IOS-XE Datapath 패킷 추적 기능으로 문제 해결

목차

소개

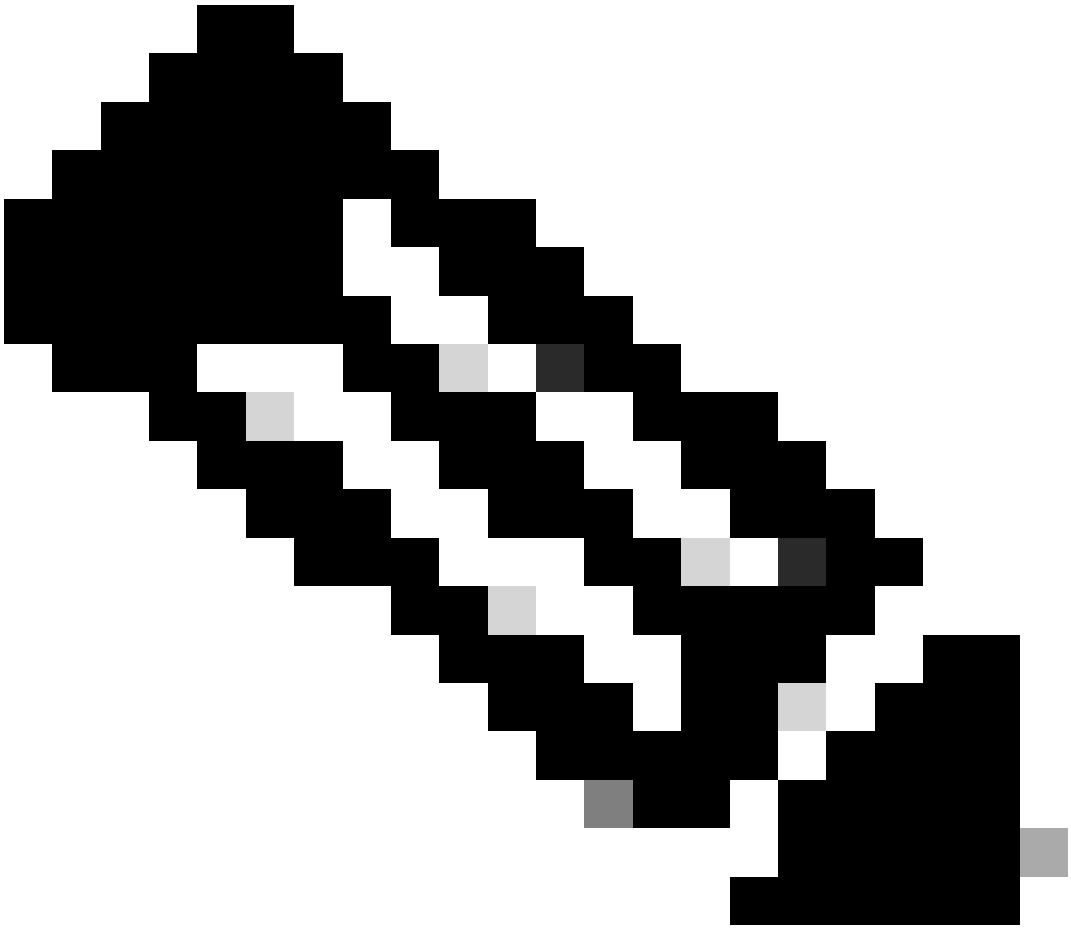
이 문서에서는 패킷 추적 기능을 통해 Cisco IOS-XE® 소프트웨어의 데이터 경로 패킷 추적을 수행하는 방법에 대해 설명합니다.

사전 요구 사항

요구 사항

Cisco에서는 다음 정보를 숙지할 것을 권장합니다.

패킷 추적 기능은 ASR1000, ISR4000, ISR1000, Catalyst 1000, Catalyst 8000, CSR1000v 및 Catalyst 8000v 시리즈 라우터를 포함하는 QFP(Quantum Flow Processor) 기반 라우팅 플랫폼에서 Cisco IOS-XE 버전 3.10 이상에서 사용할 수 있습니다. 이 기능은 Cisco IOS-XE 소프트웨어를 실행하는 ASR900 Series Aggregation Services 라우터 또는 Catalyst Series 스위치에서는 지원되지 않습니다.



참고: 패킷 추적 기능은 ASR1000 Series 라우터의 전용 관리 인터페이스인 GigabitEthernet0에서 작동하지 않습니다. 해당 인터페이스에서 전달된 패킷은 QFP에서 처리되지 않기 때문입니다.

사용되는 구성 요소

이 문서의 정보는 다음 소프트웨어 및 하드웨어 버전을 기반으로 합니다.

- Cisco IOS-XE Software 릴리스 3.10S(15.3(3)S) 이상
- ASR1000 Series 라우터

이 문서의 정보는 특정 랩 환경의 디바이스를 토대로 작성되었습니다. 이 문서에 사용된 모든 디바이스는 초기화된(기본) 컨피그레이션으로 시작되었습니다. 현재 네트워크가 작동 중인 경우 모든 명령의 잠재적인 영향을 미리 숙지하시기 바랍니다.

배경 정보

트러블슈팅 중에 컨피그레이션 오류, 용량 오버로드 또는 일반 소프트웨어 버그와 같은 문제를 식별하려면 시스템 내의 패킷에 어떤 일이 발생하는지 이해해야 합니다. Cisco IOS-XE 패킷 추적 기능은 이러한 요구 사항을 해결합니다. 사용자 정의 조건 클래스를 기반으로 패킷별 프로세스 세부 사항을 캡처하기 위해 계정 관리에 사용되는 필드 안전 방법을 제공합니다.

참조 토폴로지

이 다이어그램은 이 문서에서 설명하는 예제에 사용되는 토폴로지를 보여줍니다.

사용 중인 패킷 추적

패킷 추적 기능의 사용을 설명하기 위해 이 절 전체에서 사용되는 예는 ASR1K의 인터페이스 GigabitEthernet0/0/1에서 인그레스 방향으로 로컬 워크스테이션 172.16.10.2(ASR1K 뒤)에서 원격 호스트 172.16.20.2로 이동하는 ICMP(Internet Control Message Protocol) 트래픽의 추적을 설명합니다.

다음 두 단계를 통해 ASR1K에서 패킷을 추적할 수 있습니다.

1. ASR1K에서 추적하려는 패킷 또는 트래픽을 선택하려면 플랫폼 조건부 디버그를 활성화합니다.

2. path-trace 또는 FIA(Feature Invocation Array) 추적 옵션으로 플랫폼 패킷 추적을 활성화합니다.

빠른 시작 설명서

이 문서의 내용을 이미 잘 알고 있고 CLI를 빠르게 살펴볼 수 있는 섹션이 필요한 경우 여기를 클릭하여 빠른 시작 가이드를 참조하십시오. 다음은 툴 사용을 설명하기 위한 몇 가지 예입니다. 구문에 대해 자세히 설명하는 다음 섹션을 참조하여 요구 사항에 적합한 컨피그레이션을 사용하십시오.

1. 플랫폼 조건을 구성합니다.

<#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)

플랫폼 조건이 구성된 후 다음 CLI 명령을 사용하여 플랫폼 조건을 시작합니다.

<#root>

debug platform condition start

2. 패킷 추적을 구성합니다.

<#root>

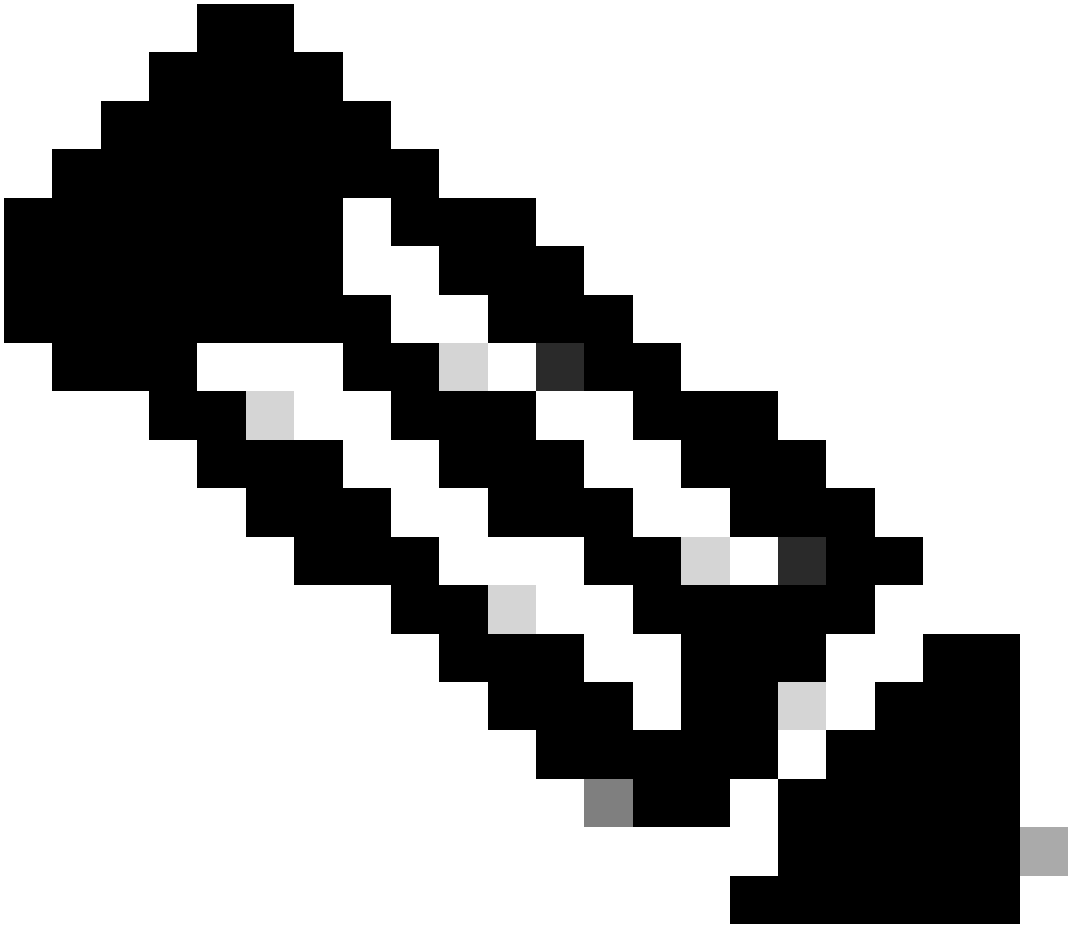
debug platform packet-trace packet 1024

  -> 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.
```



참고: 이전 Cisco IOS-XE 3.x 릴리스에서는 패킷 추적 기능을 시작하려면 debug platform packet-trace enable 명령도 필요합니다. Cisco IOS XE 16.x 릴리스에서는 더 이상 이 기능이 필요하지 않습니다.

추적 버퍼를 지우고 패킷 추적을 재설정하려면 다음 명령을 입력합니다.

<#root>

```
clear platform packet-trace statistics

  --> clear the packet trace buffer
```

플랫폼 조건과 패킷 추적 컨피그레이션을 모두 지우는 명령은 다음과 같습니다.

<#root>

clear platform condition all

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

명령 표시

필요한 것을 확보하기 위해 이전 명령을 적용한 후 플랫폼 조건 및 패킷 추적 컨피그레이션을 확인합니다.

<#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

추적된/캡처된 패킷을 확인하는 명령은 다음과 같습니다.

<#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

플랫폼 조건부 디버깅 사용

패킷 추적 기능은 추적할 패킷을 결정하기 위해 조건부 디버그 인프라를 사용합니다. 조건부 디버그 인프라는 다음을 기반으로 트래픽을 필터링할 수 있는 기능을 제공합니다.

- 프로토콜
- IP 주소 및 마스크
- ACL(Access Control List)
- 인터페이스
- 트래픽 방향(인그레스 또는 이그레스)

이러한 조건은 필터가 패킷에 적용되는 위치와 시기를 정의합니다.

이 예에서 사용되는 트래픽의 경우 172.16.10.2~172.16.20.2의 ICMP 패킷에 대해 인그레스 방향으로 플랫폼 조건부 디버그를 활성화합니다. 즉, 추적하고자 하는 트래픽을 선택합니다. 이 트래픽을 선택하기 위해 사용할 수 있는 다양한 옵션이 있습니다.

```
<#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
```

이 예에서는 다음 그림과 같이 액세스 목록을 사용하여 조건을 정의합니다.

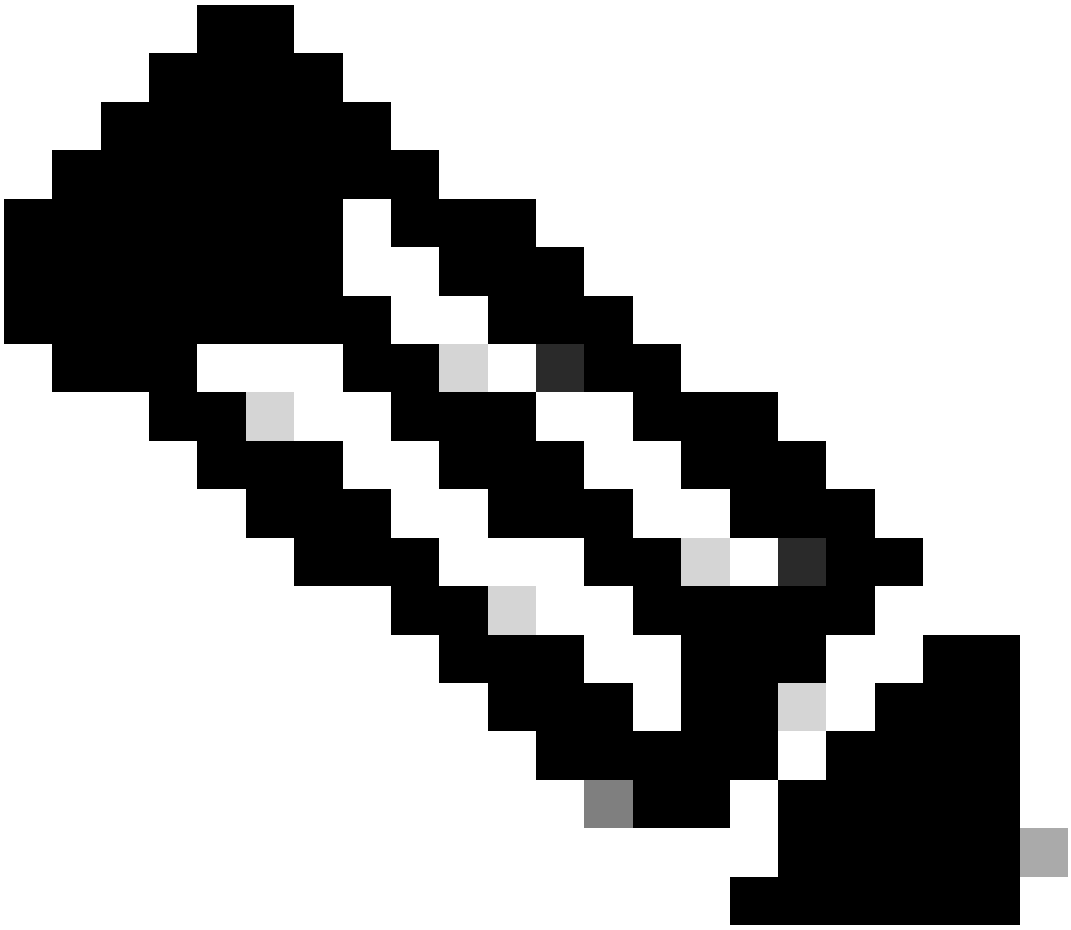
```
<#root>

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
```

조건부 디버깅을 시작하려면 다음 명령을 입력합니다.

<#root>

ASR1000#

debug platform condition start



참고: 조건부 디버깅 인프라를 중지하거나 비활성화하려면 debug platform condition stop 명령을 입력합니다.

구성된 조건부 디버그 필터를 보려면 다음 명령을 입력합니다.

<#root>

ASR1000#

show platform conditions

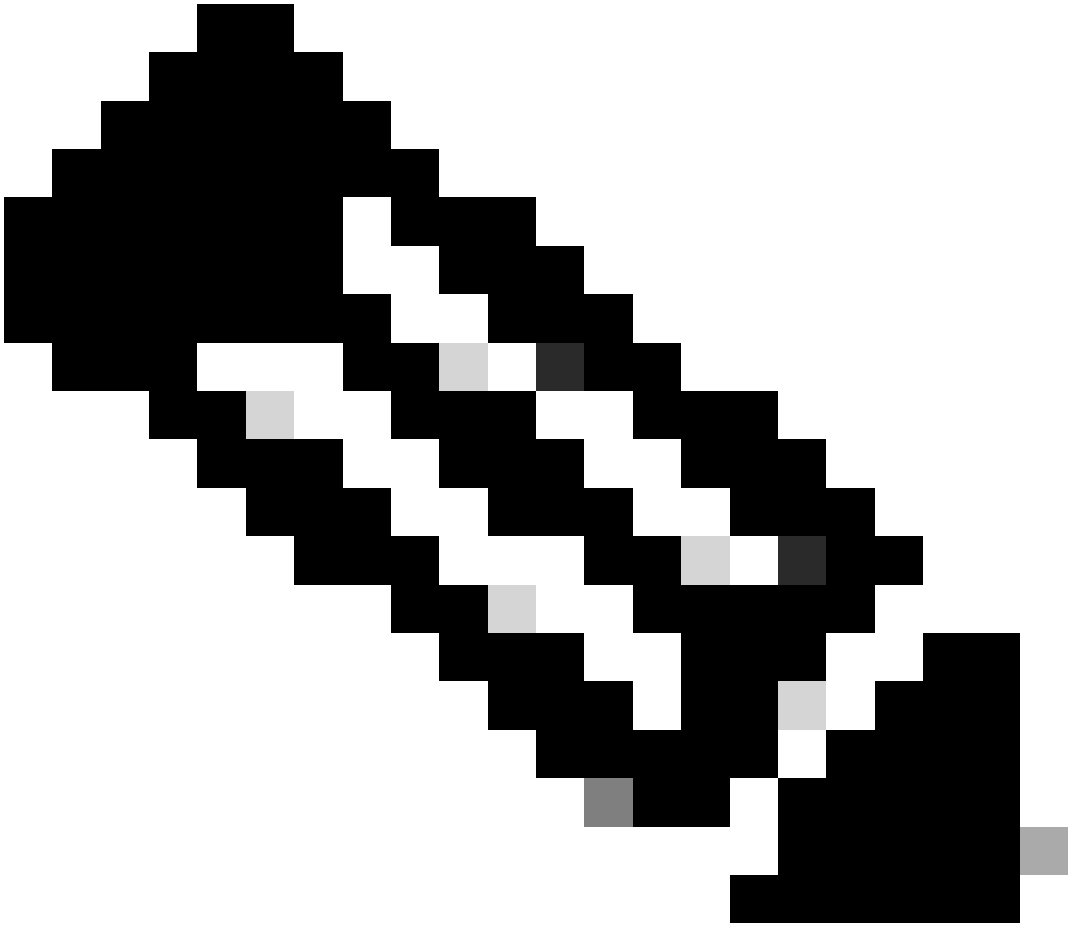
```
Conditional Debug Global State:

Start


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


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

ASR1000#
```

요약하면, 이 구성은 지금까지 적용되었습니다.

<#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
```

패킷 추적 활성화



참고: 이 섹션에서는 패킷 및 복사 옵션에 대해 자세히 설명하고, 나머지 옵션에 대해서는 이 문서의 뒷부분에서 설명합니다.

패킷 추적은 물리적 및 논리적 인터페이스(예: 터널 또는 가상 액세스 인터페이스)에서 모두 지원됩니다.

다음은 패킷 추적 CLI 구문입니다.

<#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>]
```

이 명령의 키워드에 대한 설명은 다음과 같습니다.

- pkt-num - Packet Number(패킷 번호)는 한 번에 유지되는 최대 패킷 수를 지정합니다.
- summary-only - 요약 데이터만 캡처되도록 지정합니다. 기본값은 요약 데이터와 기능 경로 데이터를 모두 캡처하는 것입니다.
- fia-trace - 경로 데이터 정보 외에 FIA 추적을 선택적으로 수행합니다.
- data-size - 경로 데이터 버퍼의 크기를 2,048~16,384바이트까지 지정할 수 있습니다. 기본값은 2,048바이트입니다.

<#root>

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

이 명령의 키워드에 대한 설명은 다음과 같습니다.

- in/out - 복사할 패킷 흐름의 방향(인그레스 및/또는 이그레스)을 지정합니다.
- L2/L3/L4 - 패킷의 복사가 시작되는 위치를 지정할 수 있습니다. 레이어 2(L2)가 기본 위치입니다.
- size - 복사되는 옥텟의 최대 수를 지정할 수 있습니다. 기본값은 64옥텟입니다.

이 예에서는 조건부 디버그 인프라에서 선택한 트래픽에 대해 패킷 추적을 활성화하는 데 사용되는 명령입니다.

<#root>

ASR1000#

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

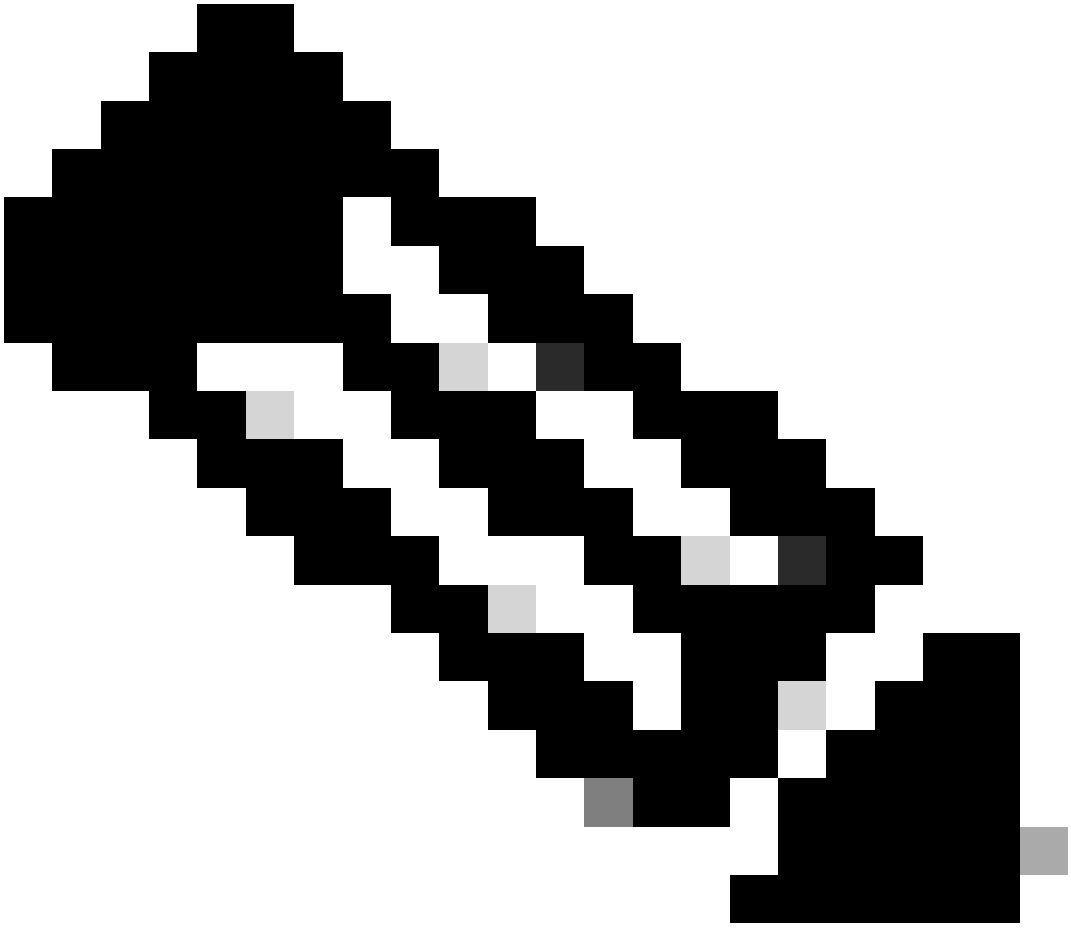
패킷 추적 컨피그레이션을 검토하려면 다음 명령을 입력합니다.

<#root>

```
ASR1000#
show platform packet-trace configuration

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

플랫폼 조건부 디버깅과 패킷 추적 컨피그레이션을 모두 보려면 show debugging 명령을 입력할 수도 있습니다.

```
<#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
```



참고: 모든 플랫폼 디버그 조건 및 패킷 추적 컨피그레이션과 데이터를 지우려면 clear platform condition all 명령을 입력합니다.

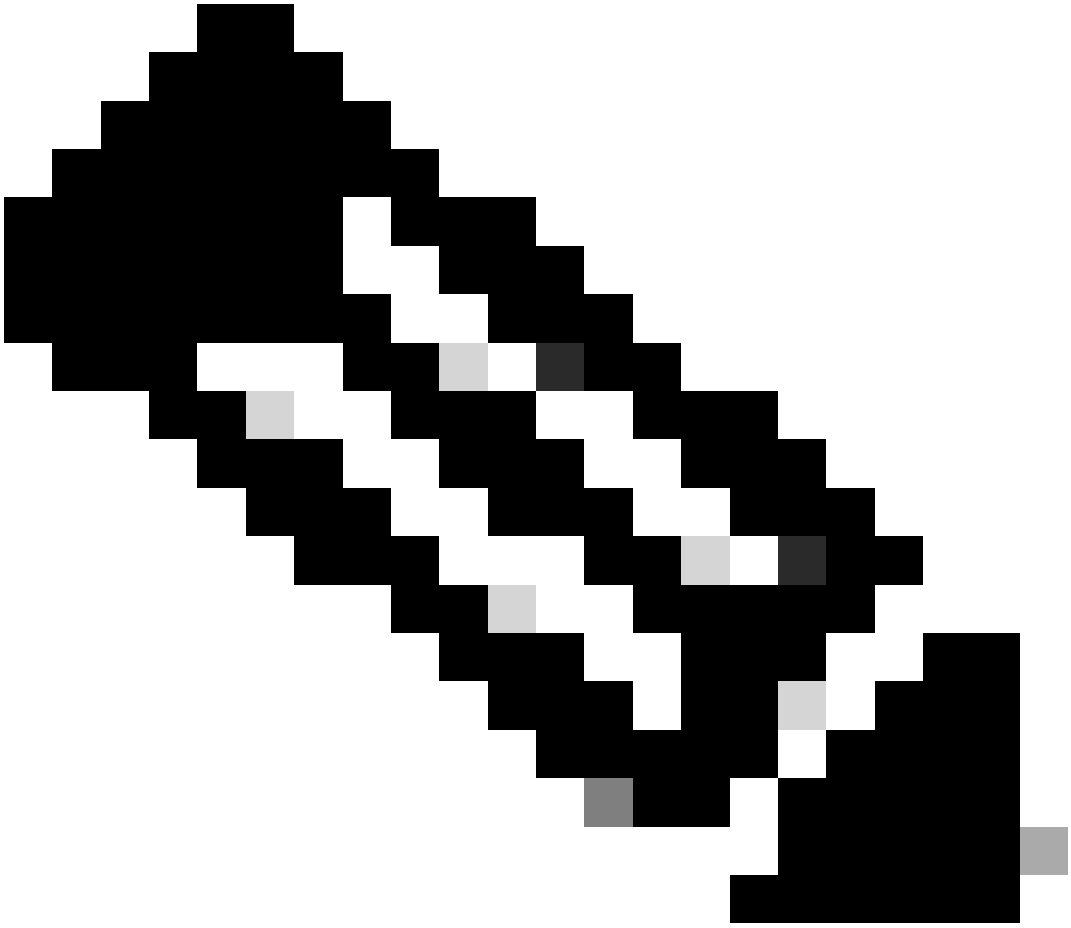
요약하면, 이 컨피그레이션 데이터는 패킷 추적을 활성화하기 위해 지금까지 사용되었습니다.

<#root>

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

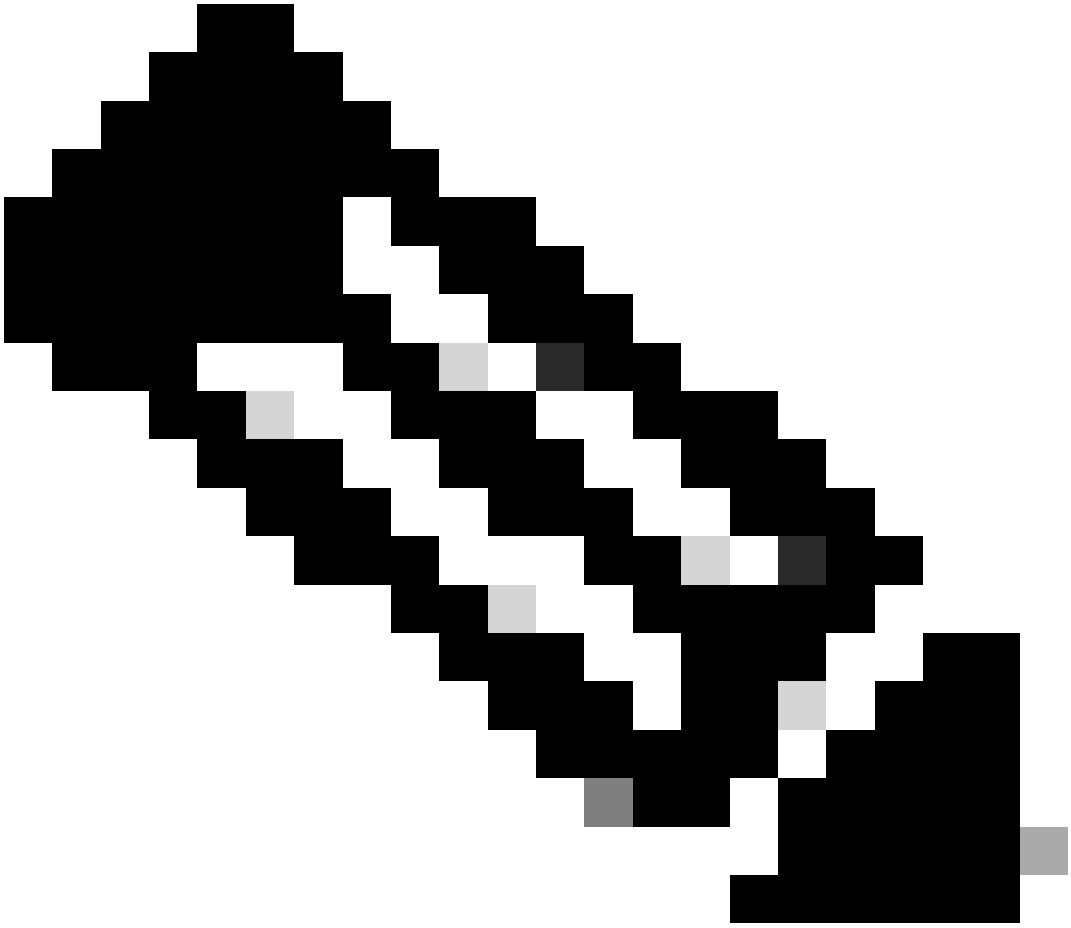
패킷 추적을 사용하는 이그레스 조건 제한

조건은 조건 필터를 정의하고 패킷에 적용되는 시기를 정의합니다. 예를 들어 디버그 플랫폼 조건 인터페이스 g0/0/0 이그레스는 패킷이 인터페이스 g0/0/0의 출력 FIA에 도달할 때 일치로 식별되므로 인그레스(ingress)에서 해당 지점까지 발생하는 모든 패킷 처리가 누락됩니다.



참고: Cisco는 가능한 한 완전하고 의미 있는 데이터를 얻기 위해 패킷 추적에 인그레스 조건을 사용하는 것을 적극 권장합니다. 이그레스(egress) 조건을 사용할 수 있지만 제한 사항에 유의해야 합니다.

패킷 추적 결과 표시



참고: 이 섹션에서는 경로 추적이 활성화된 것으로 가정합니다.

패킷 추적을 통해 다음과 같은 세 가지 특정 검사 레벨을 제공합니다.

- 어카운팅
- 패킷별 요약
- 패킷별 경로 데이터

172.16.10.2에서 172.16.20.2로 5개의 ICMP 요청 패킷이 전송되는 경우 패킷 추적 결과를 보기 위해 다음 명령을 사용할 수 있습니다.

<#root>

ASR1000#

show platform packet-trace statistics

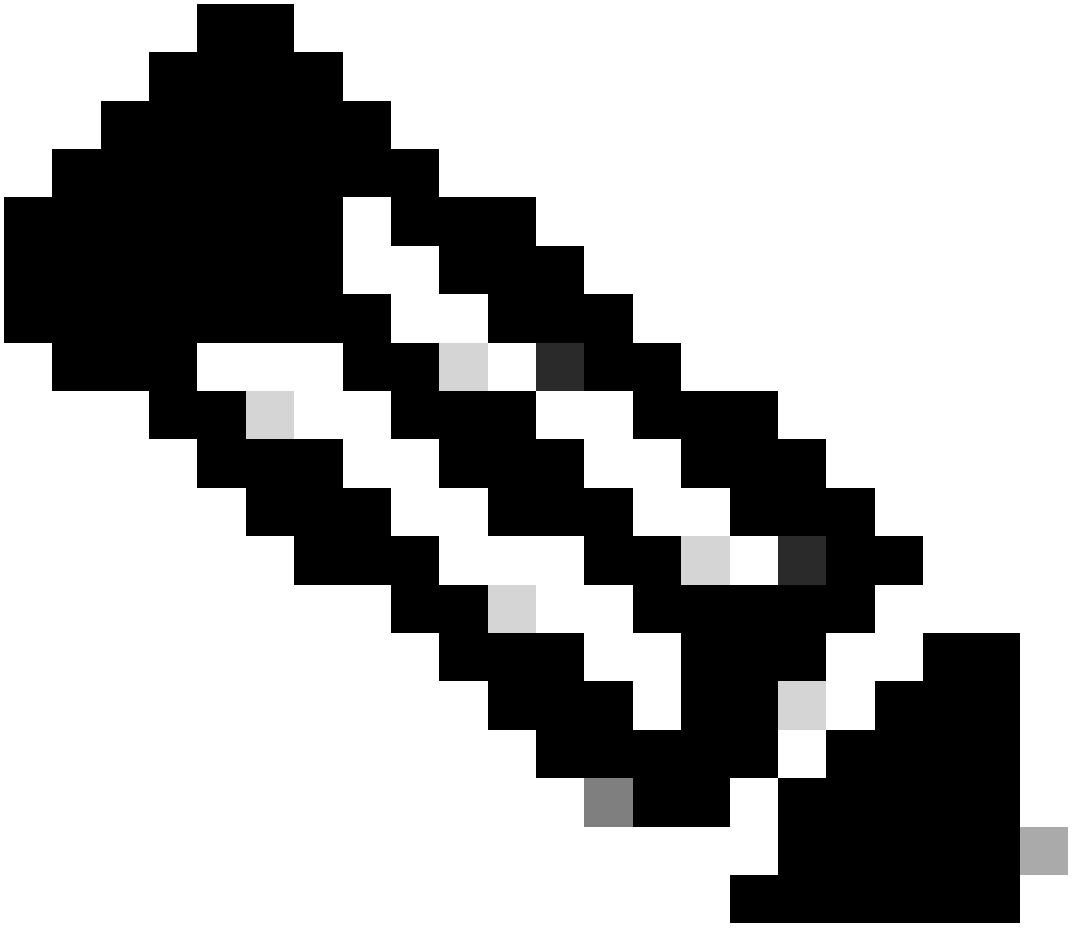
```
Packets Traced: 5


 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#
```



참고: 세 번째 명령은 각 패킷의 패킷 추적을 보는 방법을 보여 주는 예를 제공합니다. 이 예에서는 추적된 첫 번째 패킷이 표시됩니다.

이러한 출력에서 5개의 패킷이 추적되고 입력 인터페이스, 출력 인터페이스, 상태 및 경로 추적을 볼 수 있음을 알 수 있습니다.

상태	설명
FWD	패킷은 전달을 위해 예약/대기열에 추가되어 이그레스 인터페이스를 통해 다음 홉으로 전달됩니다.
펀트	패킷은 FP(Forwarding Processor)에서 RP(Route Processor)(컨트롤 플레인)로 보내집니다.
삭제	패킷이 FP에서 삭제됩니다. 삭제 이유에 대한 자세한 내용을 찾으려면 FIA 추적을 실행하거나, 전역 삭제 카운터를 사용하거나, datapath 디버그를 사용합니다.
단점	패킷은 ICMP ping 요청 또는 암호화 패킷과 같은 패킷 프로세스 중에 소비됩니다.

패킷 추적 통계 출력의 인그레스 및 삽입 카운터는 외부 인터페이스를 통해 들어오는 패킷과 컨트롤 플레인에서 삽입된 것으로 보이는 패킷에 각각 해당합니다.

FIA 추적

FIA는 패킷이 인그레스 또는 이그레스 중 하나로 전달될 때 QFP(Quantum Flow Processor)의 PPE(Packet Processor Engine)에서 순차적으로 실행되는 기능 목록을 포함합니다. 기능은 시스템에 적용된 컨피그레이션 데이터를 기반으로 합니다. 따라서 FIA 추적은 패킷이 처리될 때 시스템을 통과하는 패킷의 흐름을 이해하는 데 도움이 됩니다.

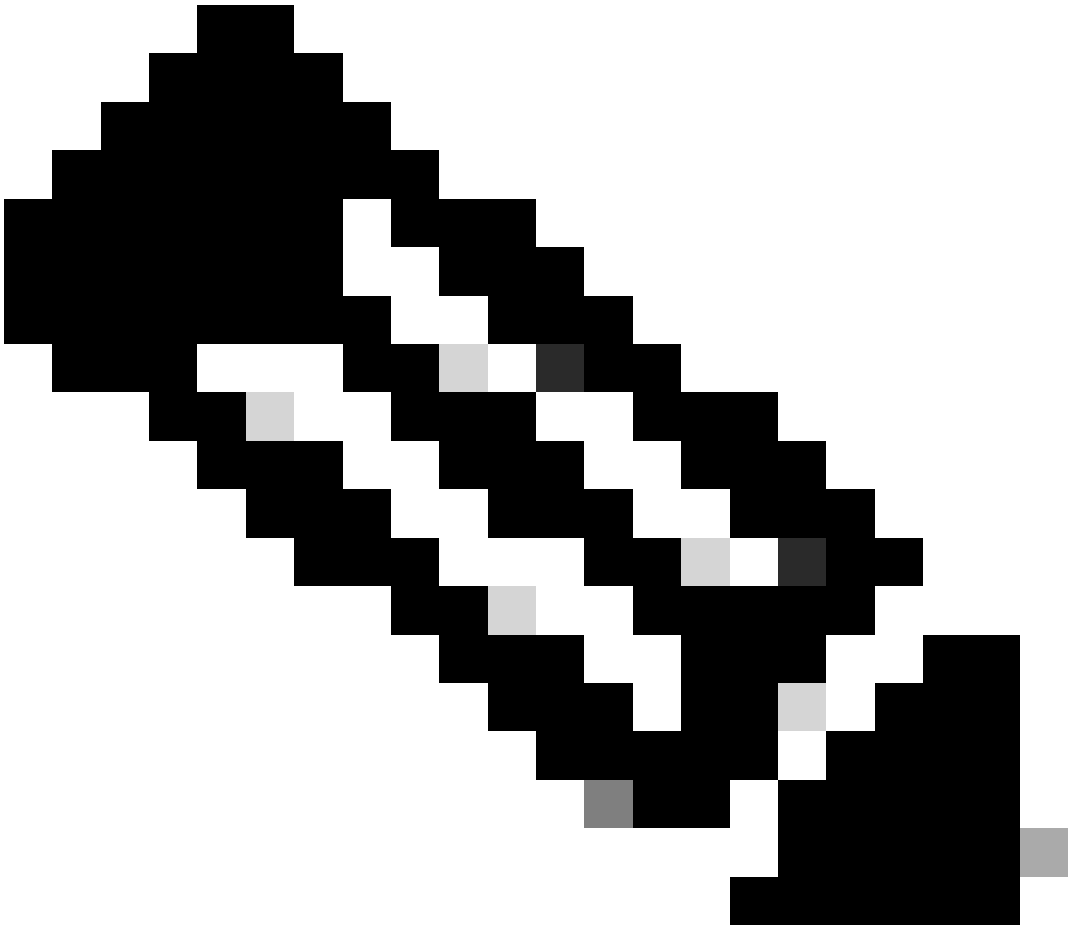
FIA를 사용하여 패킷 추적을 활성화하려면 이 컨피그레이션 데이터를 적용해야 합니다.

<#root>

ASR1000#

debug platform packet-trace packet 16 fia-trace

패킷 추적 결과 표시



참고: 이 섹션에서는 FIA 추적이 활성화되었다고 가정합니다. 또한 현재 패킷 추적 명령을 추가하거나 수정할 때 버퍼된 패킷 추적 세부 정보가 지워지므로 일부 트래픽을 다시 전송해야 추적할 수 있습니다.

이전 섹션에서 설명한 대로 FIA 추적을 활성화하는 데 사용되는 명령을 입력한 후 172.16.10.2에서 172.16.20.2로 5개의 ICMP 패킷을 보냅니다.

<#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#
```

인터페이스와 연결된 FIA 확인

플랫폼 조건부 디버그를 활성화하면 조건부 디버깅이 FIA에 기능으로 추가됩니다. 인터페이스에서 처리하는 기능 순서에 따라, 조건부 필터가 적절하게 설정되어야 합니다. 예를 들어, 조건부 필터에서 사전 NAT 주소를 사용해야 하는지 사후 NAT 주소를 사용해야 하는지 여부를 지정해야 합니다.

이 출력은 인그레스 방향으로 활성화된 플랫폼 조건부 디버깅에 대한 FIA의 기능 순서를 보여줍니다.

<#root>

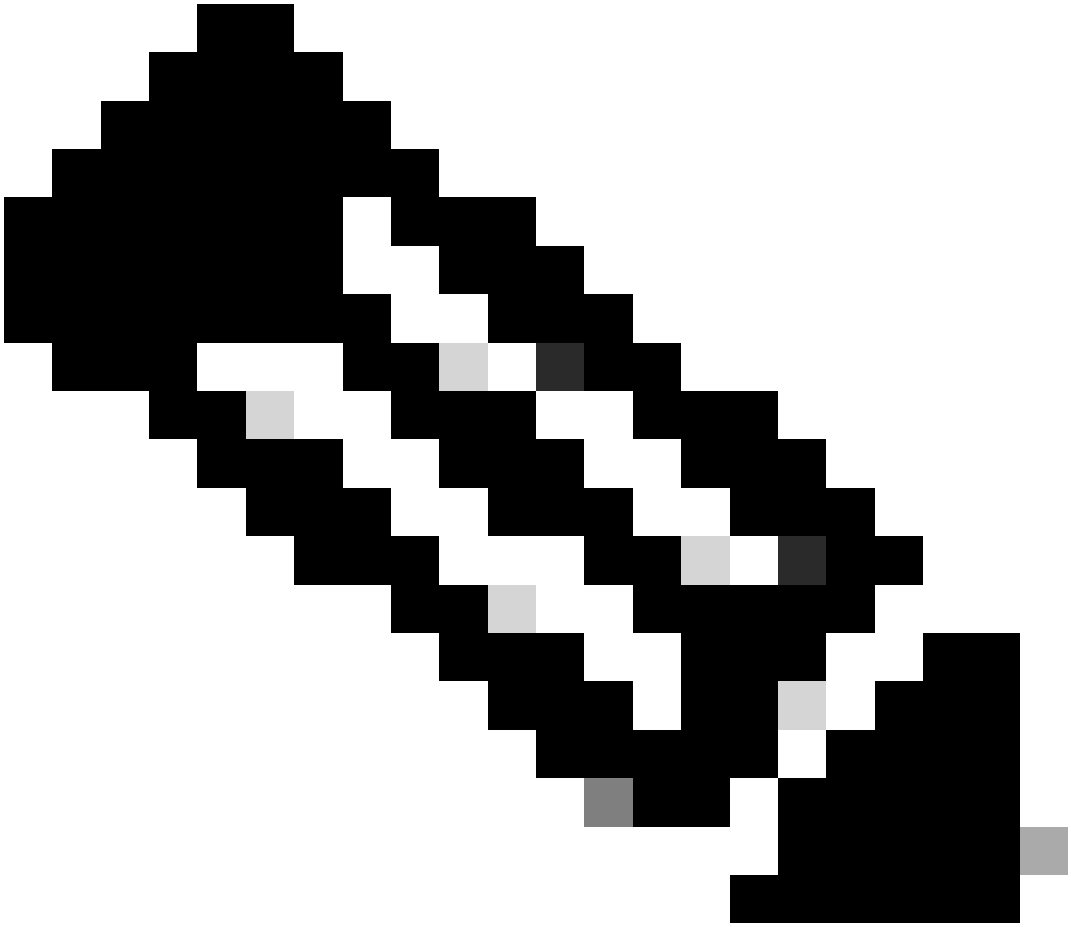
ASR1000#

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#
```



참고: CBUG_INPUT_FIA 및 DEBUG_COND_INPUT_PKT는 라우터에 구성된 조건부 디버그 기능에 해당합니다.

추적된 패킷 덤프

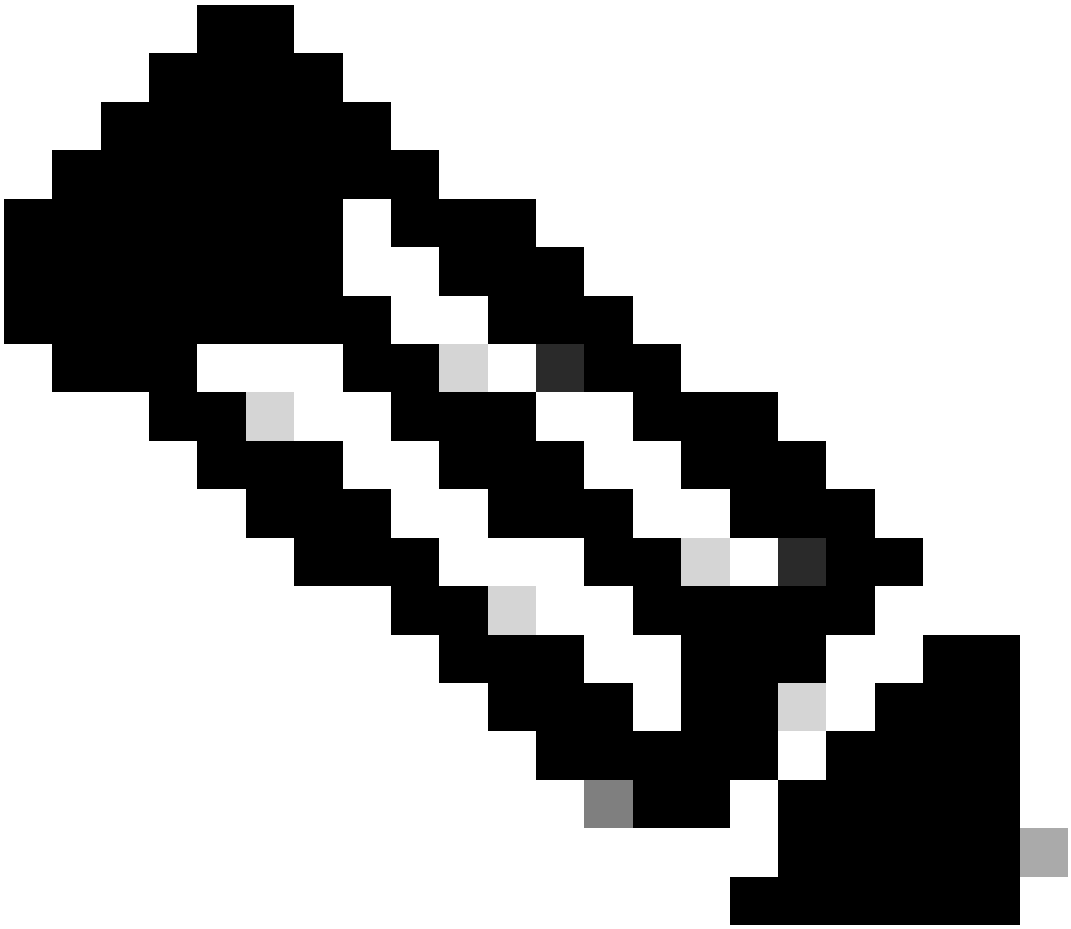
이 섹션에서 설명하는 것처럼 추적되는 대로 패킷을 복사 및 덤프할 수 있습니다. 이 예에서는 인그레스 방향(172.16.10.2~172.16.20.2)에서 최대 2,048바이트의 패킷을 복사하는 방법을 보여 줍니다.

필요한 추가 명령은 다음과 같습니다.

<#root>

ASR1000#

debug platform packet-trace copy packet input size 2048



참고: 복사된 패킷의 크기는 16~2,048바이트 범위입니다.

복사된 패킷을 덤프하려면 다음 명령을 입력합니다.

<#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#
```

추적 삭제

Drop trace는 Cisco IOS-XE Software Release 3.11 이상에서 사용할 수 있습니다. 삭제된 패킷에 대해서만 패킷 추적을 활성화합니다. 이 기능의 주요 특징은 다음과 같습니다.

- 선택적으로 특정 삭제 코드에 대한 패킷의 보존을 지정할 수 있습니다.
- 삭제 이벤트를 캡처하기 위해 전역 또는 인터페이스 조건 없이 사용할 수 있습니다.
- 삭제 이벤트 캡처는 패킷의 수명이 아니라 삭제 자체만 추적됨을 의미합니다. 그러나 조건을 개선하거나 다음 디버그 단계를 위한 단서를 제공하기 위해 요약 데이터, 튜플 데이터 및 패킷을 캡처할 수 있습니다.

다음은 드롭 유형 패킷 추적을 활성화하는 데 사용되는 명령 구문입니다.

<#root>

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

삭제 코드는 show platform hardware qfp active statistics drop detail 명령 출력에 보고된 삭제 ID와 동일합니다.

<#root>

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

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

Drop Trace 시나리오의 예

172.16.10.2에서 172.16.20.2로의 트래픽을 삭제하려면 ASR1K의 Gig 0/0/0 인터페이스에 이 ACL을 적용합니다.

```
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
```

로컬 호스트에서 원격 호스트로의 트래픽을 삭제하는 ACL이 설정된 상태에서 이 drop-trace 컨피그레이션을 적용합니다.

<#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
```

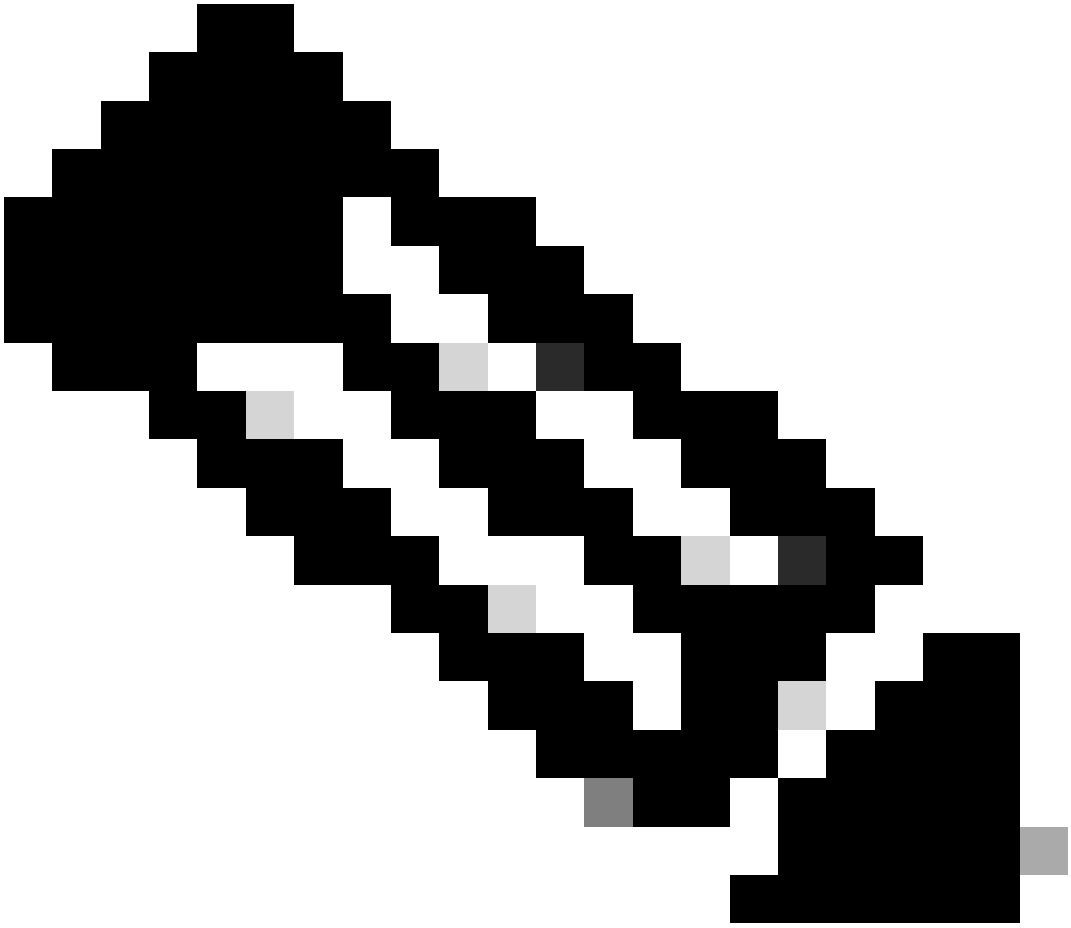
172.16.10.2에서 172.16.20.2로 5개의 ICMP 요청 패킷을 보냅니다. 삭제 추적은 다음과 같이 ACL에 의해 삭제된 이러한 패킷을 캡처합니다.

```
<#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#
```

삽입 및 펀트 추적

Inject and punt packet trace 기능은 Cisco IOS-XE Software Release 3.12 이상에서 punt(컨트롤 플레인으로 FP에서 수신하는 패킷)를 추적하고 inject(컨트롤 플레인에서 FP로 주입되는 패킷) 패킷을 추적하기 위해 추가되었습니다.



참고: 펀트 추적은 드롭 추적과 마찬가지로 전역 또는 인터페이스 조건 없이 작동할 수 있습니다. 그러나 삽입 추적이 작동하려면 조건을 정의해야 합니다.

다음은 ASR1K에서 인접 라우터로 ping할 때 punt inject packet trace 및 의 예입니다.

<#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#
```

이제 punt 및 결과를 확인할 수 nject trace r있습니다.

<#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)
```

IOSd 및 LFTS Punt/Inject Trace 및 UDF 일치를 통한 패킷 추적 개선(17.3.1의 새로운 기능)

패킷 추적 기능은 Cisco IOS-XE 릴리스 17.3.1에서 IOSd 또는 기타 BinOS 프로세스로 시작되거나 전달될 패킷에 대한 추가 추적 정보를 제공하도록 더욱 향상되었습니다.

IOSd 삭제 추적

이러한 개선을 통해 패킷 추적은 IOSd로 확장되며, 일반적으로 ***show ip traffic*** **출력**에서 보고되는 IOSd 내부의 모든 패킷 삭제에 대한 정보를 제공할 수 있습니다. IOSd 삭제 추적을 활성화하는 데 필요한 추가 컨피그레이션은 없습니다. 다음은 잘못된 체크섬 오류로 인해 IOSd에서 삭제된 UDP 패킷의 예입니다.

```
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 이그레스 경로 추적

패킷이 IOSd에서 시작되어 네트워크를 향해 이그레스 방향으로 전송됨에 따라 패킷 추적이 향상되어 경로 추적 및 프로토콜 처리 정보를 표시합니다. IOSd 이그레스 경로 추적 정보를 캡처하는 데 필요한 추가 컨피그레이션은 없습니다. 라우터를 이그레스(egress)하는 SSH 패킷에 대한 이그레스 경로 추적의 예는 다음과 같습니다.

<#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 패킷 추적

LFTS(Linux Forwarding Transport Service)는 CPP에서 전송되는 패킷을 IOSd 이외의 애플리케이션으로 전달하는 전송 메커니즘입니다. LFTS 패킷 추적 향상에서는 경로 추적 출력에 이러한 패킷에 대한 추적 정보를 추가했습니다. LFTS 추적 정보를 가져오는 데 필요한 추가 컨피그레이션은 없습니다. 다음은 NETCONF 애플리케이션에 대한 펀트된 패킷에 대한 LFTS 추적의 출력 예입니다.

<#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
```

사용자 정의 필터에 기반한 패킷 추적 패턴 일치(ASR1000 플랫폼에만 해당)

Cisco IOS-XE 릴리스 17.3.1에서는 UDF(사용자 정의 필터) 인프라를 기반으로 하는 패킷의 임의 필드에서 일치시키기 위해 새로운 패킷 일치 메커니즘이 ASR1000 제품군에 추가되었습니다. 따라서 표준 L2/L3/L4 헤더 구조의 일부가 아닌 필드를 기반으로 유연한 패킷 매칭이 가능합니다. 다음 예에서는 L3 외부 프로토콜 헤더에서 26바이트의 오프셋부터 시작하는 사용자 정의 패턴 0x4D2의 2바이트에 일치하는 UDF 정의를 보여 줍니다.

```
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
```

패킷 추적 예

이 섹션에서는 문제 해결을 위해 패킷 추적 기능이 유용한 몇 가지 예를 제공합니다.

패킷 추적 예 - NAT

이 예에서는 인터페이스 소스 NAT(Network Address Translation)가 로컬 서브넷(172.16.10.0/24)에 대한 ASR1K(Gig0/0/0)의 WAN 인터페이스에 구성됩니다.

다음은 172.16.10.2에서 172.16.20.2로의 트래픽을 추적하기 위해 사용되는 플랫폼 조건 및 패킷 추적 컨피그레이션이며, 이는 Gig0/0/0 인터페이스에서 변환(NAT)됩니다.

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

인터페이스 소스 NAT 컨피그레이션으로 172.16.10.2에서 172.16.20.2로 5개의 ICMP 패킷이 전송되는 경우 패킷 추적 결과가 다음과 같습니다.

```
      <#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#
```

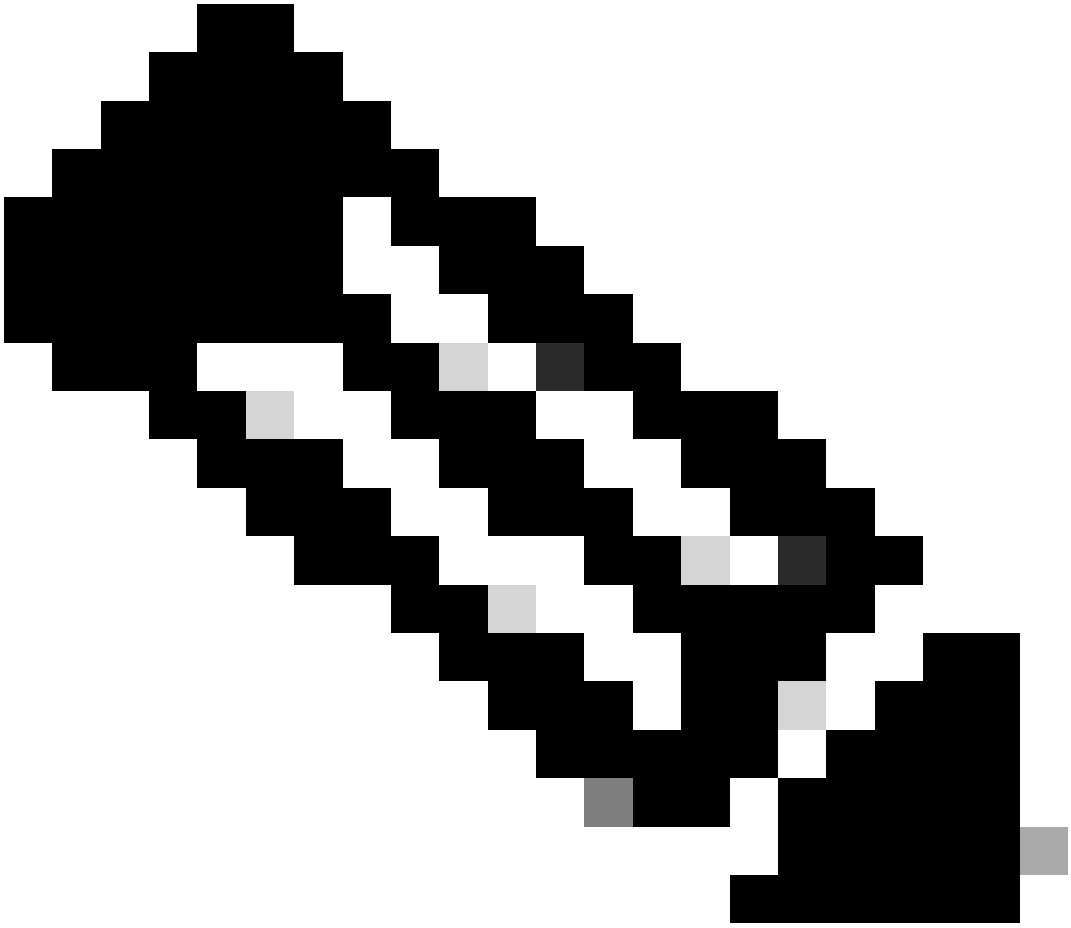
패킷 추적 예 - VPN

이 예에서는 172.16.10.0/24과 172.16.20.0/24(로컬 및 원격 서브넷) 간에 흐르는 트래픽을 보호하기 위해 ASR1K와 Cisco IOS 라우터 간에 사이트 대 사이트 VPN 터널이 사용됩니다.

다음은 Gig 0/0/1 인터페이스에서 172.16.10.2에서 172.16.20.2로 흐르는 VPN 트래픽을 추적하는 데 사용되는 플랫폼 조건 및 패킷 추적 컨피그레이션입니다.

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

이 예에서 ASR1K와 Cisco IOS 라우터 간의 VPN 터널에 의해 암호화된 172.16.10.2에서 172.16.20.2로 5개의 ICMP 패킷이 전송되는 경우 패킷 추적 출력이 다음과 같습니다.



참고: 패킷 추적은 패킷을 암호화하는 데 사용되는 추적의 QFP SA(Security Association) 핸들을 표시합니다. 이는 올바른 SA가 암호화에 사용되는지 확인하기 위해 IPsec VPN 문제를 해결할 때 유용합니다.

<#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#
```

성능에 미치는 영향

패킷 추적 버퍼는 QFP DRAM을 사용하므로 구성에 필요한 메모리의 양과 사용 가능한 메모리의 양에 주의해야 합니다.

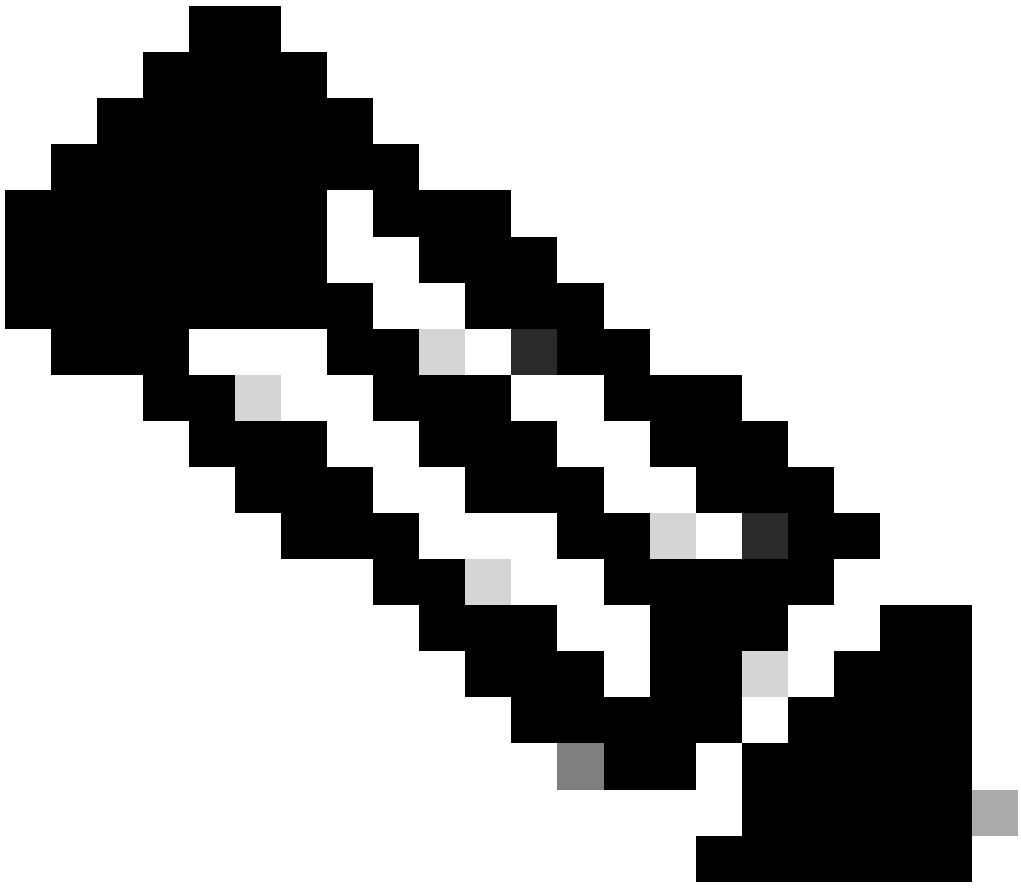
성능에 미치는 영향은 활성화된 패킷 추적 옵션에 따라 달라집니다. 패킷 추적은 추적되는 패킷의 전달 성능에만 영향을 미칩니다(예

: 사용자 구성 조건과 일치하는 패킷). 캡처할 패킷 추적을 구성하는 세부적이고 세부적인 정보가 많을수록 리소스에 더 큰 영향을 미칠 수 있습니다.

다른 트러블슈팅과 마찬가지로, 반복적인 접근 방식을 취하는 것이 가장 좋으며, 디버그 상황에서만 더 자세한 추적 옵션을 활성화하는 것이 좋습니다.

QFP DRAM 사용량은 다음 공식을 사용하여 추정할 수 있습니다.

필요한 메모리 = (통계 오버헤드) + 패킷 수 * (요약 크기 + 경로 데이터 크기 + 복사 크기)



참고: **통계 오버헤드**와 **요약 크기가** 각각 2KB 및 128B로 고정된 경우 **경로 데이터 크기**와 복사 크기**를 사용자가 구성할**

수 있습니다.

관련 정보

- **Cisco ASR1000 Series Aggregation Series Routers 소프트웨어 컨피그레이션 가이드 - 패킷 추적**
- **Cisco ASR1000 Series 서비스 라우터의 패킷 삭제**
- **Cisco 기술 지원 및 다운로드**

이 번역에 관하여

Cisco는 전 세계 사용자에게 다양한 언어로 지원 콘텐츠를 제공하기 위해 기계 번역 기술과 수작업 번역을 병행하여 이 문서를 번역했습니다. 아무리 품질이 높은 기계 번역이라도 전문 번역가의 번역 결과물만큼 정확하지는 않습니다. Cisco Systems, Inc.는 이 같은 번역에 대해 어떠한 책임도 지지 않으며 항상 원본 영문 문서(링크 제공됨)를 참조할 것을 권장합니다.