

排除Firepower威胁防御(FTD)集群故障

目录

[简介](#)

[先决条件](#)

[要求](#)

[使用的组件](#)

[背景信息](#)

[配置](#)

[集群基础知识](#)

[NGFW架构](#)

[集群捕获](#)

[集群控制链路\(CCL\)消息](#)

[集群控制点\(CCP\)消息](#)

[集群运行状况检查\(HC\)机制](#)

[集群HC故障场景](#)

[集群数据平面连接建立](#)

[故障排除](#)

[集群故障排除简介](#)

[集群数据平面问题](#)

[NAT/PAT常见问题](#)

[分段处理](#)

[ACI问题](#)

[集群控制平面问题](#)

[设备无法加入集群](#)

[CCL上的MTU大小](#)

[集群设备之间的接口不匹配](#)

[数据/端口通道接口问题](#)

[由于CCL上的可达性问题导致大脑分裂](#)

[由于暂停的数据端口通道接口而禁用集群](#)

[集群稳定性问题](#)

[FXOS回溯](#)

[磁盘已满](#)

[溢出保护](#)

[简化模式](#)

[相关信息](#)

简介

本文档介绍Firepower下一代防火墙(NGFW)上集群设置的故障排除。

先决条件

要求

Cisco建议您了解以下主题（有关链接，请参阅“相关信息”部分）：

- Firepower平台架构
- Firepower集群配置和操作
- 熟悉FTD和Firepower可扩展操作系统(FXOS)CLI
- NGFW/数据平面日志
- NGFW/数据平面packet-tracer
- FXOS/数据平面捕获

使用的组件

- 硬件：Firepower 4125
- 软件：6.7.0（内部版本65）— 数据平面9.15(1)

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

背景信息

本文档中涉及的大多数项目也完全适用于自适应安全设备(ASA)集群故障排除。

配置

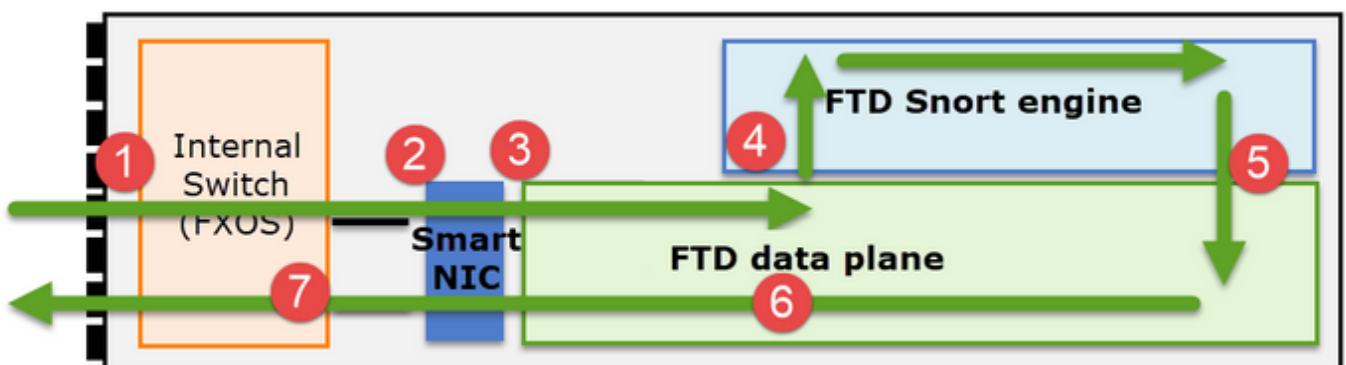
FMC和FXOS配置指南介绍了集群部署的配置部分：

- [面向Firepower威胁防御的集群](#)
- [部署Firepower威胁防御集群以实现可扩展性和高可用性](#)

集群基础知识

NGFW架构

了解Firepower 41xx或93xx系列如何处理中转数据包非常重要：



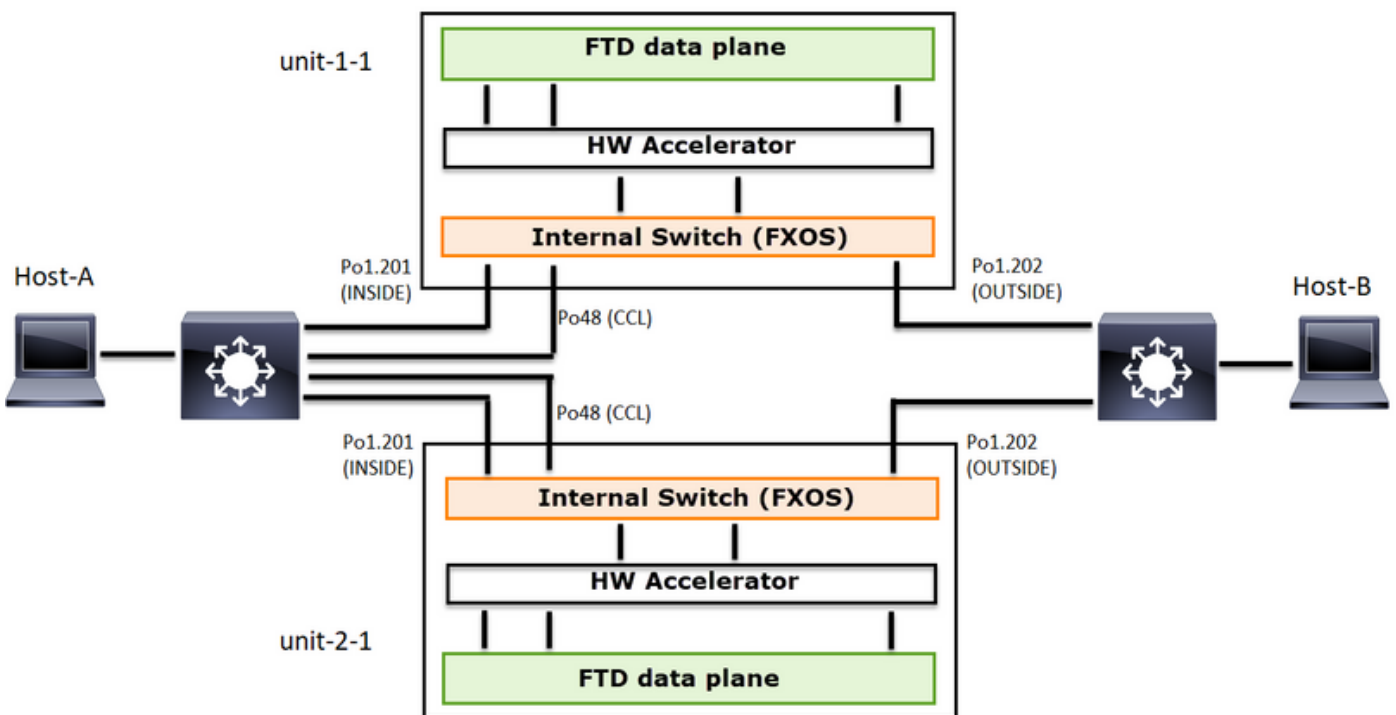
1. 数据包进入入口接口，由机箱内部交换机处理。
2. 数据包通过智能网卡。如果数据流被分流（硬件加速），则数据包将仅由智能网卡处理，然后发回网络。
3. 如果数据包未分流，它将进入主要执行L3/L4检查的FTD数据平面。
4. 如果策略需要，数据包将由Snort引擎进行检查（主要是L7检查）。
5. Snort引擎返回数据包的判定（例如，允许或阻止）。
6. 数据平面根据Snort的判定丢弃或转发数据包。
7. 数据包通过内部机箱交换机离开机箱。

集群捕获

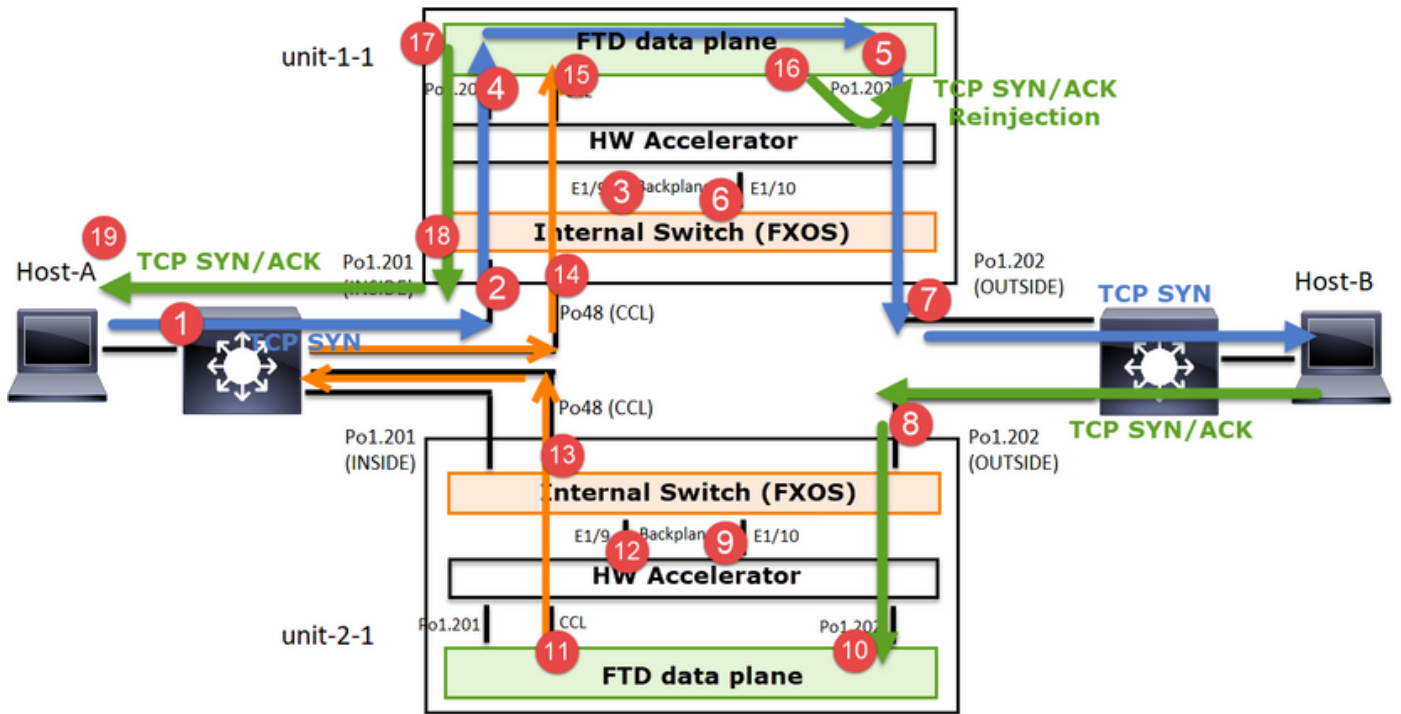
Firepower设备提供多个捕获点，用于提供对传输流的可视性。在排除故障和启用集群捕获时，主要挑战如下：

- 捕获数量随着集群中设备数量的增加而增加。
- 您需要了解集群处理特定流量的方式，才能跟踪通过集群的数据包。

下图显示一个双单元集群（例如，FP941xx/FP9300）：



对于非对称TCP连接建立，TCP SYN、SYN/ACK交换如下所示：



转发流量

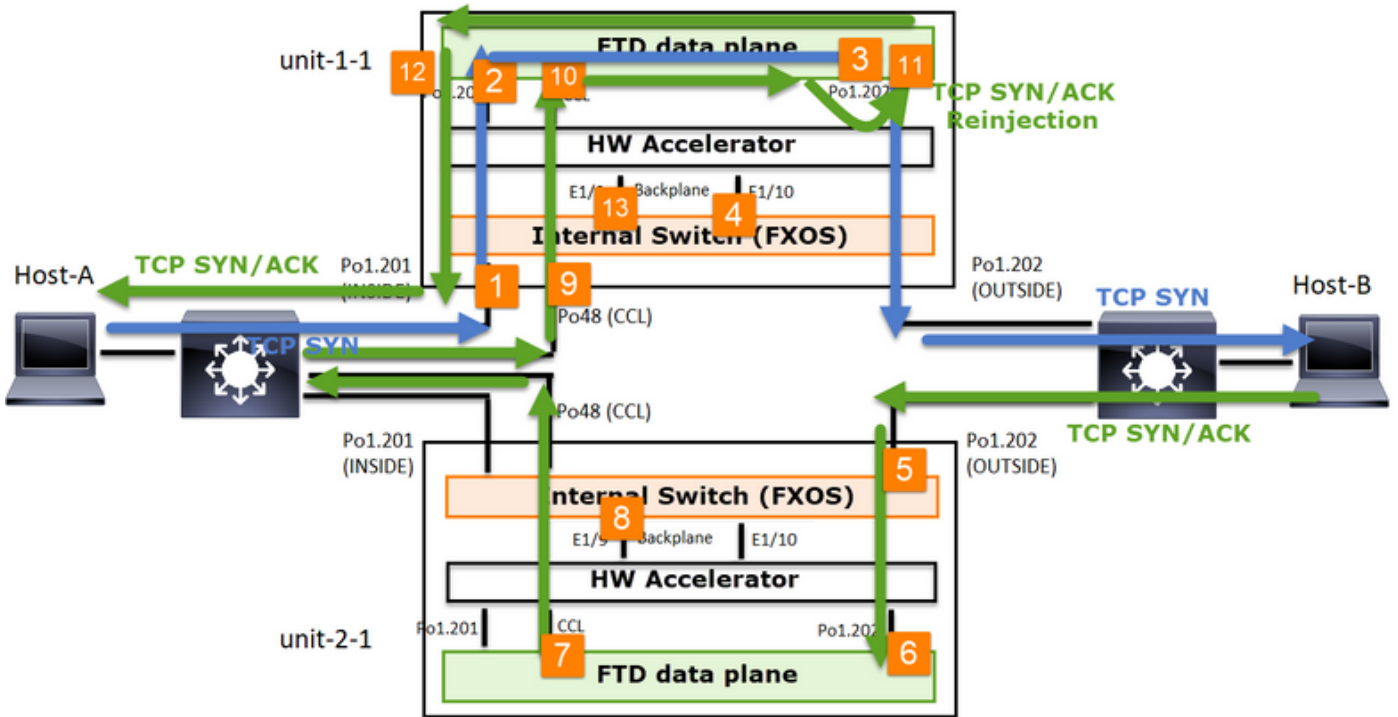
1. TCP SYN从主机A发送到主机B。
2. TCP SYN到达机箱 (Po1的一个成员) 。
3. TCP SYN通过其中一个机箱背板接口 (例如E1/9、 E1/10等) 发送到数据平面。
4. TCP SYN到达数据平面入口接口(Po1.201/INSIDE)。 在本示例中， unit1-1取得流的所有权，执行初始序列号(ISN)随机化，并对序列号中的所有权(cookie)信息进行编码。
5. TCP SYN从Po1.202/OUTSIDE (数据平面出口接口) 发出。
6. TCP SYN到达其中一个机箱背板接口 (例如E1/9、 E1/10等) 。
7. TCP SYN从机箱物理接口 (Po1的成员之一) 发送到主机B。

返回流量

8. TCP SYN/ACK从主机B发送并到达unit-2-1 (Po1的成员之一) 。
9. TCP SYN/ACK通过其中一个机箱背板接口 (例如E1/9、 E1/10等) 发送到数据平面。
10. TCP SYN/ACK到达数据平面入口接口(Po1.202/OUTSIDE)。
11. TCP SYN/ACK从集群控制链路(CCL)发送到设备1-1。默认情况下， ISN处于启用状态。因此，转发器可以在没有指挥交换机参与的情况下查找TCP SYN+ACK的所有者信息。对于其他数据包或禁用ISN时，会查询指挥交换机。
12. TCP SYN/ACK到达其中一个机箱背板接口 (例如， E1/9、 E1/10等) 。
13. TCP SYN/ACK从机箱物理接口 (Po48的成员之一) 发送至unit-1-1。
14. TCP SYN/ACK到达unit-1-1 (Po48的成员之一) 。
15. TCP SYN/ACK通过其中一个机箱背板接口转发到数据平面CCL端口通道接口 (nameif集群) 。
16. 数据平面将TCP SYN/ACK数据包重新发送到数据平面接口Po1.202/OUTSIDE。
17. 从Po1.201/INSIDE (数据平面出口接口) 向HOST-A发送TCP SYN/ACK。
18. TCP SYN/ACK通过其中一个机箱背板接口 (例如， E1/9、 E1/10等) ，并退出Po1的一个成员。
19. TCP SYN/ACK到达主机A。

有关此方案的更多详细信息，请阅读Cluster Connection Establishment Case Studies中的相关部分。

根据此数据包交换，所有可能的集群捕获点包括：



对于转发流量（例如，TCP SYN）捕获：

1. 机箱物理接口（例如，Po1成员）。此捕获是从机箱管理器(CM)UI或CM CLI配置的。
2. 数据平面入口接口（例如，Po1.201 INSIDE）。
3. 数据平面出口接口（例如，Po1.202 OUTSIDE）。
4. 机箱背板接口。FP4100有2个背板接口。FP9300上共有6个（每个模块2个）。由于您不知道数据包到达哪个接口，因此必须在所有接口上启用捕获。


对于返回流量（例如，TCP SYN/ACK）捕获：

5. 机箱物理接口（例如，Po1成员）。此捕获是从机箱管理器(CM)UI或CM CLI配置的。
6. 数据平面入口接口（例如，Po1.202 OUTSIDE）。
7. 由于数据包被重定向，下一个捕获点是数据平面CCL。
8. 机箱背板接口。同样，您必须在两个接口上启用捕获。
9. Unit-1-1机箱CCL成员接口。
10. 数据平面CCL接口（名称为集群）。
11. 入口接口（Po1.202外部）。这是从CCL到数据平面的重新注入数据包。
12. 数据平面出口接口（例如，Po1.201 INSIDE）。
13. 机箱背板接口。

如何启用集群捕获

FXOS捕获

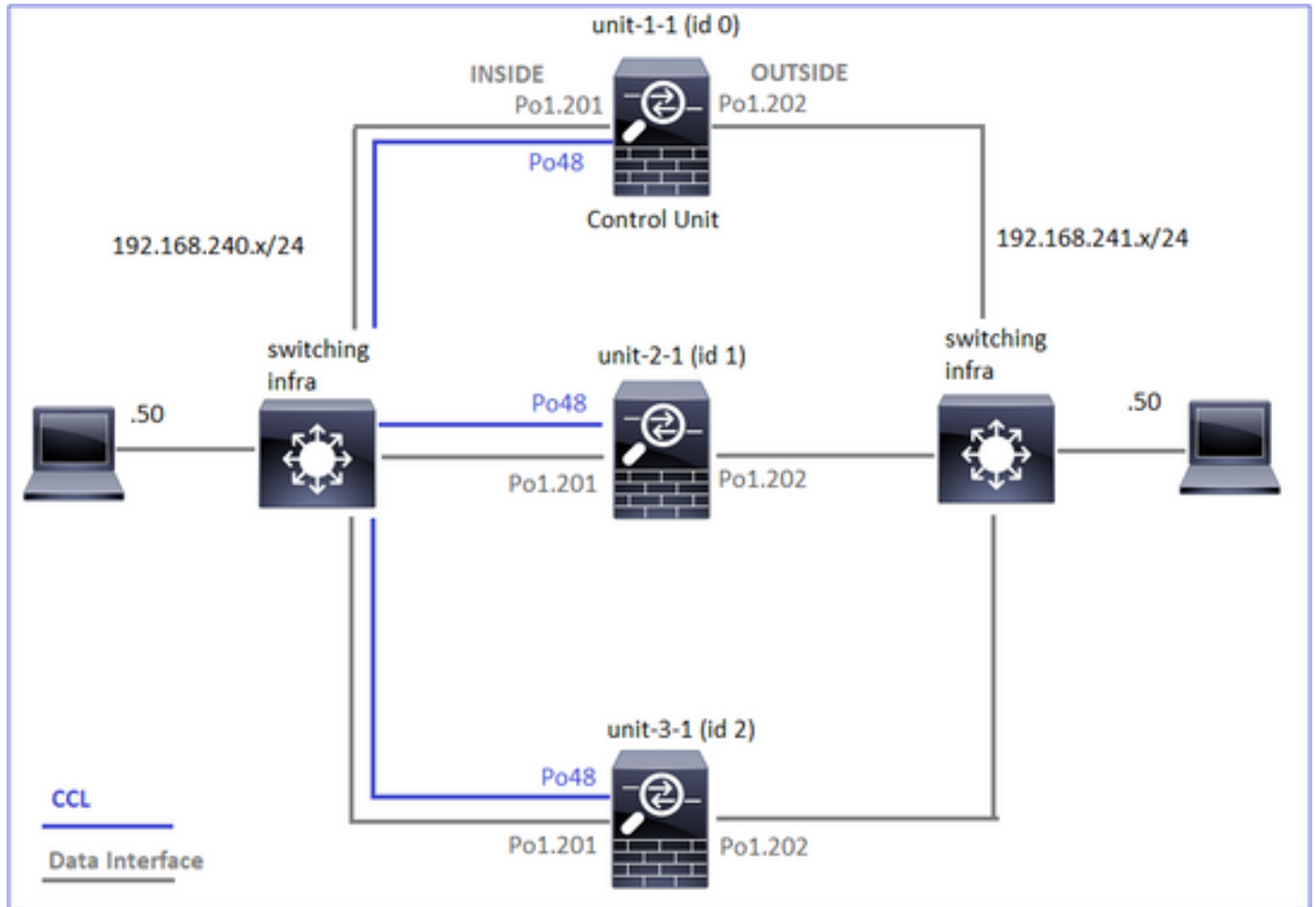
此过程在《FXOS配置指南》中描述：[数据包捕获](#)

 注意：FXOS捕获只能从内部交换机的角度在入口方向捕获。

数据平面捕获

建议在所有集群成员上启用捕获的方法是使用cluster exec命令。

考虑包含3个单元的集群：



要验证所有集群单元中是否存在活动捕获，请使用此命令：

```
<#root>
```

```
firepower#
```

```
cluster exec show capture
```

```
unit-1-1(LOCAL):*****
```

```
unit-2-1:*****
```

```
unit-3-1:*****
```

```
firepower#
```

要在Po1.201 (内部) 的所有设备上启用数据平面捕获，请执行以下操作：

```
<#root>
firepower#
cluster exec capture CAPI interface INSIDE
```

强烈建议指定捕获过滤器，并在预期流量很大时增加捕获缓冲区：

```
<#root>
firepower#
cluster exec capture CAPI buffer 33554432 interface INSIDE match tcp host 192.168.240.50 host 192.168.241.50 eq www
```

确认

```
<#root>
firepower#
cluster exec show capture

unit-1-1(LOCAL):*****
capture CAPI type raw-data buffer 33554432 interface INSIDE [Capturing - 5140 bytes]
  match tcp host 192.168.240.50 host 192.168.241.50 eq www

unit-2-1:*****
capture CAPI type raw-data buffer 33554432 interface INSIDE [Capturing - 260 bytes]
  match tcp host 192.168.240.50 host 192.168.241.50 eq www

unit-3-1:*****
capture CAPI type raw-data buffer 33554432 interface INSIDE [Capturing - 0 bytes]
  match tcp host 192.168.240.50 host 192.168.241.50 eq www
```

要查看所有捕获的内容（此输出可能很长），请执行以下操作：

```
<#root>
firepower#
terminal pager 24

firepower#
cluster exec show capture CAPI
```

```
unit-1-1(LOCAL):*****
```

```
21 packets captured
```

```
1: 11:33:09.879226 802.1Q vlan#201 PO 192.168.240.50.45456 > 192.168.241.50.80: S 2225395909:2225395909
2: 11:33:09.880401 802.1Q vlan#201 PO 192.168.241.50.80 > 192.168.240.50.45456: S 719653963:719653963(0
3: 11:33:09.880691 802.1Q vlan#201 PO 192.168.240.50.45456 > 192.168.241.50.80: . ack 719653964 win 229
4: 11:33:09.880783 802.1Q vlan#201 PO 192.168.240.50.45456 > 192.168.241.50.80: P 2225395910:2225396054
```

```
unit-2-1:*****
```

```
0 packet captured
```

```
0 packet shown
```

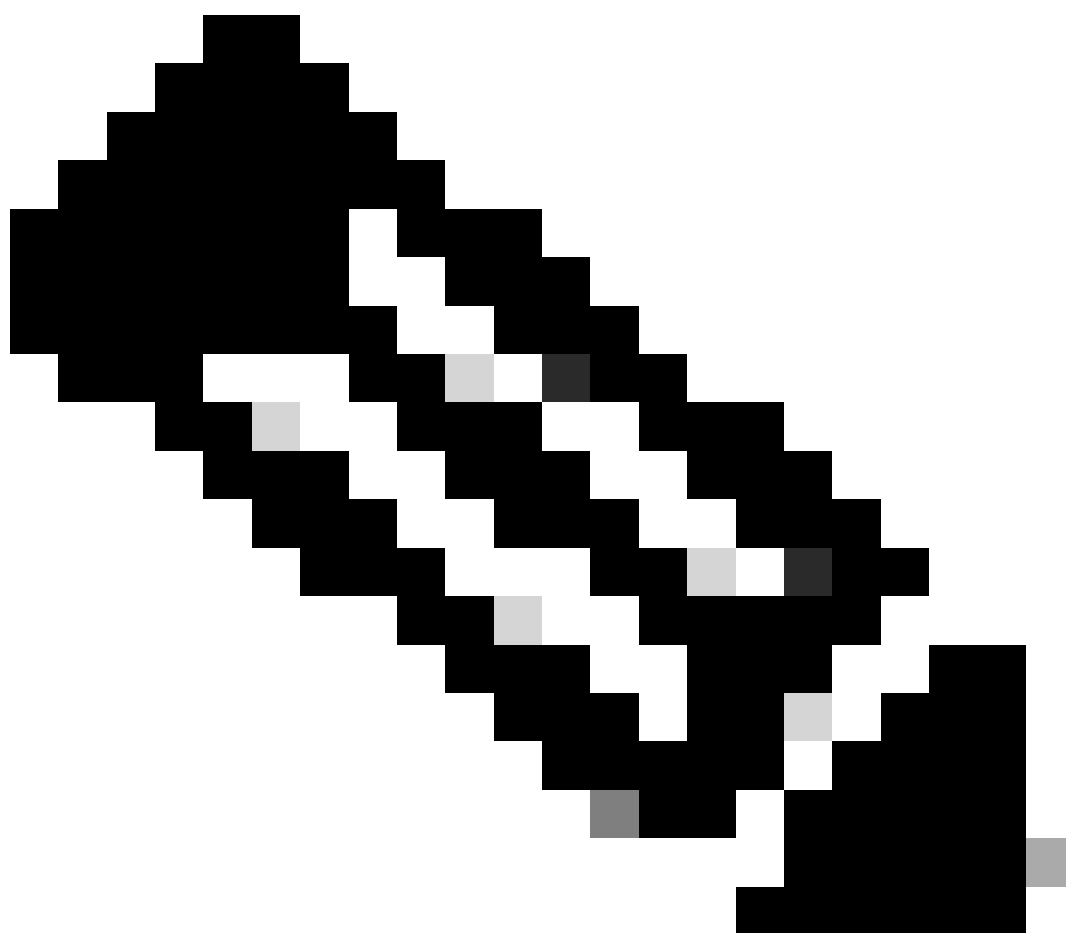
```
unit-3-1:*****
```

```
0 packet captured
```

```
0 packet shown
```

捕获跟踪

如果要查看入口数据包如何由每台设备上的数据平面处理，请使用trace关键字。这将跟踪前50个入口数据包。您可以跟踪多达1000个入口数据包。



注意：如果在一个接口上应用了多个捕获，则只能跟踪一个数据包一次。

要跟踪所有集群设备上接口OUTSIDE上的前1000个入口数据包，请执行以下操作：

```
<#root>
firepower#
cluster exec cap CAPO int OUTSIDE buff 33554432 trace trace-count 1000 match tcp host 192.168.240.50 hos
```

捕获感兴趣的数据流后，需要确保跟踪每台设备上感兴趣的数据包。需要记住的重要一点是，可以在设备1-1上配置特定数据#1，但是可以在其他设备上#2，以此类推。

在本示例中，您可以看到SYN/ACK是设备2-1上的数据包#2，但设备3-1上的数据包#1:

```
<#root>
firepower#
cluster exec show capture CAPO | include s.*ack

unit-1-1(LOCAL):*****
1: 12:58:31.117700 802.1Q vlan#202 PO 192.168.240.50.45468 > 192.168.241.50.80: S 441626016:441626016(0
2: 12:58:31.118341 802.1Q vlan#202 PO 192.168.241.50.80 > 192.168.240.50.45468:

s

301658077:301658077(0)

ack

441626017 win 28960 <mss 1460,sackOK,timestamp 1125686319 1115330849,nop,wscale 7>

unit-2-1:*****

unit-3-1:*****
1: 12:58:31.111429 802.1Q vlan#202 PO 192.168.241.50.80 > 192.168.240.50.45468:

s

301658077:301658077(0)

ack

441626017 win 28960 <mss 1460,sackOK,timestamp 1125686319 1115330849,nop,wscale 7>
```

要在本地设#2上跟踪数据包记录(SYN/ACK)，请执行以下操作：

```
<#root>
firepower#
```

```
cluster exec show cap CAPO packet-number 2 trace
```

```
unit-1-1(LOCAL):*****
```

```
2: 12:58:31.118341 802.1Q vlan#202 PO 192.168.241.50.80 > 192.168.240.50.45468:
```

```
s
```

```
301658077:301658077(0)
```

```
ack
```

```
441626017 win 28960 <mss 1460,sackOK,timestamp 1125686319 1115330849,nop,wscale 7>
```

```
Phase: 1
```

```
Type: CAPTURE
```

```
Subtype:
```

```
Result: ALLOW
```

```
Config:
```

```
Additional Information:
```

```
MAC Access list
```

```
...
```

要在远程设备上跟踪相同的数据包(SYN/ACK)，请执行以下操作：

```
<#root>
```

```
firepower#
```

```
cluster exec unit unit-3-1 show cap CAPO packet-number 1 trace
```

```
1: 12:58:31.111429 802.1Q vlan#202 PO 192.168.241.50.80 > 192.168.240.50.45468:
```

```
s
```

```
301658077:301658077(0)
```

```
ack
```

```
441626017 win 28960 <mss 1460,sackOK,timestamp 1125686319 1115330849,nop,wscale 7>
```

```
Phase: 1
```

```
Type: CAPTURE
```

```
Subtype:
```

```
Result: ALLOW
```

```
Config:
```

```
Additional Information:
```

```
MAC Access list
```

```
...
```

CCL捕获

要在CCL链路上（在所有设备上）启用捕获，请执行以下操作：

```
<#root>
```

```
firepower#
```

```
cluster exec capture CCL interface cluster
```

```
unit-1-1(LOCAL):*****
```

```
unit-2-1:*****
```

```
unit-3-1:*****
```

重新插入隐藏

默认情况下，数据平面数据接口上启用的捕获显示所有数据包：

- 从物理网络到达的
- 从CCL重新注入的

如果不想看到重新注入的数据包，请使用reinject-hide选项。如果您要验证流是否不对称，这将很有帮助：

```
<#root>
```

```
firepower#
```

```
cluster exec capture CAPI_RH reinject-hide interface INSIDE match tcp host 192.168.240.50 host 192.168.2
```

此捕获仅显示本地设备在指定接口上直接从物理网络而不是从其他集群设备接收的内容。

ASP丢包

如果要检查特定流的软件丢弃，可以启用asp-drop捕获。如果您不知道要关注哪个丢弃原因，请使用关键字all。此外，如果您对数据包负载不感兴趣，可以指定headers-only关键字。这允许您捕获的数据包数增加20至30倍：

```
<#root>
```

```
firepower#
```

```
cluster exec cap ASP type asp-drop all buffer 33554432 headers-only
```

```
unit-1-1(LOCAL):*****
```

```
unit-2-1:*****
```

```
unit-3-1:*****
```

此外，还可以在ASP捕获中指定所关注的IP:

```
<#root>
firepower#
cluster exec cap ASP type asp-drop all buffer 33554432 headers-only

match ip host 192.0.2.100 any
```

清除捕获

清除所有集群单元中运行的任何捕获的缓冲区。这不会停止捕获，但只会清除缓冲区：

```
<#root>
firepower#
cluster exec clear capture /all

unit-1-1(LOCAL):*****
unit-2-1:*****
unit-3-1:*****
```

停止捕获

有两种方法可以停止所有集群设备上的活动捕获。稍后您可以继续。

方式1

```
<#root>
firepower#
cluster exec cap CAPI stop

unit-1-1(LOCAL):*****
unit-2-1:*****
unit-3-1:*****
```

恢复

```
<#root>
firepower#
cluster exec no capture CAPI stop
```

```
unit-1-1(LOCAL):*****
unit-2-1:*****
unit-3-1:*****
```

途径2

```
<#root>
firepower#
cluster exec no capture CAPI interface INSIDE

unit-1-1(LOCAL):*****
unit-2-1:*****
unit-3-1:*****
```

恢复

```
<#root>
firepower#
cluster exec capture CAPI interface INSIDE

unit-1-1(LOCAL):*****
unit-2-1:*****
unit-3-1:*****
```

收集捕获

导出捕获的方法有多种。

方法1 — 到远程服务器

这样您就可以将捕获从数据平面上传到远程服务器（例如，TFTP）。捕获名称将自动更改以反映源设备：

```
<#root>
firepower#
cluster exec copy /pcap capture:CAPI tftp://192.168.240.55/CAPI.pcap
```

unit-1-1(LOCAL):*****

Source capture name [CAPI]?

Address or name of remote host [192.168.240.55]?

Destination filename [CAPI.pcap]?

INFO: Destination filename is changed to unit-1-1_CAPI.pcap !!!!!!!

81 packets copied in 0.40 secs

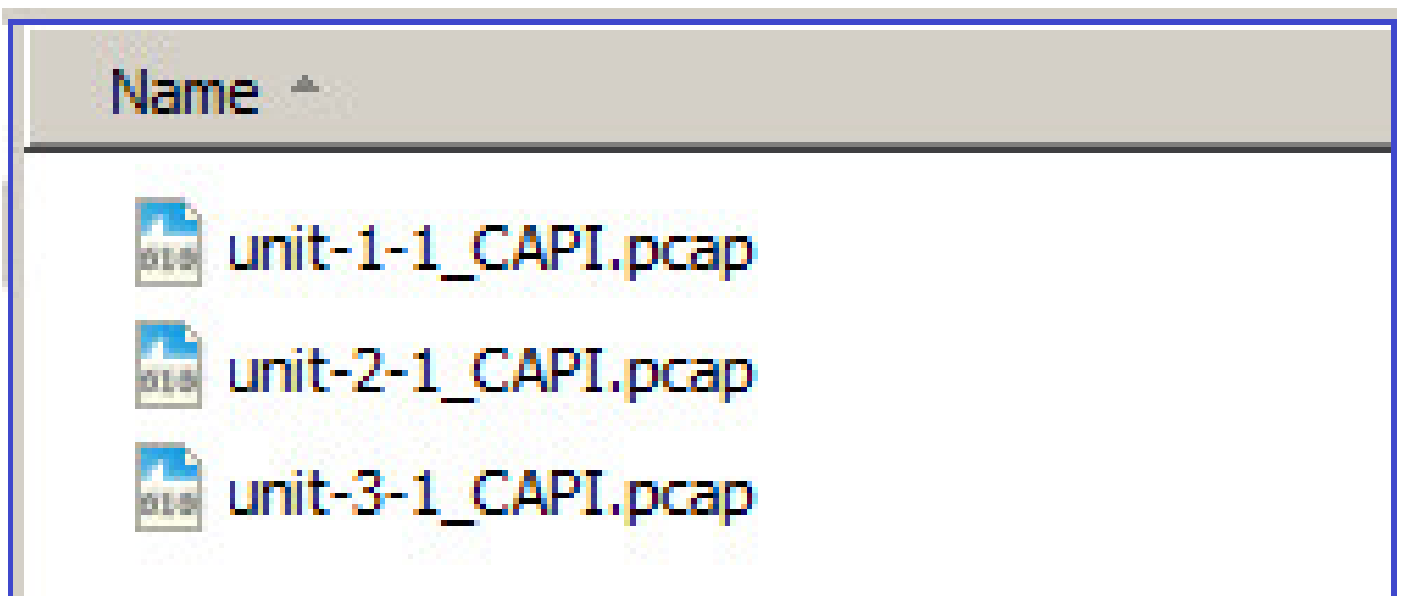
unit-2-1:*****

INFO: Destination filename is changed to unit-2-1_CAPI.pcap !

unit-3-1:*****

INFO: Destination filename is changed to unit-3-1_CAPI.pcap !

上載的pcap文件：



方法2 — 从FMC获取捕获

此方法仅适用于FTD。首先，将捕获复制到FTD磁盘：

```
<#root>
```

```
firepower#
```

```
cluster exec copy /pcap capture:CAPI disk0:CAPI.pcap
```

```
unit-1-1(LOCAL):*****
```

```
Source capture name [CAPI]?
```

```
Destination filename [CAPI.pcap]?
```

```
!!!!
```

```
62 packets copied in 0.0 secs
```

在专家模式下，将文件从/mnt/disk0/复制到/ngfw/var/common/目录：

```
<#root>
```

```
>
```

```
expert
```

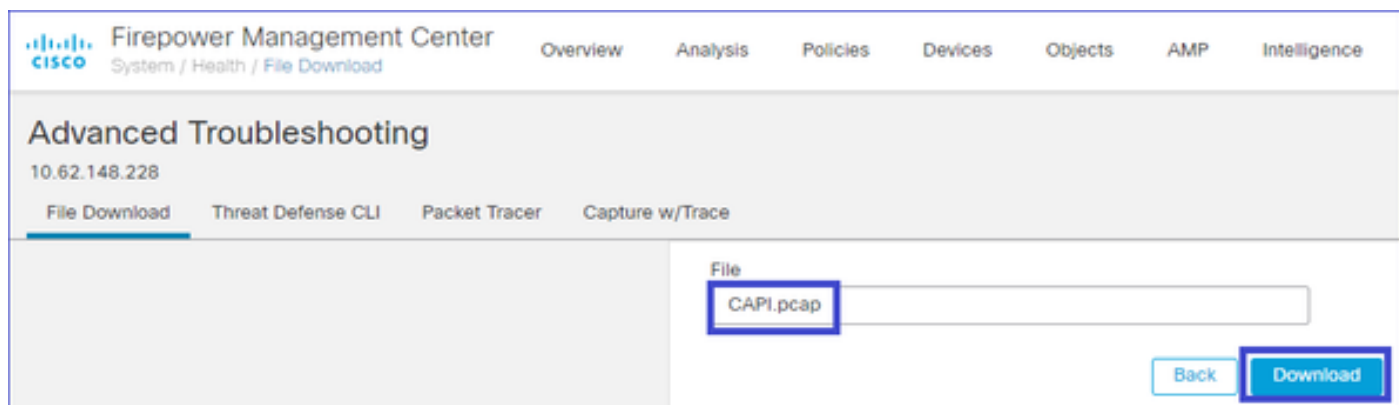
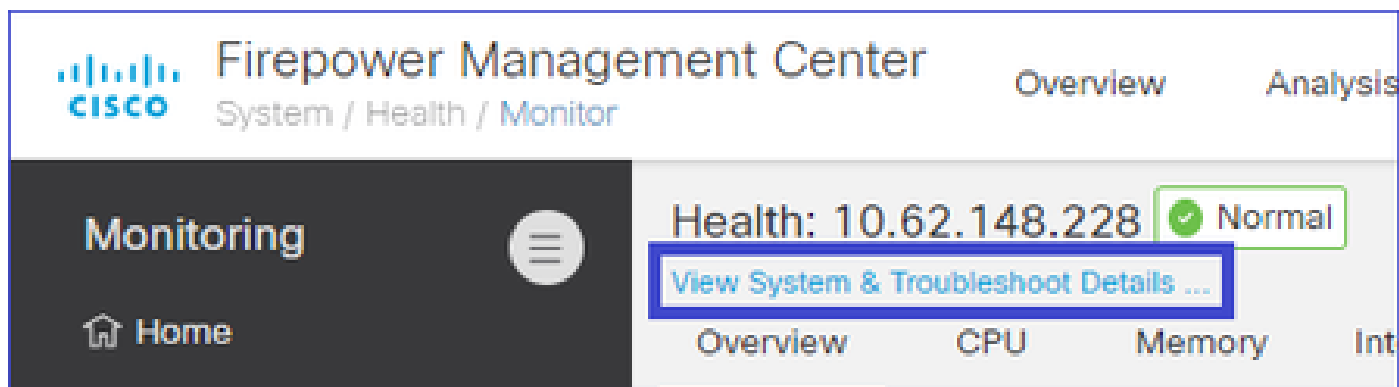
```
admin@firepower:~$
```

```
cd /mnt/disk0
```

```
admin@firepower:/mnt/disk0$
```

```
sudo cp CAPI.pcap /ngfw/var/common
```

最后，在FMC上，导航到System > Health > Monitor 部分。选择View System & Troubleshoot Details > Advanced Troubleshooting并获取捕获文件：



删除捕获

要从所有集群设备中删除捕获，请使用此命令：

```
<#root>
firepower#
cluster exec no capture CAPI

unit-1-1(LOCAL):*****
unit-2-1:*****
unit-3-1:*****
```

分流流

在FP41xx/FP9300上，可以静态（例如，Fastpath规则）或动态地将数据流分流到硬件加速器。有关流量分流的详细信息，请查看以下文档：

<https://www.cisco.com/c/en/us/support/docs/security/firepower-ngfw/212321-clarify-the-firepower-threat-defense-acc.html#anc22>

如果数据流被分流，则只有少数数据包通过FTD数据平面。其余部分由硬件加速器（智能网卡）处理。

从捕获的角度来看，这意味着如果仅启用FTD数据平面级别的捕获，您将看不到通过设备的所有数据包。在这种情况下，您还需要启用FXOS机箱级捕获。

集群控制链路(CCL)消息

如果在CCL上捕获数据，您会注意到集群设备会交换不同类型的消息。感兴趣的方面有：

协议	描述
UDP 49495	<ul style="list-style-type: none">· 群集心跳(keepalive)· L3广播(255.255.255.255)· 每个集群设备在运行状况检查保持时间值的1/3发送这些数据包。· 请注意，捕获中看49495的所有UDP数据包并非都是心跳· 心跳包含一个序列号。

UDP 4193	<p>集群控制协议数据路径消息</p> <ul style="list-style-type: none"> ·单播 ·这些数据包包含有关流所有者、指挥交换机、备份所有者等的信息 (元数据) 。这些 ISP 包括 : ·创建新流时 , “cluster add”消息从所有者发送到指挥交换机 ·当流终止时 , 从所有者向指挥交换机发送“群集删除”消息
数据包	属于通过集群的各种流量的数据包

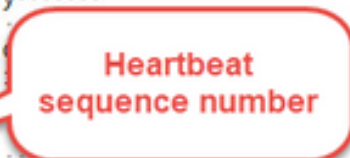
群集心跳

314	23.954349	192.222.1.1	255.255.255.255	UDP	205	49495 → 49495	Len=163
315	23.954364	192.222.1.1	255.255.255.255	UDP	205	49495 → 49495	Len=163
368	28.950976	192.222.1.1	255.255.255.255	UDP	205	49495 → 49495	Len=163
369	28.950992	192.222.1.1	255.255.255.255	UDP	205	49495 → 49495	Len=163

> Frame 314: 205 bytes on wire (1640 bits), 205 bytes captured (1640 bits)
 > Ethernet II, Src: Dell_00:01:8f (00:15:c5:00:01:8f), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
 > Internet Protocol Version 4, Src: 192.222.1.1, Dst: 255.255.255.255
 > User Datagram Protocol, Src Port: 49495, Dst Port: 49495
 * Data (163 bytes)
 Data: 010100fe00a300000000000000000000000000000001e008b0000000747524f5550310000...

```

0000  ff ff ff ff ff ff 00 15  c5 00 01 8f 08 00 45 00  .....E-
0010  00 bf a8 1f 00 00 ff 11  51 2f c0 de 01 01 ff ff  .....Q/.....
0020  ff ff c1 57 c1 57 00 ab  79 01 01 01 00 fe 00 a3  ...W-W..y.....
0030  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 1e  .....
0040  00 8b 00 00 00 07 47 52  4f 55 50 31 00 00 01 00  ....GR
0050  09 75 6e 69 74 2d 31 2d  31 00 00 02 00 09 75 6e  ..unit-1-
0060  69 74 2d 31 2d 31 00 00  03 00 01 00 00 04 00 01  ..it-1-1
0070  00 00 05 00 04 00 00 00  04 00 06 00 04 00 00 00  .....
0080  09 00 07 00 04 00 00 3a  98 00 08 00 0c 00 00 00  .....:
0090  00 c0 de 01 01 ff ff 00  00 00 09 00 02 01 1b 00  .....
00a0  0a 00 04 00 00 4e 9f 00  0b 00 0a 00 00 00 01 00  ....N.....
00b0  00 01 00 01 00 00 0c 00  08 00 00 00 00 00 00 00  .....
00c0  01 00 0d 00 08 00 00 00  00 00 00 00 00 00 00 00  .....
  
```



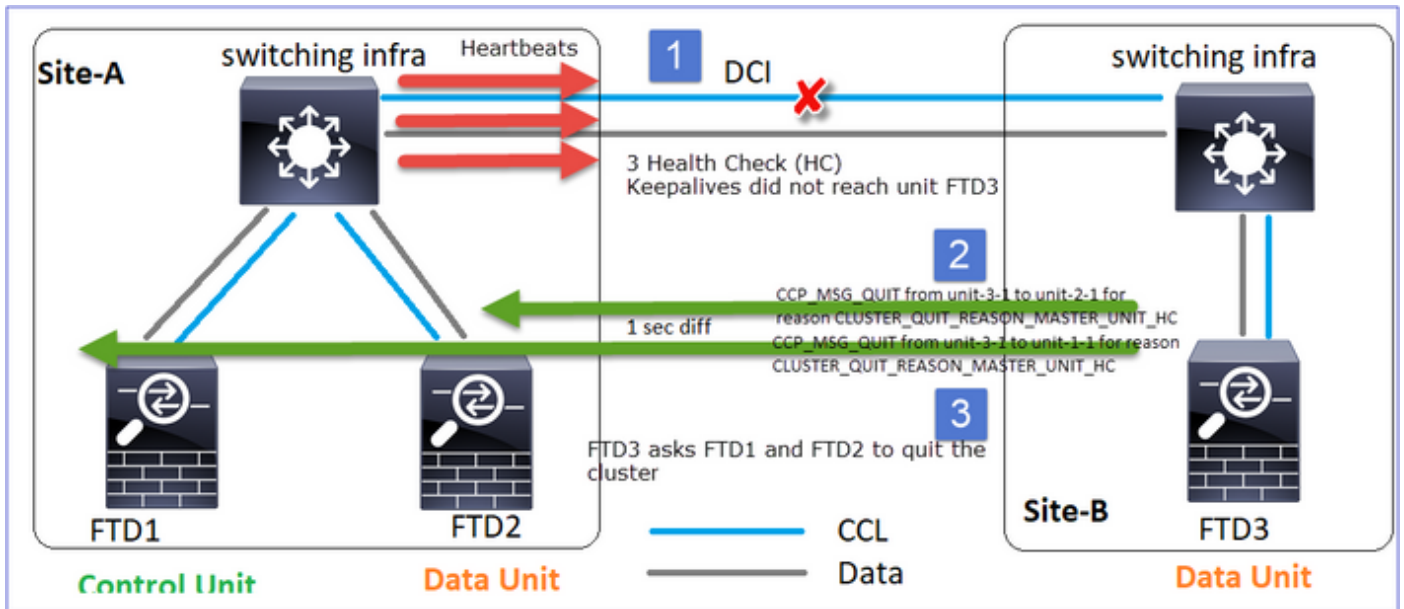
集群控制点(CCP)消息

除了心跳消息之外，在特定情况下还有大量通过CCL交换的集群控制消息。有些是单播消息，有些是广播。

CLUSTER_QUIT_REASON_PRIMARY_UNIT_HC

每当设备从控制节点丢失3个连续心跳消息时，它会通过CCL生成CLUSTER_QUIT_REASON_PRIMARY_UNIT_HC消息。此消息：

- 是单播。
- 它以1秒的间隔发送到每个单元。
- 当设备收到此消息时，退出集群 (禁用) 并重新加入。

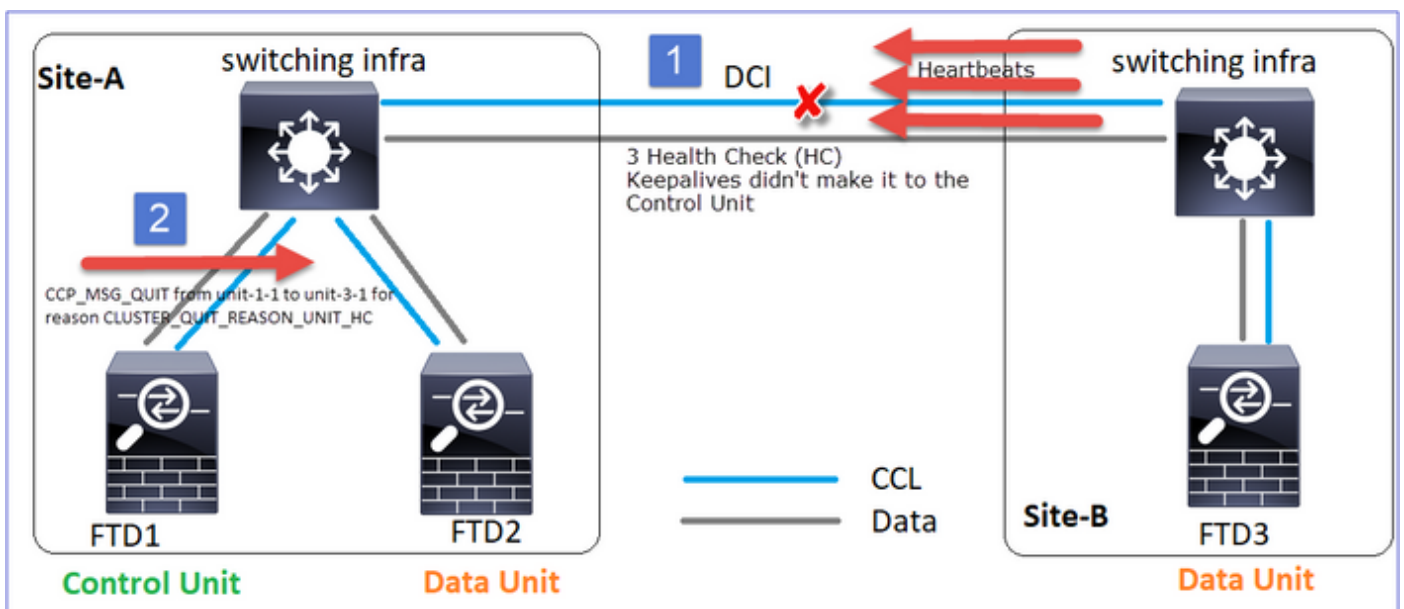


问：CLUSTER_QUIT_REASON_PRIMARY_UNIT_HC有何用途？

A.从unit-3-1(Site-B)的角度来看，它丢失了从站点A到unit-1-1和unit-2-1的连接，因此它需要尽快从成员列表中删除它们，否则，如果unit-2-1仍在其成员列表中，并且unit-2-1恰好是连接的导向器，则对unit-2-1的流查询失败，它可能会丢失数据包。

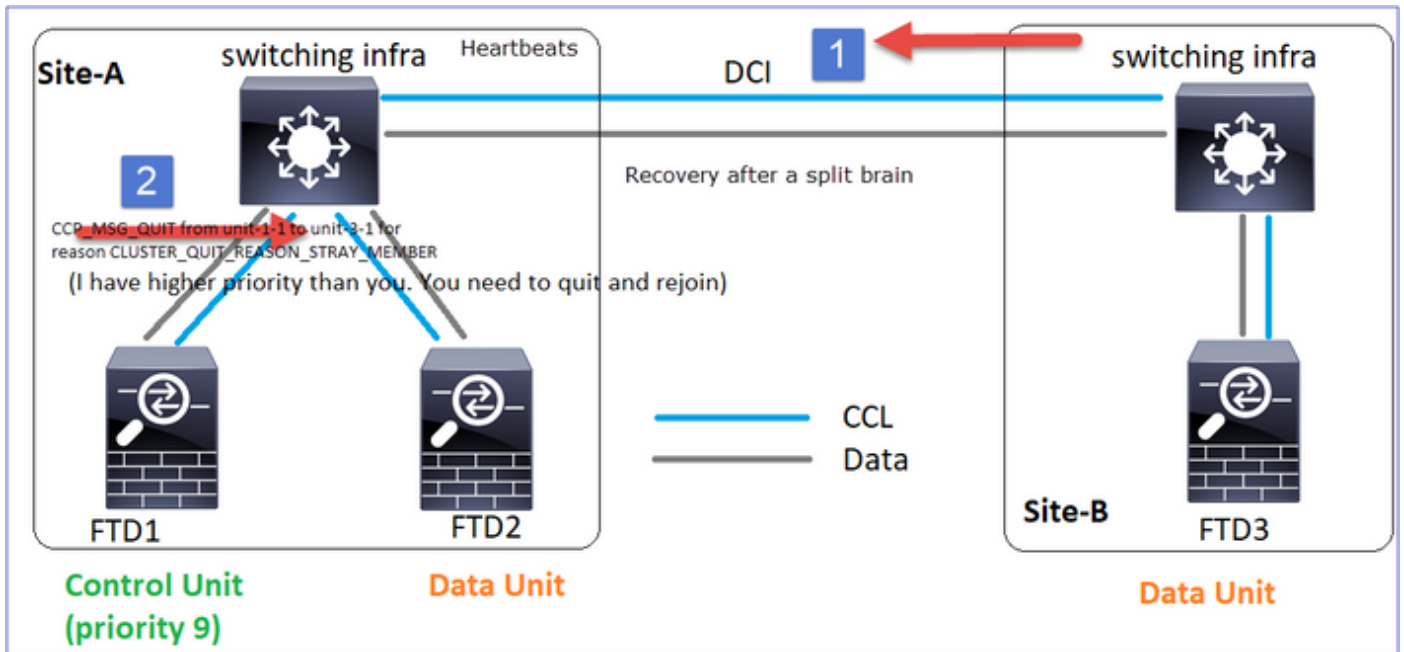
CLUSTER_QUIT_REASON_UNIT_HC

每当控制节点从数据节点丢失3个连续心跳消息时，它都会通过CCL发送CLUSTER_QUIT_REASON_UNIT_HC消息。此消息为单播。



CLUSTER_QUIT_REASON_STRAY_MEMBER

当分离分区与对等分区重新连接时，主控制单元将新数据节点视为分散成员，并以CLUSTER_QUIT_REASON_STRAY_MEMBER为原因接收CCP退出消息。



CLUSTER_QUIT_MEMBER_DROPPES

由数据节点生成并作为广播发送的广播消息。设备收到此消息后，会进入DISABLED状态。此外，自动重新连接不会启动：

```
<#root>
```

```
firepower#
```

```
show cluster info trace | include DROPOUT
```

```
Nov 04 00:22:54.699 [DEBUG]Receive CCP message: CCP_MSG_QUIT from unit-3-1 to unit-1-1 for reason
CLUSTER_QUIT_MEMBER_DROPOUT
```

```
Nov 04 00:22:53.699 [DEBUG]Receive CCP message: CCP_MSG_QUIT from unit-3-1 to unit-2-1 for reason
CLUSTER_QUIT_MEMBER_DROPOUT
```

集群历史记录显示：

```
<#root>
```

```
PRIMARY      DISABLED      Received control message DISABLE (
member dropout announcement
)
```

集群运行状况检查(HC)机制

要点

- 每个集群设备每1/3运行状况检查保持时间值发送一次检测信号到所有其他设备（广播 255.255.255.255），并使用UDP端口49495作为CCL上的传输。
- 每个集群设备使用轮询计时器和轮询计数值独立跟踪其他设备。
- 如果集群设备在心跳间隔内未收到来自集群对等设备的任何数据包（心跳或数据包），则会增加Poll count值。
- 当集群对等设备的Poll count值变为3时，对等设备被视为关闭。
- 只要接收到心跳，就会检查其序列号，并且在与先前接收到的心跳的差值不同于1的情况下，心跳丢弃计数器也会相应地增加。
- 如果集群对等体的Poll count计数器不同于0，且对等体收到数据包，则该计数器将重置为0值。

使用此命令检查集群运行状况计数器：

```
<#root>
```

```
firepower#
```

```
show cluster info health details
```

Unit (ID)	Heartbeat count	Heartbeat drops	Average gap (ms)	Maximum slip (ms)	Poll count
unit-2-1 (1)	650	0	4999	1	0
unit-3-1 (2)	650	0	4999	1	0

主列的说明

列	描述
单位(ID)	远程集群对等体的ID。
心跳计数	通过CCL从远程对等设备接收的心跳数。
心跳丢弃	丢失的心跳数。此计数器根据收到的心跳序列号来计算。
平均差距	收到的心跳的平均时间间隔。
轮询计数	当此计数器变为3时，设备将从集群中删除。轮询查询间隔与心

	跳间隔相同，但独立运行。
--	--------------

要重置计数器，请使用此命令：

```
<#root>
```

```
firepower#
```

```
clear cluster info health details
```

问：如何验证心跳频率？

A.检查平均差距值：

```
<#root>
```

```
firepower#
```

```
show cluster info health details
```

```
-----  
|                Unit (ID)| Heartbeat| Heartbeat|
```

```
Average
```

```
| Maximum|      Poll|  
|                | count|      drops|
```

```
gap (ms)
```

```
| slip (ms)|      count|
```

```
-----  
|                unit-2-1 ( 1)|      3036|          0|
```

```
999
```

```
|                1|          0|  
-----
```

问：如何更改FTD上的集群保持时间？

A.使用FlexConfig

在大脑分裂之后，谁成了控制节点？

A.具有最高优先级（最低数量）的单元：

```
<#root>
```

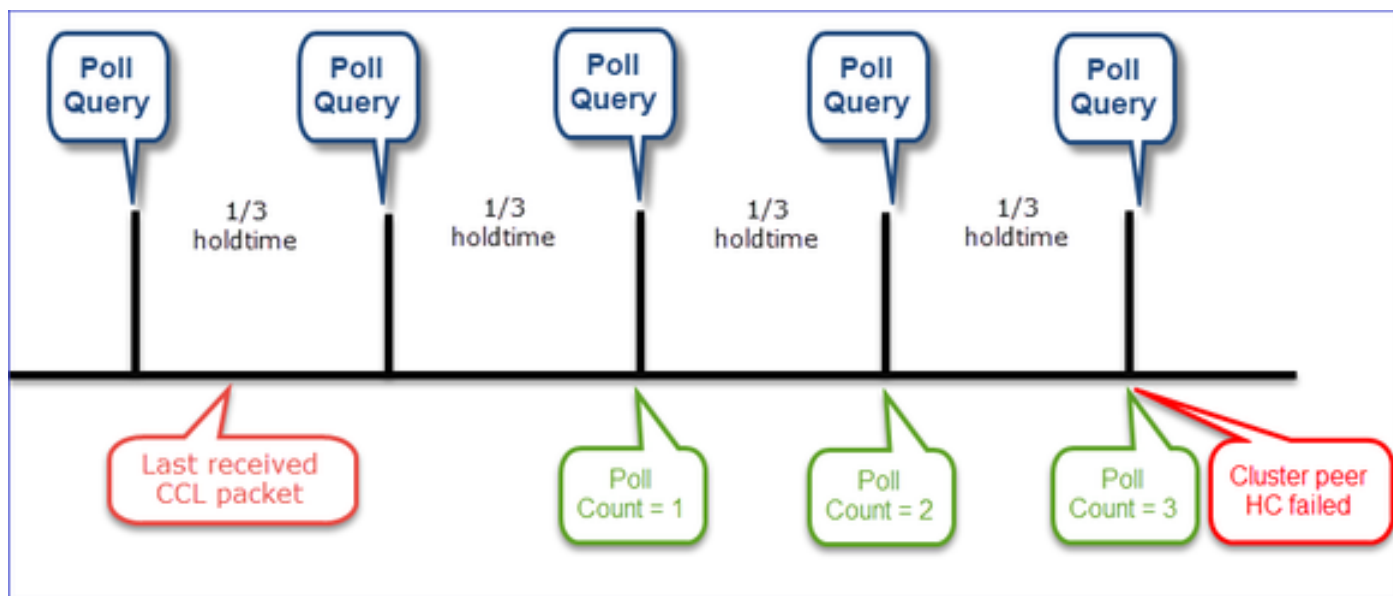
```
firepower#
```

```
show run cluster | include priority
```

```
priority 9
```

有关更多详细信息，请查看HC故障场景1。

簇状HC机制的可视化



指示计时器：min和max取决于最后收到的CCL数据包到达。

保持时间	轮询查询检查 (频率)	最小检测时间	最长检测时间
3秒 (默认)	约1秒	~3.01秒	~3.99秒
4 秒	~1.33秒	~4.01秒	~5.32秒
5 秒	~1.66秒	约5.01秒	~6.65秒
6 sec	约2秒	~6.01秒	~7.99秒
7 sec	~2.33秒	~7.01秒	~9.32秒

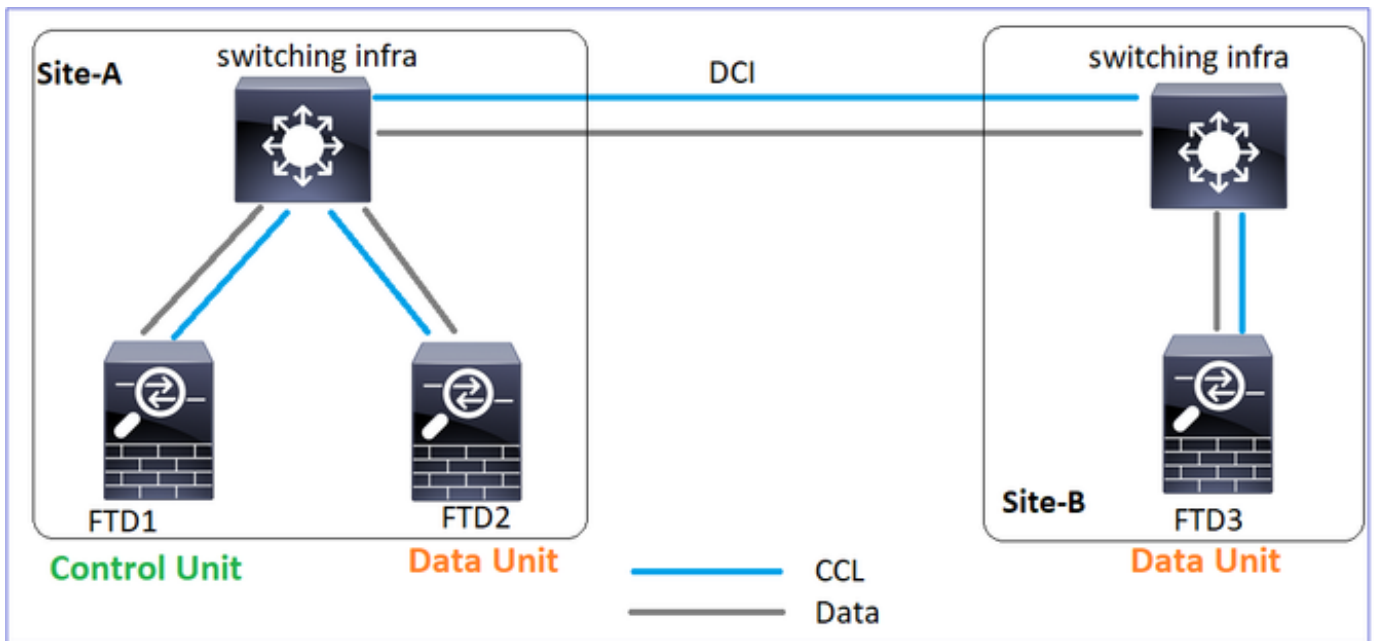
8 秒	~2.66秒	~8.01秒	~10.65秒
-----	--------	--------	---------

集群HC故障场景

本部分的目的是演示：

- 不同的集群HC故障场景。
- 如何关联不同的日志和命令输出。

拓扑



集群配置

Unit-1-1	Unit-2-1
<pre> cluster group GROUP1 key ***** local-unit unit-1-1 cluster-interface Port-channel48 ip 10.17.1.1 255.255.0.0 priority 9 health-check holdtime 3 health-check data-interface auto-rejoin 3 5 2 health-check cluster-interface auto-rejoin unlimited 5 1 health-check system auto-rejoin 3 5 2 health-check monitor-interface debounce-time 500 site-id 1 enable </pre>	<pre> cluster group GROUP1 key ***** local-unit unit-2-1 cluster-interface Port-channel48 ip 10.17.1.1 255.255.0.0 priority 17 health-check holdtime 3 health-check data-interface auto-rejoin 3 5 2 health-check cluster-interface auto-rejoin unlimited 5 1 health-check system auto-rejoin 3 5 2 health-check monitor-interface debounce-time 500 site-id 1 enable </pre>

集群状态

Unit-1-1	Unit-2-1
<pre><#root> firepower# show cluster info Cluster GROUP1: On Interface mode: spanned This is "unit-1-1" in state PRIMARY ID : 0 Site ID : 1 Version : 9.12(2)33 Serial No.: FCH22247LNK CCL IP : 10.17.1.1 CCL MAC : 0015.c500.018f Last join : 20:25:36 UTC Nov 1 2020 Last leave: 20:25:28 UTC Nov 1 2020 Other members in the cluster: Unit "unit-3-1" in state secondary ID : 1 Site ID : 2 Version : 9.12(2)33 Serial No.: FCH22247MKJ CCL IP : 10.17.3.1 CCL MAC : 0015.c500.038f Last join : 20:58:45 UTC Nov 1 2020 Last leave: 20:58:37 UTC Nov 1 2020 Unit "unit-2-1" in state SECONDARY ID : 2 Site ID : 1 Version : 9.12(2)33 Serial No.: FCH23157Y9N CCL IP : 10.17.2.1 CCL MAC : 0015.c500.028f Last join : 20:44:45 UTC Nov 1 2020 Last leave: 20:44:38 UTC Nov 1 2020</pre>	<pre><#root> firepower# show cluster info Cluster GROUP1: On Interface mode: spanned This is "unit-2-1" in state SECONDARY ID : 2 Site ID : 1 Version : 9.12(2)33 Serial No.: FCH23157Y9N CCL IP : 10.17.2.1 CCL MAC : 0015.c500.028f Last join : 20:44:46 UTC Nov 1 2020 Last leave: 20:44:38 UTC Nov 1 2020 Other members in the cluster: Unit "unit-1-1" in state PRIMARY ID : 0 Site ID : 1 Version : 9.12(2)33 Serial No.: FCH22247LNK CCL IP : 10.17.1.1 CCL MAC : 0015.c500.018f Last join : 20:25:36 UTC Nov 1 2020 Last leave: 20:25:28 UTC Nov 1 2020 Unit "unit-3-1" in state SECONDARY ID : 1 Site ID : 2 Version : 9.12(2)33 Serial No.: FCH22247MKJ CCL IP : 10.17.3.1 CCL MAC : 0015.c500.038f Last join : 20:58:45 UTC Nov 1 2020 Last leave: 20:58:37 UTC Nov 1 2020</pre>

场景 1

双向的CCL通信丢失大约4秒以上。

在失败之前

FTD1	FTD2	FTD3
站点A	站点A	站点B
控制节点	数据节点	数据节点

恢复后 (设备角色无更改)

FTD1	FTD2	FTD3
站点A	站点A	站点B
控制节点	数据节点	数据节点

分析

故障 (CCL通信丢失)。

The image shows three terminal windows side-by-side. The first window, titled 'unit-1-1 Control Unit', shows commands like 'clear cluster info trace' and 'clear cap /'. The second window, titled 'unit-2-1 Data Unit', shows a series of 'firepower#' prompts. The third window, titled 'unit-3-1 Data Unit', shows a warning about dynamic routing and a status change: 'Cluster unit unit-3-1 transitioned from SECONDARY to PRIMARY'.

单元3-1上的数据平面控制台消息：

<#root>

firepower#

WARNING: dynamic routing is not supported on management interface when cluster interface-mode is 'spanned'. If dynamic routing is configured on any management interface, please remove it.

Cluster unit unit-3-1 transitioned from SECONDARY to PRIMARY

Cluster disable is performing cleanup..done.
 All data interfaces have been shutdown due to clustering being disabled.
 To recover either enable clustering or remove cluster group configuration.

Unit-1-1集群跟踪日志 :

<#root>

firepower#

show cluster info trace | include unit-3-1

Nov 02 09:38:14.239 [INFO]Notify chassis de-bundle port for blade unit-3-1, stack 0x000055a8918307fb 0x
 Nov 02 09:38:14.239 [INFO]FTD - CD proxy received state notification (DISABLED) from unit unit-3-1
 Nov 02 09:38:14.239

[DEBUG]Send CCP message to all: CCP_MSG_QUIT from unit-1-1 to unit-3-1 for reason CLUSTER_QUIT_MEMBER_DR

Nov 02 09:38:14.239 [INFO]Notify chassis de-bundle port for blade unit-3-1, stack 0x000055a8917eb596 0x
 Nov 02 09:38:14.239

[DEBUG]Send CCP message to id 1: CCP_MSG_QUIT from unit-1-1 to unit-3-1 for reason CLUSTER_QUIT_REASON_UN

Nov 02 09:38:14.239 [CRIT]Received heartbeat event 'SECONDARY heartbeat failure' for member unit-3-1 (I

裂脑

Unit-1-1	Unit-2-1
<pre> <#root> firepower# show cluster info Cluster GROUP1: On Interface mode: spanned This is "unit-1-1" in state PRIMARY ID : 0 Site ID : 1 Version : 9.12(2)33 Serial No. : FCH22247LNK CCL IP : 10.17.1.1 CCL MAC : 0015.c500.018f Last join : 20:25:36 UTC Nov 1 2020 </pre>	<pre> <#root> firepower# show cluster info Cluster GROUP1: On Interface mode: spanned This is "unit-2-1" in state S ID : 2 Site ID : 1 Version : 9.12(2)33 Serial No. : FCH23157Y9N CCL IP : 10.17.2.1 CCL MAC : 0015.c500.028 Last join : 20:44:46 UTC Last leave : 20:44:38 UTC Other members in the cluster: </pre>

<pre> Last leave: 20:25:28 UTC Nov 1 2020 Other members in the cluster: Unit "unit-2-1" in state SECONDARY ID : 2 Site ID : 1 Version : 9.12(2)33 Serial No.: FCH23157Y9N CCL IP : 10.17.2.1 CCL MAC : 0015.c500.028f Last join : 20:44:45 UTC Nov 1 2020 Last leave: 20:44:38 UTC Nov 1 2020 </pre>	<pre> Unit "unit-1-1" in state PRIMARY ID : 0 Site ID : 1 Version : 9.12(2)33 Serial No.: FCH22247LNK CCL IP : 10.17.1.1 CCL MAC : 0015.c500.018 Last join : 20:25:36 UTC Last leave: 20:25:28 UTC </pre>
--	--

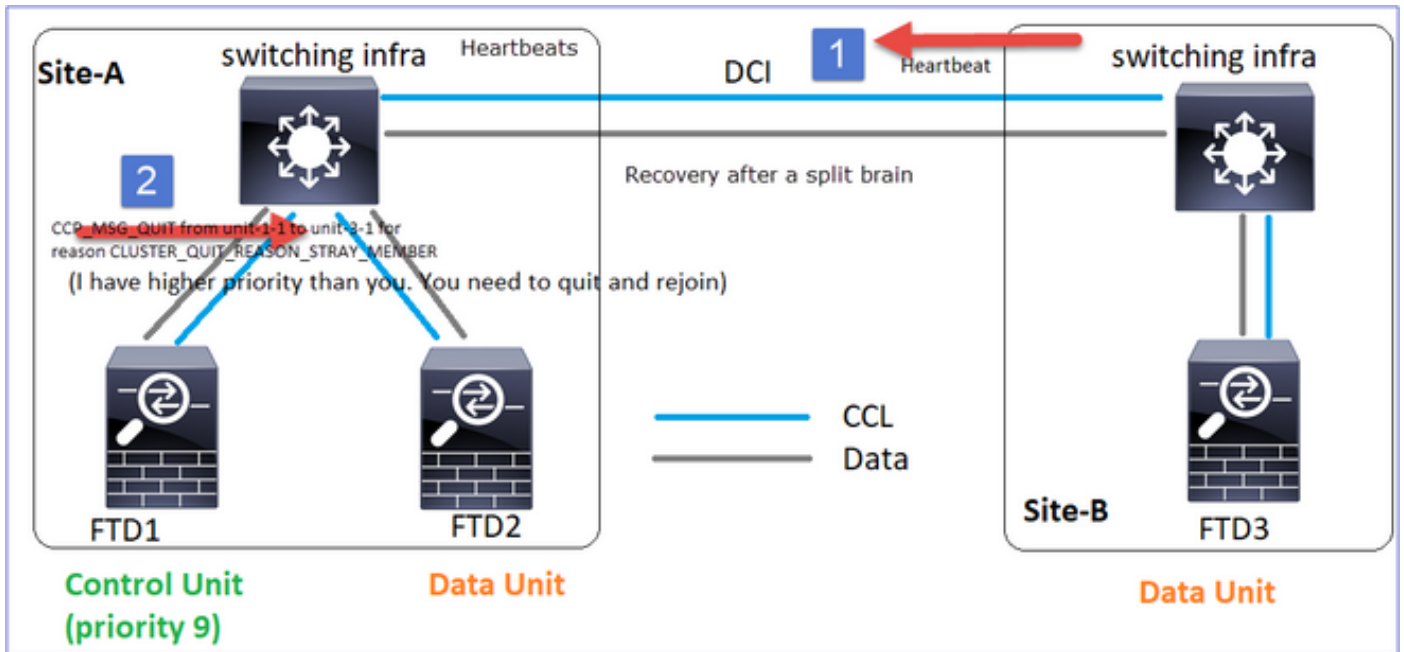
集群历史记录

Unit-1-1	Unit-2-1	Unit-3-1
无事件	无事件	<pre> <#root> 09:38:16 UTC Nov 2 2020 SECONDARY PRIMARY_POST_CONFIG Primary relinquished 09:38:17 UTC Nov 2 2020 PRIMARY_POST_CONFIG Primary Primary post config done </pre>

CCL通信恢复

Unit-1-1检测当前控制节点，并且由于unit-1-1具有更高的优先级，因此会向unit-3-1发送 CLUSTER_QUIT_REASON_STRAY_MEMBER消息以触发新的选举过程。最后，unit-3-1作为数据节点重新加入。

当拆分分区与对等分区重新连接时，主控制节点会将数据节点视为分散成员，并接收原因为 CLUSTER_QUIT_REASON_STRAY_MEMBER的CCP quit消息。



<#root>

Unit-3-1 console logs show:

```
Cluster unit unit-3-1 transitioned from PRIMARY to DISABLED
```

The 3DES/AES algorithms require a Encryption-3DES-AES activation key.

```
Detected Cluster Primart.
```

```
Beginning configuration replication from Primary.
```

```
WARNING: Local user database is empty and there are still 'aaa' commands for 'LOCAL'.
```

```
..
Cryptochecksum (changed): a9ed686f 8e2e689c 2553a104 7a2bd33a
End configuration replication from Primary.
```

```
Cluster unit unit-3-1 transitioned from DISABLED to SECONDARY
```

两台设备 (unit-1-1和unit-3-1) 均在其集群日志中显示 :

<#root>

```
firepower#
```

```
show cluster info trace | include retain
```

```
Nov 03 21:20:23.019 [CRIT]Found a split cluster with both unit-1-1 and unit-3-1 as primary units. Prima
```

```
Nov 03 21:20:23.019 [CRIT]Found a split cluster with both unit-1-1 and unit-3-1 as primary units. Prima
```

还有为拆分大脑生成的系统日志消息：

```
<#root>
```

```
firepower#
```

```
show log | include 747016
```

```
Nov 03 2020 21:20:23: %FTD-4-747016: Clustering: Found a split cluster with both unit-1-1 and unit-3-1
```

```
Nov 03 2020 21:20:23: %FTD-4-747016: Clustering: Found a split cluster with both unit-1-1 and unit-3-1
```

集群历史记录

Unit-1-1	Unit-2-1	Unit-3-1
无事件	无事件	<pre> <#root> 09:47:33 UTC Nov 2 2020 Primary DISABLED Detected a splitted cluster 09:47:38 UTC Nov 2 2020 DISABLED ELECTION Enabled from CLI 09:47:38 UTC Nov 2 2020 ELECTION SECONDARY_COLD Received cluster control me 09:47:38 UTC Nov 2 2020 SECONDARY_COLD SECONDARY_APP_SYNC Client progression done 09:48:18 UTC Nov 2 2020 SECONDARY_APP_SYNC SECONDARY_CONFIG SECONDARY application o 09:48:29 UTC Nov 2 2020 SECONDARY_CONFIG SECONDARY_FILESYS Configuration replicati 09:48:30 UTC Nov 2 2020 SECONDARY_FILESYS SECONDARY_BULK_SYNC Client progression done 09:48:54 UTC Nov 2 2020 SECONDARY_BULK_SYNC SECONDARY Client progression done </pre>

场景 2

双向的CCL通信丢失大约3至4秒。

在失败之前

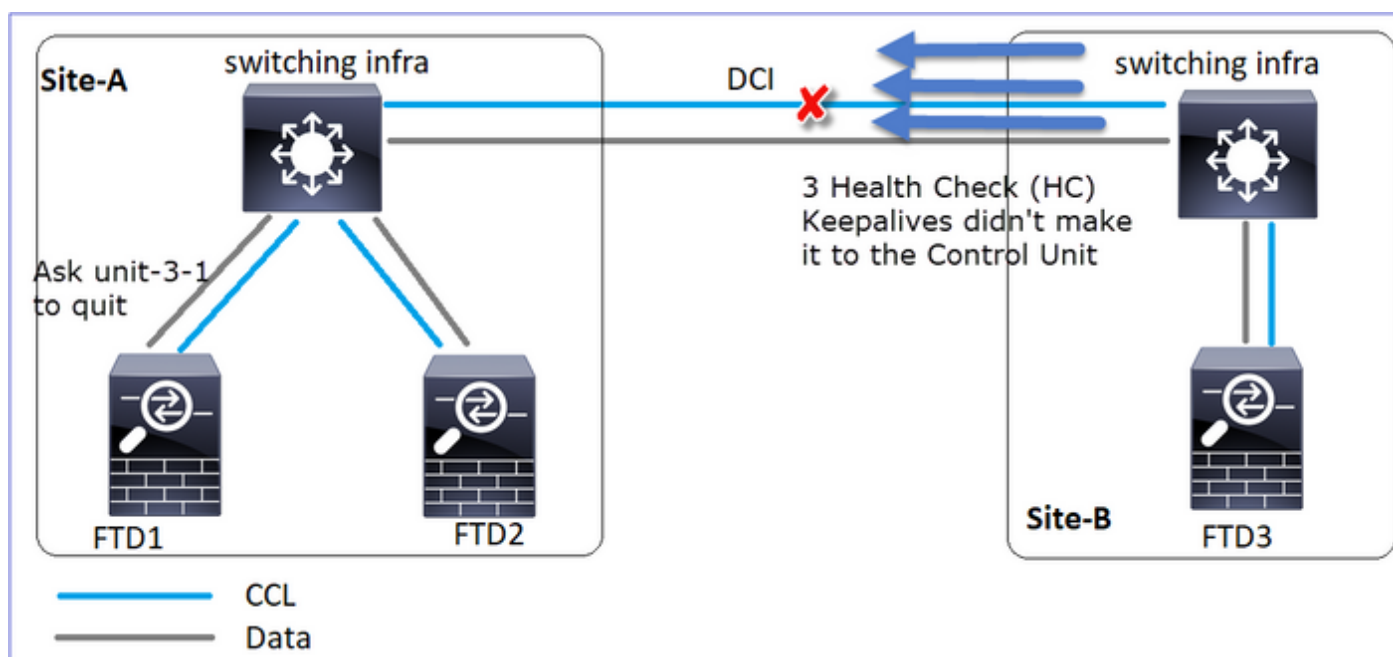
FTD1	FTD2	FTD3
站点A	站点A	站点B
控制节点	数据节点	数据节点

恢复后 (设备角色无更改)

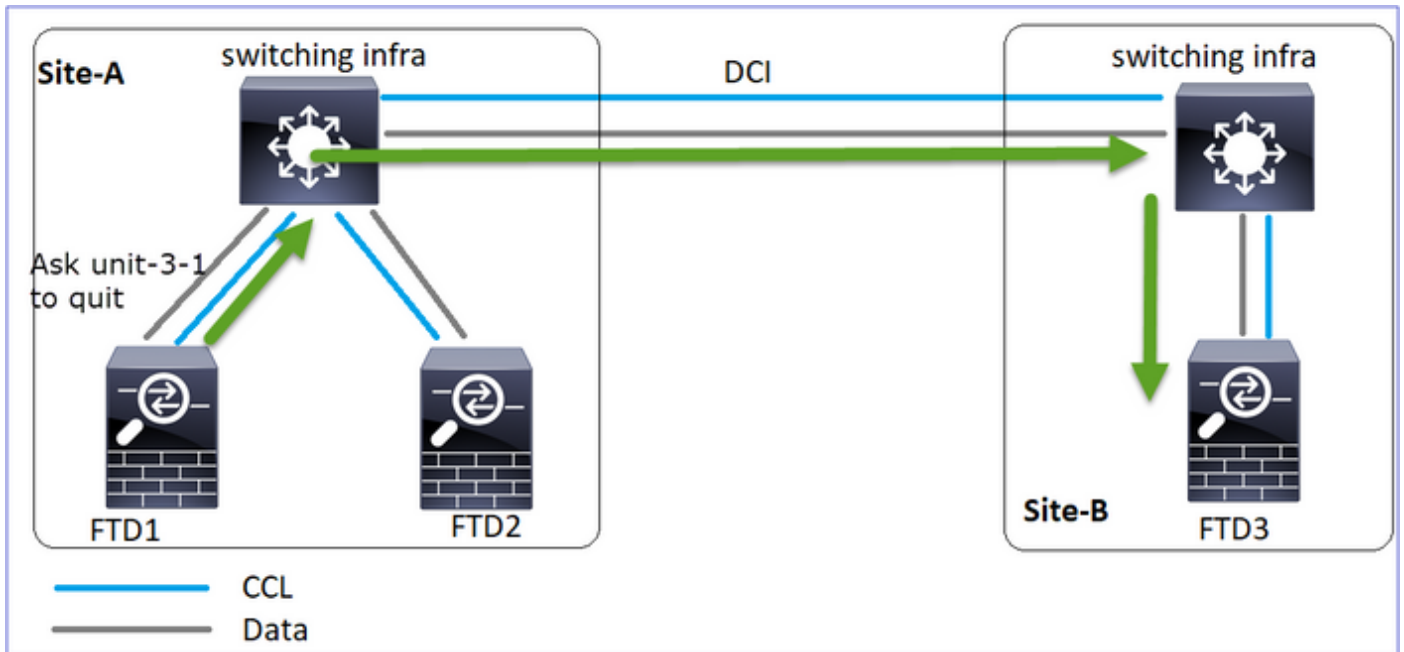
FTD1	FTD2	FTD3
站点A	站点A	站点B
控制节点	数据节点	数据节点

分析

活动1:控制节点从设备3-1丢失3个HC，并向设备3-1发送消息以离开集群。



活动2:CCL恢复非常快，来自控制节点的CLUSTER_QUIT_REASON_STRAY_MEMBER消息到达了远程端。Unit-3-1直接进入DISABLED模式，并且没有拆分大脑



在unit-1-1(control)上，您可以看到：

```
<#root>
```

```
firepower#
Asking SECONDARY unit unit-3-1 to quit because it failed unit health-check.
```

```
Forcing stray member unit-3-1 to leave the cluster
```

在unit-3-1 (数据节点) 上，您可以看到：

```
<#root>
```

```
firepower#
```

```
Cluster disable
```

```
is performing cleanup..done.
All data interfaces have been shutdown due to clustering being disabled. To recover either enable cluster
Cluster unit unit-3-1 transitioned from SECONDARY to DISABLED
```

集群单元unit-3-1转换为DISABLED状态，一旦CCL通信恢复，它将作为数据节点重新加入：

```
<#root>
```

```
firepower#
```

```
show cluster history
```

20:58:40 UTC Nov 1 2020

SECONDARY DISABLED Received control message DISABLE (stray member)

20:58:45 UTC Nov 1 2020

DISABLED ELECTION Enabled from CLI

20:58:45 UTC Nov 1 2020

ELECTION SECONDARY_COLD Received cluster control message

20:58:45 UTC Nov 1 2020

SECONDARY_COLD SECONDARY_APP_SYNC Client progression done

20:59:33 UTC Nov 1 2020

SECONDARY_APP_SYNC SECONDARY_CONFIG SECONDARY application configuration sync done

20:59:44 UTC Nov 1 2020

SECONDARY_CONFIG SECONDARY_FILESYS Configuration replication finished

20:59:45 UTC Nov 1 2020

SECONDARY_FILESYS SECONDARY_BULK_SYNC Client progression done

21:00:09 UTC Nov 1 2020

SECONDARY_BULK_SYNC SECONDARY
Client progression done

场景 3

双向的CCL通信丢失大约3至4秒。

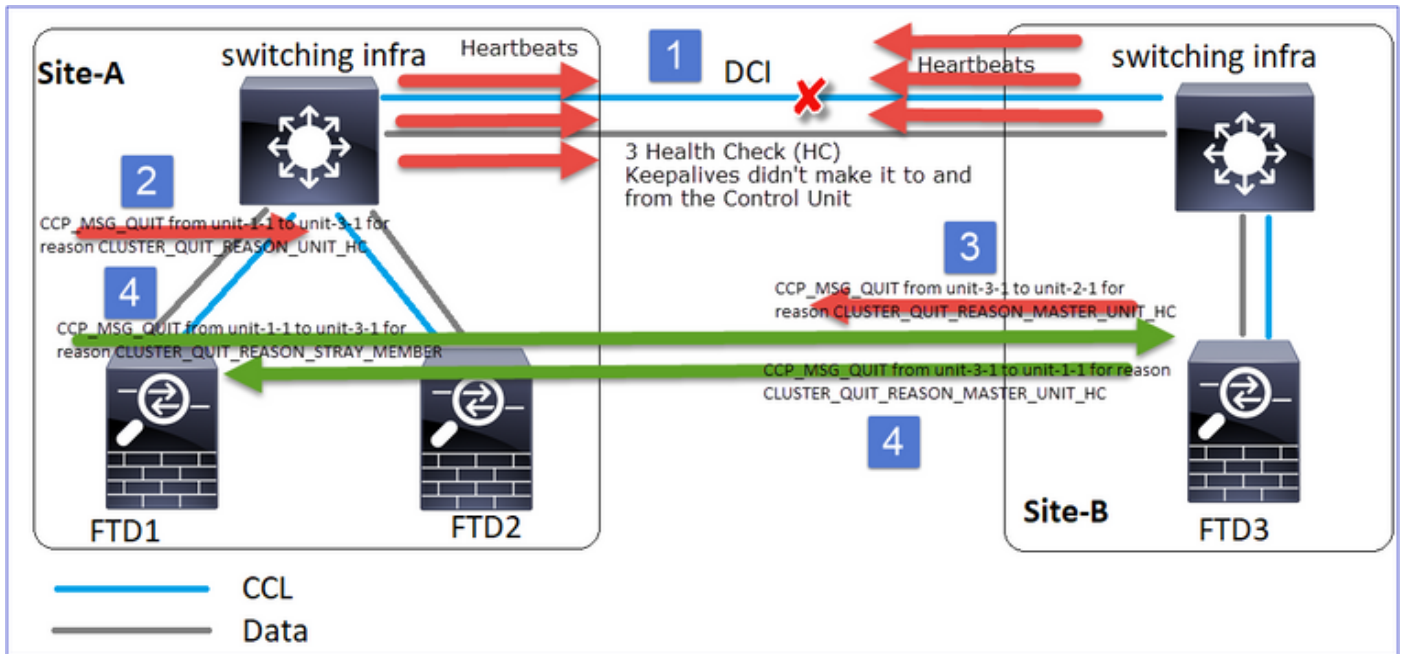
在失败之前。

FTD1	FTD2	FTD3
站点A	站点A	站点B
控制节点	数据节点	数据节点

恢复后 (控制节点已更改)。

FTD1	FTD2	FTD3
站点A	站点A	站点B
数据节点	控制节点	数据节点

分析



1. CCL关闭。
2. Unit-1-1不会从unit-3-1获得3条HC消息，而是向unit-3-1发送QUIT消息。此消息永远无法到达unit-3-1。
3. Unit-3-1向unit-2-1发送QUIT消息。此消息永远无法到达unit-2-1。

CCL恢复。

4. Unit-1-1会看到unit-3-1将自己通告为控制节点，并将QUIT_REASON_STRAY_MEMBER消息发送到unit-3-1。当unit-3-1收到此消息后，该消息将进入DISABLED状态。同时，Unit-3-1向Unit-1-1发送QUIT_REASON_PRIMARY_UNIT_HC消息并要求其退出。一旦设备-1-1收到此消息，该消息将进入DISABLED状态。

集群历史记录

```
Unit-1-1
<#root>
19:53:09 UTC Nov 2 2020
PRIMARY DISABLED
    Received control message DISABLE
                                (primary unit health check failure)
19:53:13 UTC Nov 2 2020
DISABLED ELECTION Enabled from CLI
19:53:13 UTC Nov 2 2020
ELECTION SECONDARY_COLD Received cluster control message
19:53:13 UTC Nov 2 2020
SECONDARY_COLD SECONDARY_APP_SYNC Client progression done
```

```

19:54:01 UTC Nov 2 2020
SECONDARY_APP_SYNC      SECONDARY_CONFIG      SECONDARY application configur
19:54:12 UTC Nov 2 2020
SECONDARY_CONFIG      SECONDARY_FILESYS     Configuration replication fini
19:54:13 UTC Nov 2 2020
SECONDARY_FILESYS     SECONDARY_BULK_SYNC   Client progression done
19:54:37 UTC Nov 2 2020
SECONDARY_BULK_SYNC

```

SECONDARY

Client progression done

场景 4

CCL通信丢失约3-4秒

在失败之前

FTD1	FTD2	FTD3
站点A	站点A	站点B
控制节点	数据节点	数据节点

恢复后 (控制节点更改了站点)

FTD1	FTD2	FTD3
------	------	------

站点A	站点A	站点B
数据节点	数据节点	控制节点

分析

失败

```

firepower# Cluster disable is performing cleanup..done.
All data interfaces have been shutdown due to clustering being disabled. To recover either enable clustering or remove cluster group configuration.
Cluster unit unit-1-1 transitioned from [ ] to DISABLED

firepower# Cluster disable is performing cleanup..done.
All data interfaces have been shutdown due to clustering being disabled. To recover either enable clustering or remove cluster group configuration.
Cluster unit unit-2-1 transitioned from [ ] to DISABLED
The 3DES/AES algorithms require a Encryption-3DES-AES activation key.

firepower# WARNING: dynamic routing is not supported on management interface when cluster interface-mode is 'spanned'. If dynamic routing is configured on any management interface, please remove it.
Cluster unit unit-3-1 transitioned from [ ]

```

同一故障的不同特征。在本例中，单元1-1也没有从单元3-1收到3条HC消息，一旦它收到新的keepalive，尝试使用一条杂散消息将单元3-1踢出，但消息从未到达单元3-1:

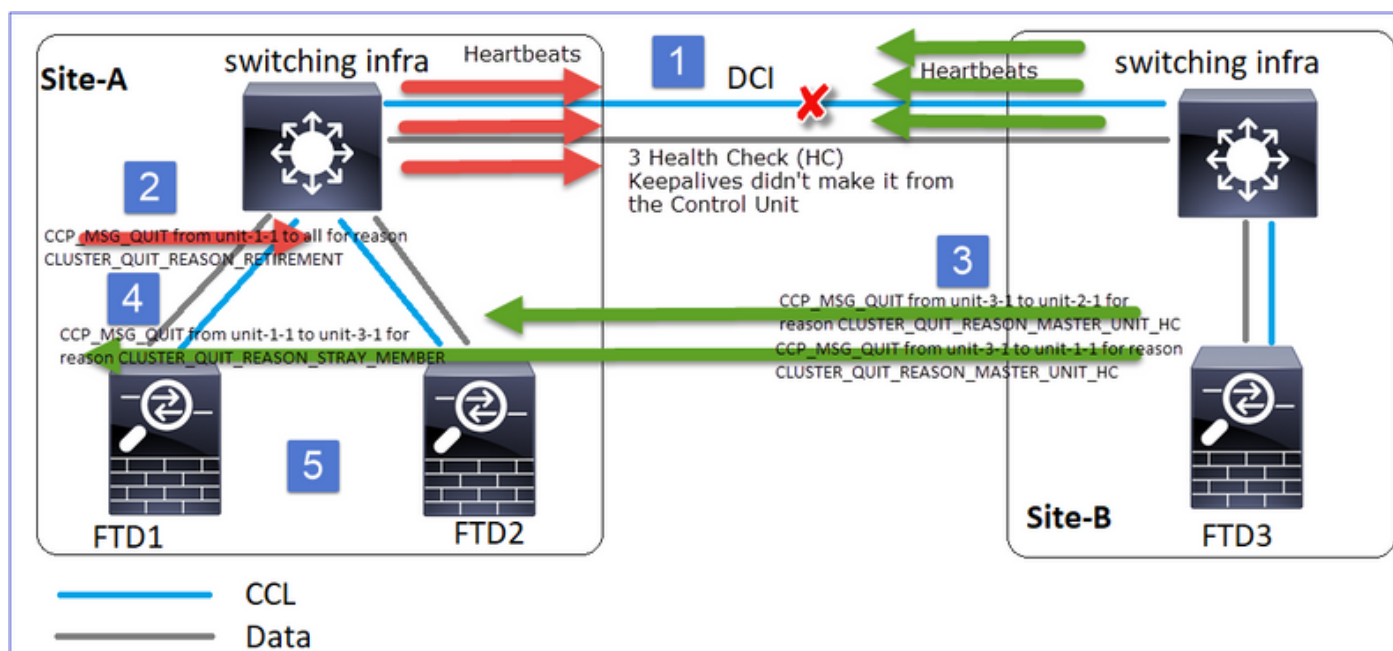
```

firepower# Asking slave unit unit-3-1 to quit because it failed unit health-check.
Forcing stray member unit-3-1 to leave the cluster.
Forcing stray member unit-3-1 to leave the cluster.
Cluster disable is performing cleanup..done.
All data interfaces have been shutdown due to clustering being disabled. To recover either enable clustering or remove cluster group configuration.
Cluster unit unit-1-1 transitioned from [ ] to DISABLED

firepower# Cluster disable is performing cleanup..done.
All data interfaces have been shutdown due to clustering being disabled. To recover either enable clustering or remove cluster group configuration.
Cluster unit unit-2-1 transitioned from [ ] to DISABLED
The 3DES/AES algorithms require a Encryption-3DES-AES activation key.

firepower# WARNING: dynamic routing is not supported on management interface when cluster interface-mode is 'spanned'. If dynamic routing is configured on any management interface, please remove it.
Cluster unit unit-3-1 transitioned from [ ]

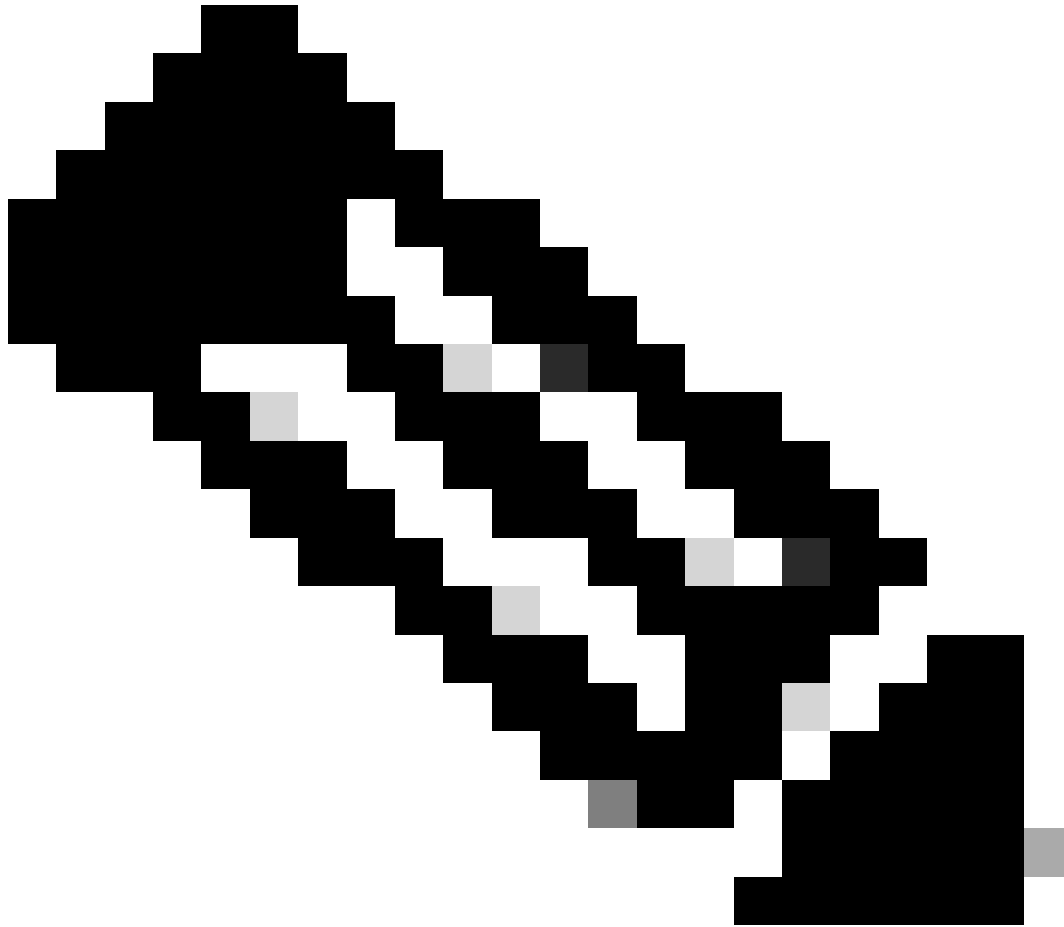
```



1. CCL在几秒钟内变成单向的。Unit-3-1不会从Unit-1-1接收3个HC消息，而是成为控制节点。
2. Unit-2-1发送CLUSTER_QUIT_REASON_RETIREMENT消息（广播）。
3. Unit-3-1向unit-2-1发送QUIT_REASON_PRIMARY_UNIT_HC消息。Unit-2-1接收该消息并退出集群。
4. Unit-3-1向unit-1-1发送QUIT_REASON_PRIMARY_UNIT_HC消息。Unit-1-1接收该消息并退

出集群。CCL恢复。

5. Units-1-1和2-1作为数据节点重新加入集群。



注意：如果在步骤5中CCL未恢复，则在site-A中，FTD1成为新的控制节点，在CCL恢复之后，它赢得新的选举。

设备1-1上的系统日志消息：

```
<#root>
```

```
firepower#
```

```
show log | include 747
```

```
Nov 03 2020 23:13:08: %FTD-7-747005: Clustering: State machine notify event CLUSTER_EVENT_MEMBER_STATE
Nov 03 2020 23:13:09: %FTD-4-747015: Clustering: Forcing stray member unit-3-1 to leave the cluster
Nov 03 2020 23:13:09: %FTD-7-747005: Clustering: State machine notify event CLUSTER_EVENT_MEMBER_STATE
Nov 03 2020 23:13:10: %FTD-4-747015: Clustering: Forcing stray member unit-3-1 to leave the cluster
Nov 03 2020 23:13:10: %FTD-6-747004: Clustering:
```

State machine changed from state PRIMARY to DISABLED

```
Nov 03 2020 23:13:12: %FTD-7-747006: Clustering: State machine is at state DISABLED
Nov 03 2020 23:13:12: %FTD-7-747005: Clustering: State machine notify event CLUSTER_EVENT_MY_STATE (sta
Nov 03 2020 23:13:18: %FTD-6-747004: Clustering: State machine changed from state ELECTION to ONCALL
```

设备1-1上的集群跟踪日志：

<#root>

firepower#

show cluster info trace | include QUIT

```
Nov 03 23:13:10.789 [DEBUG]Send CCP message to all: CCP_MSG_QUIT from unit-1-1 for reason CLUSTER_QUIT_R
Nov 03 23:13:10.769 [DEBUG]
```

Receive CCP message: CCP_MSG_QUIT from unit-3-1 to unit-1-1 for reason CLUSTER_QUIT_REASON_PRIMARY_UNIT

```
Nov 03 23:13:10.769 [DEBUG]Send CCP message to id 1: CCP_MSG_QUIT from unit-1-1 to unit-3-1 for reason C
Nov 03 23:13:09.789 [DEBUG]Receive CCP message: CCP_MSG_QUIT from unit-2-1 for reason CLUSTER_QUIT_REASO
Nov 03 23:13:09.769 [DEBUG]Send CCP message to id 1: CCP_MSG_QUIT from unit-1-1 to unit-3-1 for reason C
Nov 03 23:13:08.559 [DEBUG]Send CCP message to all: CCP_MSG_QUIT from unit-1-1 to unit-3-1 for reason CL
Nov 03 23:13:08.559 [DEBUG]Send CCP message to id 1: CCP_MSG_QUIT from unit-1-1 to unit-3-1 for reason C
```

Unit-3-1上的系统日志消息：

<#root>

firepower#

show log | include 747

```
Nov 03 2020 23:13:09: %FTD-7-747005: Clustering: State machine notify event CLUSTER_EVENT_MEMBER_STATE
Nov 03 2020 23:13:10: %FTD-7-747005: Clustering: State machine notify event CLUSTER_EVENT_MEMBER_STATE
Nov 03 2020 23:13:10: %FTD-6-747004: Clustering:
```

State machine changed from state SECONDARY to PRIMARY

```
Nov 03 2020 23:13:10: %FTD-6-747004: Clustering: State machine changed from state PRIMARY_FAST to PRIMA
Nov 03 2020 23:13:10: %FTD-6-747004: Clustering: State machine changed from state PRIMARY_DRAIN to PRIM
Nov 03 2020 23:13:10: %FTD-6-747004: Clustering: State machine changed from state PRIMARY_CONFIG to PRI
Nov 03 2020 23:13:10: %FTD-7-747006: Clustering: State machine is at state PRIMARY_POST_CONFIG
Nov 03 2020 23:13:10: %FTD-6-747004: Clustering: State machine changed from state PRIMARY_POST_CONFIG t
Nov 03 2020 23:13:10: %FTD-7-747006: Clustering:
```

State machine is at state PRIMARY

集群历史记录

Unit-1-1

<#root>

23:13:13 UTC Nov 3 2020

PRIMARY DISABLED Received control message DISABLE
(primary unit health check failure)

23:13:18 UTC Nov 3 2020

DISABLED ELECTION Enabled from CLI

23:13:18 UTC Nov 3 2020

ELECTION ONCALL Received cluster control message

23:13:23 UTC Nov 3 2020

ONCALL ELECTION Received cluster control message

...
23:14:48 UTC Nov 3 2020
ONCALL ELECTION Received cluster control message

23:14:48 UTC Nov 3 2020
ELECTION SECONDARY_COLD Received cluster control message

23:14:48 UTC Nov 3 2020
SECONDARY_COLD SECONDARY_APP_SYNC Client progression done

23:15:36 UTC Nov 3 2020
SECONDARY_APP_SYNC SECONDARY_CONFIG SECONDARY application configuration
sync done

23:15:48 UTC Nov 3 2020
SECONDARY_CONFIG SECONDARY_FILESYS Configuration replication finished

23:15:49 UTC Nov 3 2020
SECONDARY_FILESYS SECONDARY_BULK_SYNC Client progression done

23:16:13 UTC Nov 3 2020
SECONDARY_BULK_SYNC

SECONDARY

Client progression done

方案 5

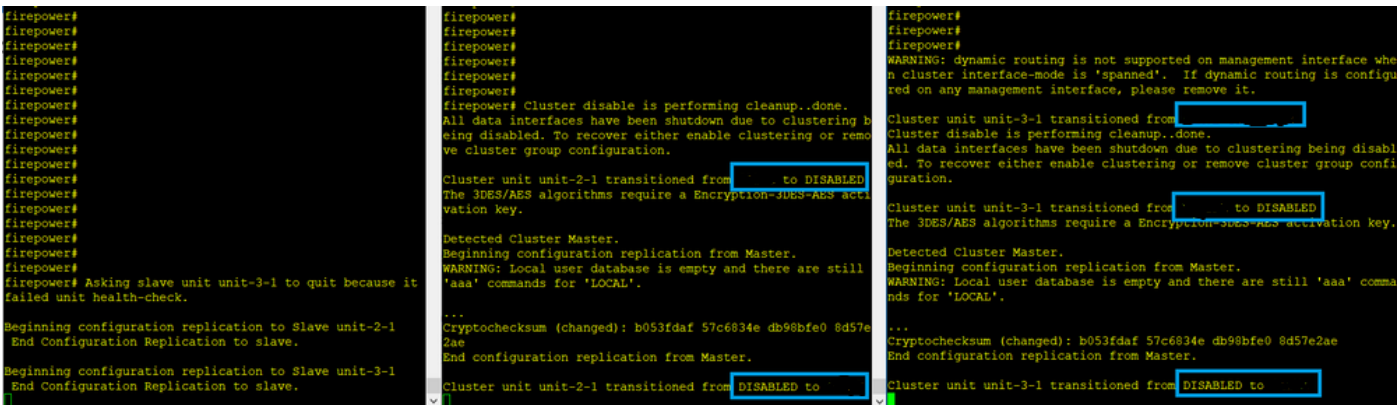
在失败之前

FTD1	FTD2	FTD3
站点A	站点A	站点B
控制节点	数据节点	数据节点

恢复后 (无更改)

FTD1	FTD2	FTD3
站点A	站点A	站点B
控制节点	数据节点	数据节点

失败



Unit-3-1向unit-1-1和unit-2-1发送了QUIT消息，但由于连接问题，只有unit-2-1接收了QUIT消息。

Unit-1-1集群跟踪日志：

```
<#root>
```

```
firepower#
```

```
show cluster info trace | include QUIT
```

```
Nov 04 00:52:10.429 [DEBUG]Receive CCP message: CCP_MSG_QUIT from unit-3-1 for reason CLUSTER_QUIT_REASON
Nov 04 00:51:47.059 [DEBUG]Receive CCP message: CCP_MSG_QUIT from unit-2-1 for reason CLUSTER_QUIT_REASON
Nov 04 00:51:45.429 [DEBUG]Send CCP message to all: CCP_MSG_QUIT from unit-1-1 to unit-3-1 for reason CL
Nov 04 00:51:45.429 [DEBUG]Send CCP message to unit-3-1(1): CCP_MSG_QUIT from unit-1-1 to unit-3-1 for r
```

Unit-2-1集群跟踪日志：

```
<#root>
```

```
firepower#
```

```
show cluster info trace | include QUIT
```

```
Nov 04 00:52:10.389 [DEBUG]Receive CCP message: CCP_MSG_QUIT from unit-3-1 for reason CLUSTER_QUIT_REASON
```

Nov 04 00:51:47.019 [DEBUG]Send CCP message to all: CCP_MSG_QUIT from unit-2-1 for reason CLUSTER_QUIT_R
Nov 04 00:51:46.999 [DEBUG]

Receive CCP message: CCP_MSG_QUIT from unit-3-1 to unit-2-1 for reason CLUSTER_QUIT_REASON_PRIMARY_UNIT

Nov 04 00:51:45.389 [DEBUG]Receive CCP message: CCP_MSG_QUIT from unit-1-1 to unit-3-1 for reason CLUSTER

集群历史记录

Unit-1-1	Unit-2-1
无事件	<pre><#root> 00:51:50 UTC Nov 4 2020 SECONDARY DISABLED Received control message DISABLE (primary unit health check failure) 00:51:54 UTC Nov 4 2020 DISABLED ELECTION Enabled from CLI 00:51:54 UTC Nov 4 2020 ELECTION SECONDARY_COLD Received cluster control message 00:51:54 UTC Nov 4 2020 SECONDARY_COLD SECONDARY_APP_SYNC Client progression done 00:52:42 UTC Nov 4 2020 SECONDARY_APP_SYNC SECONDARY_CONFIG SECONDARY application configura sync done 00:52:54 UTC Nov 4 2020 SECONDARY_CONFIG SECONDARY_FILESYS Configuration replication finis 00:52:55 UTC Nov 4 2020 SECONDARY_FILESYS SECONDARY_BULK_SYNC Client progression done 00:53:19 UTC Nov 4 2020 SECONDARY_BULK_SYNC SECONDARY Client progression done</pre>

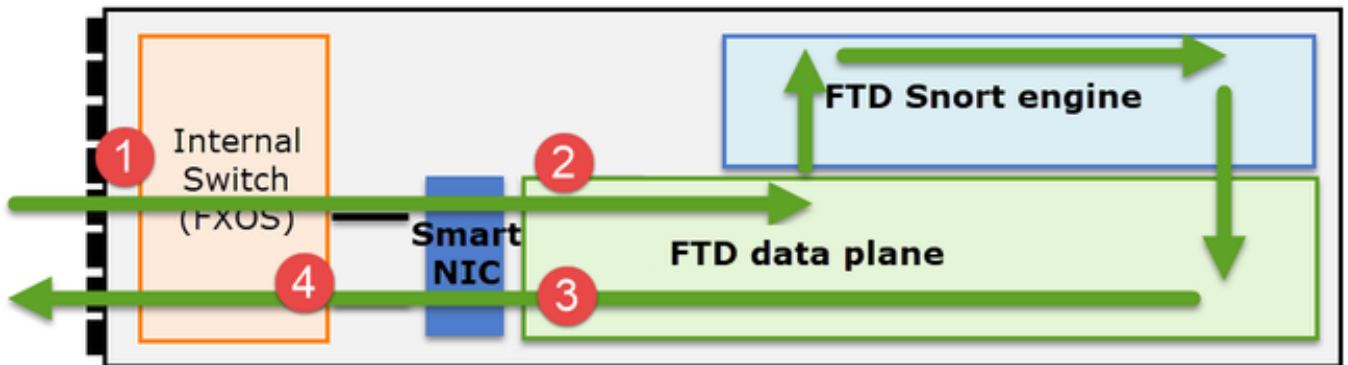
集群数据平面连接建立

NGFW捕获点

NGFW在以下方面提供捕获功能：

- 机箱内部交换机(FXOS)
- FTD数据平面引擎
- FTD Snort引擎

当您对集群上的数据路径问题进行故障排除时，大多数情况下使用的捕获点是FXOS和FTD数据平面引擎捕获。



1. 物理接口上的FXOS入口捕获
2. 数据平面引擎中的FTD入口捕获
3. 数据平面引擎中的FTD出口捕获
4. 背板接口上的FXOS入口捕获

有关NGFW捕获的其他详细信息，请查看以下文档：

集群设备流角色基础知识

可以通过多种方式通过集群建立连接，具体取决于以下因素：

- 流量类型（TCP、UDP等）
- 相邻交换机上配置的负载均衡算法
- 防火墙上配置的功能
- 网络条件（例如，IP分段、网络延迟等）

流角色	描述	标志
所有者	通常，最初接收连接的设备	UIO
导向器	处理转发器的所有者查找请求的设备。	Y

备份所有者	只要指挥交换机与所有者不是同一设备，则指挥交换机也是备用所有者。如果所有者选择自己作为指挥交换机，则会选择单独的备份所有者。	Y (如果指挥交换机也是备份所有者) y (如果指挥交换机不是备份所有者)
转发器	将数据包转发给所有者的设备	z
片段所有者	处理分段流量的设备	-
机箱备份	在机箱间集群中，当指挥交换机/备用交换机和所有者数据流均由同一机箱的单元拥有时，另一个机箱中的一个单元成为辅助备份/指挥交换机。 此角色特定于具有1个以上刀片的 Firepower 9300系列的机箱间集群。	w

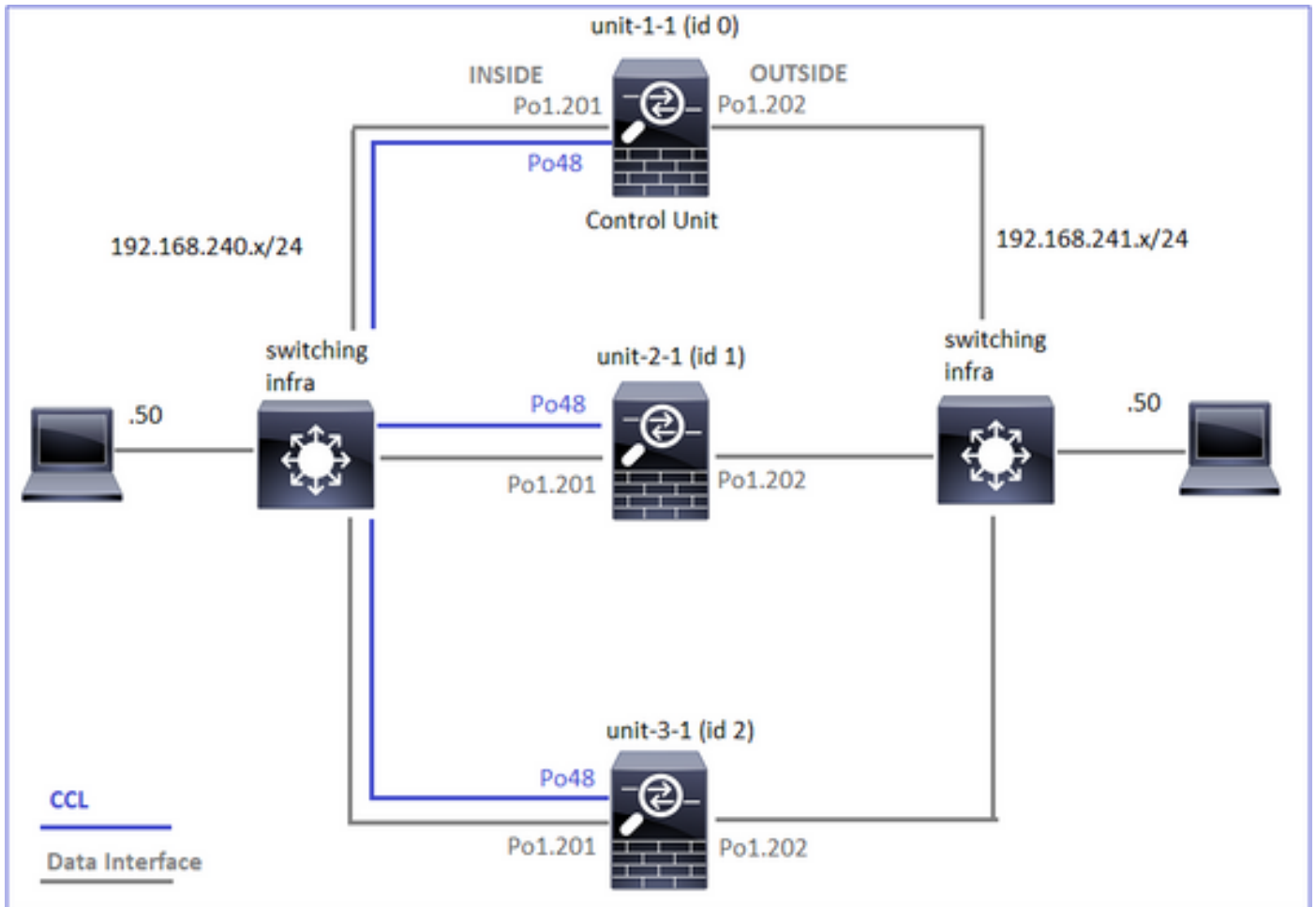
- 有关其他详细信息，请查看《配置指南》中的相关章节（参见相关信息中的链接）
- 在特定情况下（请参阅“案例研究”部分），某些标志并非始终显示。

集群连接建立案例研究

下一节将介绍各种案例研究，这些研究演示了通过群集建立连接的一些方法。其目标是：

- 熟悉不同的设备角色。
- 演示如何关联各种命令输出。

拓扑




集群设备和ID:

Unit-1-1	Unit-2-1
<pre> <#root> Cluster GROUP1: On Interface mode: spanned This is "unit-1-1" in state PRIMARY ID : 0 Site ID : 1 Version : 9.15(1) Serial No.: FCH22247LNK CCL IP : 10.17.1.1 CCL MAC : 0015.c500.018f Last join : 02:24:43 UTC Nov 27 2020 Last leave: N/A </pre>	<pre> <#root> Unit "unit-2-1" in state SECO ID : 1 Site ID : 1 Version : 9.15(1) Serial No.: FCH23157Y9N CCL IP : 10.17.2.1 CCL MAC : 0015.c500.02 Last join : 02:04:19 UTC Last leave: N/A </pre>

已启用群集捕获：

```
cluster exec cap CAPI int INSIDE buffer 33554432 match tcp host 192.168.240.50 host 192.168.241.50 eq 80
cluster exec cap CAPO int OUTSIDE buffer 33554432 match tcp host 192.168.240.50 host 192.168.241.50 eq 80
cluster exec cap CAPI_RH reinject-hide int INSIDE buffer 33554432 match tcp host 192.168.240.50 host 192.168.241.50 eq 80
cluster exec cap CAPO_RH reinject-hide int OUTSIDE buffer 33554432 match tcp host 192.168.240.50 host 192.168.241.50 eq 80
cluster exec cap CCL int cluster buffer 33554432
```

 注意：这些测试在实验室环境中运行，通过集群的流量最小。在生产中，尽量使用特定的捕获过滤器（例如，目标端口和尽可能使用源端口）来最小化捕获中的“噪声”。

案例研究1.对称流量（所有者也是主管）

观察1.重新隐藏(reject-hide)捕获仅显示unit-1-1上的数据包。这意味着两个方向的流都经过unit-1-1（对称流量）：

```
<#root>
```

```
firepower#
```

```
cluster exec show cap
```

```
unit-1-1(LOCAL):*****
capture CCL type raw-data interface cluster [Capturing - 33513 bytes]
capture CAPI type raw-data buffer 33554432 trace interface INSIDE [Buffer Full - 33553914 bytes]
match tcp host 192.168.240.50 host 192.168.241.50 eq 80
capture CAPO type raw-data buffer 33554432 trace interface OUTSIDE [Buffer Full - 33553914 bytes]
match tcp host 192.168.240.50 host 192.168.241.50 eq 80
capture CAPI_RH type raw-data

reinject-hide

  buffer 33554432 interface INSIDE [Buffer Full] -
33553914 bytes
]
match tcp host 192.168.240.50 host 192.168.241.50 eq 80
capture CAPO_RH type raw-data

reinject-hide

  buffer 33554432 interface OUTSIDE [Buffer Full] -
33553914 bytes
]
match tcp host 192.168.240.50 host 192.168.241.50 eq 80

unit-2-1:*****
capture CCL type raw-data interface cluster [Capturing - 23245 bytes]
capture CAPI type raw-data buffer 33554432 trace interface INSIDE [Capturing - 0 bytes]
match tcp host 192.168.240.50 host 192.168.241.50 eq 80
capture CAPO type raw-data buffer 33554432 trace interface OUTSIDE [Capturing - 0 bytes]
match tcp host 192.168.240.50 host 192.168.241.50 eq 80
```

```

capture CAPI_RH type raw-data
reinject-hide

  buffer 33554432 interface INSIDE [Capturing -
0 bytes
]
match tcp host 192.168.240.50 host 192.168.241.50 eq 80
capture CAPO_RH type raw-data
reinject-hide

  buffer 33554432 interface OUTSIDE [Capturing -
0 bytes
]
match tcp host 192.168.240.50 host 192.168.241.50 eq 80

unit-3-1:*****
capture CCL type raw-data interface cluster [Capturing - 24815 bytes]
capture CAPI type raw-data buffer 33554432 trace interface INSIDE [Capturing - 0 bytes]
match tcp host 192.168.240.50 host 192.168.241.50 eq 80
capture CAPO type raw-data buffer 33554432 trace interface OUTSIDE [Capturing - 0 bytes]
match tcp host 192.168.240.50 host 192.168.241.50 eq 80
capture CAPI_RH type raw-data
reinject-hide

  buffer 33554432 interface INSIDE [Capturing -
0 bytes
]
match tcp host 192.168.240.50 host 192.168.241.50 eq 80
capture CAPO_RH type raw-data
reinject-hide

  buffer 33554432 interface OUTSIDE [Capturing -
0 bytes
]
match tcp host 192.168.240.50 host 192.168.241.50 eq 80

```

观察2.具有源端口45954的流的连接标志分析

```
<#root>
```

```
firepower#
```

```
cluster exec show conn
```

```

unit-1-1(LOCAL):*****
22 in use, 25 most used
Cluster:
fwd connections: 0 in use, 1 most used
dir connections: 0 in use, 122 most used
centralized connections: 0 in use, 0 most used

```

```

VPN redirect connections: 0 in use, 0 most used
Inspect Snort:
preserve-connection: 1 enabled, 0 in effect, 2 most enabled, 1 most in effect

TCP OUTSIDE 192.168.241.50:80 INSIDE 192.168.240.50:
45954
, idle 0:00:00, bytes 487413076,
flags UIO N1

unit-2-1:*****
22 in use, 271 most used
Cluster:
fwd connections: 0 in use, 2 most used
dir connections: 0 in use, 2 most used
centralized connections: 0 in use, 0 most used
VPN redirect connections: 0 in use, 0 most used
Inspect Snort:
preserve-connection: 1 enabled, 0 in effect, 249 most enabled, 0 most in effect

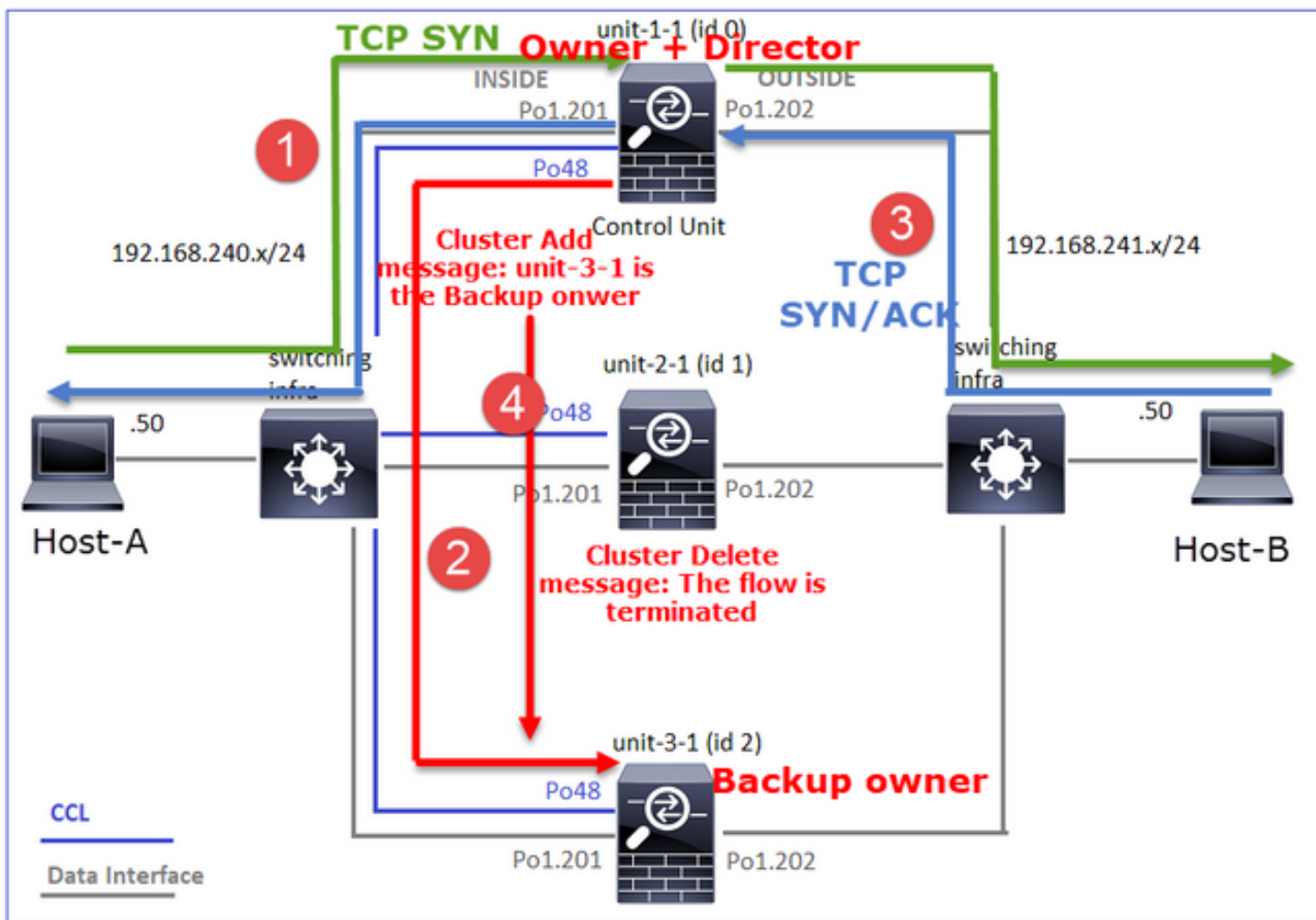
unit-3-1:*****
17 in use, 20 most used
Cluster:
fwd connections: 1 in use, 2 most used
dir connections: 1 in use, 127 most used
centralized connections: 0 in use, 0 most used
VPN redirect connections: 0 in use, 0 most used
Inspect Snort:
preserve-connection: 0 enabled, 0 in effect, 1 most enabled, 0 most in effect

TCP OUTSIDE 192.168.241.50:443 NP Identity Ifc 192.168.240.50:39698, idle 0:00:23, bytes 0, flags z
TCP OUTSIDE 192.168.241.50:80 INSIDE 192.168.240.50:
45954
, idle 0:00:06, bytes 0,
flags y

```

单元	标志	备注
Unit-1-1	UIO	·流所有者 — 设备处理流 ·控制器 — 由于unit-3-1具有“y”而不是“Y”，这意味着已选择unit-1-1作为此流的控制器。因此，由于它也是所有者，因此选择了另一设备（本例中为unit-3-1）作为备用所有者
Unit-2-1	-	-
Unit-3-1	y	设备是备份所有者

上述内容可以图形表示为：



1. TCP SYN数据包从主机A到达设备1-1。设备1-1成为流所有者。
2. Unit-1-1也被选为流导向器。因此，它还会选择unit-3-1作为备份所有者（cluster add消息）。
3. TCP SYN/ACK数据包从主机B到达设备3-1。流量是对称的。
4. 连接终止后，所有者会发送集群删除消息，以从备份所有者中删除流信息。

观察3.带跟踪的捕获显示两个方向只能通过unit-1-1。

步骤1.根据源端口，确定所有集群单元中关注的流和数据包：

```
<#root>
```

```
firepower#
```

```
cluster exec show capture CAPI | i 45954
```

```
unit-1-1(LOCAL):*****
```

```
1: 08:42:09.362697 802.1Q vlan#201 PO 192.168.240.50.45954 > 192.168.241.50.80: S 992089269:992089269(0
```

```
2: 08:42:09.363521 802.1Q vlan#201 PO 192.168.241.50.80 > 192.168.240.50.45954: S 4042762409:4042762409
```

```
3: 08:42:09.363827 802.1Q vlan#201 PO 192.168.240.50.45954 > 192.168.241.50.80: . ack 4042762410 win 22
```

```
...
```

```
unit-2-1:*****
```

```
unit-3-1:*****
```

<#root>

firepower#

cluster exec show capture CAPO | i 45954

unit-1-1(LOCAL):*****

1: 08:42:09.362987 802.1Q vlan#202 PO 192.168.240.50.45954 > 192.168.241.50.80: S 2732339016:2732339016

2: 08:42:09.363415 802.1Q vlan#202 PO 192.168.241.50.80 > 192.168.240.50.45954: S 3603655982:3603655982

3: 08:42:09.363903 802.1Q vlan#202 PO 192.168.240.50.45954 > 192.168.241.50.80: . ack 3603655983 win 22

...

unit-2-1:*****

unit-3-1:*****

步骤2.由于这是TCP流跟踪，因此三次握手数据包都会被跟踪。在此输出中可以看到，unit-1-1是所有者。为简单起见，省略非相关的跟踪阶段：

<#root>

firepower#

show cap CAPI packet-number 1 trace

25985 packets captured

1: 08:42:09.362697 802.1Q vlan#201 PO 192.168.240.50.

45954

> 192.168.241.50.80:

S

992089269:992089269(0) win 29200 <mss 1460,sackOK,timestamp 495153655 0,nop,wscale 7>

...

Phase: 4

Type: CLUSTER-EVENT

Subtype:

Result: ALLOW

Config:

Additional Information:

Input interface: 'INSIDE'

Flow type: NO FLOW

I (0) got initial, attempting ownership.

Phase: 5

Type: CLUSTER-EVENT

Subtype:

Result: ALLOW

Config:

Additional Information:

Input interface: 'INSIDE'

Flow type: NO FLOW

I (0) am becoming owner

...

返回流量(TCP SYN/ACK):

<#root>

firepower#

show capture CAPO packet-number 2 trace

25985 packets captured

2: 08:42:09.363415 802.1Q vlan#202 P0 192.168.241.50.80 > 192.168.240.50.45954:

S

3603655982:3603655982(0)

ack

2732339017 win 28960 <mss 1460,sackOK,timestamp 505509125 495153655,nop,wscale 7>

...

Phase: 3

Type: FLOW-LOOKUP

Subtype:

Result: ALLOW

Config:

Additional Information:

Found flow with id 9364, using existing flow

观察4. FTD数据平面系统日志显示所有设备上的连接创建和终止：

<#root>

firepower#

```
cluster exec show log | include 45954
```

```
unit-1-1
```

```
(LOCAL):*****
```

```
Dec 01 2020 08:42:09: %FTD-6-302013:
```

```
Built inbound TCP connection 9364
```

```
for INSIDE:192.168.240.50/45954 (192.168.240.50/45954) to OUTSIDE:192.168.241.50/80 (192.168.241.50/80)
```

```
Dec 01 2020 08:42:18: %FTD-6-302014:
```

```
Teardown TCP connection 9364
```

```
for INSIDE:192.168.240.50/45954 to OUTSIDE:192.168.241.50/80 duration 0:00:08 bytes 1024000440 TCP FIN
```

```
unit-2-1:*****
```

```
unit-3-1
```

```
:*****
```

```
Dec 01 2020 08:42:09: %FTD-6-302022:
```

```
Built backup stub TCP connection
```

```
for INSIDE:192.168.240.50/45954 (192.168.240.50/45954) to OUTSIDE:192.168.241.50/80 (192.168.241.50/80)
```

```
Dec 01 2020 08:42:18: %FTD-6-302023:
```

```
Teardown backup TCP connection
```

```
for INSIDE:192.168.240.50/45954 to OUTSIDE:192.168.241.50/80 duration 0:00:08 forwarded bytes 0 Cluste
```

案例研究2.对称流量 (所有者与指挥交换机不同)

- 与案例研究相#1，但在本案例研究中，流所有者与指挥交换机是不同的单位。
- 所有输出均与案例分析#1相似。与案例分析#1的主要区别是替代方案1的“y”标志的“Y”标志。

意见1.业主与所长不同。

源端口为46278的流的连接标志分析。

```
<#root>
```

```
firepower#
```

```
cluster exec show conn
```

```
unit-1-1(LOCAL):*****
```

```
23 in use, 25 most used
```

```
Cluster:
```

```
fwd connections: 0 in use, 1 most used
```

```
dir connections: 0 in use, 122 most used
```

```
centralized connections: 0 in use, 0 most used
```

```
VPN redirect connections: 0 in use, 0 most used
```

```
Inspect Snort:
```

preserve-connection: 2 enabled, 0 in effect, 4 most enabled, 1 most in effect

TCP OUTSIDE 192.168.241.50:80 INSIDE 192.168.240.50:

46278

, idle 0:00:00, bytes 508848268, flags

UIO N1

TCP OUTSIDE 192.168.241.50:80 INSIDE 192.168.240.50:46276, idle 0:00:03, bytes 0, flags aA N1

unit-2-1:*****

21 in use, 271 most used

Cluster:

fwd connections: 0 in use, 2 most used

dir connections: 0 in use, 2 most used

centralized connections: 0 in use, 0 most used

VPN redirect connections: 0 in use, 0 most used

Inspect Snort:

preserve-connection: 0 enabled, 0 in effect, 249 most enabled, 0 most in effect

unit-3-1:*****

17 in use, 20 most used

Cluster:

fwd connections: 1 in use, 5 most used

dir connections: 1 in use, 127 most used

centralized connections: 0 in use, 0 most used

VPN redirect connections: 0 in use, 0 most used

Inspect Snort:

preserve-connection: 0 enabled, 0 in effect, 1 most enabled, 0 most in effect

TCP OUTSIDE 192.168.241.50:80 NP Identity Ifc 192.168.240.50:46276, idle 0:00:02, bytes 0, flags z

TCP OUTSIDE 192.168.241.50:80 INSIDE 192.168.240.50:

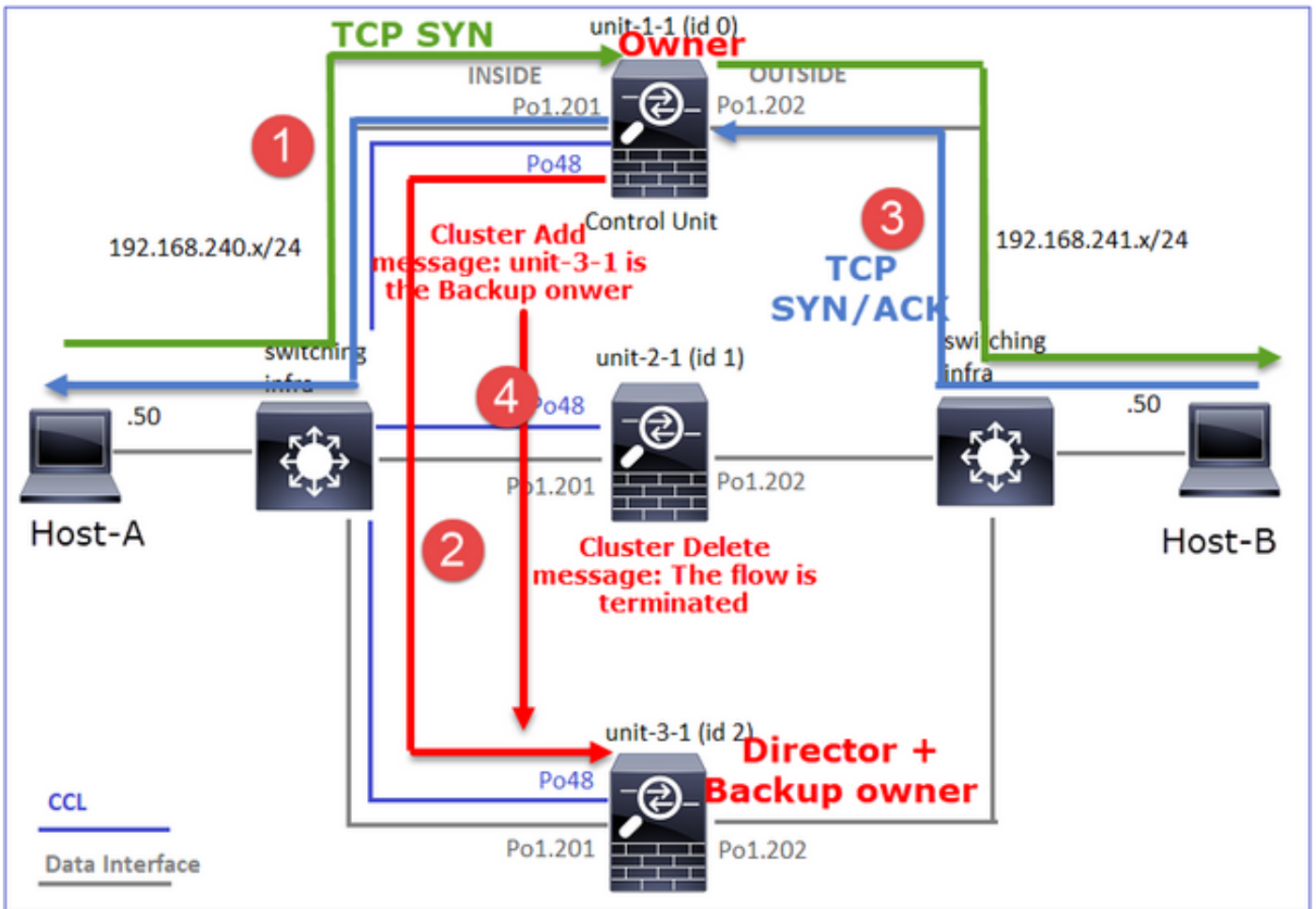
46278

, idle 0:00:06, bytes 0,

flags Y

单元	标志	备注
Unit-1-1	UIO	·流所有者 — 设备处理流
Unit-2-1	-	-
Unit-3-1	Y	·指挥交换机和备用设备所有者 — 第3-1单元具有标志Y (指挥交换机)。

上述内容可以图形表示为：



1. TCP SYN数据包从主机A到达设备1-1。设备1-1成为流所有者。
2. Unit-3-1被选为流导向器。Unit-3-1也是备份所有者（通过CCL的UDP 4193上的“cluster add”消息）。
3. TCP SYN/ACK数据包从主机B到达设备3-1。流量是对称的。
4. 连接终止后，所有者通过CCL发送UDP 4193上的“cluster delete”消息，以删除备份所有者中的流信息。

观察2.用trace捕获显示两个方向只通过unit-1-1

步骤1.使用与案例研究1相同的方法，根据源端口识别所有集群单元中的相关流和数据包：

```
<#root>
```

```
firepower#
```

```
cluster exec show cap CAPI | include 46278
```

```
unit-1-1
```

```
(LOCAL):*****
```

```
3: 11:01:44.841631 802.1Q vlan#201 P0 192.168.240.50.46278 > 192.168.241.50.80:
```

```
s
```

```
1972783998:1972783998(0) win 29200 <mss 1460,sackOK,timestamp 503529072 0,nop,wscale 7>
```

4: 11:01:44.842317 802.1Q vlan#201 PO 192.168.241.50.80 > 192.168.240.50.46278:

s

3524167695:3524167695(0)

ack

1972783999 win 28960 <mss 1380,sackOK,timestamp 513884542 503529072,nop,wscale 7>

5: 11:01:44.842592 802.1Q vlan#201 PO 192.168.240.50.46278 > 192.168.241.50.80: . ack 3524167696 win 22

unit-2-1:*****

unit-3-1:*****
firepower#

在OUTSIDE接口上捕获：

<#root>

firepower#

cluster exec show cap CAPO | include 46278

unit-1-1

(LOCAL):*****

3: 11:01:44.841921 802.1Q vlan#202 PO 192.168.240.50.46278 > 192.168.241.50.80:

s

2153055699:2153055699(0) win 29200 <mss 1380,sackOK,timestamp 503529072 0,nop,wscale 7>

4: 11:01:44.842226 802.1Q vlan#202 PO 192.168.241.50.80 > 192.168.240.50.46278:

s

3382481337:3382481337(0)

ack

2153055700 win 28960 <mss 1460,sackOK,timestamp 513884542 503529072,nop,wscale 7>

5: 11:01:44.842638 802.1Q vlan#202 PO 192.168.240.50.46278 > 192.168.241.50.80: . ack 3382481338 win 22

unit-2-1:*****

unit-3-1:*****
firepower#

步骤2.重点关注入口数据包 (TCP SYN和TCP SYN/ACK)：

<#root>

firepower#

```
cluster exec show cap CAPI packet-number 3 trace
```

```
unit-1-1(LOCAL):*****
```

```
824 packets captured
```

```
3: 11:01:44.841631 802.1Q vlan#201 P0 192.168.240.50.46278 > 192.168.241.50.80:
```

```
S
```

```
1972783998:1972783998(0) win 29200 <mss 1460,sackOK,timestamp 503529072 0,nop,wscale 7>
```

```
...
```

```
Phase: 4
```

```
Type: CLUSTER-EVENT
```

```
Subtype:
```

```
Result: ALLOW
```

```
Config:
```

```
Additional Information:
```

```
Input interface: 'INSIDE'
```

```
Flow type: NO FLOW
```

```
I (0) got initial, attempting ownership.
```

```
Phase: 5
```

```
Type: CLUSTER-EVENT
```

```
Subtype:
```

```
Result: ALLOW
```

```
Config:
```

```
Additional Information:
```

```
Input interface: 'INSIDE'
```

```
Flow type: NO FLOW
```

```
I (0) am becoming owner
```

跟踪设备1-1上的SYN/ACK:

```
<#root>
```

```
firepower#
```

```
cluster exec show cap CAPO packet-number 4 trace
```

```
unit-1-1(LOCAL):*****
```

```
4: 11:01:44.842226 802.1Q vlan#202 P0 192.168.241.50.80 > 192.168.240.50.
```

```
46278
```

```
:
```

```
S
```

```
3382481337:3382481337(0)
```

```
ack
```

```
2153055700 win 28960 <mss 1460,sackOK,timestamp 513884542 503529072,nop,wscale 7>  
Phase: 3  
Type: FLOW-LOOKUP  
Subtype:  
Result: ALLOW  
Config:  
Additional Information:
```

```
Found flow with id 9583, using existing flow
```

观察3. FTD数据平面系统日志显示所有者和备份所有者的连接创建和终止：

```
<#root>
```

```
firepower#
```

```
cluster exec show log | include 46278
```

```
unit-1-1(LOCAL):*****
```

```
Dec 01 2020 11:01:44: %FTD-6-302013:
```

```
Built inbound TCP connection
```

```
9583 for INSIDE:192.168.240.50/46278 (192.168.240.50/46278) to OUTSIDE:192.168.241.50/80 (192.168.241.50/80)
```

```
Dec 01 2020 11:01:53: %FTD-6-302014:
```

```
Teardown TCP connection
```

```
9583 for INSIDE:192.168.240.50/46278 to OUTSIDE:192.168.241.50/80 duration 0:00:08 bytes 1024001808 TC
```

```
unit-2-1:*****
```

```
unit-3-1:*****
```

```
Dec 01 2020 11:01:44: %FTD-6-302022:
```

```
Built director stub TCP connection
```

```
for INSIDE:192.168.240.50/46278 (192.168.240.50/46278) to OUTSIDE:192.168.241.50/80 (192.168.241.50/80)
```

```
Dec 01 2020 11:01:53: %FTD-6-302023:
```

```
Teardown director TCP connection
```

```
for INSIDE:192.168.240.50/46278 to OUTSIDE:192.168.241.50/80 duration 0:00:08 forwarded bytes 0 Cluste
```

案例研究3.非对称流量 (指挥交换机转发流量)。

观察1. reinject-hide捕获显示unit-1-1和unit-2-1 (非对称流) 上的数据包：

```
<#root>
```

```
firepower#
```

cluster exec show cap

```
unit-1-1(LOCAL):*****
capture CCL type raw-data buffer 33554432 interface cluster [Buffer Full - 33554320 bytes]
capture CAPI type raw-data buffer 100000 trace interface INSIDE [Buffer Full - 98552 bytes]
match tcp host 192.168.240.50 host 192.168.241.50 eq www
capture CAPO type raw-data buffer 100000 trace interface OUTSIDE [Buffer Full - 98552 bytes]
match tcp host 192.168.240.50 host 192.168.241.50 eq www
capture CAPI_RH type raw-data
```

reinject-hide

buffer 100000 interface

INSIDE

[Buffer Full -

98552 bytes

]

match tcp host 192.168.240.50 host 192.168.241.50 eq www

capture CAPI_RH type raw-data

reinject-hide

buffer 100000 interface

OUTSIDE

[Buffer Full -

99932 bytes

]

match tcp host 192.168.240.50 host 192.168.241.50 eq www

```
unit-2-1:*****
```

capture CCL type raw-data buffer 33554432 interface cluster [Buffer Full - 33553268 bytes]

capture CAPI type raw-data buffer 100000 trace interface INSIDE [Capturing - 0 bytes]

match tcp host 192.168.240.50 host 192.168.241.50 eq www

capture CAPO type raw-data buffer 100000 trace interface OUTSIDE [Buffer Full - 99052 bytes]

match tcp host 192.168.240.50 host 192.168.241.50 eq www

capture CAPI_RH type raw-data reinject-hide buffer 100000 interface INSIDE [Capturing - 0 bytes]

match tcp host 192.168.240.50 host 192.168.241.50 eq www

capture CAPO_RH type raw-data

reinject-hide

buffer 100000 interface

OUTSIDE

[Buffer Full -

99052 bytes

]

match tcp host 192.168.240.50 host 192.168.241.50 eq www

```
unit-3-1:*****
```

capture CCL type raw-data buffer 33554432 interface cluster [Capturing - 53815 bytes]

capture CAPI type raw-data buffer 100000 trace interface INSIDE [Capturing - 0 bytes]

match tcp host 192.168.240.50 host 192.168.241.50 eq www

capture CAPO type raw-data buffer 100000 trace interface OUTSIDE [Capturing - 658 bytes]

match tcp host 192.168.240.50 host 192.168.241.50 eq www


```
capture CAPI_RH type raw-data reinject-hide buffer 100000 interface INSIDE [Capturing - 0 bytes]
match tcp host 192.168.240.50 host 192.168.241.50 eq www
capture CAPO_RH type raw-data reinject-hide buffer 100000 interface OUTSIDE [Capturing - 658 bytes]
match tcp host 192.168.240.50 host 192.168.241.50 eq www
```

观察2.与源端口46502的流的连接标志分析。

```
<#root>
```

```
firepower#
```

```
cluster exec show conn
```

```
unit-1-1
```

```
(LOCAL):*****
```

```
23 in use, 25 most used
```

```
Cluster:
```

```
fwd connections: 0 in use, 1 most used
```

```
dir connections: 0 in use, 122 most used
```

```
centralized connections: 0 in use, 0 most used
```

```
VPN redirect connections: 0 in use, 0 most used
```

```
Inspect Snort:
```

```
preserve-connection: 2 enabled, 0 in effect, 4 most enabled, 1 most in effect
```

```
TCP OUTSIDE 192.168.241.50:80 INSIDE 192.168.240.50:
```

```
46502
```

```
, idle 0:00:00, bytes 448760236,
```

```
flags UIO N1
```

```
TCP OUTSIDE 192.168.241.50:80 INSIDE 192.168.240.50:46500, idle 0:00:06, bytes 0, flags aA N1
```

```
unit-2-1
```

```
:*****
```

```
21 in use, 271 most used
```

```
Cluster:
```

```
fwd connections: 0 in use, 2 most used
```

```
dir connections: 1 in use, 2 most used
```

```
centralized connections: 0 in use, 0 most used
```

```
VPN redirect connections: 0 in use, 0 most used
```

```
Inspect Snort:
```

```
preserve-connection: 0 enabled, 0 in effect, 249 most enabled, 0 most in effect
```

```
TCP OUTSIDE 192.168.241.50:80 INSIDE 192.168.240.50:
```

```
46502
```

```
, idle 0:00:00, bytes 0,
```

```
flags Y
```

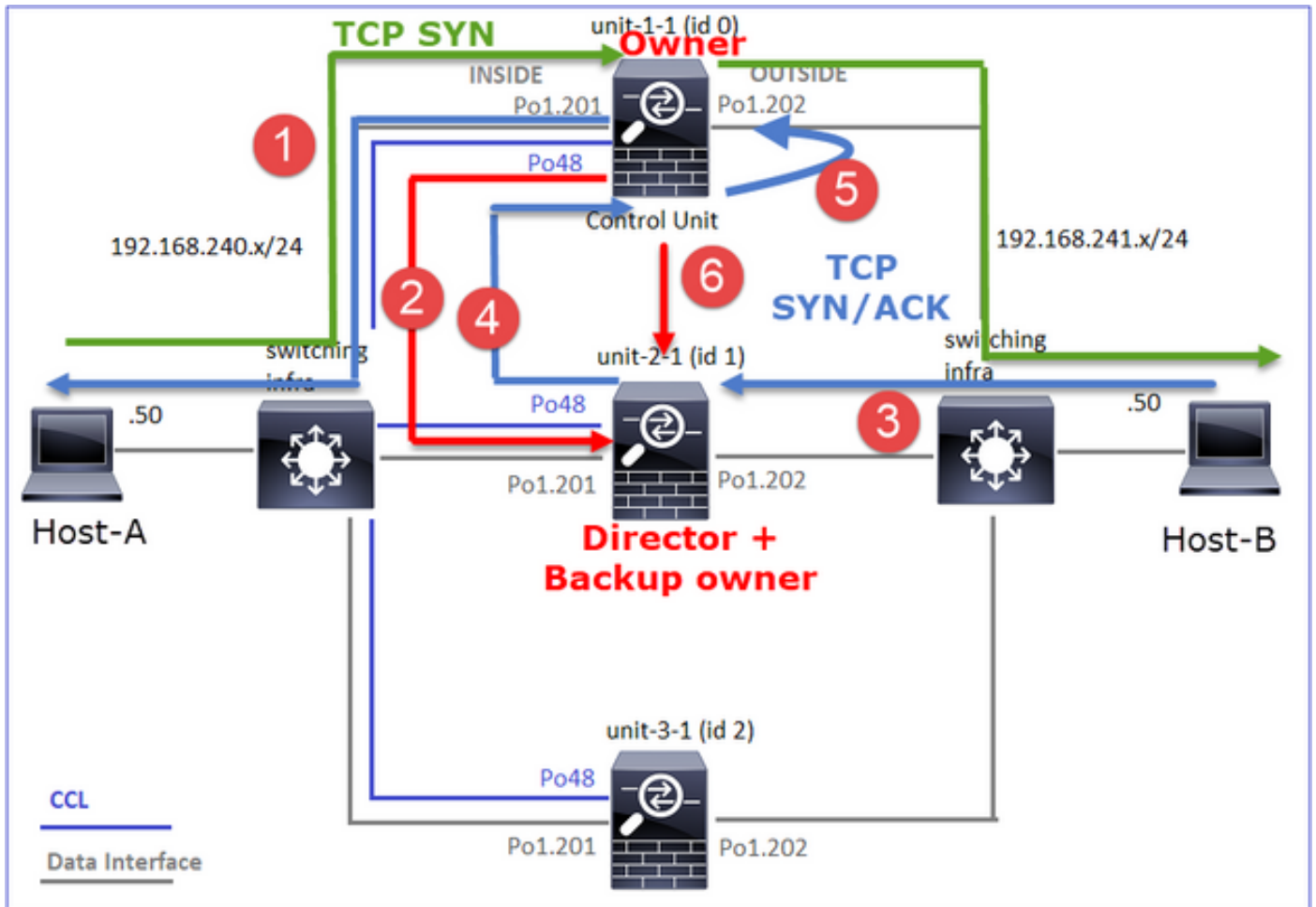
```

unit-3-1:*****
17 in use, 20 most used
Cluster:
fwd connections: 1 in use, 5 most used
dir connections: 0 in use, 127 most used
centralized connections: 0 in use, 0 most used
VPN redirect connections: 0 in use, 0 most used
Inspect Snort:
preserve-connection: 0 enabled, 0 in effect, 1 most enabled, 0 most in effect

```

单元	标志	备注
Unit-1-1	UIO	·流所有者 — 设备处理流。
Unit-2-1	Y	<p>·导向器 — 由于unit-2-1具有“Y”标志，这意味着已选择unit-2-1作为此流的导向器。</p> <p>·备份所有者</p> <p>·最后，虽然从该输出中看不出来，但从show capture和show log输出中，很明显的unit-2-1会将此流转发给所有者（尽管在本场景中技术上它不被视为转发器）。</p> <p>注意：设备不能同时是导向器（Y流）和转发器（z流），这2个角色是互斥的。控制器（Y流）仍然可以转发流量。请参阅本案例研究后面的show log输出。</p>
Unit-3-1	-	-

上述内容可以图形表示为：



1. TCP SYN数据包从主机A到达设备1-1。设备1-1成为流所有者。
2. Unit-2-1被选为流指挥和备份所有者。流所有者在UDP 4193上发送“cluster add”单播消息以通知备份所有者有关流的消息。
3. TCP SYN/ACK数据包从主机B到达设备2-1。流量不对称。
4. Unit-2-1通过CCL将数据包转发给所有者（由于TCP SYN Cookie）。
5. 所有者重新在接口OUTSIDE上注入数据包，然后将数据包转发到主机A。
6. 连接终止后，所有者会发送集群删除消息，以从备份所有者中删除流信息。

观察3.使用trace捕获显示非对称流量以及从unit-2-1到unit-1-1的重定向。

步骤1.识别属于关注流(端口46502)的数据包：

```
<#root>
```

```
firepower#
```

```
cluster exec show capture CAPI | include 46502
```

```
unit-1-1(LOCAL):*****
```

```
3: 12:58:33.356121 802.1Q vlan#201 PO 192.168.240.50.46502 > 192.168.241.50.80: S 4124514680:4124514680
```

```
4: 12:58:33.357037 802.1Q vlan#201 PO 192.168.241.50.80 > 192.168.240.50.46502: S 883000451:883000451(0
```

```
5: 12:58:33.357357 802.1Q vlan#201 PO 192.168.240.50.46502 > 192.168.241.50.80: . ack 883000452 win 229
```

```
unit-2-1:*****
```

```
unit-3-1:*****
```

返回方向：

```
<#root>
```

```
firepower#
```

```
cluster exec show capture CAPO | include 46502
```

```
unit-1-1(LOCAL):*****
```

```
3: 12:58:33.356426 802.1Q vlan#202 PO 192.168.240.50.46502 > 192.168.241.50.80: S 1434968587:1434968587
```

```
4: 12:58:33.356915 802.1Q vlan#202 PO 192.168.241.50.80 > 192.168.240.50.46502: S 4257314722:4257314722
```

```
5: 12:58:33.357403 802.1Q vlan#202 PO 192.168.240.50.46502 > 192.168.241.50.80: . ack 4257314723 win 22
```

```
unit-2-1:*****
```

```
1: 12:58:33.359249 802.1Q vlan#202 PO 192.168.241.50.80 > 192.168.240.50.46502: S 4257314722:4257314722
```

```
2: 12:58:33.360302 802.1Q vlan#202 PO 192.168.241.50.80 > 192.168.240.50.46502: . ack 1434968736 win 23
```

```
3: 12:58:33.361004 802.1Q vlan#202 PO 192.168.241.50.80 > 192.168.240.50.46502: . 4257314723:4257316091
```

```
...
```

```
unit-3-1:*****
```

步骤2.跟踪数据包。默认情况下，仅跟踪前50个入口数据包。为简单起见，省略无关的跟踪相位。

Unit-1-1 (所有者)：

```
<#root>
```

```
firepower#
```

```
cluster exec show capture CAPI packet-number 3 trace
```

```
unit-1-1(LOCAL):*****
```

```
3: 12:58:33.356121 802.1Q vlan#201 PO 192.168.240.50.
```

```
46502
```

```
> 192.168.241.50.80:
```

```
S
```

```
4124514680:4124514680(0) win 29200 <mss 1460,sackOK,timestamp 510537534 0,nop,wscale 7>
```

```
...
```

```
Phase: 4
```

```
Type: CLUSTER-EVENT
```

```
Subtype:
```

```
Result: ALLOW
```

```
Config:
```

```
Additional Information:
```

```
Input interface: 'INSIDE'
```

```
Flow type: NO FLOW
```

```
I (0) got initial, attempting ownership.
```

Phase: 5
Type: CLUSTER-EVENT
Subtype:
Result: ALLOW
Config:
Additional Information:
Input interface: 'INSIDE'
Flow type: NO FLOW

I (0) am becoming owner

Unit-2-1 (转发器)

返回流量(TCP SYN/ACK)。关注单位是unit-2-1，它是导向器/备份所有者，并将流量转发给所有者：

<#root>

firepower#

```
cluster exec unit unit-2-1 show capture CAPO packet-number 1 trace
```

```
1: 12:58:33.359249 802.1Q vlan#202 P0 192.168.241.50.80 > 192.168.240.50.
```

46502

```
: S 4257314722:4257314722(0) ack 1434968588 win 28960 <mss 1460,sackOK,timestamp 520893004 510537534,no
```

...

Phase: 4
Type: CLUSTER-EVENT
Subtype:
Result: ALLOW
Config:
Additional Information:
Input interface: 'OUTSIDE'
Flow type: NO FLOW

I (1) got initial, attempting ownership.

Phase: 5
Type: CLUSTER-EVENT
Subtype:
Result: ALLOW
Config:
Additional Information:
Input interface: 'OUTSIDE'
Flow type: NO FLOW

I (1) am early redirecting to (0) due to matching action (-1).

观察4. FTD数据平面系统日志显示所有设备上的连接创建和终止：

```
<#root>
firepower#
cluster exec show log | i 46502

unit-1-1(LOCAL):*****
Dec 01 2020 12:58:33: %FTD-6-302013:
B
uilt inbound TCP connection
 9742 for INSIDE:192.168.240.50/46502 (192.168.240.50/46502) to OUTSIDE:192.168.241.50/80 (192.168.241.50/80)
Dec 01 2020 12:59:02: %FTD-6-302014:
Teardown TCP connection
 9742 for INSIDE:192.168.240.50/46502 to OUTSIDE:192.168.241.50/80 duration 0:00:28 bytes 2048000440 TC

unit-2-1:*****
Dec 01 2020 12:58:33: %FTD-6-302022:
Built forwarder stub TCP connection
 for OUTSIDE:192.168.241.50/80 (192.168.241.50/80) to unknown:192.168.240.50/46502 (192.168.240.50/46502)
Dec 01 2020 12:58:33: %FTD-6-302023:
Teardown forwarder TCP connection
 for OUTSIDE:192.168.241.50/80 to unknown:192.168.240.50/46502 duration 0:00:00 forwarded bytes 0 Forwarder
Dec 01 2020 12:58:33: %FTD-6-302022:
Built director stub TCP connection
 for INSIDE:192.168.240.50/46502 (192.168.240.50/46502) to OUTSIDE:192.168.241.50/80 (192.168.241.50/80)
Dec 01 2020 12:59:02: %FTD-6-302023:
Teardown director TCP connection
 for INSIDE:192.168.240.50/46502 to OUTSIDE:192.168.241.50/80 duration 0:00:28 forwarded bytes 20483163

unit-3-1:*****
firepower#
```

案例研究4. 不对称流量 (所有者为总监)

观察1. reinject-hide捕获显示unit-1-1和unit-2-1 (非对称流) 上的数据包：

```
<#root>
firepower#
cluster exec show cap

unit-1-1(LOCAL):*****
```

```
capture CCL type raw-data buffer 33554432 interface cluster [Buffer Full - 33554229 bytes]
capture CAPI type raw-data buffer 100000 trace interface INSIDE [Buffer Full - 98974 bytes]
match tcp host 192.168.240.50 host 192.168.241.50 eq www
capture CAPO type raw-data buffer 100000 trace interface OUTSIDE [Buffer Full - 98974 bytes]
match tcp host 192.168.240.50 host 192.168.241.50 eq www
capture CAPI_RH type raw-data
```

reinject-hide

buffer 100000 interface

INSIDE

[Buffer Full -

98974 bytes

]

match tcp host 192.168.240.50 host 192.168.241.50 eq www

capture CAPI_RH type raw-data

reinject-hide

buffer 100000 interface

OUTSIDE

[Buffer Full -

99924 bytes

]

match tcp host 192.168.240.50 host 192.168.241.50 eq www

unit-2-1:*****

capture CCL type raw-data buffer 33554432 interface cluster [Buffer Full - 33552925 bytes]

capture CAPI type raw-data buffer 100000 trace interface INSIDE [Capturing - 0 bytes]

match tcp host 192.168.240.50 host 192.168.241.50 eq www

capture CAPO type raw-data buffer 100000 trace interface OUTSIDE [Buffer Full - 99052 bytes]

match tcp host 192.168.240.50 host 192.168.241.50 eq www

capture CAPI_RH type raw-data reinject-hide buffer 100000 interface INSIDE [Capturing - 0 bytes]

match tcp host 192.168.240.50 host 192.168.241.50 eq www

capture CAPO_RH type raw-data

reinject-hide

buffer 100000 interface OUTSIDE [Buffer Full -

99052 bytes

]

match tcp host 192.168.240.50 host 192.168.241.50 eq www

unit-3-1:*****

capture CCL type raw-data buffer 33554432 interface cluster [Capturing - 227690 bytes]

capture CAPI type raw-data buffer 100000 trace interface INSIDE [Capturing - 0 bytes]

match tcp host 192.168.240.50 host 192.168.241.50 eq www

capture CAPO type raw-data buffer 100000 trace interface OUTSIDE [Capturing - 4754 bytes]

match tcp host 192.168.240.50 host 192.168.241.50 eq www

capture CAPI_RH type raw-data reinject-hide buffer 100000 interface INSIDE [Capturing - 0 bytes]

match tcp host 192.168.240.50 host 192.168.241.50 eq www

capture CAPO_RH type raw-data reinject-hide buffer 100000 interface OUTSIDE [Capturing - 0 bytes]

match tcp host 192.168.240.50 host 192.168.241.50 eq www

观察2.与源端口46916的流的连接标志分析。

<#root>

firepower#

cluster exec show conn

unit-1-1

(LOCAL):*****

23 in use, 25 most used

Cluster:

fwd connections: 0 in use, 1 most used

dir connections: 0 in use, 122 most used

centralized connections: 0 in use, 0 most used

VPN redirect connections: 0 in use, 0 most used

Inspect Snort:

preserve-connection: 1 enabled, 0 in effect, 4 most enabled, 1 most in effect

TCP OUTSIDE 192.168.241.50:80 INSIDE 192.168.240.50:

46916

, idle 0:00:00, bytes 414682616,

flags UIO N1

unit-2-1

:*****

21 in use, 271 most used

Cluster:

fwd connections: 1 in use, 2 most used

dir connections: 0 in use, 2 most used

centralized connections: 0 in use, 0 most used

VPN redirect connections: 0 in use, 0 most used

Inspect Snort:

preserve-connection: 0 enabled, 0 in effect, 249 most enabled, 0 most in effect

TCP OUTSIDE 192.168.241.50:80 NP Identity Ifc 192.168.240.50:

46916

, idle 0:00:00, bytes 0,

flags z

unit-3-1

:*****

17 in use, 20 most used

Cluster:


```

fwd connections: 0 in use, 5 most used
dir connections: 1 in use, 127 most used
centralized connections: 0 in use, 0 most used
VPN redirect connections: 0 in use, 0 most used
Inspect Snort:
preserve-connection: 0 enabled, 0 in effect, 1 most enabled, 0 most in effect

```

```
TCP OUTSIDE 192.168.241.50:80 INSIDE 192.168.240.50:
```

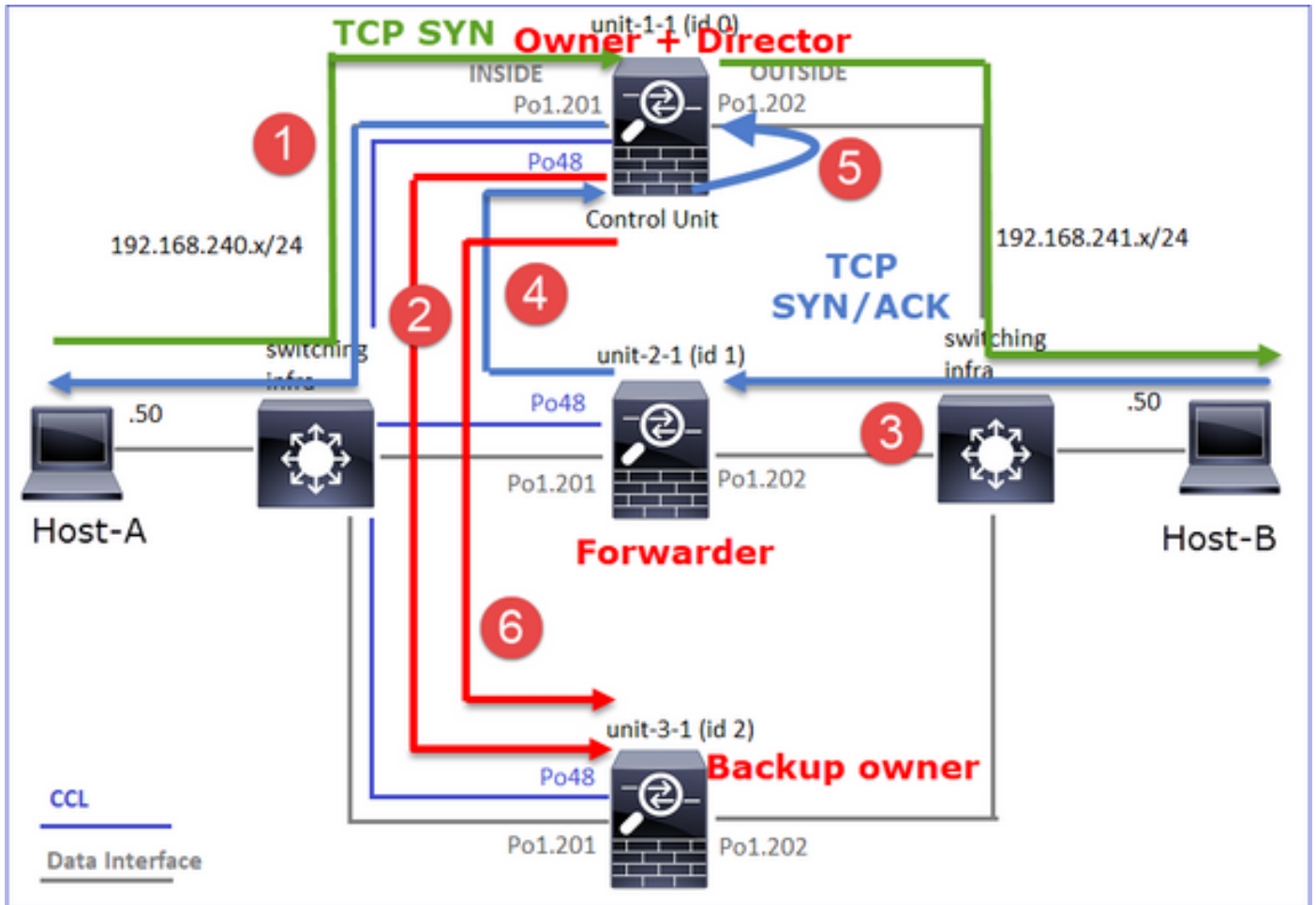
```
46916
```

```
, idle 0:00:04, bytes 0,
```

```
flags y
```

单元	标志	备注
Unit-1-1	UIO	·流所有者 — 设备处理流 ·控制器 — 由于unit-3-1具有“y”而不是“Y”，这意味着已选择unit-1-1作为此流的控制器。因此，由于它也是所有者，因此选择了另一设备（本例中为unit-3-1）作为备用所有者
Unit-2-1	z	·转发器
Unit-3-1	y	— 备份所有者

上述内容可以图形表示为：



1. TCP SYN数据包从主机A到达unit-1-1。Unit-1-1成为流所有者，并被选举为导向器。
2. Unit-3-1被选为备份所有者。流所有者在UDP 4193上发送单播“cluster add”消息以通知备份所有者有关流的消息。
3. TCP SYN/ACK数据包从主机B到达设备2-1。流量不对称。
4. Unit-2-1通过CCL将数据包转发给所有者（由于TCP SYN Cookie）。
5. 所有者重新在接口OUTSIDE上注入数据包，然后将数据包转发到主机A。
6. 连接终止后，所有者会发送集群删除消息，以从备份所有者中删除流信息。

观察3.使用trace捕获显示非对称流量以及从unit-2-1到unit-1-1的重定向。

Unit-2-1 (转发器)

```
<#root>
```

```
firepower#
```

```
cluster exec unit unit-2-1 show capture CAPO packet-number 1 trace
```

```
1: 16:11:33.653164 802.1Q vlan#202 P0 192.168.241.50.80 > 192.168.240.50.
```

```
46916
```

```
:
```

```
s
```

1331019196:1331019196(0)

ack

3089755618 win 28960 <mss 1460,sackOK,timestamp 532473211 522117741,nop,wscale 7>

...

Phase: 4

Type: CLUSTER-EVENT

Subtype:

Result: ALLOW

Config:

Additional Information:

Input interface: 'OUTSIDE'

Flow type: NO FLOW

I (1) got initial, attempting ownership.

Phase: 5

Type: CLUSTER-EVENT

Subtype:

Result: ALLOW

Config:

Additional Information:

Input interface: 'OUTSIDE'

Flow type: NO FLOW

I (1) am early redirecting to (0) due to matching action (-1).

观察4. FTD数据平面系统日志显示所有设备上的连接创建和终止：

- Unit-1-1 (所有者)
- Unit-2-1 (转发器)
- Unit-3-1 (备份所有者)

<#root>

firepower#

cluster exec show log | i 46916

unit-1-1(LOCAL):*****

Dec 01 2020 16:11:33: %FTD-6-302013:

Built inbound TCP connection

10023 for INSIDE:192.168.240.50/46916 (192.168.240.50/46916) to OUTSIDE:192.168.241.50/80 (192.168.241.50/80)

Dec 01 2020 16:11:42: %FTD-6-302014:

Teardown TCP connection

10023 for INSIDE:192.168.240.50/46916 to OUTSIDE:192.168.241.50/80 duration 0:00:09 bytes 1024010016 T

unit-2-1:*****

Dec 01 2020 16:11:33: %FTD-6-302022:

Built forwarder stub TCP connection

```
for OUTSIDE:192.168.241.50/80 (192.168.241.50/80) to unknown:192.168.240.50/46916 (192.168.240.50/46916)
Dec 01 2020 16:11:42: %FTD-6-302023:
```

```
Teardown forwarder TCP connection
```

```
for OUTSIDE:192.168.241.50/80 to unknown:192.168.240.50/46916 duration 0:00:09 forwarded bytes 1024009
```

```
unit-3-1:*****
Dec 01 2020 16:11:33: %FTD-6-302022:
```

```
Built backup stub TCP connection
```

```
for INSIDE:192.168.240.50/46916 (192.168.240.50/46916) to OUTSIDE:192.168.241.50/80 (192.168.241.50/80)
Dec 01 2020 16:11:42: %FTD-6-302023:
```

```
Teardown backup TCP connection
```

```
for INSIDE:192.168.240.50/46916 to OUTSIDE:192.168.241.50/80 duration 0:00:09 forwarded bytes 0 Cluste
```

案例研究5.不对称流量 (所有者与主管不同)。

观察1. reinject-hide捕获显示unit-1-1和unit-2-1 (非对称流) 上的数据包:

```
<#root>
```

```
firepower#
```

```
cluster exec show cap
```

```
unit-1-1
```

```
(LOCAL):*****
```

```
capture CCL type raw-data buffer 33554432 interface cluster [Buffer Full - 33553207 bytes]
```

```
capture CAPI type raw-data buffer 100000 trace interface INSIDE [Buffer Full - 99396 bytes]
```

```
match tcp host 192.168.240.50 host 192.168.241.50 eq www
```

```
capture CAPO type raw-data buffer 100000 trace interface OUTSIDE [Buffer Full - 99224 bytes]
```

```
match tcp host 192.168.240.50 host 192.168.241.50 eq www
```

```
capture CAPI_RH type raw-data
```

```
reinject-hide
```

```
buffer 100000 interface
```

```
INSIDE
```

```
[Buffer Full -
```

```
99396 bytes
```

```
]
```

```
match tcp host 192.168.240.50 host 192.168.241.50 eq www
```

```
capture CAPO_RH type raw-data
```

```
reinject-hid
```

```
e buffer 100000 interface
```

```
OUTSIDE
```

```
[Buffer Full -
```

99928 bytes

```
]
match tcp host 192.168.240.50 host 192.168.241.50 eq www
```

unit-2-1

```
:*****
capture CCL type raw-data buffer 33554432 interface cluster [Buffer Full - 33554251 bytes]
capture CAPI type raw-data buffer 100000 trace interface INSIDE [Capturing - 0 bytes]
match tcp host 192.168.240.50 host 192.168.241.50 eq www
capture CAPO type raw-data buffer 100000 trace interface OUTSIDE [Buffer Full - 99052 bytes]
match tcp host 192.168.240.50 host 192.168.241.50 eq www
capture CAPI_RH type raw-data reinject-hide buffer 100000 interface INSIDE [Capturing - 0 bytes]
match tcp host 192.168.240.50 host 192.168.241.50 eq www
capture CAPO_RH type raw-data
```

reinject-hide

buffer 100000 interface

OUTSIDE

[Buffer Full -

99052 bytes

```
]
match tcp host 192.168.240.50 host 192.168.241.50 eq www
```

```
unit-3-1:*****
capture CCL type raw-data buffer 33554432 interface cluster [Capturing - 131925 bytes]
capture CAPI type raw-data buffer 100000 trace interface INSIDE [Capturing - 0 bytes]
match tcp host 192.168.240.50 host 192.168.241.50 eq www
capture CAPO type raw-data buffer 100000 trace interface OUTSIDE [Capturing - 2592 bytes]
match tcp host 192.168.240.50 host 192.168.241.50 eq www
capture CAPI_RH type raw-data reinject-hide buffer 100000 interface INSIDE [Capturing - 0 bytes]
match tcp host 192.168.240.50 host 192.168.241.50 eq www
capture CAPO_RH type raw-data reinject-hide buffer 100000 interface OUTSIDE [Capturing - 0 bytes]
match tcp host 192.168.240.50 host 192.168.241.50 eq www
```

观察2.源端口为46994的流的连接标志分析:

<#root>

firepower#

cluster exec show conn

unit-1-1

```
(LOCAL):*****
23 in use, 25 most used
Cluster:
fwd connections: 0 in use, 1 most used
dir connections: 0 in use, 122 most used
centralized connections: 0 in use, 0 most used
```

VPN redirect connections: 0 in use, 0 most used
Inspect Snort:
preserve-connection: 1 enabled, 0 in effect, 4 most enabled, 1 most in effect

TCP OUTSIDE 192.168.241.50:80 INSIDE 192.168.240.50:

46994

, idle 0:00:00, bytes 406028640,

flags UIO N1

unit-2-1

:*****
22 in use, 271 most used
Cluster:
fwd connections: 1 in use, 2 most used
dir connections: 0 in use, 2 most used
centralized connections: 0 in use, 0 most used
VPN redirect connections: 0 in use, 0 most used
Inspect Snort:
preserve-connection: 0 enabled, 0 in effect, 249 most enabled, 0 most in effect

TCP OUTSIDE 192.168.241.50:80 NP Identity Ifc 192.168.240.50:

46994

, idle 0:00:00, bytes 0,

flags z

unit-3-1

:*****
17 in use, 20 most used
Cluster:
fwd connections: 2 in use, 5 most used
dir connections: 1 in use, 127 most used
centralized connections: 0 in use, 0 most used
VPN redirect connections: 0 in use, 0 most used
Inspect Snort:
preserve-connection: 0 enabled, 0 in effect, 1 most enabled, 0 most in effect

TCP OUTSIDE 192.168.241.50:80 INSIDE 192.168.240.50:

46994

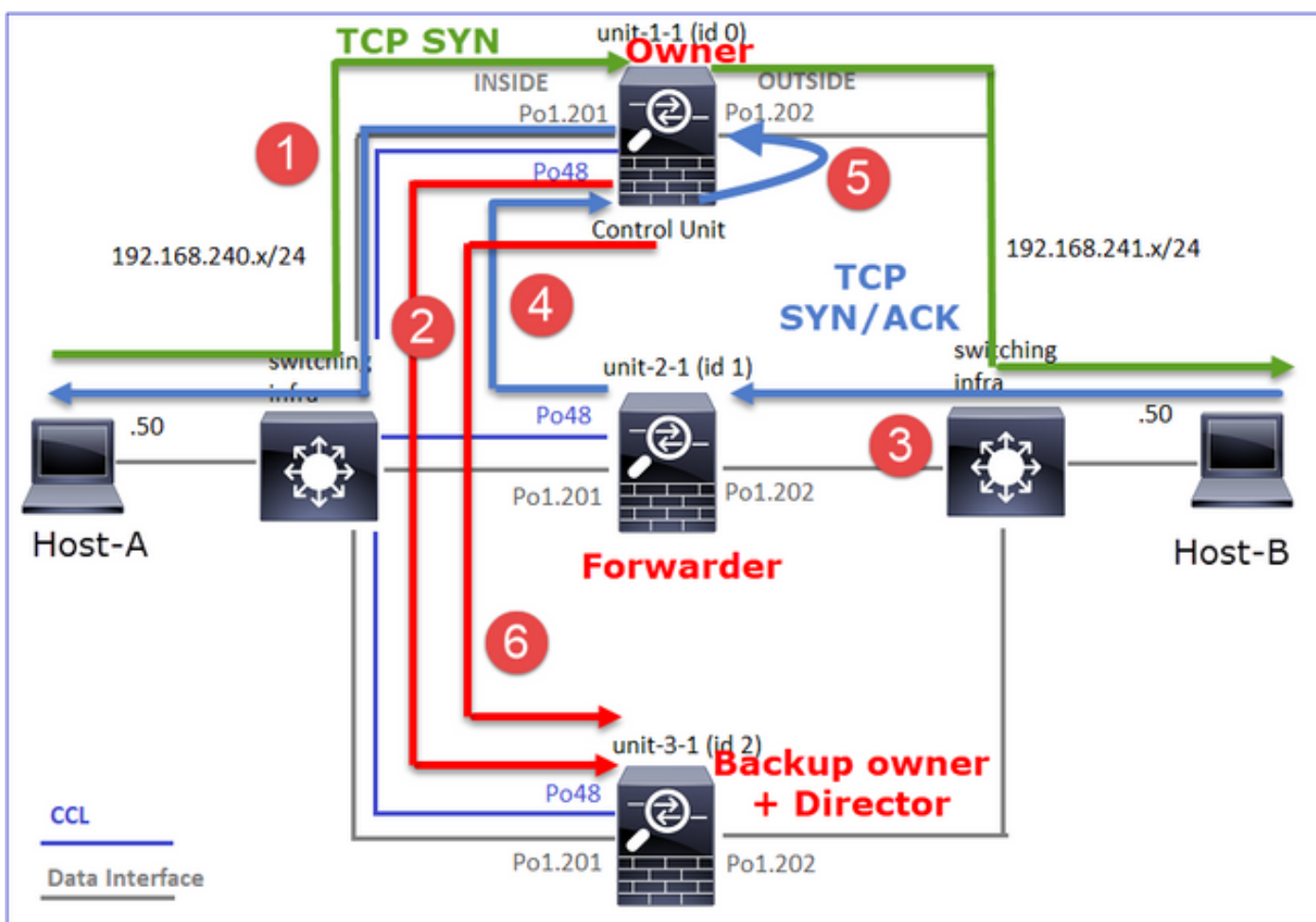
, idle 0:00:05, bytes 0,

flags Y

单元	标志	备注
----	----	----

Unit-1-1	UIO	·流所有者 — 设备处理流
Unit-2-1	Z	·转发器
Unit-3-1	Y	·备份所有者 ·董事

上述内容可以图形表示为：



1. TCP SYN数据包从主机A到达设备1-1。设备1-1成为流所有者。
2. Unit-3-1被选举为总监和备份所有者。流所有者在UDP 4193上发送“cluster add”单播消息以通知备份所有者有关流的消息。
3. TCP SYN/ACK数据包从主机B到达设备2-1。流量不对称
4. Unit-2-1通过CCL将数据包转发给所有者（由于TCP SYN Cookie）。
5. 所有者重新在接口OUTSIDE上注入数据包，然后将数据包转发到主机A。
6. 连接终止后，所有者会发送集群删除消息，以从备份所有者中删除流信息。

观察3.使用trace捕获显示非对称流量以及从unit-2-1到unit-1-1的重定向。

Unit-1-1 (所有者)

<#root>

firepower#

cluster exec show cap CAPI packet-number 1 trace

unit-1-1(LOCAL):*****

...
Phase: 4
Type: CLUSTER-EVENT
Subtype:
Result: ALLOW
Config:
Additional Information:
Input interface: 'INSIDE'
Flow type: NO FLOW

I (0) got initial, attempting ownership.

Phase: 5
Type: CLUSTER-EVENT
Subtype:
Result: ALLOW
Config:
Additional Information:
Input interface: 'INSIDE'
Flow type: NO FLOW

I (0) am becoming owner

Unit-2-1 (转发器)

<#root>

firepower#

cluster exec unit unit-2-1 show cap CAPO packet-number 1 trace

1: 16:46:44.232074 802.1Q vlan#202 P0 192.168.241.50.80 > 192.168.240.50.

46994

: S 2863659376:2863659376(0) ack 2879616990 win 28960 <mss 1460,sackOK,timestamp 534583774 524228304,no

...
Phase: 4
Type: CLUSTER-EVENT
Subtype:
Result: ALLOW
Config:
Additional Information:
Input interface: 'OUTSIDE'
Flow type: NO FLOW

I (1) got initial, attempting ownership.

Phase: 5
Type: CLUSTER-EVENT
Subtype:
Result: ALLOW
Config:
Additional Information:
Input interface: 'OUTSIDE'
Flow type: NO FLOW

I (1) am early redirecting to (0) due to matching action (-1).

观察4. FTD数据平面系统日志显示所有设备上的连接创建和终止：

- Unit-1-1 (所有者)
- Unit-2-1 (转发器)
- Unit-3-1 (备份所有者/导向器)

<#root>

firepower#

cluster exec show log | i 46994

unit-1-1(LOCAL):*****

Dec 01 2020 16:46:44: %FTD-6-302013:

Built inbound TCP connection

10080 for INSIDE:192.168.240.50/46994 (192.168.240.50/46994) to OUTSIDE:192.168.241.50/80 (192.168.241.50/80)
Dec 01 2020 16:46:53: %FTD-6-302014:

Teardown TCP connection

10080 for INSIDE:192.168.240.50/46994 to OUTSIDE:192.168.241.50/80 duration 0:00:09 bytes 1024000440 T

unit-2-1:*****

Dec 01 2020 16:46:44: %FTD-6-302022:

Built forwarder stub TCP connection

for OUTSIDE:192.168.241.50/80 (192.168.241.50/80) to unknown:192.168.240.50/46994 (192.168.240.50/46994)
Dec 01 2020 16:46:53: %FTD-6-302023:

Teardown forwarder TCP connection

for OUTSIDE:192.168.241.50/80 to unknown:192.168.240.50/46994 duration 0:00:09 forwarded bytes 1024000

unit-3-1:*****

Dec 01 2020 16:46:44: %FTD-6-302022:

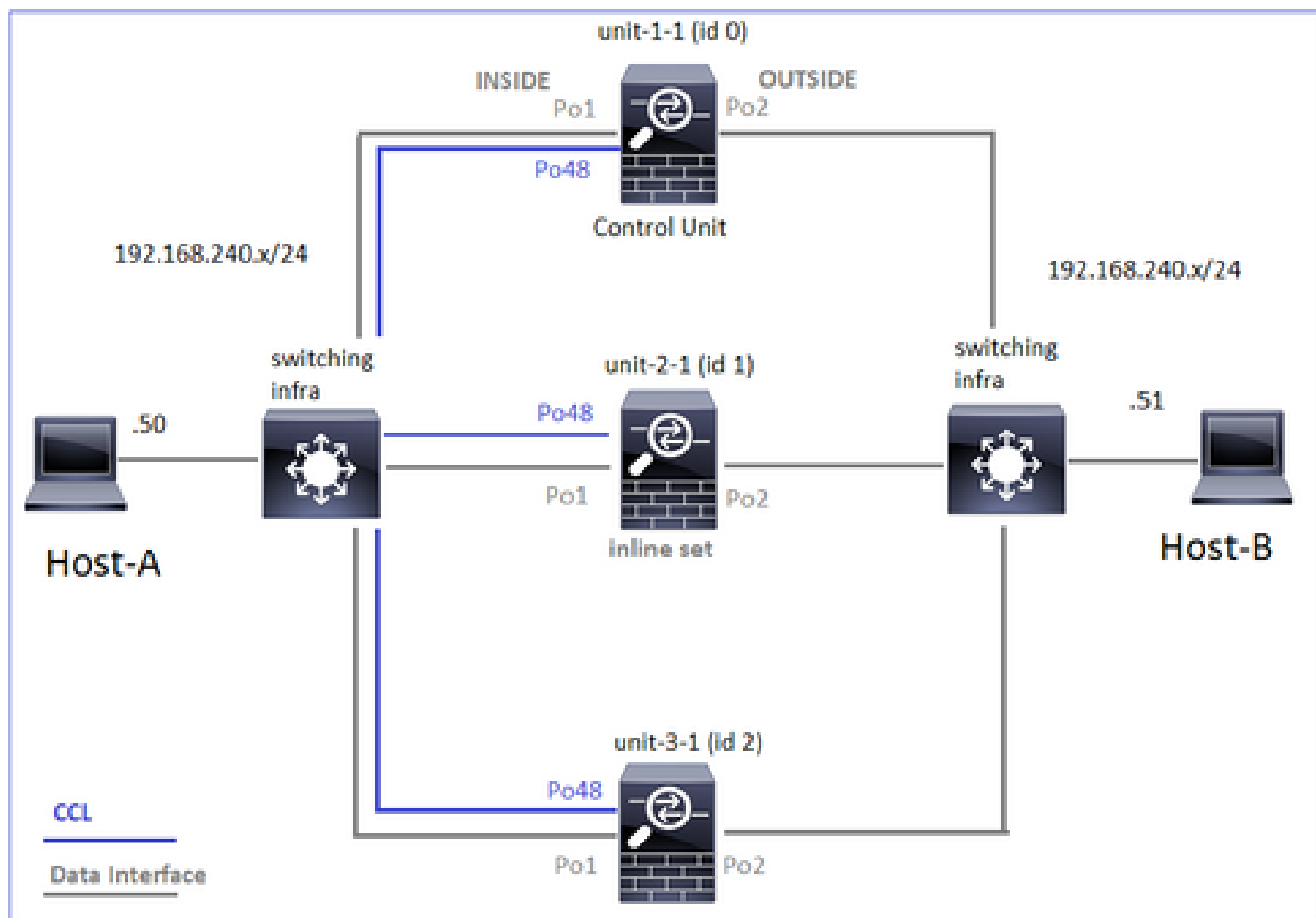
Built director stub TCP connection

for INSIDE:192.168.240.50/46994 (192.168.240.50/46994) to OUTSIDE:192.168.241.50/80 (192.168.241.50/80)
Dec 01 2020 16:46:53: %FTD-6-302023:

Teardown director TCP connection

for INSIDE:192.168.240.50/46994 to OUTSIDE:192.168.241.50/80 duration 0:00:09 forwarded bytes 0 Cluste

在下一个案例研究中，使用的拓扑基于具有内联集集群：



案例研究6.非对称流量 (内联集，所有者为导向者)

观察1.重新隐藏捕获显示unit-1-1和unit-2-1 (非对称流) 上的数据包。此外，所有者是unit-2-1 (INSIDE和OUTSIDE接口上都有用于重新隐藏 — 隐藏捕获的数据包，而unit-1-1只有 OUTSIDE上的数据包)：

```
<#root>
```

```
firepower#
```

```
cluster exec show cap
```

```
unit-1-1
```

```
(LOCAL):*****  
capture CCL type raw-data buffer 33554432 interface cluster [Buffer Full - 33553253 bytes]  
capture CAPO type raw-data trace interface OUTSIDE [Buffer Full - 523432 bytes]  
match tcp host 192.168.240.50 host 192.168.240.51 eq www
```

```

capture CAPI type raw-data trace interface INSIDE [Capturing - 0 bytes]
match tcp host 192.168.240.50 host 192.168.240.51 eq www
capture CAPO_RH type raw-data

reinject-hide

interface

OUTSIDE

[Buffer Full -

523432 bytes

]
match tcp host 192.168.240.50 host 192.168.240.51 eq www
capture CAPI_RH type raw-data reinject-hide interface INSIDE [Capturing - 0 bytes]
match tcp host 192.168.240.50 host 192.168.240.51 eq www

unit-2-1

:*****
capture CCL type raw-data buffer 33554432 interface cluster [Buffer Full - 33554312 bytes]
capture CAPO type raw-data trace interface OUTSIDE [Buffer Full - 523782 bytes]
match tcp host 192.168.240.50 host 192.168.240.51 eq www
capture CAPI type raw-data trace interface INSIDE [Buffer Full - 523782 bytes]
match tcp host 192.168.240.50 host 192.168.240.51 eq www
capture CAPO_RH type raw-data

reinject-hide

interface

OUTSIDE

[Buffer Full -

524218 bytes

]
match tcp host 192.168.240.50 host 192.168.240.51 eq www
capture CAPI_RH type raw-data

reinject-hide

interface

INSIDE

[Buffer Full -

523782 bytes]

match tcp host 192.168.240.50 host 192.168.240.51 eq www

unit-3-1:*****
capture CCL type raw-data buffer 33554432 interface cluster [Capturing - 53118 bytes]
capture CAPO type raw-data trace interface OUTSIDE [Capturing - 0 bytes]
match tcp host 192.168.240.50 host 192.168.240.51 eq www
capture CAPI type raw-data trace interface INSIDE [Capturing - 0 bytes]
match tcp host 192.168.240.50 host 192.168.240.51 eq www
capture CAPO_RH type raw-data reinject-hide interface OUTSIDE [Capturing - 0 bytes]
match tcp host 192.168.240.50 host 192.168.240.51 eq www
capture CAPI_RH type raw-data reinject-hide interface INSIDE [Capturing - 0 bytes]

```

```
match tcp host 192.168.240.50 host 192.168.240.51 eq www
```

观察2.与源端口51844的流的连接标志分析。

```
<#root>
```

```
firepower#
```

```
cluster exec show conn addr 192.168.240.51
```

```
unit-1-1
```

```
(LOCAL):*****
```

```
30 in use, 102 most used
```

```
Cluster:
```

```
fwd connections: 1 in use, 1 most used
```

```
dir connections: 2 in use, 122 most used
```

```
centralized connections: 3 in use, 39 most used
```

```
VPN redirect connections: 0 in use, 0 most used
```

```
Inspect Snort:
```

```
preserve-connection: 0 enabled, 0 in effect, 4 most enabled, 1 most in effect
```

```
TCP OUTSIDE 192.168.240.51:80 NP Identity Ifc 192.168.240.50:
```

```
51844
```

```
, idle 0:00:00, bytes 0,
```

```
flags z
```

```
unit-2-1
```

```
:*****
```

```
23 in use, 271 most used
```

```
Cluster:
```

```
fwd connections: 0 in use, 2 most used
```

```
dir connections: 4 in use, 26 most used
```

```
centralized connections: 0 in use, 14 most used
```

```
VPN redirect connections: 0 in use, 0 most used
```

```
Inspect Snort:
```

```
preserve-connection: 0 enabled, 0 in effect, 249 most enabled, 0 most in effect
```

```
TCP OUTSIDE 192.168.240.51:80 INSIDE 192.168.240.50:
```

```
51844
```

```
, idle 0:00:00, bytes 231214400,
```

```
flags b N
```

```
unit-3-1
```

```
:*****
```

20 in use, 55 most used

Cluster:

fwd connections: 0 in use, 5 most used

dir connections: 1 in use, 127 most used

centralized connections: 0 in use, 24 most used

VPN redirect connections: 0 in use, 0 most used

Inspect Snort:

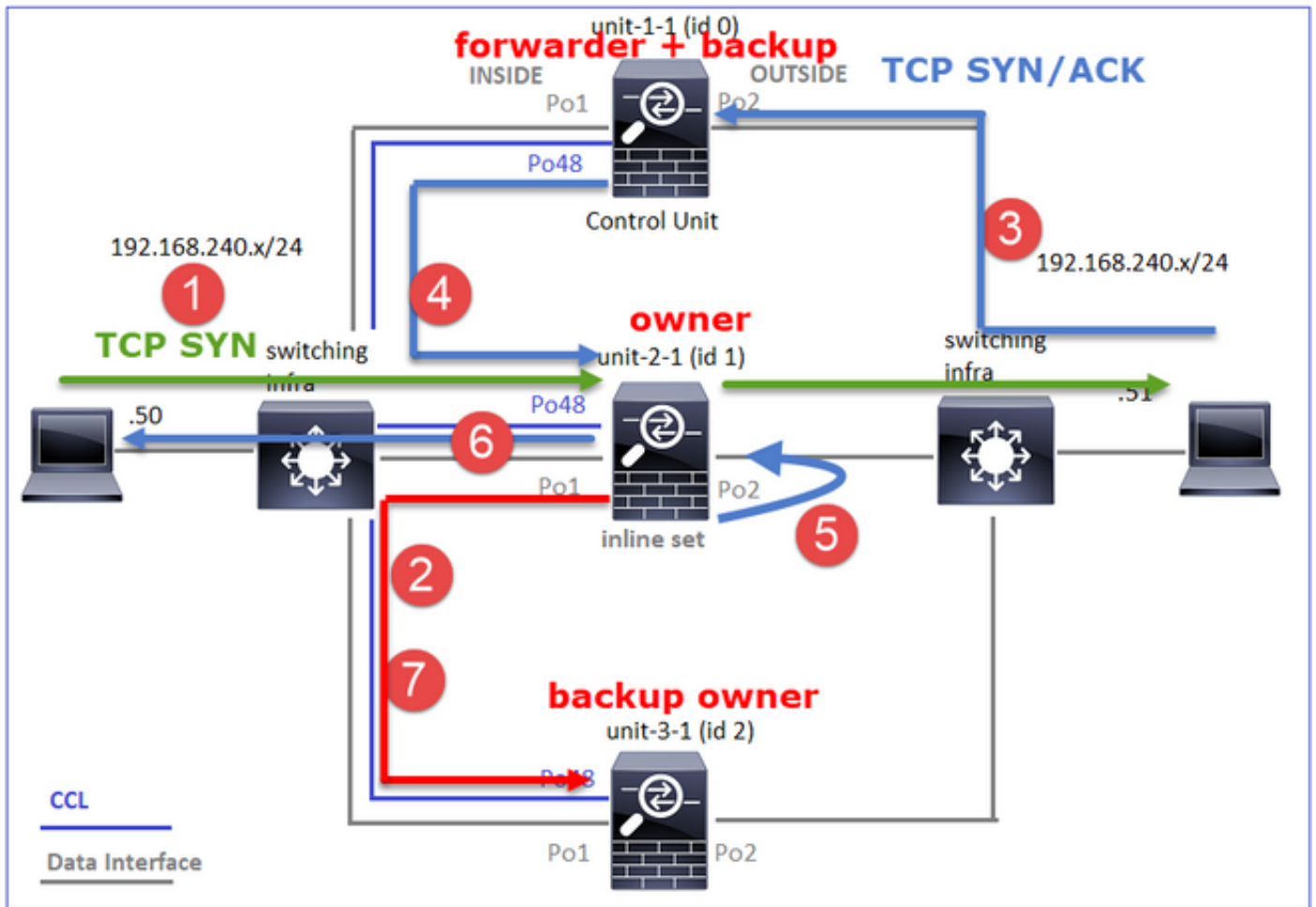
preserve-connection: 0 enabled, 0 in effect, 1 most enabled, 0 most in effect

TCP OUTSIDE 192.168.240.51:80 INSIDE 192.168.240.50:51844, idle 0:00:01, bytes 0,

flags y

单元	标志	备注
Unit-1-1	z	·转发器
Unit-2-1	b N	·流所有者 — 设备处理流
Unit-3-1	y	·备份所有者

上述内容可以图形表示为：



1. TCP SYN数据包从主机A到达unit-2-1。Unit-2-1成为流所有者并被选为指挥交换机。
2. Unit-3-1被选为备份所有者。流所有者在UDP 4193上发送“cluster add”单播消息以通知备份所有者有关流的消息。
3. TCP SYN/ACK数据包从主机B到达设备1-1。流量不对称。
4. Unit-1-1通过CCL将数据包转发到导向器(unit-2-1)。
5. Unit-2-1也是所有者，它在接口OUTSIDE上重新注入数据包。
6. Unit-2-1将数据包转发到主机A。
7. 连接终止后，所有者会发送集群删除消息，以从备份所有者中删除流信息。

观察3.使用trace捕获显示非对称流量以及从unit-1-1到unit-2-1的重定向。

Unit-2-1 (所有者/主管)

```
<#root>
```

```
firepower#
```

```
cluster exec unit unit-2-1 show cap CAPI packet-number 1 trace
```

```
1: 18:10:12.842912 192.168.240.50.51844 > 192.168.240.51.80:
```

```
s
```

```
4082593463:4082593463(0) win 29200 <mss 1460,sackOK,timestamp 76258053 0,nop,wscale 7>
```

```
Phase: 1
```

```
Type: CLUSTER-EVENT
```

Subtype:
Result: ALLOW
Config:
Additional Information:
Input interface: 'INSIDE'
Flow type: NO FLOW

I (1) got initial, attempting ownership.

Phase: 2
Type: CLUSTER-EVENT
Subtype:
Result: ALLOW
Config:
Additional Information:
Input interface: 'INSIDE'
Flow type: NO FLOW

I (1) am becoming owner

Unit-1-1 (转发器)

<#root>

firepower#

cluster exec show cap CAPO packet-number 1 trace

unit-1-1(LOCAL):*****

1: 18:10:12.842317 192.168.240.51.80 > 192.168.240.50.51844: S 2339579109:2339579109(0) ack 4082593464 v
Phase: 1
Type: CLUSTER-EVENT
Subtype:
Result: ALLOW
Config:
Additional Information:
Input interface: 'OUTSIDE'
Flow type: NO FLOW

I (0) am asking director (1).

返回流量(TCP SYN/ACK)

Unit-2-1 (所有者/主管)

<#root>

firepower#

```
cluster exec unit unit-2-1 show cap CAPO packet-number 2 trace
```

```
2: 18:10:12.843660 192.168.240.51.80 > 192.168.240.50.51844: S 2339579109:2339579109(0) ack 4082593464 v
Phase: 1
Type: CLUSTER-EVENT
Subtype:
Result: ALLOW
Config:
Additional Information:
Input interface: 'OUTSIDE'
Flow type: FULL
```

```
I (1) am owner, update sender (0).
```

```
Phase: 2
Type: FLOW-LOOKUP
Subtype:
Result: ALLOW
Config:
Additional Information:
```

```
Found flow with id 7109, using existing flow
```

观察4. FTD数据平面系统日志显示所有设备上的连接创建和终止：

- Unit-1-1 (所有者)
- Unit-2-1 (转发器)
- Unit-3-1 (备份所有者/导向器)

```
<#root>
```

```
firepower#
```

```
cluster exec show log | include 51844
```

```
unit-1-1(LOCAL):*****
```

```
Dec 02 2020 18:10:12: %FTD-6-302022:
```

```
Built forwarder stub TCP connection
```

```
for OUTSIDE:192.168.240.51/80 (192.168.240.51/80) to unknown:192.168.240.50/51844 (192.168.240.50/51844)
```

```
Dec 02 2020 18:10:22: %FTD-6-302023:
```

```
Teardown forwarder TCP connection
```

```
for OUTSIDE:192.168.240.51/80 to unknown:192.168.240.50/51844 duration 0:00:09 forwarded bytes 1024001
```

```
unit-2-1:*****
```

```
Dec 02 2020 18:10:12: %FTD-6-302303:
```

```
Built TCP state-bypass connection
```

```
7109 from INSIDE:192.168.240.50/51844 (192.168.240.50/51844) to OUTSIDE:192.168.240.51/80 (192.168.240.51/80)
```


Dec 02 2020 18:10:22: %FTD-6-302304:

Teardown TCP state-bypass connection

7109 from INSIDE:192.168.240.50/51844 to OUTSIDE:192.168.240.51/80 duration 0:00:09 bytes 1024001888 T

unit-3-1:*****

Dec 02 2020 18:10:12: %FTD-6-302022:

Built backup stub TCP connection

for INSIDE:192.168.240.50/51844 (192.168.240.50/51844) to OUTSIDE:192.168.240.51/80 (192.168.240.51/80)

Dec 02 2020 18:10:22: %FTD-6-302023:

Teardown backup TCP connection

for INSIDE:192.168.240.50/51844 to OUTSIDE:192.168.240.51/80 duration 0:00:09 forwarded bytes 0 Cluste

案例研究7.非对称流量 (内联集 , 所有者与导向者不同)

所有者是unit-2-1 (INSIDE和OUTSIDE接口上都有数据包用于重新隐藏捕获 , 而unit-3-1只有OUTSIDE) :

<#root>

firepower#

cluster exec show cap

unit-1-1(LOCAL):*****

capture CCL type raw-data buffer 33554432 interface cluster [Capturing - 13902 bytes]

capture CAPO type raw-data trace interface OUTSIDE [Capturing - 90 bytes]

match tcp host 192.168.240.50 host 192.168.240.51 eq www

capture CAPI type raw-data trace interface INSIDE [Capturing - 0 bytes]

match tcp host 192.168.240.50 host 192.168.240.51 eq www

capture CAPO_RH type raw-data reinject-hide interface OUTSIDE [Capturing - 0 bytes]

match tcp host 192.168.240.50 host 192.168.240.51 eq www

capture CAPI_RH type raw-data reinject-hide interface INSIDE [Capturing - 0 bytes]

match tcp host 192.168.240.50 host 192.168.240.51 eq www

unit-2-1

:*****

capture CCL type raw-data buffer 33554432 interface cluster [Buffer Full - 33553936 bytes]

capture CAPO type raw-data trace interface OUTSIDE [Buffer Full - 523126 bytes]

match tcp host 192.168.240.50 host 192.168.240.51 eq www

capture CAPI type raw-data trace interface INSIDE [Buffer Full - 523126 bytes]

match tcp host 192.168.240.50 host 192.168.240.51 eq www

capture CAPO_RH type raw-data

reinject-hid

e

interface

OUTSIDE

[Buffer Full -

524230 bytes

```
]
match tcp host 192.168.240.50 host 192.168.240.51 eq www
capture CAPI_RH type raw-data
```

reinject-hide

interface

INSIDE

[Buffer Full -

523126 bytes

```
]
match tcp host 192.168.240.50 host 192.168.240.51 eq www
```

unit-3-1

```
:*****
capture CCL type raw-data buffer 33554432 interface cluster [Buffer Full - 33553566 bytes]
capture CAPO type raw-data trace interface OUTSIDE [Buffer Full - 523522 bytes]
match tcp host 192.168.240.50 host 192.168.240.51 eq www
capture CAPI type raw-data trace interface INSIDE [Capturing - 0 bytes]
match tcp host 192.168.240.50 host 192.168.240.51 eq www
capture CAPO_RH type raw-data
```

reinject-hide

interface

OUTSIDE

[Buffer Full -

523432 bytes

```
]
match tcp host 192.168.240.50 host 192.168.240.51 eq www
capture CAPI_RH type raw-data reinject-hide interface INSIDE [Capturing - 0 bytes]
match tcp host 192.168.240.50 host 192.168.240.51 eq www
```

观察2.与源端口59210的流的连接标志分析。

<#root>

firepower#

cluster exec show conn addr 192.168.240.51

unit-1-1

```
(LOCAL):*****
25 in use, 102 most used
Cluster:
fwd connections: 0 in use, 1 most used
```

dir connections: 2 in use, 122 most used
centralized connections: 0 in use, 39 most used
VPN redirect connections: 0 in use, 0 most used
Inspect Snort:
preserve-connection: 0 enabled, 0 in effect, 4 most enabled, 1 most in effect

TCP OUTSIDE 192.168.240.51:80 INSIDE 192.168.240.50:

59210

, idle 0:00:03, bytes 0,

flags Y

unit-2-1

:*****

21 in use, 271 most used

Cluster:

fwd connections: 0 in use, 2 most used

dir connections: 0 in use, 28 most used

centralized connections: 0 in use, 14 most used

VPN redirect connections: 0 in use, 0 most used

Inspect Snort:

preserve-connection: 0 enabled, 0 in effect, 249 most enabled, 0 most in effect

TCP OUTSIDE 192.168.240.51:80 INSIDE 192.168.240.50:

59210

, idle 0:00:00, bytes 610132872,

flags b N

unit-3-1

:*****

19 in use, 55 most used

Cluster:

fwd connections: 1 in use, 5 most used

dir connections: 0 in use, 127 most used

centralized connections: 0 in use, 24 most used

VPN redirect connections: 0 in use, 0 most used

Inspect Snort:

preserve-connection: 0 enabled, 0 in effect, 1 most enabled, 0 most in effect

TCP OUTSIDE 192.168.240.51:80 NP Identity Ifc 192.168.240.50:

59210

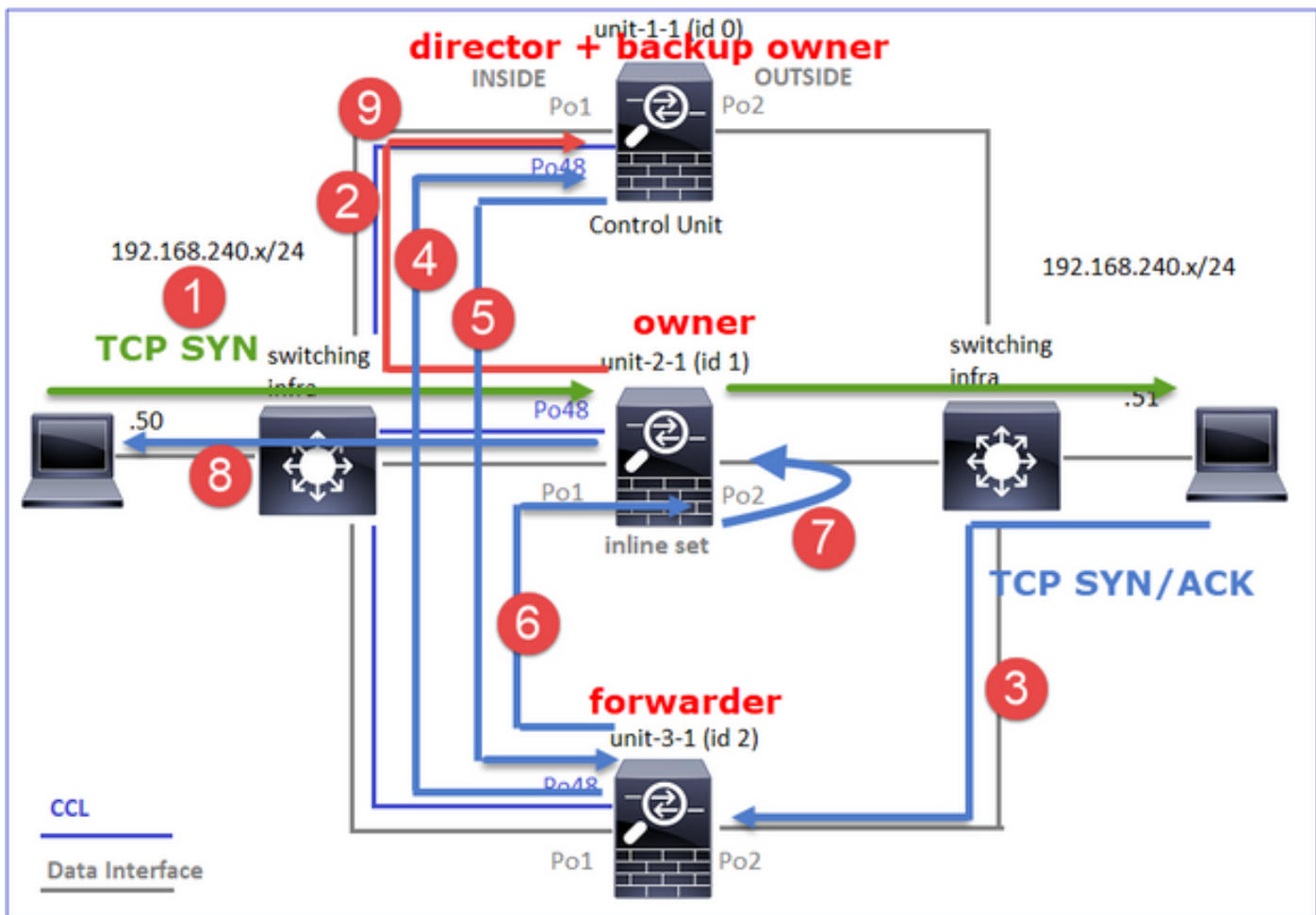
, idle 0:00:00, bytes 0,

flags z


单元	标志	备注
----	----	----

Unit-1-1	Y	·控制器/备份所有者
Unit-2-1	b N	·流所有者 — 设备处理流
Unit-3-1	z	·转发器

上述内容可以图形表示为：



1. TCP SYN数据包从主机A到达设备2-1。设备2-1成为流所有者，设备1-1被选为导向器
2. Unit-1-1被选为备份所有者（因为它是主管）。流所有者将UDP 4193上的“cluster add”单播消息发送到。通知备份所有者有关流量的信息。
3. TCP SYN/ACK数据包从主机B到达设备3-1。流量不对称。
4. Unit-3-1通过CCL将数据包转发到导向器(unit-1-1)。
5. Unit-1-1(director)知道所有者是unit-2-1，将数据包发送回转发器(unit-3-1)，并通知他所有者是unit-2-1。
6. Unit-3-1将数据包发送到unit-2-1(owner)。
7. Unit-2-1在接口OUTSIDE上重新注入数据包。
8. Unit-2-1将数据包转发到主机A。
9. 连接终止后，所有者会发送集群删除消息，以从备份所有者中删除流信息。

 注意：第2步（通过CCL的数据包）在第4步（数据流量）之前发生非常重要。在其他情况下（例如，竞争条件），指挥交换机不知道流。因此，由于数据包是内联集，因此它会将数据包转发到目的地。如果接口不在内联集中，数据包将被丢弃。

观察3.使用trace捕获显示CCL上的非对称流量和交换：

转发流量(TCP SYN)

Unit-2-1 (所有者)

<#root>

firepower#

```
cluster exec unit unit-2-1 show cap CAPI packet-number 1 trace
```

```
1: 09:19:49.760702 192.168.240.50.59210 > 192.168.240.51.80: S 4110299695:4110299695(0) win 29200 <ms
Phase: 1
Type: CLUSTER-EVENT
Subtype:
Result: ALLOW
Config:
Additional Information:
Input interface: 'INSIDE'
Flow type: NO FLOW
```

```
I (1) got initial, attempting ownership.
```

```
Phase: 2
Type: CLUSTER-EVENT
Subtype:
Result: ALLOW
Config:
Additional Information:
Input interface: 'INSIDE'
Flow type: NO FLOW
```

```
I (1) am becoming owner
```

返回流量(TCP SYN/ACK)

Unit-3-1 (ID 2 — 转发器) 通过CCL将数据包发送到unit-1-1(ID 0 - director)。

<#root>

firepower#

```
cluster exec unit unit-3-1 show cap CAPO packet-number 1 trace
```

1: 09:19:49.760336 192.168.240.51.80 > 192.168.240.50.59210:

s

4209225081:4209225081(0)

ack

4110299696 win 28960 <mss 1460,sackOK,timestamp 567715984 130834570,nop,wscale 7>
Phase: 1
Type: CLUSTER-EVENT
Subtype:
Result: ALLOW
Config:
Additional Information:
Input interface: 'OUTSIDE'
Flow type: NO FLOW

I (2) am asking director (0).

Unit-1-1(director)- Unit-1-1(ID 0)知道流所有者是unit-2-1(ID 1), 并通过CCL将数据包发送回unit-3-1 (ID 2 — 转发器)。

<#root>

firepower#

cluster exec show cap CAPO packet-number 1 trace

unit-1-1(LOCAL):*****

1: 09:19:49.761038 192.168.240.51.80 > 192.168.240.50.59210:

s

4209225081:4209225081(0)

ack

4110299696 win 28960 <mss 1460,sackOK,timestamp 567715984 130834570,nop,wscale 7>
Phase: 1
Type: CLUSTER-EVENT
Subtype:
Result: ALLOW
Config:
Additional Information:
Input interface: 'OUTSIDE'
Flow type: STUB

I (0) am director, valid owner (1), update sender (2).

Unit-3-1 (ID 2 — 转发器) 通过CCL获取数据包并将其发送到unit-2-1 (ID 1 — 所有者)。

```
<#root>
```

```
firepower#
```

```
cluster exec unit unit-3-1 show cap CAPO packet-number 2 trace
```

```
...
```

```
2: 09:19:49.761008 192.168.240.51.80 > 192.168.240.50.59210:
```

```
s
```

```
4209225081:4209225081(0) ack 4110299696 win 28960 <mss 1460,sackOK,timestamp 567715984 130834570,nop,w
```

```
Phase: 1
```

```
Type: CLUSTER-EVENT
```

```
Subtype:
```

```
Result: ALLOW
```

```
Config:
```

```
Additional Information:
```

```
Input interface: 'OUTSIDE'
```

```
Flow type: STUB
```

```
I (2) am becoming forwarder to (1), sender (0).
```

所有者重新将数据包转发到目的地：

```
<#root>
```

```
firepower#
```

```
cluster exec unit unit-2-1 show cap CAPO packet-number 2 trace
```

```
2: 09:19:49.775701 192.168.240.51.80 > 192.168.240.50.59210:
```

```
s
```

```
4209225081:4209225081(0)
```

```
ack
```

```
4110299696 win 28960 <mss 1460,sackOK,timestamp 567715984 130834570,nop,wscale 7>
```

```
Phase: 1
```

```
Type: CLUSTER-EVENT
```

```
Subtype:
```

```
Result: ALLOW
```

```
Config:
```

```
Additional Information:
```

```
Input interface: 'OUTSIDE'
```

```
Flow type: FULL
```

```
I (1) am owner, sender (2).
```

观察4. FTD数据平面系统日志显示所有设备上的连接创建和终止：

- Unit-1-1 (导向器/备份所有者)

- Unit-2-1 (所有者)
- Unit-3-1 (转发器)

<#root>

firepower#

```
cluster exec show log | i 59210
```

```
unit-1-1(LOCAL):*****
```

```
Dec 03 2020 09:19:49: %FTD-6-302022:
```

```
Built director stub TCP connection
```

```
for INSIDE:192.168.240.50/59210 (192.168.240.50/59210) to OUTSIDE:192.168.240.51/80 (192.168.240.51/80)
```

```
Dec 03 2020 09:19:59: %FTD-6-302023:
```

```
Teardown director TCP connection
```

```
for INSIDE:192.168.240.50/59210 to OUTSIDE:192.168.240.51/80 duration 0:00:09 forwarded bytes 0 Cluste
```

```
unit-2-1:*****
```

```
Dec 03 2020 09:19:49: %FTD-6-302303:
```

```
Built TCP state-bypass connection
```

```
14483 from INSIDE:192.168.240.50/59210 (192.168.240.50/59210) to OUTSIDE:192.168.240.51/80 (192.168.240.51/80)
```

```
Dec 03 2020 09:19:59: %FTD-6-302304:
```

```
Teardown TCP state-bypass connection
```

```
14483 from INSIDE:192.168.240.50/59210 to OUTSIDE:192.168.240.51/80 duration 0:00:09 bytes 1024003336
```

```
unit-3-1:*****
```

```
Dec 03 2020 09:19:49: %FTD-6-302022:
```

```
Built forwarder stub TCP connection
```

```
for OUTSIDE:192.168.240.51/80 (192.168.240.51/80) to unknown:192.168.240.50/59210 (192.168.240.50/59210)
```

```
Dec 03 2020 09:19:59: %FTD-6-302023:
```

```
Teardown forwarder TCP connection
```

```
for OUTSIDE:192.168.240.51/80 to unknown:192.168.240.50/59210 duration 0:00:09 forwarded bytes 1024003
```

故障排除

集群故障排除简介

群集问题可分类为：

- 控制平面问题 (与集群稳定性相关的问题)
- 数据平面问题 (与中转流量相关的问题)

集群数据平面问题

NAT/PAT常见问题

重要配置注意事项

- 端口地址转换(PAT)池的可用IP数必须至少与集群中的设备数相同，最好是比集群节点更多IP。
- 除非有特定原因禁用默认xlate per-session命令，否则必须保留这些命令。为禁用了xlate per-session的连接建立的任何PAT转换始终由集群中的控制节点单元处理，这可能导致性能降低。

高PAT池范围使用率，因为源自低端口的流量会导致集群IP不平衡

FTD将PAT IP划分为多个范围，并尝试将xlate保持在相同的源范围内。下表显示了源端口如何转换为同一源范围内的全局端口。

原始源端口	转换后的Src端口
1-511	1-511
512-1023	512-1023
1024-65535	1024-65535

当源端口范围已满并且需要从该范围分配新的PAT转换时，FTD将移至下一个IP以为该源端口范围分配新的转换。

症状

通过集群的NAT流量的连接问题

确认

```
<#root>
```

```
#
```

```
show nat pool
```

FTD数据平面日志显示PAT池耗尽：

```
<#root>
```

```
Dec 9 09:00:00 192.0.2.10 FTD-FW %ASA-3-202010:
```

```
PAT pool exhausted. Unable to create TCP connection

from Inside:192.0.2.150/49464 to Outside:192.0.2.250/20015
Dec 9 09:00:00 192.0.2.10 FTD-FW %ASA-3-202010:

PAT pool exhausted. Unable to create TCP connection

from Inside:192.0.2.148/54141 to Outside:192.0.2.251/443
```

缓解

配置NAT平面端口范围并包括保留端口。

此外，在6.7/9.15.1之后，只有在节点离开/加入具有大量PAT背景流量的集群时，您才可能最终获得不均衡的端口块分配。它自己恢复的唯一方式是释放端口块以在节点间重新分配。

使用基于端口块的分配，当节点分配了大约10个端口块（如pb-1、pb-2...pb-10）时，节点始终从第一个可用端口块开始并分配一个随机端口，直到其耗尽。仅当所有端口块都用尽到那一点时，分配才会移至下一个端口块。

例如，如果主机建立512个连接，则设备会随机为pb-1的所有这512个连接分配映射端口。现在，当所有这些512连接都处于活动状态时，当主机建立自pb-1耗尽以来的第513个连接时，它会移动到pb-2并从其分配一个随机端口。现在，在513个连接中，假设第10个连接完成并清除了pb-1中的一个可用端口。此时，如果主机建立第514个连接，集群单元将从pb-1而不是pb-2分配映射端口，因为pb-1现在有一个空闲端口（在第10个连接删除过程中释放了该端口）。

需要记住的重要一点是，分配是从具有空闲端口的第一个可用端口块开始的，这样在正常加载的系统中，最后一个端口块始终可用于重分发。此外，PAT通常用于短期连接。端口块在短时间内变为可用状态的概率非常高。因此，使用基于端口块的池分配，可以缩短池分配达到平衡所需的时间。

但是，如果所有端口块（从pb-1到pb-10）耗尽，或者每个端口块都保存一个用于长期连接的端口，则这些端口块永远不会快速释放并重新分配。在这种情况下，破坏性最小的方法是：

1. 识别具有过多端口块的节点(show nat pool cluster summary)。
2. 确定该节点上使用率最低的端口块(show nat pool ip <addr> detail)。
3. 清除此类端口块的xlates(clear xlate global <addr> gport 'start-end')，使其可用于重分发。

 **警告：**这会中断相关连接。

当重定向到其他目标时，无法浏览到双通道网站（如网络邮件、银行等）或SSO网站。

症状

无法浏览到双渠道网站（例如网络邮件、银行网站等）。当用户连接到要求客户端打开第二个套接字/连接的网站时，如果第二个连接被哈希到与获得第一个连接时的群集成员不同的群集成员，并且流量使用IP PAT池，则当流量从其他公共IP地址接收连接时，服务器会重置流量。

确认

进行数据平面集群捕获，查看如何处理受影响的传输流。在这种情况下，TCP重置来自目标网站。

缓解 (6.7/9.15.1之前的版本)

- 观察是否有任何多会话应用程序使用多个映射IP地址。
- 使用show nat pool cluster summary命令检查池是否均匀分布。
- 使用cluster exec show conn命令检查流量是否正确进行了负载均衡。
- 使用show nat pool cluster ip <address> detail命令检查粘滞IP的池使用情况。
- 启用syslog 305021(6.7/9.15)以查看哪些连接未能使用粘滞IP。
- 要解决向PAT池添加更多IP或微调已连接交换机上的负载均衡算法。

关于以太网通道负载均衡算法：

- 对于非FP9300，如果身份验证通过一台服务器发生：调整相邻交换机上从源IP/端口和目标IP/端口到源IP和目标IP的以太网通道负载均衡算法。
- 对于非FP9300，如果身份验证通过多个服务器进行：调整相邻交换机上从源IP/端口和目标IP/端口到源IP的以太网通道负载均衡算法。
- 对于FP9300:在FP9300机箱上，负载均衡算法固定为source-dest-port source-dest-ip source-dest-mac，并且不能更改。在这种情况下，解决方法是使用FlexConfig向FTD配置中添加xlate per-session deny命令，强制某些目标IP地址（对于有问题/不兼容的应用程序）的流量仅由机箱内集群中的控制节点处理。解决方法伴随以下副作用：
 - 不同转换流量的负载均衡（所有流量都流向控制节点）。
 - xlate插槽可能用尽（并对控制节点上其他流量的NAT转换产生负面影响）。
 - 降低机箱内群集的可扩展性。

由于池中的PAT IP不足，所有流量都发送到控制节点，因此集群性能较低。

症状

集群中的PAT IP不足，无法向数据节点分配空闲IP，因此，所有受PAT配置约束的流量都会转发到控制节点进行处理。

确认

使用show nat pool cluster命令查看每台设备的分配，并确认它们都至少拥有池中的一个IP。

缓解

对于6.7/9.15.1之前的版本，请确保您的PAT池大小至少等于集群中的节点数。在具有PAT池的6.7/9.15.1之后的版本中，您可以从所有PAT池IP分配端口块。如果PAT池使用率确实很高，导致池频繁耗尽，您需要增加PAT池大小（请参阅FAQ部分）。

由于未启用每个会话，因此所有流量都发送到控制节点，因此性能较低。

症状

通过集群控制节点处理大量高速UDP备份流量，这会影响性能。

背景

只有使用xlates且已启用每个会话的连接才能由使用PAT的数据节点处理。使用命令show run all xlate查看xlate per-session配置。

启用每个会话意味着当关联的连接断开时，会立即关闭xlate。这有助于在连接采用PAT时提高每秒连接性能。在关联的连接断开后，非每会话状态将再持续30秒，如果连接速率足够高，则每个全局IP上可用的65k TCP/UDP端口可以在短时间内用完。

默认情况下，所有TCP流量都启用每会话，只有UDP DNS流量启用每会话。这意味着所有非DNS UDP流量都被转发到控制节点进行处理。

确认

使用此命令可检查集群设备之间的连接和数据包分布：

```
<#root>
```

```
firepower#
```

```
show cluster info conn-distribution
```

```
firepower#
```

```
show cluster info packet-distribution
```

```
firepower#
```

```
show cluster info load-monitor
```

使用cluster exec show conn命令查看哪些集群节点拥有UDP连接。

```
<#root>
```

```
firepower#
```

```
cluster exec show conn
```

使用此命令可以了解群集节点间的池使用情况。

```
<#root>
```

```
firepower#
```

```
cluster exec show nat pool ip
```

缓解

为相关流量（例如UDP）配置每会话PAT(per-session permit udp命令)。对于ICMP，您不能更改默认的多会话PAT，因此，当配置了PAT时，ICMP流量始终由控制节点处理。

当节点离开/加入集群时，PAT池分布变得不平衡。

症状

- 连接问题，因为PAT IP分配可能由于设备离开和加入集群而随时间变得不平衡。
- 在6.7/9.15.1之后，可能出现新加入的节点无法获得足够端口块的情况。没有任何端口块的节点将流量重定向到控制节点。至少有一个端口块的节点会处理流量，并在池耗尽后丢弃该流量。

确认

- 数据平面系统日志显示如下消息：

<#root>

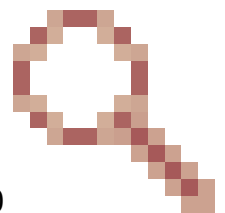
```
%ASA-3-202010:
```

```
NAT pool exhausted. Unable to create TCP connection  
from inside:192.0.2.1/2239 to outside:192.0.2.150/80
```

- 使用show nat pool cluster summary命令确定池分布。
- 使用cluster exec show nat pool ip <addr> detail命令了解集群节点间的池使用情况。

缓解

- 对于6.7/9.15.1之前的版本，Cisco Bug ID [CSCvd](#)中介绍了一些解决方法10530。
- 在6.7/9.15.1之后的版本中，使用clear xlate global <ip> gport <start-end>命令手动清除其他节点上的某些端口块，以重新分发到所需节点。



症状

集群通过PAT传输的流量的主要连接问题。这是因为FTD数据平面按设计不发送全局NAT地址的GARP。

确认

直连设备的ARP表显示了更改控制节点后集群数据接口的不同MAC地址：

```
<#root>
```

```
root@kali2:~/tests#
```

```
arp -a
```

```
? (192.168.240.1) at f4:db:e6:
```

```
33:44:2e
```

```
[ether] on eth0
```

```
root@kali2:~/tests#
```

```
arp -a
```

```
? (192.168.240.1) at f4:db:e6:
```

```
9e:3d:0e
```

```
[ether] on eth0
```

缓解

在集群数据接口上配置静态（虚拟）MAC。

受PAT影响的连接失败

症状

集群通过PAT传输的流量的连接问题。

验证/缓解

- 确保正确复制配置。
- 确保池均匀分布。
- 确保池所有权有效。
- show asp cluster counter中没有故障计数器增量。
- 确保使用正确信息创建导向器/转发器流。
- 验证是否按预期创建、更新和清理了备份副本。
- 验证是否根据“每个会话”行为创建和终止xlates。
- 启用“debug nat 2”可指示任何错误。请注意，此输出可能非常嘈杂，例如：

```
<#root>
```

```
firepower#
```

```
debug nat 2
```

```
nat:
no free blocks available to reserve for 192.168.241.59, proto 17
```

```
nat: no free blocks available to reserve for 192.168.241.59, proto 17
nat: no free blocks available to reserve for 192.168.241.58, proto 17
nat: no free blocks available to reserve for 192.168.241.58, proto 17
nat: no free blocks available to reserve for 192.168.241.57, proto 17
```

要停止调试，请执行以下操作：

```
<#root>
firepower#
un all
```

- 启用连接和NAT相关的系统日志以将信息关联到故障连接。

ASA和FTD集群PAT改进 (9.15和6.7之后)

发生了什么变化？

重新设计了PAT操作。单个IP不再分配到每个集群成员。相反，PAT IP被拆分为端口块，并结合IP粘性操作在集群成员之间均匀 (尽可能) 分配这些端口块。

新设计解决了这些限制 (请参阅上一节)：

- 多会话应用会因缺乏集群范围的IP粘性而受到影响。
- 要求的PAT池的大小至少等于集群中的节点数。
- 当节点离开/加入集群时，PAT池分布变得不平衡。
- 无系统日志指示PAT池不平衡。

从技术上讲，PAT的默认端口范围是1024-65535，而不是默认的1-511、512-1023和1024-65535端口范围。此默认范围可以扩展为包括常规PAT的特权端口范围1-1023 (“include-reserve”选项)。

这是FTD 6.7上的PAT池配置示例。有关其他详细信息，请查看《配置指南》中的相关部分：

NAT Rule:
Manual NAT Rule

Insert:
In Category NAT Rules Before

Type:
Dynamic

Enable

Description:

Interface Objects Translation PAT Pool Advanced

Original Packet	Translated Packet
Original Source:* net_192.168.240.0 +	Translated Source: Address
Original Destination: Address +	+ +
Original Source Port: + +	Translated Destination: + +
Original Destination Port: + +	Translated Source Port: + +
	Translated Destination Port: + +

Interface Objects Translation PAT Pool Advanced

Enable PAT Pool

PAT:
Address ip_192.168.241.57-59 +

Use Round Robin Allocation

Extended PAT Table

Flat Port Range ⓘ This option always enabled on device from v6.7.0 irrespective of its configured value.

Include Reserve Ports

Block Allocation

有关PAT的其他故障排除信息

FTD数据平面系统日志 (6.7/9.15.1之后)

当集群节点的粘滞IP中的所有端口用尽时，系统会生成粘性失效系统日志，并且分配会移至具有空

闲端口的下一个可用IP，例如：

```
%ASA-4-305021: Ports exhausted in pre-allocated PAT pool IP 192.0.2.100 for host 198.51.100.100 Allocat
```

当节点加入集群时，会在节点上生成池不平衡系统日志，并且不会获得任何端口块或不等份额的端口块，例如：

```
%ASA-4-305022: Cluster unit ASA-4 has been allocated 0 port blocks for PAT usage. All units should have  
%ASA-4-305022: Cluster unit ASA-4 has been allocated 12 port blocks for PAT usage. All units should have
```

显示命令

池分布状态

在show nat pool cluster summary输出中，对于每个PAT IP地址，在平衡分配方案中，各节点之间的端口块差异不得超过1个。均衡和不均衡的端口块分布的示例。

```
<#root>
```

```
firepower#
```

```
show nat pool cluster summary
```

```
port-blocks count display order: total, unit-1-1, unit-2-1, unit-3-1
```

```
IP OUTSIDE:ip_192.168.241.57-59 192.168.241.57 (126 -
```

```
42 / 42 / 42
```

```
)
```

```
IP OUTSIDE:ip_192.168.241.57-59 192.168.241.58 (126 - 42 / 42 / 42)
```

```
IP OUTSIDE:ip_192.168.241.57-59 192.168.241.59 (126 - 42 / 42 / 42)
```

分布不平衡：

```
<#root>
```

```
firepower#
```

```
show nat pool cluster summary
```

```
port-blocks count display order: total, unit-1-1, unit-4-1, unit-2-1, unit-3-1
```

```
IP outside:src_map 192.0.2.100 (128 - 32 /
```

```
22 / 38
```

```
/ 36)
```

池所有权状态

在show nat pool cluster输出中，不得存在所有者或备份为UNKNOWN的单个端口块。如果存在，则表明池所有权通信存在问题。示例：

```
<#root>
```

```
firepower#
```

```
show nat pool cluster | in
```

```
[3072-3583], owner unit-4-1, backup <
```

```
UNKNOWN
```

```
>
```

```
[56832-57343], owner <UNKNOWN>, backup <UNKNOWN>
```

```
[10240-10751], owner unit-2-1, backup <UNKNOWN>
```

端口块中端口分配的记帐

show nat pool命令通过其他选项得到增强，这些选项用于显示详细信息和过滤后的输出。示例：

```
<#root>
```

```
firepower#
```

```
show nat pool detail
```

```
TCP PAT pool INSIDE, address 192.168.240.1, range 1-1023, allocated 0
```

```
TCP PAT pool INSIDE, address 192.168.240.1, range 1024-65535, allocated 18
```

```
UDP PAT pool INSIDE, address 192.168.240.1, range 1-1023, allocated 0
```

```
UDP PAT pool INSIDE, address 192.168.240.1, range 1024-65535, allocated 20
```

```
TCP PAT pool OUTSIDE, address 192.168.241.1, range 1-1023, allocated 0
```

```
TCP PAT pool OUTSIDE, address 192.168.241.1, range 1024-65535, allocated 18
```

```
UDP PAT pool OUTSIDE, address 192.168.241.1, range 1-1023, allocated 0
```

```
UDP PAT pool OUTSIDE, address 192.168.241.1, range 1024-65535, allocated 20
```

```
UDP PAT pool OUTSIDE, address 192.168.241.58
```

```
range 1024-1535, allocated 512
```

```
range 1536-2047, allocated 512
```

```
range 2048-2559, allocated 512
```

```
range 2560-3071, allocated 512
```

```
...
```

```
unit-2-1:*****
```

```
UDP PAT pool OUTSIDE, address 192.168.241.57
```

```
range 1024-1535, allocated 512 *
```

```
range 1536-2047, allocated 512 *
```

```
range 2048-2559, allocated 512 *
```

“*”表示它是备份端口块

要解决此问题，请使用clear xlate global <ip> gport <start-end>命令手动清除其他节点上的某些端口块，以便重新分配到所需节点。

手动触发端口块重分发

- 在具有恒定流量的生产网络中，当节点离开并重新加入集群时（可能由于回溯），有时它无法获得池的相等份额，或者在最坏的情况下，它无法获得任何端口块。
- 使用show nat pool cluster summary命令确定哪个节点拥有的端口块多于所需数量。
- 在拥有更多端口块的节点上，使用show nat pool ip <addr> detail命令找出分配数量最少的端口块。
- 使用clear xlate global <address> gport <start-end>命令清除从这些端口块创建的转换，以便它们可用于重分发到所需节点，例如：

```
<#root>
```

```
firepower#
```

```
show nat pool detail | i 19968
```

```
range 19968-20479, allocated 512
range 19968-20479, allocated 512
range 19968-20479, allocated 512
```

```
firepower#
```

```
clear xlate global 192.168.241.57 gport 19968-20479
```

```
INFO: 1074 xlates deleted
```

6.7/9.15.1之后PAT的常见问题(FAQ)

问：如果集群中的可用设备数量已达到可用的IP数量，是否仍可以将每台设备的1个IP用作选项？

答：现在已不再如此，也没有在基于IP地址和基于端口块的地址池分配方案之间进行切换的功能。

基于IP地址的池分配旧方案导致多会话应用故障，来自主机的多个连接（属于单个应用事务的一部分）被负载均衡到集群的不同节点上，从而被不同的映射IP地址转换，导致目标服务器看到它们来自不同的实体。

此外，使用基于端口块的新分配方案，即使您现在可以使用低至单个PAT IP地址，也始终建议根据需要PAT的连接数量使用足够的PAT IP地址。

问：是否仍可以为该集群的PAT池保留一个IP地址池？

是的，你可以。来自所有PAT池IP的端口块分布到群集节点上。

问：如果对PAT池使用多个IP地址，是否为每个IP地址分配给每个成员的不同端口块？

答：不，每个IP都是独立分配的。

问：所有群集节点都有所有公有IP，但只有一部分端口？如果是这种情况，能否保证每次源IP使用相同的公共IP？

A.正确，每个PAT IP由每个节点部分拥有。如果某个节点上已用完所选公有IP，则会生成系统日志，指示无法保留粘滞IP，并且分配将移至下一个可用的公有IP。无论是独立、高可用性还是集群部署，IP粘性始终以尽力而为的方式进行，具体取决于池可用性。

问：所有内容是否都基于PAT池中的单个IP地址，但如果使用PAT池中的多个IP地址，则不适用？

A.它也适用于PAT池中的多个IP地址。来自PAT池中每个IP的端口块分布于群集节点。PAT池中的每个IP地址在集群中的所有成员之间拆分。因此，如果您在PAT池中有一个C类地址，则每个集群成员都有来自每个PAT池地址的端口池。

它适用于CGNAT吗？

A.是的，CGNAT同样受支持。CGNAT（也称为块分配PAT）的默认块大小为“512”，可以通过xlate块分配大小CLI进行修改。对于常规动态PAT（非CGNAT），块大小始终为“512”，这是固定且不可配置的。

问：如果设备离开集群，控制节点会将端口块范围分配给其他设备还是保留给自己？

A.每个端口块都有一个所有者和备份。每次从端口块创建xlate时，它也会复制到端口块备份节点。当节点离开集群时，备份节点拥有所有端口块和所有当前连接。由于备份节点已成为这些附加端口块的所有者，因此会为其选择新的备份，并将所有当前副本复制到该节点，以处理故障情况。

根据这个警告，可以采取什么行动来增强粘性？

有两种可能的原因可以解释为什么不能保持粘性。

原因1:流量未正确进行负载均衡，因为其中一个节点看到的连接数量比其他节点多，从而导致特定的粘滞IP耗尽。如果确保流量在集群节点间均匀分布，则可以解决此问题。例如，在FPR41xx集群上，调整已连接交换机上的负载均衡算法。在FPR9300集群上，确保机箱上的刀片数量相等。

原因2: PAT池使用率非常高，导致池频繁耗尽。要解决此问题，请增加PAT池大小。

问：如何处理对extended关键字的支持？它是否显示错误，并阻止在升级期间添加整个NAT命令，还是删除extended关键字并显示警告？

A.从ASA 9.15.1/FP 6.7开始的集群不支持PAT扩展选项。配置选项不会从任何CLI/ASDM/CSM/FMC中删除。配置（通过升级直接或间接配置）时，系统会通知您一条警告消息，并且会接受配置，但您不会看到正在运行的PAT的扩展功能。

问：转换数是否与并发连接数相同？

A.在6.7/9.15.1之前版本中，尽管它是1-65535，因为源端口在1-1024范围内从未被大量使用，但它实际上是1024-65535(64512个连接)。在6.7/9.15.1之后的实施中，将“flat”作为默认行为，其值为

1024-65535。但是，如果您要使用1-1024，则可以使用“include-reserve”选项。

问：如果节点重新加入集群，它会将旧的备份节点作为备份，而该备份节点会为其提供其旧的端口块？

A.这取决于当时端口块的可用性。当节点离开集群时，其所有端口块都将移至备份节点。然后，控制节点将累积空闲端口块并将其分发到所需节点。

问：如果控制节点的状态发生变化，是否选择新的控制节点，是保持PAT块分配，还是基于新的控制节点重新分配端口块？

A.新控制节点了解已分配哪些块，哪些块是免费的，哪些是从中开始的。

问：xlates的最大数量与具有此新行为的最大并发连接数量是否相同？

答：是的。xlates的最大数量取决于PAT端口的可用性。这与最大并发连接数无关。如果仅允许1个地址，则可能有65535个连接。如果您需要更多，则必须分配更多IP地址。如果有足够的地址/端口，您可以达到最大并发连接数。

问：添加新的集群成员时，端口块分配过程是什么？如果由于重新启动而添加了集群成员，会发生什么情况？

A.端口块始终由控制节点分配。只有存在空闲端口块时，端口块才会分配给新节点。自由端口块表示不通过端口块中的任何映射端口提供连接。

此外，在重新加入时，每个节点会重新计算其可拥有的块数。如果节点拥有的块数超出其预期数量，它会在这些端口块可用时将其释放给控制节点。然后，控制节点将它们分配给新加入的数据节点。

问：它是否只支持TCP和UDP协议或SCTP？

A.动态PAT从未支持SCTP。对于SCTP流量，建议仅使用静态网络对象NAT。

问：如果某个节点的块端口耗尽，它是否会丢弃数据包，而不使用下一个可用的IP块？

不，它不会立即掉下来。它使用来自下一个PAT IP的可用端口块。如果所有PAT IP上的所有端口块均已用尽，则会丢弃流量。

问：为了避免集群升级窗口中控制节点的过载，是否最好提前手动选择新的控制（例如，在4单元集群升级的中途），而不是等待控制节点上处理所有连接？

A.控件必须最后更新。这是因为，当控制节点运行较新版本时，除非所有节点都运行较新版本，否则它不会启动池分配。此外，当升级运行时，如果控制节点运行的是旧版本，则所有新版本的数据节点会忽略来自该控制节点的池分布消息。

要详细解释这一点，请考虑以4节点A、B、C和D为控制节点的集群部署。以下是典型的无中断升级步骤：

1. 将新版本下载到每个节点。
2. 重新加载设备“D”。所有连接、xlates都将移至备份节点。

3. 单位“D”出现，并且：

a.处理PAT配置

b.将每个PAT IP分成端口块

c.使所有端口块处于未分配状态

d.忽略从控件接收的较旧版本的集群PAT消息

e.将所有PAT连接重定向到主连接。

4.同样，使用新版本启动其他节点。

5.重新加载设备'A'控件。由于没有控制备份，所有现有连接都会被丢弃

6.新控件开始以较新的格式分发端口块

7.设备'A'重新加入，能够接受端口块分发消息并对其执行操作

分段处理

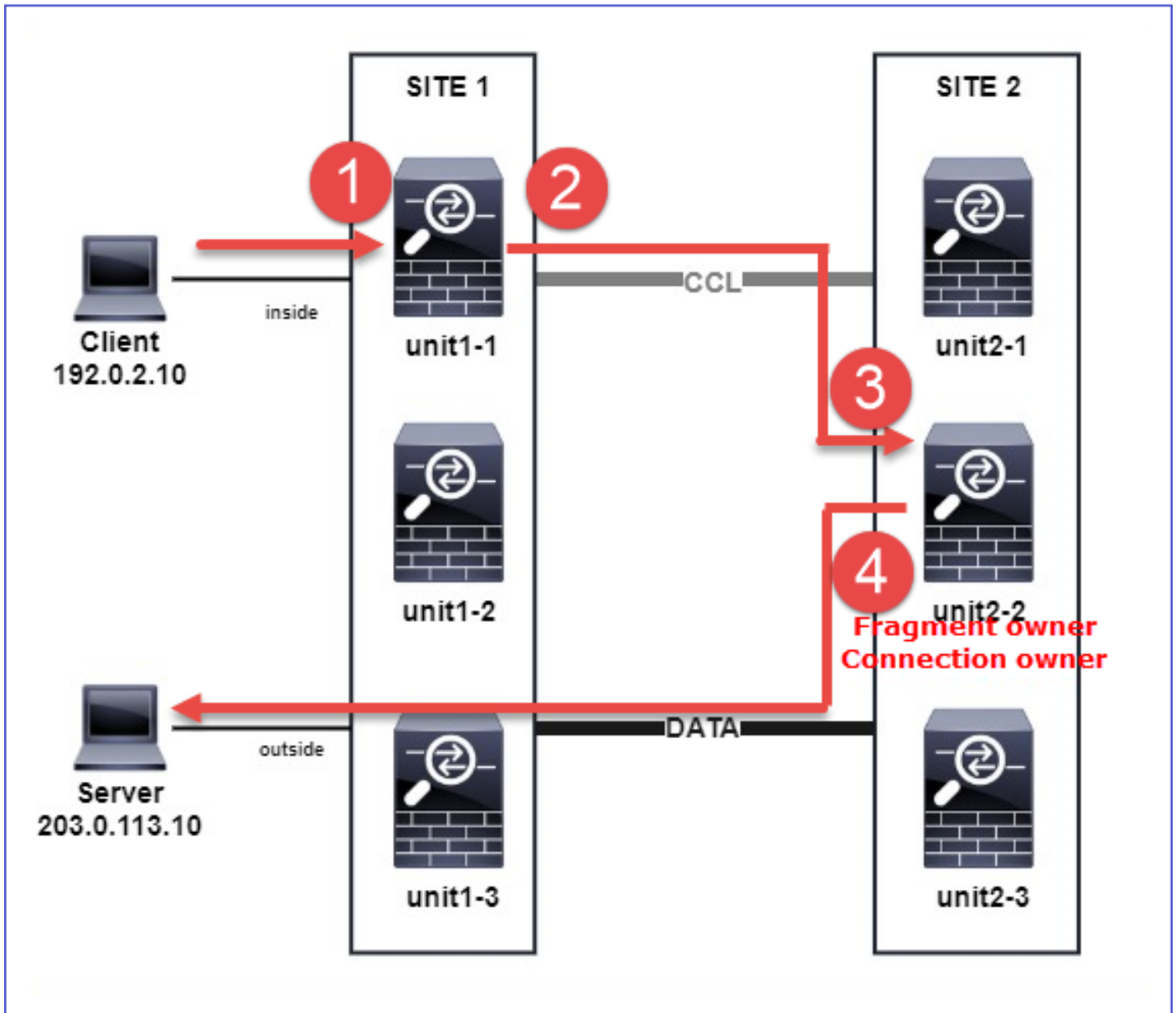
症状

在站点间集群部署中，必须在一个特定站点（站点本地流量）中处理的分段数据包仍然可以发送到其他站点的设备，因为这些站点之一可以拥有分段所有者。

在集群逻辑中，为具有分段数据包的连接定义了一个附加角色：分段所有者。

对于分段数据包，接收分段的集群单元根据分段的源IP地址、目标IP地址和数据包ID的散列确定分段所有者。然后，所有分段都通过集群控制链路转发给分段所有者。分段可以负载平衡到不同的集群单元，因为只有第一个分段包含交换机负载平衡哈希中使用的5元组。其他分段不包含源端口和目的端口，可以负载平衡到其他集群设备。分段所有者临时重组数据包，以便根据源/目标IP地址和端口的散列值确定指挥交换机。如果是新连接，则分段所有者将成为连接所有者。如果它是现有连接，则分段所有者将通过集群控制链路将所有分段转发给连接所有者。然后，连接所有者重组所有分段。

考虑以下拓扑，其中含有从客户端到服务器的分段ICMP回应请求：



为了了解操作的顺序，在内部接口、外部接口和集群控制链路接口上配置了跟踪选项，从而捕获集群范围的数据包。此外，在内部接口上配置了具有reject-hide选项的数据包捕获。

```
<#root>
```

```
firepower#
```

```
cluster exec capture capi interface inside trace match icmp any any
```

```
firepower#
```

```
cluster exec capture capir interface inside reinject-hide trace match icmp any any
```

```
firepower#
```

```
cluster exec capture capo interface outside trace match icmp any any
```

```
firepower#
```

```
cluster exec capture capccl interface cluster trace match icmp any any
```

集群内的操作顺序：

1. 站点1中的unit-1-1接收分段的ICMP回应请求数据包。

```
<#root>
```

```
firepower#
```

```
cluster exec show cap capir
```

```
unit-1-1(LOCAL)
```

```
:*****
```

```
2 packets captured
```

```
1: 20:13:58.227801 802.1Q vlan#10 P0 192.0.2.10 > 203.0.113.10 icmp: echo request
```

```
2: 20:13:58.227832 802.1Q vlan#10 P0
```

```
2 packets shown
```

2. unit-1-1选择站点2中的unit-2-2作为分段所有者，并向其发送分段数据包。

从unit-1-1发送到unit-2-2的数据包的目标MAC地址是unit-2-2中CCL链路的MAC地址。

```
<#root>
```

```
firepower#
```

```
show cap capccl packet-number 1 detail
```

```
7 packets captured
```

```
1: 20:13:58.227817
```

```
0015.c500.018f 0015.c500.029f
```

```
0x0800 Length: 1509
```

```
192.0.2.10 > 203.0.113.10
```

```
icmp: echo request (wrong icmp csum) (frag 46772:1475@0+) (ttl 3)
```

```
1 packet shown
```

```
firepower#
```

```
show cap capccl packet-number 2 detail
```


7 packets captured

2: 20:13:58.227832

0015.c500.018f 0015.c500.029f

0x0800 Length: 637

192.0.2.10 > 203.0.113.10

(

frag 46772

:603@1480) (ttl 3)

1 packet shown

firepower#

cluster exec show interface po48 | i MAC

unit-1-1(LOCAL):*****

MAC address 0015.c500.018f, MTU 1500

unit-1-2:*****

MAC address 0015.c500.019f, MTU 1500

unit-2-2

:*****

MAC address 0015.c500.029f, MTU 1500

unit-1-3:*****

MAC address 0015.c500.016f, MTU 1500

unit-2-1:*****

MAC address 0015.c500.028f, MTU 1500

unit-2-3:*****

MAC address 0015.c500.026f, MTU 1500

3.unit-2-2接收、重组分段的数据包，并成为流的所有者。

<#root>

firepower#

cluster exec unit unit-2-2 show capture capcc1 packet-number 1 trace

11 packets captured

1: 20:13:58.231845 192.0.2.10 > 203.0.113.10 icmp: echo request

Phase: 1

Type: CLUSTER-EVENT

Subtype:
Result: ALLOW
Config:
Additional Information:
Input interface: 'inside'
Flow type: NO FLOW

I (2) received a FWD_FRAG_TO_FRAG_OWNER from (0).

Phase: 2
Type: CLUSTER-EVENT
Subtype:
Result: ALLOW
Config:
Additional Information:
Input interface: 'inside'

Flow type: NO FLOW

I (2) have reassembled a packet and am processing it.

Phase: 3
Type: CAPTURE
Subtype:
Result: ALLOW
Config:
Additional Information:
MAC Access list

Phase: 4
Type: ACCESS-LIST
Subtype:
Result: ALLOW
Config:
Implicit Rule
Additional Information:
MAC Access list

Phase: 5
Type: ROUTE-LOOKUP
Subtype: No ECMP load balancing
Result: ALLOW
Config:
Additional Information:
Destination is locally connected. No ECMP load balancing.
Found next-hop 203.0.113.10 using egress ifc outside(vrfid:0)

Phase: 6
Type: CLUSTER-EVENT
Subtype:
Result: ALLOW
Config:
Additional Information:
Input interface: 'inside'

Flow type: NO FLOW

I (2) am becoming owner

Phase: 7

Type: ACCESS-LIST

Subtype: log

Result: ALLOW

Config:

access-group CSM_FW_ACL_ global

access-list CSM_FW_ACL_ advanced trust ip any any rule-id 268435460 event-log flow-end

access-list CSM_FW_ACL_ remark rule-id 268435460: PREFILTER POLICY: igasimov_prefilter1

access-list CSM_FW_ACL_ remark rule-id 268435460: RULE: r1

Additional Information:

...

Phase: 19

Type: FLOW-CREATION

Subtype:

Result: ALLOW

Config:

Additional Information:

New flow created with id 1719, packet dispatched to next module

...

Result:

input-interface: cluster(vrfid:0)

input-status: up

input-line-status: up

output-interface: outside(vrfid:0)

output-status: up

output-line-status: up

Action: allow

1 packet shown

firepower#

cluster exec unit unit-2-2 show capture capccl packet-number 2 trace

11 packets captured

2: 20:13:58.231875

Phase: 1

Type: CLUSTER-EVENT

Subtype:

Result: ALLOW

Config:

Additional Information:

Input interface: 'inside'

Flow type: NO FLOW

I (2) received a FWD_FRAG_TO_FRAG_OWNER from (0).

Result:

input-interface: cluster(vrfid:0)
input-status: up
input-line-status: up
Action: allow

1 packet shown

4.unit-2-2根据安全策略允许数据包，并通过外部接口将它们从站点2发送到站点1。

<#root>

firepower#

cluster exec unit unit-2-2 show cap capo

2 packets captured

1: 20:13:58.232058 802.1Q vlan#20 P0 192.0.2.10 > 203.0.113.10 icmp: echo request

2: 20:13:58.232058 802.1Q vlan#20 P0

观察/警告

- 与指挥交换机角色不同，片段所有者不能本地化为特定站点。片段所有者由最初接收新连接的分段数据包的设备确定，可以位于任何站点。
- 由于分段所有者也可以成为连接所有者，因此为了将数据包转发到目标主机，它必须能够解析出口接口，并查找目标主机或下一跳的IP和MAC地址。这假定下一跳也必须能够到达目的主机。
- 要重组分段数据包，ASA/FTD会维护每个命名接口的IP分段重组模块。要显示IP分段重组模块的运行数据，请使用show fragment命令：

<#root>

Interface: inside
Configuration:

size: 200

, Chain: 24, Timeout: 5, Reassembly: virtual
Run-time stats: Queue: 0, Full assembly: 0

```
Drops: Size overflow: 0, Timeout: 0,  
Chain overflow: 0, Fragment queue threshold exceeded: 0,  
Small fragments: 0, Invalid IP len: 0,  
Reassembly overlap: 0, Fraghead alloc failed: 0,  
SGT mismatch: 0, Block alloc failed: 0,  
Invalid IPV6 header: 0, Passenger flow assembly failed: 0
```

在集群部署中，分段所有者或连接所有者将分段的数据包放入分段队列。分段队列大小受使用 `fragment size <size> <nameif>` 命令配置的 Size 计数器的值限制（默认为 200）。当分段队列大小达到大小的 2/3 时，会认为超出分段队列阈值，并且会丢弃不属于当前分段链一部分的任何新分段。在这种情况下，超出分段队列阈值将递增，并生成系统日志消息 FTD-3-209006。

```
<#root>
```

```
firepower#
```

```
show fragment inside
```

```
Interface: inside
```

```
Configuration:
```

```
Size: 200
```

```
, Chain: 24, Timeout: 5, Reassembly: virtual
```

```
Run-time stats:
```

```
Queue: 133
```

```
, Full assembly: 0
```

```
Drops: Size overflow: 0, Timeout: 8178,
```

```
Chain overflow: 0,
```

```
Fragment queue threshold exceeded: 40802
```

```
,  
Small fragments: 0, Invalid IP len: 0,  
Reassembly overlap: 9673, Fraghead alloc failed: 0,  
SGT mismatch: 0, Block alloc failed: 0,  
Invalid IPV6 header: 0, Passenger flow assembly failed: 0
```

```
%FTD-3-209006: Fragment queue threshold exceeded, dropped TCP fragment from 192.0.2.10/21456 to 203.0.113.1
```

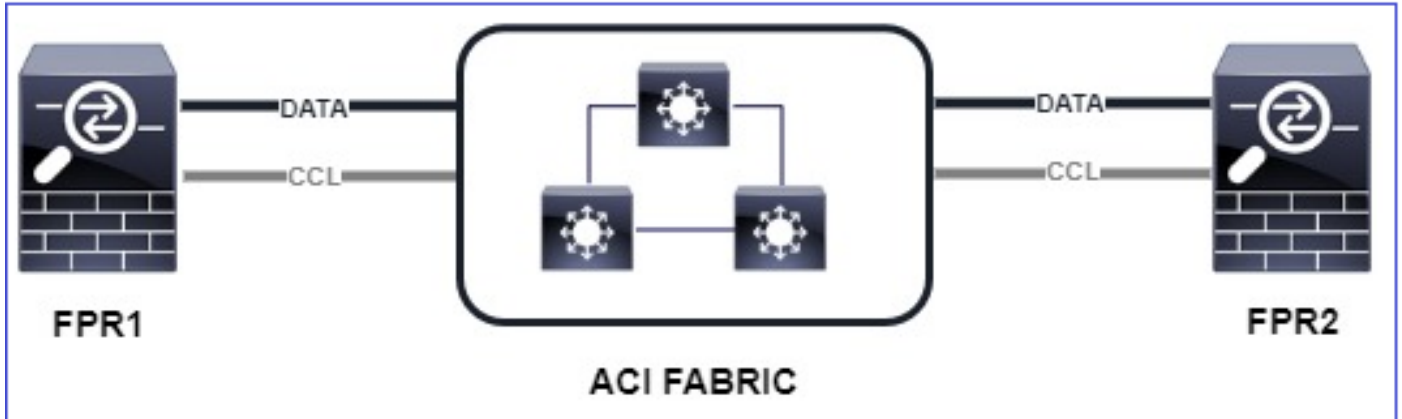
作为解决方法，请在 Firepower Management Center > Devices > Device Management > [Edit Device] > Interfaces > [Interface] > Advanced > Security Configuration > Override Default Fragment Setting 中增加大小，保存配置并部署策略。然后，监控 `show fragment` 命令输出中的队列计数器和系统日志消息 FTD-3-209006 的出现情况。

ACI 问题

由于 ACI Pod 中的活动 L4 校验和验证，导致间歇性连接问题

症状

- 通过ACI Pod中部署的ASA/FTD集群出现间歇性连接问题。
- 如果群集中只有1台设备，则不会发现连接问题。
- 从一个集群设备发送到集群中的一个或多个其他设备的数据包在目标设备的FXOS和数据平面捕获中不可见。



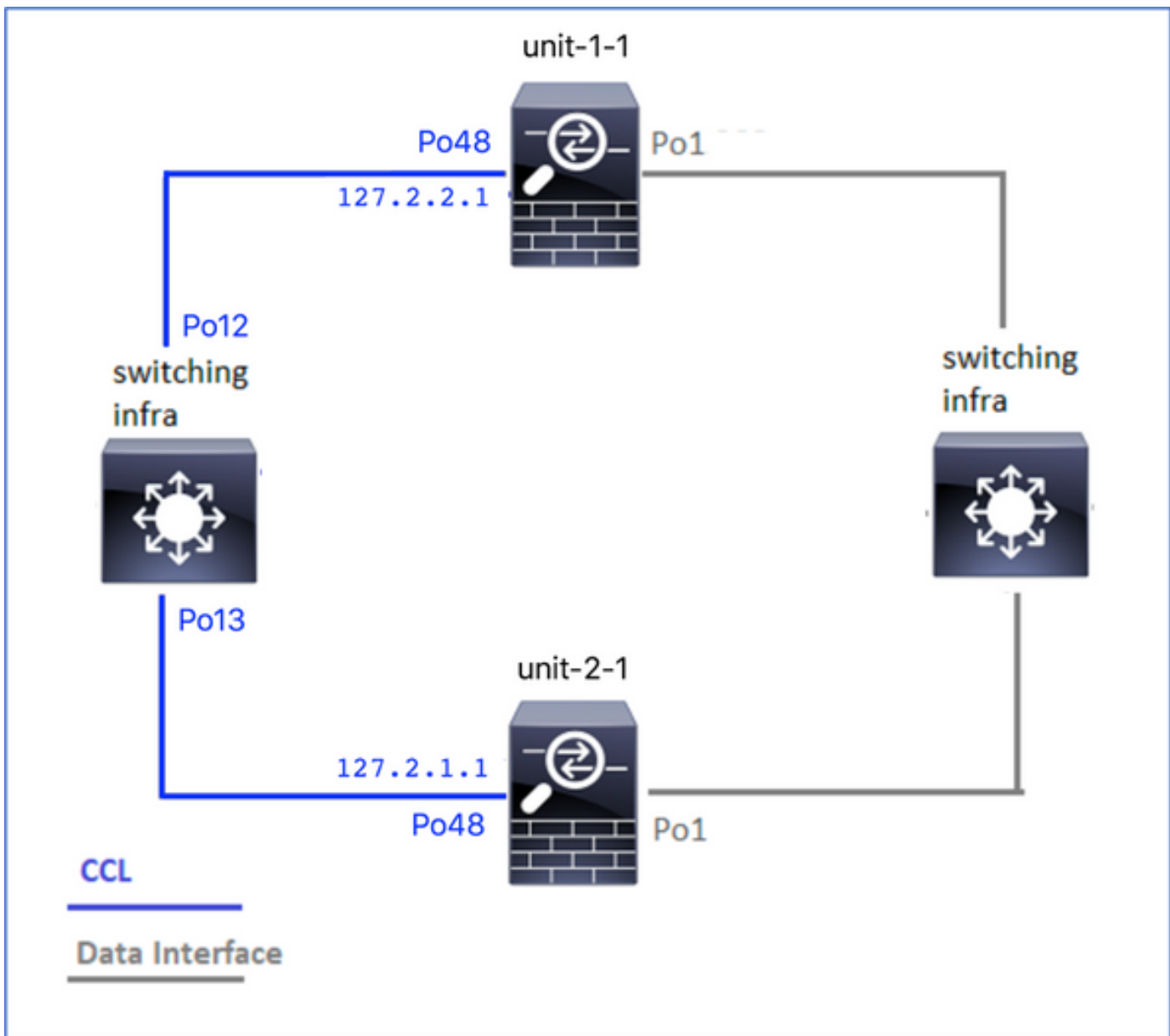
缓解

- 通过集群控制链路重定向的流量没有正确的L4校验和，这是预期行为。集群控制链路路径上的交换机不能验证L4校验和。验证L4校验和的交换机可能导致流量丢弃。检查ACI交换矩阵交换机配置，确保没有通过集群控制链路对接收或发送的数据包执行L4校验和。

集群控制平面问题

设备无法加入集群

CCL上的MTU大小



症状

设备无法加入集群，并显示以下消息：

```
The SECONDARY has left the cluster because application configuration sync is timed out on this unit. Di
Cluster disable is performing cleanup..done.
Unit unit-2-1 is quitting due to system failure for 1 time(s) (last failure is SECONDARY application co
All data interfaces have been shutdown due to clustering being disabled. To recover either enable clust
```

验证/缓解

- 在FTD上使用show interface命令，验证集群控制链路接口上的MTU至少比数据接口MTU高100字节：

<#root>

```
firepower#  
show interface  
  
Interface  
Port-channel1  
"  
Inside  
", is up, line protocol is up  
Hardware is EtherSVI, BW 40000 Mbps, DLY 10 usec  
MAC address 3890.a5f1.aa5e,  
MTU 9084
```

```
Interface  
Port-channel48  
"  
cluster  
", is up, line protocol is up  
Hardware is EtherSVI, BW 40000 Mbps, DLY 10 usec  
Description: Clustering Interface  
MAC address 0015.c500.028f,  
MTU 9184  
  
IP address 127.2.2.1, subnet mask 255.255.0.
```

- 使用size选项通过CCL执行ping操作，以验证CCL MTU上的配置是否已在路径中的所有设备上正确配置。

```
<#root>
```

```
firepower#  
ping 127.2.1.1 size 9184
```

- 在交换机上使用show interface命令检验MTU配置

```
<#root>
```

```
Switch#  
show interface  
  
port-channel12
```



```
is up
admin state is up,
  Hardware: Port-Channel, address: 7069.5a3a.7976 (bia 7069.5a3a.7976)
```

MTU 9084

bytes, BW 40000000 Kbit , DLY 10 usec

port-channel13

```
is up
admin state is up,
  Hardware: Port-Channel, address: 7069.5a3a.7967 (bia 7069.5a3a.7967)
```

MTU 9084

bytes, BW 40000000 Kbit , DLY 10 use

集群设备之间的接口不匹配

症状

设备无法加入集群，并显示以下消息：

```
Interface mismatch between cluster primary and joining unit unit-2-1. unit-2-1 aborting cluster join.
Cluster disable is performing cleanup..done.
Unit unit-2-1 is quitting due to system failure for 1 time(s) (last failure is Internal clustering error)
All data interfaces have been shutdown due to clustering being disabled. To recover either enable clustering
```

验证/缓解

登录到每个机箱上的FCM GUI，导航到Interfaces选项卡，并验证所有集群成员是否具有相同的接口配置：

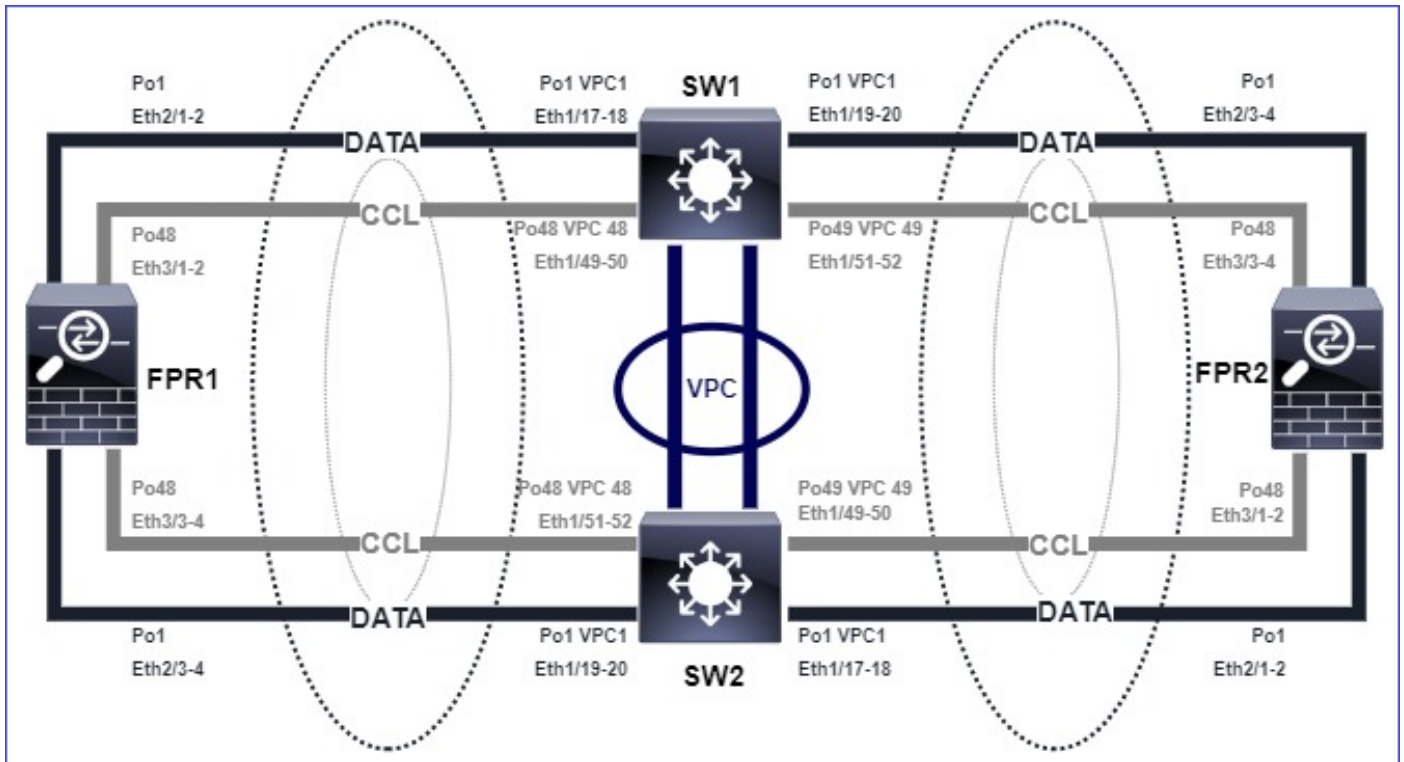
- 分配给逻辑设备的接口
- 接口的管理速度
- 接口的管理双工
- 接口状态

数据/端口通道接口问题

由于CCL上的可达性问题导致大脑分裂

症状

群集中有多个控制单元。请思考以下拓扑：



机箱1:

<#root>

```
firepower# show cluster info
```

```
Cluster ftd_cluster1: On
Interface mode: spanned
```

```
This is "unit-1-1" in state PRIMARY
```

```
ID : 0
Site ID : 1
Version : 9.15(1)
Serial No.: FLM2103TU5H
CCL IP : 127.2.1.1
CCL MAC : 0015.c500.018f
Last join : 07:30:25 UTC Dec 14 2020
Last leave: N/A
Other members in the cluster:
Unit "unit-1-2" in state SECONDARY
ID : 1
Site ID : 1
Version : 9.15(1)
Serial No.: FLM2103TU4D
CCL IP : 127.2.1.2
CCL MAC : 0015.c500.019f
Last join : 07:30:26 UTC Dec 14 2020
Last leave: N/A
Unit "unit-1-3" in state SECONDARY
ID : 3
Site ID : 1
Version : 9.15(1)
```

Serial No.: FLM2102THJT
CCL IP : 127.2.1.3
CCL MAC : 0015.c500.016f
Last join : 07:31:49 UTC Dec 14 2020
Last leave: N/A

机箱2:

<#root>

```
firepower# show cluster info
```

```
Cluster ftd_cluster1: On  
Interface mode: spanned
```

```
This is "unit-2-1" in state PRIMARY
```

```
ID : 4  
Site ID : 1  
Version : 9.15(1)  
Serial No.: FLM2103TUN1  
CCL IP : 127.2.2.1  
CCL MAC : 0015.c500.028f  
Last join : 11:21:56 UTC Dec 23 2020  
Last leave: 11:18:51 UTC Dec 23 2020  
Other members in the cluster:  
Unit "unit-2-2" in state SECONDARY  
ID : 2  
Site ID : 1  
Version : 9.15(1)  
Serial No.: FLM2102THR9  
CCL IP : 127.2.2.2  
CCL MAC : 0015.c500.029f  
Last join : 11:18:58 UTC Dec 23 2020  
Last leave: 22:28:01 UTC Dec 22 2020  
Unit "unit-2-3" in state SECONDARY  
ID : 5  
Site ID : 1  
Version : 9.15(1)  
Serial No.: FLM2103TUML  
CCL IP : 127.2.2.3  
CCL MAC : 0015.c500.026f  
Last join : 11:20:26 UTC Dec 23 2020  
Last leave: 22:28:00 UTC Dec 22 2020
```

确认

- 使用ping命令验证控制单元的集群控制链路(CCL)IP地址之间的连接：

<#root>

```
firepower# ping 127.2.1.1
```

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 127.2.1.1, timeout is 2 seconds:

?????

Success rate is 0 percent (0/5)

- 检查ARP表：

```
<#root>
```

```
firepower# show arp
```

```
cluster 127.2.2.3 0015.c500.026f 1
```

```
cluster 127.2.2.2 0015.c500.029f 1
```

- 在控制单元中，配置并检查CCL接口上的捕获：

```
<#root>
```

```
firepower# capture capccl interface cluster
```

```
firepower# show capture capccl | i 127.2.1.1
```

```
2: 12:10:57.652310 arp who-has 127.2.1.1 tell 127.2.2.1
41: 12:11:02.652859 arp who-has 127.2.1.1 tell 127.2.2.1
74: 12:11:07.653439 arp who-has 127.2.1.1 tell 127.2.2.1
97: 12:11:12.654018 arp who-has 127.2.1.1 tell 127.2.2.1
126: 12:11:17.654568 arp who-has 127.2.1.1 tell 127.2.2.1
151: 12:11:22.655148 arp who-has 127.2.1.1 tell 127.2.2.1
174: 12:11:27.655697 arp who-has 127.2.1.1 tell 127.2.2.1
```

缓解

- 确保CCL端口通道接口连接到交换机上的独立端口通道接口。
- 在Nexus交换机上使用虚拟端口通道(vPC)时，请确保CCL端口通道接口连接到不同的vPC，并且vPC配置不具有失败的一致性状态。
- 确保CCL端口通道接口位于同一广播域中，并且已在接口上创建并允许CCL VLAN。

以下是交换机配置示例：

```
<#root>
```

```
Nexus#
```

```
show run int po48-49
```

```
interface port-channel48
description FPR1
```

```
switchport access vlan 48
```

```
vpc 48
```

```
interface port-channel49
description FPR2
```

```
switchport access vlan 48
```

```
vpc 49
```

```
Nexus#
```

```
show vlan id 48
```

```
VLAN Name Status Ports
```

```
-----
48 CCL active Po48, Po49, Po100, Eth1/53, Eth1/54
```

```
VLAN Type Vlan-mode
```

```
-----
48 enet CE
```

```
1 Po1 up success success 10,20
```

```
48 Po48 up success success 48
```

```
49 Po49 up success success 48
```

```
<#root>
```

```
Nexus1#
```

```
show vpc brief
```

```
Legend:
```

(*) - local vPC is down, forwarding via vPC peer-link

vPC domain id : 1
Peer status : peer adjacency formed ok
vPC keep-alive status : peer is alive
Configuration consistency status : success

Per-vlan consistency status : success

Type-2 consistency status : success

vPC role : primary
Number of vPCs configured : 3
Peer Gateway : Disabled
Dual-active excluded VLANs : -
Graceful Consistency Check : Enabled
Auto-recovery status : Disabled
Delay-restore status : Timer is off.(timeout = 30s)
Delay-restore SVI status : Timer is off.(timeout = 10s)

vPC Peer-link status

id Port Status Active vlans

1 Po100 up 1,10,20,48-49,148

vPC status

id Port Status Consistency Reason Active vlans

1 Po1 up success success 10,20

48 Po48 up success success 48

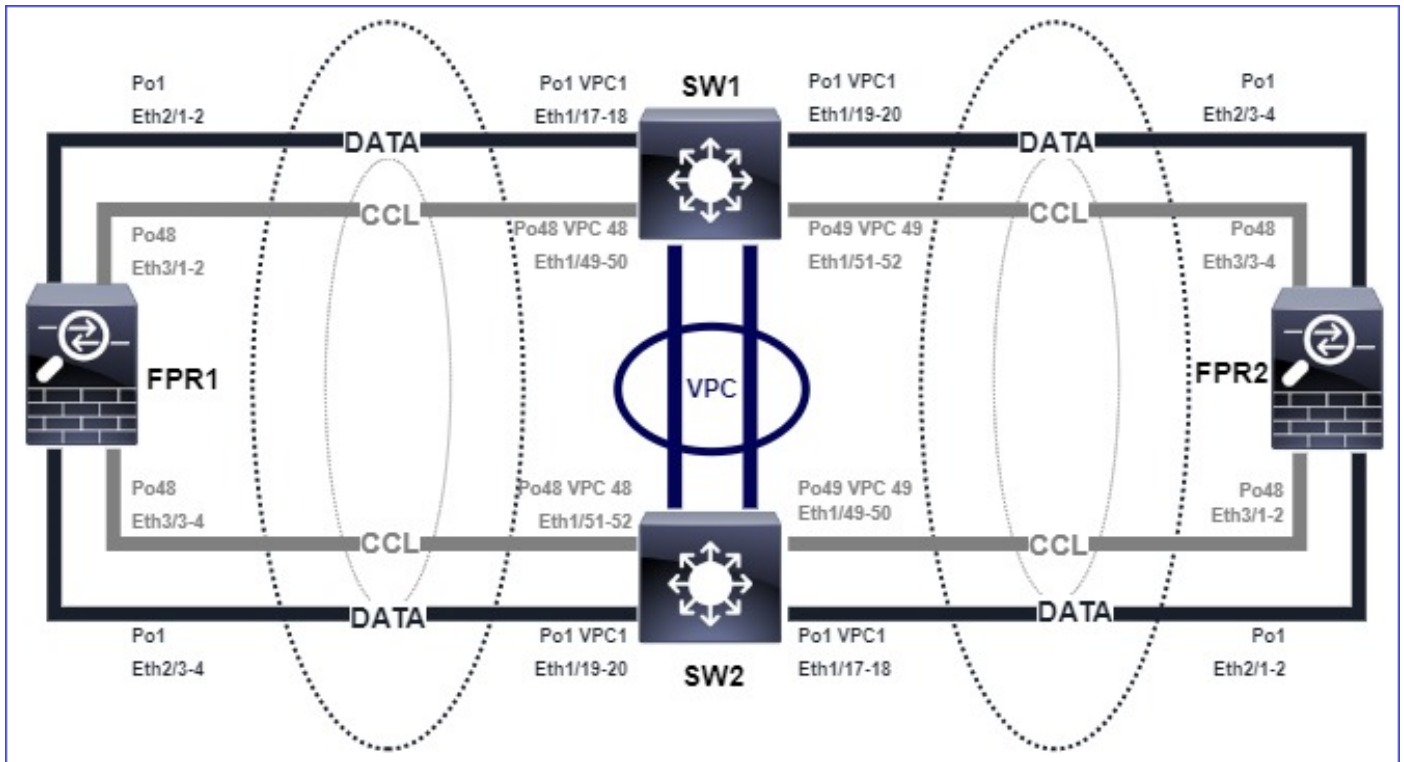
49 Po49 up success success 48

由于暂停的数据端口通道接口而禁用集群

症状

一个或多个数据端口通道接口被挂起。当管理性启用数据接口挂起时，由于接口运行状况检查失败，同一机箱中的所有集群设备都会被从集群中踢出。

请思考以下拓扑：



确认

- 检查控制单元控制台：

<#root>

firepower#

Beginning configuration replication to

SECONDARY unit-2-2

End Configuration Replication to SECONDARY.

Asking SECONDARY unit

unit-2-2

to quit because it

failed interface health

check 4 times (last failure on

Port-channel1

). Clustering must be manually enabled on the unit to rejoin.

- 检查受影响设备中的show cluster history和show cluster info trace module hc命令的输出：

<#root>

firepower# Unit is kicked out from cluster because of interface health check failure.

Cluster disable is performing cleanup..done.

All data interfaces have been shutdown due to clustering being disabled. To recover either enable clust

Cluster unit unit-2-1 transitioned from SECONDARY to DISABLED

firepower#

```
show cluster history
```

```
=====
From State To State Reason
=====
```

12:59:37 UTC Dec 23 2020

ONCALL SECONDARY_COLD Received cluster control message

12:59:37 UTC Dec 23 2020

SECONDARY_COLD SECONDARY_APP_SYNC Client progression done

13:00:23 UTC Dec 23 2020

SECONDARY_APP_SYNC SECONDARY_CONFIG SECONDARY application configuration sync done

13:00:35 UTC Dec 23 2020

SECONDARY_CONFIG SECONDARY_FILESYS Configuration replication finished

13:00:36 UTC Dec 23 2020

SECONDARY_FILESYS SECONDARY_BULK_SYNC Client progression done

13:01:35 UTC Dec 23 2020

SECONDARY_BULK_SYNC DISABLED Received control message DISABLE (interface health check failure)

<#root>

firepower#

```
show cluster info trace module hc
```

Dec 23 13:01:36.636 [INFO]cluster_fsm_clear_np_flows: The clustering re-enable timer is started to expi

Dec 23 13:01:32.115 [INFO]cluster_fsm_disable: The clustering re-enable timer is stopped.

Dec 23 13:01:32.115 [INFO]Interface Port-channel1 is down

- 在fxos命令外壳中检查show port-channel summary命令的输出：

<#root>

FPR2(fxos)#

```
show port-channel summary
```


Flags: D - Down P - Up in port-channel (members)
I - Individual H - Hot-standby (LACP only)
s - Suspended r - Module-removed
S - Switched R - Routed
U - Up (port-channel)
M - Not in use. Min-links not met

Group Port-Channel Type Protocol Member Ports

1 Po1(SD) Eth LACP Eth2/1(s) Eth2/2(s) Eth2/3(s) Eth2/4(s)

48 Po48(SU) Eth LACP Eth3/1(P) Eth3/2(P) Eth3/3(P) Eth3/4(P)

缓解

- 确保所有机箱具有相同的集群组名称和密码。
- 确保端口通道接口在所有机箱和交换机中具有相同的双工/速度配置且以管理方式启用的物理成员接口。
- 在站点内集群中，确保所有机箱中的相同数据端口通道接口连接到交换机上的相同端口通道接口。
- 在Nexus交换机中使用虚拟端口通道(vPC)时，请确保vPC配置没有失败的一致性状态。
- 在站点内集群中，确保所有机箱中的相同数据端口通道接口连接到同一vPC。

集群稳定性问题

FXOS回溯

症状

设备离开集群。

验证/缓解

- 使用show cluster history命令查看设备何时离开群集

```
<#root>
```

```
firepower#
```

```
show cluster history
```

- 使用以下命令检查FXOS是否具有回溯

```
<#root>
```

```
FPR4150#
```

```
connect local-mgmt
```

```
FPR4150 (local-mgmt)#
```

```
dir cores
```

- 收集设备离开集群后生成的核心文件，并将其提供给TAC。

磁盘已满

如果集群设备的/ngfw分区中的磁盘利用率达到94%，则设备会退出集群。每3秒进行一次磁盘利用率检查：

```
<#root>
```

```
> show disk
```

```
Filesystem Size Used Avail Use% Mounted on
rootfs 81G 421M 80G 1% /
devtmpfs 81G 1.9G 79G 3% /dev
tmpfs 94G 1.8M 94G 1% /run
tmpfs 94G 2.2M 94G 1% /var/volatile
/dev/sda1 1.5G 156M 1.4G 11% /mnt/boot
/dev/sda2 978M 28M 900M 3% /opt/cisco/config
/dev/sda3 4.6G 88M 4.2G 3% /opt/cisco/platform/logs
/dev/sda5 50G 52M 47G 1% /var/data/cores
/dev/sda6 191G 191G 13M
```

```
100% /ngfw
```

```
cgroup_root 94G 0 94G 0% /dev/cgroups
```

在这种情况下，show cluster history输出显示：

```
<#root>
```

```
15:36:10 UTC May 19 2021
```

```
PRIMARY Event: Primary unit unit-1-1 is quitting
                due to
```

```
diskstatus
```

```
Application health check failure, and
                primary's application state is down
```

或

14:07:26 CEST May 18 2021

SECONDARY DISABLED Received control message DISABLE (application health check failure)

另一种验证故障的方法是：

```
<#root>
```

```
firepower#
```

```
show cluster info health
```

```
Member ID to name mapping:
```

```
0 - unit-1-1(myself) 1 - unit-2-1
```

```
          0  1
Port-channel48 up up
Ethernet1/1 up up
Port-channel12 up up
Port-channel13 up up
```

```
Unit overall          healthy healthy
```

```
Service health status:
```

```
          0      1
```

```
diskstatus (monitor on) down down
```

```
snort (monitor on)      up      up
```

```
Cluster overall        healthy
```

此外，如果磁盘大约是100%，设备可能难以重新加入集群，直到释放了一些磁盘空间。

溢出保护

每个集群单元每5分钟检查一次本地单元和对等单元的CPU和内存利用率。如果利用率高于系统阈值（LINA CPU 50%或LINA内存59%），信息性消息显示如下：

- 系统日志(FTD-6-748008)
- 文件log/cluster_trace.log，例如：

```
<#root>
```

```
firepower#
```

```
more log/cluster_trace.log | i CPU
```

```
May 20 16:18:06.614 [INFO][
```

```
CPU load 87%
```

```
| memory load 37%] of module 1 in chassis 1 (unit-1-1) exceeds overflow protection threshold [
```

]. System may be oversubscribed on member failure.

May 20 16:18:06.614 [INFO][CPU load 87% | memory load 37%] of chassis 1 exceeds overflow protection thr

May 20 16:23:06.644 [INFO][CPU load 84% | memory load 35%] of module 1 in chassis 1 (unit-1-1) exceeds

该消息表示在设备发生故障时，其他设备资源可以超订用。

简化模式


6.3之前FMC版本的行为

- 您可以在FMC上单独注册每个集群节点。
- 然后在FMC中形成逻辑集群。
- 对于每个新群集节点添加，必须手动注册该节点。

6.3后FMC

- 通过简化模式功能，只需一个步骤即可在FMC上注册整个集群（只需注册集群的任何一个节点）。

支持的最低管理器数	受管设备	需要支持的最低受管设备版本	备注
FMC 6.3	仅FP9300和FP4100上的FTD集群	6.2.0	这仅是FMC功能

 **警告：**在FTD上形成集群后，您需要等待自动注册启动。您不能尝试手动注册集群节点（添加设备），但应使用Reconcile选项。

症状

节点注册失败

- 如果控制节点注册因任何原因失败，则集群将从FMC中删除。

缓解

如果数据节点注册因任何原因而失败，则有2个选项：

1. 每次部署到集群时，FMC都会检查是否有需要注册的集群节点，然后启动这些节点的自动注册。
2. 在集群摘要选项卡(Devices > Device Management > Cluster tab > View Cluster Status[链接](#))下提供Reconcile选项。触发协调操作后，FMC开始自动注册需要注册的节点。

相关信息

- [面向Firepower威胁防御的集群](#)
- [Firepower 4100/9300机箱的ASA集群](#)
- [关于Firepower 4100/9300机箱上的集群](#)
- [Firepower NGFW集群深入探讨 — BRKSEC-3032](#)
- [分析 Firepower 防火墙捕获以有效排除网络问题](#)

关于此翻译

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

请注意：即使是最好的机器翻译，其准确度也不及专业翻译人员的水平。

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