了解有关调试命令的重要信息

目录

简介

先决条件

- 要求
- 使用的组件

背景信息

警告

在调试之前

- 获得调试输出
  - 控制台端口
  - AUX 端口
  - VTY端口
  - 将信息记录到内部缓冲器
  - 将信息记录到UNIX Syslog服务器
- 其他预调试任务
- 停止调试

使用 debug ip packet 命令

有条件的被触发的调试

相关信息

简介

本文档介绍使用 debug 命令的一般准则，包括 Cisco IOS® 平台上可用的 debug ip packet 命令。

先决条件

要求

Cisco 建议您了解以下主题：

- 通过控制面板、aux和vty端口与路由器相连。
- Cisco IOS 常见配置问题.
- 解释 Cisco IOS 调试输出.

使用的组件

本文档不限于特定的软件和硬件版本。

本文档中的信息都是基于特定实验室环境中的设备编写的。本文档中使用的所有设备最初均采用原始（默认）配置。如果您的网络处于活动状态，请确保您了解所有命令的潜在影响。

背景信息

本文档提供在Cisco IOS®平台上使用debug命令的一般指南。其中还包括示例和条件调试概述。

debug特权EXEC命令提供有关网络事件、协议状态、数据包处理和常规网络活动的诊断信息。这些命令有助于在故障排除过程中确定特定问题的原因。

但是，debug命令可能会生成大量输出信息，并且可能会影响设备性能，特别是在已经处理高流量或高CPU使用率的路由器上。因此，请谨慎地运行debug命令，并且仅在需要排除故障时运行。



注意：本文档不介绍如何使用或解释特定的debug命令及其输出。有关单个debug命令的详细信息，请参阅相应的Cisco Debug命令参考文档。

警告

请谨慎运行debug命令。通常，在排除特定故障时，仅在技术支持代表的指导下使用这些命令。

启用调试可能会中断路由器运行，尤其是在网络负载较重时。如果启用日志记录，当控制台端口因日志消息而过载时，访问服务器可以间歇性地冻结。

在运行debug命令之前，请考虑该命令可以生成的输出量以及调试会话可以运行的时间。

例如，在具有一个基本速率接口的路由器上，debug isdn q931不太可能影响系统。但是，在具有完整E1配置的AS5800上运行相同的debug命令可以生成足够的输出使设备挂起或停止响应。

在调试之前，请通过运行show processes cpu命令检查CPU负载。在启用调试之前，请验证是否有足够的CPU容量可用，这就是方法。

例如，如果带有ATM接口的Cisco 7200路由器正在运行桥接，重新启动路由器可能会消耗大量CPU资源，具体取决于配置的子接口数量。对于每个虚电路(VC)，必须生成网桥协议数据单元(BPDU)数据包。在此关键时段启用调试会导致CPU利用率大幅提高，从而导致设备挂起或网络连接丢失。



注意：运行调试时，您通常不会看到路由器提示，特别是当调试非常密集时。但是，在大多数情况下，您可以运行no debug all或undebug all命令停止调试。

在调试之前

除了上述要点之外，请确保您了解调试对平台稳定性的影响。在启用任何debug命令之前，还必须考虑必须在路由器上连接哪个接口。

获得调试输出

路由器可显示各种接口的调试输出，其中包括控制面板、aux和vty端口。路由器还可将发送到内部缓冲器的日志消息记录到外部unix syslog服务器上。

控制台端口

如果在正常配置下连接到控制台，则无需额外工作。调试输出必须自动显示。但是，请确保已根据需要设置logging console级别，并且未使用no logging console命令禁用日志记录。



警告：对路由器的控制台端口进行过多调试会导致其被挂起。这是因为：在执行路由器其他功能之前，路由器对控制面板输出自动划分优先权。如果路由器正在处理到控制台端口的大型调试输出，则它可能会挂起。如果调试输出过多，请使用vty(telnet)端口或日志缓冲区获取调试。



注意：默认情况下，控制台端口上会启用日志记录。控制台端口始终处理debug输出，即使您使用其他端口或方法（例如aux、vty或缓冲区）捕获输出也是如此。Cisco建议在正常操作条件下，运行no logging console命令，该命令始终处于启用状态，并使用其他方法捕获调试。在必须使用控制台的情况下，临时重新打开日志控制台。

AUX 端口

如果通过辅助端口连接，请运行terminal monitorcommand。另请验证no logging on命令未在路由器上激活。



注意：如果使用aux端口监控路由器，请记住，当路由器重新启动时，aux端口不会显示启动顺序输出。连接到控制台端口以查看启动顺序。

VTY端口

如果通过辅助端口或 Telnet 进行连接，请键入 terminal monitor 命令。另请确认未使用no logging on命令。

将信息记录到内部缓冲器

默认的日志记录设备是控制台；除非另有指定，否则所有消息都会显示在控制台上。

要将消息记录到内部缓冲区，请运行logging buffered路由器配置命令。以下是该命令的完整语法：

<#root>

```
logging buffered
no logging buffered
```

日志记录缓冲命令将日志消息复制到内部缓冲区，而不是将其写入控制台。缓冲器具有循环特性，新消息将覆盖旧消息。

要显示缓冲区中记录的消息，请使用特权EXEC命令show logging。所显示的第1个消息是缓冲器中最旧的消息。您可以指定缓冲区大小和要记录的消息的严重性级别。



注意：在输入缓冲区大小之前，请确保机箱中有足够的可用内存。使用show proc mem命令查看可用内存。

通过 no logging buffered 命令，可禁用缓冲器，并将消息写入到控制面板中(缺省条件下)。

将信息记录到UNIX Syslog服务器

要将消息记录到syslog服务器主机，请运行logging router configuration命令。查看此命令的完整语法：

<#root>

logging

no logging

通过 logging 命令确定syslog服务器主机以接收日志消息。<ip-address> 参数是主机的IP地址。通过多次发出此命令，您可以构建接收日志记录消息的系统日志服务器列表。

no logging命令会从系统日志列表中删除具有指定地址的系统日志服务器。

其他预调试任务

1. 设置终端仿真器软件，使其将调试输出捕获到文件中。请参阅软件终端仿真程序文档。
2. 启用运行service timestamps命令的毫秒(msec)时间戳：

<#root>

router(config)#

service timestamps debug datetime msec

router(config)#

service timestamps log datetime msec

这些命令以MMM DD HH:MM:SS格式将时间戳添加到调试，根据系统时钟指示日期和时间。如果未设置系统时钟，日期和时间前面加星号(*)，表示日期和时间可能不正确。

通常建议配置毫秒时间戳，因为这样在查看调试输出时能提供高级别的清晰度。毫秒时间戳可以更好地指示各种调试事件的相对计时。但是，当控制台端口输出大量消息时，它们无法与事件的实际时间相关联。

例如，如果在包含200条VC的盒子上启用debug x25 all，并且输出已记录到缓冲区(运行no logging console和logging buffered命令)，则debug输出（缓冲区内）中显示的时间戳不能是数据包通过接口的确切时间。因此，不要使用msec时间标记来证明性能情况，但可使用这种时间标记来获得事件发生时间的相对信息。

停止调试

若要停止调试，则使用 no debug all 或 undebug all 命令。使用show debug命令验证调试是否已关闭。

请记住，命令no logging console和terminal no monitor只会阻止输出分别输出到控制台、aux或vty上。它不会停止调试并耗尽路由器资源。

使用 debug ip packet 命令

在传统Cisco IOS®路由器上，debug ip packet主要看到进程交换流量。通过快速交换或CEF转发的流量不会出现，除非将转发强制到进程交换路径中。但是，由于它为每个数据包生成一个输出，因此输出可能非常广泛，并导致路由器挂起。因此，请按照本节所述在最严格的控制下运行debug ip packet。

限制debug ip packet输出的最佳方法是创建链接到debug的访问列表。只有与访问列表条件匹配的数据包才能受到debug ip packet的制约。不必在所有接口上都应用此访问列表，但非常适用于调试操作。

在运行debugging ip packet之前，请注意路由器在默认情况下使用的是快速交换，或者可以使用CEF交换（如果配置为使用CEF交换）。这意味着，一旦采用这些技术，数据包就不会提供给处理器，调试过程也不会显示任何信息。要使此功能正常工作，必须在没有ip route-cache（用于单播数据包）或no ip mroute-cache（用于组播数据包）的路由器上禁用快速交换。 这必须应用到流量应流经的接口。通过 show ip route 命令来验证这种情况。



注意：在较新的平台上，转发通常由CEF或基于硬件的交换处理，因此禁用快速交换不再适用或推荐使用。因此，debug ip packet可能无法可靠地显示中转流量，而现代故障排除

通常依赖于平台特定的捕获或硬件工具。

有条件的被触发的调试

当启用条件触发调试功能时，路由器会为进出指定接口上的路由器的数据包生成调试消息；路由器不会为通过其他接口进出的数据包生成调试输出。

下面介绍条件调试的简单实施。请考虑以下场景：下面显示的路由器(trabol)有两个接口（serial 0和serial 3），均运行HDLC封装。

您可以运行normaldebug serial interface命令来观察所有接口上收到的HDLC keepalive。您可以在两个接口上观察keepalive:

<#root>

```
traxbol#

debug serial interface


Serial network interface debugging is on
traxbol#
*Mar  8 09:42:34.851:

Serial0: HDLC

 myseq 28, mineseen 28*, yourseen 41, line up

! -- HDLC keeplaive on interface Serial 0

*Mar  8 09:42:34.855:

Serial3: HDLC

 myseq 26, mineseen 26*, yourseen 27, line up

! -- HDLC keeplaive on interface Serial 3

*Mar  8 09:42:44.851: Serial0: HDLC myseq 29, mineseen 29*, yourseen 42, line up
*Mar  8 09:42:44.855: Serial3: HDLC myseq 27, mineseen 27*, yourseen 28, line up
```

为接口Serial 3启用条件调试，这意味着只显示接口Serial 3的调试。运行debug interface <interface_type interface_number >命令：

<#root>

```
traxbol#

debug interface serial 3


Condition 1 set
```

运行show debug condition命令以验证条件调试是否处于活动状态（接口serial 3的条件处于活动状态）：

<#root>

```
traxbol#

show debug condition



Condition 1: interface Se3 (1 flags triggered)
Flags: Se3
traxbol#
```

现在只显示接口serial 3的调试：

<#root>

```
*Mar  8 09:43:04.855:

Serial3: HDLC

 myseq 29, mineseen 29*, yourseen 30, line up
*Mar  8 09:43:14.855:

Serial3: HDLC

 myseq 30, mineseen 30*, yourseen 31, line up
```

运行undebug interface <interface_type interface_number>命令以删除条件调试。建议您在删除条件触发之前关闭调试（例如，使用 undebug all 命令）。通过这种办法，可避免删除条件时出现调试输出泛滥。

<#root>

```
traxbol#
```

```
undebug interface serial 3

This condition is the last interface condition set.
Removing all conditions can cause a flood of debugging
messages to result, unless specific debugging flags
are first removed.
Proceed with removal? [yes/no]:

y


Condition 1 has been removed
traxbol
```

您可以观察接口Serial 0和Serial 3的调试显示：

<#root>

```
*Mar  8 09:43:34.927:

Serial3: HDLC

 myseq 32, mineseen 32*, yourseen 33, line up
*Mar  8 09:43:44.923:

Serial0: HDLC

 myseq 35, mineseen 35*, yourseen 48, line up
```



警告：某些调试操作本身是有条件的。例如atm调试，使用ATM调试时，必须明确指定必须为其启用调试的接口，而不是在所有atm接口上启用调试并指定条件。

本部分介绍了将 ATM 数据包调试限制在一个子接口的正确方法：

<#root>

```
arielle-nrp2#

debug atm packet interface atm 0/0/0.1


!--- Note that you explicitly specify the sub-interface to be used for debugging

ATM packets debugging is on

Displaying packets on interface ATM0/0/0.1 only
```

```
arielle-nrp2#
*Dec 21 10:16:51.891: ATM0/0/0.1(O):
VCD:0x1 VPI:0x1 VCI:0x21 DM:0x100 SAP:AAAA CTL:03 OUI:0080C2 TYPE:0007
Length:0x278
*Dec 21 10:16:51.891: 0000 FFFF FFFF FFFF 0010 7BB9 BDC4 0800 4500 025C 01FE
0000 FF11 61C8 0A30
*Dec 21 10:16:51.891: 4B9B FFFF FFFF 0044 0043 0248 0000 0101 0600 0015 23B7
0000 8000 0000 0000
*Dec 21 10:16:51.891: 0000 0000 0000 0000 0000 0000 0010 7BB9 BDC3 0000 0000
0000 0000 0000 0000
*Dec 21 10:16:51.891: 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000
*Dec 21 10:16:51.891: 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000
*Dec 21 10:16:51.891: 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000
*Dec 21 10:16:51.891: 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000
*Dec 21 10:16:51.895: 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000
*Dec 21 10:16:51.895:
arielle-nrp2#
```

如果尝试在所有接口上启用atm debugging（使用应用条件），则路由器在拥有大量ATM子接口时可能会挂起。下面举例说明了atm 调试的不正确方法。

在本例中，您可以看到应用了一个条件，但也可以看到该条件没有任何影响。您仍然可以看到来自另一个接口的数据包。

在本实验场景中，您只有两个接口，流量非常少。如果接口数很高，则所有接口的调试输出也非常高，这可能会导致路由器挂起：

<#root>

```
arielle-nrp2#

show debugging condition



Condition 1: interface AT0/0/0.1

 (1 flags triggered)
Flags: AT0/0/0.1

! -- A condition for a specific interface.


arielle-nrp2#

debug atm packet
```

```
ATM packets debugging is on
Displaying all ATM packets
arielle-nrp2#
*Dec 21 10:22:06.727:
```

ATM0/0/0.2

```
(O):

! -- You see debugs from interface ATM0/0/0/.2, even though the condition
! -- specified ONLY AT0/0/0.1


  VCD:0x2 VPI:0x5 VCI:0x37 DM:0x100 SAP:AAAA CTL:03 OUI:0080C2
TYPE:000E Length:0x2F
*Dec 21 10:22:06.727: 0000 0000 0180 0000 107B B9BD C400 0000 0080
0000 107B B9BD C480 0800 0014
*Dec 21 10:22:06.727: 0002 000F 0000
*Dec 21 10:22:06.727: un a
*Dec 21 10:22:08.727: ATM0/0/0.2(O):
  VCD:0x2 VPI:0x5 VCI:0x37 DM:0x100 SAP:AAAA CTL:03 OUI:0080C2
TYPE:000E Length:0x2F
*Dec 21 10:22:08.727: 0000 0000 0180 0000 107B B9BD C400 0000 0080
0000 107B B9BD C480 0800 0014
*Dec 21 10:22:08.727: 0002 000F 0000
*Dec 21 10:22:08.727: ll
*Dec 21 10:22:10.727: ATM0/0/0.2(O):
  VCD:0x2 VPI:0x5 VCI:0x37 DM:0x100 SAP:AAAA CTL:03 OUI:0080C2
TYPE:000E Length:0x2F
*Dec 21 10:22:10.727: 0000 0000 0080 0000 107B B9BD C400 0000 0080
0000 107B B9BD C480 0800 0014
*Dec 21 10:22:10.727: 0002 000F 0000
*Dec 21 10:22:10.727:
*Dec 21 10:22:12.727: ATM0/0/0.2(O):
  VCD:0x2 VPI:0x5 VCI:0x37 DM:0x100 SAP:AAAA CTL:03 OUI:0080C2
TYPE:000E Length:0x2F
*Dec 21 10:22:12.727: 0000 0000 0080 0000 107B B9BD C400 0000 0080
0000 107B B9BD C480 0800 0014
*Dec 21 10:22:12.727: 0002 000F 0000
*Dec 21 10:22:12.727:
*Dec 21 10:22:13.931:
```

ATM0/0/0.1

```
(O):

!--- You also see debugs for interface ATM0/0/0.1 as you wanted.

VCD:0x1 VPI:0x1 VCI:0x21 DM:0x100 SAP:AAAA CTL:03 OUI:0080C2
TYPE:0007 Length:0x278
*Dec 21 10:22:13.931: 0000 FFFF FFFF FFFF 0010 7BB9 BDC4 0800 4500
025C 027F 0000 FF11 6147 0A30
*Dec 21 10:22:13.931: 4B9B FFFF FFFF 0044 0043 0248 0000 0101 0600
001A 4481 0000 8000 0000 0000
*Dec 21 10:22:13.931: 0000 0000 0000 0000 0000 0000 0010 7BB9 BDC3
0000 0000 0000 0000 0000 0000
*Dec 21 10:22:13.931: 0000 0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000
*Dec 21 10:22:13.931: 0000 0000 0000 0000 0000 0000 0000 0000 0000
```

```
0000 0000 0000 0000 0000 0000
*Dec 21 10:22:13.931: 0000 0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000
*Dec 21 10:22:13.931: 0000 0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000
*Dec 21 10:22:13.935: 0000 0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000
```

相关信息

- 拨号和接入技术支持
- 思科技术支持和下载

关于此翻译
思科采用人工翻译与机器翻译相结合的方式将此文档翻译成不同语言，希望全球的用户都能通过各自的语言得到支持性的内容。

请注意：即使是最好的机器翻译，其准确度也不及专业翻译人员的水平。

Cisco Systems, Inc. 对于翻译的准确性不承担任何责任，并建议您总是参考英文原始文档（已提供链接）。