

Como Obter Entradas CAM Dinâmicas (Tabela CAM) para os Switches Catalyst Usando o SNMP

Índice

[Introdução](#)

[Pré-requisitos](#)

[Requisitos](#)

[Componentes Utilizados](#)

[Convenções](#)

[Background](#)

[O CISCO-VTP-MIB](#)

[A BRIDGE-MIB](#)

[Detalhes das variáveis MIB--Inclui os identificadores de objeto \(os OID\)](#)

[Recupere a informação do CAM dinâmica com SNMP](#)

[Instruções passo a passo](#)

[Verificar](#)

[Informações Relacionadas](#)

Introdução

Este documento descreve como coletar entradas do Dynamic Content-Addressable Memory (CAM) para switches Catalyst usando Simple Network Management Protocol (SNMP).

Pré-requisitos

Requisitos

Antes de usar a informação neste documento, assegure-se de que você encontre as seguintes condições prévias:

- Compreenda como obter VLAN de um Catalyst Switch através do SNMP.
- Compreenda como usar a [indexação de série de comunidade snmp](#).
- Uso geral de comandos **get** e **walk** SNMP.

Componentes Utilizados

Este documento aplica-se aos Catalyst Switches que executa o OS do catalizador ou o [®] regular do Catalyst IOS que apoiam o [BRIDGE-MIB](#). As informações neste documento são baseadas nas versões de software e hardware abaixo.

- Catalyst 3524XL executando CatIOS 12.0(5)WC5a
- Catalyst 2948G que executa Cactos 6.3(3)
- NET-SNMP disponível em <http://www.net-snmp.org/>

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 você estiver trabalhando em uma rede ativa, certifique-se de que entende o impacto potencial de qualquer comando antes de utilizá-lo.

Convenções

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

Background

O CISCO-VTP-MIB

A fim alcançar o MIBs que tem uma instância separada para cada VLAN, você deve primeiramente compreender como usar a [indexação de séries de comunidades](#). Em seguida, você precisa de conhecer os VLAN específicos que são ativos em um interruptor dado. [Do CISCO-VTP-MIB](#), você pode obter os Vlan ativo em um interruptor usando o objeto do [vtpVlanState](#). A razão usar o objeto do vtpVlanState, e não o vtpVlanName ou um outro objeto, são que você pode determinar em uma operação o número do índice e que um VLAN é operacional.

Mais informação é dada no exemplo abaixo.

A BRIDGE-MIB

[Do BRIDGE-MIB](#), que é extraído do [RFC 1493](#), você pode usar o [dot1dTpFdbAddress do dot1dTpFdbTable](#), onde o valor é igual a 3 ou *aprendido*, para determinar que endereços de controle de acesso de mídia (MAC) estão na tabela do forwarding no interruptor. Este valor é armazenado como um endereço MAC unicast para que a ponte tem a transmissão e/ou a informação de filtragem. Estes valores do MAC address apenas não significam muito e podem produzir muitos dados. Consequentemente, você precisa de contar o número de entradas e de armazenar esse valor de contagem, com base em um .1.3.6.1.2.1.17.4.3.1.3) igual [dot1dTpFdbStatus](#) (a *instruído* (um valor de 3).

Note: O BRIDGE-MIB usa a indexação de séries de comunidades para alcançar uma ocorrência particular do MIB como descrito na [indexação de série de comunidade snmp](#).

Tender dados do MAC address é valiosa para manter-se a par do número total de entradas de CAM (endereços MAC) aprendidas dinamicamente pelo interruptor. Esta monitoração ajuda a manter-se a par do nivelamento em sua rede, especialmente ao correlacionar ao número total dos LAN virtuais (VLAN) pelo interruptor. Por exemplo, se você tem um VLAN definido no interruptor e no você para ver 8,000 endereços MAC, você sabe que você tem 8,000 endereços MAC para um VLAN, que é extensivo para uma sub-rede.

Um objeto MIB relacionado do [BRIDGE-MIB](#) (RFC 1493) é [dot1dTpFdbStatus](#). Este MIB fornece o estado da entrada de endereço MAC.

As definições do valor são:

- **o outro (1):** Nenhum dos seguintes. Isto inclui os casos onde algum outro objeto MIB (não instância correspondente de `dot1dTpFdbPort`, nem uma entrada no `dot1dStaticTable`) estão sendo usados para determinar se e como os endereços do quadro ao valor da instância correspondente do `dot1dTpFdbAddress` estão sendo enviados.
- **(2) inválido:** Esta entrada é já não válida (por exemplo, era instruída mas envelheceu desde para fora), mas não foi nivelada ainda da tabela.
- **instruído (3):** O valor da instância correspondente do `dot1dTpFdbPort` era instruído, e está sendo usado.
- **auto (4):** O valor da instância correspondente do `dot1dTpFdbAddress` representa um dos endereços da ponte. A instância correspondente do `dot1dTpFdbPort` indica qual das portas da ponte tem este endereço.
- **mgmt (5):** O valor da instância correspondente do `dot1dTpFdbAddress` é igualmente o valor de um exemplo existente de `dot1dStaticAddress`.

Detalhes das variáveis MIB--Inclui os identificadores de objeto (os OID)

```
vtpVlanState OBJECT-TYPE
SYNTAX INTEGER { operational(1),
suspended(2),
mtuTooBigForDevice(3),
mtuTooBigForTrunk(4) }
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The state of this VLAN.
```

The state 'mtuTooBigForDevice' indicates that this device cannot participate in this VLAN because the VLAN's MTU is larger than the device can support.

The state 'mtuTooBigForTrunk' indicates that while this VLAN's MTU is supported by this device, it is too large for one or more of the device's trunk ports."

```
::= { vtpVlanEntry 2 }
```

```
.1.3.6.1.2.1.17.4.3.1.1
```

```
dot1dTpFdbAddress OBJECT-TYPE
-- FROM BRIDGE-MIB
-- TEXTUAL CONVENTION MacAddress
SYNTAX OCTET STRING (6)
MAX-ACCESS read-only
STATUS Mandatory
DESCRIPTION "A unicast MAC address for which the
bridge has forwarding and/or filtering information."
::= { iso(1) org(3) dod(6) internet(1) mgmt(2) mib-2(1)
dot1dBridge(17) dot1dTp(4) dot1dTpFdbTable(3) dot1dTpFdbEntry(1) 1 }
```

```
.1.3.6.1.2.1.17.4.3
```

```
dot1dTpFdbTable OBJECT-TYPE
-- FROM BRIDGE-MIB
DESCRIPTION "A table that contains information about unicast
entries for which the bridge has forwarding and/or filtering information.
This information is used by the transparent bridging function in
determining how to propagate a received frame."
::= { iso(1) org(3) dod(6) internet(1) mgmt(2) mib-2(1) dot1dBridge(17)
```

```
dot1dTp(4) 3 }
```

```
.1.3.6.1.2.1.17.5.1
```

```
dot1dStaticTable OBJECT-TYPE
```

```
-- FROM BRIDGE-MIB
```

```
DESCRIPTION      "A table containing filtering information configured
into the bridge by (local or network) management specifying the set of ports
to which frames received from specific ports and containing specific destination
addresses are allowed to be forwarded.  The value of zero in this table as the
port number from which frames with a specific destination address are received,
is used to specify all ports for which there is no specific entry in this table
for that particular destination address.  Entries are valid for unicast and for
group/broadcast addresses."
```

```
::= { iso(1) org(3) dod(6) internet(1) mgmt(2) mib-2(1) dot1dBridge(17)
dot1dStatic(5) 1 }
```

```
.1.3.6.1.2.1.17.4.3.1.2
```

```
dot1dTpFdbPort OBJECT-TYPE
```

```
-- FROM BRIDGE-MIB
```

```
SYNTAX            Integer
```

```
MAX-ACCESS        read-only
```

```
STATUS            Mandatory
```

```
DESCRIPTION      "Either the value "0", or the port number of the port
on which a frame having a source address equal to the value of the corresponding
instance of dot1dTpFdbAddress has been seen.  A value of "0" indicates that the
port number has not been learned, but that the bridge does have some
forwarding/filtering information about this address (that is, in the StaticTable).
```

```
Implementors are encouraged to assign the port value to
this object whenever it is learned, even for addresses for which the corresponding
value of dot1dTpFdbStatus is not learned(3)."
```

```
::= { iso(1) org(3) dod(6) internet(1) mgmt(2) mib-2(1) dot1dBridge(17) dot1dTp(4)
dot1dTpFdbTable(3) dot1dTpFdbEntry(1) 2 }
```

[Recupere a informação do CAM dinâmica com SNMP](#)

[Instruções passo a passo](#)

Siga estas etapas para obter a informação do CAM dinâmica com SNMP.

1. Recupere os VLAN. Use o `snmpwalk` no objeto do [vtpVlanState](#) (.1.3.6.1.4.1.9.9.46.1.3.1.1.2):

```
nms-server2:/home/ccarring> snmpwalk -c public 14.32.6.17 vtpVlanState
CISCO-VTP-MIB::vtpVlanState.1.1 = INTEGER: operational(1)
CISCO-VTP-MIB::vtpVlanState.1.2 = INTEGER: operational(1)
CISCO-VTP-MIB::vtpVlanState.1.6 = INTEGER: operational(1)
CISCO-VTP-MIB::vtpVlanState.1.7 = INTEGER: operational(1)
CISCO-VTP-MIB::vtpVlanState.1.8 = INTEGER: operational(1)
CISCO-VTP-MIB::vtpVlanState.1.11 = INTEGER: operational(1)
CISCO-VTP-MIB::vtpVlanState.1.12 = INTEGER: operational(1)
CISCO-VTP-MIB::vtpVlanState.1.14 = INTEGER: operational(1)
CISCO-VTP-MIB::vtpVlanState.1.18 = INTEGER: operational(1)
CISCO-VTP-MIB::vtpVlanState.1.19 = INTEGER: operational(1)
CISCO-VTP-MIB::vtpVlanState.1.20 = INTEGER: operational(1)
CISCO-VTP-MIB::vtpVlanState.1.21 = INTEGER: operational(1)
CISCO-VTP-MIB::vtpVlanState.1.41 = INTEGER: operational(1)
CISCO-VTP-MIB::vtpVlanState.1.42 = INTEGER: operational(1)
CISCO-VTP-MIB::vtpVlanState.1.43 = INTEGER: operational(1)
CISCO-VTP-MIB::vtpVlanState.1.44 = INTEGER: operational(1)
CISCO-VTP-MIB::vtpVlanState.1.100 = INTEGER: operational(1)
```

```
CISCO-VTP-MIB::vtpVlanState.1.101 = INTEGER: operational(1)
CISCO-VTP-MIB::vtpVlanState.1.123 = INTEGER: operational(1)
CISCO-VTP-MIB::vtpVlanState.1.401 = INTEGER: operational(1)
CISCO-VTP-MIB::vtpVlanState.1.1002 = INTEGER: operational(1)
CISCO-VTP-MIB::vtpVlanState.1.1003 = INTEGER: operational(1)
CISCO-VTP-MIB::vtpVlanState.1.1004 = INTEGER: operational(1)
CISCO-VTP-MIB::vtpVlanState.1.1005 = INTEGER: operational(1)
```

2. Para cada VLAN, obtenha o [dot1dTpFdbAddress](#) da tabela de endereços MAC (que usa a [indexação de séries de comunidades](#)) (.1.3.6.1.2.1.17.4.3.1.1). No exemplo, o VLAN2 não contém nenhuma entrada na tabela:

```
nms-server2:/home/ccarring> snmpwalk -c public@1 14.32.6.17 dot1dTpFdbAddress
.1.3.6.1.2.1.17.4.3.1.1.0.208.211.106.71.251 = Hex-STRING: 00 D0 D3 6A 47 FB
```

```
nms-server2:/home/ccarring> snmpwalk -c public@2 14.32.6.17 dot1dTpFdbAddress
```

```
nms-server2:/home/ccarring> snmpwalk -c public@6 14.32.6.17 dot1dTpFdbAddress
```

```
.1.3.6.1.2.1.17.4.3.1.1.0.2.185.144.76.102 = Hex-STRING: 00 02 B9 90 4C 66
```

```
.1.3.6.1.2.1.17.4.3.1.1.0.2.253.106.170.243 = Hex-STRING: 00 02 FD 6A AA F3
```

```
.1.3.6.1.2.1.17.4.3.1.1.0.16.13.56.16.0 = Hex-STRING: 00 10 0D 38 10 00
```

```
.1.3.6.1.2.1.17.4.3.1.1.0.96.84.144.248.0 = Hex-STRING: 00 60 54 90 F8 00
```

```
.1.3.6.1.2.1.17.4.3.1.1.0.208.2.214.120.10 = Hex-STRING: 00 D0 02 D6 78 0A
```

```
.1.3.6.1.2.1.17.4.3.1.1.0.208.211.54.162.60 = Hex-STRING: 00 D0 D3 36 A2 3C
```

```
.1.3.6.1.2.1.17.4.3.1.1.0.224.30.159.10.210 = Hex-STRING: 00 E0 1E 9F 0A D2
```

```
nms-server2:/home/ccarring> snmpwalk -c public@7 14.32.6.17 dot1dTpFdbAddress
```

```
.1.3.6.1.2.1.17.4.3.1.1.0.16.13.161.24.32 = Hex-STRING: 00 10 0D A1 18 20
```

... and so forth for each VLAN discovered in the first step.

3. Para cada VLAN, obtenha o número de porta de Bridge, o [dot1dTpFdbPort](#) (.1.3.6.1.2.1.17.4.3.1.2):

```
nms-server2:/home/ccarring> snmpwalk -c public@1 14.32.6.17 dot1dTpFdbPort
.1.3.6.1.2.1.17.4.3.1.2.0.208.211.106.71.251 = INTEGER: 113
```

```
nms-server2:/home/ccarring> snmpwalk -c public@2 14.32.6.17 dot1dTpFdbPort
```

```
nms-server2:/home/ccarring> snmpwalk -c public@6 14.32.6.17 dot1dTpFdbPort
```

```
.1.3.6.1.2.1.17.4.3.1.2.0.2.185.144.76.102 = INTEGER: 113
```

```
.1.3.6.1.2.1.17.4.3.1.2.0.2.253.106.170.243 = INTEGER: 113
```

```
.1.3.6.1.2.1.17.4.3.1.2.0.6.83.198.64.173 = INTEGER: 113
```

```
.1.3.6.1.2.1.17.4.3.1.2.0.16.13.56.16.0 = INTEGER: 113
```

```
.1.3.6.1.2.1.17.4.3.1.2.0.96.84.144.248.0 = INTEGER: 113
```

```
.1.3.6.1.2.1.17.4.3.1.2.0.208.2.214.120.10 = INTEGER: 113
```

```
.1.3.6.1.2.1.17.4.3.1.2.0.208.211.54.162.60 = INTEGER: 113
```

```
.1.3.6.1.2.1.17.4.3.1.2.0.224.30.159.10.210 = INTEGER: 65
```

```
nms-server2:/home/ccarring> snmpwalk -c public@7 14.32.6.17 dot1dTpFdbPort
```

```
.1.3.6.1.2.1.17.4.3.1.2.0.16.13.161.24.32 = INTEGER: 113
```

... and so forth for each VLAN discovered in the first step.

4. Obtenha a porta de Bridge ao mapeamento do [ifIndex](#) (1.3.6.1.2.1.2.2.1.1), o [dot1dBasePortIfIndex](#) (.1.3.6.1.2.1.17.1.4.1.2):

```
nms-server2:/home/ccarring> snmpwalk -c public@1 14.32.6.17 dot1dBasePortIfIndex
```

```
.1.3.6.1.2.1.17.1.4.1.2.68 = INTEGER: 12
```

```
.1.3.6.1.2.1.17.1.4.1.2.69 = INTEGER: 13
```

```
.1.3.6.1.2.1.17.1.4.1.2.70 = INTEGER: 14
```

```
.1.3.6.1.2.1.17.1.4.1.2.71 = INTEGER: 15
```

```
.1.3.6.1.2.1.17.1.4.1.2.72 = INTEGER: 16
```

```
.1.3.6.1.2.1.17.1.4.1.2.74 = INTEGER: 18
```

```
.1.3.6.1.2.1.17.1.4.1.2.76 = INTEGER: 20
```

```
.1.3.6.1.2.1.17.1.4.1.2.77 = INTEGER: 21
```

```
.1.3.6.1.2.1.17.1.4.1.2.78 = INTEGER: 22
```

```
.1.3.6.1.2.1.17.1.4.1.2.79 = INTEGER: 23
```

```
.1.3.6.1.2.1.17.1.4.1.2.80 = INTEGER: 24
```

```
.1.3.6.1.2.1.17.1.4.1.2.81 = INTEGER: 25
```

```
.1.3.6.1.2.1.17.1.4.1.2.82 = INTEGER: 26
```

```
.1.3.6.1.2.1.17.1.4.1.2.83 = INTEGER: 27
```

```
.1.3.6.1.2.1.17.1.4.1.2.84 = INTEGER: 28
.1.3.6.1.2.1.17.1.4.1.2.85 = INTEGER: 29
.1.3.6.1.2.1.17.1.4.1.2.86 = INTEGER: 30
.1.3.6.1.2.1.17.1.4.1.2.87 = INTEGER: 31
.1.3.6.1.2.1.17.1.4.1.2.88 = INTEGER: 32
.1.3.6.1.2.1.17.1.4.1.2.89 = INTEGER: 33
.1.3.6.1.2.1.17.1.4.1.2.90 = INTEGER: 34
.1.3.6.1.2.1.17.1.4.1.2.91 = INTEGER: 35
.1.3.6.1.2.1.17.1.4.1.2.92 = INTEGER: 36
.1.3.6.1.2.1.17.1.4.1.2.93 = INTEGER: 37
.1.3.6.1.2.1.17.1.4.1.2.94 = INTEGER: 38
.1.3.6.1.2.1.17.1.4.1.2.95 = INTEGER: 39
.1.3.6.1.2.1.17.1.4.1.2.96 = INTEGER: 40
.1.3.6.1.2.1.17.1.4.1.2.98 = INTEGER: 42
.1.3.6.1.2.1.17.1.4.1.2.99 = INTEGER: 43
.1.3.6.1.2.1.17.1.4.1.2.100 = INTEGER: 44
.1.3.6.1.2.1.17.1.4.1.2.101 = INTEGER: 45
.1.3.6.1.2.1.17.1.4.1.2.102 = INTEGER: 46
.1.3.6.1.2.1.17.1.4.1.2.103 = INTEGER: 47
.1.3.6.1.2.1.17.1.4.1.2.104 = INTEGER: 48
.1.3.6.1.2.1.17.1.4.1.2.105 = INTEGER: 49
.1.3.6.1.2.1.17.1.4.1.2.106 = INTEGER: 50
.1.3.6.1.2.1.17.1.4.1.2.107 = INTEGER: 51
.1.3.6.1.2.1.17.1.4.1.2.108 = INTEGER: 52
.1.3.6.1.2.1.17.1.4.1.2.109 = INTEGER: 53
.1.3.6.1.2.1.17.1.4.1.2.110 = INTEGER: 54
.1.3.6.1.2.1.17.1.4.1.2.111 = INTEGER: 55
.1.3.6.1.2.1.17.1.4.1.2.112 = INTEGER: 56
.1.3.6.1.2.1.17.1.4.1.2.113 = INTEGER: 57
.1.3.6.1.2.1.17.1.4.1.2.114 = INTEGER: 58
```

... and so forth for each VLAN discovered in the first step.

5. Anda o [ifName](#) (.1.3.6.1.2.1.31.1.1.1.1) de modo que o valor do ifIndex obtido em etapa 4 possa ser correlated com um nome de porta adequada:

```
nms-server2:/home/ccarring> snmpwalk -c public@14.32.6.17 dot1dBasePortIfIndex
.1.3.6.1.2.1.17.1.4.1.2.68 = INTEGER: 12
.1.3.6.1.2.1.17.1.4.1.2.69 = INTEGER: 13
.1.3.6.1.2.1.17.1.4.1.2.70 = INTEGER: 14
.1.3.6.1.2.1.17.1.4.1.2.71 = INTEGER: 15
.1.3.6.1.2.1.17.1.4.1.2.72 = INTEGER: 16
.1.3.6.1.2.1.17.1.4.1.2.74 = INTEGER: 18
.1.3.6.1.2.1.17.1.4.1.2.76 = INTEGER: 20
.1.3.6.1.2.1.17.1.4.1.2.77 = INTEGER: 21
.1.3.6.1.2.1.17.1.4.1.2.78 = INTEGER: 22
.1.3.6.1.2.1.17.1.4.1.2.79 = INTEGER: 23
.1.3.6.1.2.1.17.1.4.1.2.80 = INTEGER: 24
.1.3.6.1.2.1.17.1.4.1.2.81 = INTEGER: 25
.1.3.6.1.2.1.17.1.4.1.2.82 = INTEGER: 26
.1.3.6.1.2.1.17.1.4.1.2.83 = INTEGER: 27
.1.3.6.1.2.1.17.1.4.1.2.84 = INTEGER: 28
.1.3.6.1.2.1.17.1.4.1.2.85 = INTEGER: 29
.1.3.6.1.2.1.17.1.4.1.2.86 = INTEGER: 30
.1.3.6.1.2.1.17.1.4.1.2.87 = INTEGER: 31
.1.3.6.1.2.1.17.1.4.1.2.88 = INTEGER: 32
.1.3.6.1.2.1.17.1.4.1.2.89 = INTEGER: 33
.1.3.6.1.2.1.17.1.4.1.2.90 = INTEGER: 34
.1.3.6.1.2.1.17.1.4.1.2.91 = INTEGER: 35
.1.3.6.1.2.1.17.1.4.1.2.92 = INTEGER: 36
.1.3.6.1.2.1.17.1.4.1.2.93 = INTEGER: 37
.1.3.6.1.2.1.17.1.4.1.2.94 = INTEGER: 38
.1.3.6.1.2.1.17.1.4.1.2.95 = INTEGER: 39
.1.3.6.1.2.1.17.1.4.1.2.96 = INTEGER: 40
.1.3.6.1.2.1.17.1.4.1.2.98 = INTEGER: 42
```

```
.1.3.6.1.2.1.17.1.4.1.2.99 = INTEGER: 43
.1.3.6.1.2.1.17.1.4.1.2.100 = INTEGER: 44
.1.3.6.1.2.1.17.1.4.1.2.101 = INTEGER: 45
.1.3.6.1.2.1.17.1.4.1.2.102 = INTEGER: 46
.1.3.6.1.2.1.17.1.4.1.2.103 = INTEGER: 47
.1.3.6.1.2.1.17.1.4.1.2.104 = INTEGER: 48
.1.3.6.1.2.1.17.1.4.1.2.105 = INTEGER: 49
.1.3.6.1.2.1.17.1.4.1.2.106 = INTEGER: 50
.1.3.6.1.2.1.17.1.4.1.2.107 = INTEGER: 51
.1.3.6.1.2.1.17.1.4.1.2.108 = INTEGER: 52
.1.3.6.1.2.1.17.1.4.1.2.109 = INTEGER: 53
.1.3.6.1.2.1.17.1.4.1.2.110 = INTEGER: 54
.1.3.6.1.2.1.17.1.4.1.2.111 = INTEGER: 55
.1.3.6.1.2.1.17.1.4.1.2.112 = INTEGER: 56
.1.3.6.1.2.1.17.1.4.1.2.113 = INTEGER: 57
.1.3.6.1.2.1.17.1.4.1.2.114 = INTEGER: 58
```

... and so forth for each VLAN discovered in the first step.

Agora a informação de porta obtida pode ser usada, por exemplo: De etapa 2, há um MAC address: .1.3.6.1.2.1.17.4.3.1.1.0.208.211.106.71.251 = Encantar-CORDA: 00 D0 D3 6A 47 FB De etapa 3: .1.3.6.1.2.1.17.4.3.1.2.0.208.211.106.71.251 = INTEIRO: 113 Isto diz-lhe que este MAC address (00 D0 D3 6A 47 FB) é do número de porta de Bridge 113. De etapa 4, o número de porta de Bridge 113 tem um ifIndex número 57 .1.3.6.1.2.1.17.1.4.1.2.113 = INTEIRO: 57 Da etapa 5, o ifIndex 57 corresponde à porta 2/49 .1.3.6.1.2.1.31.1.1.1.1.57 = CORDA: 2/49 Compare isso com a saída do comando **show cam dynamic output** para switch Cactos, ou o comando **show mac output** para o Switches de CatIOS. Você vê um fósforo para 1 [ALL] 00-d0-d3-6a-47-fb 2/49.

Verificar

Esta seção fornece informações que você pode usar para confirmar se sua configuração está funcionando adequadamente.

1. Telnet a seu interruptor.
2. Da linha de comando, emita o comando apropriado: Dispositivos cactos: **Comando show cam** Dispositivos de CatIOS: [show mac](#)
3. Compare a saída com os resultados obtidos pelo procedimento especificado aqui.

```
nms-2948g> (enable) show cam dynamic
```

```
* = Static Entry. + = Permanent Entry. # = System Entry. R = Router Entry.
```

```
X = Port Security Entry $ = Dot1x Security Entry
```

VLAN	Dest MAC/Route Des	[CoS]	Destination Ports or VCs / [Protocol Type]
1	00-d0-d3-6a-47-fb		2/49 [ALL]
6	00-02-b9-90-4c-66		2/49 [ALL]
6	00-02-fd-6a-aa-f3		2/49 [ALL]
6	00-10-0d-38-10-00		2/49 [ALL]
6	00-60-54-90-f8-00		2/49 [ALL]
6	00-c0-1d-99-00-dc		2/49 [ALL]
6	00-d0-02-d6-78-0a		2/49 [ALL]
6	00-d0-d3-36-a2-3c		2/49 [ALL]
6	00-e0-1e-9f-0a-d2		2/1 [ALL]
7	00-10-0d-a1-18-20		2/49 [ALL]
8	00-10-0d-38-10-00		2/49 [ALL]
8	00-10-0d-a1-18-c0		2/49 [ALL]
14	00-d0-d3-36-a2-3c		2/49 [ALL]
18	00-00-0c-07-ac-12		2/49 [ALL]

```
18      00-10-0d-38-10-00      2/49 [ALL]
18      00-d0-d3-36-a2-3c      2/49 [ALL]
19      00-d0-02-d6-78-0a      2/49 [ALL]
41      00-d0-d3-36-a2-3c      2/49 [ALL]
42      00-d0-d3-36-a2-3c      2/49 [ALL]
100     00-04-de-a9-18-00      2/49 [ALL]
100     00-10-0d-38-10-00      2/49 [ALL]
100     00-10-7b-d9-07-60      2/49 [ALL]
100     00-90-27-86-76-e2      2/49 [ALL]
100     00-d0-d3-36-a2-3c      2/49 [ALL]
100     00-e0-1e-68-33-c7      2/49 [ALL]
101     00-d0-d3-36-a2-3c      2/49 [ALL]
Total Matching CAM Entries Displayed =26
nms-2948g> (enable)
```

[Informações Relacionadas](#)

- [Indexação de série de comunidade SNMP](#)
- [Suporte Técnico - Cisco Systems](#)