

Configurando a autenticação de certificado para SSH em dispositivos Cisco IOS XE

Contents

[Introdução](#)

[Pré-requisitos](#)

[Requisitos](#)

[Componentes Utilizados](#)

[Informações de Apoio](#)

[Configurar](#)

[Diagrama de Rede](#)

[Considerações de implantação](#)

[Configurações](#)

[Dispositivo IOS-XE \(servidor SSH\)](#)

[Pragma Fortress CL \(cliente SSH instalado na máquina do usuário\)](#)

[Verificar](#)

[Troubleshooting](#)

[Problemas comuns](#)

[Erros de configuração](#)

[Informações Relacionadas](#)

Introdução

Este documento descreve a configuração do Secure Shell (SSH) em dispositivos Cisco IOS® XE usando certificados X.509v3 para autenticação, de acordo com as diretrizes definidas no RFC 6187.

Pré-requisitos

Requisitos

A Cisco recomenda que você tenha conhecimento da infraestrutura de Public Key Infrastructure (PKI).

Componentes Utilizados

As informações neste documento são baseadas nestas versões de software e hardware:

- Switch C9200L executando o Cisco IOS XE versão 17.3.5
- cliente SSH do Pragma Fortress

As informações neste documento foram criadas a partir de dispositivos em um ambiente de laboratório específico. Todos os dispositivos utilizados neste documento foram iniciados com uma configuração (padrão) inicial. Se a rede estiver ativa, certifique-se de que você entenda o impacto potencial de qualquer comando

Informações de Apoio

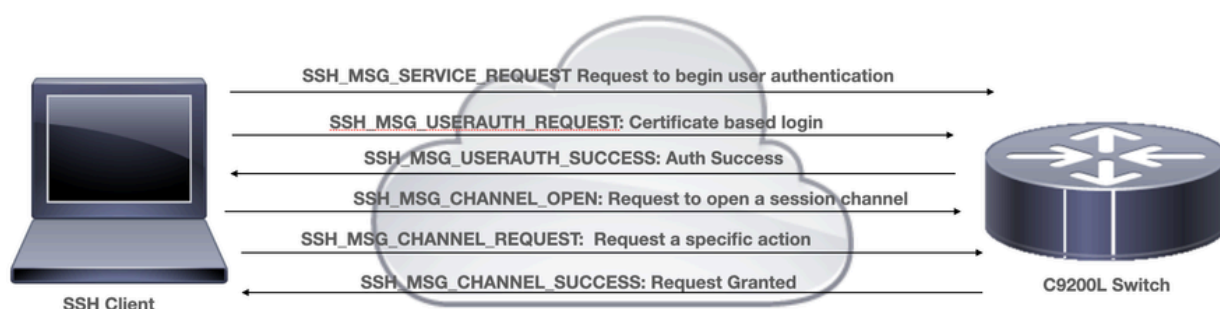
A abordagem melhora a segurança SSH ao habilitar a autenticação baseada em certificado, alinhando, assim, as práticas de gerenciamento de chave SSH com as usadas no Transport Layer Security (TLS).

O SSH fornece autenticação mútua para estabelecer uma conexão segura entre cliente e servidor. Tradicionalmente, os servidores são autenticados usando pares de chaves RSA (Rivest-Shamir-Addleman). O cliente computa uma impressão digital da chave pública do servidor e solicita que o administrador a verifique, de preferência comparando-a com um valor conhecido obtido por meio de um método seguro e fora da banda. No entanto, essa verificação manual é frequentemente ignorada devido à sua complexidade, aumentando o risco de ataques man-in-the-middle (MitM) e enfraquecendo o modelo de confiança SSH.

O RFC 6187 aborda esses problemas habilitando a autenticação baseada em certificado X.509v3, que integra SSH com PKI. Essa abordagem melhora a segurança e a escalabilidade, permitindo que a confiança seja estabelecida por meio de Autoridades de Certificação (CAs) confiáveis, oferecendo uma experiência de usuário e um modelo de confiança semelhante ao TLS.

Configurar

Diagrama de Rede



Considerações de implantação

- Um cliente SSH compatível com RFC6187 é necessário para aproveitar o recurso.
- O cliente e o servidor SSH negociam os mecanismos de autenticação suportados. Todos os

mecanismos de autenticação anteriormente suportados no dispositivo podem continuar a ser executados simultaneamente com mecanismos de autenticação baseados em x509 para garantir uma transição suave.

- O administrador pode optar por usar o método de autenticação baseado em x509 somente para o servidor, somente para o cliente ou ambos.
- Para verificar com êxito os dados de autenticação da outra parte, o cliente e o servidor precisam confiar apenas em uma CA comum. Isso significa que somente o certificado de CA que assinou o certificado do roteador deve ser instalado no armazenamento de certificados confiáveis do dispositivo cliente.
- O certificado fornece informações sobre a identidade da outra parte (o nome comum e o nome alternativo do assunto são normalmente usados para essa finalidade). O cliente deve comparar o nome do host ou o nome do endereço IP do servidor que foi fornecido como entrada pelo administrador com os dados de identidade disponíveis no certificado apresentado. Limita severamente as oportunidades do MitM ou outros ataques de personificação.

Configurações

Dispositivo IOS-XE (servidor SSH)

Configure um ponto confiável que contenha o certificado CA e, opcionalmente, o certificado do roteador.

```
crypto pki trustpoint pki-server
  enrollment pkcs12
  subject-name cn=RTR-DC01.cisco.com
  revocation-check none
  rsakeypair ssh-cert
```

! The username has to be fetched from the certificate for accounting and authorization purposes. Multip

```
authorization username subjectname commonname
```

Configure os mecanismos de autenticação permitidos usados durante a negociação do túnel SSH.

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

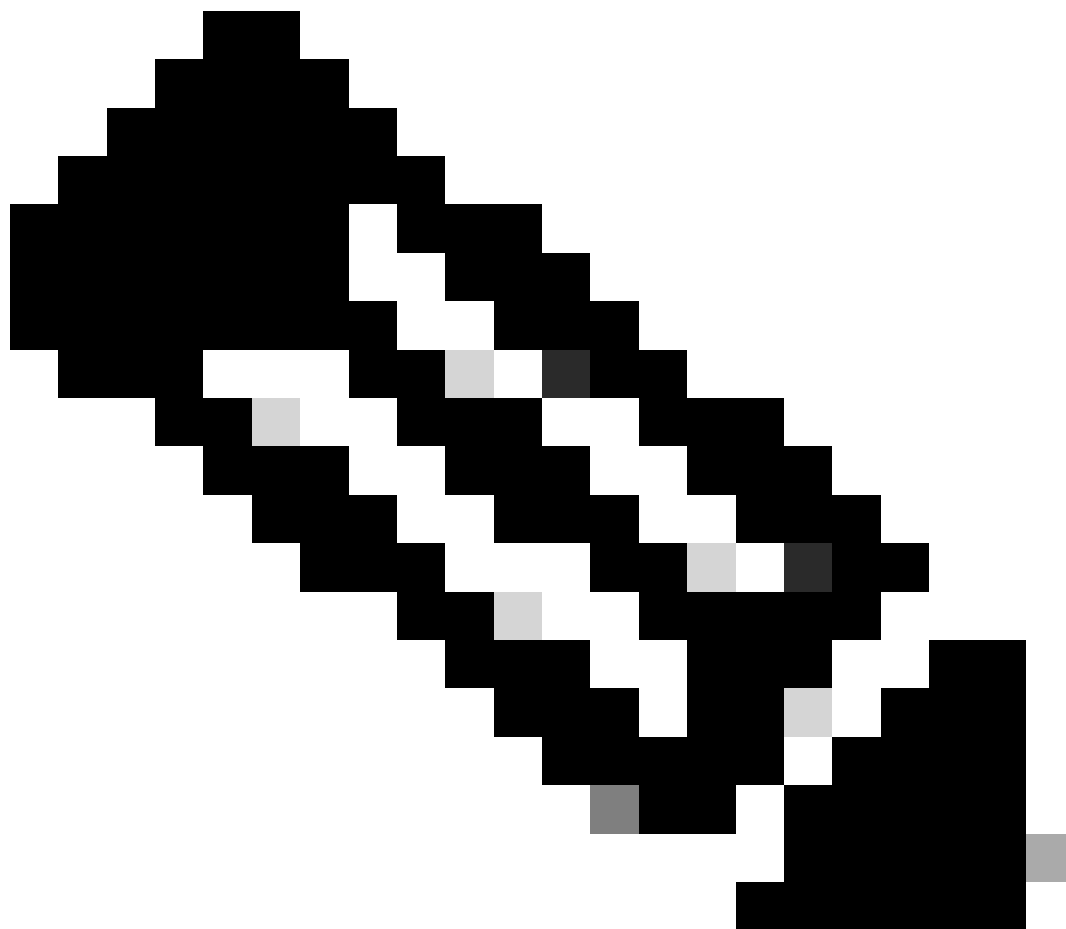
```
! Acceptable algorithms used to authenticate the client
ip ssh server algorithm authentication publickey password keyboard
```

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

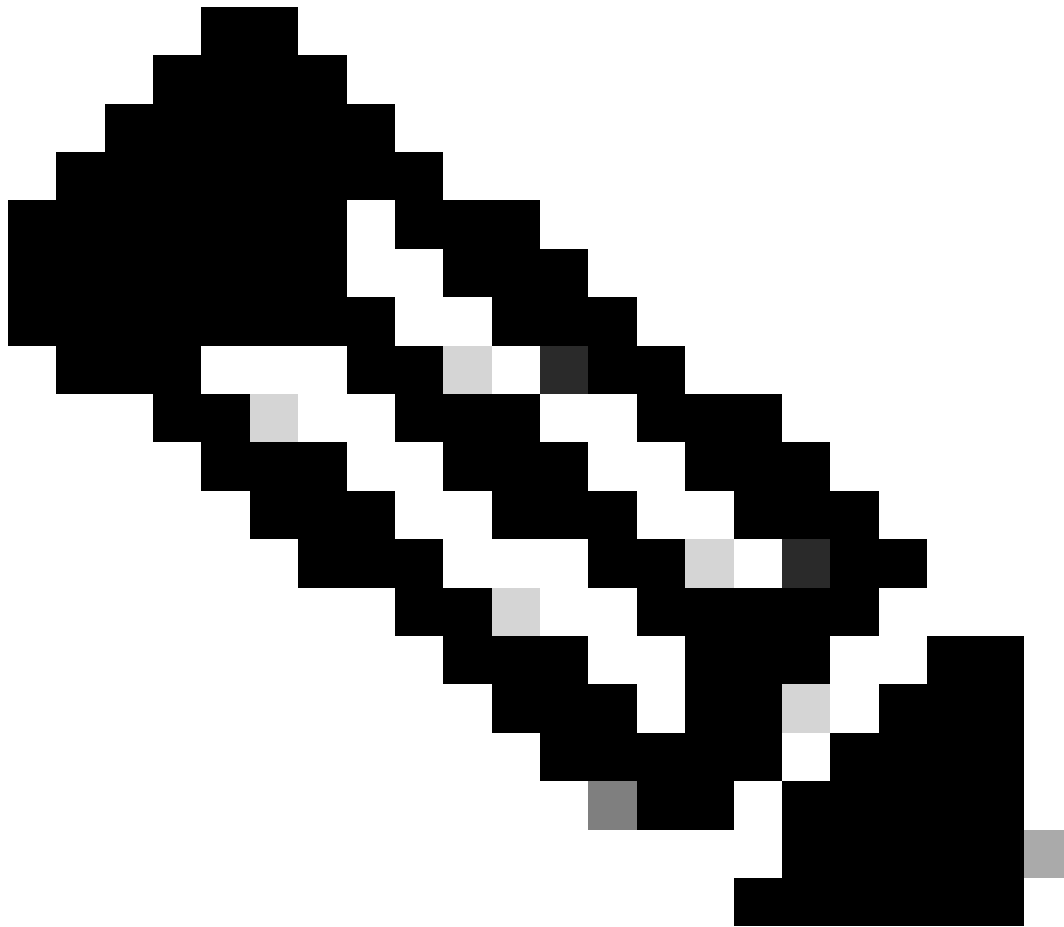
Configure o servidor SSH para usar certificados corretos no processo de autenticação.

```
ip ssh server certificate profile
server
    trustpoint sign
```

```
user
    trustpoint verify
```



Note: Certifique-se de que o servidor SSH e o cliente SSH tenham o certificado de ID emitido pelo mesmo servidor CA.



Note: Se, no caso de o cliente SSH, como a máquina Windows, tiver um certificado de ID emitido por alguma outra CA, faça com que ele seja importado no servidor SSH, ou seja, o Switch Cisco e mapeie esse ponto confiável para o perfil de certificado do servidor SSH acima na seção do usuário.

Pragma Fortress CL (cliente SSH instalado na máquina do usuário)

RTR-DC01.cisco.com

Authentication

Logging

Keyboard

Proxy

Serial

Telnet

Terminal

+ Ssh

+ Window

RTR-DC01.cisco.com

Site name:

RTR-DC01.cisco.com

Host address:

RTR-DC01.cisco.com

Host Alias:

Protocol:

ssh2

Port:

22

Comment:

Host address is the Hostname of the Cisco Device mentioned in the ID certificate issued to it by the CA server



Use Global Sites

Connect

Cancel

Apply

Site Manager X

RTR-DC01.cisco.com

- Authentication**
- Logging
- Keyboard
- Proxy
- Serial
- Telnet
- Terminal
- + Ssh
- + Window

Select the Public Key/
Certificate Auth method
and select the fill userid
which will fetch it from the
Client ID certificate present
on the machine

Authentication

UserId:

☐ Has domain Domain:

☒ Fill UserId From SC - Map:

☐ Password

☐ GSSAPI ☒ Use MIC

☐ SmartCard/CAC Reader: ☐ Key Only

☒ Public Key/Certificate

☐ Key or Certificate from file

☒ Certificate from User Store

☐ Use Global Sites

Verificar

```
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 sameer.cisco.com idle 00:02:37 192.168.1.100
```

Troubleshooting

Essas depurações são usadas para rastrear sessões bem-sucedidas:

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

Mar 27 15:35:40.103: SSH1: starting SSH control process

! Server identifies itself

Mar 27 15:35:40.103: SSH1: sent protocol version id SSH-1.99-Cisco-1.25

! Client identifies itself

Mar 27 15:35:40.106: SSH1: protocol version id is - SSH-2.0-Pragma FortressCL 5.0.10.4176

! Authentication algorithms supported by server

Mar 27 15:35:40.106: SSH2 1: kexinit sent: kex algo = diffie-hellman-group-exchange-sha1,diffie-hellman

Mar 27 15:35:40.106: SSH2 1: kexinit sent: hostkey algo = x509v3-ssh-rsa,ssh-rsa

Mar 27 15:35:40.106: SSH2 1: kexinit sent: encryption algo = aes128-ctr,aes192-ctr,aes256-ctr

Mar 27 15:35:40.106: SSH2 1: kexinit sent: mac algo = hmac-sha2-256,hmac-sha2-512,hmac-sha1,hmac-sha1-96

Mar 27 15:35:40.106: SSH2 1: SSH2_MSG_KEXINIT sent

Mar 27 15:35:40.109: SSH2 1: SSH2_MSG_KEXINIT received

Mar 27 15:35:40.109: SSH2 1: kex: client->server enc:aes256-ctr mac:hmac-sha2-256

Mar 27 15:35:40.109: SSH2 1: kex: server->client enc:aes256-ctr mac:hmac-sha2-256

! Client chooses authentication algorithm

Mar 27 15:35:40.109: SSH2 1: Using hostkey algo = x509v3-ssh-rsa

Mar 27 15:35:40.109: SSH2 1: Using kex_algo = diffie-hellman-group-exchange-sha1

Mar 27 15:35:40.109: SSH2 1: SSH2_MSG_KEX_DH_GEX_REQUEST received

Mar 27 15:35:40.109: SSH2 1: Range sent by client is - 1024 < 2048 < 8192

Mar 27 15:35:40.109: SSH2 1: Modulus size established : 2048 bits

Mar 27 15:35:40.121: SSH2 1: expecting SSH2_MSG_KEX_DH_GEX_INIT

Mar 27 15:35:40.121: SSH2 1: SSH2_MSG_KEXDH_INIT received

! SSH Server sends certificate associated with trustpoint "pki-server"

Mar 27 15:35:40.133: SSH2 1: Sending Server certificate associated with PKI trustpoint "pki-server"

Mar 27 15:35:40.133: SSH2 1: Got 2 certificate(s) on certificate chain

Mar 27 15:35:40.135: SSH2: kex_derive_keys complete

Mar 27 15:35:40.135: SSH2 1: SSH2_MSG_NEWKEYS sent

Mar 27 15:35:40.135: SSH2 1: waiting for SSH2_MSG_NEWKEYS

Mar 27 15:35:40.214: SSH2 0: channel window adjust message received 49926

Mar 27 15:35:41.417: SSH2 1: SSH2_MSG_NEWKEYS received

Mar 27 15:35:41.436: SSH2 1: Authentications that can continue = publickey,password,keyboard-interactive

Mar 27 15:35:41.437: SSH2 1: Using method = none

Mar 27 15:35:41.437: SSH2 1: Authentications that can continue = publickey,password,keyboard-interactive

Mar 27 15:35:41.438: SSH2 1: Using method = publickey

! Client sends certificate

Mar 27 15:35:41.438: SSH2 1: Received publickey algo = x509v3-ssh-rsa

Mar 27 15:35:41.438: SSH2 1: Verifying certificate for user 'sameer.cisco.com' in SSH2_MSG_USERAUTH_REQUEST

Mar 27 15:35:41.439: SSH2 1: Verifying certificate for user 'sameer.cisco.com'

Mar 27 15:35:41.439: SSH2 1: Received a chain of 2 certificate

Mar 27 15:35:41.439: SSH2 1: Received 0 ocsp-response

Mar 27 15:35:41.439: SSH2 1: Starting PKI session for certificate verification

! Client certificate is verified by the SSH-Server

Mar 27 15:35:41.444: SSH2 1: Verifying certificate for user 'sameer.cisco.com'

Mar 27 15:35:41.444: SSH2 1: Received a chain of 2 certificate

Mar 27 15:35:41.444: SSH2 1: Received 0 ocsp-response

Mar 27 15:35:41.444: SSH2 1: Starting PKI session for certificate verification

Mar 27 15:35:41.445: SSH2 1: Verifying signature for user 'sameer.cisco.com' in SSH2_MSG_USERAUTH_REQUEST

Mar 27 15:35:41.445: SSH2 1: Received a chain of 2 certificate

Mar 27 15:35:41.445: SSH2 1: Received 0 ocsp-response

! Certificate status verified successfully

```
Mar 27 15:35:41.446: SSH2 1: Client Signature verification PASSED
Mar 27 15:35:41.446: SSH2 1: Certificate authentication passed for user 'sameer.cisco.com'
Mar 27 15:35:41.446: SSH2 1: authentication successful for sameer.cisco.com
Mar 27 15:35:41.448: SSH2 1: channel open request
Mar 27 15:35:41.451: SSH2 1: pty-req request
Mar 27 15:35:41.451: SSH2 1: setting TTY - requested: height 25, width 80; set: height 25, width 80
Mar 27 15:35:41.452: SSH2 1: shell request
Mar 27 15:35:41.452: SSH2 1: shell message received
Mar 27 15:35:41.452: SSH2 1: starting shell for vty
Mar 27 15:35:41.464: SSH2 1: channel window adjust message received 9
Aug 21 20:07:32.311: CRYPTO_PKI: ip-ext-val: IP extension validation not required
```

Problemas comuns

Erros de configuração

A validação do certificado é bem-sucedida para o usuário, no entanto, devido à ausência do comando obrigatório `authorization username subjectname commonname` na configuração, a Autenticação do Certificado para o usuário falha.

```
Apr 26 01:35:32.222: SSH1: starting SSH control process
Apr 26 01:35:32.222: SSH1: sent protocol version id SSH-1.99-Cisco-1.25
Apr 26 01:35:32.224: SSH1: protocol version id is - SSH-2.0-Pragma FortressCL 5.0.10.4176
Apr 26 01:35:32.224: SSH2 1: kexinit sent: kex algo = diffie-hellman-group-exchange-sha1,diffie-hellman
Apr 26 01:35:32.224: SSH2 1: kexinit sent: hostkey algo = x509v3-ssh-rsa,ssh-rsa
Apr 26 01:35:32.224: SSH2 1: kexinit sent: encryption algo = aes128-ctr,aes192-ctr,aes256-ctr
Apr 26 01:35:32.224: SSH2 1: kexinit sent: mac algo = hmac-sha2-256,hmac-sha2-512,hmac-sha1,hmac-sha1-9
Apr 26 01:35:32.224: SSH2 1: SSH2_MSG_KEXINIT sent
Apr 26 01:35:32.234: SSH2 1: SSH2_MSG_KEXINIT received
Apr 26 01:35:32.234: SSH2 1: kex: client->server enc:aes256-ctr mac:hmac-sha2-256
Apr 26 01:35:32.235: SSH2 1: kex: server->client enc:aes256-ctr mac:hmac-sha2-256
Apr 26 01:35:32.235: SSH2 1: Using hostkey algo = x509v3-ssh-rsa
Apr 26 01:35:32.235: SSH2 1: Using kex_algo = diffie-hellman-group-exchange-sha1
Apr 26 01:35:32.235: SSH2 1: SSH2_MSG_KEX_DH_GEX_REQUEST received
Apr 26 01:35:32.235: SSH2 1: Range sent by client is - 1024 < 2048 < 8192
Apr 26 01:35:32.235: SSH2 1: Modulus size established : 2048 bits
Apr 26 01:35:32.246: SSH2 1: expecting SSH2_MSG_KEX_DH_GEX_INIT
Apr 26 01:35:32.246: SSH2 1: SSH2_MSG_KEXDH_INIT received
Apr 26 01:35:32.259: SSH2 1: Sending Server certificate associated with PKI trustpoint "pki-server"
Apr 26 01:35:32.259: CRYPTO_PKI: (A0049) Session started - identity selected (pki-server)
Apr 26 01:35:32.259: SSH2 1: Got 3 certificate(s) on certificate chain
Apr 26 01:35:32.259: CRYPTO_PKI: Rcvd request to end PKI session A0049.
Apr 26 01:35:32.260: CRYPTO_PKI: PKI session A0049 has ended. Freeing all resources.
Apr 26 01:35:32.260: CRYPTO_PKI: unlocked trustpoint pki-server, refcount is 0
Apr 26 01:35:32.273: SSH2: kex_derive_keys complete
Apr 26 01:35:32.274: SSH2 1: SSH2_MSG_NEWKEYS sent
Apr 26 01:35:32.274: SSH2 1: waiting for SSH2_MSG_NEWKEYS
Apr 26 01:35:45.664: SSH2 1: SSH2_MSG_NEWKEYS received
Apr 26 01:35:45.665: SSH2 1: Authentications that can continue = publickey,password,keyboard-interactive
Apr 26 01:35:45.666: SSH2 1: Using method = none
Apr 26 01:35:45.666: SSH2 1: Authentications that can continue = publickey,password,keyboard-interactive
Apr 26 01:35:45.675: SSH2 1: Using method = publickey
Apr 26 01:35:45.675: SSH2 1: Received publickey algo = x509v3-ssh-rsa
Apr 26 01:35:45.676: SSH2 1: Verifying certificate for user 'sameer.cisco.com' in SSH2_MSG_USERAUTH_REQ
```

Apr 26 01:35:45.676: SSH2 1: Verifying certificate for user 'sameer.cisco.com'
Apr 26 01:35:45.676: SSH2 1: Received a chain of 3 certificate
Apr 26 01:35:45.676: SSH2 1: Received 0 oosp-response
Apr 26 01:35:45.676: SSH2 1: Starting PKI session for certificate verification
Apr 26 01:35:45.676: CRYPTO_PKI: (A004A) Session started - identity not specified
Apr 26 01:35:45.676: CRYPTO_PKI: (A004A) Adding peer certificate
Apr 26 01:35:45.676: CRYPTO_PKI: Added x509 peer certificate - (1249) bytes
Apr 26 01:35:45.676: CRYPTO_PKI: (A004A) Adding peer certificate
Apr 26 01:35:45.676: CRYPTO_PKI: Added x509 peer certificate - (1215) bytes
Apr 26 01:35:45.676: CRYPTO_PKI: (A004A) Adding peer certificate
Apr 26 01:35:45.676: CRYPTO_PKI: Added x509 peer certificate - (921) bytes
Apr 26 01:35:45.676: CRYPTO_PKI: ip-ext-val: IP extension validation not required
Apr 26 01:35:45.677: CRYPTO_PKI: create new ca_req_context type PKI_VERIFY_CHAIN_CONTEXT,ident 37
Apr 26 01:35:45.677: CRYPTO_PKI: (A004A)validation path has 1 certs
Apr 26 01:35:45.677: CRYPTO_PKI: (A004A) Check for identical certs
Apr 26 01:35:45.677: CRYPTO_PKI : (A004A) Validating non-trusted cert
Apr 26 01:35:45.677: CRYPTO_PKI: (A004A) Create a list of suitable trustpoints
Apr 26 01:35:45.677: CRYPTO_PKI: Found a issuer match
Apr 26 01:35:45.677: CRYPTO_PKI: (A004A) Suitable trustpoints are: pki-server,
Apr 26 01:35:45.677: CRYPTO_PKI: (A004A) Attempting to validate certificate using pki-server policy
Apr 26 01:35:45.677: CRYPTO_PKI: ECU validation successful. Cert matches configuration.
Apr 26 01:35:45.677: CRYPTO_PKI: (A004A) Using pki-server to validate certificate
Apr 26 01:35:45.677: CRYPTO_PKI: Added 1 certs to trusted chain.
Apr 26 01:35:45.677: CRYPTO_PKI: Prepare session revocation service providers
Apr 26 01:35:45.677: CRYPTO_PKI: Deleting cached key having key id 50
Apr 26 01:35:45.677: CRYPTO_PKI: Attempting to insert the peer's public key into cache
Apr 26 01:35:45.677: CRYPTO_PKI:Peer's public inserted successfully with key id 51
Apr 26 01:35:45.678: CRYPTO_PKI: Expiring peer's cached key with key id 51
Apr 26 01:35:45.678: CRYPTO_PKI: (A004A) Certificate is verified
Apr 26 01:35:45.678: CRYPTO_PKI: Remove session revocation service providers
Apr 26 01:35:45.678: CRYPTO_PKI: Remove session revocation service providers
Apr 26 01:35:45.678: CRYPTO_PKI: (A004A) Certificate validated without revocation check
Apr 26 01:35:45.678: CRYPTO_PKI: Populate AAA auth data
Apr 26 01:35:45.678: CRYPTO_PKI: Unable to get configured attribute for primary AAA list authorization.
Apr 26 01:35:45.678: CRYPTO_PKI: (A004A)chain cert was anchored to trustpoint pki-server, and chain val
Apr 26 01:35:45.678: CRYPTO_PKI: destroying ca_req_context type PKI_VERIFY_CHAIN_CONTEXT,ident 37, ref
Apr 26 01:35:45.678: CRYPTO_PKI: ca_req_context released
Apr 26 01:35:45.678: CRYPTO_PKI: (A004A) Validation TP is pki-server
Apr 26 01:35:45.678: CRYPTO_PKI: (A004A) Certificate validation succeeded
Apr 26 01:35:45.678: SSH2 1: Could not find username field configured for certificate in trustpoint Err
Apr 26 01:35:45.678: CRYPTO_PKI: Rcvd request to end PKI session A004A.
Apr 26 01:35:45.678: CRYPTO_PKI: PKI session A004A has ended. Freeing all resources.
Apr 26 01:35:45.679: SSH2 1: Can not decode the certificate Err code = 7
Apr 26 01:35:45.679: SSH2 1: Certificate authentication failed for user 'sameer.cisco.com'

Informações Relacionadas

- [Guia de configuração de PKI](#)
- [Suporte técnico e downloads da Cisco](#)

Sobre esta tradução

A Cisco traduziu este documento com a ajuda de tecnologias de tradução automática e humana para oferecer conteúdo de suporte aos seus usuários no seu próprio idioma, independentemente da localização.

Observe que mesmo a melhor tradução automática não será tão precisa quanto as realizadas por um tradutor profissional.

A Cisco Systems, Inc. não se responsabiliza pela precisão destas traduções e recomenda que o documento original em inglês ([link fornecido](#)) seja sempre consultado.