

Como detectar e conexões de TCP claramente penduradas usando o SNMP

Índice

[Introdução](#)

[Pré-requisitos](#)

[Requisitos](#)

[Componentes Utilizados](#)

[Convenções](#)

[Informações de Apoio](#)

[Detalhes dos objetos MIB — Inclui os identificadores de objeto \(os OID\)](#)

[Use o SNMP para detectar se uma conexão de TCP pendura](#)

[Resumo](#)

[Instruções passo a passo](#)

[Use o SNMP para cancelar uma conexão de TCP que pendure](#)

[Instruções passo a passo](#)

[Informação detalhada do objeto MIB](#)

[Script de Perl a detectar e conexões de TCP claramente penduradas](#)

[Informações Relacionadas](#)

[Introdução](#)

Este documento descreve como usar o Simple Network Management Protocol (SNMP) para detectar e conexões de TCP claramente penduradas em um dispositivo IOS Cisco. O documento igualmente explica o SNMP objeto que você se usa por esse motivo.

A seção autorizada, o [script de Perl para detectar e as conexões de TCP claramente penduradas](#), fornecem um link a um script de Perl que execute estas instruções.

[Pré-requisitos](#)

[Requisitos](#)

Os leitores deste documento devem estar cientes destes tópicos:

- Compreenda como ver a informação da conexão de TCP em dispositivos Cisco
- Uso geral da **caminhada** SNMP, **comandos get, get-next, e set**
- Compreenda como configurar o SNMP em um dispositivo Cisco

[Componentes Utilizados](#)

Este documento aplica-se aos roteadores Cisco e ao Switches que executam o IOS Software que apoia o [TCP-MIB](#) e os módulos [CISCO-TCP-MIB](#).

Nota: O módulo CISCO-TCP-MIB não é carregado à revelia no NET-SNMP. Se o módulo MIB não é carregado em seu sistema, você deve usar o OID para prover um objeto em vez de seu nome.

A informação neste documento é baseada em todo o IOS Software e versões de hardware.

A informação é baseada nesta versão do NET-SNMP:

- Versão 5.1.2 NET-SNMP disponível em <http://www.net-snmp.org/>

O script de Perl foi testado com versões PERL:

- 5.005_03 no FreeBSD
- 5.8.0 em Solaris 5.8
- 5.005_02 — enviado como parte dos CiscoWorks SNM no Microsoft Windows 2000
- ActivePerl 5.8.4 no Microsoft Windows 2000, disponível em <http://www.activestate.com/Products/ActivePerl/> .

As informações neste documento foram criadas a partir de dispositivos em um ambiente de laboratório específico. Todos os dispositivos utilizados neste documento foram iniciados com uma configuração (padrão) inicial. Se a sua rede estiver ativa, certifique-se de que entende o impacto potencial de qualquer comando.

[Convenções](#)

Para obter mais informações sobre convenções de documento, consulte as [Convenções de dicas técnicas Cisco](#).

[Informações de Apoio](#)

[Detalhes dos objetos MIB — Inclui os identificadores de objeto \(os OID\)](#)

Estes são os objetos que você usa:

Do módulo [CISCO-TCP-MIB](#):

- [ciscoTcpConnInBytes](#), OID .1.3.6.1.4.1.9.9.6.1.1.1.10 número de bytes entrado nesta conexão.
- [ciscoTcpConnInPkts](#), OID 1.3.6.1.4.1.9.9.6.1.1.1.20 número de pacotes entrados nesta conexão.
- [ciscoTcpConnOutBytes](#), OID .1.3.6.1.4.1.9.9.6.1.1.1.30 número de bytes output nesta conexão
- [ciscoTcpConnOutPkts](#), OID .1.3.6.1.4.1.9.9.6.1.1.1.40 número de pacotes output nesta conexão.
- [ciscoTcpConnRetransPkts](#), OID .1.3.6.1.4.1.9.9.6.1.1.1.70 número de pacotes retransmitidos nesta conexão.
- [ciscoTcpConnRto](#), OID .1.3.6.1.4.1.9.9.6.1.1.1.90 valor de timeout da retransmissão para esta conexão.

Do módulo [TCP-MIB](#):

- [tcpConnState](#), OID .1.3.6.1.2.1.6.13.1.10 estado para esta conexão.

Há mais detalhes nestes objetos em [informação detalhada do objeto MIB](#).

Use o SNMP para detectar se uma conexão de TCP pendura

Resumo

Estas etapas ajudam-no a determinar se uma conexão de TCP pendura:

1. A fim determinar se os [ciscoTcpConnRetransPkts](#) e os objetos do [ciscoTcpConnRto](#) são apoiados no dispositivo, execute um SNMP **obter o próximo a** operação no [ciscoTcpConnRto](#) e verifique se algum objeto é retornado. **Nota:** Você precisa somente de verificar um objeto porque o apoio para ambos eles foi adicionado ao mesmo tempo. **Nota:** Não todos os dispositivos Cisco apoiam os últimos dois objetos ([ciscoTcpConnRetransPkts](#) e [ciscoTcpConnRto](#)), mas seu uso pode aumentar a precisão da detecção. Se os [ciscoTcpConnRetransPkts](#) e os objetos do [ciscoTcpConnRto](#) são apoiados, continue a etapa 2. Se os [ciscoTcpConnRetransPkts](#) e os objetos do [ciscoTcpConnRto](#) não são apoiados, continue a etapa 3.
2. Todos os objetos são apoiados. Para cada conexão de TCP verifique estes: os [ciscoTcpConnOutBytes](#) são 0. os [ciscoTcpConnOutPkts](#) são 0. os [ciscoTcpConnRetransPkts](#) são maiores de 0. o [ciscoTcpConnRto](#) é maior de 20,000. **Nota:** Os 20,000 podem ser reduzidos para acelerar a detecção. Toma um minuto ou assim para que Rto alcance 20,000 uma vez que a conexão é pendurada. Contudo, os valores menores podem reduzir a precisão do resultado. Se todos os precedentes são verdadeiros, a seguir esta conexão de TCP está pendurada e pode ser cancelada. Continue [usar o SNMP para cancelar uma conexão de TCP que pendure](#).
3. Somente os primeiros quatro objetos são apoiados. Para cada conexão de TCP verifique estes: os [ciscoTcpConnInBytes](#) são maiores de 0. os [ciscoTcpConnInPkts](#) são 0. os [ciscoTcpConnOutBytes](#) são 0. os [ciscoTcpConnOutPkts](#) são 0. Espere alguns segundos e **consiga os** objetos outra vez verificar que não era uma conexão de TCP em processo do estabelecimento. **Nota:** As primeiras duas verificações (um número positivo de bytes da entrada mas de nenhuns pacotes de entrada) podem parecer estranhas, mas elas foram verificadas contra dispositivos e Versões do IOS numerosos. **Nota:** As Versões do IOS que apoiam todos os seis objetos não podem exibir este comportamento e, conseqüentemente, o teste em etapa 2 não incluem estes primeiros dois testes. Se toda a reunião dos objetos os testes ambas as vezes então esta conexão de TCP é pendurada e pode ser cancelada. Continue [usar o SNMP para cancelar uma conexão de TCP que pendure](#).

Instruções passo a passo

Os valores neste exemplo são:

- Hostname do dispositivo a = nms-7206a (apoia todos os objetos)
- Hostname do dispositivo b = nms-1605 (apoios somente os primeiros quatro objetos)
- A comunidade de leitura = público
- A comunidade de gravação = privado

Substitua os string de comunidade e o hostname nestes comandos:

1. Determine se este suportes do dispositivo os [ciscoTcpConnRetransPkts](#) e os objetos do [ciscoTcpConnRto](#):Execute um **SNMP obter o próximo a** operação no [ciscoTcpConnRto](#):
`snmpgetnext -c public nms-7206a ciscoTcpConnRto` Se os objetos são apoiados você vê uma resposta como este:
CISCO-TCP-MIB::ciscoTcpConnRto.14.32.100.75.2065.172.18.86.111.23092 =
INTEGER: 303 milliseconds

Nota: O deslocamento predeterminado usado para estes objetos, neste caso 14.32.100.75.2065.172.18.86.111.23092, é uma concatenação do endereço IP local — 14.32.100.75, o número de porta de TCP local — 2065, o endereço IP remoto — 172.18.86.111, e o número de porta de TCP remoto — 23092.O retorno é para o [ciscoTcpConnRto](#). Vá para o Passo 2.Se os objetos não são apoio, você vê uma resposta como este:

`snmpgetnext -c public nms-1605 ciscoTcpConnRto CISCO-FLASH-`
MIB::ciscoFlashDevicesSupported.0 = INTEGER: 1 O retorno não é para o objeto do [ciscoTcpConnRto](#). O objeto exato retornado não é importante. Continue a etapa 3.

2. **Obtenha a** informação sobre cada conexão TCP para dispositivos que apoia todos os seis objetos na tabela da conexão de TCP de Cisco.Execute um **SNMP obter o próximo a** operação em [ciscoTcpConnOutBytes](#), em [ciscoTcpConnOutPkts](#), em [ciscoTcpConnRetransPkts](#), e em [ciscoTcpConnRto](#):

`snmpgetnext -c public nms-7206a ciscoTcpConnOutBytes ciscoTcpConnOutPkts`
`ciscoTcpConnRetransPkts ciscoTcpConnRto` Você vê uma resposta como este:
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 estes:[os ciscoTcpConnOutBytes](#) são 0.[os ciscoTcpConnOutPkts](#) são 0.[os ciscoTcpConnRetransPkts](#) são maiores de 0.[o ciscoTcpConnRto](#) é maior de 20,000.**Nota:** Os 20,000 podem ser reduzidos para acelerar a detecção. Toma um minuto ou assim para que Rto alcance 20,000 uma vez que a conexão é pendurada. Contudo, os valores menores podem reduzir a precisão do resultado.Se toda a estes é verdadeira, a seguir esta conexão de TCP está pendurada e pode ser cancelada. Continue [usar o SNMP para cancelar uma conexão de TCP que pendure](#).Continue a **andar a** tabela da conexão de TCP. A fim fazer isto, execute um **SNMP obter o próximo a** operação repetidamente como você verifica para ver se há conexões penduradas, usando os objetos retornados tais como estes:

`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`Verifique cada entrada usando o teste anterior até que a operação **obter o próximo** retorne objetos desse modo:

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

Você andou agora todas as conexões de TCP neste dispositivo e você é feito.

3. **Obtenha a** informação sobre cada conexão TCP para dispositivos que apoia somente os primeiros quatro objetos na tabela da conexão de TCP de Cisco.Execute um **SNMP obter o próximo a** operação em [ciscoTcpConnInBytes](#), em [ciscoTcpConnOutBytes dos ciscoTcpConnInPkts](#), e em [ciscoTcpConnOutPkts](#):

`snmpgetnext -c public nms-1605 ciscoTcpConnInBytes ciscoTcpConnInPkts ciscoTcpConnOutBytes`
`ciscoTcpConnOutPkts` Você vê uma resposta como este:
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
```

Verifique para ver se estes são verdadeiros: [os ciscoTcpConnInBytes](#) são maiores de 0. [os ciscoTcpConnInPkts](#) são 0. [os ciscoTcpConnOutBytes](#) são 0. [os ciscoTcpConnOutPkts](#) são 0. Espere alguns segundos e **obtenha os** objetos outra vez. Verifique que não era uma conexão de TCP em processo do estabelecimento. Se todos os acima **são** verdadeiros, a seguir esta conexão de TCP está pendurada e pode ser cancelada. Continue [usar o SNMP para cancelar uma conexão de TCP que pendure](#). Continue a **andar a** tabela da conexão de TCP. A fim fazer isto, execute um SNMP **obter o próximo a** operação repetidamente como você verifica para ver se há conexões penduradas, usando os objetos retornados tais como estes:

```
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
```

Verifique cada entrada usando o teste anterior até que a operação **obter o próximo** retorne objetos desse modo: 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

Você andou agora todas as conexões de TCP neste dispositivo e você é feito.

[Use o SNMP para cancelar uma conexão de TCP que pendure](#)

[Instruções passo a passo](#)

Você pode usar o SNMP para cancelar uma conexão de TCP pendurada. O comando SNMP é equivalente ao **comando clear tcp local <local_ip> <local_port> remote <remote_ip> <remote_port>**. O objeto que você se usa para cancelar uma linha é **tcpConnState**.

A fim cancelar uma conexão de TCP pendurada com SNMP, emita 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: O deslocamento predeterminado usado para estes objetos, neste caso 14.32.100.75.2065.172.18.86.111.23092, é uma concatenação do endereço IP local — 14.32.100.75, o número de porta de TCP local — 2065, o endereço IP remoto — 172.18.86.111, e o número de porta de TCP remoto — 23092.

Nota: Você deve usar o deslocamento predeterminado exato que você determinou esteve pendurado no [uso SNMP detectar se uma conexão de TCP pendura](#). Esteja ciente que este comando desliga uma conexão de TCP sem advertir.

[Informação detalhada do 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 }
```

[Script de Perl a detectar e conexões de TCP claramente penduradas](#)

Este link fornece um arquivo morto um script de Perl e os módulos MIB necessários. Clicar com o botão direito o link e salvar o arquivo a seu sistema.

- [fixTCPPhang.tgz](#)

Os arquivos no arquivo são:

- escaninho/fixTCPPhang.pl
- mibs/CISCO-SMI.my
- mibs/CISCO-TCP-MIB.my

Para extrair o script e os módulos MIB, use uma utilidade tal como o gzip e o alcatrão na Unix-como sistemas operacionais. Por exemplo, para extrair os arquivos a **/tmp** que supõe que o arquivo morto está colocado em **/tmp**:

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

Nota: Você pode precisar de editar a primeira linha do script para especificar o lugar do Perl.

Use o winzip ou as outras utilidades em sistemas operacionais de Microsoft Windows para extrair os arquivos. Se você extrai os arquivos a **c:\tmp** então que você não tem que especificar - opção m quando você executar o script.

Invoke os arquivos com este comando:

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

Para o cada as conexões de TCP penduradas encontradas lhe veem uma linha como esta saída:

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

Enquanto a série de comunidade de leitura/gravação foi fornecida e - a opção f foi especificada, o script cancelou a conexão. Note a indicação CANCELADA na extremidade da saída.

O script apoia as versões de SNMP 1, 2c, e 3. Se você especifica o SNMP Versão 3, você deve especificar toda a informação da autenticação - no argumento v. Este é um exemplo de usar SNMP v3:

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

Os comandos ios configurar SNMP v3 para o exemplo anterior são:

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

Nota: Parece estar um erro na versão do Windows do NET-SNMP usada nestes testes. O erro não permite que a autenticação SHA trabalhe corretamente.

Há diversas outras opções que você pode usar com este script. Algumas das opções do script incluem onde encontrar os utilitários comando-linha NET-SNMP e onde encontrar os módulos MIB se não estão em `/tmp/mibs`. Você pode igualmente ver este sumário daquelas opções:

```
fixTCPPhang.pl fixTCPPhang.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.
```

[Informações Relacionadas](#)

- [Suporte Técnico - Cisco Systems](#)