

Come rilevare e cancellare le connessioni TCP bloccate con SNMP

Sommario

[Introduzione](#)

[Prerequisiti](#)

[Requisiti](#)

[Componenti usati](#)

[Convenzioni](#)

[Premesse](#)

[Dettagli degli oggetti MIB - include gli identificatori di oggetto \(OID, Object Identifier\)](#)

[Utilizzare il protocollo SNMP per rilevare eventuali blocchi di una connessione TCP](#)

[Riepilogo](#)

[Istruzioni dettagliate](#)

[Utilizzare il protocollo SNMP per cancellare una connessione TCP bloccata](#)

[Istruzioni dettagliate](#)

[Informazioni dettagliate sull'oggetto MIB](#)

[Script PERL per rilevare e cancellare le connessioni TCP bloccate](#)

[Informazioni correlate](#)

[Introduzione](#)

In questo documento viene descritto come usare il protocollo SNMP (Simple Network Management Protocol) per rilevare e cancellare le connessioni TCP bloccate su un dispositivo Cisco IOS. Nel documento vengono inoltre descritti gli oggetti SNMP utilizzati a tale scopo.

La sezione [Script PERL per rilevare e cancellare le connessioni TCP bloccate](#) fornisce un collegamento a uno script PERL che implementa queste istruzioni.

[Prerequisiti](#)

[Requisiti](#)

Questo documento è utile per conoscere i seguenti argomenti:

- Come visualizzare le informazioni sulla connessione TCP sui dispositivi Cisco
- Uso generale dei comandi **walk**, **get**, **get-next** e **set** di SNMP
- Comprendere come configurare il protocollo SNMP su un dispositivo Cisco


[Componenti usati](#)

Questo documento è relativo ai router e agli switch Cisco con software IOS che supportano i moduli [TCP-MIB](#) e [CISCO-TCP-MIB](#).


Nota: per impostazione predefinita, il modulo CISCO-TCP-MIB non viene caricato in NET-SNMP. Se il modulo MIB non è caricato nel sistema, è necessario utilizzare l'OID per fare riferimento a un oggetto anziché al nome.

Le informazioni di questo documento si basano su tutte le versioni software e hardware di IOS.

Le informazioni si basano su questa versione di NET-SNMP:

- NET-SNMP versione 5.1.2 disponibile all'indirizzo <http://www.net-snmp.org/> 

Lo script PERL è stato testato con le versioni PERL:

- 5.005_03 su FreeBSD
- 5.8.0 su Solaris 5.8
- 5.005_02 — fornito come parte di CiscoWorks SNMS su Microsoft Windows 2000
- ActivePerl 5.8.4 su Microsoft Windows 2000, disponibile all'indirizzo <http://www.activestate.com/Products/ActivePerl/> 

Le informazioni discusse in questo documento fanno riferimento a dispositivi usati in uno specifico ambiente di emulazione. Su tutti i dispositivi menzionati nel documento la configurazione è stata ripristinata ai valori predefiniti. Se la rete è operativa, valutare attentamente eventuali conseguenze derivanti dall'uso dei comandi.

[Convenzioni](#)

Per ulteriori informazioni sulle convenzioni usate, consultare il documento [Cisco sulle convenzioni nei suggerimenti tecnici](#).

[Premesse](#)

[Dettagli degli oggetti MIB - include gli identificatori di oggetto \(OID, Object Identifier\)](#)

Questi sono gli oggetti che utilizzate:

Dal modulo [CISCO-TCP-MIB](#):

- [ciscoTcpConnInBytes](#), OID .1.3.6.1.4.1.9.9.6.1.1.1.1 Numero di byte immessi in questa connessione.
- [ciscoTcpConnInPkts](#), OID 1.3.6.1.4.1.9.9.6.1.1.1.2 Numero di pacchetti immessi su questa connessione.
- [ciscoTcpConnOutBytes](#), OID .1.3.6.1.4.1.9.9.6.1.1.1.3 Numero di byte di output su questa connessione
- [ciscoTcpConnOutPkts](#), OID .1.3.6.1.4.1.9.9.6.1.1.1.4 Numero di pacchetti di output su questa connessione.
- [ciscoTcpConnRetransPkts](#), OID .1.3.6.1.4.1.9.9.6.1.1.1.7 Numero di pacchetti ritrasmessi su questa connessione.
- [ciscoTcpConnRto](#), OID .1.3.6.1.4.1.9.9.6.1.1.1.9 Valore di timeout di ritrasmissione per la connessione.

Dal modulo [TCP-MIB](#):

- [tcpConnState](#), OID .1.3.6.1.2.1.6.13.1.1 Stato della connessione.

Per ulteriori informazioni su questi oggetti, vedere [Informazioni dettagliate sugli oggetti MIB](#).

Utilizzare il protocollo SNMP per rilevare eventuali blocchi di una connessione TCP

Riepilogo

Questi passaggi consentono di determinare se una connessione TCP si blocca:

1. Per determinare se gli oggetti [ciscoTcpConnRetransPkts](#) e [ciscoTcpConnRto](#) sono supportati nel dispositivo, eseguire un'operazione **get-next** SNMP su [ciscoTcpConnRto](#) e verificare se sono stati restituiti oggetti. **Nota:** è necessario controllare un solo oggetto perché il supporto per entrambi è stato aggiunto contemporaneamente. **Nota:** non tutti i dispositivi Cisco supportano gli ultimi due oggetti ([ciscoTcpConnRetransPkts](#) e [ciscoTcpConnRto](#)), ma il loro utilizzo può aumentare l'accuratezza del rilevamento. Se gli oggetti [ciscoTcpConnRetransPkts](#) e [ciscoTcpConnRto](#) sono supportati, andare al passaggio 2. Se gli oggetti [ciscoTcpConnRetransPkts](#) e [ciscoTcpConnRto](#) non sono supportati, andare al passaggio 3.
2. Sono supportati tutti gli oggetti. Per ciascuna connessione TCP, verificare quanto segue: [ciscoTcpConnOutBytes](#) è 0. [ciscoTcpConnOutPkts](#) è 0. [ciscoTcpConnRetransPkts](#) è maggiore di 0. [ciscoTcpConnRto](#) è maggiore di 20.000. **Nota:** I 20.000 possono essere ridotti per accelerare il rilevamento. Ci vuole un minuto circa perché Rto raggiunga i 20.000 una volta che la connessione è bloccata. Tuttavia, valori più piccoli possono ridurre la precisione del risultato. Se tutte le precedenti condizioni sono vere, la connessione TCP viene interrotta e può essere cancellata. Continuare a [utilizzare SNMP per cancellare una connessione TCP che si blocca](#).
3. Sono supportati solo i primi quattro oggetti. Per ciascuna connessione TCP, verificare quanto segue: [ciscoTcpConnInBytes](#) è maggiore di 0. [ciscoTcpConnInPkts](#) è 0. [ciscoTcpConnOutBytes](#) è 0. [ciscoTcpConnOutPkts](#) è 0. Attendere alcuni secondi e **ottenere** di nuovo gli oggetti per verificare che non si tratti di una connessione TCP in fase di definizione. **Nota:** i primi due controlli (un numero positivo di byte di input ma nessun pacchetto di input) possono sembrare strani, ma sono stati verificati rispetto a numerosi dispositivi e versioni IOS. **Nota:** le versioni di IOS che supportano tutti e sei gli oggetti potrebbero non presentare questo comportamento e, pertanto, il test illustrato nel passo 2 non include questi primi due test. Se tutti gli oggetti soddisfano entrambi i test, la connessione TCP viene bloccata e può essere cancellata. Continuare a [utilizzare SNMP per cancellare una connessione TCP che si blocca](#).

Istruzioni dettagliate

I valori in questo esempio sono:

- Nome host dispositivo a = nms-7206a (supporta tutti gli oggetti)
- Nome host dispositivo b = nms-1605 (supporta solo i primi quattro oggetti)

- Community di lettura = public
- Write community = privato

Sostituire le stringhe della community e il nome host in questi comandi:

1. Determinare se il dispositivo supporta gli oggetti [ciscoTcpConnRetransPkts](#) e [ciscoTcpConnRto](#): Eseguire un'operazione **SNMP get-next** su [ciscoTcpConnRto](#):

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

Se gli oggetti sono supportati, viene visualizzata una risposta simile alla seguente:

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

Nota: l'indice utilizzato per questi oggetti, in questo caso

14.32.100.75.2065.172.18.86.111.23092, è una concatenazione dell'indirizzo IP locale—14.32.100.75, del numero di porta TCP locale—2065, dell'indirizzo IP remoto—172.18.86.11 TCP remoto numero—23092. La restituzione è per [ciscoTcpConnRto](#). Procedere al passo 2. Se gli oggetti **non** sono supportati, verrà visualizzata una risposta simile alla seguente:

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

Il valore restituito **non** è per l'oggetto [ciscoTcpConnRto](#). L'oggetto esatto restituito non è importante. Procedere al passo 3.

2. **Ottiene** informazioni su ciascuna connessione TCP per i dispositivi che supportano tutti e sei gli oggetti nella tabella delle connessioni TCP di Cisco. Eseguire un'operazione get-next di SNMP su [ciscoTcpConnOutBytes](#), [ciscoTcpConnOutPkts](#), [ciscoTcpConnRetransPkts](#) e [ciscoTcpConnRto](#):

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

Viene visualizzata una risposta simile alla seguente:

```
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
```

Verificare quanto segue: [ciscoTcpConnOutBytes](#) è 0. [ciscoTcpConnOutPkts](#) è 0. [ciscoTcpConnRetransPkts](#) è maggiore di 0. [ciscoTcpConnRto](#) è maggiore di 20.000. **Nota:** I 20.000 possono essere ridotti per accelerare il rilevamento. Ci vuole un minuto circa perché Rto raggiunga i 20.000 una volta che la connessione è bloccata. Tuttavia, valori più piccoli possono ridurre la precisione del risultato. Se tutte queste condizioni sono vere, la connessione TCP viene interrotta e può essere cancellata. Continuare a [utilizzare SNMP per cancellare una connessione TCP che si blocca](#). Continuare a **visualizzare** la tabella delle connessioni TCP. A tale scopo, eseguire ripetutamente un'operazione SNMP **get-next** durante il controllo delle connessioni bloccate, utilizzando gli oggetti restituiti come i seguenti:

```
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
```

Controllare ogni voce utilizzando il test precedente fino a quando l'operazione **get-next** restituisce gli oggetti nel modo seguente:

```
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
```

A questo punto sono state eseguite tutte le connessioni TCP su questo dispositivo.

3. **Ottiene** informazioni su ciascuna connessione TCP per i dispositivi che supportano solo i primi quattro oggetti nella tabella delle connessioni TCP di Cisco. Eseguire un'operazione **get-next** di SNMP su [ciscoTcpConnInBytes](#), [ciscoTcpConnInPkts](#), [ciscoTcpConnOutBytes](#) e [ciscoTcpConnOutPkts](#):

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

Viene visualizzata una risposta simile alla seguente:

```
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
```

Verificare se sono vere: [ciscoTcpConnInBytes](#) è maggiore di 0. [ciscoTcpConnInPkts](#) è 0. [ciscoTcpConnOutBytes](#) è 0. [ciscoTcpConnOutPkts](#) è 0. Attendere alcuni secondi e **recuperare** nuovamente gli oggetti. Verificare che non sia una connessione TCP in fase di connessione. Se tutte le condizioni precedenti **sono** vere, la connessione TCP viene interrotta e può essere cancellata. Continuare a [utilizzare SNMP per cancellare una connessione TCP che si blocca](#). Continuare a **visualizzare** la tabella delle connessioni TCP. A tale scopo, eseguire ripetutamente un'operazione SNMP **get-next** durante il controllo delle connessioni bloccate, utilizzando gli oggetti restituiti come i seguenti:

```
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
```

Controllare ogni voce utilizzando il test precedente fino a quando l'operazione **get-next** restituisce gli oggetti nel modo seguente:

```
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
```

A questo punto sono state eseguite tutte le connessioni TCP su questo dispositivo.

[Utilizzare il protocollo SNMP per cancellare una connessione](#)

TCP bloccata

Istruzioni dettagliate

È possibile utilizzare il protocollo SNMP per cancellare una connessione TCP bloccata. Il comando SNMP equivale al comando `clear tcp local <local_ip> <local_port> remote <remote_ip> <remote_port>`. L'oggetto utilizzato per cancellare una riga è `tcpConnState`.

Per cancellare una connessione TCP bloccata con SNMP, usare questo 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: l'indice utilizzato per questi oggetti, in questo caso `14.32.100.75.2065.172.18.86.111.23092`, è una concatenazione dell'indirizzo IP locale—14.32.100.75, del numero di porta TCP locale—2065, dell'indirizzo IP remoto—172.18.86.11 TCP remoto numero—23092.

Nota: È necessario utilizzare l'indice esatto che si è determinato essere bloccato in [Usa SNMP per rilevare se una connessione TCP si blocca](#). Tenere presente che questo comando disconnette una connessione TCP senza preavviso.

Informazioni dettagliate sull'oggetto 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 per rilevare e cancellare le connessioni TCP bloccate](#)

Questo collegamento fornisce un file di archivio con uno script PERL e i moduli MIB necessari. Fare clic con il pulsante destro del mouse sul collegamento e salvare il file nel sistema.

- [fixTCPPhang.tgz](#)

I file nell'archivio sono:

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

Per estrarre lo script e i moduli MIB, utilizzare un'utilità come gzip e tar su sistemi operativi simili a UNIX. Ad esempio, per estrarre i file in `/tmp` presupponendo che il file di archivio sia posizionato in `/tmp`:

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

Nota: potrebbe essere necessario modificare la prima riga dello script per specificare la posizione di PERL.

Utilizzare winzip o altre utilità nei sistemi operativi Microsoft Windows per estrarre i file. Se si estraggono i file in `c:\tmp`, non è necessario specificare l'opzione `-m` quando si esegue lo script.

Richiamare i file con questo comando:

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

Per ogni connessione TCP bloccata trovata viene visualizzata una riga simile a questo output:

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

Poiché è stata fornita la stringa della community di lettura/scrittura e è stata specificata l'opzione `-f`, lo script ha cancellato la connessione. Notare l'istruzione `CLEARED` alla fine dell'output.

Lo script supporta il protocollo SNMP versioni 1, 2c e 3. Se si specifica il protocollo SNMP versione 3, è necessario specificare tutte le informazioni di autenticazione nell'argomento `-v`. Questo è un esempio di utilizzo di SNMP v3:

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

I comandi IOS per configurare SNMP v3 per l'esempio precedente sono:

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

Nota: sembra esserci un bug nella versione Windows di NET-SNMP utilizzata in questo test. Il bug

non consente il corretto funzionamento dell'autenticazione SHA.

Con questo script è possibile utilizzare diverse altre opzioni. Alcune delle opzioni di script includono dove trovare le utilità della riga di comando NET-SNMP e dove trovare i moduli MIB se non sono in `/tmp/mibs`. È inoltre possibile visualizzare il riepilogo delle opzioni seguenti:

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.
```

[Informazioni correlate](#)

- [Supporto tecnico – Cisco Systems](#)