IOS-XE配置的ZBFW排除故障指南

目录

简介

本文描述如何最佳排除故障在聚合服务路由器(ASR) 1000的区域基于防火墙(ZBFW)功能，用使用轮询在ASR的硬件丢弃计数器的命令。ASR1000是一个基于硬件的转发平台。Cisco IOS XE ®软件配置编程硬件ASIC (数量流处理器(QFP)为了执行功能转发功能。这允许提高吞吐量和更加好的性能。对此的缺点是提出一更加巨大的挑战排除故障。用于的传统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，虽然丢包不再在软件方面。

链路和文档

命令参考

- Cisco ASR 1000系列汇聚服务路由器命令参考
- Cisco IOS XE 3S命令参考

数据路径排除故障步骤

为了排除故障数据路径，您必须识别流量是否通过ASR和Cisco IOS XE代码适当地通过。对防火墙功能的特定，数据路径故障排除遵从这些步骤：

1. **验证配置**-采集配置并且检查输出为了验证连接。
2. **验证州的连接，**如果流量适当地通过， Cisco IOS XE打开在ZBFW功能的一连接。此连接跟踪流量和状态信息在客户端和服务器之间。
3. **验证丢弃计数器**-当流量不适当地时通过， Cisco IOS XE记录所有丢弃的数据包的一个丢弃计数器。检查此输出为了查出流量失败的原因。
4. **记录**-聚集Syslog为了提供更加粒状的信息在连接修造和丢包。
5. **数据包踪迹丢弃的数据包**-请使用包跟踪为了捉住丢弃的数据包。
6. **调试**-聚集调试是多数verbose选项。调试可以有条件地得到为了确认数据包的确切的转发路径。

验证配置

show tech support防火墙输出汇总此处：

```
-------- 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

连接信息可以得到，以便在ZBFW的所有连接是列出的。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]
```

它表示从14.38.112.250的TCP Telnet连接到14.36.1.206。

> **注意**：注意，如果运行此命令，将需要很长时间是否有在设备的大量连接。思科建议您运行此命令以特定过滤器如概述此处。

连接表可以被过滤下来到一特定源或目的地址。在**平台**从属方式以后请使用过滤器。选项过滤是：

```
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>
```

此连接表是从14.38.112.250来源的仅被过滤的那么连接显示：

```
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]
```

一旦连接表被过滤，详细的连接信息可以为一更加全面的anlaysis得到。为了显示此输出，请使用**详细信息**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输出更改的丢弃计数器3.9期间。在XE 3.9前，防火墙丢弃原因非常通用的。在XE 3.9以后，防火墙丢弃原因被扩展了变得更加粒状。

为了验证丢弃计数器，请执行两个步骤：

1. 确认在Cisco IOS XE的全局丢弃计数器。这些计数器显示什么功能丢弃了流量。功能示例包括服务质量(QoS)，网络地址转换(NAT)，防火墙，等等。

2. 一旦子功能识别，请查询子功能提供的粒状丢弃计数器。在此指南中，被分析的子功能是防火墙功能。

在QFP的全局丢弃计数器

基本命令取决于在提供在QFP间的所有丢包：

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

此命令显示您通用的丢包全局在QFP间。这些丢包可以在所有功能。某些示例功能是：

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

为了看到所有丢包，请包括有值为零的计数器，使用命令：

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

为了清除计数器，请使用此命令。它在显示它以后清除输出对屏幕。此命令是清楚的在读，因此输出被重设到零，**在**显示对屏幕后。

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

下面QFP全局防火墙丢弃计数器和说明列表：

防火墙全局丢弃原因	说明
FirewallBackpressure	丢包由于反压力通过记录机制。
FirewallInvalidZone	为接口配置的没有安全区。
FirewallL4Insp	L4策略检查失败。参见下面表关于更加粒状的丢弃原因(防火墙功能丢弃原因)。
FirewallNoForwardingZone	防火墙未初始化，并且流量没有允许通过。
FirewallNonsession	会话创建发生故障。它可能归结于最大会话限制到达了或内存分配失败。
FirewallPolicy	已配置的防火墙策略是丢弃。
FirewallL4	L4检查失败。参见下面表关于更加粒状的丢弃原因(防火墙功能丢弃辩解了)。
FirewallL7	丢包由于L7检查。下面请参阅关于更加粒状的L7丢弃原因列表(防火墙功能丢弃
FirewallNotInitiator	TCP，UDP的一会话不是发起者或者ICMP。会话没有创建。例如，为了ICMP这在不精确信道处理正常的数据包能发生处理或。
FirewallNoNewSession	防火墙高可用性不允许新建的会话。
FirewallSyncookieMaxDst	为了提供招待基础的SYN溢出保护，有一单个目标文件的SYN速率作为SYN溢
FirewallSyncookie	SYNCOOLIE逻辑被触发。这指示与SYN Cookie的SYN/ACK发送，并且原始S
FirewallARStandby	不对称路由没有启用，并且冗余组不是在活动状态。

在QFP的防火墙功能丢弃计数器

限制用QFP全局丢弃计数器是没有在丢弃原因的粒度，并且某些丢弃原因例如**FirewallL4**获得很超载对点有排除故障的微不足道的作用。这在Cisco IOS XE 3.9 (15.3(2)S)从那以后被提高了，其中防火墙功能丢弃计数器被添加了。这给出更加粒状的套丢弃原因：

```
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
```

. . . .
下面防火墙功能丢弃原因和说明列表：

防火墙功能丢弃原因	说明
无效报头长度	数据包是很小不可能包含层4TCP,UDP，或者ICMP报头。 它可能 1. TCP报头长度< 20 2. UDP/ICMP报头长度< 8
无效UDP数据长度	UDP数据包长度不匹配在UDP报头指定的长度。
无效ACK编号	此丢弃能由这些原因之一造成： 1. 对TCP对等体的next_seq#的不是ACK等于。 2. ACK比TCP对等体发送的最最近的SEQ-极大。
无效ACK标志	在TCP SYNSENT和SYNRCVD状态下，预计ACK#与ISN+1是相等 此丢弃能由这些原因之一造成： 1. 期待ACK请标记，但是没设置在不同的TCP状态。 2. 除ACK标志之外，其他标志(类似RST)也设置。
无效TCP发起者	这发生，当： 1. 从TCP发起者的第一数据包不是SYN (非初始TCP数据段接收 2. 最初的SYN数据包有ACK标志设置。
与数据的SYN	SYN数据包包含有效负载。不支持这。
无效TCP标志	无效TCP标志可以造成由： 1. 除SYN之外， TCP初始SYN数据包有标志。 2. 在TCP监听状态， TCP对等体接收RST或ACK。 3. 其他响应方的数据包在SYN/ACK前接收。 4. 预计SYN/ACK没有从响应方接收。
无效分段在SYNSENT状态	一无效TCP数据段在SYNSENT状态造成由： 1. SYN/ACK有有效负载。 2. SYN/ACK有(PSH， URG， FIN)设置的其他标志。 3. 接收与有效负载的传输SYN。 4. 收到从发起者的非SYN数据包。
无效分段在SYNRCVD状态	一无效TCP数据段在SYNRCVD状态能造成由： 1. 接收与有效负载的retransit SYN从发起者。 2. 接收不是SYN/ACK、RST或者FIN从响应方的一无效分段。
无效顺序	当分段来自发起者时，这在SYNRCVD状态发生。它造成由： 1. Seq-比ISN是较少。 2. 如果接收方rcvd窗口大小是0和： 分段有有效负载或者 故障中分段(seq-比接收方LASTACK极大。 3. 如果接收方rcvd窗口大小是0，并且seq-在窗口之外下跌。 4. Seq-等于到ISN，但是不是SYN数据包。
无效窗口缩放选项	无效TCP窗口缩放选项是由不正确窗口缩放选项字节长度造成的。
在窗口外面的TCP	数据包太旧有-在另一个侧的ACK后的一个窗口。这能在已建立，
在发送的FIN以后的TCP额外的有效负载	在FIN以后接收的有效负载发送。这能在CLOSEWAIT状态发生。
TCP窗口溢出	当流入分段大小溢出接收方的窗口，这发生。然而，如果vTCP启
Retran用无效标志	一被重传的数据包由接收方已经确认。
TCP故障中分段	无序信息包将传送到检查的L7。如果L7不准许OOO分段，此数据
SYN 泛洪	在TCP SYN泛洪攻击下。在一定条件下，当对此主机的当前连接超
内部犯错- synflood失败的检查分配	在synflood检查期间， hostdb的分配发生故障。 推荐的操作：检查“显示平台硬件qfp活动功能防火墙内存”检查内存
Synflood停电丢弃	若被设定半开连接被超出，并且停电时间配置，对此IP地址的所有

半打开会话限制超出	数据包丢弃了由于允许半开放会话超出。 并且请检查设置“max-incomplete上下交替”，并且“”确保的一分钟
许多Pkt每个流	inspectable允许的数据包最大每个流被超出。最大数是25。
过多的ICMP错误信息包每个流	ICMP错误信息包最大每个流允许的被超出。最大是3。
从Rsp的Unexpect TCP有效载荷到Init	在SYNRCVD状态下， TCP收到有有效负载的一数据包从响应方至
内部错误-未确定方向	取消定义的数据包方向。
SYN在当前窗口里面	SYN数据包在一已经已建立TCP连接的窗口内被看到。
RST在当前窗口里面	RST数据包在一已经已建立TCP连接的窗口内观察。
杂散的分段	不应该通过TCP状态计算机接收例如在从响应方的监听状态接收的
ICMP内部错误-未接ICMP NAT信息	ICMP数据包nat'ed，但是内部NAT信息未命中。这是内部错误。
ICMP数据包在SCB close状态	在SCB CLOSE状态接收一ICMP数据包。
在ICMP数据包的未接IP报头	在ICMP数据包的缺少IP报头。
ICMP错误没有IP或ICMP	没有IP或ICMP的ICMP错误信息包在有效负载。由一畸形的数据包
ICMP犯错太短的Pkt	ICMP错误信息包是太短的。
ICMP犯错超过突发流量限制	ICMP错误pkt超过突发流量限制10。
ICMP犯错不可达的	ICMP错误pkt不可达的超过限制。仅第1不可得到的数据包允许通
ICMP犯错无效Seq-	嵌入式数据包Seq-不匹配产生ICMP错误数据包的seq-。
ICMP犯错无效Ack	在ICMP错误的无效ACK嵌入了数据包。
ICMP操作丢弃	已配置的ICMP操作是丢弃。
没有策略映射的区域对	策略不在区域对。它可能归结于(应用层网关)不配置的ALG打开应
不现在会话未命中和的策略	会话查找失败，并且策略不是存在检查此数据包。
不现在ICMP的错误和的策略	没有在区域对配置的策略的ICMP错误。
失败的分类	一个给的区域对的分类失败，当防火墙设法确定协议是否inspecta
分类操作丢弃	分类操作是丢弃。
安全策略Misconfig	失败的分类由于安全策略误配置。这能也归结于L7数据信道的没有
对响应方的发送RST	对响应方的发送RST在SYNSENT状态，当ACK#与ISN+1不是相等
防火墙策略丢弃	策略操作是丢弃。
片段丢弃	当第一个片段被丢弃时，请丢弃剩余的片段。
ICMP Firwall策略丢弃	ICMP被嵌入的数据包的策略操作是丢弃。
L7检查回归丢弃	L7 (ALG)决定对丢弃数据包。原因能从不同的ALG统计信息被找至
L7分段Pkt不准许	已接收被分段的数据包，当ALG不尊敬它。
L7片段Pkt不准许	已接收被分段的(或VFR)数据包，当ALG不尊敬它。
未知L7原始类型	无法识别的协议类型。

排除故障防火墙丢包

一旦丢弃原因从上述全局或防火墙功能丢弃计数器识别，其他故障排除步骤也许是需要的，如果这些丢包是意外的。除配置验证外为了保证配置为启用的防火墙功能是正确，经常要求采取有问题的通信流的数据包捕获发现数据包是否是畸形的或是否有任何协议或应用程序实施问题。

记录

ASR记录日志功能生成Syslog为了记录丢弃的数据包。这些Syslog在数据包为什么提供更多细节丢弃了。有sysloggings的两种类型：

1. 本地缓冲syslogging

2. 远程高速的记录日志

本地缓冲Syslogging

为了查出丢包的原因，您能使用通用的ZBFW故障排除，例如启用日志丢包。有两种方式配置丢包记录日志。

方法 1：请使用Inspect全局parameter-map为了记录丢弃的数据包。

```
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
....
```

方法 2：请使用自定义Inspect parameter-map为了记录仅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
```

这些信息传送对日志或根据ASR如何控制为记录配置。这是丢弃日志消息的示例。

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

本地的限制缓冲Syslogging

1. 这些日志是根据Cisco Bug ID被限制的速率CSCud09943。

2. 除非特定配置应用，这些日志也许不打印。例如，除非**日志关键字**指定，由类别默认值数据包的被丢弃的数据包不会被记录：

```
policy-map type inspect ZBFW_PMAP
class class-default
drop log
```

远程高速的记录日志

高速的记录日志(HSL)生成Syslog直接地从QFP并且发送它到已配置的Netflow HSL收集器。这是ZBFW的推荐的记录日志解决方案在ASR。

对于HSL，请使用此配置：

```
policy-map type inspect ZBFW_PMAP
class class-default
drop log
```

为了使用此配置， NetFlow收集器有能力在Netflow版本9上要求。这被选派

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

包跟踪使用有条件的匹配

打开有条件调试为了启用包跟踪然后启用这些功能的包跟踪：

```
policy-map type inspect ZBFW_PMAP
class class-default
drop log
```

> **注意**：因为ACL不是必要的，匹配情况能直接地使用IP地址。这配比作为允许双向跟踪的来源或目的地。此方法，如果没有允许修改配置，可以使用。例如：调试平台情况ipv4地址192.168.1.1/32。

打开包跟踪功能：

```
policy-map type inspect ZBFW_PMAP
class class-default
drop log
```

有两种方式使用此功能：

1. 输入**drop命令调试平台的数据包踪迹**为了跟踪仅丢弃的数据包。
2. debug命令**平台数据包踪迹丢弃的**排除将跟踪匹配情况，包括部分由设备检查/通过的所有数据包。

打开有条件调试：

```
policy-map type inspect ZBFW_PMAP
class class-default
drop log
```

运行测验，然后关闭调试：

```
policy-map type inspect ZBFW_PMAP
class class-default
drop log
```

现在信息可以显示到屏幕。在本例中， ICMP数据包丢弃的归结于防火墙策略：

```
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
```

显示平台数据包踪迹数据包<num>解码命令解码信息包报头题头信息和内容。此功能在XE3.11介绍：

```
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

嵌入式数据包捕获支持在Cisco IOS XE 3.7 (15.2(4)S)被添加了。欲了解更详细的信息，请参阅

Cisco IOS和IOS-XE配置示例的嵌入式数据包捕获。

调试

有条件调试

在XE3.10中，有条件调试将介绍。条件语句可以用于为了保证与情况是相关的仅的ZBFW功能日志调试消息。有条件调试使用ACL为了限制匹配ACL元素的日志。并且，在XE3.10之前，调试消息是更难读。debug输出在XE3.10改善使他们更加容易了解。

为了启用这些调试，请发出此命令：

```
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
```

注意必须通过ACL和定向性设置condition命令。有条件调试不会实现直到他们开始与debug命令**平台情况开始**。为了关闭有条件调试请使用debug命令**平台情况终止**。

```
debug platform condition stop
```

为了关闭有条件调试，请勿使用undebug all命令。为了关闭所有有条件调试，请使用命令：

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

在XE3.14之前， **ha**和**事件**调试没有条件的。结果， debug命令**平台情况功能fw dataplane从属方式全部**造成所有日志创建，下面选择的对立于情况。这可能创建使调试困难的附加噪声。

默认情况下，有条件的日志级别是**信息**。为了增加/请减小级别记录日志，使用命令：

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

采集并且查看调试

调试文件不会打印到控制台或监视器。所有调试写入到ASR的硬盘。调试写入到在文件夹**tracelogs下的**硬盘与名称**cpp_cp_F0-0.log.<date>**。为了查看调试写入的文件，使用输出：

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

每个调试文件将存储作为**cpp_cp_F0-0.log.<date>**文件。这些是可以复制与TFTP的ASR的正常文本文件。在ASR的日志文件最大数量是1Mb。在1Mb以后，调试写入到新的日志文件。所以每日志文件是时间戳的为了指示文件的开始。

日志文件也许在这些位置存在：

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

因为日志文件只显示，在他们被转动后，日志文件可以用此命令手工转动：

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

这立即创建"cpp_cp"日志文件并且开始在QFP的新的。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)
```

此命令允许将合并的调试文件到更加容易处理的单个文件。它合并在目录的所有文件并且隔行扫描他们准时根据。当日志非常冗长和在多个文件间时，创建这可帮助：

```
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]
```