Configuration de SSH sur les routeurs

Table des matières

- Introduction
- Conditions préalables
  - Exigences
  - Composants utilisés
- Diagramme du réseau
  - Configuration de SSH sur un routeur Cisco avec authentification par mot de passe
  - Configurer SSH sur un routeur Cisco à l'aide de l'authentification par clé publique RSA
- Jeux de configuration supplémentaires
  - Configuration de SSH version 2
  - Empêcher les connexions non SSH
  - Restreindre l'accès SSH à un sous-réseau
  - Variations du résultat de la commande Banner
- Commandes Debug et Show
- Exemple de débogage Authentification publique RSA
  - Débogage du client SSH (Linux/macOS)
  - Débogage du serveur SSH (routeur Cisco)
- Configurations incorrectes
  - Mot de passe incorrect
  - Le client SSH envoie un chiffrement non pris en charge
  - Le client SSH envoie un HMAC non pris en charge
  - Erreur « %SSH-3-PRIVATEKEY : Impossible de récupérer la clé privée RSA"
- Recommandations
- Informations connexes

Introduction

Ce document décrit comment configurer et déboguer Secure Shell (SSH) sur les routeurs Cisco.

Conditions préalables

Exigences

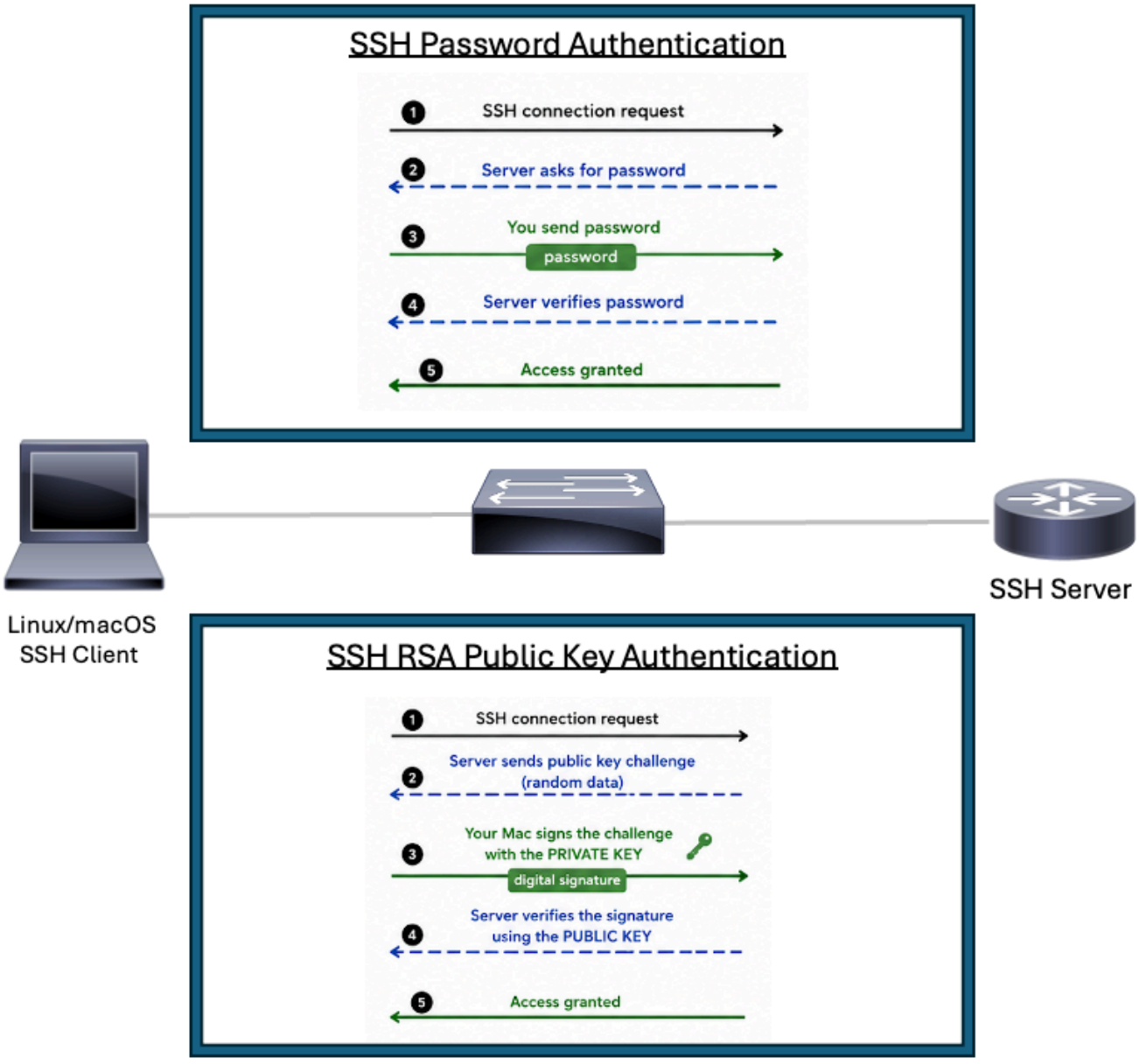
L'image ou le package logiciel Cisco doit inclure la prise en charge du cryptage pour SSH. Les exigences varient selon la plate-forme, la version du logiciel et le modèle de licence.

Composants utilisés

Les informations contenues dans ce document sont basées sur les logiciels Cisco IOS et Cisco IOS XE.

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. Si votre réseau est en ligne, assurez-vous de bien comprendre l'incidence possible des commandes.

Diagramme du réseau



Topologie SSH

Configuration de SSH sur un routeur Cisco avec authentification par mot de passe

Quelques étapes sont nécessaires pour activer la prise en charge SSH de base sur un routeur Cisco :

1. Configurez le nom d’hôte du routeur.

2. Créez un compte utilisateur local pour l'authentification SSH.

3. Configurez le nom de domaine DNS.

4. Activez la version SSH la plus élevée prise en charge par la plate-forme. Utiliser SSH version 2 si pris en charge.

5. Générez la paire de clés RSA pour SSH. Utilisez le plus grand module RSA pris en charge par la plate-forme et aligné sur votre politique de sécurité.

6. Passez en mode de configuration de ligne VTY.

7. Configurez les lignes VTY pour utiliser la base de données utilisateur locale ou aaa pour l'authentification.

8. Limitez l'accès VTY au transport SSH uniquement.

Le périphérique Cisco qui agit en tant que serveur SSH nécessite le nom d'hôte, le nom de domaine, les clés RSA, l'authentification et la configuration VTY. Un périphérique qui lance uniquement des sessions SSH nécessite la prise en charge du client SSH et une route valide vers le serveur SSH.

<#root>

Routert#

configure terminal

Router(config)#

hostname R2

R2(config)#

username cisco privilege 15 secret C1sc0

```
R2(config)#
ip domain name rtp.cisco.com
R2(config)#
ip ssh version 2
R2(config)#
crypto key generate rsa general-keys modulus 4096
R2(config)#
line vty 0 4
R2(config-line)#
login local
R2(config-line)#
transport input ssh
R2(config-line)#
end
```



Remarque : Si login local n'est pas utilisé, la commande aaa new-model entraîne l'utilisation du nom d'utilisateur et du secret locaux sur le routeur en l'absence d'autres instructions AAA.

Utilisez cette commande comme exemple de commande client SSH lors de tests à partir d'un autre routeur. Les algorithmes de chiffrement et hmac pris en charge peuvent varier selon la version du logiciel et la politique SSH :

<#root>

```
R3#
ssh -v 2 -l cisco 10.20.20.20


Password:
R2>
```

Configurer SSH sur un routeur Cisco à l'aide de l'authentification par clé publique RSA

L'authentification par clé publique RSA permet à un client SSH de s'authentifier auprès d'un routeur ou d'un commutateur Cisco sans envoyer de mot de passe ni s'y fier. Le serveur SSH stocke la clé publique. Le client SSH conserve la clé privée correspondante. Lors de la connexion, le client prouve qu'il possède la clé privée en signant numériquement les données d'authentification. Le serveur vérifie cette signature avec la clé publique configurée sur le périphérique. La clé privée reste sur le client et n'est pas transmise sur le réseau.

Avantages

- Élimine l'authentification SSH par mot de passe pour l'utilisateur configuré.
- Réduit l'exposition aux devinettes de mots de passe et aux attaques en force.
- Conserve la clé privée sur le client SSH au lieu de stocker des mots de passe réutilisables sur des périphériques réseau ou des systèmes d'automatisation.
- Fonctionne bien pour les outils d'automatisation, les scripts et les systèmes CI/CD qui nécessitent un accès sécurisé aux périphériques Cisco.
- Améliore la cohérence opérationnelle, car le serveur valide une clé publique approuvée au lieu d'un mot de passe tapé.
- Prend en charge des modèles d'authentification plus robustes pour les postes de travail des administrateurs, les hôtes de saut et les serveurs d'automatisation.

Complétez ces étapes pour configurer le serveur SSH pour effectuer l'authentification RSA basée sur la clé publique :

1. Configurez le nom d’hôte du routeur.

2. Configurez le nom de domaine DNS.

3. Activez AAA ou configurez la connexion locale sous la ligne vty.

4. Générez la paire de clés RSA utilisée par le serveur SSH.

5. Utilisez la version SSH la plus élevée prise en charge par la plate-forme. Utiliser SSHv2 lorsqu'il est pris en charge.

6. Utilisez le module RSA le plus puissant pris en charge par la plate-forme et la stratégie de sécurité. Un module de 4 096 bits est préférable lorsqu'il est pris en charge et lorsque l'impact sur

les performances est acceptable.

7. Passez en mode de configuration SSH public-key chain.

8. Configurez le nom d'utilisateur SSH qui est autorisé à s'authentifier avec l'authentification de clé publique RSA.

9. Ajoutez la clé publique RSA du client SSH pour ce nom d'utilisateur.



Remarque : Vous pouvez générer votre clé publique SSH à partir de votre périphérique Linux/macOS avec la commande ssh-keygen, par exemple, ssh-keygen -m PEM -t rsa -b 1024 -f ~/.ssh/cisco1024 après, vous pouvez lire la clé avec la commande cat ~/.ssh/cisco1024.pub.

<#root>

Router#

configure terminal

Router(config)#

hostname RSA-PublicKey

RSA-PublicKey(config)#

ip domain name rtp.cisco.com

RSA-PublicKey(config)#

aaa new-model

RSA-PublicKey(config)#

crypto key generate rsa modulus 4096

The name for the keys will be: RSA-PublicKey.rtp.cisco.com

% The key modulus size is 4096 bits
% Generating 4096 bit RSA keys, keys will be non-exportable...
[OK] (elapsed time was 4 seconds)

RSA-PublicKey(config)#

ip ssh version 2

```
RSA-PublicKey(config)#
ip ssh pubkey-chain
RSA-PublicKey(conf-ssh-pubkey)#
username ciscoadmin
RSA-PublicKey(conf-ssh-pubkey-user)#
key-string
RSA-PublicKey(conf-ssh-pubkey-data)#
$QaVSyuVNw== user@MacBook-Pro
RSA-PublicKey(conf-ssh-pubkey-data)#
end
RSA-PublicKey#
```

Jeux de configuration supplémentaires

Configuration de SSH version 2

Lors de la configuration de SSH, assurez-vous que SSHv2 est activé, car il offre un chiffrement plus fort et une sécurité nettement supérieure à SSHv1.

```
<#root>
Router(config)#
ip ssh version 2
```

Empêcher les connexions non SSH

Si vous voulez empêcher les connexions non SSH, ajoutez la commande transport input ssh sous les lignes pour limiter le routeur aux connexions SSH seulement. Les Telnet droits (non ssh) sont

refusés.

<#root>

Router(config)#

line vty 0 15

Router(config-line)#

transport input ssh

Restreindre l'accès SSH à un sous-réseau

Vous devez limiter la connectivité SSH à un sous-réseau précis, où toutes les autres tentatives du protocole SSH des adresses IP externes au sous-réseau sont abandonnées.

Vous pouvez faire de même en suivant ces étapes :

1. Définissez une liste de contrôle d’accès qui autorise le trafic provenant de ce sous-réseau spécifique.
2. Restreignez l'accès à l'interface de ligne VTY avec une `access-class`.

Ceci est un exemple de configuration. Dans cet exemple, seul l'accès SSH à partir de 10.10.10.0/24 est autorisé ; toutes les autres adresses IP source sont refusées par le refus implicite à la fin de la liste de contrôle d’accès.

<#root>

Router(config)#

access-list 23 permit 10.10.10.0 0.0.0.255

Router(config)#

line vty 0 15

Router(config-line)#

transport input ssh

Router(config-line)#

```
access-class 23 in

Router(config-line)#

exit
```

Remarque : Vérifiez la plage de lignes VTY disponible sur le périphérique, puis appliquez la liste de contrôle d'accès à toutes les lignes VTY utilisées pour l'accès à distance.

Variations du résultat de la commande Banner

Le résultat de la commande banner varie entre Telnet et différentes versions de connexions SSH. Ce tableau montre comment différentes options de commande de bannière fonctionnent avec différents types de connexions.

Options de commande de la bannière	Telnet	SSH v2
connexion bannière	S'affiche avant la connexion à l'appareil.	S'affiche avant la connexion à l'appareil.
banner motd	S'affiche avant la connexion à l'appareil.	S'affiche après la connexion à l'appareil.
banner exec	S'affiche après la connexion à l'appareil.	S'affiche après la connexion à l'appareil.

Remarque : SSH version 1 n'est plus recommandé.

Commandes Debug et Show

Avant d'émettre les commandes debug décrites ici, référez-vous à Comprendre les informations importantes sur les commandes de débogage.

debug ip ssh - Active les messages de débogage pour SSH.

show ssh - Affiche l'état des connexions du serveur SSH.

<#root>

RSA-PublicKey#

show ssh

Connection Version Mode Encryption Hmac   State   Username
0   2.0

IN

   aes128-gcm@ohmac-sha2-256-etm@openssh.com Session started

ciscoadmin

0   2.0

OUT

  aes128-gcm@ohmac-sha2-256-etm@openssh.com Session started

ciscoadmin

show ip ssh - Affiche les données de version et de configuration pour SSH.

<#root>

RSA-PublicKey#

show ip ssh

SSH Enabled - version 2.0

Authentication methods:publickey,keyboard-interactive,password
Authentication Publickey Algorithms:ssh-rsa,ecdsa-sha2-nistp256,ecdsa-sha2-nistp384,ecdsa-sha2-nistp521
Hostkey Algorithms:rsa-sha2-512,rsa-sha2-256,ssh-rsa
Encryption Algorithms:chacha20-poly1305@openssh.com,aes128-gcm@openssh.com,aes256-gcm@openssh.com,aes12
MAC Algorithms:hmac-sha2-256-etm@openssh.com,hmac-sha2-512-etm@openssh.com
KEX Algorithms:curve25519-sha256,curve25519-sha256@libssh.org,ecdh-sha2-nistp256,ecdh-sha2-nistp384,ecdh

Authentication timeout: 120 secs; Authentication retries: 3

Exemple de débogage Authentification publique RSA

Débogage du client SSH (Linux/macOS)

<#root>

ciscouser@CISCOUSER-M-R Desktop % ssh -i ~/.ssh/cisco1024 -v ciscoadmin@10.82.139.71
debug1: OpenSSH_10.2p1, LibreSSL 3.3.6
debug1: Reading configuration data /etc/ssh/ssh_config
debug1: Reading configuration data /etc/ssh/ssh_config.d/100-macos.conf
debug1: /etc/ssh/ssh_config.d/100-macos.conf line 1: Applying options for *
debug1: Reading configuration data /etc/ssh/crypto.conf
debug1: Authenticator provider $SSH_SK_PROVIDER did not resolve; disabling
debug1: Connecting to 10.82.139.71 [10.82.139.71] port 22.
debug1: Connection established.
debug1: loaded pubkey from /Users/ciscouser/.ssh/cisco1024: RSA SHA256:w5wW3BqX15fO5Qm+BP75tJ1YzaXNsDNeHLLNZTW+Roo
debug1: identity file /Users/ciscouser/.ssh/cisco1024 type 0
debug1: no identity pubkey loaded from /Users/ciscouser/.ssh/cisco1024
debug1: Local version string SSH-2.0-OpenSSH_10.2
debug1: Remote protocol version 2.0, remote software version Cisco-1.25
debug1: compat_banner: match: Cisco-1.25 pat Cisco-1.* compat 0x60000000
debug1: Authenticating to 10.82.139.71:22 as 'ciscoadmin'
debug1: load_hostkeys: fopen /Users/ciscouser/.ssh/known_hosts2: No such file or directory
debug1: load_hostkeys: fopen /etc/ssh/ssh_known_hosts: No such file or directory
debug1: load_hostkeys: fopen /etc/ssh/ssh_known_hosts2: No such file or directory
debug1: SSH2_MSG_KEXINIT sent
debug1: SSH2_MSG_KEXINIT received
debug1: kex: algorithm: ecdh-sha2-nistp256
debug1: kex: host key algorithm: rsa-sha2-512
debug1: kex: server->client cipher: aes128-gcm@openssh.com MAC: <implicit> compression: none
debug1: kex: client->server cipher: aes128-gcm@openssh.com MAC: <implicit> compression: none
debug1: expecting SSH2_MSG_KEX_ECDH_REPLY
debug1: SSH2_MSG_KEX_ECDH_REPLY received
debug1: Server host key: ssh-rsa SHA256:5wvYU0FIJQmz+eHACVJMvoaX3WSgXkyem/3tOgyOYuQ
debug1: load_hostkeys: fopen /Users/ciscouser/.ssh/known_hosts2: No such file or directory
debug1: load_hostkeys: fopen /etc/ssh/ssh_known_hosts: No such file or directory
debug1: load_hostkeys: fopen /etc/ssh/ssh_known_hosts2: No such file or directory
debug1: Host '10.82.139.71' is known and matches the RSA host key.
debug1: Found key in /Users/ciscouser/.ssh/known_hosts:2
debug1: rekey out after 4294967296 blocks
debug1: SSH2_MSG_NEWKEYS sent
debug1: expecting SSH2_MSG_NEWKEYS
debug1: SSH2_MSG_NEWKEYS received
debug1: rekey in after 4294967296 blocks
debug1: SSH2_MSG_EXT_INFO received
debug1: kex_ext_info_client_parse: server-sig-algs=<ssh-rsa,ecdsa-sha2-nistp256,ecdsa-sha2-nistp384,ecdsa-sha2-nistp521,ssh-ed25519,x509v3-ecdsa-sha

debug1: SSH2_MSG_SERVICE_ACCEPT received

debug1: Authentications that can continue: publickey,keyboard-interactive,password
debug1: Next authentication method: publickey
debug1: get_agent_identities: bound agent to hostkey
debug1: get_agent_identities: ssh_fetch_identitylist: agent contains no identities

debug1: Will attempt key: /Users/ciscouser/.ssh/cisco1024 RSA SHA256:w5wW3BqX15fO5Qm+BP75tJ1YzaXNsDNeHLL

debug1: channel 0: new session [client-session] (inactive timeout: 0)

```
debug1: Entering interactive session.
debug1: pledge: filesystem
debug1: Sending environment.
debug1: channel 0: setting env LANG = "C.UTF-8"
debug1: channel 0: setting env LC_CTYPE = "UTF-8"
```

RSA-PublicKey>en Password:

RSA-PublicKey#

show version

```
Cisco IOS XE Software, Version 17.12.04
Cisco IOS Software [Dublin], ASR1000 Software (X86_64_LINUX_IOSD-UNIVERSALK9-M), Version 17.12.4, RELEA
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2024 by Cisco Systems, Inc.
Compiled Tue 23-Jul-24 09:43 by mcpre


Cisco IOS-XE software, Copyright (c) 2005-2024 by cisco Systems, Inc.
All rights reserved.  Certain components of Cisco IOS-XE software are
licensed under the GNU General Public License ("GPL") Version 2.0.  The
software code licensed under GPL Version 2.0 is free software that comes
with ABSOLUTELY NO WARRANTY.  You can redistribute and/or modify such
GPL code under the terms of GPL Version 2.0.  For more details, see the
documentation or "License Notice" file accompanying the IOS-XE software,
or the applicable URL provided on the flyer accompanying the IOS-XE
software.


ROM: 17.3(1r)

RSA-PublicKey uptime is 2 weeks, 2 days, 10 hours, 28 minutes
Uptime for this control processor is 2 weeks, 2 days, 10 hours, 30 minutes
System returned to ROM by Reload Command
System image file is "bootflash:asr1000-universalk9.17.12.04.SPA.bin"
Last reload reason: Reload Command



This product contains cryptographic features and is subject to United
States and local country laws governing import, export, transfer and
use. Delivery of Cisco cryptographic products does not imply
third-party authority to import, export, distribute or use encryption.
Importers, exporters, distributors and users are responsible for
compliance with U.S. and local country laws. By using this product you
agree to comply with applicable laws and regulations. If you are unable
to comply with U.S. and local laws, return this product immediately.

A summary of U.S. laws governing Cisco cryptographic products may be found at:
http://www.cisco.com/wwl/export/crypto/tool/stqrg.html

If you require further assistance please contact us by sending email to
export@cisco.com.

License Type: Smart License is permanent
License Suite: AdvUCSuiteK9
Next reload License Suite: AdvUCSuiteK9
```

```
The current crypto throughput level is 20000000 kbps


Smart Licensing Status: Smart Licensing Using Policy

cisco ASR1001-HX (1SR) processor (revision 1SR) with 3684784K/6147K bytes of memory.
Processor board ID TKM20160032
Router operating mode: Autonomous
Crypto Hardware Module present
8 Gigabit Ethernet interfaces
8 Ten Gigabit Ethernet interfaces
32768K bytes of non-volatile configuration memory.
8388608K bytes of physical memory.
29401087K bytes of eUSB flash at bootflash:.

Configuration register is 0x2102


RSA-PublicKey#exit debug1: client_input_channel_req: channel 0 rtype exit-status reply 0 debug1: channel


Transferred: sent 3640, received 6336 bytes, in 114.1 seconds
Bytes per second: sent 31.9, received 55.5
debug1: Exit status 0
```

Débogage du serveur SSH (routeur Cisco)

<#root>

RSA-PublicKey#

show logging

```
Syslog logging: enabled (0 messages dropped, 3 messages rate-limited, 0 flushes, 0 overruns, xml disabl

No Active Message Discriminator.



No Inactive Message Discriminator.


    Console logging: disabled
    Monitor logging: level debugging, 0 messages logged, xml disabled,
                     filtering disabled
    Buffer logging:  level debugging, 617 messages logged, xml disabled,
                    filtering disabled
    Exception Logging: size (4096 bytes)
    Count and timestamp logging messages: disabled
    Persistent logging: disabled

No active filter modules.

    Trap logging: level informational, 161 message lines logged
        Logging Source-Interface:       VRF Name:
    TLS Profiles:
```

```
Log Buffer (102400 bytes):

*Aug 22 06:05:51.736: SSH2 0: ssh_receive: 36 bytes received
*Aug 22 06:05:58.163: SSH0: starting SSH control process
*Aug 22 06:05:58.163: SSH0: sent protocol version id SSH-2.0-Cisco-1.25
*Aug 22 06:05:58.165: SSH0: protocol version id is - SSH-2.0-OpenSSH_10.2
*Aug 22 06:05:58.165: SSH2 0:
ssh2_server_key_exchange: kexinit sent: kex algo = curve25519-sha256,curve25519-sha256@libssh.org,ecdh-
*Aug 22 06:05:58.165: SSH2 0: kexinit sent: hostkey algo = rsa-sha2-512,rsa-sha2-256,ssh-rsa
*Aug 22 06:05:58.165: SSH2 0: kexinit sent: encryption algo = chacha20-poly1305@openssh.com,aes128-gcm@
*Aug 22 06:05:58.165: SSH2 0: kexinit sent: mac algo = hmac-sha2-256-etm@openssh.com,hmac-sha2-512-etm@
*Aug 22 06:05:58.165: SSH2 0: send:packet of  length 656 (length also includes padlen of 7)
*Aug 22 06:05:58.165: SSH2 0: SSH2_MSG_KEXINIT sent
*Aug 22 06:05:58.239: SSH2 0: ssh_receive: 1300 bytes received
*Aug 22 06:05:58.239: SSH2 0: input: total packet length of 1568 bytes
*Aug 22 06:05:58.239: SSH2 0: partial packet length(block_size)8 bytes,
            needed 1560 bytes, maclen 0
*Aug 22 06:05:58.239: SSH2 0: input: padlength 4 bytes
*Aug 22 06:05:58.239: SSH2 0: SSH2_MSG_KEXINIT received
*Aug 22 06:05:58.239: SSH2 0: kex: client->server enc:aes128-gcm@openssh.com mac:hmac-sha2-256-etm@open
*Aug 22 06:05:58.239: SSH2 0: kex: server->client enc:aes128-gcm@openssh.com mac:hmac-sha2-256-etm@open
*Aug 22 06:05:58.239: SSH2 0: Using kex_algo = ecdh-sha2-nistp256
*Aug 22 06:05:58.242: SSH2 0: kexecdh_server: expecting SSH2_MSG_KEX_ECDH_INIT
*Aug 22 06:05:58.312: SSH2 0: ssh_receive: 80 bytes received
*Aug 22 06:05:58.312: SSH2 0: input: total packet length of 80 bytes
*Aug 22 06:05:58.312: SSH2 0: partial packet length(block_size)8 bytes,
            needed 72 bytes, maclen 0
*Aug 22 06:05:58.312: SSH2 0: input: padlength 5 bytes
*Aug 22 06:05:58.312: ssh2_calculate_modulus_length: modulus len 32
*Aug 22 06:05:58.312: SSH2 0: kexecdh_server: Calculated shared secret len 32
*Aug 22 06:05:58.333: SSH2 0: send:packet of  length 1160 (length also includes padlen of 10)
*Aug 22 06:05:58.333: SSH2: kex_derive_keys complete
*Aug 22 06:05:58.333: SSH2 0: send:packet of  length 16 (length also includes padlen of 10)
*Aug 22 06:05:58.333: SSH2 0: newkeys: mode 1
*Aug 22 06:05:58.333: SSH2: AES-GCM key initialized
*Aug 22 06:05:58.333: SSH2 0: SSH2_MSG_NEWKEYS sent
*Aug 22 06:05:58.333: SSH2 0: Preparing SSH2_MSG_EXT_INFO server-sig-algs<ssh-rsa,ecdsa-sha2-nistp256,e
*Aug 22 06:05:58.333: SSH2 0: send:packet of  length 260 (length also includes padlen of 19)
*Aug 22 06:05:58.333: SSH2 0: SSH2_MSG_EXT_INFO sent
*Aug 22 06:05:58.333: SSH2 0: waiting for SSH2_MSG_NEWKEYS
*Aug 22 06:05:58.411: SSH2 0: ssh_receive: 16 bytes received
*Aug 22 06:05:58.411: SSH2 0: input: total packet length of 16 bytes
*Aug 22 06:05:58.411: SSH2 0: partial packet length(block_size)8 bytes,
            needed 8 bytes, maclen 0
*Aug 22 06:05:58.411: SSH2 0: input: padlength 10 bytes
*Aug 22 06:05:58.411: SSH2 0: newkeys: mode 0
*Aug 22 06:05:58.411: SSH2: AES-GCM key initialized
*Aug 22 06:05:58.411: SSH2 0: SSH2_MSG_NEWKEYS received
*Aug 22 06:05:58.411: %SSH-5-SSH2_SESSION: SSH2 Session request from 10.189.3.9 (tty = 0) using crypto
*Aug 22 06:05:58.604: SSH2 0: ssh_receive: 52 bytes received
*Aug 22 06:05:58.604: SSH2 0: input: total packet length of 36 bytes
*Aug 22 06:05:58.604: SSH2 0: partial packet length(block_size)16 bytes,
            needed 32 bytes, maclen 0
*Aug 22 06:05:58.604: SSH2 0: input: padlength 14 bytes
*Aug 22 06:05:58.604: SSH2 0: send:packet of  length 36 (length also includes padlen of 14)
*Aug 22 06:05:58.604: SSH2 0: Authentications that can continue = publickey,keyboard-interactive,passwo
*Aug 22 06:05:58.677: SSH2 0: ssh_receive: 68 bytes received
*Aug 22 06:05:58.677: SSH2 0: input: total packet length of 52 bytes
*Aug 22 06:05:58.677: SSH2 0: partial packet length(block_size)16 bytes,
            needed 48 bytes, maclen 0
*Aug 22 06:05:58.677: SSH2 0: input: padlength 6 bytes
*Aug 22 06:05:58.677: SSH2 0: Using method = none
```

```
*Aug 22 06:05:58.677: SSH2 0: Authentications that can continue = publickey,keyboard-interactive,passwo
*Aug 22 06:05:58.677: SSH2 0: send:packet of  length 68 (length also includes padlen of 18)
*Aug 22 06:05:58.755: SSH2 0: ssh_receive: 244 bytes received
*Aug 22 06:05:58.755: SSH2 0: input: total packet length of 228 bytes
*Aug 22 06:05:58.755: SSH2 0: partial packet length(block_size)16 bytes,
            needed 224 bytes, maclen 0
*Aug 22 06:05:58.755: SSH2 0: input: padlength 5 bytes

*Aug 22 06:05:58.755: SSH2 0: Using method = publickey *Aug 22 06:05:58.755: SSH2 0: Received publickey

*Aug 22 06:05:58.755: SSH2 0: send:packet of  length 196 (length also includes padlen of 19)
*Aug 22 06:05:58.755: SSH2 0: ssh2_validate_pubkey: Authenticating 'ciscoadmin' with method: publickey
*Aug 22 06:06:02.615: SSH2 0: ssh_receive: 404 bytes received
*Aug 22 06:06:02.615: SSH2 0: input: total packet length of 388 bytes
*Aug 22 06:06:02.615: SSH2 0: partial packet length(block_size)16 bytes,
            needed 384 bytes, maclen 0
*Aug 22 06:06:02.615: SSH2 0: input: padlength 13 bytes

*Aug 22 06:06:02.615: SSH2 0: ssh2_verify_pubkey_sign: Verifying signature for user 'ciscoadmin' in SSH2

*Aug 22 06:06:02.616: SSH2 0: passed: Pubkey Authentication ssh2_authenticate_pubkey for user 'ciscoadmi

*Aug 22 06:06:02.616: SSH2 0: send:packet of  length 20 (length also includes padlen of 14)
*Aug 22 06:06:02.617: SSH2 0: authentication successful for ciscoadmin

*Aug 22 06:06:02.617: %SEC_LOGIN-5-LOGIN_SUCCESS: Login Success [user: ciscoadmin] [Source: 10.189.3.9]

*Aug 22 06:06:02.617:  SSH: bulk-mode applying server configs
*Aug 22 06:06:02.617:  SSH: bulk_mode applied server configs
```

Configurations incorrectes

Ces sections contiennent des exemples de sortie de débogage provenant de plusieurs configurations incorrectes.

Mot de passe incorrect

Débogage du routeur

<#root>

```
*Jul 29 23:45:34.428: SSH2 0: input: padlength 9 bytes
*Jul 29 23:45:34.465: SSH0: password authentication failed for cisco
*Jul 29 23:45:36.465: SSH2 0: Authentications that can continue = publickey,keyboard-interactive,password
*Jul 29 23:45:36.465: SSH2 0: send:packet of  length 60 (length also includes padlen of 10)
*Jul 29 23:45:39.444: SSH2 0: ssh_receive: 64 bytes received
*Jul 29 23:45:39.444: SSH2 0: input: total packet length of 60 bytes
*Jul 29 23:45:39.444: SSH2 0: partial packet length(block_size)8 bytes,
```

needed 56 bytes, maclen 0
*Jul 29 23:45:39.444: SSH2 0: input: padlength 5 bytes
*Jul 29 23:45:39.444: SSH2 0: Using method = password

***Jul 29 23:45:39.482: SSH0: password authentication failed for cisco *Jul 29 23:45:39.483: SSH0: AAA aut**

Le client SSH envoie un chiffrement non pris en charge

Débogage du routeur

<#root>

R2#
*Jul 29 23:43:00.126: SSH0: starting SSH control process
*Jul 29 23:43:00.126: SSH0: sent protocol version id SSH-2.0-Cisco-1.25
*Jul 29 23:43:00.129: SSH0: protocol version id is - SSH-2.0-Cisco-1.25
*Jul 29 23:43:00.129: SSH2 0:
ssh2_server_key_exchange: kexinit sent: kex algo = curve25519-sha256,curve25519-sha256@libssh.org,ecdh-sha2-nistp256,ecdh-sha2-nistp384,ecdh-sha2
*Jul 29 23:43:00.129: SSH2 0: Host-Key not configured. Skipping algo = ecdsa-sha2-nistp256
*Jul 29 23:43:00.129: SSH2 0: Host-Key not configured. Skipping algo = ecdsa-sha2-nistp384
*Jul 29 23:43:00.129: SSH2 0: Host-Key not configured. Skipping algo = ecdsa-sha2-nistp521
*Jul 29 23:43:00.129: SSH2 0: kexinit sent: hostkey algo = rsa-sha2-512,rsa-sha2-256
*Jul 29 23:43:00.129: SSH2 0: kexinit sent: encryption algo = chacha20-poly1305@openssh.com,aes128-gcm@openssh.com,aes256-gcm@openssh.com,a
*Jul 29 23:43:00.130: SSH2 0: kexinit sent: mac algo = hmac-sha2-256-etm@openssh.com,hmac-sha2-512-etm@openssh.com,hmac-sha2-256,hmac-sha2-
*Jul 29 23:43:00.130: SSH2 0: send:packet of length 736 (length also includes padlen of 10)
*Jul 29 23:43:00.130: SSH2 0: SSH2_MSG_KEXINIT sent
*Jul 29 23:43:00.131: SSH2 0: ssh_receive: 64 bytes received
*Jul 29 23:43:00.131: SSH2 0: input: total packet length of 424 bytes
*Jul 29 23:43:00.131: SSH2 0: partial packet length(block_size)8 bytes,
needed 416 bytes, maclen 0
*Jul 29 23:43:00.131: SSH2 0: input: padlength 10 bytes
*Jul 29 23:43:00.131: SSH2 0: SSH2_MSG_KEXINIT received
*Jul 29 23:43:00.131: SSH2 0: strict kex is enabled

***Jul 29 23:43:00.131: %SSH-3-NO_MATCH: No matching cipher found: client 3des-cbc server chacha20-poly130**

Le client SSH envoie un HMAC non pris en charge

Débogage du routeur

<#root>

R2#
*Jul 29 23:36:06.824: %SYS-4-LOG_CLEARED: Logging buffer was cleared by user Unknown
*Jul 29 23:36:11.418: SSH0: starting SSH control process
*Jul 29 23:36:11.418: SSH0: sent protocol version id SSH-2.0-Cisco-1.25
*Jul 29 23:36:11.421: SSH0: protocol version id is - SSH-2.0-Cisco-1.25

*Jul 29 23:36:11.421: SSH2 0:
ssh2_server_key_exchange: kexinit sent: kex algo = curve25519-sha256,curve25519-sha256@libssh.org,ecdh-sha2-nistp256,ecdh-sha2-nistp384,ecdh-sha2
*Jul 29 23:36:11.421: SSH2 0: Host-Key not configured. Skipping algo = ecdsa-sha2-nistp256
*Jul 29 23:36:11.421: SSH2 0: Host-Key not configured. Skipping algo = ecdsa-sha2-nistp384
*Jul 29 23:36:11.421: SSH2 0: Host-Key not configured. Skipping algo = ecdsa-sha2-nistp521
*Jul 29 23:36:11.421: SSH2 0: kexinit sent: hostkey algo = rsa-sha2-512,rsa-sha2-256
*Jul 29 23:36:11.421: SSH2 0: kexinit sent: encryption algo = chacha20-poly1305@openssh.com,aes128-gcm@openssh.com,aes256-gcm@openssh.com,a
*Jul 29 23:36:11.421: SSH2 0: kexinit sent: mac algo = hmac-sha2-256-etm@openssh.com,hmac-sha2-512-etm@openssh.com,hmac-sha2-256,hmac-sha2-
*Jul 29 23:36:11.421: SSH2 0: send:packet of length 736 (length also includes padlen of 10)
*Jul 29 23:36:11.421: SSH2 0: SSH2_MSG_KEXINIT sent
*Jul 29 23:36:11.421: SSH2 0: ssh_receive: 64 bytes received
*Jul 29 23:36:11.421: SSH2 0: input: total packet length of 416 bytes
*Jul 29 23:36:11.421: SSH2 0: partial packet length(block_size)8 bytes,
    needed 408 bytes, maclen 0
*Jul 29 23:36:11.422: SSH2 0: ssh_receive: 64 bytes received
*Jul 29 23:36:11.422: SSH2 0: partial packet length(block_size)8 bytes,
    needed 408 bytes, maclen 0
*Jul 29 23:36:11.422: SSH2 0: input: padlength 6 bytes
*Jul 29 23:36:11.422: SSH2 0: SSH2_MSG_KEXINIT received
*Jul 29 23:36:11.423: SSH2 0: strict kex is enabled

***Jul 29 23:36:11.423: %SSH-3-NO_MATCH: No matching mac found: client hmac-sha1 server hmac-sha2-256-etm@**

Erreur « %SSH-3-PRIVATEKEY : Impossible de récupérer la clé privée RSA"

Une modification du nom de domaine ou du nom d’hôte peut déclencher ce message d’erreur. Essayez ces solutions de contournement :

- Mettez à zéro les clés RSA et générez de nouveau les clés.

<#root>

crypto key zeroize rsa label <key-name> crypto key generate rsa label <key-name> modulus <key-size>

- Si la solution de contournement précédente ne fonctionne pas, essayez ces étapes :
    1. Mettez à zéro toutes les clés RSA.
    2. Rechargez l’appareil.
    3. Créez de nouvelles clés étiquetées pour SSH.

Recommandations

- Utilisez la version de protocole SSH la plus élevée prise en charge par la plate-forme.

SSHv2 est préférable à SSHv1.
- Utilisez le module RSA le plus puissant pris en charge par la plate-forme. Un module de 2048 bits ou supérieur est recommandé lorsqu'il est pris en charge.
- Utilisez un module 4096 bits lorsque la plate-forme le prend en charge et que l'impact sur les performances est acceptable.
- Appliquez la configuration SSH uniquement à toutes les lignes VTY disponibles, et pas seulement à la plage par défaut.
- Utilisez des informations d'identification secrètes locales ou une authentification AAA au lieu de mots de passe en texte clair.

- Si vos commandes de configuration SSH sont rejetées en tant que commandes illégales, vous n'avez pas correctement généré une paire de clés RSA pour votre routeur. Assurez-vous d'avoir indiqué un nom d'hôte et un domaine. Utilisez ensuite la commande crypto key generate rsa pour générer des paires de clés RSA et pour activer le serveur SSH.

- Lorsque vous configurez des paires de clés RSA, les messages d'erreur suivants peuvent s'afficher :

    - "Aucun nom d'hôte spécifié". Utilisez la commande de configuration globale hostname pour configurer un nom d'hôte pour le routeur.
    - "Aucun domaine spécifié". Utilisez la commande de configuration globale ip domain name pour configurer un domaine hôte pour le routeur.

- Le nombre de connexions SSH autorisées est limité au nombre maximal de connexions vty configurées pour le routeur. Chaque connexion SSH utilise une vty ressource.

- SSH utilise la sécurité locale ou le protocole de sécurité configuré par AAA sur votre routeur pour l'authentification de l'utilisateur. Lorsque vous configurez AAA, vous devez vous assurer que la console n'est pas exécutée sous AAA. Appliquez un mot-clé dans le mode de configuration globale pour désactiver AAA sur la console.

- La commande show ssh affiche les sessions SSH actives. Le message %No SSHv2 server connections running signifie qu'aucune session SSHv2 n'est actuellement active. Utilisez show ip ssh pour vérifier l'état et la configuration du serveur SSH. Si vous avez déjà configuré SSH, il est recommandé de reconfigurer le serveur SSH dans le périphérique. Suivez ces étapes pour reconfigurer le serveur SSH sur l'appareil.

1. Supprimez les paires de clés RSA à l'aide de la commande crypto key zeroize rsa. Après la suppression des paires de clés RSA, le serveur SSH est automatiquement désactivé.

Remarque : Générez des clés RSA avec un module de 2 048 bits ou plus, en fonction de la prise en charge de la plate-forme et de la politique de sécurité.

Mise en garde : Cette commande ne peut pas être annulée après l'enregistrement de votre

---

⚠ configuration. En outre, une fois les clés RSA supprimées, vous ne pouvez pas utiliser les certificats ou le CA ni participer à des échanges de certificats avec d'autres homologues IPSec, à moins de régénérer les clés RSA pour reconfigurer l'interopérabilité du CA, obtenir le certificat du CA et demander de nouveau votre certificat.

2. Reconfigurez le nom d'hôte et le nom de domaine du périphérique avec les commandes hostname <name> et ip domain name <domain>, respectivement.

3. Générez des paires de clés RSA pour votre routeur avec la commande crypto key generate rsa. Cela active automatiquement le protocole SSH.

✎ Remarque : Vous pouvez recevoir le message « `SSH2 0: Message d'`erreur « `Type de message reçu inattendu` » dû à un paquet reçu qui n'est pas compréhensible par le routeur. Augmentez la longueur de clé pendant que vous générez des clés RSA pour SSH afin de résoudre ce problème.

4. Activez SSH version 2, avec la commande ip ssh version 2. Si vous ne spécifiez pas d'autres paramètres SSH, les valeurs par défaut sont utilisées.

Informations connexes

- Secure Shell (SSH)

À propos de cette traduction
Cisco a traduit ce document en traduction automatisée vérifiée par une personne dans le cadre d'un service mondial permettant à nos utilisateurs d'obtenir le contenu d'assistance dans leur propre langue.

Il convient cependant de noter que même la meilleure traduction automatisée ne sera pas aussi précise que celle fournie par un traducteur professionnel.