Содержание

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 определенные **команды отладки** и выходные данные. **Сведения о специальных командах debug см. в соответствующем документе "Справочник Cisco по командам debug".**

Выходные данные от команд EXEC **отладки**, которым дают привилегию, предоставляют диагностическую информацию, которые включают множество событий сетевых технологий, касающихся статуса протокола и активности сети в целом.

Предварительные условия

Требования

Компания Cisco рекомендует предварительно ознакомиться со следующими предметами:

- Подключение к маршрутизатору с помощью консоли, портов aux и vty
- Общие проблемы Конфигурации Cisco IOS
- Интерпретация выходных данных отладки Cisco IOS

Используемые компоненты

Настоящий документ не имеет жесткой привязки к каким-либо конкретным версиям программного обеспечения и оборудования.

Сведения, представленные в этом документе, были получены от устройств, работающих в специальной лабораторной среде. Все устройства, описанные в этом документе, были запущены с чистой (стандартной) 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 устранении конкретных проблем.

Включение режима отладки может повлиять на работу маршрутизатора при высокой загрузке внутренней сети. Следовательно, если регистрация включена, сервер доступа может периодически заморозиться, как только консольный порт перегружен с сообщениями журнала.

Прежде чем вы запустите **команду отладки**, всегда будете рассматривать выходные данные, которые эта команда будет генерировать и период времени, это может взять. Например, если у вас будет маршрутизатор с одним интерфейсом (BRI), то **debug isdn q931**, вероятно, не будет вредить системе. Но, выполнение той же отладки на AS5800 с полной конфигурацией E1 может, вероятно, генерировать такой ввод, что это может "зависнуть" и прекратить отвечать.

Перед отладкой посмотрите на свою Загрузку ЦПУ с **командой show processes cpu**. Проверьте, что вы имеете вполне достаточный ЦП в наличии перед началом отладок. См. Устранение проблем Высокой загрузки ЦП на маршрутизаторах Cisco для получения дополнительной информации о том, как обработать высокие загрузки ЦП. 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: Когда отладки работают, вы обычно не видите командную строку маршрутизатора, особенно когда отладка интенсивна. Но в большинстве случаев можно использовать **команды никакую отладку все или undebug all** для остановки отладок. Дополнительные сведения о безопасном использовании отладки см. в разделе Получение выходных данных отладки

Условные обозначения

Дополнительные сведения об условных обозначениях см. в документе Условные обозначения технических терминов Cisco.

До отладки

Помимо аспектов, упомянутых выше, следует понять, как отладка влияет на стабильность платформы. Также следует установить, к какому интерфейсу маршрутизатора необходимо подключиться. Этот раздел имеет некоторые рекомендации.

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

Если вы связаны на консоли под стандартными конфигурациями, никакая дополнительная работа не должна быть сделана. Выходные данные отладки должны быть автоматически выведены на экран. Но, удостоверьтесь**, что уровень регистрации при входе с консоли** установлен, как желаемый и что регистрация не была отключена с **командой no logging console**. См. Использование Команд отладки для получения дополнительной информации.

 **% Warning:** Чрезмерные отладки к консольному порту маршрутизатора могут заставить его "зависать". 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 регистрация включена на консольном порте. Поэтому порт консоли всегда обрабатывает результаты отладки, даже если на самом деле для сбора результатов используется другой порт или метод (например, AUX, VTY или буфер). Следовательно, Cisco рекомендует, чтобы под обычными рабочими состояниями вам включили **команду no logging console** в любом случае и используете другие методы для получения отладок. **В ситуациях, в которых необходимо использовать консоль, временно включите вход в систему через консоль с помощью команды logging console.**

Порт AUX

Если вы связаны через Вспомогательный порт, введите **команду terminal monitor**. **Убедитесь также, что команда no logging on не активирована на маршрутизаторе.**

Примечание: При использовании Порта AUX, чтобы контролировать маршрутизатор, иметь в виду это, когда перезагрузки маршрутизатора, Порт AUX не отображает ввод загрузочной последовательности. Соединитесь с консольным портом для просмотра последовательности загрузки.

Порты VTY

Если вы связаны через Вспомогательный порт или через 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 logging buffered. Это - полный синтаксис этой команды:

logging bufferedno 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, не 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: Удостоверьтесь, что достаточно памяти доступно в коробке прежде, чем ввести размер буфера. Используйте **команду show proc mem** Cisco IOS для наблюдения 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 syslog-сервере, используйте команду logging router configuration. Полный синтаксис этой команды:

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.

Для получения дополнительной информации об устанавливании сервера системного журнала обратитесь к Использованию Команд отладки.

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.** Для получения дополнительной информации обратитесь к Получению Вывода текста от Гипертерминала. 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 MMM 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 вы включаете **debug x25 all** на коробке, которая имеет 200 VC, и выходные данные зарегистрированы к буферу (использование **команд никакой консоли регистрации и logging buffered**), метка времени, отображенная в выходных данных отладки (в буфере), не могла бы быть точным временем, когда пакет проходит 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 undebug 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 пространным 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 – это 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 учтите, что маршрутизатор выполняет по умолчанию быструю коммутацию или, при соответствующей настройке, коммутацию CEF. 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 в поле возникнет зависание или потеря подключения к одноранговому узлу.

- 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.

Давайте рассмотрим примерный сценарий:

Список доступа, конфигурированные 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.

Удалите быструю коммутацию только на одном интерфейсе на router_122. Это означает, что можно только видеть отладки для пакетов, которые предназначены для того интерфейса, как замечено с точки зрения IOS, перехватывающего пакет. От отладок такие пакеты появляются с "d =". Так как вы еще не выключили быструю коммутацию на другом интерфейсе, возвращаемый пакет не подвергается **debug ip packet**. Эти выходные данные показывают, как можно отключить быструю коммутацию:

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

Необходимо теперь активировать **debug ip packet** с access-list, определенным более ранний (access-list 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 ! -- ICMP packet from 13.1.1.1 to 10.10.10.2.
! -- This packet is displayed because it matches the ! -- source and destination requirements in
access list 10500:10:01: 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):**

```
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
! -- ICMP packet (echo) from 10.10.10.2 to 13.1.1.100:11:57: 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! -
- ICMP return packet (echo-reply) from 13.1.1.1 to 10.10.10.200:11:57: 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. Для некоторых дополнительных сведений об этой процедуре обратитесь к Пониманию Команд эхо-запроса и Traceroute.

Для получения дополнительной информации о том, как создать access-lists, обратитесь к Standard IP Access List Logging.

Отладка, запускаемая по условию

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. Для получения информации относительно использования для Условных отладок, обратитесь к Условно Инициированной Отладке.

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

Можно использовать обычную **команду debug serial interface** для наблюдения сообщений проверки активности HDLC, полученных относительно всех интерфейсов. Можно наблюдать пакеты Keepalive относительно обоих интерфейсов.

```
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
```

Включите условные отладки для последовательного интерфейса 3. Это означает, что только отлаживает для interface serial 3, отображены. Используйте команду *<interface_type interface_number>* **debug interface**.

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

Используйте **команду show debug condition**, чтобы проверить, что отладка condional активна. 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 интерфейса serial 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
```

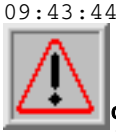
Используйте команду *<interface_type interface_number>* **интерфейса неотладки** для удаления условной отладки. Рекомендуется выключить отладки (например, с помощью undebug all) перед удалением условного триггера. Это позволит избежать потока выходных данных отладки при удалении условия.

```
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
```

Можно теперь заметить, что отображена отладка для обоих interface serial 0, а также последовательных 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
```

 **% Warning:** Некоторые операции отладки являются условным выражением собой. Примером является отладка асинхронного режима передачи atm. При отладке ATM необходимо явно указать интерфейс, для которого следует включить отладку, а не включать отладку для всех ATM-интерфейсов, указывая условие.

Этот раздел показывает правильно для ограничения отладки пакета ATM одним подинтерфейсом:

```
arielle-nrp2#debug atm packet interface atm 0/0/0.1!--- Note that you explicitly specify the
sub-interface to be used for debuggingATM 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#
```

При попытке включения отладки ATM на всех интерфейсах с помощью команды atm debugging (с примененным условием), маршрутизатор может "зависнуть", если у него имеется большое число субинтерфейсов ATM. Показан пример неправильного метода для отладки atm.

В этом случае можно увидеть, что условие применено, но не имеет эффекта. Можно все еще видеть пакет от другого интерфейса. В этом лабораторном сценарии у вас есть только два интерфейса и очень мало трафика. Если количество интерфейсов высоко, то выходные данные отладки для всех интерфейсов являются чрезвычайном высоким, и это может заставить маршрутизатор "зависать".

```
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
```

Дополнительные сведения

- Набор и поддержка технологии доступа
- Cisco Systems – техническая поддержка и документация