

Konfigurieren von SSH mit x509-Authentifizierung auf IOS-Geräten

Inhalt

[Einführung](#)

[Voraussetzungen](#)

[Anforderungen](#)

[Verwendete Komponenten](#)

[Konfigurieren](#)

[Netzwerkdigramm](#)

[Überlegungen zur Bereitstellung](#)

[Konfigurationen](#)

[\(Optional\) Integration mit TACACS Server](#)

[Überprüfen](#)

[Fehlerbehebung](#)

[Weitere Informationen](#)

Einführung

In diesem Dokument wird beschrieben, wie der SSH-Server mit der Verwendung von x509v3-Zertifikaten auf IOS-Geräten gemäß RFC6187 konfiguriert wird.

Das Secure Shell Protocol (SSH) bietet gegenseitige Authentifizierung, d. h. Client und Server werden authentifiziert. Traditionell verwendet der Server die private und öffentliche RSA-Tastatur für die Authentifizierung. Der SSH-Client berechnet die Prüfsumme des öffentlichen Schlüssels und fragt den Administrator, ob er vertrauenswürdig ist. Der Administrator sollte den öffentlichen Schlüssel mithilfe einer Out-of-Band-Methode vom Router exportieren und die Werte vergleichen. In der Praxis ist dies eine umständliche Methode, und oft wird der öffentliche Schlüssel ohne Überprüfung akzeptiert, was zu einem potenziellen Risiko von Man-in-the-Middle-Angriffen führt. Der RFC6187-Standard ist eine Lösung für dieses Problem, da er ein ähnliches Maß an Sicherheit und Benutzerfreundlichkeit bietet wie das TLS-Protokoll (Transport Layer Security), das häufig zum Schutz webbasierter Übertragungen verwendet wird.

Voraussetzungen

Anforderungen

Cisco empfiehlt, über Kenntnisse in folgenden Bereichen zu verfügen:

- PKI-Infrastruktur

Verwendete Komponenten

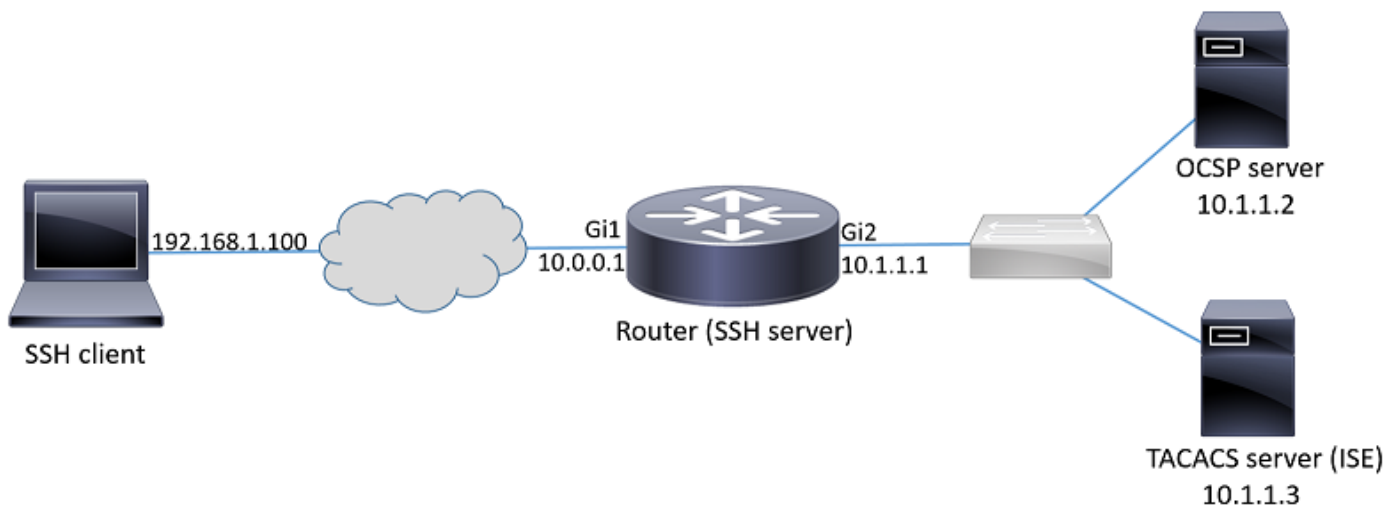
Die Informationen in diesem Dokument basieren auf den folgenden Software- und Hardwareversionen:

- CSR 1000v-Router mit IOS-XE Version 16.6.1
- SSH-Client für die Festung Pragma
- Windows Server 2016 OCSP-Server
- Identity Services Engine Version 2.1

Die Informationen in diesem Dokument wurden von den Geräten in einer bestimmten Laborumgebung erstellt. Alle in diesem Dokument verwendeten Geräte haben mit einer leeren (Standard-)Konfiguration begonnen. Wenn Ihr Netzwerk in Betrieb ist, stellen Sie sicher, dass Sie die potenziellen Auswirkungen eines Befehls verstehen.

Konfigurieren

Netzwerkdiagramm



Überlegungen zur Bereitstellung

- Ein Um diese Funktion nutzen zu können, ist ein RFC6187-kompatibler SSH-Client erforderlich.
- Diese Funktion wurde in IOS, Version 15.5(2)T, und IOS-XE, Version 15.5(2)S, implementiert.
- Der SSH-Client und der Server handeln unterstützte Authentifizierungsmechanismen aus. Alle Authentifizierungsmechanismen, die zuvor auf dem Gerät unterstützt wurden, können weiterhin gleichzeitig mit x509-basierten Authentifizierungsmechanismen ausgeführt werden, um einen reibungslosen Übergang zu gewährleisten.
- Der Administrator kann entscheiden, die x509-basierte Authentifizierungsmethode nur für Server, nur für Clients oder beide zu verwenden.
- Der IOS-Server kann überprüfen, ob das vom Client präsentierte Zertifikat nicht widerrufen wird. Zu diesem Zweck wird bei jeder Verbindung die Datenbank der widerrufenen Zertifikate aufgerufen. Dies ermöglicht den Widerruf des Zugriffs, ohne dass andere Geräte neu

konfiguriert werden müssen, falls der private Schlüssel des Zertifikats beeinträchtigt wird oder der Zugriff für einen bestimmten Benutzer widerrufen werden muss.

- Die Überprüfung auf Widerruf ist optional, es wird jedoch dringend empfohlen, den Zugriff auf Basis kompromittierter Anmeldeinformationen zu verweigern. Eine weitere Option ist die Autorisierung des vom Zertifikat abgerufenen Benutzernamens auf dem externen TACACS- (Terminal Access Controller Access Control System) oder RADIUS-Server. Wenn das Zertifikat kompromittiert wird, kann das Konto auf dem externen Server deaktiviert werden, um den Zugriff mit diesem Zertifikat zu verhindern.
- Die Autorisierung von Benutzern kann von einem externen Server durchgeführt oder übersprungen werden (alle Benutzer mit einem gültigen Zertifikat, von denen angenommen wird, dass sie über Berechtigungen für den Zugriff auf das Gerät verfügen). Die erste Methode wird in diesem Beispiel zur Vereinfachung verwendet.
- Um die Authentifizierungsdaten der anderen Partei erfolgreich zu überprüfen, müssen der Client und der Server nur einer gemeinsamen Zertifizierungsstelle (Certificate Authority, CA) vertrauen. Das bedeutet, dass nur das Zertifikat der Zertifizierungsstelle, das das Routerzertifikat signiert hat, im Zertifikatsspeicher des vertrauenswürdigen Client-Geräts installiert werden muss.
- Das Zertifikat liefert Informationen über die Identität der anderen Partei (Common Name und Subject Alternative Name werden in der Regel zu diesem Zweck verwendet). Der Client sollte den Hostnamen oder den IP-Adressnamen des Servers, der vom Administrator als Eingabe bereitgestellt wurde, mit den im vorliegenden Zertifikat verfügbaren Identitätsdaten vergleichen. Sie schränkt die Möglichkeiten von Man-in-the-Middle-Angriffen oder anderen Identitätswechsel stark ein.

Konfigurationen

Konfigurieren Sie AAA-Parameter. In einem Basisszenario (ohne externen Autorisierungsserver) kann die Autorisierung für den vom Zertifikat abgerufenen Benutzernamen übersprungen werden.

```
aaa new-model
aaa authorization network CERT none
```

Konfigurieren Sie einen Vertrauenspunkt, der das CA-Zertifikat und optional das Router-Zertifikat enthält.

```
crypto pki trustpoint SSH
enrollment mode ra
enrollment url http://10.1.1.2:80/CertSrv/mscep/mscep.dll
serial-number
ip-address 10.0.0.1
subject-name cn=10.0.0.1
revocation-check ocs
ocsp url http://10.1.1.2/ocsp
rsa-keypair SSH 2048
authorization list CERT
```

! The username has to be fetched from the certificate for accounting and authorization purposes.
Multiple options are available.

```
authorization username subjectname commonname
```

Tipp: Falls der OCSP-Server nicht erreichbar ist, kann der Administrator den Zugriff mithilfe der **OcSP**-Konfiguration für die **Widerrufsprüfung** untersagen oder den Zugriff ohne Widerruf-Prüfung mithilfe von **Revocation-Check ocsp none** zulassen (nicht empfohlen).

Konfigurieren Sie die zulässigen Authentifizierungsmechanismen, die bei der SSH-Tunnelverhandlung verwendet werden.

```
! Algorithms used to authenticate server  
ip ssh server algorithm hostkey x509v3-ssh-rsa ssh-rsa
```

```
! Acceptable algorithms used to authenticate the client  
ip ssh server algorithm authentication publickey password keyboard
```

```
! Acceptable pubkey-based algorithms used to authenticate the client  
ip ssh server algorithm publickey x509v3-ssh-rsa ssh-rsa
```

Konfigurieren Sie den SSH-Server so, dass im Authentifizierungsprozess korrekte Zertifikate verwendet werden.

```
ip ssh server certificate profile  
! Certificate used by server  
server  
trustpoint sign SSH
```

```
! CA used to authenticate client certificates  
user  
trustpoint verify SSH
```

(Optional) Integration mit TACACS Server

Nachdem der Benutzername aus dem Zertifikat abgerufen wurde, kann das IOS die Autorisierung für diesen Benutzernamen gegen den TACACS-Server durchführen. Dies ist besonders nützlich, wenn der TACACS-Server bereits für die Geräteadministration bereitgestellt wird.

Hinweis: Der IOS SSH-Server unterstützt derzeit keine Verkettung von Authentifizierungsmethoden. Das bedeutet, dass der TACACS-Server nicht für die Kennwortauthentifizierung verwendet werden kann, wenn die Zertifikate zur Authentifizierung des Benutzers verwendet werden. Es kann nur zur Autorisierung verwendet werden.

Konfigurieren Sie den TACACS-Server.

```
tacacs server ISE  
address ipv4 10.1.1.3  
key cisco123
```

Konfigurieren Sie die Autorisierungsliste für die Verwendung des TACACS-Servers.

```
aaa authorization network ISE group tacacs+
```

1. Konfigurieren Sie die ISE (Identity Services Engine). Das Konfigurationsbeispiel finden Sie unter:

<https://www.cisco.com/c/en/us/support/docs/security/identity-services-engine/200208-Configure-ISE-2-0-IO-OS-TACACS-Authentic.html>

2. Konfigurieren Sie das TACACS-Profil. Zusätzlicher Parameter **cert-application=Alle** müssen konfiguriert werden, damit die Autorisierung erfolgreich durchgeführt werden kann. Navigieren Sie zu **Work Centers > Device Administration > Policy Elements > Results > TACACS profiles > Add**.

Common Tasks

Common Task Type

<input checked="" type="checkbox"/> Default Privilege	<input type="text" value="15"/>	(Select 0 to 15)
<input checked="" type="checkbox"/> Maximum Privilege	<input type="text" value="15"/>	(Select 0 to 15)
<input type="checkbox"/> Access Control List	<input type="text"/>	
<input type="checkbox"/> Auto Command	<input type="text"/>	
<input type="checkbox"/> No Escape	<input type="text"/>	(Select true or false)
<input type="checkbox"/> Timeout	<input type="text"/>	Minutes (0-9999)
<input type="checkbox"/> Idle Time	<input type="text"/>	Minutes (0-9999)

Custom Attributes

[+ Add](#) [Trash](#) [Edit](#)

Type	Name	Value
<input type="checkbox"/> MANDATORY	cert-application	all

3. Um die Richtlinien festzulegen, gehen Sie zu **Work Center > Device Administration > Device Admin Policy Sets > Add**.

▼ Authentication Policy

Default Rule (If no match) : Allow Protocols : Default Device Admin and use : All_User_ID_Stores

▼ Authorization Policy

▼ Exceptions (1)

Local Exceptions

Status	Rule Name	Conditions (Identity groups and other conditions)	Command Sets	Shell Profiles
<input checked="" type="checkbox"/>	Certificate auth	if network admins	then Select Profile(s)	permit_lvl_15

Überprüfen

```
show ip ssh
SSH Enabled - version 1.99
Authentication methods:publickey,password,keyboard-interactive
Authentication Publickey Algorithms:x509v3-ssh-rsa,ssh-rsa
Hostkey Algorithms:x509v3-ssh-rsa,ssh-rsa
--- output truncated ---
```

```
show users
Line User Host(s) Idle Location
1 vty 0 admin1 idle 00:02:37 192.168.1.100
```

Fehlerbehebung

Diese Debuggen werden zum Verfolgen erfolgreicher Sitzungen verwendet:

```
debug ip ssh detail
debug crypto pki transactions
debug crypto pki messages
debug crypto pki validation
```

```
Aug 21 20:07:08.717: SSH0: starting SSH control process
! Server identifies itself
Aug 21 20:07:08.717: SSH0: sent protocol version id SSH-1.99-Cisco-1.25
! Client identifies itself
Aug 21 20:07:08.771: SSH0: protocol version id is - SSH-2.0-Pragma FortressCL 5.0.10.766
Aug 21 20:07:08.771: SSH2 0: kexinit sent: kex algo = diffie-hellman-group-exchange-sha1,diffie-
hellman-group14-sha1

! Authentication algorithms supported by server
Aug 21 20:07:08.771: SSH2 0: kexinit sent: hostkey algo = x509v3-ssh-rsa,ssh-rsa
Aug 21 20:07:08.772: SSH2 0: kexinit sent: encryption algo = aes128-ctr,aes192-ctr,aes256-ctr
Aug 21 20:07:08.772: SSH2 0: kexinit sent: mac algo = hmac-sha2-256,hmac-sha2-512,hmac-
sha1,hmac-sha1-96
Aug 21 20:07:08.772: SSH2 0: SSH2_MSG_KEXINIT sent
Aug 21 20:07:08.915: SSH2 0: SSH2_MSG_KEXINIT received
Aug 21 20:07:08.916: SSH2 0: kex: client->server enc:aes256-ctr mac:hmac-sha1
Aug 21 20:07:08.916: SSH2 0: kex: server->client enc:aes256-ctr mac:hmac-sha1

! Client chooses authentication algorithm
Aug 21 20:07:08.916: SSH2 0: Using hostkey algo = x509v3-ssh-rsa
Aug 21 20:07:08.916: SSH2 0: Using kex_algo = diffie-hellman-group-exchange-sha1
Aug 21 20:07:08.917: SSH2 0: Modulus size established : 4096 bits
Aug 21 20:07:08.976: SSH2 0: expecting SSH2_MSG_KEX_DH_GEX_INIT
Aug 21 20:07:09.141: SSH2 0: SSH2_MSG_KEXDH_INIT received
```

```
! Server sends certificate associated with trustpoint "SSH"
Aug 21 20:07:09.208: SSH2 0: Sending Server certificate associated with PKI trustpoint "SSH"
Aug 21 20:07:09.208: CRYPTO_PKI: (A003C) Session started - identity selected (SSH)
Aug 21 20:07:09.208: SSH2 0: Got 2 certificate(s) on certificate chain
Aug 21 20:07:09.208: CRYPTO_PKI: Rcvd request to end PKI session A003C.
Aug 21 20:07:09.208: CRYPTO_PKI: PKI session A003C has ended. Freeing all resources.
Aug 21 20:07:09.209: CRYPTO_PKI: unlocked trustpoint SSH, refcount is 0
Aug 21 20:07:09.276: SSH2: kex_derive_keys complete
Aug 21 20:07:09.276: SSH2 0: SSH2_MSG_NEWKEYS sent
Aug 21 20:07:09.276: SSH2 0: waiting for SSH2_MSG_NEWKEYS
Aug 21 20:07:16.927: SSH2 0: SSH2_MSG_NEWKEYS received
Aug 21 20:07:17.177: SSH2 0: Authentications that can continue = publickey,password,keyboard-
interactive
Aug 21 20:07:17.225: SSH2 0: Using method = none
Aug 21 20:07:17.226: SSH2 0: Authentications that can continue = publickey,password,keyboard-
interactive
Aug 21 20:07:32.305: SSH2 0: Using method = publickey

! Client sends certificate
Aug 21 20:07:32.305: SSH2 0: Received publickey algo = x509v3-ssh-rsa
Aug 21 20:07:32.305: SSH2 0: Verifying certificate for user 'admin1' in
SSH2_MSG_USERAUTH_REQUEST
Aug 21 20:07:32.305: SSH2 0: Verifying certificate for user 'admin1'
Aug 21 20:07:32.306: SSH2 0: Received a chain of 2 certificate
Aug 21 20:07:32.308: SSH2 0: Received 0 ocsf-response
Aug 21 20:07:32.308: SSH2 0: Starting PKI session for certificate verification
Aug 21 20:07:32.308: CRYPTO_PKI: (A003D) Session started - identity not specified
Aug 21 20:07:32.309: CRYPTO_PKI: (A003D) Adding peer certificate
Aug 21 20:07:32.310: CRYPTO_PKI: found UPN as admin1@example.com
Aug 21 20:07:32.310: CRYPTO_PKI: Added x509 peer certificate - (1016) bytes
Aug 21 20:07:32.310: CRYPTO_PKI: (A003D) Adding peer certificate
Aug 21 20:07:32.310: CRYPTO_PKI: Added x509 peer certificate - (879) bytes
Aug 21 20:07:32.311: CRYPTO_PKI: ip-ext-val: IP extension validation not required
Aug 21 20:07:32.311: CRYPTO_PKI: create new ca_req_context type PKI_VERIFY_CHAIN_CONTEXT,ident
31
Aug 21 20:07:32.312: CRYPTO_PKI: (A003D)validation path has 1 certs

Aug 21 20:07:32.312: CRYPTO_PKI: (A003D) Check for identical certs
Aug 21 20:07:32.312: CRYPTO_PKI : (A003D) Validating non-trusted cert
Aug 21 20:07:32.312: CRYPTO_PKI: (A003D) Create a list of suitable trustpoints
Aug 21 20:07:32.312: CRYPTO_PKI: Found a issuer match
Aug 21 20:07:32.312: CRYPTO_PKI: (A003D) Suitable trustpoints are: SSH,
Aug 21 20:07:32.313: CRYPTO_PKI: (A003D) Attempting to validate certificate using SSH policy
Aug 21 20:07:32.313: CRYPTO_PKI: (A003D) Using SSH to validate certificate
Aug 21 20:07:32.313: CRYPTO_PKI: Added 1 certs to trusted chain.
Aug 21 20:07:32.314: CRYPTO_PKI: Prepare session revocation service providers
Aug 21 20:07:32.314: CRYPTO_PKI: Deleting cached key having key id 30
Aug 21 20:07:32.314: CRYPTO_PKI: Attempting to insert the peer's public key into cache
Aug 21 20:07:32.314: CRYPTO_PKI:Peer's public inserted successfully with key id 31
Aug 21 20:07:32.315: CRYPTO_PKI: Expiring peer's cached key with key id 31
Aug 21 20:07:32.315: CRYPTO_PKI: (A003D) Certificate is verified

! Revocation status is checked
Aug 21 20:07:32.315: CRYPTO_PKI: (A003D) Checking certificate revocation
Aug 21 20:07:32.315: OCSP: (A003D) Process OCSP_VALIDATE message
Aug 21 20:07:32.315: CRYPTO_PKI: (A003D)Starting OCSP revocation check
Aug 21 20:07:32.316: CRYPTO_PKI: OCSP server URL is http://10.1.1.2/ocsp
Aug 21 20:07:32.316: CRYPTO_PKI: no responder matching this URL; create one!
Aug 21 20:07:32.316: OCSP: (A003D)OCSP Get Response command
Aug 21 20:07:32.317: CRYPTO_PKI: http connection opened
Aug 21 20:07:32.317: CRYPTO_PKI: OCSP send header size 132
Aug 21 20:07:32.317: CRYPTO_PKI: sending POST /ocsp HTTP/1.0
Host: 10.1.1.2
```

User-Agent: RSA-Cert-C/2.0
Content-type: application/ocsp-request
Content-length: 312

Aug 21 20:07:32.317: CRYPTO_PKI: OCSP send data size 312
Aug 21 20:07:32.322: OCSP: (A003D)OCSP Parse HTTP Response command
Aug 21 20:07:32.322: OCSP: (A003D)OCSP Validate DER Response command
Aug 21 20:07:32.322: CRYPTO_PKI: OCSP response status - successful.
Aug 21 20:07:32.323: CRYPTO_PKI: Decoding OCSP Response
Aug 21 20:07:32.323: CRYPTO_PKI: OCSP decoded status is GOOD.
Aug 21 20:07:32.323: CRYPTO_PKI: Verifying OCSP Response
Aug 21 20:07:32.325: CRYPTO_PKI: Added 11 certs to trusted chain.
Aug 21 20:07:32.325: ../VIEW_ROOT/cisco.comp/pki_ssl/src/ca/provider/revoke/ocsp/ocsputil.c(547)
: E_NOT_FOUND : no matching entry found
Aug 21 20:07:32.325: ../VIEW_ROOT/cisco.comp/pki_ssl/src/ca/provider/revoke/ocsp/ocsputil.c(547)
: E_NOT_FOUND : no matching entry found
Aug 21 20:07:32.326: CRYPTO_PKI: (A003D) Validating OCSP responder certificate
Aug 21 20:07:32.327: CRYPTO_PKI: OCSP Responder cert doesn't need rev check
Aug 21 20:07:32.328: CRYPTO_PKI: response signed by a delegated responder
Aug 21 20:07:32.328: CRYPTO_PKI: OCSP Response is verified
Aug 21 20:07:32.328: CRYPTO_PKI: (A003D) OCSP revocation check is complete 0
Aug 21 20:07:32.328: OCSP: destroying OCSP trans element
Aug 21 20:07:32.328: CRYPTO_PKI: Revocation check is complete, 0
Aug 21 20:07:32.328: CRYPTO_PKI: Revocation status = 0
Aug 21 20:07:32.328: CRYPTO_PKI: Remove session revocation service providers
Aug 21 20:07:32.329: CRYPTO_PKI: Remove session revocation service providers
Aug 21 20:07:32.329: CRYPTO_PKI: (A003D) Certificate validated
Aug 21 20:07:32.329: CRYPTO_PKI: Populate AAA auth data
Aug 21 20:07:32.329: CRYPTO_PKI: Selected AAA username: 'admin1'
Aug 21 20:07:32.329: CRYPTO_PKI: Anticipate checking AAA list: 'CERT'
Aug 21 20:07:32.329: CRYPTO_PKI: Checking AAA authorization
Aug 21 20:07:32.329: CRYPTO_PKI_AAA: checking AAA authorization (CERT, admin1, <all>)
Aug 21 20:07:32.329: CRYPTO_PKI_AAA: pre-authorization chain validation status (0x400)
Aug 21 20:07:32.329: CRYPTO_PKI_AAA: post-authorization chain validation status (0x400)
Aug 21 20:07:32.329: CRYPTO_PKI: (A003D)chain cert was anchored to trustpoint SSH, and chain
validation result was: CRYPTO_VALID_CERT
Aug 21 20:07:32.329: CRYPTO_PKI: destroying ca_req_context type PKI_VERIFY_CHAIN_CONTEXT,ident
31, ref count 1
Aug 21 20:07:32.330: CRYPTO_PKI: ca_req_context released
Aug 21 20:07:32.330: CRYPTO_PKI: (A003D) Validation TP is SSH
Aug 21 20:07:32.330: CRYPTO_PKI: (A003D) Certificate validation succeeded
Aug 21 20:07:32.330: CRYPTO_PKI: Rcvd request to end PKI session A003D.
Aug 21 20:07:32.330: CRYPTO_PKI: PKI session A003D has ended. Freeing all resources.
Aug 21 20:07:32.395: SSH2 0: Verifying certificate for user 'admin1'
Aug 21 20:07:32.395: SSH2 0: Received a chain of 2 certificate
Aug 21 20:07:32.396: SSH2 0: Received 0 ocsp-response
Aug 21 20:07:32.396: SSH2 0: Starting PKI session for certificate verification
Aug 21 20:07:32.396: CRYPTO_PKI: (A003E) Session started - identity not specified
Aug 21 20:07:32.396: CRYPTO_PKI: (A003E) Adding peer certificate
Aug 21 20:07:32.397: CRYPTO_PKI: found UPN as admin1@example.com
Aug 21 20:07:32.397: CRYPTO_PKI: Added x509 peer certificate - (1016) bytes
Aug 21 20:07:32.397: CRYPTO_PKI: (A003E) Adding peer certificate
Aug 21 20:07:32.398: CRYPTO_PKI: Added x509 peer certificate - (879) bytes
Aug 21 20:07:32.398: CRYPTO_PKI: ip-ext-val: IP extension validation not required
Aug 21 20:07:32.400: CRYPTO_PKI: create new ca_req_context type PKI_VERIFY_CHAIN_CONTEXT,ident
32
Aug 21 20:07:32.400: CRYPTO_PKI: (A003E)validation path has 1 certs

Aug 21 20:07:32.400: CRYPTO_PKI: (A003E) Check for identical certs
Aug 21 20:07:32.400: CRYPTO_PKI : (A003E) Validating non-trusted cert
Aug 21 20:07:32.401: CRYPTO_PKI: (A003E) Create a list of suitable trustpoints
Aug 21 20:07:32.401: CRYPTO_PKI: Found a issuer match
Aug 21 20:07:32.401: CRYPTO_PKI: (A003E) Suitable trustpoints are: SSH,

Aug 21 20:07:32.401: CRYPTO_PKI: (A003E) Attempting to validate certificate using SSH policy
Aug 21 20:07:32.401: CRYPTO_PKI: (A003E) Using SSH to validate certificate
Aug 21 20:07:32.402: CRYPTO_PKI: Added 1 certs to trusted chain.
Aug 21 20:07:32.402: CRYPTO_PKI: Prepare session revocation service providers
Aug 21 20:07:32.402: CRYPTO_PKI: Deleting cached key having key id 31
Aug 21 20:07:32.403: CRYPTO_PKI: Attempting to insert the peer's public key into cache
Aug 21 20:07:32.403: CRYPTO_PKI:Peer's public inserted successfully with key id 32
Aug 21 20:07:32.404: CRYPTO_PKI: Expiring peer's cached key with key id 32
Aug 21 20:07:32.404: CRYPTO_PKI: (A003E) Certificate is verified
Aug 21 20:07:32.404: CRYPTO_PKI: (A003E) Checking certificate revocation
Aug 21 20:07:32.404: OCSP: (A003E) Process OCSP_VALIDATE message
Aug 21 20:07:32.404: CRYPTO_PKI: (A003E)Starting OCSP revocation check
Aug 21 20:07:32.405: CRYPTO_PKI: OCSP server URL is http://10.1.1.2/ocsp
Aug 21 20:07:32.405: CRYPTO_PKI: no responder matching this URL; create one!
Aug 21 20:07:32.405: OCSP: (A003E)OCSP Get Response command
Aug 21 20:07:32.406: CRYPTO_PKI: http connection opened
Aug 21 20:07:32.406: CRYPTO_PKI: OCSP send header size 132
Aug 21 20:07:32.406: CRYPTO_PKI: sending POST /ocsp HTTP/1.0
Host: 10.1.1.2
User-Agent: RSA-Cert-C/2.0
Content-type: application/ocsp-request
Content-length: 312

Aug 21 20:07:32.406: CRYPTO_PKI: OCSP send data size 312
Aug 21 20:07:32.409: OCSP: (A003E)OCSP Parse HTTP Response command
Aug 21 20:07:32.410: OCSP: (A003E)OCSP Validate DER Response command
Aug 21 20:07:32.410: CRYPTO_PKI: OCSP response status - successful.
Aug 21 20:07:32.410: CRYPTO_PKI: Decoding OCSP Response
Aug 21 20:07:32.411: CRYPTO_PKI: OCSP decoded status is GOOD.
Aug 21 20:07:32.411: CRYPTO_PKI: Verifying OCSP Response
Aug 21 20:07:32.413: CRYPTO_PKI: Added 11 certs to trusted chain.
Aug 21 20:07:32.413: ../VIEW_ROOT/cisco.comp/pki_ssl/src/ca/provider/revoke/ocsp/ocsputil.c(547)
: E_NOT_FOUND : no matching entry found
Aug 21 20:07:32.413: ../VIEW_ROOT/cisco.comp/pki_ssl/src/ca/provider/revoke/ocsp/ocsputil.c(547)
: E_NOT_FOUND : no matching entry found
Aug 21 20:07:32.414: CRYPTO_PKI: (A003E) Validating OCSP responder certificate
Aug 21 20:07:32.415: CRYPTO_PKI: OCSP Responder cert doesn't need rev check
Aug 21 20:07:32.415: CRYPTO_PKI: response signed by a delegated responder
Aug 21 20:07:32.416: CRYPTO_PKI: OCSP Response is verified
Aug 21 20:07:32.416: CRYPTO_PKI: (A003E) OCSP revocation check is complete 0
Aug 21 20:07:32.416: OCSP: destroying OCSP trans element
Aug 21 20:07:32.416: CRYPTO_PKI: Revocation check is complete, 0
Aug 21 20:07:32.416: CRYPTO_PKI: Revocation status = 0
Aug 21 20:07:32.416: CRYPTO_PKI: Remove session revocation service providers
Aug 21 20:07:32.416: CRYPTO_PKI: Remove session revocation service providers
Aug 21 20:07:32.416: CRYPTO_PKI: (A003E) Certificate validated
Aug 21 20:07:32.417: CRYPTO_PKI: Populate AAA auth data
Aug 21 20:07:32.417: CRYPTO_PKI: Selected AAA username: 'admin1'
Aug 21 20:07:32.417: CRYPTO_PKI: Anticipate checking AAA list: 'CERT'
Aug 21 20:07:32.417: CRYPTO_PKI: Checking AAA authorization
Aug 21 20:07:32.417: CRYPTO_PKI_AAA: checking AAA authorization (CERT, admin1, <all>)
Aug 21 20:07:32.417: CRYPTO_PKI_AAA: pre-authorization chain validation status (0x400)
Aug 21 20:07:32.417: CRYPTO_PKI_AAA: post-authorization chain validation status (0x400)
Aug 21 20:07:32.417: CRYPTO_PKI: (A003E)chain cert was anchored to trustpoint SSH, and chain
validation result was: CRYPTO_VALID_CERT
Aug 21 20:07:32.417: CRYPTO_PKI: destroying ca_req_context type PKI_VERIFY_CHAIN_CONTEXT,ident
32, ref count 1
Aug 21 20:07:32.417: CRYPTO_PKI: ca_req_context released
Aug 21 20:07:32.417: CRYPTO_PKI: (A003E) Validation TP is SSH
Aug 21 20:07:32.417: CRYPTO_PKI: (A003E) Certificate validation succeeded
Aug 21 20:07:32.418: CRYPTO_PKI: Rcvd request to end PKI session A003E.
Aug 21 20:07:32.418: CRYPTO_PKI: PKI session A003E has ended. Freeing all resources.
Aug 21 20:07:32.418: SSH2 0: Verifying signature for user 'admin1' in SSH2_MSG_USERAUTH_REQUEST

```
Aug 21 20:07:32.418: SSH2 0: Received a chain of 2 certificate
Aug 21 20:07:32.418: SSH2 0: Received 0 ocsf-response
Aug 21 20:07:32.418: CRYPTO_PKI: found UPN as admin1@example.com

! Certificate status verified successfully
Aug 21 20:07:32.419: SSH2 0: Client Signature verification PASSED
Aug 21 20:07:32.419: SSH2 0: Certificate authentication passed for user 'admin1'
Aug 21 20:07:32.419: SSH2 0: authentication successful for admin1
Aug 21 20:07:32.470: SSH2 0: channel open request
Aug 21 20:07:32.521: SSH2 0: pty-req request
Aug 21 20:07:32.521: SSH2 0: setting TTY - requested: height 25, width 80; set: height 25, width
80
Aug 21 20:07:32.570: SSH2 0: shell request
Aug 21 20:07:32.570: SSH2 0: shell message received
Aug 21 20:07:32.570: SSH2 0: starting shell for vty
Aug 21 20:07:32.631: SSH2 0: channel window adjust message received 8
```

Falls das Zertifikat für admin1 widerrufen wurde:

```
Aug 21 19:39:52.081: CRYPTO_PKI: OCSP Response is verified
Aug 21 19:39:52.081: CRYPTO_PKI: (A0024) OCSP revocation check is complete 0
Aug 21 19:39:52.082: OCSP: destroying OCSP trans element
Aug 21 19:39:52.082: CRYPTO_PKI: Revocation check is complete, 0
Aug 21 19:39:52.082: CRYPTO_PKI: Revocation status = 1
Aug 21 19:39:52.082: CRYPTO_PKI: Remove session revocation service providers
Aug 21 19:39:52.082: CRYPTO_PKI: Remove session revocation service providers
Aug 21 19:39:52.082: CRYPTO_PKI: (A0024) Certificate revoked
Aug 21 19:39:52.082: %PKI-3-CERTIFICATE_REVOKED: Certificate chain validation has failed. The
certificate (SN: 750000001B78DA4CC0078DEC0700000000001B) is revoked
Aug 21 19:39:52.082: CRYPTO_PKI: (A0024)chain cert was anchored to trustpoint Unknown, and chain
validation result was: CRYPTO_CERT_REVOKED
Aug 21 19:39:52.082: CRYPTO_PKI: destroying ca_req_context type PKI_VERIFY_CHAIN_CONTEXT, ident
18, ref count 1
Aug 21 19:39:52.082: CRYPTO_PKI: ca_req_context released
Aug 21 19:39:52.083: CRYPTO_PKI: (A0024) Certificate validation failed
```

Weitere Informationen

- **PKI-Konfigurationsleitfaden:**
https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/sec_conn_pki/configuration/15-mt/sec-pki-15-mt-book.html
- **TACACS bei ISE-Konfigurationsbeispiel:**
<https://www.cisco.com/c/en/us/support/docs/security/identity-services-engine/200208-Configure-ISE-2-0-IOS-TACACS-Authentic.html>
- [Technischer Support und Dokumentation - Cisco Systems](#)