Fehlerbehebung mit der IOS-XE DataPath Packet Trace-Funktion

Inhalt

Einleitung

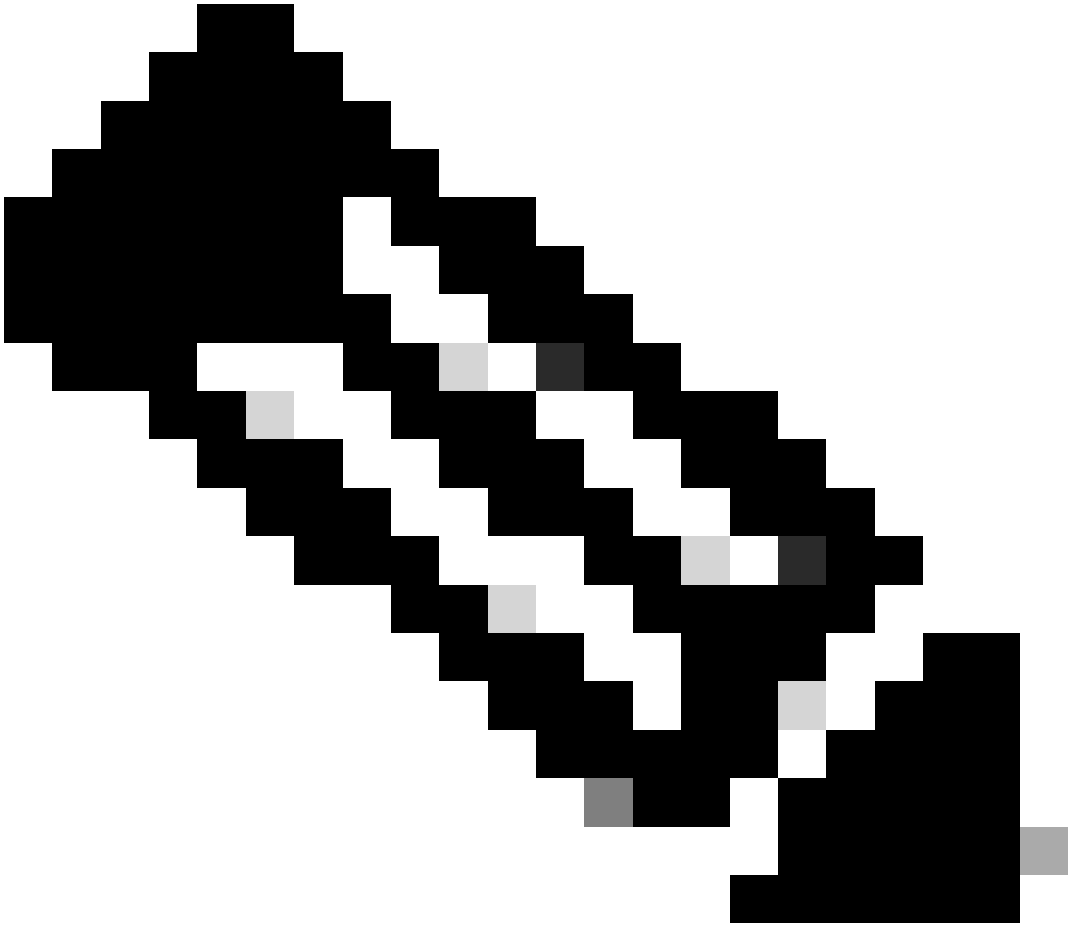
In diesem Dokument wird beschrieben, wie die Paketverfolgung über den Datenpfad für die Cisco IOS-XE®-Software mithilfe der Packet Trace-Funktion durchgeführt wird.

Voraussetzungen

Anforderungen

Cisco empfiehlt, dass Sie diese Informationen kennen:

Die Funktion zur Paketverfolgung ist in Cisco IOS-XE Version 3.10 und höheren Versionen auf QFP-basierten Routing-Plattformen (Quantum Flow Processor) verfügbar, darunter ASR1000, ISR4000, ISR1000, Catalyst 1000, Catalyst 8000 und CSR1 Router der Serien 000v und 8000v. Diese Funktion wird auf den Aggregation Services Routern der Serie ASR900 oder den Switches der Serie Catalyst, auf denen die Cisco IOS-XE Software ausgeführt wird, nicht unterstützt.



> Hinweis: Die Funktion zur Paketverfolgung funktioniert nicht auf der dedizierten Verwaltungsschnittstelle GigabitEthernet0 der Router der Serie ASR 1000, da Pakete, die über diese Schnittstelle weitergeleitet werden, nicht vom QFP verarbeitet werden.

Verwendete Komponenten

Die Informationen in diesem Dokument basierend auf folgenden Software- und Hardware-Versionen:

- Cisco IOS-XE Softwareversion 3.10S (15.3(3)S) und höher
- Router der Serie ASR 1000

Die Informationen in diesem Dokument beziehen sich auf Geräte in einer speziell eingerichteten Testumgebung. Alle Geräte, die in diesem Dokument benutzt wurden, begannen mit einer gelöschten (Nichterfüllungs) Konfiguration. Wenn Ihr Netzwerk in Betrieb ist, stellen Sie sicher, dass Sie die möglichen Auswirkungen aller Befehle kennen.

Hintergrundinformationen

Um Probleme wie Fehlkonfigurationen, Kapazitätsüberlastung oder sogar den normalen Softwarefehler während der Fehlerbehebung zu identifizieren, muss bekannt sein, was mit einem Paket in einem System geschieht. Die Cisco IOS-XE Packet Trace-Funktion erfüllt diese Anforderung. Es stellt eine vor Ort sichere Methode bereit, die für die Abrechnung und zum Erfassen der paketbezogenen Prozessdetails auf der Grundlage einer Klasse benutzerdefinierter Bedingungen verwendet wird.

Referenztopologie

In diesem Diagramm wird die Topologie veranschaulicht, die für die in diesem Dokument beschriebenen Beispiele verwendet wird:

Packet Tracing im Einsatz

Um die Verwendung der Paketablaufverfolgungsfunktion zu veranschaulichen, wird in dem in diesem Abschnitt verwendeten Beispiel eine Ablaufverfolgung des Internet Control Message Protocol (ICMP)-Datenverkehrs von der lokalen Workstation 172.16.10.2 (hinter dem ASR1K) zum Remote-Host 172.16.20.2 in Eingangsrichtung an der Schnittstelle GigabitEthernet0/0/1 auf der ASR1K

Sie können Pakete auf dem ASR1K mit den folgenden zwei Schritten verfolgen:

1. Aktivieren Sie das bedingte Debuggen der Plattform, um die Pakete oder den Datenverkehr auszuwählen, die auf dem ASR1K verfolgt werden sollen.

2. Aktivieren Sie die Plattformpaketverfolgung entweder mit der Option path-trace oder der Option Feature Invocation Array (FIA) trace.

Kurzanleitung

Wenn Sie den Inhalt dieses Dokuments bereits kennen und einen Abschnitt für einen schnellen Überblick über die CLI benötigen, finden Sie hier eine Kurzreferenz. Dies sind nur einige Beispiele, die die Verwendung des Tools veranschaulichen. In den nachfolgenden Abschnitten werden die Syntaxarten detailliert beschrieben. Stellen Sie sicher, dass Sie die für Ihre Anforderungen geeignete Konfiguration verwenden.

1. Konfiguration der Plattformbedingungen:

<#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)

Starten Sie nach der Konfiguration einer Plattformbedingung die Plattformbedingungen mit dem folgenden CLI-Befehl:

<#root>

debug platform condition start

2. Paketverfolgung konfigurieren:

<#root>

debug platform packet-trace packet 1024

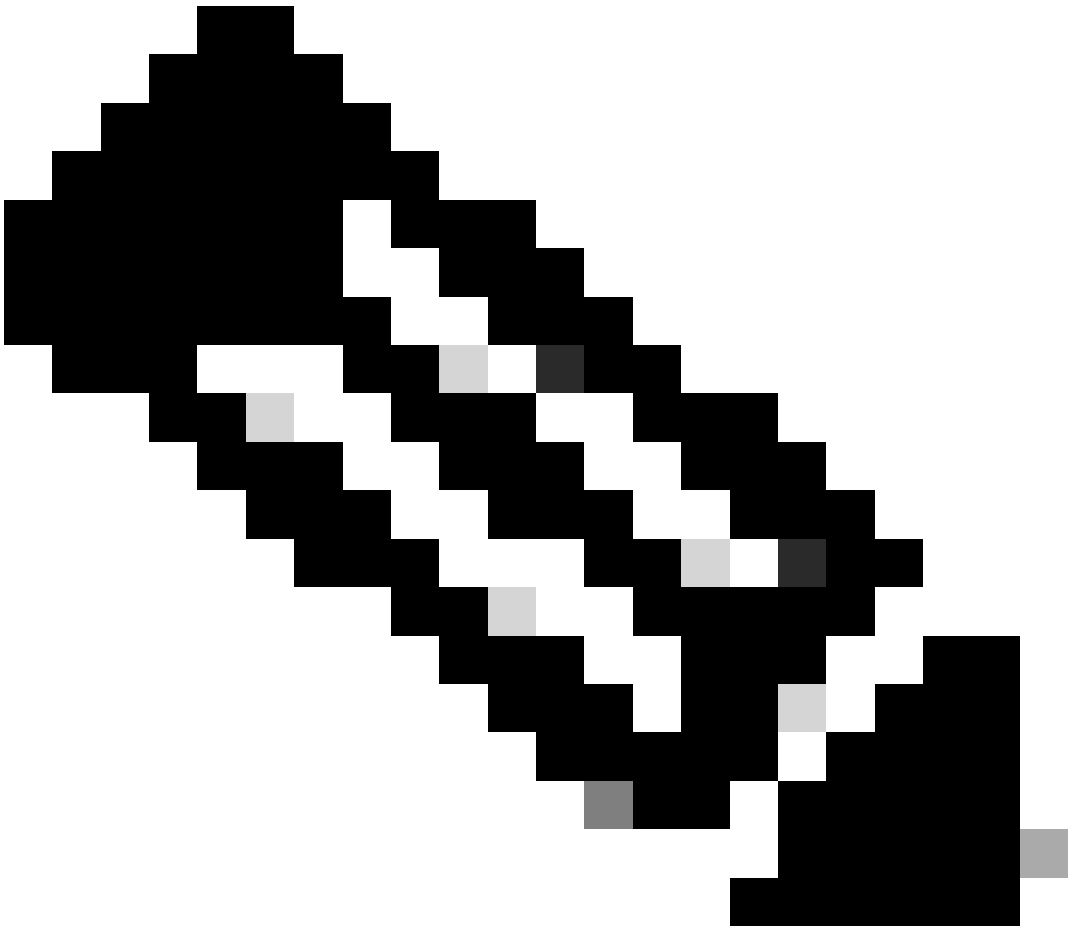
```
  -> basic path-trace, and automatically stops
 tracing packets after 1024 packets. You can use "circular" option if needed
```

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.
```



Hinweis: In früheren Versionen von Cisco IOS-XE 3.x ist der Befehl debug platform

---

packet-trace enable auch zum Starten der Funktion packet-trace erforderlich. In Cisco IOS-XE 16.x ist dies nicht mehr erforderlich.

Geben Sie den folgenden Befehl ein, um den Ablaufverfolgungspuffer zu löschen und "packet-trace" zurückzusetzen:

<#root>

clear platform packet-trace statistics

  --> clear the packet trace buffer

Der Befehl zum Löschen beider Plattformbedingungen und der Paketablaufverfolgungskonfiguration lautet:

<#root>

clear platform condition all

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

Befehle anzeigen

Überprüfen Sie den Plattformzustand und die Konfiguration der Paketverfolgung, nachdem Sie die vorherigen Befehle angewendet haben, um sicherzustellen, dass Sie über die erforderlichen Funktionen verfügen.

<#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

Nachfolgend sind die Befehle zum Überprüfen der verfolgten/erfassten Pakete aufgeführt:

<#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

Bedingte Plattformdebugs aktivieren

Die Packet Trace-Funktion verwendet die Infrastruktur für bedingtes Debuggen, um die zu verfolgenden Pakete zu bestimmen. Die Infrastruktur für bedingtes Debuggen bietet die Möglichkeit, Datenverkehr basierend auf folgenden Kriterien zu filtern:

- Protokolle
- IP-Adresse und -Maske
- Zugriffskontrollliste (ACL)
- Schnittstelle
- Richtung des Datenverkehrs (Eingang oder Ausgang)

Diese Bedingungen legen fest, wo und wann die Filter auf ein Paket angewendet werden.

Aktivieren Sie für den in diesem Beispiel verwendeten Datenverkehr bedingtes Plattform-Debuggen in Eingangsrichtung für ICMP-Pakete von 172.16.10.2 bis 172.16.20.2. Mit anderen Worten: Wählen Sie den Datenverkehr aus, den Sie verfolgen möchten. Es gibt verschiedene Optionen, die Sie verwenden können, um diesen Datenverkehr auszuwählen.

<#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

In diesem Beispiel wird eine Zugriffsliste verwendet, um die Bedingung zu definieren, wie hier gezeigt:

<#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#
```

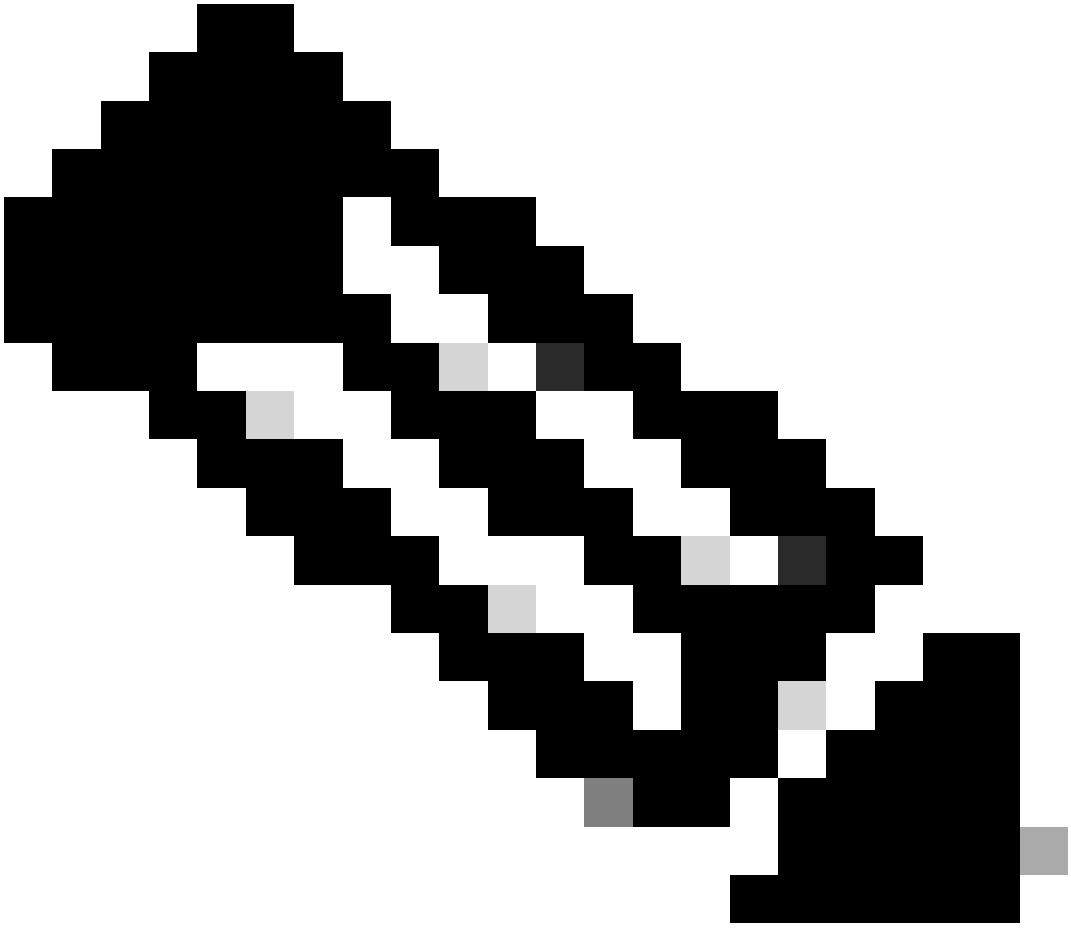
debug platform condition interface gig 0/0/1 ipv4
access-list 150 ingress

Um mit dem bedingten Debuggen zu beginnen, geben Sie den folgenden Befehl ein:

<#root>

ASR1000#

debug platform condition start



Hinweis: Um die Infrastruktur für bedingtes Debuggen zu beenden oder zu deaktivieren, geben Sie den Befehl debug platform condition stop ein.

Geben Sie den folgenden Befehl ein, um die konfigurierten Filter für bedingtes Debuggen anzuzeigen:

<#root>

ASR1000#

show platform conditions

Conditional Debug Global State:

Start

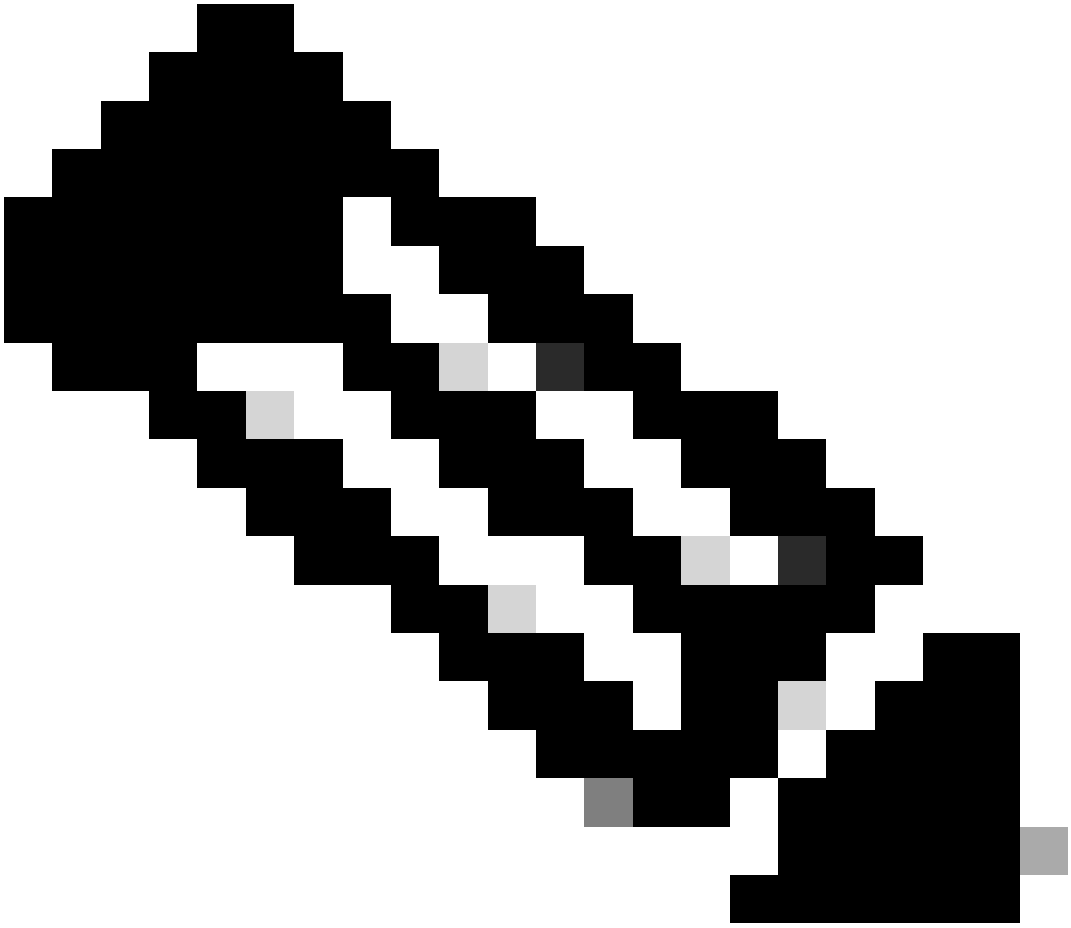
```
Conditions                                                          Direction
--------------------------------------------------------------------|---------
```

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


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

ASR1000#
```

Zusammenfassend lässt sich sagen, dass diese Konfiguration bisher angewandt wurde:

```
<#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
```

Packet Trace aktivieren



Hinweis: In diesem Abschnitt werden die Paket- und Kopieroptionen ausführlich beschrieben. Die anderen Optionen werden weiter unten in diesem Dokument beschrieben.

Packet Traces werden sowohl auf den physischen als auch auf den logischen Schnittstellen unterstützt, z. B. Tunnel- oder Virtual-Access-Schnittstellen.

Die CLI-Syntax für die Paketverfolgung lautet wie folgt:

<#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>]
```

Hier finden Sie Beschreibungen der Schlüsselwörter dieses Befehls:

- pkt-num - Die Paketnummer gibt die maximale Anzahl von Paketen an, die gleichzeitig beibehalten werden.
- summary-only: Gibt an, dass nur die zusammengefassten Daten erfasst werden. Standardmäßig werden sowohl Zusammenfassungsdaten als auch Funktionsdaten erfasst.
- fia-trace - Dieser Befehl führt optional eine FIA-Ablaufverfolgung zusätzlich zu den Pfaddateninformationen aus.
- data-size - Hier können Sie die Größe des Pfaddatenpuffers von 2.048 bis 16.384 Byte angeben. Der Standardwert ist 2.048 Byte.

<#root>

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

Hier finden Sie Beschreibungen der Schlüsselwörter dieses Befehls:

- Eingang/Ausgang - Legt die Richtung des zu kopierenden Paketflusses fest - Eingang und/oder Ausgang.
- L2/L3/L4 - Hier können Sie den Speicherort angeben, an dem die Paketkopie beginnt. Layer 2 (L2) ist der Standardspeicherort.
- size - Hier können Sie die maximale Anzahl der zu kopierenden Oktetts angeben. Der Standardwert ist 64 Achtbitzeichen.

In diesem Beispiel ist dies der Befehl, der verwendet wird, um die Paketverfolgung für den Datenverkehr zu aktivieren, der mit der Infrastruktur für bedingtes Debuggen ausgewählt wird:

<#root>

```
ASR1000#
```

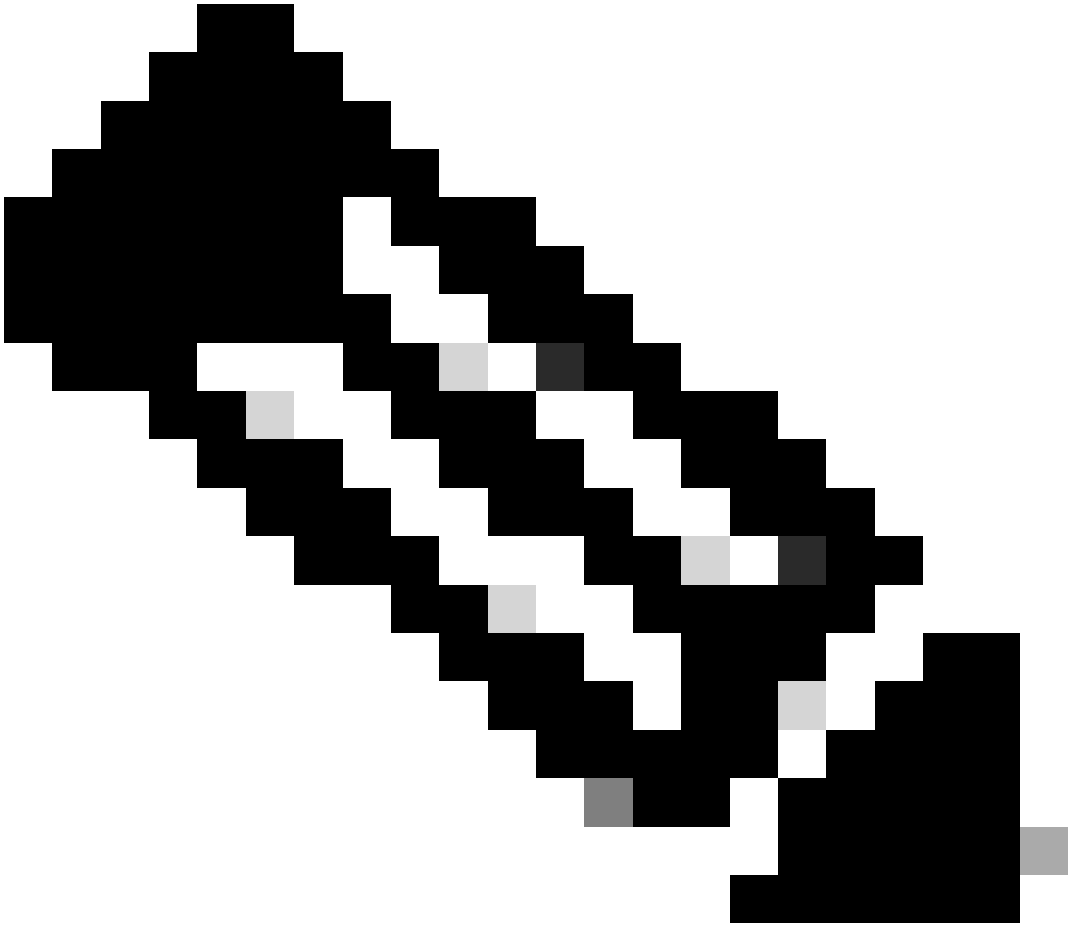
```
debug platform packet-trace packet 16
```

Geben Sie den folgenden Befehl ein, um die Konfiguration der Paketverfolgung zu überprüfen:

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

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

Sie können auch den Befehl show debugging eingeben, um sowohl die bedingten Plattformdebugs als auch die Paketablaufverfolgungskonfigurationen anzuzeigen:

```
<#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
```



Hinweis: Geben Sie den Befehl clear platform condition all ein, um alle Plattformdebugbedingungen sowie die Paketablaufverfolgungskonfigurationen und -daten zu löschen.

Zusammenfassend lässt sich sagen, dass diese Konfigurationsdaten bisher genutzt wurden, um die Paketverfolgung zu ermöglichen:

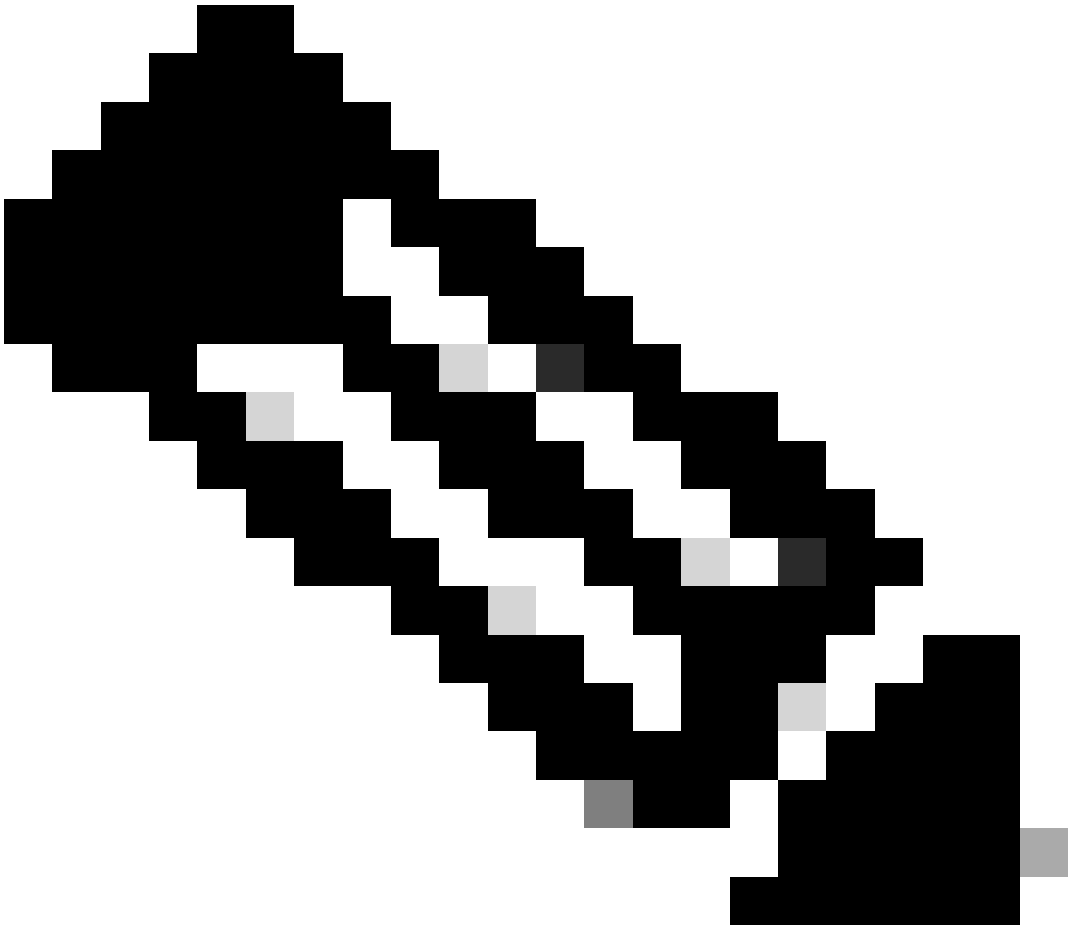
<#root>

debug platform packet-trace packet 16

Beschränkung des Ausgangszustands mit Paketverfolgungen

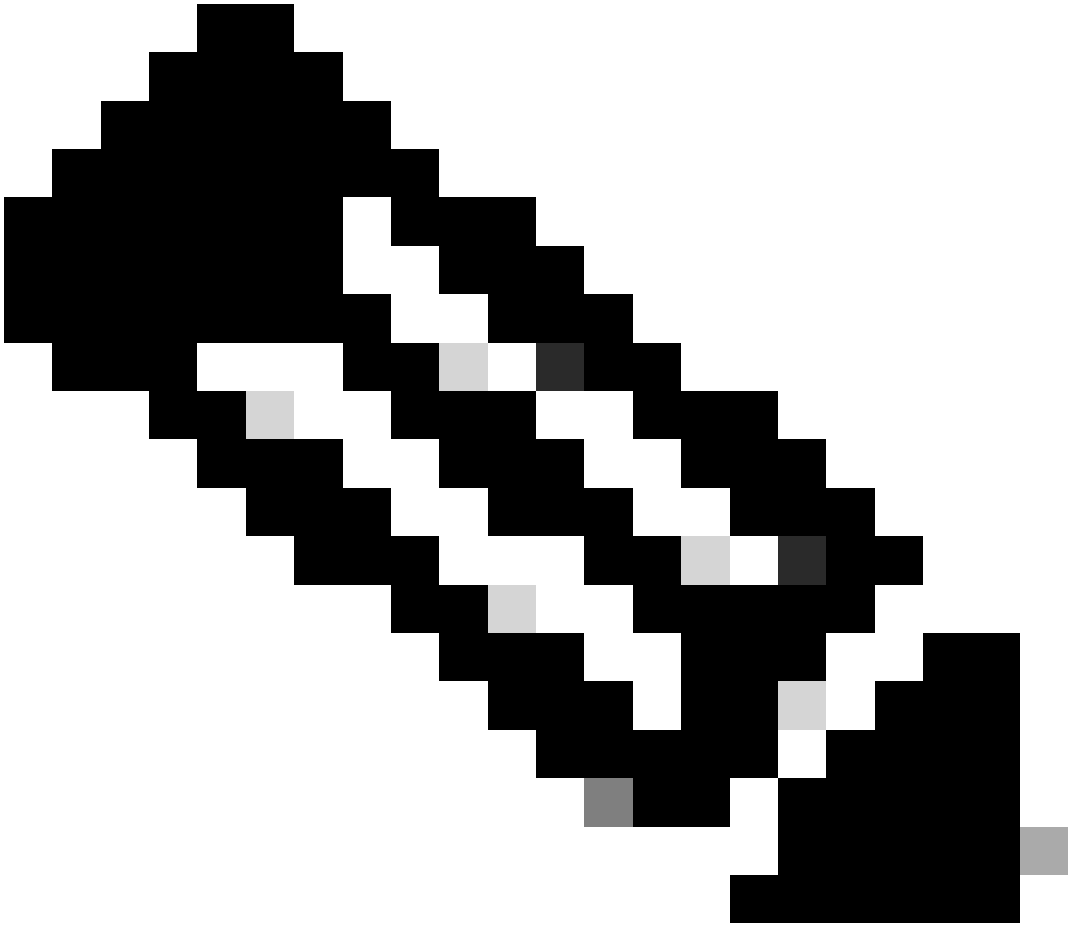
Die Bedingungen definieren die bedingten Filter und den Zeitpunkt, an dem sie auf ein Paket angewendet werden. So bedeutet beispielsweise die Debug-Plattform-Bedingungsschnittstelle

g0/0/0 ausgehend, dass ein Paket als Übereinstimmung identifiziert wird, wenn es den Ausgang FIA an der Schnittstelle g0/0/0 erreicht, sodass jede Paketverarbeitung, die vom Eingang bis zu diesem Punkt stattfindet, verpasst wird.



Hinweis: Cisco empfiehlt dringend, die Eingangsbedingungen für Paket-Traces zu verwenden, um möglichst vollständige und aussagekräftige Daten zu erhalten. Die Ausgangsbedingungen können verwendet werden, aber beachten Sie die Einschränkungen.

Paketverfolgungsergebnisse anzeigen



Hinweis: In diesem Abschnitt wird davon ausgegangen, dass path-trace aktiviert ist.

Die Paketverfolgung bietet drei spezifische Prüfungsstufen:

- Buchhaltung
- Paketbasierte Zusammenfassung
- Paketbasierte Pfaddaten

Wenn fünf ICMP-Anforderungspakete von 172.16.10.2 bis 172.16.20.2 gesendet werden, können die folgenden Befehle verwendet werden, um die Paketverfolgungsergebnisse anzuzeigen:

<#root>

ASR1000#

show platform packet-trace statistics

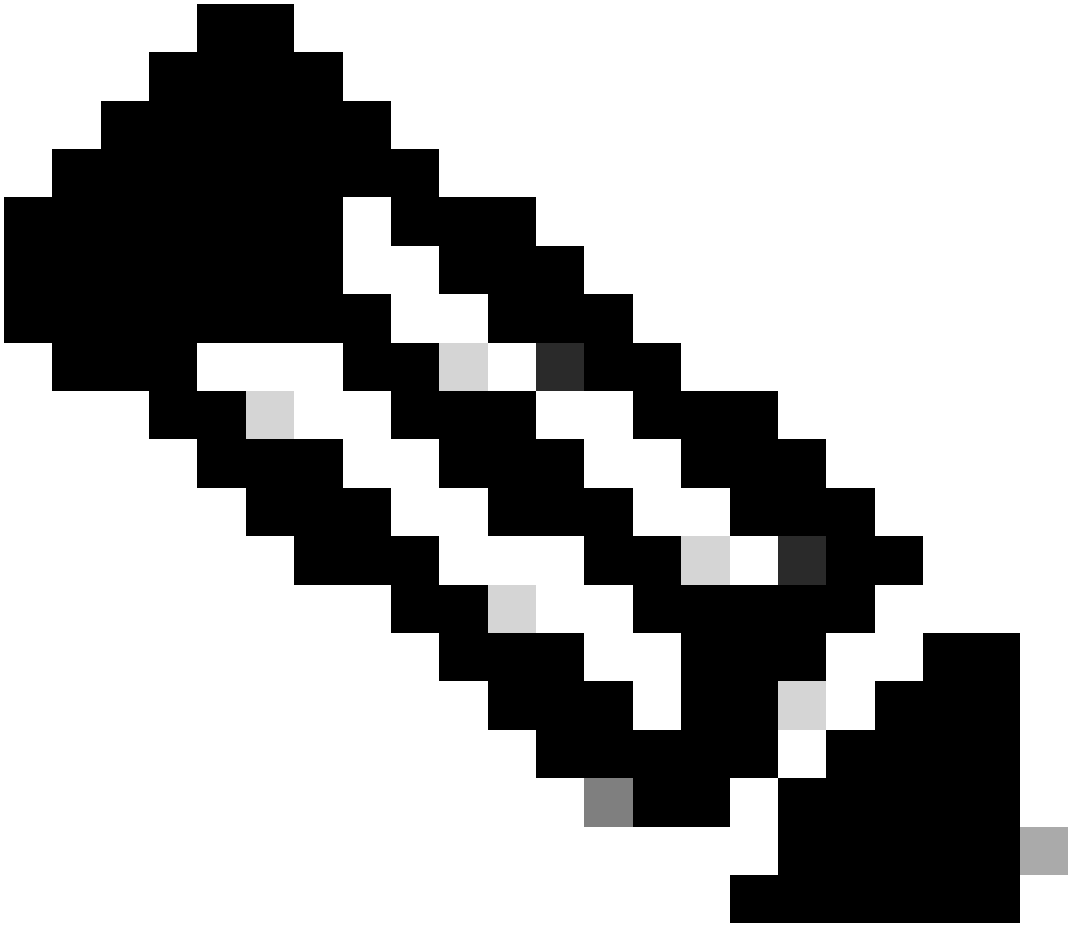
```
Packets Traced: 5


 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#
```



Hinweis: Der dritte Befehl enthält ein Beispiel, das veranschaulicht, wie die Paketverfolgung für jedes Paket angezeigt wird. In diesem Beispiel wird das erste verfolgte Paket angezeigt.

Anhand dieser Ausgaben können Sie sehen, dass fünf Pakete verfolgt werden und dass Sie die Eingangsschnittstelle, die Ausgangsschnittstelle, den Status und die Pfadverfolgung anzeigen können.

Status	Bemerkung
FWD	Das Paket wird für die Zustellung geplant/in die Warteschlange gestellt und über eine Ausgangsschnittstelle an den nächsten Hop weitergeleitet.
PUNKT	Das Paket wird vom Forwarding-Prozessor (FP) zum Route-Prozessor (RP) (Kontrollebene) gesendet.
VERWERFEN	Das Paket wird auf dem FP verworfen. Führen Sie FIA-Ablaufverfolgung aus, verwenden Sie globale Zähler für das Ablegen von Daten, oder verwenden Sie Datenpfad-Debugs, um aus Gründen für das Ablegen weitere Details zu erhalten.

KONTRA	Das Paket wird während eines Paketprozesses verbraucht, z. B. während der ICMP-Ping-Anforderung oder der Crypto-Pakete.

Die Zähler für den Eingang und den Eingang in der Paketverfolgungsstatistik entsprechen den Paketen, die über eine externe Schnittstelle eingehen, bzw. den Paketen, die als von der Kontrollebene eingekoppelt betrachtet werden.

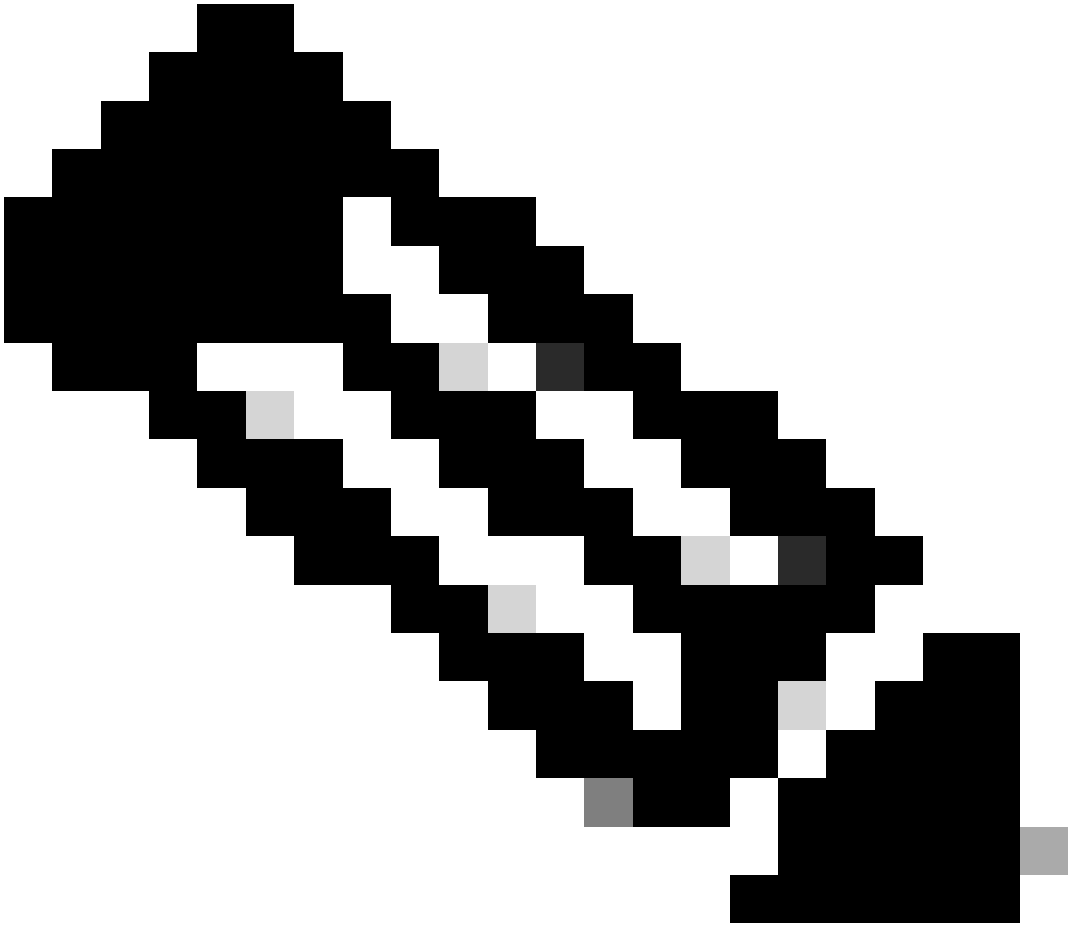
FIA-Ablaufverfolgung

Die FIA enthält die Liste der Funktionen, die sequenziell von den Packet Processor Engines (PPE) im Quantum Flow Processor (QFP) ausgeführt werden, wenn ein Paket ein- oder ausgeht. Die Funktionen basieren auf den Konfigurationsdaten, die auf den Computer angewendet werden. So hilft eine FIA-Ablaufverfolgung, den Paketfluss durch das System zu verstehen, während das Paket verarbeitet wird.

Sie müssen diese Konfigurationsdaten anwenden, um die Paketverfolgung mit FIA zu aktivieren:

```
<#root>

ASR1000#

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

Paketverfolgungsergebnisse anzeigen



Hinweis: In diesem Abschnitt wird davon ausgegangen, dass die FIA-Ablaufverfolgung aktiviert ist. Wenn Sie die aktuellen Paketablaufverfolgungsbefehle hinzufügen oder ändern, werden auch die gepufferten Paketablaufverfolgungsdetails gelöscht. Sie müssen daher erneut Datenverkehr senden, um ihn nachverfolgen zu können.

Senden Sie fünf ICMP-Pakete von 172.16.10.2 an 172.16.20.2, nachdem Sie den Befehl eingegeben haben, der verwendet wird, um die FIA-Ablaufverfolgung zu aktivieren, wie im vorherigen Abschnitt beschrieben.

<#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#
```

Prüfen der mit einer Schnittstelle verknüpften FIA

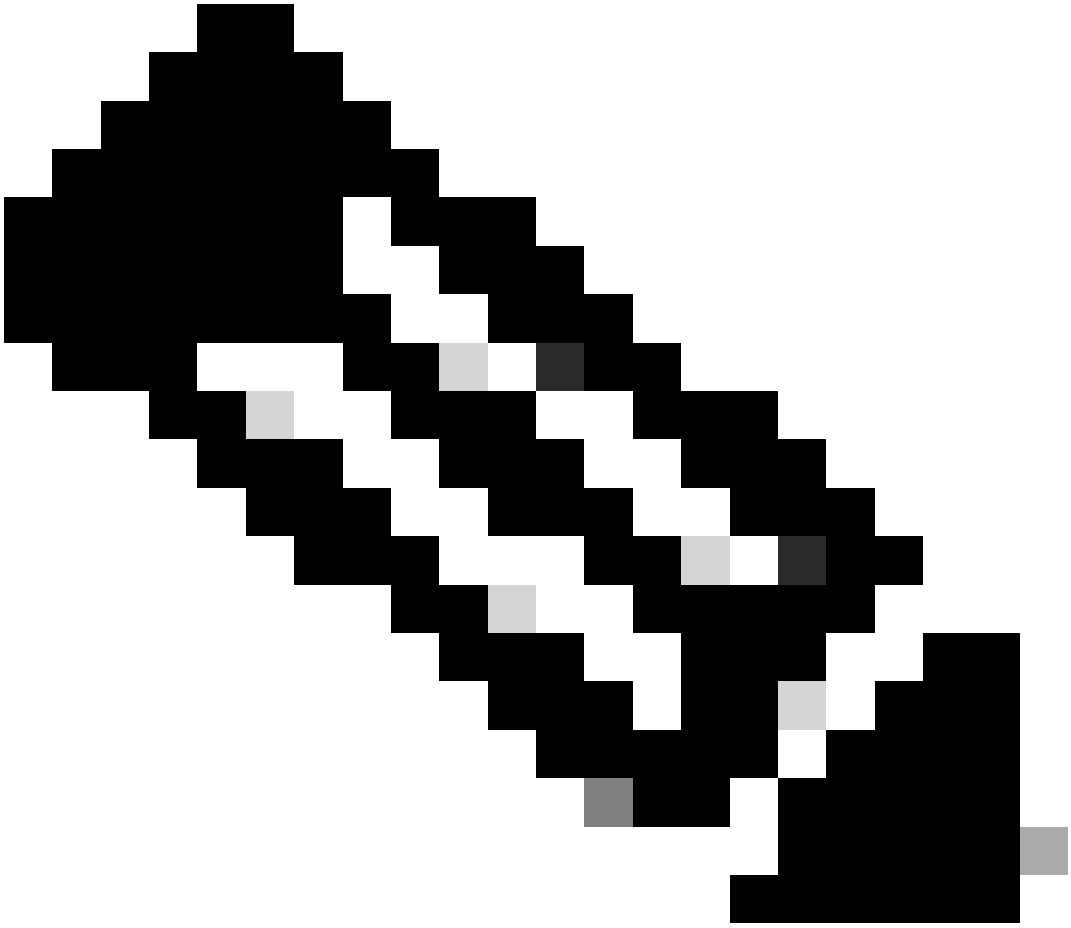
Wenn Sie das bedingte Debuggen der Plattform aktivieren, wird das bedingte Debuggen der FIA als Funktion hinzugefügt. Abhängig von der Funktionsreihenfolge der Schnittstellenverarbeitung muss der Bedingungsfilter entsprechend eingestellt werden, beispielsweise ob die Vor- oder Nachadresse im Bedingungsfilter verwendet werden muss.

Diese Ausgabe zeigt die Reihenfolge der Features in der FIA für das bedingte Plattformdebuggen, das in Eingangsrichtung aktiviert ist:

<#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#
```



Hinweis: CBUG_INPUT_FIA und DEBUG_COND_INPUT_PKT entsprechen den auf dem Router konfigurierten bedingten Debugfunktionen.

Zurückverfolgte Pakete ausgeben

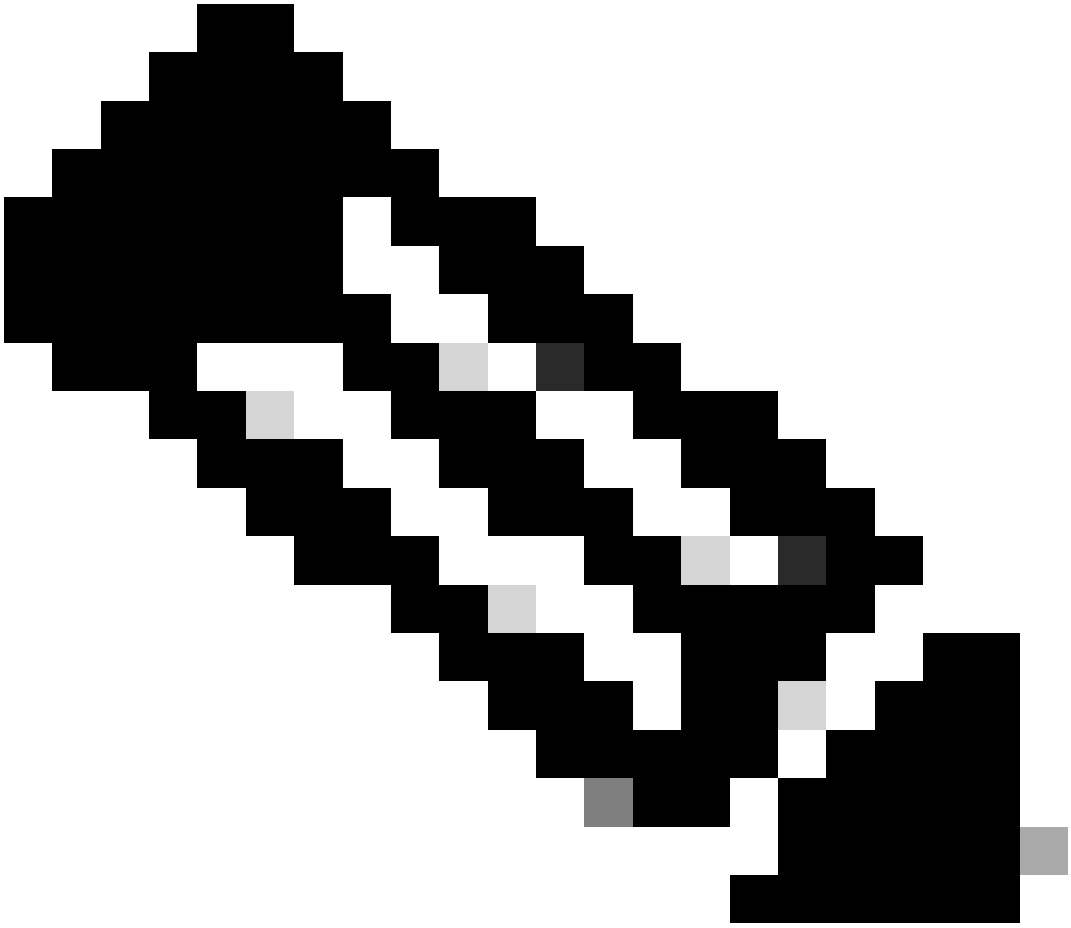
Sie können die Pakete während der Ablaufverfolgung kopieren und auslesen, wie in diesem Abschnitt beschrieben. In diesem Beispiel werden maximal 2.048 Byte der Pakete in Eingangsrichtung (172.16.10.2 bis 172.16.20.2) kopiert.

Der folgende zusätzliche Befehl wird benötigt:

<#root>

ASR1000#

debug platform packet-trace copy packet input size 2048



Hinweis: Die Größe des kopierten Pakets liegt im Bereich von 16 bis 2.048 Byte.

Geben Sie den folgenden Befehl ein, um die kopierten Pakete zu sichern:

<#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#
```

Trace ablegen

Drop-Trace ist ab Version 3.11 der Cisco IOS-XE Software verfügbar. Es aktiviert die Paketverfolgung nur für verworfene Pakete. Hier einige Highlights der Funktion:

- Optional können Sie die Beibehaltung von Paketen für einen bestimmten Dropcode festlegen.
- Sie kann ohne globale Bedingungen oder Schnittstellenbedingungen verwendet werden, um Löschereignisse zu erfassen.
- Eine Erfassung von Ereignissen bei einem Verwerfen bedeutet, dass nur der Verwerfen selbst verfolgt wird, nicht die Lebensdauer des Pakets. Es ermöglicht Ihnen jedoch weiterhin, zusammengefasste Daten, Tupeldaten und das Paket zu erfassen, um die Bedingungen zu verfeinern oder Hinweise auf den nächsten Debugschritt bereitzustellen.

Die Befehlssyntax zum Aktivieren von Drop-Typ-Paketablaufverfolgungen lautet wie folgt:

<#root>

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

Der Drop-Code entspricht der Drop-ID, die in der Ausgabe des Befehls show platform hardware qfp active statistics drop detail ausgegeben wird:

<#root>

```
ASR1000#
```

```
show platform hardware qfp active statistics drop detail

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

Beispiel-Fallverfolgungsszenario

Wenden Sie diese ACL auf die Gig 0/0/0-Schnittstelle des ASR1K an, um Datenverkehr von 172.16.10.2 auf 172.16.20.2 zu verwerfen:

```
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
```

Wenn die ACL eingerichtet ist, die den Datenverkehr vom lokalen Host zum Remote-Host verwirft, wenden Sie die folgende Konfiguration für den Drop-Trace an:

<#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
```

Senden Sie fünf ICMP-Anforderungspakete von 172.16.10.2 an 172.16.20.2. Die Ablaufverfolgung erfasst die folgenden Pakete, die von der ACL verworfen werden:

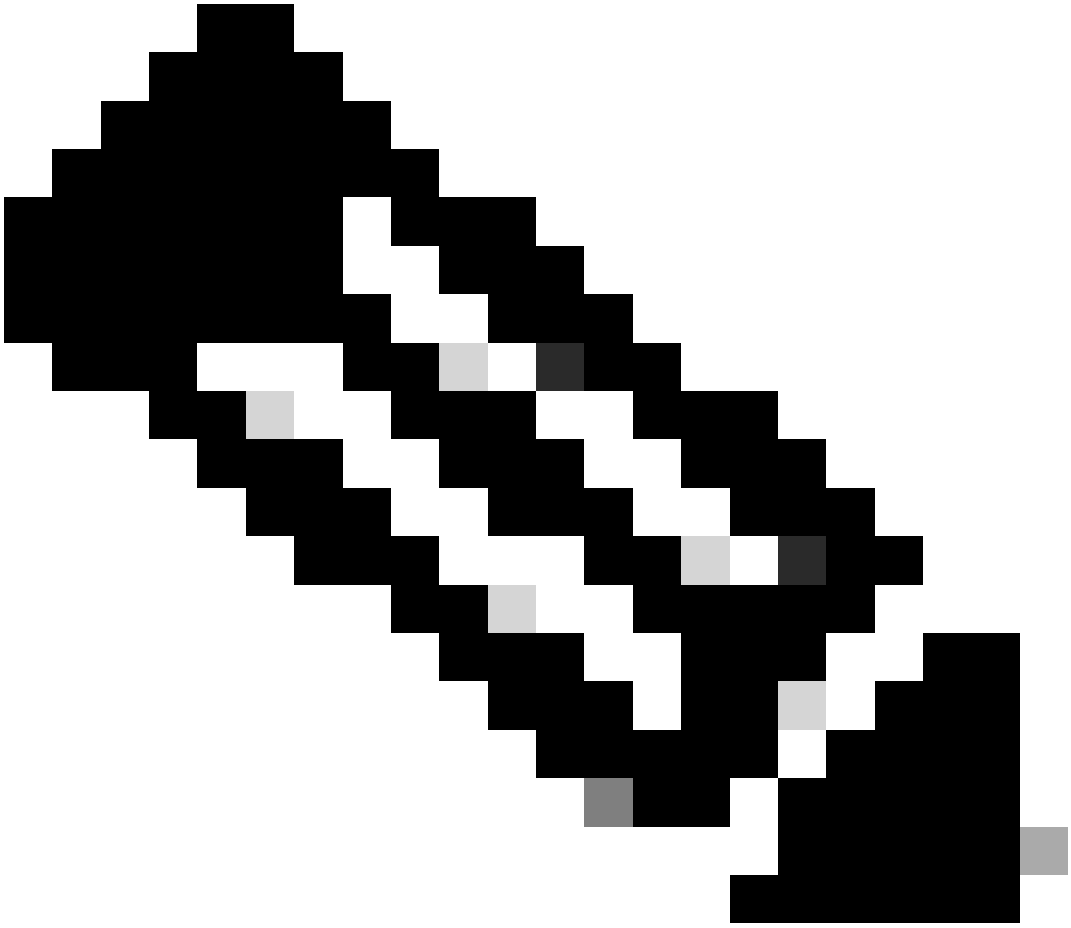
<#root>

```
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#
```

Spuren injizieren und stanzen

Die Funktion inject und punt packet trace wurde in Cisco IOS-XE Software Release 3.12 und höher hinzugefügt, um punt-Pakete (Pakete, die auf dem FP empfangen werden und auf die Kontrollebene gelocht werden) und inject-Pakete (Pakete, die vom Kontrollebene auf den FP eingespeist werden) nachzuverfolgen.



Hinweis: Die Punt-Ablaufverfolgung kann ohne die globalen oder Schnittstellenbedingungen funktionieren, genau wie eine Drop-Ablaufverfolgung. Die Bedingungen müssen jedoch definiert werden, damit eine Injection-Trace funktioniert.

Das folgende Beispiel zeigt ein punt und inject packet trace , wenn Sie einen Ping vom ASR1K an einen benachbarten Router senden:

<#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#
```

Jetzt können Sie die punt und die nject trace rErgebnisse überprüfen:

<#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)
```

Packet Trace-Optimierung mit IOSd und LFTS Punt/Inject Trace und UDF-Anpassung (neu in 17.3.1)

Die Paketablaufverfolgungsfunktion wurde weiter verbessert, um zusätzliche Ablaufverfolgungsinformationen für Pakete bereitzustellen, die von IOSd oder anderen BinOS-Prozessen in Cisco IOS-XE Version 17.3.1 stammen oder für diese bestimmt sind.

IOSd Drop Tracing

Mit dieser Erweiterung wird die Paket-Nachverfolgung auf IOSd ausgedehnt und kann Informationen über alle Paketverluste innerhalb von IOSd liefern, die in der Regel in der Ausgabe von ***show ip traffic*** gemeldet werden. Es ist keine zusätzliche Konfiguration erforderlich, um die IOSd-Ablaufverfolgung zu aktivieren. Das folgende Beispiel zeigt ein UDP-Paket, das von IOSd aufgrund eines fehlerhaften Prüfsummenfehlers verworfen wurde:

```
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-Ausgangspfad-Verfolgung

Die Paketverfolgung wird verbessert, um die Informationen zur Pfadverfolgung und Protokollverarbeitung anzuzeigen, wenn das Paket von IOSd stammt und in Ausgangsrichtung an das Netzwerk gesendet wird. Es ist keine zusätzliche Konfiguration erforderlich, um die IOSd-Ausgangspfadverfolgungsinformationen zu erfassen. Das folgende Beispiel zeigt die Ablaufverfolgung des Ausgangspfades für ein SSH-Paket, das den Router verlässt:

<#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-Paketverfolgung

LFTS (Linux Forwarding Transport Service) ist ein Transportmechanismus zum Weiterleiten von Paketen, die von CPP gesendet werden, an andere Anwendungen als IOSd. Die LFTS-Paketablaufverfolgungs-Verbesserung hat Ablaufverfolgungsinformationen für solche Pakete in der Pfadablaufverfolgungsausgabe hinzugefügt. Zum Abrufen der LFTS-Ablaufverfolgungsinformationen ist keine zusätzliche Konfiguration erforderlich. Hier ist ein Beispiel für die Ausgabe der LFTS-Ablaufverfolgung für Paket-Punts an die NETCONF-Anwendung:

<#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
```

Abgleich der Paketablaufmuster auf Basis des benutzerdefinierten Filters (nur ASR1000-Plattform)

In Cisco IOS-XE Version 17.3.1 wird den ASR1000-Produktfamilien ein neuer Paketvergleichsmechanismus hinzugefügt, um einen Abgleich in einem beliebigen Feld in einem Paket basierend auf der UDF-Infrastruktur (User Defined Filter) zu ermöglichen. Dies ermöglicht eine flexible Paketzuordnung auf der Grundlage von Feldern, die nicht Teil der standardmäßigen L2/L3/L4-Headerstruktur sind. Das nächste Beispiel zeigt eine UDF-Definition, die mit 2 Byte eines benutzerdefinierten Musters von 0x4D2 übereinstimmt, das mit einem Offset von 26 Byte vom äußeren L3-Protokoll-Header beginnt.

```
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
```

Packet Trace-Beispiele

In diesem Abschnitt finden Sie einige Beispiele, in denen die Paketablaufverfolgungsfunktion zur Fehlerbehebung nützlich ist.

Beispiel für Packet Trace - NAT

In diesem Beispiel wird eine Network Address Translation (NAT) als Schnittstellenquelle auf der WAN-Schnittstelle eines ASR1K (Gig0/0/0) für das lokale Subnetz (172.16.10.0/24) konfiguriert.

Nachfolgend finden Sie die Plattformbedingung und die Konfiguration der Paketverfolgung, die verwendet wird, um den Datenverkehr von 172.16.10.2 bis 172.16.20.2 zu verfolgen, der auf der Gig0/0/0-Schnittstelle umgewandelt wird:

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

Wenn fünf ICMP-Pakete von 172.16.10.2 an 172.16.20.2 mit einer NAT-Konfiguration für die Schnittstellenquelle gesendet werden, sind dies die Paketverfolgungsergebnisse:

```
      <#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#
```

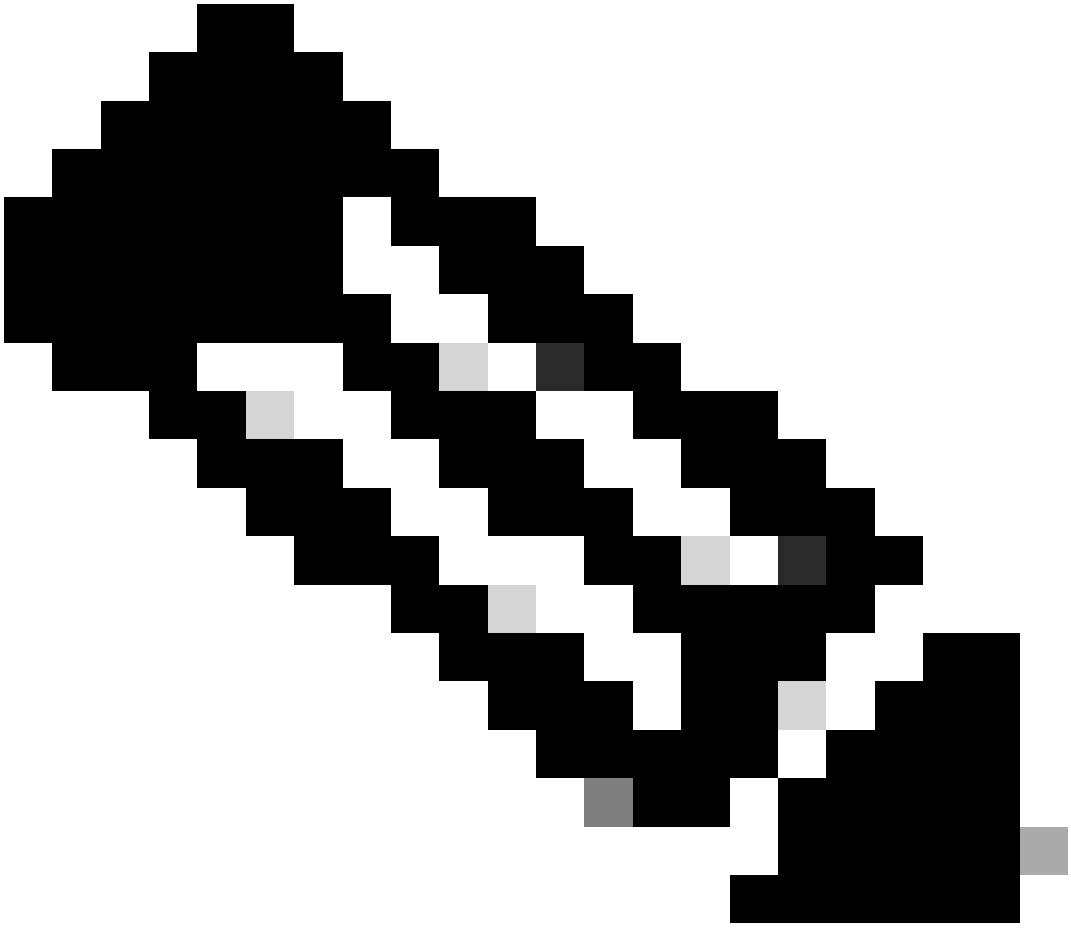
Beispiel für Packet Trace - VPN

In diesem Beispiel wird ein Site-to-Site-VPN-Tunnel zwischen dem ASR1K und dem Cisco IOS-Router verwendet, um den Datenverkehr zwischen 172.16.10.0/24 und 172.16.20.0/24 (lokale und Remote-Subnetze) zu schützen.

Nachfolgend finden Sie die Plattformbedingung und die Konfiguration der Paketverfolgung, die verwendet wird, um den VPN-Datenverkehr zu verfolgen, der von 172.16.10.2 bis 172.16.20.2 auf der Gig 0/0/1-Schnittstelle fließt:

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

Wenn fünf ICMP-Pakete von 172.16.10.2 bis 172.16.20.2 gesendet werden, die durch den VPN-Tunnel zwischen dem ASR1K und dem Cisco IOS-Router in diesem Beispiel verschlüsselt werden, sind dies die Paketverfolgungs-Ausgaben:



Hinweis: Die Paketnachverfolgungen zeigen das QFP Security Association (SA)-Handle in der Ablaufverfolgung an, die zur Verschlüsselung des Pakets verwendet wird. Dies ist nützlich, wenn Sie IPsec-VPN-Probleme beheben, um sicherzustellen, dass die richtige SA für die Verschlüsselung verwendet wird.

<#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#
```

Auswirkungen auf die Leistung

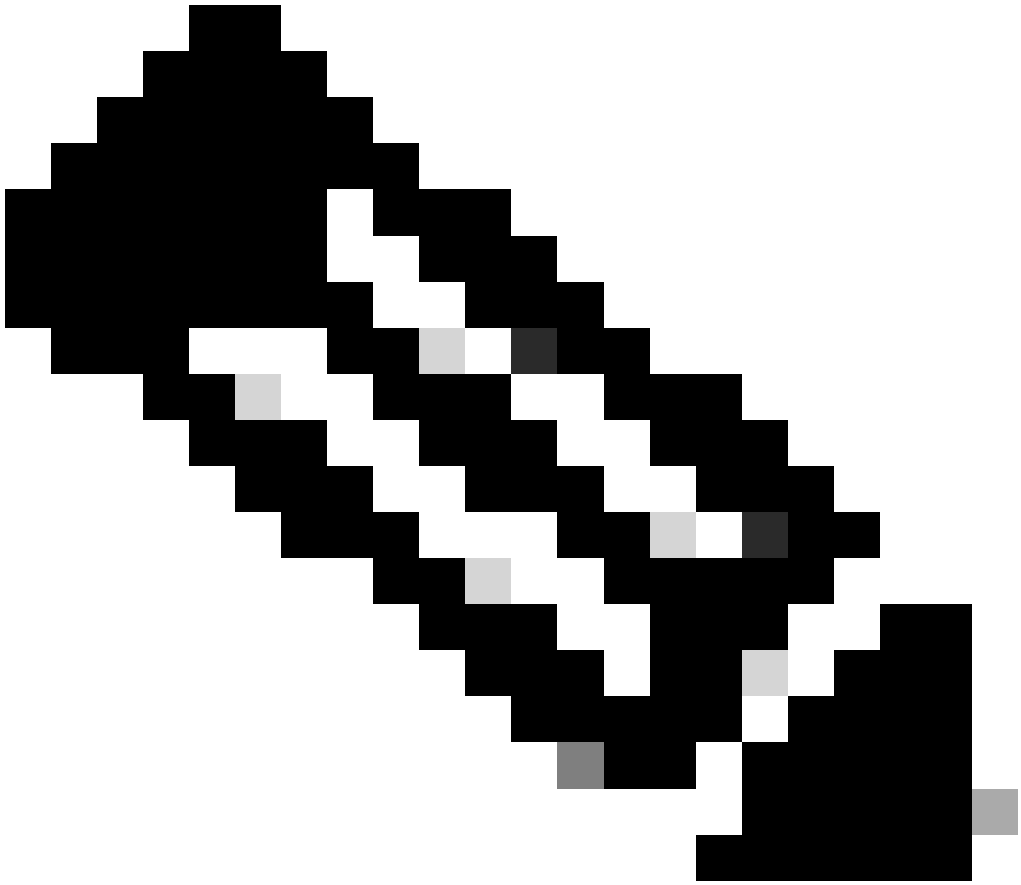
Paketablaufverfolgungspuffer belegen QFP DRAM. Achten Sie daher auf die Menge an Speicher, die für eine Konfiguration erforderlich ist, und auf die Menge an verfügbarem Speicher.

Die Auswirkungen auf die Leistung variieren je nach den aktivierten Optionen für die Paketverfolgung. Die Paketverfolgung wirkt sich nur auf die Weiterleitungsleistung der verfolgten Pakete aus, z. B. der Pakete, die den benutzerdefinierten Bedingungen entsprechen. Je detaillierter und detaillierter die Informationen sind, die Sie für die Paketerfassung konfigurieren, desto umfangreicher können sich diese auf die Ressourcen auswirken.

Wie bei jeder Fehlerbehebung empfiehlt es sich, einen iterativen Ansatz zu wählen und nur dann die detaillierteren Ablaufverfolgungsoptionen zu aktivieren, wenn eine Debugsituation dies erfordert.

Die Verwendung von QFP-DRAMs kann anhand der folgenden Formel geschätzt werden:

Benötigter Arbeitsspeicher = (statistischer Overhead) + Anzahl der Pakete * (Zusammenfassungsgröße + Pfaddatengröße + Kopiergröße)



Hinweis: Wenn der **Overhead** für **Statistiken** und die **Zusammenfassungsgröße** auf 2 KB bzw. 128 B festgelegt sind, sind die **Pfaddatengröße** und die **Kopiegröße** vom Benutzer konfigurierbar**.**

Zugehörige Informationen

- **Software-Konfigurationsanleitung für Cisco Router der Aggregation-Serie ASR1000 - Packet Trace**
- **Paketverluste bei Cisco Services Routern der Serie ASR 1000**
- **Technischer Support und Downloads von Cisco**

Informationen zu dieser Übersetzung

Cisco hat dieses Dokument maschinell übersetzen und von einem menschlichen Übersetzer editieren und korrigieren lassen, um unseren Benutzern auf der ganzen Welt Support-Inhalte in ihrer eigenen Sprache zu bieten. Bitte beachten Sie, dass selbst die beste maschinelle Übersetzung nicht so genau ist wie eine von einem professionellen Übersetzer angefertigte. Cisco Systems, Inc. übernimmt keine Haftung für die Richtigkeit dieser Übersetzungen und empfiehlt, immer das englische Originaldokument (siehe bereitgestellter Link) heranzuziehen.