ZBFW for IOS-XE配置故障排除指南

目录

简介

本文档介绍如何使用用于轮询ASR上硬件丢弃计数器的命令对聚合服务路由器(ASR)1000上的基于区域的防火墙(ZBFW)功能进行最佳故障排除。ASR1000是基于硬件的转发平台。Cisco IOS-XE®的软件配置对硬件ASIC(量子流处理器(QFP))进行编程，以执行功能转发功能。这可实现更高的吞吐量和更好的性能。缺点是对故障排除提出了更大的挑战。传统的Cisco IOS命令通过基于区域的防火墙(ZBFW)轮询当前会话和丢弃计数器不再有效，因为丢弃不再在软件中。

链接和文档

命令参考

- Cisco ASR 1000系列聚合服务路由器命令参考
- Cisco IOS XE 3S命令参考

数据路径故障排除步骤

要排除数据路径故障，您必须确定流量是否正确通过ASR和Cisco IOS-XE代码。特定于防火墙功能，数据路径故障排除遵循以下步骤：

1. **验证配置** — 收集配置并检查输出以验证连接。
2. **验证连接状态** — 如果流量正常通过，Cisco IOS-XE会在ZBFW功能上打开连接。此连接跟踪客户端和服务器之间的流量和状态信息。
3. **验证丢弃计数器** — 当流量无法正常通过时，Cisco IOS-XE会记录任何丢弃数据包的丢弃计数器。检查此输出以查明流量故障的原因。
4. **日志记录** — 收集系统日志，以便提供有关连接构建和数据包丢弃的更精细的信息。
5. **数据包跟踪丢弃的数据包** — 使用数据包跟踪以捕获丢弃的数据包。
6. **调试** — 收集调试是最详细的选项。可以有条件地获取调试，以确认数据包的确切转发路径。

验证配置

show tech support firewall**的输出总**结如下：

```
-------- show clock --------
-------- show version -------
-------- show running-config --------
-------- show parameter-map type inspect --------
-------- show policy-map type inspect -------
-------- show class-map type inspect --------
-------- show zone security --------
-------- show zone-pair security --------
-------- show policy-firewall stats global --------
-------- show policy-firewall stats zone --------
-------- show platform hardware qfp active feature firewall datapath <submode> --------
-------- show platform software firewall RP <submode> --------
```

验证连接状态

可以获取连接信息，以便列出ZBFW上的所有连接。输入此命令：

```
ASR#show policy-firewall sessions platform
--show platform hardware qfp active feature firewall datapath scb any any any any any all any --
[s=session  i=imprecise channel  c=control channel  d=data channel]
 14.38.112.250 41392 14.36.1.206 23 proto 6 (0:0)    [sc]
```

它显示从14.38.112.250到14.36.1.206的TCP Telnet连接。

> **注意：**请注意，如果运行此命令，如果设备上有大量连接，则需要很长时间。思科建议您使用此处概述的特定过滤器运行此命令。

连接表可以向下过滤到特定源地址或目标地址。在平台子模式后**使用**过滤器。要过滤的选项包括：

```
radar-ZBFW1#show policy-firewall sessions platform ?
 all                     detailed information
 destination-port        Destination Port Number
 detail                  detail on or off
 icmp                    Protocol Type ICMP
 imprecise               imprecise information
 session                 session information
 source-port             Source Port
 source-vrf              Source Vrf ID
 standby                 standby information
 tcp                     Protocol Type TCP
 udp                     Protocol Type UDP
 v4-destination-address  IPv4 Desination Address
 v4-source-address       IPv4 Source Address
 v6-destination-address  IPv6 Desination Address
 v6-source-address       IPv6 Source Address
 |                       Output modifiers
 <cr>
```

此连接表已过滤，因此仅显示源自14.38.112.250的连接：

```
ASR#show policy-firewall sessions platform v4-source-address 14.38.112.250
--show platform hardware qfp active feature firewall datapath scb 14.38.112.250
any any any any all any --
[s=session  i=imprecise channel  c=control channel  d=data channel]
 14.38.112.250 41392 14.36.1.206 23 proto 6 (0:0)    [sc]
```

一旦过滤了连接表，就可以获得详细的连接信息以便进行更全面的分析。要显示此输出，请使用**detail**关键字。

```
ASR#show policy-firewall sessions platform v4-source-address 14.38.112.250 detail
--show platform hardware qfp active feature firewall datapath scb 14.38.112.250
any any any any all any detail--
[s=session  i=imprecise channel  c=control channel  d=data channel]
 14.38.112.250 41426 14.36.1.206 23 proto 6 (0:0)       [sc]
 pscb : 0x8c5d4f20,  bucket : 64672, fw_flags: 0x204 0x20419441,
        scb state: active, scb debug: 0
 nxt_timeout: 360000, refcnt: 1,  ha nak cnt: 0,  rg: 0, sess id: 117753
 hostdb: 0x0, L7: 0x0, stats: 0x8e118e40, child: 0x0
 l4blk0: 78fae7a7 l4blk1: e36df99c l4blk2: 78fae7ea l4blk3: 39080000
 l4blk4: e36df90e l4blk5: 78fae7ea l4blk6: e36df99c l4blk7: fde0000
 l4blk8: 0 l4blk9: 1
 root scb: 0x0 act_blk: 0x8e1115e0
 ingress/egress intf: GigabitEthernet0/0/2 (1021), GigabitEthernet0/0/0 (131065)
 current time 34004163065573 create tstamp: 33985412599209 last access: 33998256774622
 nat_out_local_addr:port: 0.0.0.0:0 nat_in_global_addr:port: 0.0.0.0:0
 syncookie fixup: 0x0
 halfopen linkage: 0x0 0x0
 cxsc_cft_fid: 0x0
 tw timer: 0x0 0x0 0x372ba 0x1e89c181
 Number of simultaneous packet per session allowed: 25
   bucket 125084 flags 1 func 1 idx 8 wheel 0x8ceb1120
```

检查防火墙丢弃计数器

在XE 3.9期间，丢弃计数器输出发生了变化。在XE 3.9之前，防火墙丢弃原因非常一般。在XE

3.9之后，防火墙丢弃原因扩展为更加精细。

要验证丢弃计数器，请执行两个步骤：

1. 确认Cisco IOS-XE中的全局丢弃计数器。这些计数器显示丢弃流量的功能。服务质量(QoS)、网络地址转换(NAT)、防火墙等功能示例。

2. 确定子功能后，查询子功能提供的精细丢弃计数器。在本指南中，要分析的子功能是防火墙功能。

QFP上的全局丢弃计数器

要依赖的基本命令提供QFP上的所有丢包：

```
Router#show platform hardware qfp active statistics drop
```

此命令显示QFP上全局丢弃的泛型。这些丢包可以位于任何功能上。一些示例功能包括：

```
Ipv4Acl
Ipv4NoRoute
Ipv6Acl
Ipv6NoRoute
NatIn2out
VfrErr
...etc
```

要查看所有丢包，包括值为零的计数器，请使用命令：

```
show platform hardware qfp active statistics drop all
```

要清除计数器，请使用此命令。在将输出显示到屏幕后，它会清除输出。此命令在读取时清除，因此在输出显示到屏幕后将其重置为零。

```
show platform hardware qfp active statistics drop clear
```

以下是QFP全局防火墙丢弃计数器列表和说明：

防火墙全局丢弃原因	解释
防火墙背压	由于日志记录机制的背压，丢包。
FirewallInvalidZone	没有为接口配置安全区域。
防火墙L4Insp	L4策略检查失败。有关更精细的丢弃原因（防火墙功能丢弃原因），请参阅下
FirewallNoForwardingZone	防火墙未初始化，不允许任何流量通过。
防火墙非会话	会话创建失败。可能是因为已达到最大会话限制或内存分配失败。
防火墙策略	已配置的防火墙策略已丢弃。
防火墙L4	L4检测失败。有关更精细的丢弃原因（防火墙功能丢弃原因），请参阅下表。
防火墙L7	由于L7检测而丢弃数据包。有关更精细的L7丢弃原因（防火墙功能丢弃原因）
FirewallNotInitiator	不是TCP、UDP或ICMP的会话发起方。未创建会话。例如，对于ICMP，收到这可能发生在正常数据包处理或不精确信道处理中。
FirewallNoNewSession	防火墙高可用性不允许新会话。
FirewallSyncookieMaxDst	为了提供基于主机的SYN泛洪保护，每个目标的SYN速率为SYN泛洪限制。当
FirewallSyncookie	触发SYNCOLIE逻辑。这表示已发送带SYN Cookie的SYN/ACK，并丢弃原始S
防火墙ARStandby	非对称路由未启用，冗余组未处于活动状态。

QFP上的防火墙功能丢弃计数器

QFP全局丢弃计数器的限制是丢弃原因没有粒度，而且某些丢弃原因(如**FirewallL4**)会过载到几乎不用进行故障排除的程度。此功能在Cisco IOS-XE 3.9(15.3(2)S)中得到增强，在该版本中添加了防火墙功能丢弃计数器。这提供了一组更精细的丢弃原因：

```
ASR#show platform hardware qfp active feature firewall drop all
-----------------------------------------------------------------------------
Drop Reason Packets
-----------------------------------------------------------------------------
Invalid L4 header 0
Invalid ACK flag 0
Invalid ACK number 0
....
```

以下是防火墙功能丢弃原因和解释的列表：

防火墙功能丢弃原因	解释
标头长度无效	数据报太小，无法包含第4层TCP、UDP或ICMP报头。其原因可能是： 1. TCP报头长度< 20 2. UDP/ICMP报头长度< 8
UDP数据长度无效	UDP数据报长度与UDP报头中指定的长度不匹配。
ACK号无效	此下降可能是由以下原因之一造成的： 1. ACK不等于TCP对等体的next_seq#。 2. ACK大于TCP对等体发送的最新SEQ#。
ACK标志无效	在TCP SYNSENT和SYNRCVD状态下，ACK#应等于ISN+1，但不是。 此下降可能是由以下原因之一造成的： 1. 应为ACK标志，但未在不同的TCP状态中设置。 2. 除ACK标志外，还设置其他标志（如RST）。
TCP发起程序无效	当： 1. 来自TCP发起方的第一个数据包不是SYN（未收到有效会话的非初 2. 初始SYN数据包设置了ACK标志。
SYN与数据	SYN数据包包含负载。不支持此功能。
TCP标志无效	无效的TCP标志可能由以下因素引起： 1. TCP初始SYN数据包具有除SYN以外的标记。 2. 在TCP侦听状态下，TCP对等体接收RST或ACK。 3. 在SYN/ACK之前收到其他响应方的数据包。 4. 未从响应方接收预期的SYN/ACK。
处于SYNSENT状态的段无效	SYNSENT状态中的无效TCP数据段是由以下原因造成的： 1. SYN/ACK具有负载。 2. SYN/ACK设置了其他标志(PSH、URG、FIN)。 3. 接收带负载的传输SYN。 4. 从启动器接收非SYN数据包。
处于SYNRCVD状态的段无效	SYNRCVD状态中的无效TCP数据段可能由以下因素引起： 1. 从发起方接收带负载的重传SYN。 2. 从响应方接收非SYN/ACK、RST或FIN的无效数据段。
SEQ无效	当数据段来自发起方时，这会在SYNRCVD状态中发生。原因： 1. Seq#小于ISN。 2. 如果接收器rcvd窗口大小为0且： 网段有负载，或 无序段(seq#大于接收器LASTACK。

	3. 如果接收方rcvd窗口大小为0且seq#超出窗口。 4. Seq#等于ISN，但不等于SYN数据包。
窗口缩放选项无效	无效的TCP窗口缩放选项是由不正确的窗口缩放选项字节长度引起的。
TCP窗口外	数据包太旧 — 一个窗口位于另一端的ACK后面。这可能发生在ESTABL
发送FIN后的TCP额外负载	FIN发送后收到负载。这可能发生在CLOSEWAIT状态。
TCP窗口溢出	当传入数据段大小溢出接收方的窗口时，会发生这种情况。但是，如果启
具有无效标志的重新转换	接收方已确认重发的数据包。
TCP无序数据段	无序数据包即将传送到L7进行检查。如果L7不允许OOO网段，此数据包
SYN 泛洪	在TCP SYN泛洪攻击下。在某些情况下，当当前与此主机的连接超过配置 被丢弃。
内部错误 — 同步泛洪检查分配失败	在synflood检查期间，主机db的分配失败。 建议操作：选中“show platform hardware qfp active feature firewall mem
Synflood封锁丢弃	如果超过配置的半开连接并配置了封锁时间，则会丢弃所有到此IP地址的
半开放会话限制超出	由于超出允许的半打开会话数，数据包被丢弃。 另请检查“max-incomplete high/low”和“one-minute high/low”的设置，以确
每个流的Pkt太多	超过每个流允许的可检查数据包的最大数量。最大数为25。
每个流的ICMP错误数据包过多	超过每个流允许的ICMP错误数据包的最大数量。最大数为3。
不期望TCP负载从Rsp到Init	在SYNRCVD状态下，TCP从响应方向到发起方方向接收带负载的数据包
内部错误 — 未定义方向	未定义数据包方向。
当前窗口内的SYN	在已建立的TCP连接窗口中可以看到SYN数据包。
当前窗口内的RST	在已建立的TCP连接窗口内观察到RST数据包。
杂散网段	接收不应通过TCP状态机接收的TCP数据段，例如在侦听状态下从响应方
ICMP内部错误 — 丢失的ICMP NAT信息	ICMP数据包已经过NAT，但缺少内部NAT信息。这是内部错误。
处于SCB关闭状态的ICMP数据包	收到处于SCB CLOSE状态的ICMP数据包。
ICMP数据包中缺少IP报头	ICMP数据包中缺少IP报头。
ICMP错误无IP或ICMP	负载中没有IP或ICMP的ICMP错误数据包。可能是由格式错误的数据包或
ICMP错误PKT太短	ICMP错误数据包太短。
ICMP错误超出突发限制	ICMP错误数据包超过突发限制10。
ICMP错误不可达	ICMP错误数据包无法到达超出限制。仅允许第1个不可达数据包通过。
ICMP错误无效Seq#	嵌入式数据包的Seq#与发生ICMP错误的数据包的seq#不匹配。
ICMP错误无效确认	嵌入ICMP错误数据包中的ACK无效。
ICMP操作丢弃	配置的ICMP操作为丢弃。
没有策略映射的区域对	区域对上不存在策略。可能是因为未将ALG（应用层网关）配置为打开应
会话丢失且策略不存在	会话查找失败，且不存在用于检查此数据包的策略。
ICMP错误和策略不存在	ICMP错误，在区域对上未配置策略。
分类失败	当防火墙尝试确定协议是否可检查时，给定区域对中的分类失败。
分类操作丢弃	分类操作为丢弃。
安全策略配置错误	由于安全策略配置错误，分类失败。这也可能是因为L7数据通道没有引脚
将RST发送到响应方	当ACK#不等于ISN+1时，将RST发送到SYNSENT状态的响应方。
防火墙策略丢弃	策略操作即将丢弃。
分片丢弃	丢弃第一个分段时丢弃剩余的分段。
ICMP防火墙策略丢弃	ICMP嵌入数据包的策略操作为DROP。
L7检测返回DROP	L7(ALG)决定丢弃数据包。原因可从不同的ALG统计中找到。
L7网段PKT不允许	当ALG不执行时，收到分段数据包。
L7分段PKT不允许	当ALG不执行时，收到分段（或VFR）数据包。
未知的L7协议类型	无法识别的协议类型。

排除防火墙丢弃故障

从上述全局或防火墙功能丢弃计数器中确定丢弃原因后，如果这些丢弃是意外的，则可能需要其他故障排除步骤。除了为确保启用的防火墙功能的配置正确进行配置验证外，通常还需要对相关流量执行数据包捕获，以查看数据包是否格式错误或是否存在任何协议或应用程序实施问题。

日志记录

ASR日志记录功能生成系统日志以记录丢弃的数据包。这些系统日志提供了有关丢弃数据包的原因的更多详细信息。系统日志绑定分为两种类型：

1. 本地缓冲系统日志
2. 远程高速日志记录

本地缓冲系统日志

为了查明丢包的原因，您可以使用通用ZBFW故障排除，例如启用日志丢包。配置数据包丢弃日志记录有两种方法。

方法 1：使用inspect-global parameter-map记录所有丢弃的数据包。

```
parameter-map type inspect-global     log dropped-packets
```

方法 2：使用自定义检查参数映射只记录特定类的丢弃的数据包。

```
parameter-map type inspect LOG_PARAM
log dropped-packets
!
policy-map type inspect ZBFW_PMAP
class type inspect ZBFW_CMAP
inspect LOG_PARAM
```

这些消息会根据ASR的日志记录配置发送到日志或控制台。以下是丢弃日志消息的示例。

```
*Apr  8 13:20:39.075: %IOSXE-6-PLATFORM: F0: cpp_cp: QFP:0.0 Thread:103
TS:00000605668054540031 %FW-6-DROP_PKT: Dropping tcp pkt from GigabitEthernet0/0/2
14.38.112.250:41433 => 14.36.1.206:23(target:class)-(INSIDE_OUTSIDE_ZP:class-default)
due to Policy drop:classify result with ip ident 11579 tcp flag 0x2, seq 2014580963,
ack 0
```

本地缓冲系统日志记录的限制

1. 根据Cisco Bug ID CSCud09943，这些日志的速率受限。
2. 除非应用特定配置，否则这些日志可能无法打印。例如，除非指定log关键字，否则不会记录由class-default数据包丢弃的数据包：

```
policy-map type inspect ZBFW_PMAP
```

```
class class-default
drop log
```

远程高速日志记录

高速日志记录(HSL)直接从QFP生成系统日志，并将其发送到已配置的netflow HSL收集器。这是ASR上ZBFW的推荐日志记录解决方案。

对于HSL，请使用以下配置：

```
parameter-map type inspect inspect-global
  log template timeout-rate 1
  log flow-export v9 udp destination 1.1.1.1 5555
```

要使用此配置，需要支持Netflow版本9的NetFlow收集器。详情请参阅

[配置指南：基于区域的策略防火墙，Cisco IOS XE版本3S(ASR 1000)防火墙高速日志记录]

使用条件匹配的数据包跟踪

启用条件调试以启用数据包跟踪，然后启用以下功能的数据包跟踪：

```
ip access-list extended CONDITIONAL_ACL
 permit ip host 10.1.1.1 host 192.168.1.1
 permit ip host 192.168.1.1 host 10.1.1.1
!
debug platform condition feature fw dataplane submode all level info
debug platform condition ipv4 access-list CONDITIONAL_ACL both
```

> **注意：**匹配条件可以直接使用IP地址，因为不需要ACL。这将匹配为允许双向跟踪的源或目标。如果不允许更改配置，则可以使用此方法。例如：debug platform condition ipv4 address 192.168.1.1/32。

打开数据包跟踪功能：

```
debug platform packet-trace copy packet both
debug platform packet-trace packet 16
debug platform packet-trace drop
debug platform packet-trace enable
```

使用此功能有两种方法：

1. 输入**debug platform packet-trace drop**命令以仅跟踪丢弃的数据包。
2. 如果排除命令**debug platform packet-trace drop，将**跟踪任何与条件匹配的数据包，包括设备检查/传递的数据包。

打开条件调试：

```
debug platform condition start
```

运行测试，然后关闭调试：

```
debug platform condition stop
```

现在，信息可以显示到屏幕。在本例中，由于防火墙策略，ICMP数据包被丢弃：

```
Router#show platform packet-trace statistics
Packets Summary
 Matched  2
 Traced   2
Packets Received
 Ingress  2
 Inject   0
Packets Processed
 Forward  0
 Punt     0
 Drop     2
   Count       Code  Cause
   2           183   FirewallPolicy
 Consume  0
```

```
Router#show platform packet-trace summary
Pkt   Input            Output           State  Reason
0     Gi0/0/2          Gi0/0/0          DROP   183 (FirewallPolicy)
1     Gi0/0/2          Gi0/0/0          DROP   183 (FirewallPolicy)
```

```
Router#show platform packet-trace packet 0
Packet: 0           CBUG ID: 2980
Summary
 Input     : GigabitEthernet0/0/2
 Output    : GigabitEthernet0/0/0
 State     : DROP 183 (FirewallPolicy)
 Timestamp
   Start   : 1207843476722162 ns (04/15/2014 12:37:01.103864 UTC)
   Stop    : 1207843477247782 ns (04/15/2014 12:37:01.104390 UTC)
Path Trace
 Feature: IPV4
   Source      : 10.1.1.1
   Destination : 192.168.1.1
   Protocol    : 1 (ICMP)
 Feature: ZBFW
   Action  : Drop
   Reason  : ICMP policy drop:classify result
   Zone-pair name  : INSIDE_OUTSIDE_ZP
   Class-map name  : class-default
Packet Copy In
 c89c1d51 5702000c 29f9d528 08004500 00540000 40004001 ac640e26 70fa0e24
 01010800 172a2741 00016459 4d5310e4 0c000809 0a0b0c0d 0e0f1011 12131415
Packet Copy Out
 c89c1d51 5702000c 29f9d528 08004500 00540000 40003f01 ad640e26 70fa0e24
 01010800 172a2741 00016459 4d5310e4 0c000809 0a0b0c0d 0e0f1011 12131415
```

show platform packet-trace packet <num> decode命令可解码数据包报头信息和内容。XE3.11中引入了以下功能：

```
Router#show platform packet-trace packet all decode
Packet: 0           CBUG ID: 2980
Summary
```

```
 Input     : GigabitEthernet0/0/2
 Output    : GigabitEthernet0/0/0
 State     : DROP 183 (FirewallPolicy)
 Timestamp
   Start   : 1207843476722162 ns (04/15/2014 12:37:01.103864 UTC)
   Stop    : 1207843477247782 ns (04/15/2014 12:37:01.104390 UTC)
Path Trace
 Feature: IPV4
    Source      : 10.1.1.1
    Destination : 192.168.1.1
    Protocol    : 1 (ICMP)
  Feature: ZBFW
    Action  : Drop
    Reason  : ICMP policy drop:classify result
    Zone-pair name  : INSIDE_OUTSIDE_ZP
    Class-map name  : class-default
Packet Copy In
 c89c1d51 5702000c 29f9d528 08004500 00540000 40004001 ac640e26 70fa0e24
 01010800 172a2741 00016459 4d5310e4 0c000809 0a0b0c0d 0e0f1011 12131415
 ARPA
    Destination MAC     : c89c.1d51.5702
    Source MAC          : 000c.29f9.d528
   Type               : 0x0800 (IPV4)
 IPv4
   Version            : 4
   Header Length      : 5
   ToS                : 0x00
   Total Length       : 84
   Identifier         : 0x0000
   IP Flags           : 0x2 (Don't fragment)
   Frag Offset        : 0
   TTL                : 64
   Protocol           : 1 (ICMP)
   Header Checksum    : 0xac64
   Source Address     : 10.1.1.1
   Destination Address : 192.168.1.1
 ICMP
   Type               : 8 (Echo)
   Code               : 0 (No Code)
   Checksum           : 0x172a
   Identifier         : 0x2741
   Sequence           : 0x0001
Packet Copy Out
 c89c1d51 5702000c 29f9d528 08004500 00540000 40003f01 ad640e26 70fa0e24
 01010800 172a2741 00016459 4d5310e4 0c000809 0a0b0c0d 0e0f1011 12131415
 ARPA
   Destination MAC     : c89c.1d51.5702
   Source MAC          : 000c.29f9.d528
   Type               : 0x0800 (IPV4)
 IPv4
   Version            : 4
   Header Length      : 5
   ToS                : 0x00
   Total Length       : 84
   Identifier         : 0x0000
   IP Flags           : 0x2 (Don't fragment)
   Frag Offset        : 0
   TTL                : 63
   Protocol           : 1 (ICMP)
   Header Checksum    : 0xad64
   Source Address     : 10.1.1.1
   Destination Address : 192.168.1.1
 ICMP
   Type               : 8 (Echo)
```

```
Code                : 0 (No Code)
Checksum            : 0x172a
Identifier          : 0x2741
Sequence            : 0x0001
```

嵌入式数据包捕获

Cisco IOS-XE 3.7(15.2(4)S)中增加了嵌入式数据包捕获支持。 有关详细信息，请参阅

Cisco IOS和IOS-XE的嵌入式数据包捕获配置示例。

调试

条件调试

在XE3.10中，将引入条件调试。可以使用条件语句来确保ZBFW功能仅记录与条件相关的调试消息。条件调试使用ACL来限制与ACL元素匹配的日志。此外，在XE3.10之前，调试消息更难读取。调试输出在XE3.10中得到了改进，使其更易于理解。

要启用这些调试，请发出以下命令：

```
debug platform condition feature fw dataplane submode [detail | policy | layer4 | drop]
debug platform condition ipv4 access-list <ACL_name> both
debug platform condition start
```

请注意，condition命令必须通过ACL和方向设置。条件调试在使用debug platform condition start命令启动后才**能实现**。要关闭条件调试，请使用命令debug platform condition **stop**。

```
debug platform condition stop
```

要关闭条件调试，请**勿使**用命令**undebug all**。要关闭所有条件调试，请使用命令：

```
ASR#clear platform condition all
```

在XE3.14之前， **ha和事件**调试不是条件的。因此，命令debug platform condition feature fw dataplane submode **all会导致创建所有日志，而与下面选择的条件无关。**这可能会产生额外的噪音，使调试变得困难。

默认情况下，条件日志记录级别为**info**。要增加/降低日志记录级别，请使用命令：

```
debug platform condition feature fw dataplane submode all [verbose | warning]
```

收集和查看调试

调试文件不会打印到控制台或显示器。所有调试都写入到ASR的硬盘。调试将写入到文件夹tracelogs下**的硬**盘中，**名称为cpp_cp_F0-0.log.<date>**。要查看写入调试的文件，请使用以下输出：

```
ASR# cd harddisk:
ASR# cd tracelogs
ASR# dir cpp_cp_F0*Directory of harddisk:/tracelogs/cpp_cp_F0*

Directory of harddisk:/tracelogs/

3751962 -rwx 1048795 Jun 15 2010 06:31:51 +00:00
cpp_cp_F0-0.log.5375.20100615063151
3751967 -rwx 1048887 Jun 15 2010 02:18:07 +00:00
cpp_cp_F0-0.log.5375.20100615021807
39313059840 bytes total (30680653824 bytes free)
```

每个调试文件将存储为cpp_cp_F0-0.log.<date>文件。这些是可通过TFTP从ASR复制的常规文本文件。ASR上的日志文件最大值为1Mb。在1Mb后，调试将写入新的日志文件。因此，每个日志文件都加上时间戳以指示文件的开始。

日志文件可能存在于以下位置：

```
harddisk:/tracelogs/
bootflash:/tracelogs/
```

由于日志文件仅在旋转后显示，因此可以使用以下命令手动旋转日志文件：

```
ASR# test platform software trace slot f0 cpp-control-process rotate
```

这会立即创建“cpp_cp”日志文件，并在QFP上启动新文件。例如：

```
ASR#test platform software trace slot f0 cpp-control-process rotate
 Rotated file from: /tmp/fp/trace/stage/cpp_cp_F0-0.log.7311.20140408134406,
Bytes: 82407, Messages: 431

ASR#more tracelogs/cpp_cp_F0-0.log.7311.20140408134406
04/02 10:22:54.462 : btrace continued for process ID 7311 with 159 modules
04/07 16:52:41.164 [cpp-dp-fw]: (info): QFP:0.0 Thread:110 TS:00000531990811543397
:FW_DEBUG_FLG_HA:[]: HA[1]: Changing HA state to 9
04/07 16:55:23.503 [cpp-dp-fw]: (info): QFP:0.0 Thread:120 TS:00000532153153672298
:FW_DEBUG_FLG_HA:[]: HA[1]: Changing HA state to 10
04/07 16:55:23.617 [buginf]: (debug): [system] Svr HA bulk sync CPP(0) complex(0)
epoch(0) trans_id(26214421) rg_num(1)
```

此命令允许将调试文件合并到单个文件中，以便更轻松地处理。它会合并目录中的所有文件，并根据时间进行交互。当日志非常冗余且跨多个文件创建时，这会有所帮助：

```
ASR#request platform software trace slot rp active merge target bootflash:MERGED_OUTPUT.log
Creating the merged trace file: [bootflash:MERGED_OUTPUT.log]
 including all messages

Done with creation of the merged trace file: [bootflash:MERGED_OUTPUT.log]
```