Cómo detectar y borrar conexiones TCP bloqueadas mediante SNMP

Contenido

Introducción

Este documento describe cómo utilizar el protocolo simple de administración de red (SNMP) para detectar y borrar conexiones TCP bloqueadas en un dispositivo Cisco IOS. El documento también explica los objetos SNMP que se utilizan para este propósito.

La sección titulada PERL Script to Detect and Clear Hung TCP Connections, proporciona un enlace a un script PERL que implementa estas instrucciones.

Prerequisites

Requirements

Quienes lean este documento deben tener conocimiento de los siguientes temas:

- Entender cómo ver la información de conexión TCP en los dispositivos Cisco
- Uso general de los comandos SNMP **walk**, **get**, **get-next** y **set**
- Entienda cómo configurar SNMP en un dispositivo Cisco

Componentes Utilizados

Este documento se aplica a los routers y switches de Cisco que ejecutan el software IOS que soportan los módulos TCP-MIB y CISCO-TCP-MIB.

Nota: El módulo CISCO-TCP-MIB no se carga de forma predeterminada en NET-SNMP. Si el módulo MIB no se carga en su sistema, debe utilizar el OID para hacer referencia a un objeto en lugar de su nombre.

La información en este documento se basa en todas las versiones de software y hardware del IOS.

La información se basa en esta versión de NET-SNMP:

- NET-SNMP version 5.1.2 available at http://www.net-snmp.org/

La secuencia de comandos PERL se probó con las versiones PERL:

- 5.005_03 en FreeBSD
- 5.8.0 en Solaris 5.8
- 5.005_02 — enviado como parte de CiscoWorks SNMS en Microsoft Windows 2000
- ActivePerl 5.8.4 en Microsoft Windows 2000, disponible en 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.

Convenciones

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

Antecedentes

Detalles de los objetos MIB — Incluye identificadores de objetos (OID)

Estos son los objetos que se utilizan:

Desde el módulo CISCO-TCP-MIB:

- ciscoTcpConnInBytes, OID .1.3.6.1.4.1.9.9.6.1.1.1.1El número de bytes introducidos en esta conexión.
- ciscoTcpConnInPkts, OID 1.3.6.1.4.1.9.9.6.1.1.1.2La cantidad de paquetes introducidos en esta conexión.
- CiscoTcpConnOutBytes, OID .1.3.6.1.4.1.9.9.6.1.1.1.3El número de bytes de salida en esta conexión
- ciscoTcpConnOutPkts, OID .1.3.6.1.4.1.9.9.6.1.1.1.4El número de paquetes que se producen en esta conexión.
- ciscoTcpConnRetransPkts, OID .1.3.6.1.4.1.9.9.6.1.1.1.7El número de paquetes retransmitidos en esta conexión.
- ciscoTcpConnRto, OID .1.3.6.1.4.1.9.9.6.1.1.1.9El valor de tiempo de espera de retransmisión para esta conexión.

Desde el módulo TCP-MIB:

- tcpConnState, OID .1.3.6.1.2.1.6.13.1.1El estado de esta conexión.

Hay más detalles sobre estos objetos en Información Detallada de Objeto MIB.

Utilice SNMP para detectar si se bloquea una conexión TCP

Summary

Estos pasos le ayudan a determinar si una conexión TCP se bloquea:

1. Para determinar si los objetos ciscoTcpConnRetransPkts y ciscoTcpConnRto son compatibles con el dispositivo, realice una operación get-next SNMP en ciscoTcpConnRto y verifique si se devuelve algún objeto.**Nota:** Solo es necesario comprobar un objeto porque se agregó soporte para ambos al mismo tiempo.**Nota:** No todos los dispositivos Cisco admiten los dos últimos objetos (ciscoTcpConnRetransPkts y ciscoTcpConnRto), pero su uso puede aumentar la precisión de la detección.Si se admiten los objetos ciscoTcpConnRetransPkts y ciscoTcpConnRto, vaya al Paso 2.Si los objetos ciscoTcpConnRetransPkts y ciscoTcpConnRto no se admiten, vaya al Paso 3.
2. Se admiten todos los objetos. Para cada conexión TCP, verifique lo siguiente:ciscoTcpConnOutBytes es 0.ciscoTcpConnOutPkts es 0.ciscoTcpConnRetransPkts es mayor que 0.ciscoTcpConnRto es mayor que 20 000.**Nota:** Los 20 000 se pueden reducir para acelerar la detección. Rto tardará aproximadamente un minuto en llegar a 20.000 una vez que se cuelga la conexión. Sin embargo, los valores más pequeños pueden reducir la precisión del resultado.Si todos los anteriores son verdaderos, esta conexión TCP se bloquea y se puede borrar. Proceda a Utilizar SNMP para Borrar una Conexión TCP que Cuelga.
3. Sólo se admiten los primeros cuatro objetos. Para cada conexión TCP, verifique lo siguiente:ciscoTcpConnInBytes es mayor que 0.ciscoTcpConnInPkts es 0.ciscoTcpConnOutBytes es 0.ciscoTcpConnOutPkts es 0.Espere unos segundos y **consiga** los objetos de nuevo para verificar que no se trataba de una conexión TCP en el proceso de establecimiento.**Nota:** Las dos primeras comprobaciones (un número positivo de bytes de entrada pero sin paquetes de entrada) pueden parecer extrañas, pero se verificaron con numerosos dispositivos y versiones de IOS.**Nota:** Es posible que las versiones IOS que admiten los seis objetos no muestren este comportamiento y, por lo tanto, la prueba en el Paso 2 no incluya estas dos primeras pruebas.Si todos los objetos cumplen las pruebas ambas veces, esta conexión TCP se bloquea y se puede borrar. Proceda a Utilizar SNMP para Borrar una Conexión TCP que Cuelga.

Step-by-Step Instructions

Los valores de este ejemplo son:

- Nombre de host del dispositivo a = nms-7206a (admite todos los objetos)
- Nombre de host del dispositivo b = nms-1605 (sólo admite los primeros cuatro objetos)
- Comunidad de lectura = pública
- Comunidad de escritura = privada

Reemplace las cadenas de comunidad y el nombre de host en estos comandos:

1. Determine si este dispositivo admite los objetos ciscoTcpConnRetransPkts y ciscoTcpConnRto:Realice una operación **SNMP get-next** en ciscoTcpConnRto:
```
snmpgetnext -c public nms-7206a ciscoTcpConnRto
```

   Si los objetos **son** admitidos, verá una respuesta como esta:
```
CISCO-TCP-MIB::ciscoTcpConnRto.14.32.100.75.2065.172.18.86.111.23092 =
     INTEGER: 303 milliseconds
```

   Nota: El índice utilizado para estos objetos, en este caso `14.32.100.75.2065.172.18.86.111.23092`, es una concatenación de la dirección IP local—14.32.10 0.75, el número de puerto TCP local—2065, la dirección IP remota—172.18.86.111 y el número de puerto TCP remoto—23092.La devolución es para ciscoTcpConnRto. Continúe en el paso 2.Si los objetos **no son** compatibles, verá una respuesta como esta:
```
snmpgetnext -c public nms-1605 ciscoTcpConnRto
CISCO-FLASH-MIB::ciscoFlashDevicesSupported.0 = INTEGER: 1
```

   El valor devuelto **no** corresponde al objeto ciscoTcpConnRto. El objeto exacto devuelto no es importante. Proceda al Paso 3.
2. **Obtenga** información sobre cada conexión TCP para los dispositivos que admiten los seis objetos en la tabla de conexión TCP de Cisco.Realice una operación SNMP get-next en ciscoTcpConnOutBytes, ciscoTcpConnOutPkts, ciscoTcpConnRetransPkts y ciscoTcpConnRto:
```
snmpgetnext -c public nms-7206a ciscoTcpConnOutBytes
                       ciscoTcpConnOutPkts
                       ciscoTcpConnRetransPkts
                       ciscoTcpConnRto
```

   Puede ver una respuesta como esta:
```
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
```

   Verifique lo siguiente:ciscoTcpConnOutBytes es 0.ciscoTcpConnOutPkts es 0.ciscoTcpConnRetransPkts es mayor que 0.ciscoTcpConnRto es mayor que 20 000.**Nota:** Los 20 000 se pueden reducir para acelerar la detección. Rto tardará aproximadamente un minuto en llegar a 20.000 una vez que se cuelga la conexión. Sin embargo, los valores más pequeños pueden reducir la precisión del resultado.Si todo esto es cierto, esta conexión TCP se bloquea y se puede borrar. Proceda a Utilizar SNMP para Borrar una Conexión TCP que Cuelga.Continúe **recorriendo** la tabla de conexión TCP. Para hacer esto, realice una operación **get-next** SNMP repetidamente mientras busca conexiones bloqueadas, usando los objetos devueltos como estos:
```
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
```

   Compruebe cada entrada utilizando la prueba anterior hasta que la operación **get-next**

devuelva objetos de esta manera:

```
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
```

Ya ha analizado todas las conexiones TCP en este dispositivo y ya ha terminado.

3. **Obtenga** información sobre cada conexión TCP para dispositivos que sólo admiten los primeros cuatro objetos en la tabla de conexión TCP de Cisco.Realice una operación SNMP **get-next** en ciscoTcpConnInBytes, ciscoTcpConnInPkts ciscoTcpConnOutBytes y ciscoTcpConnOutPkts:

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

Puede ver una respuesta como esta:

```
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
```

Compruebe si son ciertas:ciscoTcpConnInBytes es mayor que 0.ciscoTcpConnInPkts es 0.ciscoTcpConnOutBytes es 0.ciscoTcpConnOutPkts es 0.Espere unos segundos y **obtenga** los objetos de nuevo. Verifique que no haya sido una conexión TCP en el proceso de ser establecida.Si todos los anteriores **son** verdaderos, esta conexión TCP se bloquea y se puede borrar. Proceda a Utilizar SNMP para Borrar una Conexión TCP que Cuelga.Continúe **recorriendo** la tabla de conexión TCP. Para hacer esto, realice una operación **get-next** SNMP repetidamente mientras busca conexiones bloqueadas, usando los objetos devueltos como estos:

```
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
```

Compruebe cada entrada utilizando la prueba anterior hasta que la operación **get-next** devuelva objetos de esta manera:

```
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
```

Ya ha analizado todas las conexiones TCP en este dispositivo y ya ha terminado.

Utilice SNMP para borrar una conexión TCP que se bloquea

Step-by-Step Instructions

Puede utilizar SNMP para borrar una conexión TCP bloqueada. El comando SNMP es equivalente al comando clear tcp local <*local_ip*> <*local_port*> ***remote*** <*remote_ip*> <*remote_port*>

. El objeto que se utiliza para borrar una línea es **tcpConnState**.

Para borrar una conexión TCP bloqueada con SNMP, ejecute este comando:

```
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)
```

Nota: El índice utilizado para estos objetos, en este caso `14.32.100.75.2065.172.18.86.111.23092`, es una concatenación de la dirección IP local—14.32.10 0.75, el número de puerto TCP local—2065, la dirección IP remota—172.18.86.111 y el número de puerto TCP remoto—23092.

Nota: Debe utilizar el índice exacto que determinó que estaba colgado en Usar SNMP para detectar si una conexión TCP se bloquea. Tenga en cuenta que este comando desconecta una conexión TCP sin previo aviso.

Información detallada del objeto 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 }
```

Guión PERL para detectar y borrar conexiones TCP bloqueadas

Este enlace proporciona un archivo de archivo con un script PERL y los módulos MIB necesarios. Haga clic con el botón derecho del ratón en el enlace y guarde el archivo en el sistema.

- fixTCPhang.tgz

Los archivos del archivo son:

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

- mibs/CISCO-TCP-MIB.my

Para extraer el script y los módulos MIB, utilice una utilidad como gzip y tar en sistemas operativos similares a UNIX. Por ejemplo, para extraer los archivos a **/tmp** suponiendo que el archivo de archivo se coloca en **/tmp**:

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

Nota: Es posible que deba editar la primera línea del script para especificar la ubicación de PERL.

Utilice winzip u otras utilidades en los sistemas operativos Microsoft Windows para extraer los archivos. Si extrae los archivos a **c:\tmp**, no tendrá que especificar la opción -m cuando ejecute el script.

Invoque los archivos con este comando:

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

Para cada conexión TCP bloqueada encontrada, verá una línea como esta salida:

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

Cuando se proporcionó la cadena de comunidad de lectura y escritura y se especificó la opción -f, el script borró la conexión. Observe la instrucción `CLEARED` al final del resultado.

El script admite las versiones 1, 2c y 3 de SNMP. Si especifica SNMP versión 3, debe especificar toda la información de autenticación en el argumento -v. Este es un ejemplo del uso de SNMP v3:

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

Los comandos IOS para configurar SNMP v3 para el ejemplo anterior son:

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

Nota: Parece haber un error en la versión para Windows de NET-SNMP utilizada en esta prueba. El error no permite que la autenticación SHA funcione correctamente.

Hay otras opciones que puede utilizar con este script. Algunas de las opciones del script incluyen dónde encontrar las utilidades de la línea de comandos NET-SNMP y dónde encontrar los módulos MIB si no están en **/tmp/mibs**. También puede ver este resumen de estas opciones:

```
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.
```

Información Relacionada

- Soporte Técnico - Cisco Systems