在路由器上配置SSH

目录

简介

先决条件

- 要求
- 使用的组件

网络图

- 使用密码身份验证在思科路由器上配置SSH
- 使用RSA公钥身份验证在思科路由器上配置SSH

其他配置集

- 配置SSH第2版
- 阻止非 SSH 连接
- 限制对子网的SSH访问
- banner 命令输出的变化

debug 和 show 命令

调试RSA公共身份验证示例

- SSH客户端调试(Linux/macOS)
- SSH服务器调试（思科路由器）

配置不正确

- 错误密码
- SSH客户端发送不支持的密码
- SSH客户端发送不支持的HMAC
- 错误“%SSH-3-PRIVATEKEY:无法检索RSA私钥”

建议

相关信息

简介

本文档介绍如何配置和调试Cisco路由器上的Secure Shell(SSH)。

先决条件

要求

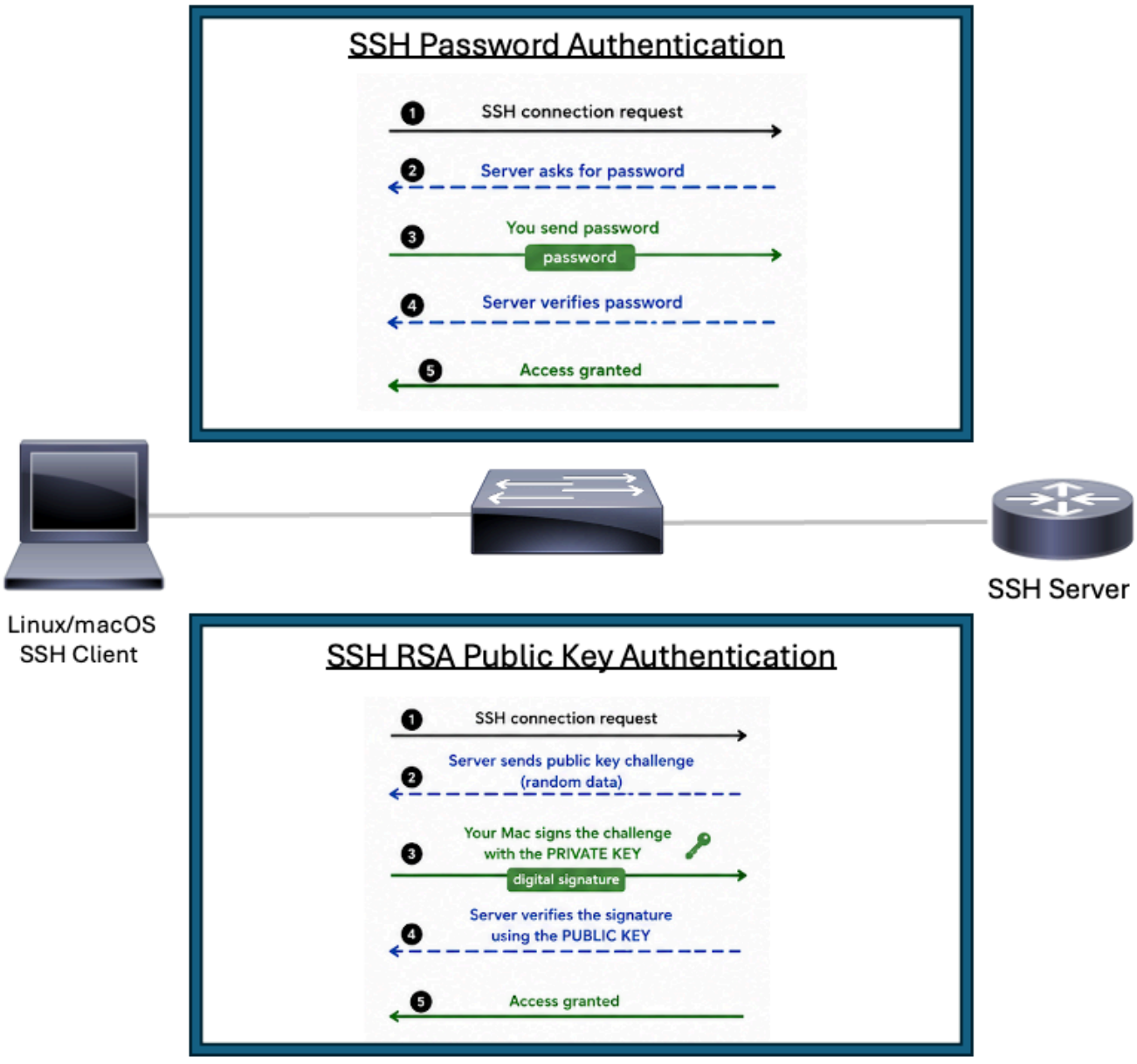
思科映像或软件包必须包含对SSH的加密支持。要求因平台、软件版本和许可模式而异。

使用的组件

本文档中的信息基于Cisco IOS和Cisco IOS XE软件。

本文档中的信息都是基于特定实验室环境中的设备编写的。本文档中使用的所有设备最初均采用原始（默认）配置。如果您的网络处于活动状态，请确保您了解所有命令的潜在影响。

网络图



SSH拓扑

使用密码身份验证在思科路由器上配置SSH

在Cisco路由器上启用SSH基本支持需要几个步骤：

1.配置路由器主机名。

2.为SSH身份验证创建本地用户帐户。

3.配置DNS域名。

4.启用平台支持的最高SSH版本。如果支持，请使用SSH版本2。

5.生成SSH的RSA密钥对。使用平台支持的最大RSA系数并与您的安全策略保持一致。

6.进入VTY线路配置模式。

7.将VTY线路配置为使用本地用户数据库或aaa进行身份验证。

8.仅限制VTY访问SSH传输。

充当SSH服务器的思科设备需要主机名、域名、RSA密钥、身份验证和VTY配置。仅发起SSH会话的设备需要SSH客户端支持和到SSH服务器的有效路由。

<#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
```



注意：如果未使用login local，则aaa new-model命令会在没有其他AAA语句的情况下使用路由器上的本地用户名和密码。

当从另一台路由器进行测试时，请将此命令用作SSH客户端命令的示例。支持的密码和hmac算法可能因软件版本和SSH策略而异：

<#root>

```
R3#

ssh -v 2 -l cisco 10.20.20.20


Password:
R2>
```

使用RSA公钥身份验证在思科路由器上配置SSH

RSA公钥身份验证允许SSH客户端向Cisco路由器或交换机进行身份验证，而无需发送密码或依赖于密码。SSH服务器存储公钥。SSH客户端保留匹配的私钥。在登录期间，客户端通过数字签名身份验证数据来证明它拥有私钥。服务器使用设备上配置的公钥验证签名。私钥保留在客户端上，不

通过网络传输。

好处

- 无需对配置的用户进行基于密码的SSH身份验证。
- 减少密码猜测和暴力攻击的风险。
- 在SSH客户端上保留私钥，而不是在网络设备或自动化系统上存储可重复使用的密码。
- 适用于需要安全访问思科设备的自动化工具、脚本和CI/CD系统。
- 提高操作一致性，因为服务器验证的是受信任的公钥而不是键入的密码。
- 支持管理员工作站、跳转主机和自动化服务器的更强身份验证模型。

完成以下步骤以配置SSH服务器以执行基于RSA公钥的身份验证：

1.配置路由器主机名。

2.配置DNS域名。

3.在line vty下启用AAA或配置login local。

4.生成SSH服务器使用的RSA密钥对。

5.使用平台支持的最高SSH版本。如果支持，请使用SSHv2。

6.使用平台和安全策略支持的最强RSA模数。如果支持并且性能影响可以接受，则首选4096位模数。

7.进入SSH公钥链配置模式。

8.配置允许使用RSA公钥身份验证进行身份验证的SSH用户名。

9.从SSH客户端为该用户名添加RSA公钥。



注意：您可以使用ssh-keygen命令从Linux/macOS设备生成SSH公钥，例如，ssh-keygen -m PEM -t rsa -b 1024 -f ~/.ssh/cisco1024，然后您可以使用cat ~/.ssh/cisco1024.pub命令读取该密钥。

<#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#
```

其他配置集

配置SSH第2版

配置SSH时，确保启用SSHv2，因为它比SSHv1提供更强的加密和明显更好的安全性。

<#root>

```
Router(config)#
```

ip ssh version 2

阻止非 SSH 连接

如果要阻止非 SSH 连接，请在语句行的下面添加 transport input ssh 命令，将路由器限制为只能使用 SSH 连接。直接（非 SSH）Telnet 将被拒绝。

<#root>

```
Router(config)#
```

line vty 0 15

```
Router(config-line)#
```

transport input ssh

限制对子网的SSH访问

您需要限制到特定子网的SSH连接，在该子网中会丢弃来自该子网外部IP的所有其他SSH尝试。

可以使用以下步骤执行相同操作：

1. 定义允许来自该特定子网的流量的访问列表。
2. 使用access-class限制对VTY线路接口的访问。

这是示例配置。在本示例中，仅允许来自10.10.10.0/24的SSH访问；所有其它源IP地址均被ACL末尾的隐式deny拒绝。

```
<#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
```

注意：验证设备上的可用VTY线路范围，然后将ACL应用到用于远程访问的所有VTY线路。

banner 命令输出的变化

使用 Telnet 和不同版本的 SSH 连接时，banner 命令的输出会有所不同。下表说明了不同的 banner 命令选项如何处理各种类型的连接。

Banner命令选项	Telnet	SSH v2
banner login	在登录到设备之前显示。	在登录到设备之前显示。
banner motd	在登录到设备之前显示。	登录设备后显示。
banner exec	登录设备后显示。	登录设备后显示。

注意：不再建议使用SSH第1版。

debug 和 show 命令

在发出debug命令之前，请参阅了解有关Debug命令的重要信息。

debug ip ssh — 启用SSH调试消息。

show ssh — 显示SSH服务器连接的状态。

<#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 — 显示SSH的版本和配置数据。

<#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

调试RSA公共身份验证示例

SSH客户端调试(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:

compression: none
debug1: kex: client->server cipher: aes128-gcm@openssh.com MAC:

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=

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
```

SSH服务器调试（思科路由器）

<#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
*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
```

配置不正确

以下部分列出了几种不正确的配置导致的调试输出示例。

错误密码

路由器调试

<#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**

SSH客户端发送不支持的密码

路由器调试

<#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**

SSH客户端发送不支持的HMAC

路由器调试

<#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@**

错误“%SSH-3-PRIVATEKEY:无法检索RSA私钥”

更改域名或主机名可能会触发此错误消息。尝试以下解决方法：

- 将RSA密钥归零，然后重新生成密钥。

<#root>

crypto key zeroize rsa label

crypto key generate rsa label

modulus

- 如果之前的解决方法不起作用，请尝试以下步骤：

    1. 将所有RSA密钥归零。
    2. 重新启动设备。
    3. 为SSH创建新的标记密钥。

建议

- 使用平台支持的最高SSH协议版本。SSHv2优先于SSHv1。
- 使用平台支持的最强RSA模数。如果支持，建议使用2048位或更高的模数。
- 如果平台支持4096位模数，并且性能影响可以接受，则使用4096位模数。
- 将仅SSH配置应用于所有可用的VTY线路，而不仅仅是默认范围。
- 使用本地密钥或AAA身份验证，而不是明文密码。

- 如果 SSH 配置命令被作为非法命令拒绝，则说明您没有成功地为路由器生成 RSA 密钥对。确保已指定主机名和域。然后使用crypto key generate rsa命令生成RSA密钥对，并启用SSH服务器。

- 配置RSA密钥对时，您会收到以下错误消息：

    - "未指定主机名"。 使用hostname全局配置命令配置路由器的主机名。
    - "未指定域"。 使用ip domain name全局配置命令配置路由器的主机域。

- 允许的SSH连接数限制为路由器配置的vty 最大连接数。每个SSH连接都使用一个vty资源。

- SSH使用本地安全性或通过路由器上的AAA配置的安全协议进行用户身份验证。配置AAA时，必须确保控制台未在AAA下运行。在全局配置模式下应用关键字以禁用控制台上的AAA。

- 命令show ssh显示活动的SSH会话。消息%No SSHv2 server connections running表示当前没有SSHv2会话处于活动状态。使用show ip ssh验证SSH服务器状态和配置。如果已经配置了

SSH，则建议您在设备中重新配置SSH服务器。要重新配置设备上的SSH服务器，请完成以下步骤。

1.使用crypto key zeroize rsa命令删除RSA密钥对。删除RSA密钥对后，将自动禁用SSH服务器。

注意：根据平台支持和安全策略，生成具有2048位模数或更强的RSA密钥。

警告：保存配置后，此命令无法撤消。此外，删除RSA密钥后，您无法使用证书或CA或参与与其他IP安全(IPSec)对等体的证书交换，除非您重新生成RSA密钥以重新配置CA互操作性、获取CA证书，并再次请求您自己的证书。

2.分别使用hostname <name>和ip domain name <domain>命令重新配置设备的主机名和域名。

3.使用crypto key generate rsa命令为路由器生成RSA密钥对。这将自动启用SSH。

注意：您可以收到“SSH2 0:路由器无法理解收到的数据包导致的意外消息type received”错误消息。要解决此问题，请在生成用于 SSH 的 RSA 密钥时增加密钥长度。

4.使用ip ssh version 2命令启用SSH版本2。如果不指定其他SSH参数，则使用默认值。

相关信息

- Secure Shell (SSH)

关于此翻译
思科采用人工翻译与机器翻译相结合的方式将此文档翻译成不同语言，希望全球的用户都能通过各自的语言得到支持性的内容。

请注意：即使是最好的机器翻译，其准确度也不及专业翻译人员的水平。

Cisco Systems, Inc. 对于翻译的准确性不承担任何责任，并建议您总是参考英文原始文档（已提供链接）。