



Важные сведения о командах Debug

Содержание

Введение

Предварительные условия

- Требования
- Используемые компоненты
- Предупреждения
- Условные обозначения

До отладки

- Получение выходных данных отладки
- Прочие задачи перед началом отладки
- Прекращение отладки

Использование команды `debug ip packet`

Условно запускаемые отладки

Дополнительные сведения

Введение

На этой странице даны несколько основных рекомендаций по использованию функций отладки, доступных на платформах Cisco IOS®, а также примеры правильного использования команды `debug ip packet` и условной отладки.

Примечание. Этот документ не объясняет, как использовать и интерпретировать специальные команды отладки `debug` и их выходные данные. Сведения о специальных командах `debug` см. в соответствующем документе "Справочник Cisco по командам debug".

Выходные данные привилегированных команд EXEC `debug` предоставляют диагностические сведения относительно множества событий межсетевого взаимодействия, относящихся к состоянию протокола и к сетевой активности в целом.

Предварительные условия

Требования

Использование данного документа требует наличия следующих знаний.

- Подключение к маршрутизатору с помощью консоли, портов aux и vty.
- Общие проблемы конфигурации IOS.
- Интерпретация выводов команд debug IOS

Используемые компоненты

Настоящий документ не имеет жесткой привязки к устройству или какой-либо версии ПО.

Сведения, содержащиеся в данном документе, были получены с устройств в специальной лабораторной среде. Все устройства, используемые в этом документе, были запущены с чистой (заданной по умолчанию) конфигурацией. В рабочей сети необходимо изучить потенциальное воздействие всех команд до их использования.

Предупреждения

Команды **debug** необходимо использовать с осторожностью. Обычно рекомендуется использовать эти команды только под руководством представителя технической поддержки своего маршрутизатора при выявлении конкретных неполадок.

Включение режима отладки может повлиять на работу маршрутизатора при высокой загрузке внутренней сети. Следовательно, если включена регистрация событий, сервер будет периодически недоступен, если порт консоли перегружен сообщениями о регистрации.

Используя команду **debug**, всегда принимайте во внимание объем выходных данных этой команды и время ее выполнения. Например, если имеется маршрутизатор с одним интерфейсом базового уровня (BRI), команда **debug isdn q931** не повредит системе. Однако такая же отладка на AS5800 с полной конфигурацией E1 приведет к такому объему входных данных, что устройство может "зависнуть".

Перед использованием отладки определите степень загруженности процессора, применив команду **show processes cpu**. Перед началом отладки убедитесь, что имеющийся CPU обладает достаточной производительностью. Дополнительные сведения об управлении процессором в случае высокой загрузки см. в документе Устранение неполадок, связанных с высокой загрузкой процессора в маршрутизаторах Cisco. Например, при наличии маршрутизатора Cisco 7200 с интерфейсом ATM, использующего мост, – в зависимости от количества настроенных субинтерфейсов – перезагрузка маршрутизатора может вызвать высокую загрузку процессора. Здесь причиной является то, что для каждого виртуального канала необходимо создать пакет BPDU. Начало отладки в столь критический период может вызвать резкое увеличение загрузки CPU и привести к зависанию или потере сетевого соединения.

Примечание. При работе отладчика приглашение маршрутизатора обычно не видно, особенно в моменты интенсивной отладки. Однако в большинстве случаев для остановки процесса отладки можно использовать команды **no debug all** или **undebug all**. Дополнительные сведения о безопасном использовании отладки см. в разделе Получение выходных данных отладки.

Условные обозначения

Дополнительные сведения об условных обозначениях в документах см. в статье Условные обозначения, используемые в технической документации Cisco.

До отладки

Помимо аспектов, упомянутых выше, следует понять, как отладка влияет на стабильность платформы. Также следует установить, к какому интерфейсу маршрутизатора необходимо подключиться. В следующем разделе приведено несколько инструкций.

Получение выходных данных отладки

Маршрутизаторы могут отображать результаты отладки для различных интерфейсов, в том числе для консольных, вспомогательных и VTY-портов. Маршрутизаторы могут также протоколировать сообщения, сохраняющиеся во внутреннем буфере, на внешнем syslog-сервере с ОС Unix. Инструкции и предупреждения относительно каждого метода обсуждаются далее.

Порт консоли

При стандартных настройках подключения к консоли дополнительных действий не требуется. Выходные данные отладки должны быть автоматически выведены на экран. Однако следует убедиться в том, что уровень регистрации при входе в консоль установлен с помощью команды **logging console level** так, как описано выше, и что вход не был запрещен с помощью команды **no logging console**. Дополнительные сведения см. в документе Использование команд debug.



Предупреждение. Избыток данных отладки в порте консоли маршрутизатора может вызвать зависание устройства. Это происходит потому, что маршрутизатор автоматически присваивает выходным данным консоли высший приоритет по отношению к другим своим функциям. По этой причине маршрутизатор может зависнуть, если он обрабатывает выходные данные отладки большого объема для порта консоли. Следовательно, если выходных данных отладки много, воспользуйтесь портами VTY (telnet) или буферами журнала для получения необходимых отладок. Подробнее об этом см. ниже.

Примечание. По умолчанию на порте консоли функция регистрации данных включена. Поэтому порт консоли всегда обрабатывает результаты отладки, даже если на самом деле для сбора результатов используется другой порт или метод (например, AUX, VTU или буфер). Следовательно, при нормальном рабочем состоянии рекомендуется, чтобы всегда была включена команда **no logging console**, и использовались другие способы сбора отладочных сведений. В ситуациях, в которых необходимо использовать консоль, временно включите вход в систему через консоль с помощью команды **logging console**.

Вспомогательный порт

Если соединение установлено через вспомогательный порт, введите команду **terminal monitor**. Убедитесь также, что команда **no logging on** не активирована на маршрутизаторе.

Примечание. Используя вспомогательный порт для отслеживания состояния маршрутизатора, помните, что при перезапуске вспомогательный порт не отображает данные последовательности загрузки. Чтобы увидеть последовательность загрузки, подключитесь к порту консоли.

Порты VTU

Если соединение установлено через вспомогательный порт или протокол Telnet, введите команду **terminal monitor**. Также убедитесь, что команда **no logging on** не использовалась.

Запись сообщений во внутренний буфер

Консоль является устройством записи данных по умолчанию; все сообщения отображаются в консоли при отсутствии специальных настроек.

Записать сообщения во внутренний буфер позволяет команда настройки маршрутизатора буферизованной регистрации. Полный синтаксис этой команды.

```
logging buffered
no logging buffered
```

Команда **logging buffered** копирует сообщения регистрации во внутренний буфер вместо записи их в консоль. Буфер имеет кольцевую природу, поэтому более поздние сообщения записываются поверх более ранних. Чтобы отобразить сообщения, зарегистрированные в буфере, используйте привилегированную команду EXEC **show logging**. Первым сообщением, отображенным на экране, является самое старое сообщение из находящихся в буфере. Можно указать размер буфера, а также уровень важности регистрируемых сообщений.

Совет. Убедитесь в достаточном количестве доступной памяти модуля, прежде чем указывать размер буфера. Используйте команду IOS **show proc mem**, чтобы определить количество доступной памяти.

Команда **no logging buffered** отменяет использование буфера и записывает сообщения в консоль (по умолчанию).

Сообщения регистрации сервера системного журнала UNIX

Чтобы начать регистрацию сообщений на сервере системных журналов, используйте команду настройки маршрутизатора регистрации. Полный синтаксис этой команды выглядит следующим образом.

```
logging <ip-address>
no logging <ip-address>
```

Команда **logging** определяет узел сервера системного журнала для получения сообщений регистрации. Аргументом **<ip-address>** является IP-адрес хоста. При использовании этой команды больше одного раза создается список серверов системного журнала,

принимающих сообщения регистрации.

Команда **no logging** удаляет из списка системных журналов сервер системного журнала с указанным адресом.

Дополнительные сведения о настройке syslog-сервера см. в документе [Использование команд debug](#).

Прочие задачи перед началом отладки

1. Настройте ПО эмулятора терминала (например HyperTerminal) для сбора данных отладки в файл. Например, в программе HyperTerminal щелкните **Transfer**, затем **Capture Text** и настройте соответствующие параметры. Дополнительные сведения см. в документе [Запись текста выходных данных программы Hyperterminal](#). Для другого ПО эмуляторов терминала см. документацию к соответствующему ПО.
2. Включить метки времени в миллисекундах (мсек) с помощью команды **service timestamps**.

```
router(config)#service timestamps debug datetime msec
router(config)#service timestamps log datetime msec
```

Эти команды добавляют метки времени в данных отладки в формате МММ ДД ЧЧ:ММ:СС, указывая дату и время в соответствии с системными часами. Если системные часы не настроены, перед датой и временем будет отображаться звездочка (*), указывающая на вероятность того, что дата и время указаны неверно.

Рекомендуется устанавливать значение метки времени в миллисекундах, что позволяет обеспечить больший уровень качества и наглядности представления выходных данных отладки. Метки времени в миллисекундах предоставляют более точное определение времени, в которое были произведены различные отладки по отношению друг к другу. Примечание. Если порт консоли выдает большое количество сообщений, они могут не совпадать с реальным моментом их появления. Например, если включена команда **debug x25** для блока, содержащего 200 виртуальных каналов, а выходные данные регистрируются в буфере (с использованием команд **no logging console** и **logging buffered**), метка времени, отображаемая в выходных данных команд **debug** (в буфере), может не соответствовать точному времени прохождения пакета через интерфейс. Таким образом, метки времени MSEC предпочтительнее использовать не для определения проблем производительности, а для получения соответствующих данных во время совершения событий.

Прекращение отладки

Чтобы остановить отладку, необходимо использовать команду **no debug all** или **undebg all**. Убедитесь, что отладка была прекращена посредством использования команды **show debug**.

Запомните, что команды **no logging console** и **terminal no monitor** препятствуют только выводу выходных данных в порте консоли, вспомогательном или vty-порте. Это не останавливает отладку и поэтому истощает ресурсы маршрутизатора.

Использование команды **debug ip packet**

Команда **debug ip packet** позволяет получить информацию о пакетах, для которых маршрутизатор не использует быструю коммутацию. Однако так как он генерирует выход для каждого пакета, выход может быть пространственным, и это приводит к зависанию маршрутизатора. По этой причине команду **debug ip packet** следует использовать под жестким контролем, как описано в этом разделе.

Самый эффективный способ ограничить выходные данные команды **debug ip packet** – создать список доступа, связанного с операцией отладки. Только пакеты, которые совпадают с критериями списка доступа станут предметом обработки команды **debug ip packet**. Этот список доступа не следует применять ни к какому интерфейсу, но он может быть применен к операции отладки.

Перед отладкой IP-пакетов с помощью команды **debugging ip packet** учтите, что маршрутизатор выполняет по умолчанию быструю коммутацию или, при соответствующей настройке, коммутациюCEF. Это означает, что когда будет задействован этот метод, пакет не передается на процессор, следовательно, отладка не показывает никаких данных. Чтобы данная функция работала, необходимо отключить в маршрутизаторе быструю коммутацию с помощью команды **no ip route-cache** (для одноадресных пакетов) или **no ip mroute-cache** (для многоадресных пакетов). Это должно применяться к интерфейсам, на которых предполагается поток трафика.

Убедитесь в этом при помощи команды **show ip route**.

Предупреждения.

- Отключение быстрой коммутации на маршрутизаторе, который обрабатывает большое количество пакетов, может привести к использованию CPU для всплеска загрузки таким образом, что устройство зависает или теряет соединения с одноранговыми узлами.
- Не отключайте быструю коммутацию на маршрутизаторе, в котором используется многопротокольная коммутация с использованием меток (MPLS). MPLS используется совместно с методом коммутации CEF. Таким образом, отключение быстрой коммутации интерфейса может оказать разрушительное воздействие.

Рассмотрим примерный сценарий.



Сконфигурированный список доступа в маршрутизаторе_122:

```
access-list 105 permit icmp host 10.10.10.2 host 13.1.1.1
access-list 105 permit icmp host 13.1.1.1 host 10.10.10.2
```

Этот список разрешает доступ любому пакету протокола (ICMP) с главного маршрутизатора_121 (с IP-адресом 10.10.10.2) на главный маршрутизатор_123 (с IP-адресом 13.1.1.1), и в обратном направлении. Важно разрешить любое направление для пакетов, в противном случае маршрутизатор может отбрасывать возвратные ICMP-пакеты.

Отключим быструю коммутацию только в интерфейсе маршрутизатора_122. Это значит, что теперь видимы только данные отладки для пакетов, предназначенных для этого интерфейса (с точки зрения IOS с поддержкой перехвата пакеты). В данных отладки такие пакеты отображаются с "d=". Так как еще не отключена быстрая коммутация в другом интерфейсе, возвратный пакет не будет обрабатываться командой **debug ip packet**. Выходные данные, приведенные ниже, показывают, как отключить быструю коммутацию.

```
router_122(config)#interface virtual-template 1
router_122(config-if)#no ip route-cache
router_122(config-if)#end
```

Теперь необходимо активировать команду **debug ip packet** с помощью списка доступа, определенного ранее (список доступа 105).

```
router_122#debug ip packet detail 105
IP packet debugging is on (detailed) for access list 105
router_122#
00:10:01: IP: s=13.1.1.1 (Serial3/0), d=10.10.10.2 (Virtual-Access1),
g=10.10.10.2, len 100, forward

00:10:01:      ICMP type=0, code=0
! -- Пакет ICMP из 13.1.1.1 в 10.10.10.2.
! -- Данный пакет отображается, так как он соответствует
! -- требованиям в отношении источника и назначения в списке доступа 105

00:10:01: IP: s=13.1.1.1 (Serial3/0), d=10.10.10.2 (Virtual-Access1),
g=10.10.10.2, len 100, forward
00:10:01:      ICMP type=0, code=0
00:10:01: IP: s=13.1.1.1 (Serial3/0), d=10.10.10.2 (Virtual-Access1),
g=10.10.10.2, len 100, forward
00:10:01:      ICMP type=0, code=0
```

Теперь удалим быструю коммутацию на другом интерфейсе (на router_122). Это означает, что все пакеты между двумя интерфейсами находятся в сети с пакетной коммутацией (что необходимо для отладки ip-пакетов с помощью команды **debug ip packet**):

```
router_122(config)#interface serial 3/0
router_122(config-if)#no ip route-cache
router_122(config-if)#end

router_122#
00:11:57: IP: s=10.10.10.2 (Virtual-Access1), d=13.1.1.1
(Serial3/0), g=172.16.1.6, len 100, forward
00:11:57: ICMP type=8, code=0
! -- Пакет ICMP (эхо) из 10.10.10.2 в 13.1.1.1

00:11:57: IP: s=13.1.1.1 (Serial3/0), d=10.10.10.2 (Virtual-Access1),
g=10.10.10.2, len 100, forward
00:11:57: ICMP type=0, code=0
! -- Возвратный пакет ICMP (эхо-ответ) из 13.1.1.1 в 10.10.10.2

00:11:57: IP: s=10.10.10.2 (Virtual-Access1), d=13.1.1.1 (Serial3/0),
g=172.16.1.6, len 100, forward
00:11:57: ICMP type=8, code=0
00:11:57: IP: s=13.1.1.1 (Serial3/0), d=10.10.10.2 (Virtual-Access1),
g=10.10.10.2, len 100, forward
00:11:57: ICMP type=0, code=0
```

Заметьте, что выходные данные команды **debug ip packet** не отображают пакеты, которые не соответствуют критериям списка доступа. Дополнительные сведения об этом способе см. в документе Общие сведения о командах ping и traceroute.

Дополнительные сведения о создании списка доступа см. в документе Настройка списков IP-доступа.

Условно запускаемые отладки

Когда функция условно запускаемой отладки включена, маршрутизатор создает сообщения с данными об отладке для пакетов, принимаемых и отправляемых маршрутизатором через определенный интерфейс; маршрутизатор не создает выходные данные для пакетов, отправляемых и получаемых через разные интерфейсы. Дополнительные сведения об условных отладках см. в документе Условно запускаемые отладки.

Рассмотрим простую реализацию условных отладок. Рассмотрим следующий сценарий: маршрутизатор, показанный ниже (trab01) имеет два интерфейса (последовательные интерфейсы 0 и 3), использующие инкапсуляцию протокола HDLC

Используем команду **debug serial interface** для обзора сообщений об активности протокола HDLC, полученных на всех интерфейсах. Можно наблюдать сообщения об активности от обоих интерфейсов.

```
trab01#debug serial interface
Serial network interface debugging is on
trab01#
*Mar  8 09:42:34.851: Serial0: HDLC myseq 28, mineseen 28*, yourseen 41, line up
! -- поддержка активности HDLC на последовательном интерфейсе 0

*Mar  8 09:42:34.855: Serial3: HDLC myseq 26, mineseen 26*, yourseen 27, line up
! -- поддержка активности HDLC на последовательном интерфейсе 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
```

Теперь включим условную отладку для последовательного интерфейса 3. Это значит, что только выходные данные для последовательного интерфейса 3 отображаются на экране. Используйте команду **debug interface <interface_type interface_number>**.

```
trab01#debug interface serial 3
Condition 1 set
```

Используйте команду **show debug condition** для проверки активности условной отладки. Обратите внимание, что для последовательного интерфейса 3 имеется активное условие.

```
traxbol#show debug condition

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

Обратите внимание, что теперь отображаются только отладки для последовательного интерфейса 3.

```
*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
```

Используйте команду **debug interface <interface_type interface_number>**, чтобы удалить условную отладку. Рекомендуется сначала отключить все операции отладки (используя, например, команду **undebug all**) перед тем, как устранить условия. Это позволит избежать потока выходных данных отладки при удалении условия.

```
traxbol#undebug interface serial 3
This condition is the last interface condition set.
Removing all conditions may 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#
```

Теперь видно, что отображается отладка для обоих последовательных интерфейсов 0, а также для последовательного интерфейса 3.

```
*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
```



Предупреждение. Некоторые операции отладки являются условиями сами по себе. Примером является отладка асинхронного режима передачи atm. При отладке ATM необходимо указать интерфейс, для которого следует включить отладку, а не включать отладку для всех ATM-интерфейсов, указывая условие.

Следующий раздел содержит описание корректного способа ограничения отладки ATM-пакетов одним субинтерфейсом.

```
arielle-nrp2#debug atm packet interface atm 0/0/0.1
!--- Имейте в виду, что мы четко указываем субинтерфейс, который используется для отладки
```

```
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
```



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

Дополнительные сведения

- [Страницы поддержки технологии доступа](#)
- [Техническая поддержка – Cisco Systems](#)

© 1992-2010 Cisco Systems, Inc. Все права защищены.

Дата генерации PDF файла: Jan 05, 2010

<http://www.cisco.com/support/RU/customer/content/10/107661/debug.shtml>
