

# Hosted TCP-verbindingen detecteren en wissen met SNMP

## Inhoud

[Inleiding](#)

[Voorwaarden](#)

[Vereisten](#)

[Gebruikte componenten](#)

[Conventies](#)

[Achtergrondinformatie](#)

[Details van de MIB-objecten — Omvat Objectidentificatoren \(OID's\)](#)

[Gebruik SNMP om te ontdekken als een TCP-verbinding wordt geblokkeerd](#)

[Samenvatting](#)

[Stapsgewijze instructies](#)

[Gebruik SNMP om een TCP-verbinding te wissen die wordt verborgen](#)

[Stapsgewijze instructies](#)

[Gedetailleerde MIB-objecten](#)

[PERL Script om Hung TCP-verbindingen te detecteren en te wissen](#)

[Gerelateerde informatie](#)

## [Inleiding](#)

Dit document beschrijft hoe u Simple Network Management Protocol (SNMP) kunt gebruiken om opgehangen TCP-verbindingen op een Cisco IOS-apparaat te detecteren en verwijderen. Het document verklaart ook de SNMP voorwerpen die u voor dit doel gebruikt.

De sectie met de titel, [PERL Script om Hung TCP-verbindingen te detecteren en te wissen](#), biedt een link naar een PERL-script dat deze instructies implementeert.

## [Voorwaarden](#)

### [Vereisten](#)

Lezers van dit document zouden kennis moeten hebben van deze onderwerpen:

- Begrijp hoe u TCP-verbindingeninformatie over Cisco-apparaten kunt weergeven
- Algemeen gebruik van SNMP **lopen**, **krijgen**, **volgende**, en **instellen** opdrachten
- Begrijp hoe u SNMP op een Cisco-apparaat kunt configureren

### [Gebruikte componenten](#)

Dit document is van toepassing op Cisco routers en switches die IOS-software uitvoeren die de [TCP-MIB](#) en de [CISCO-TCP-MIB](#) modules ondersteunen.


**Opmerking:** de CISCO-TCP-MIB module is niet standaard geladen in NET-SNMP. Als de MIB module niet op uw systeem is geladen, moet u de OID gebruiken om een object te verwijzen in plaats van de naam.

De informatie in dit document is gebaseerd op alle IOS-software en hardwareversies.

De informatie is gebaseerd op deze versie van NET-SNMP:

- NET-SNMP versie 5.1.2 beschikbaar op <http://www.net-snmp.org/> 

Het PERL script is getest met PERL versies:

- 5.005\_03 op FreeBSD
- 5.8.0 over Solaris 5.8
- 5.005\_02 — verscheept als deel van CiscoWorks SNMS op Microsoft Windows 2000
- ActivePerl 5.8.4 op Microsoft Windows 2000, beschikbaar op <http://www.activestate.com/Products/ActivePerl/> 

De informatie in dit document is gebaseerd op de apparaten in een specifieke laboratoriumomgeving. Alle apparaten die in dit document worden beschreven, hadden een opgeschoonde (standaard)configuratie. Als uw netwerk live is, moet u de potentiële impact van elke opdracht begrijpen.

## [Conventies](#)

Raadpleeg [Cisco Technical Tips Conventions](#) (Conventies voor technische tips van Cisco) voor meer informatie over documentconventies.

## [Achtergrondinformatie](#)

### [Details van de MIB-objecten — Omvat Objectidentificatoren \(OID's\)](#)

Dit zijn de objecten die je gebruikt:

Van de [CISCO-TCP-MIB](#) module:

- [cisco TCPConnBytes](#), OID .1.3.6.1.4.1.9.9.6.1.1.1.1Het aantal bytes die worden ingevoerd op deze verbinding.
- [cisco TCP/NIn-pakketten](#), OID 1.3.6.1.4.1.9.9.6.1.1.1.2Het aantal pakketten dat op deze verbinding wordt ingevoerd.
- [ciscoTCPOutBytes](#), OID .1.3.6.1.4.1.9.9.6.1.1.3Het aantal bytes dat wordt uitgevoerd op deze verbinding
- [Cisco TCP-out-filters](#), OID 4.1.3.6.1.4.1.9.6.1.1.4Het aantal pakketten dat op deze verbinding wordt uitgevoerd.
- [cisco TCP/NRetrans-poorten](#), OID .1.3.6.1.4.1.9.6.1.1.7Het aantal pakketten dat op deze verbinding opnieuw wordt verzonden.
- [cisco TCPConnRto](#), OID .1.3.6.1.4.1.9.9.6.1.1.9De waarde voor het opnieuw verzenden van timeout voor deze verbinding.

Van de [TCP-MIB](#) module:

- [TcpConnState](#), OID .1.3.6.1.2.1.6.13.1.1 De status voor deze verbinding.

Er zijn meer details over deze objecten in [Gedetailleerde MIB Objectinformatie](#).

## [Gebruik SNMP om te ontdekken als een TCP-verbinding wordt geblokkeerd](#)

### [Samenvatting](#)

Deze stappen helpen u te bepalen of een TCP-verbinding bestaat:

1. Om te bepalen of de objecten [ciscoTCPConnRetransPackets](#) en [ciscoTCPConnRto](#) in het apparaat worden ondersteund, voert **SNMP** de **volgende** handeling op [ciscoTCPConnRto](#) uit en verifieert u of er voorwerpen worden teruggegeven. **Opmerking:** U hoeft slechts één object te controleren omdat ondersteuning voor beide tegelijk is toegevoegd. **Opmerking:** niet alle apparaten van Cisco ondersteunen de laatste twee objecten ([ciscoTCPConnRetransPackets](#) en [ciscoTCPConnRto](#)), maar het gebruik ervan kan de nauwkeurigheid van de detectie verhogen. Als de objecten [ciscoTCPConnRetrans](#) en [ciscoTCPConnRto](#) worden ondersteund, gaat u naar Stap 2. Als de objecten [ciscoTCPConnRetransPkts](#) en [ciscoTCPConnRto](#) **niet** worden ondersteund, gaat u naar Stap 3.
2. Alle objecten worden ondersteund. Controleer voor elke TCP-verbinding deze: [ciscoTCPConnOutBytes](#) is 0. [Cisco TCPConnOutPkts](#) is 0. [ciscoTCPConnRetransPkts](#) is groter dan 0. [cisco TCPConnRto](#) is groter dan 20.000. **Opmerking:** De 20.000 kan worden verminderd om de detectie te versnellen. Het duurt een minuut of zo voor Rto 20.000 is als de verbinding wordt opgehangen. Kleinere waarden kunnen echter de nauwkeurigheid van het resultaat verminderen. Als al het vorige juist is, dan wordt deze TCP verbinding opgehangen en kan worden gewist. Ga verder naar [Gebruik SNMP om een TCP verbinding te wissen die handig is](#).
3. Alleen de eerste vier objecten worden ondersteund. Controleer voor elke TCP-verbinding deze: [ciscoTCPConnInBytes](#) is groter dan 0. [Cisco TCPConnInPkts](#) is 0. [ciscoTCPConnOutBytes](#) is 0. [Cisco TCPConnOutPkts](#) is 0. Wacht een paar seconden en **laat** de objecten opnieuw controleren dat het geen TCP verbinding was in het proces van vestiging. **Opmerking:** de eerste twee controles (een positief aantal ingevoerde bytes maar geen ingevoerde pakketten) kunnen vreemd lijken, maar ze zijn geverifieerd op basis van talloze apparaten en IOS-versies. **Opmerking:** IOS versies die alle zes objecten ondersteunen kunnen dit gedrag niet tonen en daarom omvat de test in Stap 2 deze eerste twee testen niet. Als alle objecten de testen beide keer ontmoeten dan wordt deze TCP verbinding opgehangen en kan deze worden gewist. Ga verder naar [Gebruik SNMP om een TCP verbinding te wissen die handig is](#).

### [Stapsgewijze instructies](#)

De waarden in dit voorbeeld zijn:

- Apparaathostname a = nms-7206a (ondersteunt alle objecten)
- Apparaathostname b = nms-1605 (ondersteunt alleen de eerste vier objecten)

- Lees gemeenschap = publiek
- Schrijf gemeenschap = privé

Vervang de community strings en de hostname in deze opdrachten:

1. Bepaal of dit apparaat de objecten [ciscoTCPConnRetransPkts](#) en [ciscoTCPConnRto](#) ondersteunt: Voer een **SNMP gezochte** volgende handeling uit op [ciscoTCPConnRto](#):  

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

Als de objecten ondersteund worden, ziet u een dergelijke reactie:

```
CISCO-TCP-MIB::ciscoTcpConnRto.14.32.100.75.2065.172.18.86.111.23092 =  
    INTEGER: 303 milliseconds
```

**Opmerking:** De voor deze objecten gebruikte index, in dit geval

14.32.100.75.2065.172.18.86.111.23092, is een aaneenschakeling van het lokale IP-adres—14.32.10.75, het lokale TCP poortnummer-2065, het externe IP-adres-172.18.86.11 en het externe TCP-poortnummer-23092. Het rendement is voor

[ciscoTCPConnRto](#). Ga verder naar stap 2. Als de objecten niet ondersteund zijn, zie je een reactie als deze:

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

De terugkeer is niet voor het [ciscoTCPConnRto](#) voorwerp. Het exacte teruggegeven object is niet belangrijk. Ga verder naar stap 3.

2. **Ontvang** informatie over elke TCP verbinding voor apparaten die alle zes objecten in de Cisco TCP verbinding tabel ondersteunen. Voer een **SNMP-operatie** uit op [ciscoTCPOutBytes](#), [ciscoTCPConnOutPkts](#), [ciscoTCPConnRetransPkts](#) en [ciscoTCPConnRto](#):

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

Je ziet een reactie als deze:

```
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
```

Controleer deze: [ciscoTCPConnOutBytes](#) is 0. [Cisco TCPConnOutPkts](#) is

0. [ciscoTCPConnRetransPkts](#) is groter dan 0. [cisco TCPConnRto](#) is groter dan

20.000. **Opmerking:** De 20.000 kan worden verminderd om de detectie te versnellen. Het duurt een minuut of zo voor Rto 20.000 is als de verbinding wordt opgehangen. Kleinere waarden kunnen echter de nauwkeurigheid van het resultaat verminderen. Als al deze

getallen waar zijn, dan is deze TCP verbinding opgehangen en kan deze worden gewist. Ga verder naar [Gebruik SNMP om een TCP verbinding te wissen die handig is](#). Ga verder naar de TCP verbindingstabel. Om dit te doen, voer een SNMP **get-next** handeling herhaaldelijk uit als u op hung verbindingen controleert, met behulp van de geretourneerde objecten zoals deze:

```
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
```

Controleer elke ingang met de vorige test tot de **get-next** handeling op deze manier objecten teruggeeft:

```
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
```

Je hebt nu alle TCP connecties op dit apparaat gelopen en je bent klaar.

3. **Ontvang** informatie over elke TCP verbinding voor apparaten die slechts de eerste vier voorwerpen in de Cisco TCP verbinding tabel ondersteunen. Voer een SNMP-**get-next** handeling uit op [ciscoTCPConnInBytes](#), [ciscoTCPConnInPackets](#), [ciscoCPNetOutBytes](#) en [ciscoTCPNetOutPkts](#):

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

Je ziet een reactie als deze:

```
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
```

Controleer of deze waar zijn: [ciscoTCPConnInBytes](#) is groter dan 0. [Cisco TCPConnInPkts](#) is 0. [ciscoTCPConnOutBytes](#) is 0. [Cisco TCPConnOutPkts](#) is 0. Wacht een paar seconden en haal de objecten terug. Controleer dat het geen TCP verbinding was in het proces van vestiging. Als al het bovenstaande waar is, wordt deze TCP-verbinding verbroken. Ga verder naar [Gebruik SNMP om een TCP verbinding te wissen die handig is](#). Ga verder naar de TCP verbindingstabel. Om dit te doen, voer een SNMP **get-next** handeling herhaaldelijk uit als u op hung verbindingen controleert, met behulp van de geretourneerde objecten zoals deze:

```
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
```

Controleer elke ingang met de vorige test tot de **get-next** handeling op deze manier objecten teruggeeft:

```
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
```

Je hebt nu alle TCP connecties op dit apparaat gelopen en je bent klaar.

[\*\*Gebruik SNMP om een TCP-verbinding te wissen die wordt verborgen\*\*](#)

## Stapsgewijze instructies

U kunt SNMP gebruiken om een opgehangen TCP verbinding te ontruimen. De SNMP-opdracht is gelijk aan de heldere lokale `<local_ip> <local_port> externe <Remote_ip> <Remote_Port>-opdracht van TCP>`. Het object dat je gebruikt om een lijn te verwijderen is `tcpConnState`.

Om een opgevoerde TCP-verbinding met SNMP te wissen, geeft u deze opdracht uit:

```
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)
```

**Opmerking:** De voor deze objecten gebruikte index, in dit geval

14.32.100.75.2065.172.18.86.111.23092, is een aaneenschakeling van het lokale IP-adres—14.32.10 7.75, het lokale TCP poortnummer-2065, het externe IP-adres-172.18.86.11 en het externe TCP-poortnummer-23092.

**Opmerking:** U moet de exacte index gebruiken die u hebt ingesteld toen u [SNMP](#) gebruikte [om te detecteren of er een TCP-verbinding is](#). Let erop dat deze opdracht een TCP-verbinding zonder waarschuwing ontkoppelt.

## Gedetailleerde MIB-objecten

```
.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 }

```

## [PERL Script om Hung TCP-verbindingen te detecteren en te wissen](#)

Deze link verschaft een archiefbestand met een PERL-script en de benodigde MIB-modules. Klik met de rechtermuisknop op de koppeling en slaat het bestand op het systeem.

- [fixTCPPhang.tgz](#)

De bestanden in het archief zijn:

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

Om het script en de MIB modules te extraheren, gebruikt u een voorziening zoals gzip en tar op een UNIX-achtig besturingssysteem. Bijvoorbeeld, om de bestanden naar **/tmp** te extraheren ervan uitgaande dat het archiefbestand in **/tmp** geplaatst is:

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

**Opmerking:** Mogelijk moet u de eerste regel van het script bewerken om de locatie van PERL te specificeren.

Gebruik winzip of andere hulpprogramma's op Microsoft Windows-besturingssysteem om de bestanden te extraheren. Als u de bestanden naar **c:\tmp** extraheert, hoeft u niet de optie **-m** op te geven wanneer u het script gebruikt.

Noteer de bestanden met deze opdracht:

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

Voor elke opgehangen TCP-verbinding die u hebt gevonden, ziet u een lijn zoals deze uitvoer:

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

Aangezien de read-Writing community string werd gegeven en de **-f** optie werd gespecificeerd, klaagde het script de verbinding uit. Merk op dat de GEDRUKTE verklaring aan het einde van de uitvoer staat.

Het script ondersteunt SNMP versies 1, 2c en 3. Als u SNMP versie 3 specificeert, moet u alle verificatieinformatie in het **-v** argument specificeren. Dit is een voorbeeld van het gebruik van SNMP v3:

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

De IOS opdrachten om SNMP v3 te configureren voor het vorige voorbeeld zijn:

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

**Opmerking:** Er lijkt een bug in de Windows versie van NET-SNMP te zijn gebruikt bij deze testen. Het bug laat SHA-verificatie niet goed werken.



Er zijn verschillende andere opties die je met dit script kunt gebruiken. Sommige script opties omvatten waar de NET-SNMP opdrachtregel nutsbedrijven te vinden en waar de MIB modules gevonden kunnen worden als ze niet in `/tmp/mibs` zijn. U kunt deze samenvatting van deze opties ook bekijken:

#### **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.

## [Gerelateerde informatie](#)

- [Technische ondersteuning - Cisco-systemen](#)