Debug Secure Shell (SSH) sur NCS1K

Table des matières

Introduction

Ce document décrit les pratiques de dépannage de base pour Secure Shell (SSH) sur la plate-forme NCS1K.

Conditions préalables

Ce document suppose une maîtrise des systèmes d'exploitation basés sur XR sur des périphériques tels que le système NCS (Network Convergence System) 1002.

Exigences

Cisco recommande que vous ayez connaissance de ces sujets pour les exigences de connexion SSH :

- Package k9sec approprié pour l'image XR
- Configuration SSH présente sur le périphérique Cisco
- Génération de clés réussie, échange de clés et négociation de chiffrement entre l’hôte et le serveur

Composants utilisés

Les informations contenues dans ce document sont basées sur les versions de matériel et de logiciel suivantes :

- NCS1002 avec XR 7.3.1
- NCS1004 avec XR 7.9.1

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.

Vérifier les packages installés

Les commandes show install active et show install committed identifiez la présence du package k9sec. Sans ce package installé, vous ne pouvez pas générer de clés de chiffrement pour lancer une session 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
```

Configuration

À tout le moins, le NCS1K nécessite la configuration ssh server v2 afin d'autoriser les connexions SSH. Saisissez show run ssh afin de s'assurer que cette configuration est présente :

<#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
```

Identifier les clés générées

Pour établir une session SSH, le NCS1K doit disposer d'une clé cryptographique publique. Identifier la présence de clés générées avec `show crypto key mypubkey { dsa | ecdsa | ed25519 | rsa }`. Le type de clé par défaut est `rsa`. La clé apparaît sous la forme d'une chaîne hexadécimale, omise ici à des fins de sécurité.

<#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>
```

Afin de générer une clé d'un type particulier, entrez la commande `crypto key generate { dsa | ecdsa | ed25519 | rsa }` et choisissez un module clé. La taille du module varie selon l'algorithme.

Type de clé	Modules/types de courbes autorisés	Longueur du module par défaut (bits)
dsa	512, 768, 1024	1024
ecdsa	nistp256, nistp384, nistp521	none
ed25519	256	256

rsa	512 à 4096	2048

Vérifiez que la clé a été générée avec succès avec show crypto key mypubkey.

Afin de supprimer une clé existante, entrez la commande crypto key zeroize { authentication | dsa | ecdsa | ed25519 | rsa } [label]. Assurez-vous que vous avez accès au périphérique par d'autres moyens, car la déconnexion d'un périphérique sans clé de chiffrement bloque l'accès avec SSH.

Identifier les capacités du serveur SSH

Le serveur et l'hôte doivent s'entendre sur un échange de clés, une clé d'hôte et un chiffrement avant d'établir une session SSH. Afin d'identifier les capacités de la plate-forme NCS1K, entrez la commande 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-sha
Key-Exchange Algorithms := ecdh-sha2-nistp521,ecdh-sha2-nistp384,ecdh-sha2-nistp256,diffie-hellman-group
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 :=
```

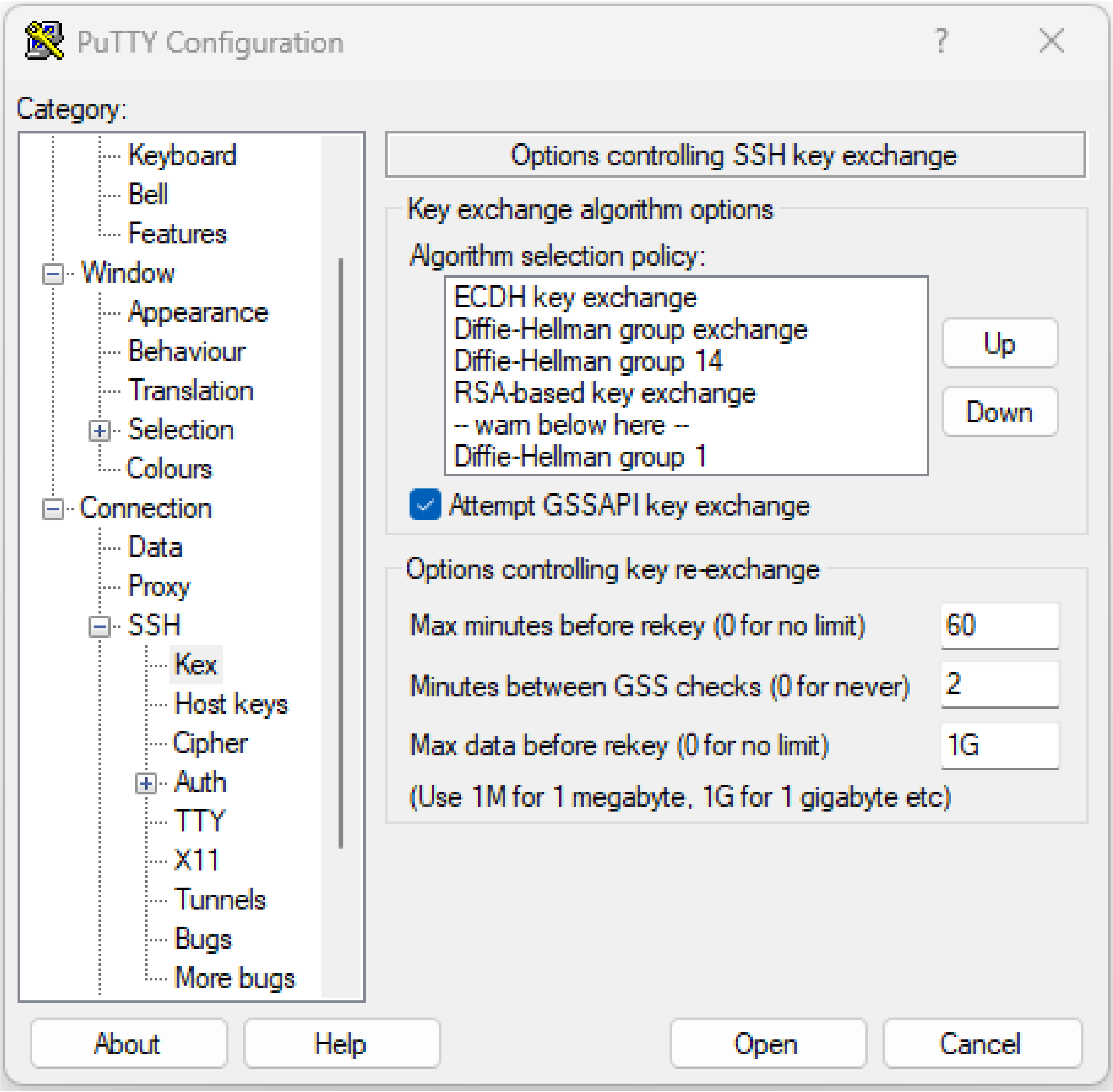
```
User Trustpoint :=
Port Forwarding := Disabled
Max Authentication Limit := 20
Certificate username := Common name(CN)
```

Identifier les capacités SSH hôte

L'hôte qui tente de se connecter doit correspondre à au moins une clé d'hôte, un échange de clés et un algorithme de chiffrement du serveur afin d'établir une session SSH.

PuTTY

PuTTY répertorie les algorithmes d'échange de clés, de clé hôte et de chiffrement pris en charge sous Connections > SSH. L'hôte négocie automatiquement les algorithmes en fonction de ses capacités, préférant l'algorithme d'échange de clés par ordre de préférence de l'utilisateur. L'option Attempt GSSAPI key exchange n'est pas nécessaire pour se connecter à un périphérique NCS1K.



Capture d'écran des options PuTTY SSH

Linux

Les serveurs Linux conservent généralement les algorithmes pris en charge dans /etc/ssh/ssh_config fichier. Cet exemple provient d'Ubuntu Server 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
```

Dépannage des connexions SSH

Ces commandes permettent d'isoler les pannes des connexions SSH.

Voir les sessions SSH entrantes et sortantes actuelles avec 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
```

Les sessions SSH historiques incluent les tentatives de connexion ayant échoué avec la commande 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
```

Les traces SSH fournissent un niveau de détail précis sur le processus de connexion avec 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
```

Configurer les valeurs de nouvelle clé SSH

La configuration SSH re-key détermine le temps et le nombre d'octets avant qu'un nouvel échange de clés ne se produise. Voir les valeurs actuelles en utilisant 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
```

Afin de définir le volume de nouvelle touche, utilisez la commande ssh server rekey-volume [size]. La taille de nouvelle clé par défaut est de 1 024 Mo.

<#root>

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

ssh server rekey-volume 4095

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

commit
```

De même, définissez la valeur du minuteur de nouvelle touche avec ssh server rekey-time [time]. La valeur par défaut est de 60 minutes.

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

Débogage SSH

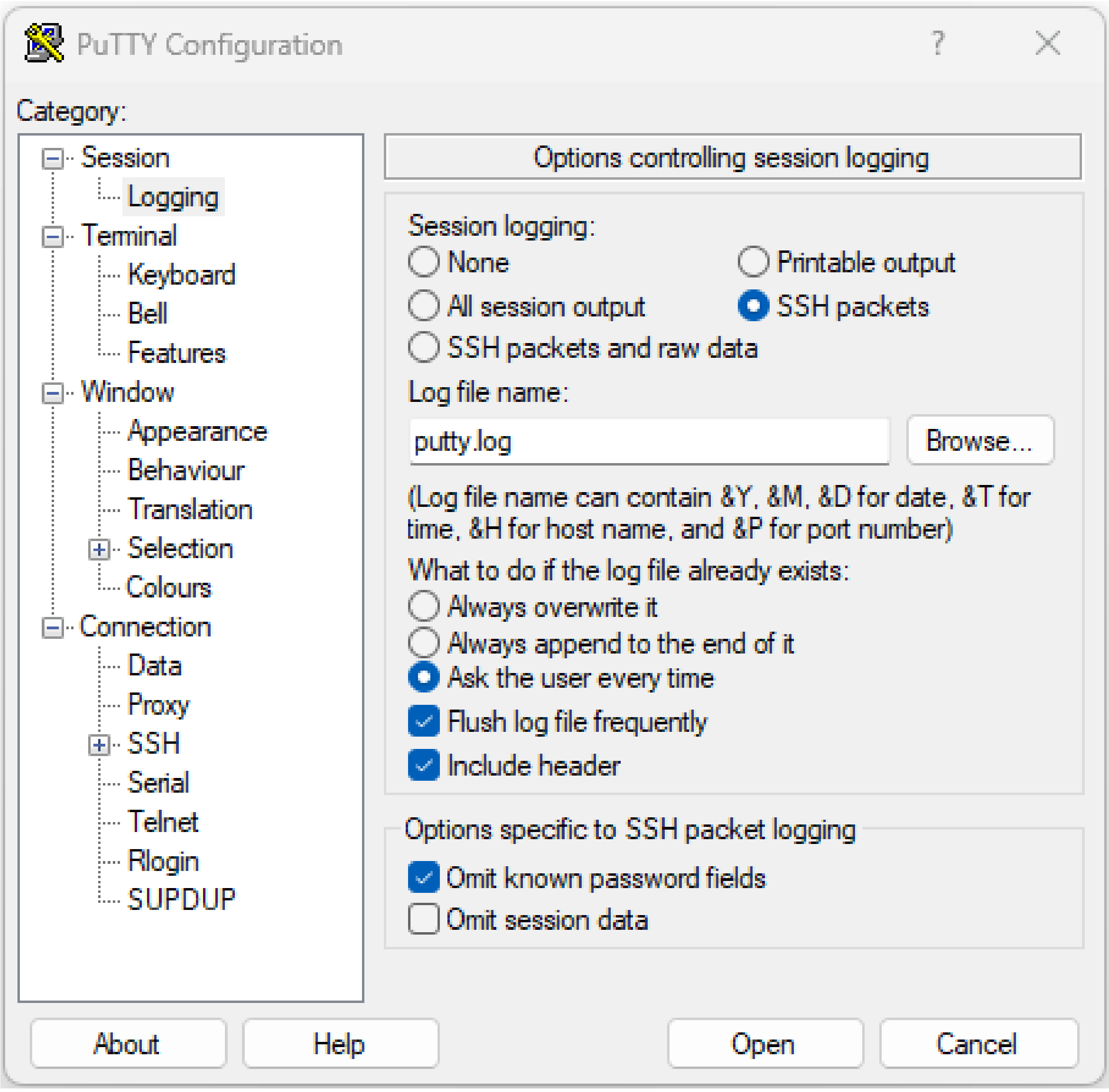
Les debug ssh server affiche les résultats en temps réel des sessions SSH actives et des tentatives de connexion. Afin de dépanner une connexion défaillante, activez le débogage, essayez la connexion, puis arrêtez le débogage avec undebug all. Consignez la session à l'aide de PuTTY ou d'une autre application de terminal pour analyse.

<#root>

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

debug ssh server
```

PuTTY inclut une fonctionnalité de journalisation des paquets SSH sous Session > Logging.



Capture d'écran de la journalisation PuTTY SSH

Sous Linux, `ssh -vv` (très détaillé) donne des informations détaillées sur le processus de connexion SSH.

<#root>

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

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

Journaux supplémentaires

Plusieurs show techs capturent des informations utiles sur SSH.

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

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.