

Функция трассировки пакетов канала передачи данных XE IOS

Содержание

[Введение](#)

[Предварительные условия](#)

[Требования](#)

[Используемые компоненты](#)

[Ссылочная топология](#)

[Пакетное отслеживание в использовании](#)

[Краткое руководство по началу работы](#)

[Включите условные отладки платформы](#)

[Включите трассировку пакетов](#)

[Выходное ограничение условия с трассировками пакетов](#)

[Отобразите результаты трассировки пакетов](#)

[Трассировка FIA](#)

[Отобразите результаты трассировки пакетов](#)

[Проверьте FIA, связанный с интерфейсом](#)

[Формируйте дампы отслеженных пакетов](#)

[Трассировка отбрасывания](#)

[Сценарий трассировки отбрасывания в качестве примера](#)

[Введите и плавите на плоскодонке трассировки](#)

[Примеры трассировки пакетов](#)

[Пример трассировки пакетов - NAT](#)

[Пример трассировки пакетов - VPN](#)

[Влияние на производительность](#)

[Дополнительные сведения](#)

Введение

Этот документ описывает, как выполнить пакетное отслеживание канала передачи данных для программного обеспечения Cisco IOS®-XE через функцию Трассировки пакетов.

Для определения проблем, таких как неверная конфигурация, перегрузка емкости, или даже обычная ошибка в программном обеспечении при устранении проблем, необходимо понять то, что происходит с пакетом в системе. Функция Трассировки пакетов Cisco IOS XE обращается к этой потребности. Это предоставляет полевой надежный метод, который используется для учета и для получения по пакетным подробных данных обработки на основе класса определяемых пользователем условий.

Предварительные условия

Требования

Cisco рекомендует ознакомиться с функцией трассировки пакетов, которая доступна в Версиях Cisco IOS XE 3.10 и позже, а также во всех платформах, которые выполняют программное обеспечение Cisco IOS XE, такое как Маршрутизаторы агрегации Серии Cisco 1000 (ASR1K), маршрутизатор (CSR1000v) Облачных сервисов Cisco 1000V Series и Маршрутизатор ISR Cisco 4451-X Series (ISR4451-X).

Используемые компоненты

Сведения, содержащиеся в данном документе, касаются следующих версий программного обеспечения и оборудования:

- Версии Программного обеспечения Cisco IOS XE 3.10S (15.3 (3) S) и позже
- ASR1K

Данные для документа были получены в специально созданных лабораторных условиях. Все устройства, описанные в данном документе, были запущены с конфигурацией по умолчанию. Если ваша сеть является оперативной, удостоверьтесь, что вы понимаете потенциальное воздействие любой используемой команды.

Ссылочная топология

Эта схема иллюстрирует топологию, которая используется для примеров, которые описаны в этом документе:



Пакетное отслеживание в использовании

Для иллюстрирования использования функции трассировки пакетов пример, который используется всюду по этому разделу, описывает трассировку трафика Протокола ICMP от локальной рабочей станции 172.16.10.2 (позади ASR1K) к удаленному хосту 172.16.20.2 (направление доступа для ASR1K в интерфейсе Gig0/0/1).

Можно отследить пакеты на ASR1K с этими двумя шагами:

1. Включите условные отладки платформы для выбора пакетов или трафика, который вы хотите отследить на ASR1K.
2. Включите трассировку пакетов платформы (трассировка пути или трассировка Массива вызова функции (FIA)).

Краткое руководство по началу работы

Если вы уже знакомы с содержанием этого документа и хотите раздел для ознакомления в CLI, вот Краткое руководство по началу работы. Это только несколько примеров, чтобы

проиллюстрировать использование программного средства, обратитесь последующие разделы, которые обсуждают синтаксисы подробно и гарантируют, что вы используете конфигурацию, которая является соответствующей вашему требованию.

1. Настройте условия платформы:

```
debug platform condition ipv4 10.0.0.1/32 both --> matches in and out packets with source or destination as 10.0.0.1/32
```

```
debug platform condition ipv4 access-list 198 egress --> (Ensure access-list 198 is defined prior to configuring this command) - matches egress packets corresponding to access-list 198
```

```
debug platform condition interface gig 0/0/0 ingress --> matches all ingress packets on interface gig 0/0/0
```

```
debug platform condition mpls 10 1 ingress --> matches MPLS packets with top ingress label 10
```

```
debug platform condition ingress --> matches all ingress packets on all interfaces (use cautiously)
```

После того, как условие платформы настроено, запустите условия платформы с этой команды CLI:

```
debug platform condition start
```

2. Настройте трассировку пакетов:

```
debug platform packet-trace packet 1024 -> basic path-trace, and automatically stops tracing packets after 1024 packets. You can use "circular" option if needed
debug platform packet-trace packet 1024 fia-trace -> enables detailed fia trace, stops tracing packets after 1024 packets
debug platform packet-trace drop [code <dropcode>] -> if you want to trace/capture only packets that are dropped. Refer to Drop Trace section for more details.
```

3. Затем, включите трассировку пакетов платформы:

```
debug platform packet-trace enable
```

Вот команды, чтобы очистить буфер и перезагрузить трассировку пакетов:

```
no debug platform packet-trace enable --> disable the packet trace, this needs to be done before you clear the buffer
```

```
clear platform packet-trace statistics --> clear the packet trace buffer
```

```
debug platform packet-trace enable --> reenables the packet trace
```

Команда для очистки и условий платформы и конфигурации трассировки пакетов:

```
clear platform condition all --> clears both platform conditions and the packet trace configuration
```

Команды "show"

Проверьте условие платформы и конфигурацию трассировки пакетов после применения предыдущих команд, чтобы гарантировать, что вы имеете то, в чем вы нуждаетесь.

```
show platform conditions --> shows the platform conditions configured
```

```
show platform packet-trace configuration --> shows the packet-trace configurations
```

`show debugging` --> this will show both platform conditions and platform packet-trace configured

Вот команды для проверки отслеженных/захваченных пакетов:

`show platform packet-trace statistics` --> statistics of packets traced

`show platform packet-trace summary` --> summary of all the packets traced, with input and output interfaces, processing result and reason. `show platform packet-trace packet 12` -> Tracing the 12th packet, with complete path trace or FIA trace details.

Включите условные отладки платформы

Функция Трассировки пакетов полагается на инфраструктуру условной отладки для определения пакетов, которые будут отслежены. Инфраструктура условной отладки предоставляет способность к трафику фильтрации на основе:

- Протокол
- IP-адрес и маска
- Список контроля доступа (ACL)
- Interface
- Направление трафика (вход или выход)

Где и когда фильтры применены к пакету, эти условия определяют.

Для трафика, который используется в данном примере, включите условные отладки платформы в направлении доступа для пакетов ICMP от 172.16.10.2 до 172.16.20.2. Другими словами, выберите трафик, который вы хотите отследить. Существуют различные варианты, которые можно использовать для выбора этого трафика.

```
ASR1000#debug platform condition ?
egress Egress only debug
feature For a specific feature
ingress Ingress only debug
interface Set interface for conditional debug
ipv4 Debug IPv4 conditions
ipv6 Debug IPv6 conditions
start Start conditional debug
stop Stop conditional debug
```

В данном примере access-list используется для определения условия, как показано здесь:

```
ASR1000#show access-list 150
Extended IP access list 150
10 permit icmp host 172.16.10.2 host 172.16.20.2
ASR1000#debug platform condition interface gig 0/0/1 ipv4
access-list 150 ingress
```

Для начала условной отладки введите эту команду:

```
ASR1000#debug platform condition start
```

Примечание: Чтобы остановить или отключить инфраструктуру условной отладки, войдите, условие платформы отладки останавливают команду.

Для просмотра фильтров условной отладки, которые настроены, вводят эту команду:

```
ASR1000#show platform conditions
```

```
Conditional Debug Global State: Start
```

Conditions Direction

-----|-----
GigabitEthernet0/0/1 & IPV4 ACL [150] ingress

Feature Condition Format Value

-----|-----|-----
ASR1000#

Таким образом, эта конфигурация была применена к настоящему времени:

```
access-list 150 permit icmp host 172.16.10.2 host 172.16.20.2
debug platform condition interface gig 0/0/1 ipv4 access-list 150 ingress
debug platform condition start
```

Включите трассировку пакетов

Примечание: В этом разделе описываются пакет и параметры копирования подробно, и другие опции описаны позже в документе.

Трассировки пакетов поддерживаются и на медосмотре и на логических интерфейсах, таких как Туннель или Интерфейсы виртуального доступа.

Вот синтаксис CLI трассировки пакетов:

```
ASR1000#debug platform packet-trace ?
copy Copy packet data
drop Trace drops only
enable Enable packet trace
inject Trace injects only
packet Packet count
punt Trace punts only
debug platform packet-trace packet <pkt-size/pkt-num> [fia-trace | summary-
only]
[circular] [data-size <data-size>]
```

Вот описания для ключевых слов этой команды:

- **цифра PKT** - Пакетный Номер задает максимальное число пакетов, которые поддержаны когда-то.
- **summary-only** - Это указывает, что только перехвачены сводные данные. По умолчанию должен перехватить оба сводных данные и данные пути характеристики.
- **fia-трассировка** - Это дополнительно выполняет трассировку FIA в дополнение к информации о данных пути.
- **размер данных** - Это позволяет вам задавать размер буфера данных пути от 2,048 до 16,384 байтов. По умолчанию составляет **2,048 байтов**.

```
debug platform packet-trace copy packet {in | out | both} [L2 | L3 | L4]
[size <num-bytes>]
```

Вот описания для ключевых слов этой команды:

- **входящий/исходящий** - Это задает направление потока пакетов, который будет скопирован - вход и/или выход.

- **L2/L3/L4** - Это позволяет вам задавать местоположение, которое запускает копия пакета. Уровень 2 (L2) является расположением по умолчанию.
- **размер** - Это позволяет вам задавать максимальное число октетов, которые скопированы. По умолчанию является 64 октетами.

Для данного примера это команды, используемые для включения трассировки пакетов для трафика, который выбран инфраструктурой условной отладки:

```
ASR1000#debug platform packet-trace packet 16
ASR1000#debug platform packet-trace enable
```

Для рассмотрения конфигурации трассировки пакетов введите эту команду:

```
ASR1000#show platform packet-trace configuration
debug platform packet-trace enable
debug platform packet-trace packet 16 data-size 2048
```

Можно также ввести команду **show debugging** для просмотра и условных отладок платформы и конфигураций трассировки пакетов:

```
ASR1000# show debugging
IOSXE Conditional Debug Configs:
```

```
Conditional Debug Global State: Start
```

```
Conditions
Direction
```

```
-----|-----
GigabitEthernet0/0/1 & IPV4 ACL [150] ingress
```

```
...
IOSXE Packet Tracing Configs:
```

```
Feature Condition Format Value
```

```
-----|-----|-----
```

```
Feature Type Submode Level
```

```
-----|-----|-----|-----
```

```
IOSXE Packet Tracing Configs:
```

```
debug platform packet-trace enable
debug platform packet-trace packet 16 data-size 2048
```

Примечание: Введите **ясное условие платформы вся** команда для очистки всего debug condition платформы и конфигураций трассировки пакетов и данных.

Таким образом, эти данные о конфигурации использовались к настоящему времени для включения трассировки пакетов:

```
debug platform packet-trace packet 16
debug platform packet-trace enable
```

Выходное ограничение условия с трассировками пакетов

Условия определяют условные фильтры и когда они применены к пакету. Например, **интерфейс условия платформы отладки g0/0/0 выход** означает, что пакет определен как соответствие, когда это достигает выходных данных FIA на интерфейсе g0/0/0, таким образом, пропущен любой пакет, обрабатывающий, который имеет место от входа до той точки.

Примечание: Cisco настоятельно рекомендует, чтобы вы использовали входные условия для трассировки пакетов для получения самых завершенных и значимых возможных данных. Выходные условия могут использоваться, но знать об ограничениях.

Отобразите результаты трассировки пакетов

Примечание: Этот раздел предполагает, что включена трассировка пути.

Три определенных уровня контроля предоставлены трассировкой пакетов:

- Учет
- Попакетная сводка
- Попакетные данные пути

Когда пять пакетов запроса ICMP переданы от 172.16.10.2 до 172.16.20.2, эти команды могут использоваться для просмотра результатов трассировки пакетов:

```
ASR1000#show platform packet-trace statistics
```

```
Packets Traced: 5
```

```
Ingress 5
```

```
Inject 0
```

```
Forward 5
```

```
Punt 0
```

```
Drop 0
```

```
Consume 0
```

```
ASR1000#show platform packet-trace summary
```

```
Pkt Input Output State Reason
```

```
0 Gi0/0/1 Gi0/0/0 FWD
```

```
1 Gi0/0/1 Gi0/0/0 FWD
```

```
2 Gi0/0/1 Gi0/0/0 FWD
```

```
3 Gi0/0/1 Gi0/0/0 FWD
```

```
4 Gi0/0/1 Gi0/0/0 FWD
```

```
ASR1000#show platform packet-trace packet 0
```

```
Packet: 0 CBUG ID: 4
```

```
Summary
```

```
Input : GigabitEthernet0/0/1
```

```
Output : GigabitEthernet0/0/0
```

```
State : FWD
```

```
Timestamp
```

```
Start : 1819281992118 ns (05/17/2014 06:42:01.207240 UTC)
```

```
Stop : 1819282095121 ns (05/17/2014 06:42:01.207343 UTC)
```

```
Path Trace
```

```
Feature: IPV4
```

```
Source : 172.16.10.2
```

```
Destination : 172.16.20.2
```

```
Protocol : 1 (ICMP)
```

```
ASR1000#
```

Примечание: Третья команда предоставляет пример, который иллюстрирует, как просмотреть трассировку пакетов для каждого пакета. В данном примере показывают первый отслеженный пакет.

От этих выходных данных вы видите, что отслежены пять пакетов и что можно просмотреть входной интерфейс, выходной интерфейс, состояние и трассировку пути.

Состояние	Комментарий
FWD	Пакет планируется/помещается в очередь для доставки, чтобы быть переданным следующему переходу через исходящий интерфейс.
ИЗБЫТОЧНОЕ НАПРАВЛЕНИЕ	Пакет плывется на плоскодонке с Процессора переадресации (FP) на Процессор маршрута (RP) (уровень управления).
ОТБРАСЫВАНИЕ	Пакет отброшен на FP. Выполните трассировку FIA, используйте глобальные счетчики сбросов или используйте отладки канала передачи данных для обнаружения большего количества подробных данных по причинам отбрасывания.
CONS	Пакет использован во время пакетного процесса, такой как во время запроса Функции проверки связности ICMP ping или крипто-пакетов.

Вход и вводит счетчики в выходных данных statistics трассировки пакетов, соответствуют пакетам, которые входят через внешний интерфейс и пакеты, которые замечены, как введено по уровню управления, соответственно.

Трассировка FIA

FIA держит список функций, которые выполняются последовательно Пакетными модулями процессора (PPE) в процессоре Quantum Flow (QFP), когда пакет передан или вход или выход. Функции основываются на данных о конфигурации, которые применены на машину. Таким образом трассировка FIA помогает понимать поток пакета через систему, поскольку обработан пакет.

Необходимо применить эти данные о конфигурации для включения трассировки пакетов с FIA:

```
ASR1000#debug platform packet-trace packet 16 fia-trace
```

Отобразите результаты трассировки пакетов

Примечание: Этот раздел предполагает, что разрешена трассировка FIA. Кроме того, когда вы добавляете или модифицируете команды трассировки текущего пакета, буферизированные подробные данные трассировки пакетов очищены, таким образом, необходимо передать некоторый трафик снова так, чтобы можно было отследить его.

Передайте пять пакетов ICMP от 172.16.10.2 до 172.16.20.2 после ввода команды, которая используется для разрешения трассировки FIA, как описано в предыдущем разделе.

```
ASR1000#show platform packet-trace summary
```

Pkt	Input	Output	State	Reason
0	Gi0/0/1	Gi0/0/0	FWD	
1	Gi0/0/1	Gi0/0/0	FWD	
2	Gi0/0/1	Gi0/0/0	FWD	
3	Gi0/0/1	Gi0/0/0	FWD	
4	Gi0/0/1	Gi0/0/0	FWD	

```
ASR1000#show platform packet-trace packet 0
```

```
Packet: 0          CBUG ID: 9
```

```
Summary
```

```
Input      : GigabitEthernet0/0/1
```



```
Output   : GigabitEthernet0/0/0
State    : FWD
Timestamp
  Start   : 1819281992118 ns (05/17/2014 06:42:01.207240 UTC)
  Stop    : 1819282095121 ns (05/17/2014 06:42:01.207343 UTC)
```

Path Trace

Feature: IPV4

```
Source      : 172.16.10.2
Destination : 172.16.20.2
Protocol    : 1 (ICMP)
```

Feature: FIA_TRACE

```
Entry      : 0x8059dbe8 - DEBUG_COND_INPUT_PKT
Timestamp  : 3685243309297
```

Feature: FIA_TRACE

```
Entry      : 0x82011a00 - IPV4_INPUT_DST_LOOKUP_CONSUME
Timestamp  : 3685243311450
```

Feature: FIA_TRACE

```
Entry      : 0x82000170 - IPV4_INPUT_FOR_US_MARTIAN
Timestamp  : 3685243312427
```

Feature: FIA_TRACE

```
Entry      : 0x82004b68 - IPV4_OUTPUT_LOOKUP_PROCESS
Timestamp  : 3685243313230
```

Feature: FIA_TRACE

```
Entry      : 0x8034f210 - IPV4_INPUT_IPOPTIONS_PROCESS
Timestamp  : 3685243315033
```

Feature: FIA_TRACE

```
Entry      : 0x82013200 - IPV4_OUTPUT_GOTO_OUTPUT_FEATURE
Timestamp  : 3685243315787
```

Feature: FIA_TRACE

```
Entry      : 0x80321450 - IPV4_VFR_REFRAG
Timestamp  : 3685243316980
```

Feature: FIA_TRACE

```
Entry      : 0x82014700 - IPV6_INPUT_L2_REWRITE
Timestamp  : 3685243317713
```

Feature: FIA_TRACE

```
Entry      : 0x82000080 - IPV4_OUTPUT_FRAG
Timestamp  : 3685243319223
```

Feature: FIA_TRACE

```
Entry      : 0x8200e500 - IPV4_OUTPUT_DROP_POLICY
Timestamp  : 3685243319950
```

Feature: FIA_TRACE

```
Entry      : 0x8059aff4 - PACTRAC_OUTPUT_STATS
Timestamp  : 3685243323603
```

Feature: FIA_TRACE

```
Entry      : 0x82016100 - MARMOT_SPA_D_TRANSMIT_PKT
Timestamp  : 3685243326183
```

ASR1000#

Проверьте FIA, связанный с интерфейсом

При включении условных отладок платформы это добавлено к FIA как функция. В зависимости от местоположения, что это добавлено к списку, вы, возможно, должны были бы отрегулировать свои условия платформы, такой как тогда, когда вы отслеживаете пакеты постепсар и предварительный епсар.

Эти выходные данные показывают заказ функций в FIA для условной отладки платформы, которая включена в направлении доступа:

```
ASR1000#show platform hardware qfp active interface if-name GigabitEthernet 0/0/1
```

General interface information

Interface Name: GigabitEthernet0/0/1

Interface state: VALID

Platform interface handle: 10

QFP interface handle: 8

Rx uidb: 1021

Tx uidb: 131064

Channel: 16

Interface Relationships

BGPPA/QPPB interface configuration information

Ingress: BGPPA/QPPB not configured. flags: 0000

Egress : BGPPA not configured. flags: 0000

ipv4_input enabled.

ipv4_output enabled.

layer2_input enabled.

layer2_output enabled.

ess_ac_input enabled.

Features Bound to Interface:

2 GIC FIA state

48 PUNT INJECT DB

39 SPA/Marmot server

40 ethernet

1 IFM

31 icmp_svr

33 ipfrag_svr

34 ipreass_svr

36 ipvfr_svr

37 ipv6vfr_svr

12 CPP IPSEC

Protocol 0 - ipv4_input

FIA handle - CP:0x108d99cc DP:0x8070f400

IPV4_INPUT_DST_LOOKUP_ISSUE (M)

IPV4_INPUT_ARL_SANITY (M)

CBUG_INPUT_FIA

DEBUG_COND_INPUT_PKT

IPV4_INPUT_DST_LOOKUP_CONSUME (M)

IPV4_INPUT_FOR_US_MARTIAN (M)

IPV4_INPUT_IPSEC_CLASSIFY

IPV4_INPUT_IPSEC_COPROC_PROCESS

IPV4_INPUT_IPSEC_RERUN_JUMP

IPV4_INPUT_LOOKUP_PROCESS (M)

IPV4_INPUT_IPOPTIONS_PROCESS (M)

IPV4_INPUT_GOTO_OUTPUT_FEATURE (M)

Protocol 1 - ipv4_output

FIA handle - CP:0x108d9a34 DP:0x8070eb00

IPV4_OUTPUT_VFR

MC_OUTPUT_GEN_RECYCLE (D)

IPV4_VFR_REFRAG (M)

IPV4_OUTPUT_IPSEC_CLASSIFY

IPV4_OUTPUT_IPSEC_COPROC_PROCESS

IPV4_OUTPUT_IPSEC_RERUN_JUMP

IPV4_OUTPUT_L2_REWRITE (M)

IPV4_OUTPUT_FRAG (M)

IPV4_OUTPUT_DROP_POLICY (M)

PACTRAC_OUTPUT_STATS

MARMOT_SPA_D_TRANSMIT_PKT

DEF_IF_DROP_FIA (M)

Protocol 8 - layer2_input

FIA handle - CP:0x108d9bd4 DP:0x8070c700

LAYER2_INPUT_SIA (M)

CBUG_INPUT_FIA

DEBUG_COND_INPUT_PKT

```
LAYER2_INPUT_LOOKUP_PROCESS (M)
LAYER2_INPUT_GOTO_OUTPUT_FEATURE (M)
Protocol 9 - layer2_output
FIA handle - CP:0x108d9658 DP:0x80714080
LAYER2_OUTPUT_SERVICEWIRE (M)
LAYER2_OUTPUT_DROP_POLICY (M)
PACTRAC_OUTPUT_STATS
MARMOT_SPA_D_TRANSMIT_PKT
DEF_IF_DROP_FIA (M)
Protocol 14 - ess_ac_input
FIA handle - CP:0x108d9ba0 DP:0x8070cb80
PPPOE_GET_SESSION
ESS_ENTER_SWITCHING
PPPOE_HANDLE_UNCLASSIFIED_SESSION
DEF_IF_DROP_FIA (M)
```

```
QfpEth Physical Information
DPS Addr: 0x11215eb8
Submap Table Addr: 0x00000000
VLAN Ethertype: 0x8100
QOS Mode: Per Link
```

```
ASR1000#
```

Примечание: CBUG_INPUT_FIA и DEBUG_COND_INPUT_PKT соответствуют функциям условной отладки, которые настроены на маршрутизаторе.

Формируйте дампы отслеженных пакетов

Можно скопировать и формировать дампы пакетов, поскольку они отслежены, как этот раздел описывает. Данный пример показывает, как скопировать максимум 2,048 байтов пакетов в направлении доступа (172.16.10.2 к 172.16.20.2).

Вот дополнительная команда, которая необходима:

```
ASR1000#debug platform packet-trace copy packet input size 2048
```

Примечание: Размер пакета, который скопирован, находится в диапазоне 16 - 2,048 байтов.

Введите эту команду для формирования дампа скопированных пакетов:

```
ASR1000#show platform packet-trace packet 0
Packet: 0 CBUG ID: 14
Summary
Input : GigabitEthernet0/0/1
Output : GigabitEthernet0/0/0
State : FWD
Timestamp
  Start : 1819281992118 ns (05/17/2014 06:40:01.207240 UTC)
  Stop : 1819282095121 ns (05/17/2014 06:40:01.207343 UTC)
Path Trace
Feature: IPV4
Source : 172.16.10.2
Destination : 172.16.20.2
Protocol : 1 (ICMP)
Feature: FIA_TRACE
Entry : 0x8059dbe8 - DEBUG_COND_INPUT_PKT
Timestamp : 4458180580929
```

<some content excluded>

Feature: FIA_TRACE

Entry : 0x82016100 - MARMOT_SPA_D_TRANSMIT_PKT

Timestamp : 4458180593896

Packet Copy In

a4934c8e 33020023 33231379 08004500 00640160 0000ff01 5f16ac10 0201ac10
01010800 1fd40024 00000000 000184d0 d980abcd abcdabcd abcdabcd abcdabcd
abcdabcd abcdabcd abcdabcd abcdabcd abcdabcd abcdabcd abcdabcd abcdabcd
abcdabcd abcdabcd abcdabcd abcdabcd abcd

ASR1000#

Трассировка отбрасывания

Трассировка отбрасывания доступна в Версиях программного обеспечения Cisco IOS XE 3.11 и позже. Это включает трассировку пакетов только для отброшенных пакетов. Вот некоторое выделение функции:

- Это дополнительно позволяет вам задавать задержку пакетов для определенного кода отбрасывания.
- Это может использоваться без глобальных или интерфейсных условий для получения событий отбрасывания.
- Перехват события отбрасывания означает, что только само отбрасывание отслежено, не жизнь пакета. Однако это все еще позволяет вам перехватывать сводные данные, данные кортежа и пакет, чтобы помочь совершенствовать условия или дать представления о следующем шаге отладки.

Вот синтаксис команды, который используется для включения трассировок пакетов типа отбрасывания:

```
debug platform packet-trace drop [code <code-num>]
```

Код отбрасывания совпадает с ID отбрасывания, как сообщается в **show platform hardware qfp активные подробные выходные данные команды отбрасывания статистики**:

```
ASR1000#show platform hardware qfp active statistics drop detail
```

ID	Global Drop Stats	Packets	Octets
60	IpTtlExceeded	3	126
8	Ipv4Acl	32	3432

Сценарий трассировки отбрасывания в качестве примера

Примените этот ACL на Концерт 0/0/0 интерфейс ASR1K для отбрасывания трафика с 172.16.10.2 до 172.16.20.2:

```
ASR1000#show platform hardware qfp active statistics drop detail
```

ID	Global Drop Stats	Packets	Octets
60	IpTtlExceeded	3	126
8	Ipv4Acl	32	3432

С ACL на месте, который отбрасывает трафик от локального хоста до удаленного хоста, применяют эту конфигурацию трассировки отбрасывания:

```
debug platform condition interface Gig 0/0/1 ingress
debug platform condition start
debug platform packet-trace packet 1024 fia-trace
debug platform packet-trace drop
debug platform packet-trace enable
```

Передайте пять пакетов запроса ICMP от 172.16.10.2 до 172.16.20.2. Трассировка отбрасывания перехватывает эти пакеты, которые отброшены ACL, как показано:

```
ASR1000#show platform packet-trace statistics
```

```
Packets Summary
Matched 5
Traced 5
Packets Received
Ingress 5
Inject 0
Packets Processed
Forward 0
Punt 0
Drop      5
Count Code Cause
5 8 Ipv4Acl
Consume 0
```

```
ASR1000#show platform packet-trace summary
```

```
Pkt Input Output State Reason
0 Gi0/0/1 Gi0/0/0 DROP 8 (Ipv4Acl)
1 Gi0/0/1 Gi0/0/0 DROP 8 (Ipv4Acl)
2 Gi0/0/1 Gi0/0/0 DROP 8 (Ipv4Acl)
3 Gi0/0/1 Gi0/0/0 DROP 8 (Ipv4Acl)
4 Gi0/0/1 Gi0/0/0 DROP 8 (Ipv4Acl)
```

```
ASR1K#debug platform condition stop
```

```
ASR1K#show platform packet-trace packet 0
```

```
Packet: 0 CBUG ID: 140
Summary
Input : GigabitEthernet0/0/1
Output : GigabitEthernet0/0/0
State   : DROP 8 (Ipv4Acl)
Timestamp
Start : 1819281992118 ns (05/17/2014 06:42:01.207240 UTC)
Stop : 1819282095121 ns (05/17/2014 06:42:01.207343 UTC)
Path Trace
Feature: IPV4
Source : 172.16.10.2
Destination : 172.16.20.2
Protocol : 1 (ICMP)
Feature: FIA_TRACE
Entry : 0x806c7eac - DEBUG_COND_INPUT_PKT
Lapsed time: 1031 ns
Feature: FIA_TRACE
Entry : 0x82011c00 - IPV4_INPUT_DST_LOOKUP_CONSUME
Lapsed time: 657 ns
Feature: FIA_TRACE
Entry : 0x806a2698 - IPV4_INPUT_ACL
Lapsed time: 2773 ns
Feature: FIA_TRACE
Entry : 0x82000170 - IPV4_INPUT_FOR_US_MARTIAN
Lapsed time: 1013 ns
Feature: FIA_TRACE
Entry : 0x82004500 - IPV4_OUTPUT_LOOKUP_PROCESS
```

```
Lapsed time: 2951 ns
Feature: FIA_TRACE
Entry : 0x8041771c - IPV4_INPUT_IPOPTIONS_PROCESS
Lapsed time: 373 ns
Feature: FIA_TRACE
Entry : 0x82013400 - MPLS_INPUT_GOTO_OUTPUT_FEATURE
Lapsed time: 2097 ns
Feature: FIA_TRACE
Entry : 0x803c60b8 - IPV4_MC_OUTPUT_VFR_REFRAG
Lapsed time: 373 ns
Feature: FIA_TRACE
Entry : 0x806db148 - OUTPUT_DROP
Lapsed time: 1297 ns
Feature: FIA_TRACE
Entry : 0x806a0c98 - IPV4_OUTPUT_ACL
Lapsed time: 78382 ns
```

```
ASR1000#
```

Введите и плывите на плоскодонке трассировки

Функция трассировки введения и пакета избыточного направления была добавлена в Версиях программного обеспечения Cisco IOS XE 3.12 и позже для отслеживания избыточного направления (пакеты, которые получены на FP, которые плывутся на плоскодонке к уровню управления), и введите (пакеты, которые введены к FP от уровня управления), пакеты.

Примечание: Трассировка избыточного направления может работать без глобальных или интерфейсных условий, точно так же, как трассировка отбрасывания. Однако условия должны быть определены для вводить трассировки для работы.

Вот пример избыточного направления, и введите трассировку пакетов, когда вы пропинговываете от ASR1K до соседнего маршрутизатора:

```
ASR1000#debug platform condition ipv4 172.16.10.2/32 both
ASR1000#debug platform condition start
ASR1000#debug platform packet-trace punt
ASR1000#debug platform packet-trace inject
ASR1000#debug platform packet-trace packet 16
ASR1000#debug platform packet-trace enable
ASR1000#
ASR1000#ping 172.16.10.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.10.2, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 14/14/15 ms
ASR1000#
```

Теперь можно проверить избыточное направление и ввести результаты трассировки:

```
ASR1000#show platform packet-trace summary
Pkt Input Output State Reason
0 INJ.2 Gi0/0/1 FWD
1 Gi0/0/1 internal0/0/rp:0 PUNT 11 (For-us data)
2 INJ.2 Gi0/0/1 FWD
3 Gi0/0/1 internal0/0/rp:0 PUNT 11 (For-us data)
4 INJ.2 Gi0/0/1 FWD
5 Gi0/0/1 internal0/0/rp:0 PUNT 11 (For-us data)
6 INJ.2 Gi0/0/1 FWD
7 Gi0/0/1 internal0/0/rp:0 PUNT 11 (For-us data)
```

```
8 INJ.2 Gi0/0/1 FWD
9 Gi0/0/1 internal0/0/rp:0 PUNT 11 (For-us data)
```

```
ASR1000#show platform packet-trace packet 0
```

```
Packet: 0 CBUG ID: 120
```

```
Summary
```

```
Input      : INJ.2
```

```
Output : GigabitEthernet0/0/1
```

```
State : FWD
```

```
Timestamp
```

```
Start : 115612780360228 ns (05/29/2014 15:02:55.467987 UTC)
```

```
Stop : 115612780380931 ns (05/29/2014 15:02:55.468008 UTC)
```

```
Path Trace
```

```
Feature: IPV4
```

```
Source : 172.16.10.1
```

```
Destination : 172.16.10.2
```

```
Protocol : 1 (ICMP)
```

```
ASR1000#
```

```
ASR1000#show platform packet-trace packet 1
```

```
Packet: 1 CBUG ID: 121
```

```
Summary
```

```
Input : GigabitEthernet0/0/1
```

```
Output : internal0/0/rp:0
```

```
State      : PUNT 11 (For-us data)
```

```
Timestamp
```

```
Start : 115612781060418 ns (05/29/2014 15:02:55.468687 UTC)
```

```
Stop : 115612781120041 ns (05/29/2014 15:02:55.468747 UTC)
```

```
Path Trace
```

```
Feature: IPV4
```

```
Source : 172.16.10.2
```

```
Destination : 172.16.10.1
```

```
Protocol : 1 (ICMP)
```

Примеры трассировки пакетов

Этот раздел предоставляет некоторые примеры, где функция трассировки пакетов полезна для целей устранения проблем.

Пример трассировки пакетов - NAT

С данным примером интерфейсная Переадресация исходной сети (NAT) настроена на Интерфейсе WAN ASR1K (Gig0/0/0) для локальной подсети (172.16.10.0/24).

Вот условие платформы и конфигурация трассировки пакетов, которая используется для отслеживания трафика от 172.16.10.2 до 172.16.20.2, который становится преобразованным (NAT) на интерфейсе Gig0/0/0:

```
ASR1000#show platform packet-trace summary
```

```
Pkt Input Output State Reason
```

```
0 INJ.2 Gi0/0/1 FWD
```

```
1 Gi0/0/1 internal0/0/rp:0 PUNT 11 (For-us data)
```

```
2 INJ.2 Gi0/0/1 FWD
```

```
3 Gi0/0/1 internal0/0/rp:0 PUNT 11 (For-us data)
```

```
4 INJ.2 Gi0/0/1 FWD
```

```
5 Gi0/0/1 internal0/0/rp:0 PUNT 11 (For-us data)
```

```
6 INJ.2 Gi0/0/1 FWD
```

```
7 Gi0/0/1 internal0/0/rp:0 PUNT 11 (For-us data)
```

```
8 INJ.2 Gi0/0/1 FWD
```

```
9 Gi0/0/1 internal0/0/rp:0 PUNT 11 (For-us data)
```

ASR1000#show platform packet-trace packet 0

Packet: 0 CBUG ID: 120

Summary

Input : INJ.2

Output : GigabitEthernet0/0/1

State : FWD

Timestamp

Start : 115612780360228 ns (05/29/2014 15:02:55.467987 UTC)

Stop : 115612780380931 ns (05/29/2014 15:02:55.468008 UTC)

Path Trace

Feature: IPV4

Source : 172.16.10.1

Destination : 172.16.10.2

Protocol : 1 (ICMP)

ASR1000#

ASR1000#show platform packet-trace packet 1

Packet: 1 CBUG ID: 121

Summary

Input : GigabitEthernet0/0/1

Output : internal0/0/rp:0

State : PUNT 11 (For-us data)

Timestamp

Start : 115612781060418 ns (05/29/2014 15:02:55.468687 UTC)

Stop : 115612781120041 ns (05/29/2014 15:02:55.468747 UTC)

Path Trace

Feature: IPV4

Source : 172.16.10.2

Destination : 172.16.10.1

Protocol : 1 (ICMP)

Когда пять пакетов ICMP передаются от 172.16.10.2 до 172.16.20.2 с интерфейсной исходной конфигурацией NAT, это результаты трассировки пакетов:

ASR1000#show platform packet-trace summary

Pkt	Input	Output	State	Reason
-----	-------	--------	-------	--------

0	Gi0/0/1	Gi0/0/0	FWD	
---	---------	---------	-----	--

1	Gi0/0/1	Gi0/0/0	FWD	
---	---------	---------	-----	--

2	Gi0/0/1	Gi0/0/0	FWD	
---	---------	---------	-----	--

3	Gi0/0/1	Gi0/0/0	FWD	
---	---------	---------	-----	--

4	Gi0/0/1	Gi0/0/0	FWD	
---	---------	---------	-----	--

ASR1000#show platform packet-trace statistics

Packets Summary

Matched 5

Traced 5

Packets Received

Ingress 5

Inject 0

Packets Processed

Forward 5

Punt 0

Drop 0

Consume 0

ASR1000#show platform packet-trace packet 0

Packet: 0 CBUG ID: 146

Summary

Input : GigabitEthernet0/0/1

Output : GigabitEthernet0/0/0

State : FWD

Timestamp

Start : 3010217805313 ns (05/17/2014 07:01:52.227836 UTC)


```
Stop : 3010217892847 ns (05/17/2014 07:01:52.227923 UTC)
Path Trace
Feature: IPV4
Source : 172.16.10.2
Destination : 172.16.20.2
Protocol : 1 (ICMP)
Feature: FIA_TRACE
Entry : 0x806c7eac - DEBUG_COND_INPUT_PKT
Lapsed time: 1031 ns
Feature: FIA_TRACE
Entry : 0x82011c00 - IPV4_INPUT_DST_LOOKUP_CONSUME
Lapsed time: 462 ns
Feature: FIA_TRACE
Entry : 0x82000170 - IPV4_INPUT_FOR_US_MARTIAN
Lapsed time: 355 ns
Feature: FIA_TRACE
Entry : 0x803c6af4 - IPV4_INPUT_VFR
Lapsed time: 266 ns
Feature: FIA_TRACE
Entry : 0x82004500 - IPV4_OUTPUT_LOOKUP_PROCESS
Lapsed time: 942 ns
Feature: FIA_TRACE
Entry : 0x8041771c - IPV4_INPUT_IPOPTIONS_PROCESS
Lapsed time: 88 ns
Feature: FIA_TRACE
Entry : 0x82013400 - MPLS_INPUT_GOTO_OUTPUT_FEATURE
Lapsed time: 568 ns
Feature: FIA_TRACE
Entry : 0x803c6900 - IPV4_OUTPUT_VFR
Lapsed time: 266 ns
Feature: NAT
Direction : IN to OUT
Action : Translate Source
Old Address : 172.16.10.2 00028
New Address : 192.168.10.1 00002
Feature: FIA_TRACE
Entry : 0x8031c248 - IPV4_NAT_OUTPUT_FIA
Lapsed time: 55697 ns
Feature: FIA_TRACE
Entry : 0x801424f8 - IPV4_OUTPUT_THREAT_DEFENSE
Lapsed time: 693 ns
Feature: FIA_TRACE
Entry : 0x803c60b8 - IPV4_MC_OUTPUT_VFR_REFRAG
Lapsed time: 88 ns
Feature: FIA_TRACE
Entry : 0x82014900 - IPV6_INPUT_L2_REWRITE
Lapsed time: 444 ns
Feature: FIA_TRACE
Entry : 0x82000080 - IPV4_OUTPUT_FRAG
Lapsed time: 88 ns
Feature: FIA_TRACE
Entry : 0x8200e600 - IPV4_OUTPUT_DROP_POLICY
Lapsed time: 1457 ns
Feature: FIA_TRACE
Entry : 0x82017980 - MARMOT_SPA_D_TRANSMIT_PKT
Lapsed time: 7431 ns
ASR1000#
```

Пример трассировки пакетов - VPN

С данным примером туннель VPN типа «узел-узел» используется между ASR1K и маршрутизатором Cisco IOS для защиты трафика, который течет между 172.16.10.0/24 и 172.16.20.0/24 (локальные и удаленные подсети).

Вот условие платформы и конфигурация трассировки пакетов, которая используется для отслеживания трафика VPN, который вытекает 172.16.10.2 к 172.16.20.2 на Концерте 0/0/1 интерфейс:

```
ASR1000#show platform packet-trace summary
```

```
Pkt Input Output State Reason
0 Gi0/0/1 Gi0/0/0 FWD
1 Gi0/0/1 Gi0/0/0 FWD
2 Gi0/0/1 Gi0/0/0 FWD
3 Gi0/0/1 Gi0/0/0 FWD
4 Gi0/0/1 Gi0/0/0 FWD
```

```
ASR1000#show platform packet-trace statistics
```

```
Packets Summary
Matched 5
Traced 5
Packets Received
Ingress 5
Inject 0
Packets Processed
Forward 5
Punt 0
Drop 0
Consume 0
```

```
ASR1000#show platform packet-trace packet 0
```

```
Packet: 0 CBUG ID: 146
Summary
Input : GigabitEthernet0/0/1
Output : GigabitEthernet0/0/0
State : FWD
Timestamp
Start : 3010217805313 ns (05/17/2014 07:01:52.227836 UTC)
Stop : 3010217892847 ns (05/17/2014 07:01:52.227923 UTC)
Path Trace
Feature: IPV4
Source : 172.16.10.2
Destination : 172.16.20.2
Protocol : 1 (ICMP)
Feature: FIA_TRACE
Entry : 0x806c7eac - DEBUG_COND_INPUT_PKT
Lapsed time: 1031 ns
Feature: FIA_TRACE
Entry : 0x82011c00 - IPV4_INPUT_DST_LOOKUP_CONSUME
Lapsed time: 462 ns
Feature: FIA_TRACE
Entry : 0x82000170 - IPV4_INPUT_FOR_US_MARTIAN
Lapsed time: 355 ns
Feature: FIA_TRACE
Entry : 0x803c6af4 - IPV4_INPUT_VFR
Lapsed time: 266 ns
Feature: FIA_TRACE
Entry : 0x82004500 - IPV4_OUTPUT_LOOKUP_PROCESS
Lapsed time: 942 ns
Feature: FIA_TRACE
Entry : 0x8041771c - IPV4_INPUT_IPOPTIONS_PROCESS
Lapsed time: 88 ns
Feature: FIA_TRACE
Entry : 0x82013400 - MPLS_INPUT_GOTO_OUTPUT_FEATURE
Lapsed time: 568 ns
Feature: FIA_TRACE
Entry : 0x803c6900 - IPV4_OUTPUT_VFR
Lapsed time: 266 ns
```

```
Feature: NAT
Direction : IN to OUT
Action : Translate Source
Old Address : 172.16.10.2 00028
New Address : 192.168.10.1 00002
Feature: FIA_TRACE
Entry : 0x8031c248 - IPV4_NAT_OUTPUT_FIA
Lapsed time: 55697 ns
Feature: FIA_TRACE
Entry : 0x801424f8 - IPV4_OUTPUT_THREAT_DEFENSE
Lapsed time: 693 ns
Feature: FIA_TRACE
Entry : 0x803c60b8 - IPV4_MC_OUTPUT_VFR_REFRAG
Lapsed time: 88 ns
Feature: FIA_TRACE
Entry : 0x82014900 - IPV6_INPUT_L2_REWRITE
Lapsed time: 444 ns
Feature: FIA_TRACE
Entry : 0x82000080 - IPV4_OUTPUT_FRAG
Lapsed time: 88 ns
Feature: FIA_TRACE
Entry : 0x8200e600 - IPV4_OUTPUT_DROP_POLICY
Lapsed time: 1457 ns
Feature: FIA_TRACE
Entry : 0x82017980 - MARMOT_SPA_D_TRANSMIT_PKT
Lapsed time: 7431 ns
ASR1000#
```

Когда пять пакетов ICMP передаются от 172.16.10.2 до 172.16.20.2, которые зашифрованы VPN-туннелем между ASR1K и маршрутизатором Cisco IOS в данном примере, это выходные данные трассировки пакетов:

Примечание: Трассировки пакетов показывают маркер Сопоставления безопасности (SA) QFP в трассировке, которая используется для шифрования пакета, который полезен, когда вы решаете проблемы IPSEC VPN, чтобы проверить, что корректный SA используется для шифрования.

```
ASR1000#show platform packet-trace summary
Pkt Input Output State Reason
0 Gi0/0/1 Gi0/0/0 FWD
1 Gi0/0/1 Gi0/0/0 FWD
2 Gi0/0/1 Gi0/0/0 FWD
3 Gi0/0/1 Gi0/0/0 FWD
4 Gi0/0/1 Gi0/0/0 FWD

ASR1000#show platform packet-trace packet 0
Packet: 0 CBUG ID: 211
Summary
Input : GigabitEthernet0/0/1
Output : GigabitEthernet0/0/0
State : FWD
Timestamp
Start : 4636921551459 ns (05/17/2014 07:28:59.211375 UTC)
Stop : 4636921668739 ns (05/17/2014 07:28:59.211493 UTC)
Path Trace
Feature: IPV4
Source : 172.16.10.2
Destination : 172.16.20.2
Protocol : 1 (ICMP)
Feature: FIA_TRACE
Entry : 0x806c7eac - DEBUG_COND_INPUT_PKT
Lapsed time: 622 ns
```

Feature: FIA_TRACE
Entry : 0x82011c00 - IPV4_INPUT_DST_LOOKUP_CONSUME
Lapsed time: 462 ns
Feature: FIA_TRACE
Entry : 0x82000170 - IPV4_INPUT_FOR_US_MARTIAN
Lapsed time: 320 ns
Feature: FIA_TRACE
Entry : 0x82004500 - IPV4_OUTPUT_LOOKUP_PROCESS
Lapsed time: 1102 ns
Feature: FIA_TRACE
Entry : 0x8041771c - IPV4_INPUT_IPOPTIONS_PROCESS
Lapsed time: 88 ns
Feature: FIA_TRACE
Entry : 0x82013400 - MPLS_INPUT_GOTO_OUTPUT_FEATURE
Lapsed time: 586 ns
Feature: FIA_TRACE
Entry : 0x803c6900 - IPV4_OUTPUT_VFR
Lapsed time: 266 ns
Feature: FIA_TRACE
Entry : 0x80757914 - MC_OUTPUT_GEN_RECYCLE
Lapsed time: 195 ns
Feature: FIA_TRACE
Entry : 0x803c60b8 - IPV4_MC_OUTPUT_VFR_REFRAG
Lapsed time: 88 ns
Feature: IPSec
Result : IPSEC_RESULT_SA
Action : ENCRYPT
SA Handle : 6
Peer Addr : 192.168.20.1
Local Addr: 192.168.10.1
Feature: FIA_TRACE
Entry : 0x8043caec - IPV4_OUTPUT_IPSEC_CLASSIFY
Lapsed time: 9528 ns
Feature: FIA_TRACE
Entry : 0x8043915c - IPV4_OUTPUT_IPSEC_DOUBLE_ACL
Lapsed time: 355 ns
Feature: FIA_TRACE
Entry : 0x8043b45c - IPV4_IPSEC_FEATURE_RETURN
Lapsed time: 657 ns
Feature: FIA_TRACE
Entry : 0x8043ae28 - IPV4_OUTPUT_IPSEC_RERUN_JUMP
Lapsed time: 888 ns
Feature: FIA_TRACE
Entry : 0x80436f10 - IPV4_OUTPUT_IPSEC_POST_PROCESS
Lapsed time: 2186 ns
Feature: FIA_TRACE
Entry : 0x8043b45c - IPV4_IPSEC_FEATURE_RETURN
Lapsed time: 675 ns
Feature: FIA_TRACE
Entry : 0x82014900 - IPV6_INPUT_L2_REWRITE
Lapsed time: 1902 ns
Feature: FIA_TRACE
Entry : 0x82000080 - IPV4_OUTPUT_FRAG
Lapsed time: 71 ns
Feature: FIA_TRACE
Entry : 0x8200e600 - IPV4_OUTPUT_DROP_POLICY
Lapsed time: 1582 ns
Feature: FIA_TRACE
Entry : 0x82017980 - MARMOT_SPA_D_TRANSMIT_PKT
Lapsed time: 3964 ns
ASR1000#

Влияние на производительность

Буферы трассировки пакетов используют DRAM QFP, так помнить количество памяти, которого конфигурация требует и количество памяти, которое доступно.

Влияние на производительность варьируется, зависящее от опций трассировки пакетов, которые включены. Трассировка пакетов только влияет на скорость переадресации пакетов, которые отслежены, такие как те пакеты, которые совпадают с настраиваемыми условиями. Чем более гранулированные и подробные сведения, которые вы настраиваете трассировку пакетов для получения, тем больше это повлияет на ресурсы.

Как с любым устранением проблем, лучше проявлять итерационный подход и только включать более подробные опции трассировки, когда ситуация с отладкой гарантирует его.

Использование DRAM QFP может быть оценено с этой формулой:

памяти было нужно = (издержки stats) + цифра pkts * (итоговый размер +, размер данных пути + копирует размер),

Примечание: Где издержки stats и итоговый размер исправлены в 2 КБ и 128 В, соответственно, размер данных пути и копируют размер, настраиваемы.

Дополнительные сведения

- [Руководство по конфигурации программного обеспечения series маршрутизаторов агрегации Cisco ASR1000 Series - трассировка пакетов](#)
- [Отбрасывание пакетов маршрутизаторов Cisco ASR 1000 Series Service](#)
- [Cisco Systems – техническая поддержка и документация](#)