SHA256	All Certificates	Polycom

Installing Platform Certificates

You can install Platform certificates using one of the following methods:

- **Configuration files.** You must enter the certificate and private key in PEM format. In the following parameters, X = 1 or 2.

```

Device.sec.TLS.customDeviceCertX.set="1"

device.sec.TLS.customDeviceCertX.publicCert="-----BEGIN CERTIFICATE--
---
MDc1RDCCAiIwDQYJKoZIhvcNAQEBBQADggIPADCCAgCggIBAN9e0DPnIKfDdBTR
+nYK6l5sW0X6W+ygsUdeclsK0+VdEApWSddYnhzfKSRTkwqYRGnxcCzEVqRNcO7c
...
b5Hac0YZQq9NLPJYjsZRQ8iAxCkwFdqu2jQ398aBMlexPvy6SYHheIkju7lgiFB6
-----END CERTIFICATE----- "
device.sec.TLS.customDeviceCertX.privateKey="-----BEGIN RSA
PRIVATE KEY-----
MIIJKAIBAAKCAgEA317QM+cgp8N0FNH6dgrqXmxbRfpb7KCxR15yWwrT5V0QC1ZJ
11ieHN8pJFOTCphEafFwLMRWpElw7txvkdpzRh1Cr00s8liOx1FDyIDEKTAV2q7a
Ndf3xoEzV7E+/LpJgeF4iSO7vWCIUhr/0lGW0TpLzwrpV/cdShMuxyfqWS+Gh8+c
...
/T42HW3TMSYlXiv6Hr2wvopq+EiRnMDJI1ImA8h7wz6BmXXwpFIBzIbqQwXpxjlJ
-----END RSA PRIVATE KEY-----"

```

Note: There aren't any individual "set" parameters for the "publicCert" and "privateKey" device parameters. The `device.sec.TLS.customDeviceCertX.set` parameter is used to handle installing the public cert and private key.

Note: Keep the new lines in the certificate PEM-body strings. Include the BEGIN and END strings.

- **Phone user interface.** You must enter the certificate as a single file with the PEM formatted certificate or PKCS#7 certificate chain and private key concatenated together.
- **System web interface.** You must enter the certificate as a single file with the PEM formatted certificate or PKCS#7 certificate chain and private key concatenated together.

Below is an example of a certificate and private key concatenated together.

```

-----BEGIN CERTIFICATE-----
MDc1RDCCAiIwDQYJKoZIhvcNAQEBBQADggIPADCCAgCggIBAN9e0DPnIKfDdBTR
+nYK6l5sW0X6W+ygsUdeclsK0+VdEApWSddYnhzfKSRTkwqYRGnxcCzEVqRNcO7c
...
b5Hac0YZQq9NLPJYjsZRQ8iAxCkwFdqu2jQ398aBMlexPvy6SYHheIkju7lgiFB6
-----END CERTIFICATE-----
-----BEGIN RSA PRIVATE KEY-----
MIIJKAIBAAKCAgEA317QM+cgp8N0FNH6dgrqXmxbRfpb7KCxR15yWwrT5V0QC1ZJ
11ieHN8pJFOTCphEafFwLMRWpElw7txvkdpzRh1Cr00s8liOx1FDyIDEKTAV2q7a
Ndf3xoEzV7E+/LpJgeF4iSO7vWCIUhr/0lGW0TpLzwrpV/cdShMuxyfqWS+Gh8+c
...
/T42HW3TMSYlXiv6Hr2wvopq+EiRnMDJI1ImA8h7wz6BmXXwpFIBzIbqQwXpxjlJ
-----END RSA PRIVATE KEY-----

```

Installing Application Certificates

You can install Application certificates using the same methods as Platform certificates. In the following parameters, **X** = 1 to 7:

```
sec.TLS.customDeviceCert.X="-----BEGIN CERTIFICATE-----
MDc1RDCCAiIwDQYJKoZIhvcNAQEBBQADggIPADCCAgCggIBAN9e0DPnIKfDdBTR
+nYK6l5sW0X6W+ygsUdeclsk0+VdEApWSddYnhzfKSRTkwqYRGnxcCzEVqRNcO7c
...
b5Hac0YZQq9NLPJYjsZRQ8iAxCkWFdqu2jQ398aBMlexPvy6SYHheIkju7lgiFB6
-----END CERTIFICATE----- "
sec.TLS.customDeviceKey.X="-----BEGIN RSA PRIVATE KEY-----
MIIJKAIBAAKCAgEA317QM+cgp8N0FNH6dgrqXmxbRfpb7KCxR15yWwrT5V0QC1ZJ
11ieHN8pJFOTCphEafFwLMRWpE1w7txvkdPzRh1Cr00s8liOx1FDyIDEKTAV2q7a
NDf3xoEzV7E+/LpJgeF4iSO7vWCIUhr/0lGW0TpLzwrpV/cdShMuxyfqWS+Gh8+c
...
/T42HW3TMSYlXiv6Hr2wvopq+EiRnMDJI1ImA8h7wz6BmXXwpFIBzIbqQwXpxjlJ
-----END RSA PRIVATE KEY-----"
```

Set Device Credentials for TLS Profiles

Use the parameters in the following sections to set the device credentials for each TLS Profile:

Platform Profile 1

```
device.sec.TLS.profile.deviceCert1.set="1"
device.sec.TLS.profile.deviceCert1="<PlatformDevCred>"
```

Where *<PlatformDevCred>* is one of the following: Builtin, Platform1, Platform2, Platform3

Platform Profile 2

```
device.sec.TLS.profile.deviceCert2.set="1"
device.sec.TLS.profile.deviceCert2="<PlatformDevCred>"
```

Where *<PlatformDevCred>* is one of the following: Builtin, Platform1, Platform2, Platform3

Application Profile 1-7 (X = 1 to 7)

```
sec.TLS.profile.X.deviceCert="<ApplicationDevCred>"
```

Where *<ApplicationDevCred>* is one of the following: Polycom, Platform1, Platform2, Platform3, Application1, Application2, Application3, Application4, Application5, Application6, Application7

TLS Profile for TLS Applications:

```
sec.TLS.profileSelection.SIP="<TLSProfile>"
sec.TLS.profileSelection.LDAP="<TLSProfile>"
sec.TLS.profileSelection.SOPI="<TLSProfile>"
sec.TLS.profileSelection.XMPP="<TLSProfile>"
sec.TLS.profileSelection.browser="<TLSProfile>"
```

Where *<TLSProfile>* can be: PlatformProfile1, PlatformProfile2, PlatformProfile3, ApplicationProfile1, ApplicationProfile2, ApplicationProfile3, ApplicationProfile4, ApplicationProfile5, ApplicationProfile6, ApplicationProfile7

TLS Profile for TLS Platform Profiles

```
device.sec.TLS.profileSelection.provisioning="<TLSPlatProfile>"
device.sec.TLS.profileSelection.provisioning.set="1"
device.sec.TLS.profileSelection.syslog="<TLSPlatProfile>"
device.sec.TLS.profileSelection.syslog.set="1"
device.sec.TLS.profileSelection.dot1x="<TLSPlatProfile>"
device.sec.TLS.profileSelection.dot1x.set="1"
```

Where *<TLSPlatProfile>* can be: PlatformProfile1, PlatformProfile2, PlatformProfile3

Software Update

Reset Alternate Software Upgrade Servers

If you want to remove an alternate software version that was installed using a HP Poly cloud service, use the following parameters to remove that server location, which restricts the allowed software version to only that which is provided by your own configuration server.

```
Device.prov.upgradeServer.set="1"
device.prov.upgradeServer=""
upgrade.custom.server.url=""
```

Software Update Rollback Prevention

In PVOS 8.3.0 or later (Edge E) or 9.1.0 or later (CCX series/Trio C60), you can prevent Poly phones from downgrading (rolling back) to earlier versions using the following parameters.

```
Device.prov.rollbackProtection.set="1"
device.prov.rollbackProtection="Enabled"
```

Permitted values include Auto, Enabled, Disabled. The default value of "Auto" means that for phones running the Microsoft Teams base profile (CCX), rollback protection is enabled by default. For all other base profiles, rollback protection is disabled by default. If set to "Enabled" the phone will not permit a downgrade to any earlier release, regardless of base profile. If set to "Disabled" then the phones can downgrade to any compatible software release.

Simple Certificate Enrollment Protocol (SCEP)

Use the following parameters to configure the Simple Certificate Enrollment Protocol to automatically and securely provision phones with a digital device certificate. This feature streamlines the certificate enrollment process for large numbers of deployed phones.

Use the following parameter to enable the SCEP feature.

```
SCEP.enable="1"
```

Use the following parameters to specify the URL address of the SCEP server accepting requests to obtain a certificate.

```
SCEP.url="<SCEP server address>"Specify the user name and password to
authenticate with the SCEP server.
SCEP.http.username="<SCEP username>"
SCEP.http.password="<SCEP password>"
```

Use the following parameter to specify the challenge password to send with the CSR when requesting a certificate.

```
SCEP.challengePassword="<challenge password>"
```

Use the following parameters to specify the following information for the CSR when requesting a certificate.

```
SCEP.csr.commonName="<common name>"
```

Tip: In bulk provisioning scenarios, if setting the device's MAC address as the CSR's common name, set this parameter to an empty value- `SCEP.csr.commonName=""`

```
SCEP.csr.country="<country name>"
SCEP.csr.email="<email address>"
SCEP.csr.locality="<locality name>"
SCEP.csr.organization="<organization name>"
SCEP.csr.organizationUnit"<organization unit name>"
SCEP.csr.state="<state name>"
```

Use the following parameter to configure the CA certificate fingerprint to confirm the authenticity of the CA response during enrollment.

```
SCEP.CAFingerprint="<CA certificate fingerprint>"
```

Use the following parameter to specify the number of times to poll the SCEP server when the SCEP server returns a Certificate Enrollment Response Message with **pkiStatus** set to pending. The default is 12. The value range is 1 to 24.

```
SCEP.certPoll.retryCount="<retry count>"
```

Use the following parameter to specify the time, in seconds, to wait between poll attempts when the SCEP server returns a Certificate Enrollment Response Message with **pkiStatus** set to pending. The default is 300 (5 minutes). The value range is 300 to 3600.

```
SCEP.certPoll.retryInterval="<number of seconds>"
```

Use the following parameter to specify the time interval, in hours, to retry certificate renewal. The default is 86400 (24 hours). The value range is 28800 to 259200 (8 to 72 hours).

```
SCEP.certRenewalRetryInterval="<retry count>"
```

Use the following parameter to specify the percentage of the certificate validity percentage threshold to initiate a renewal. The default is 80. The value range is 50 to 100.

```
SCEP.certRenewalThreshold="<percent validity>"
```

Use the following parameter to specify the number of times to retry the enrollment process in case of enrollment failure. The default is 12. The value range is 1 to 24.

```
SCEP.enrollment.retryCount="<retry count>"
```

Use the following parameter to specify the time interval, in seconds, to retry the enrollment process in case of enrollment failure. The default is 300 (5 minutes). The value range is 300 to 3600.

```
SCEP.enrollment.retryInterval="<number of seconds>"
```

Use the following parameter to specify the message type a phone uses in certificate renewal requests. The default is 1. Set the parameter to 0 if the phone requires interoperability with specific servers, such as the Microsoft Device Enrollment Service (NDES). If set to 0, the phone uses the PKCSReq message type in certificate renewal requests.

```
SCEP.useRenewalReqMessageType="1"
```

802.1X Access Port Security

The 802.1X authentication feature provides authentication services for higher security networks that use 802.1X as the authentication protocol. Poly phones support seven EAP protocols for 802.1X authentication. You can configure the 802.1X authentication feature using a central provisioning server, the system web interface, or the phone's keypad interface.

802.1X Acronyms

The following acronyms are used in this document:

EAP	Extensible Authentication Protocol
TLS	Transport Layer Security
PEAP	Protected Extensible Authentication Protocol
TTLS	Tunneled Transport Layer Security
FAST	Flexible Authentication via Secure Tunneling
MD5	Message-Digest Algorithm
MS-CHAPv2	Microsoft Challenge-Handshake Authentication Protocol (version 2)
GTC	Generic Token Card
IEEE	Institute of Electrical and Electronics Engineers
LAN	Local Area Network
WLAN	Wireless Local Area Network
EAPOL	EAP over LAN (Extensible Authentication Protocol over Local Area Network)
PAC	Protected Access Credential
MDA	Multi-Domain Authentication

Supported EAP Authentication Protocols

Poly phones support the following authentication protocols.

- EAP-TLS
- EAP-PEAPv0/MSCHAPv2
- EAP-PEAPv0/GTC
- EAP-TTLS/EAP-MSCHAPv2
- EAP-TTLS/EAP-GTC
- EAP-FAST
- EAP-MD5

EAP Authentication Requirements

You can authenticate Poly phones in 802.1X environments using each of the following supported EAP protocols. Each authentication protocol has a unique configuration.

EAP-TLS

- Device certificate
- Trusted pool of root/CA certificates
- Identity (user name)

EAP-PEAPv0/EAP-MSCHAPv2 or EAP-PEAPv0/EAP-GTC

- Trusted pool of root/CA certificates
- Identity (user name)
- Password

EAP-TTLS/EAP-MSCHAPv2 or EAP-TTLS/EAP-GTC

- Trusted pool of root/CA certificates
- Identity (user name)
- Password

EAP-MD5

- Identity (user name)
- Password

EAP-FAST

- Identity (user name)

- Password
- Optional PAC file, provisioned automatically through the network or manually using a PAC file password.

Note: Using EAP-FAST Authentication for the First Time

The first time you perform EAP-FAST dynamic PAC file provisioning (also known as in-band provisioning), the server will provision the phone with a PAC file and the 802.1X authentication will fail. This is followed by a successful 802.1X authentication. In some cases, the network switch may impose a delay of about 60 seconds before initiating the 802.1X authentication following a failed authentication attempt.

Note: Using EAP-FAST Authentication with a Network Switch in MDA Mode

If you are using a network switch in MDA mode, be aware of the following:

- MDA does not enforce the order of device authentication; however, when using an MDA-enabled port, HP recommends authenticating your voice device before a data device.
- When a network switch detects a data or voice device on a port, the switch blocks the device's MAC address until authorization succeeds. If authorization fails, there will be a delay, depending on the network switch setup, before the phone can authenticate.

Configuring 802.1X Authentication

You can configure 802.1X authentication in the following ways:

- Using a central provisioning server
- Using the system web interface
- Using the local phone user interface

If setting up more than 10 phones, HP recommends using a central provisioning server. If provisioning fewer than 10 phones, you can use the system web interface or the phone's user interface to configure 802.1x authentication.

Configuring 802.1X Using a Central Provisioning Server

The following sections outline TLS profile configuration and 802.1X setup. Each EAP protocol requires a slightly different configuration:

- If using EAP-TLS, EAP-PEAP, or EAP-TTLS, go to the "Configuring Your TLS Profile" section then go to the "Setting Up 802.1X" section.
- If using EAP-FAST or EAP-MD5, go directly to the "Setting Up 802.1X" section.

Refer to "EAP Authentication Requirements" section in this document for a list of the parameters for configuring each authentication protocol.

You can locate all the configuration parameters shown in the following sections in the **device.cfg** configuration file template located in the Config folder of your Poly software package. HP recommends you make a copy of **device.cfg** and edit the parameters directly or create a new configuration file containing only the parameters you wish to modify.

Configuring Your TLS Profile

Only EAP-TLS, EAP-PEAP, and EAP-TTLS protocols require a TLS Profile. Configure either TLS Platform Profile 1 or TLS Platform Profile 2 for these authentication protocols.

Choose the parameters ending in 1 to configure TLS Platform Profile 1 (for example, **device.sec.TLS.profile.caCertList1**) or choose the parameters ending in 2 to configure TLS Platform Profile 2 (for example, **device.sec.TLS.profile.caCertList2**). You must then specify which Platform Profile you have configured by setting the **device.sec.TLS.profile.profileSelection.dot1x** parameter shown in the TLS Profile Configuration Parameters table to **TLS Platform Profile 1** or **TLS Platform Profile 2**.

TLS Profile Configuration Parameters

Parameter	Values
device.sec.TLS.profile.caCertList1 device.sec.TLS.profile.caCertList1.set="1" device.sec.TLS.profile.caCertList2 device.sec.TLS.profile.caCertList2.set="1" Choose the CA certificate(s) to use for authentication: ▪ The built-in default certificate ▪ The built-in and Custom #1 certificates ▪ The built-in and Custom #2 certificates ▪ Any certificate (built in, Custom #1 or Custom #2) ▪ Only the Custom #1 certificate ▪ Only the Custom #2 certificate ▪ Either the Custom #1 or Custom #2 certificate	Builtin, BuiltinAndPlatform1, BuiltinAndPlatform2, All, Platform1, Platform2, Platform3, Platform1AndPlatform2
device.sec.TLS.profile.cipherSuite1 device.sec.TLS.profile.cipherSuite1.set="1" device.sec.TLS.profile.cipherSuite2 device.sec.TLS.profile.cipherSuite2.set="1" The cipher suite to use for the Platform Profile.	String
device.sec.TLS.profile.cipherSuiteDefault1 device.sec.TLS.profile.cipherSuiteDefault1.set="1" device.sec.TLS.profile.cipherSuiteDefault2 device.sec.TLS.profile.cipherSuiteDefault2.set="1" If set to 1, the default cipher suite will be used. If set to 0, the custom cipher suite will be used.	0 or 1

device.sec.TLS.profile.customCaCert1 device.sec.TLS.profile.customCaCert1.set="1" device.sec.TLS.profile.customCaCert2 device.sec.TLS.profile.customCaCert2.set="1" The custom certificate to use if device.sec.TLS.profile.caCertList is configured to use a custom certificate.	String
device.sec.TLS.profile.deviceCert1 device.sec.TLS.profile.deviceCert1.set="1" device.sec.TLS.profile.deviceCert2 device.sec.TLS.profile.deviceCert2.set="1" Choose the device certificate(s) to use for authentication.	Builtin, BuiltinAndPlatform1, BuiltinAndPlatform2, All, Platform1, Platform2, Platform3, Platform1AndPlatform2
device.sec.TLS.profile.profileSelection.dot1x device.sec.TLS.profile.profileSelection.dot1x.set="1" Choose the TLS Platform Profile that you have configured.	PlatformProfile1, PlatformProfile2, PlatformProfile3

After configuring your TLS Profile for EAP-TLS, EAP-PEAP, or EAP-TTLS protocols, go to "Setting Up 802.1X."

Setting Up 802.1X

Before configuring the EAP-TLS, EAP-PEAP, and EAP-TTLS protocols, you must first configure your certificates by setting up a TLS Profile (see Configuring Your TLS Profile). To set up 802.1X authentication, configure the parameters in the 802.1X Setup Parameters table.

802.1X Setup Parameters

Parameter	Values
device.net.dot1x.enable device.net.dot1x.enable.set="1" Enable 802.1X authentication.	1
device.net.dot1x.method device.net.dot1x.method.set="1" Specify the 802.1X authentication method where the numbers 0 to 7 refer to the following protocols: 0:None, 1:EAP-TLS, 2:EAP-PEAPv0-MSCHAPv2, 3:EAP-PEAPv0-GTC, 4:EAP-TTLS-MSCHAPv2, 5:EAP-TTLS-GTC, 6:EAP-FAST, 7:EAP-MD5	0, 1, 2, 3, 4, 5, 6, or 7 or the method option (like "EAP-TLS")
device.net.dot1x.identity device.net.dot1x.identity.set="1" The identity (user name) for authentication.	String
device.net.dot1x.password device.net.dot1x.password.set="1" The password for 802.1X authentication. This parameter is required for all methods except EAP-TLS.	String

device.net.dot1x.anonid device.net.dot1x.anonid.set="1"	string
EAP-TTLS and EAP-FAST only. The anonymous identity (user name).	
device.net.dot1x.eapFastInBandProv device.net.dot1x.eapFastInBandProv.set="1"	0 or 1
EAP-FAST only, optional. Choose 1 to enable EAP In-Band Provisioning by server unauthenticated PAC provisioning using anonymous Diffie-Hellman key exchange. Choose 0 to disable EAP In-Band Provisioning.	
device.pacfile.data device.pacfile.data.set="1"	string
EAP-FAST only, optional. The PAC file (base 64 encoded). To generate a base 64-encoded PAC file, generate the PAC file using your authentication server and then convert it to base 64. You can convert the file to base 64 using the following openssl commands: \$ openssl enc -base64 -in myfile -out myfile.b64	
device.pacfile.password device.pacfile.password.set="1"	string
EAP-FAST only, optional. The password for the PAC file	

Applying the Configuration Files to your Phone

After configuring 802.1X authentication protocols and profiles, upload the files to your phone.

To upload the configuration files to your phone:

1. Connect your phone to a staging network (a network that is not 802.1X-enabled).
2. Upload the configuration files to the phone. For more information on uploading configuration files to your phone, consult your phone's administrator guide on HP support <https://support.hp.com/us-en/poly>
3. Reboot the phone. After the phone reboots, it will be ready to connect to the 802.1X-enabled network.
4. Connect the phone to the 802.1X-enabled network and reboot the phone.
5. Verify that your phone is authenticated by making a phone call.

Example Configuration Settings

Use the following parameters to enable 802.1X authentication:

```
device.net.dot1x.enabled="1"
device.net.dot1x.enabled.set="1"
```

Use the following parameters to set the username and password for 802.1X authentication. You don't need to set a password if you set EAP-TLS for the 802.1X EAP method.

```
Device.net.dot1x.identity="<username>"
```

```
device.net.dot1x.identity.set="1"
device.net.dot1x.password="<password>"
device.net.dot1x.password.set="1"
```

Use the following parameters to set an 802.1X EAP method.

```
Device.net.dot1x.method="<EAP method>"
device.net.dot1x.method.set="1"
```

Permitted values for device.net.dot1x.method:

- "0" or "EAP-None" – No authentication
- "1" or "EAP-TLS"
- "2" or "EAP-PEAPv0-MSCHAPv2"
- "3" or "EAP-PEAPv0-GTC"
- "4" or "EAP-TTLS-MSCHAPv2"
- "5" or "EAP-TTLS-GTC"
- "6" or "EAP-FAST"
- "7" or "EAP-MD5"

If you set the 802.1X method as EAP-FAST, you can set the following parameters as well:

```
device.net.dot1x.eapFastInBandProv="<0> or <1>"
device.net.dot1x.eapFastInBandProv.set="1"
device.pacfile.data="<Optional PAC file name>"
device.pacfile.data.set="1"
device.pacfile.password="<Password for PAC file if needed>"
device.pacfile.password.set="1"
```

802.1X and the PC Port

Poly phones act as a proxy for any device connected to the PC Port by forwarding EAPOL logon/logoff messages to the Authenticator, which prevents the device from joining a secure network because the phone has been authenticated and authorized to do so.

Use the parameters in the following table to control the PC port authentication behavior:

802.1X Setup Parameters

Parameter	Values
sec.dot1x.eapollogoff.enabled When enabled, whenever a device connected to the PC port of the phone is disconnected, the phone will send an EAPOL logoff message on its behalf.	0 (default), 1
sec.dot1x.eapollogoff.lanlinkreset When enabled, the phone will cycle the LAN port link status if the UC Software application is restarted. When disabled, there will be no interruption of the LAN port link status when the phones application	0 (default), 1

restarts. By cycling the LAN port link status, this will often trigger the upstream authenticator to remove any authentication records for both the phone and any connected PC forcing both to restart 802.1X authentication again. The default setting will allow any supplicant on the PC port of the phone to receive no interruption in service if the phone should reboot.

`sec.hostmovedetect.cdp.enabled`

0 (default), 1

If set to 1 (enabled), the phone will send periodic reports for Cisco's Second Port Disconnect feature (aka "Host Movement Detection").

`sec.hostmovedetect.cdp.sleepTime`

1000 (default), 0-60000

Controls the frequency between two consecutive link-up state change reports. If `sec.hostmovedetect.cdp.enabled` is set to 1, there is an x microsecond time interval between two consecutive link-up state change reports, which reduces the frequency of dispatching CDP packets.

Troubleshooting Note: What if my Phone Doesn't Authenticate?

If your phone does not authenticate, on the phone, navigate to **Menu > Status > Platform > Configuration** and check for errors in your configuration files. If you see the message "Errors Found" instead of "Parameters Accepted" for one or more of the files, verify the parameters in the file.

Configuring 802.1X Using the System Web Interface

You can configure the 802.1X authentication parameters using the phone's system web interface. Refer to the "Configuring 802.1X Using a Central Provisioning Server" section for more information on the 802.1X configuration parameters.

To set up a TLS Profile:

1. Connect your phone to a staging network (a network that is not 802.1X-enabled).
2. Launch the system web interface by navigating to `http://<phoneIPAddress>` in a web browser. Log in using your administrator credentials.
3. Navigate to **Settings > Network > TLS**.
4. Expand the **Certificate Configuration** menu and install the required certificates.
5. Expand the **TLS Profiles** menu and configure either **Platform Profile 1** or **Platform Profile 2**.
6. Expand the **TLS Applications** and choose the **Platform Profile** that you configured (either TLS Platform Profile 1 or TLS Platform Profile 2) from the drop-down list next to the **802.1X** label.
7. Click **Save** at the bottom of the page. Your phone will reboot or restart.

To enable 802.1X authentication:

1. Launch the system web interface by navigating to `http://<phoneIPAddress>`.

2. Navigate to **Settings > Network > Ethernet**.
3. Expand the **Ethernet 802.1X** menu and configure the settings as described in **802.1X Setup Parameters** table.
4. To configure EAP-FAST with a PAC file, expand **PAC File Info** and install the PAC file (base 64 encoded)

Configuring 802.1X Using the Local Phone User Interface

You can configure the 802.1X authentication parameters using your phone's user interface. Refer to "Configuring 802.1X Using a Central Provisioning Server" section for more information the 802.1X configuration parameters.

To set up a TLS Profile:

1. On the phone, go to **Menu > Advanced > Admin Settings > TLS Security**.
2. Select **Custom CA Certificates** to configure your CA Certificates or select **Custom Device Credentials** to configure the Device Credentials.
3. From the **TLS Security** menu, select **Configure TLS Profiles** and choose either **TLS Platform Profile 1** or **TLS Platform Profile 2**.
4. Configure the profile as shown in TLS Profile Configuration Parameters table
5. From the **TLS Security** menu, select **TLS Applications > 802.1X**.
6. Select the **TLS Platform Profile** that you configured (either **TLS Platform Profile 1** or **TLS Platform Profile 2**).
7. Save the configuration. The phone will reboot.

To enable 802.1X Authentication:

1. On the phone, go to **Menu > Advanced > Admin Settings > Network Settings > Ethernet Menu**.
2. Scroll down to **802.1X Auth** and select **Enabled**.
3. From the **Ethernet** menu, select **802.1X Menu**.

See the 802.1X Setup Parameters table for the list of parameters to configure. You can also configure PAC file parameters for EAP-FAST from the **802.1X** menu by selecting **PAC File Info**. The PAC file must be base 64 encoded.

SSL/TLS Protocol Selection

Restricting TLS Versions

In UC Software 5.6.0 and later for VVX series and UC Software 5.5.3 "AA" and later for Trio C60, you can set the TLS protocol used for various applications by selecting one of the following modes:

TLSv1_0 (The default is TLSv1_0, which allows the phone to negotiate TLS 1.0 or higher.)

- TLSv1_1
- TLSv1_2
- SSLv2v3

In PVOS 9.2.0 (CCX/Trio) and PVOS 8.3.0 (Edge E), you can select TLSv1_3 as the TLS protocol.

Note: In UC Software 6.4.7 (VVX), PVOS 9.0.0 (CCX/Trio), and PVOS 8.2.0 (Edge E), the default value for the minimum TLS version for the SIP, Web Services, LDAP, SOPI, Web Server, and Exchange Services applications is TLSv1_2.

Note: In PVOS 9.2.0 (CCX/Trio), and PVOS 8.3.0 (Edge E), the default value for the minimum TLS version for all applications (including 802.1X, Syslog, and Provisioning) is TLSv1_2.

Configuring TLS Application Protocols

Use the following parameters to change the negotiated protocol for each TLS application:

```
sec.TLS.protocol.sip="TLSv1_2"
sec.TLS.protocol.ldap="TLSv1_2"
sec.TLS.protocol.sopi="TLSv1_2"
sec.TLS.protocol.webServer="TLSv1_2"
sec.TLS.protocol.xmpp="TLSv1_2"
sec.TLS.protocol.exchangeService="TLSv1_2"
device.sec.TLS.protocol.syslog="TLSv1_2"
device.sec.TLS.protocol.syslog.set="1"
device.sec.TLS.protocol.prov="TLSv1_2"
device.sec.TLS.protocol.prov.set="1"
device.sec.TLS.protocol.dot1x="TLSv1_2"
device.sec.TLS.protocol.dot1x.set="1"
```

Poodle Security Exploit – Deprecating SSLv2 & v3

In UC Software versions 4.0.8, 4.1.9, 5.1.4, 5.2.2, and 5.3.0, but prior to version 5.6.0, TLS protocols SSLv2 and SSLv3 are disabled by default to mitigate the POODLE exploit.

Use the following parameters to enable the TLS cipher suites:

```
device.sec.TLS.SSLv2v3.enabled.set="1"
device.sec.TLS.SSLv2v3.enabled="1"
```

Cipher Usage

Customizing TLS Cipher Usage

TLS specifies the protocol for establishing a secure session and makes use of cipher suite negotiation to determine how to encrypt or decrypt the data used in that session. It is important to

recognize that supporting the TLS 1.2 protocol, as all Poly phones do, does not inherently mean that you can use older and less secure ciphers in negotiation as many systems default to compatibility mode rather than a narrow field for the purpose of security.

To limit the ciphers offered or accepted by the phones, we recommend tailoring your chosen ciphers to match your needs.

Cipher control strings are defined by the OpenSSL documentation described at:

<http://www.openssl.org/docs/apps/ciphers.html>

In UC Software 6.4.7 (VVX), PVOS 9.0.0 (CCX/Trio), and PVOS 8.2.1 (Edge E), the default cipher list is:

```
sec.TLS.cipherList="ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-  
SHA256:ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384:ECDHE-  
ECDSA-CHACHA20-POLY1305:ECDHE-RSA-CHACHA20-POLY1305:DHE-RSA-AES128-GCM-  
SHA256:DHE-RSA-AES256-GCM-SHA384:DHE-RSA-CHACHA20-POLY1305"
```

Note: In software releases that support TLS 1.3, when TLS 1.3 is used, the cipher list is restricted to what TLS 1.3 allows, and the above configuration is ignored.

Web Server - System Web Interface

In UC Software 6.4.7 (VVX), PVOS 9.0.0 (CCX/Trio), and PVOS 8.2.1 (Edge E), the default cipher list for the system web interface is:

```
sec.TLS.webserver.cipherList="ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-  
AES128-GCM-SHA256:ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-  
SHA384:ECDHE-ECDSA-CHACHA20-POLY1305:ECDHE-RSA-CHACHA20-POLY1305:DHE-  
RSA-AES128-GCM-SHA256:DHE-RSA-AES256-GCM-SHA384:DHE-RSA-CHACHA20-  
POLY1305"
```

Note: In software releases that support TLS 1.3, when TLS 1.3 is used, the cipher list is restricted to what TLS 1.3 allows, and the above configuration is ignored.

FIPS Mode

In PVOS 8.3.0 and 9.2.0, Poly phones support FIPS 140-3 compatibility, and FIPS 140-3 is enabled by default. In earlier versions, you can configure your phone to use FIPS 140-2 compliant cryptography.

Use the following parameters to configure FIPS compliant cryptography:

```
device.sec.TLS.FIPS.enabled.set="1"  
device.sec.TLS.FIPS.enabled="1"
```

.

The following are important notes, restrictions, and security changes required for FIPS 140-3:

- SSL v2/v3 and TLS v1.0/1.1 are no longer available
- RSA, DSA and DH keys shorter than 2048 bits are prohibited
- ECC keys shorter than 224 bits are prohibited
- Cipher suites using RC4 are prohibited
- Cipher suites using SHA1 MAC algorithms are prohibited
- All export cipher suites are prohibited
- 3DES is prohibited
- Certificate signatures using SHA1 and MD5 are prohibited

Security Bulletins

All device security bulletins are posted on the HP Support site for each product -

<https://support.hp.com/us-en/poly>

Navigate to the product page, and click on the “Bulletins and Alerts” link:

The screenshot shows the HP Support website for the CCX IP 505 series. A green arrow points to the 'Bulletins and Alerts' link. Below, a table lists security bulletins and alerts for the CCX IP 505 series.

Title	Severity	Category	Updated on
Certain Poly Trio & CCX Devices – Potential Remote Code Execution	High	Security Bulletin	Jun 24, 2025
Some Poly phones may sign out after migration to Microsoft AOSP	---	Customer Advisory	Jun 11, 2025
Advance Notice of Changes to Manufacturing Installed Certificates (MIC)	---	Customer Advisory	Jun 6, 2025
CCX Series Shipping Configuration (May 2025)	---	Customer Advisory	May 19, 2025