使用IOS-XE数据路径数据包跟踪功能进行故障排除

目录

简介

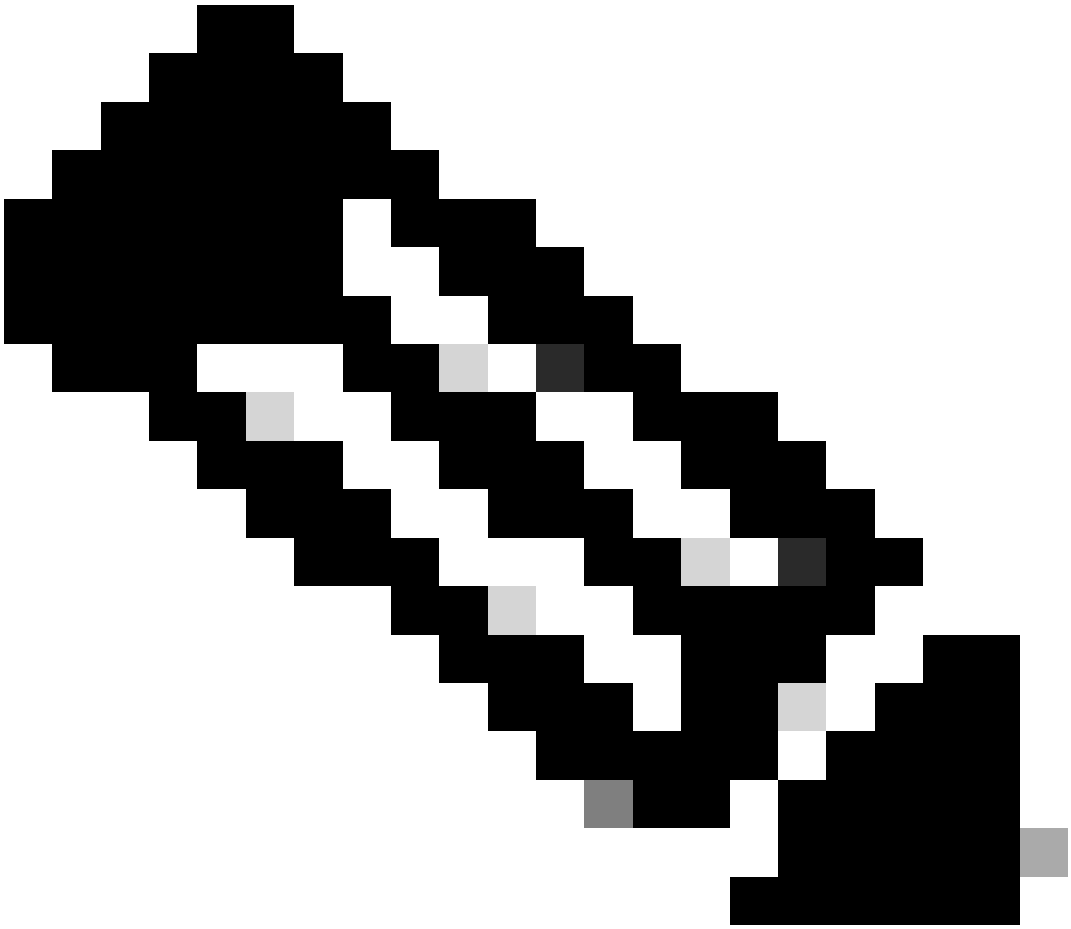
本文档介绍如何通过数据包跟踪功能对Cisco IOS-XE®软件执行数据路径数据包跟踪。

先决条件

要求

Cisco建议您了解以下信息：

数据包跟踪功能在基于QFP（量子流处理器）的路由平台上的Cisco IOS-XE版本3.10和更高版本中可用，这些路由平台包括ASR1000、ISR4000、ISR1000、Catalyst 1000、Catalyst 8000、CSR1000v和Catalyst 8000v系列路由器。运行Cisco IOS-XE软件的ASR900系列聚合服务路由器或Catalyst系列交换机不支持此功能。



> 注意：数据包跟踪功能在ASR1000系列路由器上的专用管理接口GigabitEthernet0上不起作用，因为该接口上转发的数据包不由QFP处理。

使用的组件

本文档中的信息基于以下软件和硬件版本：

- Cisco IOS-XE软件版本3.10S (15.3(3)S)及更高版本
- ASR1000系列路由器

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

背景信息

为了在故障排除时确定配置错误、容量过载或普通软件Bug等问题，必须了解系统中数据包所发生的情况。Cisco IOS-XE数据包跟踪功能可满足此需求。它提供了一种字段安全方法，用于记账并根据用户定义的条件类捕获每个数据包的进程详细信息。

参考拓扑

下图说明了用于本文档中介绍的示例的拓扑：

使用中的数据包跟踪

为了说明数据包跟踪功能的使用，本节使用的示例描述了互联网控制消息协议(ICMP)流量从本地工作站172.16.10.2（位于ASR1K之后）到ASR1K上接口GigabitEthernet0/0/1上的入口方向的远程主机172.16.20.2的跟踪。

您可以通过以下两个步骤跟踪ASR1K上的数据包：

1. 启用平台条件调试，以选择要在ASR1K上跟踪的数据包或流量。

2. 使用path-trace或Feature Invocation Array (FIA)跟踪选项启用平台数据包跟踪。

快速入门指南

如果您已经熟悉本文档的内容，并且希望通过部分快速查看CLI，请点击此处查看快速入门指南。下面仅列举几个示例来说明该工具的用法。请参阅后面详细讨论语法的部分，并确保使用适合您的要求的配置。

1. 配置平台条件：

```
<#root>

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
```

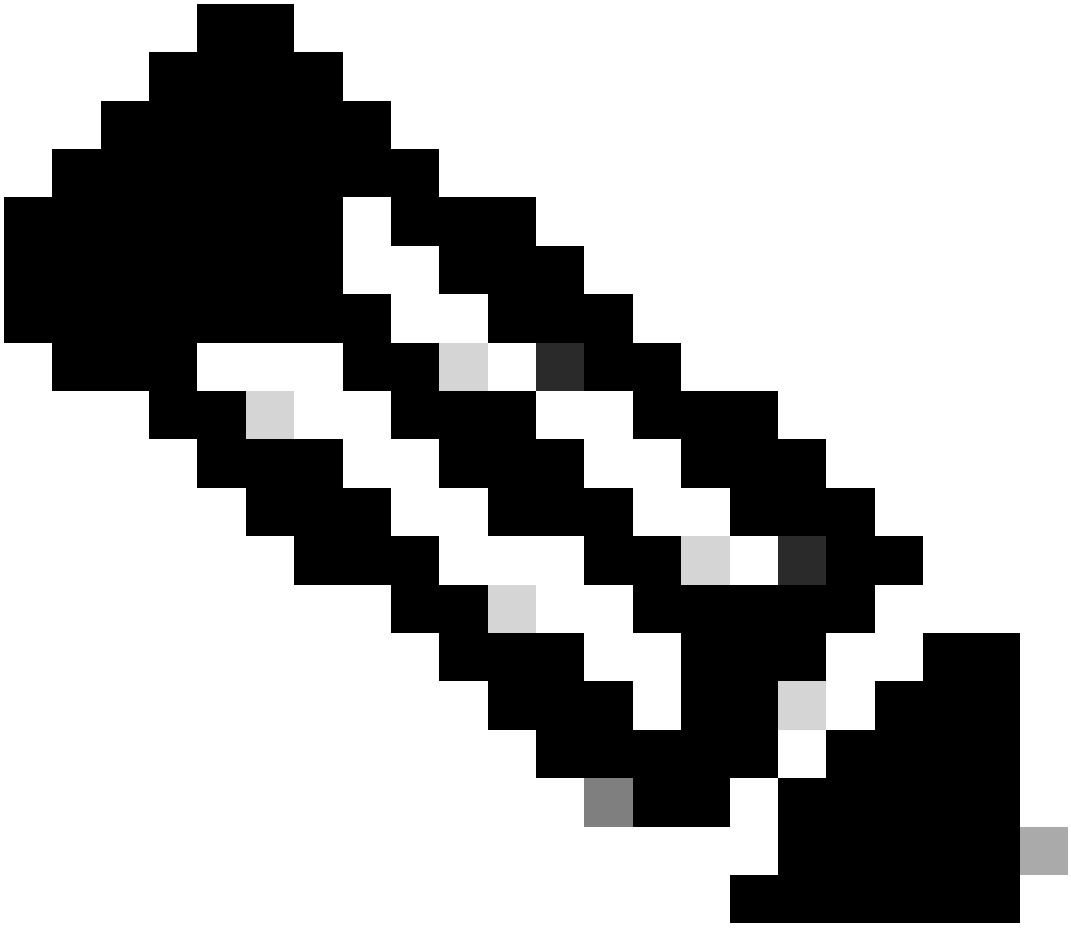
```
 defined prior to configuring this command) - matches egress packets corresponding
 to access-list 198


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)
```

配置平台条件后，使用以下CLI命令启动平台条件：

```
<#root>

debug platform condition start
```

2. 配置数据包跟踪：

```
<#root>

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.
```



注意：在较早的Cisco IOS-XE 3.x版本中，还需要使用debug platform packet-trace enable命令来启动数据包跟踪功能。Cisco IOS-XE 16.x版本不再需要此功能。

输入以下命令以清除trace buffer和reset packet-trace：

<#root>

clear platform packet-trace statistics

  --> clear the packet trace buffer

用于清除平台条件和数据包跟踪配置的命令为：

<#root>

clear platform condition all

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

显示命令

在应用上述命令之后，请验证平台条件和数据包跟踪配置，以确保您拥有所需的配置。

<#root>

```
show platform conditions

 --> shows the platform conditions configured

show platform packet-trace configuration

 --> shows the packet-trace configurations


show debugging

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

以下是用于检查跟踪/捕获的数据包的命令：

<#root>

```
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

 -> Display path trace of FIA trace details for the 12th packet in the trace buffer
```

启用平台条件调试

数据包跟踪功能依靠条件调试基础设施来确定要跟踪的数据包。条件调试基础设施能够根据以下条件过滤流量：

- 协议
- IP地址和掩码
- 访问控制列表(ACL)

- 接口
- 流量方向（入口或出口）

这些条件定义了过滤器应用于数据包的位置和时间。

对于本示例中使用的流量，为从172.16.10.2到172.16.20.2的ICMP数据包启用入口方向的平台条件调试。换句话说，选择要跟踪的流量。有多种选项可用于选择此流量。

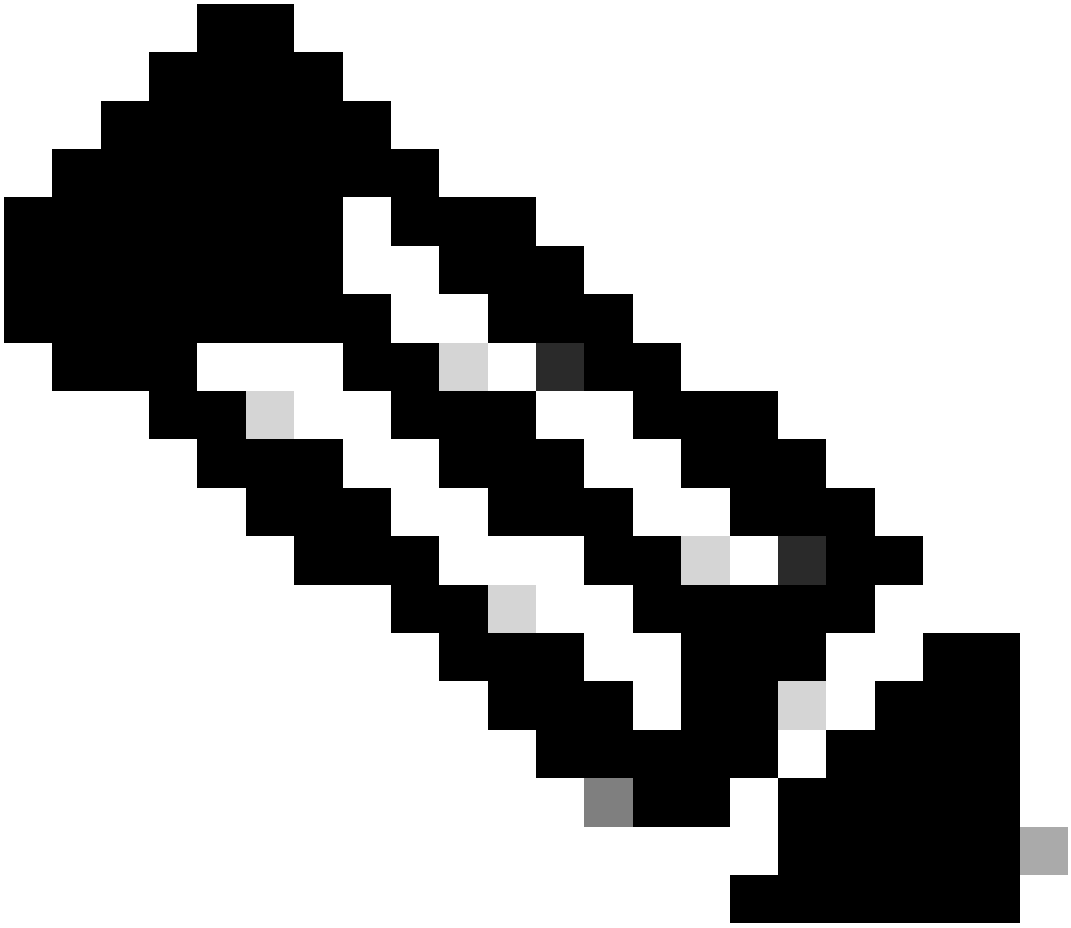
```
<#root>

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
```

在本示例中，访问列表用于定义条件，如下所示：

```
<#root>

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
```

要开始条件调试，请输入以下命令：

```
<#root>

ASR1000#

debug platform condition start
```



注意：要停止或禁用条件调试基础结构，请输入debug platform condition stop命令。

要查看配置的条件调试过滤器，请输入以下命令：

<#root>

ASR1000#

show platform conditions

Conditional Debug Global State:

Start

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

```
Feature Condition        Format                  Value
-----------------------|-----------------------|--------------------------------

ASR1000#
```

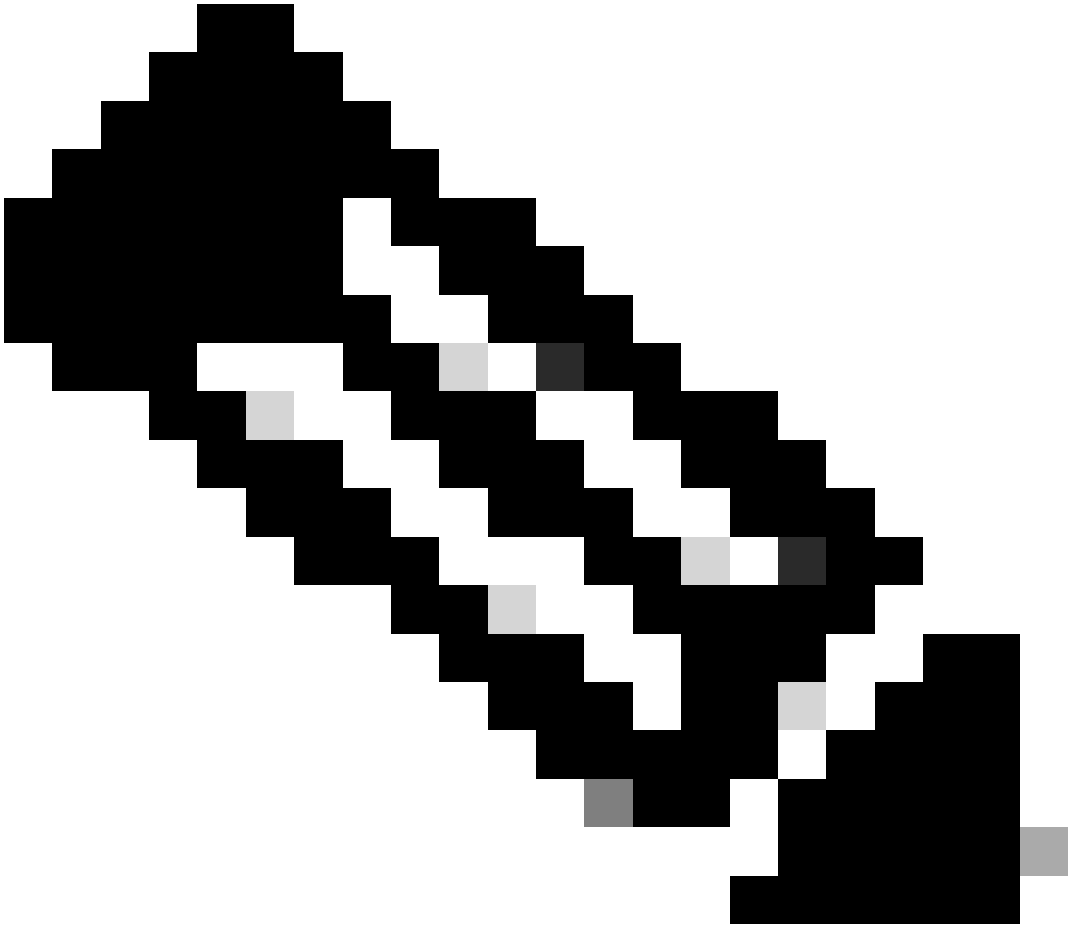
综上所述，到目前为止已应用此配置：

<#root>

```
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

启用数据包跟踪



注意：本部分详细介绍数据包和复制选项，其他选项将在本文档的后续部分进行介绍。

物理接口和逻辑接口（例如隧道接口或虚拟访问接口）都支持数据包跟踪。

以下是数据包跟踪CLI语法：

<#root>

```
ASR1000#

debug platform packet-trace

 ?
  copy    Copy packet data
  drop    Trace drops only
  inject  Trace injects only
  packet  Packet count
  punt    Trace punts only
```

<#root>

```
debug platform packet-trace packet <pkt-size/pkt-num> [fia-trace | summary-only]
 [circular] [data-size <data-size>]
```

以下是此命令关键字的说明：

- pkt-num - Packet Number指定一次维护的数据包的最大数量。
- summary-only -用于指定仅捕获摘要数据。默认值为捕获摘要数据和功能路径数据。
- fia-trace - 此选项除了执行路径数据信息外，还可以执行FIA跟踪。
- data-size -用于指定路径数据缓冲区的大小，从2,048到16,384字节。默认值为2,048字节。

<#root>

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

以下是此命令关键字的说明：

- in/out -用于指定要复制的数据包流的方向-入口和/或出口。
- L2/L3/L4 -用于指定数据包副本的开始位置。第2层(L2)是默认位置。

- size —用于指定复制的最大二进制八位数数量。默认值为64个二进制八位数。

对于本示例，以下命令用于对使用条件调试基础设施选择的流量启用数据包跟踪：

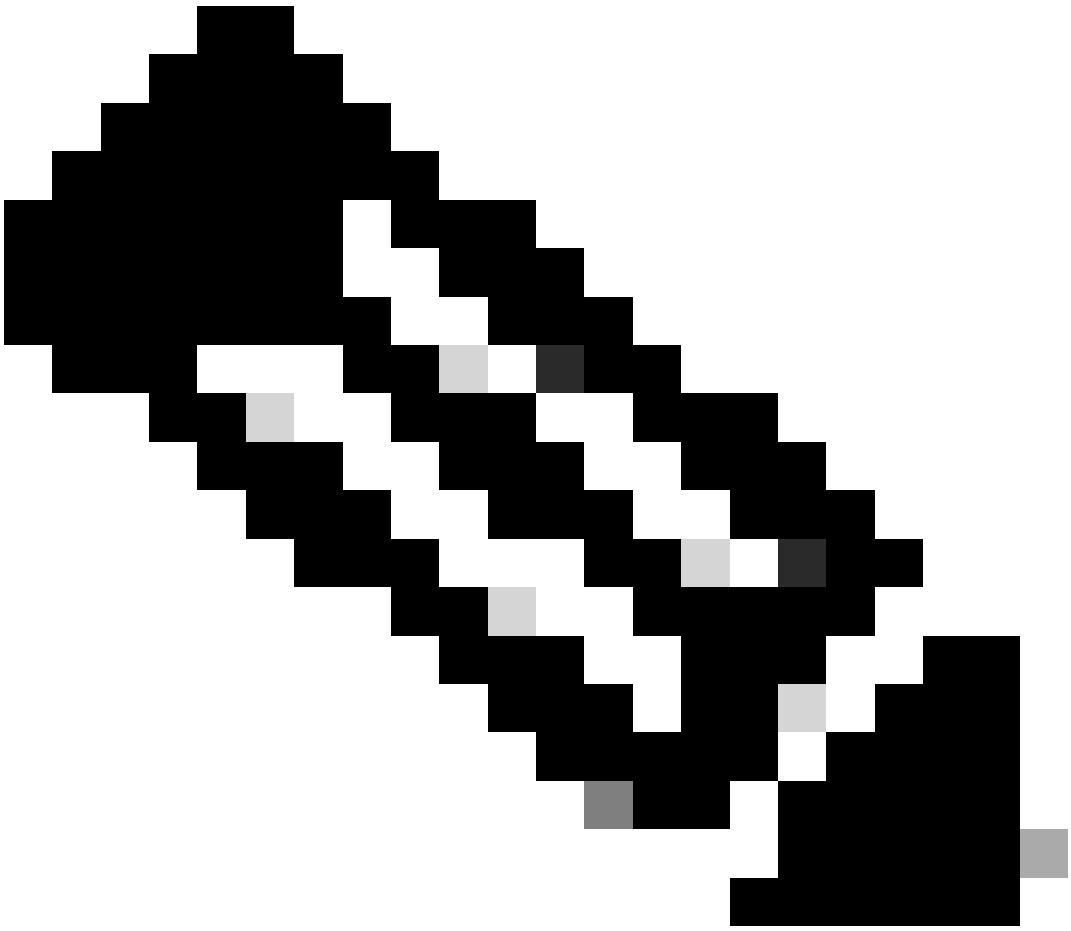
```
<#root>
ASR1000#
debug platform packet-trace packet 16
```

要查看数据包跟踪配置，请输入以下命令：

```
<#root>
ASR1000#
show platform packet-trace configuration

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

还可以输入show debugging命令查看平台条件调试和数据包跟踪配置：

```
<#root>
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 packet 16 data-size 2048
```



注意：输入clear platform condition all命令以清除所有平台调试条件以及数据包跟踪配置和数据。

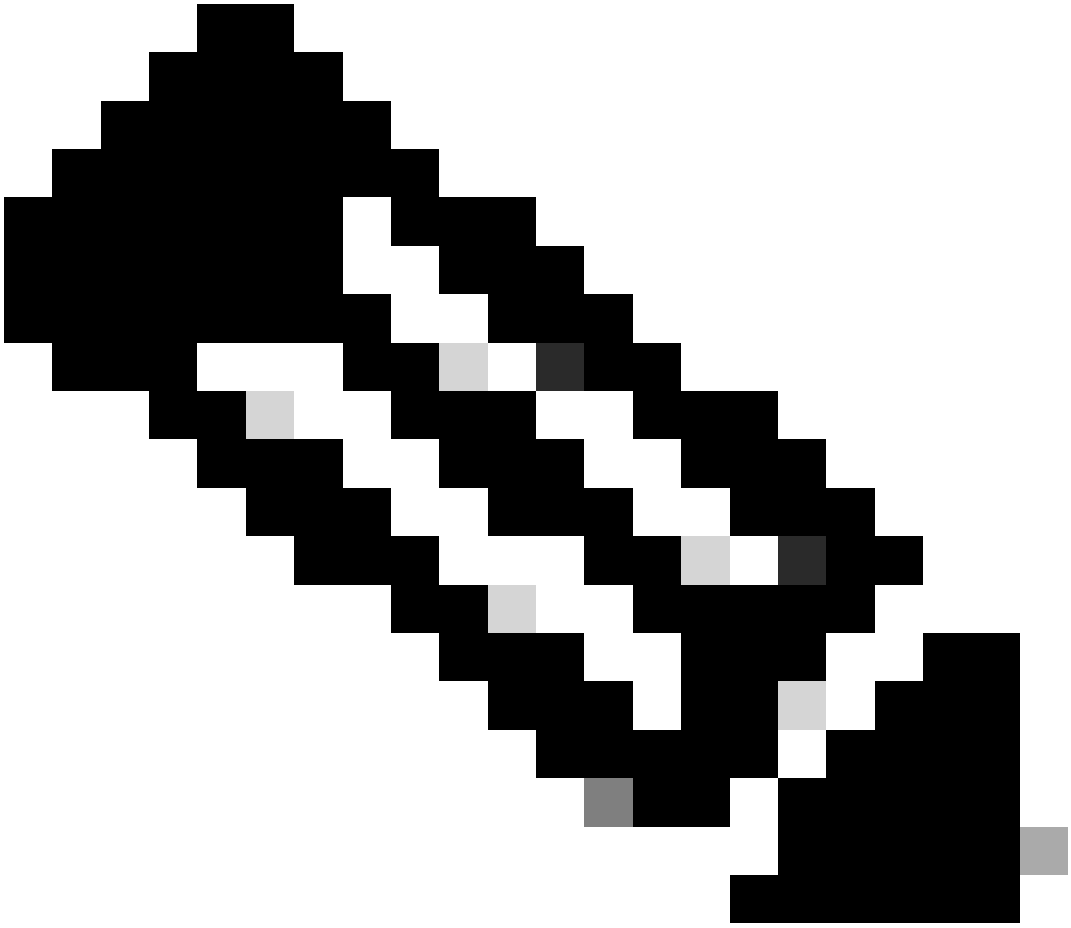
总之，到目前为止已使用此配置数据启用packet-trace：

<#root>

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

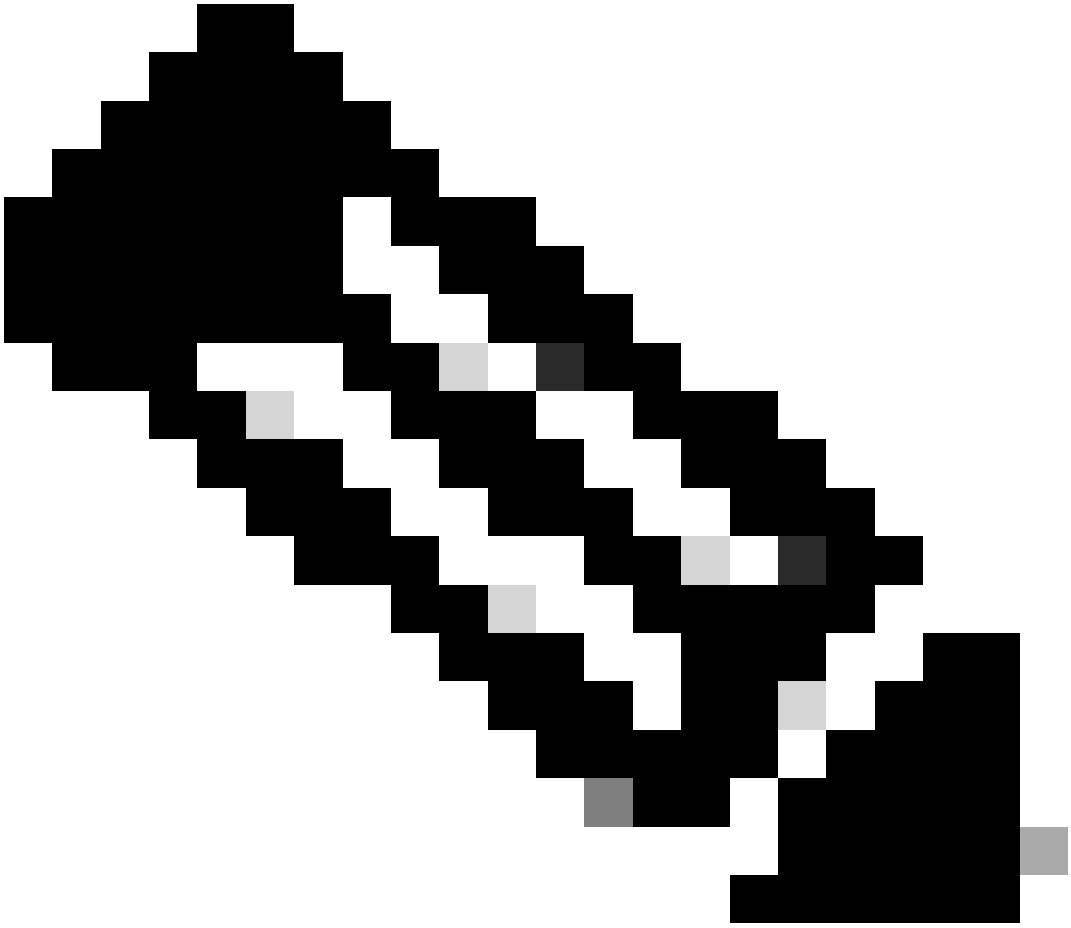
数据包跟踪的出口条件限制

条件定义了条件过滤器以及条件过滤器应用于数据包的时间。例如，debug platform condition interface g0/0/0 egress表示当数据包到达接口g0/0/0上的输出FIA时，会将其识别为匹配，因此从入口开始到该点的所有数据包处理都将被忽略。



注意：思科强烈建议您对数据包跟踪使用入口条件，以便获得尽可能完整和有意义的数据。可以使用出口条件，但请注意这些限制。

显示数据包跟踪结果



注意：本部分假定启用路径跟踪。

数据包跟踪提供三个特定级别的检查：

- 记账
- 每个数据包的摘要
- 每数据包路径数据

从172.16.10.2到172.16.20.2发送五个ICMP请求数据包时，可以使用以下命令查看数据包跟踪结果：

<#root>

ASR1000#

show platform packet-trace statistics

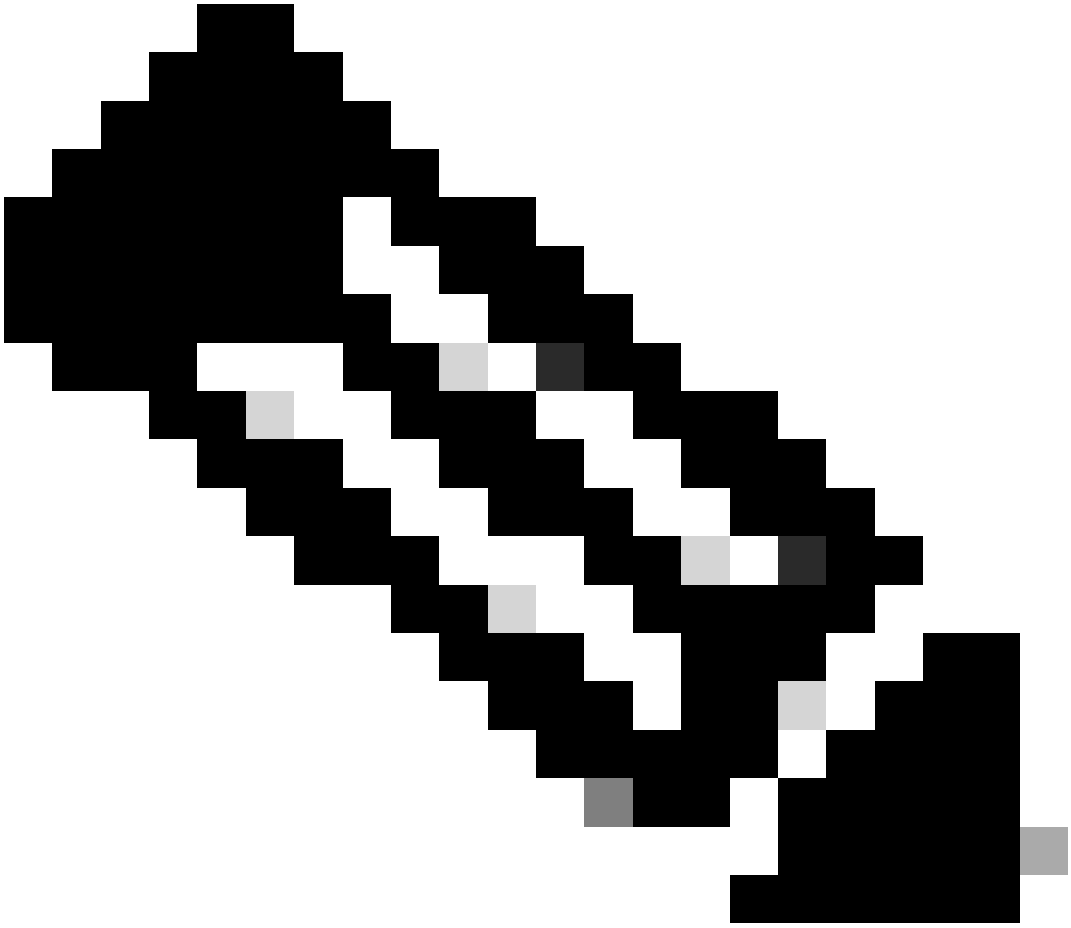
```
Packets Traced: 5


 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#
```



注意：第三个命令提供了一个示例，说明如何查看每个数据包的数据包跟踪。在本例中，显示了跟踪的第一个数据包。

从这些输出中，您可以看到跟踪了五个数据包，并且您可以查看输入接口、输出接口、状态和路径跟踪。

状态	备注
转发	数据包已安排/排队等待传输，并通过出口接口转发到下一跳。
传送	数据包从转发处理器(FP)发送到路由处理器(RP)（控制平面）。
丢弃	数据包会在FP上丢弃。运行FIA跟踪、使用全局丢弃计数器或使用数据路径调试来查找有关丢弃原因的更多详细信息。
缺点	数据包在数据包过程中（例如，在ICMP ping请求或加密数据包期间）被使用。

数据包跟踪统计信息输出中的ingress和inject计数器分别对应于通过外部接口进入的数据包和被视为从控制平面注入的数据包。

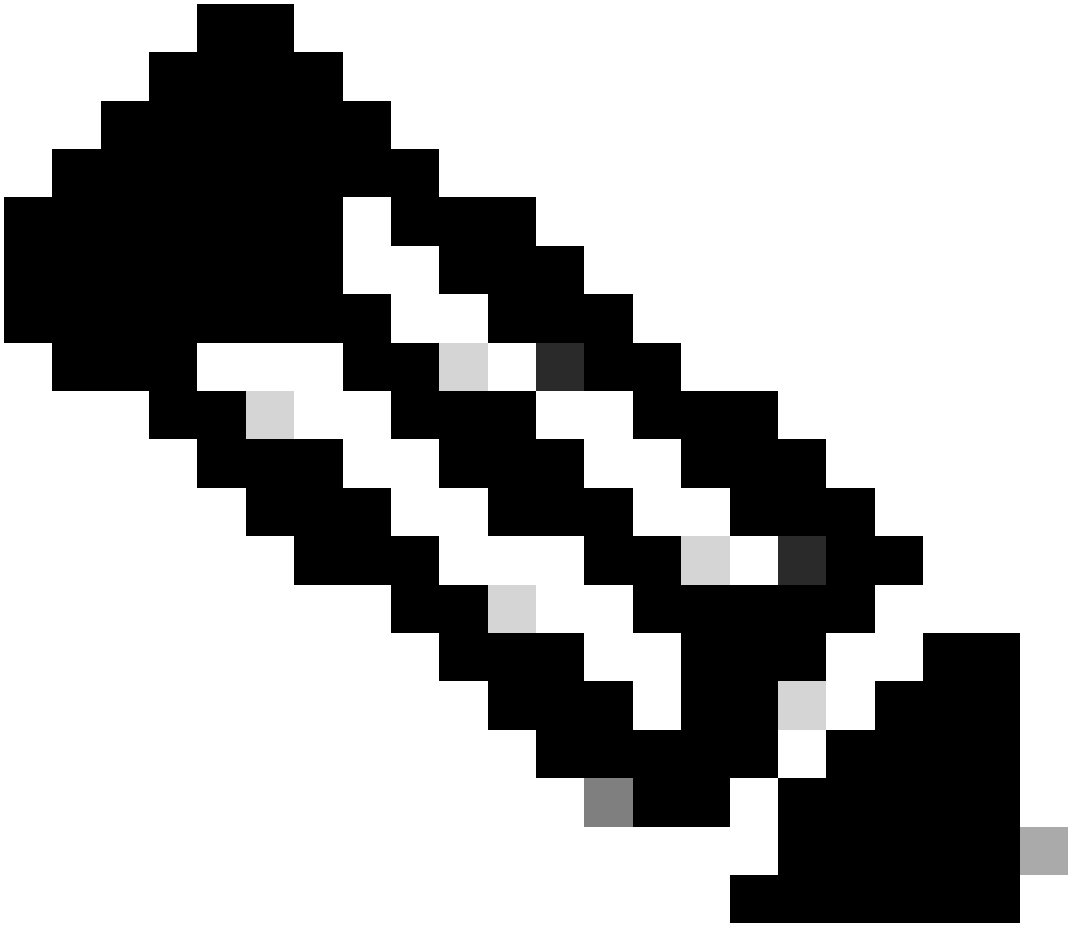
FIA跟踪

FIA保留当数据包转发到入口或出口时，量子流处理器(QFP)中的数据包处理器引擎(PPE)按顺序执行的功能列表。这些功能基于计算机上应用的配置数据。因此，FIA跟踪有助于了解数据包在处理过程中通过系统的流量。

您必须应用此配置数据，才能使用FIA启用数据包跟踪：

```
<#root>

ASR1000#

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

显示数据包跟踪结果



注意：本部分假设FIA跟踪已启用。此外，当您添加或修改当前数据包跟踪命令时，缓冲的数据包跟踪详细信息会被清除，因此您必须再次发送一些流量才能对其进行跟踪。

输入用于启用FIA跟踪的命令后，发送五个从172.16.10.2到172.16.20.2的ICMP数据包，如上一节所述。

<#root>

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#
```

检查与接口关联的FIA

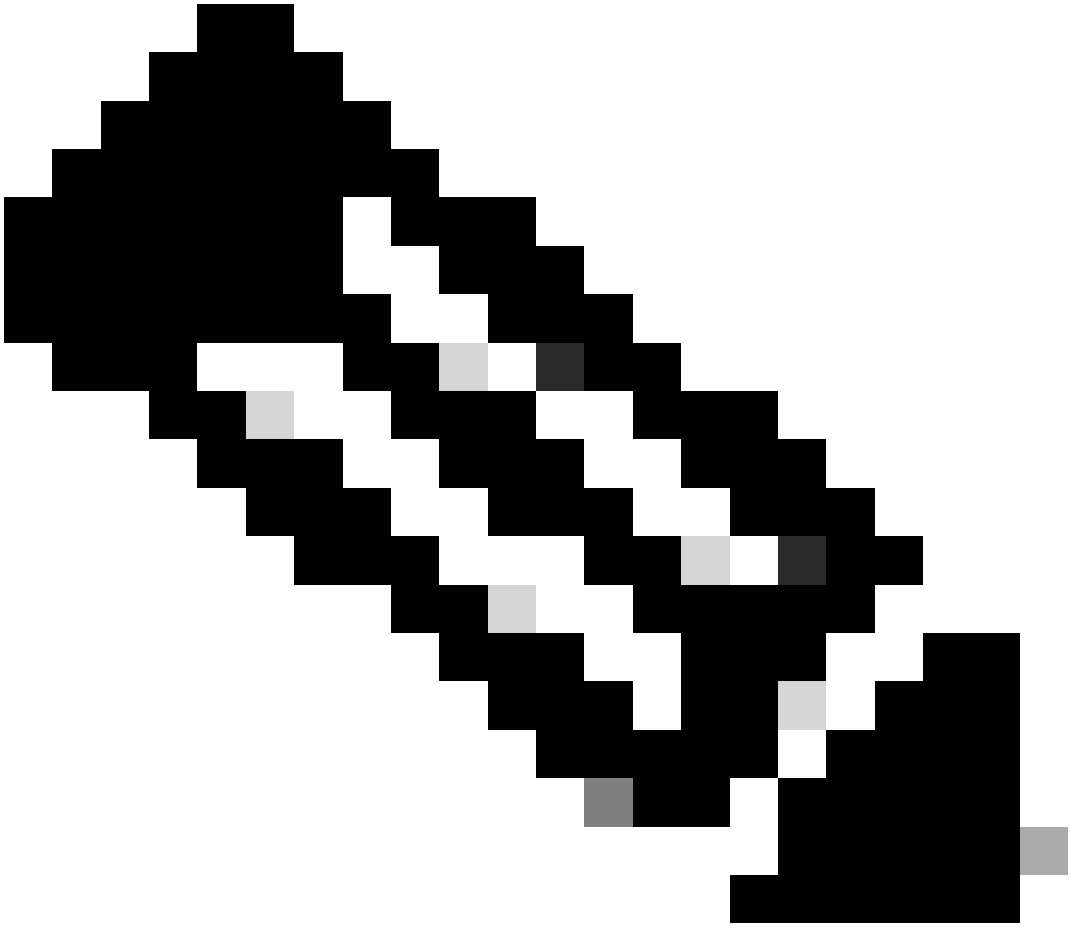
当您启用平台条件调试时，条件调试将作为功能添加到FIA。根据接口上处理的功能顺序，需要相应地设置条件过滤器，例如，在条件过滤器中必须使用NAT前地址还是后地址。

此输出显示在入口方向启用的平台条件调试的FIA中功能的顺序：

<#root>

```
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#
```



注意：CBUG_INPUT_FIA和DEBUG_COND_INPUT_PKT对应于路由器上配置的条件调试功能。

转储跟踪的数据包

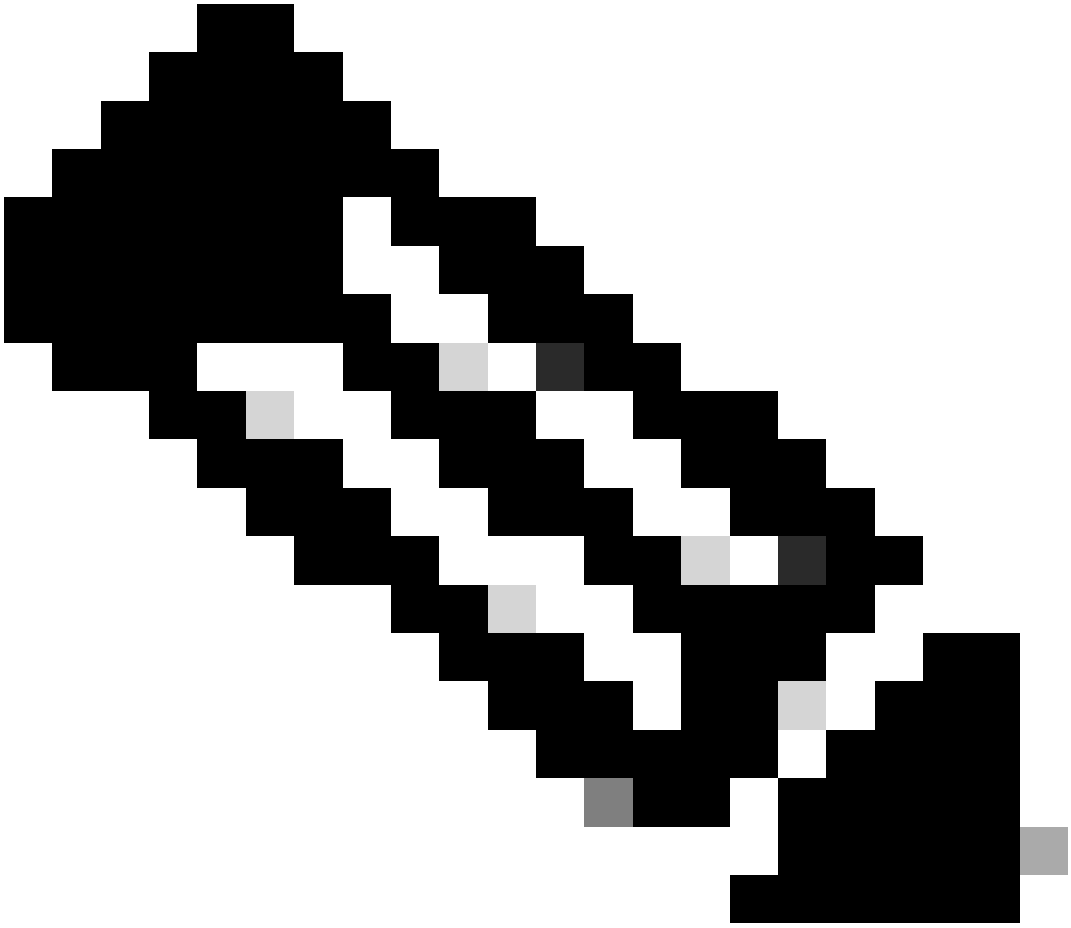
您可以复制和转储跟踪的数据包，如本部分所述。本示例显示如何在入口方向（172.16.10.2到172.16.20.2）复制最多2,048字节的数据包。

下面是所需的其他命令：

<#root>

ASR1000#

debug platform packet-trace copy packet input size 2048



注意：复制的数据包的大小在16到2,048字节的范围内。

输入以下命令以转储复制的数据包：

<#root>

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#
```

删除跟踪

Cisco IOS-XE软件版本3.11及更高版本中提供了丢弃跟踪。它只对丢弃的数据包启用数据包跟踪。以下是功能的一些亮点：

- 或者，它允许您为特定丢弃代码指定数据包的保留。
- 可以在没有全局或接口条件的情况下使用它来捕获丢弃事件。
- 丢弃事件捕获意味着仅跟踪丢弃本身，而不跟踪数据包的寿命。但是，它仍允许您捕获摘要数据、元组数据和数据包，以便帮助完善条件或提供下一步调试步骤的线索。

以下是用于启用丢弃类型数据包跟踪的命令语法：

<#root>

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

丢弃代码与丢弃ID相同，如show platform hardware qfp active statistics drop detail命令输出中所报告：

<#root>

ASR1000#

show platform hardware qfp active statistics drop detail

```
------------------------------------------------------------------------------
```

ID

```
  Global Drop Stats                          Packets                  Octets
-------------------------------------------------------------------------------

   60

  IpTtlExceeded                                3                     126

    8

  Ipv4Acl                                     32                    3432
```

丢弃跟踪场景示例

将此ACL应用于ASR1K的Gig 0/0/0接口，以丢弃从172.16.10.2到172.16.20.2的流量：

```
access-list 199 deny ip host 172.16.10.2 host 172.16.20.2
access-list 199 permit ip any any
interface Gig 0/0/0
  ip access-group 199 out
```

在已实施ACL并丢弃从本地主机到远程主机的流量时，应用以下丢弃跟踪配置：

<#root>

```
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
```

从172.16.10.2到172.16.20.2发送五个ICMP请求数据包。丢弃跟踪可捕获ACL丢弃的这些数据包，如下所示：

<#root>

```
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#
```

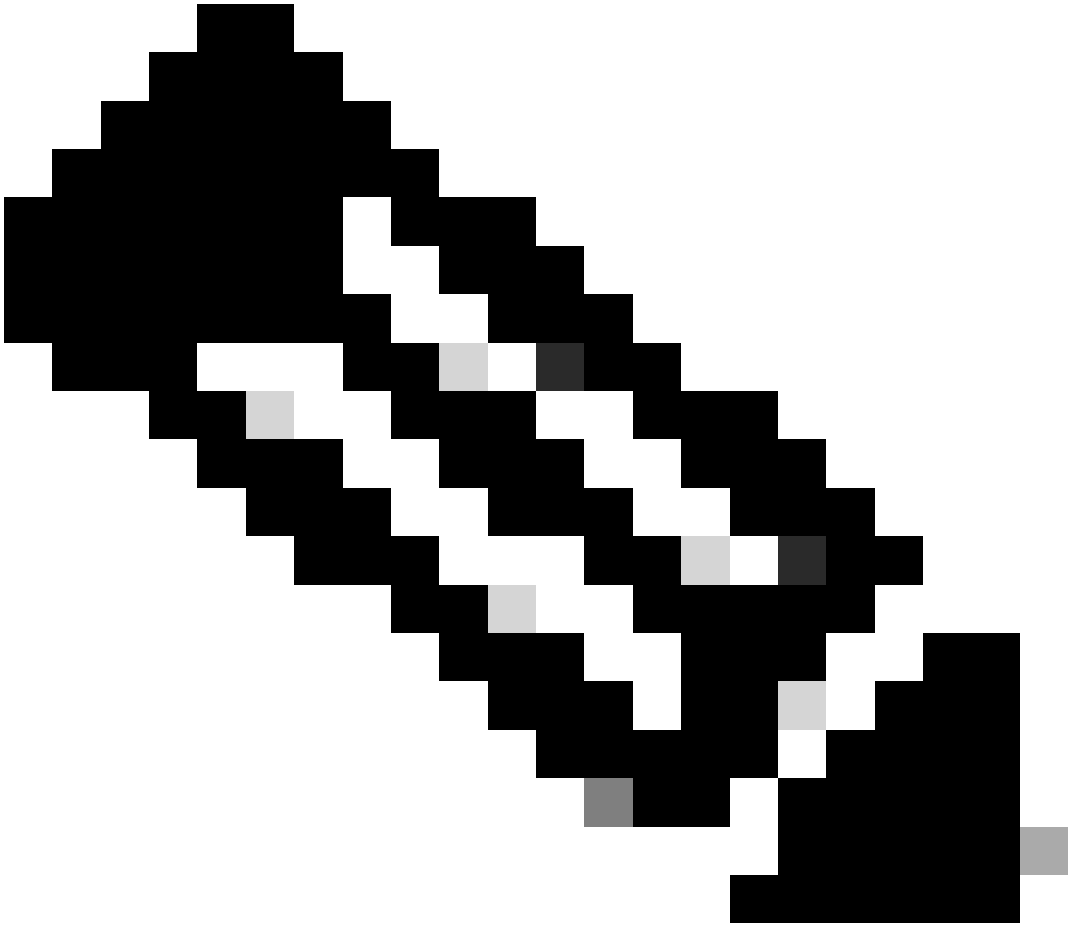
show platform packet-trace packet 0

```
 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#
```

注入和传送跟踪

在Cisco IOS-XE软件版本3.12及更高版本中添加了inject和punt数据包跟踪功能，以跟踪传送（在FP上接收的数据包被传送至控制平面）和注入（从控制平面向FP注入的数据包）数据包。



注意：传送跟踪可以在没有全局或接口条件的情况下工作，就像丢弃跟踪一样。但是，必须定义条件，注射踪迹才能起作用。

以下是从ASR1K ping相邻路由器时punt 和inject packet trace 的示例：

<#root>

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#
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#
```

现在，您可以验证punt 和nject trace r结果：

<#root>

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)
```

使用IOSd和LFTS传送/注入跟踪与UDF匹配的数据包跟踪增强（17.3.1中的新功能）

数据包跟踪功能得到进一步增强，为源或发往Cisco IOS-XE版本17.3.1中的IOSd或其他BinOS进程的数据包提供额外的跟踪信息。

IOSd丢弃跟踪

通过此增强功能，数据包跟踪扩展到IOSd，并且可以提供有关IOSd内部任何数据包丢弃的信息，通常在***show ip traffic***输出中报告。启用IOSd丢弃跟踪不需要其他配置。以下是IOSd由于错误的校验和错误而丢弃的UDP数据包的示例：

```
Router#debug platform condition ipv4 10.118.74.53/32 both
Router#debug platform condition start
Router#debug platform packet-trace packet 200
Packet count rounded up from 200 to 256

Router#
Router#show plat pack pa 0
Packet: 0           CBUG ID: 674
Summary
  Input     : GigabitEthernet1
  Output    : internal0/0/rp:0
  State     : PUNT 11  (For-us data)
  Timestamp
    Start   : 17756544435656 ns (06/29/2020 18:19:17.326313 UTC)
    Stop    : 17756544469451 ns (06/29/2020 18:19:17.326346 UTC)
Path Trace
  Feature: IPV4(Input)
    Input       : GigabitEthernet1
    Output      : <unknown>
    Source      : 10.118.74.53
    Destination : 172.18.124.38
    Protocol    : 17 (UDP)
      SrcPort   : 2640
      DstPort   : 500

IOSd Path Flow: Packet: 0    CBUG ID: 674
  Feature: INFRA
  Pkt Direction: IN
    Packet Rcvd From DATAPLANE

  Feature: IP
  Pkt Direction: IN
    Packet Enqueued in IP layer
    Source      : 10.118.74.53
    Destination : 172.18.124.38
    Interface   : GigabitEthernet1

  Feature: IP
  Pkt Direction: IN
  FORWARDED To transport layer
    Source        : 10.118.74.53
    Destination   : 172.18.124.38
    Interface     : GigabitEthernet1

  Feature: UDP
  Pkt Direction: IN
```

DROPPED

UDP: Checksum error: dropping

```
    Source      : 10.118.74.53(2640)
    Destination : 172.18.124.38(500)
```

IOSd出口路径跟踪

数据包跟踪经过增强，可显示路径跟踪和协议处理信息，因为数据包源自IOSd，并沿出口方向发送到网络。捕获IOSd出口路径跟踪信息无需其他配置。以下是流出路由器的SSH数据包的出口路径跟踪示例：

<#root>

```
Router#show platform packet-trace packet 2
Packet: 2           CBUG ID: 2
```

IOSd Path Flow:

```
  Feature: TCP
  Pkt Direction: OUTtcp0: O SYNRCVD 172.18.124.38:22 172.18.124.55:52774 seq 3052140910 OPTS 4 ACK 2346

  Feature: TCP
  Pkt Direction: OUT
  FORWARDED
    TCP: Connection is in SYNRCVD state
    ACK         : 2346709419
    SEQ         : 3052140910
    Source      : 172.18.124.38(22)
    Destination : 172.18.124.55(52774)


  Feature: IP
  Pkt Direction: OUTRoute out the generated packet.srcaddr: 172.18.124.38, dstaddr: 172.18.124.55

  Feature: IP
  Pkt Direction: OUTInject and forward successful srcaddr: 172.18.124.38, dstaddr: 172.18.124.55

  Feature: TCP
  Pkt Direction: OUTtcp0: O SYNRCVD 172.18.124.38:22 172.18.124.55:52774 seq 3052140910 OPTS 4 ACK 2346
Summary
  Input     : INJ.2
  Output    : GigabitEthernet1
  State     : FWD
  Timestamp
    Start   : 490928006866 ns (06/29/2020 13:31:30.807879 UTC)
    Stop    : 490928038567 ns (06/29/2020 13:31:30.807911 UTC)
Path Trace
  Feature: IPV4(Input)
    Input       : internal0/0/rp:0
```

```
    Output      : <unknown>
    Source      : 172.18.124.38
    Destination : 172.18.124.55
    Protocol    : 6 (TCP)
      SrcPort   : 22
      DstPort   : 52774
  Feature: IPSec
    Result    : IPSEC_RESULT_DENY
    Action    : SEND_CLEAR
    SA Handle : 0
    Peer Addr : 172.18.124.55
    Local Addr: 172.18.124.38
```

LFTS数据包跟踪

LFTS (Linux Forwarding Transport Service)是一种传输机制，用于将从CPP传送的数据包转发到除IOSd之外的其他应用。LFTS数据包跟踪增强功能在路径跟踪输出中添加了此类数据包的跟踪信息。获取LFTS跟踪信息无需其他配置。以下是向NETCONF应用传送的数据包的LFTS跟踪输出示例：

```
     <#root>

Router#show plat packet-trace pac 0
Packet: 0           CBUG ID: 461
Summary
  Input     : GigabitEthernet1
  Output    : internal0/0/rp:0
  State     : PUNT 11  (For-us data)
  Timestamp
    Start   : 647999618975 ns (06/30/2020 02:18:06.752776 UTC)
    Stop    : 647999649168 ns (06/30/2020 02:18:06.752806 UTC)
Path Trace
  Feature: IPV4(Input)
    Input       : GigabitEthernet1
    Output      : <unknown>
    Source      : 10.118.74.53
    Destination : 172.18.124.38
    Protocol    : 6 (TCP)
      SrcPort   : 65365
      DstPort   : 830
```

LFTS Path Flow: Packet: 0   CBUG ID: 461

```
  Feature: LFTS
  Pkt Direction: IN
```

```
    Punt Cause  : 11
     subCause : 0
```

基于用户定义的过滤器的数据包跟踪模式匹配（仅限ASR1000平台）

在Cisco IOS-XE版本17.3.1中，还向ASR1000产品系列添加了新的数据包匹配机制，以基于用户定义的过滤器(UDF)基础设施在数据包中的任意字段上进行匹配。这允许根据不是标准L2/L3/L4报头结构一部分的字段进行灵活的数据包匹配。下一个示例显示了UDF定义，该定义匹配2个字节的用户定义模式0x4D2，该模式从L3外部协议报头的26个字节偏移量开始。

```
udf grekey header outer l3 26 2
ip access-list extended match-grekey
 10 permit ip any any udf grekey 0x4D2 0xFFFF

debug plat condition ipv4 access-list match-grekey both
debug plat condition start
debug plat packet-trace pack 100
```

数据包跟踪示例

本节提供一些数据包跟踪功能可用于故障排除的示例。

数据包跟踪示例- NAT

在本示例中，接口源网络地址转换(NAT)在本地子网(172.16.10.0/24)的ASR1K (Gig0/0/0)的WAN接口上配置。

以下是用于跟踪从172.16.10.2到172.16.20.2（在Gig0/0/0接口上变为转换[NAT]）的流量的平台条件和数据包跟踪配置：

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

当使用接口源NAT配置从172.16.10.2到172.16.20.2发送五个ICMP数据包时，将产生以下数据包跟踪结果：

<#root>

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#
```

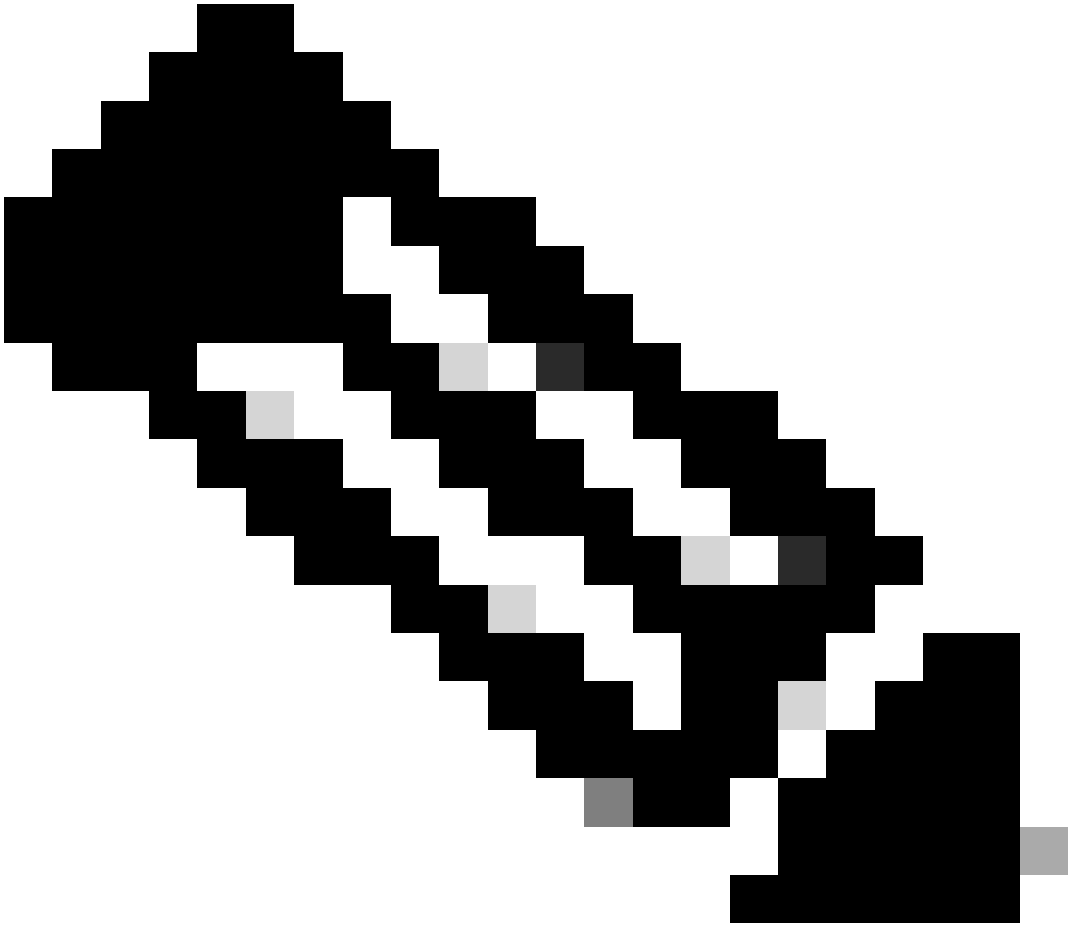
数据包跟踪示例- **VPN**

在本例中，在ASR1K和Cisco IOS路由器之间使用站点到站点VPN隧道来保护在172.16.10.0/24和172.16.20.0/24（本地和远程子网）之间流动的流量。

以下是用于跟踪在Gig 0/0/1接口上从172.16.10.2流向172.16.20.2的VPN流量的平台条件和数据包跟踪配置：

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

当从172.16.10.2发送到172.16.20.2（在本示例中通过ASR1K和Cisco IOS路由器之间的VPN隧道加密）的5个ICMP数据包时，以下是数据包跟踪输出：



注意：数据包跟踪显示用于加密数据包的跟踪中的QFP安全关联(SA)句柄，在您对IPsec VPN问题进行故障排除以验证是否使用了正确的SA进行加密时，该句柄非常有用。

<#root>

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#
```

性能影响

数据包跟踪缓冲区会消耗QFP DRAM，因此要注意配置所需的内存量和可用的内存量。

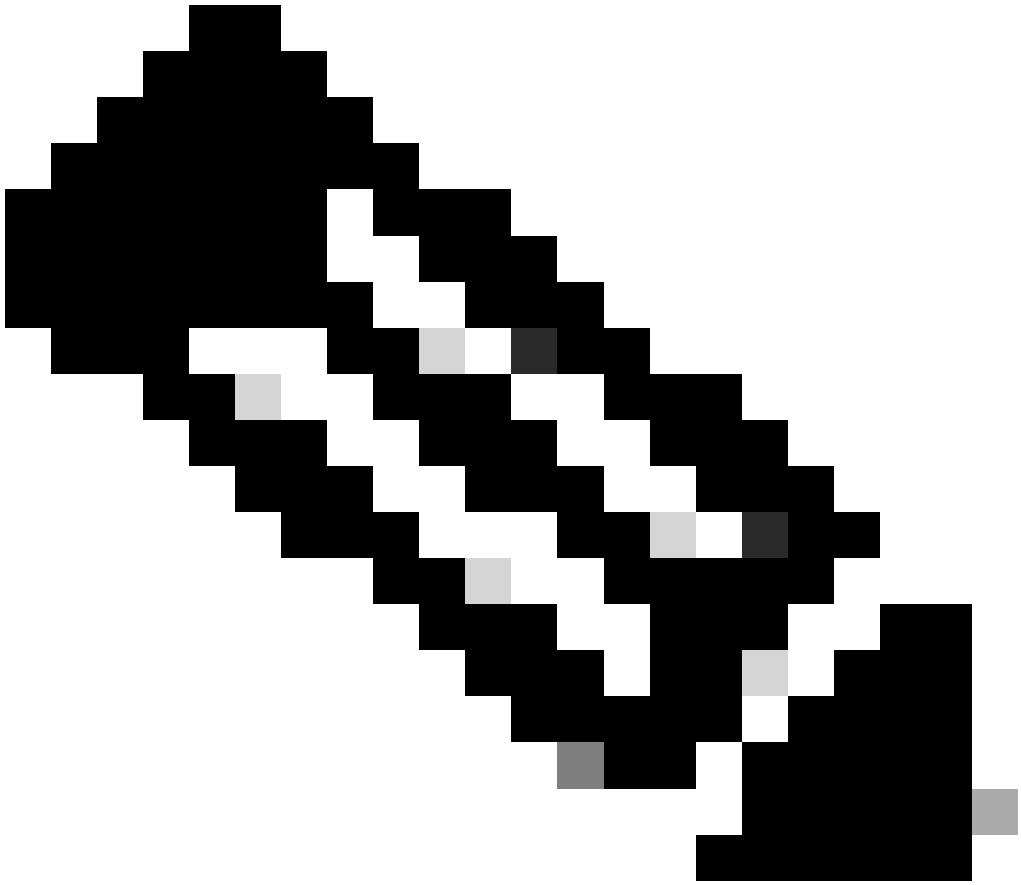
性能影响会有所不同，具体取决于启用的数据包跟踪选项。数据包跟踪仅影响跟踪的数据包的转发性能，例如那些与用户配置条件匹

配的数据包。配置要捕获的数据包跟踪的信息越精细和详细，就越能影响资源。

与任何故障排除一样，最好采用迭代方法，仅在调试情况允许时才启用更详细的跟踪选项。

QFP DRAM使用情况可通过以下公式估算：

所需内存=（统计信息开销）+数据包数*（摘要大小+路径数据大小+复制大小）



注意：如果**stats overhead**和**summary size**分别固定为2 KB和128 B，则**path data size**和**copy size**可由用户配置。

相关信息

- **Cisco ASR1000系列聚合系列路由器软件配置指南-数据包跟踪**
- **Cisco ASR1000系列服务路由器上的数据包丢弃**
- **思科技术支持和下载**

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

请注意：即使是最好的机器翻译，其准确度也不及专业翻译人员的水平。

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