21 BEIJING HOPE

21世纪网络工程师设计宝典丛书 4

Cisco Debug Commands Reference

网络核心技术内幕

Cisco Debug 命令参考

本书配套光盘内容包括：
与本书配套的电子书

21世纪网络工程师设计宝典丛书编委会　编

Cisco Debug Commands Reference

网络核心技术内幕

Cisco Debug 命令参考

本书配套光盘内容包括：
与本书配套的电子书

21世纪网络工程师设计宝典丛书编委会　编

北京希望电子出版社
Beijing Hope Electronic Press
www.bhp.com.cn

内 容 简 介

本书是21世纪网络工程师设计宝典系列之一，是专为从事网络开发和应用的人而编写的。

随着网络应用的不断深入，企业组网已经成为发展的必然趋势。如何设计企业组网的整套软、硬件解决方案已成为许多 IT 人员密切关心的问题。当网络出现故障时，尽快解决问题尤为关键。通过使用 Debug 命令，就可以快速地查找出故障发生的原因和地方，为故障的解决提供依据。

本书包括使用 Debug 命令、有条件地触发调试和 Debug 命令等三章，详细介绍了 Debug 命令的使用方法，以及命令的使用对路由器将产生的影响。对每个方法都给出了其命令格式、语法说明、使用说明等，并给出了命令的输出实例，以帮助读者掌握命令的使用。

本书结构清晰，实例丰富，技术新，使用性强。本书是企业 IT 人员、专业网络公司技术人员和系统集成人员的宝贵参考资料，是解决组网方案的重要参考手册，同时也是高等院校相关专业师生自学、教学参考用书和社会相关领域培训班教材。

本书配套光盘内容包括：与本书配套电子书。

系　列　书：21世纪网络工程师设计宝典系列（4）
书　　名：网络核心技术内幕——Cisco Debug 命令参考
文本著作者：21世纪网络工程师设计宝典丛书编写委员会
责任编辑：龙启铭
CD 制作者：希望多媒体创作中心
CD 测试者：希望多媒体测试部
出版、发行者：北京希望电子出版社
地　　址：北京海淀路82号，100080
网址：www.bhp.com.cn　E-mail：lwm@hope.com.cn
电话：010-62562329,62541992,62637101,62637102（图书发行,技术支持）
010-62633308,62633309（多媒体发行，技术支持）
010-62613322-215（门市）　010-62531267（编辑部）
经　　销：各地新华书店、软件连锁店
排　　版：希望图书输出中心
CD 生 产 者：文录激光科技有限公司
文本印刷者：北京双青印刷厂
开本 / 规格：787×1092 毫米　16 开本　　27.875 印张　646 千字
版次 / 印次：2000 年 3 月第 1 版　　2000 年 8 月第 2 次印刷
印　　数：5 001～8 000 册
本 版 号：ISBN7-900031-67-7/TP•67
定　　价：52.00 元（1CD，含配套书）

第 1 章　使用 Debug 命令

本章介绍如何使用 **debug** 命令诊断及网络连结中存在的故障。主要涉及以下的内容：

- 输入 **debug** 命令
- 使用 **debug** ？命令
- 使用 **debug all** 命令
- 生成调试输出
- 重新定向调试输出

警告　由于调试输出在 CUP 的进程中被赋予了高级的优先权，因而它可能导致系统不可用。为此，**debug** 命令只在排除某些特定故障或由 Cisco 技术维护人员排除话路故障时使用。此外，最好在网络使用的低峰期以及网上的用户较少时使用，在此期间进行调试可降低运行 **debug** 命令时影响系统使用的可能性。

1.1　输入 Debug 命令

所有 **debug** 命令的输入都使用特权 EXEC 模式，并且大部分 **debug** 命令不需要再添加参数。例如，要启用 **debug isdn q931** 命令，可在特权 EXEC 模式中输入以下命令：

debug isdn q931

要关闭 **debug isdn q931** 命令，可在特权 EXEC 模式中输入命令的 **no** 格式：

no debug isdn q931

此外，用户也可以在特权 EXEC 模式中输入 **undebug** 格式：

undebug isdn q931

要显示每个调试选项的状况，可在特权 EXEC 模式中输入以下命令：

show debugging

1.2　使用 Debug？命令

要列出并查看所有调试命令选项的简要说明，可在特权 EXEC 模式中输入以下命令：

debug ?

本书没有包含包含在 **debug ?**输出中列出所有的调试命令，而只介绍了有助于用户检测网络故障的命令。同时，本书未介绍的命令还包括在工程技术开发过程中使用的典型内部命令，在开发环境外将不使用这些命令。

1.3 使用 Debug All 命令

要启用对整个系统的诊断，可在特权 EXEC 模式中输入以下命令：

debug all

no debug all 命令可中止所有的诊断输出。使用 **no debug all** 命令是确保用户不会因疏忽而遗留下仍在运行的调试命令的简便方法。

警告 因为调试输出在网络中具有优先权，并且 **debug all** 命令比其它 **debug** 命令的输出量更大，因此它可能会严重降低路由器的使用效能，甚至可能导致路由器的不可用。在实际的测试中，最好尽量使用特定的 **debug** 命令。

1.4 生成调试命令输出

要生成某个 **debug** 命令的输出，可参照下面 **debug modem** 命令输出的例子：

```
Router# debug modem

15: 25: 51: TTY4: DSR came up
15: 25: 51: tty4: Modem: IDLE ->READY
15: 25: 51: TTY4: Autoselect started
15: 25: 51: TTY4: Autoselect failed
15: 25: 51: TTY4: Line reset
15: 25: 51: TTY4: Modem: READY->HANGUP
15: 25: 51: TTY4: dropping DTR, hanging up
15: 25: 51: tty4: Modem: HANGUP->IDLE
15: 25: 51: TTY4: restoring DTR
15: 25: 51: TTY4: DSR came up
```

在用户输入 **no debug** 命令前，路由器就会持续输出类似的信息。（本例为 **no debug modem** 命令）。

如果用户执行了某个 **debug** 命令但没有显示输出，可考虑以下几种可能性：

- 路由器可能没有按照生成用户希望查看的通信量类型进行适当配置。可使用 **more system:running-config** 命令检查其配置。
- 即使在路由器正确配置的情况下，在调试命令启用时，也可能在特定的阶段不生成用户希望查看的通信量类型。根据用户调试的协议，可使用如 TCP/IP 的 **ping** 命令以生成网络通信量。

1.5 重新定向调试及错误信息的输出

在默认的状况下，网络服务器和控制台发送调试命令输出和系统错误信息。如果用户使用了该项默认，监控调试输出将使用虚拟终端连接，而不是控制台端口。

要重新定向调试输出，可使用以下几节中介绍的 **logging** 命令选项配置。

可能的接收站包括控制台、虚拟终端、内部缓冲区和使用 syslog 服务器的 UNIX 主机。Syslog 格式要与 4.3 的“Berkeley 标准分布”（BSD）UNIX 及其派生版本。

注意　要当心用户所使用的调试接收站会影响系统的内务操作。到控制台的记录将生产大量的内务操作，而到虚拟终端的记录产生的内务操作要少一些。到 syslog 服务器的记录产生的内务操作更少，到内部缓冲器的记录是在所有方式中是最少的。

要配置信息的记录，用户需要进入配置命令模式。在 EXEC 提示符下使用 **configure terminal** 命令可进入此模式。

1.5.1　使用信息记录

要将信息登录到所有除控制台以外的受支持的目标，可输入以下内容：

logging on

默认的条件为 **logging on**。

要只定向登录到控制台上而不输出到其它接收站，可输入以下命令：

no logging on

1.5.2　设置信息记录级别

当登录信息发送到以下目标时，用户可设置记录等级：

- 控制台
- 监视器
- Syslog 服务器

表 1 列出了登录级别以及用户可用以设置此类型信息级别的相应的关键字，并进行了简要说明。最高的信息级别为 0 级：紧急（emergencies），最低的级别是 7 级：调试（debugging）。同时第 7 级显示的信息也最多。要限制这些信息的显示，可参见本章稍后的部分。

表 1　信息登录的关键字和级别

级别	关键字	说明	Syslog 定义
0	emergencies（紧急）	系统不可用	LOG_EMERG
1	alerts(报警)	需要立即采取行动	LOG_ALERT
2	critical(危急)	出现危急情况	LOG_CRIT
3	errors（错误）	存在错误情况	LOG_ERR
4	warnings（警告）	存在警告情况	LOG_WARNING
5	notification（注意）	出现普通但很重要的情况	LOG_NOTICE
6	informational(报告)	报告信息	LOG_INFO
7	debugging(调试)	调试信息	LOG_DEBUG

1.5.3　对发往控制台的登录信息类型的限制

要限制登录到控制台上的信息类型，可使用 **logging console** 路由器配置命令。该命令的语法如下：

logging console *level*

no logging console

logging console 命令限制在控制台只显示至特定严重级别的登录信息，并包括特定重要级别的登录信息。该级别的信息是由参数 *level* 指定的。

level 参数为表 1 中的关键字。表中的关键词是按照从最重要到最不重要的顺序排列。

no logging console 命令可取消到控制台的登录。

实例

以下是把控制台的登录信息级别设置为 **debugging** 级的例子，这是最不重要的级别，可显示所有的登录信息。

logging console debugging

1.5.4　到内部缓冲区的登录信息

控制台是默认的登录设备，除非特别指定，所有的信息将显示在控制台上。

要将信息登录到内部缓冲区，可使用 **logging buffered** 路由器配置命令。该命令的语法如下：

logging buffered
no logging buffered

logging buffered 命令可将登陆信息复制到内部缓冲区而不将其写入控制台。缓冲区可循环使用，因此新的信息将覆盖旧信息。要显示登录到内部缓冲区上的信息，可使用 **show logging** 特权 EXEC 命令。所显示的第一条信息是缓冲区上最早的信息。

no logging buffered 命令取消使用缓冲区而将信息写入控制台（默认）。

1.5.5　发送到其它监视器上的登录信息类型的限制

要限制登录到终端监控器的信息级别，可使用 **logging monitor** 路由器配置命令。该命令的语法如下：

logging monitor *level*
no logging monitor

logging monitor 命令限制在终端监控器而不是控制台监控器上显示的登录信息级别到根据特定 *level* 参数指定的信息级别，其中包括 *level* 参数指定的信息级别。*level* 参数是表 1 中的关键字之一。要显示在终端（虚拟控制台）上的登录信息，可使用 **terminal monitor** 特权 EXEC 命令。

no logging monitor 命令取消控制台监控器以外的终端监控器的登录。

实例

以下实例把在监控器而不是控制台上显示的信息级别设置为 **notification**。

logging monitor notification

1.5.6　UNIX Syslog 服务器的登录信息

要登录信息至 syslog 服务器，可使用 **logging** 路由器配置命令。该命令的句法如下：

logging ip-*address*
no logging *ip-address*

logging 命令可识别 syslog 服务器主机以接收登录信息。*ip-address* 参数为主机的 IP 地址。通过多次发布该命令，用户可创建接收登录信息的 syslog 服务器列表。

no logging 命令可从 syslogs 列表中删除带有指定地址的 syslogs 服务器。

1.5.7　限制发送到 Syslog 服务器的信息

要限制发送到 syslogs 服务器上的信息数量，可使用 **logging trap** 路由器配置命令。该命令的句法如下：

logging trap *level*
no logging trap

logging trap 命令限制发送到 syslogs 服务器上的登录信息到根据特定 *level* 参数指定的信息级别，其中包括 *level* 参数指定的信息级别。*level* 参数为表 1 中的关键字。

要发送登录信息到 syslogs 服务器，可用 **logging** 命令指定其主机地址。默认的中断级别为 **informational**。

no logging trap 可取消登录到 syslogs 服务器。

目前的软件可生成以下四类 syslog 信息：

- 关于软件或硬件故障的错误信息，以 **errors** 级别显示。
- 接口可用/不可用转换和系统重新启动信息，以 **notification** 级别显示。
- 重载请求和低处理堆栈信息，以 **informational** 级别显示。
- **debug** 命令的输出，以 **debugging** 级别显示。

show logging 特权 EXEC 命令显示与当前登录设置相关的地址和级别。该命令的输出包括辅助统计信息。

UNIX Sy slog 端口监控程序设置实例

要在一个 4.3 BSD UNIX 系统上设置 syslog 端口监控程序，应在*/etc/syslog. conf* 文件中包括如下一行：

```
local7.debugging /usr/adm/logs/tiplog
```

关键字 **local7** 指定可用的登录设备。

关键字 **debugging** 指定 syslog 级别。请参阅表 1 中的其它关键字。

UNIX 系统用该级别或更高的级别将信息发送到指定文件中，本例中的文件为*/usr/adm/logs/tiplog*。这个文件必须是已经存在的，同时 syslog 端口监控程序必须获得写入许可。

对于 System V UNIX 系统，该行应为：

```
local7-debug /usr/admin/logs/cisco.log
```

第 2 章　有条件地触发调试

在启用有条件地触发调试功能时，路由器将对在指定接口上进入或离开路由器的数据包生成调试信息，而不会为通过其它接口进入或离开路由器的数据包生成调试输出。用户可以明确地指定接口。例如，可以只显示一个接口或子接口的调试信息，也可以调试所有符合指定条件的接口。该项功能在拥有大量端口的拨号存取服务器上极为有用。

通常情况下，路由器会为每一个接口生成调试信息，这样会产生大量的信息。这些信息将消耗系统资源，并使得用户难以查找所需要的信息。通过限制调试信息，用户可以接收只与用户希望调试的端口相关的信息。

有条件地触发调试控制了从以下特定协议的 **debug** 命令的输出：

- debug aaa {accounting | authorization | authentication}
- debug dialer (events I packets)
- debug isdn {q921 | q931}
- debug modem {oob | trace}
- debug ppp {all | authentication | chap | error | negotiation | multilink events | packet}

当这项功能限制上述命令的输出时，不会自动激活这些命令的调试输出。只有在启用特定协议的 **debug** 命令时才生成调试信息。以下两种进程控制 **Debug** 命令的输出：

- 特定协议的 **debug** 命令指定调试哪些协议。例如，**debug dialer events** 命令生成与拨号事件相关的调试输出。
- **debug condition** 命令只生成与某一特定接口相关的调试信息。例如，**debug condition username bob** 命令只为用户名为 bob 的数据包的接口生成调试输出。

要配置有条件地触发调试，可按以下方法操作：

- 激活特定协议的调试命令
- 激活有条件的调试命令
- 指定多重条件

2.1　激活特定协议调试命令

要生成调试输出，必须激活希望输出的特定协议的 **debug** 命令。使用 **show debugging** 命令可确定被激活的调试的类型。在特权 EXEC 模式下使用以下命令可激活或取消所希望的特定协议的 **debug** 命令：

命令	目的
show debugging	决定激活的调试类型
debug protocol	激活需要的调试命令
no debug protocol	取消不需要的调试命令

如果用户不需要任何输出，可取消所有特定协议的 **debug** 命令。

2.2　激活有条件的调试命令

如果没有激活 **debug condition** 命令，将忽略接口，显示激活的特定协议的 **debug** 命令的所有调试输出。

用户输入的第一个 **debug condition** 可激活有条件的调试。路由器将只显示为符合指定条件之一的接口显示信息。如果指定了多重条件，接口就必须符合至少一个条件才能显示信息。

用户可以为明确指定的接口或符合特定条件的接口激活信息，如下所示：

- 为某一个接口显示信息
- 为多个接口显示信息
- 根据条件限制信息显示

2.2.1　某一个接口的信息显示

要取消某一接口外其他所有接口的调试信息，可在特权 EXEC 模式下使用以下命令：

命令	目的
debug condition Interface *interface*	取消某一接口外其他所有接口的调试信息

如果用户输入 **the debug condition interface** 命令，调试输出将关闭指定接口以外的所有接口的调试输出。要为所有接口重新激活调试信息，可使用 **no debug interface command** 命令。

2.2.2　多接口信息显示

要为多接口激活调试信息，可在特权 EXEC 模式下使用以下命令：

步骤	命令	目的
步骤 1	**debug condition interface** *interface*	取消某一接口外其他所有接口的调试信息
步骤 2	**debug condition interface** *interface*	为附加的接口激活调试信息。重复该命令，直到为所有希望调试的接口激活调试信息

如果用户通过多次输入此命令指定了一个以上的接口，则调试输出将为所有的指定接口显示调试输出。要关闭某一特定接口上的调试输出，可使用 **no debug interface** 命令。如果用户使用了 **no debug interface all** 命令或取消了最后一个 **debug interface** 命令，将重新激活所有接口的调试输出。

2.2.3　基于条件的信息限制

路由器可以监视接口以查看是否有数据包含有以下条件之一的特定值。

- 用户名
- 呼叫方号码
- 被呼方号码

如果输入一个条件，如呼叫号码，则所有接口将停止调试输出。然后，路由器将监视每一个接口以查看是否有含有指定呼叫方号码的数据包在接口上发送或接收。如果某一接口或子接口与条件相符，**debug** 命令将显示该接口的调试输出。当条件符合时，将“触发”某一接口的调试输出。其它接口的调试输出仍不可用。如果稍后有另一接口符合条件，则

将为该接口激活调试输出。

一旦激活某一个接口的调试输出，在该接口为不可用前将继续输出调试信息。但该接口的话路可能会发生变化，如出现新的用户名、被呼方号码和呼叫方号码，应使用 **no debug interface** 命令为特定接口重新设置调试触发机制。该接口的调试输出将被取消，直至该接口符合任一指定条件。

要限制基于指定条件的调试信息，可在特权 EXEC 模式下使用以下命令：

命令	目的
debug condition {**username** *username* | **called** *dial-string* | **caller** *dial-string*	取消有条件的调试。路由器将只为符合该条件的接口显示信息

要重新激活所有接口调试输出，可使用 **no debug condition all** 命令。

2.3　指定多重条件

要限制基于一个以上条件的调试信息，可在特许 EXEC 模式下使用以下命令：

步骤	命令	目的
步骤 1	**debug condition** {**username** *username* | **called** *dial-string* | **caller** *dial-string*}	激活有条件的调试并指定第一个条件
步骤 2	**debug condition** {**username** *username* | **called** *dial-string* | **caller** *dial-string*}	指定第二个条件，重复该任务，直到指定了所有条件

如果用户多次输入 **debug condition** 命令，当某一接口符合了至少一个条件时，则将生成调试输出。如果用 **no debug condition** 命令取消其中的一个条件，则只符合该条件的接口将不再产生调试输出。不过，符合未被取消条件的接口仍将继续产生输出。只有接口不符合任何有效条件时，该接口的输出才会被取消。

2.4　有条件地触发调试配置实例

在本例中，用以下命令设置了四个条件：

- **debug condition interface serial 0**
- **debug condition interface serial 1**
- **debug condition interface virtual-template 1**
- **debug condition username fred**

接口只符合前三个条件而不符合第四个条件。

```
Router# show debug condition

Condition 1: interface Se0 (1 flags triggered)
        Flags: SeO
Condition 2: interface Sel (1 flags triggered)
        Flags: Se1
Condition 3: interface Vt1 (1 flags triggered)
        Flags: Vt1
Condition 4: username fred (0 flags triggered)
```

在输入任一 **debug condition** 命令时，则有条件地调试的调试信息被激活。以下调试信息显示了不同接口（串口 0 和串口 1）符合条件。例如，输出的第二行显示了串口 0 符

合 username fred 条件。

```
*Mar 1 00:04:41.647: *LINK-3-UPDOWN: Interface Serial0, changed state to up
*Mar 1 00:04:41.715: Se0 Debug: Condition 4, username fred triggered, count 2
*Mar 1 00:04:42.963: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0,changed state to
up
*Mar 1 00:04:43.271: Vil Debug: Condition 3, interface Vt1 triggered, count 1
*Mar 1 00:04:43.271: %LINK-3-UPDOWN: Interface Virtual-Accessl, changed state to up
*Mar 1 00:04:43.279: Vil Debug: Condition 4, username fred triggered, count 2
*Mar 1 00:04:43.283: Vil Debug: Condition 1, interface Se0 triggered, count 3
*Mar 1 00:04:44.039:  %IP-4-DUPADDR: Duplicate address 172.27.32-114 on Ethernet 0, sourced by
00e0.le3e.2d41
*Mar 1 00:04:44.283: %LINEPROTO-5-UPDOWN: Line protocol on Interface Virtual-Accessl, changed
state to up
        *Mar 1 00:04:54.667: %LINK-3-UPDOWN: Interface Serial1, changed state to up
        *Mar 1 00:04:54.731: Sel Debug: Condition 4, username fred triggered, count 2
        *Mar 1 00:04:54.735: Vil Debug: Condition 2, interface Sel triggered, count 4
*Mar 1 00:04:55.735: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial1, changed
state to up
```

经过一段时间后，**show debug condition** 命令将显示修订后的条件列表：

```
Router# show debug condition

Condition 1: interface Se0 (2 flags triggered)
        Flags: Se0 Vil
Condition 2: interface Se1 (2 flags triggered)
        Flags: Sel Vil
Condition 3: interface Vt1 (2 flags triggered)
        Flags: Vt1 Vi1
Condition 4: username fred (3 flags triggered)
        Flags: Se0 Vi1 Se1
```

此后，串口 1 和串口 0 为不可用。当某一接口为不可用时，该接口的条件将被清除。

```
*Mar 1 00:05:51.443: %LINK-3-UPDOWN: Interface Seriall, changed state to down
*Mar 1 00:05:51.471: Se1 Debug: Condition 4, username fred cleared, count 1
*Mar 1 00:05:51.479: Vi1 Debug: Condition 2, interface Se1 cleared, count 3
*Mar 1 00:05:52.443: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial, changed stateto
down
*Mar 1 00:05:56.859: %LINK-3-UPDOWN: Interface Serial0, changed state to down
*Mar 1 00:05:56.887: SeO Debug: Condition 4, username fred cleared, count 1
*Mar 1 00:05:56.895: Vi1 Debug: Condition 1, interface Se0 cleared, count 2
*Mar 1 00:05:56.899: Vi1 Debug: Condition 3, interface Vt1 cleared, count 1
*Mar 1 00:05:56.899: Vi1 Debug: Condition 4, username fred cleared, count 0
*Mar 1 00:05:56.903: %LINK-3-UPDOWN: Interface Virtual-Accessl, changed state to down
*Mar 1 00:05:57.907: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0, changed state to
down
*Mar 1 00:05:57.907: %LINEPROTO-5-UPDOWN: Line protocol on Interface Virtual-Accessl, changed
state to down
```

最后一个 **show debug condition** 输出与接口在可用前的输出相同。

Router# **show debug condition**

Condition 1: interface SeO (1 flags triggered)
    Flags: SeO
Condition 2: interface Se1 (1 flags triggered)
    Flags: Se1
Condition 3: interface Vt1 (1 flags triggered)
    Flags: Vt1
Condition 4: username fred (O flags triggered)

第 3 章　Debug 命令

本章按字母排序对 **debug** 命令进行介绍。每个命令包括其用法的简短说明、命令的语法、使用说明、输出实例和对输出的说明。

输出的编排格式依各条命令而定。某些命令每个数据包只产生一行输出，而其它的一些命令可能会产生多行输出；某些命令会产生大量的输出，而另一些可能偶尔才有输出；一些命令产生的是文字行，而有些命令生成的信息却使用字段的格式。此外，**debug** 命令输出的方式也是多样的。例如，生成文字行的 **debug** 命令输出通常一行一行地说明，而生成信息用字段格式表示的命令通常用表格说明。

在默认方式下，网络服务将 **debug** 命令的输出发送到控制台。发送输出到一个终端（虚拟控制台）比发送到控制台所用的内务操作要少。可使用特权 EXEC 命令 **terminal monitor** 将输出发送到一个终端。如需了解关于重新定向的更多知识，请参阅“使用 **debug** 命令”一章。

3.1　debug aaa accounting

使用 **debug aaa accounting** EXEC 命令显示当事件发生时的信息。该命令的 **no** 格式取消调试输出。

[no] debug aaa accounting

使用说明

通过 **debug aaa accounting** 命令显示的信息独立于用于将计帐信息转换到服务器上的计帐协议。使用 **debug tacacs** 和 **debug radius** 协议特殊命令可获得有关协议层问题更详细的情况。

用户也可以使用 **show accounting** 命令单步调试所有的话路，并为积极计帐的功能打印所有的帐户记录。**show accounting** 命令可使用户在系统上显示有效的可计帐事件。它可让系统管理程序快速查看下一步骤，万一计帐服务器上发生某类丢失数据时对于信息的搜集也很有用。如果 **debug aaa accounting** 处于开启状态，**show accounting** 命令可显示验证、授权和计帐安全系统内在状态中的附加信息。

实例

以下是 **debug aaa accounting** 命令的范例输出。

```
Router# debug aaa accounting

16:49:21: AAA/ACCT: EXEC acct start, line 10
16:49:32: AAA/ACCT: connect start, line 10, glare
16:49:47: AAA/ACCT: connection acct stop:
task_id=70 service=exec port=10 protocol=telnet address=172.31.3.78 cmd=glare
bytes_in=308 bytes_out=76 paks_in=45 paks_out=54 elapsed_time=14
```

相关命令

debug aaa authentication
debug aaa authorization
debug radius
debug tacacs

3.2　debug aaa authentication

使用 **debug aaa authentication** EXEC 命令显示 AAA/终端存取控制器附加存取控制系统（TACASC+）的验证信息。使用此命令的 **no** 格式取消调试输出。

[no] debug aaa authentication

使用说明

利用该命令可显示正在使用的验证以及使用这些的结果。

实例

以下是 **debug aaa authentication** 命令的范例输出。本例显示了使用默认列表和第一个（即 TACAS+）的单个 EXEC 命令注册。TACAS+服务器发送 GETUSER 请求提示输入用户名，然后发送 GETPASS 请求提示输入口令，最后发送 PASS 应答以显示注册成功。数字 50996740 是话路标识符（ID），每个验证拥有唯一的话路标识符。如果同时进行几个验证，使用该标识符编号可区别不同的验证。

Router# **debug aaa authentication**

```
6 :50: 12: AAA/AUTHEN: create_user user″ ruser=″ port= ′ tty19′ rem_addr=′ 172.31.60.15′
authen_type=1 service=1 priv=1
6: 50: 12: AAA/AUTHEN/START (0 ): port= ′ tty19′ list=″ action=LOGIN service=LOGIN
6: 50: 12: AAA/AUTHEN/START (0 ): using ″ defau1tt″ 1ist
6: 50: 12: AAA/AUTHEN/START (50996740): Method=TACACS+
6: 50: 12: TAC+ (50996740): received authen response status = GETUSER
6: 50: 12: AAA/AUTHEN (50996740): status = GETUSER
6: 50: 15: AAA/AUTHEN/CONT (50996740) = continue_login
6: 50: 15: AAA/AUTHEN (50996740): status = GETUSER
6: 50: 15: AAA/AUTHEN (50996740): Method=TACACS+
6: 50: 15: TAC+: send AUTHEN/CONT packet
6: 50: 15: TAC+ (50996740): received authen response status = GETPASS
6: 50: 15: AAA/AUTHEN (5O996740): status = GETPASS
6: 50: 20: AAA/AUTHEN/CONT (50996740): continue_login
6: 50: 20: AAA/AUTHEN (50996740): status = GETPASS
6: 50: 20: AAA/AUTHEN (50996740): Method=TACACS+
76: 50: 20: TAC+: send AUTHEN/CONT packet
6: 50: 20: TAC+ (50996740): received authen response status = PASS
6: 50: 20: AAA/AUTHEN (50996740): status = PASS
```

3.3　debug aaa authorization

使用 **debug aaa authorization** EXEC 命令可显示 AAA/TACAS+授权信息。该命令的 **no** 格式调试输出。

[no] debug aaa authorization

使用说明

使用该命令可显示正在使用的授权方法以及使用这些方法的结果。

实例

以下是 **debug aaa authorization** 命令的输出实例。本例显示了用户 carrel 的 EXEC 授权。在第一行中，用户名被授权。在第二行和第三行中 AV 对（属性值）被授权。调试输出为每个 AV 对显示了一行信息。此外，显示表示明了所使用的授权方式。显示信息的最后一行表明授权处理的状态，在本例中，该状态为失败。

```
Router# debug aaa authorization

2: 23: 21: AAA/AUTHOR (0): user= ' carrel'
2: 23: 21: AAA/AUTHOR (0): send AV service=shell
2: 23: 2l: AAA/AUTHOR (0): send AV cmd*
2: 23: 21: AAA/AUTHOR (342885561): Method=TACACS+
2: 23: 21: AAA/AUTHOR/TAC+ (342885561): user=carrel
2: 23: 2l: AAA/AUTHOR/TAC+ (342885561): send AV service=shell
2: 23: 21: AAA/AUTHOR/TAC+ (342885561): send AV cmd*
2: 23: 2l: AAA/AUTHOR (342885561): Post authorization status = FAIL
```

aaa authorization 命令生成一个请求数据包，其中包括作为授权程序的一部分发送到 TACACS 后台程序的一系列 AV 对。后台程序以下列三种方式之一进行应答：

- 接收请求
- 改变请求
- 拒绝请求，因而拒绝授权

表 2 描述了可能在调试输出中显示的与 **aaa authorization** 相关联的属性值对。

表 2　授权的属性值对

属性值	说明
service=arap	正在请求 AppleTalk 远程访问的授权
service=shell	正在请求 EXEC 启动授权和命令授权
service=ppp	正在请求 PPP 授权
service=slip	正在请求 SLIP 授权
protocol=lcp	正在请求 LCP（PPP 的更低层）授权
protocol=ip	与 service=slip 一起使用，service=slip 可显示正在授权的协议层
protocol=ipx	与 service=ppp 一起使用，可显示正在授权的协议层
protocol=atalk	与 service=ppp 或 service=arap 同时使用可显示正在授权的协议层
protocol=vines	与 service=ppp 一起使用，可使 VINES 越过端对端协议（PPP）

（续表）

属性值	说明
protocol=unknown	在未定义或无支持的条件下使用
cmd=x	与 service=shell 一起使用。如果 cmd 为空，这是用于启动 EXEC 的授权请求如果 cmd 不为空，则这是一个授权请求命令，其中包含了正在授权的命令名称。例如，cmd=telnet
cmd-ang=x	与 service=shell 一起使用。当授权执行命令时，该命令的名称是由所列的每个参数的 cmd=x 对提供的。例如，cmd-arg=archie.sura.net
acl=x	与 service=shell 及 service=arap 一起使用。对于 ARA，该对包含了存取列表编号。对于 service=shell，该对包含了一个存取类编号。例如，acl=2
inacl=x	与 service=ppp 及 protocol=ip 一起使用。包含 SLIP 或 PPP/IP 的 IP 输入存取列表。例如，inacl=2
outacl=x	与 service=ppp 及 protocol=ip 一起使用。包含 SLIP 或 PPP/IP 的 IP 输出存取列表。例如，inacl=4
addr=x	与 service=slip,service=ppp 及 protocol=ip 一起使用。包含一个 IP 地址，在通过 SLIP 或 PPP/IP 连接时，远程主机应使用该地址。例如，addr=172.30.23.11
routing=x	与 service=slip,service=ppp 及 protocol=ip 一起使用。与 SLIP 和 PPP 命令中的路由标记的功能相似，可以为真，也可为假。例如，routing=true
timeout=x	与 service=arap 一起使用。在 ARA 话路切断连接前的分钟数字。例如，timeout=60
autocmd=x	与 service=shell 及 cmd=NULL 一起使用，在 EXEC 启动时指定执行一条自动命令。例如，artocmd=telnet foo.com
noescape=x	与 service=shell 及 cmd=NULL 一起使用。为用户名配置命令指定一个 noescape 选项，可以为真，也可以为假。例如，noescape=true
nohangup=x	与 service=shell 及 cmd=NULL 一起使用。为用户名配置命令指定一个 nohangup 选项，可以是真实的，也可是虚拟的，例如，nohangup=false
priv-lvl=x	与 service=shell 及 cmd=NULL 一起使用。为命令授权指定一个从 0-15 的当前特权级，例如，priv-lvl=15
zonelist=x	与 service=arap 一起使用。为 ARA 指定一个 AppleTalk 的存贮区列表，如 zonelist=5
addr-pool=x	与 service=ppp 及 protocol=ip 一起使用，为从远端主机获取地址的本地库指定名称

3.4 debug alps ascu

使用 **debug alps ascu** EXEC 命令激活对 ALPS ASCU 的调试。该命令的 **no** 格式取消调试输出。

[no] debug alps ascu {event | packet | detail | all} [*interface-id*]

语法说明

event	显示 ALC 异常事件或协议错误
packet	显示传输或接收的数据包
detail	显示所有 ALC 协议事件
all	激活事件、数据包和元件调试
interface-id	（可选）激活指定的 ASCU 的调试