

# Redimensionar chaves RSA SSH padrão nas bordas SD-WAN do Cisco IOS XE

## Contents

---

[Introdução](#)

[Pré-requisitos](#)

[Requisitos](#)

[Componentes Utilizados](#)

[Informações de Apoio](#)

[Configurar](#)

[Diagrama de Rede](#)

[Configurações](#)

[Verificar](#)

---

## Introdução

Este documento descreve como aumentar as chaves RSA SSH padrão usadas para protocolos seguros para um comprimento mais forte nas bordas SD-WAN do Cisco IOS® XE.

## Pré-requisitos

### Requisitos

A Cisco recomenda que você tenha conhecimento destes tópicos:

- Rede de longa distância definida pelo software Cisco Catalyst (SD-WAN)
- Operação básica de Chaves SSH e Certificado
- Algoritmo RSA

### Componentes Utilizados

- Cisco IOS® XE Catalyst SD-WAN Edges 17.9.4a

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

O Secure Shell (SSH) é um protocolo de rede que permite que os usuários estabeleçam

conexões remotas com dispositivos mesmo em uma rede desprotegida. O protocolo protege as sessões usando mecanismos criptográficos padrão baseados em uma arquitetura cliente-servidor.

RSA é Rivest, Shamir, Adleman: Algoritmo de criptografia (sistema criptográfico de chave pública) que usa duas chaves: Public and Private Key (Chave pública e privada), também conhecida como par de chaves. A chave RSA pública é a chave de criptografia e a chave RSA privada é a chave de descryptografia.

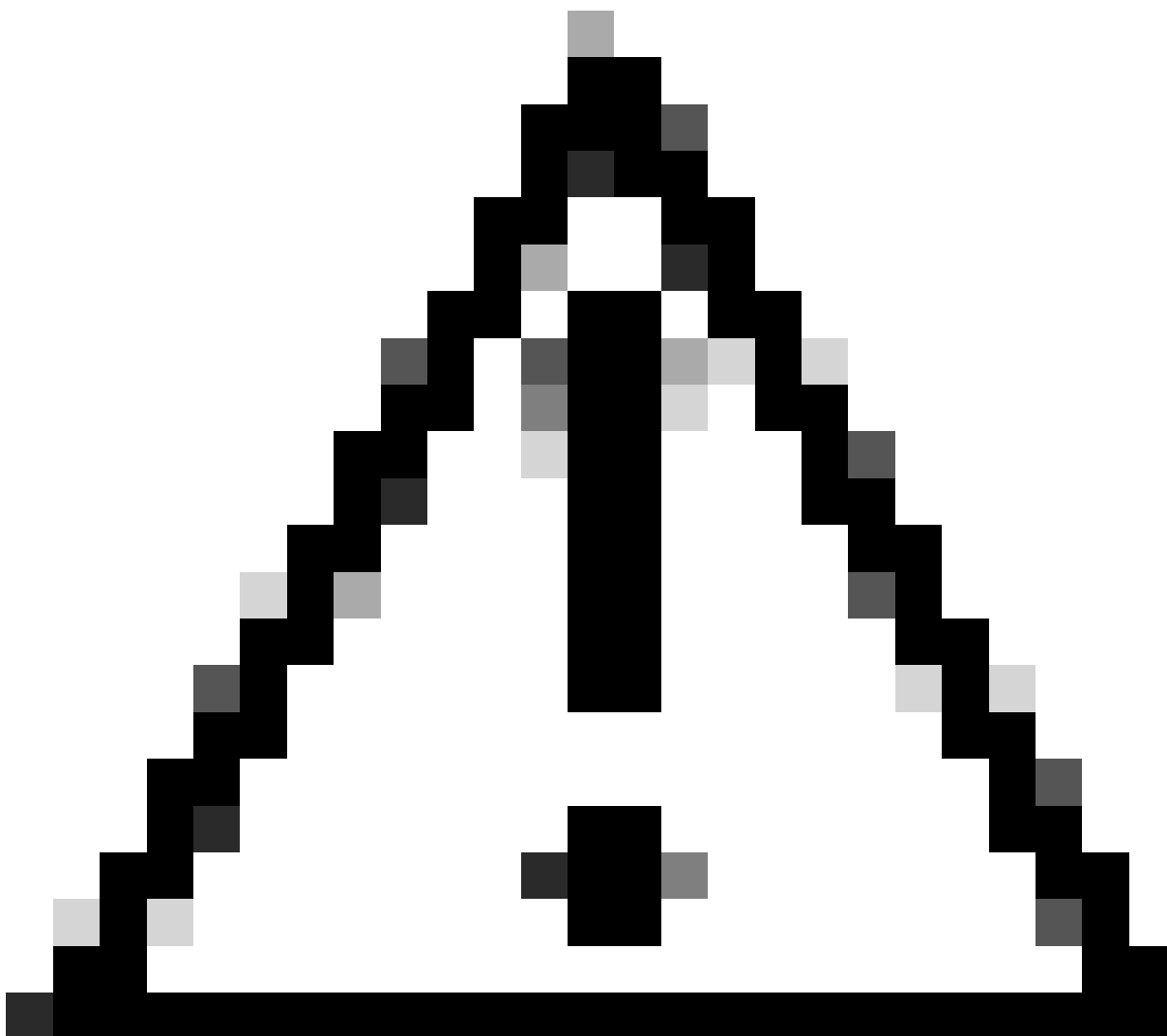
As chaves RSA têm um comprimento definido, em bits, do módulo. Quando se diz que uma chave RSA tem um comprimento de 2048 bits, isso realmente significa que o valor do módulo está entre 22047 e 22048. Uma vez que as chaves públicas e privadas de um determinado par compartilham o mesmo módulo, elas também têm, por definição, o mesmo comprimento.

Um certificado de ponto confiável é um certificado autoassinado, portanto, o nome ponto confiável, já que não depende da confiança de outra pessoa ou outra parte.

A infraestrutura de chave pública (PKI - Public Key Infrastructure) do Cisco IOS fornece gerenciamento de certificados para suportar protocolos de segurança como segurança IP (IPSec - IP Security), Secure Shell (SSH - Secure Shell) e Secure Socket Layer (SSL - Secure Socket Layer).

As chaves RSA do SSH são importantes no Cisco Catalyst SD-WAN porque são usadas pelo protocolo SSH para estabelecer a comunicação entre o gerenciador SD-WAN e os dispositivos de borda SD-WAN, já que o gerenciador SD-WAN usa o protocolo Netconf, que funciona através do SSH para gerenciar, configurar e monitorar dispositivos.

Devido a este fato, é necessário que as chaves sejam sincronizadas e atualizadas o tempo todo. Se, por conformidade e auditoria, for necessário modificar o comprimento da chave para segurança, será necessário concluir o processo descrito neste documento para redimensionar as chaves e sincronizá-las corretamente no certificado para evitar a desconexão entre o gerenciador de SD-WAN e os dispositivos de borda de SD-WAN.



Caution: Conclua todas as etapas do processo para evitar a perda de acesso ao dispositivo. Se o dispositivo estiver em produão,  recomendvel execut-lo em uma janela de manuteno e ter acesso de console ao dispositivo.

---

## Configurar

### Diagrama de Rede

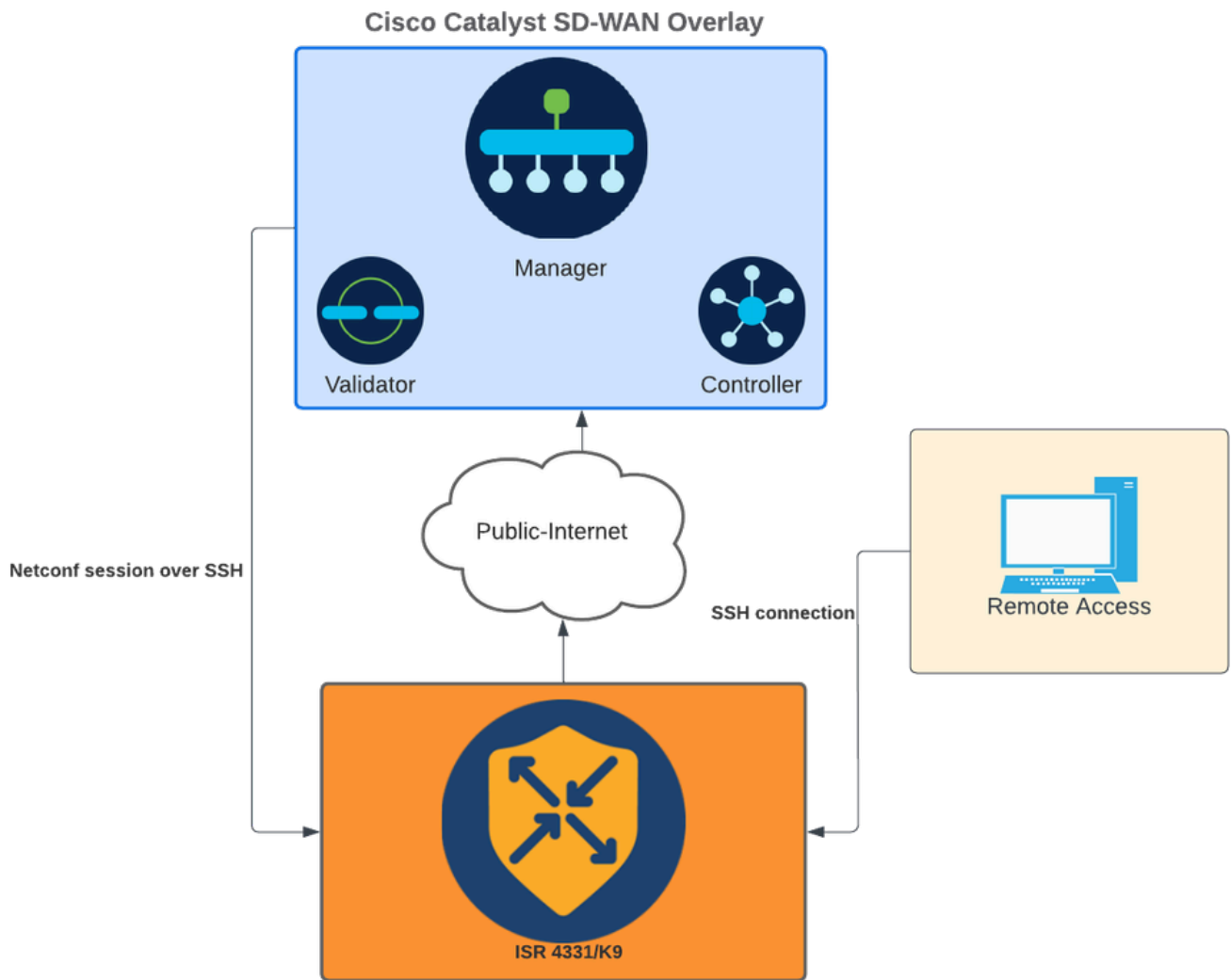


Diagrama de Rede

## Configurações

As chaves RSA nos dispositivos de borda da WAN só podem ser modificadas usando a interface de linha de comando (CLI); Não é possível usar modelos de recurso de complemento do CLI para atualizar as chaves.



aviso: É recomendável fazer o processo com o uso do console, pois a ferramenta SSH do gerenciador de SD-WAN não está disponível até que o processo seja concluído.

---



aviso: Este processo requer uma reinicialização do dispositivo. Se o dispositivo estiver em produção, é recomendável executá-lo em uma janela de manutenção e ter acesso de console ao dispositivo. Se não houver acesso ao console, configure temporariamente outro protocolo de acesso remoto como telnet.

---

Este exemplo de configuração mostra como remover o RSA 2048 e usar a chave RSA 4096.

1 - Obter o nome da chave SSH atual.

```
<#root>
```

```
Device#
```

```
show ip ssh
```

```
SSH Enabled - version 2.0
```

```
Authentication methods:publickey,keyboard-interactive,password
```

```
Authentication Publickey Algorithms:x509v3-ssh-rsa,ssh-rsa,ecdsa-sha2-nistp256,ecdsa-sha2-nistp384,ecdsa-sha2-nistp521
```

```
Hostkey Algorithms:x509v3-ssh-rsa,rsa-sha2-512,rsa-sha2-256,ssh-rsa
```

```
Encryption Algorithms:aes128-gcm,aes256-gcm,aes128-ctr,aes192-ctr,aes256-ctr
MAC Algorithms:hmac-sha2-256-etm@openssh.com,hmac-sha2-512-etm@openssh.com,hmac-sha2-256,hmac-sha2-512,
KEX Algorithms:ecdh-sha2-nistp256,ecdh-sha2-nistp384,ecdh-sha2-nistp521,diffie-hellman-group14-sha1
Authentication timeout: 120 secs; Authentication retries: 3
Minimum expected Diffie Hellman key size : 2048 bits
IOS Keys in SECSH format(ssh-rsa, base64 encoded):

TP-self-signed-1072201169 <<<< RSA Key Name
```

Modulus Size : 2048 bits

```
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQAZ5urq7f/X+AZJjUnM0dF9pLX+V0jPR8arK6bLSU7d
iGeSDDwW2MPNck/U5HBry9P/L4nKyZ1oEvAhfy7cJVvmoHD41NQW9wb/hLtimuujnRRYkKuIWLmoI7AH
y6YQoetew8XVg1VIjva+JzQ5ZX1JGm8AzN6a95RbRNhGRzgz9cTFmD7m6ArIKZPMYyQabXfrY+m/HuQ2
aytbHtJMgm0Qk2fLPak03PnQNYXpiDP3Cm0Eh3LJg82FZQ1eohmhm+mAIwU4m1LHUouigyBuq1KEBVe
z3vxjB9X8rGF3qzUcx21pHmhXaNpXWen2QQbyfAIDo8WxVoff24uLY1wCVkv
```

2 - Obter o certificado autoassinado do ponto confiável.

```
<#root>
```

```
Device#
```

```
show crypto pki trustpoint
```

```
Trustpoint TP-self-signed-1072201169: <<<< Self-signed Trustpoint name
```

```
Subject Name:
```

```
cn=IOS-Self-Signed-Certificate-1072201169
```

```
Serial Number (hex): 01
```

```
Persistent self-signed certificate trust point
```

```
Using key label
```

```
TP-self-signed-1072201169
```

Ambos os nomes de valores devem corresponder.

3 - Excluir a chave atual.

```
<#root>
```

```
Device#
```

```
crypto key zeroize rsa
```

4 - Validar que a chave antiga foi excluída com êxito.

```
<#root>
```

```
Device#
```

```
show ip ssh
```

5 - Gerar a nova chave.

```
<#root>
```

```
Device#
```

```
crypto key generate rsa modulus 4096 label
```

```
The name for the keys will be: TP-self-signed-1072201169
```

```
% The key modulus size is 4096 bits
```

```
% Generating crypto RSA keys in background ...
```

```
*Jun 25 21:35:18.919: %CRYPTO_ENGINE-5-KEY_ADDITION: A key named TP-self-signed-1072201169 has been generated
```

```
*Jun 25 21:35:18.924: %SSH-5-ENABLED: SSH 2.0 has been enabled
```

```
*Jun 25 21:35:23.205: %CRYPTO_ENGINE-5-KEY_ADDITION: A key named TP-self-signed-1072201169 has been generated
```

```
*Jun 25 21:35:29.674: %SYS-6-PRIVCFG_ENCRYPT_SUCCESS: Successfully encrypted private config file
```

Esse processo pode levar de 2 a 5 minutos para ser concluído.

6 - Validar a nova chave gerada.

```
<#root>
```

```
Device#
```

```
show ip ssh
```

```
SSH Enabled - version 2.0
```

```
Authentication methods:publickey,keyboard-interactive,password
```

```
Authentication Publickey Algorithms:x509v3-ssh-rsa,ssh-rsa,ecdsa-sha2-nistp256,ecdsa-sha2-nistp384,ecdsa-sha2-nistp521
```

```
Hostkey Algorithms:x509v3-ssh-rsa,rsa-sha2-512,rsa-sha2-256,ssh-rsa
```

```
Encryption Algorithms:aes128-gcm,aes256-gcm,aes128-ctr,aes192-ctr,aes256-ctr
```

```
MAC Algorithms:hmac-sha2-256-etm@openssh.com,hmac-sha2-512-etm@openssh.com,hmac-sha2-256,hmac-sha2-512,hmac-sha2-512-etm@openssh.com
```

```
KEX Algorithms:ecdh-sha2-nistp256,ecdh-sha2-nistp384,ecdh-sha2-nistp521,diffie-hellman-group14-sha1
```

```
Authentication timeout: 120 secs; Authentication retries: 3
```



Minimum expected Diffie Hellman key size : 2048 bits

IOS Keys in SECSH format(ssh-rsa, base64 encoded): TP-self-signed-1072201169

Modulus Size : 4096 bits <<<< Key Size

```
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQCAQDE0t/SX3oQKN6z0Wv0aFAkMcaZNzQ6JgP+7xjuX143
YS7YGmOPwIPgs8N2LWvmdLXQ/PqsQOGGsdxo2+2Y/idAFm808mb6bcWFU+t3b/Pf6GBzUv8SPnR4i4nN
5GYhZE9HX3REWYp7d+7l1YawrdzpJ6d8RgUWLOtgHSzQ7P796c0B1YLtK3eF00H1AFmFy5ec8Own7ik0
JjKtwEozImFMjHZfUEUjFuhPJELB06yYEipPwMRaZYffTRbNjM8/7S0JG1FkgFVW5nITTIgISoMV8EJv
bL18cVgATDb10ckDb7uU6PDXm3zonmZC0yqHtF10A0JxUpUa6Iry1XwMzzZqDdu32F5If4/SSCmbHV2
46P8AjCdu/2TKK5et0049UH0y0bMgPuWrJpwtk1iYA3+t6N/Qd1C5VSoua+TsMfp7Dh3k6qUTFUSy2h3
Kiibov1HKYvkcqXi6nDfAKb8o+Z8/43xbvW1DIKAuj1rbdyqPAJB411TZJk0Hk8zRP5gZ8u4jtjNKQHb
vNa3ieg4RLED0x41qCk+iSRzdddMq2te1xSWFPh67i4BnJHvhVnR6LF5Gu+uF5TWwcpy2MMOu14YDJYr
D+jnyoZr4PnfwAgk4M9U89deWS1IRPMIXYd35YmLvD60eQ5EQALNiNPUEkpdPKs4orYysEV0pRoY+HQ
```

Agora, uma nova chave é gerada. No entanto, no momento em que a chave antiga foi excluída, o certificado autoassinado que está em uso pelas sessões do Netconf também é excluído do ponto de confiança.

```
<#root>
```

```
Device#
```

```
sh crypto pki trustpoint status
```

```
Trustpoint TP-self-signed-1072201169:
```

```
Issuing CA certificate configured::
```

```
Issuing CA certificate configured:
```

```
Subject Name:
```

```
cn=Cisco Licensing Root CA,o=Cisco
```

```
Fingerprint MD5: 1468DC18 250BDFCF 769C29DF E1F7E5A8
```

```
Fingerprint SHA1: 5CA95FB6 E2980EC1 5AFB681B BB7E62B5 AD3FA8B8
```

```
State:
```


```
Keys generated ..... No <<<< Depending on the version, it can erase the key or even that, delete
```

```
Issuing CA authenticated ..... Yes
```

```
Certificate request(s) ..... None
```

Depois que a nova chave 4096 é gerada, as chaves não são atualizadas automaticamente no certificado autoassinado e é necessário concluir etapas extras para atualizá-lo.

---

 Note: Se apenas a chave for gerada, mas não for atualizada no certificado, o SD-WAN Manager perderá as sessões do Netconf e isso poderá interromper todas as atividades de gerenciamento para o dispositivo (modelos, configuração e assim por diante).

---

Há duas maneiras de gerar o certificado/atribuir a chave:

1 - Recarregar o dispositivo.

```
<#root>
```

```
Device#
```

```
reload
```

2 - Reiniciar o servidor seguro HTTP. Essa opção estará disponível apenas se o dispositivo estiver no modo CLI.

```
<#root>
```

```
Device (config)#
```

```
no ip http secure-server
```

```
Device (config)#
```

```
commit
```

```
Device (config)#
```

```
ip http secure-server
```

```
Device (config)#
```

```
commit
```

## Verificar

Após o recarregamento, valide se a nova chave foi gerada e se o certificado está em um ponto confiável com o mesmo nome.

```
<#root>
```

```
Device#
```

```
show ip ssh
```

```
SSH Enabled - version 2.0
```

```
Authentication methods:publickey,keyboard-interactive,password
```

```
Authentication Publickey Algorithms:x509v3-ssh-rsa,ssh-rsa,ecdsa-sha2-nistp256,ecdsa-sha2-nistp384,ecdsa-sha2-nistp521
```

```
Hostkey Algorithms:x509v3-ssh-rsa,rsa-sha2-512,rsa-sha2-256,ssh-rsa
```

```
Encryption Algorithms:aes128-gcm,aes256-gcm,aes128-ctr,aes192-ctr,aes256-ctr
```

```
MAC Algorithms:hmac-sha2-256-etm@openssh.com,hmac-sha2-512-etm@openssh.com,hmac-sha2-256,hmac-sha2-512,hmac-sha1
```

```
KEX Algorithms:ecdh-sha2-nistp256,ecdh-sha2-nistp384,ecdh-sha2-nistp521,diffie-hellman-group14-sha1
```

```
Authentication timeout: 120 secs; Authentication retries: 3
```

```
Minimum expected Diffie Hellman key size : 2048 bits
```

IOS Keys in SECSH format(ssh-rsa, base64 encoded): TP-self-signed-1072201169

Modulus Size : 4096 bits

```
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQCE0t/SX3oQKN6z0Wv0aFAkMcaZNzQ6JgP+7xjuX143
YS7YGmOPwIPgs8N2LWvmdLXQ/PqsQGGsdxo2+2Y/i dAFm808mb6bcWfU+t3b/Pf6GBzUv8SPnR4i4nN
5GYhZE9HX3REWYp7d+711YawrDzpJ6d8RgUWLOtghSszQ7P796c0B1YLtK3eFO0H1AFmFy5ec8Own7ik0
JjKtwEozImFmJHZfUEUjFuhPJELBO6yYEipPWMRaZYfFTRbNjM8/7SOJG1FkgFVW5nITTIgISoMV8EJv
bL18cVgATDb10ckeDb7uU6PDXm3zonmZC0yqHtF10A0JxUpUa6Iry1XwMzzZqDdu32F5If4/SSCmbHV2
46P8AjCdu/2TKK5et0049UH0y0bMgPuWrJpwtk1iYA3+t6N/Qd1C5VSoua+Tsmfp7Dh3k6qUTFUSy2h3
Kiibov1HKyvkcqXi6nDfAKb8o+Z8/43xbvW1DIKAuj1rbdyqPAJB411TZJkOHk8zRP5gZ8u4jTjNKQHb
vNa3ieg4RLED0x41qCk+iSRzdddMq2te1xSWFPh67i4BnJHvhVnR6LF5Gu+uF5TWwcpy2MMOu14YDJYr
D+jnyoZr4PnfwAgk4M9U89deWS1IRPMIXYd35YmLvD60eQ5EQALNiNPUEkpdPKs4orYysEV0pRoY+HQ
```

<#root>

Device#

```
show crypto pki trustpoint
```

```
Trustpoint TP-self-signed-1072201169: <<<< Trustpoint name
```

Subject Name:

cn=IOS-Self-Signed-Certificate-1072201169

Serial Number (hex): 01

Persistent self-signed certificate trust point

Using key label TP-self-signed-107220116

<#root>

Device#

```
show crypto pki certificates
```

Router Self-Signed Certificate

Status: Available

Certificate Serial Number (hex): 01

Certificate Usage: General Purpose

Issuer:

cn=IOS-Self-Signed-Certificate-1072201169

Subject:

Name: IOS-Self-Signed-Certificate-1072201169

cn=IOS-Self-Signed-Certificate-1072201169

Validity Date:

start date: 21:07:33 UTC Dec 27 2023

end date: 21:07:33 UTC Dec 26 2033

Associated Trustpoints: TP-self-signed-1072201169

Storage: nvram:IOS-Self-Sig#4.cer

Confirme se o SD-WAN Manager pode aplicar alterações de configuração ao roteador do dispositivo.

## 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.