Informations importantes sur les commandes debug

Contenu

Introduction

Cette page fournit des instructions générales sur l'utilisation des débogages disponibles sur les plates-formes Cisco IOS®, ainsi que des exemples d'utilisation correcte de **debug ip packet** et le débogage conditionnel.

> **Note**: Ce document n'explique pas comment utiliser et interpréter les commandes spécifiques debug et les résultats. Reportez-vous à la documentation de référence des commandes de débogage Cisco appropriée pour plus d'informations sur les **debug** .

La sortie de **debug** Les commandes du mode d'exécution privilégié fournissent des informations de diagnostic qui incluent divers événements d'interconnexion de réseaux liés à l'état du protocole et à l'activité du réseau en général.

Conditions préalables

Conditions requises

Cisco vous recommande de prendre connaissance des rubriques suivantes :

- Connexion au routeur à l'aide de la console ainsi que des ports aux et vty
- Problèmes généraux de configuration de Cisco IOS
- Interprétation des sorties de débogage de Cisco IOS

Components Used

Ce document n'est pas limité à des versions de matériel et de logiciel spécifiques.

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. If your network is live, make sure that you understand the potential impact of any command.

Avertissements

Utilisation **debug** avec prudence. D'une manière générale, il est recommandé que ces commandes soient seulement utilisées sous les orientations de l'agent d'assistance technique de votre routeur pour le dépannage de problèmes spécifiques.

Permettre le débogage peut perturber le fonctionnement du routeur quand les interréseaux connaissent des conditions de charge élevée. Par conséquent, si la journalisation est activée, le serveur d'accès peut geler de façon intermittente dès que le port de console est surchargé de messages de journal.

Avant de commencer une **debug** , tenez toujours compte du résultat que cette commande va générer et du temps que cela peut prendre. Par exemple, si vous avez un routeur avec une interface BRI (basic rate interface), **debug isdn q931** probablement ne nuira pas au système. Mais faire le même débogage sur un AS5800 avec une configuration E1 complète peut probablement générer tellement d'entrées qu'il peut suspendre et arrêter de répondre.

Avant de déboguer, examinez la charge de votre processeur avec l' **show processes cpu** erasecat4000_flash:. Vérifiez que vous disposez d'un processeur suffisant avant de commencer les débogages. Référez-vous à Dépannage de l'utilisation élevée du CPU sur les routeurs Cisco pour plus d'informations sur la façon de gérer les charges élevées du CPU. Par exemple, si vous avez un routeur Cisco 7200 avec une interface ATM effectuant le pontage, le redémarrage du routeur peut utiliser une grande partie de son processeur, selon la quantité de sous-interfaces configurées. La raison à cela est que, pour chaque circuit virtuel (VC), un paquet Bridge Protocol Data Unit (BPDU) doit être généré. Initier des débogages à ce moment critique peut provoquer l'augmentation excessive de l'utilisation du CPU et résulter en une suspension et en une perte de connectivité du réseau.

> **Note**: Lorsque les débogages sont en cours d'exécution, vous ne voyez généralement pas l'invite du routeur, en particulier lorsque le débogage est intensif. Mais, dans la plupart des cas, vous pouvez utiliser les commandes no debug all ou undebug all afin d'arrêter les débogages. Référez-vous à la section Obtention des sorties de débogage pour plus d'informations sur l'utilisation sécurisée des débogages.

Conventions

Pour plus d'informations sur les conventions utilisées dans ce document, reportez-vous à Conventions relatives aux conseils techniques Cisco.

Avant de déboguer

En plus des points mentionnés ci-dessus, assurez-vous de comprendre l'incidence des débogages sur la stabilité de la plate-forme. Vous devriez également considérer sur quelle interface du routeur vous devriez vous connecter. Cette section contient des directives.

Obtenir des résultats du débogage

Les routeurs peuvent afficher des résultats du débogage sur diverses interfaces, y compris la console ainsi que les ports aux et vty. Les routeurs peuvent également consigner des messages vers un tampon interne vers un serveur syslog unix externe. Les instructions et les mises en garde pour chaque méthode sont discutées ci-dessous :

Port de console

Si vous êtes connecté à la console, dans des configurations normales, aucun travail supplémentaire n'est nécessaire. Le résultat du débogage devrait être automatiquement affiché. Mais assurez-vous que **logging console level** est défini comme souhaité et que la journalisation n'a pas été désactivée avec le **no logging console** erasecat4000_flash:.

> Avertissement : Les débogages excessifs du port de console d'un routeur peuvent causer sa suspension. Ceci se doit au fait que le routeur donne automatiquement la priorité à la sortie de console vis-à-vis d'autres fonctions du routeur. Par conséquent, si le routeur traite d'un grand résultat de débogage au port de console, il peut être suspendu. Par conséquent, si le résultat du débogage est excessif, utilisez les ports vty (telnet) ou les mémoires tampon de journal pour obtenir vos débogages. Plus d'informations sont fournies ci-dessous.

> Note: Par défaut, la journalisation est activée sur le port de console. Par conséquent, le port de console traite toujours du résultat de débogage même si vous employez réellement un autre port ou méthode (tel qu'Aux, vty ou mémoire tampon) pour saisir le résultat. Par conséquent, Cisco recommande que, dans des conditions normales de fonctionnement, la commande no logging console soit activée à tout moment et que vous utilisiez d'autres méthodes pour capturer les débogages. Dans les situations où vous devez utiliser la console, activez temporairement de nouveau logging console.

Port AUX

Si vous êtes connecté via un port auxiliaire, tapez la **terminal monitor** erasecat4000_flash:. Vérifiez également que **no logging on** n'a pas été activée sur le routeur.

> Note: Si vous utilisez le port aux pour contrôler le routeur, gardez présent à l'esprit que, quand le routeur redémarre, le port Aux n'affiche pas la séquence de démarrage. Connectez-vous au port de console afin d'afficher la séquence de démarrage.

Ports VTY

Si vous êtes connecté via un port auxiliaire ou via telnet, tapez la **terminal monitor** erasecat4000_flash:. Vérifiez également que **no logging on** n'a pas été utilisée.

Messages de journalisation à un tampon interne

La console est le périphérique de journalisation par défaut ; tous les messages sont affichés sur la console sauf indication contraire.

Pour enregistrer les messages dans une mémoire tampon interne, utilisez la commande **logging buffered** commande de configuration du routeur. Voici la syntaxe complète de cette commande :

```
logging buffered
no logging buffered
```

Les **logging buffered** copie les messages du journal dans une mémoire tampon interne au lieu de les écrire sur la console. La mémoire tampon est de nature circulaire, donc les messages les plus récents écrasent les messages plus anciens. Pour afficher les messages qui sont connectés dans la mémoire tampon, utilisez la commande EXEC privilégiée **show logging**. Le premier message affiché est le message le plus ancien dans la mémoire tampon. Vous pouvez spécifier la taille de la mémoire tampon ainsi que le niveau de gravité des messages à consigner.

> **Astuce** : Assurez-vous que suffisamment de mémoire est disponible dans la boîte avant d'introduire la taille du tampon. Utiliser Cisco IOS **show proc mem** afin de voir la mémoire disponible.

Les **no logging buffered** annule l'utilisation de la mémoire tampon et écrit les messages sur la console (valeur par défaut).

Messages de journalisation vers un serveur Syslog UNIX

Aux messages du journal à l'hôte du serveur syslog, utilisez la commande de configuration du routeur de journalisation. La syntaxe complète de cette commande suit :

```
logging no logging
```

Les **logging** identifie un hôte de serveur syslog pour recevoir les messages de journalisation. L'argument **<** *IP address* **>** est l'adresse IP de l'hôte. En émettant cette commande plus d'une fois, vous établissez une liste de serveurs syslog qui reçoivent des messages de journalisation.

Les **no logging** supprime le serveur syslog avec l'adresse spécifiée de la liste des syslogs.

D'autres tâches de prédébogage

1. Installez votre logiciel émulateur de terminal (par exemple, HyperTerminal) de sorte qu'il puisse saisir le résultat du débogage dans un fichier. Par exemple, dans HyperTerminal, cliquez sur **Transfer**, puis cliquez sur **Capture Text**, puis sélectionnez les options appropriées. Pour plus d'informations, reportez-vous à Saisir un résultat de texte de l'hyperterminal. Pour l'autre logiciel émulateur de terminal, reportez-vous à la documentation du logiciel.
2. Activer les horodatages millisecondes (ms) à l'aide de la **service timestamps** commande :

```
router(config)#service timestamps debug datetime msec
router(config)#service timestamps log datetime msec
```

Ces commandes ajoutent des horodatages aux débogages au format MMM DD HH:MM:SS, en indiquant la date et l'heure en fonction de l'horloge système. Si l'horloge système n'a pas été réglée, la date et l'heure sont précédées par un astérisque (*) pour indiquer que la date et l'heure

ne sont probablement pas correctes.

Il est généralement recommandé de configurer des horodatages en millisecondes dans la mesure où cela confère un haut niveau de clarté aux résultats du débogage. Les horodatages en millisecondes fournissent une meilleure indication de la temporisation des divers événements de débogages les uns envers les autres. Cependant, notez que, quand le port de console émet beaucoup de messages, ceux-ci pourraient ne pas correspondre à la véritable temporisation de l'événement. Par exemple, si vous activez **debug x25** tous sur une zone contenant 200 circuits virtuels, et la sortie est consignée dans la mémoire tampon (à l'aide de **no logging console** et **logging buffered**), l'horodatage affiché dans la sortie de débogage (dans la mémoire tampon) peut ne pas être l'heure exacte à laquelle le paquet passe par l'interface. Par conséquent, n'utilisez pas les horodatages en millisecondes pour justifier des problèmes de performances, mais pour obtenir une information relative sur le moment où les événements se produisent.

Pour cesser de déboguer

Pour arrêter un débogage, utilisez la **no debug all** ou **undebug all** . Vérifiez que les débogages ont été désactivés à l'aide de la commande **show debug**.

N'oubliez pas que les commandes **no logging console** et **terminal no monitor** empêche uniquement la sortie d'être affichée sur la console, respectivement Aux ou vty. Elle n'arrête pas le débogage et utilise donc des ressources du routeur.

Utilisation de la commande debug ip packet

Les **debug ip packet** génère des informations sur les paquets qui ne sont pas commutés rapidement par le routeur. Cependant, puisqu'il génère un résultat pour chaque paquet, le résultat peut être étendu et causer ainsi la suspension du routeur. Pour cette raison, utiliser uniquement **debug ip packet** sous les contrôles les plus stricts décrits dans la présente section.

Le meilleur moyen de limiter la sortie de **debug ip packet** est de créer une liste d'accès liée au débogage. Seuls les paquets qui correspondent aux critères de liste d'accès seront soumis à **debug ip packet**. Cette liste d'accès n'a pas besoin d'être appliquée sur aucune interface, mais est plutôt appliquée à l'opération de débogage.

Avant utilisation **debugging ip packet**, notez que le routeur effectue la commutation rapide par défaut, ou peut effectuer la commutation CEF si elle est configurée pour le faire. Ceci signifie que, une fois que ces techniques sont en place, le paquet n'est pas fourni au processeur, par conséquent, le débogage ne montre rien. Pour que cela fonctionne, vous devez désactiver la commutation rapide sur le routeur avec **no ip route-cache** (pour les paquets de monodiffusion) ou **no ip mroute-cache** (pour les paquets de multidiffusion). Ceci devrait être appliqué sur les interfaces où le trafic est censé s'écouler. Vérifiez-le avec le **show ip route** erasecat4000_flash:.

Avertissements :

- Désactiver la commutation rapide sur un routeur qui prend en charge un grand nombre de paquets peut causer un pic d'utilisation du CPU de sorte que le boîtier se suspende ou perde sa connexion avec ses pairs.
- Ne désactivez pas la commutation rapide sur un routeur exécutant une commutation multiprotocole par étiquette (MPLS). MPLS est utilisé en conjonction avec CEF. Par

conséquent, désactiver la commutation rapide sur l'interface peut avoir un effet désastreux. Considérons un exemple de scénario :

La liste d'accès configurée sur router_122 est :

```
access-list 105 permit icmp host 10.10.10.2 host 13.1.1.1
access-list 105 permit icmp host 13.1.1.1 host 10.10.10.2
```

Cette liste d'accès permet à n'importe quel paquet d'Internet Control Message Protocol (ICMP) de l'hôte router_121 (avec l'adresse IP 10.10.10.2) pour héberger router_123 (avec l'adresse IP 13.1.1.1) aussi bien que dans l'autre direction. Il est important que vous permettiez les paquets dans les deux directions, faute de quoi le routeur peut abandonner le paquet ICMP de renvoi.

Supprimez la commutation rapide sur une seule interface sur le routeur 122. Cela signifie que vous ne pouvez voir que les débogages des paquets destinés à cette interface, comme le montre le point de vue de l'IOS interceptant le paquet. À partir des débogages, de tels paquets apparaissent avec « d=« . Puisque vous n'avez pas encore désactivé la commutation rapide sur l'autre interface, le paquet de retour n'est pas soumis à **debug ip packet**. Ce résultat montre comment désactiver la commutation rapide :

```
router_122(config)#interface virtual-template 1
router_122(config-if)#no ip route-cache
router_122(config-if)#end
```

Vous devez maintenant activer **debug ip packet** avec la liste d'accès définie précédemment (liste d'accès 105).

```
router_122#debug ip packet detail 105
IP packet debugging is on (detailed) for access list 105
router_122#
00:10:01: IP: s=13.1.1.1 (Serial3/0), d=10.10.10.2 (Virtual-Access1),
g=10.10.10.2, len 100, forward

00:10:01:     ICMP type=0, code=0
! -- ICMP packet from 13.1.1.1 to 10.10.10.2. ! -- This packet is displayed because it matches
the ! -- source and destination requirements in access list 105 00:10:01: IP: s=13.1.1.1
(Serial3/0), d=10.10.10.2 (Virtual-Access1), g=10.10.10.2, len 100, forward 00:10:01: ICMP
type=0, code=0 00:10:01: IP: s=13.1.1.1 (Serial3/0), d=10.10.10.2 (Virtual-Access1),
g=10.10.10.2, len 100, forward 00:10:01: ICMP type=0, code=0
```

Maintenant, supprimons la commutation rapide sur l'autre interface (sur le router_122). Cela signifie que tous les paquets de ces deux interfaces sont désormais commutés par paquets (ce qui est une condition requise pour **debug ip packet**:

```
router_122(config)#interface serial 3/0
router_122(config-if)#no ip route-cache
router_122(config-if)#end

router_122#
00:11:57:  IP: s=10.10.10.2 (Virtual-Access1), d=13.1.1.1
(Serial3/0), g=172.16.1.6, len 100, forward
00:11:57:  ICMP type=8, code=0
! -- ICMP packet (echo) from 10.10.10.2 to 13.1.1.1 00:11:57: IP: s=13.1.1.1 (Serial3/0),
d=10.10.10.2 (Virtual-Access1),
g=10.10.10.2, len 100, forward
00:11:57:  ICMP type=0, code=0
! -- ICMP return packet (echo-reply) from 13.1.1.1 to 10.10.10.2 00:11:57: IP: s=10.10.10.2
(Virtual-Access1), d=13.1.1.1 (Serial3/0), g=172.16.1.6, len 100, forward 00:11:57: ICMP type=8,
code=0 00:11:57: IP: s=13.1.1.1 (Serial3/0), d=10.10.10.2 (Virtual-Access1), g=10.10.10.2, len
100, forward 00:11:57: ICMP type=0, code=0
```

Notez que le résultat debug ip packet ne montre aucun paquet qui ne correspond pas aux critères de la liste d'accès. Pour des informations supplémentaires sur cette procédure, reportez-vous à Comprendre les commandes ping et traceroute.

Pour plus d'informations sur la façon d'établir des listes d'accès, reportez-vous à Journalisation de la liste d'accès IP standard.

Débogages conditionnellement déclenchés

Quand la fonction de débogage conditionnellement déclenchée est activée, le routeur génère des messages de débogage pour des paquets entrant ou sortant du routeur sur une interface spécifiée ; le routeur ne génère pas le résultat du débogage pour des paquets entrant ou sortant par une interface différente.

Regardez une implémentation simple des débogages conditionnels. Considérez ce scénario: le routeur représenté ci-dessous (trabol) a deux interfaces (série 0 et série 3) exécutant toutes les deux une encapsulation HDLC.

Vous pouvez utiliser la **debug serial interface** afin d'observer les keepalives HDLC reçus sur toutes les interfaces. Vous pouvez observer les keepalives sur les deux interfaces.

```
traxbol#debug serial interface
Serial network interface debugging is on
traxbol#
*Mar  8 09:42:34.851: Serial0: HDLC myseq 28, mineseen 28*, yourseen 41, line up
! -- HDLC keeplaive on interface Serial 0 *Mar 8 09:42:34.855: Serial3: HDLC myseq 26, mineseen
26*, yourseen 27, line up
! -- HDLC keeplaive on interface Serial 3 *Mar 8 09:42:44.851: Serial0: HDLC myseq 29, mineseen
29*, yourseen 42, line up *Mar 8 09:42:44.855: Serial3: HDLC myseq 27, mineseen 27*, yourseen
28, line up
```

Activez les débogages conditionnels pour l'interface série 3. Ceci signifie que seuls les débogages pour l'interface de série 3 sont affichés. Utilisez **debug interface <*interface_type interface_number*>** erasecat4000_flash:.

```
traxbol#debug interface serial 3
Condition 1 set
```

Utilisez **show debug condition** afin de vérifier que le débogage conditionnel est actif. Notez qu'une

condition pour l'interface de la série 3 est active.

```
traxbol#show debug condition

Condition 1: interface Se3 (1 flags triggered)
Flags: Se3
traxbol#
```

Notez que maintenant seuls les débogages pour l'interface de la série 3 sont affichés

```
*Mar  8 09:43:04.855: Serial3: HDLC myseq 29, mineseen 29*, yourseen 30, line up
*Mar  8 09:43:14.855: Serial3: HDLC myseq 30, mineseen 30*, yourseen 31, line up
```

Utilisez **undebug interface <*interface_type interface_number*>** afin de supprimer le débogage conditionnel. Il est recommandé de désactiver les débogages (par exemple, en utilisant undebug all) avant de supprimer le déclencheur conditionnel. Ceci permet d'éviter un déluge de résultats de débogage quand la condition est supprimée.

```
traxbol#undebug interface serial 3
This condition is the last interface condition set.
Removing all conditions may cause a flood of debugging
messages to result, unless specific debugging flags
are first removed.
Proceed with removal? [yes/no]: y
Condition 1 has been removed
traxbol
```

Vous pouvez maintenant observer que le débogage de l'interface série 0 et de l'interface série 3 s'affiche.

```
*Mar  8 09:43:34.927: Serial3: HDLC myseq 32, mineseen 32*, yourseen 33, line up
*Mar  8 09:43:44.923: Serial0: HDLC myseq 35, mineseen 35*, yourseen 48, line up
```

> **Avertissement :** Quelques opérations de débogage sont conditionnelles en elles-mêmes. Un exemple réside en le débogage atm. Avec le débogage ATM, vous devriez explicitement spécifier l'interface pour laquelle des débogages devraient être activés plutôt qu'activer des débogages sur toutes les interfaces atm et spécifiant une condition.

Cette section montre la manière correcte de limiter le débogage de paquets ATM à une sous-interface :

```
arielle-nrp2#debug atm packet interface atm 0/0/0.1
!--- Note that you explicitly specify the sub-interface to be used for debugging ATM packets
debugging is on Displaying packets on interface ATM0/0/0.1 only
arielle-nrp2#
*Dec 21 10:16:51.891: ATM0/0/0.1(O):
VCD:0x1 VPI:0x1 VCI:0x21 DM:0x100 SAP:AAAA CTL:03 OUI:0080C2 TYPE:0007
Length:0x278
*Dec 21 10:16:51.891: 0000 FFFF FFFF FFFF 0010 7BB9 BDC4 0800 4500 025C 01FE
0000 FF11 61C8 0A30
*Dec 21 10:16:51.891: 4B9B FFFF FFFF 0044 0043 0248 0000 0101 0600 0015 23B7
0000 8000 0000 0000
*Dec 21 10:16:51.891: 0000 0000 0000 0000 0000 0000 0010 7BB9 BDC3 0000 0000
0000 0000 0000 0000
*Dec 21 10:16:51.891: 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000
```

```
*Dec 21 10:16:51.891: 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000
*Dec 21 10:16:51.891: 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000
*Dec 21 10:16:51.891: 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000
*Dec 21 10:16:51.895: 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000
*Dec 21 10:16:51.895:
arielle-nrp2#
```

Si vous essayez d'activer **atm debugging** sur toutes les interfaces (avec une condition appliquée), le routeur peut se suspendre s'il possède un grand nombre de sous-interfaces ATM. Un exemple de la méthode incorrecte pour le débogage atm est montré.

Dans ce cas, vous pouvez voir qu'une condition est appliquée, mais vous voyez également que ceci n'a aucun effet. Vous pouvez toujours voir le paquet de l'autre interface. Dans ce scénario de travaux pratiques, vous n'avez que deux interfaces et très peu de trafic. Si le nombre d'interfaces est élevé, le résultat du débogage de toutes les interfaces est extrêmement élevé et peut entraîner le blocage du routeur.

```
arielle-nrp2#show debugging condition
Condition 1: interface AT0/0/0.1 (1 flags triggered)
Flags: AT0/0/0.1
! -- A condition for a specific interface. arielle-nrp2#debug atm packet
ATM packets debugging is on
Displaying all ATM packets
arielle-nrp2#
*Dec 21 10:22:06.727: ATM0/0/0.2(O):
! -- You see debugs from interface ATM0/0/0/.2, even though the condition ! -- specified ONLY
AT0/0/0.1 VCD:0x2 VPI:0x5 VCI:0x37 DM:0x100 SAP:AAAA CTL:03 OUI:0080C2 TYPE:000E Length:0x2F
*Dec 21 10:22:06.727: 0000 0000 0180 0000 107B B9BD C400 0000 0080 0000 107B B9BD C480 0800 0014
*Dec 21 10:22:06.727: 0002 000F 0000 *Dec 21 10:22:06.727: un a *Dec 21 10:22:08.727:
ATM0/0/0.2(O): VCD:0x2 VPI:0x5 VCI:0x37 DM:0x100 SAP:AAAA CTL:03 OUI:0080C2 TYPE:000E
Length:0x2F *Dec 21 10:22:08.727: 0000 0000 0180 0000 107B B9BD C400 0000 0080 0000 107B B9BD
C480 0800 0014 *Dec 21 10:22:08.727: 0002 000F 0000 *Dec 21 10:22:08.727: ll *Dec 21
10:22:10.727: ATM0/0/0.2(O): VCD:0x2 VPI:0x5 VCI:0x37 DM:0x100 SAP:AAAA CTL:03 OUI:0080C2
TYPE:000E Length:0x2F *Dec 21 10:22:10.727: 0000 0000 0080 0000 107B B9BD C400 0000 0080 0000
107B B9BD C480 0800 0014 *Dec 21 10:22:10.727: 0002 000F 0000 *Dec 21 10:22:10.727: *Dec 21
10:22:12.727: ATM0/0/0.2(O): VCD:0x2 VPI:0x5 VCI:0x37 DM:0x100 SAP:AAAA CTL:03 OUI:0080C2
TYPE:000E Length:0x2F *Dec 21 10:22:12.727: 0000 0000 0080 0000 107B B9BD C400 0000 0080 0000
107B B9BD C480 0800 0014 *Dec 21 10:22:12.727: 0002 000F 0000 *Dec 21 10:22:12.727: *Dec 21
10:22:13.931: ATM0/0/0.1(O):
!--- You also see debugs for interface ATM0/0/0.1 as you wanted. VCD:0x1 VPI:0x1 VCI:0x21
DM:0x100 SAP:AAAA CTL:03 OUI:0080C2 TYPE:0007 Length:0x278 *Dec 21 10:22:13.931: 0000 FFFF FFFF
FFFF 0010 7BB9 BDC4 0800 4500 025C 027F 0000 FF11 6147 0A30 *Dec 21 10:22:13.931: 4B9B FFFF FFFF
0044 0043 0248 0000 0101 0600 001A 4481 0000 8000 0000 0000 *Dec 21 10:22:13.931: 0000 0000 0000
0000 0000 0000 0010 7BB9 BDC3 0000 0000 0000 0000 0000 0000 *Dec 21 10:22:13.931: 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 *Dec 21 10:22:13.931: 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 *Dec 21 10:22:13.931: 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 *Dec 21 10:22:13.931: 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 *Dec 21 10:22:13.935: 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
```

Informations connexes

- Numérotation et accès de l'assistance technique
- Support et documentation techniques - Cisco Systems