Comment détecter et effacer les connexions TCP bloquées à l'aide de SNMP

Contenu

Introduction

Ce document décrit comment utiliser le protocole SNMP (Simple Network Management Protocol) pour détecter et supprimer les connexions TCP suspendues sur un périphérique Cisco IOS. Le document explique également les objets SNMP que vous utilisez à cette fin.

La section intitulée Script PERL pour détecter et effacer les connexions TCP bloquées fournit un lien vers un script PERL qui implémente ces instructions.

Conditions préalables

Conditions requises

Les lecteurs de ce document devraient avoir connaissance des sujets suivants :

- Comprendre comment afficher les informations de connexion TCP sur les périphériques Cisco
- Utilisation générale des commandes **walk** SNMP, **get**, **get-next** et **set**
- Comprendre comment configurer le SNMP sur un appareil Cisco

Components Used

Ce document s'applique aux routeurs et commutateurs Cisco exécutant le logiciel IOS prenant en charge les modules TCP-MIB et CISCO-TCP-MIB.

Remarque : Le module CISCO-TCP-MIB n'est pas chargé par défaut dans NET-SNMP. Si le module MIB n'est pas chargé sur votre système, vous devez utiliser l'OID pour référencer un objet au lieu de son nom.

Les informations de ce document sont basées sur toutes les versions du logiciel et du matériel IOS.

Ces informations sont basées sur cette version de NET-SNMP :

- NET-SNMP version 5.1.2 disponible au http://www.net-snmp.org/

Le script PERL a été testé avec les versions PERL :

- 5.005_03 sur FreeBSD
- 5.8.0 sous Solaris 5.8
- 5.005_02 — livré dans le cadre de CiscoWorks SNMS sous Microsoft Windows 2000
- ActivePerl 5.8.4 sous Microsoft Windows 2000, disponible à l'adresse http://www.activestate.com/Products/ActivePerl/.

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.

Conventions

For more information on document conventions, refer to the Cisco Technical Tips Conventions.

Informations générales

Détails des objets MIB - Inclut les OID (Object Identifier)

Voici les objets que vous utilisez :

Àpartir du module CISCO-TCP-MIB :

- ciscoTcpConnInBytes, OID .1.3.6.1.4.1.9.9.6.1.1.1.1Nombre d'octets entrés sur cette connexion.
- ciscoTcpConnInPkts, OID 1.3.6.1.4.1.9.9.6.1.1.1.2Nombre de paquets entrés sur cette connexion.
- ciscoTcpConnOutBytes, OID .1.3.6.1.4.1.9.9.6.1.1.1.3Nombre d'octets de sortie sur cette connexion
- CiscoTcpConnOutPkts, OID .1.3.6.1.4.1.9.9.6.1.1.1.4Nombre de paquets en sortie sur cette connexion.
- PaquetsRétransConnCisco, OID .1.3.6.1.4.1.9.9.6.1.1.1.7Nombre de paquets retransmis sur cette connexion.
- ciscoTcpConnRto, OID .1.3.6.1.4.1.9.9.6.1.1.1.9Valeur de délai de retransmission pour cette connexion.

Àpartir du module TCP-MIB :

- tcpConnState, OID .1.3.6.1.2.1.6.13.1.1État de cette connexion.

Vous trouverez plus de détails sur ces objets dans Informations détaillées sur les objets MIB.

Utiliser SNMP pour détecter si une connexion TCP est bloquée

Résumé

Ces étapes vous aident à déterminer si une connexion TCP est suspendue :

1. Afin de déterminer si les objets ciscoTcpConnRetransPkts et ciscoTcpConnRto sont pris en charge dans le périphérique, exécutez une opération **get-next** SNMP sur ciscoTcpConnRto et vérifiez si des objets sont retournés.**Remarque :** Il vous suffit de vérifier un objet car la prise en charge des deux a été ajoutée en même temps.**Remarque :** Tous les périphériques Cisco ne prennent pas en charge les deux derniers objets (ciscoTcpConnRetransPkts et ciscoTcpConnRto), mais leur utilisation peut améliorer la précision de la détection.Si les objets ciscoTcpConnRetransPkts et ciscoTcpConnRto **sont** pris en charge, passez à l'étape 2.Si les objets ciscoTcpConnRetransPkts et ciscoTcpConnRto ne **sont pas** pris en charge, passez à l'étape 3.
2. Tous les objets sont pris en charge. Pour chaque connexion TCP, vérifiez les points suivants :ciscoTcpConnOutBytes est égal à 0.ciscoTcpConnOutPkts est égal à 0.ciscoTcpConnRetransPkts est supérieur à 0.ciscoTcpConnRto est supérieur à 20 000.**Remarque :** Les 20 000 peuvent être réduites pour accélérer la détection. Il faut environ une minute pour que Rto atteigne 20 000 une fois la connexion suspendue. Cependant, des valeurs plus petites peuvent réduire la précision du résultat.Si tous les précédents sont vrais, cette connexion TCP est suspendue et peut être effacée. Passez à l'utilisation de SNMP pour effacer une connexion TCP qui se bloque.
3. Seuls les quatre premiers objets sont pris en charge. Pour chaque connexion TCP, vérifiez les points suivants :ciscoTcpConnInBytes est supérieur à 0.ciscoTcpConnInPkts est égal à 0.ciscoTcpConnOutBytes est égal à 0.ciscoTcpConnOutPkts est égal à 0.Patientez quelques secondes et **récupérez** les objets pour vérifier qu'il ne s'agissait pas d'une connexion TCP en cours d'établissement.**Remarque :** Les deux premiers contrôles (un nombre positif d'octets d'entrée mais aucun paquet d'entrée) peuvent sembler étranges, mais ils ont été vérifiés par rapport à de nombreux périphériques et versions d'IOS.**Remarque :** les versions IOS qui prennent en charge les six objets peuvent ne pas présenter ce comportement et, par conséquent, le test de l'étape 2 n'inclut pas ces deux premiers tests.Si tous les objets répondent aux deux tests, cette connexion TCP est suspendue et peut être effacée. Passez à l'utilisation de SNMP pour effacer une connexion TCP qui se bloque.

Step-by-Step Instructions

Les valeurs de cet exemple sont les suivantes :

- Nom d'hôte du périphérique a = nms-7206a (prend en charge tous les objets)
- Nom d'hôte du périphérique b = nms-1605 (ne prend en charge que les quatre premiers objets)
- Communauté de lecture = public
- Communauté d'écriture = privé

Remplacez les chaînes de communauté et le nom d'hôte dans ces commandes :

1. Déterminez si ce périphérique prend en charge les objets ciscoTcpConnRetransPkts et ciscoTcpConnRto :Exécuter une opération **SNMP get-next** sur ciscoTcpConnRto :
   ```
   snmpgetnext -c public nms-7206a ciscoTcpConnRto
   ```

   Si les objets **sont** pris en charge, une réponse comme celle-ci s'affiche :
   ```
   CISCO-TCP-MIB::ciscoTcpConnRto.14.32.100.75.2065.172.18.86.111.23092 =
        INTEGER: 303 milliseconds
   ```

   Remarque : L'index utilisé pour ces objets, dans ce cas `14.32.100.75.2065.172.18.86.111.23092`, est une concaténation de l'adresse IP locale : 14.32.10 0.75, le numéro de port TCP local (2065), l'adresse IP distante (172.18.86.111) et le numéro de port TCP distant (23092).Le retour est pour ciscoTcpConnRto. Passez à l'étape 2.Si les objets **ne** sont **pas** pris en charge, vous voyez une réponse comme celle-ci :
   ```
   snmpgetnext -c public nms-1605 ciscoTcpConnRto
   CISCO-FLASH-MIB::ciscoFlashDevicesSupported.0 = INTEGER: 1
   ```

   Le retour **n'est pas** pour l'objet ciscoTcpConnRto. L'objet exact retourné n'est pas important. Passez à l'étape 3.
2. **Obtenir** des informations sur chaque connexion TCP pour les périphériques qui prennent en charge les six objets de la table de connexion TCP Cisco.Effectuez une opération SNMP get-next sur ciscoTcpConnOutBytes, ciscoTcpConnOutPkts, ciscoTcpConnRetransPkts et ciscoTcpConnRto :
   ```
   snmpgetnext -c public nms-7206a ciscoTcpConnOutBytes
                         ciscoTcpConnOutPkts
                         ciscoTcpConnRetransPkts
                         ciscoTcpConnRto
   ```

   Voici une réponse :
   ```
   CISCO-TCP-MIB::ciscoTcpConnOutBytes.14.32.100.75.2065.172.18.86.111.23092 = Counter32:
   383556
   CISCO-TCP-MIB::ciscoTcpConnOutPkts.14.32.100.75.2065.172.18.86.111.23092 = Counter32: 8061
   CISCO-TCP-MIB::ciscoTcpConnRetransPkts.14.32.100.75.2065.172.18.86.111.23092 = Counter32: 2
   CISCO-TCP-MIB::ciscoTcpConnRto.14.32.100.75.2065.172.18.86.111.23092 = INTEGER: 303
   milliseconds
   ```

   Vérifiez ces éléments :ciscoTcpConnOutBytes est égal à 0.ciscoTcpConnOutPkts est égal à 0.ciscoTcpConnRetransPkts est supérieur à 0.ciscoTcpConnRto est supérieur à 20 000.**Remarque :** Les 20 000 peuvent être réduites pour accélérer la détection. Il faut environ une minute pour que Rto atteigne 20 000 une fois la connexion suspendue. Cependant, des valeurs plus petites peuvent réduire la précision du résultat.Si tous ces éléments sont vrais, cette connexion TCP est suspendue et peut être effacée. Passez à l'utilisation de SNMP pour effacer une connexion TCP qui se bloque.Continuez à **parcourir** la table de connexion TCP. Pour ce faire, exécutez une opération **get-next** SNMP à plusieurs reprises lorsque vous recherchez des connexions bloquées, à l'aide des objets retournés tels que :
   ```
   snmpgetnext -c public nms-7206a ciscoTcpConnOutBytes.14.32.100.75.2065.172.18.86.111.23092
                         ciscoTcpConnOutPkts.14.32.100.75.2065.172.18.86.111.23092
                         ciscoTcpConnRetransPkts.14.32.100.75.2065.172.18.86.111.23092
                         ciscoTcpConnRto.14.32.100.75.2065.172.18.86.111.23092
   ```

Vérifiez chaque entrée à l'aide du test précédent jusqu'à ce que l'opération **get-next** retourne les objets de cette manière :

```
CISCO-TCP-MIB::ciscoTcpConnInPkts.14.32.100.75.2065.172.18.86.111.23092 = Counter32: 8097
CISCO-TCP-MIB::ciscoTcpConnElapsed.14.32.100.75.2065.172.18.86.111.23092 =
  Timeticks: (17296508) 2 days, 0:02:45.08
CISCO-TCP-MIB::ciscoTcpConnFastRetransPkts.14.32.100.75.2065.172.18.86.111.23092 =
Counter32: 0
CISCO-FLASH-MIB::ciscoFlashDevicesSupported.0 = INTEGER: 5
```

Vous avez maintenant effectué toutes les connexions TCP sur ce périphérique et vous avez terminé.

3. **Obtenir** des informations sur chaque connexion TCP pour les périphériques qui prennent uniquement en charge les quatre premiers objets de la table de connexion TCP Cisco.Effectuez une opération **Get-next** SNMP sur ciscoTcpConnInBytes, ciscoTcpConnInPkts ciscoTcpConnOutBytes et ciscoTcpConnOutPkts :

```
snmpgetnext -c public nms-1605 ciscoTcpConnInBytes
                       ciscoTcpConnInPkts
                       ciscoTcpConnOutBytes
                       ciscoTcpConnOutPkts
```

Voici une réponse :

```
CISCO-TCP-MIB::ciscoTcpConnInBytes.14.32.6.185.23.14.32.100.33.2249 = Counter32: 68
CISCO-TCP-MIB::ciscoTcpConnInPkts.14.32.6.185.23.14.32.100.33.2249 = Counter32: 12
CISCO-TCP-MIB::ciscoTcpConnOutBytes.14.32.6.185.23.14.32.100.33.2249 = Counter32: 170
CISCO-TCP-MIB::ciscoTcpConnOutPkts.14.32.6.185.23.14.32.100.33.2249 = Counter32: 17
```

Vérifiez si ces informations sont vraies :ciscoTcpConnInBytes est supérieur à 0.ciscoTcpConnInPkts est égal à 0.ciscoTcpConnOutBytes est égal à 0.ciscoTcpConnOutPkts est égal à 0.Attendez quelques secondes et **récupérez** les objets. Vérifiez qu'il ne s'agissait pas d'une connexion TCP en cours d'établissement.Si tous les éléments ci-dessus **sont** vrais, cette connexion TCP est suspendue et peut être effacée. Passez à l'utilisation de SNMP pour effacer une connexion TCP qui se bloque.Continuez à **parcourir** la table de connexion TCP. Pour ce faire, exécutez une opération **get-next** SNMP à plusieurs reprises lorsque vous recherchez des connexions bloquées, à l'aide des objets retournés tels que :

```
snmpgetnext -c public nms-1605 ciscoTcpConnInBytes.14.32.6.185.23.14.32.100.33.2249
                       ciscoTcpConnInPkts.14.32.6.185.23.14.32.100.33.2249
                       ciscoTcpConnOutBytes.14.32.6.185.23.14.32.100.33.2249
                       ciscoTcpConnOutPkts.14.32.6.185.23.14.32.100.33.2249
```

Vérifiez chaque entrée à l'aide du test précédent jusqu'à ce que l'opération **get-next** retourne les objets de cette manière :

```
CISCO-TCP-MIB::ciscoTcpConnOutBytes.14.32.6.185.23.14.32.100.33.4184 = Counter32: 170
CISCO-TCP-MIB::ciscoTcpConnOutPkts.14.32.6.185.23.14.32.100.33.4184 = Counter32: 17
CISCO-TCP-MIB::ciscoTcpConnInPkts.14.32.6.185.23.14.32.100.33.4184 = Counter32: 12
CISCO-TCP-MIB::ciscoTcpConnElapsed.14.32.6.185.23.14.32.100.33.4184 = Timeticks: (4345)
0:00:43.45
```

Vous avez maintenant effectué toutes les connexions TCP sur ce périphérique et vous avez terminé.

Utiliser SNMP pour effacer une connexion TCP qui se bloque

Step-by-Step Instructions

Vous pouvez utiliser SNMP pour effacer une connexion TCP interrompue. La commande SNMP est équivalente à la commande clear tcp local *<local_ip>* *<local_port>* **remote** *<remote_ip>* *<remote_port>*. L'objet que vous utilisez pour effacer une ligne est **tcpConnState**.

Afin d'effacer une connexion TCP interrompue avec SNMP, émettez cette commande :

```
snmpset -c private nms-7206a tcpConnState.14.32.100.75.2065.172.18.86.111.23092 integer
deleteTCB
TCP-MIB::tcpConnState.14.32.100.75.2065.172.18.86.111.23092 = INTEGER: deleteTCB(12)
```

Remarque : L'index utilisé pour ces objets, dans ce cas `14.32.100.75.2065.172.18.86.111.23092`, est une concaténation de l'adresse IP locale : 14.32.10 0.75, le numéro de port TCP local (2065), l'adresse IP distante (172.18.86.111) et le numéro de port TCP distant (23092).

Remarque : Vous devez utiliser l'index exact que vous avez déterminé comme étant suspendu dans Utiliser SNMP pour détecter si une connexion TCP est interrompue. Notez que cette commande déconnecte une connexion TCP sans avertissement.

Informations détaillées sur les objets MIB

```
.1.3.6.1.4.1.9.9.6.1.1.1.1
ciscoTcpConnInBytes OBJECT-TYPE
        -- FROM CISCO-TCP-MIB
        SYNTAX          Counter
        MAX-ACCESS      read-only
        STATUS          Current
        DESCRIPTION     "Number of bytes that have been input on this TCP
                        connection."
::= { ciscoTcpConnEntry 1 }

.1.3.6.1.4.1.9.9.6.1.1.1.2
ciscoTcpConnOutBytes OBJECT-TYPE
        -- FROM CISCO-TCP-MIB
        SYNTAX          Counter
        MAX-ACCESS      read-only
        STATUS          Current
        DESCRIPTION     "Number of bytes that have been output on this TCP
                        connection."
::= { ciscoTcpConnEntry 2 }

.1.3.6.1.4.1.9.9.6.1.1.1.3
ciscoTcpConnInPkts OBJECT-TYPE
        -- FROM CISCO-TCP-MIB
        SYNTAX          Counter
        MAX-ACCESS      read-only
        STATUS          Current
        DESCRIPTION     "Number of packets that have been input on this TCP
                        connection."
::= { ciscoTcpConnEntry 3 }

.1.3.6.1.4.1.9.9.6.1.1.1.4
ciscoTcpConnOutPkts OBJECT-TYPE
        -- FROM CISCO-TCP-MIB
        SYNTAX          Counter
        MAX-ACCESS      read-only
```

```
        STATUS          Current
        DESCRIPTION     "Number of packets that have been output on this TCP
                        connection."
::= { ciscoTcpConnEntry 4 }

.1.3.6.1.4.1.9.9.6.1.1.1.7
ciscoTcpConnRetransPkts OBJECT-TYPE
        -- FROM CISCO-TCP-MIB
        SYNTAX          Counter
        MAX-ACCESS      read-only
        STATUS          Current
        DESCRIPTION     "The total number of packets retransmitted due to a timeout -
                        that is, the number of TCP segments transmitted containing
                        one or more previously transmitted octets."
::= { ciscoTcpConnEntry 7 }

.1.3.6.1.4.1.9.9.6.1.1.1.9
ciscoTcpConnRto OBJECT-TYPE
        -- FROM CISCO-TCP-MIB
        SYNTAX          Integer
        MAX-ACCESS      read-only
        STATUS          Current
        DESCRIPTION     "The current value used by a TCP implementation for the
                        retransmission timeout."
::= { ciscoTcpConnEntry 9 }

.1.3.6.1.2.1.6.13.1.1
tcpConnState OBJECT-TYPE
        -- FROM RFC1213-MIB
        SYNTAX          Integer { closed(1), listen(2), synSent(3), synReceived(4),
                        established(5), finWait1(6), finWait2(7), closeWait(8), lastAck(9),
                        closing(10), timeWait(11), deleteTCB(12) }
        MAX-ACCESS      read-write
        STATUS          Mandatory
        DESCRIPTION     "The state of this TCP connection.

                        The only value which may be set by a management
                        station is deleteTCB(12).  Accordingly, it is
                        appropriate for an agent to return a `badValue'
                        response if a management station attempts to set
                        this object to any other value.

                        If a management station sets this object to the
                        value deleteTCB(12), then this has the effect of
                        deleting the TCB (as defined in RFC 793) of the
                        corresponding connection on the managed node,
                        resulting in immediate termination of the
                        connection.

                        As an implementation-specific option, a RST

                        segment may be sent from the managed node to the
                        other TCP endpoint (note however that RST segments
                        are not sent reliably)."
::= { tcpConnEntry 1 }
```

Script PERL pour détecter et effacer les connexions TCP bloquées

Ce lien fournit un fichier d'archive avec un script PERL et les modules MIB nécessaires. Cliquez avec le bouton droit sur le lien et enregistrez le fichier sur votre système.

- fixTCPhang.tgz

Les fichiers de l'archive sont les suivants :

- bin/fixTCPhang.pl
- mibs/CISCO-SMI.my
- mibs/CISCO-TCP-MIB.my

Pour extraire le script et les modules MIB, utilisez un utilitaire tel que gzip et tar sur un système d'exploitation UNIX. Par exemple, pour extraire les fichiers dans **/tmp** en supposant que le fichier d'archive est placé dans **/tmp** :

```
cd /tmp; gzip -dc fixTCPhang.tgz | tar -xvf -
```

Remarque : Vous devrez peut-être modifier la première ligne du script pour spécifier l'emplacement de PERL.

Utilisez winzip ou d'autres utilitaires sur les systèmes d'exploitation Microsoft Windows pour extraire les fichiers. Si vous extrayez les fichiers vers **c:\tmp** alors vous n'avez pas à spécifier l'option -m lorsque vous exécutez le script.

Appelez les fichiers avec cette commande :

```
fixTCPhang.pl -c public -C private -f nms-7206a
```

Pour chaque connexion TCP interrompue trouvée, une ligne comme celle-ci apparaît :

```
Found bad TCP connection: Local IP: 14.32.100.75 port 23 Remote IP: 172.18.100.33 port 47878:
CLEARED
```

Lorsque la chaîne de communauté en lecture-écriture a été fournie et que l'option -f a été spécifiée, le script a effacé la connexion. Notez l'instruction CLEARED à la fin du résultat.

Le script prend en charge SNMP versions 1, 2c et 3. Si vous spécifiez SNMP version 3, vous devez spécifier toutes les informations d'authentification dans l'argument -v. Voici un exemple d'utilisation de SNMP v3 :

```
fixTCPhang.pl -v "3 -a MD5 -u chelliot -A chelliot -l authNoPriv" -f nms-dmz-ap1200-b
```

Les commandes IOS permettant de configurer SNMP v3 pour l'exemple précédent sont les suivantes :

```
snmp-server group chelliot-group v3 auth write v1default
snmp-server user chelliot chelliot-group v3 auth md5 chelliot
```

Note : Il semble y avoir un bogue dans la version Windows de NET-SNMP utilisé dans ce test. Le bogue ne permet pas à l'authentification SHA de fonctionner correctement.

Vous pouvez utiliser plusieurs autres options avec ce script. Certaines des options de script incluent où trouver les utilitaires de ligne de commande NET-SNMP et où trouver les modules MIB s'ils ne sont pas dans **/tmp/mibs**. Vous pouvez également afficher ce résumé de ces options :

```
fixTCPhang.pl
fixTCPhang.pl [-dfhV -c <read_community> -C <write_community> -m <mib_directory>
               -p <command_path> -t <timeout> -v <snmp_version>] <device>
  Version 1.2
  Detect hung TCP connections on <device>, optionally clearing them.
  Options:  -c  Specify read community string. Defaults to public.
            -C  Specify the readwrite community string. No default.
                 Must be supplied for the script to clear hung connections.
            -d  Turn on debug mode.
            -f  Fix or clear any hung TCP connections found.
            -h  Print this message.
            -m  Specify the directory to find CISCO-SMI.my and CISCO-TCP-MIB.my.
                 Defaults to /tmp/mibs.
            -p  Where to find the net-snmp utilities.
                 Optional if the utilities are in the path.
            -t  SNMP Timeout value. Defaults to 5 sec.
            -v  Specify SNMP version to use: One of 1, 2c, or 3.
                 If 3 is specified then this option must include all of the
                 authentication information for SNMPv3. For example:
                 "3 -a MD5 -u chelliot -A chelliot -l authNoPriv"
                 Note: NET-SNMP seems to have a bug with SHA authentication on Windows.
                 See the NET-SNMP documentation for more information.
                 Defaults to SNMP version 1.
            -V  Print version number.
```

Informations connexes

- Support technique - Cisco Systems