Debug Secure Shell (SSH) su NCS1K

Sommario

Introduzione

In questo documento vengono descritte le procedure di base per la risoluzione dei problemi relativi a Secure Shell (SSH) sulla piattaforma NCS1K.

Prerequisiti

In questo documento si presume che i sistemi operativi basati su XR siano efficienti su dispositivi quali Network Convergence System (NCS) 1002.

Requisiti

Cisco raccomanda la conoscenza dei seguenti argomenti per i requisiti della connessione SSH:

- Il pacchetto k9sec rilevante per l'immagine XR
- Configurazione SSH presente sul dispositivo Cisco
- Generazione di chiavi, scambio di chiavi e negoziazione di cifratura tra host e server completati

Componenti usati

Le informazioni fornite in questo documento si basano sulle seguenti versioni software e hardware:

- NCS1002 con XR 7.3.1
- NCS1004 con XR 7.9.1

Le informazioni discusse in questo documento fanno riferimento a dispositivi usati in uno specifico ambiente di emulazione. Su tutti i dispositivi menzionati nel documento la configurazione è stata ripristinata ai valori predefiniti. Se la rete è operativa, valutare attentamente eventuali conseguenze derivanti dall'uso dei comandi.

Verifica pacchetti installati

I comandi `show install active` e `show install committed` identificare la presenza del pacchetto k9sec. Senza questo pacchetto installato, non è possibile generare chiavi crittografiche per avviare una sessione 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
```

Configurazione

Come minimo, NCS1K richiede la configurazione `ssh server v2` per consentire le connessioni SSH. Inserire `show run ssh` per garantire la presenza di questa configurazione:

```
<#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
```

Identifica chiavi generate

Per stabilire una sessione SSH, l'NCS1K deve avere una chiave crittografica pubblica presente. Identifica la presenza di chiavi generate con show crypto key mypubkey { dsa | ecdsa | ed25519 | rsa }. Il tipo di chiave predefinito è rsa. La chiave viene visualizzata come stringa esadecimale, omessa per motivi di sicurezza.

```
<#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>
```

Per generare una chiave di un determinato tipo, immettere il comando crypto key generate { dsa | ecdsa | ed25519 | rsa } e scegliere un modulo chiave. Le dimensioni del modulo variano a seconda dell'algoritmo.

Tipo di chiave	Tipi di modulo/curva consentiti	Lunghezza modulo predefinita (bit)
dsa	527, 768, 1024	1024
ecdsa	nistp256, nistp384, nistp521	nessuna
ed25519	256	256

rsa	da 512 a 4096	2048

Verificare la chiave generata correttamente con show crypto key mypubkey.

Per rimuovere una chiave esistente, immettere il comando crypto key zeroize { authentication | dsa | ecdsa | ed25519 | rsa } [label]. Verificare di avere accesso al dispositivo con altri mezzi, poiché la disconnessione da un dispositivo senza chiavi crittografiche blocca l'accesso con SSH.

Individuazione funzionalità server SSH

Prima di stabilire una sessione SSH, il server e l'host devono accordarsi su uno scambio di chiavi, una chiave host e una cifratura. Per identificare le funzionalità della piattaforma NCS1K, immettere il 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 :=
```

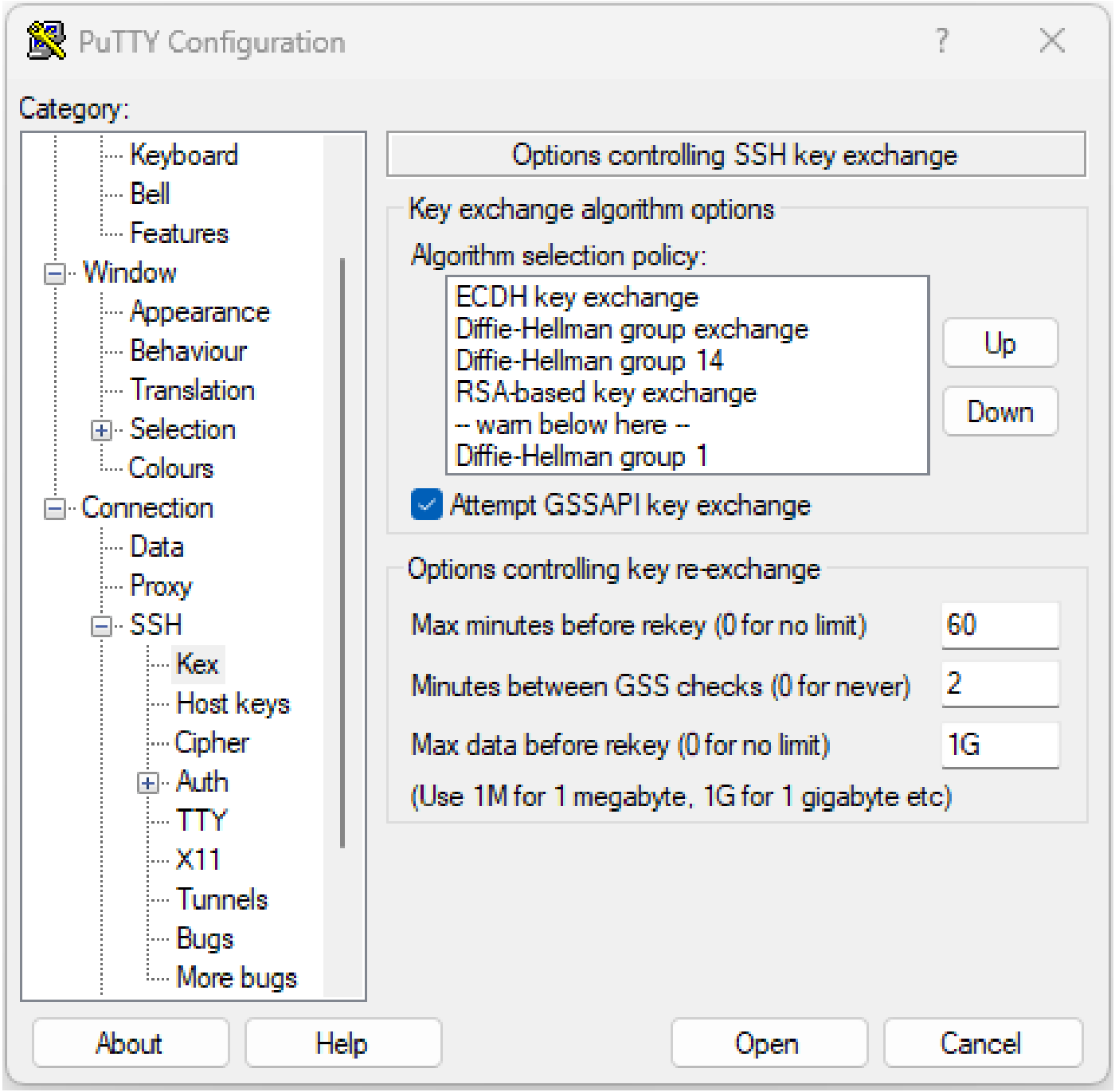
```
User Trustpoint :=
Port Forwarding := Disabled
Max Authentication Limit := 20
Certificate username := Common name(CN)
```

Identificazione delle funzionalità SSH dell'host

Per stabilire una sessione SSH, l'host che tenta di connettersi deve corrispondere ad almeno una chiave host, uno scambio di chiavi e un algoritmo di crittografia del server.

PuTTY

PuTTY elenca gli algoritmi supportati per lo scambio di chiavi, la chiave host e la cifratura in Connections > SSH. L'host negozia automaticamente gli algoritmi in base alle proprie capacità, preferendo l'algoritmo di scambio chiave in ordine di preferenza dell'utente. L'opzione Attempt GSSAPI key exchange non è necessaria per la connessione a un dispositivo NCS1K.



Schermata delle opzioni PuTTY SSH

Linux

I server Linux in genere mantengono gli algoritmi supportati nel /etc/ssh/ssh_config file. Questo esempio ha origine da 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
```

Risoluzione dei problemi di connessione SSH

Questi comandi possono aiutare a isolare gli errori nelle connessioni SSH.

Vedere le sessioni SSH in entrata e in uscita correnti 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
```

Le sessioni SSH cronologiche includono i tentativi di connessione non riusciti con il 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
```

Le tracce SSH forniscono un preciso livello di dettaglio nel processo di connessione 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
```

Configurazione dei valori di reimpostazione chiave SSH

La configurazione della rigenerazione della chiave SSH determina il tempo e il numero di byte prima che si verifichi un nuovo scambio di chiave. Visualizzare i valori correnti utilizzando 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
```

Per impostare il volume per la reimpostazione della chiave, utilizzare il comando ssh server rekey-volume [size]. La dimensione predefinita della chiave di ripristino è 1024 MB.

<#root>

```
RP/0/RP0/CPU0:NCS1004_1(config)#
ssh server rekey-volume 4095

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

Analogamente, impostare il valore di reimpostazione chiave del timer con ssh server rekey-time [time]. Il valore predefinito è 60 minuti.

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

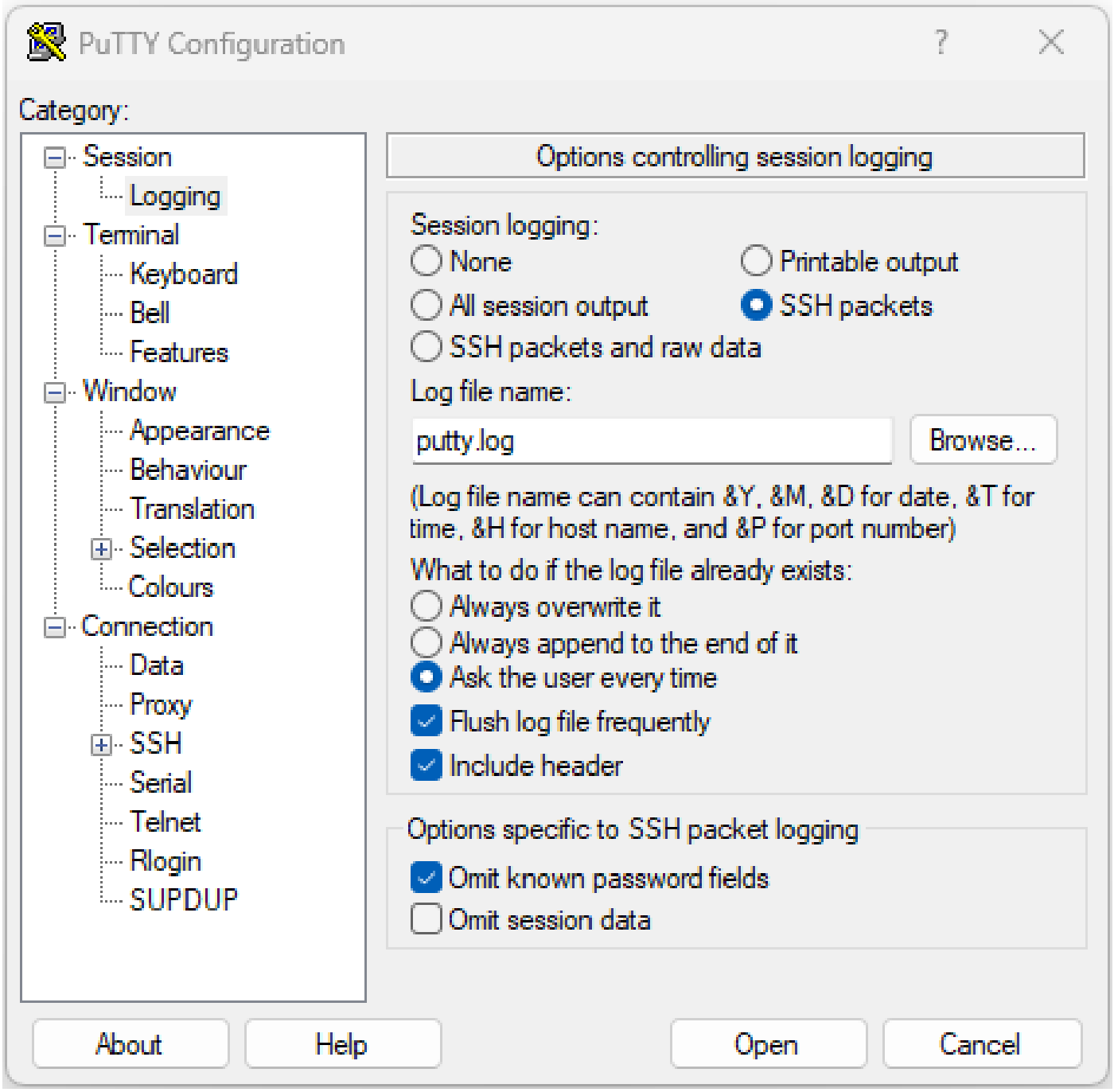
Debug SSH

OSPF (Open Shortest Path First) debug ssh server Questo comando visualizza gli output in tempo reale delle sessioni SSH attive e dei tentativi di connessione. Per risolvere i problemi relativi a una connessione non riuscita, abilitare il comando debug, tentare la connessione e quindi interrompere il debug con undebug all. Registrare la sessione utilizzando PuTTY o un'altra applicazione terminale per l'analisi.

<#root>

```
RP/0/RP0/CPU0:NCS1002_1#
debug ssh server
```

PuTTy include una funzione per registrare i pacchetti SSH in Session > Logging.



Schermata di registrazione di PuTTY SSH

In Linux, `ssh -vv` (molto dettagliato) offre informazioni dettagliate sul processo di connessione SSH.

```
<#root>

ubuntu-18@admin:/$

ssh -vv admin@192.168.190.2
```

Registri aggiuntivi

Diversi show tech catturano informazioni utili sul protocollo SSH.

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

Informazioni su questa traduzione

Cisco ha tradotto questo documento utilizzando una combinazione di tecnologie automatiche e umane per offrire ai nostri utenti in tutto il mondo contenuti di supporto nella propria lingua. Si noti che anche la migliore traduzione automatica non sarà mai accurata come quella fornita da un traduttore professionista. Cisco Systems, Inc. non si assume alcuna responsabilità per l’accuratezza di queste traduzioni e consiglia di consultare sempre il documento originale in inglese (disponibile al link fornito).