Debug Secure Shell (SSH) en NCS1K

Contenido

Introducción

Este documento describe las prácticas básicas de solución de problemas para Secure Shell (SSH) en la plataforma NCS1K.

Prerequisites

En este documento, se asume que es competente con los sistemas operativos basados en XR en dispositivos como Network Convergence System (NCS) 1002.

Requirements

Cisco recomienda que tenga conocimiento de estos temas para los requisitos de conexión SSH:

- El paquete k9sec relevante para la imagen XR
- Configuración SSH presente en el dispositivo Cisco
- Generación satisfactoria de claves, intercambio de claves y negociación de cifrado entre el host y el servidor

Componentes Utilizados

La información que contiene este documento se basa en las siguientes versiones de software y hardware.

- NCS1002 con XR 7.3.1
- NCS1004 con XR 7.9.1

La información que contiene este documento se creó a partir de los dispositivos en un ambiente de laboratorio específico. Todos los dispositivos que se utilizan en este documento se pusieron en funcionamiento con una configuración verificada (predeterminada). Si tiene una red en vivo, asegúrese de entender el posible impacto de cualquier comando.

Verificar paquetes instalados

Los comandos `show install active` y `show install committed` identifique la presencia del paquete k9sec. Sin este paquete instalado, no puede generar claves criptográficas para iniciar una sesión SSH.

<#root>

```
RP/0/RP0/CPU0:NCS1002_1#
show install active


Wed Jul 19 09:31:18.977 UTC
Label : 7.3.1

Node 0/RP0/CPU0 [RP]
Boot Partition: xr_lv58
Active Packages: 4
ncs1k-xr-7.3.1 version=7.3.1 [Boot image]
ncs1k-mpls-te-rsvp-3.1.0.0-r731
ncs1k-mpls-2.1.0.0-r731
ncs1k-k9sec-3.1.0.0-r731

RP/0/RP0/CPU0:NCS1002_1#
show install committed


Wed Jul 19 09:31:37.359 UTC
Label : 7.3.1

Node 0/RP0/CPU0 [RP]
Boot Partition: xr_lv58
Committed Packages: 4
ncs1k-xr-7.3.1 version=7.3.1 [Boot image]
ncs1k-mpls-te-rsvp-3.1.0.0-r731
ncs1k-mpls-2.1.0.0-r731
ncs1k-k9sec-3.1.0.0-r731
```

Configuración

Como mínimo, NCS1K requiere la configuración `ssh server v2` para permitir las conexiones SSH. ingrese `show run ssh` para asegurarse de que esta configuración está presente:

<#root>

```
RP/0/RP0/CPU0:NCS1004_1#
show run ssh

Wed Jul 19 13:06:57.207 CDT
ssh server rate-limit 600
ssh server v2
ssh server netconf vrf default
```

Identificar claves generadas

Para establecer una sesión SSH, el NCS1K debe tener presente una clave criptográfica pública. Identificar la presencia de claves generadas con show crypto key mypubkey { dsa | ecdsa | ed25519 | rsa }. El tipo de clave predeterminado es rsa. La clave aparece como una cadena hexadecimal, omitida aquí por motivos de seguridad.

<#root>

```
RP/0/RP0/CPU0:NCS1002_1#
show crypto key mypubkey rsa

Wed Jul 19 10:30:09.333 UTC
Key label: the_default
Type : RSA General purpose
Size : 2048
Created : 11:59:56 UTC Tue Aug 23 2022
Data : <key>
```

Para generar una clave de un tipo determinado, ingrese el comando crypto key generate { dsa | ecdsa | ed25519 | rsa } y elija un módulo clave. El tamaño del módulo varía según el algoritmo.

Tipo de clave	Tipos de módulos/curvas permitidos	Longitud predeterminada del módulo (bits)
dsa	512, 768, 1024	1024
ecdsa	nistp256, nistp384, nistp521	ninguno
ed25519	256	256
rsa	512 a 4096	2048

Compruebe que la clave se ha generado correctamente con show crypto key mypubkey.

Para quitar una clave existente, ingrese el comando crypto key zeroize { authentication | dsa | ecdsa | ed25519 | rsa } [label]. Asegúrese de tener acceso al dispositivo a través de otros medios, ya que la desconexión de un dispositivo sin claves criptográficas bloquea el acceso con SSH.

Identificación de Capacidades de Servidor SSH

El servidor y el host deben acordar un intercambio de claves, una clave de host y un cifrado antes de establecer una sesión SSH. Para identificar las capacidades de la plataforma NCS1K, ingrese el comando show ssh server.

<#root>

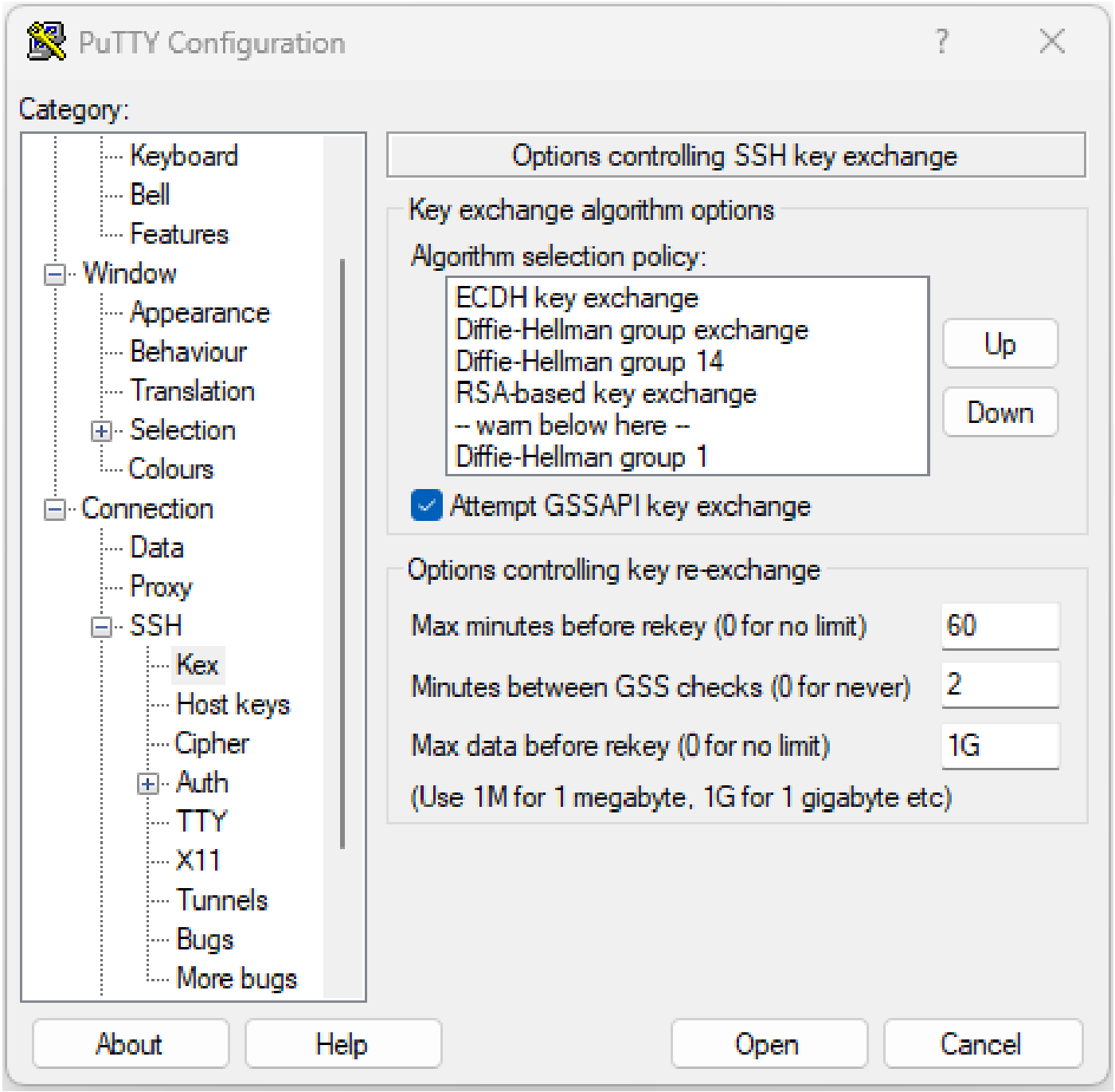
```
RP/0/RP0/CPU0:NCS1004_1#
show ssh server


Wed Jul 19 13:28:04.820 CDT
---------------------
SSH Server Parameters
---------------------

Current supported versions := v2
SSH port := 22
SSH vrfs := vrfname:=default(v4-acl:=, v6-acl:=)
Netconf Port := 830
Netconf Vrfs := vrfname:=default(v4-acl:=, v6-acl:=)

Algorithms
---------------
Hostkey Algorithms := x509v3-ssh-rsa,ecdsa-sha2-nistp521,ecdsa-sha2-nistp384,ecdsa-sha2-nistp256,rsa-sh
Key-Exchange Algorithms := ecdh-sha2-nistp521,ecdh-sha2-nistp384,ecdh-sha2-nistp256,diffie-hellman-grou
Encryption Algorithms := aes128-ctr,aes192-ctr,aes256-ctr,aes128-gcm@openssh.com,aes256-gcm@openssh.com
Mac Algorithms := hmac-sha2-512,hmac-sha2-256,hmac-sha1

Authentication Method Supported
-----------------------------------
PublicKey := Yes
Password := Yes
Keyboard-Interactive := Yes
Certificate Based := Yes

Others
------------
DSCP := 16
Ratelimit := 600
Sessionlimit := 64
Rekeytime := 60
Server rekeyvolume := 1024
TCP window scale factor := 1
Backup Server := Disabled
Host Trustpoint :=
User Trustpoint :=
Port Forwarding := Disabled
Max Authentication Limit := 20
Certificate username := Common name(CN)
```

Identificación de Funciones SSH de Host

El host que intenta conectarse debe coincidir al menos con una clave de host, intercambio de claves y algoritmo de cifrado del servidor para establecer una sesión SSH.

PuTTY

PuTTY enumera los algoritmos de intercambio de claves, clave de host y cifrado admitidos en Connections > SSH. El host negocia automáticamente los algoritmos en función de sus capacidades, y prefiere el algoritmo de intercambio de claves por orden de preferencia del usuario. La opción Attempt GSSAPI key exchange no es necesario para conectarse a un dispositivo NCS1K.



Captura de pantalla de opciones de PuTTY SSH

Linux

Los servidores Linux normalmente mantienen los algoritmos soportados en el /etc/ssh/ssh_config archivo. Este ejemplo se origina en el servidor Ubuntu 18.04.3.

```
Host *
# ForwardAgent no
# ForwardX11 no
# ForwardX11Trusted yes
# PasswordAuthentication yes
# HostbasedAuthentication no
# GSSAPIAuthentication no
# GSSAPIDelegateCredentials no
# GSSAPIKeyExchange no
# GSSAPITrustDNS no
# BatchMode no
# CheckHostIP yes
# AddressFamily any
# ConnectTimeout 0
# StrictHostKeyChecking ask
# IdentityFile ~/.ssh/id_rsa
# IdentityFile ~/.ssh/id_dsa
# IdentityFile ~/.ssh/id_ecdsa
# IdentityFile ~/.ssh/id_ed25519
# Port 22
# Protocol 2
# Ciphers aes128-ctr,aes192-ctr,aes256-ctr,aes128-cbc,3des-cbc
# MACs hmac-md5,hmac-sha1,umac-64@openssh.com
# EscapeChar ~
# Tunnel no
# TunnelDevice any:any
# PermitLocalCommand no
# VisualHostKey no
# ProxyCommand ssh -q -W %h:%p gateway.example.com
# RekeyLimit 1G 1h
SendEnv LANG LC_*
HashKnownHosts yes
GSSAPIAuthentication yes
```

Troubleshooting de Conexiones SSH

Estos comandos pueden ayudar a aislar fallas con conexiones SSH.

Vea las sesiones SSH actuales entrantes y salientes con show ssh session details.

```
<#root>

RP/0/RP0/CPU0:NCS1002_1#

show ssh session details
```

```
Wed Jul 19 13:08:46.147 UTC
SSH version : Cisco-2.0

id key-exchange pubkey incipher outcipher inmac outmac
-----------------------------------------------------------------------------------------------
Incoming Sessions
128733 ecdh-sha2-nistp256 ssh-rsa aes256-ctr aes256-ctr hmac-sha2-256 hmac-sha2-256
128986 diffie-hellman-group14 ssh-rsa aes128-ctr aes128-ctr hmac-sha1 hmac-sha1
128988 diffie-hellman-group14 ssh-rsa aes128-ctr aes128-ctr hmac-sha1 hmac-sha1

Outgoing sessions
```

Las sesiones SSH históricas incluyen intentos de conexión fallidos con el comando show ssh history detail.

```
<#root>

RP/0/RP0/CPU0:NCS1002_1#

show ssh history details


Wed Jul 19 13:13:26.821 UTC
SSH version : Cisco-2.0

id key-exchange pubkey incipher outcipher inmac outmac start_time end_time
-----------------------------------------------------------------------------------------------
Incoming Session
128869diffie-hellman-group14-sha1ssh-rsa aes128-ctr aes128-ctr hmac-sha1 hmac-sha1 19-07-23 11:28:55 19
```

Los seguimientos SSH proporcionan un nivel fino de detalle en el proceso de conexión con show ssh trace all.

```
<#root>

RP/0/RP0/CPU0:NCS1002_1#

show ssh trace all


Wed Jul 19 13:15:53.701 UTC
3986 wrapping entries (57920 possible, 40896 allocated, 0 filtered, 392083 total)
Apr 29 19:13:19.438 ssh/backup-server/event 0/RP0/CPU0 t6478 [SId:=0] Respawn-count:=1, Starting SSH Se
Apr 29 19:13:19.438 ssh/backup-server/shmem 0/RP0/CPU0 t6478 [SId:=0] Shared memory does not exist duri
```

Configuración de Valores SSH Re-Key

La configuración de re-clave SSH determina el tiempo y el número de bytes antes de que ocurra un nuevo intercambio de claves. Consulte los valores actuales mediante show ssh rekey.

<#root>

```
RP/0/RP0/CPU0:NCS1004_1#
```

show ssh rekey

```
Wed Jul 19 15:23:06.379 CDT
SSH version : Cisco-2.0

id RekeyCount TimeToRekey(min) VolumeToRekey(MB)
------------------------------------------------------------
Incoming Session
1015      6      6.4              1024.0
1016      0      58.8             1024.0

Outgoing sessions
```

Para configurar el volumen de re-key, utilice el comando ssh server rekey-volume [size]. El tamaño predeterminado para volver a introducir la clave es de 1024 MB.

<#root>

```
RP/0/RP0/CPU0:NCS1004_1(config)#
```

ssh server rekey-volume 4095

```
RP/0/RP0/CPU0:NCS1004_1(config)#
```

commit

Del mismo modo, establezca el valor del temporizador re-key con ssh server rekey-time [time]. El valor predeterminado es 60 minutos.

```
RP/0/RP0/CPU0:NCS1004_1(config)# ssh server rekey-time 120
RP/0/RP0/CPU0:NCS1004_1(config)# commit
```

Depuración SSH

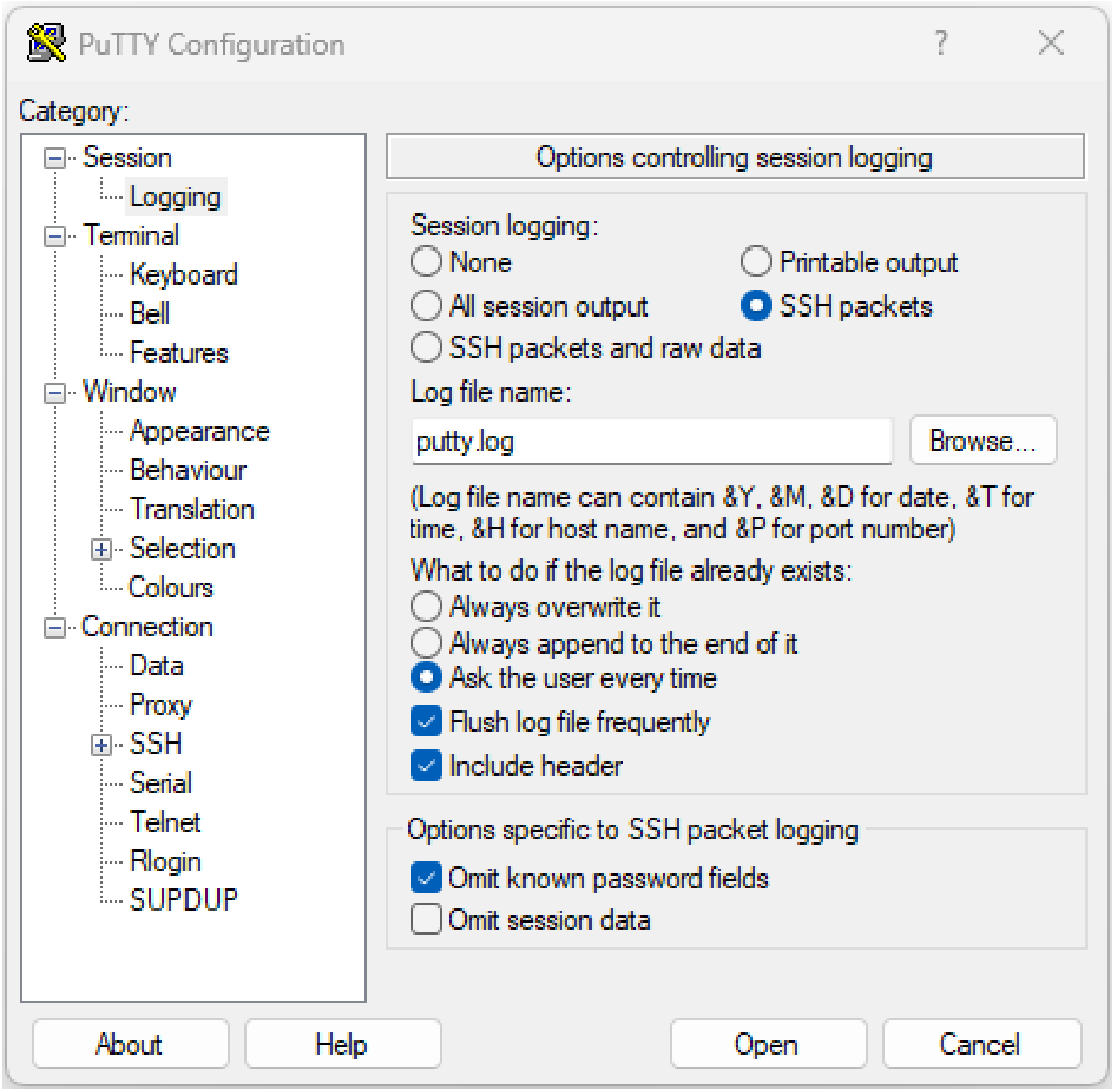
debug ssh server muestra salidas en tiempo real para sesiones SSH activas e intentos de conexión. Para resolver problemas de una conexión defectuosa, habilite el debug, intente la conexión y luego detenga el debug con undebug all. Registre la sesión mediante PuTTY u otra aplicación de terminal para su análisis.

<#root>

```
RP/0/RP0/CPU0:NCS1002_1#
```

```
debug ssh server
```

PuTTY incluye una función para registrar paquetes SSH en Session > Logging.



Captura de pantalla del registro SSH de PuTTY

En Linux, ssh -vv (muy detallado) proporciona información detallada sobre el proceso de conexión SSH.

<#root>

```
ubuntu-18@admin:/$
```

```
ssh -vv admin@192.168.190.2
```

Registros adicionales

Varios show techs capturan información útil sobre SSH.

- **show tech { ncs1k | ncs1001 | ncs1004 } detail**
- **show tech crypto session**
- **show tech ssh**
- **admin show tech { ncs1k | ncs1001 | ncs1004 }-admin**

Acerca de esta traducción
Cisco ha traducido este documento combinando la traducción automática y los recursos humanos a fin de ofrecer a nuestros usuarios en todo el mundo contenido en su propio idioma.

Tenga en cuenta que incluso la mejor traducción automática podría no ser tan precisa como la proporcionada por un traductor profesional.

Cisco Systems, Inc. no asume ninguna responsabilidad por la precisión de estas traducciones y recomienda remitirse siempre al documento original escrito en inglés (insertar vínculo URL).