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

Contenu

Introduction

Ce document décrit comment employer le Protocole SNMP (Simple Network Management Protocol) pour détecter et les connexions TCP clairement arrêtées sur un périphérique de Cisco IOS. Le document explique également le SNMP objecte que vous utilisez à cet effet.

La section autorisée, le script Perl à détecter et les connexions TCP clairement arrêtées, fournit un lien à un script Perl qui implémente ces instructions.

Conditions préalables

Conditions requises

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

- Comprenez comment visualiser les informations de connexion TCP sur des périphériques de Cisco
- L'utilisation générale de l'**inspection** SNMP, **obtiennent**, obtenir-**prochain**, et des **commandes set**
- Comprenez comment configurer le SNMP sur un périphérique de Cisco

Composants utilisés

Ce document applique à Cisco des Routeurs et des Commutateurs exécutant le logiciel IOS prenant en charge le TCP-MIB et les modules CISCO-TCP-MIB.

Note: 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 employer l'OID pour mettre en référence un objet au lieu de son nom.

Les informations dans ce document sont basées sur tout le logiciel et versions de matériel IOS.

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

- Version 5.1.2 NET-SNMP disponible chez http://www.net-snmp.org/

Le script Perl a été testé avec des versions Perl :

- 5.005_03 sur le FreeBSD
- 5.8.0 sur Solaris 5.8
- 5.005_02 — expédié en tant qu'élément des CiscoWorks SNMS sur le Microsoft Windows 2000
- ActivePerl 5.8.4 sur le Microsoft Windows 2000, disponible chez http://www.activestate.com/Products/ActivePerl/ .

Les informations contenues dans ce document ont été créées à partir des périphériques d'un environnement de laboratoire spécifique. Tous les périphériques utilisés dans ce document ont démarré avec une configuration effacée (par défaut). Si votre réseau est opérationnel, assurez-vous que vous comprenez l'effet potentiel de toute commande.

Conventions

Pour plus d'informations sur les conventions de documents, reportez-vous à Conventions relatives aux conseils techniques Cisco.

Informations générales

Détails des objets MIB — Inclut les identifiants d'objet (les OID)

Ce sont les objets que vous utilisez :

Du module CISCO-TCP-MIB :

- ciscoTcpConnInBytes, OID .1.3.6.1.4.1.9.9.6.1.1.1.1Le nombre d'octets a entré sur cette connexion.
- ciscoTcpConnInPkts, OID 1.3.6.1.4.1.9.9.6.1.1.1.2Le nombre de paquets a entré sur cette connexion.
- ciscoTcpConnOutBytes, OID .1.3.6.1.4.1.9.9.6.1.1.1.3Le nombre d'octets a sorti sur cette connexion
- ciscoTcpConnOutPkts, OID .1.3.6.1.4.1.9.9.6.1.1.1.4Le nombre de paquets a sorti sur cette connexion.
- ciscoTcpConnRetransPkts, OID .1.3.6.1.4.1.9.9.6.1.1.1.7Le nombre de paquets retransmis

sur cette connexion.
- ciscoTcpConnRto, OID .1.3.6.1.4.1.9.9.6.1.1.1.9La valeur du dépassement de durée de retransmettre pour cette connexion.

Du module TCP-MIB :

- tcpConnState, OID .1.3.6.1.2.1.6.13.1.1L'état pour cette connexion.

Il y a plus de détails sur ces objets dans les informations détaillées d'objet MIB.

Employez le SNMP pour le détecter si une connexion TCP s'arrête

Résumé

Ces étapes vous aident à déterminer si une connexion TCP s'arrête :

1. Afin de déterminer si les ciscoTcpConnRetransPkts et les objets de ciscoTcpConnRto sont pris en charge dans le périphérique, exécutez une exécution de snmp get-next sur le ciscoTcpConnRto et la vérifiez si des objets sont retournés.**Note**: Vous devez seulement vérifier un objet parce que le soutien de chacun d'eux a été ajouté en même temps.**Note**: Non tous les périphériques de Cisco prennent en charge les deux derniers objets (des ciscoTcpConnRetransPkts et ciscoTcpConnRto), mais leur utilisation peut augmenter la précision de la détection.Si les ciscoTcpConnRetransPkts et les objets de ciscoTcpConnRto sont pris en charge, passez à l'étape 2.Si les ciscoTcpConnRetransPkts et les objets de ciscoTcpConnRto ne sont pas pris en charge, passez à l'étape 3.
2. Tous les objets sont pris en charge. Pour chaque contrôle de connexion TCP ceux-ci :les ciscoTcpConnOutBytes est 0.les ciscoTcpConnOutPkts est 0.les ciscoTcpConnRetransPkts est plus grand que 0.le ciscoTcpConnRto est plus grand que 20,000.**Note**: Les 20,000 peuvent être réduits pour accélérer la détection. Cela prend une minute ou ainsi pour que Rto atteigne 20,000 une fois que la connexion est arrêtée. Cependant, de plus petites valeurs peuvent réduire la précision du résultat.Si tous les précédents sont vrais, alors cette connexion TCP est arrêtée et peut être effacée. Poursuivez pour employer le SNMP pour effacer une connexion TCP qui s'arrête.
3. Seulement les quatre premiers objets sont pris en charge. Pour chaque contrôle de connexion TCP ceux-ci :les ciscoTcpConnInBytes est plus grand que 0.les ciscoTcpConnInPkts est 0.les ciscoTcpConnOutBytes est 0.les ciscoTcpConnOutPkts est 0.Attendez quelques secondes et **obtenez les** objets de nouveau pour vérifier que ce n'était pas une connexion TCP en cours d'être établi.**Note**: Les deux premiers contrôles (un nombre positif d'octets d'entrée mais d'aucun paquets en entrée) peuvent sembler étranges, mais eux ont été vérifiés contre de nombreux périphériques et versions IOS.**Note**: Les versions IOS qui prennent en charge chacun des six objets peuvent ne pas montrer ce comportement et, en conséquence, le test dans l'étape 2 n'inclut pas ces deux premiers tests.Si tout les rassemblement d'objets les tests les deux fois alors cette connexion TCP est arrêté et peut être effacé. Poursuivez pour employer le SNMP pour effacer une connexion TCP qui s'arrête.

Instructions pas à pas

Les valeurs dans cet exemple sont :

- Adresse Internet de périphérique a = nms-7206a (prend en charge tous les objets)
- Adresse Internet de périphérique b = nms-1605 (supports seulement les quatre premiers objets)
- La communauté à accès en lecture = le public
- Écrivez la communauté = privé

Remplacez les chaînes de la communauté et l'adresse Internet dans ces commandes :

1. Déterminez si ce périphérique prend en charge les ciscoTcpConnRetransPkts et les objets de ciscoTcpConnRto :Exécutez une exécution de **snmp get-next** sur le ciscoTcpConnRto :

   ```
   snmpgetnext -c public nms-7206a ciscoTcpConnRto
   ```

   Si les objets sont pris en charge vous voyez une réponse comme ceci :

   ```
   snmpgetnext -c public nms-7206a ciscoTcpConnRto
   ```

   Note: L'index utilisé pour ces objets, dans ce cas `14.32.100.75.2065.172.18.86.111.23092`, est un enchaînement de l'adresse IP locale — 14.32.100.75, le nombre local de port TCP — 2065, l'adresse IP distante — 172.18.86.111, et le nombre distant de port TCP — 23092.Le retour est pour le ciscoTcpConnRto. Passez à l'étape 2.Si les objets ne sont pas support, vous voyez une réponse comme ceci :

   ```
   snmpgetnext -c public nms-1605 ciscoTcpConnRto
   CISCO-FLASH-MIB::ciscoFlashDevicesSupported.0 = INTEGER: 1
   ```

   Le retour n'est pas pour l'objet de ciscoTcpConnRto. L'objet précis retourné n'est pas important. Passez à l'étape 3.
2. **Obtenez les** informations sur chaque connexion TCP pour les périphériques qui prennent en charge chacun des six objets dans la table de connexion TCP de Cisco.Exécutez une exécution de **snmp get-next** sur des ciscoTcpConnOutBytes, des ciscoTcpConnOutPkts, des ciscoTcpConnRetransPkts, et le ciscoTcpConnRto :

   ```
   snmpgetnext -c public nms-7206a ciscoTcpConnOutBytes
                         ciscoTcpConnOutPkts
                         ciscoTcpConnRetransPkts
                         ciscoTcpConnRto
   ```

   Vous voyez une réponse comme ceci :

   ```
   snmpgetnext -c public nms-7206a ciscoTcpConnOutBytes
                         ciscoTcpConnOutPkts
                         ciscoTcpConnRetransPkts
                         ciscoTcpConnRto
   ```

   Vérifiez ces derniers :les ciscoTcpConnOutBytes est 0.les ciscoTcpConnOutPkts est 0.les ciscoTcpConnRetransPkts est plus grand que 0.le ciscoTcpConnRto est plus grand que 20,000.**Note**: Les 20,000 peuvent être réduits pour accélérer la détection. Cela prend une minute ou ainsi pour que Rto atteigne 20,000 une fois que la connexion est arrêtée. Cependant, de plus petites valeurs peuvent réduire la précision du résultat.Si toute la ces

derniers est vraie, alors cette connexion TCP est arrêtée et peut être effacée. Poursuivez pour employer le SNMP pour effacer une connexion TCP qui s'arrête.Continuez **à marcher la** table de connexion TCP. Afin de faire ceci, exécutez une exécution de snmp get-next à plusieurs reprises comme vous vérifiez les connexions arrêtées, utilisant les objets retournés de ce type :

```
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 utilisant le test précédent jusqu'à ce que l'obtenir-**prochaine** exécution renvoie des objets de cette manière :

```
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
```

Vous avez maintenant marché toutes les connexions TCP sur ce périphérique et vous êtes fait.

3. **Obtenez les** informations sur chaque connexion TCP pour les périphériques qui prennent en charge seulement les quatre premiers objets dans la table de connexion TCP de Cisco.Exécutez une exécution de snmp get-next sur des ciscoTcpConnInBytes, des ciscoTcpConnOutBytes de ciscoTcpConnInPkts, et des ciscoTcpConnOutPkts :

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

Vous voyez une réponse comme ceci :

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

Vérifiez pour voir si ce sont vrais :les ciscoTcpConnInBytes est plus grand que 0.les ciscoTcpConnInPkts est 0.les ciscoTcpConnOutBytes est 0.les ciscoTcpConnOutPkts est 0.Attendez quelques secondes et **obtenez les** objets de nouveau. Vérifiez que ce n'était pas une connexion TCP en cours d'être établi.Si tous les ci-dessus **sont** vrais, alors cette connexion TCP est arrêtée et peut être effacée. Poursuivez pour employer le SNMP pour effacer une connexion TCP qui s'arrête.Continuez **à marcher la** table de connexion TCP. Afin de faire ceci, exécutez une exécution de snmp get-next à plusieurs reprises comme vous vérifiez les connexions arrêtées, utilisant les objets retournés de ce type :

```
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 utilisant le test précédent jusqu'à ce que l'obtenir-**prochaine** exécution renvoie des objets de cette manière :

```
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
```

Vous avez maintenant marché toutes les connexions TCP sur ce périphérique et vous êtes fait.

Employez le SNMP pour effacer une connexion TCP qui s'arrête

Instructions pas à pas

Vous pouvez employer le SNMP pour effacer une connexion TCP arrêtée. La commande SNMP est équivalente au **<local_ip > au <local_port locaux de clear tcp > <remote_ip distant > <remote_port >** commande. L'objet que vous utilisez pour effacer une ligne est **tcpConnState**.

Afin d'effacer une connexion TCP arrêtée avec le 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)
```

Note: L'index utilisé pour ces objets, dans ce cas `14.32.100.75.2065.172.18.86.111.23092`, est un enchaînement de l'adresse IP locale — 14.32.100.75, le nombre local de port TCP — 2065, l'adresse IP distante — 172.18.86.111, et le nombre distant de port TCP — 23092.

Note: Vous devez utiliser l'index précis que vous avez déterminé étiez SNMP en service arrêté au détecter si une connexion TCP s'arrête. Rendez-vous compte que cette commande déconnecte une connexion TCP sans avertissement.

Les informations détaillées d'objet MIB


```
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)
```


Script Perl au détecter et connexions TCP clairement arrêtées

Ce lien fournit à un fichier d'archivage un script Perl et les modules nécessaires MIB. Cliquez avec le bouton droit le lien et sauvegardez le fichier à votre système.

- fixTCPhang.tgz

Les fichiers dans les archives sont :

- coffre/fixTCPhang.pl
- mibs/CISCO-SMI.my

- mibs/CISCO-TCP-MIB.my

Pour extraire le script et les modules MIB, utilisez un utilitaire tel que le gzip et le goudron sur les systèmes d'exploitation comme une UNIX. Par exemple, pour extraire les fichiers à **/tmp** supposant que le fichier d'archivage est placé dans **/tmp** :

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

Note: Vous pouvez devoir éditer la première ligne du script pour spécifier l'emplacement du Perl.

Employez le winzip ou d'autres utilitaires sur des systèmes d'exploitation de Microsoft Windows pour extraire les fichiers. Si vous extrayez les fichiers à **c:\tmp** alors que vous ne devez pas spécifier - option m quand vous exécutez le script.

Appelez les fichiers avec cette commande :

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

Pour le chaque les connexions TCP arrêtées trouvées te voient une ligne comme cette sortie :

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

Pendant que la chaîne de caractères de la communauté en lecture-écriture était fournie et - l'option f a été spécifiée, le script a effacé la connexion. Notez la déclaration `EFFACÉE à l'`extrémité de la sortie.

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

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

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

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

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

Il y a plusieurs autres options que vous pouvez utiliser 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 si elles ne sont pas dans **/tmp/mibs**. Vous pouvez également visualiser 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